Games and Culture
Volume 3 Number 2
April 2008 175-198

10.1177/1555412008314129
http://gac.sagepub.com
hosted at
http://online.sagepub.com

Rounds, Levels, and Waves

The Early Evolution of Gameplay Segmentation

José P. Zagal
Georgia Institute of Technology
Clara Fernández-Vara
Massachusetts Institute of Technology
Michael Mateas
University of California, Santa Cruz

This article explores the early evolution of the structure and management of gameplay in videogames. The authors introduce the notion of gameplay segmentation to capture the role that design elements such as level, boss, and wave play in videogames and identify three modes of segmentation. Temporal segmentation limits, synchronizes, and/or coordinates player activity over time. Spatial segmentation breaks the game's virtual space into sublocations. Challenge segmentation presents the player with a sequence of self-contained challenges. The authors describe each mode, and additional submodes, by analyzing vintage arcade games. The analyses illustrate how these games represent a "primordial soup" in which many current game design conventions were first explored. Their simplicity provides the authors with access to the original "building blocks" of videogames, thus allowing them to develop a rich vocabulary for the discussion.

Keywords: *videogames; arcade; gameplay; game analysis; game ontology*

Gameplay Segmentation in Vintage Arcade Games

This article explores how early videogames found ways to provide players with novel and complex gameplay experiences, particularly how these experiences were structured and managed over time using innovative game design ideas. In abstract terms, we have called this concept *segmentation of gameplay* and have developed it in the context of the Game Ontology Project (GOP) at the Georgia Institute of Technology (Zagal, Mateas, Fernandez-Vara, Hochhalter, & Lichti, 2005).[1] In this article, we introduce and discuss the term and explain its relevance through an analysis of selected classic videogames.

Segmentation of gameplay refers to the manner in which a game is broken down into smaller elements or *chunks* of gameplay.[2] Words such as *level*, *boss*, and *wave*, among many others, refer to ways gameplay is segmented. We explore what these terms bring to the understanding of games through the lens of a particular period, the early arcade years. Referred to as the "golden age of videogames," the 1970s and early 1980s witnessed an explosion of games and game styles, defining many videogame genres still with us today (Kent, 2001, p. 123; Sellers, 2001).

Classic arcade videogames are ideal objects of analysis for unpacking different varieties of segmentation because they introduced what are now canonical varieties of segmentation while also displaying design innovation by blurring and mixing segmentation styles. For instance, Donkey Kong, the canonical platformer, demonstrates a clear notion of *level*, enriched with other types of segmentation, as we explore below (Nintendo, 1981).

We selected the following games for analysis: Asteroids (1979), Battlezone (1980), Donkey Kong (1981), Mappy (1983), Marble Madness (1984), Moon Patrol (1982), Phoenix (1980), Robotron: 2084 (1982), and The Amazing Maze (1976). These games provide clear examples of different forms of segmentation while also displaying design innovations that blur the edges between categories. Each game provides a strong example of some segmentation form and often a weak example of others. This last point is important because we do not presume that our typifications of gameplay segmentation are in any way decisive or complete. Certainly, new gaming hardware, input and output devices, and novel game designs introduce new ways of segmenting gameplay and force us to reexamine the ones proposed here.

Game Ontology Project (GOP)

The ensuing analysis uses concepts and terminology developed as part of GOP (Zagal et al., 2005). GOP offers a framework for describing, analyzing, and studying games by defining a hierarchy of concepts abstracted from an analysis of many specific games. The project borrows concepts and methods from prototype theory and grounded theory to achieve a framework that is continually evolving with each new game analysis or particular research question (Glaser & Strauss, 1967; Lakoff, 1987). The term *ontology* is borrowed from computer science rather than used in the philosophical sense. It refers to the identification and (oftentimes formal) description of entities within a domain. The elements are usually derived from common game terminology (e.g., level and boss), then refined by both abstracting more general concepts and identifying more precise or specific ones. An ontology in this sense is different from a game taxonomy in that, rather than organizing games by their characteristics or elements, it is the elements themselves that are organized.

GOP is distinct from design rules and design patterns approaches that offer imperative advice to designers (Fabricatore, Nussbaum, & Rosas, 2002; Falstein, 2004). It intends not to describe rules for creating good games but rather to identify the abstract commonalities and differences in design elements across a wide range of concrete examples. The ontological approach is also distinct from genre analyses and related attempts to answer the question "What is a game?" Rather than develop definitions to distinguish between games and nongames or among their different types, it focuses on analyzing design elements that cut across a wide range of games. Its goal is not to classify games according to their characteristics and/or mechanics (Lundgren & Björk, 2003) but to describe the design space of games.[3] The ontology abstracts away the representational details of games. Issues of setting (e.g., medieval castle, spaceship), genre (e.g., horror, sci-fi), and leveraging of representations from other media (e.g., player's knowledge of the *Star Wars* universe) are bracketed by our analyses. Because its goal is to characterize the game design space, such bracketing is necessary to achieve broad coverage without having to characterize abstract notions of setting and genre. A deep reading of any particular game may require an analysis of its sociohistorical context, conventions, and types of representations. The ontology helps position the more formal or structural elements of the game within the game design space, which is not incompatible with incorporating our terminology with critical analyses or sociological studies of videogames.

Each ontology entry consists of a title or name, a description of the element, a number of strong and weak examples of games that embody the element, a parent element, potentially one or more child elements, and potentially one or more part elements (elements related by the part-of relation). The examples describe how the element is instantiated in specific games. We include both strong and weak examples, with the latter describing border cases of games that partially reify the element. The parent–child relationship captures the notion of subtype (subset); child elements are more specific or specialized concepts than are parent elements. Finally, the part-of relation captures the notion of compound elements that are constructed out of other elements (parts).

This article uses the sections of the ontology that refer to segmentation of gameplay, whose terms are expanded below. Because a description of the entire ontology, currently consisting of more than 190 elements, is beyond the scope of this article, we refer interested readers to the GOP Web site, which is an end-user-editable wiki where the most recent work is available for reading and participation.[4]

Games and Segmentation

Playing a game takes place over time. Different game designs regulate gameplay time in different ways. This process of managing and regulating the development of gameplay experience through the design of a game is what we call segmentation of gameplay. To borrow from non-videogames, consider a game of football (soccer), in which the match is divided into 45-min halves. Splitting the total duration of the game in half is a way of segmenting gameplay. Another segmentation form involves coordinating players' actions. For example, many games force players to take turns, alternating periods of participation and observation. In these cases, gameplay is segmented by forcing the players to coordinate their actions so that individual players cannot simultaneously affect the state of the game. A game in which players take turns is different from one in which everyone participates simultaneously. This is another example of what we mean by segmentation of gameplay.

Segmentation of gameplay describes how a game is broken down into smaller or shorter units of gameplay. As seen in the examples above, it is not new or particular to videogames. However, videogames have greatly extended the varieties of segmentation, making the concept richer and more sophisticated. Specifically, videogames have introduced new vocabulary referring to gameplay segmentation. For instance, words such as *level*, *boss*, and *wave* refer to particular ways of segmenting gameplay that have become essential in describing and analyzing videogames. These words, however, are also used informally, so that novel forms of segmentation are sometimes conflated under these general terms.

