

MIT Open Access Articles

*From "open mailbox" to context mechanics:
shifting levels of abstraction in adventure games*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: Clara Fernandez-Vara. 2011. From "open mailbox" to context mechanics: shifting levels of abstraction in adventure games. In Proceedings of the 6th International Conference on Foundations of Digital Games (FDG '11). ACM, New York, NY, USA, 131-138.

As Published: <http://dx.doi.org/10.1145/2159365.2159383>

Publisher: Association for Computing Machinery (ACM)

Persistent URL: <http://hdl.handle.net/1721.1/100262>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



From “Open Mailbox” to Context Mechanics: Shifting Levels of Abstraction in Adventure Games

Clara Fernández-Vara
Singapore-MIT GAMBIT Game Lab
Massachusetts Institute of Technology
telmah@mit.edu

ABSTRACT

Abstracting the fictional world to essential components is one of the first steps to design the system of a game. The amount of detail with which the fictional world is implemented as the system determines the level of abstraction of the simulation of the game [9]. This paper is a historical analysis of the design of a specific genre, adventure games, and how the levels of abstraction have shifted through time. Early adventure games, such as *Zork* or *The Lurking Horror*, had a wide range of possible actions and had more detailed simulations of the game world. Through the more than thirty years of history of adventure games, such as *Space Quest*, *Myst*, *Indigo Prophecy* or the recent *Machinarium*, the nuance of the simulation has diminished, as well as the variety of possible actions. There are two basic reasons for this simplification: first, to make the interface easier to use, and second, in order to facilitate players finding and identifying the elements of the puzzles and advance the story. This historical exploration of adventure games design provides insight on the trade-offs of choosing different levels of abstraction in the design, which may be extensible to other videogame genres.

Categories and Subject Descriptors

J.m [Computer Applications]: Miscellaneous – *video games*.

General Terms

Design, Human Factors, Theory.

Keywords

Game design, history, adventure games, interactive narrative, abstraction.

1. INTRODUCTION

Selecting the essential components of the system of the game is a basic step in the process of game design; determining the relationship between the rules and the fictional world also defines how the player participates in that world. This paper examines a specific videogame genre, adventure games, and the historical shift in the way they have implemented the game world. A more detailed definition of adventure games will be provided below; for

the moment, let it suffice to say that the phrase “adventure games” refers to the kinds of games that were derived from Crowther and Wood’s *Adventure*, from text adventure games / interactive fiction (*Zork*, *The Lurking Horror*), to point-and-click adventure games such as *Space Quest IV: Roger Wilco and the Time Rippers*, *The Secret of Monkey Island*, *Myst*, or *Machinarium*; recent releases such as the *Phoenix Wright: Ace Attorney* series and *Heavy Rain* are also considered part of the genre according to the definition below.

Adventure games are significant because they seem to follow an inverse pattern from most other videogames—while new technologies have allowed for developing more complex interactions and game worlds, adventure games have become more streamlined. For example, in the early first person shooter *Wolfenstein 3D*, aiming was paired with moving in space; most enemies followed a walking pattern, and would start shooting at the player character as soon as he entered the room. In *Half-Life 2*, on the other hand, the player can move towards one direction and look / aim at another; apart from having a wider range of actions (such as jumping and crouching), enemies have different behaviors depending on the situation. For instance, early in the game, soldiers will look at the player character suspiciously, push him back if he gets too close, and finally attack him with a taser if he insists on transgressing.

The following sections will discuss how adventure games seem to have followed an inverse approach, losing the range of possible actions in the world, as well as presenting environments that are more static and where characters seem to function more like a signpost than a character responding to the actions of the player. The reasons for this shift have to do with the trade-offs of narrowing the world in favor of helping the player solve the puzzles and thus advance the story of the game.

The historical approach here followed does not imply that the appearance of new game conventions or designs cancels out the previous ones. New design conventions co-exist with older ones, so that the transformations on how the game world is implemented in the game actually expand the design palette of adventure games.

2. DEFINING THE TERMS: FICTIONAL WORLD, STORY, SIMULATION

Videogames take place in a fictional world, which is the constructed virtual space that the player explores and navigates in. Fictional worlds are “imagined worlds” [8] where the actions and events of the game take place; “fictional” in this context refers to something fake pretending to be real [10]. The fictional world

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

FDG’11, June 29-July 1, 2011, Bordeaux, France.

Copyright 2011 ACM 978-1-4503-0804-5/11/06 ... \$10.00.

provides a frame of reference to the player's actions and becomes the setting of the story of the game.

The term "fiction" will be avoided in this discussion because it is very ambiguous. While Juul uses it interchangeably with "fictional world" [8], Aarseth seems to make it equivalent to non-interactive content [2]; in traditional narratology, the term is used to refer to story [16]. Thus this paper will use the more specific terms "story" and "fictional world", which refer to different aspects. In the context of videogames, "story" is defined according to narratological concepts, as the structured sequence of events which take place in the fictional world of the game [16].

