2-6-2015 12:00 AM

An Investigation of Cognitive Implications in the Design of Computer Games

Robert Haworth, *The University of Western Ontario*

Supervisor: Dr. Kamran Sedig, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Computer Science

An Investigation of Cognitive Implications in the Design of Computer Games

(Thesis format: Integrated Article)

by

Robert Haworth

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

Computer games have been touted for their ability to engage players in cognitive activities (e.g., decision making, learning, planning, problem solving). By 'computer game' we mean any game that uses computational technology as its platform, regardless of the actual hardware or software; games on personal computers, tablets, game consoles, cellphones, or specialized equipment can all be called computer games. However, there remains much uncertainty regarding how to design computer games so that they support, facilitate, and promote the reflective, effortful, and conscious performance of cognitive activities. The goal of this dissertation is to relieve some of this uncertainty, so that the design of such computer games can become more systematic and less ad hoc. By understanding how different components of a computer game influence the resulting cognitive system, we can more consciously and systematically design computer games for the desired cognitive support. This dissertation synthesizes concepts from cognitive science, information science, learning science, human-computer interaction, and game design to create a conceptual design framework. The framework particularly focuses on the design of: gameplay, the player-game integrated cognitive system, the interaction that mediates gameplay and the cognitive system, and the components of this interaction. Furthermore, this dissertation also includes a process by which researchers can explore the relationship between components of a computer game and the resulting cognitive system in a consistent, controlled, and precise manner. Using this process, three separate studies were conducted to provide empirical support for different aspects of the framework; these studies investigated how the design of rules, visual interface, and the core mechanic influence the resulting cognitive system. Overall then, the conceptual framework and three empirical studies presented in this dissertation provide designers with a greater understanding of how to systematically design computer games to provide the desired support for any cognitive activity.

Keywords

Games; Computer Games; Digital Games; Electronic Games; Video Games; Educational Games; Serious Games; Gameplay; Cognitive Gameplay; Complex Cognition; Distributed

Cognition; Human-Computer Interaction; Interaction Design; Game Design; Game Evaluation; Design Framework

Co-Authorship Statement

Chapter 1 was written by me, to introduce the dissertation and explain the connections between chapters. **Chapters 3, 4, 7 and 8** were a collaborative effort with my supervisor, Kamran Sedig. **Chapters 2 and 5** were also a collaborative effort with my supervisor, but included the help of another graduate student, Paul Parsons, working in the area of interaction design. **Chapter 6**, which focuses on the design of a particular game, was another collaborative work with my supervisor, but it also included the help of an undergraduate student, Michael Corridore, who worked on the implementation of the game. **Chapter 9** was again written by me, to summarize the dissertation and outline future areas of research.

Since the framework presented in this dissertation is a result of heavy collaboration, it may be difficult to identify which contributions can be attributed to me alone. However, some general comments can still be made. In synthesizing material, my contributions were associated with harmonizing the material, terminology, and attitudes of the game community (both researchers and practitioners) with those of other research communities. In theorizing material, my contributions were associated with ensuring design concepts and patterns applied effectively to the context of games. In testing material, my contributions were associated with the design and some implementation of the games. Such comments may not assist in identifying any particular concept or phrase as mine. Rather, it is intended to highlight the point that, throughout all of the collaboration and synthesizing, my main contribution has been to ensure that this framework includes concepts from game research and is applicable to game design.

Acknowledgments

First, I would like to thank my supervisor, Dr. Kamran Sedig. Our regular conversations and collaborations have been productive, insightful, and supportive throughout the time spent developing this dissertation. These discussions will continue to be a part of my growth long after my PhD is complete. Likewise, I would like to thank my examiners for their time and effort reviewing this dissertation and for their comments and constructive criticism.

I would also like to thank those graduate students who have provided me with help. Foremost among these students is (now Dr.) Paul Parsons, whose many comments, questions, and general camaraderie has been a boon to my writing, understanding, and well-being.

Last, but not least, I would like to thank my parents and my sister for their regular and continual love and support over the many years of my education, as well as my wife Gillian for her patience and love both before and after our wedding.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Generally speaking, the current literature on game design can be divided into two groups. The first consists of books, documents, and articles from those in the game industry (e.g., Adams, 2010; Crawford, 2003; Rouse, 2005; Novak, 2007). The second consists of materials from academics, such as journal articles (e.g., Barr et al., 2007; Bedwell et al., 2012), books (e.g., Koster, 2004; Salen & Zimmerman, 2004), and most frequently conference papers (e.g., Aarseth, 2003; Cox et al, 2012; Hunicke et al., 2004). Historically, the communication between these two groups has been problematic (Turner & Browning, 2010). As a result, the theoretical and practical aspects of game design have often been disconnected; game developers (i.e., practitioners) may not consider relevant research, and it may be unclear how studies and theories developed by researchers can be used by developers in creating better games.

Some researchers are focused on games for particular purposes, such as games for education, health, and political or social causes (e.g., Aleven et al., 2010; Burke et al., 2009; Dondlinger, 2007; Gros, 2007). Games can also be designed to engage the player in particular high-level cognitive activities (e.g., decision making, learning, problem solving). However, there is limited research into the design of such games. Some game researchers ignore the cognitive aspect of the player's experience entirely (e.g., Cowley et al., 2008). Other researchers focus on particular cognitive activities (e.g., when players engage in learning; see Aleven et al., 2010), and their research may also be in the context of design but only for those activities. Other researchers study cognition in general but not in relation to game design (e.g., Connolly et al., 2012; Quiroga et al., 2011). Therefore, although research is currently being conducted on game design and cognition in games, we currently lack a systematic understanding of how to influence player cognition in game design. Hence, we need a conceptual framework that can provide a common foundation for game design and cognition. This framework would be useful for analysis, design, and experimental study but would have to be general enough to be useful irrespective of the particular context of application. In other words, the framework could be used for particular cognitive activities while allowing the same concepts to be used in other contexts. For example, this framework could be used to assist the design of games for learning while being just as applicable for games that are intended for planning or decision making.

As all games engage the player in some degree of cognition, designing a game so that a particular activity occurs is not a major problem. The main problem that this dissertation attempts to solve is identifying some decisions that designers should consider in order to engage the player in particular *depths and degrees of* cognition. For instance, “what are some considerations that designers should have to create a game intended for the conscious, effortful, and reflective performance of problem solving?” The answer to that question is the focus of this dissertation and not “what does a designer need to do to include problem solving in a game?”

The solution presented is a conceptual framework with several components, each of which is explained in particular chapters. The framework is a product of synthesizing research and concepts from multiple fields, most notably: cognitive science, human-computer interaction, information visualization, learning sciences and education, game studies, and general game design. Despite such synthesizing we approach this framework from a predominantly computer science perspective, focusing specifically on the design of the user interface. For example, we discussed the interaction afforded by the user interface (a traditional computer science focus) rather than action possibilities or ‘mechanics’ within particular game contexts (as is commonly done in game studies). As another example, we abstracted away from particular content (again, a common computer science approach) to focus on the representation and manipulation of that content (unlike educational game research). It is not our intension to neither argue in favor of this perspective nor argue against other perspectives; alternative frameworks and perspectives can easily complement our own framework. However, there are no frameworks that unite the disciplines mentioned above in the context of user interface design. In other words, there are no frameworks to help us understand how to design a computer game’s user interface so as to influence the depth of the player’s cognition; ours is able to do this.

1.1 Structure of Our Framework

Our research group has been developing a number of conceptual frameworks for user interface design in the context of computerized cognitive tools. The framework presented in this dissertation is a specialized and expanded version of this more general research. It is specialized because previous concepts have been reworked and focused towards the context of computer games. It is expanded because new concepts have been added that only arose out of designing and analyzing computer games. Defining the structure of this framework is difficult because it is not wholly separate or distinct from other ongoing research. For the framework presented in this dissertation though, several components can be identified. A very brief description of each component is provided below. More detailed discussions occur in the various chapters of this dissertation.

The **User Interface** refers the part of the game through which the player interacts. It is composed of **Visual Representations** (VRs) of the game's information or content, and it is upon these VRs which the player acts. Although the design of VRs and their perceptual and cognitive effects are relevant for this dissertation, no detailed discussion of these topics has been included. Brief discussions are available in Chapters 2 and 5, but a more detailed and elaborate discussion can be found in Parsons and Sedig (2014a).

The **Core Mechanic** is the set of essential interactions that occur between the player and the game. By interaction is meant the player acting on a VR and the game reacting, resulting in a change of the VRs of the user interface. The player experiences gameplay as a result of repeatedly performing these interactions.

Cognitive Gameplay is the cognitive component of the experience that emerges as the player interacts with the game (i.e., engages with the core mechanic). We indirectly design cognitive gameplay through directly designing the game (i.e., cognitive gameplay is second-order design). Since this dissertation is ultimately concerned with design decisions related to the depth and kind of cognition, we are actually concerned with the design of cognitive gameplay.

Common **Patterns of Interaction** can be identified, which enable more systematic design and analysis of the core mechanic. Although some of the patterns appropriate to games have been discussed in this thesis, a broader and more comprehensive list (including some patterns that do not make sense in the context of computer games) can be found in Sedig and Parsons (2013).

Interactivity is the quality of interaction, such that the interactivity of the core mechanic refers to the quality of the core mechanic (i.e., the quality of the essential interactions of the game). Designing interactivity involves designing the operational forms of specific structural elements of each essential interaction. Interactivity affects the resulting gameplay, such that deeper, more effortful, and more reflective cognitive gameplay occurs as a result of the design decisions related to interactivity.

The **Game Rules** provide an unambiguous description of the structure and functionality of the game. This includes the game's core mechanic, content (e.g., game objects), VRs, and interactivity. These rules can be written at different levels of abstraction, depending on the amount of detail desired. For example, the rules for the game of *Chess* include a description of the pieces and how these pieces can be moved on the board. However, the rules could be very concrete, such as stating that the pieces must be physical objects made of wood, or the rules could be very abstract, such as not even describing the visual form that a piece could take.

The **Solution Space** is the abstract space within which the solution to a problem is constructed. This space is shared across the computer game and the player: some portion of the space is represented on the user interface of the game, and another (potentially overlapping) portion is represented within the mind of the player. The solution space serves as an example of how cognition can be distributed across the player and the game, and how we can design the game to manipulate this distribution. Ultimately, its design is a more concrete example of what we mean by designing cognitive gameplay.

To test design choices, the framework includes a **Creative Design Process** by which we can systematically create multiple isomorphic variations of a computer game. By isomorphic is meant that the variations are structurally similar at some degree of abstraction. This allows us to isolate and test structural changes more rigorously, by being able to identify the points along which two games are similar.

Cognitive Toys are simple non-computerized activities that engage the mind to some degree, such as math and logic puzzles. These serve as a source of ideas for a computer game that could require some manner of cognitive engagement from the player, and act as the starting point for this framework's design process.

In addition to these components, the framework is founded upon **Distributed Cognition** and **General Systems Theory (GST)**. It is within the theory of distributed cognition that our understanding of cognitive gameplay is founded; the integrated and reciprocal cognitive system that is formed as the player interacts with the game is very much an explanation of cognition derived from the theory of distributed cognition. This dissertation does not include a detailed discussion of cognition or distributed cognition, but the limited discussions provided in almost every chapter are sufficient.

GST provides a simple, comprehensive, and consistent means by which systems can be described and analyzed. Since any physical or conceptual entity can be conceptualized as a system, GST is applicable to any component our framework. For example, a computer game can be analyzed as a system with various sub-systems and sub-sub-systems. The experience arising from the player interacting with the game can also be analyzed as a system, as can the cognition system or cognitive gameplay. However, we could also look at the user interface as a system, in which VRs are its components and each VR is also a sub-system. A detailed discussion of GST is not included in this dissertation, but a brief summary of it and some of the ways that we have used it are provided in Chapter 4. For more information about GST, see Skyttner (2005).

Lastly, this framework has undergone a series of **Empirical Evaluations**. The purpose of these evaluations was to show how we can evaluate the cognitive gameplay of a computer game. The design process was used to create several games, which were then used to investigate two components of the framework (interactivity and the solution space) and differences in cognitive gameplay were detected. Hence, these evaluations provided empirical support for the framework by suggesting that at least some of the design decisions are important for designing cognitive gameplay.

Figure 1 depicts the relationships between some of the components of the framework. The player acts upon a visual representation (VR_1), it is transformed into a new visual representation (VR_2) as part of the reaction, the player perceives and mentally processes this change, and then she acts again. This cycle is enabled by the core mechanic, and cognitive gameplay emerges out of it. Interaction design is involved in the creation of the core mechanic, and representation design is involved in the creation of the VRs on the user interface. Interactivity design encompasses interaction design, since it involves designing the micro-level structural components of specific interactions. However, interaction design, interactivity design, and representation design are all encompassed by the rules of game; the rules define all of these components, such that anyone designing the rules of a game is involved in the design of these other components. In addition, the rules define lower-level internal aspects of the game (e.g., scoring, mechanics, game component restrictions, etc.) which are not discussed in this dissertation.

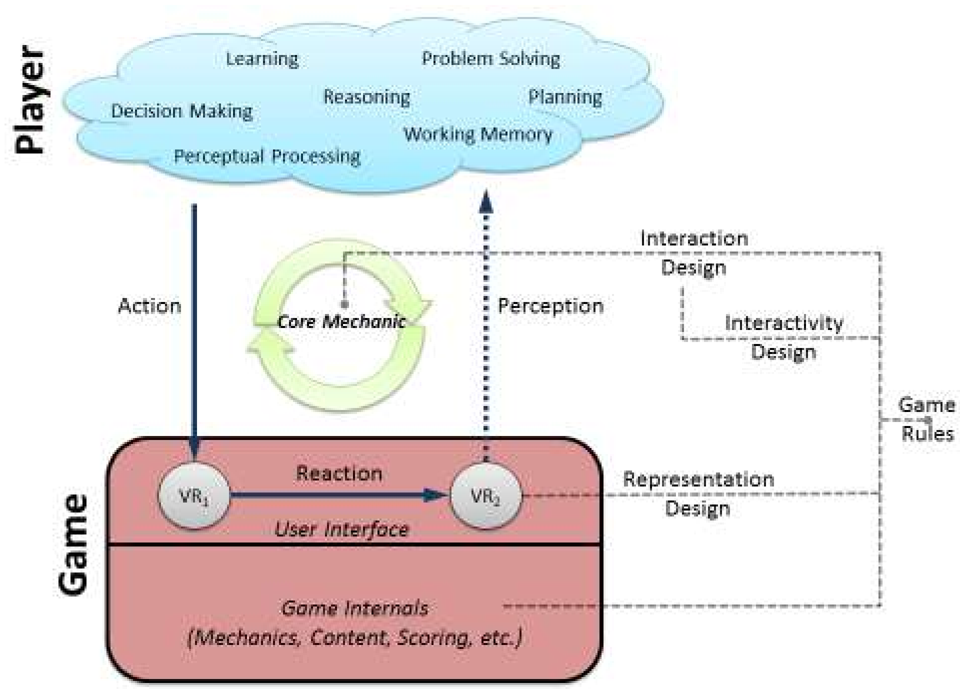

Figure 1-1: Some of the framework's components covered in this dissertation.

1.2 Scope of this Dissertation

This dissertation is only concerned with computer games, whereby "computer games" is meant games whose interaction is mediated by computational technology. For example, games on a personal computer, laptop, game console (e.g., XBox, Playstation), or cell phone would all be considered computer games. In contrast, non-computer games would include things such as board games, card games, and physical sports. Although it is possible for a game's rules to be sufficiently abstract that the rules could be implemented as a computer or non-computer game, our interest is in the computer game version. For instance, "the game of Chess" may mean a physical instantiation (i.e., board game) or computerized instantiation (i.e., computer game) of a set of rules, but we are only interested in the computerized version. In addition, the aspect of the computational technology with which we are interested is the conceptual- and software-level of the user interface. In other words, we are not interested in forms of input hardware (e.g., specialized input devices), mixed- or augmented-reality devices, or the lower-level details of a game (e.g., rules describing game objects, how they are interrelated, scoring, etc.).

Even though the framework in this dissertation is applicable to many different kinds of computer games, the examples and discussions are restricted to only single-player puzzle-like games. A game that involves only one human player has a simpler cognitive system than a game with multiple human players. Even though the concepts in this framework would still apply, we want to first establish the most basic case of a cognitive system with games: a system in which there is only one human player and one game. For a similar reason, the games in this dissertation do not involve computerized opponents (i.e., they are not multi-player games which involve only one human player and several artificial intelligence players). The main form of opposition the player faces is derived strictly from how the game components react to the player and the initial configuration of these components (typically what are called puzzle games). The games we use as examples are also simple in nature; they are not complex, multi-staged role-playing or strategy games. Again, the purpose behind such simplification is to establish and more easily explain the fundamental concepts of this framework. It is much easier to understand how to design the interactivity of the core mechanic of *Tetris* than *World of Warcraft*, even though the design concepts of this framework do apply to a game like *World of Warcraft*.

For this dissertation, we also did not include a deep discussion or investigation of human cognition. Even we emphasize cognitive gameplay and include studies that have been conducted to evaluate cognitive gameplay we do not go into detail about the different kinds or levels of cognitive processes. We also do not discuss cognition at the neurological or physiological level, even though it would be possible to evaluate cognitive gameplay using such means. Instead, we consider cognition in a less technical sense, as is common for research within the human-computer interaction discipline of computer science. More qualitative and subjective measures of cognition are used than quantitative ones, and the possibilities for greater specificity in cognitive processes serves as an area of future work.

1.3 Structure of this Dissertation

The rest of this dissertation is divided into eight chapters. Their contents and the relationships between these chapters are briefly described below.

In **Chapter 2**, we discuss the computer game components that are covered in our framework, how these components are conceptualized, and some brief thoughts on how they can be designed to influence the player's cognition.

In **Chapter 3**, we discuss interaction and show how computer games can be analyzed and categorized using their interaction patterns. In addition, we briefly discuss the cognitive activity most closely associated with particular interaction patterns.

In **Chapter 4**, we introduce the cognitive toy as a conceptual seed for designing computer games with a particular cognitive activity or goal. We also discuss the role played by rules in design, and how computer games can be designed using the lens of general systems theory. These concepts are then combined into a process for designing isomorphic computer games.

In **Chapter 5**, we discuss interactivity and the core mechanic. Twelve structural elements of interaction are identified that, collectively, affect interactivity and the resulting cognitive gameplay. Unlike in Chapter 3 where we discuss particular interaction patterns, in this chapter we focus on the manner in which any one of those interactions could be operationalized to better promote or support the desired degree of cognitive engagement.

In **Chapter 6**, we take the process introduced in Chapter 4 and use it for empirical testing of computer games. As an example, we present an investigation into how we could study the effect of minor structural differences in a game on player cognition.

In **Chapter 7**, we use the design process discussed in Chapter 4 to create isomorphic games that differ in the operationalization of one of the structural elements identified in Chapter 5. We then use the study methodology described in Chapter 6 to investigate differences between the cognitive gameplay of these games. While a method for empirically validating our framework is presented in Chapter 6, an actual example of such work is shown in this chapter. In addition, empirical support for the central goal of this framework—that certain design decisions actually impact the resulting cognitive engagement—is presented.

In **Chapter 8**, we present a different study, which investigated the representational component of the user interface. This chapter provides evidence that cognition can be influenced by more than just the game component studied in Chapter 7. This chapter also suggests that the study methodology presented in Chapter 6 can be used to empirically investigate and validate any part of the framework.

In **Chapter 9**, we draw some conclusions from the previous chapters, discuss how this dissertation contributes to the study of games, and identify some areas of future research.

Finally, readers should keep in mind that the chapters of this dissertation can be read sequentially or individually. Chapters 2, 3, 4, 6, and 7 have been published; Chapters 5 and 8 are in the midst of review. As such, chapters 2 to 8 are written as self-contained papers and could be read in isolation from the rest of the dissertation.

1.4 Brief Note on Terminology

The chapters of this dissertation were written at different times of my Ph.D. and reflect the process of developing the concepts involved as well as the terms that refer to those concepts. Since we occasionally used concepts that could be foreign to the target audience or had a significantly different definition than their familiar one, each chapter provides a definition of its main concepts. Even though some of the concepts were gradually developed and refined, their usage within a particular chapter should be clear from the chapter's background section. The differences between terms in the various chapters are minor, but there is one situation where terminological differences may result in some confusion: the games on which we focused our framework.

We are primarily interested in games that use computational technology as a platform, and we initially referred to these games as "digital games" and then later as "computer games." These terms and others, such as video game and electronic game, are used interchangeably in the game literature. It is often clear what kind of game is meant by these terms, but sometimes authors use terms in a very specific manner. For instance, to one person "video game" could mean a game played on a game console (e.g., Microsoft XBox, Sony Playstation) or a personal computer; to another person "video game" only means games played on a game console and the term "computer game" means games played on a personal computer. In other words, in one case video game included multiple hardware platforms while in the other case it only included one platform. We wanted a term that indicated we did not care about the hardware or software used in the platform, since the concepts we were investigating were abstract and could be implemented on any computational platform. Eventually, we settled on using "computer game," given the greater proliferation of that term, and included a full definition of what we meant by it to eliminate any possibility of confusion.

A bigger problem emerged when it came to cognition. Initially, we wanted to emphasize that our framework was intended for designers who wanted to consciously design games that encouraged, promoted, and supported a deeper and more reflective cognitive performance. For example, our framework was intended for games where the player must spend significant mental effort on the presented material in order to make decisions, and not merely to produce a game that required the player to make decisions. Since this emphasis was focused on the type of game, we wanted to differentiate these games from others. At first, we used the term “digital epistemic games,” and then changed to the more familiar-sounding “digital cognitive games.” However, this term was still a source of confusion for many readers and reviewers. In response, we changed our focus from the game to the gameplay (i.e., the experience resulting from playing the game). The term “cognitive gameplay” is now used; the goal of the framework is to assist the design, analysis, and evaluation of cognitive gameplay. Therefore, when reading the various chapters of this dissertation, it is necessary to realize that the same phenomenon has been intended all along but the terms used to identify and define it have changed. Again, if it is necessary to know the particular term and definition used within a chapter, such terms are explained within the background section of the chapter itself.

Chapter 2: Design of Digital Cognitive Games: Some Considerations

This chapter has been published as Haworth, R., Parsons, P., and Sedig, K. (2013). Design of digital cognitive games: Some considerations. *The International Journal of Cognitive Technology*, *18* (1), 22-27. It is reprinted here with permission.

Note that the format was changed to match the format of this dissertation, and the references were moved to the end of the dissertation. Figure numbers have also been changed to be relative to chapter numbers. For example, “Figure 2-1” is the first figure of Chapter 2 and the figure is labeled as such. However, this figure is referred to as “Figure 1” in the text of the chapter. The same is true for tables. In addition, when the phrase “this paper” is used, it refers to this chapter.

2.1 Introduction

There is a growing body of research suggesting that playing digital games can enhance the performance of cognitive activities (Blumberg & Ismailer, 2009; Green et al., 2006; Sedig, 2007). Furthermore, digital games are increasingly being targeted towards purposes other than pure entertainment. In addition to their motivational properties, they are also being conceptualized as cognitive technologies (McDaniel & Vick, 2010). Already digital games are being used to facilitate such activities as making sense of climate change patterns, analyzing economic policies, learning about mathematical representations, reasoning with decision trees and complex structures, and exploring health issues (for some examples, see: Belman & Flanagan, 2010; Gros, 2007; Ke & Grabowski, 2007; Rowhani & Sedig, 2009; Haworth et al., 2010; Linehan et al., 2011). However, despite their potential for enhancing and developing cognitive activities, and despite their growing popularity, the design of digital games is often not well informed by human-computer interaction research, nor current research in the cognitive and learning sciences (Barr, Noble, & Biddle, 2007; Rambusch, 2010; Sedig, 2008). Studies in these areas indicate a relationship between design decisions and the performance of high-level cognitive activities such as learning and problem solving (e.g., Chmiel, 2010; Habgood & Ainsworth, 2011; Sedig, 2007, 2008; Sharritt, 2010; Svendsen, 1991). In addition, existing design frameworks tend to focus on general principles (e.g., see Dipietro et al., 2007; Wilson et al., 2009; Sedig & Haworth, 2012) which, while useful for design, do not allow for a systematic understanding of the relationship between design decisions and cognitive effects. Therefore, it is difficult to consistently design digital games to achieve their full potential for enhancing the performance of high-level cognitive activities (Haworth & Sedig, 2011). Rather than being concerned with all digital games, this paper is concerned with a particular category of digital games—digital cognitive games (DCGs)—whose primary function is to mediate (i.e., facilitate, develop, and/or promote) the controlled, reflective, effortful, and/or mindful performance of high-level cognitive activities. This paper combines and integrates research from game studies, cognitive and learning sciences, and human-computer interaction design to discuss some components of DCGs that influence cognitive processes, and are thus essential considerations for the design of effective DCGs.

The structure of the paper is as follows. Before discussing design considerations, some recent trends and advances in the learning and cognitive sciences, as well as the basic characteristics of DCGs, are discussed. Next, some important components of DCGs are discussed; these include game content, representations, interaction, core mechanic, and interactivity. Subsequent to this, a brief design scenario is presented. The final section summarizes the paper and provides suggestions for future research.

2.1.1 Learning and cognition

Towards the end of the twentieth century, epistemological shifts and technological advances stimulated researchers and educators to expand their conceptions of learning and learning environments (Land & Hannafin, 2000). Indeed, learning theories developed over the past two decades have asserted that higher-order thinking and the performance of high-level cognitive activities, such as problem solving and planning, are vital components of learning (Jonassen, 2011). For example, a learning activity may involve solving a problem, making sense of a body of information, planning some future actions, and/or making decisions; that is, it may involve any number of high-level cognitive activities. Accordingly, in this paper, learning is considered as a high-level cognitive activity that may have any number of other cognitive activities embedded within it.

Concurrent to epistemological shifts in the learning sciences, cognitive science researchers began to posit that cognitive processes are fundamentally influenced by one's surrounding environment. Evidence began to suggest that cognition not only is situated within social and contextual settings, but also is embodied and distributed across the brain and its external environment. For example, Kirsh and Maglio (1994) studied people playing *Tetris*, and discovered that cognitive processes during gameplay were extended into the external environment through the performance of epistemic actions—actions performed to facilitate mental operations rather than to achieve physical or pragmatic goals. Further research into human cognition has demonstrated that the external environment not only mediates and facilitates cognitive processes, but also is an integral component of what can be understood as an extended and distributed cognitive system (see Clark, 2008). When playing a game, cognitive processes are distributed across the player and the game. Cognitive activities emerge from the interactions among the components of the system—that is, the player and the game. Consequently, the unit of analysis of learning and other cognitive activities that games mediate *must be the player-game cognitive system*. That is, to examine if and how games support learning, the player and the game must be viewed as a distributed cognitive system, and not as isolated entities.

2.1.2 Digital cognitive games

Games have been defined in many different ways over the years. In this paper, a game is defined as a system in which players engage in artificial conflict, defined by rules, resulting in a quantifiable outcome (Salen & Zimmerman, 2004). A digital game, then, is a subset of general games which operates on electronic, computational devices or platforms. In other words, digital technology that is interactive necessarily mediates the play of a digital game, whereas such mediation is not a necessary characteristic of games in general.

Digital Cognitive Games (DCGs) are digital games that facilitate, support, and/or promote the performance of reflective, mindful, controlled, and/or effortful high-level cognitive activities. While playing a digital game, the player may be engaged in cognitive activities—such as problem solving or learning—using habitual, automatic, and/or trial-and-error mental strategies, but not necessarily thinking in a reflective or controlled manner. If, on the other hand, a digital game is intentionally designed such that the player is reflecting on his actions, thinking carefully about the task at hand, and/or engaging in deep mental processing of the information, then the game is considered a DCG.

When the primary cognitive activity of a DCG is learning, it can also be called a learning game, educational game, or serious game. However, DCG is a broader term since the primary cognitive activity may be something other than learning. For example, a digital game that engages the player in contemplation of social justice issues would be a serious game, but may not be a learning game or a DCG. A digital game in which the player learns and develops mathematical skills would be a learning game or an educational game, but it could also be a DCG. However, a digital game in which the player navigates through a maze—but must engage in reflective planning to do so—would be a DCG, but may not be considered a learning game or a serious game. This paper will focus on DCGs, as we are interested in the components of digital games that influence cognitive processes during the performance of high-level cognitive activities, irrespective of the content with which such processes are engaged. These same ideas can be applied to learning and educational games that are also DCGs, and thus will benefit researchers and designers of such games.

2.2 Design Considerations for DCGs

This section will briefly examine some components of DCGs that fundamentally influence cognitive processes, and thus affect learning outcomes and the performance of cognitive activities. Considering these components when designing DCGs can enable a systematic design process in which design decisions are based on a conscious understanding of their cognitive effects. These components include game content, representations, interaction, core mechanic, and interactivity.

2.2.1 Game content and representations

Historically, educational content of games has been overemphasized at the expense of other essential components, leading to many poorly designed games that did not achieve their intended learning outcomes (Habgood, 2007). While content is certainly important, research in cognitive science has demonstrated that the manner in which content is represented, rather than the content *per se*, significantly influences cognitive processes (e.g., see Zhang & Norman, 1994). In fact, since the only access the player has to content is through representations at the visually-perceptible interface of the game, from the player's perspective the *representation is the content* (Cole & Derry, 2005). Designers must consider not only the content that is being provided to the player, but also the manner in which the content is represented. Moreover, it is often the case that game content can be represented with different representational forms that are informationally equivalent but computationally non-equivalent. That is, although different representations may depict the same content, each may require differing amounts of cognitive effort for processing and interpretation. While performing cognitive activities, cognitive processes are distributed across mental representations of the player and visual representations of the DCG; as a result, designers must carefully consider representation design and how design decisions impact cognitive processes and activities (e.g., see Sharritt, 2010).

2.2.2 Interaction and the core mechanic

An essential characteristic of DCGs is the interaction that takes place between the player and the DCG. Interaction is often discussed in the context of high-level pedagogical considerations, such as whether the DCG promotes constructivist learning, goal-based learning, cognitive apprenticeship, and so on. In this paper, however, interaction refers to lower-level individual instances of action and reaction between the player and the DCG, such as a player performing an action and a shape rotating as a result. At this level, interaction design is concerned with low-level action-reaction considerations, such as whether and how a DCG allows the player to rearrange tiles, transform shapes, move through a game space, and assign behaviors and/or properties to game entities, and the cognitive effects of such interactions. Much research demonstrates that design considerations at this level of interaction have significant effects on cognitive processes (e.g., see Sedig & Parsons, 2013).

A DCG often includes interactions that are not essential to playing the game, such as pausing or saving the state of the game. Although such interactions may be performed, there is typically a core set of interactions that are essential for proper gameplay and are repeated throughout the game. This core set of interactions occurs again and again to form a cycle, and can be referred to as the core mechanic of a game (see Sicart, 2008). For example, in the game *Tetris*, the basic interactions are rotation and movement of a shape, and these are repeated over and over and form the core mechanic. In DCGs, the core mechanic is the cyclical pattern of interaction that binds the player and the game into an integrated cognitive system. Consequently, it is primarily through the core mechanic that the player accesses and engages with the content of the DCG. Moreover, it is primarily through the core mechanic that the player and the DCG are tightly coupled into an integrated cognitive system, resulting in the emergence of higher-level cognitive activities. The core mechanic can therefore be considered the *epistemic nucleus* of the DCG. Consequently, this component must be designed with a conscious awareness of how cognitive processes of the player are affected.

2.2.3 Interactivity

High-level cognitive activities, such as learning, emerge from the sustained interaction between the player and the DCG that is enabled by the core mechanic. As a result, the quality of this interaction is a critical determinant of the quality of the cognitive activities that emerge from the interaction. While interaction refers to action and reaction, interactivity, by adding the suffix 'ity', refers to the *quality of interaction* (Sedig & Liang, 2006). The authors are currently developing a framework that identifies and characterizes a number of elements that collectively give structure to each individual interaction in the context of DCGs. Each element has different operational forms, and varying the operationalization of these elements can significantly impact interactivity. As an individual interaction has both an action and a reaction component, the elements that affect interactivity can be categorized into action elements and reaction elements. To illustrate, let us examine one action element and one reaction element. One element of action that has been identified as affecting cognitive processes is *agency*, which is concerned with the metaphoric way through which the player expresses an action. Two operational forms of agency are: verbal and manual. If the agency of an action is verbal, the player expresses an action using his 'mouth', as though speaking to the DCG, such as by typing a command into a console. If the agency of an action is manual, the player expresses an action using his ‘hands’, as though reaching into the interface and grasping and manipulating representations, such as using a mouse cursor to click and rotate an object. A study by Svendsen (1991) demonstrated that the operational form of this action element significantly influenced cognitive processes during the performance of a problem solving activity. One reaction element that has been identified as affecting cognitive processes is *activation*, which is concerned with the commencement of reaction after the player has committed an action. Three operational forms of activation are: immediate, delayed, and on-demand. If activation is immediate, the reaction occurs instantaneously after an action is committed. If activation is delayed, an action is committed and then a span of time passes before the reaction occurs. If activation is on-demand, the reaction only occurs once the player requests it. This element of reaction can be operationalized to promote different degrees of mental effort and engagement with the underlying content. For example, if activation is delayed, the player may be forced to engage in deep, reflective thought before committing an action, since the feedback from the action (i.e., the reaction) is not immediate.

2.3 Design Scenario

This section will examine the design considerations of a particular DCG in order to demonstrate how the above components may be integrated and considered in the design process. The DCG in question requires the player to find a path through a maze that leads to an exit, while passing through several checkpoints along the way. Thus, to properly play the game, the player must identify possible paths and checkpoints, analyze them, make decisions about which path to take, and plan a course of action. The manner in which the components are designed affects how much cognitive effort is required to identify paths, how easily the player can assess the suitability of a path, how much the player is encouraged to reflect before making decisions, and so on. Each component and its different design options are discussed below. An exploratory study that was previously conducted with the DCG is also briefly mentioned to provide empirical support.

2.3.1 Content and representation

One crucial design decision is how the content of the game (e.g., the player, the maze space, paths through the maze space) should be represented. Three possibilities, each with different effects on cognitive processes, will be discussed. First, the paths could be represented in a grid-like fashion and the player's current position could be represented as an avatar, which is typical of many maze and puzzle games (Figure 1). This representation forces the player to exert cognitive effort to identify implicit, hidden paths, and to determine their suitability. A second option is to represent the paths explicitly with a tree diagram (Figure 2). Positions in the maze can be represented as nodes in the tree, and paths can be represented as links between the nodes. With this representation, the player is not required to expend much cognitive effort to identify paths, as they are explicit and visible, and can instead focus cognitive resources on analyzing the paths to determine which one is the best to follow. A third option is to use both representations simultaneously, so that the cognitive effects of each may be combined.

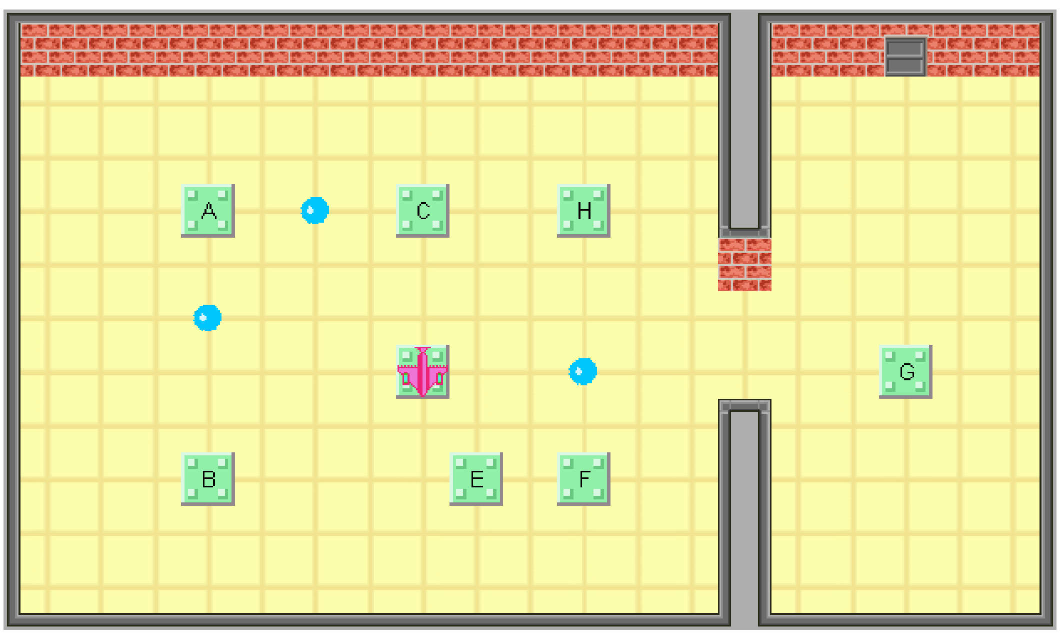

Figure 2-1: An implicit representation of paths through a maze.

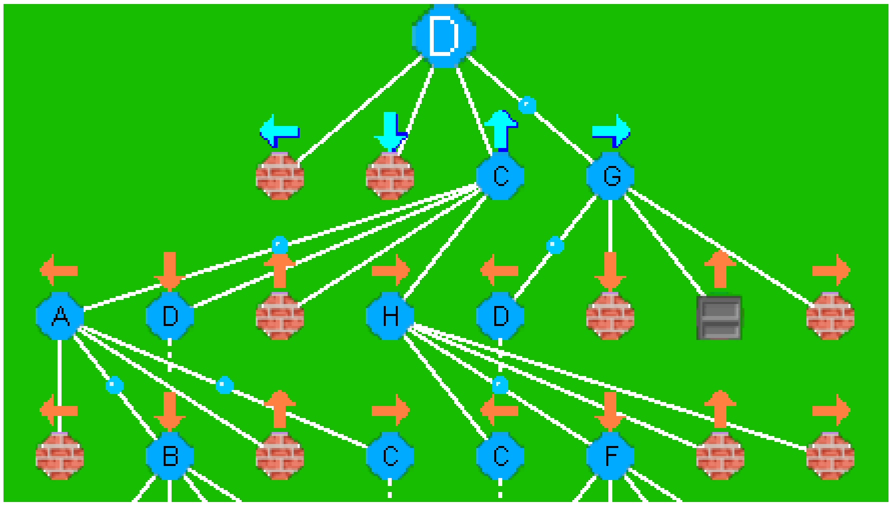

Figure 2-2: An explicit representation of paths through a maze.

2.3.2 Interaction and core mechanic

The main action that the player performs in this DCG is to move through a path in a maze. Thus, an important issue for designing interaction is to determine the way in which the player can follow a path through the maze, while integrating this interaction with a chosen representation for the paths. Assuming that the paths are represented as a tree diagram, interaction could be designed in the following manner. For the action component, the player selects one of the root node's immediate children as the next step in the path. For the reaction component, the tree changes such that the selected node becomes the tree's new root node. Both these two components are repeated in a cycle, until the player has passed through all checkpoints and has selected the goal node. The selecting of nodes to form the path through the maze would thus constitute the core mechanic of the DCG. Alternatively, use of the grid-like representation would alter the design of the core mechanic, and interaction design would be concerned with moving an avatar around the grid. For the action component, the player could assign a direction to the avatar. For the reaction component, the avatar could move in the assigned direction until it reached some point at which it would stop. The core mechanic is thus the cycle of assigning directions to the avatar, and it moving accordingly through the maze.

2.3.3 Interactivity

For interactivity design, each structural element of an interaction can be examined and operationalized according to the desired cognitive effect. To exemplify this in light of the DCG currently being examined, consider the interaction discussed in the previous section in which the player assigns a direction to an avatar. The two previously discussed interactivity elements, *agency* and *activation*, which are concerned with action and reaction respectively, will be discussed.

The *agency* element of the action could be operationalized such that the player is given four directional buttons to click (i.e., manual agency). Doing so would assign the button's corresponding direction to the avatar. Alternatively, the agency element could be operationalized such that the player types a command, such as 'north', to assign a direction (i.e., verbal agency). The activation *element* of the reaction could be operationalized such that each time the player assigns a direction to the avatar it immediately moves in the assigned direction (i.e., immediate activation). On the other hand, the player could queue a series of directions for the avatar, then at some point select a button to initiate the reaction such that the avatar would move in each of the queued directions in the order in which they were assigned (i.e., on-demand activation). As discussed in the interactivity section above, each of these would have different effects on the cognitive processes of the players.

2.3.4 Evaluation

An exploratory study was conducted using multiple versions of this DCG to determine whether different design decisions affected decision making (see Haworth, Tagh Bostani, & Sedig, 2010 for a detailed discussion of the study). In the study, four versions of the DCG were developed. In version one, both a grid and a tree diagram were used to represent the paths, and the player interacted with the DCG by selecting nodes in the tree. In version two, interaction was changed so that the player assigned a direction to the avatar. In version three, the player could interact with the DCG either by selecting nodes or assigning a direction. In version four, only a grid was used to represent the paths, and the player interacted with the DCG by assigning a direction to the avatar. The results of the study indicated that participants who played version one referred to the tree diagram to extract paths in difficult mazes, showed more awareness of the consequences of their decisions, and avoided choosing paths that would be detrimental to their progress. Participants who played versions two and three paid less attention to the tree diagram, and spent more effort extracting paths from the grid. Participants who played version four showed more difficulty in determining the correct path. The researchers concluded that the different ways of designing interaction and representation components of the DCG did have an effect on the way in which participants analyzed their possibilities and made decisions regarding the best path to take.

Although the results indicate that such design decisions do affect the cognitive processes of players, only two of the components discussed in this paper—interaction and representation—were studied. A brief discussion of this study has been included simply to provide additional empirical evidence that the consideration of at least these two components is necessary. Future studies can examine the other components and their relationships in more detail.

2.4 Summary

As recent theories of learning and instruction promote more situated and active learning strategies, DCGs have the potential to take a more important role in educational settings. To do so, however, their design must be well informed by relevant research. This paper has drawn from research in the cognitive and learning sciences, game studies, and human-computer interaction design, to examine some components that are important to consider in the design of DCGs. Representation design, interaction design, design of the core mechanic, and design for interactivity, have all been discussed in terms of their cognitive impacts. Future research in this area will hopefully elaborate on these components and integrate them into more comprehensive design frameworks. Although design of DCGs until now has typically been ad hoc and/or based on anecdotal evidence, the development of such research can assist designers in making systematic design decisions that are based on an awareness of their cognitive effects. As a consequence, DCGs can be consciously designed to engage particular cognitive processes and to achieve intended cognitive outcomes.

Chapter 3: A Strategy for Analyzing Digital Epistemic Games

This chapter has been published as Haworth, R., and Sedig, K. (2010). A Strategy for analyzing digital epistemic games. In Soria, A., & Maldonado, J. (Eds.), *Computer games: Learning objectives, cognitive performance, and effects on development* (pp. 53-84). New York, NY: Nova Science Publishing. It is reprinted here with permission from Nova Science Publishing, Inc.

Note that the format was changed to match the format of this dissertation, and the references were moved to the end of the dissertation. Figure numbers have also been changed to be relative to chapter numbers. For example, "Figure 3-1" is the first figure of Chapter 3 and the figure is labeled as such. However, this figure is referred to as "Figure 1" in the text of the chapter. The same is true for tables.

Note also that this chapter uses the term "Digital Epistemic Game," which is equivalent to the term "Digital Cognitive Game" used in chapters 2 and 4. Despite minor differences in definition, the concept of a digital cognitive game is the same between chapters 2, 3, and 4. See Section 1.2 of the introduction for the origin and transformation of this term.

3.1 Introduction

Digital games are now a noticeable part of popular culture and the digital game industry continues to grow, generating $10.5 billion USD in the United States alone last year (Siwek, 2010). Due to this popularity, there is much speculation about whether digital games have any learning potential (De Aguilera & Mendiz, 2003; Kafai, 2006; Prensky, 2000; Van Eck, 2006). A variety of games have been examined in the context of learning or used in different training and learning situations, such as history (Squire, 2006), literacy (Gee, 2007), spatial reasoning (Crawford, 2003), organizational and memory strategies (Jacobs & Dempsey, 1993), and mathematics and science (Kafai, 1995; Randel et al., 1992). However, it would be incorrect to say that all games provide learning potential (Van Eck, 2006). Thus, this chapter only focuses on digital epistemic games, which are computer-based games that mediate mental, intellectual, knowledge-oriented activities. To mediate here suggests that games support, facilitate, transform, and enhance. Given this definition, digital epistemic games can be used to improve players' problem solving, planning, information analysis, reasoning, critical thinking skills and other related cognitive skills and abilities.

Digital epistemic games are not a specific category or genre of games, and are not limited to any specific game-playing platform. To determine whether a game is considered epistemic or not, it is best to determine what cognitive benefits the game provides. If a game provides very little cognitive benefit, or none at all, then it would not be considered an epistemic game. This chapter presents a strategy for analyzing digital epistemic games, thus allowing anyone to better determine what the cognitive benefits of a game are. This is of particular importance to educators who wish to integrate digital epistemic games into their learning material, since knowing what cognitive benefits a game provides will explain how the game can best supplement the learning material.

This chapter will first examine different methods of categorizing games, to determine whether and how these methods will help describe the cognitive benefits that different digital epistemic games can provide. Next, this chapter will examine patterns and describe a pattern-based strategy to categorize epistemic games, followed by a list of identified patterns. Lastly, the pattern-based strategy will be used to analyze several digital epistemic games, providing an example of how this strategy can be used by others.

3.1.1 Game Categorization

Taxonomies provide a structure for organizing objects, based on common properties, and for comparison or analysis of the organized objects. Different taxonomies or typologies exist to categorize digital games, each focusing on different common properties. Games are traditionally categorized according to genres, with a game placed into one genre or a sub-genre. However, there is some concern that the current genres used in the digital game industry are influenced by marketing and journalism (Järvinen, 2002; Elverdam & Aarseth, 2007), mainly due to a lack of consistency among the genres used (Alpert, 2007). Additionally, there is disagreement about how the genres categorize digital games. Some suggest genres categorize by the representational and aesthetic aspects of games (Novak, 2007; Caldwell, 2004), while others suggest the categorization is based on the gameplay elements and forms of interaction (Rollings & Adams, 2003; Apperley, 2006; Myers, 1990). Still others suggest the current genres involve the narrative, mood, and feel of the game (Elverdam & Aarseth, 2007).

Of these different interpretations of genres for categorizing games, categorizing by gameplay and forms of interaction are most likely to describe what cognitive skills the player will need to use. As an example, Crawford (1984) suggested dividing games into the two general categories of skill-and-action games, which focus on physical dexterity and reaction time, and strategy games, which focus on cognitive skills. This categorization would be somewhat useful, as any game in the strategy genre would be a digital epistemic game. However, Crawford’s categorization is outdated so it is better to look at the more current genres used for categorization. Several of the more commonly used genres, with the interpretation that the genres categorize games by gameplay and by types of interaction, are shown in Table 1. Several specific genres, such as racing, are not included in the list as they can be considered sub-genres.

Table 3-1: Common game genres (Rollings & Adams, 2003; Alpert, 2007; Novak, 2007)

Action	Involves physical challenges and personal conflict, relying on hand-eye coordination, reaction time, and reflexive actions.
Adventure	Involves exploration, discovery, and solving puzzles or problems.
Puzzle	Involves one or a series of puzzles or logical challenges, often without much time pressure. These tend to be short in length.
Role-Playing	Involves one or more characters that progress and develop in various ways as the game advances, often coinciding with a story. These games also involve exploration, tactical conflict, and some logistical or economic conflicts.
Simulation	Involves replication of real-world scenarios or systems, such as vehicle simulations, sports games, or business and management simulations. The problems and goals of these games are similar to those of the replicated situation or system.
Strategy	Involves strategic, tactical, and logistical challenges requiring players to perform careful resource management. The depth and complexity of the challenges depend on the time constraints.

The puzzle genre from Table 1 seems to be the most likely category to locate epistemic games. However, there may also be epistemic games in the adventure genre. The remaining genres, with the possible exception of action, may also provide some degree of cognitive benefit. It is this "degree" which poses the biggest problem. Given that there are only vague notions of what cognitive skills are necessary, and no indication of the degree to which those skills are needed, categorization by genre does not provide the necessary information for an educator to properly select a game.

However, other taxonomies exist and could be examined. Wolf (2001) created a taxonomy based on types of interaction for gameplay and similar representational or thematic style, resulting in 42 different categories of which some may not be considered digital games. Although these categories differentiate between types of interaction, the focus is on how the game is played and not on what cognitive skills the game requires or teaches. Apperley (2006) suggests that games should be thought of as "belonging to several genres at once" rather than keep the restrictive view that games exist in one genre only. Although Wolf's taxonomy does not do this, it is also a combination of interactional style and representational style, suggesting that perhaps a multi-dimensional taxonomy would be better suited to categorize games.

Several multi-dimensional taxonomies are available, allowing games to be categorized based on a combination of properties rather than just one property. For example, rather than creating categories based simply on how the player interacts with the game, one dimension of the category could be interaction while another dimension could be how time affects the gameplay or how different narratives are used. Aarseth et al. (2003) developed a multi-dimensional taxonomy to describe games based on structural properties – space, time, player-structure, control, and rules – to understand in more detail how various games are similar or different at a structural or spatial level. Lewis et al. (2007) used multidimensional scaling techniques to compare the similarity of games on a 2-Dimensional map, in order to determine how players decide why a game is similar to another game. This map resulted in a categorization of games with respect to their aesthetics versus their mechanics. The Game Ontology Project (Zagal et al., 2005) is another framework that focuses more on analysis of games than just categorization, examining the various structural components of games and the relationships of these components. In contrast, the Game Design Rules (Falstein, 2002) focus on principles for better game design and the Game Design Patterns (Björk et al., 2003) identify common structural components of various games. These two frameworks provide an alternate methodology to examine the components of games and to better compare, analyze, and understand games. However, these frameworks can also be used for categorization (Björk & Peitz, 2007).

3.1.2 Epistemic Patterns

Although none of the categorizations described in the previous section organized games based on their epistemic utility, one deserves extra attention. The pattern-based approach to categorization used by Björk et al. (2003) focused on the structural aspects of digital games, but the same pattern-based approach can be used to categorize other aspects of games. Patterns are anything that repeat in a predictable and organized manner. As far as design is concerned patterns can be used as guidelines for the creation of, analysis of, or solution to commonly occurring problems or features.

Patterns were first applied to design by Alexander et al. (1977) to examine common problems in architecture. By identifying architectural problems as patterns, solutions could be discussed that were known to work well against those problems. Thus, the patterns became useful as a problem-solving tool for designers. The idea of using patterns to identify common problems, and their solutions, has since been applied to a variety of other disciplines such as software design (Beck & Cunningham, 1987; Gamma et al., 1995; Buschmann et al., 1996; Fowler, 1996), human-computer interaction (Borchers, 2001), game design (Björk & Holopainen, 2005; McNaughton et al., 2004; Mor et al., 2006), and business (Sadtler et al., 2006). (For an in depth discussion of design patterns and their applications see Dearden & Finlay, 2006).