Considering vintage arcade games in particular, we identify three general modes in which gameplay is usually segmented. These modes relate to time, space, and challenge (see Table 1). The first mode is temporal segmentation, which means limiting, synchronizing, and/or coordinating player activity over time. For example, games in which players take turns segment gameplay by defining the order and manner in which players may participate and implying that a player cannot play during someone else's turn. Another variety of temporal segmentation stipulates fixed time periods that define the duration of the game. Many sports games favor segmentation of this type by enforcing real-world time limits.

The second form of segmentation is spatial—the game's virtual space is broken down into sublocations. Some words used to describe particular forms of spatial segmentation include *levels*, *maps*, and even *worlds*.

The third form is challenge. It occurs when subunits are presented as self-contained challenges to be negotiated by the player, with successive challenges implying greater difficulty. Consider a game in which the player solves a series of puzzles. Solving a particular puzzle allows the player to attempt the next one, and so on.

Most games—contemporary videogames in particular—include multiple forms of segmentation that are interrelated or even co-occur. Although it is rare for a game to exhibit a single form of segmentation, one form is often more salient because of its greater impact on gameplay. Donkey Kong has spatial segmentation in the form of levels and temporal segmentation because the player has a fixed time limit to complete each level. Changing either form of segmentation affects gameplay, but eliminating the time limit has less impact than changing the game so that it no longer has levels. We argue that Donkey Kong without levels ceases to be Donkey Kong, whereas without a time limit it remains similar, although easier. Other games use several modes of segmentation in different occasions—a game may be temporally segmented in one section, whereas the rest could be spatially segmented. Sometimes,

Table 1
General Types of Segmentation of Gameplay

1	Temporal
2	Spatial
3	Challenge

when analyzing a game, it may be useful to choose a particular type of segmentation as the basis for further analysis. For example, when exploring the pacing and tension as perceived by the player, it might be productive to focus on temporal segmentation.

We examine these issues by providing detailed descriptions of each form of segmentation, including some subtypes we identify. The descriptions are supplemented with discussions of particular games. The object of each discussion is to highlight a particular form (or subform) of segmentation, recognize other forms that may be present, and describe how the overall gameplay experience is mediated by the design decisions made regarding segmentation of gameplay. We conclude by recapitulating on the different forms of segmentation in the context of in-depth analyses of Battlezone and Robotron: 2084.

Temporal Segmentation

Perhaps the most traditional form of segmentation is temporal. In most nonelectronic games, temporal segmentation is the only way of breaking down gameplay. This type of segmentation takes two distinct forms—the first regulates who plays when, the second specifies time limits or periods of gameplay. The first is about *coordination*, whereas the second uses time as a *resource*. Table 2 shows the main forms of temporal segmentation and some specific cases that are described in the following subsections.

Temporal Coordination

Temporal segmentation by coordination refers to how a game regulates the actions of a player in a game and how these actions occur over time. The most traditional form of coordination is players taking turns. In many games, at any given moment, only one player can perform actions in the game, whereas the other players wait for their turn to play; tic-tac-toe is a classic example of this type of coordination.[5] In games that use rounds,[6] players independently decide their actions and then resolve the consequences of these actions simultaneously. Once the actions have been resolved, a new round begins and the players again decide their actions. Rock-paper-scissors is a classic example of this type of coordination.[7] This form of segmentation regulates players' actions over time but does not constrain the length of time their moves can take. A

Table 2
Temporal Segmentation and Specific Forms

Temporal Segmentation	
Temporal coordination	Turn taking, rounds, interleaved games
Temporal resource	Time limit, subdivisions of total available time

game of chess can take from a few minutes to as long as the players think reasonable to meditate their next move. The duration of the game is thus at the discretion of the players, not the game (Salen & Zimmerman, 2004, p. 129).

Vintage arcade games introduced a new form of coordination, which we call *interleaved games*. This form of alternating gameplay, informally called "hotseat multiplayer" because players often take turns sitting in the game's seat (Wikipedia, 2005), consists of taking turns over two or more independent game sessions (Björk & Holopainen, 2005; Zagal, Nussbaum, & Rosas, 2000). This occurs in games in which the first player plays until he or she loses a life, at which point it is the second player's turn. Both players alternate until they both run out of lives. Everyone is playing on the same hardware, and there is usually some visual representation acknowledging the other player's game and its status. For example, the top-left corner of the screen may show the current score of Player 1, whereas the top-right corner displays Player 2's score. This new form of coordination is afforded by technology; the computer is well suited to simultaneously manage and maintain multiple game states.

Temporal Resource

A game can establish its total duration and any subperiods and their length. When duration is explicitly regulated by a game, time is being treated as a resource. For instance, a game might last 10 minutes played in two halves of 5 minutes each. Examples of early arcade games that employed this form of segmentation include Atari Football, Atari Basketball, and Atari Soccer (Atari, 1979c, 1979d, 1979e).

A temporal subdivision does not have to be constant. Time can be considered a resource in the sense that certain actions, rules, or events may modify the game's duration. For example, in Atari Basketball, inserting a quarter in the machine granted the player 1 minute of playtime.[8] Segmentation via temporal resource is common in sports games, such as basketball or football. In football, a game lasts 90 minutes, played in two halves of 45 minutes. The referee, however, may grant extensions on the duration. Another form of temporal segmentation occurs when the player is allotted a specific amount of time to complete a certain task, fulfill a goal, or simply do the best he or she can. Also, many games use a different visual representation for what is essentially a time limit. Donkey Kong has a numerical counter that usually starts at 5,000 and then decreases in increments of 100.

Temporal resource is not necessarily equivalent to a time-dependent goal. For example, in racing games, the winning condition typically requires reaching a certain location as quickly as possible. In these cases, it is important to have a "low time," and there is an implicit "maximum time" (the slowest opponent will eventually finish). This time, however, is neither constant nor explicitly communicated to the player; therefore, we cannot argue that temporal resource segments gameplay, unless the win condition involves getting to a certain location in a certain amount of time.

Analysis: Marble Madness (1984)

Marble Madness can be a one- or two-player game; two players may play at the same time, competing to reach the goal first (Atari, 1984, 1985). The goal of the game is to reach the end as soon as possible, using a trackball controller to maneuver a marble down a total of six unrelated isometric courses. There are numerous obstacles and unfriendly creatures that hinder the player's progress. A timer ticks away; when it reaches "00," the game ends. Falling off the edge, or from too high, causes the marble to reappear in a previous location (further "up" the course).

In Marble Madness, space is segmented in six levels. Each level is larger than the screen; as the player's marble rolls down, the screen scrolls to reveal more of the level. The most salient form of gameplay segmentation in Marble Madness is via temporal resource—the player is assigned a fixed amount of time to complete each level. Of interest, the temporal resource segmentation allows the player to begin a level adding leftover time if he or she performed well on the previous one (though this is not true of all levels). If the player loses the marble over the edge, he or she loses time because the timer is not paused while the new marble is repositioned. The lure of reaching the end goal with seconds to spare, combined with the agony of losing time after a precipitate maneuver, results in a tense gameplay experience.