As Juul argues, the aspects of the fictional world implemented in the rules become the simulated / virtual aspects of the game [9, 10] (see Figure 1). The rules of the game determine how the fictional world works as a simulation: which doors open and how, how gravity works, when certain events take place, how items can affect other items. By being simulated, the fictional world becomes participatory, distinguishing fictional worlds in videogames from those in other media.

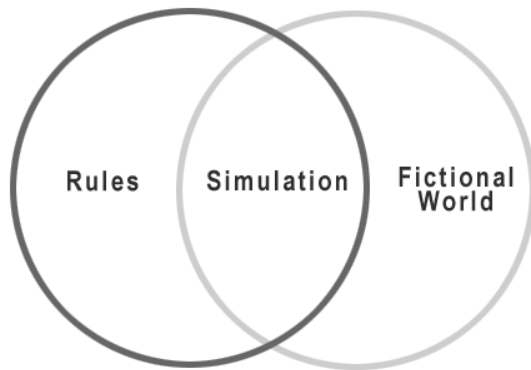


Figure 1: The intersection between the rules and the fictional world is the simulation of the game.

Those aspects of the fictional world which are not simulated are represented. Cut-scenes, backdrops, descriptions, are representations of the fictional world, which are essential for the player to make sense of what happens in the world, as well as get cues as to what to do next (see [11] for a detailed discussion of the role of cut-scenes in videogames). Conversely, there are rules of the game that are not part of the game world, such as the score system or the saving/loading features. Although these rules provide information related to state of the game, they are not part of the simulation.

It must be noted that we are not invoking the term "representation" as opposed to "simulation" and therefore not part of the game, as Frasca [6] and Aarseth [2] do. In the theoretical framework presented here, "representation" refers to the aspects of the fictional world which are not simulated, but are still integral to the videogame.

In order to implement the fictional world, it must be abstracted into basic components and made into a system. Abstraction is a primordial step in the design of the simulation [7], it is a selection process that determines what is essential to allow the player to interact with. The simulation does not implement the whole world; it only incorporates the aspects that are relevant to gameplay. Only some parts of the fictional world are simulated;

what is not simulated is usually not relevant to the core mechanics of the game. For instance, when running on the racing tracks of *Wipeout HD*, the player can see there is a larger fictional world, there are futuristic buildings and advertisements, but there is no information about the society or culture where those races take place. The expected interaction in the simulation of *Wipeout HD* is racing. The rest of the world is represented, so there is a hint of its existence, but the player cannot participate or manipulate it. Fighting games, such as *Super Street Fighter IV*, present a similar case—the player can only fight against the other player or computer-controlled character, but cannot attack any of the spectators, run away, or explore any of the lavish spaces in which the fights take place.

The simulation in a game is designed to foster specific interactions, and that is what may delimit its size and elaboration—the designer must carry out a selective process before turning the world into a simulation.

3. LEVELS OF ABSTRACTION

The first step to create a simulation from a fictional world is to establish the degree and nuance of the system. The concept of "level of abstraction" is commonly used in game design to refer to how a specific model is implemented as a game system [7, 15]. Juul refers to the level of abstraction of a videogame as the border between the fictional world and how it is implemented in the rules, within which the player can operate. "[T]he player can only act on a certain level, outside which the world is either crudely implemented [...], simply represented [...], or simply absent [...]" [9]

Juul uses *Cooking Mama* as an example, where the fictional world is a kitchen, but the possible actions in it are more limited than in real life: cucumbers can only be cut in a specific way, and one cannot order take out instead of cooking [9]. The parts of the game that are not relevant to gameplay have less functionality (e.g. the player sees a picture of the dish you cooked, but there is no action to eat it), or are not implemented at all (e.g. the player cannot visit the rest of the house where the kitchen is). Following this logic, the events in the fictional world that cannot take place as a result of the game system will either be represented or left out.

Whereas Juul understands the level of abstraction as a border between the simulation and the fictional world, it is more productive to interpret it as the area of intersection between rules and the world. Thus, the level of abstraction is the amount of overlap between both. This illustrates more clearly how there may be different degrees of abstraction. A smaller overlap, where either the rules or the fictional world present a larger area in the diagram, indicates a less fine-grained interaction. The larger the area of overlap between rules and the fictional world, the higher fidelity in the simulation is (see Figure 2).

The level of abstraction can also be applied to the rules that establish how the player interacts with the simulation and how much nuance it provides. For example, in *Half-Life 2*, the player character must gain momentum in order to make a longer jump because of the detailed physics system of the world, whereas in *Yoshi's Island: Super Mario Advanced 3* the player character may jump in mid-air, but there is no inertia.

The game may select what type of action is going to be performed depending on the context. These are *context mechanics*, defined by Sicart as "mechanics that are triggered depending on the

context of the player presence in the game world” [18]. Quick time events, such as the ones found in *Resident Evil 4*, are a type of context mechanics since pressing two buttons once in a timely manner or repeatedly as fast as possible can be equivalent to different actions, from dodging, to running, to attacking. Point-and-click interfaces are also a prime example of contextual mechanics, since a click will mean different things depending on the context—more on these mechanics in the sections below.