However, in many of those cases the patterns are used to solve design problems. The patterns this chapter will examine do not involve design problems, nor do they involve best practices or design decisions to avoid. This chapter uses the notion of a pattern as a commonly occurring idea or theme, specifically with regards to knowledge and learning. This chapter introduces epistemic patterns, which are patterns of thinking or patterns of coming to know something. When applied to games, these patterns determine what cognitive skills are utilized, learned, and improved in the completion of tasks in the game. Digital epistemic games can be analyzed to determine which epistemic patterns exist in the game. Knowing the epistemic patterns that a game contains will indicate what cognitive skills and knowledge the game will teach, as players will have to learn and use those skills and knowledge in order to advance further in the game.

From here a taxonomy can be created to categorize games based on what epistemic patterns they contain, as games are likely to contain more than one pattern. Since some patterns may not have as significant an impact in a game as others, patterns in a game may be primary patterns or supplementary patterns. Primary patterns are those that the game focuses upon or that exist regularly in the game. The patterns that are performed in addition to the primary patterns for given tasks, or that are not as pervasive in the game, are supplementary patterns. This taxonomy is more beneficial to educators, as they are able to identify what cognitive benefits a game contains and to what degree those benefits exist.

To identify these patterns, multiple games as well as large collections of puzzles were analyzed. The patterns are at an abstract level, keeping the detail of how the pattern exists in a game or puzzle at a minimum, and focus more on the cognitive and learning benefits the patterns provide than on how the patterns are used in games. This list of patterns is still a work in progress. More patterns may become apparent as more digital games and puzzles are examined, and as digital games are analyzed in more detail.

In the next section the patterns are listed and explained in detail. Multiple games are then examined to explain how the patterns are used in those games, and how the games can be categorized according to these patterns.

3.2 Pattern List

3.2.1 Arranging

The arranging pattern deals with changing the spatial positions of game objects or components of those objects. Arranging objects in space can lead to knowledge building, sensemaking, and improved problem solving (Ankerst et al., 1998; Kirsh, 1995; Kastens et al., 2008; Yang et al., 2003). Digital games that are designed with arranging can help players to understand underlying patterns, trends, and relationships between objects, (Ankerst et al., 1998), as well as help with organizational skills (Kirsh, 1995; Kastens et al., 2008).

The arranging itself is usually restricted in some manner, dependent on the game itself. For example, objects could be arranged only on a grid, obstacles could prevent objects from being arranged past them, physical laws such as gravity may be simulated to affect where the object ultimately ends up after the arrangement, and so on. There are a variety of purposes for arranging, such as placing objects in a specific way, sorting objects, or aligning objects to give insight into a bigger problem. Additionally, in games where a player manipulates a representative character – such as many action, adventure, and role-playing games – the act of simply moving that character around may not qualify as arranging. Moving a representative character is dealt with in the *searching* pattern.

3.2.2 Assigning

This pattern refers to the act of giving objects within the game meaning, properties, functionality, and so on. Assigning meaning or properties to objects within a game has the cognitive benefits of improving sensemaking, problem solving, and learning (MacKeracher, 2004; Renkl, 1997). By assigning meaning to objects or actions within the game, players are able to make sense of the game environment, the objects within it, and the relationships among the objects (MacKeracher, 2004). This is a more contextual example of making sense of anything in the world (MacKeracher, 2004) and supports more successful learning in general (Renkl, 1997).

Players may assign meaning to objects as they play the game, though this is usually not part of the game itself. However, changing the properties, functionality, and behavior of game objects is a part of the game, and the player is limited in what properties and functionality can be changed, added, or removed by the game itself. To be considered part of the assigning pattern, any changes that players make must advance the game in some manner and cannot be merely aesthetic changes. Although some games provide the option to change properties of objects for aesthetic purposes, by definition these changes do not advance the player further in the game.

3.2.3 Collecting

This pattern refers to gathering various objects for a specific purpose. Simply collecting objects may assist the cognitive development provided by other patterns, but collecting information about the game provides insight on how to organize and analyze information one collects (Kuhn & Ho, 1980). Often collecting is the goal of a game, where players must collect all of some object or a specific number of some object. In other cases collecting is simply a bonus, where players can collect all of some object and are given a bonus reward for doing so. When integrated within a digital game, collecting can be the motivation players have for trying to advance or the game can require players to figure out how they can collect all the necessary objects in order to advance further in the game.

3.2.4 Comparing

Comparing refers to examining objects to better understand their similarities and differences. This leads to matching similar objects, finding counterparts for objects, discovering the object that does not belong or is different, and so on. Comparison provides several cognitive benefits, such as learning, sensemaking, and problem solving (Gentner et al., 2007; Oakes & Ribar, 2005). Additionally, comparing is a key component of the problem solving process, organizing objects, and experimenting with objects (Smith & Medin, 1981). The complexity of similarity can be much greater than simple visual differences though, such as comparing the meanings of words or comparing two objects based on the relationship of their components. The comparing pattern often supplements other patterns, and is rarely an independent pattern in a game. For example, the comparison pattern would supplement the arranging or collecting pattern by requiring players to compare different objects in order to arrange or collect them.

3.2.5 Composing

This pattern refers to the act of bringing objects together and merging them into one larger object, although it can also be seen as creating new objects out of components. The cognitive benefits that composition provides focus on problem solving, planning, and learning (Abrahamson, 2006; Cox, 1999; Grossen & Carnine, 1990). Composing also facilitates analytical reasoning (Grossen & Carnine, 1990) and creativity with respect to creating new ideas (Cox, 1999). Merging objects to create a larger object may be the goal of the game, in which case this pattern is what motivates the player to advance through the game, but the merging of smaller objects may just be one part of the game.

3.2.6 Filtering

The filtering pattern allows players to change the amount of information, or type of information, displayed to them. The information is often provided in layers, allowing players to select and view what information is relevant. This decreases the perceptual and cognitive processing players need to perform in order to play the game, providing cognitive benefits such as reasoning, problem solving, decision making, and learning (Stolte et al., 2002; Kastens et al., 2008). Filters may be necessary in a game, as the information available to the player is too much display all at once, or they may be optional, allowing players to decide how much information they want displayed.

3.2.7 Linking

This pattern involves connecting, joining, relating, or associating objects in some way, as well as understanding the connections and relationships between various objects. Linking aids in organization and incremental decision making (Foster & Stefik, 1986). In addition, linking provides the cognitive benefits of planning, learning, sensemaking, and problem solving (Peterson & Snyder, 1998; Faisal, et al., 2009). The relationships between objects may be obvious in a game, or explained in a manual or tutorial. Some games though do not explain the relationships between objects, or make these relationships unclear, requiring players to understand how objects are connected in order to advance or complete the game.

3.2.8 Probing

The probing pattern involves examining, focusing on, and looking into game objects for more detailed information. Given that players cannot attend to all the information available to them at once (Ormrod, 1995) probing helps to focus their attention and process information more deeply, which is a significant requirement for learning (Hannafin & Hooper, 1993). Games provide probing to players in a variety of ways, such as zooming or panning a level in the game, inspecting individual game objects for information, and even requesting in-game help for specific objects or game concepts. Probing may be included in a game but it is not required in order to successfully complete the game. This is particularly noticeable if the player has already completed the game, as the player may already know whatever information probing could provide.

3.2.9 Searching

The searching pattern involves players finding objects within the game, finding information about the game or game objects, or finding a path through a portion of the game in order to advance. Searching is a directed activity, unlike simply wandering through the game world, as players are actively trying to understand the problem given to them and develop a solution for it (Bates, 1986). This pattern also provides the cognitive benefits of sensemaking and problem solving, as searching helps with acquiring new knowledge and updating existing knowledge (Rowley & Hartley, 2008). Finding a hidden object in a game may only be part of the solution, as players may need to overcome other obstacles in order to actually obtain the hidden object. As well, games may include searching as an optional task to provide hints or bonus rewards. Finding the exit in a maze-like environment, or finding a safe path through many obstacles in the game, are also examples of the searching pattern regardless of whether finding that path involves moving a representative character around the environment or not.

3.2.10 Selecting

The selecting pattern refers to choosing an object in the game and marking it, such as to keep track of the object or to remember it for further investigation later. Selecting usually precedes other related actions players may perform, such as selecting which object to arrange, and is usually connected to other patterns in a manner that enriches exploration and discovery (Yi et al., 2007). Some games involve objects that are always selected, or the game automatically selects objects for the player. These games may contain the selection pattern, depending on whether the selection involved helps the player keep track of the selected object or not.

3.3 Analysis of Games Using these Patterns

In order to better understand how these patterns are used in games, and how these patterns can be used to categorize games, several examples are shown below. Each example game is first described, to give a sense of what players do when they play the game. Then the patterns in the game are described to make it clear why the games contain those patterns. Although games often contain multiple patterns some are obviously more important than others. These patterns are denoted as primary patterns, in that they form the primary way of thinking or playing the game. Other patterns are supplementary, in that they supplement the other patterns or provide an aid to the thinking of the player but are not the primary task of the game. Not all games contain patterns that can be described in this manner. All the patterns in those are primary, and thus there is no distinction between primary and supplementary. However even this distinction should be considered as a range, since the presence of the same pattern at the same level (primary or supplementary) in multiple games does not imply the pattern is equally utilized in them.

3.3.1 Adventures of Lolo

The game Adventures of Lolo[1] is broken down into many different levels that contain a variety of obstacles. To complete each level players collect all the heart objects. Once these hearts are collected a treasure chest that is somewhere on the level will open and the player must collect what is inside. Once the item inside the chest is collected the player exits through the door and proceeds to the next level. However, to collect all the hearts the player must avoid the many obstacles in the level. The obstacles come in different forms. There are objects that form walls, such as rocks and trees or a river, which the player must walk around. There are enemies that can defeat the player, and force the player to restart the level, as well as enemies that just block the player's path. Both of these enemies may stay in one place, may wander the level in a set pattern, or may chase the player. For some enemies, the player is defeated upon entering their line of sight.

---

[1] Copyright 1989 Hal Laboratory, Inc. All Rights Reserved. Adventures of Lolo is part of the Eggerland series, developed between 1985 and 2000 by Hal Laboratory, Inc.

There are also boxes that the player can push, but not pull, and otherwise act exactly as a wall.

In most levels the player has a limited number of magic shots, which allows the player to turn enemies into an egg. Shooting the egg will remove it from the level but the enemy will come back, within a few seconds, in the same location or sometimes a different location. Once the enemy has turned into an egg, the player can push it around the same as the player can push boxes around. To complete the game, players must recognize how to use the obstacles in the level to their advantage. For example, there may be a heart close to an enemy that defeats players as soon as players enter its line of sight. Players could get the heart by blocking the enemy's line of sight with a box or with an enemy that was turned into an egg. Although some levels require players to act quickly, possibly to avoid an enemy that is chasing them, players usually have plenty of time to figure out how to collect all of the hearts.

The Adventures of Lolo game contains the patterns of *arranging*, *collecting*, *linking*, and *searching*. First, since the goal of each level is to collect the hearts the *collecting* pattern plays a key role. *Collecting* provides the motivation and objective for each level, and occasionally provides another ability that the player needs to complete the level. The order in which objects are collected may matter, so it is not always possible for players to simply collect all the hearts however they want. Players use the *arranging* pattern in order to move objects around. *Arranging* provides the means for players to avoid obstacles, open new paths through the level, and collect the needed objects.

Searching is required for players to find the correct path through all the obstacles. Both *searching* and *arranging* are interconnected here, because to find the correct route players may need to move objects out of the way or move an object to prevent other enemies from stopping them. Players may also need to figure out the path to an object which, when moved, will open up a path elsewhere. Finally, *linking* is required in order for players to understand how the various objects on the level are interrelated. Players must recognize how enemies could be used in a beneficial way, how to use other obstacles to protect themselves, the order in which to collect the hearts to prevent players from becoming trapped, and so on. Without this understanding, players will not be able to complete any of the levels in the game.

For categorization purposes, Adventures of Lolo focuses on *arranging*, *linking*, and *searching* as primary patterns with *collecting* as a supplementary pattern.

3.3.2 Bejeweled

The game Bejeweled[2] consists of a grid of jewels, each jewel having one color out of a set of possible colors. Players must arrange these jewels such that there are three or more jewels of the same color in a horizontal or vertical line. However, players can only swap two adjacent jewels and are not allowed to move them in any other way. As well, players can only swap jewels if it creates a line of three or more jewels of the same color. When a line of jewels of the same color is created, the jewels disappear and all the jewels above that line in the grid are moved down, with new jewels of randomly chosen colors placed at the top.

---

[2] Copyright 2001 – 2007 PopCap Games, Inc. All Rights Reserved. Screenshot used with permission.

Figure 3-1: A game of Bejeweled in progress. Two vertical lines of jewels were simultaneously created, and the jewels were just removed from the board. Note that the jewels are of different shape as well as different color.

The primary pattern that exists in this game is *arranging*. This pattern provides the primary way for players to reach the objective of the game: by swapping the positions of the jewels in order to create lines of jewels of the same color. The *comparing* pattern is connected with the *arranging* pattern. Players compare the colors of jewels in order to determine which ones to arrange. The positions of the jewels are also compared, in order to determine if swapping two jewels will give the desired result.

This game also contains the *composing* pattern, which is the goal of the game. Players are not arranging the jewels blindly; they are arranging the jewels in order to compose a horizontal or vertical line. Lastly, this game contains the *selection* pattern, which is connected to how players swap jewels. Players must first select one jewel, which then becomes highlighted to indicate it is the one selected. Players then select another jewel that is adjacent to the selected one and the swap is performed.

For categorization purposes, this game contains *arranging* and *composing* as primary patterns with the *comparing* and *selecting* patterns as supplementary.

3.3.3 Blocksum

The game of Blocksum[3] focuses on a grid of squares. Blocks of varying sizes and shapes slowly rise from the bottom of the screen to fill up the grid. Each block contains a number, which is not necessarily related to its size. For example, a block could have the number two and be as large as two grid squares. However, a block could also have the number four and be one grid square in size. Players are able to combine individual blocks with adjacent ones, causing the new block to be the combined size and shape of the component blocks. As well, combined blocks have a number that is the sum of the numbers of the component blocks. Thus, combining a block with the number three and a block with the number four, both of which are one grid square in size, will create a block with the number seven that is two grid squares in size.

The purpose of the game is to combine blocks in order to eliminate them. Blocks are eliminated when they are adjacent to other blocks with the same number, provided there are at least as many adjacent blocks as the number on the blocks. For example, there must be at least three adjacent blocks each with the number three in order for the blocks to disappear. By removing blocks from the grid, players receive points and create space for the new blocks that continue to rise.

---

[3] Copyright 2004 – 2006 Shintaro Sato, Ginger, XOR http://infotech.rim.zenno.info/products/blocksum/en/

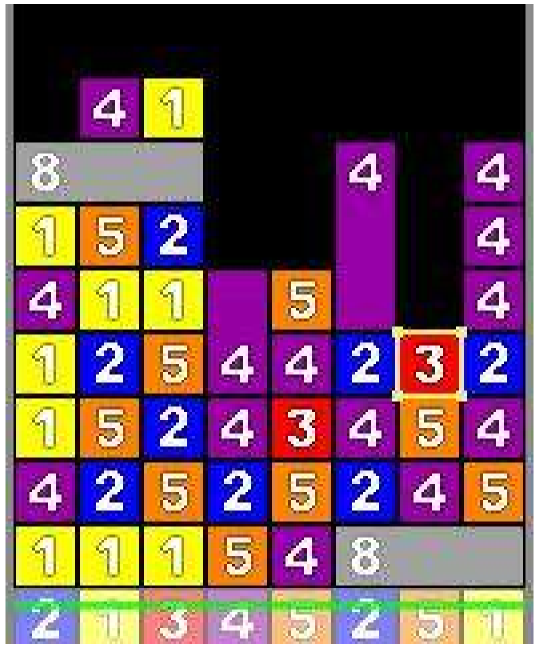

Figure 3-2: An adaptation of the game Blocksum. The player's cursor is currently on a three block. Note the new blocks rising from below the green line. Also note that the size of the blocks do not always correspond to their number.

This description suggests that the main pattern of the game is *composing*. The *composing* pattern provides the goal of the game, since players advance by *composing* blocks. This pattern also describes how players actually play the game: by merging blocks together to form new ones. Players that wish to advance far in the game must plan ahead to determine which blocks to compose immediately and which blocks to leave for later in the game.

However, there are other patterns that exist in this game. The *searching* pattern plays a minor role, as players may need to search through the blocks to find specific numbers. The *comparing* pattern is related to this. Players need to compare the numbers on the different blocks in order to determine which blocks they want to compose. The *selection* pattern is also required in order for players to select the blocks they will compose. All three of these patterns play a relatively minor role in the game, especially compared to the *composing* pattern.

Thus, Blocksum can be categorized with the *composing* pattern as its primary pattern and with *comparing*, *searching*, and *selecting* as supporting patterns.

3.3.4 Bookworm

This game[4] provides players with a set of lettered tiles on a grid. Players need to find words within the set of the tiles and select the tiles in order to spell out the word. When words are found, the tiles that compose that word are removed from the grid and new letters are added. Some tiles flash red and slowly burn the tiles below them. Should the burning reach the bottom of the grid, the game ends. These burning tiles add a sense of urgency and timing to a game that otherwise does not force any time constraints on the player.

Figure 3-3: A game of Bookworm in progress. The player has found and highlighted the word "star" and now just needs to submit it. The player could also select the 'E' below the 'R' to form "stare" instead.

---

[4] Copyright 2003 – 2004 PopCap Games, Inc. All Rights Reserved. Screenshot used with permission.

The main pattern in this game is the *searching* pattern. The game requires that players continually search through all the letters to find words. Players can search for smaller words, or look for longer words to obtain a higher score. *Composing* also plays a significant role in this game, as players create the words by composing the individual letters together. The *selection* pattern provides a supporting role. Players must select the tiles in order to create words, and the tiles become highlighted whenever they are selected. As well, the order in which the tiles are selected is shown to better describe what word players are forming.

Bookworm can be categorized with *composing* and *searching* as primary patterns, and with the *selecting* pattern as supplementary.

3.3.5 Civilization 4

This game[5] is played on a large grid of tiles, with each tile representing a different type of terrain that provides different benefits. Players take on the role of guiding their civilization to victory in different ways such as culturally, diplomatically, economically, militaristically, or scientifically. Players do this by directing individual units around on the grid in order to explore, defend, or attack an opponent as well as build cities to provide a source of income and area of production for new units. Players must also manage their resources, to ensure they have a steady income and to provide enough funding for research to increase the effectiveness of their civilization.

---

[5] Sid Meier's Civilization IV: Beyond the Sword copyright 2005 – 2009 Firaxis Games and Take-Two Interactive Software. All Rights Reserved. Screenshots courtesy of Firaxis Games and Take-Two Interactive Software, Inc.

Figure 3-4: The main game screen for Civilization 4. The city in the center provides information about its size (9), what it is building (stables), and any defenders (currently one archer). Special resources are highlighted and gridlines are turned on to ease distance calculations for the player.

One of the primary patterns in this game is the *arranging* pattern. Players arrange units around on the grid to further various goals. However, players do not normally need to find a route for their units as the game always shows the shortest route to the grid destination the player selects. Given that exploration plays an important role, especially early in the game, *searching* is another important pattern that exists. Players must search through the world to find good locations to build, to find their opponents, and to find other bonuses that may be hiding.

In order to move any unit around on the grid, the player must select the unit. The *selection* pattern can be seen when the player selects a unit, as the unit becomes marked to remind the player about which unit is currently selected. Players can direct the selected unit around, but the unit can also be assigned specific instructions. Players can choose to have a unit automatically explore the map, to try to build roads between their cities, defend a location, and so on. This is an example of the *assigning* pattern.

To build a city in the best location players must take into account what resources are close by, how far away the resources are from the city, what benefits the terrain around the city gives, and if there are any strategic advantages to the location. The same tiles that contain this information also contain cities the player has built and units that the player controls, as well as cities and units of the opposing players. Thus, the *comparing* pattern is used here. Players compare various tile locations in order to determine the optimal location for a city. Players also compare the strength of their civilization with the strengths of their opposing civilizations in areas such as military strength, scientific progress, financial prowess, and so on.

Figure 3-5: The same location as in Figure 4, except an economic filter was turned on. Each tile now shows what economic bonuses it provides at a glance, so players do not have to examine each tile to see what is provided. In contrast to Figure 4, the filter to show special resources was turned off.

When all the terrain information is displayed it can be very difficult to see where different units are, particularly opposing units. The *filtering* pattern is available in this game through various filters the player can turn on and off. By using these filters, players are able to see the information they require when they want it and can turn off extra information when it begins to clutter the screen. *Probing* can provide extra information that may or may not be available through the filters. Players can probe individual units for extra information, such as probing an archer unit to investigate what other units it could upgrade to. Players can also probe their own cities to get extra information about its production, its resources and income, and so on. The probing functionality is also linked with the in-game help, so that players who require more information on how to use a certain unit can easily access that information.

Figure 3-6: The same location as in Figure 4, but all the filters are turned off except the terrain filter. Now the player only sees the underlying terrain and its improvements, such as roads and the city. The player's units, economic information about game tiles, and the grid are all hidden.

Thus, Civilization 4 contains the *arranging*, *assigning*, *comparing*, *filtering*, *probing*, *searching*, and *selecting* patterns. No individual pattern is a primary pattern in this game, as the patterns complement each other and are interconnected. In order to achieve the best score, players should take advantage of the various features each pattern gives – particularly the *comparing*, *filtering*, and *probing* patterns – but the game can be played without significantly considering the *assigning*, *filtering*, and *probing* patterns.

3.3.6 The Incredible Machine

The Incredible Machine[6] is broken down into a large number of puzzles, with each puzzle requiring some different goal. Each puzzle is a Rube Goldberg-inspired machine, which is a complex-looking machine with various parts that performs a simple task such as catching a falling ball or lighting a candle. Some of the pieces create the structure of the machine, such as walls, and are static. Other pieces are more dynamic, such as the freely moving tennis ball or balloon, and react to other pieces as well as cause other pieces to react to them.

For example, a tennis ball could fall on a mouse cage. This causes the mouse inside the cage to start running on a wheel. The wheel is attached to a conveyor belt, which will start to move a bowling ball off the edge of the belt. The bowling ball falls and lands on an incline, causing it to roll down off the bottom of the screen.

Players have access to specific pieces that they need to position somewhere on the level, connect to other existing pieces, and then run their machine to see what happens. If the pieces are attached correctly, the puzzle will be solved. Otherwise, players will have to rearrange the pieces and try again. Due to this, one of the primary patterns of the game is *arranging*. Players need to arrange the game pieces in the correct manner to solve the puzzle. The pieces that are already placed on the level only form a partial solution. Players cannot arrange those pieces, but must take them into account when placing and arranging the remaining pieces.

---

[6] Copyright 1993 – 2002 Activision Blizzard, Inc. and its subsidiary Activision, Inc. All Rights Reserved.

Linking is another important pattern in the game. Players need to understand the connection between the various pieces of the game in order to utilize them. For example, one level could contain a cannon and the objective is to put a cannon ball in a basket on the other side of the cannon. Even if the cannon is fired, the cannon ball will move in the direction opposite to the goal. Players need to identify the connection between the cannon and objects that will move a cannon ball, such as an incline or conveyor belt. Players also need to recognize what can be used to fire the cannon, such as a flashlight to generate light and a magnifying glass to focus the light onto the fuse of the cannon. *Linking* is also used when connecting the various pieces of the game. For example, a balloon could be floating freely or it could be connected to a rope that is connected to a gun. When the balloon flies into the air, it will pull the trigger of the gun and fire it.

Other supplementary patterns exist as well in this game, such as *collecting*, *selecting*, and *searching*. The *collecting* pattern exists as a goal in several levels. Players may need to collect a certain number of objects in some manner, such as collecting three basketballs in a bin at the top of the screen. The *selecting* pattern is connected to the *arranging* pattern, since players must select the object to arrange before it can be moved. The object being moved is always highlighted but when no object is selected, the mouse can be positioned over other objects. Positioning the mouse over an object that can be moved will highlight it, notifying the player that the object can be arranged. Players use the *searching* pattern to plan routes for objects through the level. For example, if a player needs to move a basketball from the top of the screen to the bottom, that player will need to find what paths the basketball can take depending on what obstacles are placed in the way. This task of finding routes through a level is often central to deciding how to arrange the given objects.

For categorization purposes, The Incredible Machine contains *arranging* and *linking* as primary patterns and *collecting*, *selecting*, and *searching* as supplementary patterns.

3.3.7 Legend of Zelda: Ocarina of Time

In this game[7] players direct their representative character through a variety of different environments. Some of these areas are very open and are connected in an obvious fashion, such as a road leading from a town towards the highlands. Other areas are dungeons or temples and are much more restrictive, containing different puzzles and obstacles that players must try to pass. Throughout the game are numerous different enemies that players must defeat, often by finding and exploiting the enemies' weakness.

Although navigating throughout the environments in the game is usually straightforward, navigating through several of the dungeon levels is often more difficult. It is in these dungeon areas that *searching* is necessary. In one dungeon, for example, walls of fire impede the player's movement but these walls are not visible until the player almost walks into them. Thus, players must walk carefully to see whether a hidden wall exists and try to determine how to reach their destination using this technique. Note that in the areas outside the dungeons it is not necessary to search for paths, as the areas contain very open spaces with the only obstacles being easy to avoid and the paths often clearly marked. Players may also need to search areas for special hidden items, either to advance in the area or as a bonus. Thus, the *collecting* pattern exists in the game. Although in each dungeon area there is one major item the player must find and collect, the game also involves a large number of bonus items that the player can collect but does not need to. For example, one of the major collectable items in the game is a golden token. Early in the game these tokens are hidden in rather obvious places, such as in an alcove by the exit door. Later in the game, they are hidden inside boxes or placed behind walls that the player must break through with a bomb. Searching becomes a key activity then if the player wants to collect all of these items

Linking is one of the primary patterns in this game, with players constantly required to understand how different objects are related. For example, one dungeon shows a locked door, two unlit torches beside the door, and one lit torch elsewhere in the room. Players

---

[7] Copyright 1998 – 2007 Nintendo Company, Ltd. All Rights Reserved.

must make the connection that the torches should be lit, and a hint in the game suggests that the torches are related to the locked door. In order to light the torches, players must again make the connection that by holding a wooden stick close to the torch, it will catch on fire and can be used to light other torches. Once players have discovered this, further puzzles involving the torches are made harder by increasing the difficulty of lighting the torches. Since it is assumed the player knows about lighting the torches, the difficulty then becomes reaching the torches before the player's stick is extinguished. The burning stick could be extinguished by water falling on the player, an enemy knocking the player down, or the stick becoming completely burnt. Also, players may need to perform other actions before they can reach the desired torch, such as first moving a ladder into place.

Arranging is also used in the game for many puzzles. Several dungeon areas involve large blocks that the player can stand on. However, the player can also push and pull these blocks along specific tracks. The player must figure out where to move the blocks in the area in order to open a path to the exit, or in order to reach a higher room. As another example, one dungeon is filled almost entirely with water. The player can swim but cannot dive very far, so it is advantageous to drain some of the water from the dungeon in order to reach other rooms. However, draining the water affects the entire dungeon and many of the rooms in the dungeon contain blocks that float on top of the water. By changing the water level the player will rearrange these blocks and open new paths, while closing others. Players need to consider which room they can get to with the water at what height in order to complete the dungeon.

The patterns of *arranging*, *linking*, and *searching* are often interrelated. For example, in one dungeon the player must cross a pool of water. The player can swim across, but the other side of the pool is too high to climb out of. Instead, the player must make the connection – hence linking – that firing a magic arrow will temporarily freeze part of the water and create a platform. However, the player can only create a limited number of these frozen platforms and the platforms do not last forever. Thus, the player must figure out how to arrange the ice platforms to discover the route to the other side of the pool.

For categorization, the Legend of Zelda: Ocarina of Time contains *arranging*, *linking*, and *searching* as primary patterns and *collecting* as a supplementary pattern.

3.3.8 Myst

The purpose of the game Myst[8] and its sequels is to investigate the worlds presented in each game. Players must examine objects in the world, try to understand the purpose of the objects, and use that knowledge to complete the various puzzles that arise. The main patterns that exist in this game are *linking* and *searching*. Players must search each of the areas to find clues. Then, players must understand what the clues explain and connect that with the objects around them. However, understanding how different objects work through experimenting with them is sometimes allowed. By *probing* objects in this manner, players are able to gain a better understanding of the meaning and purpose of those objects.

For example, in the original Myst game players could visit a planetarium that showed the position of stars at various times of the year. Players could view the star positions for any night of the year, but the purpose of doing so was not clear unless the players read a book in the library that mentioned the position of stars. Players then had to make the connection between the star maps in the library book, the planetarium, and a clue about the correct dates. By choosing the correct dates, the players were given symbolic clues that they then needed to connect with symbols shown on another part of the island. Making that connection allowed them to proceed to the next area of the game.

Thus, Myst is categorized as containing *linking* and *searching* as its primary patterns and *probing* as a supplementary pattern.

Figure 3-7: This shows the planetarium's star screen in Myst, where the player enters the date and time using sliding bars on the right to receive a star map on the left.

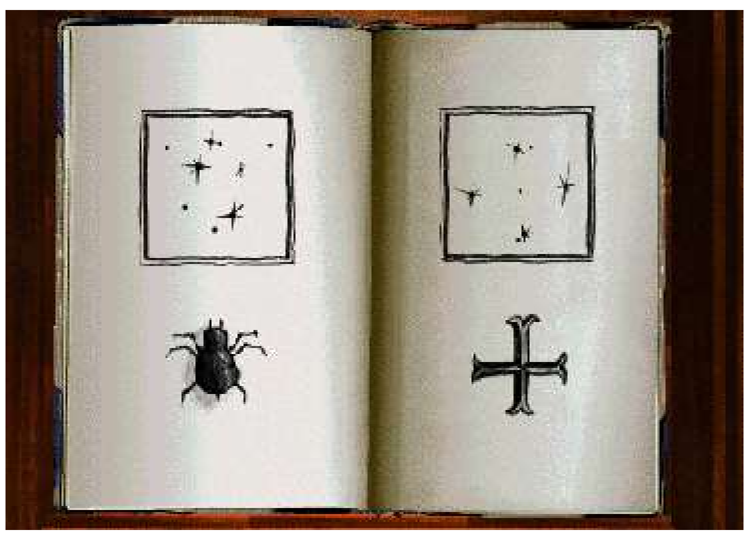

Figure 3-8: In this image, the library book is shown to help the player match up the stars shown in the planetarium with a symbol. With the star map from figure 8, the player has found the insect symbol.

3.3.9 Pacman

In this game[9], players direct their representative character through a maze. Four ghosts chase the player through this maze, and the player must try to avoid the ghosts while attempting to collect all the pellets scattered throughout the maze. There are bonus items that players do not have to collect but are awarded a bonus if they do. There are also power pills that, when collected, allow players to eat the ghosts chasing them. Again, the power pills are not necessary to collect but provide a temporary advantage if players choose to collect them.

Searching is one of the main patterns in this game. Players must find their way through the maze in each level, trying to avoid the constantly moving ghosts, in order to collect all the pellets. The other main pattern in this game is *collecting*. In order to advance in the game, players collect the pellets that are all throughout the level. There are other objects players can collect as well, such as the power pills, but these are not required in order to complete the level. For categorization purposes, Pacman contains *collecting* and *searching* as primary patterns.

3.3.10 Step-By-Step

This game[10] is broken down into multiple different levels. Each level contains several round tiles, in different layouts, that players must walk over. The tiles disappear after the player walks on them a certain number of times, with the exact number depending on the color of the tile. For example, blue tiles disappear after the player walks on them once, green tiles after the player walks on them twice, and so on. The game involves finding a path from the starting location that covers every tile of the level the exact number of times needed to remove the tiles. Since tiles disappear after the player walks on them, each level contains at least one tile that does not disappear. This type of tile is where players must end their route.

---

[9] Copyright 1980 – 2007 Namco Bandai Games, Inc. and its subsidiary Namco. All Rights Reserved.

[10] Developed by Manfred Kopp, released under the GNU Public License version 3.

This game is almost entirely the *searching* pattern, as players need to find the correct path through each level. Although the correct path may initially seem obvious, it becomes much more difficult once certain tiles must be walked over multiple times. Players must carefully plan their path or else they may become stuck. The *comparing* pattern provides a supporting role in this game. Players compare the colors and types of different tiles to aid in the planning of their route through the level. As no other patterns are used in the game, Step-By-Step is categorized by *searching* as its primary pattern and *comparing* as its supplementary pattern.

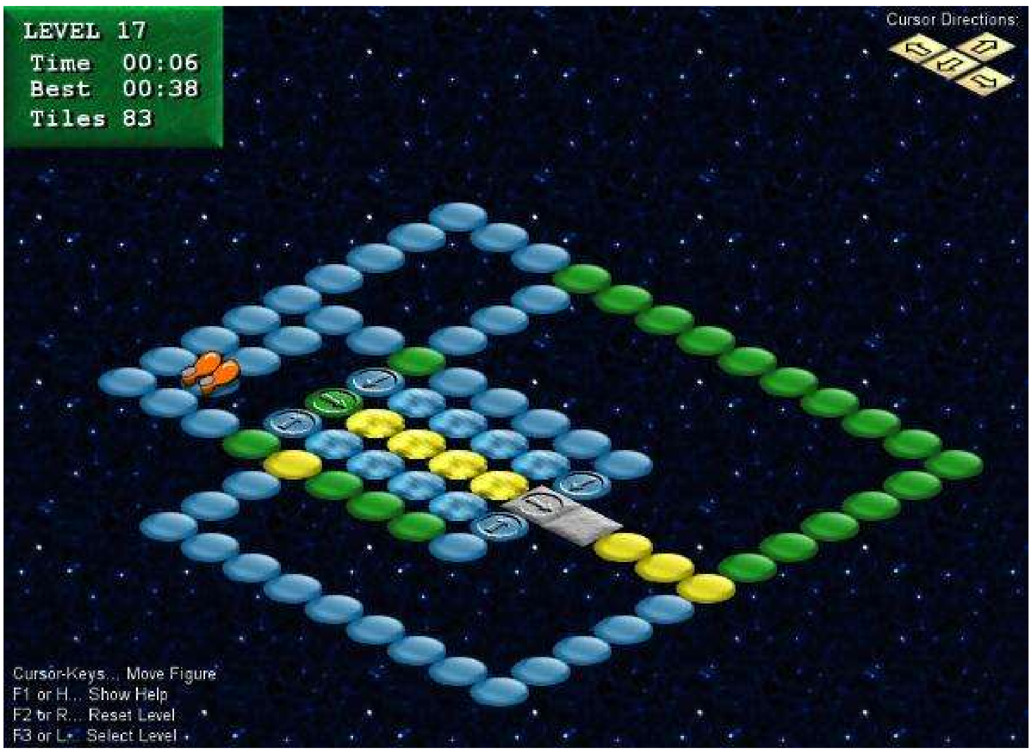

Figure 3-9: A level in Step-By-Step. Note the yellow tiles, which take three steps to remove, and the tiles with arrows, which force the player to move in that direction.

3.3.11 Tetris

This game[11] involves a single screen with a grid of squares. Blocks of specific shapes and sizes appear at the top of the grid and automatically move down to the bottom of the grid, at a constant speed depending on how far in the game the player has progressed. Players can arrange the falling block by moving it to the left or right, and by rotating it clockwise, but they cannot move any other block. To help the player decide where to arrange the falling block, the next block that will appear is shown on the side of the screen. Whenever the player fills every grid square in a horizontal line, that line of blocks disappears and the player is awarded points. If no new blocks can be added to the central topmost squares of the grid, the game is over and the player has lost. Some versions of the game have a maximum "level" at which the player can reach and win, while others have no winning level and players keep playing to achieve a high score.

The primary pattern used in Tetris is the *arranging* pattern. Players must rotate and position the falling block in the best location in order to advance. This can be especially difficult when players are given a block that is not currently useful, and the block must be arranged such that it is used later in the game. This game also uses the *composing* pattern. When players arrange the blocks to form a horizontal line across the grid, the players have composed a row of blocks. Composition provides the goal of the game, with *arranging* providing the method by which players accomplish that goal. Thus, Tetris is categorized with both *composing* and *arranging* as its primary patterns.

[11] Copyright 1985 – 2009 Tetris Holding, LLC. All Rights Reserved.

Figure 3-10: A screenshot from a game of TetrisZone in progress. The blue L-shaped block on the left is the block currently being arranged. Note that the next three blocks that will appear are shown on the right rather than next one block.

3.4 Conclusion

This chapter presented a strategy for analyzing digital epistemic games, based on ten epistemic patterns. Categorizing games based on broad genres, such as action or adventure, does not provide enough indication about what cognitive benefits the games provide, if any. Taxonomies other than genres exist, but these categorize games in ways that are also not beneficial towards explaining their cognitive benefits. Thus, a categorization based on the presented patterns was discussed. By organizing games based on the epistemic patterns, the cognitive benefits that the games provide become apparent.

Thus, these patterns help with choices about which game to use for different educational purposes. This pattern-based strategy is still incomplete, with the potential for other patterns to be discovered. Further research is needed to determine the complete set of patterns as well as a framework for applying these patterns to the design of digital epistemic games.

Chapter 4: Creative Design of Digital Cognitive Games: Application of Cognitive Toys and Isomorphism

This chapter has been published as Sedig, K., and Haworth, R. (2012). Creative design of digital cognitive games: Application of cognitive toys and isomorphism. *Bulletin of Science, Technology, & Society*, *32* (5), 413-426. http://dx.doi.org/10.1177/0270467612469069 It is reprinted here with permission from SAGE Publishing as per their Journal Author Reuse Policy.

Note that the format was changed to match the format of this dissertation, and the references were moved to the end of the dissertation. Figure numbers have also been changed to be relative to chapter numbers. For example, “Figure 4-1” is the first figure of Chapter 4 and the figure is labeled as such. However, this figure is referred to as “Figure 1” in the text of the chapter. The same is true for tables. In addition, when the phrase “this article” is used, it refers to this chapter.

4.1 Introduction

Game-playing has been, and remains, a popular leisure activity. Recent popularity and ubiquity of games is partially due to the proliferation and adoption of digital games (Siwek, 2010). According to surveys, 97% of American teenagers and over 50% of American adults played digital games in 2008 (Lenhart, Jones, & Macgill, 2008; Lenhart, Kahne, et al., 2008). The increased popularity of game-playing is precipitating the use of digital games for purposes other than pure entertainment (e.g., see Burke et al., 2009; Gordon, Lent, & Velsen, 2004; Sánchez, Gutiérrez, Cabrera, & Zea, 2009). Digital game design techniques, principles, and mechanics are even recently being used in nontraditional applications, such as training, marketing, and health and wellness initiatives (Anderson & Rainie, 2012). Digital games have increasingly been used to support reasoning, problem solving, planning, learning, and other such cognitive activities (Barab, Thomas, Dodge, Carteaux, & Tuzun, 2005; Gros, 2007; Haworth, Tagh Bostani, & Sedig, 2010; Ke & Grabowski, 2007; Linehan et al., 2011). A rapidly growing body of research suggests that playing digital games can enhance the performance of cognitive activities (Blumberg & Ismailer, 2009). The support and development of such activities, however, may be incidental and not a primary function of many games. This article is concerned with a particular category of digital games—digital cognitive games (DCGs)—whose primary purpose is to mediate (i.e., support, develop, and enhance) such aforementioned cognitive activities.

Digital games are not that effective in non-entertainment contexts, unless their design is informed by well-researched and systematic frameworks. Despite their growing popularity and potential for supporting cognitive activities, the design of DCGs is neither simple nor often informed by research (Haworth & Sedig, 2011; Rambusch, 2010; Turner & Browning, 2010). Indeed, there are no systematic design processes for the creation and analysis of such games. Game design has been largely based on ideas and experiences from the gaming industry (Barr, Noble, & Biddle, 2007). Although such experience is valuable for game designers, whether academic or otherwise, it is primarily anecdotal and intuitive, composed of general principles and best practices rather than structured and systematic design processes (e.g., see Adams, 2010; McGuire & Jenkins, 2009). This is also true when it comes to the design of DCGs. For instance, even some researchers who discuss how to design educational games (e.g., DeVane, Durga, & Squire, 2010; Dipietro, Ferdig, Boyer, & Black, 2007; Dondlinger, 2007) fail to provide or follow a systematic design process. We believe and demonstrate in this article that it is possible to have a structured and systematic approach to the design of games that does not restrict or hamper the creativity inherent in the design of digital games. In this article, we will present a preliminary process for the design of DCGs that is structured and systematic, yet allows for creativity on the part of designers. At the heart of the process is the application of two fundamental ideas: cognitive toys and isomorphism. As far as the authors are aware, these two ideas have never been combined to inform the design of DCGs.

Cognitive toys are simple, non-digital activities that people use for mental stimulation and amusement. These toys can act as sources of inspiration for digital games, either as the initial seed from which a more complete game arises or as one component of a game. How one might take inspiration from a cognitive toy and use it to create a digital game relies on another concept, that of isomorphism. Isomorphism refers to similarity between systems in terms of structure, form, function, and other deep characteristics (Gunaratne, 2008; Whitchurch & Constantine, 1993). In viewing cognitive toys and digital games as systems, designers can use features and components of a toy to inspire similar characteristics in a game. Therefore, the design process presented in this article is a formalized way in which one can systematically and creatively use cognitive toys, as sources of inspiration, to design DCGs.

The remainder of this article is divided into the following sections. In Section 2, a more detailed discussion of some foundational concepts is provided. In Section 3, the design process itself is discussed. In Section 4, an example of using this process to design two games is given. Finally, in Section 5, this article is summarized and future works are discussed.

4.2 Background

This section consists of four subsections, which will provide the conceptual and terminological background for the development of the rest of the ideas in this article. First, we will define and exemplify cognitive toys and discuss why they can be sources of inspiration for design of DCGs. Second, we will introduce general systems theory (GST) as a framework within which designers can analyze cognitive toys and design DCGs. Third, we will define isomorphism from the perspective of GST and illustrate its application in the design of DCGs. Finally, we will discuss game rules, their usefulness, and their complementary role, along with GST and isomorphism, in the creative design of DCGs.

4.2.1 Cognitive Toys

Cognitive toys are known by various other names, such as brain teasers, brain games, brainmatics, mind-benders, thinker toys, and puzzles (Kroehnert, 1991; Michalko, 2006; Moscovich, 2000, 2005, 2006, 2009). As such, the term *cognitive toy* can become problematic if interpreted too strictly. Given that a definite distinction between toys, puzzles, and games can lead to problematic definitions (Koster, 2004; Salen & Zimmerman, 2004), we define the term *cognitive toy* somewhat loosely. As such, a cognitive toy can be defined as any simple, small, non-digital, amusing, play activity that can be used to engage and stimulate the mind. Putting it slightly differently, any non-digital activity that is simple to perform, is intended for entertainment, and requires cognitive effort to perform (Bottino, Ferlino, Ott, & Tavella, 2007; Chu & MacGregor, 2011), regardless of whether or not the activity has a definite solution or a practical purpose, can be considered a cognitive toy for the purposes of the design process discussed in this article. A number of mathematical problems that have been developed for entertainment (Danesi, 2002), given how ancient some of them are (Connaughton, Tache, & Burley, 2010; Gillings, 1962; Olivastro, 1993), could also be considered cognitive toys. Additionally, a non-digital game that has very simple rules, and requires cognitive effort to play, could also be considered a cognitive toy. In each case, giving an activity the label of cognitive toy does not require one to remove a previous label. For instance, a non-digital game can be considered a cognitive toy but can also be thought of as and called a game. Two examples of cognitive toys are provided in Figures 1 and 2.

3		5	15
7			10
		14	1
			8

Figure 4-1: A sample *Magic Square* cognitive toy (See Gardner, 2001 for more). Players need to insert the numbers from 1 to 16 such that the sum of each row and column are the same.

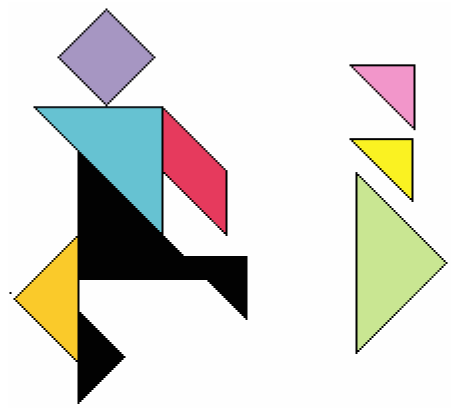

Figure 4-2: A sample *Tangram* cognitive toy that is partially complete. Players need to arrange the shapes into the outline. This specific one is from Moscovich (2005, p. 12).

Cognitive toys can be used as sources of inspiration when designing DCGs. There are at least four reasons for this. (a) Engaging with cognitive toys requires some degree of cognitive effortful thinking; thus, they can inspire the design of DCGs irrespective of their ultimate purpose. As such, DCGs that are intended to entertain and facilitate and promote cognitive activities skills, such as problem solving and reflective reasoning, could all be based on cognitive toys. (b) People enjoy playing with cognitive toys, as the popularity and proliferation of toys such as Sudoku and other puzzles can attest (Crute & Myers, 2007; Huang, Cheng, & Chan, 2007; Smith, 2005); this is not just a recent phenomenon but a historical pattern (Danesi, 2002). Designers can, therefore, use cognitive toys to create games, knowing that the source of their design is popularly considered to be fun. (c) Thousands of cognitive toys have been developed over the years and new ones continue to be developed (Connaughton et al., 2010; Danesi, 2002; Gillings, 1962; Olivastro, 1993). As such, the number and variety of cognitive toys from which designers can choose is large and continues to grow. (d) Cognitive toys can be useful for finding patterns within or similarity between DCGs.

4.2.2 General Systems Theory

General systems theory is the science of systems and provides a theoretical framework for the study of structure, properties, and characteristics of systems (Hofkirchner &

Schafranek, 2011). By a system is meant any whole that is formed from interacting, interrelated, and interdependent entities (Skyttner, 2005). In GST, systems that have physical existence—such as a tree, the circulatory system, or an ecosystem—are studied with the intent of identifying general or universal characteristics that can then be applied to the study of any system (Drack & Schwarz, 2010). Thus, GST provides a conceptual framework that assists with the structuring and organizing of thoughts and actions when analyzing or designing a system.

A system can be analyzed by identifying and describing its components and characteristics. All systems can be described through four main things: entities that comprise them, properties of those entities, relationships between entities and properties, and environment in which entities exist (Littlejohn, 2003; Skyttner, 2005). Each entity in a system can be simple or composite. Simple entities have no components or subsystems. Composite entities are composed of other entities and thus are, in fact, systems themselves. As such, a system can simultaneously have many subsystems as well as be a subsystem of another system, that is, its supersystem.

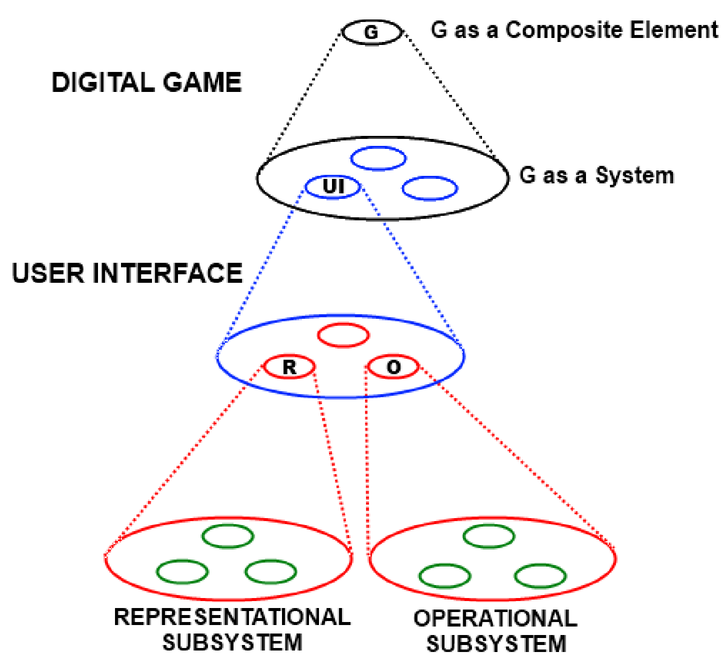

Figure 4-3: A digital game as a hierarchy of systems, with the user interface and its subsystems as an example. Adapted for digital games from Skyttner (2005).

A digital game can be analyzed as a system (Geurts, de Caluwé, & Stoppelenburg, 2000; Krek, 2008; Salen & Zimmerman, 2004; Sweetser, 2008). At a macroscopic level, a game can be decomposed into a series of embedded subsystems. At the highest level, one subsystem of a digital game is its user interface (UI; see Figure 3). The UI is a composite entity and thus a system, with two main subsystems: a representational subsystem and an operational subsystem. The representational subsystem is composed of all the information representations that are visible at the UI, each of which encodes some information about the game and may itself be a composition of other representations. The operational subsystem is composed of all the interactions that the player can perform, that is, all of the actions the player can perform on various representations, and the reaction, or change in the representational subsystem that results.

A digital game can also be analyzed at a microscopic level by choosing one specific subsystem and describing its components in detail. As an example, we will analyze both the representational and operational subsystems of the game *Tetris*[12]. First, we will identify the entities of the representational subsystem. Figure 4 provides a screenshot of *Tetris*, along with a decomposition of its representational subsystem. The main representation is itself a composition of several label representations and container representations. The containers are also composite, in that within them are embedded other representations of individual *Tetris* blocks (called Tetriminos). All of these representations are the entities of the representational subsystem. Once identified, we can determine properties of each entity. For instance, some of the properties of Tetriminos would include their shape, size, and orientation. We can also determine relationships these entities may have, such as the spatial arrangement of Tetriminos in the "playing area" container. Second, we will identify the entities of the operational subsystem of Tetris. This would include the two interactions the player can perform: rotating a Tetrimino, and arranging its spatial position in the "playing area."

---

[12] Tetris ® and © 1985-2012 Tetris Holding.

