2010

Aligning Problem Solving and Gameplay : A Model for Future Research and Design

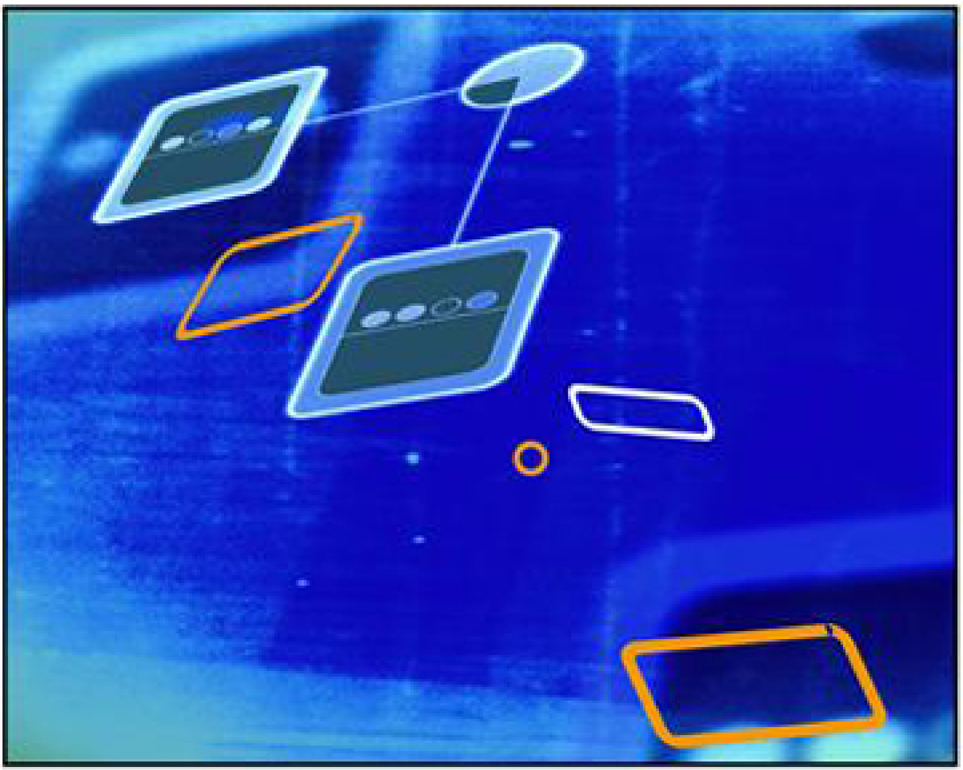
Woei Hung
University of North Dakota, woei.hung@und.edu

Richard Van Eck
University of North Dakota, richard.vaneck@und.edu

PREMIER REFERENCE SOURCE

Interdisciplinary Models and Tools for Serious Games

Emerging Concepts and Future Directions



RICHARD VAN ECK

Interdisciplinary Models and Tools for Serious Games:
Emerging Concepts and Future Directions

Richard Van Eck
University of North Dakota, USA

INFORMATION SCIENCE REFERENCE
Hershey · New York

Director of Editorial Content: Kristin Klinger
Director of Book Publications: Julia Mosemann
Acquisitions Editor: Lindsay Johnston
Development Editor: Elizabeth Ardner
Publishing Assistant: Sean Woznicki
Typesetter: Devvin Earnest
Production Editor: Jamie Snavely
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com/reference

Library of Congress Cataloging-in-Publication Data

Interdisciplinary models and tools for serious games : emerging concepts and future directions / Richard Van Eck, editor.
p. cm.

Includes bibliographical references and index.
Summary: "This book discusses the need for interdisciplinary awareness in the study of games and learning"--Provided by publisher.

ISBN 978-1-61520-719-0 (hardcover) -- ISBN 978-1-61520-720-6 (ebook) 1. Educational games. I. Van Eck, Richard, 1964-
LB1029.G3.I68 2010
371.39'7--dc22

2009044470

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Editorial Advisory Board

List of Reviewers

Copy Editor

Table of Contents

Section 5
Future Directions

Chapter 10
Aligning Problem Solving and Gameplay:
A Model for Future Research and Design

Woei Hung
University of North Dakota, USA

Richard Van Eck
University of North Dakota, USA

ABSTRACT

Problem solving is often discussed as one of the benefits of games and game-based learning (e.g., Gee, 2007a, Van Eck 2006a), yet little empirical research exists to support this assertion. It will be critical to establish and validate models of problem solving in games (Van Eck, 2007), but this will be difficult if not impossible without a better understanding of problem solving than currently exists in the field of serious games. While games can be used to teach a variety of content across multiple domains (Van Eck, 2006b, 2008), the ability of games to promote problem solving may be more important to the field of serious games because problem-solving skills cross all domains and are among the most difficult learning outcomes to achieve. This may be particularly important in science, technology, engineering, and math (STEM), which is why serious game researchers are building games to promote problem solving in science (e.g., Gaydos & Squire, this volume; Van Eck, Hung, Bowman, & Love, 2009). Current research and design theory in serious games are insufficient to explain the relationship between problem solving and games, nor do they support the design of educational games intended to promote problem solving. Problem solving and problem-based learning (PBL) have been studied intensely in both Europe and the United States for more than 75 years. Most recently, researchers (e.g., Jonassen, 1997, 2000, & 2002; Hung, 2006a; Jonassen & Hung, 2008) have made advances in both the delineation and definition of problem types and models for designing effective problems and PBL. Any models and research on the relation of games and problem solving must build on the existing research base in problem solving and PBL rather than unwittingly covering old ground in these areas. In this chapter, we present an overview of the dimensions upon which different problems vary, including domain knowledge and structuredness and their associated learning outcomes. We then propose a classification of gameplay (as opposed to game genre) that accounts for the cognitive skills encountered during gameplay, relying in part on pre-

DOI: 10.4018/978-1-61520-719-0.ch010

vious classifications systems (e.g., Apperley, 2006), Mark Wolf's (2006) concept of grids of interactivity (which we call iGrids), and our own cognitive analysis of gameplay. We then use this classification system, the iGrids, and example games to describe eleven different types of problems, the ways in which they differ, and the gameplay types most likely to support them. We conclude with a description of the ability of problems and games themselves to address specific learning outcomes independent of problem solving, including domain-specific learning, higher-order thinking, psychomotor skills, and attitude change. Implications for future research are also described. We believe that this approach can guide the design of games intended to promote problem solving and points the way toward future research in problem solving and games.

INTRODUCTION

It has been argued that games are a kind of disruptive technology (e.g., Strawn, 2007), but they can only be so to the extent that they solve a widely recognized problem that has value to sufficient numbers of people. For game-based learning to truly become a disruptive technology, it must address a critical need that is difficult to meet any other way. Many have argued that games address critical thinking and problem-solving skills (e.g, Gee, 2007a; Greenfield, in press; Van Eck, 2006a, 2007; Yanuzzi & Behrenhausen, this volume) that our current educational system is failing to provide (e.g., Broussard, La Lopa, & Ross-Davis, 2007; OECD, 2004).

Problem solving may well be the most powerful pedagogical benefit of commercial games in general and of game-based learning and serious games specifically. Whether our current educational system recognizes the need for problem solving as a learning outcome, and whether or not it can support it with existing resources and infrastructure, it seems clear that problem solving and the related research and design we do will remain an important area of study in the field of serious games.

Unfortunately, while researchers have begun to move the discussion of problem solving beyond descriptive to theoretical (e.g., Gee, 2007a & 2007b; Van Eck, 2007) and the practical (Van Eck, 2008), the majority of our discussion can be summed up as "Games are problems being solved by players; therefore, playing games will help people be better problem solvers." Our research tends to be primarily descriptive, wherein we describe the admittedly complex behavior involved in working one's way through a game like *World of Warcraft* (Blizzard, 2001) as evidence that problem solving must surely be going on during that process. This is sufficient for making the case that games most likely address problem solving and are therefore worthy of further study, but this is not sufficient to guide our development of serious games to directly address problem solving as a learning outcome. Problem solving is far more complex than many first realize, just as games are more complex than they appear at first to the general public. For example, we cannot discuss problem solving without understanding what type of problem we are referring to: creating a menu for guests who have different diet restrictions, troubleshooting a car that won't start, diagnosing a patient's back pain problem, or solving global warming. Each type of problem differs significantly in structuredness, requirements for prior knowledge, ability to embed other subproblems, cognitive structure, etc. Just as we recognize that game genres (e.g., first-person shooter, adventure, role-playing games [RPGs], massively multiplayer online games [MMOGs]) encourage different gameplay experiences, we need to recognize the different types of problem solving that exist in the world.

It is almost universally agreed in the field of instructional design (Dick, Carey, & Carey, 2005; Gagne, Wager, Golas, & Keller, 2005; Smith & Ragan, 2005) that instruction is most effective when the instructional strategies (in this case, styles of gameplay) that are employed appropriately afford the learners' development of the desired learning outcomes (in this case, types of problem solving). If we don't understand the full typology and complexity of different problem types, we cannot begin to formulate theory or practice in serious games and problem solving.

Fortunately, we are not starting from scratch in this regard. Cognitive psychology and instructional design have been studying problem solving for many years, and a rich body of research exists that can help inform our studies and design of problem solving in games. In this chapter, we will discuss problem-solving theory and research from a cognitive perspective. We begin with a brief discussion of problems and problem solving in general, then move into a discussion of the nature of different types of problems (problem typology). We will attempt to bridge theory and practice by examining the relationships between games, problems, their cognitive processes, and instructional design, including heuristic tools and examples of those problem types as they may be mapped onto different gameplay experiences typically afforded by different genres of games.

BACKGROUND

The Problem with Problem Solving

Research on problem solving goes back at least to the 1930s and Gestalt psychology.[1] Early attempts to study problem solving were hampered by assumptions that most researchers have now come to believe are flawed. Chief among these assumptions was that all problem solving was essentially the same for all individuals and, most critically, for all kinds of problems and domains.[2] Research in this tradition was focused on controlling for prior knowledge through the use of simple, novel tasks, the assumption being that this would uncover general problem-solving skills shared by all problems and all problem solvers. The tasks used to observe problem solving all had prescribed "best" solutions and were thus easy to use to compare problem solvers' moves to optimal solutions. The *Towers of Hanoi* (2002) is a well-known example. A mathematical game developed in the 1800s, this game required one to shift a pyramid-shaped stack of disks from one of three posts to another. All three posts are identical and aligned in a single row, with one of the outside (i.e., leftmost or rightmost) posts initially containing the stack of disks (often referred to as the initial state). The rules require the player to move only one ring at a time, either one or two posts distance, and to only place smaller rings on top of larger rings, until the pyramid stack of disks has been replicated on the third post (often referred to as the goal state). The solution to this task requires that the player make short-term moves which appear incorrect in order to achieve the final goal. Problems like this were thought to scale up, or generalize, to other problems in other domains.

It was not until nearly 50 years later that researchers (e.g., Bhaskar & Simon, 1977) came to believe that a general theory of problem solving was not possible, and that problem solving was very much context- and domain-dependent and that, further, problems themselves were all different according to the domain in which they were situated. For the last 30 years, problem solving has proceeded under these assumptions, although by differing methods and approaches in Europe and the United States.

Nevertheless, some elements of problem-solving research in the last century remain useful for talking about problem solving, despite their origins in abstract, well-defined problems. Early research suggested that a problem has two states: an initial state and a goal state. The initial state is the set of information and resources present at the beginning of the problem. This is the starting point for the problem, if you will. The goal state is the information and resources that will be present when the goal has been met. It is the ending point of the problem, after it has been solved. A problem, then, can be thought of as an attempt by the problem solver to do things that reduce the disparity between the initial state and the goal state. The strategies she uses and the process by which she thinks about moving toward the goal state within the constraints of the problem and prior knowledge are collectively referred to as the problem space. Problem-solving research describes the solution process with a variety of terms but most commonly as searching the problem space (Newell & Simon, 1972). We can see this in games as well, where games have an initial state and goal state (introduced by the game box, cut scenes, Web site reviews, and word-of-mouth among players), and where the playing of the game becomes the problem space.

While our conceptualization of problem solving today is domain dependent and recognizes a variety of ill-structured vs. well-structured problems (more on this later), these concepts can still be helpful in discussing problem solving. For example, while we recognize that the initial state and the goal state are complex concepts defined by the learner herself as she integrates what is known about the problem with her existing schema for the domain/problem, they remain useful labels for discussing those aspects of the problem-solving process.

Most recently, Jonassen (e.g., 2000, 2002) and Jonassen and Hung (2006, 2008) have proposed a typology of problems and associated prescriptions for the design of problem-based learning and instruction to promote problem solving in general. Given the widely held belief that games themselves are examples of problem solving, the potential for this body of research to inform research and design in serious games warrants a closer inspection of this literature to see if and how it can be mapped to the study and design of serious games.

The Heart of the Matter

The problem with problems in instructional contexts is that many are poorly formed and articulated, thus dooming from the beginning any instruction designed to promote problem solving. According to Jonassen (2002), all good problems share two characteristics. First, they have some kind of goal, which he refers to as the "unknown." By unknown, he means that the learner does not know how to reach the goal, not that the learner does not know what s/he is trying to achieve in the first place. Consequently, the goal requires the generation of new knowledge, which can be a combination of two or more elements of prior knowledge and/or the generation and combination of new knowledge with prior knowledge. The second characteristic all good problems share is a value to the learner in solving them (i.e., in generating the knowledge needed to achieve the unknown). It is not much of a stretch to see how this potentially aligns well with games and problem-solving environments; games have an overarching goal that the learner does not know how to achieve and which requires the generation of new knowledge (the unknown), and games (at least, good ones) have a value to the learner in achieving the goal (unknown).