As a side note, the level of abstraction refers to how detailed the system of the game is, independently of its visual representation. Adventure games provide a clear example: the lavish backgrounds of *Myst* were beautifully rendered graphics when the game was released, but the functionality of the world was much more limited than the one in *King’s Quest: Quest for the Crown* with its heavily pixelated EGA graphics and limited color palette.

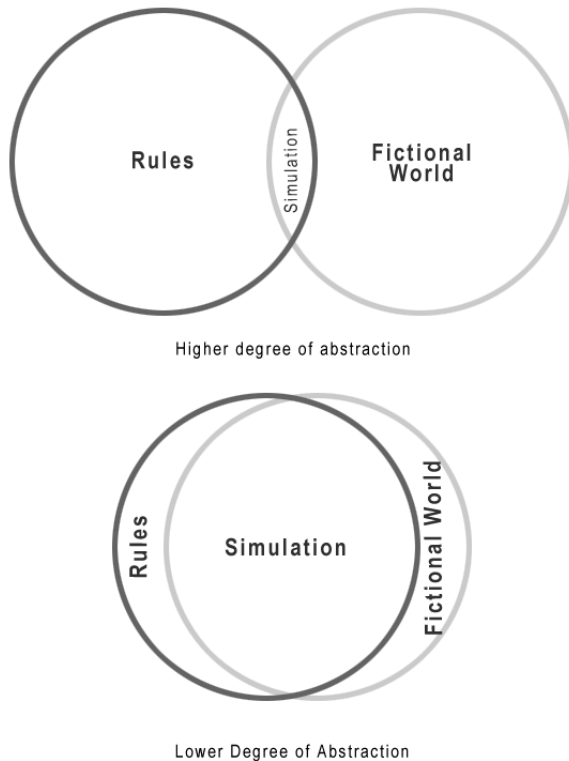


Figure 2: Level of Abstraction is the overlap between the rules and the fictional world.

4. ADVENTURE GAMES: DEFINITION

The term “adventure games” refers to a particular set of games that have their origins in the text game *Adventure*. This does not mean that adventure games are text-only: the earliest games inspired by *Adventure* are text adventures (*Zork: The Great Underground Empire*, *The Hitchhiker’s Guide to the Galaxy*), but later the graphical capabilities of computers have given way to graphic adventure games (*King’s Quest*, *The Secret of Monkey Island*). With the generalization of the mouse as an input device, point-and-click has become the standard interface of adventure games, such as *Myst* or *Machinarium* more recently. These games share five features [5], which when appearing together set them apart from other games: Adventure games are story-driven videogames, which encourage exploration and puzzle solving and

always have at least one player character. The basic interaction of adventure games is based on object manipulation and spatial navigation. Their challenges usually appear in the form of concatenated puzzles, which are integrated in the fictional world. Let us unpack what each of these features means.

4.1 Story-driven

Adventure games present strong examples of how a story can drive the game experience. Adventure games are one type of story-driven games, along with computer role-playing games (e.g. *Mass Effect*, *Final Fantasy VII*), or some action-adventure games (*The Legend of Zelda*, *Grand Theft Auto: Vice City*).

In story-driven games, there is a set of story events that take place as the player advances in the game. These events are usually pre-determined, and constitute an “ideal walkthrough.” The player is usually expected to make progress in the game by following specific steps, or at least to reach certain milestones, so that a concrete state of affairs is reached and the story unfolds along with the gameplay.

The key difference between story-driven games, such as *The Secret of Monkey Island* or *Mass Effect*, and games with in which progress is framed as a story, such as *Rock Band* or *Super Street Fighter IV* is that in the first case the manipulable entities and actions in the game are also the agents and events in the story. There must be a substantial overlap between what the characters do as agents in the story and as entities as part of the game, as well as between the values and properties of the objects in the game and in the game world. Adventure games always have a story inextricable from gameplay—advancing in the game means advancing in the story, because each challenge and its solution constitute an event in the story of the game. The story thus dictates what the player has to do in order to traverse the game successfully.

4.2 Puzzle-solving

Adventure games are not competitive; physical skills and reflexes are not necessary to play the game. Through the game, the player must solve a series of problems in the game world. Each problem is a puzzle that is integrated in the environment, and solving it constitutes an event in the story of the game. The puzzles in the game are interrelated, so that solving one may open up a new one, or facilitate solving another.

In the case of adventure games, there is normally only one correct sequence of actions that will provide the solution; if there is more than one way to solve the puzzle, it has to be designed into the game. These puzzles are usually interwoven, so that by solving a puzzle the player obtains an object or information to solve another one, or a new puzzle or set of puzzles appears.

Since adventure games are story-driven, their puzzles are integrated in the world of the game. The entities involved in the puzzle are also entities in the world; solving the puzzle means achieving a specific state of affairs in the fictional world.

4.3 Player Character

Adventure games always have at least one character that the player controls. The player character is the main game entity which provides the point of view and carries out the player’s commands. This character also defines how the player interacts with the world, since she cannot affect it directly. Typically, adventure games have only one player character, although some games feature different characters that the player can control through the game.