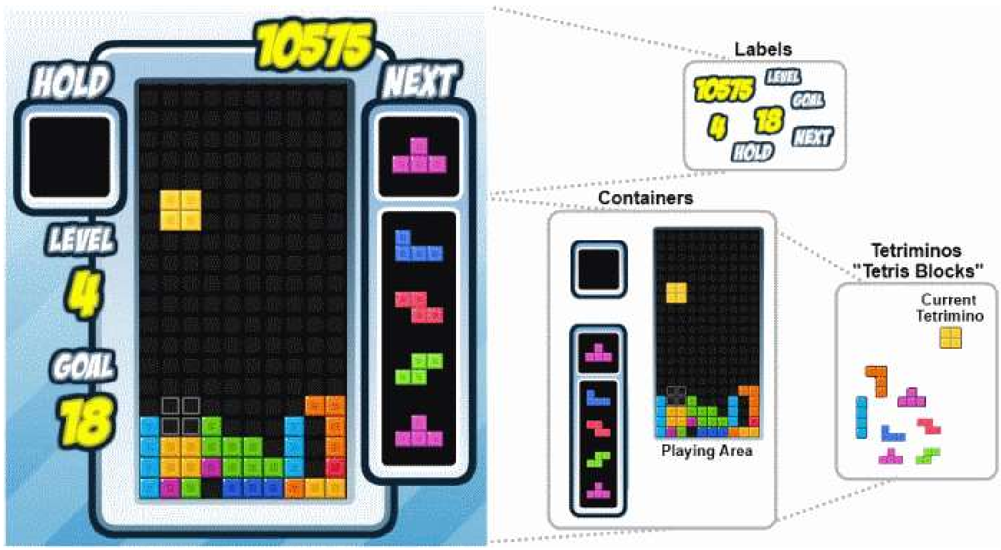

Figure 4-4: A screenshot from the game *Tetris* and the decomposition of its representational subsystem.

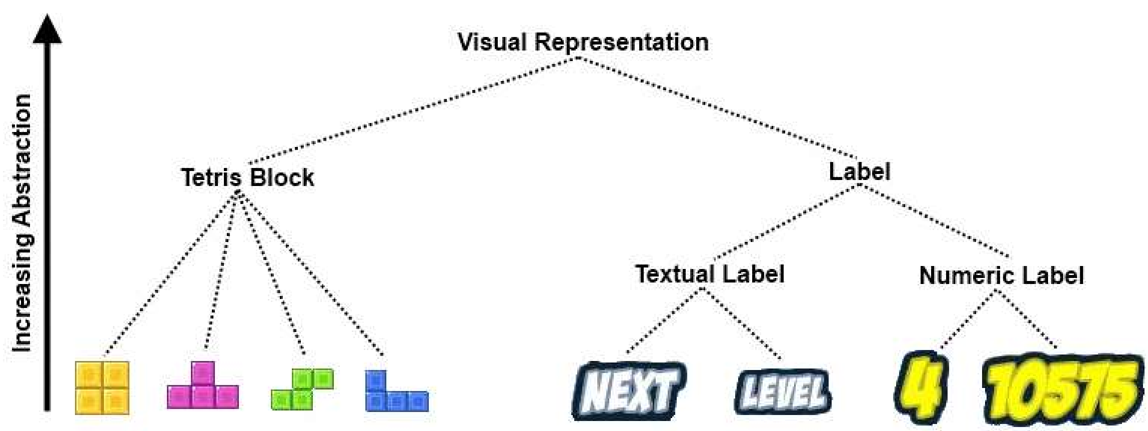

Figure 4-5: Degrees of abstraction when examining entities of the representational subsystem of *Tetris*.

4.2.3 Isomorphism

Isomorphism refers to similarity between two systems at a deep level (Whitchurch & Constantine, 1993). This similarity can be conceptualized in terms of correspondence between two systems: in terms of their forms, shapes, structures, operations, internal relations, representations, and/or other characteristics. In GST, finding isomorphism between two systems helps with the analysis and understanding of an unknown or lesser known system's entities and its behaviors with respect to a well-known system. Thus, comparing two systems at an abstract level allows us to see if they are isomorphic. For instance, at a certain level the *Magic Square* cognitive toy can be isomorphic to the *Number Scrabble* game, and both are isomorphic to the *Tic-Tac-Toe* game (see Dou et al., 2010). Similarity can occur at multiple levels of abstraction, but the isomorphism with which we are concerned in this article occurs only at higher levels of abstraction. Once entities of a system are considered in terms of abstract patterns, isomorphism between entities and between systems becomes apparent.

To discover the abstract patterns of any game, both the representational and the operational subsystems of the game can be analyzed using GST as a conceptual framework. For instance, consider the entities of the representational subsystem of Tetris, as shown in Figure 5, and discussed in the previous section. These entities can be examined at higher or lower levels of abstraction. If we focus on each entity in terms of all its properties in a specific manner, we achieve little abstraction. As we group and classify entities, a higher degree of abstraction is achieved. For example, using the term Tetrimino or *Tetris* block is a generalization and is more abstract than referring to a specific Tetrimino in the UI. A similar generalization can be made for the various labels on the interface. At higher levels of abstraction, both the labels and the blocks are simply representations.

Once similar entities are grouped together under a specific term, this abstraction allows us to discover isomorphism. For instance, when exploring *Tetris* blocks, we find that they can be regarded as isomorphic with regard to function and operation, as they are all grouped under the abstract representational label of "Tetrimino." As such, the more that various, disparate entities in cognitive toys and DCGs are grouped together, the higher the degree of abstraction that is used to examine those entities. Just as we can abstract entities within a system to discover isomorphic entities, we can do the same across systems.

Taking the above into consideration, isomorphism then refers to similarity correspondence between structural (representations and their organization) and/or operational patterns of a toy and those of a digital game. Therefore, DCGs that are based on the same cognitive toy can also be isomorphic, either structurally or operationally or both.

4.2.4 Game Rules

Although a digital game can be described in terms of a hierarchy of systems, it can be difficult to determine the functionality of a game based solely on such a description. This is especially true when the functionality in question is exactly how one plays the game and what is required to either win or successfully complete the game. Thus, a systems-based description of a game can be complemented with a rules-based description. The rules of a game define its functionality in an explicit and unambiguous manner.

Part of the functionality includes the representational and operational subsystems, specifically: how current and future game states are represented, how possible actions are represented, and the possible interactions available to the player. Much of this information is already contained in the systems-based description but worded differently. However, the rules also define underlying details of the flow of the game, such as the starting and ending conditions of the game, winning and losing conditions for the player, as well as game pieces and their starting state. This information is highly relevant for understanding the functionality of a game but is often not well expressed in a systems-based description. Clear and unambiguous rules are useful to designers: for them to understand how the game should operate and to avoid design-related problems early in development. Although the rules of a game should be unambiguous, they can be written with different degrees of abstraction. For the purposes of this article, two levels of rules will be discussed: abstract and concrete (see Salen & Zimmerman, 2004 for a similar concept).

Abstract rules define the functionality of a game at a high level, without too much specificity and detail. These rules define the fundamental functionality of a game but lack details regarding its visible or physical implementation. Thus, from reading a set of abstract rules, one understands the various pieces of a game, the conditions for winning, and what actions the player can perform, but there is no guidance for what the pieces would actually look like or specifically how an action is performed. For example, an abstract rule could be the following: "Players take turn arranging objects in an *n*-by-*n* grid." This rule is sufficient for understanding the functionality of the game but does not provide the details necessary to actually play it. Information regarding the size or shape of the grid, what objects are arranged, and how these objects are arranged is necessary for a physical or digital implementation of the game but not necessary for analysis. Since abstract rules are written at a high degree of abstraction, they are useful for determining isomorphisms between games.

Concrete rules define the functionality of a game at a moderate to low degree of abstraction. All details for the functioning and implementation of a game are contained within concrete rules, such that anyone reading the rules could both analyze the game and play an implementation of it. For example, a concrete rule corresponding to the above abstract rule could be the following: "In a 5-by-5 grid of equal-sized cells, players alternate moving colored entities from one cell to an adjacent cell." This rule is still somewhat abstract, in that it neither defines exact physical game piece nor specifies lines of software code, but it is sufficiently concrete that a physical or digital implementation could be made from it.

4.3 Process for Design of Digital Cognitive Games

This section discusses a process for the design of DCGs that uses cognitive toys as a means of initial inspiration. Thousands of cognitive toys have been developed over the years, and some are quite ancient in their conception (Connaughton et al., 2010; Danesi, 2002; Gillings, 1962; Olivastro, 1993). As such, there is a wide variety of cognitive toys available for designers. A carefully chosen cognitive toy can provide designers with a game idea that is both fun and mentally stimulating for players.

The design process we are presenting in this article is separated into five stages: (a) selecting a cognitive toy, (b) identifying and generalizing the patterns of the cognitive toy, (c) creating isomorphic abstract games from the patterns in the second stage, (d) creating concrete games from the abstract patterns in the third stage, and (e) implementing one or more of the concrete games from the fourth stage. Different games can result from this process based on designers' creativity and ideas generated at each stage, as outlined in Figure 6. The design process, outlined here, is systematic and structured as it relies on GST as its conceptual framework. It is also creative due to the freedom that designers are afforded at each stage of the process to choose the way in which they abstract details from a cognitive toy and instantiate, or make concrete, those same details. There are many ways in which designers can instantiate an abstract systems-based description or a set of abstract rules, thus resulting in many possible implementations of the resulting game. Any resulting games will be isomorphic to the initially chosen cognitive toy but may be very different from the toy at the surface or detailed level. This proposed design process has the potential to inspire designers to creatively come up with many isomorphic games out of a selected cognitive toy.

Each of the stages of the design process is discussed below, and a detailed example of this process is provided in Section 4.

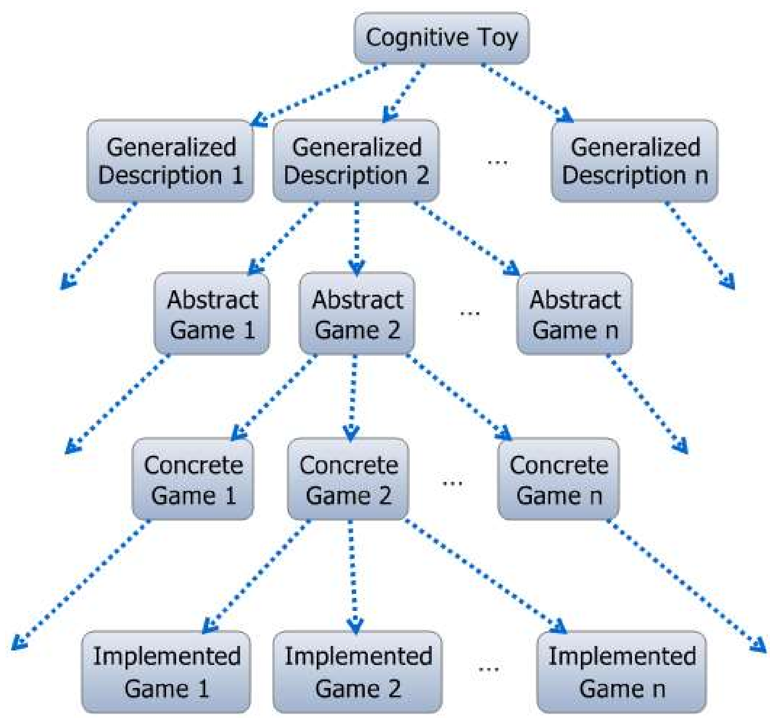

Figure 4-6: A diagram of the possibilities for the variation in resulting games at each stage of the design process.

4.3.1 Stage 1: Selecting a Cognitive Toy

In the first stage of the process, designers search through a variety of cognitive toys and choose one. The chosen toy can then be analyzed as a system, and a systems-based description is created. In other words, the entities of the toy, and their properties and relationships, are described, along with any hierarchical nesting of subsystems. Designers focus on identifying the entities of the toy and tasks that are equivalent to the representational and operational subsystems of a game.

To develop creative ideas for the representational subsystem of DCGs, designers must first identify the toy's entities and its container(s), if any. Entities inspire designers to determine the objective that the player must accomplish, the obstacles that must be overcome, and the tools available to the player. The containers inspire the design of the space which the resulting game may have and the boundaries of this space. For example, consider the *Tangram* cognitive toy, shown in Figure 2 above. In this example, the entities would be the various shaped polygons, while the container would be a shaped area in which the entities are to be placed.

To develop creative ideas for the operational subsystem of DCGs, designers must identify what the player does with the toy as a whole and with its entities. This includes the actions available to the player and the reactions or results that occur on performing each action. In other words, designers must determine how the player is able to successfully accomplish the toy's objective, the methods of interaction with the toy, and any physical artifacts with which the player must interact. The physical artifacts correspond to the entities in the representational subsystem. Designers should take note of actions necessary to accomplish the objective as well as actions that a player can take with the entities that are possible but not explicit. For example, in the *Tangram* toy, the player is able to pick up an entity, rotate it, and place it somewhere within the container. However, the player is also able to rotate the entity again or take it out of the container completely. These are actions that are possible with physical artifacts; hence, their allowance in a digital game may be necessary or at least important to note.

Although we have described this stage of the design process as though one toy is chosen to generate design ideas, designers can choose multiple toys and combine the resulting ideas.

4.3.2 Stage 2: Generalizing the Cognitive Toy's Patterns

In the second stage of the design process, designers generalize the description(s) created in the previous stage so as to develop a set of generalized patterns of the cognitive toy. Many of these patterns will be at a micro level, generalizing features specific to parts of the toy's system. For example, in the *Magic Square* cognitive toy, a micro-level pattern is that the entities each have a property which can be compared and combined. This pattern could be instantiated in different ways but is specifically instantiated in the Magic Square as "a numeric value." Other patterns will be at a macro level, generalizing features applicable to the whole of the toy's system. For example, in the *Tangram* cognitive toy, a macro-level pattern is that the task is to construct an object with a specific form. Table 1 shows a possible generalization of the patterns of the *Magic Square* cognitive toy.

It is important to note that features identified in the previous stage may already be described in abstract, general terms. For example, the container of the *Magic Square* cognitive toy could be identified as a 4-by-4 grid or a two-dimensional (2D), cell-based container. Designers do not need to wait until this stage to identify patterns, but the two stages have been separated to ensure that important details are not missed and that a high level of abstraction is used in the description of the toy for the next stage.

Table 4-1: Possible Generalization of the *Magic Square*'s Patterns.

The toy has the following generalized patterns:
• a set of unique objects
• a grid container, which is an *n*-dimensional space segmented into equal-sized cells
• individual objects can be placed in the cells of the grid container
• each object has an attribute that identifies it (we can generalize the above to state that the object has several attributes
• objects can be arranged on the grid cells
• objects' attributes can be connected (the connecting action modulates the arranging action)

4.3.3 Stage 3: Creating Isomorphic Abstract Games

In the third stage of the process, designers formalize the description from the previous stage into a set of isomorphic abstract games. An abstract game is a game in which all the rules are abstract. To formalize the representational patterns, designers must describe the entities of the game, the attributes of those entities, and the relationships among the entities. Different types of entities are described, but all descriptions should remain abstract. To formalize the operational patterns, designers must explain what actions the player performs, what entities of the game the player acts on, and why those actions are performed. These are the operational patterns described in the previous stage, but these patterns can now be modulated by environmental properties in addition to other patterns.

To create each abstract game, designers expand and elaborate on the patterns from the previous stage but keep the results abstract. The patterns previously identified should be used for inspiration, not just copied verbatim.

Table 2 shows four possible isomorphic abstract games derived from the *Magic Square* cognitive toy.

Table 4-2: Three Possible Isomorphic Abstract Games Derived from the *Magic Square* Cognitive Toy.

Game possibility	Abstract rules or descriptions
Game with objects	• the game is split into levels • each level contains an *n*-by-*n* grid ○ the size of the grid is set at the beginning of the game but it may change between levels ○ each grid cell can contain a single object ○ each grid cell may start empty or with an object in it • the player chooses a unique object from a set of objects and arranges that object on the grid • when the objects in each row are combined, the combination must satisfy some criteria ○ the same occurs for each column, and each main diagonal ○ if all combinations satisfy the same criteria, the level is complete • each time a level is completed, the size of the grid increases • when the player completes all the levels, the game ends and the player wins
Game with objects, plus time limit	• the game is split into levels • each level contains an *n*-by-*n* grid ○ the size of the grid is set at the beginning of the game but it may change between levels ○ each grid cell can contain a single object

	○ each grid cell may start empty or with an object in it • the player chooses a unique object from a set of objects and arranges that object on the grid • when the objects in each row are combined, the combination must satisfy some criteria ○ the same occurs for each column, and each main diagonal ○ if all combinations satisfy the same criteria, the level is complete • each time a level is completed, the size of the grid increases • when the player completes all the levels, the game ends and the player wins • each level contains a time limit, with the limit dependent on the size of the grid ○ the timer is set to its maximum value at the start of the level and counts down to a specific minimum value ○ when the timer reaches the minimum value, the game ends and the player loses • when the player completes all the levels, the game ends and the player wins
Game with progressive objects, plus time and score	• the game is split into levels • the player starts the game with a specific score • each level contains an *n*-by-*n* grid ○ the size of the grid is set at the beginning of the game but it may change between levels ○ each grid cell can contain a single object ○ each grid cell may start empty or with an object in it • the player chooses a unique object from a set of objects and arranges that object on the grid • when the objects in each row are combined, the combination must satisfy some criteria ○ the same occurs for each column, and each main diagonal ○ if all combinations satisfy the same criteria, the level is complete • each time a level is completed: ○ the player's score is increased by a value specific to the difficulty of the level completed and the amount of time remaining ○ the size of the grid increases by a set amount, up to a maximum value ○ when the size of the grid reaches its maximum value, the grid becomes an *n*-by-*n*-by-*n* grid of a smaller *n* • each level contains a time limit, with the limit dependent on the size of the grid ○ the timer is set to its maximum value at the start of the level and counts down to a specific minimum value ○ when the timer reaches the minimum value, the game ends and the player loses • when the player completes all the levels, the game ends and the player wins

4.3.4 Stage 4: Creating Concrete Games

In the fourth stage of the design process, designers create a set of concrete games. A concrete game is one in which designers instantiate the representational and operational patterns or rules of one of the abstract games from the previous stage into a set of concrete rules. As suggested above, the concrete rules created in this stage should be sufficiently detailed such that a digital game can easily be implemented from them. In other words, it should be clear from the set of concrete rules exactly how the game is played; however, the aesthetic details of the game need not to be clear, as they can be implemented in different ways.

In this stage, a possible UI implementation based on these rules should be clear, and the look and feel of UI entities can be considered but need not be fully detailed. Similarly, the theme, narrative, setting, and images for the concrete game can be considered but do not have to be completely established. In the abstract game, a theme, narrative, or setting is only included if it is relevant to the functioning of the game.

The concrete rules developed in this stage can be written such that they do not include technology- and platform-specific details. For example, a digital game designed for a personal computer may include controls that utilize a mouse, but a mouse does not exist in a handheld game console. The rules for the concrete game could be written for any kind of pointing device, such that a mouse would apply but so would a touch screen, leaving the rules platform independent.

Table 3 shows three possible concrete games derived from the second abstract game in Table 2, namely, *game with objects, plus time limit*.

4.3.5 Stage 5: Implementing Games

In the fifth, and final, stage of the design process, designers instantiate the concrete rules into a fully implemented digital game. This includes the development of images, audio, narrative details, layouts of each level or area of the game, and software code. The details of this stage are beyond the scope of this article, as it is closer to that involved in the development of any software project based on a specific design.

Table 4-3: Possible Concrete Games Based on the *Game With Objects, Plus Time Limit* Abstract Game.

Game possibility	Concrete rules or descriptions
Color Crystals	• the game is split into levels • each level contains a grid with square-shaped cells ○ the grid starts at a size of 4-by-4 ○ every fifth level, the grid increases in size by 1, up to a maximum size of 10-by-10 ○ each grid cell can contain 1 of a set of 10 possible color crystals, but a cell can also be empty

	○ at the start of a level, 10% of the grid cells will have a crystal • in each level, with a grid size of *n*, the player is given exactly *n* crystals of *n* different colors, that is, with a grid size of 4 the player is given 16 crystals: 4 crystals of 4 colors each ○ some of those crystals may already be placed on the grid ○ the player is randomly shown one of those crystals beside the grid ○ the player must select in which cell of the grid to place that crystal ○ when the crystal is placed, the next crystal is shown ○ the next crystal shown may be of the same color • when each grid cell is filled with a color: ○ if each row, column, and the two main diagonals of the grid are filled with one of each color then the player successfully completes the level • if a row, column, or one of the main diagonals contains 2 of the same color, then the game ends and the player loses, regardless of how many cells remain empty • each level contains a time limit that counts down by 1 every second ○ the timer is set to its maximum value at the start of the level, initially being 2 minutes ○ the timer's maximum value is increased by 2 minutes whenever the grid increases in size ○ when the timer reaches 0, the game ends and the player loses • when the player completes Level 35, the game ends and the player wins
Number Swapper	• the game is split into levels • each level contains a grid with square-shaped cells ○ the grid starts at a size of 5-by-5 ○ every third level, the grid increases in size by 1, up to a maximum size of 20-by-20 ○ each grid cell can contain one number between 1 and 400 ○ each cell is filled with a number at the start of a level • each level will have a numeric value, specifically chosen for that level ○ the available numbers must provide a correct solution • the player can select any number on the grid and swap its position on the grid with any adjacent number • during the level, if the sum of all the numbers in a row, column, or one of the main diagonals is equal to the numeric value chosen for the current level, then that row, column, or diagonal is lit up ○ when all rows, columns, and the two main diagonals are lit up, the player successfully completes the level • each level contains a time limit that counts down by 1 every second ○ the timer is set to its maximum value at the start of the level, initially being 5 minutes ○ the timer's maximum value is increased by 3 minutes whenever the grid increases in size ○ when the timer reaches 0, the game ends and the player loses • when the player completes Level 50, the game ends and the player wins
Egg Pusher	• the game is split into levels • each level contains a grid with square-shaped cells ○ the grid starts at a size of 5-by-5 ○ every tenth level, the grid increases in size by 2, up to a maximum size of 11-by-11 ○ each grid cell can contain one egg or barrel with 1 out of 10 possible colors • in each level of the game

	○ the top left grid cell contains a player-controlled character ○ five of the grid cells contain an egg ○ the remaining grid cells are filled with barrels ○ in each level after the first, one more egg is used instead of a barrel • each level contains a time limit that counts every second ○ for every 45 seconds that pass, one of the eggs on the level will crack ○ an egg can crack three times before no further cracks develop ○ when all the eggs on the grid are cracked three times, the next time a crack would develop all the eggs break, the game ends, and the player loses • the player directs his or her character in any of the four cardinal directions ○ when the character moves, it pushes any eggs or barrels in front of it in the direction of the movement ○ an egg or barrel pushed off the edge of the grid will wrap around to the other side of the grid in the same row or column ○ if the character is adjacent to an egg with three cracks, and pushes that egg, then the egg breaks and the game ends with the player losing • the player can change the color of his or her character to any of the colors available for that level • during the level, if any row, column, or one of the main diagonals contains one barrel or egg of each color, then that row, column, or diagonal is lit up ○ when all rows, columns, and the two main diagonals are lit up, the player successfully completes the level ○ the color of the player's character is included as well • when the player completes Level 50, the game ends and the player wins

4.4 An Integrated Example of the Design Process

In this section, we present an integrated example of how one might apply the design process to create two isomorphic DCGs[13]. The two games seem very different on the surface, but at an abstract level they are isomorphic. The example games are intentionally small in scope, as there is insufficient space to explain the rules of games with larger scopes.

The first three stages of the process are included below. Following that, the remaining two stages are included in sections specific to the game that is under consideration.

---

[13] It is important to note that many more DCGs can be derived from the chosen cognitive toy.

4.4.1 Choosing a Cognitive Toy

A popular cognitive toy is one shown in Figure 7 (Moscovich, 2000). This toy involves nine octagons. Each octagon has a fixed number of colored slices. The octagons can be rotated but not moved. The objective of this toy is to rotate the octagons such that each pair of adjacent octagon slices has the same color. Using GST, we can analyze this toy as a system and identify its components (see Table 4).

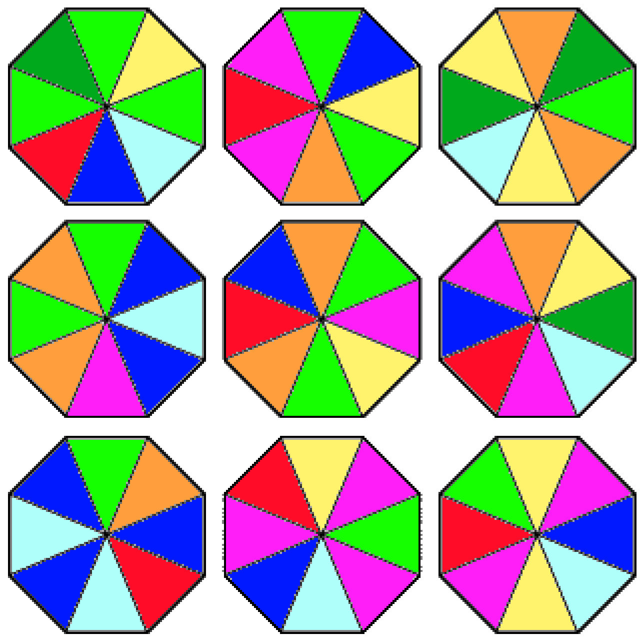

Figure 4-7: A cognitive toy adapted from Moscovich (2000, pg 158).

Table 4-4: Components of the octagon toy when analyzed as a system.

Entity	Properties	Relationships
Container	3 cells x 3 cells in size	N/A
Octagon	8 colored slices	Adjacent to other octagons
		Adjacent slice colors match

4.4.2 Generalizing the Cognitive Toy's Patterns

The components of the toy, as described above, the interaction available to the player, and the overall goal of the toy can be abstracted to a high or very general level. The patterns are then organized in terms of representational and operational features. The representational features include the following:

- Multifaceted entities (derived from multicolored polygons)
    - Multiple values of an attribute are encoded in the entity (derived from different colors)
    - Attribute values of two adjacent entities can match, creating a “matched” relationship between them (derived from same color)
- An n-by-n, cell-based container, into which the composite entities are arranged (derived from 3-by-3 grid)

The operational features include the following:

- The player can transform an entity (derived from rotating the polygons)
    - This changes which attribute values are compared between entities, and thus creates or destroys a “matched” relationship
    - When all adjacent entities have a “matched” relationship, a correct solution is obtained

4.4.3 Creating an Abstract Game

Once general patterns have been identified, they can be used to create rules for an abstract game. One possible set of abstract rules is the following:

- The game is divided into multiple levels
    - Each level is composed of several multifaceted objects
    - These objects are placed into a 2D, cell-based container
    - Relationships are formed between nearby objects when there is a specific correspondence between their attributes
    - The goal of each level is to create a specified relationship between nearby objects

- The player can transform each object, one at a time, to change its relationships with nearby objects

The rules in the abstract game use the generalized patterns of the toy, "multifaceted entity," and the necessary relationships. The division of the game into levels is a new rule that does not correspond to any generalized pattern above. This rule is just one example of how designers do not have to be restricted to the general patterns of the toy and can be creative in how they design the game.

4.4.4 Digital Game 1: Control the Flow

In this subsection, the last two steps in the design process will be discussed for a sample digital game, titled *Control the Flow*.

4.4.4.1 Concrete Game

Using the abstract rules in Section 4.3, one set of concrete rules can be the following:

- The game is divided into multiple levels
    - In each level there is a 2D grid
    - Four kinds of objects are arranged into this grid: a gate, a lever, grass, a water channel
    - Water can flow through water channel objects into adjacent water channels
    - Gates are opened, to allow water to flow through a channel, or closed, to prevent such flow
    - A lever will open or close gates of the same color as the lever
    - An avatar is placed into grid
- The player navigates an avatar through the level

  - An avatar can move among grid cells that contain closed gates or grass
  - When an avatar is adjacent to a lever, the lever can be manipulated to open or close gates
- The objective of each level is to open gates such that water flows from one edge of the grid to the opposite edge

Since the avatar, representing the player, can move across closed gates, pulling various levers will create paths to other levers. As a result, the player needs to determine the state in which each lever should be in order to complete each level.

The set of abstract rules above have been instantiated into this set of concrete rules in the following manner. The 2D cell–based container is instantiated as a 2D grid. The gates and levers are instantiations of multifaceted objects. The facets of a gate object form its state: whether it is opened or closed. Similarly, the facets of a lever object form its state. Gates and levers have a relationship whereby the state of a lever affects the state of various gates. Gates also have a relationship with water channel objects, in that water can flow through an open gate. In the abstract rules, the objective is to create a specified relationship. Similarly, in these concrete rules, the objective is to create the right relationships between levers and gates, so that the appropriate relationships can be formed between water channel objects.

Figure 4-8: A screenshot from the digital game *Control the Flow*.

4.4.4.2 Implemented Game

With the concrete rules completed, the digital game *Control the Flow* can be implemented. One way in which the concrete rules can be instantiated into a digital form is shown in Figure 8. The grid is not explicit, but is instead implicit in the arrangement of objects. All the other objects have an explicit form, as can be seen in Figure 8.

An example of playing the game, based on Figure 8, is as follows. Water needs to flow through the green gates or the purple gates in order to reach the bottom of the level. The player cannot reach the switch to open the green gates, unless she or he first closes the red gates. Thus, the player must move her or his avatar toward the red switch, close the red gates, move toward the green switch on the right, and then open the green gates. This will complete the section of the level currently shown in the screenshot, though more may need to be done to successfully complete the level.

4.4.5 Digital Game 2: Space Industries

In this subsection, the last two steps in the design process will be discussed for a second sample digital game, titled *Space Industries*.

4.4.5.1 Concrete Game

Using the abstract rules in Section 4.3, one set of concrete rules can be created:

- The game is divided into multiple levels
    - In each level there is a 2D grid
    - In some cells are arranged space colony objects or trade route objects
- Trade route objects between two space colonies create a trading relationship between them
- Space colony objects also have industries that provide goods
    - Each industry demands a specific good to function and produces a specific good
    - Colonies that have a trading relationship share produced goods
    - A colony is satisfied if each demand is met
- The player can change some industries on some colonies
    - In doing so, the player can change which goods are produced on a colony and thus whether trading colonies have their demands met
- The objective of each level is to ensure that each colony is satisfied

The set of abstract rules in Section 4.3 have been instantiated into this set of concrete rules in the following manner.

The 2D cell–based container is instantiated as a 2D grid. The space colony objects are instantiations of multifaceted objects. The facets of a space colony are its industries, the goods the colony produces, and the goods the colony demands. Whether a colony is satisfied is dependent on the state of other colonies with which it has a trading relationship. This is an instantiation of the "specified relationship" in the abstract rules.

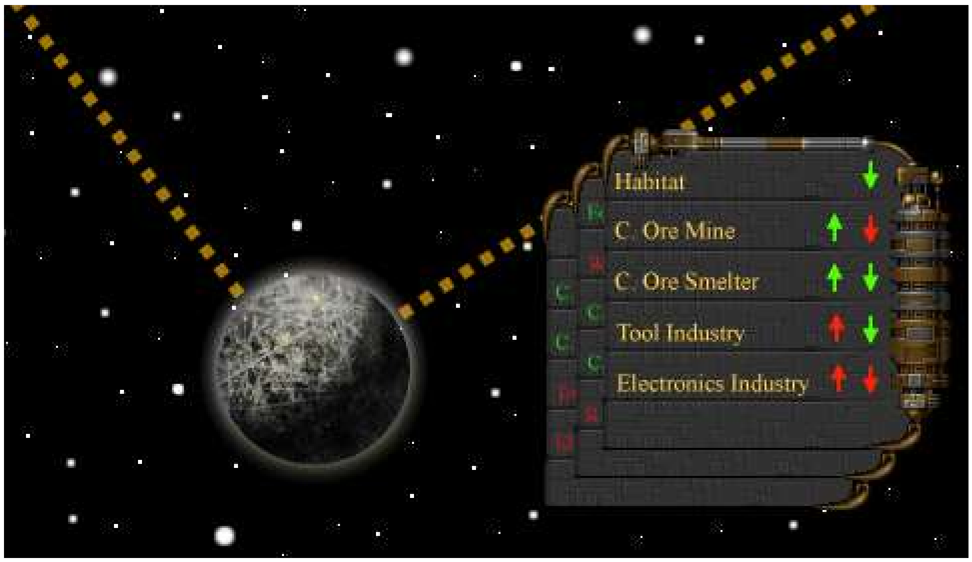

Figure 4-9: A colony in Space Industries that trades with two nearby colonies. Note that the colony demands two imports, and it exports two goods that nearby colonies do not demand.

4.4.5.2 Implemented Game

With the concrete rules completed, the digital game *Space Industries* can be implemented. One way in which the concrete rules can be instantiated into a digital game is shown in Figure 9. One space colony is shown, along with its industries. The goods demanded are shown as the second list, hidden behind the industry list. The goods produced are shown as the third list, hidden at the bottom.

An example of playing the game, based on the screenshot in Figure 9, is as follows. The colony in the figure demands food, water, common ores, common metals, and rare metals to be satisfied. The colony produces common ores, common metals, tools, and electronics. However, the demand for water and rare metals has not been fulfilled. In addition, the colony is producing two goods that are not fulfilling the demands of any colonies with which it has a trade route. One of these goods is electronics. Although the player cannot add an industry to this colony that would create rare metals, it could change the electronics industry into a resort. This would remove the demand for rare metals, create a demand for tourists, and stop the colony from producing electronics, which are not fulfilling another colony's demands anyway. If a nearby colony produces tourists but not rare metals, this would create a relationship that fulfills more demands of the colony. As such, changing the electronics industry to a resort would move the player closer toward satisfying this colony, and thus completing the level successfully.

4.5 Summary and Future Work

This article discusses a preliminary process for designing DCGs, using the concepts of isomorphism and cognitive toys. This process facilitates a structured and systematic approach to design, yet encourages the creativity of designers. Designers first choose a cognitive toy and analyze it from the perspective of GST. Then, they create a description of the toy deriving its general patterns, use those patterns to create a set of isomorphic abstract games, choose one of the abstract games and instantiate it into a set of concrete games, and finally choose one of the concrete games and implement it on a digital platform. At each stage of the process, designers are free to use an extensive amount of creativity. This results in a wide variety of possible games inspired by a single chosen cognitive toy. The inspired games can look very different with regard to their surface features but can be isomorphic at the abstract level of their description, that is, deeper levels of structure and operation.

To enhance this design process, a classification of cognitive toys, through systematic analysis, is still required. Such classification would greatly assist designers in the first two stages of the process. Additionally, further exploration of the various components of a DCG would aid designers at all stages of the process. Most important, further research is required to investigate the impact of different design choices made at different stages of the process on the final DCG. A design decision made on one component of a DCG can affect the resulting experience of playing the game (Hunicke, LeBlanc, & Zubek, 2004). To ensure that the game provides the intended experience, designers must be able to understand the implications of their design decisions. This requires empirical studies to evaluate the effect of different design decisions on the quality of the game.

While the design process presented in this article is simple to follow, it can potentially inspire designers to use simple cognitive toys that people enjoy in order to come up with countless novel and engaging DCGs. Finally, it is important to note that the process is flexible enough to allow designers to blend the features of different cognitive toys into created DCGs.

Chapter 5: Toward a Science of Interactivity for Cognitive Gameplay: Design of the Core Mechanic

This chapter has been submitted to the *International Journal of Cognitive Technology* and is currently under review. Its format was changed to match the format of this dissertation, and the references were moved to the end of the dissertation. In addition, figure numbers are relative to chapter numbers. For example, "Figure 5-1" is the first figure of Chapter 5 and the figure is labeled as such. However, this figure is referred to as "Figure 1" in the text of the chapter. The same is true for tables. In addition, when the phrase "this paper" is used, it refers to this chapter.

5.1 Introduction

In recent years there has been increasing interest in using computer games for purposes other than mere entertainment. One such purpose is to engage the player[14] in some form of high-level cognitive activity, such as critical thinking, decision making, learning, planning, or problem solving (Gee, 2007; Spires, 2008). Of these high-level activities, computer games that engage the player in learning are especially popular among researchers (e.g., Barab et al., 2005; Gros, 2007; Ke & Grabowski, 2007; Linehan et al., 2011; Sedig, 2008). When it comes to designing games for this purpose we should carefully consider the role of cognitive gameplay.

Generally speaking, gameplay refers to the experience that emerges from the player interacting with the game (Salen & Zimmerman, 2004). Cognitive gameplay refers to the cognitive component of the gameplay experience: the cognitive processes, tasks, and activities that emerge from the interaction between the player and the game (Sedig & Haworth, 2014). Since most computer games engage the player in some form of cognitive activity (Adams, 2010; Costikyan, 2002; Gee, 2007; Haworth & Sedig, 2012), most computer games have cognitive gameplay. However, the quality of cognitive

---

[14] Throughout the paper the term 'player(s)' is used in a generic sense to refer to any class of player (e.g., learners, students, teenagers, adults). Its particular meaning depends on the context in which it is used.

gameplay differs between games. This quality influences the overall enjoyment of the player (Cox et al., 2012; Connolly et al., 2012) and, more importantly, the player's engagement in high-level cognitive activities such as learning and problem solving (Sedig, 2008; Sedig & Haworth, 2014). For instance, consider a computer game that is intended to support the player in learning about biology. In such a situation, the cognitive activity of learning must be intimately connected to play, so that the learning is essential rather than incidental. Hence, when the player interacts with the game, the resulting cognitive gameplay would include learning about biology. Although this may happen, the quality of the cognitive gameplay could still be low. For instance, while playing the game, the player may not develop a deep understanding of the material; thus, the resulting learning would be relatively shallow. In this case, the game should be redesigned so that it enables, supports, and facilitates deeper and more effortful learning (e.g., Sedig et al., 2001; Sedig, 2008). In other words, it should be redesigned so that the quality of its cognitive gameplay is higher.

Since cognitive gameplay is an emergent phenomenon, we cannot design it directly. Instead, we design the game so that the desired cognitive gameplay can emerge. In other words, the design of cognitive gameplay is second-order design. While many game components can influence gameplay (e.g., content, graphics, goals, narrative, mechanics, rules, etc.), the interaction component has a much greater effect than the others because gameplay emerges from it (Adams, 2010; Salen & Zimmerman, 2004). In designing interaction though, we are particularly interested in the game's core mechanic. The core mechanic is the set of essential interactions that are repeatedly performed in a cycle, and it is primarily from the core mechanic that gameplay emerges (Campbell et al., 2008; Salen & Zimmerman, 2004). This cycle comprises the main dialogue between the player and the game, and is the primary channel through which a dialogical relationship is established and mutual causal influence occurs (Brey, 2005; Kirsh, 2005).

Through the interaction enabled by the core mechanic, the player and the game are coupled together into an integrated cognitive system. Therefore, the core mechanic is the nucleus of the cognitive system. It is from the interchange between the player and the game, sustained by the core mechanic, that cognitive gameplay emerges (e.g., high-level cognitive activities like learning and problem solving). As such, the core mechanic can be considered the *epistemic nucleus* of a game. Consequently, the manner in which the core mechanic is designed has a significant influence on the quality of cognitive gameplay. As the player engages with the game through the core mechanic, the quality of this interaction affects the performance of high-level cognitive activities and hence the quality of cognitive gameplay. This quality of interaction can be referred to as interactivity. By having the suffix 'ity', interactivity denotes the quality or condition of interaction (Sedig et al., 2014). As such, in this paper, interactivity refers to the quality of the cognitive coupling between the player and the game. The better the game is in mediating high-level cognitive activities, the higher its degree of interactivity.

In order to effectively design cognitive gameplay, it is important to examine the core mechanic and how its operational features contribute to interactivity. There is a shortage of research, however, that systematically examines how design decisions influence the interactivity of computer games (i.e., the quality of the cognitive coupling between the player and the game). It is known that playing computer games can enhance the performance of cognitive activities (Blumberg & Ismailer, 2009; Connolly et al., 2012) and that design features can affect cognitive processes (Bedwell et al., 2012). Yet, the design of computer games is often not well informed by research in human-computer interaction and the cognitive and learning sciences (Rambusch, 2010; Turner & Browning, 2010). At this stage of the development of research, due to the complexity of game design, there are still no comprehensive models for the design of computer games that emphasize and focus on cognitive gameplay (Dondlinger, 2007; Haworth & Sedig, 2012). Similarly, there is a lack of clarity when it comes to the design of interaction and interactivity in this context (Domagk, 2010; Kücklich, 2005). As a result, computer games are being designed without a systematic understanding of cognitive gameplay (e.g., Elliott et al., 2002). One area of research that has received some attention is educational computer games. There is a large body of evidence suggesting that games can be used to promote, facilitate, and enhance learning (Barab et al., 2005; Dondlinger, 2007; Alessi & Trollip, 2001; De Aguilera & Mendiz, 2003; Gee, 2007; Sedig, 2008; Squire, 2006). Learning theories developed over the past two decades have placed increasing emphasis on higher-order thinking and high-level cognitive activities, such as problem solving and planning, as vital components of learning (Jonassen, 2011). Therefore, the ideas presented in this paper are relevant and applicable not only to regular computer games but also to educational, learning, and serious games.

This paper presents a framework, INFORM (INteractivity design For the cORe Mechanic), that can facilitate systematic thinking about, and inform the design of, the core mechanic with respect to its interactivity and, hence, the desired quality of cognitive gameplay. In doing so, the INFORM framework makes a contribution towards a science of interactivity for computer games. The structure of this paper is as follows: Section 2 examines some necessary background concepts and terminology. Section 3 identifies and characterizes twelve elements of the INFORM framework that affect interactivity. Section 4 provides an integrated example that demonstrates the application of the interactivity framework in a design scenario. Finally, Section 5 provides a summary and discusses some areas of future research.

5.2 Foundational Background Concepts

This section examines some concepts and terminology necessary for presenting and developing the framework. First, we will present some recent research in cognitive science followed by a definition of the games in which we are interested. Afterward, we will provide a more detailed characterization of the information content of games, interaction, the core mechanic as the epistemic nucleus, and interactivity.

5.2.1 Cognition

Over the past half-century, cognitive scientists have largely departed from the model of humans as discrete information-processing systems. Toward the end of the 20th century, researchers began to increasingly promote the idea of cognition as being fundamentally influenced by the environment in which one is situated (e.g., Brown et al., 1989; Lave & Wengner, 1991; Salomon, 1993). In contrast to earlier models, cognitive activities were no longer considered as processes whereby decontextualized information was simply transmitted to a person; rather, cognitive activities were viewed as situated within specific contexts and embedded within particular social and physical environments. Evidence within cognitive science research also began to suggest that cognition was not only situated, but also embodied and distributed across the brain and its external environment (Clark & Chalmers, 1998; Hutchins, 1995). In other words, not only is cognition influenced by social, cultural, and contextual factors, but also by objects and by the ways in which people interact with them.

Research in cognitive science has demonstrated that objects external to the brain and body can be deeply intertwined in cognitive processes. For instance, Kirsh and Maglio (1994) studied people playing the game of *Tetris*, and discovered that cognitive processes during gameplay were extended into the external environment through the performance of epistemic actions—actions performed to facilitate mental operations rather than to achieve physical or pragmatic goals. For example, participants would often rotate and translate Tetris shapes within the game—not to achieve the pragmatic goal of placing the shape in a desired location—but rather, to facilitate mental computation. Moreover, the study determined that it was quicker, easier, and less costly, in terms of attention and memory for participants, to operate on the shapes in the game than to operate on them in the head alone. In other words, the study suggested that the manner in which the participants interacted with objects within the game had a significant impact on their cognitive processes.

Further research into human cognition has demonstrated that the external environment not only mediates and facilitates cognitive processes, but is an integral component of what can be understood as an extended and distributed cognitive system (Clark & Chalmers, 1998; Hutchins, 1995). That is, cognitive processes extend into the external environment and are distributed among the brain, body, and external entities such as people and artifacts. One consequence of these newer models of human cognition is that the unit of analysis of cognition is no longer restricted to the brain or even the body alone—rather, it must include external entities that extend and distribute cognitive processes.

When the player interacts with a game to solve problems or make decisions, cognitive processes are distributed across the player and the game. This distribution results in a coupled cognitive system that involves both the player and the game (see Clark &

Chalmers, 1998; Brey, 2005). Such a coupling enables high-level cognitive activities to emerge from the interactions among the components of the system—that is, the player and the game. Consequently, the unit of analysis of problem solving, decision making, reasoning, and other such activities must be the *player-game cognitive system*. In other words, it is insufficient to consider the game or the player in isolation to understand how games support, for example, learning or problem solving. Although HCI researchers have been incorporating recent models of human cognition into HCI literature (e.g., see Brey, 2005; Dourish, 2004; Hollan et al., 2000), little effort has been made to do so in the context of computer games (Rambusch, 2010).

5.2.2 Computer Games

For many years game researchers and developers have discussed, debated, and attempted to identify the essential characteristics of games (e.g., Costikyan, 2002; Crawford, 1994; Suits, 1990). After identifying common features from some of the more popular definitions, Salen and Zimmerman (2004) claimed to have constructed a definition of a game that captures its essential characteristics: “A game is a system in which players engage in artificial conflict, defined by rules, that results in a quantifiable outcome” (p. 80). In this paper, we will use their definition of a game.

A computer game is a subset of games; it is a system, it engages the player in non-real or artificial conflict that is defined by rules, and it has a quantifiable outcome. In addition, it operates on an interactive, electronic, and computational device or platform. The actual hardware or software used for this platform is largely irrelevant as far as the definition of a computer game is concerned. For instance, a computer game could be implemented on a personal computer, tablet computer, game console, or mobile device. The essential characteristic of a computer game, and its key distinction from non-computer games, is that computational technology mediates the interaction between the player and the game. Other specific terms have been used in the literature (e.g., video games, digital games, console games), but these can be considered synonymous with the term computer game. For the rest of this paper, unless otherwise stated, the term ‘game’ refers to a computer game.

Although non-computer games can be part of a player-game cognitive system, computer games offer greater possibilities for distributing cognition, because computational technology is highly malleable. With computational technology, we can design interactions and challenges that are difficult or impossible to create in non-computer games. For example, consider a game in which the player must arrange various 3D objects of different shapes and sizes to construct a larger 3D object in the fewest number of steps. If we wanted to allow the player to resize some of the objects or construct the larger object anyway she wants, this would be rather trivial to implement in a computer game, but very difficult to implement in a non-computer game. Likewise, it would be trivial in a computer game but difficult in a non-computer game to place certain restrictions on the player, such as: the order in which the objects can be used, the number of times an object can be arranged, and whether objects can be rotated. Many such possibilities may influence the manner in which the player thinks while playing the game. Hence, we can more easily create computer games with a much wider range of possibilities for supporting the performance cognitive activities.

5.2.3 Cognitive Gameplay

The term gameplay can refer to the interaction that occurs between the player and the game (Ang, 2006). However, it can also refer to the player's subjective experience that arises from this interaction. To more clearly indicate when gameplay refers to this experience, some researchers are using terms such as gameplay experience (e.g., Ermi & Mayra, 2005) or game experience (e.g., Poels et al., 2007). In this paper, we will use gameplay to mean the player's subjective experience that arises from her interacting with the game.

Gameplay is a composite construct, in that the experience can decomposed into specific dimensions (Poels et al., 2007). Many of these dimensions focus on the emotional or aesthetic side of the experience, such as immersion, tension, or flow (Ermi & Mayra, 2005). There is also the cognitive dimension of the player's experience, which we call cognitive gameplay in this paper. This dimension includes the cognitive processes and operations that occur within the player-game cognitive system (Sedig & Haworth, 2014).

Cognitive gameplay includes high-level cognitive activities in which the player is engaged (e.g., problem solving, planning, learning) as well as lower level cognitive tasks and processes. For example, consider the game *KAtomic*[15] in which the player must arrange atoms to form a given molecule (see Figure 1). To some extent, the cognitive gameplay of this game involves learning molecule structures. However, it also involves problem solving since the player must determine the best way to arrange the atoms and the order in which to move them. Cognitive gameplay is also influenced by the components of the game with which the player must perceive, understand, and/or interact (e.g., Callele et al., 2010). Although these components are important to consider for cognitive gameplay, they are part of the game and not necessarily gameplay. For instance, the atoms, target molecule, and maze in *KAtomic* influence cognitive gameplay since they are the objects with which the player must think.

Figure 5-1: A screenshot of the game *KAtomic*.

Designing gameplay means designing the player's experience. Hence, designing cognitive gameplay means designing the player-game cognitive system. If we want to create a game for learning, for example, we can design a game with cognitive gameplay

---

[15] Copyright 1998-2012 KAtomic Team, released under the GNU General Public License (GPL) version 2.

that emphasizes learning. However, as stated previously, gameplay is an emergent phenomenon, meaning that it cannot be designed it directly. We must design it indirectly through designing the components of the player-game cognitive system. Two of these components that are important are content and interaction.

5.2.4 Information Content of Games

When considering cognitive gameplay that emphasizes learning, researchers often make a distinction between gameplay that serves to entertain and that serves to educate (e.g., Lee & Peng, 2006). This tendency for distinction highlights the implicit assumption that cognitive gameplay that includes learning is inherently not entertaining. As a result, the design approach to learning is often one of finding the best way to combine the entertainment aspect of a game and the desired cognitive goals (e.g., promoting learning) (Ritterfeld & Weber, 2006). Moreover, such goals are often content-focused—that is, the primary focus is on the content that should be delivered to the player—thus placing content at the heart of design (e.g., see Fisch, 2005; Moreno-Ger et al., 2008). Even when researchers consider cognitive activities other than learning, the design of cognitive gameplay remains focused on content such as puzzles or other mental challenges (e.g., Callele et al., 2010; Connolly et al., 2012; Cox et al., 2012).

This approach leads to an inflation of the role of content in the design of cognitive gameplay. Much research suggests that other factors are at least as important when it comes to engaging, developing, and enhancing the player's cognition. For example, research in cognitive science has demonstrated that the manner in which content is represented, rather than the content *per se*, significantly influences cognitive processes (e.g., see Cox & Brna, 1995; Larkin & Simon, 1987; Zhang & Norman, 1994). In fact, from the perspective of the player, the *representation is the content*—that is, since the only access the player has to content is through visual representations, there is a unity of meaning between the content and its representation (Cole & Derry, 2005). It is not adequate, then, to provide content to the player without considering the manner in which the content is represented. The representational component of games requires careful consideration.

During gameplay, the player engages with content through representations that are displayed at the visually perceptible interface of the computer game.[16] Since the content is represented in a visual manner, we will refer to the representations as visual representations (VRs). VRs can encode content intended to engage the player in problem solving, learning, and other cognitive activities, such as mathematical functions and historical information, and can also encode other information, such as the current game state and possible actions the player could perform. In other words, the visually perceptible interface of any computer game is comprised entirely of VRs. For example, consider again the game *KAtomic*. In a screenshot of the game above (Figure 1), we can identify various VRs and the type of information that they encode. Some VRs encode game content, such as VRs for individual atoms, walls, and the molecule which the player needs to create. Some VRs also encode educational content, such as the chemical composition and form of specific molecules. There are also VRs that encode possible actions, such as the arrows on which the player can click so as to move an atom in a specific direction. Some of the VRs act as containers, to visually group other VRs on the interface. When all of these are considered, one can see how the entire interface of this game is comprised of VRs.

