Gameplay Segmentation in Vintage Arcade Games

José P. Zagal, Clara Fernández-Vara and Michael Mateas

Introduction

This chapter explores how early videogames found ways to provide players with novel and complex gameplay experiences; in particular, 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 we have developed it in the context of the *Game Ontology Project* at the Georgia Institute of Technology. [i,63] In this chapter we introduce and discuss the term, as well as 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.[ii] For example, words like *level*, *boss*, and *wave*, among many others, refer to ways gameplay was 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 1970's and early 1980's witnessed an explosion of games and game styles, defining many videogame genres still with us today.[31,53]

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 will explore below.[42]

We selected the following arcade 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 begin to 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 wish to presume that our typifications of gameplay segmentation are in any way final, decisive or complete. Certainly, new gaming hardware, input and output devices, as well as novel game designs, will introduce new ways of segmenting gameplay and force us to re-examine the ones proposed here.

The Game Ontology Project

The ensuing analysis uses concepts and terminology developed as part of the Game Ontology Project (GOP).[63] GOP offers a framework for describing, analyzing and studying games, by defining a hierarchy of concepts abstracted from an analysis of many specific games. GOP borrows concepts and methods from prototype theory as well as grounded theory to achieve a framework that is continually evolving with each new game analysis or particular research question.[24,34] The use of the term *ontology* is borrowed from computer science. 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 both by abstracting more general concepts and identifying more precise or specific ones. An ontology is different than 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.[22,23] It does not intend 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/non-games 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, but to describe the design space of games.[iii,35]

The ontology abstracts away the representational details of games. Issues of setting (e.g. medieval castle, spaceship), genre (e.g. horror, sci-fi), and the 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 in order to achieve broad coverage without having to characterize abstract notions of setting and genre. Thus, we avoid the Sisyphean task of building an abstract model of human culture. A deep reading of any particular game would require an

analysis of its representational conventions, allusions and connotations. The ontology helps position the more formal or structural elements of the game within the game design space; other methods and techniques would be required to unpack representational issues.

The top level of the ontology consists of five elements: interface, rules, goals, entities, and entity manipulation. The *interface* is where the player and game meet, the mapping between the embodied reactions of the player and the manipulation of game entities. It refers both to how the player interacts with the game and how the game communicates to the player. The *rules* of a game define and constrain what can or can't be done in a game; they lay down the framework, or model, within which the game shall take place. Rules regulate the development of the game and determine the basic interactions that can take place within it.[iv] *Goals* are the objectives or conditions that define success in the game. *Entities* are the objects within the game that the player manages, modifies or interacts with at some level. Finally, *entity manipulation* encompasses the alteration of the game made either by the player or by in-game entities. Entity manipulation thus refers to the actions or verbs that can be performed by the player and by in-game entities.

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. As explained previously, we include both strong and weak examples; the weak examples describe 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 the parent element. Finally, the part-of relation captures the notion of compound elements that are constructed out of other elements (parts).

In this chapter, we will be using the sections of the ontology that refer to segmentation of gameplay, whose terms are expanded below. Since a description of the entire ontology, currently consisting of more than 150 elements, is beyond the scope of this chapter, we refer interested readers to Zagal *et. al*. for further details.[63]

Games and Segmentation

Playing a game, alone or with other players, 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), where the match is divided into forty-five-minute halves. Splitting the total duration of the game in two is an example of a way to segment gameplay. Another common form of gameplay segmentation is by coordinating players' actions. For example, many board games force players to take turns, alternating periods of action 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 where players take turns is very 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/shorter elements or chunks of gameplay. As we saw in the examples above, it is not something 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 example, words like *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 have identified 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, as well as 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. As mentioned previously, 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 sub-locations. Some words used to describe particular forms of spatial segmentation include *boards*, *levels*, *maps*, and even *worlds*.

The third form is challenge. In this case a game has sub-units that 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 must solve a series of puzzles. Solving a particular puzzle allows the player to attempt the next one, and so on.

	General Types of Segmentation of Gameplay
1	Temporal
2	Spatial
3	Challenge

Table 1: General Types of Segmentation of Gameplay

Most games – contemporary videogames in particular – include multiple forms of segmentation that are often interrelated or even co-occur. Although it is rare for a game to exhibit a single form of segmentation, often one form is more salient because of its greater impact on gameplay. For example, *Donkey Kong* has spatial segmentation in the form of levels, as well as 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 would have less of an 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 very similar, although easier. Also, other games use several modes of segmentation in different occasions—a game may be

temporally segmented in one section, while the rest of it could be spatially segmented. We also note that sometimes, 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 begin by focusing on temporal segmentation.