4.4 Object Manipulation

The interaction with the world is mediated through the player character, which is the entity that manipulates the objects and deals with the non-player characters. The player's input is a command [13], which directs the player character to do something in the world. Object manipulation can involve one object (e.g. "pick up lamp), or two objects at the same time, for example "use wire with outlet" (*Broken Sword: The Shadow of the Templars*). It can also involve an object and a character, e.g. "give hot dog to boy" (*Gabriel Knight: Sins of the Fathers*) or "use fake barf with Harold" (*Day of the Tentacle*).

As I argued elsewhere [4], the command input has evolved through time, from typing a verb + object command, to choosing the verb from a menu and then clicking on the object that is going to be manipulated, to choosing an icon that stands for a verb or several verbs and then the object, to clicking on the object that has a specific verb associated with it. Thus the cycle structure has gone from being explicit (in the text commands) to context mechanics (by clicking on an object that is associated with a specific action). The evolution of the command input along with the general interface design, shows how adventure game developers pursued a direct manipulation model of interaction [17], so that the player can participate in the simulation of the fictional world. This has taken the form of gestural interfaces in some current adventure games, where the commands are have become equivalent to moving the controller in a specific way, imitating the gesture to perform the required action (e.g. *Indigo Prophecy*, or some puzzles in the Nintendo DS game *Hotel Dusk: Room 215*).

4.5 Exploration of Space and Action

The last distinctive feature of the adventure game genre is that it encourages exploration of the game space through interaction. This means that part of the gameplay is based on probing the space of possibility in the game, by investigating the space and interacting with the objects and characters.

Exploration starts by navigating the space, moving from location to location, and examining all the objects available. The affordances of the space of adventure games are exploration and problem-solving, so that the source of the challenges mostly consists of the environment itself.

The exploration of the space and objects also leads to experimenting with the possible actions in the world, thus probing the possibility space of the game. Reaching a specific state of affairs that will solve a puzzle requires the player to manipulate the objects in the world, investigating their properties and affordances, to then figure out how to those objects can help solve the puzzles at hand. Thus typical actions are picking up objects, open or close them, combining items, or observing reactions of other characters when they are offered an object. In *Space Quest IV: Roger Wilco and the Time Rippers*, for example, the player may smell and taste any available item in the simulation, even if most times it is not an action that helps finishing the game.

During their historical evolution, adventure games have changed the way exploration takes place in them, in order to make clearer to the player what could and could not be done in the game world. It is not a matter of usability alone—too much freedom can be intimidating for some players, particularly novice ones, who need to be guided through the world.

5. THE SIMULATION OF ADVENTURE GAMES

Although Aarseth repeatedly argues that adventure games are not simulations [1,2], when the definition introduced above is applied to adventure games, we find that some aspects of the fictional world are indeed simulated while others are represented. The simulation and the actions that could take place in this fictional world create the space of possibility in adventure games. Players can roam the dungeons of the Great Underground Empire of *Zork*, meet pirates at the Scumm Bar in *The Secret of Monkey Island*, or find the way to escape the diner where the player character just murdered someone in *Indigo Prophecy*.

It is the simulative aspects that provide adventure games with one of their defining characteristics, the exploration of action and space. If the world were not simulated, players could not try picking things up, opening or closing doors, or talk to other characters. The fact that there is a sequence of events that must take place in a specific order, by solving the puzzles in a specific way, is different from the fictional world not being simulated. Adventure games may only feature one series of steps that must be followed in order to complete the game; however, this does not invalidate the existence of the simulation. The player needs to experiment in the simulation in order to learn the right thing to do. The player becomes like an actor without a script, trying things out in the simulation and combining different actions to figure out what the correct action is. When an action or series of actions work out, the player obtains the solution to the puzzle, which is also an event in the story of the game. The design brings about a different type of gameplay, which characterizes the genre; as we will see below, the focus on problem-solving is probably why the design conventions of adventure games have evolved differently from other genres.

6. THE SPECTRUM OF LEVELS OF ABSTRACTION IN ADVENTURE GAMES

The necessity of selecting the elements of the game system purposely was obvious for early developers of adventure games, such as some of the interactive fiction pioneers. As David Lebling and Mark Blank, two of the original implementors of *Zork*, noted: "Obviously, no small computer program can encompass the entire universe. What it can do, however, is simulate enough of the universe to appear more intelligent than it really is. This is a successful strategy only because CFS games [Computer Fantasy Simulations] are goal-directed. As a consequence, most players try to do only a small subset of the things they might choose to do with an object if they really had one in their possession." [12]

That is, given the limitations of the computer to recreate a simulated world, they only created a subset of it, establishing what would be possible. Lebling et al. argue that the gaps in the simulation can be disguised by giving a goal to the player—in the case of *Zork*, finding all the treasure in the dungeon and putting it into a trophy case. Murray refers to this goal-oriented design as "scripting the interactor" [14], since it is the fictional world of the game, which evokes treasure raiding in *Dungeons and Dragons* campaigns. The guidance is not hard-coded in the design, but evoked by the fictional world, which provides the goals of the game.

