7-9-2010

A Taxonomy and Framework for Designing Educational Games to Promote Problem Solving

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

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

A Taxonomy and framework for designing educational games to promote problem solving

Richard Van Eck
University of North Dakota

Woei Hung
University of North Dakota

Abstract
Problem solving is often discussed as one of the benefits of games and game-based learning, yet little empirical research exists to support this assertion. It will be critical to establish and validate models of problem solving in games, but this will be difficult if not impossible without a better understanding of problem solving than currently exists in the field of serious games. Problem solving and problem-based learning (PBL) have been studied intensely in both Europe and the United States for more than 75 years. 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 paper, we present an overview of the dimensions upon which different problems vary as well as their associated learning outcomes. We also propose a classification of gameplay (as opposed to game genre) that accounts for the cognitive skills encountered during gameplay, relying in part on previous classification systems, Mark Wolf's concept of grids of interactivity (which we call iGrids), and our own cognitive analysis of gameplay. We then briefly describe eleven different types of problems, the ways in which they differ, and the gameplay types most likely to support them using our gameplay topology. We believe that this approach can guide the design of games intended to promote problem solving and that it points the way toward future research in problem solving and games.

Key Words: Problem Solving, Games, Game Design, Serious Games, Taxonomy, Grids of Interactivity, iGrids

*****

1. Statement of the Problem

Many have argued that games address critical thinking and problem-solving skills.[1][2][3][4][5]Problem solving may well be the most powerful pedagogical benefit of commercial games in general and of game-based learning and serious games specifically.

Unfortunately, while researchers have begun to move the discussion of problem solving beyond descriptive to theoretical, 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[6] as evidence that problem solving must surely be going on during that process. 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 realise. 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 [MMOs]) encourage different gameplay experiences, we need to recognize the different types of problem solving that exist in the world. 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 which can help inform our studies and design of problem solving in games. In this paper, we will attempt to bridge theory and practice by examining the relationships between games, problems, their cognitive processes, and instructional design.

2. Problem Solving

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. It was not until nearly 50 years later that researchers came to believe that a general theory of problem solving was not possible, and that problem solving was very much context and domain dependent.

Nevertheless, some elements of early problem-solving research remain useful for talking about problem solving. For example, it is generally accepted that a problem, as represented in the mind of the problem solving, 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. The goal state is the information and resources that *will* be present when the goal has been met, and the problem solver uses a representation of that goal state when considering how to proceed. A problem, then, can be thought of as an attempt 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 her prior knowledge are collectively referred to as the problem space. 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.

Most recently, Jonassen[7] [8] and Jonassen and Hung [9] [10] have proposed a typology of problems and associated prescriptions for the design of problem-based learning and instruction to promote problem solving in general. If games themselves are examples of problem solving, a closer inspection of this literature to see if and how it can be mapped to the study and design of serious games may yield important findings.

3. Games & Problem Solving

Jim Gee[11]has argued convincingly that all games are situated, complex problem-solving opportunities in which players are immersed in a culture and way of thinking. Others have made the same point, such as Kiili[12], who contended that "a game itself is a big problem that is composed of smaller causally linked problems". To be sure, games are more than just problems to be solved, but it is difficult to conceive of a game that does not incorporate problems to be solved. 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 lie at the heart of games leads us to a wealth of previous research findings to draw upon, 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.

The core of our argument is that problems are highly differentiated by context, purpose, and domain, that different types of gameplay have their own affordances, and that it is necessary to understand problem types and gameplay types in order to align them meaningfully in the design of games to promote problem solving. It is not, however, possible to discuss how problems are differentiated without also discussing some of the aspects by which this differentiation occurs. Structuredness, cognitive components, and domain knowledge are key dimensions along which problems vary. Space does not allow a full accounting these dimensions, and the reader is referred to our work on this elsewhere.[13] Likewise, we rely on an in-depth analysis of gameplay types which we are able only to touch upon here, and the reader is referred to the aforementioned chapter for full accounting of gameplay types and interactivity.