We will examine many of these issues by providing detailed descriptions of each form of segmentation, including some sub-types we identified. The descriptions will be supplemented with discussions of particular games, in the form of case studies. The object of each case study is to highlight a particular form (or sub-form) of segmentation, which other forms may be present, and how the overall gameplay experience is mediated by the decisions the game designers made regarding segmentation of gameplay. The last section of this chapter will recapitulate the different forms of segmentation in the context of additional case studies of select vintage arcade games.

Temporal Segmentation

Perhaps the most traditional form of segmentation is temporal. In most non-electronic games, temporal segmentation is the only way of breaking down gameplay. This type of segmentation can take two distinct forms—the first regulates who plays when; the second specifies time limits or periods of gameplay. The first form is about *coordination*, while the second uses time as a *resource*. Table 2 shows the main forms of temporal segmentation as well as some specific cases that will be described in further detail in the following subsections.

Temporal Segmentation	
Temporal Coordination	Turn taking, Turn based, Interleaved games
Temporal Resource	Time limit, Subdivisions of total available time

Table 2: Temporal Segmentation and Specific Forms

Temporal Coordination

Temporal segmentation by coordination refers to how a game regulates the actions of a player in a game and how these actions are distributed in 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, while the other players wait for their turn to play; Tic-Tac-Toe is a classic example of this type of coordination.[v] In turn-based (rather than turn-taking) games, players independently decide their actions and then resolve the consequences of these actions simultaneously. Once the actions have been resolved, a new turn begins and the players must again decide their actions. Rock-Paper-Scissors is a classic example of turn-based coordination.[vi] This form of segmentation regulates players' actions over time, but does not constrain the length of time their moves can take. For instance, completing a 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.[48]

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[58], consists of taking turns over two or more independent game sessions.[14,64] For example, this occurs in games where the first player plays until he 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 players' game and its status. For example, the top left corner of the screen may show the current score of player 1 while the top right corner displays player 2's score. Notably, this new form of coordination is due to the affordances of technology; in this case, the computer is well-suited to manage and maintain multiple game states simultaneously.

Temporal Resource

A game can establish its total duration as well as any sub-periods and their length. When duration is explicitly regulated by a game, then time is being treated as a resource. For instance, a game might last ten minutes played in two halves of five minutes each. In an obvious attempt to reflect the sport it emulates, examples of early arcade games that employed this form of segmentation include *Atari Football*, *Atari Basketball* and *Atari Soccer*.[5,6,7]

A temporal subdivision does not have to be constant for the duration of the game. Time can be considered a resource, in the sense that there might be certain actions, rules, or events that modify the game's duration. For example, in *Atari Basketball*, inserting a

quarter in the machine granted the player one minute of playtime.[vii] Segmentation via temporal resource is common in sports games, such as basketball or soccer. In soccer, a game lasts ninety minutes, played in two halves of forty-five minutes. The referee, however, has the authority to grant extensions on the duration of the half period if he believes certain in-game events warrant it. Another form of temporal segmentation takes place when the player is allotted a specific amount of time to complete a certain task, fulfill a goal, or simply do the best he can. Also, many games use a different visual representation for what is essentially a time-limit. For example, *Donkey Kong* has a numerical counter that usually starts at five thousand and then decreases in increments of one hundred.

Temporal resource is not necessarily equivalent to a time-dependent goal. For example, in racing games the winning condition typically requires taking the least amount of time to reach a certain location. In these cases, it is important to have a "low time" and there is an implicit "maximum time" (even 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 involved 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.[11,12]. 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 6 unrelated

isometric courses. There are numerous obstacles and unfriendly creatures that will hinder the player's progress. A timer ticks away; when it reaches "00" the game ends. Falling off the edge or from too high will cause the marble to re-appear in a previous location (further "up" the course). Since the re-appearance is not immediate, there is also a loss of several precious seconds on the timer.