Games as Problem-Solving Environments

Jim Gee (2007a, 2007b) has argued convincingly that all games are situated, complex problem-solving opportunities in which players are immersed in a culture and way of thinking. We've already discussed how games can be conceptualized in similar ways to problem solving from the cognitive sciences (e.g., initial state, problem space, and goal state; goal/unknown, generation of new knowledge, and of value to the learner in solving the problem). Others have made the same point, such as Kiili (2007), who contended that "a game itself is a big problem that is composed of smaller causally linked problems" (p. 396).

To be sure, games are more than just problems to be solved and will often contain not only multiple kinds of problems of varying type, structuredness, and complexity but also a variety of other learning and entertainment outcomes with their associated strategies. Nonetheless, it is difficult if not impossible to conceptualize a game that does not incorporate problems to be solved, and thus, problems can be seen as the raw materials for producing games, which can themselves be thought of as problem-solving domains.

The realization that problems are at the heart of games not only opens a new avenue for game research but also leads us to a wealth of previous research findings to draw upon. Problem solving is a long-studied cognition research area (see, for example, Frensch & Funke, 1995a; Greeno, 1980; Hayes, 1980; Jonassen, 1997; Larkin & Simon, 1987; Newell & Simon, 1972), and it behooves us as researchers in this growing field to be aware of this research as we attempt to refine our understanding about the cognitive benefits of games.

PROBLEMS, INTERACTIVITY, AND GAMES

In the next four sections, we will describe different kinds of problems and the cognitive processes they require, the learned capability outcomes they support, and their connection to gameplay styles. While the largest part of our argument will be that problems are highly differentiated by context, purpose, and domain, it is not possible to discuss this without also discussing some of the aspects by which this differentiation occurs. Therefore, in the first section we will look at structuredness, cognitive components, and domain knowledge as key dimensions along which problems vary.

Further, in order to describe how different games may or may not align with different problems, we must first establish a common set of terminology and definitions for what we mean by different types of games. While problem typologies and classification systems are well-established and accepted in the learning sciences, the same cannot be said for game classifications. Traditionally, the field has relied on genres of games to organize discussions of different types of gameplay. However, this approach has led to several challenges. There are competing genre classification systems (e.g., Apperley, 2006; Bogost, 2007), for example, that are valuable but not necessarily compatible nor widely accepted and adopted. Also, games often employ multiple gameplay strategies from different genres within the same game, leading to hybridized descriptions like action–adventure that work against meaningful classification (Kallay, this volume). This creates a significant challenge for our purpose in this chapter: how to describe what kinds of games support what kinds of problems and vice versa?

Because we believe that problem types and their associated cognitive-processing requirements will be most impacted by *gameplay* rather than game *genre* and that interactivity captures the most salient features of gameplay as it relates to problem solving, we have adopted Mark Wolf's (2006) concept of a grid of interactivity (which we refer to as an iGrid) to help quantify the interaction required by the different gameplay types. In the second section, we will describe this metric, which we believe is a better way of discussing the interaction of problem type and games than traditional genre classifications would be. Such an approach avoids the first challenge above (incompatible genre classifications that confound gameplay, platform, and marketing terminology) and solves the related second challenge (hybridized categories resulting from multiple gameplay styles within a single game).

Problems with genre classifications notwithstanding, we also recognize their value in shared understanding and familiarity. Therefore, in the third section, we have attempted to synthesize existing genre classifications with the express purpose of mapping them to the cognitive processes required by the game in a manner that takes advantage of common terminology in game studies. This is necessary in order to further map problem types, which themselves differentially support and require cognitive processes. We will rely on iGrids to help make the appropriate delineations and comparisons.

In the fourth section, we will describe the eleven different types of problems (Jonassen, 2000) and relate them to gameplay types, relying again on both iGrids and on our own gameplay classifications before closing with a discussion of the learned capability outcomes (psychomotor, attitude, etc.) in the context of games and problem types. Again, our purpose is not to perpetuate problematic genre classification systems nor to propose new ones but rather to map problem solving and problem types to appropriate kinds of gameplay design.

Problems

Structuredness

A broad definition of problem structuredness was articulated by Wood (1983) as the degree to which the information in the problem is known or knowable to the problem solver. Jonassen (1997) further refined this concept in his discussion of the continuum of well-structured and ill-structured problems, where he argued that structuredness describes the reliability of the problem space in terms of the ratio of the information about the problem known and unknown to the problem solver, the number of variables involved, number of solution paths, and the degree of ambiguity about the criteria for assessing the success of solving the problem. More specifically, he states that the factors that characterize the structuredness include known versus vaguely defined or unknown states of the problem (initial state, goal state, and operators), regular versus unconventional uses of rules and principles involved, stated constraints versus hidden constraints, predictable operators versus highly unpredictable and unprescribed operators, a preferred and prescribed solution versus multiple viable solutions, and definite versus vague criteria for evaluating the solutions.

Video games may run the gamut from highly structured (as with the need to fire weapons against hordes of zombies in *Left 4 Dead* [Valve, 2008] in order to stay alive) to poorly structured (how to win *Spore* [Entertainment Arts, 2008]). Therefore, structuredness becomes one dimension upon which we can categorize both games and problems.

Cognitive Composition of Problems

In addition to varying along the dimension of structure, solving different problems also relies on different kinds of cognition. Building a civilization over the span of 3000 years via multiple strategies such as economics, diplomacy, industry, and arts (e.g., the *Civilization* series of games) is a fundamentally different problem than trying to get from safe house to safe house without being killed by "hunters," "smokers," "boomers," "tanks," or swarming hordes of zombies (*Left 4 Dead*, Valve, 2008). Therefore, it is not logical to expect that strategies learned in one game will necessarily transfer to another. Likewise, problems in general require one or more kinds of thinking, some of which are supportive of one another and some of which are completely different. We examine here six main kinds of cognitive processes from cognitive psychology and instructional design that we will later rely on (along with structuredness and other dimensions) in our discussion of problem solving.

Logical Thinking

This cognitive process refers to the mental process that infers an expected event as a result of the occurrence of its preceding event or evaluates the validity of the conditional relations of these events. Most people are not particularly conscious of engaging in this type of thinking process; yet, in fact, it is a fundamental cognitive process that humans utilize to process and reason everyday matters (Houdé & Tzourio-Mazoyer, 2003). For example, if I see John walk into the room with a wet, dripping umbrella, I might infer that it was raining. Likewise, a jury member might notice a logical flaw in testimonial statements indicating that a suspect was seen in two locations 300 miles apart within a 1-hour period of time.

Analytic Thinking

The analytic cognitive process mainly focuses on identifying and separating an object, essay, substance, or system into its constituent components, examining their relationships as well as understanding the nature, behaviors, and specific functions of each component. This cognitive process is essential in developing a deep understanding about a subject area, a system, or a problem. An individual needs to be able to isolate individual parts in order to understand their unique nature and functions in relation to the whole. Therefore, analytic thinking can be seen as the initial cognitive process that an individual has to perform in understanding what is being studied. Analyzing bank operations in order to develop banking system software is an example of employing analytic thinking.

Strategic Thinking

Mintzberg (1994) argued that strategic thinking is an integration process of synthesizing and evaluating the analytical results of a given situation and generating a most viable plan with intuition and creativity. Liedtka (1998) further characterized strategic thinking as the cognitive processes that are intent-focused, hypothesis-driven, thinking in time, intelligent opportunism, and reasoning from a systems perspective. Thus, strategic thinking involves a goal-oriented planning process with an understanding of past and current situations, the generation and testing of hypotheses, flexible adaptation to the dynamic nature of the environment, and the taking of a systemic view during the entire thinking process. The ability to think strategically is a key to effective problem solving. Managing a multinational enterprise in a cutting-edge technology business is an example that requires effective and intensive strategic thinking skills.

Analogical Reasoning

According to Holyoak and Thagard (1997), analogical reasoning refers to the mental process in which an individual "reason[s] and learn[s] about a new situation (the *target* analog) by relating it to a more familiar situation (the *source* analog) that can be viewed as structurally parallel" (p. 35). For example, when the concept of the Internet first became known to the public, the analogy of a highway system and traffic was used to help people understand its structure and function. Moreover, analogical reasoning is the core of case-based learning (Kolodner, 1997), which is a common learning strategy employed by people in their everyday lives as well as an effective instructional approach.

Systems Thinking

Systemic thinking refers to the cognitive reasoning processes that consider complex, dynamic, contextual, and interdependent relationships among constituent parts, and the emerging properties of a system (Capra, 1996; Ossimitz, 2000). According to Sterman (2002), this cognitive skill is considered very complex and highly counterintuitive. Because of its complex nature, the cognitive load of performing systems thinking is usually beyond an individual's capacity. Therefore, cognitive tools such as modeling software are often required. One example that requires intensive systems thinking processes is constructing a weather or ecological system model.

Metacognitive Thinking

Metacognition refers to the cognitive process that an individual is consciously aware of and which he or she articulates to various aspects of his or her own thinking processes. In simpler terms, it is "cognition about cognition" (Flavell, 1985, p. 104). Metacognition and its cognitive processes and skills are the core elements for successful self-regulated, self-directed learning (Driscoll, 2005). Metacognitive thinking is a highly complex cognitive process that involves all of the cognitive thinking skills mentioned above at some point and at different levels during the thinking process.

Domain Knowledge

In addition to structuredness and cognitive composition, problems will vary by the domain knowledge they require. One cannot solve a problem if one has not mastered the prerequisite domain knowledge. We cannot expect a student to solve an algebraic equation without having mastered subordinate skills such as addition, multiplication, division, and subtraction. What has been a persistent challenge for serious games is that problem solving that is dependent on domain knowledge has sometimes resulted in edutainment that requires mastery of content delivered in a highly instructivist manner without regard to the ludic nature of video games. The focus of this chapter lies in problem solving, however, and a discussion of how to design games to incorporate the need for mastery of domain content is beyond the scope of our discussion here. We argue here only that if the designer's goal is to promote problem solving and that problem requires prerequisite knowledge, one must include prerequisite knowledge as a design goal or the problem must be reconceptualized such that it does not require that prior domain knowledge.

Domain knowledge required for problem solving can include declarative knowledge, procedural knowledge, concepts, and principles. Declarative knowledge refers to things that can be stated, also often called verbal information, and includes labels, names, and facts. Procedural knowledge is knowledge of how to conduct a process, whether it is the order of mathematical operations in solving an algebra problem or knowing how to send an e-mail. Concepts are little more slippery, both because everyone thinks they know what a concept is and because concepts can be concrete or abstract. An abstract concept (also called a defined concept) is something that cannot be pointed to but must instead be evaluated according to criteria or a definition. Patriotism is an example, in that whether something is or is not patriotic depends on its relation to a nation's laws, values, and social considerations. In contrast, concrete concepts are things that can be identified and agreed on by virtue of their nature. A ball or a chair is an example of a concrete concept, despite the fact that it can vary tremendously in appearance and surface characteristics. On the other hand, principles are defined by Sugrue (1995, p. 29) as "the rules that involve relationships among the concepts." PBL problems usually involve several concepts. The learners must conceptually interconnect the concepts based on the principles in order to apply the concepts to solve a complex problem.

It is not necessary to be an expert in applying this terminology so much as it is critical that each type of knowledge be explicitly examined during the problem design stage to ensure that all domain-specific prerequisite knowledge be identified and classified so it can be pretested and because each requires different strategies for mastery should it be determined that the strategies must be addressed within the game. It is important to note also that this is all part of the initial instructional analysis and design stages, and does not imply any kind of instructivist approach will become part of the gameplay itself!

iGrids and Gameplay Types

While serious game researchers may not agree on game genre classifications, most would agree that interactivity is one of the hallmarks of video games (Wolf, 2006). As such, interactivity is a good place to start in our attempts to classify games in ways that do not suffer from the problems current genre classifications do. Wolf argues that:

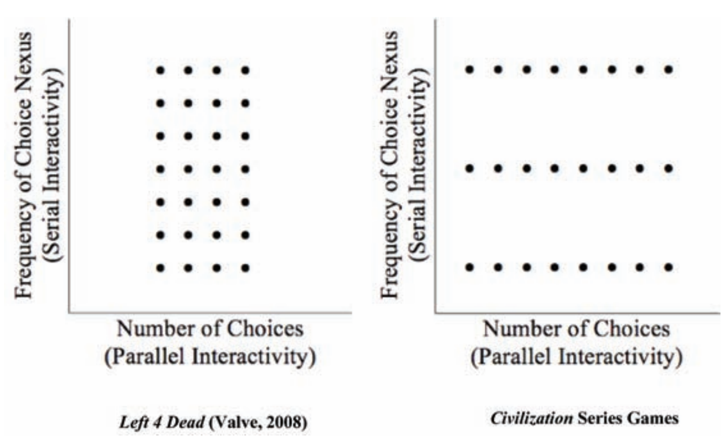
The smallest unit of interactivity is the choice.... Choices are made in time, which gives us a two-dimensional grid of interactivity that can be drawn for any game. First, in the horizontal direction, we have the number of simultaneous (parallel) options that constitute the choice that a player is confronted with at any given moment. Second, in the vertical direction, we have the number of sequential (serial) choices made by a player over time until the end of the game." (p. 80)

Wolf (2006) calls this a Grid of Interactivity, but for semantic reasons, we will refer to it as an Interactivity Grid, or iGrid. Because the frequency of choices and the number of choices make good initial measures of both pace and complexity or cognitive load, and because we believe (and evidence supports) that these constructs are likely to impact problem solving in general and problem typology differentially, iGrids make a good place to start this discussion. Of course, Wolf himself points out that it is not possible to map an entire game space on a graph. For example, in addition to the number of choices at a given time (x axis) and the frequency of opportunities for choice (or choice nexus[3]), he argues that we should further evaluate the consequences of individual choices (from trivial to game-changing). We would further argue that the amount of complexity of the information required at each nexus would be of further value in this analysis. Nonetheless, such plots remain a useful tool for conceptualizing the issue of interactivity and one which we can rely on to further define the kinds of gameplay that differentially support different problem typology.

