A Planning-Based Approach for Generating Narrative Events in Video Games

by

Le Fang

Bachelor of Science, Simon Fraser University, 2013

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
School of Interactive Arts and Technology
Faculty of Communication, Art and Technology

SIMON FRASER UNIVERSITY
Fall 2017

Approval

Name:	**Le Fang**
Degree:	**Master of Science**
Title:	**A Planning-Based Approach for Generating Narrative Events in Video Games**
Examining Committee:	**Chair: Wolfgang Stuerzlinger** Professor
	Jim Bizzocchi Senior Supervisor Associate Professor
	Steve DiPaola Supervisor Professor
	Dave Fracchia External Examiner Professor of Professional Practice Master of Digital Media Program Centre for Digital Media
Date Defended/Approved:	**October 10, 2017**

Abstract

This project is an attempt to find a solution for the replayability problem in video games from a narrative perspective. The project's goal is to improve replayability in narrative-focused video games by providing variations in the narrative experience. The outcome of this project is a game prototype that takes the form of a detective adventure game, in which the player needs to find the murderer. This game prototype uses a combination of scripted narrative and generative narrative with computational approaches. The generative narrative part uses a probability-based random event scheduler combined with STRIPS planning to generate narrative components in order to provide variety in narrative. The outcome of this project is a first step to revealing the possibilities of building narrative-focused games with some generative narrative aspects to enhance variety in narrative and thus enhance replayability.

Keywords: Interactive Narrative; Generative Narrative; Computer Planning; Video Game; Artificial Intelligence

This work is dedicated to my maternal grandfather Hua Yang,

whom passed away during my studies abroad.

Acknowledgements

First of all, I would like to express my sincere gratitude to my senior supervisor Associate Professor Jim Bizzocchi. This thesis would not be possible without his guidance and support throughout my study. He has offered me a great deal of help in everything throughout this research. It was a great time for me to work with him. Thank you, Jim.

Also, I would like to give my grateful thanks to my supervisor Professor Steve Dipaola. He provided me a great deal of support during the development of the game prototype, as well as the exploration into a lot of interesting ideas. And it was a great fun for me to listen to him talking about his projects, ideas, and everything.

Next, I should express my thanks to my friend, Xie Zhou. I have learned a lot of invaluable things from lengthy and interesting discussions with him on topics about video games and everything over the past 10 years. These conversations have given me a great deal of useful information and inspirations, despite the fact that we constantly disagree with each other and he constantly mocks me in every way possible.

I would like to thank Xujian Tang, Xin Gao, and my other friends in Canada, in China and in other corners around the world as well. Their support and encouragement are very essential to me, I could not make this far without their help.

And last, I would like to thank my mother for her selfless support and love. Thank you, mom.

Table of Contents

List of Figures

Chapter 1.
Introduction

1.1. Early Days of Narrative in Video Games and the First Adventure Game

The video game is a very new media form compare to others - it has only gone through 44 years of evolution since Pong. Regardless the rather short age, video games have evolved from simple white dots and dashes with minimal gameplay rules, to amazingly detailed and believable game world rendered in high resolution, populated by all sorts of complex objects and regulated by complex yet fun gameplay rules. One can hardly believe such significant evolution only took less than 50 years to happen.

1.1.1. Early Days of Video Games

The early video games did not have too much content to offer, especially in terms of presentation. Most of them can only focus on the fundamental function of the video game: “play”, a behavior that includes all possible activities within a more regulated structure (Zimmerman, 2004). In this case, “play” a game is meant to be a ludic activity, to be “fun”. Many early video games, especially arcade games and games on early video game consoles, have little or no explanation of narrative context to justify the content of the game. These games have simplistic graphics, sound effect, and gameplay rules, but highly fun and addictive for their time. One of such examples is the world-famous arcade game *Space Invaders*, which became a social phenomenon back in the 1970s and early 1980s. Taito, the company released this game, have installed 100,000 Space Invader arcades within the first few months of its release in Japan, and these coin operated machines even made Bank of Japan to increase the production of 100 yen coins. This game has grossed 2 billion dollars by 1982 (Executive, 1982). These early arcade style video games were also well received when they ported to game consoles such as the Atari 2600, and much later, the Family Computer (Nintendo Entertainment System in the US), so they can be played at home.

Due to the restriction of technology and lack of methodologies in production, the early video games are often very simplistic. There was usually not enough space for

narrative back then. Many video games back in those days have chosen to leave textual narrative out of the game completely, although some developers made some effort in narrative by stuffing a background story somewhere outside the game, such as in user's manual or magazine, to justify actions in the game, and give players something to read and think about while playing. For example, *Legend of Zelda* did this in its user's manual (Nintendo, 1987). This way of embedding narrative is far from interactive narrative. The narrative material is completely out of the video game, hence not interactive at all.

Back then, dedicated gaming devices specialized in what were then advanced graphics and fast animation. On the other hand, the computer games are a somewhat different story. Early computers were not built to handle graphics, as a result, the first computer games did not have proper graphical presentation at all. However, even under a situation which the beauty of graphics cannot be utilized, the early computer games blooms in a form which does not necessarily require the existence of graphics and animation.

1.1.2. The First Adventure Game and Its Descendants

The first computer games were the digital interactive narrative at the purist form: text interface only, nothing else. Let us take a look at *Adventure* (or *Colossal Cave Adventure*, 1976/1977) as an example. *Adventure* is the first computer adventure game in existence, and the very first video game which features the freedom of space navigation and exploration (Lessard, 2012). This game features a series of interconnected locations. All of these locations are described by texts, including the surrounding environment of current location, objects of interest, the status of the player, and the hints for possible dangers or treasures. The player interacts with the game by typing text commands, they are allowed to use several key words to navigate in the maze (such as "n" for go north, or "d" for go down), as well as interact with objects and points of interest in the environment (Wolf, 2008).

This interaction design is not very intuitive by today's standards, but it was a huge success back then, because it was a totally new way of interaction. The text command based nature of this interface design has successfully created an illusion which the player is actually "talking" to the computer, boosts a unique interact experience. What makes things more interesting is the fact that the early developers of

adventure games were also aware of this pseudo-dialog nature, and implemented reactions for a variety of interesting unsupported words and combinations of invalid commands (such as “pick up house”), making the “talking to computer” illusion a more satisfactory and believable experience. Players would also eager to attempt a lot of different combination of words in order to find all these commands, only to see if the program can respond, for a moment of good time (Wolf, 2008). On the other hand, the text based interface makes these text-based adventure games very narrative-oriented. They have very rich content in terms of narrative even though it has no graphic representation at all.

Although the pure text interface is somewhat complex and not as intuitive as the more streamlined point-and-click interface which comes years later, it did not stop the passion of players to play and love this game. The success of *Adventure* has well proved that narrative can become a central focus of a video game, and interactive narrative can be “played” and enjoyed in a video game format. As a matter of fact, *Adventure* has motivated a lot of developers to devote themselves into the development of adventure games.

The later games have evolved a lot from the original *Adventure*. *Zork* has more sophisticated overall game design and puzzles, as well as a more complicated system. Sierra’s great hit *King’s Quest* introduced EGA standard powered colorful graphical representation. *Déjà Vu* developed ICOM Simulations was the first game to have a point and click interface. A series of adventure games such as *Maniac Mansion, Indiana Jones and the Last Crusade* and *The Secret of Monkey Island* developed by LucasArts introduced the “No dying” concept (the player character would never reach a “die” or “game over” state), animated characters and backgrounds as well as more refined gameplay and narrative design. *I have no mouth and I must scream* showed the possibility of adaptation and inspiration from traditional literature. *Seventh Guest* introduced Full Motion Video (FMV) into video games. And last but not least, *Myst* redefined the genre by making a lot of successful attempts in terms of graphics, sound, gameplay and puzzle design as well as the narrative design. (Moss, 2011)

1.2. Gameplay-Focused Video Games and Narrative-Focused Video Games

There was a huge debate among digital media scholars around whether the video games should be focus on being “fun”, or should be used as a new way to tell stories, the two sides of the debates are known as ludologists - which support that video games should be fun and story is not a focus - and narratologists - which think video games should be used as an interactive narrative device (Bizzocchi, 2006). This kind of debate was not only common in academic domains, it was - and it still is, although different from the original debate - a huge topic among players, video game medias and video game developers.

Such debate often ends up in nowhere since this is not a simple back/white or yes/no question. What we can observe is that there are only a few video games in a small set of specific genres are pure in either category: a handful puzzle games (e.g.: the original Tetris) and some board / card games have no story; “visual novel” and “interactive drama” have minimal interactive options which can hardly be called “gameplay”. Other than these, the majority of video games, even many action adventure, shoot’em up and platformer games in the 1980s, are products of a combination of both gameplay and narrative. Thus we can change the question from a simple yes/no one to a “how much” one: how much gameplay is in a specific game, and how much narrative is there? (Zimmerman, 2004)

Based on this question, I would like to present my observations on video games between the balance of gameplay and narrative. These games often present in two forms: games that privilege gameplay, and games that are more focused on interactive storytelling. From my experience, game genres rooted in arcade as well as in non-narrative table top games, such as action games (including shoot’em up, on-rail shooter, first/third person shooters, beat’em up, platformer and fighting games), simulation games and board games are the main contributors to the first form; and those genres developed from table top adventure/role-playing games and their digital forms in early ages, such as adventure games, and later, majority of role-playing games (mainly Japanese RPGs like *Dragon Quest* and *Final Fantasy*, and Western RPGs like *Baldur’s Gate* and *Fallout*), are the main contributors to the second form. This observation is similar to Juul’s

categorization on games of progression and games emergence (Juul, 2005), I will explore Juul's idea in a later chapter.

In the early years, video games were more extreme in regard to this balance, and showed significant differences between the two kinds. Games which fall in the first form often feature fast and stimulating gameplay with minimal storytelling, and games belong to the later form often progress at a slow pace, the actual gameplay itself is usually significantly less stimulating as games in the first category. Later, these differences between the two categories started to blur. The evolution of technology and video game design methodologies make it possible for video games to have both complex narrative presentation and action-oriented gameplay, crossbreed genres such as action-adventure (e.g.: *Resident Evil, Silent Hill*) and action role-playing (e.g.: *Secret of Mana, Diablo*) also start to exist during this period.

Adventure mentioned in the previous section has two game genres as its direct descendants: adventure games and role-playing games. Many role-playing games (RPGs) are heavily narrative-driven and exploration centered as the original *Adventure*, while they also differ themselves from the original *Adventure* by having complex systems built around character progressing and combat to enhance the gameplay experience. On the other hand, the adventure games have stayed more faithful to its origin. The gameplay methodology of adventure games remains largely unchanged throughout its existence, although the presentation and interface have changed a lot. Player still needs to navigate through various spaces, collect items by investigating the surroundings and talking to non-player characters, and finally solve puzzles in order to progress. The only thing keeps this genre alive is its beautiful visual-audio presentation and its ability to represent and tell a complex story in a most expressive way through digital media. Narrative is the core of this genre (Lessard, 2012).

1.3. Addressing Replayability in Gameplay-Focused Video Games

Among the five properties of digital media recognized by Ryan, one important trait of digital media is its ability to be repeatedly experienced with somewhat different progressing in the digital text without making modifications to the media itself (Ryan, 2004). As a form of digital media, video games certainly should be able to be played

repeatedly for a different experience. However, the problem is whether these video games worth the effort to be revisited.

In certain game genres, for example, simulation games, real-time strategy games, and puzzle games such as *Tetris*, replayability is barely a problem, since the player is responsible for creating a considerable part of content of the game using tools and components given by the game. The situations across different game sessions are often different from each other by a considerably large amount, thus these genres can keep the player entertained for a long time.

For action games, there are also several approaches for developers to expand the value of revisiting. Action games in the 1980s commonly use a simple tactic: loop the game from the first stage once beaten, and make stages harder by introducing more enemies, increasing enemies' attack power or make enemies more aggressive. This strategy was somewhat effective in those days, since players were not get nearly as many good games as we are now, so they had no choices but stick to the same game again and again, the beefed up difficulty can pose a new challenge to the player, and keep them entertained for an extended time. There is also a modern version of this approach, video games nowadays provide the player with extra difficulty level or extra modes upon completing the game, in order to give players more reason to come back later for more challenge.

Some video games extend this approach by not only offering new difficulty levels or extra modes to the game, but also new content additions in the later playthroughs. Some games would offer new narrative scenes (e.g.: the true ending of *Ghosts'n Goblins* can only be accessed if the game is beaten twice in a row, *Halo* has some additional ending scenes can be viewed only if you have beaten the Legendary difficulty), challenging goals with certain limitations or require the player to perform a certain action at a specific time (e.g.: the achievement/trophy system on modern consoles), or offering a "new game plus" option to allow players to start the new playthrough with bonus items and weapons, or even the ability to carry over all your weapons, items and powerups to blast through the whole game with a much easier ride (games like *Resident Evil 4*, *Dead Space* and certain Japanese RPGs have a new game plus mode).

Some of the games go even further by implementing procedurally generated random levels. Games like dungeon crawler RPGs have random generated enemy layouts and trap positions, or even the dungeon itself is randomly generated, the idea is to use algorithms to decide the layout of levels instead of designing them by hand, offering a new game experience to counter player's knowledge to the game, thus keeps the player's interest on the game longer.

Though random generated levels sound more than enough for replayability, there is a catch. The levels generated are usually much less interesting and less elegant than those created by human, even the balance of levels may become problematic due to the simplicity of algorithm design. So a few developers choose to build more complex and organized systems for the automated level design mechanisms, these systems can control the progress for generating levels with more precise conditions. A typical example is *Left 4 Dead*. *Left 4 Dead* includes a system called "the Director", which monitors the game world, as well as the location, health, weapon and ammo count of all players, and change the level design on the fly by generating enemies in different numbers and different types, as well as adjusting supplies around them. For example, If they are sticking together, "the director" will spawn enemies with excellent area-of-effect attacks, if players are spreaded, "the director" will spawn enemies with the ability to restrain a single player; if players have enough ammo and health, "the director" will spawn a large number of enemies, if players are low on health and ammo, "the director" will spawn weapons, medpacks and ammo piles nearby, etc. (Booth, 2009). The idea behind the director is to keep a certain amount of pressure on the party of players by controlling the level of difficulty against them, and force them to keep their tactics changing all the time. By keeping the difficulty at a challenging but not annoying level, which matches the "challenge-based immersion": the level of immersion is maximized if the challenge presented is at a "correct" level which just hit the "sweet spot" of individual players (Ermi and Mayra, 2005), the replayability is significantly increased.

All these ways to enhance replayability are from the gameplay perspective, which means they cannot be applied to adventure games, since, as I stated in the previous section, adventure games are narrative-driven, the gameplay component is somewhat lacking, all these ways can hardly be applied to adventure games, posing a serious replayability issue to the adventure genre, I will discuss about these issues in the following section.

1.4. The Replayability Problem and the Fall of Adventure Games

The replayability problem for adventure games did not stop adventure games becoming popular in the 1980s and 1990s. The main reason for adventure games became so popular is that, during the time, this genre was almost the only one that can provide a strong narrative-driven experience. During the same time period, the narrative in other game genres such as action were generally simplistic, lackluster or close to non-existent in game. Also, due to technology limitations in that time, the presentation of narrative in other genres was significantly worse than that in adventure games since they have to focus their attention elsewhere. The RPG genre has the potential to have a complex story that can compete with adventure games, but RPGs during that time usually use plain text and simplistic in-game animations as the device of narrative. This is usually the best narrative a video game other than adventure games can get in those days. On the other side, adventure games such as *Gabriel Knights* and *the Secret of Monkey Island* had the luxury to have beautifully crafted graphics as well as animations. Later in the 1990s, adventure games are among the first games to feature audio dialogues, in some games, even full motion video clips are used extensively. The advantage both in narrative and presentation have secured the place for adventure games on the market during that time.

However, things started to change in the late 1990s, video games in general start to have complex stories, the presentation also caught up with the help of technology advancements as well as borrowing methodologies from adventure games (e.g.: RPG games like *Final Fantasy VII* and action-adventure games such as *Resident Evil* start to use pre-rendered backgrounds and CGI movie clips just as adventure games do), even suppress the presentation of adventure games. On the other hand, the adventure games remained largely unchanged, most of them still using the clunky point and click user interface, and the core gameplay mechanism remains the same: traverse between locations, find items, and find where to use these items to solve mysteries, and move the narrative forward. Once the game is beat, the mysteries are solved, and the narrative comes to a conclusion, there are very few reasons to start over a fresh playthrough.

Since adventure games have very little replayability, this genre has to find another way to stretch the time duration for a single playthrough, so they have designed

a lot of difficult puzzles scattered throughout the game. These puzzles can pose a good challenge and a good pleasure for the players if they are cleverly designed and can be solved with proper logical inference or reasoning. Some adventure games in the late 1990s indeed had challenging yet interesting puzzles, such as *Broken Sword*, but in most cases, this was not the case. Designing a considerable amount of quality puzzles cost too much, so it was very popular among adventure game developers to implement "cheap" puzzles which the solution is very counter-intuitive, such as requiring players to use a combination of random items to make a nonsense contraption, or use an item on a random place that makes no sense logically. Finding a solution to such a puzzle become a wild guess instead of rational thinking. These illogical puzzles can drag the time needed to complete the game by a considerably huge amount, causing huge frustration to the players, what is more terrible, some puzzles were made near impossible to solve deliberately, in attempt to sell players additional services such as official strategy guide and hints through paid phone service (Digital Game Museum, 2011).

This kind of brutal puzzle design approach combined with uninspiring gameplay design seriously hindered players' interest in adventure games. By the late 1990s, the other genres have already caught up with adventure games in the domains once adventure games had advantages in. Players no longer need adventure games for a presentation pleasure or complex narrative. First person shooters have stunning visual impacts, while players value narrative and exploration can simply go for the RPG games. The adventure genre started to fall apart in the late 1990s, especially the old school point-and-click genre, a few masterpieces around the millennium such as *The Longest Journey* and *Syberia* did not prevent the fall of adventure games (Digital Game Museum, 2011). Today, the most famous adventure games often take the form of "interactive drama" by mainly focus on film or drama like interactive experience utilizing possible narrative choices and quick time events, but less about puzzle solving and exploration, such as *Heavy Rain* and the *Walking Dead*; or combines adventure with other genres like action or open world, such as *L.A. Noire*. All these efforts are made to enhance the one-time experience, but they are still not truly worthy for a revisit in terms of replayability.

A way to extend the replayability of narrative-driven games, such as adventure games, is to make branching stories instead of a single storyline. The user would be more satisfactory if they get more choices, so it is better to design many choices for the

users (Crawford, 2003). Unfortunately, story and narrative is heavily depending on the carefully written content, unlike gameplay mechanics, it is very hard to automate the creation process of narrative. There are many efforts put into generative narrative, but they are not ready for use yet. Hence the replayability issue still a menace troubles adventure games. And in general, troubles the experience of interactive narrative.

1.5. About this Project

In the previous section, I have raised the replayability problem in narrative-focused games, such as adventure games. While most video games can easily make replay worthwhile by a variety of approaches, these approaches are not suitable for narrative-driven games. These games are still suffering from replayability issues. To solve or improve replayability in narrative-driven games, variety is required. Choices and branching stories are the common way among narrative-driven games, but the number of choices and branches are often limited due to the limitation of time and resources, since the content of narrative still need to be created by human.

This project is an experiment on the idea which combining algorithm controlled content generate with narrative. It takes the form of a detective adventure game. The bare bone idea is classic and simple: several characters presents in an enclosed area which have several different locations and a set of items, and murders would occur, just like the novel *Then there were none* by Agatha Christie, the only difference is that the player assumes a character of a “detective”, his task is to solve the murder cases and find out who is the murderer.

The narrative has a fixed narrative arc, which the huge frame of the story is predetermined, but instead of having a fixed series of events (or multiple series), the game would feature a database of tagged candidate event segments, a planning system with a proper knowledge representation world is used pick and schedule these events on the fly according to the changes in state of the world as well as some randomized factors to form a “reasonable” story line as well as a logically solvable situation for the player.

Through the development of this prototype, I would like to verify this research question: is it possible to to improve replayability in narrative-focused games by using limited content generation approach? The reason why I have chosen adventure game as

a presentation form is that the adventure genre is very narrative-focused, and the gameplay factor is minimum. It is better for this prototype because I can see more clearly see the dynamics of my research question. The reason why I choose limited content generation is because the technology is not quite efficient enough to create complex meaningful contents, so I have to operate on a predefined domain and pre-authored database of contents. This project is meant to be somewhere between video games with predefined narrative contents just like most video games do, and the experimental programs which the artificial intelligent agent is fully responsible for creating new content from scratch.

1.6. The Structure of This Thesis

The rest of this thesis is divided into 6 chapters.

Chapter 2 (Literature Review) provides a series of reviews on various topics related to the development of this game prototype. These topics include new media design, narrative, interactive narrative, game narrative and generative narrative, as well as various discussions on video game design.

Chapter 3 (Event Planning and System Details) includes a detailed introduction to the two essential components of the event planning system the game prototype uses: the STRIPS based planning system, the probability based event scheduler.

Chapter 4 (Game Prototype Details) gives a detailed description on the game prototype. It includes both its specification, game design, and narrative design, as well as how the event planning system works in this game prototype.

Chapter 5 (Discussion) provides an in-depth review and analysis of the completed game prototype. In this chapter, the game prototype is examined and discussed extensively using materials introduced in chapter 2. The designs and choices of the game prototype as well as the rationales behind these designs and choices are also discussed in detail.

Chapter 6 (Limitations and Future Works) lists a number of problems and issues of the game prototype, and discusses possible future fixes and extensions to this

prototype. It also briefly discusses the future research direction of my ideas illustrated in this thesis.

Chapter 7 (Conclusion) is a summary that briefly reviews the previous discussions and high points of the thesis.

Chapter 2.
Literature Review

2.1. New media Design

Numerous scholars did extensive researches over new media. Among them, a good number of theories and principles emerge from these researches are well concluded. Outcomes from these researches can serve as valuable theoretical basis for all future researches in this field. This project is a research in the new media field, so it is worthwhile for me to examine these previous researches, and find out what I can benefit from these theories and principles.

2.1.1. Manovich's 5 Principles on New Media

Lev Manovich suggests 5 principles that he thought all digital media artifacts should follow in order to be classified as a new media artifact. These principles suggest that 1) new media artifacts should be described or represented digitally and can be controlled by algorithms; 2) it should has a modular structure, which every part of the new media object can be reused in another object; 3) the new media object can generate new content from a database automatically on the fly according to certain algorithms or requirements, human can be removed from part or even all of the creation process; 4) the new media objects should be flexible to changes, resulting in infinitely many versions; 5) all new media objects consist two layers: "cultural layer" containing content intended for human viewer and "computer layer" containing representations for computers to process (Manovich, 2000).