5. Problem Structuredness

Jonassen[14] 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. Because video games run the gamut from highly structured to poorly structured, structuredness becomes one dimension upon which we can categorize both games and problems.

6. Cognitive Processes in Problem Solving

In addition to varying along the dimension of structure, solving different problems also relies on different kinds of cognition. There are six main cognitive processes relevant to problem solving as we discuss it here in our paper: Logical thinking (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), analytic thinking (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), strategic thinking (an integration process of synthesizing and evaluating the analytical results of a given situation and generating a most viable plan with intuition and creativity), analogical reasoning (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")[15], systems thinking (the cognitive reasoning processes that consider complex, dynamic, contextual, and interdependent relationships among constituent parts, and the emerging properties of a system)[16-17], and metacognitive thinking (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). Different problems and different gameplay types differ in their support for these different types of thinking, which makes them important in understanding how gameplay and problem solving can be aligned.

7. Classifying Gameplay Types Using iGrids

The variance of problems along dimensions of structuredness and cognitive processes presents one challenge to the research and development of games for promoting problems solving. Yet games themselves vary greatly as well, as can be seen in classification systems.[18-19] And because no one system is widely accepted and nor are they completely compatible, our task is made even more difficult. 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. While serious game researchers may not agree on game genre classifications, most would agree that interactivity is one of the hallmarks of video games.

> "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."[20]

Wolf[21] calls this a Grid of Interactivity. For semantic reasons, we refer to them as Interactivity Grids, or iGrids. 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, they 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, and we agree. For example, the consequences of individual choices (from trivial to game-changing) and the 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" game seen in Figure 1.

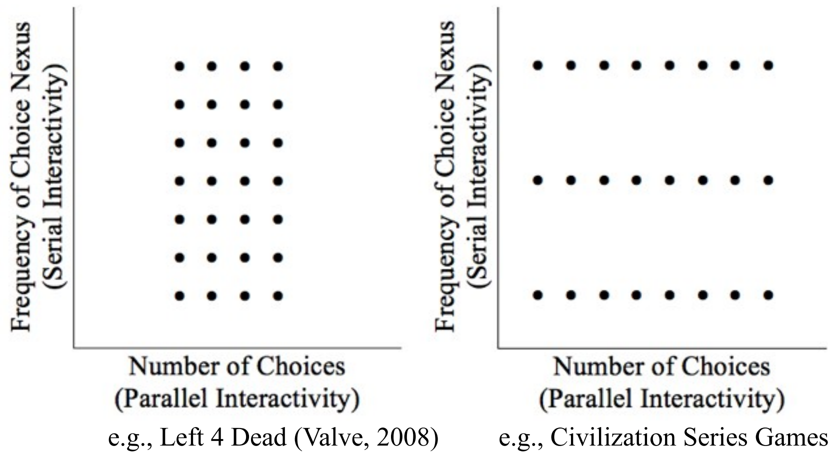

Figure 1. iGrids for two different gameplay types.

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 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." 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.

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.

Likewise, games like those in the Civilization series allow 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.

iGrids, as measures of gameplay type, become useful tools for discussing the differences in games that are likely to impact learning. In our discussion, we will rely on terminology regarding gameplay which we have fully articulated elsewhere.[22] This terminology, while based initially on several existing taxonomies (most notably, Apperley, 2006), differs from common parlance, in some cases significantly. With this in mind, gameplay types in the following discussion are divided into six main categories: Action, Strategy, Simulation, Adventure, Role-Playing, and Puzzles.

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. See the iGrid for Left 4 Dead in figure one for an example of Action gameplay type.

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, coordination among cognition, sensory information processing, and muscular movement control. Strategy and simulation are conflated terms in many taxonomies, but we reserve the term “strategy” for games like the SimCity series (see below). See figure 2.