To do this, we can imagine Aristotelian archetypes of different game genres. For example, in our descriptions of action games and simulation games, *Left 4 Dead* (Valve, 2008) and the *Civilization* series games, we might conceptualize an iGrid as seen in Figure 1.

The x-axis represents parallel interactivity, which is the number of choice options a player has at a given point in time (called a choice nexus), while the y-axis represents how often the player is presented with a choice nexus. For example, the game represented by the iGrid on the left of Figure 1 forces the player to make choices frequently over the course of the game with little time between choices but presents few options to choose from at those points. In the iGrid on the right, we see a game that presents many options to choose from but forces the player to make choices fewer times

Figure 1. iGrids for two different gameplay types



over the course of the game with long periods of time between choices.

Left 4 Dead (Valve, 2008) is a game in which players must fight their way across a city filled with zombies trying to kill them. While there are ostensibly many choices to make during gameplay, (which path to take, how long to wait between "runs," which of five or six weapons to use, or where to take cover), at any given moment (choice nexus), there are only a few choices that can be made. For example, one cannot literally choose from ANY place to take cover, as there are only a few places within immediate reach before one is likely to be attacked. Likewise, there are only a few logical weapon choices to make at any given choice nexus; the assault rifle is best for mowing down hordes of swarming zombies, while the Molotov and shotgun are best for killing large zombies called "tanks." What *Left 4 Dead* and other games with stereotypical action gameplay lack in number of choices (what Wolf calls parallel interactivity) is in this case made up for in the frequency of choice nexus over time (what Wolf calls serial interactivity). There is very little time to consider your individual choice options because gameplay in *Left 4 Dead* is predominantly characterized by repeated choice nexus with little latency. This makes a certain amount of sense from the perspective of extraneous cognitive load; high choice numbers (parallel interactivity) AND high frequency choice nexus (serial interactivity) would quickly overload the abilities of most players, and game testing reveals these limitations. As one aspect increases, the other should, in general, decrease. This can be seen in games like those in the *Civilization* series (see Figure 1).

Some might argue (and we would not disagree) that there are action games with more parallel choices (e.g., weapons, running vs. hiding, inventory, armor, etc.) and periods of gameplay with lower choice nexus frequency. However, just because a game has many *potential* choices at a given juncture, only a subset of those choices is related to that particular juncture. While any game theoretically has access to all of the game controller options—graphics levels, armor, weapons, navigation throughout the environment, etc.—serial interactivity junctures will of necessity limit those options to what is thematically relevant and chronologically possible. I may have 100 different things I *could* do, but if I am in the middle of a firefight, I am not going to check inventory, change armor, invert my game controller axis, etc. There may be other junctures in the game where I can pause and reload, equip weapons, etc., but the archetypal interaction in a first-person shooter (FPS) game is firing while under fire.

Likewise, games like those in the *Civilization* series support near-continuous serial opportunities for interaction, but they do not require it. In fact, they encourage systemic changes (high parallel interactivity) interspersed with periods of observation (serial interactivity) using time compression tools. So any games that share similar features and characteristics of games like the *Civilization* series will be characterized predominantly by an iGrid as seen in Figure 1. Of course, one can imagine any number of games that blend or bend genres, but one can also easily imagine that iGrids could be developed for different parts of those games, and that they would capture the archetypal patterns we imagine for different genres, accordingly.

Gameplay Types and Game Genres

iGrids, as measures of gameplay type, become useful tools for discussing the differences in games that are likely to impact learning. While it is possible to rely solely on these grids in our discussion of gameplay and problem solving, we recognize that mapping what for many is a new tool/concept to existing mental models of game genres will be helpful. Accordingly, we have pulled from several existing taxonomies (most notably, Apperley, 2006) to construct a basic framework of a game classification based not on genre (although we rely on many of the same names and labels as existing genre-based classification systems) but on gameplay characteristics that are commonly associated with such systems. By clarifying or redefining (for some types of games that have been defined differently by researchers) the stereotypical *characteristics* of each gameplay type, we are better able to support the analysis of the problem types and the cognitive processes in gameplay as well as the interrelationships with different types of learning.

We make no assertion that this has any value to the field beyond the ability to organize our discussion of problem solving and games. The reader is referred to any number of excellent texts on genre and game classification, including Ian Bogost's *Persuasive Games: Videogames and Procedural Rhetoric*, and the works of Lee Sherlock, Jasmina Kallay, and Sanna-Mari Äyrämö & Raine Koskimaa, all in this volume. Rather, our system is used solely as a tool for the larger analysis of problem typology and learning outcomes. By providing a description, an example, and an iGrid for each gameplay type below, our analysis can be adapted or applied to whatever classification system is desired.

With this in mind, gameplay types in the following discussion will be divided into six main categories: Action, Strategy, Simulation, Adventure, Role-Playing, and Puzzles. In the following sections, we will also discuss these categories in terms of the nature of the games, muscular–sensory coordination, muscular–cognition coordination, and reflex requirements during the gameplay for each type of game.

Action

We define action games as the type of games where the gameplay mainly consists of activities that require fast reaction time, eye–hand coordination, and reflexes, and in many cases also a familiarity with attack patterns of the game system. FPS, sports games, fighting games, and platform games are typical games under this category. We recognize that conceptually, many readers will be troubled by conflating sports and fighting games within the same category, but we remind the reader that our system is based on the alignment of gameplay with the cognitive, structural, and domain requirements of different problem types, not on narratological or fantasy characteristics of the games. FPSs refer to games where the gameplay is characterized by avoiding being killed and eliminating all enemies with the means (usually in the form of shooting) provided in the game. Sports games are electronic versions of sports that are played in the real world, such as football, tennis, or baseball. Fighting games usually feature one-on-one fighting (e.g., *Mortal Kombat*, Midway, 2004). The player wins the game by defending him/herself and also executes quick and effective attacks to defeat the opponent. Platform games generally refer to the types of games that require the player to perform a number of actions such as jumping, bouncing, running, and so forth, in order to advance through the game. The context and actions for platform games are usually fanciful or imaginary (e.g., *Super Mario Bros*, Nintendo, 1985). iGrids like the one for *Left 4 Dead* in Figure 1 are typical of the gameplay observed in what we call Action games.

Simulation Games

Simulation games are a somewhat problematic category because of the inconsistent categorization of simulation games by different researchers or game designers. For example, Frasca (2003) defined simulations as any game that simulates real-world activities. Apperley (2006) followed the same line of reasoning and included sports games and simulation of the dynamics of city growth as examples of simulations. However, we find this definition to be too broad and problematic in terms of distinguishing specific cognitive processes during gameplay. For example, by Frasca's (2003) definition, both a computer game that simulates the TV game show *Deal or No Deal* and the video game *SimCity* would be classified as simulations. Yet, the cognitive processes in which the player engages are completely different when playing these two games. Whereas *Deal or No Deal* is a game of chance and *SimCity* is a test of the ability to optimize a system by strategically balancing a variety of factors.

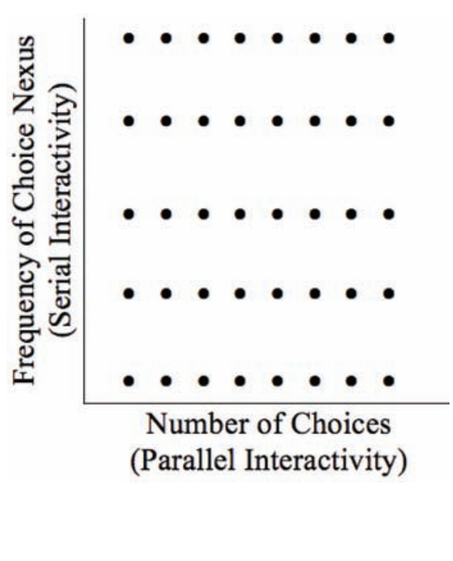
To better support analyzing the cognitive processes entailed in these types of gameplay, as well as the problem types associated with them, we define simulation games more narrowly as being characterized by the operations of a given system, for example, flying an airplane, driving a car, or operating machinery. We reserve the term "strategy" for games like the *SimCity* series (see below). The defined characteristics of simulation games include a requirement of specific domain knowledge about the system, specific procedural knowledge about operating the system in normal conditions as well as handling emergency situations, and coordination among cognition, sensory information processing, and muscular movement control. Simulations are also a performative oriented type of game. Successful simulation gameplay consists of accurate, effective, and efficient coordination among the player's domain knowledge, receiving and processing environmental information (situation awareness), quick response to changes in order to make optimal decisions, and performing precise muscular–motor skills in response to the desired course of action. Figure 2 presents an iGrid for a typical simulation game. Depending on the complexity of the simulation, which typically is simplified at early levels with more and more complexity of the system revealed as the player builds expertise, iGrids can vary in number of parallel choices. In general, however, they will look like the grid pictured in Figure 2.

Strategy Games

As we discussed in the simulation game section, strategy games and simulation games share a blurry boundary because of a lack of consensus on the definitions of these two categories of games. In this chapter, we define a strategy game as being characterized primarily by gameplay that involves regular episodes of careful planning, decision making, execution of actions, and adjustment of the actions in order to reach the goal of the game, which typically comprises optimizing the system the player is managing. The most prominent distinction between strategy games and simulation games, in our definition, is the degree of physical, muscular, or psychomotor manipulations involved in the reaction execution during the gameplay. Playing strategy games requires a high level of cognitive processing power in order to engage in analytic reasoning, logical thinking, strategic reasoning, and systemic reasoning. When a decision has been made, the player can execute the desired actions by giving commands. The analysis and decision-making processes in strategic games usually do not require fast reaction times as they do in simulation games. The *SimCity* series of games are examples of strategy games.

The gameplay in strategy games requires a highly sophisticated level of cognitive thinking skills and relatively advanced domain knowledge (although not prior knowledge, necessarily). In order to play and win the game, the player has to

Figure 2. iGrid for simulation games



develop an understanding of the system (e.g., a city) and the nature and behaviors of all its components. Also, to maintain and ultimately optimize the system (e.g., balancing health and growth), the player has to strategically balance all components, elements, and aspects that constitute the system. Ideally, there will be multiple ways to reach the goal of the game in the game design. iGrids such as the one in Figure 1 for the *Civilization* series games are typical of strategy games.

Adventure

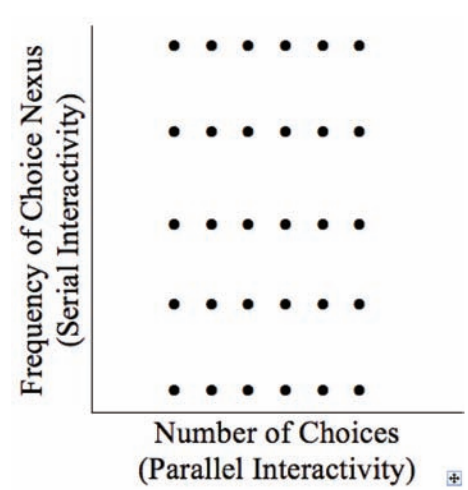
Although Apperley (2006) has classified adventure and role-playing games in the same category, in this chapter, we will define these two types of games as independent categories. We will discuss the reason for this decision shortly. We define adventure games as a broad category of fantasy games in which the player has to overcome a series of obstacles (usually related narratively) to reach the final goal or destination. The contexts of adventure games are usually some kind of fantasy, which allows endless possibilities for the contexts (the game world) of the games. Adventure games can place the player as a hero on a quest in a mythical land, as an artist in a dreamworld, as a detective (or wrongly accused fugitive) in a city, or as any number of characters and contexts anywhere on the continuum from realism to fantasy. In addition, the obstacles encountered can be of any type, ranging from simple puzzles to complex strategy play, depending upon the complexity and sophistication of the game. Thus, adventure games can be seen as a category that may combine a variety of other categories and gameplay characteristics. However, the most critical distinction between adventure and other categories of games seems to us to be the degree of fantasy (in the sense of narrative backdrop or context rather than the genre) in the game. The elements and degree of fantasy determine how the game player reasons through the problems encountered and solved, which Myers (2003) refers to as the "laws of physics" and "law of play" (p. 12). For the purpose of studying games from an instructional design perspective, this distinction is absolutely critical. Not only does this allow us to distinguish simulations from adventure games, but the distinction also provides us with a means of judging the relevance of subject matter or domain-specific knowledge, reasoning, and skills.

Because adventure games can combine so many different play characteristics, they are perhaps the hardest to capture with an iGrid. However, while there may certainly be periods of interaction in an adventure game that are characteristic of action games, adventure games are most *likely* to comprise opportunities for reflection and choices that require long-term planning and strategy. As a result, they might best be characterized by the grid in Figure 3.