Marble Madness also offers a two-player mode. However, the activities of the different players are not regulated by temporal coordination because both players participate simultaneously. In fact, players decide whether they want to compete or not; the game does not enforce or suggest competition, it merely allows it.

Spatial Segmentation

Computer games are spatial—most games convey a notion of place to the player, whose participation in the game is within the boundaries of a virtual world (Murray, 1997). This world, although different from the physical world we inhabit, is still subject to its own rules and constraints; it has an internal consistency that allows the player to recognize it as a world (Juul, 2004).

Because of technical limitations, early arcade games such as Computer Space and Pong faced significant difficulties in representing something on the screen that could even be identified as a space (Atari, 1972; Nutting, 1971). Early arcade games also tended to present the entire gameworld in one screen (Fernandez-Vara, Zagal, & Mateas, 2007; Wolf, 2001); nothing happened off screen, where the player could not see. Games that unfolded over a series of different, discrete screens, such as Gorf and Donkey Kong, soon appeared (Midway, 1981; Nintendo, 1981). In many cases, there was an explicit continuity of space that was communicated to the player; he or she had to navigate the gameworld in discrete segments. These multiple screen games are early examples of what we call spatial segmentation.

Spatial segmentation results from the division of the gameworld into different spaces that also partition gameplay. In these cases, the gameworld is presented not as a continuous whole but rather as distinct subspaces that are navigated separately and that may even have their own special rules. Each subspace may be larger than what can be displayed on the screen; what matters is whether they are distinguished as separate locations and whether there are gameplay restrictions or differences between each location. In referencing a strong sense of spatial segmentation, it is important that the player perceives he or she is participating in a virtual space larger than its onscreen representation and that this space is traversed in parts. A series of disconnected screens that bear no sense of relationship could be considered an example of spatial segmentation, albeit a weak one. An early arcade example of a game in which the gameworld is larger than the screen, yet it is perceived as a non-segmented whole, is Defender (Williams, 1980). In this case, the screen scrolls smoothly, allowing the player to see the rest of the gameworld.

Spatial segmentation is common in videogames because computers can efficiently and cheaply store the data needed to represent or generate expansive virtual worlds (Murray, 1997). Storing additional information, such as specific rules for different spatial segments, is relatively easy. In fact, virtual spaces are a habitat for imagination where physics can be redefined or even altogether invented (Novak, 1991). On the other hand, the spatial aspects of non-videogames are usually constrained by the limitations of actual physical space. Consider the problems of building a board game with 40 square boards, each having an area of 1 square meter, or a basketball court 20 kilometers in length!

Level

A *level* is a recognizable subspace of the gameworld. As spatial forms of segmentation of gameplay, levels are "discrete virtual locations containing tasks that must be accomplished before players can advance" (Laidlaw, 1996). However, the word *level* is problematic because it is also often used to refer to the degree of difficulty a player encounters. In this last sense, the higher the level, the more difficult the game. In games such as the pencil-and-paper role-playing game Dungeons & Dragons, the word *level* has multiple uses within the same game (TSR, 1974).[9] In the case of our ontology, references to the difficulty are considered under challenge segmentation (discussed below).

What helps distinguish a level from other forms of spatial segmentation is the discontinuity in gameplay and in space between one level and another; the more evident the discontinuity, the greater the notion of level. Often, the discontinuity is highlighted through the use of a transitional screen or cut scene. However, that discontinuity cannot come at the expense of losing the necessary spatial relationship between the spatial segments. The tension in discontinuity and relationship between spatial segments can be resolved by exploring a series of questions. For instance, "Do enemies from one area follow the player to the next?" "If the player fires a shot and it goes off-screen, can it kill an enemy in the next area?" "Are any variables, such as health, enemy positions, and so on, reset when the player moves from one area to another?" "Could the player draw a map of the areas visited and where they are located in relation to each other?" Affirmative answers to these questions strengthen the notion of levels in a game. Another distinguishing feature of levels is differentiation, because they represent different locations. Games such as Pac-Man, we argue, do not have spatial levels because the maze is always the same (Namco, 1980a).[10]

As parts of a gameworld, levels are often grouped together by representational themes, (e.g., "ice" or "lava") or by particular aspects of gameplay (e.g., "flying" or "driving").

Analysis: Donkey Kong (1981)

Donkey Kong allows players to control a character called Jumpman (later renamed Mario), who tries to recover his beloved from the clutches of a gorilla named Donkey Kong (Kent, 2001). Jumpman can move left and right, jump, and climb up and down ladders.

Donkey Kong has multiple, coexisting forms of gameplay segmentation. In two-player mode, the game allows interleaved games. When the two-player mode is selected, players' scores are represented in the top-left and top-right areas of the screen. Donkey Kong has four distinct screens—levels in this case—that indicate spatial segmentation. Each screen has enough visual detail to be recognized as a representation of a physical, although fantastical, place. There is also an early example of a boss challenge, a form of challenge segmentation (more on this later). The game also employs temporal segmentation through an indicator labeled "Bonus"—when this indicator reaches zero, the player loses a life and must restart the level.

This game is an interesting example of spatial segmentation because the game has four distinct screens, referred to as Girders, Pie Factory, Elevator, and Rivet levels (Butler, 2003a), and also because the player's progress involves negotiating a series of specific level configurations (see Table 3). After pressing the "start" button, the player is taunted with a message, "How high can you get?" The game then begins on the girder level, and an indicator in the corner displays "L = 01." Clearing this level takes the player to the rivet level, after which a new sequence of levels begins. At this time the taunting message reappears, and the label "L = 01" changes to "L = 02."

Table 3
Level Sequence in Donkey Kong

Label of Sequence	Level Sequence
L01	Girders, rivet
L02	Girders, elevator, rivet
L03	Girders, pie factory, elevator, rivet
L04	Girders, pie factory, girders, elevator, rivet
L05, L06, etc.	Girders, pie factory, girders, elevator, girders, rivet

Table 3 describes the five different configurations in total, in which each successive sequence introduces a new level and revisits previous ones in a more challenging way (e.g., by adding more opponents). Each sequence always begins with the girder level and ends with the rivet level. The sequences of levels exemplify a form of challenge segmentation because the levels become harder and it takes longer to get to the top to defeat Kong.

Completing each sequence provides the player with a sense of progression. Each level within a sequence is higher (labeled in meters) than the previous, the rivet level being the one at the highest altitude, reached after climbing up through the previous ones. As the player reaches the top of each level, except the rivet one, Donkey Kong escapes with the girl, climbing up a ladder to the next level. Only on reaching the final rivet level can Jumpman defeat Kong—removing all the rivets that support the structure of the level will send Kong crashing down on his head. The rivet level is a boss challenge (described below) because to defeat the special antagonist (boss), the player needs strategies different from those used in previous levels. In Donkey Kong, this means removing rivets instead of trying to reach the girl.