5.2.5 Interaction

Although games for learning have historically had components that serve a representational function (e.g., tiles that represent numbers or other concepts), the representations were static. Computational VRs, however, due to their malleable nature, have the potential to be highly interactive. Making VRs interactive can enhance their expressiveness by enabling the player to engage in a discourse with the underlying information in a dynamic fashion (Sedig & Sumner, 2006; Sedig & Parsons, 2013).

Although it is widely agreed upon that interaction is beneficial, its characterization is often vague and encompasses all kinds of phenomena that involve the player and the

---

[16] Digital information can also be represented and communicated to the player through auditory, tactile, and other modalities. However, such considerations are outside the scope of this paper.

game. As a result, it is commonly unclear as to what exactly is being discussed when the word interaction is used. In a broad sense, interaction refers to a reciprocal active relationship between the player and the game. However, the interaction between the player and the game can be discussed at different levels of granularity: from low-level interface events, to interaction techniques, to interaction patterns, to specific tasks and knowledge discovery activities. At the highest level, interaction refers to pedagogical and philosophical issues dealing with the overall relationship between the player and the content, such as whether the game offers the player a behaviorist or a constructionist form of interaction, whether learning is in a situated context, whether there is cognitive apprenticeship, and so on. At lower levels, interaction refers to low-level cognitive and visual tasks and action considerations, such as whether and how a game allows the player to rearrange tiles, transform shapes, move through a game space, or assign behavior and/or properties to game entities. At the lowest level, interaction can refer to physical actions and events, such as mouse clicks and drags. In this paper, we are mainly concerned with lower-level interaction patterns and their role in the core mechanic of games. By ‘interaction’ here is meant the player performing an action and the interface of the game reacting to the action. The reciprocal influence resulting from action and reaction is what connects the player to the game, and a strongly coupled cognitive system is formed when the game’s actions and reactions are operationalized appropriately.

There are many variations of games, rules, input and output techniques, and hardware and software platforms. This has resulted in a huge number of interactions and techniques for implementing them. Consequently, characterizing individual interactions at this level is an unmanageable task. Characterizing interactions at a higher level, however—at the level of patterns rather than techniques—allows a manageable set of interaction patterns to be identified and discussed in a consistent manner. Using this approach, a number of techniques and variations can be categorized according to their common features (Sedig & Parsons, 2013). For example, consider the following three interactions: in a strategy game, such as *Age of Empires*, the player designates a character to be a forager; in a mathematical game, the player binds some value to an algebraic entity; and in a sports game, such as *FIFA Soccer*, the player assigns a defensive strategy to the whole team. At the level of particular implementation details and input techniques these three interactions are distinct. At a higher level, however, there is a shared characteristic: the player interacts with VRs to assign a feature (e.g., value, function, behavior) to them. Accordingly, all three interactions can be categorized and given a common label, such as *assigning*. Each interaction, then, is an *instance* of the assigning pattern. This approach characterizes interactions at a level that is not dependent on any particular technology, game, platform, or technique—that is, it characterizes interactions as general patterns. Viewing individual interactions at the level of general patterns allows for a common characterization of the core mechanic; a common method of analyzing the structural elements of individual interactions that comprise the core mechanic; and, a common vocabulary for conceptualizing and discussing the emergence of interactivity from the operational features of the core mechanic.

5.2.6 Core Mechanic: The Epistemic Nucleus

All games have mechanics (Adams & Dormans, 2012; McGuire & Jenkins, 2009). However, there is no clear agreement among industry practitioners and researchers as to what constitutes the mechanics of a game (Sicart, 2008). The term 'game mechanics' is used as a broad construct that includes such things as rules, methods, feedback, interactions, player behaviors, game objects, and algorithms (e.g., see Adams, 2010; Hunicke et al., 2004; Lundgren & Björk, 2003; Rouse, 2005). Although what constitutes the mechanics of a game is not clear, there is typically a *core mechanic* that can be identified in any game. This core mechanic has been defined as the "patterns of repeated behavior" (Salen & Zimmerman, 2004) and the "essential interactions which a player repeats during play" (Campbell et al., 2008). In other words, the core mechanic of a game is the continual pattern of interaction in which the player and game are engaged. Although this provides some clarity as to what constitutes the core mechanic, further explication is required. As was mentioned earlier, interaction between the player and the game can be analyzed at many levels of granularity. In the context of the core mechanic, however, interaction refers to low-level patterns of action and reaction. That is, the player performs an action and the game responds in some way. This pattern of action-reaction is then repeated again and again to form a cycle, and this cycle is what constitutes the core mechanic. For example, consider the game *Breakout*. In this game, the player moves a paddle to either the left or the right to intercept a ball that is bouncing around the game space. This action-reaction pattern is repeated again and again, and makes up the core mechanic of the game. As another example, consider the *Super Mario Brothers* game. In this game, the player moves the character Mario around on the screen. Mario can walk, run, and jump to different heights. The player can also perform the occasional special move to more easily defeat enemies or overcome obstacles, such as launching a small fireball. Thus, the core mechanic of this game is composed of walking, running, jumping, and occasionally performing a special attack.

The core mechanic of any game, then, consists of a set of interactions that repeatedly occur between the player and the game. Hence, it is mainly through the core mechanic that the player acts upon the information content of the game, is engaged in critical thinking, and performs cognitive activities pertaining to the game. As such, the core mechanic is the epistemic nucleus of cognitive gameplay. For effective design of cognitive gameplay, then, it is important to examine how the operationalization of the core mechanic affects interactivity.

5.2.7 Interactivity

Interactivity has been discussed in various domains, such as media and communication studies (e.g., Bucy, 2004; Jensen, 1998), advertising and marketing (e.g., Liu & Shrum, 2009; Yoo et al., 2010), human-computer interaction (e.g., Burgoon et al., 2000; Svanaes, 2000), educational and learning technology (e.g., Aldrich et al., 1998; Sedig & Liang, 2006), and computer games (e.g., Kücklich, 2005; Lee et al., 2006). Characterizations of interactivity, however, remain vague and inconsistent. Kücklich (2005) notes that, soon after its inception, the concept of interactivity was appropriated by marketers and political-interest groups, leading to “an inflation of the term’s meaning that eventually emptied it of all analytical value” (p. 232). Other researchers have suggested that the concept of interactivity lacks an underlying theoretical model, is vague and blurry, and lacks a common language (Bucy, 2004; Sedig & Liang, 2006; Kiousis, 2002; Mann, 2002). Although interactivity is sometimes discussed in the context of computer games, very little focus has been given to characterizing it in a thorough manner at a consistent level of granularity. For example, Lee et al. (2006) define interactivity in computer games as a "communication process" that is characterized by "turn-taking, feedback, and choice behaviors" (p. 263). While discussing interactivity in the context of cognitive gameplay, Lieberman (2006) suggests that good interactivity design "gives them [players] a great deal of control, involves them in active decision making, and provides continuous feedback...". Although such characterizations are not necessarily incorrect, they do not explicate the features of interactivity in an analytical manner, and are therefore not suitable to guide design and evaluation of cognitive gameplay in a systematic fashion. It is very likely that researchers and practitioners would benefit from having access to frameworks that elaborate on the elements which give structure to and bring about the interactivity construct, particularly in the context of cognitive gameplay.

5.3 INFORM: Interactivity Design for the Core Mechanic

One of the challenges in discussing interactivity is that the terms 'interaction' and 'interactivity' are often used loosely and interchangeably. Although they are related, a distinction can be made between them. Interaction refers to a reciprocal active relationship between the player and the game. As discussed in section 2.5, an interaction in this paper refers to the player performing an action and the interface of the game reacting to the action. By having the suffix 'ity', then, interactiv*ity* signifies the quality or condition of interaction (Sedig, 2009). As such, in this paper, interactivity refers to the *quality of the cognitive coupling* between the player and the game. The better the game is in facilitating cognitive activities and desired cognitive processes, the higher its degree of interactivity. Notice too that the quality of the cognitive coupling is the same as the quality of cognitive gameplay. Hence, designing high quality cognitive gameplay can be achieved through carefully and consciously designing interactivity appropriately. This distinction between interaction and interactivity is important; even though a game is interactive, if the quality of interaction is not good, cognitive processes will not be supported effectively. To provide an example, consider a game intended to facilitate learning about world geography in which the player must move shapes that represent countries into their correct locations. To achieve the goal, the player repeatedly moves the shapes around the interface until they are arranged properly. The interaction can be operationalized in many different ways, however. For example, the player could choose a shape and drag it to a different location. Alternatively, the player could type a command to the game to have the shape moved to a particular location. Furthermore, there could be time constraints on the interaction or there could be no time limit. In all previous cases, the interaction remains constant (i.e., arranging shapes); its quality changes, however, according to the manner in which the interaction is operationalized. Despite knowing that these changes in quality affect cognitive gameplay, unless we have a framework to structure and systematize the design of interactivity it will be difficult to consciously design cognitive gameplay.

As mentioned in Section 2.6, the core mechanic of any game is comprised of a number of interactions that occur in a cyclical fashion. Any single interaction that is part of the core mechanic has a number of elements that collectively give it structure. Moreover, each element has different operational forms, and varying the operationalization of these elements determines the quality of the interaction. As an individual interaction has both an action and a reaction component, the elements that affect interactivity can be categorized into action elements and reaction elements. Twelve elements, six for action and six for reaction, have been identified and are characterized within the INFORM framework. The elements of action are: agency, flow, focus, granularity, presence, and timing. The elements of reaction are: activation, context, flow, spread, state, and transition. Table 1 provides a list of these elements.

In what follows, we will characterize and discuss each element and some possible forms in which each can be operationalized. Examples of existing games will also be given. Where applicable, studies that have investigated a particular element are discussed. The following three points should be considered while examining this section: 1) the terms used here may not be found in the literature since a framework, such as INFORM, does not currently exist; as a result, some of the terms have been devised for the INFORM framework. 2) The studies that are discussed did not necessarily use the same terminology as we are using here, even though they are examining the same phenomenon. 3) As this is a young area of research, not every element has been studied; hence, we cannot include studies for every single element. Future research will need to conduct systematic studies on how different forms of these elements affect cognitive gameplay. In addition to existing research in game studies, such an endeavor should be well informed by research in HCI design, motivation and experience design, and cognitive and learning sciences. Finally, it must be stated that the integration of different forms of these elements, and their aggregate mutual influences within the core mechanic, contribute to the emergence of interactivity. As such, the INFORM framework can facilitate systematic thinking about how design decisions influence interactivity. By identifying and characterizing each element and its operational forms, the INFORM framework is a step toward the development of a science of interactivity for cognitive gameplay.

Table 5-1: Elements of interaction in the INFORM framework.

Action	Reaction
agency	activation
flow	context
focus	flow
granularity	spread
presence	state
timing	transition

5.3.1 Elements of Action

There are at least six elements that make up the action component of an interaction. Each of these is discussed below.

5.3.1.1 Agency

This element is concerned with the metaphoric way through which the player expresses an action. In other words, this element deals with the manner in which the player articulates an action. There are at least two forms of agency: verbal and manual. In verbal agency, the player expresses an action using her ‘mouth’, as though she speaks to a VR, such as by typing a command into a console. In manual agency, the player expresses an action using her ‘hands’, as though she is reaching into the interface and grasping and manipulating a VR, such as using a mouse cursor to drag a VR. An example of the different forms of agency in the context of the mathematical puzzle game, *Tower of Hanoi* (Petković, 2009; Sniedovich, 2001), is presented below.

Tower of Hanoi consists of three pegs and a number of disks of different sizes. The disks can slide onto any peg. To start the game, the disks can be placed in a conical shape by stacking them on top of each other in ascending order of size on one peg. The goal of the game is to move the entire stack to another peg. The rules of the game are as follows: only one disk may be moved at a time from one peg to another; only the upper disk from one of the pegs can be moved and be put on top of another disk that may already be present on another peg; and, finally, no disk may be placed on top of a smaller disk.

Svendsen (1991) conducted a study to investigate the effects of agency on thinking and problem solving in the context of a game based on the *Tower of Hanoi*. Two versions of the game were created that differed in terms of agency. One version had manual agency, in which participants used a mouse to click-and-drag discs from one peg to another. The other version had verbal agency, in which participants typed a command to move a disc from one peg to another. The results of the study showed that verbal agency was more conducive to reflective thought while problem solving. Participants who used the game with verbal agency made fewer mistakes. This study suggests that the form of agency of an interaction can affect the player's cognitive processes—in this case, verbal agency being more conducive to reflective cognition. As such, the operationalization of this element should be considered carefully when designing the core mechanic.

5.3.1.2 Flow

This element is concerned with how an action is parsed in time. There are two main forms of flow: discrete and continuous. An action with discrete flow occurs instantaneously in time and/or is punctuated. An action with continuous flow occurs over a span of time in a fluid manner. As a simple example of different forms of action flow, let us assume that there is a game in which the player has to move a VR from its current location to a new location. Using discrete flow, the player can click at the new location to which the VR is to move and it will appear at that location. Using continuous flow, the player can drag the VR to the new location. An example of a study about the cognitive effects of different forms of flow, both for action and reaction components of interaction, is provided later in Section 3.2.3.

5.3.1.3 Focus

This element is concerned with the VR to which the player attends in order to act upon a VR of interest—that is, the focal point of action. There are two main forms of focus: direct and indirect. Using direct focus, the player acts directly on the VR of interest. Using indirect focus, the player acts on an intermediary VR to affect the VR of interest. An example of these different forms of focus is provided in the following study.

Sedig et al. (2001) conducted a study using several versions of a computer game. The game was based on the *Chinese Tangram* puzzles (Slocum, 2007)—see Section 3.1.4 for a description of this puzzle. The game involved children learning transformation geometry concepts by solving tangram puzzles. Two versions of the game differed in terms of how the participants operated on the polygons. In one version, called Direct Object Manipulation, they acted upon the polygons directly (i.e., direct focus) in order to move them to the desired screen locations. Thus, upon selection of desired transformation functions, participants would click on a desired polygon to move it. In the second version, called Direct Concept Manipulation, participants acted on VRs that represented transformation geometry functions such as a VR of the arc of rotation, a VR of the line of reflection, and a VR of the translation vector. In order to move a tangram shape, participants would adjust the parameters of these VRs and, once satisfied, apply it to the polygon (i.e., indirect focus). The polygon would then move. Two groups of participants were used. Pre- and post-tests, containing 51 transformation geometry questions, were administered to these groups before and after using their respective version. The results showed that different forms of focus affected learning significantly. Even though there was no significant difference in their pre-test results, the group that had used the Direct Concept Manipulation version (i.e., indirect focus) performed significantly better on the post-test than the group that used the Direct Object Manipulation version (i.e., direct focus). The authors concluded that the focal point of the players' action affects their attentive processes. In this study, in the first version, the participants' focal point of attention had been the polygon shapes, paying little attention to the transformation operations. In the second version, their focal point of attention had been the transformation geometry VRs, hence paying more attention to how to adjust those than paying attention to the polygons. As can be seen in the study, the operationalization of focus in an interaction should be considered carefully when designing the core mechanic.

5.3.1.4 Granularity

This element is concerned with the steps that the player needs to compose an action. There are two main forms of granularity: atomic and composite. An action which has atomic granularity cannot be decomposed into steps. In other words, an atomic action is itself the only step. An action which has composite granularity can be broken down into more than one step.

Let us assume we are designing a game that is based on the *Chinese Tangram* puzzles (Slocum, 2007). A tangram puzzle includes a 2D outline or silhouette and seven 2D polygons of various shapes and sizes. The objective is to arrange (i.e., move) the polygons so that all of them fit inside the outline without overlapping each other. Each puzzle differs in the outline given but the same seven polygons are always used. This puzzle can be used to explore both geometric properties of shapes as well as transformation geometry operations. In a physical environment, the player does not have to think about how to translate, rotate, and reflect the polygons to move them to desired locations in the outline. However, in the context of a computer game, the player needs to do these explicitly. Since there is more than one polygon, the player has to select a polygon and then perform a transformation operation on it to move it from one location to another (i.e., one of translation, rotation, or reflection). Let us examine how this action can be designed using different forms of granularity. If the action has atomic granularity, then it can be operationalized in the following fashion: the player clicks on a polygon and drags it to translate it; the player double-clicks on a polygon to reflect it; the player right-clicks on a polygon to rotate it clockwise or counter-clockwise by a pre-determined amount. As can be seen, in this scenario, the player does not get to see or examine the finer granularity of these operations. If the action has composite granularity, then one way in which it can be operationalized is presented next. First, the player clicks on a polygon to select it. Then the player clicks on a button to select the transformation operation (e.g., translation). This results in a representation of the vector of translation appearing in the play area of the screen. Then the player adjusts the parameters of the vector (i.e., its magnitude and direction). Once satisfied with the settings of these parameters, the player commits the action by clicking 'run' or 'go'. As can be seen, in this scenario, the player can examine the finer details of the operation. Both forms of granularity can then be operationalized for moving the other polygons in the context of varied tangram puzzles. Therefore, this element of interaction can play a role in how the player engages with information in the context of a game.

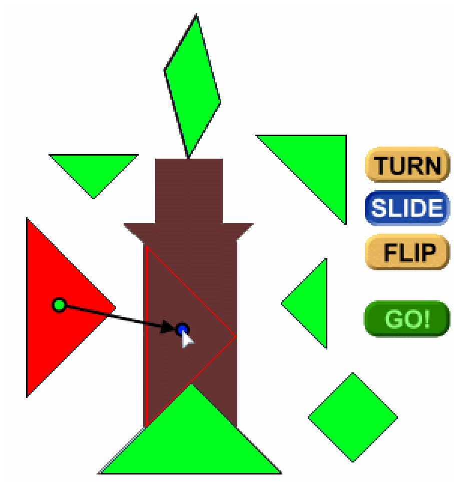

Figure 5-2: A game of *Chinese Tangram* puzzles. The player is moving a polygon. The translation (slide) operation is selected, and the player is currently adjusting a VR of the translation vector.

5.3.1.5 Presence

This element is concerned with whether the game advertises the existence of an action to the player. It has two main forms: explicit and implicit. In explicit presence, the availability or existence of an action is explicitly advertised to the player. In implicit presence, even though an action is present, its availability is not advertised to the player, and it is assumed that the player knows that it exists.

As an example, imagine that there is a game in which the player arranges the position of blocks. The VRs of the blocks could be designed to indicate that the player can act on them, such as having an image of a set of arrows on the block. There could also be a label on the interface which explains how to move the blocks. In both cases, presence is explicit as the player is made aware that she can perform an action on a block. If no label or instructions were provided, and the VRs for the blocks do not suggest that the player can act on them, then presence is implicit.

In terms of potential cognitive effect, there is a tradeoff between implicit and explicit presence. With explicit presence, too many VRs notifying the player of action possibilities can be overwhelming. Making too many things explicit can lead to confusion. On the other hand, with implicit presence, extra cognitive load may be placed on the player by requiring her to remember the existence of action possibilities. The player may also have to search for actions or information regarding how to act, spending time on tasks unrelated to the game. Thus, designers should consciously decide which form of presence is appropriate in a particular context, so as to influence cognitive processes in a desirable manner (e.g., not to burden the player with cognitive load where it is not beneficial).

5.3.1.6 Timing

This element is concerned with the amount of time available to the player to compose and/or commit an action. There are at least two main forms of timing: self-paced and system-paced. In self-paced timing, the player is not constrained by any time limitations for composing and committing an action. Using this form of timing, the player has as much time as needed to think about and examine a situation before committing an action. In system-paced timing, the player has a set amount of time to compose and commit an action before the system acts, usually involving negative consequences, such as the action being automatically canceled, the player being prevented from acting again on the same VR, or the player having to restart the current puzzle or game level.

For example, consider again the game *KAtomic*. The core mechanic of this game involves moving atoms within the game space, which is an instance of the arranging pattern. The timing element of this interaction can be operationalized to be self-paced or system-paced. If timing is self-paced, then the player has no time restrictions. She can take as much time as she wants to move an atom and to choose the order in which to move. Given that the player has no time constraints, the game provides the opportunity for the player to plan carefully all of her moves before committing an action. If timing is system-paced, it could be operationalized in the following manner. An atom is always selected to be moved, and the player can change which atom is selected. After the lapse of thirty seconds, the atom automatically moves in a random direction. The timer would restart each time an atom is moved. This form of timing forces the player to act quickly, because she may not want the atoms to move in an undesired direction.

Another example of the operationalization of the two forms of timing can be seen in the mathematical puzzle game, *Tower of Hanoi* (Petković, 2009; Sniedovich, 2001), described in Section 3.1.1. When designing the digital version of this game, the timing of the action of moving the disks can be either self-paced or system-paced. If it is self-paced, then the player has unlimited time for each action (i.e., each move). It can be seen that this form of timing provides the player with the opportunity for reflective cognition and look-ahead planning and decision-making. If it is system-paced, depending on how fast each action should be performed, then the player's thinking is constrained by the system's pace. However, it should be noted that the system-paced form of timing can be used to promote quick decision-making and planning by the player.

5.3.2 Elements of Reaction

There are at least six elements that make up the reaction component of an interaction. Collectively, these six elements can also be referred to as feedback. Although feedback has been discussed by a number of researchers and game developers, the term is often used in a manner that does not distinguish between levels of interaction and interactivity. Feedback at the level of an individual interaction can be better understood by characterizing the structural elements that make up the reaction component of the interaction. In this paper, reaction refers to effects of an action that are visibly-perceptible at the interface and not effects that may occur internally in the game and are hidden from view of the player. Furthermore, an action often results in the interface going through fluctuations before the reaction process is completed and the interface reaches equilibrium. Therefore, some of the reaction elements discussed here are concerned with the reaction during fluctuation while others are concerned with the reaction as the interface reaches an equilibrium and the reaction process is completed.

5.3.2.1 Activation

This element is concerned with the commencement of reaction after the player has committed an action. There are at least three forms of activation: immediate, delayed, and on-demand. In immediate activation, the reaction occurs instantaneously after an action is committed. In delayed activation, an action is committed and then a span of time passes before the reaction occurs. In on-demand activation, the reaction only occurs once the player requests it. An example of these three forms is provided in the following study.

Sedig and Haworth (2014) conducted a study of the relationship between activation and cognitive gameplay in two puzzle-based games: *Laser Dilemma* (LD) and *Temple Swap* (TS). In LD, the player arranges operators to redirect the path of a laser beam with the goal of directing the laser beam through a series of goal points. In TS, the player is given a set of tiles with different symbols. She can swap adjacent tiles to form rows or columns of the same symbol, with the goal of ensuring each tile is in at least one such row or column. Both games had two versions; in the first version, the operational form of activation was immediate (LD-I and TS-I) while the operational form of activation was on-demand in the second version (LD-D and TS-D). For example, in LD-I the laser would fire to test the solution of the puzzle immediately after each operator was placed. In LD-D though, the laser would only fire when the player clicked on the fire button. Similarly, in TS-I the tiles would immediately move once the player has selected two tiles to swap, while in TS-D the player can select tiles to swap but no swapping would actually occur until after she has clicked on the go button.

In the study, participants were randomly assigned to one version of one game and played it. Both during and after playing the game, participants described the cognitive gameplay they experienced. These experiences were then organized and analyzed. The results of the study indicated that on-demand activation promoted more effortful and long-term thinking while immediate activation prevented this or encouraged less effortful and shorter-term thinking. In LD-I, participants tried to plan carefully but reported difficulty doing so due to activation being immediate. In TS-I, participants did not try to plan carefully but merely moved tiles around and eventually discovered a solution. In LD-D, participants were able to plan carefully and felt that the game required them to think this way. Although delayed activation was not tested, this study suggests that on-demand activation is more appropriate for encouraging effortful and reflective forms of thinking while immediate activation is more appropriate for encouraging effortless and quick or reactionary forms of thinking.

5.3.2.2 Context

This element is concerned with the general context in which VRs exist as the interface reaches equilibrium. There are two forms of this element: changed and unchanged. Before an action is committed, a VR exists within some context. During the reaction process, that context can change or it can remain the same. Once the reaction process finishes, if a VR is in a different context, then the form is changed.

For an example, we will again use the computer game *KAtomic*. When the player performs an action to move an atom, no context change occurs. Let us now imagine a different scenario in which a context change does occur. Whenever the player clicks on an arrow to move an atom in a certain direction, the game changes to a first-person perspective of the maze centered on the atom. The movement of the atom is shown from this perspective as it goes through the maze, and then the game returns to the original context after the movement is complete. The different view of the movement of the atom is a different context, due to a drastic change in the VRs of the interface.

Unexpected changes in context can have an effect on memory and processing of information. Changing a person's physical location can affect her ability to remember information (see Smith & Vela, 2001). Similarly, changing the mental context in which a person is thinking can also affect her ability to remember (Delaney et al., 2010).

5.3.2.3 Flow

This element is concerned with how a reaction is parsed in time. There are two main forms of flow: discrete and continuous. A reaction with discrete flow occurs instantaneously in time and/or is punctuated. A reaction with continuous flow occurs over a span of time in a fluid manner. For example, consider a game in which the player launches a projectile at a target. Once the launch parameters are entered, the projectile gradually moves across the screen following the path of the defined arc. In this case, the reaction has continuous flow. Instead, the projectile could appear immediately at the end point of the arc once the launch parameters are entered. In this latter case, the reaction has discrete flow. As was stated in Section 3.1.2, an example of a study about the cognitive effects of different forms of flow, for both action and reaction components of interaction, is reported next.

Liang et al. (2010) conducted a study to compare different design options of an interactive visualization tool. The testbed for this study was 3D Platonic and Archimedean geometric solids. The tool was designed to help high school students explore and make sense of the structural properties of and relations among these solids. Students needed to understand how truncating or augmenting the edges and/or vertices of solids would transform them into other solids. For instance, truncating the vertices of a cube leads to various intermediary solids, but eventually an octahedron is created. The tool contained maps which represented relationships between solids, such as how truncating the edges would transform one solid into another. The tool also contained an enlarged solid, which represented the currently selected solid on a map on a larger scale.

Based on the combinations of different forms of flow, four versions of the tool were created with different interfaces: C-C (continuous flow of action, continuous flow of reaction), C-D (continuous flow of action, discrete flow of reaction), D-C (discrete flow of action, continuous flow of reaction), and D-D (discrete flow of action, discrete flow of reaction). A multi-method empirical study was conducted to evaluate the usability of these four interfaces and their effect on learning, visual thinking, and exploration. Using C-C, participants could click on a solid in one of the maps and drag it to any position within that map (continuous action flow). As participants dragged the solid, an enlarged solid updated its form so as to match the solid at that point in the transition (continuous reaction flow). Using C-D, participants could act in the same manner as previously discussed (continuous action flow). However, as participants dragged a solid across a map the enlarged solid did not change. The enlarged solid updated itself immediately once participants released the mouse button and stopped acting on the map (discrete reaction flow). Using D-C, participants could only click on a position in a map (discrete action flow). Participants could not drag a solid as in the previous two versions. Once participants clicked somewhere on a map, a smooth animation showed the solid transform as it moved from the previous position on the map to the newly chosen position (continuous reaction flow). Similarly, the enlarged solid slowly updated its form to match the animated solid. Finally, using D-D, participants acted in the same manner as in version D-C. However, once a participant clicked somewhere on a map, no animation was played but instead the enlarged solid immediately updated its form (discrete reaction flow).

Participants were divided up such that they each used only one version of the tool. Before and after using the tool, participants were given a pre- and post-test containing multiple choice questions regarding geometry. The results showed that the form of flow affected learning: the group that used the D-D version (discrete action and reaction flow) performed significantly better on the post-test than the other groups, and the group that used the C-C version (continuous action and reaction flow) performed the worst. Participants who used the C-C version did not have to reflect much on the content they were exploring, as it made the performance of action and interpretation of reaction less demanding than the versions with discrete flow. Participants who used the D-D version needed to spend more time reflecting and planning their actions. The authors concluded that the ease and intuitiveness of continuous flow may not always be desirable, as this form of flow can be counter-productive when reflective cognition and investment of mental effort are needed in learning. Even though the reported study does not deal with a game, it nonetheless demonstrates that how the flow element is operationalized can affect the cognitive processes of the player and should be designed with care.

5.3.2.4 Spread

This element is concerned with the spread of effect that an action causes. An action can cause a change to occur in the VR of interest. However, other VRs may be affected as well. There are two main forms of spread: self-contained and propagated. A reaction which has self-contained spread only causes a change to occur in the VR of interest. A reaction which has propagated spread causes a change in other VRs on the interface. In other words, the effect of a reaction propagates such that other VRs are affected.

As an example of operationalizations of the two forms of spread, we will use a game in which the player creates visual tiling patterns composed of various polygons. The player creates a pattern by inserting polygons into the working space, one at a time. The player could insert a new polygon adjacent to an existing polygon, such that the existing polygon remains unaffected. In this case, spread is self-contained; the reaction only affects the newly inserted polygon and none of the other polygons in the working space are affected. However, the player could also insert a polygon between two or more existing ones. The newly inserted polygon would push the others away from it, so as to make room in the working space for it. In this case, spread is propagated because the reaction affects multiple polygons.

5.3.2.5 State

This element is concerned with the conditions of the interface (i.e., the interface's VRs) once the reaction process is complete and the interface reaches equilibrium. There are three main forms of state (that is, the states that VRs affected by an action can assume): created, deleted, or altered. In created state, new VRs are created which did not exist before the action was committed. In deleted state, some VRs are deleted from the interface. Finally, in altered state, some VRs' properties (e.g., their values, positions, etc.) are modified.

Let us see how the same interaction can result in different forms as its reaction element of state. As an example, let us use a game in which the player must guide a robot through a grid-based maze full of different types of objects that have different behaviors. The main interaction pattern is navigating—that is, moving on or through a representation space. For instance, the player can move the robot by 5 grid blocks. As the robot is moving, it can hit an object resulting in a change in the position of that object (i.e., altered state). As a result of a similar movement, the robot can, for instance, pass over a switch, triggering the disappearance of certain objects from the grid (i.e., deleted state). Alternatively, the robot can pass over another kind of switch, causing the spawning of new objects that are placed on the path of the robot (i.e., created state).

5.3.2.6 Transition

This element is concerned with how change is presented on a 2D display. VRs in games that are dynamic and/or interactive are spatio-temporal entities. As such, when an action causes change in them, the change can take place along both the temporal and spatial dimensions of VRs (Tufte, 1997). This makes presentation of change difficult, often resulting in the distortion of one of the dimensions. Hence, there are two general forms of transition: stacked and distributed (Tufte, 1997). If transition is stacked, changes in a VR are sequentially stacked one on top of another in time. Although the VR is visually changing over a duration of time, only the current state at one point in time is perceivable and past stages of change disappear. This form of reaction behaves like a movie, where changes to a scene are stacked in time and one scene replaces another. If transition is distributed, multiple stages of change in a VR are spatially distributed, such that they are all perceivable. Of all the changes a VR may pass through, several are chosen as snapshots. Those snapshots are then displayed as new VRs, such that the player can view them in parallel on the screen, without previous stages disappearing in time. This form of transition is similar to a storyboard, where transitional scenes are shown as separate images. Hence, while stacked transition constrains the visual change to one location, distributed transition can communicate changes in a VR over a region of space. An example of a study about the cognitive effects of different forms of transition is presented next.

Sedig et al. (2005) conducted a study to investigate how to design interfaces that can support the formation of cognitive maps of processes of change in objects (encoded as VRs)—where cognitive maps refer to mental maps of these objects and paths of change from one form to another. Cognitive maps involve three levels of knowledge: landmark (knowing the main objects), route (knowing the paths of change or transition), and survey (knowing the overall landscape of objects and paths). The investigation was conducted using 3D Platonic and Archimedean geometric solids. These solids can be morphed (i.e., changed) to each other by truncating and/or augmenting them, as was discussed in section 3.2.2. Three versions of a tool using three different interfaces were designed. Stacked interface would allow participants to interact with a 3D solid to morph it in place (i.e., stacked transition). Distributed interface would allow participants to interact with a VR that displayed the distribution of these solids in space with paths connecting solids that can be morphed to one another (i.e., distributed transition). This interface would allow participants to click on an arrow and observe how one solid changed to another along the connecting paths. Hybrid interface had the stacked and distributed interfaces integrated and coupled together as one interface (see Figure 3). This interface would allow participants to interact with either the stacked or the distributed side of the interface and observe its effect on the other side.

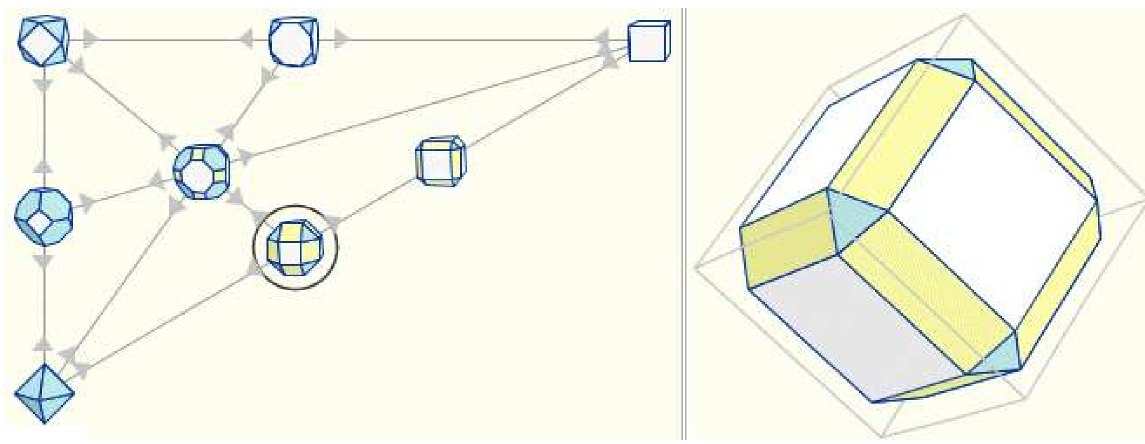

Figure 5-3: An interface example in which transition is distributed, for the map on the lefthand side, and stacked for the solid on the righthand side.

A multi-method empirical study was conducted to evaluate the usability of the three interfaces. The results of the study showed that all three interfaces supported the formation of cognitive maps, but to different degrees. The hybrid interface was the most effective of the three interfaces. The study suggested that the group that used the stacked interface were more focused on the individual solids (landmark knowledge), the group that used the distributed interface were more focused on the paths that connected the solids (route knowledge), and the group that used the hybrid were focused on both the solids and the paths (survey knowledge). This study suggests that each form of transition has its own inherent strengths and weaknesses. For instance, stacked transition avoids forcing the player to make constant back-and-forth movements of the eye, something that distributed transition does (Tufte, 1997). However, distributed transition forces the player to compare different states of an object from memory, as one cannot see multiple states of the VR simultaneously, a limitation that stacked transitions do not have. As these have different perceptual and cognitive effects, they can influence how well the player performs cognitive activities. As such, they require more systematic investigation.

5.4 Application of INFORM in a Design Scenario

In this section, a scenario is presented in which the framework, INFORM, is applied to the design of the core mechanic of a computer game. This scenario is intended to demonstrate how INFORM facilitates systematic design decisions for cognitive gameplay. To this end, we will first examine the game and its core mechanic. Next, we will systematically analyze the different ways in which the interactions that comprise the core mechanic can be operationalized. This will be done by examining the operational forms of each element that collectively give structure to an interaction. In doing so, the many possibilities for varying the structure of the core mechanic of a game to achieve different cognitive goals or ends should become apparent. In other words, designers can methodically analyze the combinatorial possibilities that the operational forms of interaction elements create in terms of design variations of cognitive gameplay. For example, if each interaction has 12 elements, each of which has at least 2 forms, the number of possible ways to operationalize an interaction is at least 2^{12}, or 4096. It should be noted that not all elements are applicable or have significant cognitive effects in every game. However, even if only 4 out of 12 elements have a significant influence on cognitive processes in a particular context, the possible combinations are still 2^4, or 16. Furthermore, as this calculation is only applicable to a single interaction, this number is increased depending on the number of interactions within the core mechanic.

Although not all elements are always significantly important, we examine each element and its operational forms in this section for the sake of thoroughness. As a result, some of the examples may seem contorted and unlikely to appear in the particular game presented here. We have included these examples, however, simply to demonstrate how INFORM facilitates a systematic design process. In addition, other design considerations, such as player motivation and enjoyment, score keeping, and graphical design, are not discussed. Although such considerations are important, they are omitted here to keep the focus on how design of interactivity affects cognitive gameplay.

The cognitive gameplay for the game in this integrated example includes spatial reasoning activities. That is, the game is intended to develop and enhance the player's ability to visualize spatial patterns and mentally manipulate them. In this game, the player must recreate a series of patterns in a step-by-step fashion. For every level of the game, the player is given a pattern to copy and a 3D geometric solid. The pattern is broken down into sections which have the same size and shape as the faces on the solid. Each face on the solid has a different image that corresponds to sections of the pattern. In other words, the pattern is a composition of the faces on the solid. The player is also given a blank working space in which to create a copy of the given pattern. To create a copy, the player rotates the solid and stamps one of its faces into a section of the working space. The face that is inserted is the one which is most prominently visible. This is repeated until each section of the working space is equivalent to the corresponding section of the pattern. Therefore, in order to create a copy of the pattern, the player needs to rotate the solid such that the desired face is the most visible one. To do so, it is necessary for the player to determine how to rotate the solid in as few rotations as possible. To play the game effectively, the player must consciously engage in a series of spatial reasoning tasks. Moreover, to achieve the intended outcome (i.e., winning the game), the player must develop her spatial reasoning abilities in order to meet increasingly difficult challenges. As the epistemic nucleus, the core mechanic has a primary role in facilitating the cognitive processes of the player. By systematically adjusting how different forms of the interaction elements are operationalized, the designer can create many different variations of the core mechanic, each of which potentially influences cognitive processes in different ways. These may then be implemented in different circumstances within a game (e.g., in different levels), or may be used to implement different versions of a game that can be empirically studied. In either case, the point is that the framework gives the designer a tool for thinking systematically about the design of the core mechanic.

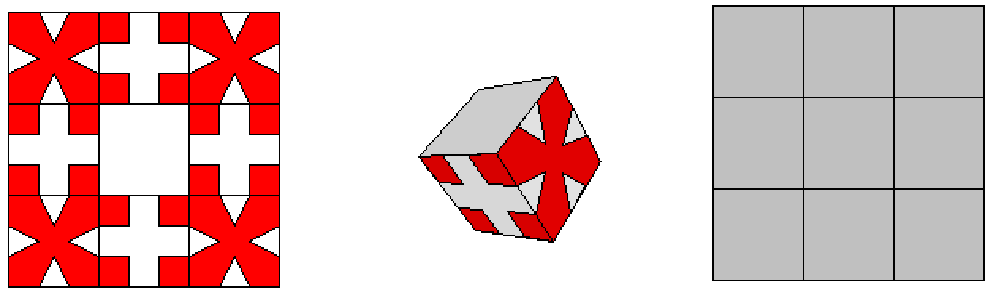

Figure 5-4: The pattern to recreate, the 3D solid the player will use, and the working space in which a copy of the pattern is created.

The core mechanic of the game described above is comprised of two interaction patterns. The first is transforming, which refers to a category of interactions in which the geometric form of a VR is changed, such as by twisting, rotating, bending, folding, or stretching. The second is inserting, which refers to a category of interactions in which a new VR is interjected into a representation space (e.g., working space in this game), such as by stamping, imprinting, or implanting. This game has only one instance of each pattern. To play this game, the player first rotates the solid—an instance of the transforming pattern. Once the desired face is reached, the player then stamps one face of the solid into the working space—an instance of the inserting pattern. These two interactions—rotating and stamping—are repeated again and again in a cyclical fashion and constitute the core mechanic of the game.

To illustrate how the INFORM framework facilitates systematic design decisions for the above-mentioned core mechanic, the rest of this section will analyze the possible ways of operationalizing the elements of the rotating interaction. Although proper design requires analysis of both the rotating and stamping interactions, including the stamping interaction in our analysis would add considerable length to this section of the paper. For the sake of brevity, only the rotating interaction is analyzed below. This method of analysis, however, should clearly indicate how a similar analysis could be done for the inserting interaction. Indeed, it should clearly indicate how any interaction within any game can be systematically analyzed. In what follows, each element of the rotating interaction will be analyzed according to how its possible forms can be operationalized.

Agency: *manual* or *verbal*. With manual agency, this interaction could be designed such that the player can rotate the solid using the mouse. For example, the player could click and drag the solid, and it would rotate as it is dragged. As another example, arrows could appear around the solid as the mouse cursor approaches it. The player could click on an arrow of the desired direction to rotate the solid in that direction. With verbal agency though, this interaction would be designed differently. For example, the player could have to type commands into a console to rotate the solid. The command "rotate left" would rotate the solid to the left. There could also be a command such as "rotate left 3", which would rotate the solid to the left three times.

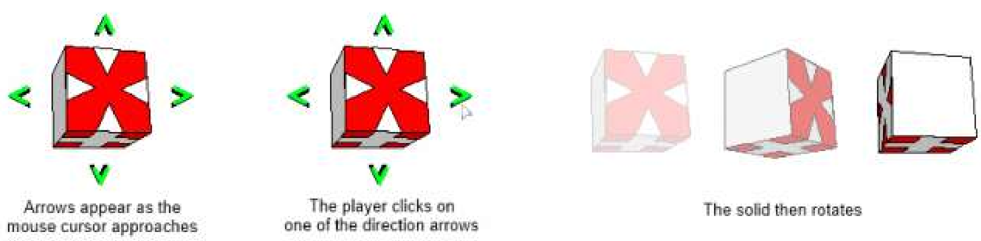

Figure 5-5: An example of rotating the solid when agency is manual.

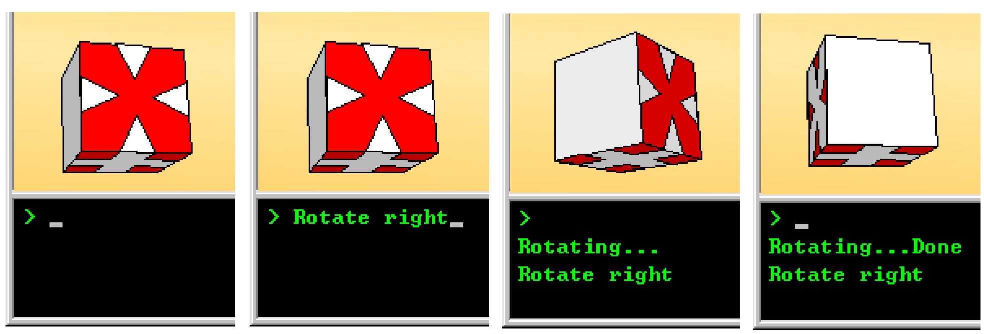

Figure 5-6: An example of rotating the solid when agency is verbal.

Focus: *direct* or *indirect*. With direct focus, the solid would be the focal point of action. As an example, the player could click on and drag the solid so as to rotate it. As the interaction is directed toward the VR for the solid, focus is direct. However, focus could be made indirect. This would require another VR to become the focal point of action. A simple example is having arrows appear around the solid. When the player clicks on an arrow, the solid rotates in that direction. This is indirect focus, as the interaction is directed toward a separate VR: one of the arrows. As another example, the player could be given the solid and a slider bar beneath it. The player cannot click on the solid, but can click on and drag the slider. As she drags the slider, the solid rotates correspondingly along one axis. For instance, if she drags the slider to the right, the solid will rotate to the right along the x-axis.

Flow (action): *discrete* or *continuous*. Consider the above examples in which the player clicks on an arrow button to rotate the solid. These are examples in which action flow is discrete, as action occurs instantaneously: the player just clicks on a button. As another example, the player could right-click on the solid to rotate it in one direction. The player could also left-click on the solid to rotate it in another direction. However, if the player could drag a slider to specify the amount of rotation, then action flow would be continuous; the action happens over a period of time in a fluid manner. Similarly, another example of continuous flow would be if the player dragged the solid itself in a certain direction to cause reaction.

Granularity: *atomic* or *composite*. With atomic granularity this interaction would only have one step. For instance, the player could just drag the solid in any direction and that would rotate it once in that direction. This is atomic granularity because the player has only performed one step: dragging the solid. With composite granularity, the interaction would have more than one step. For example, in order to rotate the solid the player would have to click on it first. Doing so would cause directional arrows to appear. The player would then click on an arrow of the desired direction and it would become highlighted. Another button would then appear over the solid to confirm the rotation, and the player would need to click this button to commit the action. Upon clicking it, the reaction would occur (i.e., the solid would rotate in that direction). In this particular example, composite granularity seems tedious and unnecessary. However, if the player could supply additional parameters to the action (such as the angle or speed of rotation) and these parameters mattered for the game then deciding between atomic or composite granularity would be a more relevant design decision.

Presence: *implicit* or *explicit*. With implicit presence, this interaction could be designed such that the possibility of rotating the solid is not advertised to the player. The shape would be visible at the interface, but the player would be required to have existing knowledge that the shape could be rotated and how to go about rotating it. With explicit presence, the possibility of rotating the solid would be advertised to the player. One obvious example is to have text below the solid stating something such as: "To rotate the solid, click and drag it". Alternatively, the solid could be wiggling with a small rotation sign attached to it to suggest the possibility of this interaction.

Timing: *self-paced* or *system-paced*. If timing is self-paced, the player can take as long as she wants to rotate the solid. However, if timing is system-paced then some time restriction is placed on the player. For example, there could be a timer that begins at 60 seconds and keeps counting down. When it reaches 0 seconds, the player loses points, as if an extra rotation was performed without effect. Every time the player performs the rotating action, the timer for this interaction can be reset.

Activation: *immediate*, *delayed*, or *on-demand*. With immediate activation, the solid would rotate as soon as the player acts. For example, the player clicks an arrow button to rotate the solid in some direction, and the solid rotates immediately afterward. With delayed activation though, the solid would not rotate until a period of time has elapsed or some other event occurs. For instance, the player clicks on the arrow button to rotate the solid to the left. After clicking on it, the solid does not rotate. Then, she clicks on the arrow button to rotate the solid upward. After clicking, the solid would rotate to the left since this was the previous action performed. Next, the player stamps the solid onto the workspace and the solid rotates upward.

Lastly, activation could also be on-demand. In this case, the player would be able to perform multiple rotations but the solid would never rotate until she clicks on a separate button to request the feedback. For example, the player could click on an arrow of the desired direction, and an arrow of the same direction would appear in a bar below the solid. This would repeat as long as she clicked on arrows, allowing her to create a sequence of rotations. Once the player has the desired rotation sequence, she clicks a 'rotate' button and this would request the feedback to occur. The reaction would then occur; i.e., the solid would rotate according to the indicated sequence.

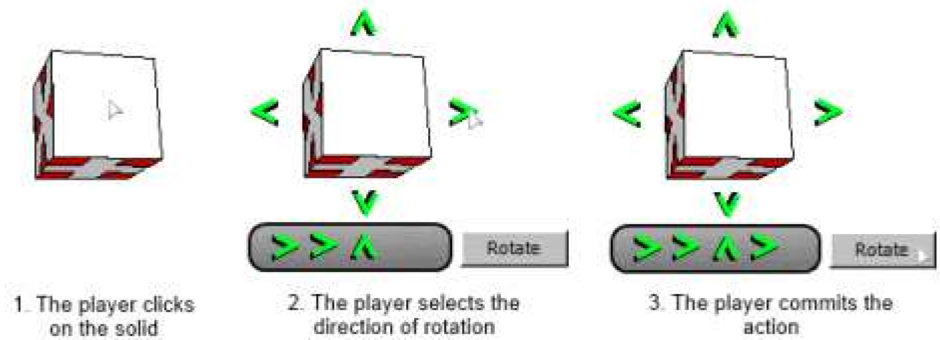

Figure 5-7: An example of multiple rotations of the solid with on-demand activation.

Context: *changed* or *unchanged*. In all of the examples thus far, context is unchanged; the context in which the interaction occurs does not change once the reaction is finished. However, the interaction could be designed in such a way that if the player rotates the solid beyond the limits imposed by the game, the game would then react by aborting the current interaction and switching to a new context to give instructions to the player about more effective forms of play. The player would then have to leave this context to return to the previous one and continue playing. This is an example of changed context.

Flow (reaction): *discrete* or *continuous*. One way in which continuous reaction flow can be operationalized is the following. The player can click on and drag a slider to rotate the solid. As she drags the slider, the solid does not change. Once the player releases the mouse button to finish dragging the slider, then the solid gradually rotates until its orientation matches that specified by the slider's position. This is an example of continuous reaction flow, since the reaction occurs fluidly over a period of time. If discrete reaction flow was desired, the following change could be made. Once the player finishes dragging the slider, the solid's orientation immediately changes to match that specified by the slider's position. It does not change gradually, but all at once. In this case, the reaction occurs instantaneously without any fluid motion. Although this operationalization may seem contorted and unrealistic, separating action and reaction may be conducive to mindful planning in certain situations (see Liang et al., 2010 for detailed examples of how these forms are operationalized and the resulting cognitive effect).

Spread: *self-contained* or *propagated*. In the case of self-contained spread, only the focal VR is affected. For instance, when the player drags the solid to rotate it, and the VR for the solid is the only VR affected on the interface, then spread is self-contained. However, in the example below in which transition is distributed, the player types a command to rotate the cube and, as a result, multiple VRs are created to display the orientation of the solid at different stages in the rotation. In this example, spread is propagated. These other VRs are created as part of the reaction, and thus the effect of the reaction has propagated to other VRs. When the player rotates the solid again, the VRs for the previous rotation will be deleted, and new ones created. From this it is clear that more than only one VR is affected.

State: *created*, *deleted*, and/or *altered*. To demonstrate how this element can be designed, a new rule is added to the game: a limit on the number of times the solid can rotate. Once this limit is reached, the player can no longer rotate the solid and must restart the puzzle. Consider the case in which the player rotates the solid by typing a 'rotate' command into a console. When the player presses enter to commit the rotating action, the solid rotates accordingly. Assume that, in addition to the solid rotating, VRs elsewhere in the interface, representing the number of performed rotations, are also affected. One possibility is that, as the solid is rotated, the color and/or arrangement of other VRs change to reflect the number of remaining rotations that are available to the player. In this case, the properties of the VRs (i.e., their colors and positions) are altered—an example of an altered state. Another possibility is that each time the solid is rotated, a VR is removed from the interface to indicate that one less rotation is available to the player—an example of a deleted state. For instance, there could be a row of small cubes, and each rotation results in one of these cubes being removed. A third possibility is that VRs representing the number of rotations are added to the interface after the performance of each rotation—an example of a created state. There may be an empty grid, for instance, and each rotation results in a small copy of the rotated solid being created and placed in the grid to signify that an interaction has taken place.

Transition: *stacked* or *distributed*. For stacked transition, imagine that the solid rotates such that previous orientations are not displayed; only the current orientation in the rotation process is displayed. In all of the above examples, transition is stacked. However, we could make transition distributed. For instance, the player types a command to rotate the solid three times to the right. The VR for the solid is then replaced by several VRs, each of which displays the solid at a certain orientation within the rotation process. All of the VRs remain on the screen, so that the player can see the different orientations which the solid had while it was rotating.