Establishing clear goals, however, does not prevent the player from attempting to do other things that may not have anything to do with those goals, especially if they are not clear to the player.

Since adventure games require the player to experiment and explore the game, poking and probing the simulation is expected, even if she is given a specific goal. Experimenting and pushing the boundaries of the world is also one of the attractions of simulated worlds, so exposing the limitations of the simulation is actually part of how the game is played.

In order to cover up those limitations, designers have to anticipate what players may try to do, even more so in the case of text adventure games, where players can in theory type in any command. The possible actions are associated to every object in the game. A simulated wooden chair would support sitting on it, standing on it, or pushing it around (or not, if it is really heavy), but it cannot be put inside a pocket or eaten. Other more specific behaviors, such as taking it apart and turning it into firewood should also be predicted by the rules.

Designing such a highly detailed simulation requires a lot of time and effort on the game designer's part, and it is not particularly efficient—there will always be something that a player tries that is not anticipated by the rules. One obvious strategy to prevent implementing a whole world is selecting what aspects of the fictional world will be most relevant to gameplay. If the player does not need to take the chair apart, but the player tries it anyways, the player will obtain a response such as "Dismantling the chair would hardly be of help here." The designer can script the interactor through by using text: a shelf can invite the player to reach the box on it with the following description: "There is a red box on the shelf, but it is too far for you to reach." The chair should support the player standing on it, so that when the chair is near the shelf and stands on it, the player can get access to the red box. In this context, trying to dismantle the chair hardly makes any sense, but standing on it becomes a useful action.

The opposite side of the spectrum of levels of abstraction is occupied by point-and-click adventure games with no menu for the different actions. *Myst* is a prototypical example of how contextual mechanics and a world with lower fidelity guide the player through the simulation. At the beginning of the game, the player is literally dropped on an island, and has to figure out steadily what the goal of the game is. By roaming around, reading books and notes, and operating different technologies, the player learns about the different Ages and the events that took place in them, as well as how each technology works, although the goal is not explicit. The player is presented with a dilemma: two brothers have been made prisoner in two magic books, so the player must decide which of the two brothers should be released by bringing back pages of their books; the player may eventually decide to not release either of them. The overall goal becomes finding the pages; how to find them, and what the intermediate goals are depends on the player exploring the island and the different ages.

The possible actions of *Myst* are very limited—the player interacts with the world through contextual mechanics. This way, the player does not have to guess what the correct action is, since there are few objects that can be interacted with, and each object only does one thing (turn, flip, open/close, switch on/off). There is no inventory; the player character can only carry one object at a time, one book page. This forces the player to choose between two pages in each area (Age) to bring back to the island of *Myst*, turning the choice posed by the goal of the game into a core mechanic. Restricting the type of objects that the player can pick up makes clear that the player does not have to hoard objects in the world, but rather figure out how the strange machinery works.

On the other end of the spectrum, *Zork* represents its world textually, creating a world where the interactions and the

behaviors of objects are more detailed. Most objects in the game could be picked up and dropped, the player can try to burn objects and then put out the fire with water to see what happens. The simulation of the objects in the world is more fine-grained, and the mechanics of the game also offer more nuanced actions to the player. In short, *Zork* creates a more fine-grained simulation, while in the simulation of *Myst* the rule system is more simplified, thus creating a more limited simulation and offering the player less variety in the available actions (Figure 3).

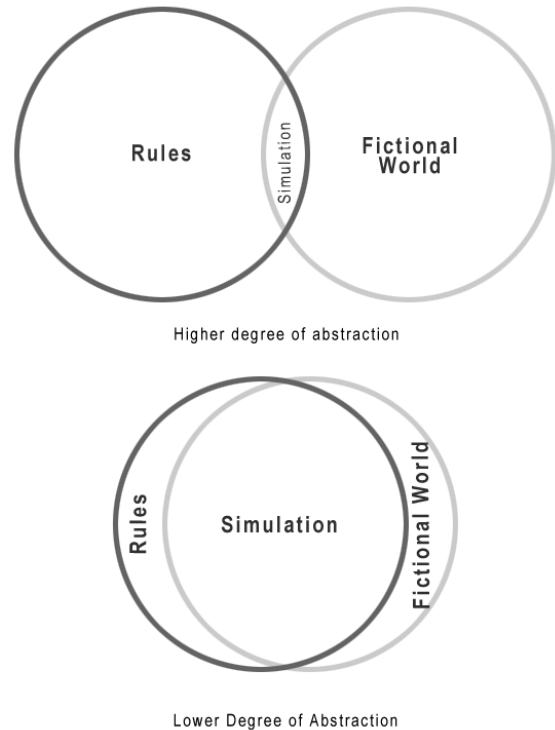


Figure 3: Levels of abstraction of *Zork* vs. *Myst*.

The decreased nuance in the simulations of adventure games also means that the events of the story have less room in the simulation, so they have to be represented. If there are fewer possible actions for the player, but there is still a story that must unfold, the events have to take place in the representations (again, cut-scenes, descriptions), curtailing the player's agency over the events.