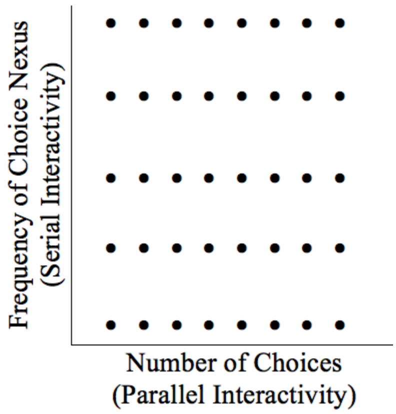

Figure 2. iGrid for simulation games.

As mentioned above, strategy games and simulation games share a blurry boundary because of a lack of consensus on the definitions of these two categories of games. 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. See the iGrid for Civillization in Figure 3.

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. 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, on balance, 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.

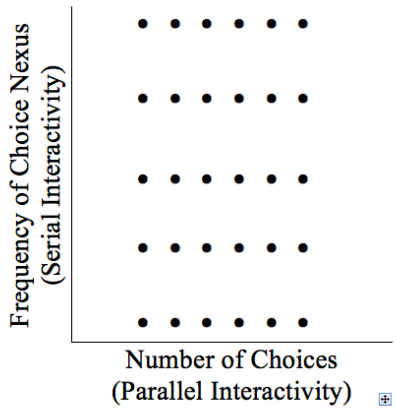

Figure 3. iGrid for typical adventure games.

Adventure and **role-playing games** are sometimes classified as the same type of gameplay. One difference we see between these two gameplay types lies in the player identification with the protagonist. Players may be more likely develop a psychological or emotional attachment to the character they are playing in role-playing games than adventure games. Another reason for separating these two categories is the availability and increasing popularity of massive 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. Grids 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.

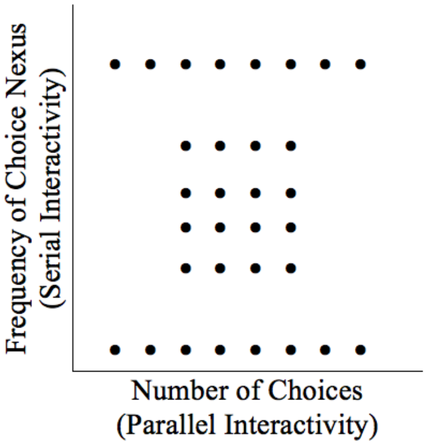

Figure 4. iGrid for role-playing games.

Puzzle games refer to any games that are relatively low- or noncontextualized, with few rules, and which can usually can be solved through logical reasoning. 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

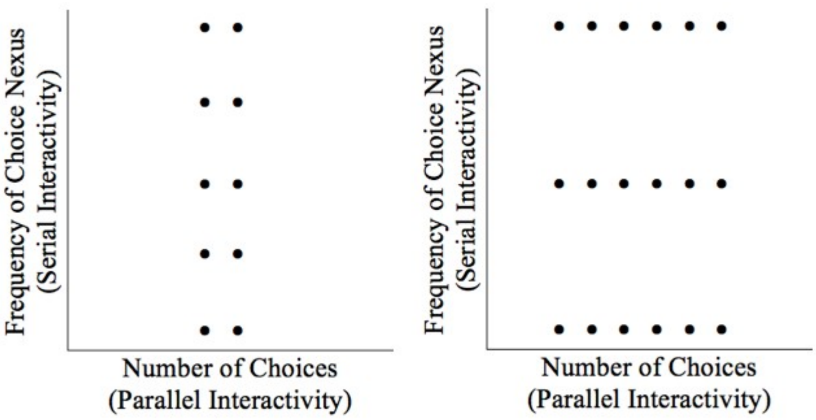

Figure 5. iGrids for the most common forms of puzzle games.

8. Problem Typology

Now that we have outlined our gameplay typology, we turn out attention to problems themselves. Jonassen[23] 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

Space does not allow for a full accounting of all these problem types, iGrids, and examples. The reader is referred to Jonassen's text referenced above and the chapter by Hung & Van Eck, referenced earlier for a full description.