Donkey Kong's screens are canonical levels. Each is higher than the last, and there is a sense of spatial relationship between them. Together they are part of a skyscraper. Each sequence of levels is thus implicitly a different skyscraper, which Jumpman must climb in pursuit of the ape.[11] The rivet level is always the top of the skyscraper from which Kong falls defeated. At the beginning of each level, the timer is reset, and Jumpman must begin again from the bottom. Though levels are distinct, they are also related through particular gameplay elements. Skills developed by using the hammer on the girder level are used in subsequent levels. This satisfies the apparently contradictory requirements of discontinuity and relationship that define a level.

Spatial Checkpoint

Spatial checkpoints divide a space into sublocations that follow one another continuously (in contrast to the discreteness witnessed in levels),[12] being the boundaries between these sublocations. To be considered a form of spatial segmentation, there must be differences that affect gameplay when the player moves from one sublocation to the next, otherwise there would not be any segmentation. Some examples of affecting gameplay in this way include prohibiting the player from moving back to a previous sublocation once a spatial checkpoint has been reached, resetting or modifying game variables (e.g., the amount of time the player has to reach the next sublocation), awarding bonus points, or resuming the game from the spatial checkpoint last cleared before losing a life. The player must also realize he or she has moved from one sublocation to another to consider a spatial subdivision a checkpoint. Games with gameworlds larger than the screen and that scroll continuously when the player moves do not usually feature spatial checkpoints unless some of the above conditions are met.

Analysis: Moon Patrol (1982)

In Moon Patrol, the player controls a moon buggy that can jump, fire missiles to the front and into the air, and accelerate and decelerate its movement (Irem, 1982). Moon Patrol allows interleaved playing for two players, at the time sporting the uncommon feature of allowing players to continue from the last checkpoint if they insert another coin after losing all lives (Sellers, 2001).

Moon Patrol has multiple hierarchical forms of spatial segmentation. At the highest level, the player completes a series of "courses" or levels, in which the player drives the moon buggy from the moon base to a point marked "Z." The first level is the beginner course, and the later ones are called champion courses. The champion course is distinguishable from the beginner one by its greater difficulty and the color of the buggy.

As illustrated in Figure 1, each course is subdivided into five sections labeled "E," "J," "O," "T," and "Z." As the moon buggy traverses the course, the red bar "fills up," (darker grey in the figure) indicating the overall progress the player has made. When the buggy reaches each letter marked above the bar, the player is informed of the time taken to travel from the previous letter, and whether that time is low enough to score extra points. Each newly entered section is distinguished from the previous by a changing background.

The course subdivisions are a weak example of a level. They can be considered levels because the next section is a different place (represented by the new background) and the timer used to calculate bonus points is reset. Also, the player has a new goal: reach the next section in the shortest time possible. Finally, there is a distinct pause in the gameplay when the player is informed of how well he or she did and whether or not any bonus points are scored. However, the case for calling them levels is weakened because each section continues from the next.

Moon Patrol has a strong example of spatial checkpoints because each level is also subdivided. In Figure 2, the player is in level (E-J) and is about to reach point "G." To get to the spatial checkpoint "J," he or she will have to pass letters "H" and "I" (Schultz, 2002b). These spatial checkpoints are important for two reasons: Each

Figure 1
Course Subdivisions in Moon Patrol

Figure 2
Reaching Point "G"

one represents a different challenge, in terms of enemies and obstacles, and they mark an additional checkpoint in the player's progress. If the player loses a life between points "D" and "E," he or she does not restart the course but automatically resumes playing from "D"—this is another example of how this game uses spatial checkpoints. We do not consider these subdivisions as levels because the player moves continuously from one to the next with no pause in gameplay; undefeated enemies from the previous subsection can also follow the player into the new subsection.

Challenge Segmentation

Segmentation by challenge is difficult to define because most games, regardless of how their gameplay is segmented, try to increase the difficulty and challenge as the game develops. Thus, it is harder to recognize segmentation based on providing different challenges. In essence, segmenting by challenge is to have the player resolve a series of discrete self-contained challenging situations, their most salient feature being that they are perceived by the player as separate tests or trials. Specific forms of challenge segmentation include puzzles, boss challenges, and waves.

Increasing challenge is often accomplished by making enemy characters more dangerous, by making the player character more vulnerable, or by enforcing shorter time limits. Other ways include modifying the rules and introducing new ones, such as allowing the player to perform new actions. For example, Pac-Man limits the duration of the effect of the power pills, whereas Robotron: 2084 introduces new enemies that require different strategies to defeat (VidKidz, 1982).

Wave

A wave is a kind of challenge segmentation generally observed in games that require quick reflexes and good hand–eye coordination. A wave is a group of usually similar enemy entities that must be avoided or destroyed as they approach the player. There is usually a pause, or respite, between waves. One of the defining aspects of the wave is player inaction results in the end of the game (game over). Segmentation with waves is primarily used to increase the tension of the gameplay. Classic examples of segmentation using waves include Space Invaders and Missile Command (Atari, 1980c; Taito, 1978). The former was described by S. Iwata as a game in which "a little neglect may breed great tension" (Kent, 2001, p. 118),[13] whereas in the latter "you were only postponing the inevitable. This war was hell, and it was never long before you died in a blaze of glory" (Sellers, 2001).

Analysis: Asteroids (1979)

In Asteroids, the player controls a triangular ship that fires in the direction it faces (Atari, 1979a). Pressing the "left rotate" or "right rotate" buttons causes the ship to rotate in the respective direction. In addition, the player can press the "thrust" button to move forward or the "hyperspace" button to disappear and reappear in a random location on the screen. The objective of the game is to shoot and destroy as many asteroids as possible before all spaceships are destroyed. When the player has shot all asteroids, a new set of large asteroids appear. At the beginning of the game, 4 large asteroids appear; the next cycle starts with 6 large asteroids, then 8, and thereafter 10, increasing the challenge (Atari, 1979b).

Asteroids' main form of segmentation is challenge segmentation in waves. The player faces successive waves with more and more asteroids. Destroying all asteroids in a wave gives the player a brief respite before a new batch of asteroids appears. The wave sensation is enhanced by the starting position of the player's ship, in the center of the screen, surrounded by drifting asteroids. Successive waves affirm the intention of increasing the difficulty level. Of interest, there is no indication on the game's interface of which wave is being played. As a deviation from the prototypical wave, there is no unavoidable death in Asteroids. The player has no urgency in destroying the last little bit of space rock to trigger the next wave. In fact, it is in the player's best interest not to because there is a reward of 1,000 points for destroying a flying saucer that periodically flies by. This strategy, referred to as "lurking," allows players to set incredible top scores (Kent, 2001).

Puzzle

Perhaps the clearest form of challenge segmentation is the puzzle. However, it is a rare form of segmentation in early arcade games. The static nature of puzzles, coupled with a slower pace and emphasis on problem solving rather than reflexes, probably limited the commercial potential of puzzle games in the arcade.