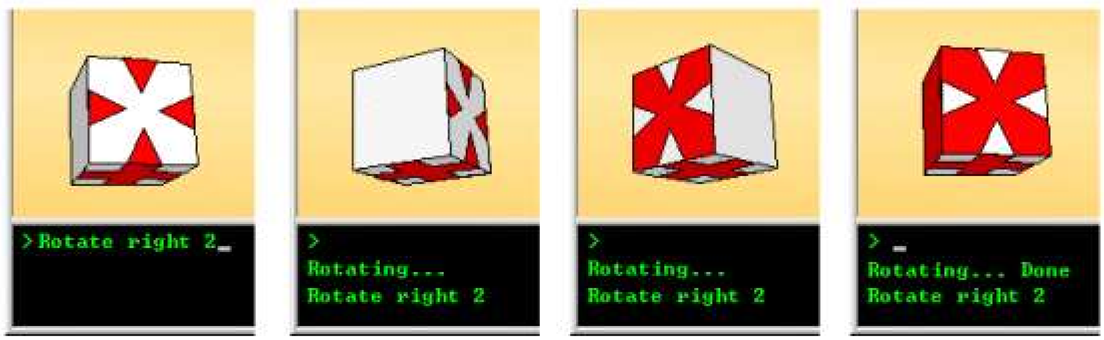

Figure 5-8: An example of the solid rotating with the stacked transition.

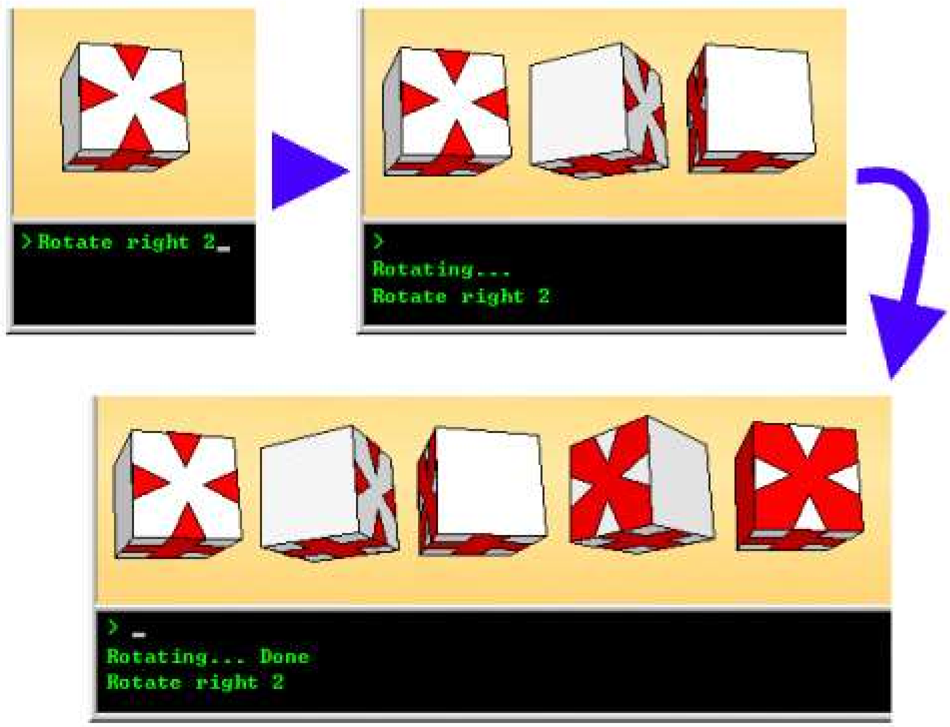

Figure 5-9: An example of the solid rotating with distributed transition.

5.5 Summary

The cognitive gameplay of a computer game is the cognitive component of the player's experience, which emerges from the interaction between the player and the game. As such, it includes high-level cognitive activities such as problem-solving, critical thinking, planning, decision making, and/or learning. Conscious design of cognitive gameplay requires the consideration of a number of issues, such as game mechanics, narrative, motivation, rules, information content, graphics, and outcomes. This paper has focused on the core mechanic—i.e., the interactions that bind the player and the game together, the cyclical performance of which results in gameplay. It is mainly the core mechanic that couples the player and the game together to form an integrated cognitive system. Through this coupling, and the reciprocal active relationship between the player and the game, high-level cognitive activities (i.e., cognitive gameplay) emerge. Therefore, the core mechanic can be considered as the *epistemic nucleus* of the game. The manner in which the core mechanic is designed affects cognitive gameplay. It is important, then, to design the core mechanic carefully and to examine how its operational features contribute to interactivity—i.e., the quality of the cognitive coupling between the player and the game.

This paper presents the INFORM framework (INteractivity design For the cORe Mechanic). This framework enables designers to systematically analyze the different ways in which the interactions that comprise the core mechanic can be operationalized. INFORM identifies and characterizes twelve elements that collectively give structure to an individual interaction, and identifies and describes some operational forms of each element. Table 2 provides a summary of these interaction elements. Designers can analyze the interactions that comprise a core mechanic according to their elements and their possible operational forms. In doing so, designers can consider many possible variations of the structure of the core mechanic that each affect cognitive processes and activities in different ways. This not only enables systematicity, but also helps to stimulate creativity in the design process. For instance, designers can vary the structure of the core mechanic for different situations within a game, different levels within a game, or even different versions of a game, each of which achieve different cognitive goals or ends. Although not all elements are applicable and/or important for every game, even if only half of the elements have a significant influence on cognitive processes in a particular context, the possible combinations for each interaction are 2^6, or 64. Without a descriptive and analytical framework, such as INFORM, it would be very difficult to consider the many possibilities of design in a systematic manner.

Table 5-2: Summary of interaction elements in the INFORM framework.

Component	Element	Concern	Forms
action	agency	metaphoric way through which action is expressed	verbal, manual
	flow	parsing of action in time	discrete, continuous
	focus	focal point of action	direct, indirect
	granularity	steps required to compose an action	atomic, composite
	presence	existence and advertisement of action	explicit, implicit
	timing	time available to player to compose and/or commit action	self-paced, system-paced
reaction	activation	commencement of reaction	immediate, delayed, on-demand
	context	context in which VRs exist once	changed,

		reaction is complete	unchanged
	flow	parsing of reaction in time	discrete, continuous
	transition	presentation of change	stacked, distributed
	spread	spread of effect that action causes	self-contained, propagated
	state	condition of VRs once reaction process is complete	created, deleted, altered

The increasing popularity and economic impact of computer games suggests that their influence will continue to grow in the foreseeable future. Consequently, there is potential for further emphasis on the design of cognitive gameplay. In addition, as recent theories of learning and instruction promote more situated and active learning strategies, games have the potential to take a more important role in educational settings. To do so, however, the design of cognitive gameplay must be well-informed by research in human-computer interaction and the cognitive and learning sciences. Frameworks such as INFORM are a step towards informing the design of cognitive gameplay in this manner.

As part of INFORM, where applicable, studies that have investigated a particular element are discussed. However, as not much systematic research has been done in this area, not all elements have been studied. Future research will need to conduct studies on these elements and how their different forms affect cognitive processes and activities (i.e., cognitive gameplay). Another line of future research can be in integrating INFORM into a larger framework that considers other aspects of game design. For example, the affective and emotional components of gameplay (e.g., motivation, immersion, flow) can be examined in relation to the operational features of the core mechanic. Furthermore, future research can establish more exhaustive design principles, guidelines, and prescriptive and descriptive frameworks and models, to develop a mature science of interactivity. By identifying and characterizing some features of the core mechanic that influence cognitive processes, the INFORM framework makes a contribution toward such a science of interactivity for cognitive gameplay.

Chapter 6: Investigating Variations in Gameplay: Cognitive Implications

This paper has been published as Sedig, K., Haworth, R., and Corridore, M. (2015). Investigating Variations in Gameplay: Cognitive Implications. *International Journal of Computer Games Technology*, *2015*, 1-16. doi:10.1155/2015/208247 It is reprinted here as per the Creative Commons Attribution License.

Note that the format was changed to match the format of this dissertation, and the references were moved to the end of the dissertation. Figure numbers have also been changed to be relative to chapter numbers. For example, “Figure 6-1” is the first figure of Chapter 6 and the figure is labeled as such. However, this figure is referred to as “Figure 1” in the text of the chapter. The same is true for tables. In addition, when the phrase “this paper” is used, it refers to this chapter.

6.1 Introduction

Computer games have now become popular and ubiquitous in our society, especially as the proliferation of games on social media and cell phones increases. At the same time, interest in using computer games in non-entertainment contexts has been increasing (e.g., Deterding et al., 2011; Habgood & Ainsworth, 2011; Spires, 2008). Games within most of these contexts are more effective when they engage the player in deep and meaningful cognition, for example, reflective learning of mathematics, solving complex engineering problems, and planning a sequence of actions in order to accomplish some goal. Therefore, we need to design these games with conscious attention given to their cognitive gameplay. Generally speaking, the term gameplay refers to the experience which emerges from the interaction between the player and the game (Ermi & Mäyrä, 2005; Salen & Zimmerman, 2004). Cognitive gameplay refers to the cognitive processes that emerge from the player interacting with the game. In other words, cognitive gameplay is the cognitive component of the experience of playing a game. The quality of cognitive gameplay is the primary factor affecting the ability of a game to engage the player in cognitive tasks and activities (Sedig, 2008). For example, if we wanted to create a computer game for mathematics education, the game should engage the player in the cognitive activity of learning specific mathematical concepts. Hence, it would be insufficient to simply embed these concepts into the game; we would also need to carefully design the cognitive gameplay so that the game is conducive to deep and conscious reflection of the embedded mathematical concepts (see Elliott et al., 2002; Sedig, 2008).

Design of cognitive gameplay is a second-order design problem (Adams, 2010; Salen & Zimmerman, 2004). We cannot directly design gameplay, since it emerges from the interaction that occurs between the player and the game. Instead, we must design the interaction afforded by the game, as it is from this interaction that gameplay emerges. In other words, we can control the resulting gameplay by carefully designing the interaction. Similarly, to design cognitive gameplay we need to design the interaction so that it facilitates, promotes, and supports the desired cognitive engagement (Haworth et al., 2013). For example, if we wanted to design a game for learning mathematics, and thus wanted cognitive gameplay that is conducive to learning, then we need to design the game's *interaction* in such a way that it enables and promotes deep and conscious reflection upon the embedded mathematical concepts.

Currently, there are no frameworks or methods for the systematic design of cognitive gameplay. Many of the existing frameworks and methods for designing gameplay tend to focus on the non-cognitive components of gameplay, such as enjoyment, immersion, or challenge (e.g., Alexander et al., 2013; Connolly et al., 2012; Isbister & Schaffer, 2008; Sweetser & Wyeth, 2005). Cognitive gameplay can influence non-cognitive components, such as enjoyment, but a framework that focuses on these other components does not inform us about how to engage the player in specific cognitive tasks or activities. Other frameworks relate specific structural features of games to learning (e.g., Aleven et al, 2010; Bedwell et al., 2012). While these frameworks are more useful for the design of cognitive gameplay, the structural features on which they focus are quite broad (e.g., narratives, evaluation, and personalization); usually they are not directed toward interaction design. In addition, these frameworks focus on the specific cognitive activity of learning and hence may not generalize well to other high-level cognitive activities, for example, decision making, planning, or problem solving.

Sedig and colleagues (2014) have developed a framework that informs the design of interactivity (i.e., the quality of interaction) at a micro-level for visual tools. Although the current context of this framework is visual tools, aspects of it may be applicable to the design of computer games. Specifically, this framework could inform the design of micro-level interactivity in a computer game and thus inform the design of cognitive gameplay to some extent (see Haworth et al., 2013). However, to further understand the systematic design of cognitive gameplay, we need to investigate the relationship between micro-level interactivity and cognitive gameplay. In a more general sense, we need a method for studying the effects of structural elements of computer games on cognitive gameplay.

Typical studies of cognitive gameplay investigate the relationship between the game as a whole and some internal change in the player (see Connolly et al., 2012; Powers et al., 2013). For example, Quiroga and colleagues (2011) conducted a study to determine whether playing *Big Brain Academy* for multiple trials would influence general intelligence test scores. The results of their study indicated that some test scores were influenced and, thus, the cognitive gameplay of *Big Brain Academy* involves general intelligence. Although these results are important and useful, they do not help us determine what component(s) of *Big Brain Academy* were responsible for this effect. For instance, we cannot conclude from this study whether the interaction afforded by the game influenced the results of the general intelligence test; other components of the game may be responsible, such as the game's content, the manner in which this content is represented, and the game's goals. Without a method of identifying and isolating components of a game, it is very difficult to determine the source of any cognitive effect. Thus, while this study is still valuable, it does not further our understanding of how to design a computer game's essential interactions in order to create some desired cognitive gameplay. Although other studies that investigate the effect of particular components of a computer game on cognitive gameplay (e.g., Sedig et al., 2001) are more useful, they are still insufficient. Their emphasis tends to be on a particular cognitive activity (e.g., knowledge acquisition), and, as previously mentioned, the results may not be generalizable to other high-level cognitive activities. The lack of generalizability is primarily due to the methods used for designing the tested computer games.

To study the design of cognitive gameplay, we need a systematic process for design that is (1) general in terms of the possible cognitive activities that can result and (2) capable of isolating specific game components for further study. The process for designing computer games presented in (Sedig & Haworth, 2012) may be applicable for this purpose, but it has not been tested. Hence, we investigated whether this process would assist in studying the design of cognitive gameplay. We conducted a simple investigation of the cognitive gameplay of two computer games. These games were designed using the process described in (Sedig & Haworth, 2012) and were designed such that structural differences could be isolated and controlled. To test whether it was possible to isolate only one structural difference, we designed these games so that the only structural difference between them was one element of interaction. When these games were played by our volunteers, differences in cognitive gameplay were observed. These differences were based on self-reported descriptions of the volunteers' experiences and their own observations of the differences between the games. The difference in cognitive gameplay could be associated with the structural differences in interaction between the two games. In other words, we found some promising evidence that (1) careful design of interaction can affect cognitive gameplay, (2) these effects can be studied in a systematic and controlled manner, and (3) the design process mentioned in (Sedig & Haworth, 2012) could be of benefit for such studies. However, this is still preliminary research.

In this paper, then, we will present the results of this investigation and some of the theoretical considerations surrounding the investigation. In Section 2 below, some of the design concepts of computer games will be discussed as well as the design process used for the two games. Following this, in Section 3, the two games and the investigation procedure will be explained. Then, in Section 4, the results of the investigation will be presented, followed by a discussion in Section 5. Some conclusions about how this investigation furthers our understanding of the design of cognitive gameplay will be presented at the end, in Section 6.

6.2 Background

In this section we will discuss some of the background concepts and terminology before we present our investigation.

6.2.1 Computer Games

Many terms have been used throughout the literature to describe the kind of games that we talk about in this paper, such as video game, digital game, or electronic game. We will simply adopt the term computer game, though the other terms can be considered as equally applicable. By computer game we mean a game that is implemented on some form of computational platform. The physical hardware used is unimportant, as the concepts in this paper equally apply to a game that functions on a personal computer, mobile device, tablet computer, or game console. However, using computational technology as the means through which interaction occurs gives computer games a greater degree of interactive potential. This is due to the interaction that can be designed for computer games as well as the removal of physical limitations inherent in non-computer games (e.g., card games, board games, and physical sports). As a result of this interactive potential, computer games also have a greater degree of cognitive potential and higher potential for more variety and depth of cognitive gameplay.

6.2.2 Cognition

Over the past few decades, researchers in cognitive science have increasingly promoted the idea that human cognition is fundamentally influenced by the surrounding environment (e.g., Brown et al., 1989; Hollan et al., 2000; Lakoff & Johnson, 1999; Salomon, 1993). Although there are cognitive processes that operate within a human, such internal processes do not simply operate on some decontextualized input. Artifacts external to a human can cause the operation of internal processes to be augmented, constrained, or offloaded. In other words, one can conceptualize the operation of cognitive processes as though they were distributed across multiple objects in an environment (Clark, 2008; Hutchins, 1995).

A more comprehensive model for cognition, in which external artifacts are taken into account, is one where cognitive processes occur within an integrated human-artifact cognitive system. This model allows us to analyze the whole cognitive system rather than merely focus on internal cognitive processes in isolation of the environment. The cognitive system most relevant to this paper is the one composed of a player and a computer game. The player is a subsystem, composed of internal cognitive processes, expectations towards games, skill with certain games, and so on. The computer game is also a subsystem, composed of various User Interface (UI) elements which the player perceives and with which she interacts. By analyzing the whole player-game cognitive system we can identify the way in which cognitive processes are distributed across the player and the computer game (for an example of this in a context outside of computer games, see Parsons & Sedig, 2014b).

For example, Kirsh and Maglio (1994) studied people playing the computer game *Tetris* and found that certain cognitive processes were extended into the game. In *Tetris*, the player arranges differently shaped blocks in a play space so as to create horizontal rows. To do this, she needs to quickly determine the location and orientation of each block and arrange it accordingly. Arranging a block involves a combination of two operations: rotating and translating. To determine the best location and orientation of a block, the player could internally visualize the result of possible operations on it. However, Kirsh and Maglio found that their participants would often operate on blocks in the game and use the result of those operations to determine the best location and orientation of blocks. Instead of internally visualizing the result of an operation the participants simply performed the operation, saw the result, and used this information to help them arrange the corresponding block. They would do so despite having limited time to act on any block and despite the potential risk of having the block end up in a location or orientation that was detrimental to their progress. Performing such operations on a block is an example of a cognitive process being distributed within the player-game cognitive system: the player offloaded her cognitive process of visualizing onto the game when she operated on blocks in the game instead of internally visualizing the same operation.

6.2.3 Interaction

The distribution of cognitive processes is enabled by the relationships that exist between elements in the cognitive system. In the case of the player-game cognitive system, the most influential relationship is the interaction that occurs between the player and the computer game. A single interaction has two components: an action that the player performs on one or more elements of the UI and a reaction from the game. By reaction is meant some perceivable change of the UI, such as an element of the UI changing its position or new UI elements being created. In playing the game, the player performs an action, the game responds with a reaction, the player perceives this reaction, and then the player performs a new action. As this action-reaction cycle repeats, the player and the game become coupled together into a two-way bilateral dialogue (see Kirsh, 2005; Salomon & Perkins, 2005 for elaboration, even though they are not in the context of computer games). Cognition emerges from this dialogue, in that the interaction engages and mediates cognitive processes and facilitates their distribution (Sedig et al., 2014).

6.2.4 Interactivity

The term interactivity refers to the quality of interaction, and therefore it means the quality of the cognitive coupling between the player and the computer game (Haworth et al., 2013). We can conceptualize an interaction in a computer game as a composite of a set of elements that collectively determine its structure. In the context of the design of visual tools, Sedig and colleagues (2014) have identified twelve such elements, which could also apply to the design of computer games; some of these elements include activation, context, flow, and focus. These elements have various operational forms that affect the quality of the cognitive coupling between the player and the game. Thus, designing interactivity of a game involves determining how the elements of its interactions should be operationalized.

For example, one of the structural elements of interaction is focus. Focus can be operationalized as direct or indirect. In a game with direct focus the player acts on a target UI object, while in a game with indirect focus the player acts on an intermediary UI object to influence a target UI object. In a study that explored designing computer games for learning mathematical concepts (Sedig et al., 2001), two computer games were developed that differed in the way in which focus was operationalized (though the authors of the study do not use the term focus). The results of this study indicated that there were differences in cognitive gameplay; specifically, the game with indirect focus led to deeper and more effortful learning of the embedded mathematical concepts than the game with direct focus. In other words, operationalizing this structural element of interaction in one way resulted in cognitive gameplay that was characterized by deep and mindful learning, while operationalizing it in a different way resulted in cognitive gameplay that promoted shallow and habituated learning. This study is also an example of how we could investigate the effect of structural differences of interaction on cognitive gameplay. However, in (Sedig et al., 2001) there was no discussion of a systematic and generalized method for the design or investigation of their games. As previously mentioned, they also focused entirely on learning (specifically acquisition of mathematical knowledge) and made no indication that their study could be generalized to other cognitive activities.

6.2.5 Cognitive Gameplay

Roughly speaking, the term gameplay refers to the subjective experience of the player while playing a game (Salen & Zimmerman, 2004). However, it can also refer to the features of a game that affect the kind of experience that the player may have (e.g., Aarseth, 2003; Adams, 2010). In this paper though, we will use the definition provided by Ermi and Mäyrä, who say that gameplay "is not a property or a direct cause of certain elements of a game but something that emerges in a unique interaction process between the game and the player." (2005, page 2) Although the experience that emerges is still subjective, interaction is what facilitates and mediates its emergence.

Cognitive gameplay is a subset of gameplay and can be defined as the cognitive processes that emerge through the bilateral dialogue between the player and the game (Haworth et al., 2013). In other words, it refers only to the cognitive aspect of gameplay. Other aspects of gameplay can be influenced by cognitive gameplay and vice versa, but the term cognitive gameplay does not refer to these. For example, the pleasure gained from aesthetic features of a game, the degree of visual immersion or engagement, and the perception of challenge are all aspects of gameplay but are not part of what we mean by cognitive gameplay. The player mindfully and deliberately determining the best position for some object, planning a path through a navigation space, devising a strategy to solve a puzzle, deciding which options to take, taking into account a concept to overcome an obstacle, creating a sequence of actions to create a structure, or trying to remember a previously failed attempt so as to avoid it are all examples of experiences captured by the term cognitive gameplay.

Since both gameplay and cognition emerge from interaction between the player and the computer game, we can design cognitive gameplay by designing the interaction of the computer game. Specifically, it is interactivity design that is most relevant for designing cognitive gameplay. This is because interactivity determines the quality of the cognitive coupling between the player and the game, and hence interactivity determines the quality of the resulting cognitive gameplay (Haworth et al., 2013; Sedig et al., 2014). Likewise, we can gain a better understanding of how to systematically design cognitive gameplay through investigating the effect of a computer game's interactivity on its resulting cognitive gameplay.

6.2.6 Investigating Cognitive Gameplay

In order to investigate cognitive gameplay, it is first necessary to identify the structural component(s) of the computer game that will be explored. Simple computer games are often best for this, since they have few structural components. For example, the computer game *Tetris* has only two actions (rotate a block and translate a block), a handful of reactions, and one game object (the *Tetris* block in its various forms). However, to determine whether a component influences cognitive gameplay, we need to vary that component and compare games which have those variations. Comparing the gameplay of one computer game with the gameplay of a different computer game can be problematic, as both games require deep structural similarity before differences between them can be comprehensively identified. In other words, the two games need to be isomorphic at a deep level. When two games are isomorphic it means that they are structurally identical at a certain level of abstraction (e.g., see Dou et al., 2010). The level of abstraction can vary, such that two games may be isomorphic at one level but not at a different level.

For example, consider two games that are variations of *Tetris*. In one game, the blocks are multicolored and glow when translated or rotated. In the other game, the whole display is black-and-white. These two games are different at the surface level of abstraction, or the presentation level of information, but they are isomorphic at deeper levels since they share the same rules and interactions. As another example, consider two other variations of *Tetris* in which there is one difference between them: in one game only the next block that will appear is shown, while in the other game the next four blocks that will appear are shown. This change is at the level of rules and hence is deeper than the presentation level. Thus, these two games are not isomorphic at the surface level or at the level of rules. However, they are still isomorphic at a more abstract level where both games share the abstract rule “present the next set of blocks.” In one game this abstract rule is specified as “show the next block” and in the other game the rule is specified as “show the next four blocks.”

A design process was developed and proposed by Sedig and Haworth (2012) in which multiple games can be designed and the structural differences between these games can be identified and controlled. Through this process, a series of games could be produced that are isomorphic at a very abstract level: they share the same source idea. In addition, the games can maintain isomorphism at other levels of abstraction, depending on how they are designed. The process involves five stages: (1) choosing a cognitive toy as a source of inspiration, (2) extracting some general patterns from the toy, (3) choosing one of the general patterns and using it to design a set of abstract rules, (4) using the abstract rules to create a set of concrete rules, and (5) using the concrete rules to implement a full computer game. Each stage after the first can be performed multiple times, branching out into separate paths for creating a computer game. For instance, one cognitive toy could be chosen, from which two sets of general patterns could be extracted. Three sets of abstract rules could be generated from each general pattern, giving a total of six sets of rules, and eventually resulting in at least six different computer games that are isomorphic to varying degrees. However, there was no mention in (Sedig & Haworth, 2012) of this process being used to empirically investigate the effect of structural differences of interaction on cognitive gameplay, although such a possibility was implied.

An example of using this process to create games is shown in Figure 1. A cognitive toy is chosen and four different computer games are produced at the end of the process. All four games are isomorphic at a very deep level, since they are all based on the same cognitive toy. The games are also isomorphic at the level of abstract rules, since they share the same set of abstract rules. The two games in the middle are isomorphic at the level of concrete rules, since they share the same set of concrete rules. The four games are different at the presentation level, and the first and fourth games are different at the level of concrete rules. The structural changes between the first and second games are much greater than the changes between the second and third games, given the difference in degree of isomorphism between those three games.

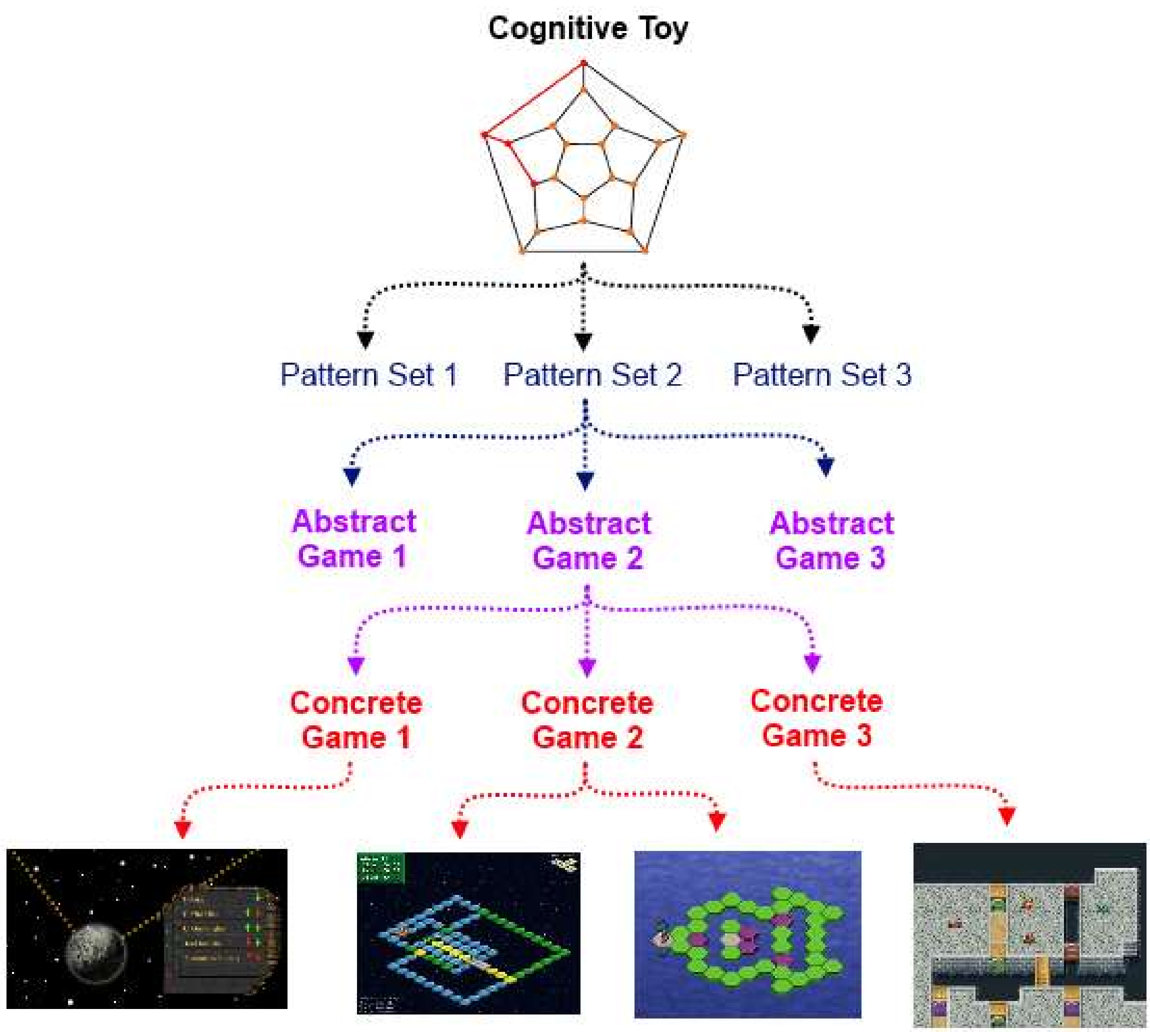

Figure 6-1: An example of the design process discussed in Sedig and Haworth (2012) being used to create four isomorphic computer games.

The stage in the process from which two games diverge determines the depth and degree of isomorphism. Since the designer determines when divergence occurs in the process, and the design that is created at each stage, he therefore controls the degree and kind of difference that occurs between the resulting games. In other words, by using this process we can create multiple games that differ in terms of exactly the structural component(s) that we want to investigate.

6.3 Methodology

In this section we will briefly discuss the two isomorphic computer games that we developed and the procedure for our investigation.

6.3.1 Computer Games

For our investigation of the effect of structural variations of interaction on cognitive gameplay, two isomorphic computer games were developed: *Fixed Play Space* (Game F) and *Rotating Play Space* (Game R). Both are 2D maze-like puzzle games that were developed by our research team using the design process and theoretical concepts described in Section 2.

The cognitive toy *Labyrinth* (see Figure 2) was chosen in the first stage of the design process. In this toy, the player turns dials to tilt the board and cause a marble to roll in a certain direction. The objective is to move the marble from a start position to an end position such that it does not fall into one of the many holes in the board. From this toy, we designed a simple abstract game, in which the player must navigate an orb from a start position to a goal position while avoiding various obstacles. Some obstacles stop the orb from moving while others force the player to restart. The orb moves in a random direction each time it collides with an obstacle. In addition, the player can arrange a limited number of operators. These operators cause the orb to move in a specific direction or to stop moving. Thus, the player cannot directly move the orb but she can influence its direction by using various operators.

Figure 6-2: A photo of a classic *Labyrinth*, which was the inspiration for the two computer games that we developed.

From this abstract game we derived two concrete rule sets and then implemented those rules as two computer games. Both games are divided into eight levels. Each level has a start point, a goal point, a specific arrangement of obstacles and bonus collectibles, and specific operators available to the player. The levels are intended to be progressively more difficult, based on the obstacles and the number and kind of operators available. The random directions of the orb were instantiated as a direction queue. This queue shows a sequence of twelve directions, and the player can click an action button to cause the orb to move in a way related to the first entry in the direction queue. The first entry is then removed and a new random one is added to the end, effectively making the queue infinite in length.

The UI is divided into two parts: the play space and the control panel (see Figure 3). The play space contains the orb and the level-specific arrangement of elements (the goal, bonus collectibles, and obstacles). This is also the space in which the player places and arranges the operators available for the current level. The control panel contains the direction queue, the available operators, the action button, and the player's current score and remaining lives.

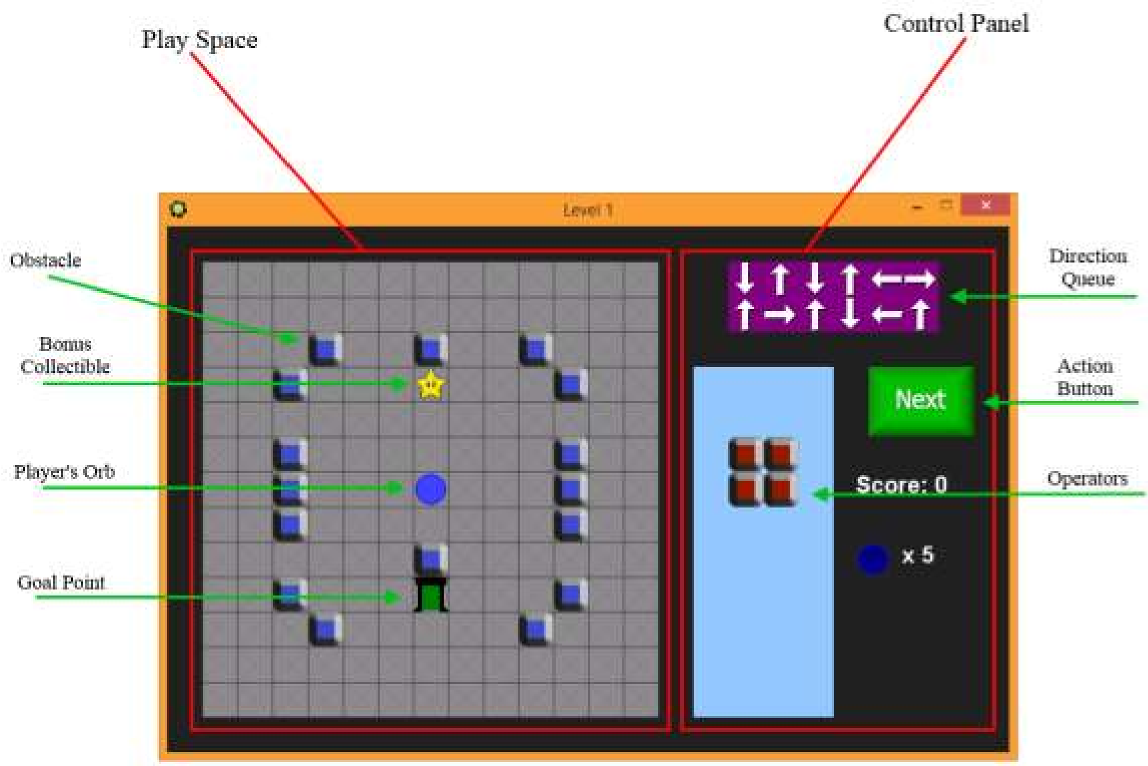

Figure 6-3: Screenshot of Game F with its UI components labeled.

In addition to clicking the action button, the player can arrange the given operators. This is done by clicking on an operator in the control panel and dragging it onto the play space. The player can move an operator after placing it, but she cannot move operators while the orb is in motion nor can she hold an operator somewhere while the orb is moving. In other words, operators must be placed and then left alone while the orb moves. An operator can be placed anywhere on the play space except on top of another operator or within one tile of the orb. Hence, the player cannot place any operator directly beside the orb. There are two types of operators in the game: walls and arrows. When the orb collides with the wall operator, the orb stops moving. Wall operators give the player more control over the ending position of the orb. When the orb collides with the arrow operator, it moves again in the direction of the arrow. For example, if an orb was moving to the right and collided with an "up" arrow, the orb would then begin moving upward and would continue until it collided with something else. Arrow operators give the player more control over the direction of the orb's movement. The player was not given the same number or type of operators in each level; in some levels the player may have walls and arrows, while in others she may have only arrows or only walls.

Bonus objects are placed throughout each level. One type of bonus object awards the player 50 points when the orb collides with it. The player cannot receive points in any other way. The other type of bonus object awards the player an extra life when the orb collides with it. About halfway through the game, black hole objects were also placed on the level. When the orb collides with a black hole, the player loses a life and the level restarts. When all the player's lives are lost, the game ends in failure. Colliding with a black hole is the only way in which the player can lose lives and thus is the only way in which she can lose the game. If the player reached the goal point of the final level, she wins the game and is shown her final score. We did not attempt to provide any other scoring mechanism, such as points for operators used or time spent per level, as these might become additional factors influencing the behavior of players.

Slightly different graphics were used in each game, which means that the two games differed at the presentation level. The games also had different directions in the direction queue and responses to the action button being clicked. In other words, the games differ in the reaction component of interaction. This is the structural difference that we explored. Other than this one structural difference, and the presentation differences, the two games are identical.

6.3.1.1 Fixed Play Space (Game F)

In Game F, the direction queue contains a series of cardinal directions: up, down, left, and right. When the action button is clicked, the orb begins moving in a straight line in the direction of the first entry in the queue and stops when it collides with a wall or the edge of the play space. There is an animation of the orb as it moves, so that the player can see it move from its start to end position. Only the orb moves; all other objects in the play space remain fixed in place. See Figure 4 for an example of this movement. If the orb collides with an arrow operator, it moves in the same fashion (i.e., gradually until it collides with another operator). The goal point is presented as a door, the bonus collectible that gives score is presented as a star, and the bonus collectible that gives an extra life is presented as a circle that looks similar to the player's orb. The wall operators have the same appearance as existing walls except for a difference in color.

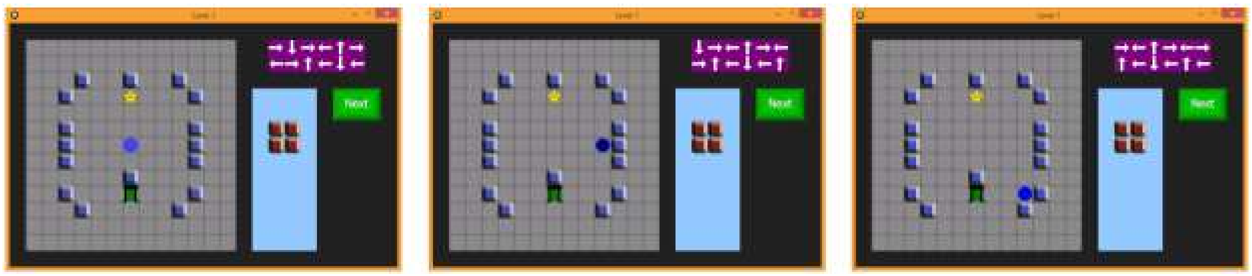

Figure 6-4: Screenshots from Game F that show the ending position of the orb after the player clicks the action button. From left to right: the starting position, after one movement right, and after one movement down.

6.3.1.2 Rotating Play Space (Game R)

In Game R, the direction queue contains a series of rotation directions: 90-degree clockwise turn, 90-degree counter-clockwise turn, and a 180-degree turn. When the action button is clicked, the play space and all elements within it, including the orb, rotate according to the first entry of the direction queue. This rotation is animated (i.e., the player can see the objects move). After the rotation is finished, the orb moves down towards the bottom of the play space. This movement is similar to the orb's movement in Game F (i.e., gradual, animated, and stopping upon colliding with another object or the edge of the play space). See Figure 5 for an example of the rotation. The orb does not collide with anything during the rotation, only afterwards when it is moving down toward the bottom of the play space. The game objects look the same as in Game F except minor color differences in the background, UI, and walls, and the bonus collectible that gives score is presented as a coin instead of a star.

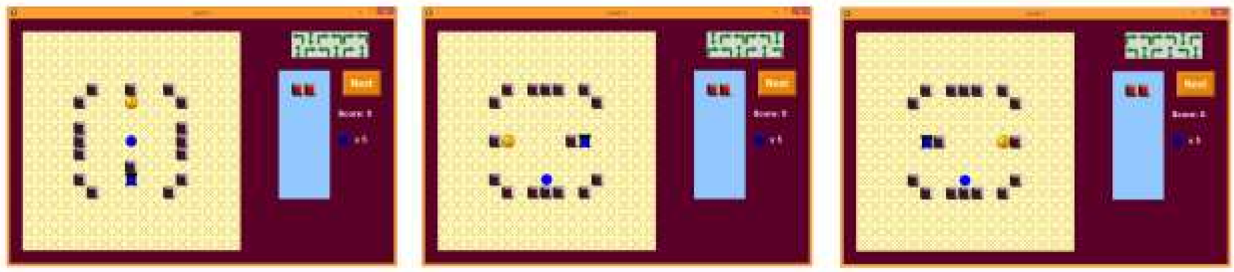

Figure 6-5: Screenshots from Game R that show the orb moving. From left to right: the starting position, after a 90-degree left rotation, and after a 180-degree rotation.

6.3.2 Investigation Procedure

We recruited eight volunteers from our university through word-of-mouth to participate in our exploratory investigation ($N = 8$). These volunteers are henceforth referred to as players. None of them had previously played the games that we had created. These players were randomly divided into two groups: Group 1 and Group 2. The procedure for our investigation was divided into four parts: (1) players filled out a demographics survey on paper; (2) they played one of the two games for about 30 minutes; (3) they filled out a design questionnaire on paper; (4) they played the other game for about 30 minutes. The whole procedure lasted about 90 minutes per person. People in Group 1 played Game F and then Game R, while those in Group 2 did the reverse.

We collected 4 types of data to analyze cognitive gameplay: (1) video-recordings of the game screen, which captured in-game interaction of the players; (2) audio-recordings, which captured their verbal comments; (3) two paper surveys, which captured specific opinions about the game and the experienced gameplay; and (4) direct observations obtained from watching the players interact with the game. The demographics survey was intended to gather past game-playing behavior and preferences to provide context for their comments. The design questionnaire was intended to gather their opinions about the game. For example, they were asked: whether they liked the game and why, whether they had any difficulties playing the game, the amount and nature of challenge the game provided, the kind of cognitive tasks in which they engaged, and the mental effort involved in these tasks. The design questionnaire only applied to the first game that they played.

For each game, players were given a one-minute summary of how to play the game. We answered any questions they had about how to play either game. While playing the first game, we asked them some questions to help elaborate or explain their behavior and to encourage them to vocalize their thoughts and the operation of their internal cognitive processes. When playing the second game, they were asked these same questions as well as a set of specific interview questions to encourage them to compare the two games. For example: do you like this game more than the previous one and why? Which game required more mental effort? What kind of mental effort was involved? Which game would you prefer to play again and why?

We were primarily interested in two types of observations. First, we wanted to explore how players experienced the game in cognitive terms. Specifically, we wanted to know what players were thinking about, what they were mentally emphasizing, and why they were engaged in this behavior. If players found it difficult to verbally express these experiences, or felt as though they were not thinking strongly about anything, then we could interpret this to mean they were not aware of their thoughts; in other words, the cognitive gameplay was more automatic and unconscious than reflective and conscious. Although we might want a game in which players think more deeply or are more aware of their thinking, we did not want to measure that in this study nor did we want to conclude which game was better. Our interest was to explore any possible differences in cognitive gameplay between playing these two games. Second, we wanted to explore differences in perceived enjoyment. We were interested in players indicating whether they enjoyed one game over the other, if they experienced any frustration, and explaining their thoughts on the source of this enjoyment or frustration. While this whole investigation could be conducted using quantitative means (see Section 5.4), collecting qualitative data was deemed more appropriate. Furthermore, the extremely low sample size rendered quantitative analysis irrelevant. Within the standard expectations of quantitative studies (e.g., $\alpha = 0.05$; effect size = 0.1; power = 0.8) the statistical power of our sample is roughly 0.05, far below what is needed to make any generalizable claims (see Cohen, 1992). As such, quantitative analysis was deemed unnecessary and only qualitative results will be presented.

6.4 Results

This section is divided into three subsections. First, we provide a brief summary of some data gathered from the demographics surveys and scoring, to obtain a clearer picture of the players. The remaining two subsections contain the qualitative results for the two games, one section per game. Each subsection begins with a summary of the results for the corresponding game and continues with more detailed results. These results include a combination of comments from the players, their responses on the paper surveys, and observations made by the researchers. Where relevant, we mention whether the player was in Group 1 or Group 2. Players are referred to as P<#> (e.g., P1, P5) when they are quoted in the results. The quotes are verbatim, minus the removal of filler words (e.g., “uhh,” “umm,” “like,” and “you know”). We have reported and emphasized the qualitative results over the quantitative results, even though some quantitative data is reported.

6.4.1 Player Demographics and Scoring

In this subsection, we present some of the demographic data gathered about the players (see Table 1). This data is sorted into the two groups to help clarify the context of detailed results given in Sections 4.2 and 4.3. Statistical analysis of this data has not been conducted due to the small sample size. However, it is interesting to note that three of the players self-identified as non-gamers and none of the players identified puzzle games as a type of game with which they had experience playing.

Table 6-1: Summary of player demographics.

Age[1]	Gender	Gamer Status[2]	Experience[3]	Experienced Game Genres[4]
Group 1				
28	M	NG	E	FPS, Sports, Strategy
29	F	NG	SE	Cell Phone Games, Word Games
22	M	C-A	SE	Adventure, Sports
22	M	A	VE	FPS, Sports, Strategy
Group 2				
28	M	C	SE	Adventure, Mystery, Sports
26	M	H	X	MMORPG, Strategy
22	M	C-A	VE	Adventure, FPS, RPG, Sports
23	M	NG	E	Adventure, FPS, RPG

[1] Ages are in years
[2] "Do you consider yourself:" NG=non-gamer, C=causal gamer, C-A=between C and A, A=average gamer, A-H=between A and H, H=hardcore gamer
[3] "In general, how experienced are you with playing games?" N=novice, SE=slightly experienced, E=experienced, VE=very experienced, X=expert
[4] "Which kinds of games do you feel most experienced with?"

In Tables 2 and 3 we present some scoring results. The scores that players obtained could be interpreted as representing their level of motivation for obtaining optional content in the games, since score was not awarded for merely completing a level. Obtaining a high score could mean that the player was willing and interested in additional challenge; obtaining a low score could mean that the player was unwilling or unable to go after the bonus objectives in the game. The maximum score in both games was 1100. For added context with interpreting this data, the highest level reached in the game is also reported. Each game had 8 levels, and Tables 2 and 3 report the level of the game at which the experiment ended. An entry of "end" means that the player completed the game. Total play time for each player per game was roughly the same: 30 minutes. Again, this data was not analyzed statistically due to low sample size. Relevant data for P8 was lost due to a recording error.

Table 6-2: Player scoring and game completion data for Group 1.

	Score		**Level Reached**	
	Game F	**Game R**	**Game F**	**Game R**
P1	500	250	7	4
P3	150	0	7	3
P5	1100	250	End	7
P7	1100	700	End	End

Table 6-3: Player scoring and game completion data for Group 2.

	Score		**Level Reached**	
	Game F	**Game R**	**Game F**	**Game R**
P2	250	200	8	7
P4	1100	700	End	8
P6	1050	600	End	End

6.4.2 Results of Game F

Players in Group 1 played Game F first, while those in Group 2 played it second. In this game they reported that the main cognitive task in which they were engaged was planning, planning in Game F required less mental effort than in Game R, Game F had a low level of difficulty, Game F's low level of difficulty affected how deeply they planned, and Game F was more enjoyable than Game R but considered trivial in comparison.

6.4.2.1 Main Cognitive Task: Planning

Planning a path was the primary cognitive task in which players engaged. For example, the following is a portion of P2 verbalizing his thoughts while working through one level:

> "*First of all, if I go up all the way to the upper corner here it won't be good for me, because the ball will go up and then it will go right and then it will go up and … up and down and then up and to the right. So, this is the level where I really need to use at least one of these [wall operators]. Now, I have four so I need to create a path based on them. So, if I go up, these two are probably the best places where I need to stop the ball, which means that I have to put something – one of those blocks – in either of those two cells. If not, I have to avoid this [one location in the play space] because the next move will go towards the black hole. So then if I go to the right, I can't stop here of course, or here because the next move is up and then I will lose.*"

The fact that the players engaged in planning could also be seen in other ways. While reflecting on the game, P8 wrote in the design questionnaire that the game "*required a decent amount of foresight for some of the levels, which if you did not do, may result in death or a very long time to beat it*." While playing the game, players pointed to specific bonuses or locations in the play space which they were trying to reach and planned a path to it. P4 explained that, "*my strategy is to get the highest amount of points. I am just looking at the next directions, and trying to predict the first direction*." Later on, when creating a plan in a level with several objectives, P4 continued to elaborate his strategy: "*Right now I do not care about these directions [the ones near the end of the queue], I just pick one goal—one star or one life—and match the first direction to where I want to go*."

Players mainly engaged in one kind of behavior to externalize the path they were planning: they would trace the path of the orb with the mouse or their finger. This path could be where they wanted the orb to go, how it would move when affected by other operators, or simply where it would go based on the sequence of directions. Players also indicated goal points, by orienting the mouse toward them or pointing at them with a finger, while engaged in forming a plan. Sometimes they would also move an operator to a particular location but keep it in one location instead of placing it and would then trace the path with a finger. In some cases, players did no tracing but placed operators in a way that suggested they had determined a path mentally. Regardless of the number of operators they used or how far ahead they planned, none of the players expressed difficulty (verbally or behaviorally) with tracing a path.

The players in Group 2 also engaged in planning. However, planning in Game F was different than planning in Game R, due to the exclusion of visualizing the rotation:

> "*I'm still planning moves ahead but I don't struggle to figure out where I'm going to end up. … I'm not struggling with rotating in my head. … That's part of the planning but you have to actually do that spatial reasoning to say 'this is going to be here and then it's going to go left' and that's very difficult to make multiple rotations in your mind and then plan in the context of that. … You still have to plan [in Game F] no matter what, there's no way you can randomly complete the game. So you still have to plan but you don't have to put as much effort into planning.*" (P1)

However, the amount of planning in Game F compared to Game R was potentially greater:

> "*I know where it's going to be, so I'm not spending my time trying to figure out where it's going to be for the next move. My plan is more from the starting point to the end point, I'm spending time rearranging things so it follows the path. Before [in Game R] I had an end point, I had a beginning, but I had no idea how to go from beginning to end because I couldn't see what each move did. But now I*

know what each move does, so I'm able to develop a strategy; I can plan more." (P3)

6.4.2.2 Lower Mental Effort Compared to Game R

People in Group 2 considered Game F to be easier than Game R, primarily due to less mental effort required:

> "*Well in terms of planning ahead, it's much easier. I can say 'I'm going to go up, I'm going to go right, I'm going to go up, I'm going to go down' I can plan the whole level with very little mental effort, or at least it seems that way.*" (P1)

> "*I love this game! I don't have to do the rotations and the ball is the only thing that moves, so right now when I click next the ball is going to move right there. So now I click next, and the ball goes there, and I knew that was going to happen.*" (P3)

> "*This is very different. It's definitely not as complex as the other one. It's still involves some kind of thinking but nowhere near the level of the other one.*" (P5)

When questioned about the difference in the mental effort between games, the players responded with the following:

> "*In terms of planning and how hard it is mentally I would say this one is certainly easier. Because I can very easily plan all the moves, I just follow it with the mouse. … And I can, based on that, place my blocks where I want them. There's no confusion really, saying 'is that really what's going to happen?' It's easier to predict. They're both predictable, if you know what's going to happen, but this is easier much to predict accurately.*" (P1)

> "*[the mental effort I need to exert is] significantly reduced, because everything is not moving only one thing is moving in the game. And because only one thing is moving you can keep track of it, you know where it currently is and you know where it's going to end up. And because everything else is not moving, there's only one variable you have to follow. In [Game R] you had to follow many*

> *variables, where the door is, where the possible blocks that you're putting are…*" (P3)

> "*The other game was very much mentally draining. With this game it's like, I should be thinking, I can think multiple moves ahead but it's almost like I don't really have to because there's not as much risk involved.*" (P5)

In elaborating specifically the difference in mental effort, P1 focused on visualizing the rotation as the primary source:

> "*[Game F] doesn't require any type of spatial reasoning, mental rotation. That's what's very taxing mentally, especially when it comes to sequential processing of that. This is much easier. You don't really have to do this in your head. I mean, you have to imagine the ball moving but you can literally put your finger there and know where it's going to go. There's very little mental effort.*" (P1)

6.4.2.3 Low Level of Difficulty

All players in Group 1 indicated that the game was rather easy. They rated the difficulty of the game on the design questionnaire as a 2 or 3 out of 5 (mean = 2.5) and rated the increase in difficulty as a 2 or 3 out of 5 (mean = 2.75). P4 commented that "*if I could change the game, I would make it more challenging*," while P8 wrote that "*I liked that the levels weren't so hard to make me frustrated, but were hard enough for a good challenge*." All the players had a chance to see the full progression of difficulty, even if they did not complete the game. When comparing the final level of Game F to the earlier ones, P2 said "*it's not really much more difficult*" and he indicated that the difficulty was the obstacles that forced resetting the level: "*The challenges are, of course, the large number of black holes*." All the players indicated that they had no problems or confusion with the UI. P6 even made this more explicit by writing in the design questionnaire that "*the controls were concise and could be used effectively*."