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. However, 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. Interestingly, the temporal resource segmentation allows the player to begin a level adding leftover time if he performed well on the previous one (though this is not true of all levels). However, if the player loses the marble over the edge, he 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 offered a two-player mode. However, the activities of the different players are not regulated by temporal coordination since both players could participate simultaneously. In fact, players could decide whether they wanted 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.[38] This world, while 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.[30]

Due to technical limitations, early arcade games like *Computer Space* and *Pong* faced significant difficulties in representing something on the screen that could even be identified as a space.[2,46] Early arcade games also tended to present the entire gameworld in one screen [62]; nothing happened off-screen where the player could not see. Soon afterwards appeared games that unfolded over a series of different, discrete screens; *Gorf* and *Donkey Kong* are examples of this.[37] In many cases, there was an explicit continuity of space that was communicated to the player; he had to navigate the gameworld in discrete segments or chunks. 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 when this division also partitions gameplay. In these cases the gameworld is not presented as a continuous whole but rather as distinct subspaces that are navigated separately, and which 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, as well as 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 that he is participating in a virtual space larger than its onscreen representation, and that this space is traversed in parts. A series of disconnected screens which 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*.[59] In this case, the screen scrolls smoothly allowing the player to see the rest of the gameworld.

Spatial segmentation is quite common in videogames since computers can efficiently and cheaply store the data needed to represent or generate expansive virtual worlds.[38] Storing additional information, such as specific rules for different spatial segments, is relatively easy for computers. In fact, virtual spaces are a habitat for imagination where physics can be redefined or even altogether invented.[45]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 boardgame with 40 separate square boards, each having an area of one square meter, or a basketball court 20 kilometers in length!

Level

A *level* is a recognizable subspace of the gameworld. As a spatial form of segmentation of gameplay, *levels* are "discrete virtual locations containing tasks that must be accomplished before players can advance".[33] However, the word *level* is inherently problematic when referring to games. For instance, *level* has also been used to refer to the

degree of difficulty a player encounters. In this sense of challenge, the higher the level, the more difficult the game. In fact, in games such as the pencil and paper role-playing game *Dungeons & Dragons*, the word *level* has multiple uses within the same game.[viii,56] In the case of our ontology, references to the difficulty are considered under challenge segmentation (discussed in detail 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, intermediate animation, or cut-scene. However, that discontinuity cannot come at the expense of a loss of the necessary spatial relationship between the spatial segments. The tension between 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, etc, re-set 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 would 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.[ix,39]

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

Analysis: Donkey Kong (1981)

Nintendo's *Donkey Kong* allows players to control a humanoid character called Jumpman (later renamed Mario[31]), who desperately tries to recover his beloved from the clutches of a stubborn gorilla named Donkey Kong. Jumpman can move left and right, jump, and climb up and down ladders.

Donkey Kong is a great example of multiple co-existing forms of gameplay segmentation. First, in two-player mode, the game allows interleaved-games. When the two-player mode is selected, their scores are represented in the top-left and top-right areas of the screen. *Donkey Kong* has four distinct screens – levels in this case – which 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 level, 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 re-start 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[15]), and

because the player's progress through the game involves a series of specific screen 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 his time the taunting message re-appears and the label "L=01" changes to "L=02". Table 3 describes the five different configurations in total, where each successive sequence introduces a new level and re-visits 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, since the levels become harder and it takes longer to get to the top to defeat Kong.

Label/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: Level Sequence in Donkey 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 upon 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 stage because in order to defeat the special antagonist (boss), the player needs strategies different from those used in previous levels. In the case of *Donkey Kong* this means removing rivets instead of trying to reach the girl.

Donkey Kong's screens are a canonical example of level. Since each level is higher than the last, there is a sense of spatial relationship between them. There is the implicit idea that sequences of levels are different skyscrapers, which Jumpman must climb in pursuit of the ape.[x] The rivet level is always the top of the skyscraper from which Kong must fall defeated. At the beginning of each level the timer is reset and Jumpman must begin again from the bottom. However, though each level is distinct, they are also related through particular gameplay elements. For example, 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) [xi], being the boundaries between these sublocations. In order to be considered a form of spatial segmentation, there must be differences that impact 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/modifying some game variables – such as 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. Finally, the player must also be aware he has made this transition from one sublocation to another in order to consider a spatial subdivision a checkpoint. Games with gameworlds larger than the screen and which scroll continuously when the player moves do not usually feature spatial checkpoints unless some of the conditions above 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.[28] *Moon Patrol* also allowed interleaved playing for two players, sporting the, at the time, uncommon feature of allowing them to continue from the last checkpoint if they inserted another coin after losing all lives.[53]

Moon Patrol is also an excellent example of hierarchical forms of spatial segmentation. At the highest level, the player must complete a series of "courses" or levels, in which the player must get 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 greater its 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" indicating the overall progress the player has made. When the buggy reaches each of the letters marked above the bar, the player is informed of the time he took to travel between the previous letter and the current one, and whether his time is low enough to score extra points. Each newly entered section is distinguished from the previous by a changing background.