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.

Algorithmic 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. Like 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.

Story problems, sometimes also called word problems, are subject matter-bound, although not necessarily realistic. Solving story problems requires domain-specific declarative knowledge, procedural knowledge, concepts, and principles. Story problems are one of the most well-structured problems types and are more complex than logical and algorithmic problems. Thus story problems can be deemed as precontextualized problems that lie between pure 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 typically requires logical and analytic thinking. 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?"

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. Rule-using problems are often seen as part of more complex types of 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. 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. 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.

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 as simple as troubleshooting a broken light bulb. 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.Troubleshooting typically involves recognizing the symptoms (abnormal behaviors of system), identifying possible causes, testing the hypotheses, and then applying corrective procedures. 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.

Diagnosis–solution 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 man-made systems. 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.

In Jonassen's 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.[24] 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.

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. 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.

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. Therefore, on the continuum of problem structuredness, design problems are at the far end of ill-structured and complex. Engineering design problems, instructional design problems, and interior design 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.

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 comparing to other solutions. The Israeli–Palestinian conflict is a prime example of a dilemma problem. 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, principles are the most vital form of domain knowledge for this type of problem solving. 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, and will also experience some degree of psychological or emotional realization, which could result in attitude change.

9. Implications for Design

Space does not allow a full accounting of every problem type and every gameplay type,[25] but general description and example may suffice to illustrate the logic behind blending problem and game typologies. Knowing about different problem types allows us to see existing games in a new light. For example, dilemma problems can be seen in persuasive games such as Darfur is Dying.[26] But more importantly, knowing how those problem types themselves vary along the dimensions of domain-specific knowledge and required cognitive processes shows us that what superficially may appear to be similar games are in fact quite different in terms of their ability to support problem solving. For example, many might say that September 12 and Darfur is Dying are both dilemma games, when in fact September 12 is too well structured and stripped of context to fully support dilemma problems. Relying on iGrid typologies of gameplay rather than on genre classifications similarly promotes more precise analyses of games and problem solving. By focusing on archetypal gameplay styles, we can see how strategy and roleplaying-games seem best suited for dilemma problems, for example. Further, we are able to apply this reasoning to hybridized games that might at first glance appear to not support different kinds of problem solving. Extending our example of the dilemma problem, the game Bioshock, which many might categorize as adventure-action hybrid, is in fact a hybridization of action, adventure, and strategy. The game Bioshock pits the player against a variety of challenges in an underwater city named “Rapture.” As with Left 4 Dead (Valve, 2008), the player 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, which 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 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. Thus, it is possible to support dilemma problem solving across the full arc of a game which itself is interspersed with other gameplay types, which in their own right may support other kinds of problem solving.

Finally, while our purpose is to outline a mechanism by which problem types, with their associated cognitive requirements, can be matched to different styles of gameplay, the end result also provides significant guidance for design and development of the games themselves. Because the study of problem solving within education and instructional design has been going on for decades, a rich body of research and best practices exists for supporting problem solving. Knowing, for example, that a problem is highly structured implies that less support should be provided for its solution, while ill-structured problems will require addition scaffolding and strategies to avoid cognitive overload. On the other hand, well-structured problems that occur during games with hybridized gameplay styles may indicate the need for more support than otherwise. When the problem solving itself is driving the game design, we may deliberately modify the form and frequency of a different gameplay styles in order to better support the problem. Knowing the kinds of cognitive processes involved also may help guide our selection of in-game tools, story structure, and objectives as well. Figure 6 presents a matrix of problem types by gameplay type (according to our definitions outlined earlier) in terms of the cognitive processes supported. These alignments are derived in similar fashion to those described above, and more fully articulated elsewhere.