A clear example of how the simulation has changed over time can be observed if we compare *Sam and Max Hit The Road*, a menu-driven point-and-click adventure game, with the more recent series of episodic games based on the same characters, *Sam and Max: Season One*. The wild bunny Max accompanies and collaborates with Sam, the player character, through the game in both cases. In *Sam & Max Hit the Road*, as the player explores the world by controlling Sam, Max will start doing things on his own, from jumping on beds, to ask insistently to go to the bathroom, to trying to wrench an axe from a mannequin in a carnival ride. Instead of ordering Max what to do, Sam uses Max with different objects and other characters—Max is actually part of the inventory; "using" Max means unleashing him to do something unexpected.

Conversely, in *Sam and Max: Season One*, Max mainly functions as a provider of information (provides the player with hints about what to do), and comments on the recent events. The bunny does not have independent behaviors any more nor reacts to the environment while the player is doing something else, but rather waits for the player to interact with him. Max cannot be “used,” but rather the player may cue him to do something through dialogue and depending on the context. This is a new type of contextual mechanic, where depending on the context, Max will carry out certain actions depending on what the story needs.

7. Levels of Abstraction in Game Mechanics

The lesser nuance in the simulation of the game has also led to a restriction in the number of actions available to the player, which is reflected by the changes in the interface of adventure games. The interface of adventure games has also transformed through time, going from having to type the commands in natural language, which allowed for a relatively large range of actions, to clicking on an object to perform an action, which the game itself selects depending on the context. As I have argued elsewhere [4], this change has taken place in order to encourage direct manipulation [17]: instead of using the command line as an input, which may or may not return a successful output (the action is possible in the simulation), direct manipulation provides continuous and immediate feedback on the actions of the player—the player moves the mouse and can locate what objects can be operated; if an action is not possible, clicking will not have any effect on the simulation. Thus, the player does not have to fight the interface in order to participate in the world, but concentrate on solving the puzzles.

Context mechanics make the interaction less error-prone—the program does not have to return an error message whenever she performs an unsuccessful attempt to do something, because the action is what the player is supposed to do. Immediate feedback and reduction of error messages are qualities that improve the usability of computer programs [17]. This is a way to address one of the problems posed by Lebling, Blank, and Anderson when they referred to how to implement the simulation efficiently—the player should be solving the problem in the game rather than the program [12]. However, being more efficient in the interaction can also get in the way of experimenting and exploring the world, since trial and error still constitutes a basic strategy to learn the rules of the simulation. Improving the usability of the program affects the range of actions, since it can prevent the player from trying out actions that help her learn more about the fictional world.

Adventure games can have a relatively large list of verbs: text adventure games could have relatively powerful parsers, which would accept a good range of verbs. The original *Zork*, for example, recognizes “71 distinct ‘actions’” [12]. Graphic adventures, such as *King’s Quest V*, present the player with five commands (go to, look at, pick up, talk, use (inventory item), examine (inventory)), plus the save / load options and audio/visual settings of the game. This given list of actions is relatively standard, and was devised as a way to prevent word-hunting, i.e. finding the right phrase to carry out an action, which was typical of text adventure games. The player could still find some items that did not provide much information, or where a specific action was not possible (the player character would respond “I don’t think that’s a good idea” for example). More recent point-and-click games, such as *Sam & Max: Season One*, have done away with the verb list and use context mechanics like the ones found in

Myst. The player clicks on an object, and the game chooses for the appropriate action—clicking on a character is equivalent to choosing “talk to”; clicking on a closed door will be interpreted as trying to open it. Therefore, very object only has one action associated with it, simplifying both the design and the interaction with the game.

A significant attempt at revamping the adventure game interface can be found in *Indigo Prophecy* and the recent *Heavy Rain*. In these games, the set of actions depends on the immediate surroundings of the player character. To choose the action, the player has to move the controller stick in a specific way, usually imitating the physical movement that the player character has to carry out in the simulation. There is a misleading nuance in the actions—the player is imitating the gesture, but how accurately one does it does not usually matter. In the case of *Indigo Prophecy*, some actions require certain basic coordination skills, so that the player must move the two analog sticks of a controller in a specific pattern and within a limited amount of time. These actions become quick time events, and are used for action scenes, such as fights or chases. Although there seems to be more nuance in the actions, in this second case there is a disconnect between the movements of the controller and those of the player-controlled character. For instance, the player may have to dodge and then punch to the right, but the sequence is randomly set, so that the movement the player has to carry out is up-down-left. This is a difficulty metaphor [8], but it also seems to disconnect the player from the directly operating within the simulation of the game.

8. CASE STUDY: THE MICROWAVE

The level of abstraction of a game thus affects both the range of responses to player interaction and how nuanced the interaction with the objects is. A meticulous simulation means that the player can try more actions involving that object, and that those actions will very likely receive a response, even if the action is not part of solving a puzzle. On the other hand, when the simulation is restricted to essential actions, it means that an object will only respond to the specific action that makes it relevant to solving a puzzle.