Figure 1 : Course subdivisions in Moon Patrol

The course subdivisions are a weak example of a level. They can be considered levels because there is a distinct notion that 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 now has a new goal which is to 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 did, and whether or not he scored any bonus points in the level. However, the case for calling them levels is weakened because each section continues from the next.

Figure 2: Reaching point "G"

Moon Patrol also has a strong example of spatial checkpoints, since each level is also subdivided. For example, in Figure 2, the player is in level (E-J) and is about to reach point “G”. In order to get to the spatial checkpoint “J” he will have to pass letters “H” and “I”.[50] These spatial checkpoints are important for two reasons: each 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 would not have to 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, since 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

One of the least straightforward forms of gameplay segmentation is segmentation by challenge. It is particularly 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. The essence of this form of segmentation 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 a test or trial. Specific forms of challenge segmentation include puzzles, boss levels, and waves.

Sometimes the increase in challenge is accomplished by making enemy characters more dangerous, making the player character more vulnerable, or enforcing shorter time limits. Other ways include slight modifications of the rules, or 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 while *Robotron: 2084* introduces new enemies that require different strategies to defeat.[57]

Wave

A wave is a particular form 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 completely destroyed as they approach the player. There is usually a pause, or respite, between the appearance of different waves. One of the defining aspects of the wave is that the player's inaction will result in the premature 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.*[10,54] The former was described by S. Iwata as a game where "A little neglect may breed great tension" while in the latter "you were only postponing the inevitable. [xii,31] This war was hell, and it was never long before you died in a blaze of glory."[53]

Analysis: Asteroids (1979)

In *Asteroids,* the player controls a small triangle-shaped ship which fires in the direction it faces.[3] Pressing the "left rotate" or "right rotate" buttons causes the ship to rotate in the respective direction. Additionally, 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 players have shot all asteroids, a new set of large asteroids appear. At the beginning of the game, four large asteroids appear, the next cycle starts with six large asteroids, the next time eight, and thereafter ten – increasing the challenge.[4]

Asteroids' main form of segmentation is challenge segmentation in waves. The player faces a series of 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. The successive waves affirm the intention of increasing the difficulty level. Interestingly, 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 in order to trigger the next wave. In fact, it is usually in his best interest not to, since there is an award 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 as well as duration of gameplay.[31]

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. Therefore, it is not surprising that this form of segmentation flourished with the advent of home computers.

Games that use puzzle segmentation present the player with a series of puzzles, which must be solved before the next one is available. At times the progression through puzzles of increased difficulty need not be strictly linear. A puzzle can be recognized as a challenge, where there is no active agent against which the player is competing[20], i.e. a puzzle is static. Typically, it will have at most a few correct solutions, requiring problem-solving skills rather than good hand-eye coordination and quick reflexes.[47] This form of segmentation is commonly seen in adventure games. For instance, if the player must re-arrange crates in order 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 solution allows the player to progress through the gameworld.

Analysis: The Amazing Maze Game (1976)

In *The Amazing Maze Game* (AMG) the player must find the way out of a single screen maze before his opponent, either the computer or another player.[13] When the computer controls the opponent, it moves more slowly than the player but never makes a mistake.

AMG is one of the few early arcade games where we see challenge segmentation by puzzles.[xiii] It is arguably a weak form of puzzle segmentation because there is an active agent against whom the player is competing—he 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. In other words, 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.

If *AMG* 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, then he can continue to play, solving new mazes against an ever-faster opponent until he is eventually defeated. The player, regardless of his 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 in order 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 be increased significantly, the player might have to resort to new tactics, or there might be additional conditions/restrictions on the player's actions. As the name implies, the player may also have to face the boss, an opponent that is 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.[1,32] The first wave in *Time Pilot* is an example of this[xiv]—after the required fifty-six enemy planes have been shot down, the player must destroy a blimp before "time warping" to the next wave.[52,53]

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 the case of *Donkey Kong*, Kong appears in every level; he conveniently manages to escape, until the last level – the boss challenge – where he can finally be defeated. Defeating Kong in this case requires using a different tactic, releasing all the bolts holding up the structure, rather than simply climbing up.

Analysis: Phoenix (1980)

Phoenix is a game where 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. Additionally, the player can activate a "force field" as means of protection from the missiles and birds. The birds, called phoenixes, are destroyed when they collide with the force field or are shot by a rocket.[19]

The player faces a series of waves. The first two waves consist of sixteen phoenix fighters. The third and fourth waves consist of 8 phoenixes larger than the fighters, though harder to destroy.[17] 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.