These principles generalize all the basic traits of new media artifacts stripped to bare bones. The first four principles basically say that every new media artifact should be a computer program which features a function structure, can form new situations from source material automatically, and is subject to infinite times of change.

The fifth principle is more interesting, because this principle is draw a line between the front end (cultural layer) and the back end (computer layer). The idea is much similar to the "wall" analog seen in the famous computer science data structure

textbook “Walls and Mirrors” (Carrano & Prichard, 2006). The textbook refers abstract data types (ADTs) as “walls”, since ADTs hide their actual implementation details from the user, and users only need to access its public interfaces to make use of its functions. The “culture layer” and “computer layer” principle takes the “wall” metaphor and applies it to a larger extent, in which a wall separates the complex details of “computer layer”, so the user only need to interact with the “culture layer”. The two-layer structure is a good thing because the user does not need to care about how the artifact actually works, what they need to know is how to interact with “culture layer” and get most out of it, while leaving the construction details to the author.

2.1.2. Ryan on New Media Properties

Ryan identified the ambiguity of the term “media” itself. She gave two definitions summarizing media from two different perspectives. From user’s perspective, media is “a channel or system of communication, information, or entertainment”; and from author’s perspective, media is “material or technical means of artistic expression” (Ryan, 2006). Also, she noticed that the media should be viewed both as technologies and culture practices, this is similar to Manovich’s principle about “culture layer” and “computer layer”.

In addition, Ryan has her own set of properties which she thought new media objects should have. New media objects should be able to respond to changes in virtual world environment and user input, should contain multiple sensory and semiotic channels, might have online capabilities, the digital text should be able to be refreshed or rewritten without throw away any material supports, and should be a modular design (Ryan, 2004). These properties are amazingly similar to Manovich’s 5 principles, especially in the sense of reusable materials and modules.

2.1.3. Murray on 4 Properties and 3 High Level Pleasures of Digital Media

Murray discussed on the properties of digital medium, and stated that there are four types of affordances in digital mediums, which are encyclopedic, spatial, procedural and participatory. Encyclopedic means that digital mediums are capable of transmit a huge among to information to the user, the information can be indexed and organized in

very efficient ways that the user can search and retrieve information from the medium in multiple different scales. Spatial means that the computer can construct virtual spaces in which the user can navigate freely inside. Procedural means digital mediums should be able to “represent and execute conditional behaviors”, they should be designed and implemented in a way which can process and respond to user input according to a certain set of conditions and rules in a specific procedure. Participatory means interactivity, the digital medium should allow interactivity and be able to communicate with the user with certain flexibility, and provide proper feedback to any allowed input from the user (Murray, 2011).

Murray also contributes a lot in the principles of new media design by identifying the three high level pleasures from new media artifacts. She discussed about these pleasures in her book “Hamlet on the Holodeck” (Murray, 1998), the three high level pleasures are immersion, transformation and agency. Among these three pleasures, immersion and agency are the two more important ones for the background of this research, so I would like to explore in these two pleasures in details.

Murray thought that the the fact of being able to enter a fiction world in new media artifacts is a type of pleasure, even despite the content itself. She called this pleasure immersion after the metaphor of the experience which being submerged in water. Immersion is an experience that the user feels that they are submerged into the artifact they are experiencing. This definition of immersion is comparable to Coleridge’s “suspension of disbelief” in narrative (Coleridge, 1906) - the active willingness to give in yourself for the story. However, when experiencing a new media artifact, most of the time the user can interact with the artifact, so the definition of “suspension of disbelief” for traditional media forms such as novels and films is a bit of passive for describing the experience in new media. So Murray went one step further and suggested that the user is actually creating a belief actively during the experience, it is an active process that reflects the interactive nature of the new media, instead of the passive experience of narrative in traditional media.

Agency is another important high level pleasure in new media artifacts. Murray defined agency as the satisfaction that the user’s ability to take meaningful actions and witness the consequence of their choices (Murray, 1998). If we put this definition in author’s perspective, then we can restructure this definition the other way around: new

media artifacts must allow the user to take meaningful actions and choices, and reflect their actions in the world later on, in order to satisfy the user. In this manner, the definition turns into a fundamental guideline for how to make a new media artifact. Out of all three pleasures mentioned previously, agency might be the most important one to video games since this is a more detailed discussion about what kind of experience a video game should provide to keep the user entertained and immersed, so it would be beneficial to explore further in this topic.

Murray continued on to identify 3 categories of pleasures under agency. One of which is the pleasure emerge from the ability of navigating in the virtual space, another one is the pleasure of story, either from a single path progression story or a story which direction is determined by the user, and the last one is the pleasure emerge from problem solving (Murray, 1998). All these three pleasures are more specific design principles for new media artifacts in general, including video games. These pleasures will be discussed in details later in this chapter, under narrative and video game sections respectively.

2.1.4. Zimmerman on Against Hypertext

Hypertext based interactive artifacts were commonly seen in the early days of internet, Zimmerman was against it and regarded it as not interactive enough. He made his focus on the interactivity in his essay “against hypertext” and made a strong claim that interactivity is the most important trait which separates new media from “old” medias (Zimmerman, 2000).

To back up his opinion that hypertext based artifacts have poor interactivity, Zimmerman went on to review structures in interactive artifacts. He identified two main categories: content-based or embedded structure which have pregenerated content only, and system-based or emergent structure which are set of rules and procedures to generate new and unexpected content. He then pointed out that an explicitly interactive media should have both of these structures, while hypertext only have the first structure, hence it is a poor interactive artifact (Zimmerman, 2000). Interestingly, Zimmerman’s strong claim was more or less accurate, as those hypertext-based artifacts quickly diminished along the progression of internet and new media design principles. They lost

their popularity mostly due to their lackluster design which severely hindered their power of presentation and interactivity.

2.1.5. Bolter et al. on Remediation and Oscillation between Reality and Virtual World

Bolter and Grusin focused their attention on another interesting perspective by discussing the new media itself as a medium. They talked about transparent immediacy, in which the media itself should erase its existence and invisible to the user, so the user can have better immersion into the content. They also talked about hypermediacy, in which the media is “not act as a window on to the world, but rather as ‘windowed’ itself”, by acknowledge its existence to the viewer using different representations (Bolter & Grusin, 2000). Their idea on hypermediacy is somewhat close to the famous “medium is the message” from McLuhan. The idea behind this is to make the viewer be aware of, and appreciate the media itself, rather than the content it delivers.

The idea of immediacy and hypermediacy leads to the discussion of remediation: “the representation of one medium in another” (Bolter & Grusin, 2000). They argued that it is the “defining characteristic of the new digital media” (Bolter & Grusin, 2000). They pointed out that there are four kinds of remediation, out of the four kinds, the final one is the most interesting one to look at. It is a combination of immediacy and hypermediacy, in which the new media absorbs the old one entirely, that way the original content is still there, but the user has all the freedom to interact with it (Bolter & Grusin, 2000). Many video games are the remediation of old medias in some ways, they offer fresh takes on the same material from older medias, so that both the audiences and creators from older media and new media can benefit from this kind of remediation.

Bolter, along with Gromala, also discussed on the relationship between the virtual world and the existence of reality in digital media. They argued that the digital designs should acknowledge the existence of both virtual world and the importance of the reality around the interface. There is no design that can be purely abstract or completely detached from the physical world, the design itself is oscillating between the physical environment and the virtual world (Bolter & Gromala, 2003). From the user’s point of view, their mind is often oscillating between immersed into the media and aware of the surroundings and the “window” to the virtual world itself in the reality. Manovich

also mentioned a similar oscillation, in which he identifies the oscillation between the illusion created by the virtual world inside computer and the destruction of the illusion by the presence of interface which reminds the user it is not real (Manovich, 2000).

2.2. Narrative, Interactive Narrative and Video Games

2.2.1. Definitions of Narrative

Narrative is a very broad topic with a lot of different definitions. There are quite a few similarities among them, but they are all different from each other in one way or another. Most of these definitions agree on the fact that the narrative is composed by a number of events, as a matter of fact, the presentation of these events alone is the original definition of storytelling (Chatman, 1978).

In comparison, Abbott's definition on story and narrative has one more factor in addition to the collection of events definition: narrative is a sequence of events (or a single event) represented to the audience (Abbott, 2002, cited by Ryan, 2006), by defining narrative as a sequence of events instead of a collection of events, he made a point that the order of the event matters, only a series of events represented in order can be called narrative. In reality, the sequence of events need to be in a specific order (such as ordered by time or ordered by relativity around a character or a core event) to establish a meaningful relationship between cause and effect, or the narrative would be chaotic and hardly make any sense.

Bordwell has another perfect definition for narrative, which is a more definitive one compared to the previous two. The narrative is considered as "a chain of events in cause-effect relationship occurring in time and space". This definition of narrative is usually referred as "story" by many people. By definition, story contains both exactly what the events explicitly represented in the narrative, and anything we can infer or presume based on given information in the text. On the other hand, plot is strictly anything that represented visually or auditorily in the text of the narrative itself. Its main difference compared to story is that plot does not contain anything which is not in the text directly, but can be inferred by the user / audience. Also, plot contains "nondiegetic materials" such as the background music and certain visuals which have nothing directly connected with the story (Bordwell & Thompson, 2008).

The definition by J. Hillis Miller, as quoted by Zimmerman moves once another step further in this series of evolution on narrative definitions. According to Zimmerman, Miller's definition was organized in three parts: the first part defines narrative as a story, he stated that a narrative should have an initial state, a change in that state, and an insight brought by that change. The second and third part treat narrative as a plot, in which he claimed that the sequence of events should be "a personification of events through a medium such as language", and the representation should be a part of certain patterning and repetition (Miller, 1995, cited by Zimmerman, 2004). This definition is a further step from the previous one. It agrees on the fact that the narrative is a sequence of events, too. But on top of that, it further claims that the sequence of events should result in some form of change to the initial story world in terms of story content. The narrative should also have its own unique personality, while its structure should be a well-established one that was used in past stories. Since Zimmerman is a games scholar, the change in state is an important part of the definition of narrative to him. He sees game narratives as a dynamic process that should change in the context of game play. This is also one of the reasons why I am adapting the planning approach to generate sequences of events: assemble event sequence with different events in order to achieve different cause-effect relations (the details in the specific narrative is different), while still maintaining the same overall structure (in general, the narrative tells a same kind of story).

2.2.2. Interactivity

Interactivity is also an essential factor in narrative for new media artifacts. Some scholars such as Zimmerman regarded it as the most important trait of new media, while the other scholars like Ryan and Murray also discussed about it extensively, since it is indeed plays a central role in new media environment, especially in video games.

In "Against Hypertext", Zimmerman also discussed 4 modes of interactivity in addition to embedded structure and emergent structure. These 4 modes of interactivity are cognitive interactivity or interpretive participation with a text, functional interactivity or utilitarian participation with a text, explicit interactivity or participation with designed choices and procedures in a text and meta-interactivity or cultural participation with a text that only happens psychologically and emotionally. The first one states that cognitive interactivity is the the perception and interpretation of the text. The second one

is the functional interaction with the medium of the text. The third one is explicit and designed interactivity is the interactivity mostly discussed in new media ground, such as clicking links in hypertext websites or playing video games. And the last one is outside the text itself, being the reader participate in a discussion, deconstruction, or reconstruction of the original material (Zimmerman, 2000). The third interactivity mode - explicit and designed interactivity - is the one we are talking about most of the time whenever we discuss interactivity in new media. These modes are not specific to new media objects. The first one is emotional and mental activities inside the user's brain, the second one is related to the interface of the artifact, and the last one involves processing the original material with other people. It is good to keep these in mind when designing a new media object, in which the first and last modes reminds the author that audiences / users also have their own interpretation of the artifact, so it would be beneficial to design some thought-inducing content; while the second one suggests for a good design of human-computer interface (HCI). A well-designed HCI can indeed boost the experience of interactivity.

Manovich had a brief discussion on HCI design. He thought that the human-computer interface is actually a "remediation" of old media, he suggests that the interface often has two conventions. One of them is the interface itself acts like a window to the virtual world, allowing the user to see through it. And another one is the traditional sense of HCI, which the user actually interacts with (Manovich, 2000). The remediation idea here is exactly the same as what Bolter and Grusin mentioned as the last kind of remediation: the new media (HCI) absorbs the old media (physical interfaces) (Bolter & Grusin, 2000): providing interactivity in the virtual space to the user, while its function still maps its physical counterpart in some way.

Ryan contributed a lot in the discussion of interactivity by providing her take on user participation in digital media. She identified user participation in digital media by 4 modes in two groups, which are internal / external mode, and exploratory / ontological mode. In internal mode, the user project themselves into the virtual world either from a first person view or assuming the identification of an avatar inside the world; in external mode, the user is outside of the virtual world, controlling the virtual world by either "playing god" from above or perform an activity which close to browsing a database. In exploratory mode, the user can browse the virtual world as a database but cannot affect it; in ontological mode, the user's decision can change the world (Ryan, 2004). These

participation modes can serve as a useful tool to identify and evaluate interactive new media objects. For example, most adventure games are in internal mode, since players usually assume control of the protagonist of the story, and they are ontological, since the goal for the player is to solve puzzles and change current state, not just observing the world without consequences.

2.2.3. Interactive Narrative

We have finished discussing narrative and interactivity separately, now we can go further by examining papers and researches about narrative in an interactive new media environment. Although interactive narrative still follows many traditions from narrative in old media forms, there are many theories and principles from traditional media forms that just cannot directly fit into this new domain. A lot of changes and modifications, even total overhauls to these theories and principles are required to make them work in an interactive narrative environment.

Ryan has a flexible and scalable “fuzzy” definition of narrative that dismantling narrative into several essential components: a world setting, a set of characters to populate the world, a set of events involving these characters, and a series of changes to the world brought by these events (Ryan, 2004). Interestingly, these components have their comparable counterparts in a computer planning system, which also has a definition of world (or knowledge base), agents (comparable to characters), events (actions which can be used change the state of the world), and a plan to change the state of the world. Hence it is very reasonable to model narrative using such planning system, I will talk about the planning system in details in the next chapter.

Manovich also has an interesting approach on narrative, his approach is more focused on a specific type of narrative in digital media: the database. Manovich thought database is an alternative way to model the real world. Any object in the world can be modeled as a database structure, and any process can be reduced to algorithm, even narrative can also be reduced to database (world, characters, and events) and algorithm (how the sequence of events carries out). Following a narrative is actually traversing through the database following links between database items created by the author. If we see this from the author’s point of view, creating a digital media object is a process of creating the database and an algorithm to establish relations between data. He also

acknowledges that a random combination of data drawn from database is not narrative, these items must be in an order that cause-and-effect relations matters to be a narrative. This definition matches other definitions of narrative mentioned in this section earlier. However, the unsorted items in databases and the cause-and-effect related items in narrative both represents different culture in new media objects. The interactive narrative also acts as a mask to the underlying mechanics. From the representation, it gives the user an illusion of creating their own story, since narrative need to satisfy a set of conditions, not just connecting random accesses to the database (Manovich, 2000). These ideas of Manovich are a further deconstruction of narrative from a digital media perspective, by describing narrative as database and algorithm, it suggests that if we can replace the author in the event sequence creation process with an algorithm to automate this process, which is exactly what this project is trying to accomplish.

Crawford also talked about the architecture of interactive narrative. Crawford focused on the core of interactive narrative structure itself, and mainly discussed about choices and consequences in interactive narrative artifacts. He has identified several narrative structures, such as the following structures: A linear structure which is the same as traditional narrative. A branching storytree with multiple branchpoints leading to different consequences. A foldback structure which features choices that eventually foldback to the single main storyline. A “kill’em if they stray” structure, an obstruction structure which is basically linear structure with some kind of obstacles the user need to overcome in order to advance to the next story node. Finally, a hand-wired storytree in which the creator manually designs every connection between story nodes to form a complex interactive story network (Crawford, 2003).

From my observation, most video games today mainly take advantage of an essentially linear structure, such as most first or third person shooter games, or a foldback structure, such as most role-playing games. These basic narrative structures are often used with an obstruction structure to block off story progression by actual gameplay and challenges, players have to complete these gameplay sections in order to advance in storyline. The reason why nobody uses the simple branching storytree is simple as Crawford stated in the research: the number of branches would grow to an extremely large number very quickly along with the increasing length of the narrative, before long the number of states in the storytree becomes too large to achieve. Nevertheless, Crawford did not like the approach most video games took, he values the

user choice over anything in interactive media artifacts. He thought the interactivity inside obstructionist approach is “skinner box interactivity”, in which the user is just like “a rat in a maze, require to push the correct levers to get the reward - and punished for failure”. So he went on to set up a criterion of excellence for interactive narrative based on the number of meaningful choices the user can make. He also argued about a few ways that he thought would be effective to increase interactivity. Such as decreasing conceivable states, increasing accessable states, adding more variables, using arithmetic variables instead of boolean variables, and using computed branching (Crawford, 2003).

Apart from his obsession with quantifying and maximizing user choice, he has a good point that branching and choices can indeed increase the user’s interest in interactive media, as well as suggesting a computed approach for the branching factors. Also, his description of gameplay and challenges as “skinner box” activity is very interesting and smart. Although we all love these “skinner box” activities (Crawford, 2003), improve the experience from somewhere outside of gameplay itself is surely a good option.

2.2.4. Game Narrative

The narrative arc in the interactive digital media environment is an interesting topic to start this discussion. The narrative arc in traditional media usually follows the following path: setup, complication, development, resolution, and denouement. The power of narrative through such a narrative arc is largely depending on the careful control of pacing and implementation of other details. It is possible to accomplish that kind of tight control in a traditional media such as film, since the creators has complete control over the media object. However, it is not the case in an interactive scenario such as video games, simply because the user plays a huge role in the narrative itself, so the precise timing and pacing control is impossible to achieve since the creator just cannot anticipate the variables brought by the largely diverse user base (Bizzocchi, 2006). But this does not mean narrative arc is completely useless in an interactive environment, it can still apply to the overall storyline. Although precise pacing and timing control is impossible, creators still have a somewhat indirect control over the pacing of the narrative with some new tools at their disposal to influence the user’s experience and guide their emotion to a proper state. Such as level design, enemy placement, item

placement and characters; also, some old techniques from traditional media such as dialogue and cut-scenes are also available and is widely used by video games.

Based on the observation that the traditional narrative arc cannot directly apply to a new media artifact, Bizzochi suggested a new framework for analyzing narrative in interactive media objects such as video games. This framework somewhat resembles part of Ryan's definition on narrative by having a list of components such as story world and character, there are also additions like emotion guidance, narrative interface and micro-narrative (Bizzocchi, 2006). Out of the three, emotion guidance and micro-narrative is more interesting for the discussion of my project.

The emotion guidance is a device that narrative can use to guide the user to a certain emotional state, and I believe interactive new media artifacts can do this better than traditional media thanks to the power of interactivity. Although Bizzocchi suggested that there are no video games can inflict strong emotional involvement such as "make the player cry" through his observation, I can think of a few video games which really did a great job on emotion guidance. Such as the clever combination of artistic design, micro-narratives and an impressive main storyline riddled with complex psychological metaphors to reflect the deep inner struggle of characters in *Silent Hill 2.* Or the story of a distorted "American dream" inside an illegal immigrant goes wrong step by step in *Grand Theft Auto IV*.

Micro-narrative is another device interactive narrative artifacts can take advantage of. They are localized sub events and sub stories that can be used to enhance the main storyline as well as the story world. They utilize the pacing control of traditional narrative arc locally, and gain limited control through this segmented progress. These efforts provide a small but strong narrative experience, and contribute to the main storyline as well as a part of the virtual world. A good example of micro-narrative usage would be *Dark Souls*, in which the whole background story is fragmented into the form of micro-narratives.

Jenkins identified some aspects of narrative design which video games can take advantage of. He has identified several narrative architectures that can be utilized in an interactive environment. The first architecture is environmental storytelling, that is storytelling using the virtual space itself in addition to all kinds of texts. The second one

is the enacting story, or micro-narrative, which just mentioned in the last paragraph. The third one is embedded narratives, what it says is that the narrative information in an interactive environment “must be presented redundantly across a range of spaces and artifacts, because one cannot assume the player will necessarily locate or recognize the significance of any given element” (Jenkins, 2004). In that spirit, scattering narrative objects such as documents and voice recordings around the play space as collectables, along with other objects in the game environment which contain meaningful narrative information can be an effective strategy to encourage exploration. Players can discover these items and interact with them to gain more comprehension to the story of the game. This strategy is widely adopted by many modern video games. The last storytelling narrative is a rather unique one, in which the media itself is not exactly the storyteller, the player themselves are the storyteller, they make their own story through all tools provided by the media, this is called emergent narratives, such as a playthrough in The Sims or Sid Meier’s Civilization (Jenkins, 2004).

2.3. Computing, Interactive Narrative and Generative Narrative

Interactive narrative and generative narrative exists even in the pre-computer ages. There are plenty of non-electrical games existed earlier than video games have born which inspired computer games as well as other digital media artifacts a lot, including this project. I think it is worthwhile to examine them first.

2.3.1. Non-Electronic Interactive Narrative and Generative Narrative Works

The Exquisite Corpse is a good example of a simple form of generated narrative from the pre-computer era. This game of creation emerged in the early ages of Surrealist movement. This game requires several people to play, each of them write down a phrase on the same piece of paper, fold the paper to hide what they wrote and pass the paper to the next people, in attempt to form a whole sentence (“About Exquisite Corpse”, n.d.). It is an attempt to generate a meaningful sentence through collective effort from a group, in which they do not communicate with each other about the text they have created. It can be treat as a generative narrative attempt before computers exist, if we treat the people take part in this game as the processors, and the game rule as an

algorithm. This activity also has a drawing version, which works in a parallel fashion - each “player” contributes images, without knowing what the other players are contributing. In both cases, text and drawing, the completed artwork is generated by the participants working independently within the system of creation.

Another interesting artifact is Queneau’s “A hundred thousand billion poems” (Queneau, 1961). It is a book with a set of ten sonnets which all the sentences have the same rhyme scheme, so any of the lines can combine with any other 9 lines to form a new ten sonnet, resulting in a very large number of possible ten sonnets. The book’s pages are sliced horizontally so you can mix and match lines from across the book. It shares similar ideas with the “Exquisite Corpse”, but in a more elegant way.