Comparing the microwave ovens in *The Lurking Horror* (a text adventure) and *Maniac Mansion* (graphic adventure) provides some perspective on how different levels of abstraction generate different functionalities. A microwave is not a versatile object that can be used repeatedly in a game, but its inclusion makes the fictional world richer. In *The Lurking Horror* all the buttons of the oven are operational: it can be opened/closed, the player can put food inside to cook it (Chinese food in this case). In order to cook, the player has to press the right buttons: enter the time using the numeric pad (press 5, 0, 0 which means five minutes), set the level (high) and then start the oven. On the other hand, the microwave oven in *Maniac Mansion* can be opened/closed and turned on/off. The player character will refuse to use it if it is empty, and there are only two objects that can be used in the oven.¹

The functionality of the microwave in *Maniac Mansion* is less fine-grained than the one in *The Lurking Horror*, since there is a smaller range of actions that apply to the specific object. The

¹ One is a hamster, who obviously does not survive, and the other is a glass jar full of radioactive water. Either of these actions kills the player character.

more restricted the functionality of an object is, the less verbs can be applied successfully to it. This difference between levels of abstraction presents an interesting dilemma. Being able to do more things with an object means that it will be more difficult for the player to find out what to do with it, so the player will be more prone to error. On the other hand, a narrower range of actions means that the simulation will also be less rich, making the world less interesting to explore, more restricted and more obviously artificial.

Heating food in the microwave is a key puzzle to advance in *The Lurking Horror*, but since it requires several actions related to the microwave, it actually turns out to be quite complicated. The player is forced to think about every step that needs to be done, when in real life it is something that she may do without thinking.

Thus, choosing the right level of abstraction is a fundamental tool to establish how much room the player has to fail and experiment. A less detailed simulation is more likely to require a specific series of steps, which constitute a particular sequence of events, because there is less room for behaviors that deviate from the one expected from the player. Therefore, restricting the number of objects to interact with and the possible actions associated with them facilitates generating the story, simply because there is not much else left for the player to do. On the other hand, objects that have a detailed functionality and support a wider range of actions give more leeway to the player to explore the world and experiment, but also make it more likely for the player to carry out actions that will not allow her to advance in the game. These two extremes characterize the trade-offs that adventure games design must make when creating the simulation of the world.

9. HOW SHIFTING LEVELS OF ABSTRACTION CHANGED ADVENTURE GAMES

Specifying the level of abstraction mainly determines how players explore the simulation, and what events of the story are simulated and which ones are represented. In more nuanced simulated worlds, such as the ones found in most text adventures / interactive fiction or some early graphic adventures, players explore the boundaries and limits of the simulation. There is more information, not because there are necessarily more objects, but because there are more things that can be done with them. The main question the player asks throughout a text adventure game is “what can I do with this object?” or “how can I interact with this character?”

For example, in *The Lurking Horror*, one of the key inventory items is a bottle of Classic Coke, which helps the player character stay awake and alert by drinking it. The more nuanced simulation allows the player to pour Coke anywhere, although that is not a good move. Without caffeine, the player character falls asleep and the game is over. The simulation, however, allows the player to waste the contents of the bottle, and thus arrive to a state of affairs in the simulation in which it is impossible to complete the sequence of events in the story.

On the other hand, finding out what is possible in the world can be a source of pleasure for players too, particularly if the player tries to do something particularly off-the-wall. For instance, there is a bucket in one of the rooms in *Zork*, and if the player chooses to kick it, the player character will “die” and become a spirit in the world, so the player must find her body in the game in order to

revive. “Kicking the bucket” is a joke which also has consequences in the simulation of the game.

The pleasures of exploration in text adventure games usually reside in probing the depth of the world. This is of course easier for players who are more experienced, who identify the goals of the game quickly, and who are already familiar with the repertoire of actions that the parser of the game will recognize. New players, however, usually find the command line intimidating, and figuring out the effective commands an ordeal, particularly now that most games use Graphic User Input. Thus a less detailed simulation is a way to introduce new players to the genre, or encourage players who do not wish to fight the parser of the game.

The kind of exploration in adventure games has therefore shifted to other types of activities. Since the range of actions has been reduced, exploration starts by finding what objects and characters are simulated, and then figuring out what they do. The main questions the player asks in a lower-fidelity simulation are “what does this object do?” and “what information can I obtain from this character?” By learning the specific role of each object in the game, the player focuses on establishing relationships between the objects and what they do.

Context mechanics also provide consistency to the world, since each object will behave the same way whenever it is clicked on, creating expectations about what to expect from an object. For example, in *Machinarium*, the player character is a robot, and must disguise itself as one of the guardians in order to enter the city. The guard is tall, and wears a blue cone-shaped hat with a bulb on top. The player character is short, but can extend up its body, which allows it to reach objects that are high up and makes him look taller. A pile of striped traffic cones, a street lamp and a bucket of white paint are in the scene. To solve the puzzle, the player has to find the relationship between the objects—using an object with the bucket of paint turns it white, which is not the right color. The street lamp has a bulb. Clicking on the first traffic cone moves it to the inventory; continuing clicking reveals more cones, until a small bucket of deep blue paint appears under the last one. It is too small to dip a traffic cone in it, but it can be mixed with the bucket of white paint, resulting in the exact color of the guard's hat. Every action provides the player with information, none of them is an immediate solution to the problem, but gets the player a tad closer to it. Contextual interfaces emphasize discovering what objects *do*, instead of what can *potentially* be done.