Figure 3 : Boss Level in Phoenix

Thus *Phoenix*'s main form of segmentation is in waves. As described previously, the player must face a series of waves with opponents that become more dangerous and harder to defeat. For the purposes of this case study, however, the fifth wave is the most interesting. After clearing the fourth wave, the player must face a boss challenge: the attack of the "spacefortress". While avoiding waves of phoenix fighters, the player must blast through a protective barrier shielding a space creature inside the spacefortress (See Figure 3).[19] In this case, defeating waves of phoenix fighters is not enough, since only more will appear. The player must act quickly to defeat the boss since the spacefortress slowly inches its way down to the bottom of the screen. Players delaying the destruction of the spacefortress until the last second are rewarded extra points.

Bonus Stage

Of all the forms of challenge segmentation we have explored, the bonus stage is perhaps the most unusual. Strictly speaking, the purpose of a bonus stage is to present the player with an opportunity to earn rewards without the risk of losing the game. While 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. Thus the challenge perceived by the player changes drastically, becoming a sort of anti-challenge: you can't lose the game while playing a bonus stage. Nevertheless, bonus stages are usually quite challenging to those players who want to score maximum points. Early arcade games with bonus stages include *Mario Bros.* and *Mappy*.[41,43]

A bonus stage is different from a period of invulnerability within a normal part of a game. Each of the mazes in *Pac-Man* has four power-pills that must be consumed. Whenever Pac-Man "eats" a power-pill, a reversal occurs—for a limited time, the player can defeat the ghosts. If he catches them before the effect wears out, he 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".[21,40,53]. Starting with the 3rd level and every 4 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 however, because the player can still lose lives by crashing into rocks or the

motionless cars.[51] Another weak example of a bonus stage occurs in *Joust*.[60] Joust has "egg waves" that occur every 5 waves.[29] Instead of enemy riders, the player is presented with 12 harmless eggs distributed among the different platforms.[16] The eggs will hatch into enemy riders after a time, but until then they are "basically free points".[29] Both *Rally-X* and *Joust* have weak example of bonus rounds because, although you can lose the game, the rounds are a break from traditional gameplay and they begin with an extensive amount of time during which the player is effectively invulnerable.

Analysis: Mappy (1983)

Mappy, a commercially unsuccessful game in the United States [53], features a trampoline-jumping police mouse who must recover stolen items from a gang of cat thieves led by Nyamco (changed to Goro in the United States[49]). 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 multi-storied house (see Figure 4) with doors, stolen items – such as computers, tv sets and safes – and trampolines. The trampolines are represented by a thin horizontal line that changes color whenever it is bounced upon; if a trampoline is bounced on three consecutive times it breaks.[49] When the player recovers the ten stolen items in the house, he starts a new one. Different houses have different internal layouts as well as varying roof colors. The player can tell which house, or level number, he is playing by referring to the number of balloons in the bottom-right corner of the screen.