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; Action
Story	+	+	+	+	+	+	+							Adventure; Puzzle
Rule-use	+	~	~	+	+	+								Action; Strategy; Roleplaying; Adventure; Puzzle
Decision–making		~	+	+	+	+		+	~	~				Action; Strategy; Role-playing; Simulations; Adventure
Troubleshooting		+	+	~	+	+	+	+	~	~				Simulations
Diagnosis–solution		+	+	+	+	+	+	+	+	+				Simulations; Strategy
Strategic Performance		+	+	+	+	+	+	+	+	+	+	+		Action; Roleplaying; Simulations; Adventure
Case Analysis			+	+	~	+	+		~	+			~	Strategy
Design			+	+	+	+	+	+	+	+				Strategy
Dilemma				+	+	+	~	+	+	+			+	Strategy; Roleplaying

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."

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.

8. 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.

Notes

1 J P Gee, "Games and learning: Issues, perils and potentials." In J. P. Gee (Ed.), Good video games and good learning: Collected essays on video games, learning and literacy (pp. 129-174). New York: Palgrave/Macmillan, 2007.
2 P M Greenfield, "Video Games." In Gaming & cognition: Theories and perspectives from the learning sciences, Richard Van Eck, Editor. Hershey, PA: IGI Global, 2010.
3 R Van Eck, "Six ideas in search of a discipline." In Brett Shelton & David Wiley (Eds.) The educational design and use of computer simulation games. Boston, MA: Sense, 2006.
4 R Van Eck, "Building intelligent learning games." In David Gibson, Clark Aldrich, & Marc Prensky (Eds.) Games and simulations in online learning: Research & development frameworks. Hershey, PA: Idea Group, 2006.
5 T J Yannuzzi, & B G Behrenshausen, "Serious games for transformative learning: A communication perspective on the radical binarisation of everyday life." In Interdisciplinary models and tools for serious games: Emerging concepts and future directions, Richard Van Eck, Editor. Hershey, PA: IGI Global, 2010.
6 Blizzard. "World of Warcraft" [PC game]. USA: Blizzard, 2001.
7 D H Jonassen, "Toward a design theory of problem solving." ETR&D, 48(4), 63-85, 2000.
8 D H Jonassen, "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, 2002.
9 D H Jonassen, & W Hung, W, "Learning to troubleshoot: A new theory based design architecture." Educational Psychology Review, 18(1), 77-114, 2006.
10 D H Jonassen, & W Hung, "All problems are not equal: Implications for PBL." Interdisciplinary Journal of Problem-Based Learning, 2(2), 6-28, 2008.
11 See note 1.
12 K Kiili, "Foundation for problem-based gaming." British Journal of Educational Technology, 38(3), p. 396, 2007.
13 W Hung & R Van Eck, 'Aligning Problem Solving and Gameplay: A Model for Future Research and Design', in R Van Eck (ed), Interdisciplinary approaches to serious games: Emerging concepts, theory, and future directions, IGI Global, 2010, p. 227-263.
14 D H Jonassen, "Instructional design models for well-structured and ill-structured problem solving learning outcomes." ETR&D, 45(1), 65-94, 1997.
15 K J Holyoak, & P Thagard, "The analogical mind." American Psychologist, 52(1), p. 37, 1997.
16 F Capra, "The web of life: A new scientific understanding of living systems." New York: Doubleday, 1996.
17 G Ossimitz, "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, 2000.
18 T H Apperley, "Genre and game studies: Toward a critical approach to video game genres." Simulation & Gaming, 37(1), 6-23, 2006
19 I Bogost, "Persuasive games: The expressive power of videogames." Cambridge, MA: MIT Press, 2007.
20 M J P Wolf, "Emerging world of anime and manga. Mechademia, 1, p. 80, 2006
21 See note 19.
22 See note 12.
23 See note 7.
24 See note 7.
25 See W Hung & R Van Eck, 'Aligning Problem Solving and Gameplay: A Model for Future Research and Design', in R Van Eck (ed), Interdisciplinary approaches to serious games: Emerging concepts, theory, and future directions, IGI Global, 2010, p. 227-263.
26 mtvU, "Darfur Is Dying." [Online game]. Retrieved March 29, 2009, from http://www.darfurisdying.com