Role-Playing

As we mentioned above, adventure and role-playing games are sometimes classified as the same type of gameplay (e.g., Apperley, 2006). One difference we see between these two gameplay types lies in the player identification with the protagonist. Players may be more likely to develop a psychological or emotional attachment

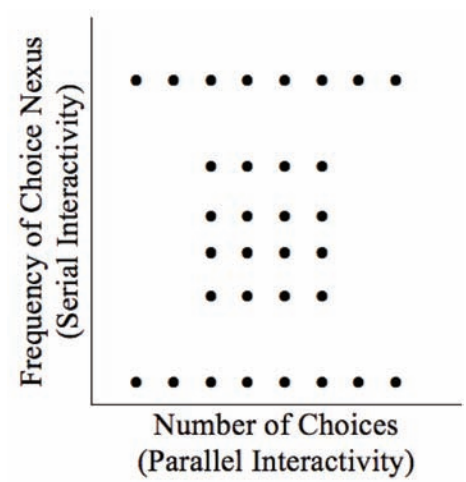
Figure 3. iGrid for typical adventure games



to the character they are playing in role-playing games than adventure games. This is because the characters in role-playing games may carry more salient and complex personalities than the characters do in adventure games and because the player has much more control over and investment in their character's looks and abilities in role-playing games than adventure games.

Another reason for separating these two categories is the availability and increasing popularity of massively multiplayer online role-playing games (MMORPGs). The addition of persistent worlds which continually evolve in the player's absence, and the cooperative play element inherent in MMORPGs brings a whole new dimension into gameplay that the adventure game, which we define predominantly as a single-player game, does not afford. The cooperative dimension of role-playing games enriches the complexity of psychological and social interactions in the gameplay. Similar to adventure games, role-playing games can and often are a combination of other game types, such as shooter plus war strategy. Yet, we argue that role-playing should be in its own category as it contains unique human psychological and social dimensions that could have significant instructional implications (e.g., see Yannuzzi & Behrenhausen, this volume, and Anderson, in press), especially in terms of problems and problem solving. iGrids for role-playing games are most likely to reflect the one depicted in Figure 4, where periods of fighting or action gameplay are interspersed with time for reflection and intense periods of modification of characters and resources (e.g., selling inventory, equipping items, forging new items, trading, building).

Figure 4. iGrid for role-playing games



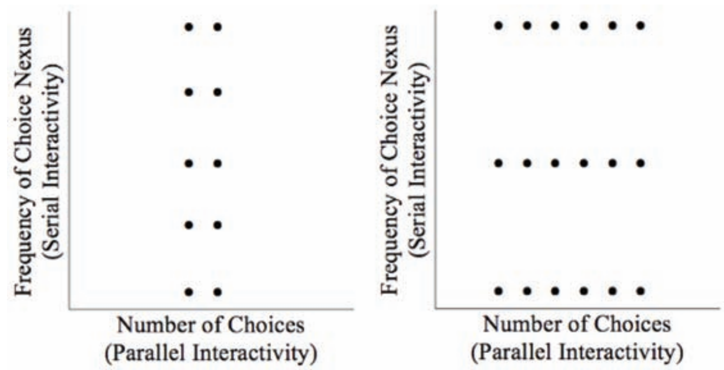
Puzzles

Puzzle games refer to any games that are relatively low- or noncontextualized, with few rules, and which can usually be solved through logical reasoning. The criteria for winning these games are often tied to the number of moves (e.g., match sticks game), the length of time spent, etc. There can be an indefinite number of variations developed from one basic puzzle, and puzzle games can appear as stand-alone games. Very often, however, they are embedded in other types of games, such as adventure, role-playing, or action games. Puzzles are often critical to enriching the engagement, challenge, and entertainment of gameplay in other gameplay types. Because puzzles can incorporate time constraints, the frequency of choice nexus can be varied. However, given the inverse relationship between serial and parallel interactivity described earlier, we can envision two typical forms of iGrids for puzzles that differ primarily in the number of choices presented at a given time, as seen in Figure 5. In the first example, we see a puzzle that provides a choice nexus at a fairly frequent rate over the course of the game, with only two options available at each point. In the second iGrid, we see a puzzle that presents fewer opportunities to make a choice, but requires the player to choose from more options each time.

A Typology of Problems

In the last three sections, we have discussed different types of gameplay and the cognitive skills and other dimensions upon which problems may vary.

Figure 5. iGrids for the most common forms of puzzle games



Having now outlined these four key components (structuredness, cognitive composition, prerequisite domain knowledge and iGrids/gameplay types), we now turn our attention to problem typology. In doing so, we will discuss 11 different problem types proposed by Jonassen (2000) within the context of their structuredness, cognitive composition, domain knowledge, and likely gameplay types and corresponding iGrids. We will also provide examples of both the problem types and the kinds of games we envision best supporting them and, where possible and relevant, the other problem types that may possibly be combined or related to a given problem type.

Jonassen (2000) has constructed a comprehensive typology to categorize different types of problems and their nature and characteristics. This typology consists of 11 types of problems:

- Logical problem
- Algorithm problem
- Story problem
- Rule-use problem
- Decision making problem
- Troubleshooting problem
- Diagnosis–solution problem
- Strategic performance problem
- Case analysis problem
- Design problem
- Dilemma problem

In the following sections, we will describe these types in terms of the activities the problems require to solve them, the context in which the problems usually appear, the nature of the problems, and the structuredness of the problems. These problems are, in *general*, presented in order from most to least structured, and from least to most complex. We will further map these to the gameplay types and iGrids that are best aligned with them and provide examples of some games that exemplify our classification.

Regarding these latter two mappings, however, it should be noted that the scope and complexity of any given problem plays a key role in determining which kind of gameplay would be suited for supporting the given problem type. If the problems are small in scope, they may be integrated into a wide variety of games. This is because any given game may employ multiple gameplay types. It is worth repeating that our gameplay types are NOT genres, nor are they intended to necessarily represent any single game; rather, they are descriptions of gameplay that (in some cases) share their names with game genre classifications. Accordingly, when we associate iGrids and gameplay type labels, we do so to indicate the style of gameplay that, *while occurring*, will best support the problem type and its requirements regardless of whatever other gameplay types might occur at other times in a given game. Further since all problem solving may have fluency and/or automaticity as a long-term goal, our assertions about pace of play (e.g., action vs. adventure vs. strategy) should be understood as most applicable to novices during early or intermediate stages of expertise. Players with high-expertise levels within a domain and problem type might be expected to have reduced intrinsic cognitive load and therefore be able to solve problems with gameplay types with more frequent serial interaction, parallel interaction, or both, which may open up additional gameplay types for potential use.

Logical Problems

Logical problems usually involve overcoming a small number of obstacles and a set of rules which have to be complied with in order to achieve the goal. This type of problem is at the far end of well-structured in Jonassen's (1997) structuredness continuum of problems. Solving logical problems typically involves utilizing concept and principle types of knowledge (e.g., propositional logical principles) and logical thinking and analytic thinking processes. Logical problems are often abstract and context-free. Therefore, domain-specific knowledge is not required. *Towers of Hanoi* is an example of a logical problem. There could be situations where subject matter domain knowledge may be required when logical problems are embedded in a more complex, context-specific problem, for example, writing an essay that flows logically.

iGrid and Gameplay Type

Because there are a small number of rules/constraints involved in logic puzzles, and because of the high degree of structure, we believe that puzzle gameplay (Figure 5) is most compatible with logical problems, followed closely by adventure (Figure 3). A logic problem will most commonly have rules and constraints that determine how certain resources can be arranged. This might make logic problems more appropriate for puzzle gameplay like that depicted in the second iGrid in Figure 5. On the other hand, solving a logic problem may, at least for novices, involve isolating one or two variables/constraints and making small moves to test results, which is closer to the gameplay type depicted in the first iGrid of Figure 5. Adventure gameplay also seems well suited to logic problems because of the high parallel interactivity. In both cases, the serial interactivity remains low enough to allow for processing, designing, and evaluating solutions to the problem between moves.

Example Game

Perhaps one of the best examples of gameplay that supports of the logic problem is the seminal game *7th Guest* (Trilobyte, 1992). This popular game took place in a mansion filled with puzzles that had to be solved, many of them logic problems. For example, a cake puzzle required that the cake be cut into six pieces, each having two gravestones, two skulls and one blank square, and with all the squares touching on at least one side (two rules, three characteristics).

Algorithmic Problems

These types of problems require applications of one or a series of procedures to be performed in order to solve a mathematical equation. The problem solver has to execute the steps in the procedure(s) in a certain order to reach the final goal. Algorithmic problems are well structured, abstract, and noncontextual in nature. When solving algorithmic problems, the most critical knowledge includes domain-specific (i.e., mathematics) procedural knowledge, concepts, and principles, and typically involves logical thinking processes. Problem solvers do not need subject matter domain knowledge in order to solve algorithmic problems. These types of problems are commonly seen in school settings. Similar to logical problems, algorithmic problems are often part of more complex problems, such as story problems or design problems. Examples of algorithmic problems include solving [(3+7)*6]/4, calculating the standard deviation of a set of data, or the nonmathematical example of the procedure for changing a tire.

iGrid and Gameplay Type

Their reliance on procedures and steps make these kinds of problems applicable perhaps to several gameplay types. Because they involve logical thinking and are highly structured, like logic problems, they are suited to puzzle and adventure gameplay types (see Figures 3 and 5). However, the addition of procedures also opens them up to a lesser extent for use with action gameplay. Many games provide problems that must be solved through execution of several actions in a specific sequence. When games do this with a time limit on the puzzles, the gameplay begins to resemble an action game (see Figure 1). Regardless, the key characteristic that differentiates gameplay related to algorithmic vs. logic problems is sequence. Logic problems *may* require sequential actions, but algorithmic problems *always* do.

Example Game

A good example of gameplay that supports algorithmic problem solving (albeit nonmathematical in nature) is *Phantasmagoria* (Sierra On-Line, 1995). In the final chapter of *Phantasmagoria*, the main character (Adrienne) confronts the truth about the haunted house she is living in and the disastrous effect it has had on her husband's mental health. In the culminating scene, she must execute at a minimum (depending on which items have been gathered during prior gameplay) 18 separate steps in the correct sequence (e.g, grab acid, throw it, pick up a book, hand an object to her tormentor, pull a lever, exit through a secret passage, and so on). This gameplay is characteristic of action gameplay, despite the fact that most of the gameplay in *Phantasmagoria* is closer to the adventure gameplay type.

Story Problems

Story problems, sometimes also called word problems, are context-bound, although not necessarily realistic. Solving story problems requires domain-specific declarative knowledge, procedural knowledge, concepts, and principles. In Jonassen's (1997) well- and ill-structuredness continuum of problems, story problems are one step away from the well-structured end and more complex than logical and algorithmic problems. Thus story problems can be deemed as precontextualized problems that lie between purely abstract problems (such as algorithmic problems) and fully contextualized problems (such as configuring a subway train schedule). Engaging in the process of solving these types of problems requires logical and analytic thinking, unless the problem solver merely employs formula-based methods (van Heuvelen, 1991) or direct translation strategy (Jonassen, 2003). At times, analogical reasoning could facilitate the solving of story problems (e.g., using worked-examples; Van Merriënboer, 1997). Story problems are often seen as part of the more complex types of problems that we will discuss next. A typical story problem might be "A train drives at a speed of 70 miles per hour, and there is an average of 5 miles between stops on a subway train route. Given that there are 10 stops on this route, how many hours would it take for the train to travel between the starting and the end points?"

iGrid and Gameplay Type

Depending on the nature of its incorporation into any given game, a story problem could once again be well suited to adventure and puzzle gameplay types (Figures 3 and 5). This is because story problems may have at their heart algorithm or logic problems. But because they may *also* be context-bound, be less structured than the first two types, and allow for other strategies (e.g., analogical reasoning), they are unique. This is most likely to manifest itself in the narratological/endogenous fantasy aspects of their integration within a game, more so than in different gameplay types, per se. When they are integrated within more complex problems (as outlined later), story problems may be associated with other gameplay types.

Example Game

Frankly speaking, it is difficult to identify game examples of story problems as opposed to logic or algorithmic problems that are or could be contextualized as stories within given gameplay. The format and structure of a "traditional" story problem is somewhat antithetical to gameplay conventions, in part because story problems are highly structured; many games want to allow players to discover essential elements of the problem distributed throughout the game. Accordingly, it is most relevant to consider story problems in games as types of distributed algorithmic or logic problems with short duration and that may be more implied than explicitly presented. As a hypothetical example, one might be able to integrate a story problem in a game like *Agatha Christie: And Then There Were None* (AWE Productions, 2005). If the trip by boat to the island takes 54 minutes, if we know that four people came out on three trips during one day, and if we have alibi statements from characters regarding their locations at certain times, then it might be possible to disprove one or more alibis by using math to check out their statements. This could be used as a design heuristic, perhaps, in order to integrate the problem type by requiring the player to assemble the relevant parts of the story and to recognize (transfer) story problem-solving strategies in order to solve this part of the problem. Once the player has done so, the gameplay type during that solving process best reflects adventure and puzzle gameplay.