Games that use puzzle segmentation present a series of puzzles that must be solved before the next is available. The progression need not be strictly linear. A puzzle is a challenge in which there is no active agent against which the player is competing (i.e., a puzzle is static; Crawford, 2003). Typically, it has at most a few correct solutions, requiring problem-solving skills rather than good hand–eye coordination and quick reflexes (Rollings & Adams, 2003) This form of segmentation is common in adventure games. If the player must rearrange crates to leave a room, there is a clear instance of puzzle segmentation. In adventure games, it is usual for the game to be organized as a series of puzzles whose solutions allow the player to progress through the gameworld.

Analysis: The Amazing Maze Game (1976)

In Amazing Maze Game, the player must find the way out of a single screen maze before his or her opponent, either the computer or another player (Bally/Midway, 1976). When the computer controls the opponent, it moves more slowly than the player but never makes a mistake.

Amazing Maze Game is one of the few early arcade games in which we see challenge segmentation by puzzles.[14] It is arguably a weak form of puzzle segmentation because there is an active agent against whom the player is competing—he or she has to reach the exit before the computer-controlled opponent does. However, the opponent cannot hinder the player's progress in any way (and vice versa), so the computer opponent functions as a timer. The opponent's progress through the maze is like a lit fuse—it will reach the end regardless, without speeding up or slowing down. The time it takes it to reach the exit is how long the player has to solve the maze. In this sense, the game also uses a weak form of temporal resource segmentation using time as a resource.

When Amazing Maze Game is played by two players, it also uses temporal segmentation, with a total time limit of 6 minutes. In the one-player game, the player has the chance to beat the computer-controlled opponent by being the fastest to solve three mazes. If the player succeeds, he or she can continue to play, solving new mazes against an ever-faster opponent until he or she is eventually defeated. The player, regardless of his or her performance, always has a chance to play three mazes.

Boss Challenge

A boss challenge is a capstone of gameplay activity. It is a milestone of a player's progress in a game because it embodies a particularly difficult challenge that must be overcome to continue or finish the game. Boss challenges (games may have more than one) are unique because they present a trial that does not spring from the natural progression of previous challenges. For example, the difficulty may increase significantly, the player might have to resort to new tactics, or there might be additional conditions or restrictions on the player's actions. As the name implies, the player may also face the boss, an opponent substantially different from previous ones in terms of size, power, and vulnerability. Early arcade games such as Phoenix, Gorf, and Time Pilot feature boss challenges (Amstar, 1980; Konami, 1982; Midway, 1981). The first wave in Time Pilot is an example of this—after the required 56 enemy planes have been shot down, the player destroys a blimp before "time warping" to the next wave (Schultz, 2003; Sellers, 2001).[15]

Often, the boss challenge is present in the context of some other form of segmentation. For example, the last level of a series may contain the "boss." This level would be considered a boss challenge; succeeding here is a climax to the player's activities in previous levels. In Donkey Kong, Kong appears in every level but conveniently escapes until the last level—the boss challenge. Defeating Kong requires a different tactic, releasing all the bolts holding up the structure, rather than simply climbing up.

Analysis: Phoenix (1980)

In Phoenix, the player controls a spaceship that maneuvers left and right, fires rockets, and attempts to evade missiles and the birds dropping them as they dive at the spaceship. In addition, the player can activate a "force field" as a means of protection from the missiles and birds. The birds, called phoenixes, are destroyed when they collide with the force field or when they are hit by a rocket (Centuri, 1980).

The player faces a series of waves. The first two waves consist of 16 phoenix fighters. The third and fourth waves consist of 8 phoenixes larger than the fighters though harder to destroy (Butler, 2003c). The birds in the third and fourth waves begin as small eggs that get larger as they cross the screen until they turn into full-sized birds.

Phoenix's main form of segmentation is waves. The player faces a series of waves with opponents that become more dangerous and harder to defeat. The fifth wave is the most interesting. After clearing the fourth wave, the player faces a boss challenge: the attack of the "space fortress." While avoiding waves of phoenix fighters, the player blasts through a protective barrier shielding a space creature inside the space fortress (see Figure 3; Centuri, 1980). In this case, defeating waves of phoenix fighters is not enough because more appear. The player must also act quickly to defeat the boss because the space fortress slowly inches its way down to the bottom of the screen. A player delaying the destruction of the space fortress until the last second is rewarded extra points.

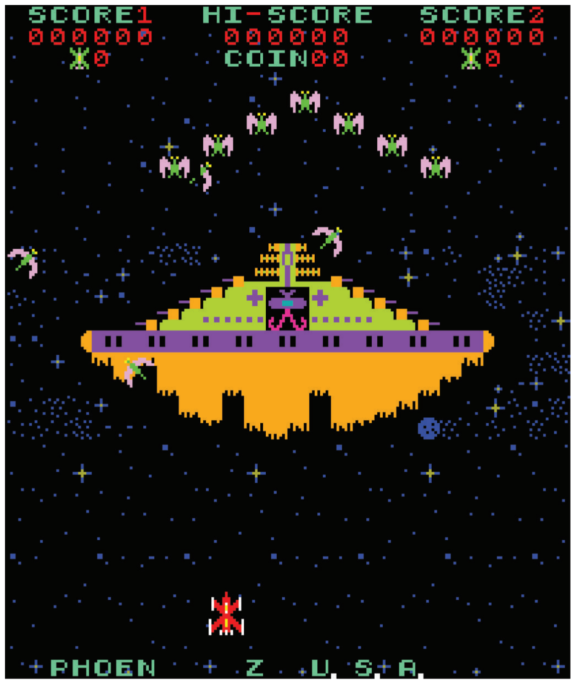
Bonus Stage

The purpose of a bonus stage is to present the player with an opportunity to earn rewards without the risk of losing the game. Although the use of in-game resources in bonus stages differs from game to game, the uniting factor is that the player is momentarily liberated from the potential of losing. The challenge perceived by the player changes drastically, becoming a sort of antichallenge: You cannot lose the game while playing a bonus stage. Nevertheless, bonus stages are challenging if you want to score maximum points. Early arcade games with bonus stages include Mario Bros. and Mappy (Namco, 1983; Nintendo, 1983).

A bonus stage is different from a period of invulnerability within a normal part of a game. Whenever Pac-Man "eats" a power pill, a reversal occurs—for a limited time, the player can defeat the ghosts. If the player catches them before the effect wears out, he or she scores extra points. However, the objectives of the game remain the same. The player is still in the same maze, under the same victory conditions: eat all the pellets. There is no sense of reward because eating the power pills is a requirement that cannot be ignored.

Rally-X has been acknowledged as the first game with a "bonus round" (DeMaria & Wilson, 2004; Namco, 1980b; Sellers, 2001). Starting with the third level and every four levels after that, the player gets to play a special level in which the enemy cars are motionless until the player runs out of fuel. It is a weak example because the player can still lose lives by crashing into rocks or the motionless cars (Schultz, 2002c). Another weak example of a bonus stage occurs in Joust (Williams, 1982a). Joust has "egg waves" that occur every five waves (Janiec, 2004). Instead of enemy riders, the player is presented with 12 harmless eggs distributed among the different platforms (Butler, 2003b). The eggs hatch into enemy riders after a time, but until then they are "basically free points" (Janiec, 2004). Both Rally-X and Joust have weak example of bonus stage because, although you can lose the game, the rounds are a break from traditional gameplay and begin with an extensive amount of time during which the player is effectively invulnerable.