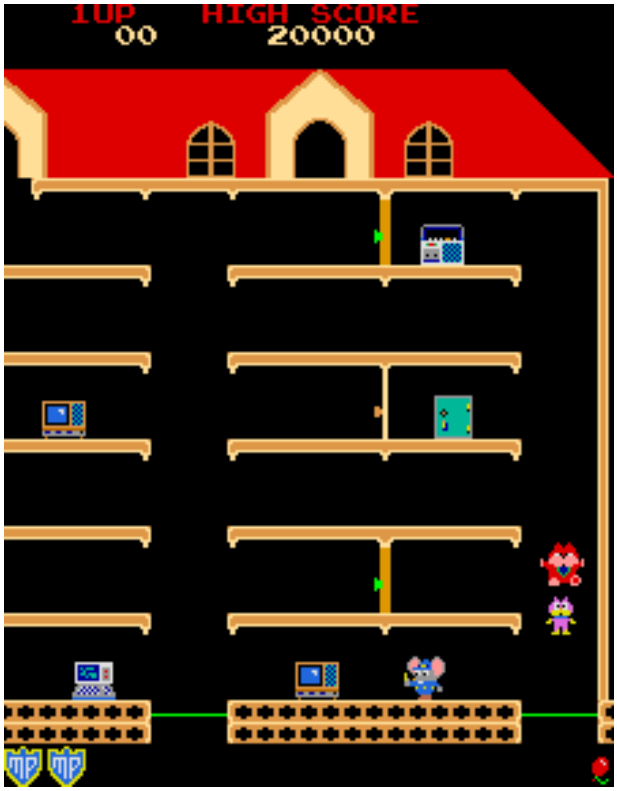

Figure 4: Mappy's First Level

Levels 3, 7, 11 and 15 are bonus stages and are distinct from the normal levels for several reasons. First of all, there are no floors, only trampolines. There are no enemies either, though there is a new objective: clearing all the balloons. If the player fails to pick all the balloons before the music stops playing or if he falls through a trampoline off-screen, he resumes play on the next level with no lives lost. *Mappy*'s bonus stages are an excellent opportunity to get a lot of extra points—at the end of each bonus stage is a special balloon with a picture of Nyamko, which alone is worth 2,000 pts., on top of the point-value of all the other balloons in the level, and the "perfect" bonus award.

Select Analyses

In this section we will take a look at two specific games: *Battlezone* and *Robotron:2084.* Using the analysis and discussion of these games we will attempt to demonstrate how classic arcade games represent a sort of "primordial soup" where we can identify the weak instances of what in later years would become canonical forms of segmenting gameplay. In these games we can observe early – sometimes clumsy, sometimes brilliant – attempts at toying with notions of segmented gameplay.

Analysis: Battlezone (1980)

Battlezone is an early example of the deathmatch-style of games that would become very popular fifteen years later, with games like *Doom* and *Quake*, where the player battles opponents that are similar to himself in an enclosed arena. [26,27] Atari's *Battlezone* is a one-player game depicting a first-person view from inside a tank.[8] 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 pyramids and boxes, which can provide temporary protection for players and cannot be destroyed by shots. In addition to this, flying saucers appear periodically; they don't shoot at players, but shooting them can award the player very high points.[9]

Battlezone makes for an interesting case study because it highlights a tension between the general form of challenge segmentation and its sub-type 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, since the opponents "become more evasive as the game progresses".[9] However, each successive opponent is 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.[xv] In fact, the progression in difficulty is not communicated openly to the player and is not particularly transparent. In order 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. Since 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 next, a series of enemy missiles are 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 towards him, in a straight line or zigzagging. The player has to destroy this homing missile before it reaches his 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 decides to constantly move away from his 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* only has 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.

Case Study: Robotron: 2084 (1982)

Robotron: 2084 is a game where one or two players can alternate to control a mutant clone, in order 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".[61] The game is played with two eight-directional joysticks which 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. In order 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."[25] The player deals with "the tension of having the world converge on you from all sides simultaneously"[25], 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).[18]

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 how they are related, each wave does introduce 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, and 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, as well as 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 1980's. 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. Other times, such as in flashbacks where 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 segments and structures gameplay into an Aristotelian dramatic arc.[36] Additionally, traditional forms of segmentation, such as challenges, are increasingly being presented to the player under a narrative pretense which is simply too important to ignore. For instance, consider a militaristic aviation game, where 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, while a third might require the player simply to reach a certain location without firing his weapons. Examples of games in which narratively framed missions are important include the *Wing Commander* and *X-Wing* series. In each of the examples mentioned, changes in gameplay, particular restrictions, and the 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. Their simplicity, 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 re-designing new games under the light of a richer conceptual vocabulary for the discussion, analysis and critique of games.

References

1 Amstar. Phoenix. Centuri, Inc., Hialeah:FL, 1980.
2 Atari. Pong. Sunnyvale, CA: Atari Inc, 1972.
3 ---. Asteroids. Sunnyvale, CA: Atari Inc, 1979.
4 ---. Asteroids: Operation, Maintenance and Service Manual. Sunnyvale, California: Atari Inc, 1979.
5 ---. Atari Basketball. Sunnyvale, CA: Atari Inc, 1979.
6 ---. Atari Football. Sunnyvale, CA: Atari Inc, 1979.
7 ---. Atari Soccer. Sunnyvale, CA: Atari Inc, 1979.
8 ---. Battlezone. Sunnyvale, CA: Atari Inc., 1980.
9 ---. Battlezone: Operation, Maintenance and Service Manual. Sunnyvale, CA: Atari Inc., 1980.