Rule-Using Problems

Rule-using problems, in essence, are the types of problems that likely have multiple solution paths, yet the actions taken along the solution paths are constrained by a set of restrictive rules. They can be highly noncontextual, such as chess or card games, or they can be fully contextualized and fairly complex, such as filing a tax return. The structuredness of rule-using problems can range from well structured to semi-well-structured, depending upon the complexity of the problem. Domain-specific declarative knowledge is *usually* required to solve rule-using problems, while domain concepts and procedural knowledge *may* be needed in some cases. When solving rule-using problems, the problem solvers usually engage in the processes of logical and analytic thinking while complying with the rules. For example, rule-using problems can themselves be logic problems that require rules, such as arranging seating for guests in a diplomatic formal dinner where the formal dinner seating convention has to be complied with and the guests' preferences also need to be taken into account. Rule-using problems are often seen as part of more complex types of problems, for example, decision-making problems, strategic performance problems, and others that we will discuss shortly.

iGrid and Gameplay Type

All games are, to a certain extent, rule-using problems themselves, so it may come as no surprise that we believe that nearly all gameplay types are potentially useful for these types of problems. Of all gameplay types, strategy and role-playing are perhaps best suited to rule-using problems, however. Role-playing gameplay is perhaps the most open-ended of gameplay types, placing a premium on socially negotiated paths among multiple paths constrained by rules for navigation, fighting, interaction, resources at hand, etc. Figure 4 shows the iGrid associated with this gameplay type. Simulation gameplay type is not included here because it has unique requirements (based on our definition) that include psychomotor skills and decision making. Rule-using problems are primarily associated with nonphysical contexts and also do not necessarily involve decision making. Strategy gameplay is also highly open in its support of multiple strategies and paths to the end goal (see Figure 1). Regardless of the gameplay type selected, rule-using problems should have opportunities for low serial interactivity to allow for processing and thinking, although the faster pace associated with parts of role-playing gameplay (as with action gameplay) could be adapted as well for more expert learners.

Example Game

An example of a game that supports rule-using problems via the role-playing gameplay type is *Sacred 2: Fallen Angel* (Ascaron, 2009). In order to defeat many of the major monsters (bosses) in the game, the player must master different strategies, weapons, and abilities, which are in turn impacted by all the attributes of the character, as the player makes choices about where to invest resources. As a Dryad (one of five character types), you may choose to specialize in ranged weapons (e.g., bows) that do a certain amount of damage. However, because you know that some bosses are more or less susceptible to damage related to fire or ice, and because you can equip bows to do more damage by "forging" them with ice crystals or lava rocks, you need to have two bows: one for each damage type. Likewise, you have three "combat aspects," each with five combat "arts," one of which is considered a "buff," and all of which are improved by eating "runes." Combat arts each have their own respective damage type that will be better or worse for certain bosses. The more runes you eat, the more powerful you get, EXCEPT if you exceed your character-level abilities, in which case the runes slow down your regeneration time (how soon you can use them again). ALL of this has to be managed within the context of a given fight. For instance, fighting the octagolamus (a giant squid–snail thing) requires causing fire damage, but it cannot be damaged as fast as it regenerates without using something else. In the Dryad's case, this might mean getting close enough to cast a combination of three combat arts (e.g., "tangled vine" to hold the octagolamus in place, "edaphic lances" to create a series of thorns that do damage while it is held in place, and "black curse," which lowers the boss's attributes so damage is more effective) and *then* firing the fire bow to cause more damage than the boss can repair. This represents just one small part of such role-playing gameplay in which rule-using problems are routine.

Decision-Making Problems

Like rule-using problems, decision-making problems also typically involve multiple options for which the problem solver has to evaluate the advantages and disadvantages and make the most viable selection. When solving decision-making problems effectively, the problem solvers are engaging in the processes of researching as much relevant information as possible, analyzing and assessing the pros and cons of the options, making a value judgment of each option, and then ultimately deciding which option to take. Decision-making problems fall in the middle of the structuredness continuum. Domain-specific concepts and principles are the foundation for solving this type of problem, with the assistance of domain declarative knowledge. In order to perform the necessary problem-solving tasks, logical, analytic, and strategic thinking are key cognitive skills. Systemic and metacognitive thinking may or may not occur, depending upon the nature of the situation. Choosing a retirement plan or deciding which school to attend is an example of a decision-making problem. A decision-making problem can sometimes be a complex version of a combination of logical problem and rule-using problem or can be part of the following types of problems.

iGrid and Gameplay Type

Like rule-using problems, decision problems may incorporate other problem types and are also good for a wide variety of gameplay types, including action (Figure 1), role-playing (Figure 4), and adventure (Figure 3). However, unlike rule-using problems, decision-making problems are not well suited to puzzles because the complexity of decision-making problems outstrips the representational ability of most puzzle gameplay. Decision-making problems also bring the possibility of supporting simulation gameplay (Figure 2) for the first time. Decision-making problems, with their more complex and sophisticated nature, begin to get at what simulations often require. In our opinion, however, strategy gameplay (Figure 1) may hold the most potential for supporting decision-making problems, given the prevalence of decision making, the number of choices presented at a given time, and the resulting need for reflection (low serial interactivity) in this type of gameplay.

Example Game

The classic strategy game series *SimCity* is a well-known example of a game that supports decision-making problems. In this game, the player must make a series of decisions, beginning with decisions about a location to begin building on (e.g., by a river or by arable land) and progressing over time to include tax rates, amount of land or revenue to devote to industry vs. residential vs. the arts, transportation, farmland, infrastructures like fire and police, etc. All of these options require continual evaluative decisions based on tradeoffs (taxes pay for police, but high taxes lead to poverty, dissatisfaction, and riots, which all require police). If the player tries to make decisions once and never revisits those issues, the system quickly spins out of control.

Troubleshooting Problems

Troubleshooting problems are commonly seen in everyday lives. They may be as complex as scientists troubleshooting a computer glitch on the Spirit rover on Mars, or a mechanic troubleshooting an alternator problem in a car or as simple as troubleshooting a lamp with a burned-out light bulb. In terms of problem structuredness, troubleshooting problems can range from semi-well-structured to semi-ill-structured. Solving troubleshooting problems usually involves highly specific domain knowledge, including concepts and principles. Prior domain declarative knowledge is necessary but not the focus of learning how to troubleshoot. It is assumed that problem solvers already possesses a certain degree of declarative knowledge when they troubleshoot or learn how to troubleshoot problems (this assumption is also true of all of the following problem types). Hegarty (1991) suggested that domain procedural knowledge may be critically important in troubleshooting problems when a fault is identified and a procedure needs to be executed in order to restore the system to its normal state. Troubleshooting typically involves recognizing the symptoms (abnormal behaviors of system), identifying possible causes, testing the hypotheses, and then applying corrective procedures (Jonassen & Hung, 2006). Thus, analytic, strategic, and logical reasoning are the main cognitive activities during the troubleshooting process. An experienced troubleshooter also relies on analogical reasoning when encountering similar problems. Systemic and metacognitive thinking may not necessarily be performed by all troubleshooters, but when they are, troubleshooting skills are elevated.

iGrid and Gameplay Type

Whereas we have gone from problem types that are supported by a few gameplay types to those that are supported by a majority of gameplay types, with troubleshooting the picture is much clearer. Simulations (Figure 2) seem to be the only gameplay type suited to troubleshooting. While they range in complexity, the scope of troubleshooting problems remains narrow—never reaching the scale of a *SimCity*, for example. Simulation gameplay focuses on systems (as does strategy gameplay), but they are narrower systems that are tractable via hypothesis testing, for example.[4] Their emphasis on procedural knowledge also makes troubleshooting problems well suited to simulation gameplay.

Example Game

Any game that makes significant use of simulation gameplay makes a good example, and those who have played any variant of the *FlightSim* games will easily see the connection. Since the authors have not played a lot of simulation games and because we suspect this may also be true of many readers, we will focus on *The Incredible Machine* (aka *Contraptions*; Dynamix, 1993) a game that our students have considered for use in K-12 classrooms to teach science. *The Incredible Machine* requires players to design contraptions out of a variety of moving parts (e.g., conveyor belts, funnels, hard or soft surfaces, springs, tunnels) to accomplish different goals (e.g., move the ball from Point A to Point B). Once each machine has been initially designed, gameplay shifts to troubleshooting as the player begins to figure out how and why the system is breaking down. Players move parts, replace parts, change speed, etc., to test what happens and use the results to refine their model of the system and where it is breaking down.

Diagnosis–Solution Problems

These types of problems are similar to troubleshooting problems in terms of the cognitive processes involved. The most common diagnosis–solution problems are medical in nature. Doctors diagnose patients' complaints, identify possible causes of the disease or discomfort, and give a prescription to remedy the problem. Both diagnosis–solution and troubleshooting problems start with a display of symptoms or a fault state that needs to be restored back to a normal state. However, diagnosis–solution problems are usually more ill-structured and complex than troubleshooting problems because there is much more unknown with respect to human physiology than with man-made systems, which results in a higher degree of intransparency of the problem space (Frensch & Funke, 1995b; Jonassen & Hung, 2008; Spering, Wagener, & Funke, 2005). It should be noted, however, that a diagnosis–solution problem need not always be medical. To the degree that a system is open, ill-structured, complex, and intransparent (much is unknown about the system), diagnosis–solution problems may be found. Solving diagnosis–solution problems requires all types of domain knowledge and the process is cognitively engaged at a deep level. The problem solver has to analyze the symptoms, logically rule out the irrelevant or the impossible, analogically reason with similar cases, strategically test the hypotheses, and then prescribe solutions from a holistic (systemic) perspective. Moreover, metacognitive thinking is critical in this type of problem solving because it is an important mechanism for problem solvers to accumulate their knowledge repertoire, skills, and experiences. Diagnosing a patient with an irregular heartbeat rhythm and determining why a marker species is dying off in an otherwise healthy water ecosystem are examples of this type of problem.

iGrid and Gameplay Type

As might be expected, diagnosis–solution problems are supported by simulation gameplay, just as troubleshooting problems are (see Figure 2). The key to supporting diagnosis–solution problems with simulation gameplay lies in the characteristics of the system under diagnosis, as described above. Medical simulations will support medical diagnosis–solution problems, of course, but it is important to remember that much depends on the underlying conceptual model of the simulation. It would be possible to build a highly limited, well-structured, closed-system medical simulation that would in fact NOT reflect true diagnosis–solution problems. When designing such systems and games, it is necessary to at least simulate complexity through random factors and/or to rely on algorithm-based programming (see Crawford, this volume). A good way to do this is to collect real-world case data (e.g., actual medical diagnosis records), including the false leads and the data that led to them.

Because of the complexity, ill-structuredness, and intransparency of the systems underlying diagnosis–solution problems, strategy gameplay may also support these problem types (see Figure 1). We argued before that strategy gameplay does not support rule-using problems because the underlying systems in strategy gameplay tend to be too open, ill-structured, and complex. Here, this is precisely what allows this type of gamplay to support this problem type. There tend to be many factors and criteria to consider at one time (high parallel interactivity), but changes take time to occur and require significant cognitive processing to evaluate and use as inputs to generate new hypotheses and courses of action (low serial interactivity).

Example Game

There are several examples of serious games that focus on medical training (e.g., *Pulse*, by BreakAway Ltd.), but that does not mean that diagnosis–solution problems are being implemented by these games. Game artifical intelligence is not easily able to completely simulate the complexity of the human body, especially since so much is unknown. Because we can only design a game to simulate a system based on what is known, most medical games tend to focus on well-structured problems for which we can specify prescribed solutions. Best-case practices are therefore the content under study, rather than the "messy" real-world complexity of true diagnosis. The degree to which we can simulate, if not replicate, the ill-structured intransparent nature of systems is the degree to which we can support true diagnosis–solution problems.

An example of this kind of approach to supporting this problem type is a game we are currently developing to teach scientific problem solving to middle school students (also see Gaydos and Squire's description of *Citizen Science*, this volume). Based on the National Science Education Standards for science as inquiry, science as a human endeavor, and science in personal and social perspectives, this game requires students to solve a variety of environmental problems that face their hometown. In doing so, they engage in the process of problem identification through solution, implementation, and evaluation. Problems have multiple potential causes and solutions, however, and diagnosing potential causes and proposing solutions requires several rounds of testing and evaluation; information seeking from multiple, conflicting resources; and public buy-in from constituent groups with disparate and often incompatible views. In one scenario, the player hears a news story about potential neurological disabilities on the rise. In researching the story, they find that there are several potential causes (randomness, nutrition, lead poisoning), each of which has several potential sources (e.g., lead poisoning could be waterborne or soil-based from lead paint chips, agricultural runoff, or a petroleum spill), each of which must be ruled out or in. Eventually, students must conduct soil sampling at a specific site in the game to see if there is contamination there. We randomly assign a central source of contamination, using algorithms to radiate the contamination out from that point in weakening amounts, using further random generation within a range of expected contamination values. By constraining the number of tests the player can "afford," we simulate the ill-structuredness of the diagnosis process; taking a few samples at different places yields a range of values that in some cases will be high enough to indicate contamination but in more cases will indicate values within an acceptable "normal" range. The question for the player becomes whether values at the high range of normal indicate randomness or proximity to a site with even higher concentrations. This is similar also to the old board game *Battleship* (Milton Bradley, 1943), in which one "samples" on a grid of coordinates and finds they have missed, near-missed, or scored a direct hit.