On the other hand, the early tabletop role-playing games such as “Dragons & Dungeons” (D&D) resembles interactive narrative. In these D&D game sessions, the game master hosts the whole game and responsible for everything in the game. They are responsible to build every piece of settings in the fantasy game world, from the world setting to every single enemy the other players may encounter. They also responsible for enforcing rules and game systems, such as the random factors such as damage and consequences of decisions are controlled by rolling dices. If we replace the game master and dices with a computer program, then we get a computer role-playing game. The computer role-playing games can be seen as evolved versions of this interactive narrative which can be played with as few as only one player.

2.3.2. Projects involving Computational and Generative Approaches

Moving onto the computer age, generating narrative or improving interactive narrative experience using computational approach is not a new topic in computing science. There are a plenty bunch of projects with limited success in the past, and more successful ones later on. Among them, there are some projects using planning approach or methodologies. In this section, I will go through some of these projects.

Ramirez and Bulitko’s (2014) project focused on a new AI based player experience manager called Player-specific Automated Storytrelling (PAST). The system continuously to accommodate player behavior with pre-defined goals. Whenever the player’s action breaks current goal, the system would generate an alternative narrative

to re-enable the goal using automated planning. The generative narrative is directed by a player modeling system which evaluates player action and select the most suitable accommodation for the current play session.

Kumar and Ramakrishnan (2008) introduced a way to model the algorithm for storytelling by utilizing a data mining strategy called redescriptions. Their approach including defining events as redescriptions and stories as chains of redescriptions, and then using A* search on results from redescription algorithm to design a storytelling algorithm. They also tested its scalability and its potential for knowledge discovery.

Michael Young's (2007) paper a bipartite model of narrative generation in virtual worlds proposed a new way to model the narrative in an interactive 3D rendered virtual world by separating the model into a bipartite model, which contains a narrative model and a discourse model. The narrative model contains the story elements, while the discourse model contains the elements for how the story is told. The paper also presents a system called Mimesis which implemented the model using DPOCL planner co-developed by the author, along with a game engine responsible for present the virtual world. The paper also presented an example to prove the ground of this model.

Hartsook, et al.'s (2011) project is about a research on a system which can procedurally generate "complete, executable" game worlds for computer role-playing games. They use a story presentation which breaks the story of an RPG game into a list of "plot points", which are important events in the whole story, and a set of axioms to represent the state of the story. They implemented a game plot adaptation algorithm to generate a storyline from a pre-defined partial plot as well as the preferences of the player, and they put the result of this algorithm into a game world generation system to generate an executable linear game world, using mainly genetic algorithm.

Ware, Young, Harrison and Roberts' (2014) paper described a planning algorithm based on an existing planning algorithm which generates intentions and actions in chronological order for each character, and the conflict of these intentions in a story. The CPOCL (Conflict Partial-Order Link) algorithm described in this paper is an algorithm based on POCL and focuses on the generation of conflict in the narrative. In order to make an interesting story, the author identified conflict as an important element in the stories. He has also identified seven dimensions of conflict in story, which are

participants, reason, duration, balance, directness, stakes, and resolution. The paper goes on to reveal the technical details of this algorithm, and validated and discussed various experiment cases on planning structures and the dimensions. They recognized their work as an extension to current existing story generation researches which focused on intention-driven character actions, and believe this research is a good demonstration of the potential of automated story creation via computer systems.

Also, the projects I have worked on, such as the “Seasons” project by Jim Bizzocchi has provided me some inspirations for this research. “Seasons” is based on a generative engine which picks clips from a database according to a scheme. All clips in the database are associated with two layers of tags: a season tag and a set of content tags, the scheme specifies number of clips per content tag as well as content tags per season tag, and transition effects, then the generative engine would pick clips according to this scheme to assemble an ambient video on the fly according to these specifications. The database structure from this project has given me some initial ideas about how to handling data in a generative system (Bizzocchi, 2015).

2.4. Video Game

Despite the fact that narrative is the central focus of this research, the outcome of the research is still counted as a video game. Creating this prototype is an exercise in video game design. The purpose of the research is rooted in the context of video games: it focuses on finding a solution to an ongoing challenge in video game design - supporting replayability through coherent variations in the game narrative. It is very worthwhile to examine some literature on the topic of video game itself, as well as its design principles, in order to find some suitable tools and criteria for analyzing this game prototype.

2.4.1. Definitions and Characteristics of Play and Games

If we want to discuss video games, it is best to start with the definition of “play” and “game”. Even before video game exists, there were researches on games in general. Caillois is one such scholar, in his influential book “Man, Play and Games”, he did a critical review based on Huizinga’s theories, and summarize a set of six core characteristics for the definition of “play”. These characteristics are described as follows:

the activity should be emerging from one's free will and should not be obligatory; the activity should have its time and space separated from others; the activity should have uncertainty to its course and outcome; the activity should be unproductive; the activity should have rules; and finally, the activity should involve an awareness of an imaginary reality which against real life. (Caillois, 1961) In spite of the year which this book came out predates the existence of video games, these definitions of play still fit all games including modern video games surprisingly well.

Zimmerman's definition on "game" is a somewhat different to Caillois' definition on "play": the "game" is a voluntary interactive activity that one or more players engaging in an artificial conflict following rules, which leads to a "quantifiable outcome" (Zimmerman, 2004). The definition partially reflects Caillois' definitions on "play", but with a shift of focus. The definition named 5 elements, Zimmerman thought all of these elements should be met in order to qualify as a "game": first of all, players participate voluntarily; the activity should be interactive, so the player(s) is actually participate in the activity, so non-interactive activities watching a movie or TV drama does not count as playing a game; a "game" should have one or more core artificial conflict; a "game" should have a rule to limit players' behavior; and last, the outcome of a game should be easily measured with some form of quantifiable standards, such as scores.

Both Caillois' six characteristics of play and Zimmerman's definition agree on the fact that a game should have rules to limit players' behavior, and it should be a voluntary activity which players join with their free will. They also both agree on that it should have an outcome, but Caillois thought the outcome should be unpredictable to make "play" interesting, while Zimmerman considered "quantifiable" as an essential element. Also, Zimmerman valued the conflict inside a game, while Caillois thought alternative reality make-believe and separate time and space for game is more important.

In addition, another scholar, Jesper Juul, also has his own idea on games. His ideas are somewhat similar to Zimmerman's statement and some part of Caillois' six characteristics. Juul put together six features that a video game should have and called it a new definition for games. These features include rules, variable, quantifiable outcome, as well as the valorization of outcome. The valorization of outcome stands for how the game judges you using its own evaluation system, player effort, players' emotional attachment to outcome, and negotiable consequences that may or may not

have real world impacts (Juul, 2005). He agrees Zimmerman's opinion on rules and quantifiable outcome, while he was more focused on the outcome of games than both Caillois and Zimmerman. In his statements, five out of the six features are established around the outcome of a game, including the evaluation system, the relationship between the player and the outcome and the actual real world consequences. His opinion also provides a new focus on game research, which is something worth taking into consideration when designing a game.

Apart from the great six characteristics definition, Caillois still has some great definitions and ideas worth examining. The most notable one being the term paidas and ludus. Paidas is the spontaneous play, with self-induced non-consistent rules or no rules at all; while ludus is the opposite, which is play that controlled by rules induced by the game. Caillios thought that the progress of play is often the continuum between these two states (Caillois, 1961). Zimmerman's definition that suggests "game play" is the freedom of activity inside a structure of rules matches the continuum idea from Caillios, while Zimmerman is more precise on the topic. Caillios also divided games into 4 categories according to their nature: AGON (competition, such as athletics), ALEA (chance, such as gambling), MIMICRY (simulation, such as role playing) and ILINX (vertigo, such as horse riding) (Caillois, 1961). Modern video games are often falls into multiple categories, or even all of them.

Csikszentmihalyi's famous "flow" framework is also an important guideline for designing play and games (Csikszentmihalyi, 1990). It was mentioned by a lot of scholars in their own works, such as Zimmerman, Juul and Bizzocchi. According to this framework, the player can enjoy a challenge if and only if the difficulty of the challenge matches his ability. When the difficulty of the challenge is inside the right zone for a player, they will experience a state of "flow". Their game focus will be enhanced, and their challenge that is too easy for the player will lead to boredom, and a challenge that is too hard would lead to frustration and anxiety.

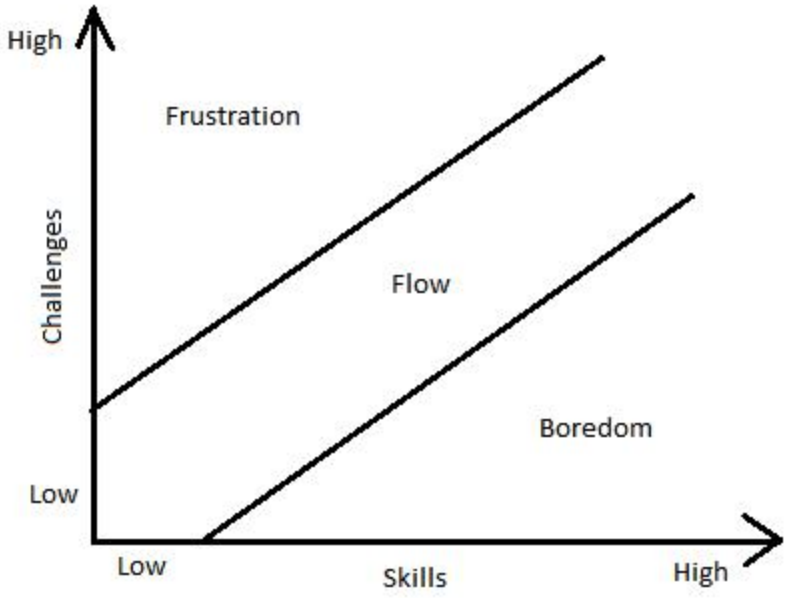

Figure 2-1: "Flow" of Player

2.4.2. Space in Games

The design of navigable space in a game is important, since it is a core mechanism inside a video game, and its design can directly affect people's experience of interactivity inside such a game. I have previously mentioned that Murray has identified the ability of navigation in virtual space as one major pleasure of agency, she also thought adventure maze and spatial journey stories such as "Adventure" and "Zork" are suitable for digital environment since "the story is tied to the navigation of space" (Murray, 1998). From this comment we can see Murray values the ability of navigation as an essential component of digital media.

On the other hand, Manovich's discussion on navigable space also agreed with Murray's point. He identified that in some linear video games such as Doom and Myst, the spatial journey is the core to the narrative, which connects to the form of spatial journey driven ancient stories. Navigable space and computer benefits from each other greatly. Computer can hold various navigable spaces, and the methodology of navigable spaces can be used as a powerful tool in computer. The navigable space can be treated as a unique culture form, all operations in new media can be represented as navigable space (Manovich, 2000).

In Jenkins' discussion about spatial stories and environmental storytelling, the design of space in games also plays an important role. He claims that games usually tell stories through the design of game worlds and game spaces, every aspect in the space should be carefully designed so that it would reinforce the experience of narrative. Also, some stories are strongly driven by the journey that hero traverse through spaces (Jenkins, 2004). Jenkins' idea also resonances with the design of adventure games, which requires player to traverse through locations in order to drive narrative forward, and the design of spaces often plays an essential role in storytelling.

Wei (2010) also stated the importance of space design in narrative in her paper about time and space in game narrative. She suggested a framework to analyze the space design in video games. The framework contains the following aspects: space as topological structure which treats the game space as a static object, space as operational structure which sees the game world unfold as the events in the story moves forward, and presentational structure which treats the space as the dynamic presentation of all visual and auditory material from the game.

2.4.3. Game Time

As Manovich suggested, in spatial journeys where the narrative is driven by the spatial movement of the main character, time and narrative are equated with the movement (Manovich, 2000). In fact, time in video games is not only important to narrative, but also in every other aspect, including gameplay and player experience, so it is also a worthy topic to examine along with navigable spaces.

Juul have put some extensive effort in game time, in his book "Half-real", he devoted a chapter in space and time. According to him, time has a duality in video games, with both the real world play time and fictional time in the fictional world both present. He continued this discussion by further develop different types of time flow in games. There are games with only real world play time exists, since these games are abstract and have no or minimal fictional world, such as chess and card games. There are games with a parallel virtual world exists, but the fictional time flows at the same speed as the play time. And there are games with fictional time flows at a different speed than the play time. The last type of time flow is more interesting to examine, it creates a need to make projection between fiction time and the play time we spend to play the

game, which means the fictional time is a proportional mapping to the real time. For example, the time can flow faster than the real time as in a lot of sports games and grand strategy games, or slower than the real time such as in “Max Payne” when the “bullet time” is triggered. The game time can also be set at any period in the history, in an alternative universe, or in the future. He also mentioned cut-scenes as a special case, since it blocks player from playing the game, so it does not count as play time, but the fictional time still continues to move on (Juul, 2005).

Although Juul already made a very good discussion about the duality and projection of time, he did not stop there. He went on to discuss the chronology in games. He has identified that most video games are in chronological order, with some games feature flashbacks. He thought flashforward being highly problematic since it renders player’s action and effort useless, and it is generally not a good idea to do in an interactive artifact. He also pointed out the breaks and pauses in video games with coherent worlds, such as loading screens, can briefly break the continuity of play time while the continuity in fictional time is still maintained; while arcade games with different levels breaks the continuity in both play time and fictional time, although each individual level still maintains its own local continuity in time. Furthermore, there are games which the action speed does not match the flow speed of fictional time, such as in sports games, although the time flows at multiple times faster than play time, the action speed is only a bit faster than the speed in real world. At the end of the argument, Juul also defined a term I think would be very useful, which is “dead time”. It indicates the time when player have to perform uninteresting and unchallenging tasks in order to reach an objective in game, the perfect example of this would be the level grinding in role-playing games (Juul, 2005).

Eskelinen also discussed this topic in his article. He argues that there is only one-time scheme in games, which is “the movement from the beginning to the winning or some other situation” (Eskelinen, 2004). He also argues that the major temporal relationship is between user time and event time, and which events in game can be manipulated or cannot be manipulated are different from game to game (Eskelinen, 2004).

Eskelinen went on to proposed six categories to look into in terms of the difference in temporal relations in games. The six categories are order, frequency, speed,

duration, the time of action, and simultaneity. Order looks into whether the player can affect the order of events or not; frequency pays attention on the number of times which a certain event or action repeats itself; speed focuses on the pacing of the game and and who takes control of the pacing; duration examines how the game handles the duration; the time of acton checks the possibilities for the timing of triggering a certain action; and simultaneity mainly take a note about the increase and decrease of simultaneous events by the player over the course of gameplay (Eskelinen, 2004).

In Wei's (2010) research, she also presented an analytic structure which is similar to Eskelinen's idea. Her structure contains order, speed, frequency as described in Eskelinen's proposal, and polychrony that borrows Herman's (2002) idea of fuzzy temporality, which again itself is an extension to Genette's (1980) study on temporality based on the relationship between story time and discourse time.

The fuzzy temporality indicates that a set of events can be arranged in the following four ways: 1) Full ordering in which for any pair of two events in the sequence, there is an explicit temporal order between them. 2) Random ordering in which a event sequence that can be ordered in any of the random permutations of these events. 3) Alternative/multiple ordering in which all possible orderings are in a weighted random system. 4) Partial ordering in which "some events can be 'uniquely sequenced relative to all others, some only relative to some others, and some relative to none'" (Herman, 2002, cited by Wei, 2010).

Based on this concept, event sequence in a polychronic narrative system would be inexactly ordered just as the last three types of ordering mentioned in fuzzy temorality, or inexactly coded in which some events are placed at a fuzzy time on the timeline of the narrative (Wei, 2010: Herman, 2002).

2.4.4. Game Rules

Besides game time, Juul also discussed extensively on rules in games. Zimmerman discussed about rules in his other works, too, along with a handful of game design books, since it directly ties to gameplay. I will explore theories and discussions on rules in games briefly in the following.

Gameplay is the core of a game, without gameplay, the artifact cannot be called a “game”. But what is the core of gameplay? I think the answer lies in the rules of a game. From their own definitions, we can conclude that Caillois, Zimmerman and Juul along with many other scholars regarded rules to be an important part for games. This seems to be contradictory since games are meant to be fun, “play” are meant to be voluntary activity according to Caillois and Zimmerman, and in general, rules are something to regulate and to restrict our actions. With these restrictions, rules are generally not meant to be fun. However, this is absolutely not the case in games, since rules in games function more than just put limits on players’ actions. They are also specifying allowed actions, and make those otherwise meaningless action meaningful in games. In other words, “rules give games structure” (Juul, 2005).

2.4.4.1 Juul and Game Rules

In addition to see rules as a set of explicit instructions to tell players what actions are allowed and what are prohibited, so that players can understand and play along, Juul also viewed rules in a game as a state machine that receive player interactions and react accordingly. On the other hand, the state machine itself can be perceived as a tree of finite possibilities, playing a game is roughly equal to explore the tree of possibilities. Inside the tree of possibilities, there should be multiple outcomes, since a game need to have a quantifiable outcome. Among these different outcomes, there should be some outcomes better than the others, with the better ones harder to achieve than worse ones, so naturally the player would like to work towards the better outcomes. The obstacles he faced during this process are challenges, and the way the player tries to play the game in order to overcome these challenges are called gameplay (Juul, 2005). If rules of a game are not fun, then the game play cannot be anywhere close to interesting.

Juul went on further to categorize games into two types of rule structures based on how rules and challenges are implemented in the game, I have mentioned these two rule structures in my introduction section, now I would like to explain them in details. The first rule structure is games of emergence. traditional games are all in this form. This type of games set up rules for players, but do not place artificial challenges directly in front of players. Instead they induce challenges indirectly by other opponents which play under the same set of rule, or other random factors that keep changing during the course of gameplay. In other words, challenges are emerged during gameplay session.

The second rule structure, games of progression is rather new, but widely adopted by video games. These games have a bunch of artificial challenges directly set up one after another along the progress of the game, and they are to be experienced by players in the exact order they are supposed to be (Juul, 2005). Games of emergence provide more freedom in gameplay, more suitable for competitive gameplay and have more variability between different game sessions, hence more replayability. On the other hand, games of progress are based on stronger control of in game elements in attempt to provide an experience that is more close to film and novel, both of which have strong and strict control over every story, visual and sound elements. So games of progression can do better on narrative and controlled progression to provide a more exciting and interesting experience in a single play session, but once these artificial challenges are exhausted, there is little reason to revisit these games.

To enjoy a game means the player should engage and enjoy rules set by the game. Juul also discussed this topic, mainly on two forms of challenges players usually encounter during playing a game, namely choices and puzzles.

Unlike Crawford whom discussed this topic from an interactive narrative perspective, Juul discussed choices based on the criteria that if the player can enjoy the choice or not. He mentioned Sid Meier's opinion on what he called the "interesting choice". Sid Meier stated that the choices should be made in a way that no single choice is the best option, so players can have the freedom to pick anything they want; the choices are not equally good, so that players' choice can be meaningful; and the player should be informed before the choice is made.

On the challenges topic, Juul quoted the "aesthetics of mind" by Marcel Danesi. Danesi's discussion was about puzzles, and these discussions can in turn be applied on all challenges in video games. Danesi thought that the level of interest of a puzzle is inversely proportional to the complexity of the solution, and is proportional to the effort which the puzzle tries to hide its solution. Juul extended this argument with a note that different people has different knowledge base and experience, and this affects the way people approach challenges, so a single challenge is not equally challenging for different people (Juul, 2005).

2.4.4.2. Zimmerman and Salen on Game Rules

Just as Juul had an extensive discussion on rules in games, Zimmerman also dedicated on this topic. He authored a book along with Salen on this topic. Although often mistaken as a textbook for step-by-step detailed game design, this book provides a lot of useful theoretical insights for game design by discussing mainly about natures and approaches for designing game rules. I would like to discuss part of his opinions covered in his book.

Juul's discussion was originated from game rules, then expand on all impacts and effects these rules have brought to games later on. In contrast, Salen and Zimmerman focused their research on game rules themselves. They have pointed out that rules in games are tricky to identify, since games are usually complex and naturally very structural. So they came up with a way to identify games by stripping all "aesthetic quality" and "representational identity" from the game, and only look at the formal identity of the game (Salen & Zimmerman, 2004). This approach basically treats rules in games as an abstract formal system, and explore it in this form.

Salen and Zimmerman made a list of rule characteristics in games. Among these characteristics, several characteristics were also mentioned by Juul, while they went down a little further in exploring the nature of rules in games. First of all, rules are there to limit player actions just like how rules work in general; the second one is mentioned by Juul as well, which is the rule itself should be explicit and unambiguous, so that they are easy to follow; thirdly, the same set of rules are to be followed by all players take part in the game; the fourth characteristic is that rules are fixed and would not change during the game session; the fifth one is rules have its authority in games and meant to be followed; and the final one is that rules are repeatable and portable across any game sessions with any sets of players (Salen & Zimmerman, 2004). In practice, a lot of video games breaches third and fourth characteristics in favor of narrative or representation, or simply provide a better challenge since the artificial intelligence player is too incompetent. The best example for the former breach is Uncharted series, this game is known for very inconsistent rules throughout the game (player character has inconsistent jumping distance, hitbox size, damage tolerance, hitpoints and weapon damage across different parts of the game). But thanks to these inconsistent rules, this game has some impressive cinematic-like experiences and pacing control. Examples for the later breach

are the strategy games such as *Civilization* and *StarCraft*, the AI players in these games have a various of advantages such as accelerated resource gathering speed and research time over human players in order to make up for their incompetence against human players.

They also introduced three kinds of rules to gain a better understanding on games as a formal system. The first kind out of the three kinds of rules is operational rules which represents the general interpretation of rules: restrictions and guidelines to shape the gameplay by telling the player what can be done and what cannot; the second kind is the constituative rules that cover the mathematical, algorithmic, logical or other technical rules lies under what is presented to the players; and the last kind is the implicit rules that is not directly implemented by the game itself, but are often followed by most players in the game (Salen & Zimmerman), such as good manner to other players and not exploiting in game glitches and bugs in multiplayer games. The game rules discussed in this chapter up until this point are mainly operational rules.

2.4.5. Zimmerman and Salen on Uncertainty and Randomness

Uncertainty was also a topic on Salen and Zimmerman's list. They regarded it as a "central feature of every game" and "a key component to meaningful play" to illustrate its importance in games. Even in those more deterministic games without dice roll or algorithmic randomness, uncertainty is still an important element since players would not know how the game is played out and the outcome of the game prior to the game. If the outcome of a game is deterministic, every action involved in this game lost its meaning, hence no meaningful play. This kind of sense in games is related to the macro-level uncertainty according to Salen and Zimmerman, as opposed to the micro-level uncertainty which refers to the specific probability based processes inside the game system design. The relationship between uncertainty and meaningful play is also interesting, while uncertainty seems to take part of the power of choice and agency away from the player, it at the same time gives the player a feeling that their actions have more impact in a game due to the uncertain outcomes (Salen & Zimmerman, 2004).