Figure 3
Boss Level in Phoenix

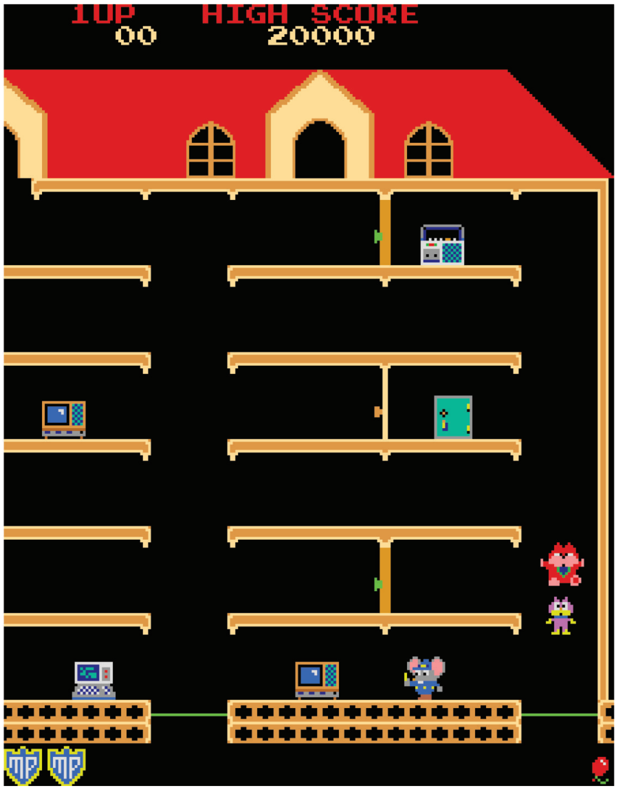


Analysis: Mappy (1983)

Mappy features a trampoline-jumping police mouse who must recover stolen items from a gang of cat thieves led by Nyamco, or Goro in the United States (Schultz, 2002a). The player can move Mappy left and right and, by pressing a button, open and close doors to knock down the cats.

Mappy's main form of segmentation is using levels. Each level represents a side view of a multistoried house (see Figure 4) with doors, stolen items—such as computers and safes—and trampolines. The trampolines are represented by a thin horizontal line that changes color whenever it is bounced on; if a trampoline is bounced on three consecutive times, it breaks (Schultz, 2002a). When the player recovers the 10 stolen items in a house, he or she starts a new one. Different houses vary in internal layouts and roof color. The player can tell which house, or level number, he or she is playing by referring to the number of balloons in the bottom-right corner of the screen.

Figure 4
Mappy's First Level



Levels 3, 7, 11, and 15 are bonus stages and are distinct for several reasons. First of all, there are no floors, only trampolines. There are no enemies, and there is a new objective: clear all the balloons. If the player fails to pick all the balloons before the music stops or if he or she falls through a trampoline off screen, he or she resumes play on the next level with no lives lost. Mappy's bonus stages are an opportunity for winning extra points—at the end of each bonus stage is a special balloon with a picture of Nyamko, which alone is worth 2,000 points.

Select Analyses

We now discuss two specific games: Battlezone and Robotron: 2084. By analyzing and discussing these games, we demonstrate how classic arcade games represent a sort of "primordial soup" in which we identify the weak instances of what, in later years, would become canonical forms of segmenting gameplay. In these games we observe early—sometimes clumsy, sometimes brilliant—attempts at toying with notions of segmented gameplay.

Analysis: Battlezone (1980)

Battlezone is an early example of the death match–style of games that would become popular 15 years later with games such as Doom and Quake, in which the player battles opponents similar to himself or herself in an enclosed arena (idSoftware, 1994, 1996). Atari's Battlezone is a one-player game depicting a first-person view from inside a tank (Atari, 1980a). The player must destroy enemy tanks, intelligent missiles, and fast "super" tanks. The battles are fought in a large valley surrounded by unreachable mountains and volcanoes. The ground contains indestructible pyramids and boxes that provide protection. Also, flying saucers appear periodically; they do not shoot at the player, but shooting them can award the player high points (Atari, 1980b).

Battlezone makes for an interesting case study because it highlights a tension between the general form of challenge segmentation and its subtype wave. Battlezone uses arguably a general form of challenge because defeating an opponent triggers a small pause before the next one appears; there is also a development in terms of difficulty because the opponents "become more evasive as the game progresses" (Atari, 1980b). However, successive opponents are visually indistinguishable from the last; there is no record of the number of enemies the player has defeated, nor is there any attempt to humanize or individualize the tanks as they are encountered.[16] In fact, the progression in difficulty is not openly communicated to the player and is not particularly transparent. For a game to exhibit a strong example of generic challenge segmentation, the challenges should be strongly distinguished; Battlezone seems to be a weak case of that. Because each tank is indistinguishable from the next, there is no perception of facing a new challenge.

Occasionally, between the appearance of one tank and the appearance of the next, a series of enemy missiles is fired at the player in succession (the player has to destroy one at a time). The missiles are different from the tanks because they fall from the sky away from the player and then move directly toward the player, in a straight line or zigzagging. The player has to destroy this homing missile before it reaches his or her tank. This variation is interesting because a *wave* of missiles is purposefully targeting the player. The homing missile is also used to get the player back in the game if he or she decides to constantly move away from his or her opponent. When the distance between the player and the enemy tank is too great, a warning sound is emitted and a missile is fired at the player.

If we accept the notion that Battlezone has only wave segmentation, then it is interesting to note that there are two distinct types of waves: individual tanks and homing missiles. If we subscribe to the notion of general challenge segmentation, then the player faces a series of challenges against tanks that are punctuated by waves of missiles.

Analysis: Robotron: 2084 (1982)

Robotron: 2084 is a game in which one or two players can alternate to control a mutant clone to "deactivate six types of robots with a laser gun, while the robots' armada (including tanks, electrodes and cruise missiles) will be deployed against the mutant" (Williams, 1982b). The game is played with two 8-directional joysticks that independently control the movement of the character and the direction of the laser fire (e.g., the player can move left while simultaneously firing down).

On the surface, Robotron segments gameplay using waves. The player begins each wave in the center of the screen, surrounded by enemies closing in on him or her. To survive, the player must eliminate all vulnerable opponents; other opponents, though, are invincible. Each successive wave has a configuration of enemies different from the previous one. According to Eugene Jarvis, Robotron's game designer,

> The philosophy of enemy design was to create a handful of AI opponents as unique as possible from one another, with unique properties of creation, motion, projectile firing, and interaction with the player. The enemies would be deployed in a wave related fashion, with distinct themes for each wave. (Hague, 2002)

The player deals with "the tension of having the world converge on you from all sides simultaneously" (Hague, 2002) and must employ different strategies to succeed. For example, every five waves there is a "Brain Robot Wave," which consists mainly of brain robots (enemies) and human family members (not enemies). One of the brain robots' attacks is "reprogramming" a human into an evil prog that homes in on the player. It is in the player's best interest to rescue as many humans as fast as possible (the point system also rewards this behavior; Butler, 2005).