Strategic Performance Problems

In Jonassen's (2000) definition of problem typology, strategic performance problems often involve psychomotor skill performance with cognitive processes and metacognitive processes operating consciously or unconsciously within the performer. Solving these types of problems requires the problem solver to fully maintain situational awareness in order to make adjustments in response to the change of the situation/environment. Typical strategic performance problems include operating an airplane, playing in a tennis match, or driving a car. Strategic performance problems are typically ill-structured in nature, since there are a number of courses of action (solution paths) that the problem solver can take. All types of knowledge are needed when solving strategic performance problems, especially procedural knowledge. The most critical cognitive activities during problem solving of this type are strategic and metacognitive thinking. A performer could well possess the domain knowledge, yet the coordination between his or her cognition and muscular control may not occur smoothly or efficiently. Some people will need more practice with muscular–cognition coordination than others. When this happens, strategic and metacognitive thinking become critical to the acceleration of learning and refinement of the performance. In addition, analytic, logical, analogical, and systemic thinking are also supportive in most strategic performance problem cases. In some cases, this type of problem may contain subproblems of troubleshooting or diagnosis–solution problems.

iGrid and Gameplay Type

Given the name of this problem, we might expect that strategy gameplay would support this problem type. That is not the case, however. The primary characteristic of gameplay for this problem type is medium to high serial interactivity, with varying degrees of parallel interactivity, making it appropriate for action, simulation, adventure, and role-playing gametypes (Figures 1–4, respectively). The key lies in the requirement for situational awareness usually coupled with psychomotor skills. When flying a plane, one has to monitor airspeed, pitch, yaw, and altitude, using them in concert to make adjustments using pedals, throttle, and other controls. There are no long periods of time in between adjustments as there are with strategy gameplay.

Example Game

Earlier we described a game with simulation gameplay (flight simulator), and simulations are perhaps the easiest type of gameplay to see in terms of strategic performance. Instead, we will describe an action game with medium-high serial and parallel interactivity because of its fit, its contemporary nature, and because it makes a good example of how problem types can be instantiated in gameplay in ways you might not immediately classify as appropriate. In the game *Left 4 Dead* (Valve, 2008), whether played cooperatively or as single-player, players must work their way from safe house to safe house through a dark, postapocalyptic urban landscape populated by zombies bent on killing humans. In the safe house and during movement through the city, players can pick up health kits and a variety of weapons with different characteristics, each of which has trade-offs and benefits (stopping to pick them up puts you at risk but not doing so puts you at risk later). If players wait in one place too long, the game AI sends more zombies after them, so the game requires consistent (though not continuous) movement through buildings, streets, subways, tunnels, and so on (continuous situational awareness). Attacks come from six different zombie types: Tanks (strong, cause high damage, and are hard to damage), Hunters (fast, unpredictable movement, and cause average damage), Smokers (attack from a distance with prehensile tongues, hold you in place for others to damage, easy to kill), Witches (stay in one place unless disturbed, very fast, high damage, very hard to kill), Boomers (projectile vomit that causes little damage but summons Hordes), and Hordes (swarms of zombies that are easy to kill individually but must be killed rapidly to avoid becoming overwhelmed). As the player moves through the landscape, auditory and visual cues signal the presence of different zombie types (situational awareness), which requires in turn the selection of and rapid switching among appropriate weapons (metacognitive thinking and selection of options, e.g., shotguns and Molotov cocktails for Tanks or automatic assault rifles for Hordes) while charting a future path through the environment (multiple solution paths). One quickly learns the value of cover and the foolishness of running into the woods or trapping oneself in a dead-end room.

Case Analysis Problems

Case analysis problems are often used to help an individual, a company, or an organization understand the individual elements and the intercausal relationships among them in a current situation from a similar situation that has happened in the past. These types of problems have long been used in law schools, business schools, and medical education. They can be seen as semi-ill-structured because there is relatively more known than unknown in the problem space because the problem occurred in the past. Because case analysis problems are highly contextualized, domain-specific knowledge is required. The problem solver's domain concepts and principles serve as the foundation of his or her ability to solve the problem. Procedural knowledge may also be required. In terms of cognitive activities, analytic thinking dominates the problem-solving process with the assistance of analogical thinking and sometimes systemic or logical thinking. This type of problem solving also involves psychological and/or emotional evolution throughout the process when attitude change is involved either consciously or unconsciously. Again, some of this type of problem may contain subproblems of troubleshooting or diagnosis–solution problems.

iGrid and Gameplay Type

Because cases are highly contextualized, rely on analogical reasoning, and often require systemic thinking, strategy gameplay is probably best suited to this problem type (see Figure 1). These problem types require significant time for reflection, making low serial interactivity a necessity. Parallel interactivity will likely be determined by the structuredness of the domain, the amount of domain knowledge required, and the complexity of the case.

Example Game

As an example game for this problem type and the next, we return to the *SimCity* series of games. One of our students developed a lesson plan around these games that required students to design a solution to rebuilding a city that had been destroyed by a variety of natural disasters. The point was not to have students learn about natural disasters in urban planning but rather to allow them to explore the various paths possible in urban planning and the differences that philosophical beliefs make in the long-term evolution of a city. Students were required to establish key goals and indicators for the redevelopment of their cities (e.g., focus on the arts, on public spaces, on industry, on entertainment) and to rebuild their cities accordingly. The resulting cities were then compared across different groups to discuss the impacts that planning decisions have on long-term success and how successful cities can be highly divergent. This in itself is more of a design problem (more on that later), but it is a short step to taking the resulting cities as cases and putting them in the hands of learners who face a different, but related problem. For example, simply using a differently configured city and cause of disaster would qualify the resulting experience as a case study. One might also present a different problem; a city with an insufficient tax base and low industry could be presented with the documented case of a city that recovered from a natural disaster by focusing on industry or a different city that focused on building a "greener" city that balanced industry and environment.

Design Problems

Design problems are highly complex and ill-structured. They usually have a vague goal state and ill-defined criteria for evaluating the success of solving the problem, and an indefinite number of solution paths (Jonassen, 2000). Therefore, on the continuum of problem structuredness, design problems are at the far end of ill-structured and complexity. Engineering design problems, instructional design problems, and interior design problems are examples of design problems. These types of problems are extremely contextualized, thus requiring a solid, domain-specific knowledge base, especially concepts and principles. Also, because of their highly ill-structured and complex nature, solving these problems is a cognitively intense process. All of the higher-order thinking skills we have discussed here are required at some point in the process of solving this type of problem.

iGrid and Gameplay Type

Once again, strategy is the best gameplay type (see Figure 1) for supporting design problems, and we have described one such approach in the prior section (*SimCity*). The key lies in placing tools in the hands of the players to design solutions, whether of a physical or abstract nature (e.g., mechanical engineering vs. human engineering). As such, gameplay requires multiple iterations interspersed with time for reflection and evaluation (medium serial interactivity) and many possible solution paths and decisions (high parallel interactivity).

Example Game

Whereas the focus in our previous example was on using a case (a city that had been redeveloped after natural disaster) to reason about a new, analogous problem, here the focus is on the prior activities we described that lead up to that use of the *SimCity* games as cases. Building a city is itself a design problem, but without constraints, the pedagogical value for novices may be limited. Imposing design constraints (e.g., building for the arts, entertainment, or industry) helps to concentrate the activity as a design problem. Another game we described earlier, *The Incredible Machine* (Dynamix, 1993), supports strategy gameplay for significant portions of the game. Because players must build machines to specifications, engineering design problems are well suited to that game. *The RollerCoaster Tycoon* and *Zoo Tycoon* series of games are also appropriate examples but only to the extent that constraints (some of which will likely need to be external to the game) are added. David Williamson Shaffer's book *How Computer Games Help Children Learn* details other examples of design problems within games such as *Sodaconstructor* (Sodaplay, 2007).

Dilemma Problems

According to Jonassen (2000), dilemma problems are often deemed to have no best solutions. Any solution to a dilemma problem often inherently incurs a similar amount of sacrifices or harm to the individuals involved or the situation when compared to other solutions. The Israeli–Palestinian conflict is a prime example of a dilemma problem.

In addition, dilemma problems usually consist of multiple interest groups or stakeholders whose interests often conflict with each other. Similar to design problems, dilemma problems are also extremely complex, highly contextualized, and very ill-structured. Excluding analogical reasoning, which may or may not be required depending on the nature of the problem, the problem solver engages in the tasks that demand exceptionally high levels of all other types of cognitive processes and thinking skills. While domain-specific knowledge is also critical to dilemma problems because of the high level of context specificity, principles are the most vital form of domain knowledge for supporting this type of problem solving. One potential unintended outcome of solving dilemma problems (it may not be true for all problem solvers) is a change of attitude. This change may be too subtle to notice. Yet, it is logical to assume that a person who goes through solving a dilemma problem has to take all sides of concerns into consideration as well as consider the problem from a systemic or holistic perspective. This person will also experience some degree of psychological or emotional realization, which could result in attitude change.

iGrid and Gameplay Type

Dilemma problems are at the heart of many games for change or persuasive games. For example, *September 12* (Newsgaming.com, 2003) presents the player with a dilemma of whether to kill terrorists (and civilians in the process, thereby creating more terrorists) or allow terrorists to have a free reign (the implication being that terrorist attacks will continue in the world). However, the dilemma in this game is highly simplified and far too well structured to be a good example of this problem type. It is, in essence, a dilemma problem that has been distilled down to the core of two choices. The gameplay types that best support dilemma problems are strategy and role-playing (see Figures 1 and 4, respectively). The more complex and ill-structured the dilemma problem is, the more likely it is that the different nuances and longer interaction times will result in attitude change. Therefore, games like *Darfur is Dying* (mtvU, 2006) have the potential to present larger, more complex dilemmas and thus impact attitude change. Games that employ role-playing will also support dilemma problems in part because of the personal investment players have in their avatars and the social aspects of this kind of gameplay type. Therefore, strategy-roleplaying hybrid games should be ideally suited to dilemma problems and attitude change. Regardless, gameplay type should reflect low-to-medium serial interactivity to allow for consideration of the different factors underlying the dilemma and to identify possible paths for resolution. The exception to this is in role-playing game types, where it is possible to have periods of high serial interactivity (e.g., fighting sequences) that are themselves interspersed throughout gameplay with lower serial interactivity. In theory, there will be higher parallel interactivity as a result of problem complexity, lack of structure, and required domain knowledge.

Example Games

The game *Bioshock* (2K, 2007) pits the player against a variety of challenges in an underwater city named "Rapture." As with *Left 4 Dead* (Valve, 2008), players must make their way through the city without being killed by Big Daddies (giant modified humans in diving suits) and demented humans while collecting weapons and resources. Among these resources are plasmids, which grant special powers by virtue of genetic modifications, and which are injected via syringes. They key to unlocking the powers of plasmids lies in the collection of ADAM, which can only be obtained in the game from Little Sisters, who appear to be preadolescent girls. Little Sisters are always accompanied by Big Daddies, who must be killed before the player can collect ADAM. The dilemma problem in the game occurs with the decision on how to harvest the ADAM. One way results in the death of the Little Sister but results in a large amount of ADAM. The other way saves the Little Sister but results in less ADAM. While this choice seems to be pretty simple (two choices) the choices have a significant impact on the difficulty of the game and the way it proceeds. Additionally, whereas the binary choice in *September 12* (Newsgaming.com, 2003) is limited to the same instances and has the same results easily seen in a short period of time, in *Bioshock* these choices are distributed over the course of up to 50 hours of gameplay with relatively high frequency (medium serial interactivity), and the effects of these choices are not fully realized until near the end of the game.

LEARNED CAPABILITY OUTCOMES, PROBLEM TYPES, AND GAME PLAYING

We have discussed Jonassen's (2000) typology of problems in light of their nature, knowledge required, and cognitive processes, as well as the degree of abstractness and contextualization. We have further matched problems and associated cognitive processes and learned capability outcomes with different gameplay types. The final results of these interrelationships can be seen in Figure 6.

We used types of problems to mediate types of learning and types of gameplay. The reasoning for this is twofold. First, gameplay is a goal-based activity that consists of a series of problem-solving events (Kiili, 2007). Therefore, the type of problems in a game determines the type of cognitive activities involved in gameplay. So identification of the type of problems in gameplay can function as an indicator of what type of learning can be supported. Second, Jonassen's (2000) typology of problems not only explains the nature of different types of problems, but also discusses the learning outcomes with which these problems are usually associated. While we have discussed these in passing in the previous section, a discussion of the specific learned capabilities that each problem type best supports is important to complete the picture of games, problems, and instructional learning outcomes. Therefore, we conclude with a discussion of the relationships between types of learning, problem solving, and gameplay.

Domain-Specific Knowledge Learning

Domain knowledge learning is sometimes referred to as verbal information learning (Gagné, Wager, Golas, & Keller, 2005). Although this type of learning is at a lower level of learning in Bloom's taxonomy of learning, it provides the fundamental building blocks for enabling the learners to engage in higher-order learning. As we can see in Figure 6, domain knowledge learning occurs in all types of problem solving except for logical problems, which can be solved without any specific domain knowledge. Although all problem-solving types involve domain knowledge, there are different subtypes of knowledge acquisition and application that occur among these problems. For example, solving the types of problems that are less complex and with lower levels of contextualization, such as story problems and rule-use problems, requires more declarative knowledge learning. On the other hand, solving more complex and contextualized problems, such as decision making, troubleshooting, diagnosis–solution, and strategic performance problems entails more conceptual and principle knowledge and relies heavily on procedural knowledge. Yet, case analysis problems, design problems, and dilemma problems mainly focus on the integration and flexible utilization of concepts and principles.