Their discussion then included the tree types of decision-outcome relationships model by mathematician Epstein, which are certainty, uncertainty and risk (Epstein, 1977). While a game with completely certainty, as mentioned above, can hardly be

considered as a proper game, the other two categories are commonly seen in games and are what most people actually perceived as uncertainty. To be precise, risk is a type of uncertain situation which the player knows the uncertainty has a potential negative impact as well as a positive benefit, while uncertainty refers to a situation with completely randomness, the player has no idea about the possible outcomes at all. A mix of all three categories is what we usually expect from majority games (Salen & Zimmerman, 2004).

They have also discussed the perception of randomness from the players' perspective, which is a very interesting topic. Randomness is an important aspect for fun and enjoyment in games, however players' perception of randomness is not necessarily linked to the case that the nature of the game is really involving computer randomness or probability based dice rolls. From the game designer's perspective, to make games more enjoyable, it is more important to make players feel the randomness than actually put probability based mechanisms inside the game (Salen & Zimmerman, 2004).

The design of probability based mechanisms can also be tricky, because what players need is a sense of randomness, which is not necessarily to be true probability based dice game. So before designing a mechanism involving random number generator, it is important to know the connection between the probability and player choices as well as the outcome. Although probability can bring uncertainty and chaos into a game, it can also reduce unknown uncertainties to known risks and increase the overall certainty while still maintain the feeling of randomness to players, if we understand how probability works mathematically (Salen & Zimmerman, 2004).

Also, there was another interesting observation related to the discussion on feeling of randomness from players' perspective: the actual probability distribution often does not match what player's expectation of how the randomness should work. First of all, computer does not generate real random numbers, what they do is to produce pseudo-random numbers using different algorithms combined with different seeds to mimic a "true" random number generator. So it is not strange to see a game stuck with several specific "random" outcomes showing up more often than others due to a bad random number generating algorithm design. There are also situations which players called it a game of chance, but actually it is not a game purely depending on probability,

and players often use strategies to manipulate the chance in favor of their own (Salen & Zimmerman, 2004).

The last one is maybe the most important one, which is players' understanding of how randomness works in a specific game is rarely correct, and going for some wrong expectations from the outcomes of a game of chance. Such as some of them think betting on a high gain, low probability outcome is more profitable than a low gain, high probability outcome in a long shot; some of them do not think the successive events of chance as a product of probabilities but as addictive; some of them think the chance for a win after a loss would increase despite the fact that probability for each event is independent; some of them believe the positive outcome have a larger chance to happen, given a positive and a negative outcome both have very unlikely probability to happen; some of them believe negative outcomes would not likely to happen repeatedly; and finally, the belief in luck (Salen & Zimmerman, 2004). A good utilization of game of chance should take these factors in mind, and design strategies to guide the expectations of players, or even fulfill their expectations by tweaking the game system.

Chapter 3.
Event Planning System Details

3.1. Planning, STRIPS Representation and STRIPS Planner

3.1.1. Planning

The representation of knowledge base and actions are the basics of classic artificial intelligence research. They are developed to represent the real world more precisely by adding “changes” into the representation. Based on the knowledge base - action representations, we can represent a variety of problems as how to arrange your actions to make a specific condition true, the progress of finding the correct arrangement of actions is called planning. (Brachman, 2004)

Planning is a technique that developed during research in the state-space search, theorem proving and control theory, and it is currently widely used in robotics, scheduling, as well as other domains (Russell & Norvig, 2010). Planning is considered to be the central focus of the research of knowledge representation in symbolic artificial intelligence. It is a very useful way for an artificial intelligent agent to utilize the knowledge base it has to reason in a more “realistic” world, where actions and their consequences are the central focus. It can be very powerful in solving some categories of problems when combined with other proper techniques.

Situation calculus was developed to adopt the requirement of representing a changing world. Comparing to the First Order Logic (FOL) which only represents one static situation, situation calculus has no current state, it can hold multiple situations in one formula which is called a fluent. It can also represent an action using precondition and effect axioms (each composed by one or multiple fluents). In addition, it can even represent the fluents unaffected using frame axioms, to shape all dynamics in a specific situation. With the situation calculus, it is possible to represent and reason about planning in formal language. (Brachman, 2004)

However, if we want to solve more complex problems with planning, the situation calculus becomes overly complicated, making it very difficult to use in more realistic

situations due its large overhead and the requirement to maintain too much unnecessary information and details. So we have to find a simpler way to improve the representation.

3.1.2. STRIPS Planning and STRIPS Representation

Stanford Research Institute Problem Solver (STRIPS) was developed at SRI International in 1971 as a derivation from work on the “Shakey” robot in the 1960s, and it was originally developed as a reasoning and problem solving system for autonomous robots. It is a planning based system works in a world with symbolic representations of world knowledge, where only the current state of the world, the desired goal state and a set of actions to change the state of the world are specified, the past states of the world, as well as other knowledge are irrelevant in this system. (Orkin, 2006)

STRIPS is one of the most important early systems in this domain. Aside from its contribution as a planning system, its representation of world, which is called the STRIPS representation, is considered as an important alternative representation to the overly complex pure situation calculus. (Brachman, 2004)

The STRIPS representation shares some similar traits to situation calculus, mainly in the basic representations. Such as STRIPS representation also contains a list of preconditions and a list of postconditions for each action just as the preconditions and effect axioms in situation calculus. Individual conditions in STRIPS representation also represented in a similar manner as axioms in situation calculus. For example, agent A is holding a box can be represented as *holding(A, box)* both in situation calculus and STRIPS representation.

The main difference of STRIPS representation is that it assumes the following situations for the sake of simplification: only one action occurs at a time, action occurs instantaneously (i.e.: the time cost of actions is not take into consideration), and nothing changes the world state other than actions taken. These three assumptions are called the STRIPS assumption. Although these assumptions seem to ignore some important aspects of the real world, they still work quite well even with complex real world environment set-up. The STRIPS assumptions shape the world which the original STRIPS can work in and remove information about the past and possible future situations from the representation. The world in STRIPS is only represented as current

state, the history of the world is not stored, and we deal with a single world state at a time, which makes the representation easier to implement complex problems than situation calculus.

The STRIPS representation also removes further unnecessary details and operators presented in situation calculus formulas in order to make the whole representation more approachable and more efficient. The whole representation of situations in the world is also simplified as the following components: an initial state, a goal and a set of actions.

The initial state represents the situation before any actions taken, it contains a set of conditions which covers all information potentially useful to the planning process. The goal specifies a desirable situation which the intelligent agent tries to reach through the planning process, it contains a set of conditions describing the desirable situation, when the set of goal conditions become a subset of world state, then the goal is met. Reaching the goal marks the successful end of the planning process. Actions are the operations to be used by intelligent agents to change the current state of the world, they are the only way to change the state of world. Each action is represented by a set of preconditions and a set of postconditions. The list of preconditions contains a set of conditions which must be met in order to execute the specified action. The list of postconditions contains a set of conditions which describe the consequences of the action, it is separated into two lists: an addition list and a delete list, the addition list specifies a set of conditions to be added to the world state after this action has carried out, and the delete list specifies a set of conditions to be removed from the world state after this action has carried out.

This is a simple example of an action described in STRIPS representation:

Action: eat pizza
Precondition: On(Pizza, Table), Hungry(A)
Postcondition:
Add List: Full(A)
Del List: Hungry(A), On(Pizza, Table)

In this example, the action “eat pizza” requires both conditions which “the pizza is on the table” (means there is a pizza to eat) and “agent A is hungry” (means he/she actually needs to eat something) to be true to execute. When the action is carried out, the condition “agent A is full” is added to the current situation. At the same time, conditions “agent A is hungry” and “the pizza is on the table” has been removed from the current situation. Then the world has been changed by the consequences of the action.

The process of how actions work in the planning system is explained as follows: During the execution of planning system, when all preconditions of an action is met for current state of the world, it will be placed into the pool of all possible actions. If a certain action has been selected by the system for execution, conditions in add list are added to the current state, and conditions in delete list are removed from the current state. Then the world is now affected by the consequences of the action just taken.

To demonstrate how actions actually change the world, let us take previous action “eat pizza” as an example to show how it changes the world. First, the world before the action is taken is defined as follows:

World: On(Pizza, Table), Hungry(A), On(Coke, Table), In(B, Library)

Note that there are two conditions in the world state which is not related to the preconditions of the “eat pizza” action. Now we execute this action, and remove conditions *Hungry(A)* and *On(Pizza, Table)* from the world. Since A has eaten the pizza, so the pizza is not existing and A is not hungry anymore, then we can add the condition Full(A) to the world. Since A has already fed by pizza, now we have the world that:

World: Full(A), On(Coke, Table), In(B, Library)

Note that only conditions appear in the postcondition lists of the executed action is affected, the other conditions not related to the execution of current action is left untouched.

3.1.3. Progressive Planning Algorithm

The basic algorithm for STRIPS planning is straightforward. There are basically two ways of doing the basic planning: progressive and regressive. Progressive planning works from the initial state towards the goal, and regressive planning works backwards, from the goal toward the initial state. For this project, I have chosen to use the progressive planning algorithm since it is most intuitive way to do, plus there are other important reasons for this, which I will be explain in the next chapter. The pseudo-code for progressive (forward state-space) search STRIPS planning algorithm is shown as follows:

```
//Start of pseudo-code
Plan(World, Goal):
   If Goal is a subset of World then return empty

      for each action(actionName, precondition, addlist, dellist) which pre is
      satisfied by World:
         World2 = World + addlist - dellist
         Plan = Plan(World2, Goal)
         If Plan <> fail then return Action:Plan
      End for

return Fail
//End of pseudo-code
```
(Brachman, 2004)

This algorithm loops on all available actions, it checks the set of preconditions for each of these actions and find ones which preconditions matches the current world state (i.e.: the set of preconditions is a subset of the set of conditions in current world state), and try to execute each satisfied action. Then again for each world state created by each of these actions, the algorithm would call itself again and repeat the above steps. It runs recursively until a series of actions lead to the goal is found, or the fail condition is met. This algorithm does not find an optimized solution, but will eventually find a solution if it exists, or return fail if such solution is non-existent.

Here is an example to illustrate the flow of this algorithm. Imagine we have a situation in which a person called A is hungry, and we need to reach a goal which A is not hungry. We also have a set of available options for him to resolve the “hungry” problem, such as he can eat a pizza or he can eat some rice. But if he needs to eat something, the corresponding food must exist in this world, or he would have nothing to eat under that current situation. In this case, he must first look for available options in the current situation, and then look for further actions which are made available by the previous action. For example, he looks up in his phone and finds a phone number for a pizza restaurant, then he calls the number and orders a pizza, then he can wait for the delivery of the pizza. But if there is anything that went wrong during this consequence, such as if he cannot find the proper phone number (then the action of calling would not happen), or if the restaurant is closed today (then he cannot order a pizza), then he is forced to retreat from this plan and backtrack steps to find other options. He would keep doing this until he finds a way to get some food to eat to solve the “hungry” problem, or he would run out of options, and fails to find something to eat, thus fails the plan (which means such a plan does not exist).

There is one issue with this basic algorithm worth mentioning. The algorithm does not prevent itself from picking repetitive actions from current available actions during planning, as long as preconditions of those actions are satisfied by current situation, this is known as the Sussman Anomaly. (Nilsson, 1998) For example, in a planning progress, such an action sequence could happen: the agent picked up a box from ground, and then it puts the box down, and then picks it up again immediately. The most efficient way to prevent this is to use a regressive algorithm, but since I choose to use progressive planning, I needed to find another way to reduce the frequency of this anomaly through the design of actions, by embedding additional preconditions into these actions to prevent them from executing repetitive action sequences consecutively (i.e.: they would not behave as what I mentioned in the example, even if they are wasting actions, they’ll do it with a larger loop).

3.1.4. A* Algorithm

However, since the algorithm is executing steps recursively for every action, it is somewhat inefficient, and it would become impractical with this basic algorithm if the problem scale is large. For my project, although the scale of database is not as large as

real world problems, it still needs to run in real-time, which means some heuristic is needed for the algorithm to perform well. That is the reason why I have to introduce another heuristic algorithm to work with the planner, to simplify the situation.

The A* algorithm is the most famous heuristic search, it is a very popular search algorithm for pathfinding tasks, especially in robotics and video games. In this game prototype, during the murder, the murderer needs to traverse the game space and execute actions in order to commit the murder. This is where the A* algorithm comes in for the murderer to find its way from action to action.

The algorithm itself is very simple, it operates in a map where every path has a cost assigned to them, and it finds the solution with lowest cost from start to goal. It calculates and keeps tracking of *g(n)*, the cost to reach a specific node *n*, and estimates *h(n)*, the estimated cost of the cheapest path from node n to goal, then we have an optimal solution $f(n) = g(n) + h(n)$ (Russel and Norvig, 2010). For this algorithm to work, we need a good estimated cost *h(n).* In this game prototype, I would like to make a list of direct distances between any two locations on the map (measured by the number of rooms between these two locations), and use these values as an *h(n)*. As the map in this game prototype is not terribly large, this can simplify the complexity of calculation significantly since there is only one lookup in the list to get *h(n)* for any given node. (Russell & Norvig, 2010)

The above explanations conclude the basic algorithm used for the core mechanism of the planning system. The planning system will utilize this algorithm mainly to create murders in the storyline of the game prototype, which is the main mystery for players to solve. The motivation behind this decision will be discussed in a later section. However, the STRIPS planning algorithm alone is not enough to fulfill all requirements in this project, further modifications and additions are needed, these are covered in the next section.

3.2. The Probability Based Event Scheduler

As I have stated in the previous section, the core mechanism of the event planning system for generating the murder scenario is based on the STRIPS using

progressive algorithm assisted by A* algorithm for calculating paths to be traversed by the agent (the murderer).

The event planning system runs at a specific time inside the storyline and generates new murders for the game to provide varieties to the core gameplay contents. However, generating murders from a pre-scripted previous situation every time would not ensure enough variety. So I decided to make the situation just before the murder happens as a variable as well, that is the intention for the probability based event scheduler.

The purpose of this additional event scheduler itself is to generate a sequence of events which is both reasonable and “interesting”: This event scheduler has a much simpler algorithm design and has less restrictions in creating sequences of events than the event planning system. At the same time, it would generate sequences of events which these events inside the sequence does not contradict each other. As a part of the system, the role for this scheduler is an additional layer which is responsible for scheduling the sequence of events that take place before the murder happens. Consequences of these events adds another layer of complexity and variety to the murder generated by planning system. This section will explain the probability based event scheduler in detail.

The probability based event scheduler, as the name suggests, chooses and arranges events from its own event database using a simple probability based, or “random” approach. The scheduler operates inside the same world as the event planning system, so the representation and operation of events for the scheduler are similar to the STRIPS representation used in the event planning system, which are basically actions with its own sets of preconditions and post conditions, and their consequences are also reflected in the world by adding/removing conditions inside the world knowledge base. However, they are not exactly the same since the scheduler is running on probability based approach, so there are three extra attributes assigned to each events(actions) in order to make the scheduler work.

The first extra attribute is cost. Unlike STRIPS, this probability based approach does not work like a planner which attempts to calculate a plan towards a goal state, instead, the goal is simplified as a cost limit. The cost of each event represents the time

needed for this individual event to carry out, and the cost limit represents the total length of time available for actions. The algorithm would continue to pick and execute new events while keep tracking their costs, until the cost limit is met or exceeded.

The second extra attribute is base weight. The selection of events can be completely random if each event has an equal chance to be chosen, but complete randomness cannot accurately match story development and setting of characters. So it would be better if a weight based mechanism is introduced, in which each event has a weight assigned, indicating how likely an event is going to happen for each character. Based on these weights, some events are more likely to occur than the other ones. The selection of events is based on the weight of each available events, the larger the weight, the more likely it going to be chosen.

The weight mechanism is represented as follows: there is a base weight directly attached to every event to represent the likelihood of each event. These base weights can be changed by multipliers. These multipliers are attached to all characters to represent their personal preferences and hobbies, if a character is interested in something, they are more likely to do certain activities.

How does multipliers for weights actually works? This is when the third extra attribute “category” come into use. Each individual event in the database for probability based event scheduler has a category tag attached, which indicate which category does this event fall into. The set of weights attached to each character is also in a format which maps a weight to a single category tag, a multiplier of 2 means the character likes to involve in events in this category, a multiplier of 0.5 means the character does not like to involve in these events, a multiplier of 0 means the character would not involve in the specified events at all, and a multiplier of 1 represents a neutral attitude. For example, the multiplier set *{[art, 1], [reading, 0], [idle, 2], [eating, 2], [sports, 0.5]}* means the character likes to eating, sleeping/doing nothing, does not like to do sports, and would not read anything at all, also, he/she has a neutral attitude towards arts. When the scheduler executes, the base weight would be multiplied by each of the multipliers according to their corresponding categories, then the probability based selection would carry out upon calculated new weights.

Now that we have discussed the representations of weight and multiplier, we can explain how exactly the weight based selection works. Let us imagine that we have a list of candidate actions, we go through every one of them and calculate their weights by multiplying its base weight with corresponding multiplier associated with individual character, then we calculate an accumulated weight for this action by adding current weight to previous accumulated weight, and assign this weight to this action as its "lottery number". Once every action in the list has its "lottery number", we generate a random number in the interval (0, total accumulated weight), and see which interval the generated number falls in. The m th action in the list gets selected if the generated number is greater than the "lottery number" of the (m-1) th action and smaller than the "lottery number" of the m th action.

The overall algorithm is described as follows:

```
//Start of pseudo-code
PBPlan(World, Cost limit, Characters, Actions):
   for each character({category:multipliers}):
  While currentCost < costLimit:

      for each action(actionName, precondition, addlist, dellist, cost, weight,
      category)
      which Pre is satisfied by World:

         TotalAccumulatedWeight += weight * (category:multiplier)
         action.lotteryNumber = current TotalAccumulatedWeight

      randomNum(0, TotalAccumulatedWeight)

      selectedAction = action m which action m-1.lotteryNumber < randomNum <
      action m.lotteryNumber

      World = World + selectedAction.addlist - selectedAction.dellist
      currentCost += selectedAction.cost
      selectedAction:Plan
```

Return Plan
//end of pseudo-code

Here is a detailed explanation about the procedure of this algorithm. This algorithm takes a world representation, a goal state (which is a cost limit), a list of events (actions) and a list of characters as input. The algorithm runs for all characters in the list, for each character, it first searches for actions which their lists of preconditions can be satisfied by current state of world (i.e.: actions that can be executed now) and form a list of all currently available actions; then it selects an action from the list using the weighted probability based selection described before; once the action is selected, it would be executed to change the situation of current world, the cost of this action would then add up with the previous cumulative cost to calculate current cumulative cost to complete the loop for this single action; then the algorithm loops and uses the same procedure to determine the next action while keep tracking the cumulative cost; when the cumulative cost have reached or exceeded the specified cost limit, the algorithm finishes running and the final event sequence is complete.

In the game prototype, the probability based event scheduler is implemented to handle all generative events that is not directly related to the murder itself. Including events associated with NPCs during first player interactive session in day one, the events parallel to the murder, and events during investigation sessions. Some of these events do not impact the game world (i.e.: do not change states which are critical to the investigation of murder mysteries), while some of them are indeed very critical to player's investigation.

3.3. Summary

Algorithms behind this game prototype can be divided into two main parts. One of which is the event planning system, which is used for organizing murder events that is the core of this game prototype both in terms of narrative and gameplay. The event planning system is based on STRIPS representation, and uses simple progressive planning approach in order to find a suitable plan for the murder, while using A* algorithm to find a route to connect actions in the generated plan. On the other hand, the probability based event scheduler pulls events from a separate event database to the

event planner with weighted probability based generators, and form a sequence of events to cover needs other than the murder itself. Making good use of these two algorithms is the key to this project, we will discuss how exactly they are used in the next chapter.

Chapter 4.
Game Prototype Details

4.1. Basic Information

This section includes basic information about game genre, presentation and gameplay, as well as the development tools used to make this prototype.

4.1.1. Background Information

As I have described in introduction section, the game prototype is designed to have a similar core gameplay to simple adventure games, which can be boiled down to collect items and find where to use them on, plus a few decision making sequences. The narrative is driven forward mainly by satisfying certain trigger conditions through player interaction either directly or indirectly. I choose to follow the methodology of classic adventure games mainly because I would like to focus all attentions on my research question: “can we improve replayability of video games using a purely narrative approach?” So I would like to keep other elements in this prototype at minimum. The limitation of time and resources is another reason for this decision.

On the other hand, although this prototype has a similar core gameplay mechanism to classic adventure games, it does not feature a similar control scheme to the classic adventure games such as command input in early text based adventure games and point-and-click in graphical adventure games. This prototype is using a direct control scheme similar to role-playing games, in which the player controls the movement of the character using direction keys directly, interactions with environment and non-player characters are done by using “Enter” key. The reason of using direct control is to simplify the control scheme and make it more intuitive to use, and also more convenient to develop since it comes by default in the development tool I have used.

The graphical presentation of this game prototype resembles Japanese style role-playing games in the SNES and MD era since it is made by the RPG Maker. Although the dot matrix based 2D graphic does not possess the same presentation power of the 3D games today, I think its clarity and presentation is good enough to

showcase my thoughts on the research problem. The sound design is minimal, apart from some sound effects, there is no music, sound and voices in this prototype.

Figure 4-1: A screenshot of the game prototype

Similar to many adventure games, the goal for the player in this game prototype is to find out an answer to a certain question, there are multiple outcomes depending on how the player performs in game. Also, the research focus of this game prototype is narrative, a story cannot be omitted if I want to talk about narrative. So, in the next section, I will talk about everything around the story and endings of this game prototype.