What “depth” means in the context of the simulation has also changed over time. In more detailed simulations it means more possible actions, some of which may not be necessary to complete the game, or may make it impossible to complete the game as was the case in *The Lurking Horror*. Conversely, depth in lower-fidelity simulations is usually associated with representation, where the player can learn more about the world through cut-scenes or examining objects, as is the case with the books that tell the player about the history of *Myst*. The focus shifts from more complex behaviors, which make it easier to carry out actions that do not advance the story, to essential interactions and behaviors, which let the player focus on advancing the story.

10. CONCLUSION

Adventure games demonstrate how selecting a specific level of abstraction transforms and extends the design palette of a genre. Choosing a specific level of can even generate new genres. For instance, it can be argued that hidden object games derive from

finding the active objects in an adventure game; escape the room games can be understood as adventure games where all the puzzles are exclusively of the lock-and-key type. The examples above also evidence that the level of abstraction may appeal to different audiences—more complex simulations may attract more seasoned players, particularly if they are familiar with the conventions of a specific genre, and bring a set of expectations about what can be done in the simulation. Lower fidelity simulations, on the other hand, introduce players to the conventions of the genre, mainly by preventing actions which do not get the player closer to the goal of the game. This is true of adventure games, but may also apply to other videogame genres.

Since adventure games have pursued lower fidelity simulations through their design history, it would be interesting to explore how they compare to other genres. Does less nuance appeal to newer players or casual players? If so, it may provide a hint on why certain genres, like first person shooters or real-time strategy games, remain hardcore player genres. It also gives us a hint that game designers may appeal to new audiences by creating simulated worlds that focus on the most essential aspects of the system.

11. ACKNOWLEDGMENTS

Thanks to Jason Begy, Todd Harper, Konstantin Mitgutsch and Mia Consalvo for their feedback on this paper.

12. REFERENCES

- [1] Aarseth, E.J. 2004. Genre Trouble: *Narrativism And The Art Of Simulation. First Person : New Media As Story, Performance, And Game*. N. Wardrip-Fruin and P. Harrigan, eds. MIT Press. xiii, 331 p.
- [2] Aarseth, E.J. 2005. Doors And Perception: Fiction vs. Simulation in Games. *Digital Arts and Culture* (Copenhagen, 2005).
- [3] Crawford, C. 2005. *Chris Crawford On Game Design*. New Riders.
- [4] Fernández Vara, C. 2008. Shaping Player Experience in Adventure Games: History of the Adventure Game Interface. *Extending Experiences*. A. Fernandez et al., eds. Lapland University Press.
- [5] Fernández Vara, C. 2009. The Tribulations Of Adventure Games: Integrating Story Into Simulation Through Performance. Georgia Institute of Technology.
- [6] Frasca, G. 2001. SIMULATION 101: Simulation versus Representation. <http://www.ludology.org/articles/sim1/simulation101.html>.
- [7] Gingold, C. 2003. Miniature Gardens & Magic Crayons: Games, Spaces, & Worlds. Georgia Institute of Technology.
- [8] Juul, J. 2005. *Half-Real: Video Games between Real Rules and Fictional Worlds*. MIT Press.
- [9] Juul, J. 2007. A Certain Level of Abstraction. *Situated Play: DiGRA 2007 Conference Proceedings* (Tokyo, 2007)
- [10] Klastrup, L. 2003. A Poetics of Virtual Worlds. *Digital Arts and Culture* (Melbourne, 2003).
- [11] Klevjer, R. 2002. In Defence of Cutscenes. *Computer Games and Digital Cultures Conference Proceedings* (Tampere, 2002).
- [12] Lebling, P.D. et al. 1979. Zork: A Computerized Fantasy Simulation Game. *IEEE Computer*. 12, 4 (1979), 51-59.
- [13] Montfort, N. 2003. *Twisty Little Passages: An Approach to Interactive Fiction*. MIT Press.
- [14] Murray, J.H. 2001. *Hamlet on the Holodeck : the Future of Narrative in Cyberspace*. The MIT Press.
- [15] Pearce, C. 2002. Sims, BattleBots, Cellular Automata God and Go: A Conversation with Will Wright. *Game Studies*. 2, 1 (2002).
- [16] Prince, G. 2003. *Dictionary of Narratology (Revised Edition)*. University of Nebraska Press.
- [17] Shneiderman, B. 2003. Direct Manipulation: A Step Beyond Programming Languages. *The new media reader*. N. Wardrip-Fruin and N. Montfort, eds. MIT Press. 486-499.
- [18] Sicart, M. 2008. Defining Game Mechanics. *Game Studies*. 8, 2 (2008).