How does this information help design effective instructional games? Since all types of problems, except for logical problems, involve different degrees and types of acquisition, comprehension, and application of domain knowledge,

Figure 6. Problem types, their associated cognitive processes, and learned capability outcome, and the gameplay types that might best support them. This analysis depicts the main cognitive processes involved in the problem-solving process. For the problem types that are more complex and highly contextualized, the acquisition of domain knowledge is assumed to be required, and for purposes of readability is not marked in this figure

Knowledge and Cognitive Process														
	Domain–specific knowledge[1]				*Higher–order thinking*						*Psychomotor skills*[2]		*Attitude change*[2]	
Problem type	Declarative	Procedural	Concepts	Principles	Logical	Analytic	Analogical	Strategic	Systemic	Metacognitive	Muscular movement	Muscular–cognitive coordination	Shift of belief system	**Game type**
Logical					+	+								Adventure; Puzzle
Algorithmic		+	+	+	+									Adventure; Puzzle; .
Story	+	+	+	+	+	+	+							Adventure; Puzzle
Rule-use	+	~	~	+	+	+								Action; Strategy; Roleplaying; Adven Puzzle
Decision–making		~	+	+	+	+		+	~	~				Action; Strategy; Ro playing; Simulation: Adventure
Troubleshooting		+	+	~	+	+	+	+	~	~				Simulations
Diagnosis–solution		+	+	+	+	+	+	+	+	+				Simulations; Strateg
Strategic Performance		+	+	+	+	+	+	+	+	+	+	+		Action; Roleplaying Simulations; Adven
Case Analysis			+	+	~	+	+		~	+			~	Strategy
Design			+	+	+	+	+	+	+	+				Strategy
Dilemma				+	+	+	~	+	+	+			+	Strategy; Roleplayir

[1] For Psychomotor Skills and Attitude Change: domain-specific procedural and principle knowledge and metacognitive thinking are assumed.
[2] For the learning type under Domain Knowledge, application of the knowledge is also assumed in this chart.
\+ signifies "always required."
~ signifies "sometimes required."

all types of gameplay could theoretically support basic domain-specific knowledge learning. As our problem–gameplay analysis reveals, puzzle, adventure, and action games may better support acquisition of declarative knowledge; simulations, action, and adventure could engage learners in honing their procedural knowledge; and strategy, simulation, and sometimes role-playing games may best support concept and principle knowledge application types of learning.

Higher-Order Thinking

Learning higher-order thinking skills includes a variety of cognitive reasoning skills, including logical, analytic, analogical, strategic, systemic, and metacognitive thinking. Although logical thinking could happen in a context-free condition, most higher-order thinking occurs in some type of context and involves various degrees of domain knowledge. As shown in Figure 6, diagnosis–solution, strategic performance, design, and dilemma problems require most of the higher-order thinking skills. This is consistent with our analysis that these types of problems tend to be complex, highly contextualized, and highly ill-structured. In order to deal with the level of complexity and intricacy, a sophisticated practice of integration of these cognitive reasoning skills is critical to the success of solving the problem. Thus, requiring students to solve these types of problems could providing practice opportunities for use of the associated cognitive activities, thus addressing those "critical thinking" skills we hear so much about but which are rarely operationally defined. On the other hand, case analysis, troubleshooting, and decision-making problems could engage students in various higher-order thinking activities as well. This set of problems could be considered specialized problems in terms of enhancing certain types of students' higher-order thinking skills because they tend to rely on one or two aspects of higher-order thinking. For example, case analysis problems emphasize analytic ability, troubleshooting problems focus on logical, analytic, and strategic thinking, while decision-making problems require logical, analytic, and systemic reasoning abilities to select a most viable option based on the condition given. Lastly, algorithmic, story, and rule-use problems could also be used to help students exercise higher-order thinking skills, but given their limited complexity and structuredness, the instructional effects would be less than with other types of problems.

Strategy and simulation games are perhaps the most appropriate types of games for promoting students' learning of higher-order thinking skills. Strategy games are highly cognitively oriented. They should be particularly effective in exercising students' higher-order thinking skills because the activities of these two types of gameplay involve all types of cognitive processes. Simulation games are very effective in facilitating students' development of higher order thinking skills because they require a high level and variety of cognitive reasoning in order to perform the muscular movement and to manipulate the system operation to an optimal level. However, the requirement of muscular movement control would likely take up some capacity of working memory. As a result, the exercise of cognitive processing would run at full capacity in simulation games, as opposed to strategy games where the full working memory power is devoted to cognitive processing. Therefore, simulation games may be somewhat less versatile than strategy games in enhancing the development of (pure) cognitive thinking skills. Action, role-playing, and adventure games may also provide opportunities for students to develop higher-order thinking skills, but because these games usually involve other types of gameplay activities (e.g., eye–hand coordination, quick reflex), they are less effectively used for the sole learning goal of developing higher-order thinking skills. Lastly, when the learning goal targets one or two particular types of thinking skills, then adventure and puzzle games may be an appropriate option. Puzzle games could also be embedded in more complex games, such as adventure, simulations, or strategy games. Thus they could train for one particular type of thinking skill (e.g., logical thinking) or be aggregated with several puzzle games to form an adventure game to address multiple skills.

Psychomotor Skills

Strategic performance is the main type of problem that requires the necessary cognitive processes and physical performance to support psychomotor skill learning. Psychomotor skill learning (e.g., flying an airplane, operating a crane ship, or hitting a baseball) involves perfecting both the muscular movements in a specific order and the smoothness of the transitions between each movement (Gange et al., 2005). The key to psychomotor skill learning is the coordination between cognition and muscular movements, as well as the strategic thinking that supports optimal performance. A psychomotor skill performance is usually dynamic and involves interaction between either two performers or a performer and a system.

Because of this dynamic, interactive nature of the task, cognition (i.e., strategic thinking) sometimes plays an even more important role than muscular movements. Therefore, psychomotor skill learning is an intensely muscular–cognitive process.

A number of gameplay types could afford the learning of psychomotor skills. Simulations, adventure, action, and role-playing games are appropriate but with different emphases and degrees. When the learning goal is within a specific context or profession, simulations may be the most suitable gameplay type because simulations naturally set the gameplay in a highly contextualized (or authentic) environment. Other types of gameplay, such as action, adventure, or role-playing, could be used to provide practice with eye–hand coordination or multiple modalities of inputs that are not profession-specific but could be useful in other capacities. Another advantage of these gameplay types is training for quick responses or reflexes. These are critical skills in most strategic performance problem solving and, therefore, are essential to psychomotor skill learning.

Attitude Change

Attitude change is a higher-order level of learning. It involves not only cognitive processing but also psychological, social, emotional, and affective changes of state (Gagne et al., 2005). The end result is a shift in one's belief system. The problem types that involve these internal changes of state include case analysis and dilemma problems. Both case analysis problems and dilemma problems require the problem solver to analyze all parties involved in the problem at a very deep, personal, psychological, or emotional level. However, these problems also require the problem solver to examine all parties from multiple, holistic, societal, and even global perspectives. Going through these examinations and contemplations psychologically, emotionally, scientifically, and socioculturally, it is possible that the problem solver also goes through a fundamental and philosophical retrospective journey. Therefore, solving these types of problems may bring about attitude change.

Strategy and role-playing games may be the second tier of gameplay that is likely to afford this type of learning. Strategy gameplay requires the player to manage the system (e.g., a city, a business, or a battle) to its optimal state. In order to reach that goal, the player has to have a deep understanding of each component in the system. Role-playing games have the advantage that the player is likely to develop a deep personal psychological attachment to the character that he or she plays. The features of strategy and role-playing games are capable of affording the cognitive requirements for attitude change. However, the requisite psychological, emotional, or sociocultural components in most commercial strategy and role-playing games are absent. This is understandable because commercial games are not designed to fulfill educational goals. Here, we are simply arguing that if appropriate psychological, emotional, and sociocultural components are incorporated into the game design, strategy games and role-playing games could afford attitude change learning.

FUTURE RESEARCH DIRECTIONS

The tools and processes and relationships described in this chapter suggest a variety of design research activities, some of which are summarized briefly below.

Studies of Problem Solving

If we are right about the use of iGrids and serial vs. parallel interactivity, using these grids to classify different gameplay (which will occur to varying degrees within a given game) should lead to experimental designs of specific kinds of problem-solving skills as supported by specific gameplay types. Existing research in problem solving has created a large body of potential problems that could be used as outcome measures of specific problem-solving types. Validated problem design models (e.g., 3C3R by Hung, 2006a) should be used to design analog problems that reflect the serial and parallel interactivity characteristics supported by the gameplay to increase contextual similarities.

iGrids and problem types should also be used to develop specific gameplay types, and the resulting effects on learning should be measured to validate the assumptions of our approach. Likewise, combinations of gameplay types within a given game and their ability to support corresponding problems should also be studied to see what additive or interactive effects occur, but only after validation of these types have occurred in single-mode studies.

Consequences of Choices

Wolf (2006) argues that in addition to serial and parallel interactivity, we should also examine the *consequences* of choices as another dimension of interactivity. What role does choice consequence play in learning, problem solving, and attitude change? How is choice consequence related to high and low interactivity (serial and parallel)? This is an independent variable that, like serial and parallel interactivity, can be manipulated and controlled to examine the effects on a variety of dependent variables (e.g., problem solving).

Cognitive Load of Domain Knowledge Required for Individual Choices

While cognitive load is very much influenced by individual characteristics, prior knowledge, and expertise, it is also determined by the nature of the content, the interface of the instructional medium, and the type of problem being solved. Suffice it to say that aspects of the different problem types we have addressed will require different amounts of time for processing and solving and that the demands of the interface (the game) must be designed appropriately for these processes. Further, different game ontologies often, but imperfectly, captured by notions of genre, will support this processing time differentially. For example, FPS games often have elements that require continuous attention with little time for reflection. Such elements privilege automaticity and fluency of action–reaction over planning and reflection.

Researchers and designers should look at issues of extraneous, germane, and intrinsic cognitive load (Low, Sweller, and Jin in press) at the choice nexus as a result of both the number and the complexity of choices (parallel interactivity), and at the cumulative effect of choices over time (parallel interactivity). Researchers should also examine the cognitive load that results from the interaction of gameplay type and problem elements to establish ideal frequencies for problem-solving nexuses. Researchers should further examine the role that problem complexity (germane cognitive load) plays in the amount of time needed for problem solving and metacognition. All of the different elements of problem solving may be expected to differentially impact cognitive load as well (e.g., type of cognition, the role of prior domain knowledge, structuredness of the domain) and should be studied first independently and then later for interactions.

CONCLUSION

If serious game designers hope to create games that promote problem solving, they must build on existing problem-solving research and generate new research and design heuristics on the alignment of problem solving and different gameplay types. In this chapter, we used Jonassen's typology of problem types to help analyze the cognitive processes involved in different types of gameplay and, in turn, dissected gameplay that brought the essential characteristics (for problem solving, at any rate) to light. With an understanding of the cognitive, physical, and domain knowledge requirements of each type of gameplay, instructional designers and game developers will have a better idea of what types of gameplay will most appropriately afford given learning goals and objectives. This chapter is not intended to provide a comprehensive set of guidelines for designing instructional games or selecting commercial games for instructional purposes but to promote a more cogent model for what we mean by problem solving in games and to provide a starting point for future research, design, and discussion of games to promote problem solving.

REFERENCES

2K2007). *Bioshock*. [Console game]. USA: 2K.

Anderson, B. (in press). MMOGs in support of learning: Current trends and future uses. In Van Eck, R. (Ed.), *Gaming & cognition: Theories and practice from the learning sciences*. Hershey, PA: IGI Global.

Apperley, T. H. (2006). Genre and game studies: Toward a critical approach to video game genres. *Simulation & Gaming*, *37*(1), 6–23. doi:10.1177/1046878105282278

Armstrong, S., & Warlick, D. (2004). The new literacy. *TechLearning*, *25*(2), 24–31.

Ascaron. (2009). *Sacred 2: Fallen Angel*. [Console game]. USA: Ascaraon.

Äyrämö, S.-M., & Koskimaa, R. (in press). Narrative definitions for game design: A concept-oriented study of nine computer game design books. In Van Eck, R. (Ed.), *Interdisciplinary models and tools for serious games: Emerging concepts and future directions*. Hershey, PA: IGI Global.

Bhaskar, R., & Simon, H. A. (1977). Problem solving in semantically rich domains: An example from engineering thermodynamics. *Cognitive Science*, *1*, 193–215.

Blizzard. (2001). *World of Warcraft.* [PC game]. USA: Blizzard.

Bogost, I. (2007). *Persuasive games: The expressive power of videogames*. Cambridge, MA: MIT Press.

BreakAway. (n.d.). *Pulse: Virtual Clinical Learning Lab for Health Care Training*. [Video game]. USA: BreakAway. *Ltd.*

Broussard, S. R., La Lopa, J. M., & Ross-Davis, A. (2007). Synergistic knowledge development in interdisciplinary teams. *Journal of Natural Resources and Life Sciences Education*, *36*, 129–133.

Capra, F. (1996). *The web of life: A new scientific understanding of living systems*. New York: Doubleday.

Crawford, C. (in press). Interactivity, process, and algorithm. In Van Eck, R. (Ed.), *Interdisciplinary models and tools for serious games: Emerging concepts and future directions*. Hershey, PA: IGI Global.

Dick, W., Carey, L., & Carey, J. O. (2005). *The systematic design of instruction* (6th ed.). Boston: Pearson.

Driscoll, M. P. (2005). *Psychology of learning for instruction* (3rd ed.). Boston: Pearson.

Dynamix. (1993). *The Incredible Machine*. [PC game]. USA: Dynamix.

Entertainment Arts. (2008). *Spore* [PC game]. USA: EA Games.

Flavell, J. H. (1985). *Cognitive development* (2nd ed.). Englewood Cliff, NJ: Prentice Hall.

Frasca, G. (2003). Simulation versus narrative: Introduction to ludology. In Wolf, M. J. P., & Perron, B. (Eds.), *The video game theory reader*. New York: Routledge.