4.1.2. Development Environment

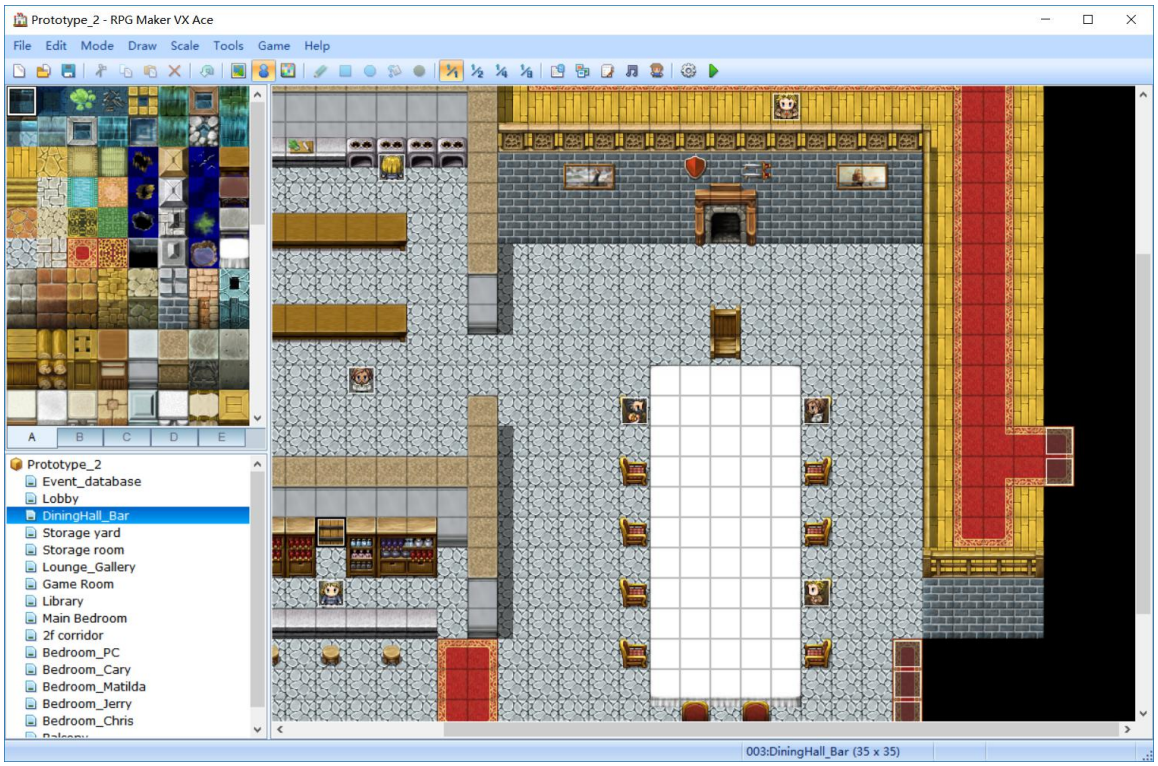

Figure 4-2: The interface of RPG Maker VX ACE

This prototype is developed using RPG Maker VX ACE, which is a game development tool which focuses in, as the name suggests, making role-playing games. Specifically, Japanese style role playing games those similar to early Dragon Quest and Final Fantasy games on NES and SNES. It can make 2D sprites based maps with ease using its map editor. It also comes with a capable event editor, a database management system and animation management system for combat and meta data management, and other misc functions for making an RPG game. This development tool is not only capable of making a trivial RPG without any coding, but it also comes with the support of scripting with its scripting language called RGSS 3 (Ruby Game Scripting System 3), which is based on standard Ruby 1.9.2, extended with its own class structure and built-in libraries.

Although RPG Maker VX ACE is a tool for making role-playing games, it suits the majority of my needs perfectly, since only a subset of features provided by this tool are sufficient to make this prototype. The main advantage of this tool is that it comes with a set of default assets which any user whom purchased this tool can use without

limitations, as well as a huge community which its member created a lot of free assets for all RPG Maker users to play with. These feature can save a lot of time for me from starting with scratch. There are also a lot of basic functions such as movement control, title screen, game over screen, save and load functions which are provided by default. The tool also comes with a very capable map editor and event editor, which means I can design a considerable part of the prototype with ease. The scripting language is a widely used programming language that is very quick to learn, allowing me to make custom scripts to suit my own needs, such as the core event planning system.

4.2. Game Narrative, Gameplay Flow and Rules

In this section, I will explain everything around the story of this game prototype: the background settings, characters, the plot itself, and endings of the story.

4.2.1. Background Setup

The setup for this game prototype is inspired by Agatha Christie's "*And then there were none*". Just as in the novel, the story of this game prototype occurs on an isolated island with hardly any way to make contact to the outside world. There are a number of people on the isolated island for different purposes, and they would stay there for 3 days in a situation which help from the outside cannot reach the island. During these days, one or two murders happen depending on main character's action. The task for the main character is to find out who is the murderer. According to random factors as well as player's interaction, the game would reach different endings. Some of these endings would provide detailed stories about the main character himself, or about other characters.

The detailed background setting is as follows. The story happens on an island about 50 kilometers away from the mainland, it is owned privately by a then-famous successful merchant called Edward Cromwell. Edward wrote to 3 people with different backgrounds and declared that his health condition is fading and he is willing to part his fortune to them, so he has asked them to come over to the island. The main character, a freelance journalist, also received a letter by an unknown sender which claimed that he is an old friend of his father, whom died years ago. The anonymous stranger knew something would happen on Edward's island, and gave him an invitation that should be

able to grant him access to the island as well. When they arrived at the island, the three guests are told that they have to compete for the fortune by staying on the island in the next 3 days, and it is only up to Edward to decide which one would take all of the fortune. During the 3 days, there is no way for them to get off the island, and they also cannot get help from the outside due to extreme weather. So they are on their own during this time period.

The intention of this background setting is pretty straightforward: to establish a classic isolated environment as the stage for the murder mystery, just like what was in "*And then there were none*". The reason to choose this situation is because an isolated place excludes the interference of outside world and limits the story to a rather small set of elements. It is a viable strategy for this prototype, because the time and cost only allow me to build a small scale prototype with a relatively simple set of characters, items and events. On the other hand, the success of the novel "*And then there were none*" along with other media artifacts with such background settings have proved this setting is a viable tactic. Also, the great story in "*And then there were none*" indicated that this kind of enclosed environment can still capable of delivering a very solid mystery story. This is another reason for me to select such a background.

4.2.2. Characters

There are six characters in this story including the main character, they are all from different backgrounds, in which three of them are there for the similar reason: to compete for Edward's fortune. The other three character includes Edward himself, Edward's maid, and last but not least, the main character. The design principle of these characters is to make their backgrounds and personalities as diverse as possible. Since there are only 5 non-player characters, this is relatively easy to achieve.

The main character is a freelance journalist, he is recently out of work for a while, and eager to get his hands on a valuable piece of news. He travels to the island because a mysterious letter tells him that there will be something "interesting" on Edward's island. He is not a candidate of the fortune, and he does not know anyone on the island prior to his visit. However, his father has some connection with Edward. This character is also player's avatar in the game, he will have a lot of dialog lines instead of being a "silent" hero.

The owner of the island is Edward Cromwell (Ed). He is an old retired merchant, he has a very legendary past, but he has disappeared from the news focus for about 20 years. The island and the mansion on the island is solely owned by him. Some time prior to the start of the story he claimed he was suffering from a serious disease and would pass away in a few months. So he invited three guests to his mansion and promised one of them that they can get a huge fortune from him. His intention is somewhat mysterious and the player might discover about it during the gameplay.

Matilda is a maid works in Edwards' mansion. She is around 20 years old. She started to serve Edward several years ago. Edward has ordered the other servants to leave 3 years ago with an unknown reason, so she is the only maid in the mansion now. She is an orphan, and her background remains a myth. She is very polite to anyone she meets, and loves her job with a passion, she is also well educated, which is uncommon for a servant. Her background is also a myth which the player can discover during the gameplay.

The three guests invited to the island are all from totally different backgrounds. One of them is Jerry Tigers, a farm owner in his mid-40s from the South East, he has some tricks up his sleeve in terms of inventions, he owns several potentially profitable patents but lack the fund to turn them into a profit. He always has a very strong yearning for a luxurious life like a real high society man. He is poorly-educated, short tempered and rude, he hates people which are both far richer than him and poorer than him.

The second guest is Cary Sherman. He is around 40 years old. He is a battle-worn war hero from the last war. He suffers from serious PTSD, and he is heavily addicted to alcohol and has an issue with sleeping. He owes a lot of money from the mafia and needs to pay them back in the near future. He does not like to talk to other people. When he is not drunk, he maintains a calm mind and has a very sharp situational awareness, but when he drunk, he is very aggressive and potentially dangerous.

The last guest is Christopher Voroshilov (Chris for short), a 51-year-old politician with an immigrant background. He climbed to his position using tactics and promises which favor the rights for the immigrants in this country. Now he plans to climb higher, but cannot get enough funding. Edward is very important to him, since Edward has the

potential to possibly support him with enough funds. He is a very well educated and socialized person, he loves talk with people and can give very persuasive and moving talks, he is also open to different ideas and people with different backgrounds. However, he is very selfish and greedy, when it comes to power and money, he would become short sighted.

All the 5 non-player characters can be a potential victim of murder, but Edward can only be killed in the second murder attempt since I decided that it would be better to keep Edward alive for a bit longer for the sake of story development. In terms of the murderer, only 4 out of the 5 non-player characters can be a potential murderer, with the exception of Edward. The different murderer / victims can result in a different branch in the story, but these branches are predetermined in the story, in order to simplify the system and deliver a more detailed and complex story.

4.2.3. Game Flow

Before explaining the detailed story and branches of the game prototype, let us take a look at the flow of the game prototype first. The actual in game time frame is roughly 3 days, which is the time frame which the 3 guests competing for the fortune need to spend on the island. During each day, there is at least one section for player to interact with the environment and non-player characters. Outside these sections, the game progresses with automatically unfolding events to tell the majority part of the story.

The game begins at day 1. The day starts in the afternoon, which is the time when the journalist arrives on the island along with the 3 candidates for the fortune. The very first event in this day serves as an introductory sequence that presents the background story and settings, as well as characters to the player. After the introductory sequence, there is the first section which player can move freely and interact with the game world. During this section, the player can talk to them and join them, as well as find things to do on their own. The outcome of this section will not actually change any states in the game world, it acts as another “introductory” section for the players in terms of interaction. In this section, the player can understand the basic controls and gameplay of this game during this section, while have a glance of the probability based event scheduler, since all non-player characters’ behaviors is generated by this scheduler. After the first interactive section ends, there is the dinner event, in which the game

introduces the final character Edward. The first murder then occurs after that, in day one night.

In the morning of day 2, there would be an event which the murder is discovered. After that, the player has a section to collect evidences by investigating the environment as well as interrogate all the remaining non-player characters. When the investigation section ends, the game advances to the evening sequence, in which the journalist reports to Edward about the investigation, and have some conversations with other non-player characters. At night, there is a second murder attempt that is either successful, causing a second victim, or ends in a fail if certain condition is met. Then the time advances to day 3.

During day 3, the player is given a final investigation chance, in which they do the similar investigations as in the first one. If the player cannot find out enough evidence to find out the murder, the games ends in a fail; if the player finds sufficient evidences, they have to identify and confront the murderer in the conviction section, then the game advances to the ending sequence, during which most of the mysteries would be solved.

Here is a flow chart of the game prototype.

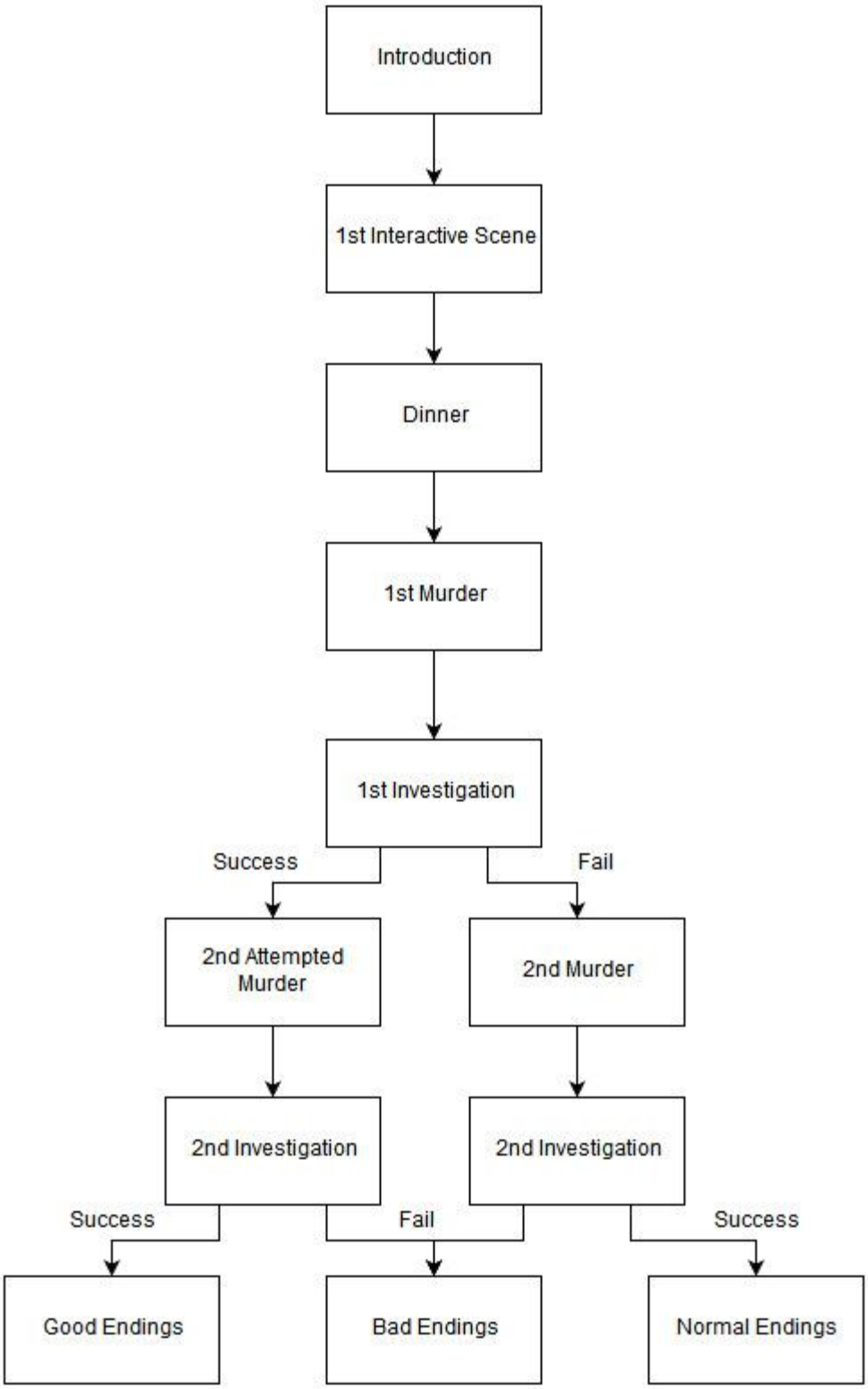

Figure 4-3: Flowchart for the game prototype, individual endings are determined by who is murder and who are victims

The flow of the game is not very complex in terms of branches and narrative structure. Some theorists maintain that more branches in narrative lead to a better interactive experience. Crawford suggests that you can measure the interactive narrative with the number of meaningful choices a story can have. By meaningful, he means the real ones which can result in a different branch (Crawford, 2003). I did not follow this design metric when designing of player choices, instead, I choose to keep the number of branches in scripted dialogue scenes at a manageable level in order to decrease the development complexity and time significantly. In addition, I also have faith in the event planning system along with the probability based event scheduler, which is capable of generating a relatively large number of possible event sequences, I will discuss this in details in the discussion chapter.

4.2.4. Branching

There are mainly four branches in the narrative of this game prototype, two of which have a specific murderer, and both of them have one ending each. The other two branches have 3 endings each. There are also four bad endings. So the total number of endings is 12, if we do not count in the different combination of victims (since who the victim is does not affect the content of ending, with the exception of Ed). The main branching of story is not totally determined by the user choice, but hugely by the factor that who is the murderer and who is the victim. For example, one branch requires one of the three candidates to be the murderer, and Edward must be alive, while another requires Matilda to be the murderer, etc. Some of the branches and endings are meant to be reached based on player's performance. Among these endings, some of them are the "good endings", in which the murderer is discovered and caught, and more stories about certain characters are revealed. In order to reach these "good endings", the player need to find enough evidence during first investigation session, so that the second murder attempt would end in a failure. There are also normal endings, in which the murder mystery is solved and the murderer is identified, but the murderer manages to get away and cannot be found anymore, or commits suicide, resulting in a situation that is less than perfect, while the goal of the main character is partially achieved. These endings can be achieved if the player cannot find enough evidence in the first investigation section and the second murder attempt is a success. There are also bad endings, they serve as "game over" states. Some of them are result of not discovering

enough evidence during final investigation section, and the others are result of convicting the wrong person during the conviction section.

The story itself is basically established around the murder, but in some branches, deeper background stories about some of the characters are told. Background stories of some NPCs would be told in different endings. In order to view all of them, the player need to go through the game several times. This is a traditional approach to make the game worth visiting multiple times after beaten once, which I have described in the introduction section. This approach would work together with the planning approach to improve the replay experience.

Here are two sample screenshots from two different endings, featuring different murderers.

Figure 4-4: Screenshots from 2 different endings

4.2.5. Additional Rules for Gameplay

As described before, the player's ultimate goal is to find out who is the murderer. This is done by collect evidences and testimonies during investigation sections, but there are certain rules the player must follow.

First of all, there is a limit for the investigation section, the player cannot wander for unlimited time in this section, there is a limitation implemented to mimic a time limit. The limitation is bond with steps player take, when the player takes a step, the step counter goes up by one, when the steps limitation is reached, the investigation section would end upon player exit current location. This option is cheaper to implement and more graceful than an actual real time counter, it can give the player more time to think about their action, as long as they do not move, the time in the game can be seen as stopped. The decision which end the investigation after player exit current location instead of end immediately is also a way to be more graceful to the player, so player can have more freedom to explore this final location.

Also, the player is required to find enough evidence during the first investigation section in order to gain access to best outcomes (the "good endings" mentioned in previous section). However, the player is not required to find all of the evidences, instead, they can omit a few pieces of evidence and still be granted access to better endings (the "normal endings" mentioned in previous section). However, during the final confrontation, the player must choose carefully, if the player confronted the wrong NPC, the game would end immediately with worst endings (the "bad endings" mentioned in previous section).

4.3. Planning System Managed Scenes and Scripted Scenes

Generally speaking, all events in this game prototype which player can see in interactive sections are outcomes generated by the event planning system or probability based planning system, while all long dialogues are scripted scenes. The details is shown as follows.

The first interaction section is directed by probability based planner only, the reason behind this is to have a section which gives the user a glance of how the planner

generated section looks like (although they might not know this in advance), and also impacts the story by affecting the relationships between characters. Also, it gives the returning players a hint that this game is not composed by scripted events only. Finally, this section is also served as a test for the later plannings.

The two murders in this game prototype is the main events that involving both planning systems. In each murder, the probability based planning system generates a series of events for each non-player character first. The main reason to do this is to randomize their locations, making the whole planning phase to have more varieties. The other reason is to mimic a series of behavior which individual characters would do if no murder is involved. It is an attempt to add some believable factors to the game world. After the probability based planning system finishes its job, the system picks a murderer and a victim based on the evaluation of relationships between characters generated from previous interactive section, and use STRIPS planning to generate the whole murder, including how did the murderer approach the victim, how did the murderer kill the victim (weapon, location, direction, etc.), how did the murderer move the body, how did the murderer leave the scene of crime, and how did the murderer dump the weapon and other evidences. At the same time, the other non-player characters also carry out their own plans, they can observe each other when they come across each other, including the murderer and victim, and they would report these meetings to player during investigation sections. The STRIPS planner is an attempt to emulate the execution of the whole murder, just as the ones written by people. Some detailed mechanisms will be explained in the next section.

Here is a sample sceenshot featuring a non-player character reporting his encounter with Jerry to the player.

Figure 4-5: NPC reporting a meeting to the player

The long dialogues in this game prototype are all scripted events. These events are all non-interactive, the player is meant to sit back and watch its unfold. These events need to be scripted since the planning system in this game prototype cannot handle the generation of complex content, only these scripted events can handle complex setup and development of the story. There are a few branches among these scripted events, the decision of choice is not made explicitly by the player, instead, it is made by the system based on how good the player performs in investigation section in terms of whether you get the “good endings” or not, as well as pure randomized factors in terms of which main story branch you get. There is a difference between the two, the branches based on player behavior would have a “good” and “bad” branch difference in them to encourage the player to behave better in the game, but the branches based on randomized factors are meant to be “equal”, they are just different stories with different contents to unfold, there should be no branches in which the situation is better than the other.

4.4. Planning System Managed Scenes

One of the core parts of this project is the murder sequence controlled by the event planning system as well as every decision the system made related to the murder. I have explained the algorithmic details about the event planning system in the previous chapter, but how does the event planner work in this specific game? How does the output look like in the actual game? Does the output from this planner actually make sense? In this section, I would like to explain the mechanism how outcomes from both planning systems would actually look like in the game.

As I described in the previous chapter, the planner is built for an agent to operate towards a goal in a world. In this specific game prototype situation, non-player characters (excluding Ed, whom does not control by either planning systems) can be treated as individual agents, and they have their different goals throughout the game. The world knowledge base is everything related to these tasks in the environment around them.

Let us start by describing the world. In this game prototype, the whole world is the mansion on the island. In the actual game, there are 5 maps covering the public areas, which are the dining room / bar / kitchen / 2nd floor west wing map, the lounge / gallery map, the game room map, the balcony map and the library map. There are also 2 small maps for underground storage room and outdoor storage area, 2 hub maps including the 1st floor lobby and 2nd floor east wing, and finally 6 individual room maps for every character. Among the 4 public area maps, there are 2 maps which have several different rooms on the same map, they are considered as different rooms in the planning system as well as in the pathfinding process.

Here is a map of the world in this game prototype.