Robotron is interesting because of the tension between the notions of level and wave. Although there are no particular indications that successive waves occur in different locations (levels) or regarding how they are related, each wave introduces new gameplay elements through the use of new enemies. As Jarvis describes, the different themes are problematic to the idea of wave because the player must employ different strategies rather than rely on sharper reflexes.

Conclusions

We have described a particular moment in time, the golden age of videogames, when the design space for games witnessed an incredible growth as the new possibilities offered by a new medium were explored. An example of this creative explosion can be seen in how segmentation of gameplay became richer and more complex than in previous forms of games. However, we do not pretend to indicate that these are the only forms of segmentation that can be observed. Rapid advances in the technological capabilities of computing devices—increased storage capacities, better graphics, and faster processors—have resulted in incredible improvements in the representational aspects of videogames and enhanced possibilities for gameplay.

The influence of technology on the design and development of games is still in force today. We are beginning to observe what might be new forms of gameplay segmentation that would not have been foreseeable with the technology of the early 1980s. For example, we now see games whose gameplay seems to be subdivided according to narrative elements or the dramatic needs of a story. In many cases, these subdivisions (e.g., chapters, acts or scenes) may seem to have a rather weak impact on gameplay. At other times, such as in flashbacks in which the player character can be controlled, the narrative segmentation is usually accompanied by a stronger segmentation of another type. However, games such as Mateas and Stern's Façade dynamically segment and structure gameplay into an Aristotelian dramatic arc (Mateas & Stern, 2003). In addition, traditional forms of segmentation, such as challenges, are increasingly being presented to the player under a narrative pretense that is simply too important to ignore. For instance, consider a militaristic aviation game, in which the player controls a jet fighter and has to go through a series of missions. The essential gameplay remains the same, in the sense that the player controls the same jet fighter; however, certain missions may specify some constraints on what can be done. For example, one mission may involve protecting an objective from enemy fighters, the goal of another mission might be to destroy an enemy base, whereas a third might require the player to simply reach a certain location without firing his or her weapons. Examples of games in which narratively framed missions are important include the Wing Commander and X-Wing series. In the examples mentioned, changes in gameplay, particular restrictions, and goals of a mission are all framed in a narrative context, without which the player would have great difficulty figuring out how to succeed.

We have shown, through the lens of segmentation of gameplay, how classic arcade videogames can be ideal objects for the analysis of games, gameplay, and the design space of games in general. In our eyes, classic arcade games are the "primordial soup" from which many of the future conventions of game design were proposed, explored, and tested on the public. It is interesting to note how many of these conventions remain even as the technological constraints that afforded their creation no longer exist. The reasons for this are yet to be explored, particularly how these conventions have come to structure the expectations of players together while contributing toward defining game literacy.

The simplicity of classic arcade games, relative to contemporary games, provides us, as researchers, with direct access to the original "building blocks" of videogames. In contemporary games, these building blocks are obscured by multiple layers of complexity and higher degrees of interrelations between them. Analyzing classic arcade games allows us to understand these building blocks and define a rich vocabulary to discuss them. We can only expect the design space of games to continue growing, as new technologies are developed and new design features are explored. Therefore, we look forward to exploring, understanding, recombining, and redesigning new games under the light of a richer conceptual vocabulary for the discussion, analysis, and critique of games.

Notes

1. Further details are available at http://www.gameontology.org.
2. Our use of the term *chunk* should be understood colloquially as a noteworthy portion of gameplay.
3. Another project seeking the same goals using a different methodological approach can be seen in (Björk & Holopainen, 2005).
4. See http://www.gameontology.org.
5. Tic-tac-toe is a game between two players, "O" and "X," who alternate in marking the spaces in a 3 × 3 board. A player wins by getting three of his or her own marks in a horizontal, vertical, or diagonal row.
6. We use rounds instead of turns to avoid confusion with taking turns.
7. Alternate names include RoShamBo, JanKenPon, Mora, and Farkle.
8. Atari called this mechanism Add-A-Coin. The actual time awarded per quarter depended on how the machine was configured by its operator.
9. In Dungeons & Dragons, level can refer to the amount of power and experience a character has (fifth level fighter vs. second level fighter) and the depth of a dungeon (Level 3 of the dungeon of doom).
10. We argue that Pac-Man's segmentation of gameplay is by challenge, not space. In Pac-Man, the player must always "solve" the same maze but in each case faces an ever-increasing challenge. The power-pills' effect is shortened, giving the player less time to chase and potentially gobble up ghosts.
11. A reference to the skyscraper-climbing, woman-kidnapping gorilla in the movie *King Kong*.
12. The term *spatial checkpoints* is used to differentiate it from *checkpoint*, which is another element of our ontology that is not necessarily spatial in nature.
13. Originally from an interview in 1982. Iwata is currently president of Nintendo Co. Ltd.
14. Another example is Crazy Balloon, in which the player must guide a balloon through a maze without touching the walls. There is no opponent or time limit, though a long period of inaction triggers the appearance of a face that blows the balloon into the walls, where it pops (Taito, 1980).
15. Successive waves work similarly but with different bosses and objectives for each wave.
16. Later games, such as Punch-Out!!, would refine and further differentiate the general form of challenge segmentation by developing distinct opponents with different fighting styles, personalities, and graphical representations (Nintendo, 1984).

References

Amstar. (1980). Phoenix. Hialeah, FL: Centuri.
Atari. (1972). *Pong*. Sunnyvale, CA: Author.
Atari. (1979a). *Asteroids*. Sunnyvale, CA: Author.
Atari. (1979b). *Asteroids: Operation, maintenance and service manual*. Sunnyvale, CA: Author.
Atari. (1979c). *Atari Basketball*. Sunnyvale, CA: Author.
Atari. (1979d). *Atari Football*. Sunnyvale, CA: Author.
Atari. (1979e). *Atari Soccer*. Sunnyvale, CA: Author.
Atari. (1980a). *Battlezone*. Sunnyvale, CA: Author.
Atari. (1980b). *Battlezone: Operation, Maintenance and Service Manual*. Sunnyvale, CA: Author.
Atari. (1980c). *Missile Command*. Sunnyvale, CA: Author.
Atari. (1984). *Marble Madness*. Sunnyvale, CA: Author.
Atari. (1985). *Marble Madness: Operators manual*. Sunnyvale, CA: Author.