10 ---. Missile Command. Sunnyvale, CA: Atari Inc, 1980.
11 ---. Marble Madness. Atari Games, Inc., Sunnyvale, CA, 1984.
12 ---. Marble Madness: Operators Manual. Sunnyvale, CA: Atari Inc, 1985.
13 Bally/Midway. The Amazing Maze Game. Bally Midway, 1976.
14 Bjork, Staffan, and Jussi Holopainen. Patterns in Game Design. Game Development Series. Hingham, Massachusetts: Charles River Media Inc., 2005.
15 Butler, Kevin. "Donkey Kong Faq". 2003. Txt File. 1.01. March 8 2005. <http://db.gamefaqs.com/coinop/arcade/file/donkey_kong_a.txt>.
16 ---. "Joust Faq". 2003. Txt File. 1.01. Aug 30 2005. <http://db.gamefaqs.com/coinop/arcade/file/joust.txt>.
17 ---. "Phoenix Faq". 2003. Txt File. 1.0. August 26 2005. <http://db.gamefaqs.com/coinop/arcade/file/phoenix.txt>.
18 ---. "Robotron 2084 Faq". 2005. Txt File. 1.04. March 30 2005. <http://db.gamefaqs.com/coinop/arcade/file/robotron_2084.txt>.
19 Centuri. Phoenix Service Manual. Hialeah, Florida: Centuri Inc, 1980.
20 Crawford, Chris. Chris Crawford on Game Design. Indianapolis, Indiana: New Riders Publishing, 2003.
21 DeMaria, Russel, and Johnny I. Wilson. High Score!: The Illustrated History of Electronic Games, 2nd Edition. Emeryville, California: McGraw Hill/Osborne, 2004.
22 Fabricatore, Carlo, Miguel Nussbaum, and Ricardo Rosas. "Playability in Action Videogames: A Qualitative Design Model." Human Computer Interaction 17.4 (2002): 311-68.
23 Falstein, Noah. "The 400 Project". 2004. Oct 29 2004. <http://www.theinspiracy.com/400_project.htm>.
24 Glaser, B., and A. Strauss. The Discovery of Grounded Theory: Strategies for Qualitative Research. Chicago: Aldine, 1967.
25 Hague, James. Halcyon Days:Interviews with Classic Computer and Video Game Programmers. http://www.dadgum.com/halcyon/index.html, 2002.
26 idSoftware. Doom. Mesquite:TX: id Software, 1994.
27 ---. Quake. Mesquite:TX: id Software, 1996.
28 Irem. Moon Patrol. Williams Electronics, Inc., Chicago:IL, 1982.
29 Janiec, Joseph. "Joust Video Game Guide". 2004. Txt File. 1.00. Aug 30 2005. <http://db.gamefaqs.com/coinop/arcade/file/joust_arc.txt>.
30 Juul, Jesper. "Introduction to Game Time." First Person: New Media as Story, Performance, and Game. Eds. N. Wardrip-Fruin and P. Harrigan. Cambridge, Massachusetts: The MIT Press, 2004. 131-42.
31 Kent, Steven L. The Ultimate History of Videogames. Roseville, California: Prima Publishing, 2001.
32 Konami. Time Pilot. Centuri, Inc., Hialeah:FL, 1982.
33 Laidlaw, Marc. "The Egos at Id." Wired 4.8 (1996).
34 Lakoff, George. Women, Fire, and Dangerous Things: What Categories Reveal About the Mind. Chicago: The University of Chicago Press, 1987.
35 Lundgren, S., and Staffan Bjork. "Describing Computer-Augmented Games in Terms of Interaction." Technologies for Interactive Digital Storytelling and Entertainment (TIDSE). Darmstadt, Germany, 2003.

36 Mateas, Michael, and Andrew Stern. "Facade: An Experiment in Building a Fully-Realized Interactive Drama." Game Developer's Conference: Game Design Track. San Jose, CA, 2003.

37 Midway. Gorf. Midway Mfg. Co. , Franklin Park:IL, 1981.

38 Murray, Janet H. Hamlet on the Holodeck: The Future of Narrative in Cyberspace. New York: The Free Press, 1997.

39 Namco. Pac-Man. Franklin Park:IL: Midway Mfg. Co, 1980.

40 ---. Rally-X. Franklin Park:IL: Midway Mfg. Co, 1980.

41 ---. Mappy. Franklin Park:IL: Midway Mfg. Co, 1983.

42 Nintendo. Donkey Kong. Nintendo of America Inc. , 1981.

43 ---. Mario Bros. Nintendo of America Inc. , 1983.

44 ---. Punch-Out!! Nintendo of America Inc. , 1984.

45 Novak, Marcos. "Liquid Architectures in Cyberspace." Cyberspace: First Steps. Ed. Michael Benedikt. Cambridge, MA, USA: MIT Press, 1991. 225-54.

46 Nutting. Computer Space. Nutting Associates, 1971.

47 Rollings, Andrew, and Ernest Adams. Andrew Rollings and Ernest Adams on Game Design. Indianapolis, IN: New Riders Publishing, 2003.