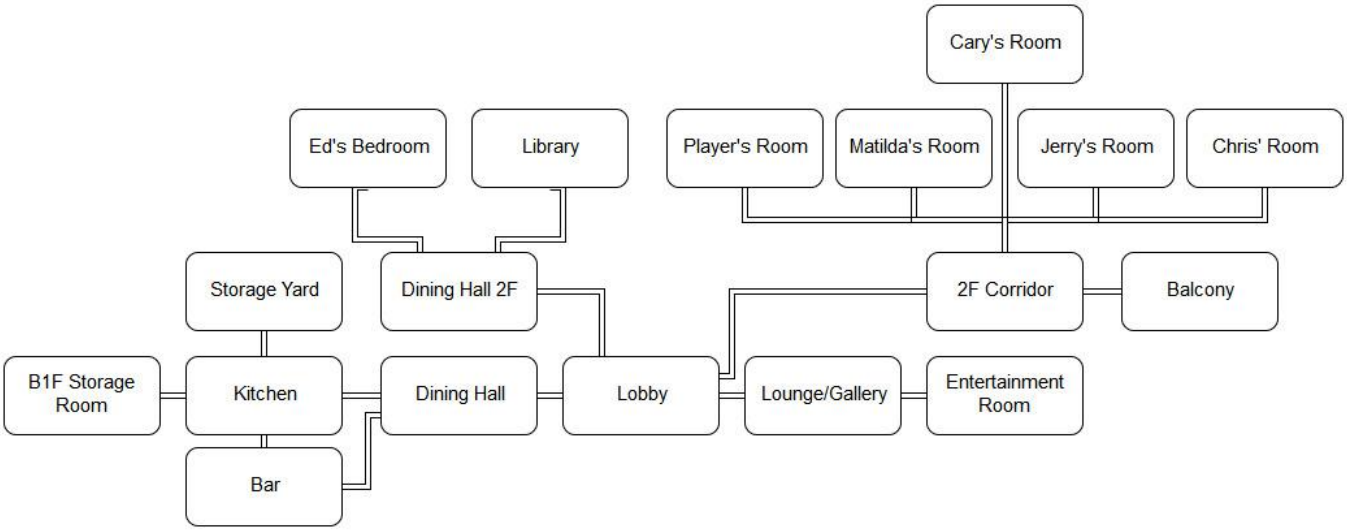

Figure 4-6: Map of the Game World

4.4.1. Scenes with Probability Based Event Scheduler and Affiliation Values

The probability based event scheduler is used alone in the first interactive section in day 1, in which the player can free roaming and observe NPCs doing random actions, and it also used in combination with the event planning system during the murder scene.

The logic behind the first interactive section is as follows. First of all, every NPC is assigned with a total free time value which mimicking the total free time they have during this section. They are free to perform any combination of random actions in the database. These actions in database are all have a value attached, this value indicates how much time is needed to complete the action. Each time an action is performed, the amount of time required to complete that action is deducted from the total free time that NPC has. Each NPC would keep doing this until their total free time reaches 0, so they can each have a sequence of random actions. After all action sequences are generated, the player can finally take control of the hero, when the player traverse through the virtual world, they can observe NPCs performing the last action in their action queue.

For example, let us assume Chris has 60 minutes of free time. He decides to eat something first, and this action would cost him 20 minutes, so this 20 minutes should be deducted from his total free time, so he has 40 minutes of free time left. Then he decides to read some book, which would take him about 30 minutes, then he still has 10 minutes of free time at his disposal. At last, he decides to play darts in the entertainment room,

and this action would cost 30 minutes, and he would have no free time left to spend after that, so this action marks the end of his action sequence. Additionally, when the player roaming in the game world, they would find Chris still playing darts in the entertainment room.

However, there are more things going on behind this process. While the player can only observe their last action directly in the game world, these NPCs would retain and report their past actions, their encounters with other NPCs and their impressions with other NPCs in an indirect way. This information would be useful in further progress and have an impact on player's experience. The way these NPCs inform the player about their past actions is through their dialogue with the player.

To achieve this, we need a new value for each NPC to represent their relationships with each other, I call it the affiliation value. Each character has a list of 3 affiliation values representing their relationship towards the other 3 non-player characters (that includes every NPC except Ed). These values start at 0 from the beginning of the game, a positive value means this NPC is friendly to the specific other NPC, and vice versa.

When each NPC carrying out their action plans, they would found themselves come across each other if they are in the same location at a specific time. If this happens, the system would register this meeting as an off-screen encounter and randomly decide the outcome of the dialog by changing corresponding affiliation values between this pair of characters. The values are individually randomly decided, so it is possible to have an outcome which character A is positive towards character B, but character B is negative towards character A. Once every encounter is solved and every actions in the plan sequence have carried out for each character, these characters would report these encounters to the player so player can have an idea about the relationship between these characters, and these relationships have a huge impact on deciding who is the murderer and who is the first victim.

Here is an example to illustrate how the affiliation value works. Cary has his affiliations towards Chris, Jerry and Matilda, and these affiliations would represent in the form of a list of values. These values starts as [Chris 0, Jerry 0, Matilda 0]. Then Cary bump into Chris in dining room, and they had a somewhat unpleasant conversation, so

the value list become [Chris -1, Jerry 0, Matilda 0]. Then he met Matilda in the lobby, and he really enjoyed the conversation, so the value list would become [Chris -1, Jerry 0, Matilda 1]. Finally, when the player talks to him, he would reply with something like “I have met Chris when I was eating something in the dining room, we had a conversation, but it was unpleasant”, “I have met Matilda in the lobby, I really liked her”.

Figure 4-7: NPC reporting his impression on another NPC.

On the other hand, when the probability based event scheduler is used along side with the event planning story, things gets a bit more complicated. This part would be explained in the next section along with the usage of event planning system.

4.4.2. Murder Event with Event Planning System and A* Pathfinding

The situation in the murder scene is more complex than other situations managed by planning system in this game. The murder event happens off-screen, but the consequences would be observed by the player in later stages of the game. The goal of this event is to generate a complete simulation for a murder, and leave enough trace for the player to reveal the process the whole murder and find out the murderer.

Generating a complete simulation for a murder is not hard to complete with a STRIPS planner, if enough actions are given. Basically, the murderer is expected to complete the following sequence of important events in order to complete the murder:

- The murderer is expected to first find a murder weapon, then find the victim;
- If the victim is alone, then the murderer would kill him on site;
- Else if the victim is not alone in the room, the murderer will bring the victim to another room which is guaranteed to be empty (such as the balcony, the victim's room or the murderer's room) and kill him;
- After that the murderer would move the body of the victim to another room in order to hide the body, and move to another room to dispose the murder weapon.

There are a number of different actions available for the murderer during each step, so the consequence from the murder would be different during each individual playthrough.

Actions in the murderer's plan would only happen in a specific location, and the transition from location to location is not a part of the plan in order to make the planner more efficient. Transitions between locations are handled by the A* pathfinding algorithm, operating on a graph which represents the interconnected locations of the mansion, and the cost from a location to its adjacent locations are all set to 1 for simplification.

At the same time, other NPCs would form their own action plans and carry them out parallel to the murder event, so that the other NPCs can witness murderer and victim's movement if they come across each other. Similar to the section which directed by probability based event scheduler, these NPCs are also given a time value, and the system chooses actions to fill up the time. These actions in plans progress alongside with the murder plans. Every action involved in the murder event, including both murder's movements/actions and "random" actions by other NPCs, are all assigned with a time cost. When all plans are set and the scene goes into action, the system would mimic the progress of the game by advancing one unit of time at a time. During each time unit, the system would evaluate the locations of all NPCs, including the murderer and victim, and record all meetings between NPC characters. Then progress to next time unit, the system would keep doing this until the victim is killed. If their actions are exhausted before the victim is killed, they would stay at their last location until the

murder is done. For simplification, I have made an assumption that the murderer would hide in the murder scene with the victim's body until everyone is gone, and all NPCs excluding the murderer and the victim would back to their own room once the victim is killed, giving the murder the space he needed to hide the body and dispose the murder weapon.

There is a special scenario for the second murder attempt, in which the murder attempt has failed. In that situation, the murderer would stop at the stage which he finds the victim, then instead of killing the victim (since the murder attempt is supposed to be failed), the murderer would retreat, and directly jump to the action which he would dispose the murder weapon. So that the traces of the murderer can still be tracked by the player.

Once all of these actions are done, the information about meetings between NPCs would pass on to individual dialogues during player investigation section, and player can gain such information by talking to these NPCs. The murderer would also report who did he meet during the murder event to the player just as the other NPCs. All murderer's actions would also leave certain trace on his movement routes for the player to collect.

Chapter 5.
Discussion

5.1. General Design Choices

During the designing phase and the implementation phase of this game prototype. I have made a lot of choices based on numerous reasons. In this section, I would like to explain some choices I have made, including why I think these choices are suitable for this project, what are the purposes of certain choices, as well as choices I made in favor of the simplicity.

5.1.1. The Prototype as an Adventure Game

I have explained briefly in the introduction chapter that the reason why I have chosen adventure game as the presentation form for this project. There are a few interesting questions regarding the game prototype: How does this prototype qualify as a video game? Also, how does it qualify as an adventure game?

According to Zimmerman's definition of game, interactive, artificial conflict, consistent rules and quantifiable result seems to be the more important aspects, while "voluntary involvement" often comes naturally (Zimmerman, 2004). This prototype is definitely interactive with all these free explore and investigation sections; it has an artificial conflict between the player and the system, with the opponent personified as the murderer. The player struggles to find out who is the murderer; the rule is simple and similar to classic adventure games, which is to explore the space and collect evidences to solve the mystery; finally, the game has a quantifiable outcome in the form of different endings, and these endings are not equally good. The better outcome is to find out the murderer, and the worse outcome is not being able to find out the murderer, hence the "win" state is easily distinguished from the "lose" one. So I would like to argue that this prototype can definitely classified as a game.

Compared to *Adventure*, the first computer adventure game in history, this game prototype has all the essential elements which *Adventure* had. Such as a set of interconnected locations which are free to traverse and explore by the player, hints of

potential “treasure” (in the case of this game prototype, it is the solution to the crime), and the player interactivity with the environment (Lessard, 2012). If we compare it to newer adventure games, this game prototype also shares some significant trait with them, such as narrative-focused, the basic game rule and the objective of the game: traversing locations, collecting items and clues, and solving problems. That makes this game prototype an adventure game.

On the other hand, if we analyze this game with three characteristics out of Caillois’ six characteristics on “play” (Caillois, 1961), we can have some new findings. The prototype sets in a virtual world in an alternative timeline, so it follows the characteristic which play should occur in a separate time and space; the endings and planning system managed scenes have a certain degree of uncertainty, which follows the requirement of uncertainty in course and outcome; and finally, player engaged in this prototype should be aware of its virtual nature and separate it from the reality. So generally, this prototype also follows Caillois’ idea of “play”.

Adventure games are heavily narrative-driven with a reduced gameplay factor, if I choose to take advantage of this game genre in my project, I can have a better chance to see about how my idea works out. There are more meanings behind the motivation of choosing adventure game as the presentation form for this project, I would like to explain them in this section.

Just as many other digital media forms, immersion is an essential pleasure from video games. The core of immersion in video games is what Murray called an “active creation of belief” (Murray, 1998). Similar to other video game genres, adventure games usually achieve the “active creation of belief” through believable character and story settings, as well as the design of interactive components which make players actively participate in the virtual world.

What makes adventure games stand out is the transition between interactive and non-interactive components. Game genres such as action games which often have pacing problems between actions and the narrative parts mainly because the user expectation, in which they usually want to get into action quickly and treat narrative components as “down times”. This can result in the inequity of balance between action and narrative components, and break immersion between the transition. An infamous

example is *Metal Gear Solid 4.* It is an action-oriented game with extensive non-interactive narrative components, it even has one of the longest cutscene sequence in video game history, at 71 minutes (Rounder, 2014). This horrendous length of cutscenes definitely detaches player from “active creation of disbelief” inside a video game.

On the other hand, adventure games usually do a great job on maintaining the pacing between interactive gameplay sections and dedicated narrative sections, since the pace of its interactive components often keep up with the pace of narrative. The switch between interaction and narrative in adventure games is more frequent, and there are more narrative elements inside the interactive puzzles and quests than that in the action game, so user should recognize the transition of interaction and non-interaction potions less often, hence the immersion breaks less often in an adventure game. This frequent switch between interactive and non-interactive components in adventure games also perfectly resonances Manovich’s idea on video games: the oscillation between interactive and non-interactive components (Manovich, 2000).

Another important pleasure from digital media is agency according to Murray, by which she meant players’ actions should be meaningful and have respective consequences (Murray, 1998). The narrative in adventure games are pushed forward by solving puzzles through player interaction. Some adventure games also feature loop back or branching structure, allowing players’ choices to lead to different consequences. Murray also suggested 3 different pleasures under agency, which are navigation, story and puzzle solving (Murray, 1998). Adventure games can fulfill these 3 pleasures amazingly well, especially the pleasure of story and puzzle solving, because adventure games are built around the power of narrative and the gameplay is fulfilled by puzzles.

5.1.2. The Detective Story

The story of this prototype takes the form of a murder mystery, which is the dominant type of detective story. Taking James’ definition on detective story, a typical detective story includes the following elements: a mysterious death as the central conflict of the story, one or a few suspects with rational motive, a proper opportunity for committing the murder, and, last but not least, a detective as the key and solution to the story (James, 1997). From this definition, we can see the fact that this type of story is highly formulated, many stories falls into the genre follow this formula. But as James

stated, the real exciting thing is that how well this formula can apply to a huge variety of stories and authors, and how imaginative stories can get under this formula (James, 1997). For me, the reason to adopt detective story form lies exactly the same: it is highly formulated, yet it can embrace all kinds of variety.

Under the detective story model, there is a fixed story structure and a background story in the game prototype which includes all the indigents I need. The story is set on an isolated island to meet the requirement of creating an easy opportunity for murders without the intervention and help of outside forces such as police. The motive of the crime is also simplified as the lust for a good fortune for most of the potential suspects with the exception of Matilda (her story is another major branching), so the planning system can make anyone of them to be the suspect or victim under a reasonable excuse. This setting also gives the freedom for the creation of murder scene. Under a similar setting, with a limited number of indigents, a considerable number of different possible murder scenarios can be generated. The player take control of the detective is a common design for such kind of game, since the detective is always the center of a detective story, they are the major motivation for the story to move forward. Although the detective is not as “almighty” as detectives in some stories such as the Holmes series or the Big Sleep, the main character of this game prototype still possesses something very different to the other NPCs in terms of narrative, such as he is beyond the conflicts between the other characters, and he can express his wisdom and capability of action through the player’s interaction.

Generally speaking, a detective story setting is ideal for a partially generative story like the one featured in this project, because of the well accepted yet potentially highly diverse story formula. Just as Miller stated in his narrative definition: the narrative should be a part of a certain pattern or repetition, yet brings in its own new content (Miller, 1995, cited by Zimmerman, 2004).

5.1.3. Player Interactivity

On a macro scale, this game prototype can be seen as a remediation of a number of different traditional media: it is a remediation of traditional novel, it is a remediation of the classic detective story formula, and more importantly, it is a “remediation” of the classic adventure game genre.

I have put a quotation mark around the last “remediation” because when Bolter and Grusin talked about remediation, they generally discussed one medium being remediated within a different form of media (Bolter & Grusin, 2000). I would like to argue that while this game prototype is indeed in the same media form as classic adventure games. The game prototype still takes the form of original material and absorbs it, making the old genre into something which has a very different design philosophy than the source material, but the “soul” of the classic adventure genre is still kept in the form of narrative-focused, puzzle-solving centered gameplay.

Compared to the third “remediation”, the first two is easier to identify: this game prototype undertook the structure and traditions from these old media forms, and transform them into an interactive form. The text descriptions of the traditional story form transformed to the HCI of a digital media, which is a window to a complete virtual world, so that the user can see through this window, as well as a device for the player to interact with this world through the HCI provided by this prototype (Manovich, 2000).

On a more detailed micro level, the interactive design in this game prototype can be analyzed by Zimmerman’s 2 interactive structures. According to this classification, this game prototype is a hybrid of emergent structure and embedded structure, with the planning system managed scenes as highly emergent, while the scripted scenes are mostly embedded. This kind of hybrid structure matches Zimmerman’s vision perfectly, in which he stated that an “explicit interactive media” should have both of these structures, and it is the emergent structure really set interactive digital media apart from other media types (Zimmerman, 2000). The event planning system in this game prototype is the core of this research project, and it is responsible for the most part of the emergent structure in this prototype.

Also, according to Zimmerman’s 4 modes of interaction (Zimmerman, 2000), this game prototype mainly features the third type of explicit interactivity as most of interactive digital media do, and all the functional interactivity comes free with any media; for the rest 2 modes, I would like to argue that the generative design this game prototype itself is a deconstruction and reconstruction of traditional narrative design.

The simple HCI design is a refinement and simplification over classic adventure games, while maintaining the core gameplay of these old adventure games. Although

the HCI is a free gift from the development tool, I would choose this type of direct control over the clunky point-and-click style control event if the direct control option is not included in the development tool, since the point-and-click control is not as intuitive as the direct control, and can be very frustrating to navigate characters or find the correct object to interact at times.

According to Ryan's criteria on user participation modes in digital media, typical adventure games belong to the internal / ontological mode. Because in these games, players usually assume control of a hero character in the virtual world, and they would change the state of the virtual world by solving puzzles and complete quests (Ryan, 2004). Taking direct control of the hero in the game can improve immersion, since the player can project themselves directly into the game through their avatar in the game. On the other hand, most video games involve the change in state in order to progress further into the game. In the case of a narrative-focused game such as an adventure game, it is even more obvious and important, since, according to Miller's definition, the change of state is an essential part of the narrative (Miller, 1995, cited by Zimmerman, 2004). In adventure games, the changes to state of virtual world is done through solving puzzles and completing quests and sub goals, so that this interactive progress acts as an essential device to push the narrative forward.

5.1.4. Space Design

The virtual world in this game prototype has an important role. Murray valued the freedom of navigation in games as one of the basic pleasure of agency in her arguments, she also mentioned that the navigation in virtual space serves an important role in narrative in these video games, especially in adventure games (Murray, 1998). Manovich also listed the "spatial journey" in video games as the "core of the narrative" (Manovich, 2000). So I have decided to construct a virtual world for this game prototype that the player can navigate freely in, at the same time, it is also capable for all the events in the narrative.

The most important guideline I have kept in mind in designing the space in this game prototype is to make the world looks believable. So I have paid close attention to the layout of maps as well as the layout of individual rooms. The whole game world is a mansion for a rich person to live in, so the place should be reasonably large, but not too

huge. The collection of rooms should cover all the daily needs and luxury needs of daily life and laid in a reasonable layout, by putting rooms with similar functions together. For example, rooms like the lounge, the gallery, the entertainment room, and the guest rooms are all located in the east wing of the mansion, since they are all meant for guests, so they can have their activity space together, and people who live in this mansion for a life time does not need to access that wing on a regular basis. In contrast, the dining room, the kitchen, the library and the master bedroom are all in the west wing, they have included all needs for daily life, and also provide some hint in the narrative. For example, the library is located near the master bedroom, which means the master of this mansion might love to read. The layout for each room is also designed with its functionality in mind, all bedrooms would have a window, a desk, a bed, some storage space as well as a restroom. Some rooms even have additional furniture to suit the special needs for characters, such as the dressing table in Matilda's room and extra couches in the master bedroom.

The design of the space is also a resonance to Murray of Manovich's idea of "spatial journey". These details in world design should be able to make the world looks more believable, as well as deliver some background settings and stories. The design of the world also has the awareness of the narrative of the game in mind, that way the world is capable of delivering the story built inside it.

The world is also designed with the easiness of access and traverse of players in mind. Each individual map is not terribly huge, while still have a decent amount of space for players to move around. The structure of each individual room is not too complex, so it would not confuse the player, the number of rooms and their layout is also designed with the complexity in mind in order to make sure the player would not have too much trouble in remembering locations.

5.1.5. Game Rules

As Zimmerman and Caillios stated in their own definitions, rules are the most important part in games. Games can exist and available for meaningful play only if rules exist to limit actions of participants, and participants are voluntarily to obey rules (Juul, 2005).

Salen and Zimmerman had a set of definitions to identify rules in games (Salen & Zimmerman, 2004), which can be applied to this game prototype as well. Stripping all the detailed narrative contents and graphics from the game prototype, gameplay rules in this game prototype can be boiled down to traversing in a virtual space and finding required objects, ultimately the player need to make the right choice based on given information in order to get the better outcomes. This game prototype played safe on designing these rules so they are not that different from the average adventure games, the gameplay experience should be similar to what the player might expect from a classic adventure game. On the other hand, the internal system rules are more complex since it involves a planning system, however, rules in these systems still follows what Salen and Zimmerman have stated. They regulate how NPCs act, all non-scripted NPCs actions are managed by the same set of rules, they are implemented as an explicit algorithm, they are not getting changed during the entire playthrough, neither they would change across all play sessions. So, according to Salen and Zimmerman's definition, both the gameplay rules and system rules are solid and reasonable in this game prototype. Also, according to Salen and Zimmerman's classification of rules (Salen & Zimmerman, 2004), gameplay rules and abstract system rules are belonging to the operational rules, and the details behind system rules belong to the constitutive rules.

Players in this game prototype are bounded by rules just like the other games, and the rules are meaningful to ensure a fun experience. Juul views rules as finite state machines with a tree of finite number of possibilities, as well as different outcomes, with some of them better than the others, the process of working towards the better outcomes are known as gameplay (Juul, 2005). This prototype takes this definition and tried to enhance gameplay aspect with a strong narrative involvement, it involves a component which embraces a typical computational approach in the narrative design in order to expand the fun with its rules. I would like to argue that the inclusion of event planning system significantly increased the number of finite possibilities in this game prototype, although the number of ultimate outcomes has not changed, it is interesting that the planning system managed scenes have more variable consequences to provide a more interesting experience.

Juul also quoted Sid Meier's statements on designing choices that indicated no best option, no equally good options and informed choice (Juul, 2005). In my opinion, the informed choice is the most important aspect. While no best option and no equally

good options sounds very interesting, but at the same time they are extremely hard to design at times due to the difficulty in balancing the consequences, the consequences might be offering no best options or offering different pros and cons in developer's vision, but the player might not have the same vision. On the other hand, making players well informed before choices would usually provide a better experience to the player compared to make them just blind guess the consequences, so I have tried my best to propagate the explicit limitations and consequences in this game prototype to players using dialogues or monologues from the hero or other NPCs, such as the steps limitation in the investigation section and the worse consequence players would get if they cannot find enough evidence in time. Also, the purpose of the design that NPCs would report their encounters with other NPCs and their impression to other NPCs is to give player more information to judge who is the murderer, while they are not counted towards direct requirements of evidence count. These designs are all attempts to keep them as informed as possible so they can think carefully before making choices during their playthrough.

The designing of puzzle is another interesting topic. Juul indicated that the aesthetics of puzzle lies within its simpleness of solution and its complexity of the thinking progress, by quoting Danesi (Juul, 2005). In this game prototype, the biggest puzzle is generated by the planning system, so even the developer does not know the answer, yet the answer itself is as simple as a name. The process the player needs to go through in order to find the truth is similar to the fetch quests commonly seen in adventure games, but I would like to argue the uncertainty nature adds more complexity to the puzzle, ultimately it would provide more interesting experience over multiple playthroughs.