When comparing the difficulty of Game F and Game R, players in Group 2 indicated that Game F was much easier: "*It does not really even compare. The other one [Game R] had* *just so much more going on that it was way harder. …I always had to prepare for the next move no matter what, whether or not it allowed me to get closer to the goal*.” (P5)

All players, in both groups, indicated that the operators gave them more control over the movement of the orb. Since the arrow operators affected the direction of the orb, these were considered the most useful. However, they also made the game much easier, as is mentioned by the players. P4 said “*with four arrows you can do anything. It is very easy*.” P6 explained that the arrows “*made the game easier than if we just would have had blocks. It allowed you to control the movements up to 4 moves in advance*.” However, P8 complained that “*the arrows may be too overpowered and those levels [in which arrow operators were available] were generally easier*.”

6.4.2.4 Difficulty Affects Depth of Planning

Two of the players in Group 2 said that they could plan up to twelve steps ahead, using all the directions in the movement sequence. The other two said they could only plan about three or four moves ahead. For example, P5 said “*[I'm looking] three maybe four [moves ahead]. You always try to look as far as ahead possible, but in the other game it was a lot more difficult to do that. So usually it was just one, maybe two [moves]*.”

However, the ease with which the players could plan in this game may have discouraged them from planning far in advance. P1 suggested that “*It seems I don't really need to plan that much in advance, I can just move. I know where a few moves are going to take me and then just go from there.*” Giving the player too many arrow operators also seemed to reduce the need to plan further in advance. When asked about the influence of the arrow operators, P7 replied: “*They make life easier, for sure. You don't have to construct a long plan, because you can get to each objective every turn. So I found myself thinking less because you don't really have to. It is the same across both games.*” A similar comment was made by P1, “*I'm using the arrows to try to plan multiple moves at once. I like the challenge of planning multiple things and then seeing it all happen, but that's my strategy; the game doesn't make me do that.*”

6.4.2.5 More Trivially Enjoyable than Game R

Despite how easy the game may have been, the players in Group 1 still enjoyed it. In the design questionnaire, P4 commented "*I liked the thinking involved in this game*," while P8 commented "*I liked how you have to be clever in how to use actions to avoid death*." Complaints were directed toward the perceived low difficulty, and not the UI or the mental effort involved.

When players in Group 2 were asked which game they enjoyed playing more, three said that Game F was more enjoyable because the challenge of the other game was too high for them. One player said that Game R was more enjoyable because he preferred its higher level of difficulty. However, when asked which game they would prefer to play again, two said that they would prefer Game R over Game F and a third one said that he would only play Game F again if it was made harder: "*I would get bored playing this all day, as it becomes trivial quickly*." (P7) The main difference between the two games was well captured by P5:

> "*They're both games, but [Game F] is more of a conventional fun game. The other one is more of a mind workout. I see the value of both but depending on the mood I'm in or what I'm prepared to do [I would play one or the other]. I'd be interested in playing both [games] but at different times. If I knew I couldn't play these games outside the context of this [investigation] then I'd ask to play the other one because it would do more for me than playing [Game F]. It is still slightly challenging. If I want a break from work, which is how people normally pick-up games, then I'd pick this one. But, if I wanted something that would really alter my thought process then I'd pick the other one for sure.*"

6.4.3 Results of Game R

The players of Group 2 played Game R first, while those in Group 1 played it second. In this game, players reported that the main cognitive task in which they were engaged was visualizing, visualizing required a high amount of mental effort, the high mental effort needed for visualizing also made Game R more difficult to play than Game F, it was difficult to plan more than a couple moves ahead, and Game R was considered enjoyable but in a different way than Game F.

6.4.3.1 Main Cognitive Task: Visualizing

Various comments made by the players while playing the game suggest that they were engaged in planning. For instance, P1 said the following while verbalizing his thoughts:

> "*Maybe get it into the middle. Let's try this. [placed wall operator] It will go there, and then [clicked action button] and then it's going to fall this way. … And then it's going to rotate, so if I go like this. [clicked action button] and then, oh that's… [looked at direction queue, saw next entry was a 180-degree turn] oh boy, I have to plan all these things ahead.*"

As another example, consider the following excerpt from P3:

> "*So the ball will end up one rotation here, and this guy [the exit] will end up one rotation here. So what we want is that when the ball gets up here, we want it to fall this way. It will come all the way back down and we want it to go across, so let's take this [wall operator] and place it so that it's here. Now, think about this again. … Instead of stopping the ball here let's stop the ball so it's over here, because if the ball stops somewhere over here then it'd be in this quadrant. If I get the ball in the same quadrant [as the goal] then the next time it moves, I can use these four blocks and put them where they need to be.*"

Similarly, after P5 clicked the action button and did not like the result he said "*I should have a put a block here, then I would have got a life after. In this context, you have to look ahead to see where things will be after the rotation.*"

However, players also visualized the rotation of the play space. As can be seen in the above quotations, visualizing was performed at each step of the plan and thus planning could not be performed without visualizing. The players engaged in a variety of strategies to help them visualize the rotation of the play space. One tilted his head to see the screen from different orientations. Another arranged the operators on the screen to indicate where certain elements would end up after the rotation. For example, after placing a wall operator at some position on the screen P3 said "*this wall is just a place-setter, to mark where the ball will end. I'm using these blocks for other things just so I can visualize what's going on.*" Most of the players used the mouse or their finger to trace the arc of rotation for some object. Players in Group 1 also performed the same actions, with P2 even explaining his strategy for using the wall operators as mental aids: "*I have enough of these [wall operators] that I do not have to think about it, so I'm using the resources as a way to ease the way I think, and then the mental effort will probably decrease.*"

6.4.3.2 High Mental Effort Needed for Visualizing

Visualizing had a high mental-resource cost for the players. When P7 was asked what required the most mental effort he responded: "*probably visualizing the rotation, then planning to get every coin since I'm committed to that*." Similarly, P2 explained that the thing which required the most mental effort was "*trying to visualize the effect of each action*," and then afterward "*I have that sort of visualization in my mind … and I'm trying to put these blocks in that visualization as a component to see what would work*." When P5 was asked whether he was thinking about the way in which the play space rotates he replied:

> "*That's always in the back of my head, because that's the nature of the game. It's a very abstract way of thinking, the whole world moving around, and it's not something that I'm used to in any game in any setting. That's the main problem that I'm having right now.*"

The high effort can also be seen in the mistakes made by the players and the confusion or frustration expressed upon realizing that the orb moved in a different direction or to a different location than anticipated. For example, after clicking the action button P1 said "*that was not how I wanted it to go. … It rotated the way I expected but I do not know why I thought it was going to fall a different way*." Similarly, after placing some operators in anticipation of a certain rotation P3 clicked the action button and, while watching the reaction, said "*No! This guy needed to be here!*" Even after determining the rotation of the play space, players would frequently double-check the anticipated outcomes to ensure a mistake was not made. This occurred more regularly for some players after they started making mistakes, but all players performed double-checking regardless of when they made mistakes.

The players in Group 1 had a similar experience, but it was unexpected given how well they played Game F: "*Now I sometimes make mistakes. I thought one thing would happen but now I see that something else happened. But I have to make sure that even if I make mistakes I won't lose*." (P4) Likewise, P6 said that the game "*is much more frustrating, because you cannot make the ball go where you want it to. [There is] definitely less control*." Due to the greater effort involved, some of the Group 1 players said they would just focus on solving the level:

> "*[in Game F] I would have gone for scoring the coins, or the stars, but in this one the first thing that I am thinking about is getting out of the door. Even though I know that there are only two black holes, but yet because I am not sure about the effects I think that is what I am going to do.*" (P2)

6.4.3.3 Higher Difficulty than Game F

Generally speaking, the players stated that this game was difficult to initially play and took some time to learn. They spent between 5 and 10 minutes on the first two levels of the game, trying to understand how the elements within the play space would rotate and practicing visualizing the rotation. Regardless of the time spent practicing, they all expressed difficulty with visualizing. For example, while verbalizing his thoughts, P3 said: "*so it rotates this way, and then the bottom guy becomes this top guy. So, the ball will be here. But that's not where I want it, I want it here. This game is too difficult, [laughs] I have to get through at least level 2!*" As another example, while P7 was reflecting on his planning he said "*I usually think about where I want to be and then I try to get there, but going more than five [rotations] is really tough. The board flips make it tougher.*" When talking about the difficulty of predicting the location of the orb, P5 said: "*Obviously it's hardest with the inverse flip. Right now the world's gonna invert, and this coin will end up on top and this will end up there, and I can't do anything to stop that I think. I want the ball to stay here close to the coin, ideally.*"

On the paper surveys, the players rated the difficulty of the game at a 4 or 5 out of 5 (an average of 4.5) and also rated the increase in difficulty at a 3 or 4 out of 5 (an average of 3.5). In further elaboration, P5 wrote in the design questionnaire that the game was "*very challenging. Not so much a 'fun' game but an exercise for the mind*." Similarly, P1 wrote that "*the cognitive load and spatial reasoning was demanding. The game was very forgiving in terms of number of moves, which made it seem less difficult than it could have been*." The players indicated that the difficulty of the game came from having to visualize the rotation of the play space. For instance, P3 wrote in the design questionnaire that the game was very difficult "*because I didn't understand what the [180-degree rotation] did or how things rotated. It was hard to visualize the result of a possible action*." When asked to elaborate, P3 said:

> "*The problem is that I can't see-- I'm having a difficult time anticipating where everything is going to be. Because now I know, 'ok this is here and this is here', and then they will go back to the position where they were before and nothing's changed. But then I do this guy [the 180-degree rotation] and I can't even make a strategy.*"

Players in Group 1 indicated that Game R was more difficult than Game F, and that playing Game R required more mental effort than playing Game F. For instance, P4 mentioned that Game R "*is much harder because you cannot predict objects and their place. And if you predict, there will be problems. If I had arrow [operators] in this level it would be so easy*." Similarly, P2 explained the difficulty that he had with visualizing:

> "*First of all, it's definitely hard to understand the way that these blocks and the ball move. It needs more than one or two test actions. Since the effect of the left and right arrows is somehow different than the other ones, this will pose another challenge. So, the first challenge is to understand what are the elements out there doing, in terms of the actions, and then when I could understand that then I could come up with solutions based on using the red blocks.*"

Finally, P6 was the most explicit about the difference between the two games when he said "*this game [Game R] is definitely harder because it's harder to visualize a few moves in advance, especially the flipping. This requires more mental effort [to play]*."

6.4.3.4 Visualizing Limited Planning Depth

Players in both Group 1 and Group 2 were only able to plan a few moves ahead:

> "*It really is much easier just to compute one move at a time. If there wasn't rotation you could do multiple moves which– It's hard to do that spatial reasoning. To say 'ok, it's going to be like this' and* then *from* there *'it's going to be like this'. I can do it in some cases but then sometimes you make a computation error because you think 'in two stages away it's going to be like this' but then you forgot that something is going to rotate in a certain way.*" (P1)

P6 mentioned that he was thinking "*two moves ahead, maybe three in some cases*," while P4 explained that "*I just look at the next arrow and I put some things down to avoid losing and be closer to the current goal*." When P2 was asked whether he planned a sequence of several actions he replied with:

> "*No, because the uncertainty of actions and sub-levels—a sequence of two or three actions in the same level—are much higher than the previous one [Game F]. The previous one was exact. In this one [Game R], the uncertainty exists and it's to a high degree. So that will stop me thinking ahead of time for maybe even more than one step, because I'm not really good at spatial recognition and memorizing things based on the place that they are or where they are going to be.*"

Three of the four players in Group 1 also said that they engaged in more planning in Game F, since it was too difficult in Game R to create long plans.

6.4.3.5 Different Enjoyment than Game F

When asked which game was more enjoyable, three of the four players in Group 1 said Game R was more enjoyable since they preferred the higher level of challenge. For instance, P2 said "*I had a more enjoyable time with this game than the previous one*," and P6 agreed: "*I had more fun playing this one more, because it's harder*." P4 disagreed though:

> "*If I want to say which game I liked more, I liked [Game F] more. This one [Game R] is more challenging, but I think less. It's like you play a game, and it's very hard, but you don't think you just play the hard game.*"

However, they expressed the opinion that Game R was more about avoiding obstacles. For instance, P8 said "*The previous game required an offensive strategy, whereas this game uses a defensive strategy as you try not to die*." When Group 1 players were asked what game they would play again, three said they would prefer to play Game R since Game F was not challenging enough for them:

> "*It depends if I really wanted to have a challenge or I just wanted to enjoy. If I just wanted to enjoy, then of course the first one [Game F] because I thought I could score more. If I really wanted to head for challenge then this one [Game R]. Solving the challenge in this game is definitely more enjoyable than in the previous game.*" (P2)

When asked to compare the two games, P2 continued this thought and said: "*[These games are] different. The goals are the same, but the way that I am thinking about obtaining those goals are different.*"

6.5 Discussion

As can be seen from the results, there was a difference between the cognitive gameplay of Game F and Game R. When the players played Game F they engaged in planning, to create a path through the play space, but it seemed that they did not engage in visualizing. Yet, when the same people played Game R they engaged in visualizing the rotation of the play space in a very effortful manner. This suggests that structural differences in interaction may affect cognitive gameplay. However, other implications arise from these results, which can be roughly grouped into three areas: challenge and enjoyment, interaction design, and cognitive gameplay. The implications in these three areas will be discussed in more detail in the subsections below. At the end of this section, we will discuss some of the limitations of this investigation and the conclusions that should not be drawn based on these results.

6.5.1 Implications for Challenge and Enjoyment

There was a difference in difficulty between Game F and Game R. All the players reported that Game R was more difficult than Game F, because they needed to visualize the rotation of the play area. In other words, a difference in the cognitive gameplay also resulted in a difference in the difficulty that players experienced while playing the game.

Visualizing the rotation of the play area in Game R can be considered an additional cognitive challenge that players needed to overcome. By challenge is meant an obstacle which the player must overcome in some way to progress through the game (see Adams, 2010). Challenges in a game can be cognitive in nature, such as recognizing a specific series of patterns or working through a logic problem. In the design of cognitive gameplay then, it seems that we can incorporate specific cognitive tasks into the structure of a game by turning them into challenges. From the perspective of systematic design of games, it would be best to introduce cognitive challenges intentionally. If we are not aware of how cognitive tasks can turn into challenges, we may unintentionally disrupt the difficulty balance of our game and potentially turn an enjoyable game into a boring or frustrating one.

Existing research already indicates a correlation between the difficulty of a game and the enjoyment experienced by the player (e.g., Alexander et al., 2013; Cowley et al., 2008; Sedig, 2007). When the player encounters a game with a difficulty level that is too low or too high for their own expertise, they find the game less enjoyable. Such an effect was also seen in this investigation. Over half of the players indicated that they preferred playing Game R over Game F, and this was because they found Game R more difficult and liked that higher degree of difficulty. For them, Game F was enjoyable but too easy for long-term enjoyment. The other players did not prefer playing Game R and cited the high difficulty as the reason. They found Game R too difficult, were regularly frustrated, and found relief and enjoyment in Game F. They indicated that Game R was too difficult because they were unable to correctly and efficiently visualize the rotation of the play area. Even though balancing the difficulty of a game should already be a conscious design choice, we need to consider the impact of cognitive gameplay on difficulty since cognitive challenges will affect the difficulty and enjoyment experienced by the player. Similarly, the player may enjoy cognitive challenges; we should then focus on the nature and difficulty of the challenge, since that will probably affect enjoyment more than the presence or absence of cognitive challenges in the game

Lastly, it is interesting to note that most of the players who preferred playing Game R were those who played it as the second game. Perhaps the mental effort involved in visualizing rotation may be too much to be immediately introduced. Although the players who played Game R as the second game indicated that they found it harder and enjoyable, they seemed to have less difficulty with visualizing the rotation than the players who encountered it immediately. This could be due to differences in the players' abilities. Another possible explanation is that players who played Game R second only had to learn how to visualize the rotation. First-time players of Game R had to learn all the rules of the game while also learning how to visualize the rotation. Since visualizing rotation was so integral to the game, they may not have enough mental resources to learn the functionality of various operators as well as discover paths through the level. Thus, it is possible that Game R required too much mental effort from first-time players. When they played Game F afterwards, they encountered only one new rule (the orb moving in specific directions) and it was much easier to learn. Therefore, introducing rotation in later levels of Game F might be better, as this would give the player the opportunity to first learn the rules of the game and then have more resources available to learn and practice visualizing the rotation.

6.5.2 Implications for Interaction Design

The difference between Game F and Game R is structural, but it is the reaction component of interaction that was changed. When the action button was clicked in Game F, the reaction was simply that the orb moved in a certain direction. When the action button was clicked in Game R, the reaction was that all objects in the play space rotated in a certain direction. We could attempt to derive conclusions about the relationship between spread of reaction and cognitive gameplay. However, given the nature of this investigation, we should instead merely note that cognitive gameplay may be influenced by changes in the structural elements of interaction. For our investigation, one element of the reaction component was changed. However, an element of the action component could have been changed instead, or a different element of the reaction component could have been changed. Without a framework that identifies possible structural elements of interaction, it will be difficult to comprehensively study the relationship between interaction and cognitive gameplay. This is also true for structural elements other than those related to interaction (e.g., lives, narratives, scoring, and time limits). A framework of structural elements of games and their relationship to cognitive gameplay does not exist yet, though similar frameworks are being developed (e.g., Aleven et al., 2010; Bedwell et al., 2012).

The type of interaction that is available can also influence cognitive gameplay in ways other than those explicitly mentioned by the players in this investigation. For example, the main interaction in *Tetris* was also used to offload certain cognitive processes (Kirsh & Maglio, 1994). If that interaction was different or could not be used to offload cognition, then the resulting cognitive gameplay would be different; there would be a difference in the distribution of cognitive processes within the player-game cognitive system. A result similar to that found by Kirsh and Maglio (1994) was seen in our investigation. When playing Game R, some players used the available operators to help visualize the rotation. This was unintentional in the design of Game R. In general, the player may interact with the game to offload their cognitive processes. However, we can enhance such offloading when we are aware of this possibility and intentionally design the game to better facilitate it. With the example of Game R, we could have allowed the player to mark certain locations in the play space or to watch an animation of the play space rotating in a certain direction. Either of these changes may have assisted players in visualizing. Likewise, we could have designed Game R so that an operator could not be moved once it was placed. This would greatly discourage players from using operators for anything other than their intended purpose and hence would have removed one of their only in-game aids for visualizing.

Finally, the difficulty that players had in visualizing the rotation might have been influenced by the way in which rotation was implemented. Rather than the play space rotating as a continuous whole, only the objects within the play space moved. If players focused on all of the elements and tried to visualize them individually, this may have been why players found rotation difficult to visualize and resource intensive. Additionally, players found the 180-degree rotation the most difficult to understand. This rotation did not involve two 90-degree rotations; instead, objects moved in more-or-less a straight line to their destination point. Although objects ended up at a position that was a 180-degree rotation from their starting position, the path these objects took may have confused players. Both points regarding how rotation was implemented suggest that the way in which we implement the design of a computer game may unintentionally affect the difficulty of its tasks. Likewise, it may also affect cognitive gameplay; although this is not something we explored, the method of investigation presented in this paper could be used for such an exploration.

6.5.3 Implications for Cognitive Gameplay

While many of the implications mentioned in the previous two subsections also apply to cognitive gameplay, two further issues need to be mentioned. First, when looking at the results of this investigation, it is clear that players engaged in planning when playing either Game F or Game R. When playing Game F, players said that they could plan from five to twelve steps ahead. However, most of the time they only planned with the next one or two entries in the direction queue in mind. Most of them reported that they felt the game did not require them to plan further ahead, so they only developed very short plans. They could keep clicking the action button without concern for a more efficient solution, and they had multiple attempts at each level; so if they did not prevent the orb from being destroyed, they could still try the level again. It seems that such few restrictions made Game F rather easy and less cognitively engaging than Game R. Some players tried to challenge themselves by self-enforcing various restrictions, or attempting to achieve a high score, but these activities point to the shallowness of the cognitive gameplay. Hence, even though Game F allowed the players to plan ahead, there were too few in-game restrictions or rewards for encouraging much in the way of long-term planning. Further research, then, is required to explore possibilities for other factors that can be incorporated into a computer game to increase the depth of cognitive gameplay. As previously mentioned, a framework detailing the structural elements of a game and their influence on cognitive gameplay would greatly assist with this.

In Game R, an alternative problem arose. Since planning was dependent on correctly visualizing the rotation of the play space, the players who found it too difficult to visualize the rotation were unable to create any kind of plan. At most, they attempted to guess the result of the first entry in the direction queue and used the operators at their disposal, but they were not able to consider further entries in the queue. However, only some of the players found it too difficult, and these players were able to create long plans in Game F. Therefore, it is likely that these players had difficulty planning in Game R not because of their inability to plan but because of the intense cost of visualizing the rotation. Just as some minor structural change can make the cognitive gameplay too shallow, as in the case of Game F, it could also make the cognitive gameplay too mentally demanding, as in the case of Game R. The resource cost of various tasks, and what factors of interaction can affect this cost, is another possible area of further research.

6.5.4 Limitations of this Investigation

First and foremost, we did not conduct a study to determine which game produced better or deeper cognitive gameplay. Our interest was purely in exploring the cognitive gameplay that resulted from a single structural change. Although this structural change was associated with the reaction component of interaction, further conceptual clarification is needed in order to study it in more detail. These two games could be studied again, but it is not clear how the same reaction could be translated into other games. Likewise, there are many other structural elements that we could study using this same method, but doing so without an associated conceptual framework will make it difficult to translate the results of such studies into a general framework for designing cognitive gameplay. Existing literature detailing such structural elements tends to be vague in terms of specific components (e.g., Aleven et al., 2010), although this could be due to focusing on very general components (e.g., story, challenge) rather than particular ones (e.g., a rule limiting the number of actions). A potential framework for the structural elements of interaction exists (see Sedig et al., 2014), but it is not within the context of computer games and may require some adaptation.

In our investigation, we only gathered qualitative data. This was sufficient for exploring the experiences of our players but seems insufficient for conducting a detailed analysis of the differences in cognitive gameplay. In particular, further analysis could be performed on the depth of planning involved, the amount of mental resources used, and the effect of cognitive offloading on any internal visualizing. Interview and in-game questions that were specifically structured for such an analysis would be an improvement over our current investigation. Gathering quantitative data would also be an improvement, particularly for analyzing the memory usage and memory capacity of the players. Other metrics could also be gathered, such as the actions that players performed, the exact time between actions, and the time spent on each level. However, such data requires a much larger sample size for it to be statistically valid, so any future studies would need more participants. The limitation of small sample size prevented quantitative data from being particularly useful, and hence we had to rely on qualitative data. Such limitations are acceptable for the investigation presented in this paper, but they need to be overcome for more formal studies in the future.

6.6 Conclusion

This paper presented an exploratory investigation of the effect of structural variations of interaction on cognitive gameplay. Two computer games were designed and developed (Game F and Game R). These games were isomorphic at a deep structural level, and their only difference was in one component of interaction. Eight volunteers played both of these games, their behavior and comments were recorded, and a difference in cognitive gameplay was observed. When they played Game F they were primarily engaged in planning, but when they played Game R they were engaged in visualizing rotation. The mental load needed to visualize was reported as being so heavy that the players found it difficult or impossible to also engage in planning. Also, there was a difference in difficulty between the two games, which led to a difference in enjoyment. All the players said that Game R was the harder of the two. Those who found Game R too difficult preferred Game F, while the rest found Game F too easy and preferred to play Game R.

This investigation was conducted to better understand how we can study the relationship between structural components of interaction and cognitive gameplay. The design process that we used in this paper (see Sedig & Haworth, 2012) can enable researchers to develop numerous isomorphic computer games. Since we can control the structural differences between these games, we can also control the features that we want to isolate for further study. This process has not been used to study the effects of structural elements until this paper. Even though we only explored one structural element of interaction, the investigation procedure presented in this paper could be used to study many different interaction elements and other structural elements unrelated to interaction. As previously mentioned though, a framework identifying and conceptualizing these elements would greatly assist in such an investigation.

Having control over the differences between games allows us to draw stronger conclusions about how to design cognitive gameplay. We only investigated the cognitive processes in which the players were engaged and not whether the game was responsible for any improvement in the players. However, more detailed and elaborate studies need to be conducted, building upon the simple investigation presented in this paper, to provide evidence for the micro-level features of a computer game that improve or hinder specific kinds and depth of cognitive gameplay. In all such cases, more formal studies of the relationship between the structure of interaction and cognitive gameplay need to be conducted, and a more systematic conceptual framework of the structure of interaction in games needs to be developed. It is our hope that such studies will be conducted in the future, and that this paper provides a promising start.

Chapter 7: Interaction Design and Cognitive Gameplay: Role of Activation Time

This paper has been published as Sedig, K., and Haworth, R. (2014). Interaction design and cognitive gameplay: Role of activation time. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play (CHI Play)* (pp. 247-256). New York, NY: ACM. http://dx.doi.org/10.1145/2658537.2658691 It is reprinted here with permission from the ACM.

Note that the format was changed to match the format of this dissertation, and the references were moved to the end of the dissertation. Figure numbers have also been changed to be relative to chapter numbers. For example, "Figure 7-1" is the first figure of Chapter 7 and the figure is labeled as such. However, this figure is referred to as "Figure 1" in the text of the chapter. The same is true for tables. In addition, when the phrase "this paper" is used, it refers to this chapter. Lastly, note that the term "Activation Time" refers to the same structural element that was called "Activation" in Chapter 5. We used a slightly different term to make it more understandable for the original target audience.

7.1 Introduction

Recently, there has been increasing interest in using computer games for more than just entertainment. For instance, games have been designed with a focus on engaging players in deep and meaningful cognition—e.g., learning content, improving working memory, practicing spatial reasoning, or developing problem-solving skills. For such games, cognitive gameplay has a very important role. In general, gameplay refers to the experience that emerges from the player interacting with a game (Salen & Zimmerman, 2004). Cognitive gameplay, then, signifies the emergent cognitive processes within the overall gameplay experience. In other words, it is the cognitive component of the game-playing experience.

Cognitive gameplay is the primary factor that determines how well games engage the player in cognitive tasks and activities (Haworth et al., 2013). For example, if we want to design a game for the cognitive activity of learning mathematics, we need to attentively design cognitive gameplay so that it is conducive to deep reflection focused on the mathematical concepts embedded in the game (Sedig, 2008). Since cognitive gameplay is emergent, we must design it indirectly (Salen & Zimmerman, 2004) by designing the interaction from which it can emerge. Thus, to design cognitive gameplay in a systematic manner we need to be guided by a conceptual framework that clarifies the relationship between interaction and cognition. Currently, however, there are no such frameworks.

There are some frameworks and methods for the design of gameplay (e.g., Connolly et al., 2012; Isbister & Schaffer, 2008; Sweetser & Wyeth, 2005); but they focus on how to design for usability or enjoyment. As such, the cognitive aspects of gameplay are either ignored (e.g., Sweetser & Wyeth, 2005) or considered in relation to the enjoyment or challenge it causes (e.g., Connolly et al., 2012; Cox et al., 2012). Such frameworks and methods are necessary for designing effective and enjoyable games, and for analyzing games to understand what makes them enjoyable. However, they do not assist in understanding how to make players think more deeply or meaningfully.

There are also numerous frameworks for design in the context of learning (e.g., Aleven et al., 2010; Bedwell et al., 2012; see especially Connolly et al., 2012), some of which even relate learning outcomes to specific game features (e.g., Bedwell et al., 2012). While these are certainly more useful for determining how to make players think, they have several limitations. First, the features are often very broad in scope, such as narrative or the right level of challenge (Bedwell et al., 2012); this information is important for macro-level design, but it is less useful for micro-level design, especially micro-level design of interaction. Second, these frameworks are focused on a given activity (i.e., learning), and thus do not always generalize to other high-level activities that emerge in the context of cognitive gameplay—e.g., planning, problem solving, or decision-making. And, third, it is uncommon for these frameworks to be empirically supported (Connolly et al., 2012; Ravaja & Kivikangas, 2009). Little empirical support is a problem for research about general gameplay though, with randomly-controlled trials and properly qualitative studies seemingly lacking (Connolly et al., 2012). Existing studies on cognitive gameplay have mixed conclusions; some suggest there are cognitive benefits to playing games (e.g., Quiroga et al., 2011) while others suggest that cognitive benefits are limited in scope (e.g., Owen et al., 2010). Such confusion is reached because of the way in which games have been studied (see Ravaja & Kivikangas, 2009). Furthermore, these studies are often disconnected from any design-related theory; in other words, the study indicates the kind and/or depth of cognitive gameplay that playing a game seems to have but has little to say about how to potentially duplicate this effect in other games.

Despite the significant work conducted in cognition for games, most especially in the area of learning, what is lacking from existing studies and frameworks is some way to identify and investigate the relationship between specific components of the interaction in a game and the resulting cognitive gameplay. Sedig and colleagues (2014) have recently developed a framework to inform the design of the micro-level interaction elements that affect the interactivity of visual tools. Since games are also visual tools, in this paper, we will use one element of this framework, activation time, to study cognitive gameplay. Each interaction in a game can be considered as a composite of a set of elements that collectively determine its structure. One of these structural elements is activation time. This element refers to the timing of the action response of an interaction. All the structural elements of interaction can take on different operational forms, depending on how the interaction itself is designed. Two operational forms which activation time can assume include: immediate and on-demand. When activation time is operationalized as immediate, the action response occurs as soon as the player performs an action. When activation time is operationalized as on-demand, the action response does not occur until after the player explicitly requests it. For example, consider a game in which the essential interaction is arranging shapes in some container. If the activation time element of this interaction is operationalized as immediate, whenever the player acts on a shape to move or rotate it, the shape will immediately change its position or orientation. If activation time is operationalized as on-demand instead, whenever the player acts on a shape, it will not change. The player will have to apply the changes separately, such as by clicking a “move shapes” button. Once that button is clicked, all the shapes on which the player has acted will change their position or orientation accordingly.

Since previous research has examined other aspects of interaction (e.g., Kirsh & Maglio, 1994; Sedig, 2008; Sedig et al., 2001), we decided to explore the usefulness of activation time for the systematic design of cognitive gameplay, which is the focus of this paper. We will discuss how we investigated the relationship between different operational forms of activation time and cognitive gameplay, particularly the effect of activation time on deep and reflective planning. First, we designed two 2D puzzle games, each of which had two versions that only differed in the way that activation time was operationalized. We hypothesized that operationalizing activation time as immediate would promote a fragmentary and reactionary approach to planning, while operationalizing activation time as on-demand would promote a more holistic and reflective approach to planning. We then conducted an empirical study with these versions to test the difference between the cognitive gameplay that participants experienced, with the results confirming our hypothesis.

Therefore, we will discuss three things in this paper: 1) a summary of our conceptual framework for the design of cognitive gameplay as it pertains to the studied games, 2) the games that we developed, and 3) the study that we conducted and its results. Thus, the key contribution of this paper is to further our understanding of interaction design for computer games, particularly in relation to cognitive gameplay.

7.2 Theoretical foundations

In this paper, our sole interest is computer games—that is, games that are implemented on some form of computation technology. The exact form of this technology is irrelevant as a computer game could use any software or hardware platform (e.g., a tablet, game console, personal computer, or cell phone). In other words, by computer game we mean a game in which computational technology is the means through which the player interacts with the game, and not simply a game functioning on a personal computer. Several other terms are used in the literature to describe this medium (e.g., video game, digital game, electronic game), but we will treat them as synonymous with computer game.

Computer games are more suitable for cognitive gameplay than non-computer games (e.g., board games, physical sports) because of the greater interactive potential that computational technology provides (Salomon & Perkins, 2005; Sedig & Parsons, 2013). For example, consider a computer game version of *Soccer*. The player could act in the game by directing a player and pressing buttons to click the ball, which would result in similar interaction as non-computer game versions. However, the player could also interact by typing the velocity for the ball each time it is kicked, or by arranging the path that the ball will take each time it collides with a specific team member. These forms of interaction are relatively simple to design for computer games but are impractical or impossible for non-computer games. It is this greater range of possibilities for interaction that allows computer games to facilitate more appropriate forms of cognitive gameplay, since the design of interaction plays the greatest role in how cognitive gameplay unfolds (Haworth et al., 2013).

7.2.1 Interaction

Computer games are interactive, meaning that there is a two-way dialogue between the player and the game. Thus, interaction can be divided into two components: an action that the player performs on some object at the user interface (UI) level of the game, and a reaction from the game. The reaction typically includes some perceivable effect, such as an object on the UI changing in size or position. It is also synonymous with other terms such as action response and visual feedback, though these terms may have more specific meanings than the general one intended by reaction.

Interaction can refer to low-level software or hardware events, such as clicking the mouse, pressing a key, or tapping a touch-sensitive surface. In this paper though, the term interaction refers to a general pattern of interaction, such as arranging a set of tiles, transforming one or more shapes, navigating through a game space, or assigning some behavior to game entities. Although the designer can choose patterns appropriate for the goal of a game, each pattern also differs in the cognitive engagement that it provides (for a comprehensive list of such action patterns and their cognitive utility, see Sedig & Parsons, 2013).

7.2.2 The Core Mechanic

Computer games often have several interactions that are incidental, such as pausing, saving the current game state, or adjusting the sound volume. Yet, there is usually a set of one or more interactions that are essential for playing the game. The term core mechanic typically refers to these essential interactions (Salen & Zimmerman, 2004; Sicart, 2008). When playing the game, the player performs these essential interactions repeatedly, forming a cycle of action-reaction. In other words, the core mechanic of a game is the continual pattern of essential interactions in which the player and the game are engaged.

For example, consider the computer game *Tetris*. There are two essential interactions in this game: rotating and sliding. As such, *Tetris* has a very simple core mechanic: the repeated cycle of acting on a shape and it reacting by moving or rotating.

For usability reasons, the incidental interactions should be carefully designed. However, the core mechanic forms the heart of gameplay; it is the repeated performance of these essential interactions that gives rise to the experience called gameplay (Ermi & Mäyrä, 2005; Salen & Zimmerman, 2004). Therefore, the design of the core mechanic will have the greatest effect on cognitive gameplay (e.g., Sedig, 2008; Sedig et al., 2001).

7.2.3 Cognitive Gameplay

Over the past few decades, cognitive scientists have largely departed from models of cognition as discrete information processing that occurs within the brain to models that encompass a broader system (e.g., Clark & Chalmers, 1998; Hutchins, 1995). This is due to increasing evidence that objects external to the brain not only mediate and facilitate the operation of cognitive processes but also are integral parts of what can be viewed as an extended and distributed cognitive system (see Clark, 2008).

Cognition in the context of computer games can therefore be understood as the player-game cognitive system, which is composed of a player and a computer game. Although the player is engaged in some cognitive task, the task itself is enabled, mediated, and supported by the game (Haworth et al., 2013; Kirsh & Maglio, 1994). Thus, cognition is the mutual coordination and cooperation between the player and the game. This occurs as a result of the core mechanic. It is through the core mechanic that the player and the game are coupled together in a mutual and dynamic dialogue (Kirsh, 2005; Salomon & Perkins, 2005; Sedig et al., 2001). In other words, it is through the cognitive coupling enabled by the core mechanic that cognitive processes and activities (e.g., planning, problem solving, decision-making, pattern recognition) can emerge.

For example, consider again the computer game *Tetris*. Through the core mechanic the player is continually moving and rotating shapes. However, the goal of the game is to form solid rows so that the components of the row will be removed from the play area. The player must identify optimal positions for the current block, in anticipation of future ones, in order to succeed. This is entirely facilitated by the core mechanic. In other words, the player can move and rotate shapes to aid the process of identifying possible and optimal locations, and to aid the task of planning the positions of blocks (Kirsh & Maglio, 1994). This is a simple illustration of the cognitive coupling between the player and the game. An important research question involves the quality of this cognitive coupling—that is, how should the core mechanic be designed to create desired forms of coupling?

Therefore, the term cognitive gameplay refers to the cognitive processes that emerge through the mutual dialogue between the player and the game, enabled by the core mechanic. In considering the quality of cognitive gameplay though, a number of factors could be involved such as: the kinds of processes in which the player is engaged (e.g., Bedwell et al., 2012; Sedig & Parsons, 2013), how deeply or consciously the player performs these processes (Schneider & Chein, 2003), and the amount of mental effort that these processes require (Sedig et al., 2001; Sweller, 2010). As such, these could be the factors that designers consider in order to promote desired kinds of cognitive gameplay.

7.2.4 Interactivity

Since cognitive gameplay is an emergent phenomenon, it cannot be designed directly. Design of cognitive gameplay is a second-order activity and must be done indirectly. It is the core mechanic that is designed directly, enabling the desired cognitive gameplay to emerge. The term interactivity refers to the quality of interaction, and thus the quality of the cognitive coupling between the player and the game (Sedig et al., 2014). The operational form of the interaction elements that make up the core mechanic of a game affect the game's interactivity and hence the quality of its cognitive coupling and its cognitive gameplay.

As previously mentioned, each interaction in the core mechanic has a set of structural elements. Sedig and colleagues (2014) have identified twelve such elements, some of which include: agency, context, flow, focus, granularity, and timing. These elements have various operational forms that affect the quality of the cognitive coupling between a user and any visual tool, including games. Hence, designing interactivity of a game at a micro-level involves determining how each of these elements should be operationalized.

An example of one of these elements that has been studied is focus. This element can be operationalized as direct or indirect. In the former the player acts on an interface object directly, whereas in the latter the player acts on an intermediary interface object to influence the target object. In a study of a game dealing with mathematical concepts, a comparison of these two operationalizations of focus resulted in different cognitive gameplay. The results showed that indirect focus led to deeper and more effortful learning among the subjects than direct focus (Sedig et al., 2001). Careful operationalization of this interaction element in one version of the game led to cognitive gameplay that was characterized by reflective, mindful learning as opposed to the other version of the game that promoted experiential cognition and shallow learning.

7.2.5 Activation Time

The interaction element investigated and reported in this paper is activation time. As previously mentioned, this element is concerned with the commencement of the reaction after the player has committed an action. Two operational forms of activation time are: immediate and on-demand. If activation time is immediate, the reaction occurs instantaneously after an action is committed. If activation time is on-demand, the reaction does not occur after an action is committed but only once the player requests it.

The operationalization of this element may promote different degrees of mental effort and engagement. For example, if activation time is on-demand, the player may be forced to engage in deep, reflective thought before committing an action, since the feedback from the action (i.e., the reaction) is not immediate. However, to test whether and how different operationalizations of activation time affect cognitive gameplay, we conducted a study.

7.3 Study Methodology

To investigate the role of activation time in cognitive gameplay we created two simple 2D puzzle-based computer games: *Temple Swap* (TS) and *Laser Dilemma* (LD). One game may be insufficient for proper generalization and validation of the study [17], so we developed two. We used puzzle games so that some cognitive effort would already be required; hence, we could measure difference of, and not existence of, cognitive gameplay. We used simple games to reduce the likelihood that game factors other than activation time would influence cognitive gameplay, and because simple games were easier to produce and learn how to play.

To test the different operationalizations of activation time, we created two versions for each game. Following the procedure outlined by Sedig and Haworth (2012), the only change between the two versions was the way in which activation time was operationalized. Although at the surface level these two versions are different games, they are still the same game at a higher level of abstraction (see Sedig & Haworth, 2012). As differences in the versions are due to the design decisions that we are interested in studying (i.e., activation time), it is reasonable to correlate any differences in cognitive gameplay to the operationalization of activation time. Two versions were thus necessary to study within-game effects of activation time, while two games were necessary to study between-game effects of activation time (see Ravaja & Kivikangas, 2009).

7.3.1 Game 1: Temple Swap (TS)

TS has 20 levels of slowly increasing difficulty. In each level, the player is given a grid of tiles. Each tile has one of four different symbols on it. The goal of the game is to arrange the tiles so that each one is in a horizontal or vertical set of three or more with matching symbols. A level is complete when all its tiles are in a set.

The core mechanic includes one interaction: swapping the position of two adjacent tiles. Tiles can be vertically or horizontally adjacent, but not diagonally adjacent (i.e., connected at the corner). One incidental interaction is also included: resetting the level, which returns all the tiles to their original position. The player is awarded points for completing each level, with the number of points based only on the difficulty of the level. The same levels were used in both versions of the game and were play-tested to ensure a similar increase in difficulty across both versions.

7.3.1.1 Temple Swap – Immediate Activation Time (TS-I)

In this version, the activation time element of the core mechanic was operationalized as immediate. This was implemented as follows. The player selects one tile by clicking on it and then selects another adjacent tile by clicking on it. Then, the tiles move so that their positions are swapped. The movement is gradual, taking just under one second to complete, and is accompanied by a sound effect. The player cannot select tiles while this movement is occurring. After each swap, the game checks if a correct solution has been found and rewards the player accordingly.

7.3.1.2 Temple Swap – On-Demand Activation Time (TS-D)

In this version, the activation time element of the core mechanic was operationalized as on-demand. This was implemented as follows. The player selects tiles exactly as described for TS-I. Once the player has selected a pair of tiles, this pair is added to the end of a queue. The queue lists all the locations on the grid that will be swapped and in the order that this swapping will occur. The player can click on a “go” button to request the reaction. Then, each pair in the queue, in its appropriate order, will swap exactly as described for TS-I. The player cannot select tiles until all pairs in the queue have been swapped. If the sequence of swaps in the queue does not result in a correct solution, then the level resets as though the player had pressed the reset button. In other words, once the player clicks the “go” button, the proposed solution must correct or else a whole new solution will have to be created.

Figure 7-1: A screenshot of game TS-D.

7.3.2 Game 2: Laser Dilemma (LD)

LD has 20 levels of slowly increasing difficulty. In each level, the player is given a maze-like arrangement of objects: goal rings, a laser gun, walls, and reflector blocks. The laser gun fires an energy ball in a certain direction. This ball travels in a straight line until it collides with a wall, which destroys it, or a reflector block, which changes its direction. When the ball collides with a goal ring, the ring becomes activated. The goal of the game is to have the energy ball activate all the goal rings on the level before the ball is destroyed.

The core mechanic includes two interactions: the player can move reflector blocks around on the level, and she can change the orientation of these blocks. Only certain reflector blocks in each level are interactive. These are marked with a thick black border around their edge, and are placed in a box on the left-hand side of the UI at the start of each level. There is also one incidental interaction: the player can reset the level, which returns all the interactive reflector blocks to their original position.

The player is awarded points for completing each level, with the number of points based only on the difficulty of the level. The same levels were used in both versions of the game and were play-tested to ensure a similar increase in difficulty across both versions.

7.3.2.1 Laser Dilemma – Immediate Activation Time (LD-I)

In this version, the activation time element of the core mechanic was operationalized as immediate. The activation time in this case does not refer to the placing or rotating of reflector blocks, which is also immediate, but to the laser gun firing to test the solution. In other words, every time the player places a reflector block on the play area, the laser gun will fire and the solution will be tested. Thus, the player can always and immediately see whether her current solution is correct and, if not, continue to revise it.

7.3.2.2 Laser Dilemma – On-Demand Activation Time (LD-D)

In this version, the activation time element of the core mechanic was operationalized as on-demand. As above, the activation time refers to when the laser gun fires. In this version, the player can place and rotate as many reflector blocks as desired and the laser gun will never fire. To fire the gun, the player must click the "fire" button, at which point the game will test the solution exactly as it occurs in LD-I. However, if the solution is not correct then the level resets as though the player had pressed the reset button, and she will have to create a whole new solution.

Figure 7-2: A screenshot of game LD-D.

7.3.3 Study Procedure

We conducted the study with a mixed-methods approach that emphasized qualitative data, and divided the study into two sessions: a play session and an interview session (both are described below). We recruited 40 university students (n=40) using on-campus posters and in-class presentations. Two randomly chosen participants won a cash prize, but the rest were not compensated. This method of compensation was mentioned during recruitment but the size of the prize ($150 CA) was not mentioned in accordance with ethical requirements. Participant demographics are described in the participants sub-section below. We randomly assigned each participant to one game version: TS-I, TS-D, LD-I, or LD-D. At least one male and female played each version, but the number of each gender varied in each version. All 40 participants performed the play session, while 12 of them were chosen to participate in the interview session. Interview criteria included: the quality of their comments during the play session and their preference and availability for being interviewed. Most participants were excluded from the interview due to being uninterested or unavailable.

7.3.3.1 Play Session

Each play session was approximately 60 minutes in length, involving three steps in the following order: 1) participants filled out a demographics survey on paper, 2) they played their assigned game version for about 25 minutes, and 3) they filled out a design questionnaire on paper. They only played the version to which they were assigned, and they received the same instructions as others who were assigned to their version. The game provided the instructions, though the researchers answered questions from participants.

We collected four sources of data: 1) video-recordings of the game screen, to capture in-game interaction; 2) audio-recordings, to capture verbal comments of the participants; 3) two paper surveys, to capture specific opinions and experiences; and 4) direct observations obtained from watching participants interact with the game.

We constructed the two surveys, using previous HCI-based computer game studies (e.g., Sedig, 2008) as examples. The demographics survey gathered demographic data, previous game-playing behavior, and game-playing preferences. The design questionnaire gathered participants' opinions of different aspects of the game and gameplay—e.g., whether there were usability problems, the degree of challenge the game provided and the nature of this challenge, the kind of cognitive tasks in which they engaged, and the mental effort involved in these tasks. Some questions in both surveys used a 5-point Likert scale, such as "Do you think the game interface made it easier for you to create plans for each level?" with a scale from "Absolutely" to "Definitely No". Most of the questions were open-ended or contained an open-ended portion, such as "What did or didn't you find challenging in the game," and "If you ever reset one of the levels, why did you reset it?"

7.3.3.2 Interview Session

The purpose of the interview session was to let participants experience the other version of their assigned game so that they could compare the two versions. The interview was separate so that the original session time could be kept short but participants would still have sufficient time to discuss their thoughts on the two versions in detail. Each interview session was performed two to three weeks after the participant had performed the play session. The interview session was approximately 60 minutes in length, during which time the participants were shown both versions of their assigned game, were asked several questions about both versions, and were given a brief opportunity to play. The interview was semi-structured in format (see Lazar et al., 2010). The questions were mostly open-ended, and were intended to gather opinions of each version from several perspectives—e.g., perceived difficulty or challenge, anticipated effort required for planning and cognitive load for memory, and the way in which they would think while playing it. The interview was audio-recorded, and the game screen was video-recorded to capture any playing by the participants.

7.3.3.3 Participants

In this sub-section, we will present a summary of some data gathered from the demographics survey. Participants were equally split in terms of gender (20 male, 20 female), came from a variety of disciplines[17], and were both graduate (17) and undergraduate (23) students. When asked to list the kinds of games which participants had experience, only 5 wrote puzzle games; most wrote games or categories that were reaction-oriented in nature (e.g., First-Person Shooter, apps like Fruit Ninja). The most commonly mentioned games that participants preferred playing[18] were Tetris (8), Candy Crush Saga (4), and Call of Duty (4). Most participants considered themselves non-gamers and either a total novice or slightly experienced gamer (see Table 1).

---

[17] Including: biology, business, chemistry, computer science, economics, education, engineering, English, health science, kinesiology, law, library science, mathematics, media studies, medicine, modern language studies, psychology.

[18] In response to: "List some of the computer games that you most prefer playing." Participants wrote between 1 and 4 games or game-genres. Over 80 unique entries were given.

Table 7-1: Participant demographics summary.

Self-Identification of Gamer Status[19]					
NG	C	C-A	A	A-H	H
23	3	5	3	5	1

Self-Assessed Game-Playing Experience Level[20]				
N	SE	E	VE	X
11	14	7	6	2

7.4 Study Results

In this section, we will present some results from the study. This section is divided into four sub-sections, one for each treatment group. Each sub-section begins with a summary of the findings for that group. Since the data was merged during the analysis phase, the results include a combination of observations, recordings, verbal comments from both sessions, and survey responses. Participants were encoded using P<#> (e.g., P4, P27) when the study was conducted, and are referred to accordingly when they are quoted.

7.4.1 Game TS-I

In this version, the cognitive gameplay that participants experienced included fragmentary planning, undirected planning, and a low degree of mental effort relative to effort in TS-D.

7.4.1.1 Fragmentary Planning

The participants who played this version attempted to solve each level by moving tiles around until they found a correct solution. In some cases, this movement was part of a larger strategy for solving one part of the puzzle: "*I just moved pieces around, for the most part. If I knew a certain set was supposed to be somewhere, I stuck it there. Like, I didn't move the keys once they were there … unless I really needed to.*" (P23) In other cases, the tiles were moved because the attempted solution was incorrect and the

---

[19] "Do you consider yourself …" Choices are: NG=non-gamer, C=causal gamer, C-A=between C and A, A=average gamer, A-H=between A and H, H=hardcore gamer.