48 Salen, Katie, and Eric Zimmerman. Rules of Play: Game Design Fundamentals. Cambridge, Massachusetts: The MIT Press, 2004.

49 Schultz, Andrew. "Mappy Faq". 2002. Txt File. 1.0.0. Aug 31, 2005 2005. <http://db.gamefaqs.com/coinop/arcade/file/mappy.txt>.

50 ---. "Moon Patrol (Arcade) Faq". 2002. Txt File. 1.0.0. March 30 2005. <http://db.gamefaqs.com/coinop/arcade/file/moon_patrol.txt>.

51 ---. "Rally-X (Arcade) Faq". 2002. Txt File. 1.0.0. Aug 30, 2005 2005. <http://db.gamefaqs.com/coinop/arcade/file/rally_x.txt>.

52 ---. "Time Pilot (Arcade) Faq". 2003. Txt File. 1.0.0. April 4, 2005 2005. <http://db.gamefaqs.com/coinop/arcade/file/time_pilot.txt>.

53 Sellers, John. Arcade Fever: The Fan's Guide to the Golden Age of Videogames. London: Running Press, 2001.

54 Taito. Space Invaders. Bally Midway, 1978.

55 ---. Crazy Balloon. Taito Corporation, 1980.

56 TSR. Dungeons & Dragons. Lake Geneva:WI: Tactical Studies Rules, 1974.

57 VidKidz. Robotron: 2084. Williams Electronics, Inc. , Chicago:IL, 1982.

58 Wikipedia. "Hotseat - Wikipedia Entry." 2005.

59 Williams. Defender. Chicago, Illinois: Williams Electronics Inc., 1980.

60 ---. Joust. Chicago, IL: Williams Electronics Inc., 1982.

61 ---. Robotron: 2084 Instruction Manual for Upright and Compact Games. Chicago, IL: Williams Electronics Inc., 1982.

62 Wolf, Mark J. P. "Space in the Video Game." The Medium of the Video Game. Ed. Mark J. P. Wolf. Austin, Texas: University of Austin Press, 2001. 51-75.

63 Zagal, Jose, Michael Mateas, Clara Fernandez-Vara, Brian Hochhalter, and Nolan Lichti. "Towards an Ontological Language for Game Analysis." Changing Views: Worlds in Play, Selected Papers of Digra 2005. Eds. Suzanne de Castell and Jen Jenson. Vancouver, Canada, 2005. 3-14.

64 Zagal, Jose, Miguel Nussbaum, and Ricardo Rosas. "A Model to Support the Design of Multiplayer Games." Presence: Teleoperators and Virtual Environments 9.5 (2000): 448-62.

Endnotes

[i] Further details are available at http://egl.gatech.edu/gamedesign/

[ii] Our use of the term "chunk" should be understood colloquially as a noteworthy portion of gameplay.

[iii] Another project seeking the same goals but favoring a different methodological approach can be seen in: Staffan Bjork and Jussi Holopainen, Patterns in Game Design, Game Development Series (Hingham, Massachusetts: Charles River Media Inc., 2005).

[iv] See Rules of Play for an overview of other definitions of rules in the context of games. Katie Salen and Eric Zimmerman, Rules of Play: Game Design Fundamentals (Cambridge, Massachusetts: The MIT Press, 2004).

[v] Tic-Tac-Toe, is a paper and pencil 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 their own marks in a horizontal, vertical or diagonal row.

[vi] Alternate names include RoShamBo, JanKenPon, Mora and Farkle.

[vii] Atari called this mechanism Add-A-Coin™. The actual time awarded per quarter depended on how the machine was configured by its operator.

[viii] In *Dungeons & Dragons*, level can refer to the amount of power and experience a character has (5th level fighter vs. 2nd level fighter) and the depth of a dungeon (level three of the dungeon of doom).

[ix] 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 cases faces an ever-increasing challenge since the duration of power-pills is shortened, providing the player with less time to chase and potentially gobble up the ghosts.

[x] This idea is conveyed via *Donkey Kong's* reference to the skyscraper climbing, woman-kidnapping gorilla in the movie *King Kong*.

[xi] 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.

[xii] Originally from an interview in 1982. Iwata is currently president of Nintendo Co. Ltd.

[xiii] Another example is *Crazy Balloon* where 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, Crazy Balloon, Taito Corporation, 1980.

[xiv] Successive waves work similarly, but with different bosses and objectives for each wave.

[xv] 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, Punch-Out!!, Nintendo of America Inc. , 1984.