5.2. The Narrative

The central focus of this research project is based on interactive narrative and generative narrative. The discussion of narrative for this game prototype is an essential part of the whole discussion. In this section, I would like to talk about the overall design of narrative, the design of narrative arc, pacing control, micro-narratives, branchings and endings, as well as how the planning system managed narrative work out.

5.2.1. Narrative Design

As a narrative-focused game, the narrative in this prototype was designed while keeping the definitions of narrative in mind. Bordwell's definition stated that the narrative has to is a series of events in cause and effect relationship and occur in its own time and space (Bordwell & Thompson, 2008). So the narrative in this game prototype has been written in a way that to make sure the cause-effect chain between events are clear to the player by stating the cause clearly to the player in the opening moments of the game, then showing player how things played out later in the game, and make sure outcomes stay logical and reasonable to the cause of these incidents. The time and space settings of this game are also stated clearly to the player in its opening scene.

Several narrative design traits are important within games, two of these are particularly useful for understanding game narrative. First, a narrative is a series of events (Chatman, 1978), and second, these events need to make sense (Ryan, 2006; Abbott, 2002). The traits that are implied in traditional definitions of narrative form the foundation of this project, since the generative design is based on traits included within these definitions. In most games, making a reasonable sequence of event is the responsibility of the developer, if we want more variations in the narrative, additional human effort is required. Replacing the effort partially with an algorithm would effectively result in more diverse narrative, hence I came up with this project.

This project also instantiates elements of Ryan's definition of narrative in digital media as well as Manovich's argument on database narrative. Ryan stated the nature of narrative in a computational environment by describing it as a world, a set of characters, a set of events and a series of changes to the world come by as the consequences of previous events (Ryan, 2004). The definition is very close to the structure of planning system, since the planning system also has similar components: a world knowledge base for agents to operate in, a set of agents to carry out actions, and a set of actions for agents to use which can change the world. We can see from the similarities that the planning system can potentially be used to generate narrative, and the outcome proves this assumption holds to a degree. Manovich also held a similar point of view, he thought everything in digital media can be reduced to database and algorithm, including narrative. His model of narrative somewhat resonances with Ryan's idea by also taking a computational approach to the same subject (Manovich, 2000). These two arguments

have laid the research ground for this project, and made the development of this game prototype easier. Also, the generative component is carefully designed to ensure the sequence of events is in a logically correct cause-effect relationship, and they can make sense to the user. Manovich thought a random combination of data cannot be considered as narrative (Manovich, 2000), so the sequence of event can be a reasonable narrative only if the sequence can make sense to the user.

Miller's definition on narrative is also an interesting argument to look at. His definition also took a form similar to a computational approach, in which he divided narrative into different states, stated its nature as the change between states (Zimmerman, 2004; Miller, 1995). Zimmerman quoted this definition in his paper on the subject of game narrative, because this definition gives "meaningful play" a great theoretical ground: the interactive sections in game can be the force which changes the virtual world between states, so that the user interactions can be meaningful to the narrative. This game prototype also follows this philosophy and lets the player take control of the hero during the most crucial part of the narrative: investigation and revealing the truth. I believe this is the most powerful component, as well as the most enjoyable moment in the narrative experience of a video game.

5.2.2. Narrative Arc

The narrative arc of this game prototype is designed with the classic narrative arc in mind, it has a similar overall narrative structure following the classic structure of setup, complication, development and resolution / denouement (Bizzocchi, 2006). The introduction scene, the first interactive scene and the dinner scene can be seen as the setup phase, these scenes introduce characters, location and the basic conflict to the player, and hint for a further complication in the story. The first murder and first investigation scene can be seen as both complication and development phase, these scenes have made the conflict between characters more obvious and more serious, and complicate the situation into a much more serious one due to the murder. The second murder attempt and the second investigation can also be seen as a part of the development phase, since it further deepens the conflict between characters and pushes the suspense further. Finally, the confrontation of murderer and ending can be seen as the climax / resolution / denouement, since in these scenes, the puzzle of murders would

be solved, murderer would be revealed, their motivation would also be explained, and a little bit of the aftermath would also be touched in the end.

The narrative arc is adopted to ensure the narrative of this game prototype follows the tradition of narrative, so it can deliver a solid story by following a reasonable structure which has already been proved by countless other stories. On the other hand, as Bizzocchi pointed out, due to the interactive nature of video games, the pacing control in narrative in video games cannot be as precise as a traditional media artifact such as a film (Bizzocchi, 2006). It is inevitable that the developer would lose their tight control in the narrative pacing in interactive sections since they cannot predict player's action, but implementing a proper narrative arc can still give a decent amount of positive impact to the in-game narrative, and many video games involving a narrative element embraced this approach in order to deliver a better narrative experience. Especially in the adventure games, where they can often have much tighter controls over the narrative arc and pacing in the narrative than other game genres, at the cost of player's interactivity. The most famous examples in recent years can be interactive drama style games directed by David Cage, such as *Heavy Rain* and *Beyond: Two Souls*. While this game prototype did not take that much freedom from the player in interactive sections, it does take away control from the player during scripted dialog scenes in order to develop the story.

The planning system managed scenes also plays a huge role in the narrative arc, the investigation scenes are import to complication and development phases, since they act as the mystery solving process in the story. The pacing in this section is meant to be slower so it is okay to give player more freedom in interactivity, however I think it is not a good idea to let players take too long in this section, so I have set up a limit in order to move the narrative to next stage if players have taken extremely long time in this section. This setting acts as an attempt to gain a loose control of narrative pacing in interactive sections in this game prototype.

5.2.3. Time

The game progression that players would be experiencing is designed to match the narrative arc. However, the flow of time during a play session in the real world is

designed in a way which does not match the speed of time flow in the narrative world just as many video games did. The design of time is an interesting topic to discuss.

The flow of time in this game is somewhat complex. On the surface, it can be perceived as the same speed to the real world time, which is an approach identified by Juul as commonly used in video games (Juul, 2005); but under the surface, the flow of time during interactive sections in this game prototype works in a different way as the time in real world.

As a matter of fact, the constant flow of time is stopped in the first interactive section which the player gets to explore the world and interact with NPCs. No matter how long the player decides to wander in the game world, the game world would not change as long as the player does not go back to their own room in order to progress to the next scene. Situations in the investigation sections are somewhat different, since every step the player takes count towards the progression of time, but the time under this situation would not move forward as long as the player stands still.

The former situation is what I would like to call a trigger based event progression, which is extremely common in video games that do not feature any kind of in-game clock. In these games, events are activated based on various triggers, the progression of time is virtually stopped as long as conditions for these triggers are not met. This approach can be seen as a major difference between game time and real time, and it is there to give the freedom of pacing control to the player, so they can progress in the game with their own pace.

The step count limit in investigation sections can be seen as a kind of time flow, but it is not exactly like the flow of time in real world, because the time is not moving forward as long as the player choose to stand still. This approach can still give the player enough time to think about the next moves, and control their own pace in playing, while implement some sort of time limit to restrict their actions, in order to bring in some degree of challenge.

Although Juul did not mention this aspect of game time in his discussion, he is not wrong on identifying the flow of time in different games. I would still recognize the time in this game prototype as a near 1:1 projection to the time in real world, since the control input is directly reflected by the hero in game, and the movement speed is

neither obviously faster nor slower than the movement speed of people in real life, these aspects are considered by Juul as the traits of a “real-time game with worlds” (Juul, 2005). On the other hand, I would like to argue that Juul’s discussion on the break of game time can also apply to this situation. In his discussion, he talked about how GTA 3 can have an accelerated clock while the speed of actions is 1:1 proportional to the real world (Juul, 2005). In the situation I described above, it is similar, the speed of actions is also nearly 1:1 proportional to the real world, while the flow of background time does not match the speed of action. In fact, it is not moving forward at all as long as the player does not trigger the next event.

Juul also talked about how games can set its time at a fictional time, as well as how games can have discontinuity between different levels (Juul, 2005). These are all reflected in my designs for this game prototype: the game prototype is set at a fictional time period, and there are breaches of continuity between scenes, in which the flow of time is omitted, so that the game prototype can cover a time span of three days in a relatively short play time span.

5.2.4. Micro-Narrative and Environmental Storytelling

In Bizzocchi’s idea of game narrative analysis framework, he suggested that micro-narrative is an important component for video games to provide a locally engaging narrative experience since the developer can have more control over the narrative pacing in this specific piece of story (Bizzocchi, 2006). This game prototype also takes advantage of micro-narratives to enhance the narrative experience.

Some micro-narratives in this game prototype are in interactive sections which require players to explore the virtual world carefully in order to gain access to them. They come in mainly two forms: one of them is in the form of scripted events, they are random events that managed by probability based event scheduler and would be placed in the game world if they are the last event in the randomly generated event sequence. For example, the NPC would decide to eat something in the dining hall, or take a drink in the bar. The other form is the NPC’s report about who did he/she met during his random event sequence, this is simply a dialogue stating these facts, such as “I have met Matilda at the gallery, and I don’t like our conversation”. These micro-narratives are all

used in a more or less generative manner, and they have more meanings in the course of gameplay as well as overall narrative in addition to their own local narrative.

These micro-narratives in interactive sections only feature limited narrative contents as they only last for a few lines. However, in future works, these micro-narrative contents can be extended to a more complete story. There are also some stories that function as micro-narratives inside the scripted dialogue scenes, especially towards the end of the game. In these scenes, there are dialogues not directly link to the main murder incident, but they also offer some additional information about specific character's background.

Jenkins talked about environmental narrative and embedded narratives which suggested to take advantage of the interactive virtual world of the game and provide additional narrative information to the player through the design of the virtual world itself as well as "redundant" pieces of information scattered through the world (Jenkins, 2004). This game prototype follows Jenkins' idea closely, when designing the spaces of this game prototype, I have carefully designed the aesthetic elements and layout of the world, so it can really deliver a representation of the world that matches what was specified in the background settings of the story, as well as making sure to have all the locations needed in the story. Evidences and dialogues managed by the planning systems are also efforts towards embedded narrative, as they are there to provide narrative information to players in order to help them solve the murder mystery, and dialogues about meetings between NPCs contain redundant information about NPC locations.

5.2.5. Branching and Endings

Branching is an important way to extend interactivity as well as the experience of narrative in digital media. Crawford discussed this matter extensively. According to his identifications on different narrative structures, the narrative structure in this game prototype is somewhat a mixed one. Specifically, this game prototype uses a branching story tree with some foldback features (Crawford, 2003). There are two main branching points in the story of this game prototype, one of which is based on who is the murderer, and this branching is controlled by the planning system, while the other one depending on player's performance. There are also minor branching points which may lead to the bad endings if the player failed to collect enough evidence during second investigation

section or confronted the wrong character as murderer. The foldback features are mainly controlled by the planning system, and exist in the form of different combinations of murderer / victims, the detailed process of murder as well as random events performed by NPCs. Also, the traditional video game narrative structure which features obstacles that players have to overcome between narratives exists in this game prototype as investigation sections. The only difference is that the player does not require to complete this section in order to advance the narrative, but they would get a worse situation if they did not successfully complete one.

Crawford stated that a storytree with a huge number of branches is very impractical because the number of states would quickly get out of control (Crawford, 2004). So an important topic of this project is to find a way to fill up these different states not by human hand, but by combinations of different elements generated by an algorithm. The planning system is my answer to this question, it can assemble a rather large number of events by creating different combinations of narrative elements from a limited database. When facing this large number of possibilities, the player has to make different interactive actions in order to react to the situation they encounter, hence the interactive possibilities increases if we take into account the different playthroughs, even if the arbitrary choices of different combinations are specified by the system, not by the player. So we end up with a situation which the player has enough interactive varieties, while the number of possibilities in scripted scenes is low. I would like to argue it is a reasonable approach to achieve variety in narrative when the resource for creating new contents is limited.

Crawford also set up criteria which measures the excellence of interactive narrative by calculating the ratio of conceivable states and accessible states, as well as gave some advice on achieving the excellence in his mind. He argues that a good interactive design should aim for a match between conceivable states and accessible states. The player should not conceive of things that they cannot do (Crawford, 2004). This game prototype aims to decrease the number of conceivable states by limiting user interaction to solving the mystery itself only, and leave other event construction and branching decisions to the system itself. Branches which affected by player performance is not impacted by player's decision directly, but through a somewhat indirect way by go through the evaluation of player's performance first. Crawford also suggested to use a computed branching structure rather than a hard-wired branching structure to increase

interactivity (Crawford, 2004). This is exactly what this project does, as computed branching is implemented as the form of planning systems.

5.3. Planning Systems Managed Scenes

The two planning systems are the heart of this project, they are my solution for the research question raised in this project. In this section, I would like to take some time and discuss the design decisions in these systems in details.

5.3.1. The Randomness and Certainty in Probability Based System Managed Scenes

Randomness is a very important topic to explore in video games. Salen and Zimmerman even regarded it as the “central feature of every game” and believe the meaningful play can exist only if randomness exists (Salen & Zimmerman, 2004). Most video games include randomness as a part of the system in some way, as it can bring unforeseen situations to the player, and make the situation more unpredictable yet giving more power to user choices while still be fun.

This game prototype also has an implementation of randomness, mainly in the form of the probability based event scheduler as well as the event planning system, in order to give the narrative more variety and flavor. According to Epstein’s classification quoted by Salen and Zimmerman (Salen & Zimmerman, 2004), I think the randomness brought by this system falls under the “uncertainty” category since the outcomes of these random events are not going to benefit the player in a simple direct way. However, the combination of these random events may give the player some advantages in predicting the upcoming murder or identifying the murderer, so there is some “risk” element in my design.

I made some effort to improve the players’ experience with these random based elements. Salen and Zimmerman pointed out that the computer generated random numbers are pseudo ones. So I have adopted a technique which is widely used when involving random number generators, which is to use the system time as the “seed” of the random number generator. Since the time is always changing, so there is a bigger

chance to generate a different number each time, reducing the possibility which the random number generator keeps pumping out the same numbers.

Salen and Zimmerman stated that a good balance between certainty, risk and uncertainty is the key to design a successful meaningful play (Salen & Zimmerman, 2004). I am also aware that users often have different expectations and feelings towards randomness. They expect more variety in results across different “dice rolls”, rather than the result distribution of a near-true randomness. I therefore revised the conditions of each action NPCs can take, and make sure they would never do certain actions twice in a single scheduling or planning session, as well as abnormal actions, such as moving in circles or doing things unrelated to their current goals.

On the other hand, as an adventure game, players also expect a level of certainty and consistency from such a game. For example, the outcome of solving each puzzle should be the same, and the way of solving the puzzle should follow fixed patterns or routines to avoid wild guessing. So the method for investigation remain the same as a certainty across all play sessions. The relationship between user performance and the consequences are also mapped specifically in the scripted scenes, so the player can expect a certain outcome based on their performance.

The overall design of certainty, uncertainty, and risk aims to meet Salen and Zimmerman’s requirement of balance across these related variables (Salen & Zimmerman, 2004).

5.3.2. The Planning System and the Murder Scene

The murder scene is the core of this game prototype, the system behind the murder scene is the most important component in this project - the STRIPS planner. During the murder scene, it works together with the probability based event scheduler in an effort to simulate the progress of a murder, and generate proper evidences for the player to solve the murder, and reproduce this progress along with their solution.

The process of generating the murder scene involves both STRIPS based event planning system and probability based system. The STRIPS planning handles the murderer’s actions and the probability based system handles the rest NPCs’ actions, their actions and meetings would be recorded and report to the player in the form of

items and dialogues. The idea behind this design is to simulate the whole murder scene with all the actors and happenings, then leave evidence along the course of simulation, rather than the approach that generates the after scene directly, and then retrospectively generates the process of the murder. I believe generating the scene in chronological order is a better way to simulate events in the story, since it is similar to the process of a murder in real life hence more convincing in its design.

One of the most important inspirations for this project is the paper on the 2005 video game *F.E.A.R.* from GDC 2006 (Orkin, 2006). In that paper, I was first introduced to the possibilities of STRIPS planning in games. I have witnessed how amazingly capable the A.I. enemies were in that game. Also, when I read the paper, I was impressed by the fact that the technology behind this game's A.I. system is a such simple and elegant one. It is even simpler than the traditional decision trees because it has fewer possible states. I decided to create a game prototype with the implementation of a planning system, but with a different focus: while most of the games focus on enhancing the gameplay, since I was interested in exploring on the interactive narrative side.

To start with, I have first explored a few examples from pre-computer age, and to my surprise, there was a good deal of literature and research dedicated to generative narrative through combinations of elements from a large database. Some typical examples are the *Exquisite Corpse* game from the avant garde tradition ("About Exquisite Corpse", n.d.), the *A hundred thousand billion poems* by Queneau (Queneau, 1961), as well as the early table-top role playing games. These three examples represent three kinds of generative narrative in the pre-computer age: 1) collective effort that different people responsible for different parts in the structure, 2) generate new contents by permutations of contents from a predetermined database, and 3) improvise on site according to certain rules.

In the field of computing, the combination of first and second is the easiest way to go because human creation is still responsible for a large amount of content creation as well as a very simple algorithmic design. The downside is that the output often does not feel very original and fresh, as well as seemingly too similar and too random. So the improvisation way appears to be a better choice from the computational perspective, but it requires very complex design and can be less effective in more realistic scenarios. The

planning approach in this game prototype lies somewhere between the two. Its algorithmic design is more complex than simple permutations, while its database has complete sentences and paragraphs pre-written by human rather than those generative narrative attempts completely rely on linguistics or more atomic elements to generate contents word by word.

I have also explored a number of different approaches covering computational narrative, generative narrative and game narrative research domain. I have found some interesting approaches, some of them have given me some precious inspirations. These projects have used a variety of techniques and algorithms including data mining, searching, advanced planning and genetic algorithms, and attempted to improve the process by focusing on different subjects like user behavior, narrative structure and conflict. These interesting approaches have given me some novel and useful ideas and references. However, I still decided to implement my system based on the simple STRIPS planning approach to keep the size of this project at a manageable state, while exploring some additions from the ideas that inspired me when I explored other projects. I think the capacity of this planning based project is suitable for a exploration of my ideas, the more complex approaches might be explored in the future.

5.4. Scripted Scenes

5.4.1. The Existence of Scripted Scenes

The narrative is not generative throughout the whole game prototype. Only several important scenes are managed by planning systems, and the rest of the scenes are scripted. These scripted scenes have important functions in the narrative structure of this game prototype. They are there to handle the more complex part of a story, namely characters, setup and the resolution of a story.

The main reason why I decided to use scripted scenes in this game prototype, rather than using a generative approach throughout the game is that, I believe generative approaches alone cannot provide a narrative that is complex enough to be interesting in a project of this scale. The results from several projects I have examined did not show anything satisfactory enough to form a complete story. Here are some examples. Young’s Mimesis system focused on producing single scene and short stories

(Young, 2007). Another project by Ware, et al. about narrative conflict based model (Ware et al., 2014) was on building the structure of a story rather than the detailed content itself. And Kumar et al.'s project was about the formalization of a storytelling algorithm and evaluation of the algorithm (Kumar et al., 2008), it is more of a theoretical research rather than a research on an application level, and the narrative it creates are mainly stubs and frameworks without a proper degree of details.

On the other hand, there are also some projects using an approach similar to this project, which is a combination of computational approach and scripted events. One example is the project by Ramirez and Bulitko, in which they effectively used a series of human written stories and a branching / loopback narrative structure, and build an algorithm to analyze player's preference and control the branching of the narrative to match player's preference (Ramirez and Bulitko, 2014). Thus I believe scripted scenes are still valuable in a game prototype like this, even when the main focus is on the generative component.

Using scripted scenes can also save a lot of development resources and time while still provide a complete game narrative experience. These written dialogues can bring down the complexity of the game significantly. So this approach is capable of bringing down the resources and time needed during development. These written narratives also decrease the probability of game glitches and errors. I think the design choice to include a number of scripted scene not only made the development process more efficient, but also can be a potential strategy for future projects using more realistic scenarios.

Finally, there is another reason to implement scripted scenes in this game prototype, and the reason is very simple: to effectively push the narrative forward. The narrative in these scripted scenes in this game prototype are not different compared to similar scenes in other classic adventure games, and their functions are also comparable to dialogue scenes, cut-scenes or set-pieces in other video game genres. The individual scripted scene is written in a manner similar to those in traditional media such as novels and films. These scripted scenes use dialogue to drive the storytelling, augmented by occasional animations performed with game engine. According to Bizzocchi, the precise pacing control that is common in films is not available generally in video games, since the interactive nature would often break the intended pacing

(Bizzocchi, 2006). For games to gain some pacing control and take advantage of narrative structure, it is common to take away some freedom from the player in order to show them something, such as using cut-scenes or set-pieces. This game prototype has used a similar method by inserting non-interactive scripted dialogue scenes during important narrative moments, in order to gain some control in narrative pacing control.

5.4.2. The Transition between Scripted Scenes and Planning System Managed Scenes

The differences between scripted scenes and planning system managed scenes are very obvious.

Scripted scenes can only have accounted for a limited amount of possibilities and branches as far as specific writing and design by human is required, the amount of possibilities and branches is strictly limited by the fact that how many scripted scenes are written by human authors and implemented by human developers. As Crawford describes, the effort to increase the number of scripted scenes can reach a completely unreasonable state very quickly, due to the exponential growth in nodes required for more branching (Crawford, 2003).

On the other hand, the number of states resulted by generative scenes such as scenes managed by planning system in this game prototype can be rather large, since the number of possible combinations and permutations of raw data from database can be fairly large, even if the number of raw data entries is very limited. Take the example of the combination of murderer and victims in this game prototype. If we exclude the special case which Matilda can kill Ed, we have 4 NPCs that can potentially kill each other, which is a fairly small number. But even if we only consider the first murder involving one murderer and one victim, we have a permutation problem $P(4, 2)$, which results in 12 different murderer-victim situations. If we take the second murder attempt into account, the possibilities would be even larger, not to mention we have not considered the variety in murder locations and murder weapons yet. So how can we connect a planning system managed scene with so many possible outcomes, to a scripted scene with obviously restricted branching ability?

The answer is simple, yet might be a bit disappointing. We can solve this situation by carefully writing the scripted scene so that it would fit in any outcomes from

the planning system managed scene, with a few additional lines to represent the specific outcome from the planning system managed scene, so that the player would proceed without noticing this solution in their first playthrough.