[20] "In general, how experienced are you with playing games?" Choices are: N=novice, SE=slightly experienced, E=experienced, VE=very experienced, X=expert.

participant was exploring other possibilities. When P4 was asked how he recovered from an incorrect solution, he responded: "*I still tried to make sets of three. I just kept trying different shapes, kept moving things around until it worked.*" However, just randomly moving pieces around to get through the game was a possible strategy. For instance, P34 wrote in the design questionnaire that the game had low difficulty because "*one could try and try and move tiles randomly until they matched.*"

7.4.1.2 Undirected Planning

All the participants sometimes performed actions that had little pragmatic utility. For example, they would move a tile from one location on the grid to another and then move it back to its original position. Other times they would swap two tiles that had the same symbol. P34 did this a few times, and one of those times commented "*Oh, that was pointless.*" At some point, all of them realized that a solution they were building was incorrect. Some of them also developed a correct solution without realizing it. This occurred to P23 three times, and on the second time he remarked: "*I didn't see that coming either. [The game] said it was right though, so fair enough.*"

7.4.1.3 Low Mental Effort

In the questionnaire, one participant wrote that the UI 'maybe' supported memory because "*immediately after [I made] a move I could undo it*" (P34). Another answered 'maybe' to the same question because "*I didn't remember the steps or action I performed in the game.*" (P4) Participants were observed repeating similar actions when attempting to solve one puzzle. For instance, P23 found one incorrect solution and then attempted to create a new one. However, once this solution was complete the participant exclaimed with a laugh "*I just did the same thing [but] on the other side!*"

7.4.2 Game TS-D

In this version, the cognitive gameplay that participants experienced included effortful visualizing and a high demand on working memory. Furthermore, the cognitive gameplay of TS-D was deemed superior to that of TS-I.

7.4.2.1 Effortful Visualizing

The participants indicated that the game engaged them in visualizing the sequence of moves and state of the board: "*I'm trying to mentally see all the swaps.*" (P6) For some of the participants, this activity was the very difficult. For instance, P2 said "*The hard thing is I can't visually imagine what happens next. If I could do it one at a time and have them swap, I could see it but I have to think instead.*" P18 echoed this sentiment:

> "*It's hard to plan ahead, I can't see everything. I do one move in my head, and then the next one, but then I begin to forget what I did. You don't see it as you do it, you have to do it all in your head and then press go. It's hard to remember all that. If it was one move at a time it would be a lot easier for me.*"

Similarly, when asked in the design questionnaire whether the UI made it easier to know the outcome of their actions, the most common answer (60%) was 'not really.' Some reasons why included: "*Because I had to perform all steps and did not see intermediary actions on the board,*" (P8) and "*You had to visualize your moves as opposed to seeing them one by one.*" (P6)

7.4.2.2 High Memory Demand

The participants also indicated that they needed to remember the mental visualization created and/or the sequence of actions. This was also a difficult and demanding task, as explained by P7: "*The most difficult part is to remember where the pieces are […] that's why I keep failing and trying again. I keep forgetting my moves. I forget one step in my plan or where I moved one thing.*" The high memory demand was also reported in the design questionnaire. When asked whether the UI made it easier to remember things, the most common answer (70%) was 'not really'. The main reason repeated by the participants was similar to P32's response: "*I was holding the information in my head to figure out what to do next.*"

In response to the high memory demand, some participants created partial solutions, tested them, and then revised them. For example, when working through part of a solution, P6 said "*I need this here, and I need to swap those to get it … I think it will be* *helpful to see it,*" after which the participant tested the solution. When the solution was completed the participant carefully looked at the screen to see the end state, but exclaimed when the level reset shortly after that: "*Oh no! I needed to see that longer.*"

Other participants broke the puzzle down into sections and solved the sections one at a time. The value of this strategy was explained by P9: "*I divided the puzzle into smaller parts, and that makes it much easier to solve. […] Because memorizing all of those places is hard.*" A more detailed explanation of this strategy was given by P3:

> "*I tried to break it down into smaller parts. For this puzzle, you've got three birds down at the bottom, and I said 'Ah, my first two moves would be to get those birds down.' I can now forget about those birds, because I've logically sealed then off. […] So now I've almost started a fresh game in my head, because I made that one logical move. So that's how I try to break it down into smaller and smaller parts.*"

7.4.2.3 Superior Cognitive Gameplay Compared to TS-I

When the interview participants were shown version TS-I they all thought that it had inferior cognitive gameplay than TS-D, based on their experiences with TS-D:

> "*My thoughts are that you've broken any need to play the game. You've eliminated any memory issues. […] You still have to figure out the end state […] but now you can just fool around and try things and get there eventually. […] If playing this game has any exercise benefits to the mind, […] this state of the game would ruin all of it.*" (P29)

> "*Now it's become a clicky game, it's really simple. Because I don't have to track my moves anymore […] I'm just playing a clicky game now. […] A game like this I'd get tired of very quickly. […] I don't think this would push you as much as the other versions, [even if] you had a maximum number of moves.*" (P3)

> "*You don't have to remember anything [in TS-I]. You make a step, and then based on that you make a new plan. […] [But,] there's no planning. You're not planning three or four steps ahead, you're not having to think of the bigger picture […]*

> *because the moment you click it, it switches. […] I think it's not as fun because it's so easy.*" (P8)

7.4.3 Game LD-I

In this version, the cognitive gameplay that participants experienced included wayfinding and planning, but the participants found themselves restricted from fully performing those activities.

7.4.3.1 Hindered Wayfinding

All of the participants engaged in the cognitive activity of wayfinding; they were trying to find a path through the level. This activity was externalized by tracing the path with their hand and with the mouse. However, the participants reported that wayfinding was influenced by the way that activation time was operationalized:

> "*[The laser firing after each block] kind of hinders me a bit. […] I would rather plan it and map it out first and fire when I'm done. Then I'd see where I went go wrong and take that singular one out.*" (P22)

> "*Here I lose a lot of time for a misplaced piece or just mentally thinking things through, and for every single piece I don't need to test the entire system.*" (P16)

Occasionally, participants would move a reflector block to a desired location but instead of placing it, and triggering the reaction, they would leave it and trace a path using their hand. One participant also placed blocks in front of the laser, to shorten the reaction time. Sometimes, participants would intentionally pick up a block and place it in the same location to trigger the reaction, and watch the result.

7.4.3.2 Hindered Planning

The participants were also engaged in planning; they had to place a sequence of reflector blocks in order to proceed through the game. When P22 was asked whether he had to do much planning, his response was "*not really. I mean, it shoots every time so I just go with whatever it's doing.*" When the participants were asked how the well they could plan with immediate activation time, their responses were largely negative:

> "*I found it frustrating. I wish I could have said 'ok, execute', or had a button to say 'go now.' […] Also, it was not helpful because it was slow, it shoots then it goes 'bom… bom…', but I'm working faster than that. I know what many of the contingencies are, I know what's going to happen in the event that I do that, so I don't need you to play it all out for me.*" (P13)

> "*I guess by it going step-by-step this forces me to—it almost solves it for you or works in a certain sequence. Because I have to put one down each time, and it shows me the solution each time, this is a step-by-step process.*" (P16)

7.4.3.3 Inferior Cognitive Gameplay Compared to LD-D

When the interview participants were shown version LD-D they thought it required more effortful planning and supported memory better:

> "*There is definitely more effort in planning, since I only have essentially one shot at it. And so, instead of putting a few pieces and figuring it out later, I basically have to map it all out by moving the mouse around.*" (P16)

> "*In this one you put [reflector blocks] all there, you can offload your memory onto [the game], and then test it. Whereas in [LD-I], you're keeping them in your memory. You're offloading them but you have to offload them one at a time. At any time you could forget one of them. Whereas [in LD-D] you can put them all there at one time, and that helps memory.*" (P13)

7.4.4 Game LD-D

In this version, the cognitive gameplay that participants experienced included effortful visualizing and a moderate memory demand.

7.4.4.1 Effortful Visualizing

As in LD-I, all the participants engaged in wayfinding. They externalized this activity by tracing the path with their hand and with the mouse. However, they also put significant effort into mentally visualizing the path:

> "*I'm attempting to predict the entire path […] I'm trying to simulate the entire thing in my mind, based on where it's going to go before I make a change, and then what changes I can make to get it to go where I want.*" (P10).

> "*[My strategy is to] predict the way the ball's going to go. It's tough to visualize it. I'd say [my strategy is] maybe a combination of trial-and-error and just like intense forethought before I press the actual fire button.*" (P20)

In the design questionnaire, P24 explained his strategy: "*I did not really make plans. For certain levels, it seemed like it would be easy and beneficial for me to make plans first, but I just did 'trial-and-error' in my mind a lot, which was why it was so demanding on my memory!*"

At some point, all the participants started creating partial solutions and testing them. For instance, while explaining the path being developed, P21 said "*if I place this here it will go up like this… no. I need to try this first to see how it works, because I have no idea.*" Some of the participants tested their plan quite frequently while others did not. For P29, for example, it was the latter: "*I check that it works; when I'm sure that it works [then] I click the fire button.*" For some participants it was necessary to test the solution to help them visualize the path, while others were able to mentally visualize it unassisted. For instance, compare P21's comment above with that of P25: "*Wow, I just don't even know what to do [for this level]. Part of me just wants to press fire and see where it goes, but I know exactly where it will go.*"

7.4.4.2 Moderate Memory Demand

When trying to build partial solutions and test them, some participants experienced difficulty remembering the solution they tested: "*It's hard to remember what I just did. I totally saw [the correct solution], if this could just come back down there. … It's hard, this is really hard.*" (P25) When asked in the design questionnaire whether the UI made it easier to remember things, the responses were mixed. Half agreed, "*I can see it. I don't need to remember everything,*" (P26) while the other half disagreed, "*I had to press reset a lot, and for levels with a lot of pieces I sometimes forgot where things were before I* *pressed reset.*" (P15) This memory demand was involved in both the visualizing, as shown in P24's comment above, and planning. P31 wrote in the questionnaire that planning was difficult because "*The [background] lines helped, but much of the state had to be maintained in working memory.*"

7.5 Discussion and Conclusion

In this paper we discussed activation time, one micro-level interaction element, and its role in cognitive gameplay. To further investigate this role, we developed two computer games each with two versions corresponding to the two operationalizations of activation. We then conducted a study to test the effect on cognitive gameplay. The results indicated that a difference in cognitive gameplay occurred across both games. The game versions with on-demand activation time engaged participants in more comprehensive planning and demanded more mental resources (e.g., working memory). The versions with immediate activation discouraged the participants from engaging in effortful cognitive gameplay and, in one of the games, interfered with their ability to plan in a comprehensive manner.

This study has several limitations, of which four will be mentioned. First, the sample size is small with very diverse backgrounds. This diversity assists generalizability (Ravaja & Kivikangas, 2009) but needs a larger sample size than what we used. However, our current sample size is typical for usability studies (Lazar et al., 2010) and, at the very least, is sufficient to identify that something warranting further study with a larger sample is occurring.

Second, our study gathered primarily self-report measures for the depth and kind of cognitive engagement. This was sufficient, since we were not interested in measuring gains from playing (e.g., improvement in working memory) but whether participants felt engaged in certain cognitive activities. Future studies that include objective measures of cognitive performance would more precisely indicate the cognitive processes that are affected by activation time.

Third, very little in the way of standard usability or game-enjoyment testing was incorporated into the study. This was intentional, since the focus was on cognitive gameplay. Informal usability testing was performed on the games, but it was not part of the study nor was it reported in this paper. Future studies incorporating usability testing would help to rule out extraneous UI issues that might confound the effect of activation time. As well, cognitive gameplay is known to have an effect on enjoyment (e.g., Connolly et al., 2012; Cox et al., 2012). Hence, measuring enjoyment more thoroughly would give finer granularity to the effects of activation time on gameplay. This may be difficult, especially considering the current problems with measuring enjoyment in games (see Nacke et al., 2010), and is another reason why such testing was not included in our study.

Fourth and finally, the games we tested were both puzzle games. This was intentional, so that we could first identify whether similar effects occur across different games in the same genre. In theory though, the effects of activation time should occur regardless of the game genre. However, puzzle games inherently require some cognitive effort, so our results may be limited to puzzle games. Future studies with games of different genres should be conducted to eliminate such bias.

The results of the study presented in this paper have implications for interaction design of computer games. They suggest that incorporating on-demand activation time into the core mechanic of puzzle-based games may result in more engaging, effortful, and reflective cognitive gameplay. They also suggest that incorporating immediate activation time in the core mechanic may result in less optimal forms of thinking, planning, problem solving, and visualizing. Therefore, if designers are not aware of activation time, the game they design may not be conducive to the intended cognitive gameplay. If designers want to systematically design cognitive gameplay, they should carefully consider how activation time is operationalized in the core mechanic.

7.6 Acknowledgments

This research has been financially supported by the Natural Sciences and Engineering Research Council of Canada.

Chapter 8: Solution Space in a Computer Game: Effect on Cognitive Gameplay

This chapter is being prepared for submission to Human-Centric Computing and Information Sciences. Its format was changed to match the format of this dissertation, and the references were moved to the end of the dissertation. Figure numbers have also been changed to be relative to chapter numbers. For example, “Figure 8-1” is the first figure of Chapter 8 and the figure is labeled as such. However, this figure is referred to as “Figure 1” in the text of the chapter. The same is true for tables. In addition, when the phrase “this paper” is used, it refers to this chapter.

8.1 Introduction

In recent years, researchers have suggested using computer games for purposes beyond mere entertainment. For instance, we could design computer games to engage players in deep and meaningful cognition, such as learning complex mathematical concepts, developing critical thinking skills, or improving context-specific decision-making abilities. To do so would require not only careful design of the game itself but also the cognitive aspect of the game. When we look at the design of computer games from a systems theory perspective we can see that a computer game is a system, since it is composed of interrelated parts that together form a complex whole (Salen & Zimmerman, 2004). For example, there are several components (e.g., user interface, rules, objectives, challenges, etc.) which are not computer games on their own but result in a computer game when combined.

We can also look at the broader player-game system. The player-game system has two components – the computer game and the player – and the main relationship that binds them together is the interaction that occurs. In light of the Theory of Distributed Cognition, the player-game system can be conceptualized as an integrated cognitive system (Clark, 2008). Within this integrated cognitive system are representations of information and processes that operate on the information. Both representation and processing can be distributed to different degrees among the player and the game, and this distribution is enabled and mediated by the interaction afforded by the game

(Haworth et al., 2013). Depending on the distribution, the player will have to exert different amounts and kinds of mental effort. In other words, the cognition in which the player is engaged depends on the distribution of representation and processing within the player-game integrated cognitive system.

For example, consider a computer game in which the player is given a set of unique pieces that she must arrange to form a target shape. The game is representing information to the player, such as the location of pieces and the target shape to form. The player interacts with the game through moving and rotating pieces. In doing so, the player is internally processing the information represented by the game. For instance, the player is identifying pieces, comparing pieces with the target shape, and visualizing the results of rotation. Since we cannot design any components internal to the player, if we want to change the distribution of processing between the player and the game we must do so through designing the game. For instance, the game could be designed so that more processing is done by the game (e.g., the game visualizes the results of rotation) or more representation is done by the player (e.g., the player must internally represent, or remember, the target shape).

By designing the distribution of cognition we are designing cognitive gameplay. Generally speaking, gameplay is the experience that emerges when the player repeatedly and continually interacts with the game (Salen & Zimmerman, 2004). Cognitive gameplay is the cognitive component of this experience; it is the cognitive system that emerges from the interaction between the player and the game. The quality of a computer game's cognitive gameplay determines how well the game engages the player in deep and meaningful cognition (Sedig, 2008). Since both gameplay and cognitive gameplay are emergent constructs, they cannot be directly designed. Instead, we need to design the computer game so that the desired cognitive gameplay emerges.

However, existing frameworks and methods for designing gameplay typically focus on non-cognitive components, such as challenge, flow, immersion, and presence (e.g., Ermi & Mayra, 2005; Sweetser & Wyeth, 2005). Hence, they are ill-suited for the design of cognitive gameplay. Other game design frameworks are situated in the context of learning (e.g., Aleven et al., 2010; Bedwell et al., 2012). These frameworks typically relate learning effectiveness to common game features, such as assessment, challenge, fiction, and goals. Despite the usefulness of these frameworks for learning, they may not generalize well for other kinds of cognitive gameplay—e.g., computer games that focus on decision making, problem solving, or planning. Furthermore, these frameworks tend to place little emphasis on interaction. Since interaction enables cognitive gameplay to emerge (Haworth et al., 2013) the design of cognitive gameplay should include interaction design. Therefore, we currently have a limited understanding of how to design cognitive gameplay.

To address this limited understanding, we are developing a framework for the design of cognitive gameplay. Two components of this framework include the representation and interaction sub-systems of the computer game. These two sub-systems should affect the distribution of cognition within the integrated cognitive system, and hence should affect cognitive gameplay. To test this possibility, we designed a 2D puzzle-based computer game with different implementations of the solution space. The solution space is the space through which the player navigates to construct a solution to some problem. This space contains various problem states, transitions between states, and the overall path (i.e., solution) from the start state to the goal state that the player is constructing. Since this space is part of the player-game cognitive system, it can be distributed across the player and the game. Changing this distribution is an example of designing cognitive gameplay. Hence, we implemented three different distributions of the solution space by designing different representation and interaction sub-systems. We then conducted an empirical study to look for any differences in cognitive gameplay that participants experienced between the different implementations. The results of the study indicated that a difference occurred, and that it was mediated by the solution space. In other words, through the design of representation and interaction – key components of our design framework – we were able to manipulate the kind and depth of cognitive engagement that the player experienced.

The remainder of this paper is divided as follows. In the next section, we will discuss the theory for designing cognitive gameplay. Then, we will discuss the methodology for the study along with the game and its versions. After that, we will present the results of the study. This paper will end with a summary and some future directions for this research.

8.2 Theoretical Foundations

To become conscious designers of cognitive gameplay, we need to be aware of research and concepts from multiple disciplines (e.g., Cognitive Science, Game Studies, Human-Computer Interaction, Information Sciences, Learning Sciences, etc.). Given such an interdisciplinary approach, a framework that could structure the transference and integration of concepts between these disciplines would provide a strong conceptual foundation for design. General Systems Theory provides such a framework, since it attempts to elucidate the essential principles of a system, regardless of its content or context (Skyttner, 2005). These principles can form a common conceptual foundation to facilitate integration among disparate theories and disciplines. Using General Systems Theory then, a computer game is a composite system (i.e., it has sub-systems) that is embedded within a super-system (the player-game system). While the game system captures all the internal components of the computer game itself, the player-game captures gameplay since it includes the play, the game, and the interaction between the two.

8.2.1 Cognition

Over the past several decades, researchers in cognitive science have been collecting evidence that objects external to the brain mediate and facilitate the operation of cognitive processes and activities (e.g., Clark & Chalmers, 1998; Hutchins, 1995). As a result, many of these researchers revised their models of cognition so that the brain is no longer viewed as an isolated information processor but is viewed as part of an extended and distributed cognitive system (see Clark, 2008). By 'extended' is meant that objects external to brain are included within the system, and by 'distributed' is meant that the operation of cognition is conceptualized as being distributed across multiple objects.

For example, consider a cognitive system composed of a human, a notebook, and a calculator. The human can read information recorded in the notebook and write into it, using the notebook as though it were an extension of her memory. Similarly, she could use the calculator to perform arithmetic calculations instead of performing them internally, augmenting her ability to calculate. Thus, memory can be seen as distributed across the human and the notebook, while arithmetic computation can likewise be seen as distributed across the human and the calculator.

Even though we cannot design the internal components of a human, we can design the other objects within the cognitive system. These objects can be designed to change the distribution of cognition within the cognitive system. For instance, the cognitive system in the example above would be very different if the human could not write in the notebook or if the calculator could also perform complex mathematical calculations instead of just simple ones. Furthermore, these external objects also change the way in which humans mentally operate (e.g., Kirsh & Maglio, 1994; Zhang & Norman, 1994). Carrying on with the same example, the complexity of calculations performed by the calculator would change how much internal mathematical work the human would do. Gradually, she would optimize her ability to use the calculator instead of being able to do the calculations herself, an effect referred to by Salomon and colleagues (1991) as 'cognitive residue'. In other words, the human component of the cognitive system gradually changes to optimize its ability to function within the system (Clark, 2008).

Therefore, we can influence and change the way in which people think through designing the objects with which they interact. In the context of games, the player-game system is also the cognitive system. Hence, we can design the game so as to change the distribution of cognition across the player and the game. Since, as was just discussed, such a distribution will occur regardless of how the game is designed, ideally we should be informed of what design decisions affect this distribution and the effect that occurs so as to prevent undesired distributions.

8.2.2 Computer Games

Thus far, the concepts discussed in this section are applicable to all games, regardless of their format, genre, number of players, or platform. From this point onward though, we will only focus on single-player computer games. In single-player games, it is easier to isolate cognitive influences since only one human is involved.

By computer game we mean a game that is implemented on some form of computational platform. Similar terms have been used in the literature (e.g., video game, digital game, electronic game, etc.), but we will treat them as synonymous with the term computer game. Although computer games require computational technology – such as a personal computer, game console, mobile device, or tablet – our interest is not the technology itself but rather the possibilities that it provides for the cognitive system. Two major sub-systems in any game are the representation sub-system, governing how the content in the game is represented to the player, and the interaction sub-system, which determines the interaction available between the player and the game (Haworth et al., 2013). Since both of these sub-systems can play a strong role in cognition (e.g., Sedig & Parsons, 2013; Zhang & Norman, 1994), their design would be the best to focus on when it comes to the cognitive system. Computational technology is malleable with respect to both representation and interaction. Hence, computer games have a much higher representative and interactive potential than non-computer games (e.g., physical sports, board games). For the greatest cognitive influence then, we should focus on the design of representation and interaction of computer games.

8.2.3 Representation Sub-System

Discussions surrounding the design of computer games often focus on the content to include, especially in the context of learning (e.g., Fisch, 2005; Moreno-Ger et al., 2008). Yet the player does not interact with content but a visual representation (VR) of it. VR refers to the visual form that content takes when it is displayed at the user interface (UI) of a computer game. Since it is only through VRs that the player has access to the content embedded in a computer game, the player makes no distinction between the content and the VR which encodes it (Cole & Derry, 2005).

However, the manner in which content is visually represented (i.e., the VR chosen for the content) has a significant influence on the operation of cognitive processes (e.g., Cox & Brna, 1995; Zhang & Norman, 1994). For example, we found that the external representation chosen for paths through a maze affected how participants navigated in a maze-based computer game (Haworth et al., 2010). In one case, participants were given a representation that implicitly encoded the paths. Participants with this representation exerted more effort discerning the correct path and frequently repeated themselves or ran into obstacles. In another case, participants were given a representation that explicitly encoded the paths in a tree-diagram. Participants with this representation avoided obstacles better and had fewer repeated mistakes. Although in this example one representation may be better than the other, typically we should choose a representation based on the cognitive effect that we want (for examples of how this can be done in the context of visual tools, see Parsons & Sedig, 2014a).

8.2.4 Interaction and Core Mechanic

Computer games are interactive, which means that there is a two-way dialogue between the player and the game (Kirsh, 2005; Haworth et al., 2013). As such, we can decompose the interaction of a computer game into two parts: action and reaction. The player performs an action on a VR and the game provides a reaction, typically some perceivable change of the UI—e.g., a VR changing in size or spatial position, or new VRs being created. Thus, the continual dialogue between the player and the game can also be seen as a repeated action-reaction cycle.

Interaction occurs at different levels of granularity (see Sedig et al., 2014). At the lowest level are physical actions and software or hardware events, such as tapping a touch screen or pressing buttons on a game-console controller. The number of possible interactions at this level is extremely high, given the large number of software implementations, hardware devices, and interaction techniques. At a higher level are patterns of interaction, such as selecting a group of units in a real-time strategy game, transforming a geometric shape in a puzzle game, or navigating an avatar through the world of a role-playing game. Characterizing interaction at the level of patterns allows a manageable set of interactions to be identified and discussed in a consistent manner. Furthermore, by using a defined list of interaction patterns we can consistently and systematically study their effect on cognitive gameplay. In this paper then, we will focus on interaction at the level of patterns, and will use an existing list of such patterns and their cognitive implications (Sedig & Parsons, 2013) to inform further discussion. Although this list is in the broader context of visual tools, it is still applicable to computer games.

Some interactions in a computer game are incidental, such as saving the current game state or adjusting game parameters like sound volume and difficulty level. However, computer games tend to include a set of one or more interactions that are essential for playing them. In other words, the act of playing a game involves continually performing a set of essential interactions, which we will call the ‘core mechanic’ (Salen & Zimmerman, 2004; Sicart, 2008). Since the design of interaction can influence the operation of cognitive processes, the design of the core mechanic will have the greatest interaction-related effect on cognitive gameplay (Haworth et al., 2013; Sedig, 2008).

8.2.5 Cognitive Gameplay

Gameplay refers to the continual performance of the core mechanic (Ang, 2006; Ermi & Mayra, 2005). The emphasis is not so much the core mechanic itself but rather the performance of the core mechanic: the actual dialogue that occurs between the player and the game, and the experience that emerges from this dialogue (Salen & Zimmerman, 2004). Given the conceptual confusion that has occurred with the term gameplay (Ermi & Mayra, 2005), some researchers have emphasized that gameplay primarily refers to the player’s experience by using terms such as gameplay experience (e.g., Ermi & Mayra, 2005; Nacke et al., 2010) or game experience (e.g., Poels et al., 2007). Thus, designing gameplay means designing the player’s experience.

However, gameplay is not only subjective but also multi-dimensional (Poels et al., 2007). As such, gameplay can be viewed as a composite and this is typically how it is measured. For instance, Ermi and Mayra (2005) explored the gameplay component of immersion, while Sweetser and Wyeth (2005) explored several additional components such as concentration and challenge. Law and Sun (2012) examined other components, such as tension and negative affect. There is also the cognitive component of gameplay—what we will call cognitive gameplay in this paper—which has been explored in specific contexts, such as cognitive challenge or learning (Connolly et al., 2012). More generally though, cognitive gameplay refers to the cognitive processes and operations that occur within the cognitive system during gameplay.

For example, consider the computer game *Bejeweled*. The goal of this game is to form rows of three or more jewels of the same color. When a row is formed, the player is awarded points and the row is removed from the play space. The core mechanic is composed of one interaction: swapping the position of two adjacent jewels (an instance of the arranging pattern). The cognitive gameplay of this game involves identifying patterns and, to an extent, planning. The player engages in planning because she can chain the formation of rows: the removal of one row causes other jewels to adjust their position in such a way that a new row is formed. Chaining gives the player more points, a reward mechanism to encourage planning. Hence, by designing *Bejeweled* so that chaining is possible and rewarded, its cognitive gameplay includes planning.

Despite planning being a possibility in *Bejeweled*, the cognitive gameplay experienced by the player may not actually involve it. Similarly, the player may engage in planning or pattern identification at a very shallow level, where little mental effort is exerted. Thus, it is insufficient to design only the ability to engage certain cognitive operations (i.e., the kind of cognitive gameplay); we also need to design the quality of the cognitive gameplay. By quality is meant, for example, the depth of cognition, amount of mental effort exerted, or amount of conscious awareness of the thinking that occurs.

How to design this quality remains an open research question. One possibility is that we can affect the quality of cognitive gameplay through changing the distribution of cognition within the player-game cognitive system. We explored this through designing the solution space of a computer game.

8.2.6 The Solution Space

The space through which the player navigates to construct a solution is called the solution space. This space can be conceptualized as a graph, where each node in the graph is a problem state, the links between nodes are transitions between states, and the path from the start state to the goal state is the solution (Russell & Norvig, 2009). When the player is engaged in problem solving, she is constructing a path through this graph and is transitioning between states through performing the core mechanic. Problem solving is part of cognitive gameplay, and the solution space is involved in problem solving. By carefully designing the solution space we can explore possible design decisions that would impact the quality of problem solving.

The solution space itself is distributed across the player and the game; some subset of the information within the space is represented internally (i.e., in the player's mind) and another subset is encoded by VRs of the computer game. How much information is distributed onto the player or the game might affect the amount of mental effort and depth of processing that can be performed as part of problem solving. Likewise, the way the player can interact with any VRs of the solution space might cause a similar effect. If this is true, it would offer some possibilities regarding how to design the quality of cognitive gameplay. It would also suggest that design decisions regarding the representation and interaction sub-systems do affect the quality of cognitive gameplay, and thus at least these sub-systems should be designed with a conscious awareness of that fact.

Two different representative distributions of the solution space were considered: *internally represented* and *externally represented*. An *internally represented* solution space would limit the amount of the solution space encoded into VRs, which should push most of the representation onto the mind of the player. An *externally represented* solution space would instead represent much of the solution space on the game, through VRs, reducing the amount which the player would need to keep in her mind.

Two different means of interaction were also considered: mutable and immutable solution space. A *mutable solution space* is one in which the player can change the VRs in some way, while an *immutable solution space* does not allow such changes.

8.3 Methodology

To explore the effect of different representative distributions and interaction means on the quality of cognitive gameplay, we designed a computer game. This computer game had three versions, in which the solution space was: 1) externally represented and mutable, 2) externally represented and immutable, and 3) internally represented. The mutable dimension only applies to externally represented solution spaces, since there are no VRs with which the player can interact in an internally represented solution space.

8.3.1 Studied Game: Laser Dilemma (LD)

Laser Dilemma (LD) is a 2D puzzle game with 20 levels of gradually increasing difficulty. In each level, the player is presented with a maze-like arrangement of objects: goal rings, a laser gun, walls, and reflector blocks. The laser gun fires an energy ball in one direction, which travels in a straight line until it collides with a wall or a reflector block. Collision with a reflector block changes the direction of the ball. If the ball collides with a goal ring, it activates the ring. The goal of each level is to activate all the goal rings before the ball collides with a wall. The player is awarded points for completing a level, with the number of points based solely on the difficulty of the level. The same levels were used across all three versions.

The core mechanic is composed of two interactions: the player can move reflector blocks around on a level (instance of the arranging pattern), and she can rotate these blocks so that they reflect the ball in a different direction (instance of the transforming pattern). Some reflector blocks are not interactive, and cannot be moved or rotated. By arranging the reflector blocks, the player is constructing a solution. She can test her solution at any time by clicking a button labeled ‘fire,’ which cases the laser gun to fire. If all the goal rings were activated, then the solution was correct; the player is awarded points and the game proceeds to the next level. However, if the solution is incorrect then the level will reset and the player will have to construct a new solution from scratch.

Figure 8-1: A screenshot of the game LD, specifically version LD-ERM.

8.3.1.1 Laser Dilemma – Externally Represented and Mutable Solution Space (LD-ERM)

In this version, the start state, the current problem state, and the constructed solution are all externally represented. The corresponding VRs are also interactive. The player constructs the solution through moving and rotating reflector blocks, and she can also revise her solution by moving or rotating blocks after they have been placed.

8.3.1.2 Laser Dilemma – Externally Represented and Immutable Solution Space (LD-ERI)

In this version, the same external representation is used as in version LD-ERM. However, the VRs for the constructed solution are not interactive. Although the player can interact with some VRs to construct a solution, she cannot revise her solution. The player can move and rotate reflector blocks as in version LD-ERM, and the position and orientation of all reflector blocks are always visible. However, once the player has placed a block it cannot be moved or rotated again (*unlike* version LD-ERM). If she wants to change the position or orientation of a block, she has to reset the level.

8.3.1.3 Laser Dilemma – Internally Represented Solution Space (LD-IR)

In this version, some of the solution space is internally represented. The start state and possible transitions between problem states are externally represented, but the constructed solution and current problem state are internally represented. This was implemented as follows.

The reflector blocks available to the player are initially visible. However, once she places a block somewhere it disappears from view. Hence, no VRs exist for the current problem state or the player's solution. When the player tests the solution, all of the placed reflector blocks reappear. However, since testing an incorrect solution resets the level, the player cannot change the position or orientation of placed blocks.

8.3.2 Study Procedure

We recruited fifteen university students (n=15) and randomly assigned them to one of the three game versions. The participants were almost equally split in terms of status (7 graduates, 8 undergraduates) and gender (8 female, 7 male). This study was divided into two sessions: a play session, and an interview session. All participants performed the play session, and 5 of them also performed the interview session. The procedures for the two sessions are described below.

8.3.2.1 Play Session Procedure

The play session was approximately one hour in length, involving three steps in this order: 1) participants filled out a demographics survey on paper, 2) participants played their assigned game version for 25 minutes, and 3) participants filled out a design questionnaire on paper. All participants were given the same set of version-specific instructions.

Four sources of data were collected and used to examine cognitive gameplay: 1) video-recordings of the game screen, to capture participants' in-game actions; 2) audio-recordings of the participants, to capture their verbal comments; 3) two paper surveys, a demographics survey and design questionnaire, to capture opinions about and experiences of playing their game version; and 4) direct observations of the participants playing the game.

The demographics survey gathered basic demographic data, such as age and gender, and game-playing behavior and preferences. The design questionnaire gathered participants' opinions of the game and its gameplay. Most of the questions were open-ended or contained an open-ended portion. For example: "Did you have difficulties with the controls? If yes, what were these difficulties?" and "Did the game interface make it easier to remember things? Why or why not?" Some questions used a 5-point expanded Likert scale for answers. For example, the question "How demanding was the game on your memory?" had a scale of 1 to 5, with 1 being written as 'no demand at all' and 5 written as 'more demand than I could handle'.

8.3.2.2 Interview Session Procedure

The interview session was approximately one hour in length, and it was performed two to three weeks after the participant performed the play session. During the interview, participants: were shown the other two versions of LD, were given an opportunity to briefly play them, were asked questions about them, and were asked to compare each version to the first one that they played. The questions were primarily open-ended in format and considered multiple perspectives of cognitive gameplay, such as perceived difficulty, anticipated effort required to plan, expected cognitive load for memory, and how they mentally constructed solutions. The interview format was semi-structured (see Lazar et al., 2010). Two sources of data collection were used in the interview: 1) audio-recordings, to capture the participants' comments; and 2) video-recordings of the game screen, to capture the actions performed by participants when shown the other versions.

8.3.2.3 Hypothesis

We made the following two hypotheses:

H1: The cognitive gameplay of LD-IR would be more distributed toward the player than the cognitive gameplay of LD-ERM or LD-ERI. In other words, participants who played

LD-IR would engage in more mental effort and more internal processing than participants of LD-ERM or LD-ERI.

H2: The cognitive gameplay of LD-ERI would be slightly more distributed toward the player than in LD-ERM. Participants of LD-ERI would engage in more pre-planning of actions (i.e., more careful play) than those who played LD-ERM.

8.4 Results

This section is divided into four sub-sections: one for each treatment group, and one for common results. The results include a combination of observations and recordings, verbal comments from both sessions, and survey responses. Participants were encoded using P<#> (e.g., P2, P11) during the study, and are referred to accordingly in the results. Comments from participants are verbatim except for removal of filler words (e.g., "uhh", "so", "like").

8.4.1 Common Results

In the design questionnaire, only three participants reported minor control problems (e.g., forgot that they could rotate reflectors, initially misunderstood how to move a reflector) and the rest reported no problems with the controls. All of the participants reported that they understood the graphical component of the UI without difficulty.

All participants, regardless of version, engaged in the activity of problem solving. They exerted mental effort determining the placement and orientation of reflectors necessary to complete each level. This activity is evident from their general behavior, excluding behavioral variations related to the kind of solution space. Verbal comments indicative of problem solving were also similar across all versions:

> "*Right now I'm thinking how to let the laser pass this one. […] If I put the stick like this it will just hit the wall, so I'm trying to think of a way around that.*" (P3, LD-ERM)

"*When you press fire, you basically know if you're going to solve it or not. It's very predictable. Obviously I know if it bounces off [this reflector] it's going to go a certain way.*" (P5, LD-ERI)

"*One thing that I am almost sure of it is that there should be a mirror here, because it's separated from all the rest. And there's no way to hit it from this other side because of the walls. There's no hole like this, since it's blocked here. I could go around it but I would lose two of my mirrors for that.*" (P6, LD-IR)

"*I think if something comes here, from here, I can cover this, this, this, and then this. […] So I need one, two, three [counting reflectors] … those are covered and then … four, five—oh, how many do I have? I have six. Hmm, that may not work.*" (P7, LD-ERM)

"[I'm] predicting the way the ball's gonna go. […] I'll start with the most logical path for the ball to follow in the first place and then I'll try to build off that. I'll probably look for a start point, like the first thing that it should be hitting, and then an end point, the last thing it should be hitting." (P8, LD-IR)

"*I'm just firing it in my mind. I feel like it's not enough to have just one [reflector, in order to solve the level]*." (P9, LD-ERI)

8.4.2 Version LD-ERM:

The major result for this version was that participants externalized problem solving.

8.4.2.1 Externalizing Problem Solving

Several observations and comments from participants indicate that the act of problem solving was aided by the VRs of the solution space. First, participants would use the mouse to trace the path they expected the laser to follow. Second, verbal comments during the play session indicated that the arrangement of reflectors aided participants in problem solving:

> "*What if I put this one here? It comes to here, to this one … oh, maybe this will work. [Place a reflector then traced a path] Yes, it does!*" (P4)

> "*I've been visualizing the orientation of the reflectors and, when the laser hits it, where it would go. I'm trying to start from the last circle I want to get to, and see how I can connect it from the last one to where I started. It's difficult. I'm thinking, initially it wants to go in here [pointing to one spot on the screen] … unless I do this [places a reflector]. Ah! I think this will work. Let's see. [Begins tracing a path]*" (P10)

Third, when interview participants were shown this version they mentioned that they would use the external representation to assist problem solving:

> "*Given both that you can see them [reflectors] and pick them back up again I would only engage in very cursory pre-planning before I started putting elements on the playfield, I think. I would outsource some of my planning to the playfield, plan actively rather than cognitively beforehand, and then try to execute my plan.*" (P1)

> "*[This version] offloads some of the mental processing and memory demand onto here [pointing to the screen]. Because, rather than having to keep the whole path in my mind at once, I have flexibility to put something here, trace part of it, realize that won't work, and then move it again. [Asked 'Do you think you would play the game that way?'] Probably, aside from the possible scenario of trying to challenge myself […] I would probably follow a similar strategy in which I would look at the whole thing, and try to devise a pathway. But rather than going carefully through every part of it I would maybe just place some of them there, and it would just offload some of that mental computation.*" (P12)

Fourth, some of the design questionnaire responses indicated that participants used the external representation for this purpose. For instance, in response to the question "Do you think the game interface made it easier for you to know the outcomes of your actions?"

participants answered mostly in the affirmative (1,2,2,2,4; with 1 meaning absolutely, 5 meaning definitely no) and explained:

> "*I was able to see where the laser won't fire and trace the path it took off each reflector to eventually reach the goal.*" (P10)

> "*Because the laser went in a straight line, it was easier to determine where it would go based on the objects on the screen*." (P15)

Fifth, participants frequently experimented with possibilities. They would place one or more reflectors, trace the path through them, and then continue constructing their solution or change the existing one; this included removing some reflectors and changing the position of others. Even though a reset button was available to clear the level, only one participant used it and did so to "*start the problem with a new approach*" (P10). The others would just refine their solution instead of clearing and restarting, or would create a new solution after they clicked the fire button to test their solution.

However, by externalizing problem solving in this way it is possible that participants were not engaging in careful problem solving. This can be seen in comments from interview participants who were assigned a different version and then shown this one:

> "*[This version is] easier. For example, if I did this [put a reflector in a bad orientation], I could rotate it and it would be fine. If it didn't work, I could do it again. I wouldn't have to think why it didn't work I could just use trial-and-error.*" (P9)

> "[Asked 'would you think differently in this version?'] *Oh yeah. I would just throw them down. I'd have so much less care. I'd just throw them down, make it very rough, and then go through it and refine it later. [Asked 'Would you still review it at the end?'] Of course, because I don't want to waste a fire. Mostly because it takes time, especially if it messed up at the start.*" (P5)

8.4.3 Version LD-ERI

The major results for this version were that participants externalized their memory on the VRs of the solution space and they internalized problem solving.

8.4.3.1 Externalizing Memory

Since participants were unable to modify their solution in this version, they behaved as though the VRs were a memory aid. They would place reflectors, as a way of externally constructing their solution, but would then still work the solution in their mind. This could be seen in three ways. First, participants would pick up a reflector and hold it in some location on the playfield but not place it. Since they did not place it, they could still move it to another location. Participants then used their hand to trace the path of the laser. The mouse was still used to trace paths, as in version LD-ERM, but only among reflectors already placed.

Second, in the design questionnaire participants indicated that they used the reset button frequently (2,2,3,3,5; where 1 is all the time and 5 is never). Sometimes this was done because a reflector was misplaced: "*Most of the time just a slight arrangement change would lead to the solution.*" (P14) Other times, it was because they wanted to create a new solution: "*Already put down pieces, so [I] needed to move them with reset button.*" (P9) Although they would use the fire button to test their solution, they would still reset without clicking the fire button. This suggests participants had already realized the outcome of their solution and saw no need to formally test it.

Third, participants explained in the interview that they used only the VRs to offload their memory. Some of the best comments regarding this are:

> "*[Compared to version LD-IR, in this version] the load on your memory is a bit diminished. […] I think the amount of planning that you have to do is still relatively high. You still have to have a relatively complete notion of what you're doing before you start placing pieces, but I think the memory component of it is a little bit more manageable; because at least you can outsource some of it to the play field.*" (P1)

> "*The thing I'm trying to visualize is the pathway for the laser. […] I want to be able to go from start to finish along the pathway, in my head, while looking at this. And then what I have to keep in memory, as I do that, is where have I gone so far along the path and, as part of the planning, how many reflectors do I have? How many I have used? What types do I have? Keeping that in my memory, I get to this point and say ok, what do I need? Do I have it available? But most of the reflector memory is offloaded onto here, because I can see it.*" (P12)

8.4.3.2 Internalizing Problem Solving

Even though participants could use VRs for recording the placement of reflectors, they still indicated that most of the problem solving was done internally. For example, P9 wrote in the design questionnaire that the game was demanding on his memory. However, in explaining why this was case, he wrote:

> "*For certain levels, it seemed like it would be easy, and beneficial for me to make plans first, like which route to take, but I just did 'trial-and-error' in my mind a lot, which was why it was so demanding on my memory!*"

Verbal comments from the participants indicated that they needed to do problem solving internally because they could not revise their solution in the game:

> "*I have to look at the whole picture, think about which ones I want to use, and then where to put them. […] In some ways it's good that you can only put [reflectors] down once, because it really makes you think about what you're going to do.*" (P5)

> "*[Not being able to pick up the reflectors] should be part of the game. If you could move everywhere it'd be too easy, there would be no memory involved. […] [If I could move them] I'd just continually click fire, and try it, and move it from here to here, I wouldn't need to reset at all. So, it'd be really quick and on to the next level. I wouldn't really think that [much].*" (P9)

> "*You really have to think about your whole plan before you start placing things. […] It's a little bit more challenging since you weren't able to move [the reflectors]*." (P10)

Interview participants, when shown this version, also expressed the same thoughts about how they would approach playing it:

> "*[Compared to version LD-IR] you get the advantage of being able to see and not have to remember where everything is, [but compared to version LD-ERM] you lose the ability to experiment the same way. It's a final choice […] With how many elements you're obviously going to have to place to solve this [difficult] puzzle, having them all be permanent is a bit problematic but at least you can work gradually towards a solution. Because if you're confident that your initial placement is correct, once you place it there it becomes a feature of the level, it's no longer part of what you're doing and so you can stop thinking about that and it seems to simplify the problem.*" (P1)

> "*The way that I would solve the level, I would do it all ahead of time before placing any of the reflectors. So, by the time I placed one, it should already be the optimal solution and I don't have to think much more about it […] it was already solved as far as I was concerned, and I was just putting them in the place they need to be.*" (P12)

8.4.4 Version LD-IR

The major results for this version were that participants internalized their memory, and they internalized problem solving.

8.4.4.1 Internalizing Memory

Compared to versions LD-ERM and LD-ERI, where memory was externalized onto VRs, the participants needed to keep more in their memory. They behaved and commented about having to spend significant mental effort on remembering:

"*I've lost track of what's going to happen now […] Even though it's clearly the blue one in the field, once you place it you don't remember whether you've got a [blue one] or not. You can easily forget, even with only a few pieces. It's deceptive how easy it is to lose track of that.*" (P1)

"*With only one or two [reflectors] it doesn't really cause any trouble; once there is more than that I need more concentration. […] It requires more memory, and I need to concentrate in order to remember better.*" (P6)

As was the case for LD-ERI, some participants would hold pieces but not place them and then use their hand to trace paths. Also, participants indicated in the design questionnaire that having no external representation of their solution made the game difficult and mentally taxing:

"*Much of the challenge was a function of the interface not the underlying task. Disappearance of the pieces was the single biggest contributor to the difficulty.*" (P1)

"*It was a challenge to remember the locations of objects and clearly a crucial part of the game.*" (P8)

When interview participants were shown LD-IR they discussed how they expected it to pose a much higher demand on their memory:

"*I know something has to go here, but if I can't see where I put it I might think 'did I put it there already?' and if I didn't then I might fire and if that happens it fails. So I have to remember all the places I put it and try again.*" (P9)

"*That one [LD-IR] obviously places more burden on the memory of the player. […] For some of these lower levels, in which it is possible to maintain those in memory, it's certainly more challenging and it could be enjoyable if the player really likes the difficult challenge. […] I wouldn't have the same visual cues. I wouldn't be able to visualize the trajectory or pathway as easily. I'd have to consciously try to remember where the reflector was, even if I know I put it right* *around somewhere. I'd still probably remember; I'd just have to exert that effort to determine it.*" (P12)

8.4.4.2 Internalizing Problem Solving

Although participants engaged in problem solving, they attempted to construct as much of the solution in their mind as they could and then test their solution. Three of the participants would construct the solution in pieces and test each piece, but this was difficult to do as they often forgot parts of the solution. The other participants just attempted to construct a complete solution, then implement their solution and test it. All of the participants tried to construct as much of the solution as they could in their mind first:

> "*I'm trying to actually trying to simulate the entire thing in my mind, based on where it's going to go before I make a change. And then, what changes I can make to get it to go where I want.*" (P1)

> "*[I'm] predicting the way the ball's gonna go. It's tough to visualize it, I'd say [my problem solving uses] a combination of trial-and-error and just intense forethought before I press the actual fire button.*" (P8)

Similar comments could be found in the design questionnaire. For instance, in response to the question "Do you think the game interface made it easier for you to know the outcomes of your actions?" P8 wrote: "*It required quite a bit of focus and concentration to predict the outcome while the pieces were invisible.*" In response to the question "What did you find challenging?" P6 wrote "*Planning ahead of time to find the path that was dependent on multiple variables.*"

In the interview, P1 further reflected on the cognitive gameplay that he experienced and elaborated on his previous comments:

> "*I tried to formulate as much of the plan as I could, because I knew that the second I put something down I would forget where it was. So, I tried to come up with the whole thing and know exactly what I was doing, which is somewhat*

> *harder to do. […] Even if I could just tell where they were and not have to know what state they were in it would be less cognitive load for me I think. […] It's hard to keep track of—if I make a small part of a plan, like I want a bidirectional gate here so I'm going to put this in one position or the other, and then I'm going to put it down, and now it's gone, and so I know that there's something here, I'm probably going to remember that for the duration of the time that I'm putting pieces down … but then, these two are pretty close so, I don't know if that's actually going to interfere with the other piece that I put down or not, I can't tell.*"

When other interview participants were shown this version, they also indicated that playing this version would require significant mental resources:

> "*[it's] very hard and I wouldn't be able to plan as well. I couldn't know what my plan was, I'd be shooting in the dark each time. I could be like 'I know it will probably come out here, and I can put one here and there,' so now I've made my plan and I put it down. I guess I laid it out in my mind, but it was uncomfortable. Because if I messed up a little bit, I'd have to go through my whole plan.*" (P5)

> "*I think [the way I would solve the level] may not actually change much. Because [in LD-ERI] I would solve it in my head ahead of time, before placing any of the reflectors for the most part, and then I would place them. So I would know where they should be, and it wouldn't matter so much that I couldn't see them. But it's still a bit more demand on working memory.*" (P12)

8.5 Conclusion

In this paper, we discussed the design of cognitive gameplay. This included the theory behind cognitive gameplay, and its manipulation through designing the representation and interaction sub-systems of a computer game. We also presented empirical evidence of this with the example of the solution space of a 2D puzzle-based computer game. Study participants who played this game experienced different cognitive gameplay depending on the way in which the solution space was designed. The difference was in terms of the mental effort the participants exerted and the way in which cognitive processes were distributed between the player and the game.

For instance, when the solution space was externalized (in versions LD-ERM and LD-ERI) participants offloaded their solution onto VRs. Rather than keep the entirety of the plan in their mind they would place some of the reflectors onto the play space, so that they could think about and review their plan more easily and reduce the cognitive load on their memory. In contrast, when the solution space was internalized (version LD-IR) participants experienced a much higher cognitive load with respect to memory. Participants experienced difficulty remembering what reflectors they had already placed (i.e., difficulty remembering the solution as they constructed it), as well as difficulty keeping the solution in their mind long enough to solve the harder problems. Hence, whether the solution space was internalized or externalized influenced the distribution of working memory, with respect to multiple aspects of problem solving (e.g., analyzing the problem, constructing a solution, implementing and testing the solution).

Similarly, whether the solution space was mutable or immutable (i.e., whether participants could change their solution as they constructed it) also affected cognitive gameplay. When the solution space was mutable (version LD-ERM) participants used the VR as an extension of their working memory as well as to enhance their problem solving ability. Various parts of the problem solving process (e.g., analyzing the problem, testing solutions, etc.) were offloaded onto the UI, as shown in the way that participants used VRs to solve each level. Although this led to a more haphazard and less careful approach to problem solving, it did require less mental effort. In contrast, when the solution space was immutable (version LD-ERI) participants offloaded less of the problem solving process onto the UI. For instance, participants used VRs to record their solution and visually test it, but they constructed and revised their solution mentally. Hence, playing LD-ERI involved more mental effort and participants were more careful in the construction of their solution than participants who played LD-ERM.

Several limitations to this study should be mentioned. First, we primarily assessed cognitive gameplay through self-report measures. This was acceptable, since we were more interested in the participants' subjective opinion of their cognitive engagement than whether any cognitive improvement occurred. However, future studies could include more objective measures of cognitive performance. Second, future studies should have more participants so that greater statistical power can be achieved. The current study had too few participants for any quantitative results to be statistically significant in a general population, though the qualitative results are still useful. And third, this study only tested one game. Although a difference in cognitive gameplay was identified, this difference may have been caused by the particular implementation. Thus, future studies should include multiple games to help reduce implementation-specific confounds in the results.

Therefore, in summary, this paper provides evidence and theory that guides us in the design of cognitive gameplay. We can clearly see that it is possible to manipulate the distribution of cognition and influence the depth of thinking that occurs in the player. As such, this line of research should be pursued further so as to better understand the relationship between the design of computer games and the related cognitive processing that occurs within the player.

Chapter 9: Conclusion and Future Work

In this dissertation, we presented a conceptual framework for the design of cognitive gameplay in computer games. Each chapter focused on one or more particular components of the framework. This final chapter is divided into three sections: 1) the dissertation is summarized and the main purpose of each chapter is presented, 2) some general conclusions are drawn with respect to how this dissertation contributes to the wider scientific literature, and 3) areas of future research are identified.

9.1 Dissertation Summary

In **Chapter 2**, we discussed a high-level overview of our framework, presenting some of the framework's main components: cognition, representation, interaction, the core mechanic, and interactivity. These are topics about which designers should be cognizant, so as to make appropriate decisions for designing cognitive gameplay. To argue this point, we provided an example of how thinking about these main components can assist in designing cognitive gameplay.