Bally/Midway. (1976). *The Amazing Maze Game*. Franklin Park, IL: Author.
Björk, S., & Holopainen, J. (2005). *Patterns in game design*. Hingham, MA: Charles River Media.
Butler, K. (2003a). *Donkey Kong FAQ. 1.01*. Retrieved March 8, 2005, from http://db.gamefaqs.com/coinop/arcade/file/donkey_kong_a.txt
Butler, K. (2003b). *Joust FAQ. 1.01*. Retrieved August 30, 2005, from http://db.gamefaqs.com/coinop/arcade/file/joust.txt
Butler, K. (2003c). *Phoenix FAQ. 1.0*. Retrieved August 26, 2005, from http://db.gamefaqs.com/coinop/arcade/file/phoenix.txt
Butler, K. (2005). *Robotron 2084 FAQ. 1.04*. Retrieved March 30, 2005, from http://db.gamefaqs.com/coinop/arcade/file/robotron_2084.txt
Centuri. (1980). *Phoenix service manual*. Hialeah, FL: Author.
Crawford, C. (2003). *Chris Crawford on game design*. Indianapolis, IN: New Riders.
DeMaria, R., & Wilson, J. I. (2004). *High score! The Illustrated history of electronic games* (2nd ed.). Emeryville, CA: McGraw Hill/Osborne.
Fabricatore, C., Nussbaum, M., & Rosas, R. (2002). Playability in action videogames: A qualitative design model. *Human Computer Interaction, 17*(4), 311-368.
Falstein, N. (2004). *The 400 project*. Retrieved Oct 29, 2004, from http://www.theinspiracy.com/400_project.htm
Fernandez-Vara, C., Zagal, J., & Mateas, M. (2007). Evolution of spatial configurations in videogames. In S. de Castell & J. Jenson (Eds.), *Worlds in play: International perspectives on digital games research* (pp. 159-168). New York: Peter Lang.
Glaser, B., & Strauss, A. (1967). *The discovery of grounded theory: Strategies for qualitative research*. Chicago: Aldine.
Hague, J. (2002). *Halcyon days: Interviews with classic computer and video game programmers*. Retrieved January 7, 2008, from http://www.dadgum.com/halcyon/index.html
idSoftware. (1994). *Doom*. Mesquite, TX: Author.
idSoftware. (1996). *Quake*. Mesquite, TX: Author.
Irem. (1982). Moon patrol. Chicago: Williams Electronics.
Janiec, J. (2004). *Joust video game guide 1.00*. Retrieved August 30, 2005, from http://db.gamefaqs.com/coinop/arcade/file/joust_arc.txt
Juul, J. (2004). Introduction to game time. In N. Wardrip-Fruin & P. Harrigan (Eds.), *First person: New media as story, performance, and game* (pp. 131-142). Cambridge, MA: MIT Press.
Kent, S. L. (2001). *The ultimate history of videogames*. Roseville, CA: Prima.
Konami. (1982). *Time Pilot*. Hialeah, FL: Centuri.
Laidlaw, M. (1996). The egos at Id. *Wired, 4*(8), 122-127, 186-189.
Lakoff, G. (1987). *Women, fire, and dangerous things: What categories reveal about the mind*. Chicago: University of Chicago Press.
Lundgren, S., & Björk, S. (2003, March). *Describing computer-augmented games in terms of interaction.* Paper presented at Technologies for Interactive Digital Storytelling and Entertainment, Darmstadt, Germany.
Mateas, M., & Stern, A. (2003, March). *Facade: An experiment in building a fully-realized interactive drama.* Paper presented at the Game Developer's Conference: Game Design Track, San Jose, CA.
Midway. (1981). *Gorf*. Franklin Park, IL: Author.
Murray, J. H. (1997). *Hamlet on the holodeck: The future of narrative in cyberspace*. New York: Free Press.
Namco. (1980a). *Pac-Man*. Franklin Park, IL: Midway.
Namco. (1980b). *Rally-X*. Franklin Park, IL: Midway.
Namco. (1983). *Mappy*. Franklin Park, IL: Midway.
Nintendo. (1981). *Donkey Kong*. Redmond, WA: Author.
Nintendo. (1983). *Mario Bros.* Redmond, WA: Author.
Nintendo. (1984). *Punch-Out!!* Redmond, WA: Author.

Novak, M. (1991). Liquid architectures in cyberspace. In M. Benedikt (Ed.), *Cyberspace: First steps* (pp. 225-254). Cambridge, MA: MIT Press.

Nutting. (1971). *Computer space*. Mountain View, CA: Author.

Rollings, A., & Adams, E. (2003). *Andrew Rollings and Ernest Adams on game design*. Indianapolis, IN: New Riders.

Salen, K., & Zimmerman, E. (2004). *Rules of play: Game design fundamentals*. Cambridge, MA: MIT Press.

Schultz, A. (2002a). *Mappy FAQ. 1.0.0*. Retrieved August 31, 2005, 2005, from http://db.gamefaqs.com/coinop/arcade/file/mappy.txt

Schultz, A. (2002b). *Moon Patrol (arcade) FAQ. 1.0.0*. Retrieved March 30, 2005, from http://db .gamefaqs.com/coinop/arcade/file/moon_patrol.txt

Schultz, A. (2002c). *Rally-X (arcade) FAQ. 1.0.0*. Retrieved August 30, 2005, from http://db.gamefaqs.com/coinop/arcade/file/rally_x.txt

Schultz, A. (2003). *Time Pilot (arcade) FAQ. 1.0.0*. Retrieved April 4, 2005, from http://db.gamefaqs.com/coinop/arcade/file/time_pilot.txt

Sellers, J. (2001). *Arcade fever: The fan's guide to the golden age of videogames*. London: Running Press.

Taito. (1978). *Space Invaders*. Franklin Park, IL: Bally Midway.

Taito. (1980). *Crazy Balloon*. Wheeling, IL: Author.

TSR. (1974). *Dungeons & Dragons*. Lake Geneva, WI: Author.

VidKidz. (1982). *Robotron: 2084*. Chicago: Williams Electronics.

Wikipedia. (2005). *Hotseat*. Retrieved November 21, 2005, from http://en.wikipedia.org/wiki/Hotseat

Williams. (1980). *Defender*. Chicago: Author.

Williams. (1982a). *Joust*. Chicago: Author.

Williams. (1982b). *Robotron: 2084 instruction manual for upright and compact games*. Chicago: Author.

Wolf, M. J. P. (2001). Space in the video game. In M. J. P. Wolf (Ed.), *The medium of the video game* (pp. 51-75). Austin: University of Texas Press.

Zagal, J., Mateas, M., Fernandez-Vara, C., Hochhalter, B., & Lichti, N. (2005). Towards an ontological language for game analysis. In S. de Castell & J. Jenson (Eds.), *Changing views: Worlds in play: Selected papers of DIGRA 2005* (pp. 3-14). Vancouver, British Columbia, Canada: Digital Games Research Association.

Zagal, J., Nussbaum, M., & Rosas, R. (2000). A model to support the design of multiplayer games. *Presence: Teleoperators and Virtual Environments, 9*(5), 448-462.

José P. Zagal is a PhD candidate at the College of Computing at the Georgia Institute of Technology. His doctoral dissertation investigates educational and learning issues involved in studying videogames. His research interests also include the formal and structural analysis of games.

Clara Fernández-Vara is a research associate in the Singapore-MIT GAMBIT Game Lab at the Massachusetts Institute of Technology. She is writing her dissertation on storytelling in adventure games. Her research interests focus on the development of theoretical frameworks to apply them to videogame analysis, criticism and design.

Michael Mateas, PhD, is an assistant professor in the Computer Science Department at University of California, Santa Cruz. His research interests include artificial intelligence for autonomous characters and interactive stories, artificial intelligence for games, and formal and structural game analysis.