Frensch, P. A., & Funke, J. (1995a). *Complex problem solving: The European perspective*. Mahwah, NJ: Lawrence Erlbaum Associates.

Frensch, P. A., & Funke, J. (1995b). Definitions, traditions, and a general framework for understanding complex problem solving. In Frensch, P. A., & Funke, J. (Eds.), *Complex problem solving: The European perspective* (pp. 3–25). Hillsdale, NJ: Erlbaum.

Gagné, R. M., Wager, W. W., Golas, K. C., & Keller, J. M. (2005). *Principles of instructional design* (5th ed.). Belmont, CA: Wadsworth/Thomson Learning.

Gaydos, M., & Squire, K. (in press). Citizen science: Designing a game for the 21st century. In Van Eck, R. (Ed.), *Interdisciplinary models and tools for serious games: Emerging concepts and future directions*. Hershey, PA: IGI Global.

Gee, J. P. (2007a). Games and learning: Issues, perils and potentials. In Gee, J. P. (Ed.), *Good video games and good learning: Collected essays on video games, learning and literacy* (pp. 129–174). New York: Palgrave/Macmillan.

Gee, J. P. (2007b). Learning and Games. In Salen, K. (Ed.), *The Ecology of games: Connecting youth, games, and learning* (pp. 21–40). Cambridge: MIT Press.

Greenfield, P. M. (in press). Video Games. In Van Eck, R. (Ed.), *Gaming & cognition: Theories and practice from the learning sciences*. Hershey, PA: IGI Global.

Hegarty, M. (1991). Knowledge and processes in mechanical problem solving. In Sternberg, R. J., & Frensch, P. A. (Eds.), *Complex problem solving: Principles and mechanisms* (pp. 253–185). Hillsdale, NJ: Erlbaum.

Holyoak, K. J., & Thagard, P. (1997). The analogical mind. *The American Psychologist*, *52*(1), 35–44. doi:10.1037/0003-066X.52.1.35

Houdé, O., & Tzourio-Mazoyer, N. (2003). Neural foundations of logical and mathematical cognition. *Nature Reviews. Neuroscience*, *4*, 507–514. doi:10.1038/nrn1117

Hung, W. (2006a). The 3C3R model: A conceptual framework for designing problems in PBL. *Interdisciplinary Journal of Problem-based Learning*, *1*(1), 55–77.

Hung, W. (2006b). *Application of 3C3R PBL problem design model: A 9-step design process*. Paper presented at the annual meeting of the American Educational Research Association, April 7–11, San Francisco.

Jonassen, D. H. (1997). Instructional design models for well-structured and ill-structured problem solving learning outcomes. *ETR&D*, *45*(1), 65–94. doi:10.1007/BF02299613

Jonassen, D. H. (2000). Toward a design theory of problem solving. *ETR&D*, *48*(4), 63–85. doi:10.1007/BF02300500

Jonassen, D. H. (2002). Integration of problem solving into instructional design. In R. A. Reiser & J. V. Dempsey (Eds.), Trends and issues in instructional design & technology 107-120). Upper Saddle River, NJ: Merrill Prentice Hall.

Jonassen, D. H. (2003). Designing research-based instruction for story problems. *Educational Psychology Review*, *15*(3), 267–296. doi:10.1023/A:1024648217919

Jonassen, D. H., & Hung, W. (2006). Learning to troubleshoot: A new theory based design architecture. *Educational Psychology Review*, *18*(1), 77–114. doi:10.1007/s10648-006-9001-8

Jonassen, D. H., & Hung, W. (2008). All problems are not equal: Implications for PBL. *Interdisciplinary Journal of Problem-Based Learning*, *2*(2), 6–28.

Kallay, J. (in press). Rethinking genre in computer games: How narrative psychology connects game and story. In Van Eck, R. (Ed.), *Interdisciplinary models and tools for serious games: emerging concepts and future directions*. Hershey, PA: IGI Global.

Kiili, K. (2007). Foundation for problem-based gaming. *British Journal of Educational Technology*, *38*(3), 394–404. doi:10.1111/j.1467-8535.2007.00704.x

Kolodner, J. L. (1997). Educational implications of analogy: A view from case-based reasoning. *The American Psychologist*, *52*(1), 57–66. doi:10.1037/0003-066X.52.1.57

Liedtka, J. M. (1998). Strategic thinking: Can it be taught. *Long Range Planning*, *31*(1), 120–129. doi:10.1016/S0024-6301(97)00098-8

Low, R., Jin, P., & Sweller, J. (in press). Learners' cognitive load when using educational technology. In Van Eck, R. (Ed.), *Gaming & cognition: Theories and practice from the learning sciences*. Hershey, PA: IGI Global.

Midway Games. (2004). *Mortal Kombat* [PC game]. USA: Midway Games.

Milton Bradley. (1943). *Battleship* [Board game]. USA: Milton Bradley.

Mintzberg, H. (1994). *The rise and fall of strategic planning*. New York: The Free Press.

mtvU. (2006). *Darfur Is Dying*. [Online game]. Retrieved March 29, 2009, from http://www.darfurisdying.com

Myers, D. (2003). *The nature of computer games: Play as semiosis*. New York: Peter Lang.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Newsgaming.com. (2003). *September 12th*. [Online game]. Montevideo, Uruguay: Newsgaming.com. Retrieved March 29, 2009, from http://newsgaming.com/newsgames.htm

Nintendo. (1985). *Super Mario Brothers*. [Console game]. Japan: Nintendo

OECD. (2004). *Problem solving for tomorrow's world: First measures of cross-curricular competencies from PISA 2003*. Organization for Economic Co-Operation and Development.

Ossimitz, G. (2000). *Teaching system dynamics and systems thinking in Austria and Germany*. A paper presented at the system dynamics 2000 conference in Bergen, Norway, August 2000. Retrieved February 27, 2004, http://www.uni-klu.ac.at/~gossimit/pap/ossimitz_teaching.pdf

Productions, A. W. E. (2005). *Agatha Christie: And Then There Were None* [Console game]. USA: AWE Productions.

Shaffer, D. W. (2006). *How computer games help children learn*. New York: Palgrave Macmillan. doi:10.1057/9780230601994

Sherlock, L. (in press). Toward a rhetoric of serious game genres. In Van Eck, R. (Ed.), *Interdisciplinary models and tools for serious games: Emerging concepts and future directions*. Hershey, PA: IGI Global.

Sierra On-Line. (1995). *Phantasmagoria* [PC game]. USA: Sierra On-Line.

Smith, P. L., & Ragan, T. J. (2005). *Instructional design* (3rd ed.). San Francisco: Jossey-Bass Education.

Sodaplay. (2007). *Sodaconstructor*. [Online game]. Retrieved from http://sodaplay.com.

Spering, M., Wagener, D., & Funke, J. (2005). Brief report: The role of emotions in complex problem-solving. *Cognition and Emotion*, *19*(8), 1252–1261. doi:10.1080/02699930500304886

Sterman, J. D. (2002). All models are wrong: Reflections on becoming a systems scientist. *System Dynamics Review*, *18*(4), 501–531. doi:10.1002/sdr.261

Strawn, G. (2007). *Change: the constant of modern times.* Keynote presentation for the Instructional Technology Conference, April 2 – 4, Middle Tennessee State University, Murfreesboro, TN.

Sugrue, B. (1995). A theory-based framework for assessing domain-specific problem-solving ability. *Educational Measurement: Issues and Practice*, *14*(3), 32–35. doi:10.1111/j.1745-3992.1995.tb00865.x

Towers of Hanoi. (2002). In *Wikipedia, the free encyclopedia*. Retrieved July 13, 2009 from http://en.wikipedia.org/wiki/Tower_of_Hanoi

Trilobyte. (1992). *The 7th Guest.* [PC game]. USA: Trilobyte.

Valve. (2008). *Left 4 Dead.* [Console game]. USA: Valve.

Van Eck, R. (September, 2006a). Building intelligent learning games. In D. Gibson, C. Aldrich, & M. Prensky (Eds.), Games and simulations in online learning: Research & development frameworks. Hershey, PA: Idea Group.

Van Eck, R. (2006b, March/April). Digital game-based learning: It's not just the digital natives who are restless…. *EDUCAUSE Review*, *41*(2).

Van Eck, R. (2007). Six ideas in search of a discipline. In Shelton, B., & Wiley, D. (Eds.), *The educational design and use of computer simulation games*. Boston, MA: Sense.

Van Eck, R. (2008). COTS in the Classroom: A teachers guide to integrating commercial off-the-shelf (COTS) games. In Ferdig, R. (Ed.), *Handbook of research on effective electronic gaming in education*. Hershey, PA: Idea Group.

Van Eck, R., Hung, W., Bowman, F., & Love, S. (2009). 21st century game design: a model and prototype for promoting scientific problem solving. In *Proceedings of the International Association of Science and Technology for Development's annual Computers and Advanced Technology in Education conference*, November 22 -24, St. Thomas, US Virgin Islands.

Van Merriënboer, J. (1997). *Training complex cognitive skills: A four-component instructional design model for technical training*. Englewood Cliffs, NJ: Educational Technology Publications.

Wolf, M. J. P. (2006, December). Emerging world of anime and manga. *Mechademia*, *1*, 78–85.

Yannuzzi, T. J., & Behrenshausen, B. G. (in press). Serious games for transformative learning: A communication perspective on the radical binarisation of everyday life. In Van Eck, R. (Ed.), *Interdisciplinary models and tools for serious games: Emerging concepts and future directions*. Hershey, PA: IGI Global.

"MUST READS" FOR THIS TOPIC

Apperley, T. H. (2006). Genre and game studies: Toward a critical approach to video game genres. *Simulation & Gaming*, *37*(1), 6–23. doi:10.1177/1046878105282278

Frensch, P. A., & Funke, J. (1995). Definitions, traditions, and a general framework for understanding complex problem solving. In Frensch, P. A., & Funke, J. (Eds.), *Complex problem solving: The European perspective* (pp. 3–25). Hillsdale, NJ: Erlbaum.

Hung, W. (2006). The 3C3R model: A conceptual framework for designing problems in PBL. *Interdisciplinary Journal of Problem-Based Learning*, *1*(1), 55–77.

Hung, W. (2006b). *Application of 3C3R PBL problem design model: A 9-step design process*. Paper presented at the annual meeting of the American Educational Research Association, April 7–11, San Francisco.

Jonassen, D. H. (1997). Instructional design models for well-structured and ill-structured problem solving learning outcomes. *ETR&D*, *45*(1), 65–94. doi:10.1007/BF02299613

Jonassen, D. H. (2000). Toward a design theory of problem solving. *ETR&D*, *48*(4), 63–85. doi:10.1007/BF02300500

Jonassen, D. H., & Hung, W. (2008). All problems are not equal: Implications for PBL. *Interdisciplinary Journal of Problem-Based Learning*, *2*(2), 6–28.

Kallay, J. (in press). Rethinking genre in computer games: How narrative psychology connects game and story. In Van Eck, R. (Ed.), *Interdisciplinary models and tools for serious games: Emerging concepts and future directions*. Hershey, PA: IGI Global.

Kiili, K. (2007). Foundation for problem-based gaming. *British Journal of Educational Technology*, *38*(3), 394–404. doi:10.1111/j.1467-8535.2007.00704.x

Low, R., Jin, P., & Sweller, J. (in press). Learners' cognitive load when using educational technology. In Van Eck, R. (Ed.), *Gaming & cognition: Theories and practice from the learning sciences*. Hershey, PA: IGI Global.

Wolfe, M. J. P. (2006, December). Assessing interactivity in video game design. *Mechademia*, *1*, 78–85.

TOP TEXTS FOR INTERDISCIPLINARY STUDIES OF SERIOUS GAMES

Cassell, J., & Jenkins, H. (Eds.). (2000). *From Barbie to Mortal Kombat: Gender and computer games*. Cambridge, MA: MIT Press.

Gee, J. P. (2003). *What video games have to teach us about learning and literacy*. New York: Palgrave MacMillan.

Koster, R. (2005). *A theory of fun for game design*. Scottsdale, AZ: Paraglyph.

Perron, B. (2009). *Video Game Theory Reader 2*. New York: Routledge.

Raessens, J., & Goldstein, J. (Eds.). (2005). *Handbook of computer game studies*. Cambridge, MA: MIT Press.

Rieber, L. P. (1996). Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games. *Educational Technology Research and Development*, *44*(2), 43–58. doi:10.1007/BF02300540

Salen, K. (2007). *The ecology of games: Connecting youth, games, and learning*. Cambridge, MA: MIT Press.

Shaffer, D. W. (2006). *How computer games help children learn*. New York: Palgrave Macmillan. doi:10.1057/9780230601994

Wardrip-Fruin, N., & Harrigan, P. (2004). *First person: New media as story, performance, game*. Cambridge, MA: MIT Press.

Wolf, M. J. P., & Perron, B. (2003). *Video game theory reader*. New York: Routledge.

ENDNOTES

1. Historical data presented here are generally agreed on by researchers in the U.S. and Europe, despite different perspectives taken in the study of complex problem solving. Information presented here is based on Frensch & Funke, 1995.
2. It is ironic that the Gestaltists believed this, as their view of the importance of experience and the real world might have sooner led to the realization that problems and problem solving were likely to be differentiated by the varied nature of problem solving in different contexts.
3. Nexus is both the singular and plural form
4. It is true that one could conceptualize strategy gameplay as a series of sub-systems amenable to hypothesis testing, but larger systems like those underlying most games that employ strategy gameplay tend to be open systems that are not amenable to such approaches over time.