In **Chapter 3**, we described the most common interaction patterns that can be found in games. We also discussed the high-level cognitive activities that each pattern enables and supports, as evidence that the choice of interaction pattern can influence cognitive gameplay. In addition, we discussed how these patterns could be used as a means for categorizing and organizing computer games. This is more conducive to recognizing the high-level cognitive activities in which the game might engage the player than existing means of categorization (e.g., game genres).

In **Chapter 4**, the focus changed to the core structure of a computer game. We first discussed how General Systems Theory can be used to conceptualize, analyze, and design the inner structure of a computer game as well as identify game isomorphisms at particular levels of abstraction. Then, we discussed how the inner structure can also be described in terms of rules, and that there can be rules at different levels of abstraction. Lastly, we discussed a process for creating isomorphic games, using the concepts of General Systems Theory, rules of different levels of abstraction, and cognitive toys. By using cognitive toys as a source of inspiration, designers can start with some mentally-engaging and enjoyable task and use the rest of the process to mold this task into a game with cognitive gameplay.

We returned to interaction in **Chapter 5**, but instead of examining general interaction patterns as in Chapter 3 we focused on the core mechanic of a game, and how the operationalization of its structural elements gives rise to interactivity. Since interactivity influences the quality of cognitive gameplay, design of interactivity is just as important as the choice of interaction. Twelve structural elements were identified and their potential impact on the quality of cognitive gameplay was discussed.

Then, in **Chapter 6**, we showed how the design process from Chapter 4 could be used to empirically investigate cognitive gameplay. Starting from a cognitive toy, two isomorphic computer games were produced. These games differed by only one structural component. Since this component had been isolated, two possible instantiations of that component could be studied as two values of a dependent variable. The chapter ended with an example of how such a study could be conducted, with some data that suggests the studied structural component (i.e., rules) affects cognitive gameplay.

In **Chapter 7**, we used the same method of empirically investigating cognitive gameplay from Chapter 6. This time, one of the structural elements identified in Chapter 5 was investigated (Activation) and the study was more thorough than the example presented in Chapter 6. The results indicated that the operationalization of activation seemed to influence cognitive gameplay, in terms of the degree of mental effort required to engage in problem solving.

In **Chapter 8**, we introduced the last part of the framework: the solution space. We showed how the distribution of the solution space can be explicitly manipulated through the design of the user interface, and validated this claim with some empirical evidence. In doing so, we provided strong evidence for the capacity of computer games to distribute cognition. Since the investigation of the solution space used the method from Chapter 6, Chapter 8 provided further support that this method can be used to design and test any component of the framework and not only the structure of interaction.

9.2 Scientific Contributions

As mentioned in the introduction, the current research on the design of cognitive gameplay is divided up into multiple isolated disciplines. Synthesizing this research into a conceptual framework would greatly benefit the games research community, as it would provide a common language to bridge the isolated disciplines. The framework presented in this dissertation is an example of such a framework; it combines research from multiple disconnected disciplines, abstracts the concepts to a more universal level, and provides a common language. Hence, one contribution of this dissertation is the framework as an example of how other such frameworks could be developed.

In addition, the framework presented in this dissertation is itself a contribution to the scientific literature. Until now, there were no frameworks for the conscious and systematic design of cognitive gameplay. Furthermore, this framework addresses multiple aspects of game design. It provides guidance at both the macro- (e.g., game design process) and micro-level (e.g., particular game components). Multiple components are discussed (e.g., rules, interaction, representation, the core mechanic) along with the potential of each component for influencing cognitive gameplay. Some of these components were also empirically tested using random-control trials; studying computer games in this way is uncommon but is something game research community needs (Connolly et al., 2012; Ravaja & Kivikangas, 2009).

Lastly, the framework's high level of abstraction, such as its emphasis on cognitive gameplay over more specific cognitive activities, gives it a more universal applicability. For example, a designer wanting to make a computer game that is conducive to learning mathematics could use this framework. However, another designer wanting to make a computer game to improve the decision-making skills of managers could also use this framework. A third designer interested in a computer game for regular entertainment could also use this framework to incorporate mentally-taxing challenges in his game. In other words, people in all of the popular game-design contexts (e.g., education, entertainment, health, and training) would benefit from the same framework. In addition, this framework complements rather than supersedes existing design frameworks. This enables designers to get the best advance of more specific frameworks with the general principles expressed in this framework. For example, someone designing a computer game for learning could combine a framework discussing the components needed for learning (e.g., Aleven et al., 2012) with our framework to design the game's interaction.

9.3 Future Research

The framework presented in this dissertation is sufficient for both researchers and practitioners, yet there are still many ways in which it could be expanded. Although the framework currently contains significant detail on interaction and interactivity, less detail is available when it comes to other components (e.g., mechanics, representation, rules). Previous research on some components has already been conducted, but the concepts and terminology of our framework would need to be adjusted to incorporate it.

Another important area of research is to better incorporate this framework with research on specific cognitive activities (such as learning) and methods for studying the performance of those activities. The method provided in our framework is sufficient for controlling the study variables, but it may not be clear for researchers outside of cognitive science and psychology how to measure cognitive gameplay in a more detailed way. Further studies that combine this framework with other methods for measuring cognition would greatly improve its usefulness, and facilitate better incorporation with existing game research in disciplines other than computer science.

There is also future work needed to improve the prescriptive utility of the framework for development. Although several components have been identified, the framework lacks corresponding lists and/or taxonomies of them. For instance, cognitive toys have been suggested as a tool for starting the design process, but no comprehensive list or taxonomy of cognitive toys was provided; this limits the ability of developers to use cognitive toys in practice. As another example, a list was provided in Chapter 3 of some interaction patterns and their relationship to particular cognitive activities. No such list was provided for rules or representations. Similarly, no list of cognitive activities was provided (aside from a brief mention in Chapter 3). Such a list would be especially helpful for designers, but also for researchers wishing to study more particular aspects of cognitive gameplay.

As another area of future research, this framework needs better integration with the terminology employed by game developers in the industry. Although many of familiar terms have been included in the framework and clearly defined (e.g., core mechanic, rules, and interaction) others related to the implementation of computer games have not been included (e.g., challenge, data, game state, goals, levels). In addition, the concept of "game mechanic," which is prevalent in the industry, should also be included (Sicart, 2008). Such mechanics need to be: defined within the common language of this framework, enumerated and categorized, and their potential cognitive utility identified. This research would greatly advance the utility of this framework within the industry, while also improving its utility in other areas of research.

Finally, more empirical studies need to be conducted on the framework's various components. Although the studies included in this dissertation are enlightening, more studies are needed before particular claims regarding cognitive performance can be made. In particular, specific cognitive activities should be better tested and longitudinal studies should also be conducted to compare the use of this framework with the desires of researchers in other communities (e.g., games to fight cognitive decline).

References

Aarseth, E. (2003). Playing research: Methodological approaches to game analysis. In B. K. Walther (Ed.), *5th International Digital Arts and Culture Conference*, Fine Art Forum.

Aarseth, E., Smedstad, S. M., and Sunnanå, L. (2003). A multi-dimensional typology of games. In *Proceedings of the 1st Digital Games Research Association International Conference* (pp. 48-53), Utrecht, The Netherlands.

Abrahamson, D. (2006). Mathematical representations as conceptual composites: Implications for design. In *Proceedings of the Twenty Eighth Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education* (pp. 464-466), Mérida, Yucatán, México: Universidad Pedagógica Nacional.

Adams, E. (2010). *Fundamentals of Game Design* (2nd ed.). Berkeley, CA: Pearson Education.

Adams, E., and Dormans, J. (2012). *Game Mechanics: Advanced Game Design*. Berkeley, CA: New Riders Press.

Aldrich, F., Rogers, Y., and Scaife, M. (1998). Getting to grips with "interactivity": Helping teachers assess the educational value of CD-ROMS. *British Journal of Educational Technology*, *29* (4), 321-332.

Alessi, S. M., and Trollip, S. R. (2001). *Multimedia for Learning: Methods and Development* (3rd ed.). Boston, MA: Allyn and Bacon.

Aleven, V., Myers, E., Easterday, M., and Ogan, A. (2010). Toward a framework for the analysis and design of educational games. In *Proceedings of DIGITEL 2010* (pp. 69-76). IEEE.

Alexander, C., Ishikawa, S., and Silverstein, M. (1977). *A pattern language: Towns, buildings, construction*. New York, NY: Oxford University Press.

Alexander, J. T., Sear, J., and Oikonomou, A. (2013). An investigation of the effects of game difficulty on player enjoyment. *Entertainment Computing*, *4* (1), 53-62.

Alpert, F. (2007). Entertainment software: Suddenly huge, little understood. *Asia Pacific Journal of Marketing and Logistics*, *19* (1), 87-100.

Anderson, J., and Rainie, L. (2012). The future of gamification (Pew Internet & American Life Project Report). Retrieved from http://pewinternet.org/Reports/2012/Future-of-Gamification.aspx

Ang, C. S. (2006). Rules, gameplay, and narratives in video games. *Simulation & Gaming*, *37* (3), 306-325.

Ankerst, M., Berchtold, S., and Keim, D. A. (1998). Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings of the IEEE Symposium on Information Visualization* (pp. 52-60).

Apperley, T. H. (2006). Genre and game studies: Toward a critical approach to video game genres. *Simulation & Gaming*, *37* (1), 6-23.

Barab, S., Thomas, M., Dodge, T., Carteaux, R., and Tuzun, H. (2005). Making learning fun: Quest Atlantis, a game without guns. *Educational Technology Research and Development*, *53* (1), 86-107.

Barr, P., Noble, J., and Biddle, R. (2007). Video game values: Human-computer interaction and games. *Interacting with Computers*, *19* (2), 180-195.

Bates, M. J. (1986). An exploratory paradigm for online information retrieval. In *Proceedings of the 6th International Research Forum in Information Science* (91-99), London.

Beck, K., and Cunningham, W. (1987). Using pattern languages for object-oriented programs. In *Workshop on the Specification and Design for Object-Oriented Programming* (OOPSLA-87), Florida.

Bedwell, W. L., Pavlas, D., Heyne, K., Lazzara, E. H., and Salas, E. (2012). Toward a taxonomy linking game attributes to learning: An empirical study. *Simulation & Gaming*, *43* (6), 729-760.

Belman, J., and Flanagan, M. (2010). Designing games to foster empathy. *The International Journal of Cognitive Technology*, *14*(2)-*15*(1), 11-21.

Björk, S., and Holopainen, J. (2005). *Patterns in game design*. Hingham, MA: Charles River Media.

Björk, S., and Peitz, J. (2007). Understanding pervasive games through gameplay design patterns. In *Proceedings of the 3rd Digital Games Research Association International Conference*, University of Toyko, Japan.

Björk, S., Lundgren, S., and Holopainen, J. (2003). Game design patterns. In *Proceedings of the 1st Digital Games Research Association International Conference* (pp. 180-193), Utrecht, The Netherlands.

Blumberg, C., and Ismailer, S. S. (2009). What do children learn from playing digital games? In U. Ritterfeld, M. J. Cody, and P. Vorderer (Eds.), *Serious Games: Mechanisms and Effects* (pp. 131–142). New York, NY: Routledge.

Borchers, J. O. (2001). A pattern approach to interaction design. *AI & Society*, *15* (4), 359-376.

Bottino, R. M., Ferlino, L., Ott, M., and Tavella, M. (2007). Developing strategic and reasoning abilities with computer games at primary school level. *Computers & Education*, *49*, 1272-1286.

Brey, P. (2005). The epistemology and ontology of human-computer interaction. *Minds and Machines*, *15*, 383-398.

Brown, J. S., Collins, A., and Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, *18* (1), 32-42.

Bucy, E. P. (2004). Interactivity in society: Locating an elusive concept. *The Information Society*, *20* (5), 373-383.

Burgoon, J. K., Bonito, J. A., Bengtsson, B., Cederberg, C., Lundeberg, M., and Allspach, L. (2000). Interactivity in human-computer interaction: A study of credibility, understanding, and influence. *Computers in Human Behavior*, *16*, 553-574.

Burke, J. W., McNeill, M. D. J., Charles, D. K., Morrow, P. J., Crosbie, J. H., and McDonough, S. M. (2009). Serious games for upper limb rehabilitation following stroke. In *Proceedings of the 2009 conference in Games and Virtual Worlds for Serious Applications* (pp. 103-110). Washington, DC: IEEE.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M. (1996). *Pattern-oriented software architecture: A system of patterns*. New York, NY: John Wiley and Sons.

Caldwell, N. (2004). Theoretical frameworks for analysing turn-based computer strategy games. *Media International Australia Incorporating Culture and Policy*, *110*, 42-51.

Callele, D., Neufeld, E., Schneider, K. (2010). A proposal for cognitive gameplay requirements. In *Fifth International Workshop on Requirements Engineering Visualization* (pp.43-52). Sydney, Australia: IEEE. doi:10.1109/REV.2010.5625658

Campbell, T., Ngo, B., and Fogarty, J. (2008). Game design principles in everyday fitness applications. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work* (pp. 249-252), New York, NY: ACM Press.

Chmiel, M. (2010). Game design towards scientific literacy. *The International Journal of Cognitive Technology*, *14*(2)-*15*(1), 32-42.

Chu, Y., and MacGregor, J. N. (2011). Human performance on insight problem solving: A review. *Journal of Problem Solving*, *3*, 119-150.

Clark, A. (2008). *Supersizing the Mind: Embodiment, Action, and Cognitive Extension*. New York, NY: Oxford University Press.

Clark, A., and Chalmers, D. (1998). The extended mind. *Analysis*, *58* (1), 7-19.

Cohen, J. (1992). A Power Primer. *Psychological Bulletin*, *112* (1), 155-159.

Cole, M., and Derry, J. (2005). We have met technology and it is us. In R. Sternberg and D. Preiss (Eds.), *Intelligence and technology: The impact of tools on the nature and development of human abilities* (pp. 210-227). Mahwah, NJ: Lawrence Erlbaum.

Connaughton, S. P., Tache, K., and Burley, D. V. (2010). Taupita: A 3000-year-old shell game in the Lapita cultural complex of Tonga. *Journal of Social Archaeology*, *10*, 118-137.

Connolly, T. M., Boyle, E. a., MacArthur, E., Hainey, T., and Boyle, J. M. (2012). A systematic literature review of empirical evidence on computer games and serious games. *Computers & Education*, *59* (2), 661-686.

Costikyan, G. (2002). I have no words & I must design: Toward a critical vocabulary for games. In F. Mäyrä (Ed.), *Proceedings of Computer Games and Digital Cultures Conference* (pp. 9-33).

Cowley, B., Charles, D., Black, M., and Hickey, R. (2008). Toward an understanding of flow in video games. *Computers in Entertainment*, *6* (2), 1-27.

Cox, R. (1999). Representation construction, externalised cognition and individual differences. *Learning and Instruction*, *9* (4), 343-363.

Cox, R., and Brna, P. (1995). Supporting the use of external representations in problem solving: The need for flexible learning environments. *Journal of artificial intelligence in education*, *6* (2/3), 239-302.

Cox, A., Cairns, P., Shah, P., and Carroll, M. (2012). Not doing but thinking: The role of challenge in the gaming experience. In *Proceedings of SIGCHI 2012* (pp. 79-88), ACM Press.

Crawford, C. (1984). *The Art of Computer Game Design*. Berkeley, CA: Osborne/McGraw-Hill.

Crawford, C. (2003). *Chris Crawford on Game Design*. Thousand Oaks, CA: New Riders Publishing.

Crute, T. D., and Myers, S. A. (2007). Sudoku puzzles as chemistry learning tools. *Journal of Chemical Education*, *84*, 612-613.

Danesi, M. (2002). *The puzzle instinct: The meaning of puzzles in human life*. Bloomington, IN: Indiana University Press.

De Aguilera, M., and Mendiz, A. (2003). Video Games and Education: Education in the Face of a "parallel school". *Computers in Entertainment*, *1* (1), 10-23.

Dearden, A., and Finlay, J. (2006). Pattern languages in HCI: A critical review. *Human-Computer Interaction*, *21* (1), 49-102.

Delaney, P. F., Sahakyan, L., Kelley, C. M., and Zimmerman, C. A. (2010). Remembering to forget: The amnesic effect of daydreaming. *Psychological science*, *21* (7), 1036-1042.

Deterding, S., Sicart, M., Nacke, L., O'Hara, K., and Dixon, D. (2011). Gamification: Using game design elements in non-gaming contexts. In P*roceedings of the 2011 Annual Conference Extended Abstracts on Human Factors in Computing Systems* (pp. 2425-2428).

DeVane, B., Durga, S., and Squire, K. (2010). "Economists who think like ecologists": Reframing systems thinking in games for learning. *E-Learning and Digital Media*, *7*, 3-20.

Dipietro, M., Ferdig, R. E., Boyer, J., and Black, E. W. (2007). Towards a framework for understanding electronic educational gaming. *Journal of Educational Multimedia and Hypermedia*, *16* (3), 225-248.

Domagk, S., Schwartz, R. N., and Plass, J. L. (2010). Interactivity in multimedia learning: An integrated model. *Computers in Human Behavior*, *26* (5), 1024-1033.

Dondlinger, M. J. (2007). Educational video game design: A review of the literature. *Journal of Applied Educational Technology*, *4* (1), 21-31.

Dou, W., Ziemkiemkiewicz, C., Harrison, L., Jeong, D., Ryan, R., Ribarsky, W., and Chang, R. (2010). Comparing different levels of interaction constraints for deriving visual problem isomorphs. In *Symposium on visual analytics science and technology* (pp. 195-202). Piscataway, NJ: IEEE.

Dourish, P. (2004). *Where the action is: The foundations of embodied interaction*. Cambridge, MA: MIT Press.

Drack, M., and Schwarz, G. (2010). Recent developments in general system theory. *Systems Research and Behavioral Science*, *27*, 601-610.

Elliott, J., Adams, L., and Bruckman, A. (2002). No magic bullet: 3D video games in education. In *Proceedings of International Conference of the Learning Sciences*. Seattle, WA.

Elverdam, C., and Aarseth, E. (2007). Game classification and game design: Construction through critical analysis. *Games and Culture*, *2* (1), 3-22.

Ermi, L., and Mäyrä, F. (2005). Fundamental components of the gameplay experience: Analysing immersion. In *Proceedings of DiGRA 2005*.

Faisal, S., Attfield, S., and Blandford, A. (2009). A classification of sensemaking representations. In *Proceedings of CHI 2009*, *a Workshop on Sensemaking*. Boston, MA: ACM Press.

Falstein, N. (2002). Better by Design: The 400 Project. *Game Developer Magazine*, *9* (3), 26.

Fisch, S. M. (2005). Making educational computer games educational. In *Proceedings of the 2005 conference on interaction design and children* (pp. 56-61). New York, NY: ACM.

Foster, G., and Stefik, M. (1986). Cognoter: Theory and practice of a colab-orative tool. In *Proceedings of the 1986 ACM Conference on Computer-Supported Cooperative Work* (pp. 7-15). Austin, TX: ACM.

Fowler, M. (1996). *Analysis patterns: Reusable object models*. Boston, MA: Addison-Wesley Professional.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software*. Westford, MA: Addison-Wesley Professional.

Gardner, M. (2001). *A Gardner's workout: Training the mind and entertaining the spirit*. Natick, MA: A. K. Peters.

Gee, J. P. (2007). *What video games have to teach us about learning and literacy* (2nd ed.). New York, NY: Palgrave MacMillan.

Gentner, D., Loewenstein, J., and Hung, B. (2007). Comparison facilitates children's learning of names for parts. *Journal of Cognition and Development*, *8* (3), 285-307.

Geurts, J. L. A., de Caluwé, L. I. A., and Stoppelenburg, A. (2000). *Changing organisations with gaming/simulations*. The Hague, Netherlands: Elsevier.

Gillings, R. J. (1962). Problems 1 to 6 of the Rhind Mathematical Papyrus. *The Mathematics Teacher*, *55*, 61-65.

Gordon, A., Lent, M. V., and Velsen, M. V. (2004). Branching storylines in virtual reality environments for leadership development. In R. Hill & N. Jacobstein (Eds.), *Proceedings of the Sixteenth Innovative Applications of Artificial Intelligence Conference* (pp. 844-851). Palo Alto, CA: AAAI Press.

Green, C. S., Bavelier, D., and In, T. A. (2006). The cognitive neuroscience of video games. *Digital Media: Transformations in Human Communication*, 211-223.

Gros, B. (2007). Digital games in education: The design of games-based learning environments. *Journal of Research on Technology in Education*, *40* (1), 23-38.

Grossen, B., and Carnine, D. (1990). Diagramming a logic strategy: Effects on difficult problem types and transfer. *Learning Disability Quarterly*, *13* (3), 168-182.

Gunaratne, S. A. (2008). Understanding systems theory: Transition from equilibrium to entropy. *Asian Journal of Communication*, *18*, 175-192.

Habgood, M. P. J. (2007). *The effective integration of digital games and learning content*. (Doctoral dissertation). University of Nottingham, United Kingdom.

Habgood, M. P. J., and Ainsworth, S. E. (2011). Motivating children to learn effectively: Exploring the value of intrinsic integration in educational games. *Journal of the Learning Sciences*, *20* (2), 169-206.

Hannafin, M. J., and Hooper, S. R. (1993). Learning principles. In M. Fleming & W. H. Levie (Eds.), *Instructional message design: Principles from the behavioral and cognitive sciences* (2nd ed., pp. 191-231). Englewood Cliffs, NJ: Educational Technology Publications.

Haworth, R., and Sedig, K. (2011). The importance of design for educational games. In A. Méndez-Vilas (Ed.), *Education in a technological world: Communicating current and emerging research and technological efforts* (pp. 518-522). Badajoz, Spain: Formatex Research Center.

Haworth, R., and Sedig, K. (2012). Designing Digital Cognitive Games that Facilitate Mindful Reasoning and Decision Making. In H. Yang & S. Wang (Eds.), *Cases on e-learning management: Development and implementation* (pp. 196-222). Hershey, PA: IGI Global.

Haworth, R., Parsons, P., and Sedig, K. (2013). Design of digital cognitive games: Some considerations. *The International Journal of Cognitive Technology*, *18* (1), 22-27.

Haworth, R., Tagh Bostani, S. S., and Sedig, K. (2010). Visualizing decision trees in games to support children's analytic reasoning: any negative effects on gameplay?. *International Journal of Computer Games Technology*, *2010*, 1-11. http://dx.doi.org/10.1155/2010/578784

Hofkirchner, W., and Schafranek, M. (2011). General system theory. In C. Hooker (Ed.), *The philosophy of complex systems* (pp. 177-194). Amsterdam, Netherlands: Elsevier.

Hollan, J., Hutchins, E., Kirsh, D. (2000). Distributed cognition: Toward a new foundation for human-computer interaction research. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *7* (2), 174-196.

Huang, O. W., Cheng, H. N., and Chan, T. (2007). Number jigsaw puzzle: A mathematical puzzle game for facilitating players' problem-solving strategies. In *Proceedings of the First IEEE International Workshop on Digital Game and Intelligent Toy Enhanced Learning* (pp. 130-134). Washington, DC: IEEE.

Hunicke, R., LeBlanc, M., and Zubek, R. (2004). MDA: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI* (25-29 July 2004) (pp. 1-5). Palo Alto, CA: AAAI Press.

Hutchins, E. (1995). *Cognition in the Wild*. Cambridge, MA: MIT Press.

Isbister, K., and Schaffer, N. (2008). *Game Usability*. Burlington, MA: Morgan Kaufmann.

Jacobs, J. W., and Dempsey, J. V. (1993). Simulation and gaming: Fidelity, feedback, and motivation. In J. V. Dempsey & G. C. Sales (Eds.), *Interactive instruction and feedback* (pp. 197-227). Englewood Cliffs, NJ: Educational Technology Publications.

Järvinen, A. (2002). Halo and the anatomy of the FPS. *Game Studies*, *2* (1). Retrieved from http://www.gamestudies.org/0102/jarvinen/

Jensen, J. (1998). Interactivity: Tracking a new concept in media and communication studies. *Nordicom Review*, *19* (1), 185-204.

Jonassen, D. H. (2011). Designing for problem solving. In R. Reiser & J. Dempsey (Eds.), *Trends and issues in instructional design and technology* (3rd ed., pp. 64-74). Boston, MA: Pearson.

Kafai, Y. B. (1995). *Minds in play: Computer game design as a context for children's learning*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and Culture*, *1* (1), 36-40.

Kastens, K. A., Liben, L. S., & Agrawal, S. (2008). Epistemic actions in science education. In *Proceedings of the International Conference on Spatial Cognition VI: Learning, Reasoning, and Talking about Space* (pp. 202-215). Freiburg, Germany.

Ke, F., and Grabowski, B. (2007). Gameplaying for maths learning: cooperative or not?. *British Journal of Educational Technology*, *38* (2), 249-259.

Kiousis, S. (2002). Interactivity: A concept explication. *New Media & Society*, *4* (3), 355-383.

Kirsh, D. (1995). The intelligent use of space. *Artificial Intelligence*, *73*(1-2), 31-68.

Kirsh, D. (2005). Metacognition, distributed cognition and visual design. In P. Gardinfas & P. Johansson (Eds.), *Cognition, Education and Communication Technology* (pp. 1-22). Mahwah, NJ: Lawrence Erlbaum.

Kirsh, D., and Maglio, P. (1994). On distinguishing epistemic from pragmatic action. *Cognitive Science*, *18*, 513-549.

Koster, R. (2004). *A theory of fun for game design*. Phoenix, AZ: Paraglyph Press.

Krek, A. (2008). Games in urban planning: The power of a playful public participation. In M Schenk (Ed.), *Proceedings of 13th international conference on urban planning, mobility nodes as innovation hubs, regional development and information society* (pp. 669-683).

Kroehnert, G. (1991). *100 Training games*. New York, NY: McGraw-Hill.

Kücklich, J. (2005). From interactivity to playability: Why computer games are not interactive. In N. Garrelts (Ed.), *Computer gameplay: Essays on the nexus of the game and gamer*. Jefferson, NC: McFarland.

Kuhn, D., and Ho, V. (1980). Self-directed activity and cognitive development. *Journal of Applied Developmental Psychology*, *1* (2), 119-133.

Land, S. M., and Hannafin, M. J. (2000). Student-centered learning environments. In D. H. Jonassen & S. M. Land (Eds.), *Theoretical foundations of learning environments* (pp. 1-23). Mahwah, NJ: Lawrence Erlbaum Associates.

Lakoff, G., and Johnson, M. (1999). *Philosophy in the flesh: The embodied mind and its challenge to western thought*. New York, NY: Basic Books.

Larkin, J., and Simon, H. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, *11*, 65-99.

Lave, J., and Wengner, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. New York, NY: Cambridge University Press.

Law, E. L.-C., and Sun, X. (2012). Evaluating user experience of adaptive digital educational games with Activity Theory. *International Journal of Human-Computer Studies*, *70* (7), 478-497.

Lazar, J., Feng, J. H., and Hochheiser, H. (2010). *Research Methods in Human-Computer Interaction*. West Sussex, UK: Wiley and Sons.

Lee, K. M., and Peng, W. (2006). What do we know about social and psychological effects of computer games? A comprehensive review of the current literature. In P. Vorderer and J. Byrant (Eds.), *Playing Video Games: Motives, Responses, and Consequences* (pp. 383-407). Mahwah, NJ: Lawrence Erlbaum Associates.

Lee, K. M., Park, N., and Jin, S.-A. (2006). Narrative and interactivity in computer games. In P. Vorderer and J. Byrant (Eds.), *Playing Video Games: Motives,* *Responses, and Consequences* (pp. 304-322). Mahwah, NJ: Lawrence Erlbaum Associates.

Lenhart, A., Jones, S., and Macgill, A. (2008). Adults and video games (Pew Internet & American Life Project Report). Retrieved from http://pewinternet.org/Reports/2008/Adults-and-Video-Games.aspx

Lenhart, A., Kahne, J., Middaugh, E., Macgill, A., Evans, C., and Vitak, J. (2008). Teens, video games and civics (Pew Internet & American Life Project Report). Retrieved from http://pewinternet.org/Reports/2008/Teens-Video-Games-andCivics.aspx

Lewis, J. P., McGuire, M., and Fox, P. (2007). Mapping the mental space of game genres. In *Proceedings of the 2007 ACM SIGGRAPH Symposium on Video Games* (pp. 103-108). San Diego, CA: ACM Press.

Liang, H.-N., Parsons, P., Wu, H., and Sedig, K. (2010). An Exploratory Study of Interactivity in Visualization Tools: "Flow" of Interaction. *Journal of Interactive Learning Research*, *21* (1), 5-45.

Lieberman, D. A. (2006). What can we learn from playing interactive games? In P. Vorderer and J. Byrant (Eds.), *Playing Video Games: Motives, Responses, and Consequences* (pp. 304-322). Mahwah, NJ: Lawrence Erlbaum Associates.

Linehan, C., Lawson, S., Doughty, M., Kirman, B., Haferkamp, N., Krämer, N. C., and Schrembi, M. (2011). Teaching group decision making skills to emergency managers via digital games. In A. Lugmayr, H. Franssila, P. Näränen, O. Sotamaa, J. Vanhala, & Z. Yu (Eds.), *Media in the ubiquitous era: Ambient, social and gaming media* (pp. 111-129). New York, NY: IGI Global.

Littlejohn, S. W. (2003). *Theories of human communication* (7th ed.). Beijing, China: Tsinghua University Press.

Liu, Y., and Shrum, L. J. (2009). A Dual-Process Model of Interactivity Effects. *Journal of Advertising*, *38* (2), 53-68.

Lundgren, S., and Björk, S. (2003). Game mechanics: Describing computer-augmented games in terms of interaction. In *Proceedings of Technologies for Interactive Digital Storytelling and Entertainment* (pp. 45-56).

MacKeracher, D. (2004). *Making Sense of Adult Learning* (2nd ed.). Toronto, Canada: University of Toronto Press.

Mann, S. (2002). Conversation as a Basis for Interactivity. In *Proceedings of the 15th Annual Conference of the National Advisory Committee on Computing Qualifications* (pp. 281-288). Hamilton, NZ.

McGuire, M., and Jenkins, O. C. (2009). *Creating games: Mechanics, content and technology*. Natick, MA: AK Peters.

McNaughton, M., Cutumisu, M., Szafron, D., Schaeffer, J., Redford, J., and Parker, D. (2004). ScriptEase: Generative design patterns for computer role-playing games. In *Proceedings of the 19th International Conference on Automated Software Engineering* (88-99). Linz, Austria.

Michalko, M. (2006). *Thinkertoys* (2nd ed.). Berkeley, CA: Ten Speed Press.

Mor, Y., Winters, N., Cerulli, M., and Björk, S. (2006). Literature review on the use of games in mathematical learning, part I: Design. In *Report of the Learning Patterns for the Design and Deployment of Mathematical Games project*.

Moreno-Ger, P., Martínez-Ortiz, I., Luis Sierra, J., and Fernández-Manjón, B. (2008). A content-centric development process model, *Computer*, *41* (3), 24-30.

Moscovich, I. (2000). *Probability games and other activities*. New York, NY: Workman.

Moscovich, I. (2005). *Sensational shape problems & other puzzles*. New York, NY: Sterling.

Moscovich, I. (2006). *The big book of brain games: 1000 Playthinks of art, mathematics & science*. New York, NY: Workman.

Moscovich, I. (2009). *BrainMatics: Logic puzzles*. Cologne, Germany: Tandem Verlag.

Myers, D. (1990). Computer Game Genres. *Play and Culture*, *3*, 286-301.

Nacke, L., Drachen, A., and Göbel, S. (2010). Methods for Evaluating Gameplay Experience in a Serious Gaming Context. *International Journal of Computer Science in* Sport, 9 (2).

Novak, J. (2007). *Game development essentials: An introduction* (2nd ed.). Clifton Park, NY: Delmar Cengage Learning.

Oakes, L. M., and Ribar, R. J. (2005). A comparison of infants' categorization in paired and successive presentation familiarization tasks. *Infancy*, *7*(1), 85-98.

Olivastro, D. (1993). *Ancient puzzles: Classic brainteasers and other timeless mathematical games of the last 10 centuries*. New York, NY: Bantam.

Ormrod, J. E. (1995). *Human Learning* (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.

Owen, A. M., Hampshire, A., Grahn, J. A., Stenton, R., Dajani, S., Burns, A. S., Howard, R., and Ballard, C. G. (2010). Putting Brain Training to the Test. *Nature*, *465*, 775-778. doi:10.1038/nature09042

Parsons, P., and Sedig, K. (2014a). Common visualizations: Their cognitive utility. In W. Huang (Ed.), *Handbook of Human Centric Visualization* (pp. 671-691). New York, NY: Springer New York.

Parsons, P., and Sedig, K. (2014b). Distribution of information processing while performing complex cognitive activities with visualization tools. In W. Huang (Ed.), *Handbook of Human Centric Visualization* (pp. 693-715). New York, NY: Springer New York.

Peterson, A. R., and Snyder, P. J. (1998). Using mind maps to teach social problems analysis. *Paper Presented at the 48th Annual Meeting of the Society for the Study of Social Problems*. San Francisco, CA.

Petković, M. S. (2009). *Famous Puzzles of Great Mathematicians*. Providence, RI: American Mathematical Society.

Prensky, M. (2000). *Digital game-based learning*. New York, NY: McGraw-Hill.

Poels, K., Kort, Y. de, & IJsselsteijn, W. (2007). “It is always a lot of fun!”: Exploring dimensions of digital game experience using focus group methodology. In *Future Play 2007* (pp. 83-89). Toronto, ON.

Powers, K. L., Brooks, P. J., Aldrich, N. J., Palladino, M. A., and Alfieri, L. (2013). Effects of video-game play on information processing: A meta-analytic investigation. *Psychonomic Bulletin & Review*, *20* (6), 1055-1079.

Quiroga, M.Á., Román, F.J., Catalán, A., Rodríguez, H., Ruiz, J., Herranz, M., Gómez-Abad, M., and Colom, R. (2011). Videogame Performance (Not Always) Requires Intelligence. *International Journal of Online Pedagogy and Course Design*, *1* (3), 18-32.

Rambusch, J. (2010). *Mind games extended: Understanding gameplay as situated activity* (Unpublished doctoral dissertation). Linköping, Sweden: Linköping University.

Randel, J. M., Morris, B. A., Wetzel, C. D., and Whitehill, B. V. (1992). The effectiveness of games for educational purposes: A review of the research. *Simulation and Gaming*, *23* (3), 261-276.

Ravaja, N., and Kivikangas, M. (2009). Designing game research: Addressing questions of validity. In U. Ritterfeld, M. J. Cody, & P. Vorderer (Eds.), *Serious games: Mechanisms and effects* (404-410). New York, NY: Routledge.

Renkl, A. (1997). Learning from worked-out examples: A study on individual differences. *Cognitive Science: A Multidisciplinary Journal*, *21* (1), 1-29.

Ritterfeld, U., and Weber, R. (2006). Video games for entertainment and education. In P. Vorderer and J. Byrant (Eds.), *Playing Video Games: Motives, Responses, and Consequences* (pp. 471-490). Mahwah, NJ: Lawrence Erlbaum Associates.

Rollings, A., and Adams, E. (2003). *Andrew Rollings and Ernest Adams on Game Design*. New Riders Games.

Rouse, R. (2005). *Game Design Theory and Practice* (2nd ed.). Plano, Texas: Wordware Publishing.

Rowhani, S., and Sedig, K. (2009). Scaffolded gameplay as a strategy for engaging learners with complexity. In G. Siemens & C. Fulford (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 784-792). Chesapeake, VA: AACE.

Rowley, J. E., and Hartley, R. J. (2008). *Organizing knowledge: An introduction to managing access to information* (4th ed.). Burlington, VT: Ashgate Publishing, Ltd.

Russell, S., and Norvig, P. (2009). *Artificial intelligence: A modern approach* (3rd ed.). Upper Saddle River, NJ: Prentice Hall Press.

Sadtler, C., Aggarwal, A., Cotignola, D., Grewal, S., Hood, P., Shrimali, S., and Teixeira, F. (2006). *Patterns: Implementing self-service in an SOA environment* (2nd ed.). Armonk, NY: IBM Corporation.

Salen, K., and Zimmerman, E. (2004). *Rules of play: Game design fundamentals*. Cambridge, MA: MIT Press.

Salomon, G. (Ed.). (1993). *Distributed cognitions: Psychological and educational considerations*. New York, NY: Cambridge University Press.

Salomon, G., and Perkins, D. (2005). Do technologies make us smarter? Intellectual amplification with, of, and through technology. In R. Sternberg and D. D. Preiss, (Eds.), *Intelligence and technology: The impact of tools on the nature and development of human abilities* (pp. 71-86). Mahwah, NJ: Lawrence Erlbaum.

Salomon, G., Perkins, D., and Globerson, T. (1991). Partners in cognition: Extending human intelligence with intelligent technologies. *Educational Researcher*, *April*, 2-9.

Sánchez, J. L. G., Gutiérrez, F. L., Cabrera, M., and Zea, N. P. (2009). Design of adaptative videogame interfaces: A practical case of use in special education. In V. Lopez Jaquero, F. Montero Simarro, J. P. Molina Masso, & J. Vanderdonckt, J. (Eds.), *Computer-aided design of user interfaces VI* (pp. 71-76). London, UK: Springer-Verlag.

Schneider, W., and Chein, J. S. (2003). Controlled & automatic processing: Behavior, theory, and biological mechanisms. *Cognitive Science*, *27* (3), 525-559.

Sedig, K. (2007). Toward operationalization of "flow" in mathematics learnware. *Computers in Human Behavior*, *23* (4), 2064-2092.

Sedig, K. (2008). From play to thoughtful learning: A design strategy to engage children with mathematical representations. *Journal of Computers in Mathematics and Science Teaching*, *27* (1), 65-101.

Sedig, K. (2009). Interactive mathematical visualizations: Frameworks, tools, and studies. In E. Zudilov-Seinstra, T. Adriaansen, & R. van Liere (Eds.), *Trends in Interactive Visualization: A State-of-the-Art Survey* (pp. 112-132). London, UK: Springer-Verlag.

Sedig, K., and Haworth, R. (2012). Creative design of digital cognitive games: Application of cognitive toys and isomorphism. *Bulletin of Science, Technology, & Society*, *32* (5), 413-426. http://dx.doi.org/10.1177/0270467612469069

Sedig, K., and Haworth, R. (2014). Interaction design and cognitive gameplay: Role of activation time. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play (CHI Play)* (pp. 247-256). New York, NY: ACM. http://dx.doi.org/10.1145/2658537.2658691

Sedig, K., and Liang, H.-N. (2006). Interactivity of visual mathematical representations: Factors affecting learning and cognitive processes. *Journal of Interactive Learning Research*, *17* (2), 179-212.

Sedig, K., and Sumner, M. (2006). Characterizing Interaction with Visual Mathematical Representations. *International Journal of Computers for Mathematical Learning*, *11* (1), 1-55.

Sedig, K., and Parsons, P. (2013). Interaction design for complex cognitive activities with visual representations: A pattern-based approach. *AIS Transactions on Human-Computer Interaction*, *5* (2), 84-133.

Sedig, K., Klawe, M., and Westrom, M. (2001). Role of interface manipulation style and scaffolding on cognition and concept learning in learnware. *ACM Transactions on Computer-Human Interaction*, *8* (1), 34-59.

Sedig, K., Rowhani, S., and Liang, H.-N. (2005). Designing interfaces that support formation of cognitive maps of transitional processes: An empirical study. *Interacting with Computers: The Interdisciplinary Journal of Human-Computer Interaction*, *17* (4), 419-452.

Sedig, K., Parsons, P., Dittmer, M., and Haworth, R. (2014). Human-Centered Interactivity of Visualization Tools: Micro- and Macro-level Considerations. In W. Huang (Ed.), *Handbook of Human Centric Visualization* (pp. 717-743). New York, NY: Springer New York. doi:10.1007/978-1-4614-7485-2

Sharritt, M. (2010). Designing game affordances to promote learning and engagement. *The International Journal of Cognitive Technology*, *14*(2)-*15*(1), 58-65.

Sicart, M. (2008). Defining game mechanics. *Game Studies*, *8* (2). Retrieved from http://gamestudies.org/0802/articles/sicart

Siwek, S. E. (2010). *Video games in the 21st century: The 2010 report*. The Entertainment Software Association. Retrieved from http://www.theesa.com/facts/pdfs/VideoGames21stCentury_2010.pdf

Skyttner, L. (2005). *General systems theory: Problems, perspectives, practice*. Hackensack, NJ: World Scientific.

Slocum, J. (2007). *The Tao of Tangram: History, problems, solutions*. New York, NY: Barnes & Noble.

Smith, D. (2005, May 15). So you thought Sudoku came from the Land of the Rising Sun. *The Observer*. Retrieved from http://www.guardian.co.uk/media/2005/may/15/pressandpublishing.usnews

Smith, E. E., and Medin, D. L. (1981). *Categories and Concepts*. Cambridge, MA: Harvard University Press.

Smith, S. M., and Vela, E. (2001). Environmental context-dependent memory: A review and meta-analysis. *Psychonomic bulletin & review*, *8* (2), 203-220.

Sniedovich, M. (2002). OR/MS Games: 2. The Towers of Hanoi Problem. *INFORMS Transactions on Education*, *3* (1), 34-51.

Spires, H. A. (2008). 21st century skills and serious games: Preparing the N generation. In L. A. Annetta (Ed.), *Serious Educational Games: From Theory to Practice* (pp. 13-23). Rotterdam, The Netherlands: Sense Publishers.

Squire, K. (2006). From content to context: Videogames as designed experience. *Educational Researcher*, *35* (8), 19-29.

Stolte, C., Tang, D., and Hanrahan, P. (2002). Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, *8* (1), 52-65.

Suits, B. (1990). *Grasshopper: Games, Life, and Utopia*. Boston, MA: David R. Godine.

Svanaes, D. (2000). Understanding interactivity - Ateps to a phenomenology of human-computer interaction (PhD Dissertation). Trondheim, Norway: Norwegian University of Science and Technology.

Svendsen, G. (1991). The influence of interface style on problem solving. *International Journal of Man-Machine Studies*, *35* (3), 379-397.

Sweetser, P. (2008). *Emergence in games*. Boston, MA: Charles River Media.

Sweetser, P., and Wyeth, P. (2005). GameFlow: A model for evaluating player enjoyment in games. *Computers in Entertainment*, *3* (3), 1-24.

Sweller, J. (2010). Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational Psychology Review*, *22* (2), 123-138.

Tufte, E. R. (1997). *Visual Explanations: Images and Quantities, Evidence and Narrative* (3rd ed.). Cheshire, CT: Graphics Press.

Turner, J., and Browning, D. (2010). Workshop on HCI and game interfaces: A long romance. In *OZCHI 2010: Design – Interaction – Participation*. Brisbane, Australia: Queensland University of Technology.

Van Eck, R. (2006). Digital game-based learning: It's not just the digital natives who are restless. *Educause Review*, *41* (2), 16-30.

Whitchurch, G. G., and Constantine, L. L. (1993). Systems theory. In P. G. Boss, W. J. Doherty, R. LaRossa, W. R. Schumm, & S. K. Steinmetz (Eds.), *Sourcebook of family theories and methods: A contextual approach* (pp. 325-355). New York, NY: Plenum Press.

Wilson, K. A., Bedwell, W. L., Lazzara, E. H., Salas, E., Burke, C. S., Estock, J. L., et al. (2009). Relationships between game attributes and learning outcomes: Review and research proposals. *Simulation & Gaming*, *40* (2), 217-266.

Wolf, M. J. P. (2001). Genre and the video game. *The Medium of the Video Game* (pp. 113-134). Austin, TX: University of Texas Press.

Yang, J., Peng, W., Ward, M. O., and Rundensteiner, E. A. (2003). Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *Proceedings of the IEEE Symposium on Information Visualization* (105-112). Seattle, WA.

Yi, J. S., ah Kang, Y., Stasko, J. T., and Jacko, J. A. (2007). Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, *13* (6), 1224-1231.

Yoo, W. S., Lee, Y., and Park, J. (2010). The role of interactivity in e-tailing: Creating value and increasing satisfaction. *Journal of Retailing and Consumer Services*, *17* (2), 89-96.

Zagal, J. P., Mateas, M., Fernández-Vara, C., Hochhalter, B., and Lichti, N. (2005). Towards an ontological language for game analysis. In *Proceedings of the 2nd Digital Games Research Association*, (pp. 3-14). Vancouver, Canada.

Zhang, J., and Norman, D. (1994). Representations in distributed cognitive tasks. *Cognitive Science: A Multidisciplinary Journal*, *18* (1), 87-122.

Curriculum Vitae

Name: Robert Haworth

Post-secondary Education and Degrees:

The University of Western Ontario
London, Ontario, Canada
2009-2015 Ph.D.

The University of Western Ontario
London, Ontario, Canada
2003-2008 B.Sc. Honours

Honours and Awards:

Province of Ontario Graduate Scholarship
2009 – 2010, 2011 – 2014

Nominated Best Paper Award at CHI Play
2014

Best Presentation in HCI / Game Design at UWORCS
2012, 2013

UWO Gold Medal in Computer Science
2008

Edna Jeffery Scholarship in Computer Science
2005

Nortel Networks Scholarship of Distinction in Computer Science
2003 – 2006, 2007 – 2008

Related Work Experience:

<u>Lecturer</u>
The University of Western Ontario
2012, 2015 (Human-Computer Interaction)

<u>Teaching Assistant</u>
The University of Western Ontario
2009 – 2013
Courses:

- The Internet: Behind the Curtain [Spring, 2013]
- Introduction to Medical Computing [Fall, 2012]
- Computer Graphics I [Fall, 2011]
- Human-Computer Interaction [Spring, 2011]
- Game Engine Design [Fall, 2010]

Design of Thinking:
Games as Tools for the Mind [Spring, 2010]
Computer Science Fundamentals I [Fall, 2009]
Design of Thinking:
Games as Tools for the Mind [Spring, 2009]

Professional Activities:

Reviewer
CHI Play 2014, ACM (1 paper), 2014
Cases on E-Learning Management, IGI (3 chapters), 2012

Research in Computer Science Conference Session Judge
The University of Western Ontario
2011, 2013, 2014

Computer Science Graduate Student Council VP Communications
The University of Western Ontario
2012-2013

Computer Science Graduate Student Council VP Finance
The University of Western Ontario
2010-2012

ED-Media International Conference Volunteer
2010

Refereed Journal Publications:

Sedig, K., Haworth, R., & Parsons, P. (submitted). Toward a Science of Interactivity for Cognitive Gameplay: Design of the Core Mechanic. *The International Journal of Cognitive Technology*.

Sedig, K., Haworth, R., & Corridore, M. (2015). Exploring Variations in Gameplay: Cognitive Implications. *International Journal of Computer Games Technology*, *2015*, 1-16. doi:10.1155/2015/208247

Haworth, R., Parsons, P., & Sedig, K. (2013). Design of digital cognitive games: Some considerations. *The International Journal of Cognitive Technology*, *18* (1), 22-27.

Sedig, K., & Haworth, R. (2012). Creative Design of Digital Games: Isomorphism and Cognitive Toys. *Bulletin of Science, Technology & Society*, *32* (5), 413-426. doi:10.1177/0270467612469069

Haworth, R., Tagh Bostani, S.S., & Sedig, K. (2010). Visualizing Decision Trees in Games to Support Children's Analytic Reasoning: Any Negative Effects on Gameplay?. *International Journal of Computer Games Technology*, *2010*, 1-11. doi:10.1155/2010/578784

Refereed Book Chapters:

Sedig, K., Parsons, P., Dittmer, M., & Haworth, R. (2014). Human-Centered Interactivity of Visualization Tools: Micro- and Macro-level Considerations. In W. Huang (Ed.), *Handbook of Human Centric Visualization* (pp. 717–743). New York, NY: Springer New York. doi:10.1007/978-1-4614-7485-2

Haworth, R., & Sedig, K. (2013). Designing Digital Cognitive Games that Facilitate Mindful Reasoning and Decision Making. In Yang, H., & Wang, S. (Eds.), *Cases on E-Learning Management: Development and Implementation* (196-222). Hershey, PA: IGI Global.

Haworth, R., & Sedig, K. (2011). The Importance of Design for Educational Games. In Méndez-Vilas, A. (Ed.), *Education in a Technological World: Communicating Current and Emerging Research and Technological Efforts*. (518-522). Badajoz, Spain: Formatex Research Center.

Haworth, R., & Sedig, K. (2010). A Strategy for Analyzing Digital Epistemic Games. In Soria, A., Maldonado, J. (Eds), *Computer Games: Learning Objectives, Cognitive Performance, and Effects on Development* (pp. 53-84). New York, NY: Nova Science Publishing.

Refereed Conference Publications:

Sedig, K., & Haworth, R. (2014). Interaction design and cognitive gameplay: Role of activation time. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play (CHI Play)* (pp. 247-256). New York, NY: ACM. doi:10.1145/2658537.2658691

Haworth, R., & Sedig, K. (2010). Supporting Children's Analytic Reasoning in Digital Games. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2010* (pp. 758-765). Chesapeake, VA: AACE.

Sedig, K., & Haworth, R. (2010). Using Cognitive Toys to Build Digital Games. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2010* (pp. 711-720). Chesapeake, VA: AACE.

Presentations:

<u>An Investigation of Cognitive Implications in the Design of Computer Games</u>
PhD Thesis Presentation, *Oral Presentation*
Western University, London Canada, 6 February 2015

<u>Interaction Design and Cognitive Gameplay: Role of Activation Time</u>
International Conference, *Oral Presentation*
First ACM SIGCHI annual symposium on Computer-human interaction in play (CHI Play 2014)
Toronto Canada, 19 October to 22 October 2014 [presented on 22 October]

<u>Understanding the Design of Digital Cognitive Games</u>
PhD Thesis Proposal, *Oral Presentation*
Western University, London Canada, 28 February 2014

<u>The Anatomy of Digital Cognitive Games</u>
Institutional Conference, *Oral Presentation*
University of Western Ontario Research in Computer Science (UWORCS) 2013
Western University, London Canada, 12 April 2013

<u>Externalization in Digital Cognitive Games</u>
Institutional Conference, *Oral Presentation*
University of Western Ontario Research in Computer Science (UWORCS) 2012
Western University, London Canada, 13 April 2012

<u>Supporting Children's Analytic Reasoning in Digital Games</u>
International Conference, *Oral Presentation*
World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA) 2010
Toronto Canada, 29 June to 2 July 2010

<u>Using Cognitive Toys to Build Digital Games</u>
International Conference, *Oral Presentation*
World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA) 2010
Toronto Canada, 29 June to 2 July 2010

<u>The Design of Digital Cognitive Games</u>
Institutional Conference, *Oral Presentation*
University of Western Ontario Research in Computer Science (UWORCS) 2010
Western University, April 2010