It is a solution that is similar to something widely adopted in a lot of situations in real life which involving choices. It is often called “the illusion of choice”, which means you are basically choose between one thing or nothing at all (Barrett, 2009). In terms of video games, it often means the player has a number of choices to pick, but they all lead to the same consequence with few variants to briefly reflect the player’s choice, if there is any. *Mass Effect* series is a great example for doing this in its game design, every decision the player makes seems to matter a lot in the narrative, but actually every choice lead to the same consequence. There are also some games put in more effort to convince players to think that each of their choices would count, such as *the Walking Dead* video game series would display a message like “character x will remember your choice” after a seemingly important choice is made. These techniques are effective to make players engaged in the narrative during their first gameplay, but this immersion would fall apart when players revisit the game for a second time.

So, why am I still using this technique in this game prototype? Because there is a difference between my implementation and the way games like *Mass Effect* have taken. According to Sid Meier’s definition on interesting choices (Juul, 2005), the way *Mass Effect* handles player choices is not a good one, since the game basically provides only one consequence across all choices, so the choices are equally good, and the game deliberately hides the fact that their choices actually does not matter, hence it is not an informed choice.

In this game prototype however, the numerous outcomes from the planning system managed scene are not the result of user interaction and user choices, but decisions made by the automated planning system, so that players’ interaction and choices are not affected by this trick. There is another fact that the consequence from the planning system managed scene does not come by just one pattern. There are a few branches in the subsequent scripted scenes, in which the branch is decided by the specific combinations of murderer-victim combination, so there is actually a few number of choices for the planning system, hence it is not a completely “illusion of choice”. On the other hand, I took caution when designing the narrative around player interactions,

so that players' interaction would actually have an impact on the outcome of the game. In their repeated gameplay sessions, they are likely to find out that their actions are actually meaningful.

Unlike the transition from a planning system managed scene to a scripted scene, the transition from a scripted scene to a planning system managed scene is pretty straightforward. Since scripted scenes would not actually change any status of the virtual world which the planning system is using, the planning system can load right back in and pick up the situation left off by the previous planning system managed scene right away.

5.5. Replayability

Just as I stated in the introduction chapter, replayability is the core of my research question in this project. Now it is the time to discuss this very subject to see if the approaches implemented in this game prototype can improve replayability.

5.5.1. Computational Approaches and Replayability

This game prototype uses a computational approach in attempt to enhance the replayability of the game. The planner based approach is deployed to provide a different experience across different playthroughs, so players would have something new to look at when they revisit the game. There are also quite a few games that already do this, the element which made this game prototype different is it focused on providing different narrative experiences across different playthroughs.

The implementation of computational approach in this game prototype is similar to games with a procedural generation feature. In most of these games, the geographical layout of the level, the location of contraptions and traps, the location of enemies and the locations of loots are not specifically designed by the developer of the game. Instead, they use algorithms to generate these designs on the fly, so players would encounter seemingly infinite number of levels across different playthroughs. The earliest example is *Rogue*. It is a dungeon crawler role-playing game, in which the player controls an adventurer to explore procedurally generated dungeons. Dungeons are generally composed by rooms and corridors connects them, and the layout of rooms and corridors for the next map would be generated on the fly right before the player enters

the level. This design guarantees the player would meet constantly engaged with new contents, each playthrough would be fresh (Wichman, 1997).

Although dungeon crawling games featuring procedure generation gives a hint to solve the replayability problem, this project needs something more complex than the simplistic solution relies on random number generator in these dungeon crawling games in order to form a reasonable narrative. There are some interesting video games feature more modern approaches in procedural generation and automated level design, *Left 4 Dead* is the one worth mentioning. *Left 4 Dead* is a cooperative first person shooter in which 4 players should work together to find a way to escape zombie-flooded environments. Developers of this game have realized the importance of replayabilty, as well as the fact that unpredictable experience during player interactions can encourage players return for more, thus enhance replayability. So they have implemented the A.I. Director in this game. The basic idea is to have a system work as a director to monitor the game world from above, and respond to current situation accordingly by spawning / de-spawning enemies and resources, in order to keep players engaged with the game (Booth, 2009). The idea that to have a system watch the game world from above and manage the world according to its situation. This game prototype takes lesson from this idea, and designed the planning system to also work as a “director” of the scene, since its work is to monitor the current state of game world and generate new narrative from this state using a computational approach.

The approaches *Rogue* and *Left 4 Dead* implemented to improve replayability has been proved by their popularity among players. Rogue was really popular among computer players in the early 1980s (Wichman, 1997). It was the first example of level design using procedure generation, and proved to be a well-received feature by later followers: this game has spawned a whole new genre of dungeon-crawler games called “roguelike”. *Left 4 Dead* and its sequel *Left 4 Dead 2* has more than 18 million players combined, and *Left 4 Dead 2* still has more than 9 million active players that have played the game recently as of the end of June, 2017 (SteamSpy, 2017). This game prototype follows ideas of enhancing replayability in these games, and apply the computational approach in its narrative component, I have faith that its replayability should be an improvement over classic adventure games.

5.5.2. Game Narrative and Replayability

There are also some narrative-driven games which have experimented with the idea of enhanced replayability from a narrative perspective. Some of these approaches are commonly seen across many games, and there are also some specific approaches that can hardly be seen among games, but they are interesting ideas nevertheless. I have spent some time to examine a few such examples, and applied a few ideas to this game prototype as well.

The different narrative structures mentioned by Crawford (2003) are all very common in video games, especially in “games of progression” as categorized by Juul (2005). While many games feature linear narrative structure as well as one ending, there are some exceptions. Japanese adventure games and visual novels often feature multiple branches of totally different storylines, role-playing games and modern adventure games often feature foldback structure with a variety of different outcomes for a single localized narrative event, and some of these games features multiple endings.

The opening scene in David Cage’s *Indigo Prophecy* has demonstrated both the advantages and disadvantages in the branching / foldback structure: during this scene, the hero would find himself just killed someone inside the toilet of a diner, the player would need to control him to find a way to escape the diner, with a variety of interactive options to shape his escape very differently, such as clean the blood around the murder, back to the diner and finish his meal, rush out of the diner or walk slowly out of the diner. After the hero is escaped, the player would take control of a police detective that arrives at the crime scene later on. The player can investigate the crime scene as the detective and spot the consequences caused by the hero earlier, player’s action in the previous scene can cause different consequences in this scene, as well as the detective’s conclusion from this scene. For example, if the player cleaned the blood around the murder scene earlier, finished his meal and paid for the meal, the detective would conclude he is a professional killer, not an amateur. I would like to argue that this is a powerful scene to demonstrate player interactivity and its impact on the narrative, since every action from the player has a different consequence. The narrative in this scene has a lot of conceivable states which are actually achievable, so it would pass Crawford’s test on interactivity with a satisfactory result (Crawford, 2003). But there is a catch: this kind of scene is extremely hard to write and develop, since it requires a large

amount of work. It also reflected in *Indigo Prophecy* itself: the opening scene is the most interesting scene in the whole game, no scene afterwards has such interactivity, not even close.

There are also games that partially rely on random factors to decide their branching and endings. Westwood Studio's *Blade Runner* is an adventure game which features random factors as an important element to affect narrative branches. In this game, the goal for the hero is to find out who are replicants from 15 suspects just like what Harrison Ford did in the famous 1982 movie. Out of the 15 suspects, only 2 of them are guaranteed to be replicants, the other 13, however, are determined by random elements. Each of them can either be a replicant or a human across different game playthroughs, so their behaviors are not scripted, and controlled by their own AI system, each of them has a goal to complete and they would work toward that goal, and the player has to find out and "retire" correct replicants. If they "retire" a human, the police might be attempt to arrest the player (Bates, 1997).

Sometimes game designers have gone more extreme and tried to experiment with groundbreaking ideas. A good example is the Japanese niche simulation game called *Koukidou Gensou Gunparade March*. This game gives the player a huge amount of freedom, the player can play as any of the 22 characters in game and has a huge set of choices in order to complete the goal of the game in a variety of ways, hence, according to Crawford's criteria, the player can enjoy a huge number of accessable state in this game (Crawford, 2003). Thanks to this huge freedom, the player is encouraged to revisit the game multiple times. From my observation, among all the systems, the most interesting one is the relationship system between characters. Developers of this game designed a system focused on relationships between the player character and the other 21 characters. The player character can talk freely and develop a variety of relationships with any of the other 21 characters. NPCs respond differently to the player character based on their relationships, and a huge collection of events can be triggered based on the relationship network the player builds.

All three examples mentioned above had some interesting ideas on game narrative in attempt to increase the replayability. Although these games are not very well known in recent years, their ideas still stand out today. Because these ideas can effectively boost interactivity and replayability, but are not used or explored as fully as

they could in other games. These ideas inspired this game prototype in different ways, and reflected in some way in this game prototype. The design of letting a murderer execute his action and have a detective to investigate the consequence left by the murderer is inspired by Indigo Prophecy's first scene. The design that allows some random elements to affect narrative branching during a game session is loosely based on the idea featured in Blade Runner. The simulation of interactions and affiliations between NPCs is inspired by Gunparade March. I believe it is a solid choice to implement these designs, and the replayability would benefit from these ideas.

Chapter 6.
Limitations and Future Works

This project serves as a first step of my ideas of applying AI and computational approaches on narrative in video games, there are several potential extensions to this project that are interesting and useful in the future. Also, during the development of this prototype, there are several ideas and designs that did not make it into the final outcome due to scale of the project and limitations of resources. There are also some features and contents that have been revised and simplified in order to favor the development constraint. However, I think these abandoned ideas and designs are still worth further explorations.

One of the biggest restriction of this prototype is the visual presentation. The environment used to develop this project is RPG Maker VX ACE. It is an ideal tool for building indie games and quick prototypes, because it is easy to uses and comes with a limited selection of functions which suits the basic needs of game development. However, this tools suffers from various restrictions, especially in the graphics design. Although the sprite-based 2D graphics saved me a lot of time in developing the project, it looks primitive by current standards and might not attract many people. I think it will be necessary to switch to another platform if I want to explore this research further. Another limitation is the lack of sound effects and music. Sound effects and music do a great deal to build storyworld, emotion, and tension. In this project, these aspects are omitted since I had to focus on another variable: generative plot construction. In the possible future development, the sound design would be an essential part to help the prototype become more attractive in its presentation.

Also, the story and length of the prototype also needs improvement. This prototype is short, the background settings, character settings and the complexity of the story are also simple. Future works can expand the length and complexity of the story in order to further explore computational approaches to plot development in video game narrative.

An important idea of this prototype is to use computational approaches to generate narrative which provides a different situation for players in different

playthroughs, and the process of generating these narrative situations is simulating actions as if these events were actually happened. In the situation of this prototype, the murder case is generated by actually simulating how the murder is happened. The problem in this prototype is that the simulation of the process is far from perfect: the decision of the murderer and victims feels too simple and arbitrary, the reason for killing is also simplified in favor of the simplicity of the project. Also, for the process of the crime, the murderer also magically knows where the victims instead of doing a proper search. The variety of the murder itself is also very restrictive. So the most essential task for future works is to design more complex and reasonable mechanics which makes the events generation more realistic and believable.

One possible solution to the decision of murderer and victims can be establishing a character model for each character, somewhat similar to Bungie's approach in designing enemy AI for Halo 2. In Halo 2, each class of enemy has their own distinct behavior tree, as well as different degrees of self-preservation impulses and morale. When facing the player, some enemies would engage actively, attempt to retreat only when severely damaged, while other enemies would flee on receiving first shot, or upon the death of the squad leader (Isla, 2005). Similar to the approach used in Halo 2, These character models would define these characters' backgrounds, personalities, preferences and so on. Then behaviors of these characters during the simulation can refer to these models as a guide, rather than resort to random number generator that I have implemented in current prototype. I believe that would make the behaviors of characters more believable, as well as make the simulation of interactions between characters more realistic. The outcomes from these interactions and events would be more believable and more natural. This approach would allow me to implement more complex motivations for the murder for different murderer/victim combinations. Dialogues for non-player characters can also be more personalized, or even blend in some lies inside their dialogues. This would result in a more variable narrative-rich situation.

The actual murder simulation can also have more complex variations. For example, the murderer could automatically generate real-time searches for the victim, instead of magically knowing where he/she is. An increase in variations in order to create a more sophisticated crime scene will provide a more interesting challenge to the

player. These more complicated narrative options will increase variation in player experience, and therefore increase replayability.

The core planning system itself can also be upgraded in the future. For this prototype, I have implemented STRIPS planning as the base of the planning system, because it is reliable and simple to use. However, STRIPS cannot handle a world with overly complicated knowledge base. I think future projects could explore the possibility of using more advanced planning approaches which are capable of handling more complex and realistic situations. Some of the approaches I have mentioned in the literature review are good places to start this exploration, such as Ware, Young, Harrison and Roberts' CPOCL system (2014). Also, I think other AI techniques other than planning are also worth researching, such as natural language processing for more complex generative narrative, or even genetic algorithms for mutating part of the narrative to add more indeterminate factors.

While I think it is good to have the non-player characters report their activities and meetings to the player, I am not very satisfied with the way it was implemented. In this prototype, characters simply report out to the player with a list of characters they met and where did they met, as well as their impressions on the character they met. While these reports contain essential information for solving the crime, the form of these reports is too primitive. For future projects, it would a better idea to make this function interactive by taking the form of presenting a list of questions for the player to ask, as in other detective games. A better option might be to add a language parsing function that would allow the user to type their own questions.

The overall interactive design of this prototype needs serious improvement to become truly attractive to players. This prototype is a draft version of my concepts, so it only contains basic functions that are just enough to instantiate and test out my thoughts. The game prototype has not been optimized through gameplay and iterative design in order to make the game really fun to play.

Although there is no formal user test done during the development of this game prototype, I did invite some friends to do some informal playing of this prototype. While they do like the concept of having a generative narrative section inside the game, I have observed that they had some difficulties during their play sessions. They seem to be

puzzled by some aspects of the gameplay during the investigation section, and some of them think the gameplay design is too simplistic. I believe there is some future work that can put more effort into making the interactive design more interesting, while keeping and improving its narrative focus. The project would then really become a solid game that is narrative-focused, replayable, and fun to play.

Last but not least, I think my ideas in this project not only apply to the adventure game, but also can be extend to other game genres which narrative plays a role. I believe using a computational approach to generate plot variations in other game genres can also bring new attractions to players. Applying the idea of using a generative narrative system in other game genres such as first person shooters or action games would be an interesting topic to explore in the future.

Chapter 7.
Conclusion

The main purpose of this thesis is to design and develop a game prototype that reflects my idea on solving the replayability problem in video games from a narrative perspective. Narrative-driven video games such as adventure games are the central focus of this research. This project is also an experiment to implement a methodology that combines traditional scripted narrative elements with computationally generative narrative processes using computational approaches within a video game system. In this project, the narrative part which managed by algorithmic approaches is designed to add variety into the central conflict of the the story, while the scripted part shapes the background and structure of the narrative that are difficult to be generative using simple algorithms.

The outcome of this project is a game prototype designed to follow theories, definitions, discussions and guidelines set up by several scholars and early games. This prototype features gameplay and a system developed within the adventure game genre. The major reason why I have decided to adopt the adventure game genre in this project is that this genre is usually narrative-oriented, where the pleasure from narrative is more important than that from gameplay, so it fits with my research focus. The narrative-driven nature of adventure games gives this genre a good narrative pace and hence a better immersive experience of narrative.

The design of interactivity in this game prototype also complies with elements in various interactive media theories and discussions remediation and modes of interaction. The navigable space in this game prototype is designed in a way that keeps size, variety and certain degree of realism, as well as the easiness for access in mind.

The design of game rules is taking the system of a simple adventure game and follows Zimmerman and Salen's definition of game rules. These designs are also complying with Juul's point of view on game rules: game rules as finite state machines, informed choices and puzzle designs, in order to improve the consistency and experience of this game prototype.

The narrative design of this prototype is a combination of narrative in traditional sense and generative narrative. The fundamental of generative narrative in this game prototype are based on the traditional nature of narrative as articulated by Chatman, Abbott, and other narrative theorists. The design of generative narrative in this game prototype also benefits from Ryan's, Manovich's and Miller's more algorithmic take on the nature of narrative. The generative narrative approach in this game prototype is designed to deploy a combination of algorithms to choose appropriate events from a collection of events and form a meaningful event sequence which can be considered as narrative.

On the other hand, the narrative design also complies with the narrative arc present in traditional media works as described in Bizzocchi's Paper. I have also put some effort in this game prototype to address the pacing problem raised by Bizzocchi by implementing some loose pacing controls. The design around time and micro-narrative also follows the ideas from Juul, Jenkins and Bizzocchi.

The inclusion of the algorithmic solution in the narrative is in response to Crawford's arguments on narrative branching and user choices. The approach implemented in this prototype attempted to reduce the difference of numbers between conceivable states and accessible states in order to comply with Crawford's argument.

The probability-based event scheduler is implemented as a potential answer to the need of variety in video games. I have explored Zimmerman and Salen's ideas around uncertainty, risk and certainty in video games, and designed this system based on these ideas in order to introduce more variety and uncertainty in this game prototype, in order to bring in some added variety. The outcome from this probability based event scheduler seems to be successful in creating varied situations, and I think it satisfies Salen and Zimmerman's ideas.

The core system responsible for generating the murder scenes is implemented with STRIPS planning, while the database it uses is filled with short events pre-written by human hand. STRIPS is a classic planning algorithm which its capability and effectiveness has been tested and proved by many previous projects from different research fields, including video games, so it would be a mature algorithm and system to use. On the other hand, while STRIPS has been used in some video games, it has been

barely used in narrative design in video games. I believe the inclusion of this algorithm can bring some fresh contribution in video game development regardless the fact that it is an old system.

The STRIPS based system implemented in this game prototype is using pre-written contents to simplify the situation the system handles, in order to achieve a reasonably variety with a manageable cost. Also, non-player character's movement between locations are calculated using A* algorithm for further reduced complexity.

The idea of using such a generative planning system is inspired by several earlier analog generative artifacts, such as the *Exquisite Corpse* game and *A hundred thousand billion poems* by Queneau. The planning system itself also takes inspirations from several similar researches in interactive narrative and generative narrative using planning techniques, along with other computational approaches.

There are also a number of scripted scenes that exist in this game prototype. The main reason for doing this is to let them handle the complex and detailed parts of the narrative which a simple generative narrative solution cannot handle. There are also some other projects doing this as well, and their results are promising. This approach can also be cost efficient and decrease the chance of generating game glitches and errors.

Replayability is the core problem this project tries to resolve. During the design phase of this game prototype, I have looked into design decisions made in various games that had a positive impact on extending replayability, and decided to adopt some aspects in these examples in my project. The idea that using computational approaches to manage certain game contents can be spotted in games featuring procedurally generated levels. There are also some aspects outside the planning system inspires this game prototype in different ways.

In the end, I would like to consider this game prototype as a successful reflection of my ideas. However, it is only a beginning in the long road of exploring the possibility of solving replayability problem from a narrative perspective in video games. There is still a lot to be done in order to make this idea truly excel, but I believe that it is an interesting idea to explore. The potential rewards are high, since interactive narrative in video games could be just as interesting as gameplay itself.

References

Abbott, H. P. (2002). Defining Narrative. In *The Cambridge Introduction to Narrative* (pp. 13-24). UK: Cambridge University Press.

Barrett, G. (2009). *What's a "Hobson's Choice"? A Way with Words*. Retrieved from https://www.waywordradio.org/whats-a-hobsons-choice/

Bates, J. (1997). Westwood's Blade Runner, *PC-GAMER, 4(9), September*. Retrieved from http://media.bladezone.com/contents/game/BR-PCGame1.html

Bizzocchi, J. (2006). Game and Narrative: An Analytical Framework. In *Proceedings of CGSA 2006 Symposium*. Canada: Authors & Canadian Games Study Association CGSA.

Bizzocchi, J. (2015). *Seasons: A Generative Multimedia Installation*. Retrieved from: https://ambientvideo.org/seasons/

Bolter, J. D. & Grusin, R. (2000). Immediacy, Hypermediacy, and Remediation. In *Remediation: Understanding New Media* (pp. 20-50). Cambridge, MA: MIT Press.

Bolter J.D.& Gromala, D. (2003). T-Garden: The Materiality of New Media. In *Windows and Mirrors: Interaction Design, Digital Art, and the Myth of Transparency*. Cambridge, MA: MIT Press.

Booth, M. (2009). *The AI Systems of Left 4 Dead*. Retrieved from

http://www.valvesoftware.com/publications/2009/ai_systems_of_l4d_mike_booth.pdf

Bordwell, D. & Thompson, K. (2008). *Plot and Story in Film Art: An Introduction* (8th Ed.). New York, NY: McGraw-Hill Company, Inc.

Brachman, R. J. & Levesque, H. J. (2004). Planning. In *Knowledge Representation and Reasoning* (pp. 305-323). San Francisco, CA: Elsevier, Inc.

Branting, L. K. (1999). A generative model of narrative cases. In *Proceedings of the 7th international conference on Artificial intelligence and law* (pp. 1-8). New York, NY: ACM.

Caillois, R (1961). The Definition of Play. In *Man, Play and Games* (The Free Press of Glencoe, Inc. Trans.) (pp. 3-10). New York, NY: The Free Press of Glencoe, Inc. (Original Work published 1958)

Carrano, F. M. & Prichard, J. J. (2002). Abstract Data Types. In *Data Abstraction and Problem Solving with C++: Walls and Mirrors* (pp. 112-117). London, UK: Pearson Educations Inc.

Coleridge, S.T. (1817). Chapter XIV. In *Biographical Sketches of My Literary Life and Opinions.* Retrieved from:

http://www.english.upenn.edu/~mgamer/Etexts/biographia.html

Crawford, C. (2003). Architectures. In *The art of Interactive Design* (pp. 77-90). San Francisco, CA: No Starch Press.

Csikszentmihalyi, M. (1990). The Conditions of Flow. In *Flow: The Psychology of Optimal Experience* (pp. 71). New York, NY: Harper and Row.

Digital Game Musem (n.d.). *The Rise and Fall of Adventure Games*.

Retrieved at: https://www.digitalgamemuseum.org/the-rise-and-fall-of-adventure-games/

Ermi, L. & Mayra, F. (2005). Fundamental components of the gameplay experience: Analysing immersion. In S. de Castell & J. Jenson (Eds.), *Changing views: Worlds in play, Digital Games Research Association Conference Proceedings*. Vancouver, BC: DiGRA.

Eskelinen, M. (2004). Towards Computer Game Studies. In Wardrip-Fruin, et al. (Eds.), *First Person: New Media as Story, Performance and Game* (pp. 118-130). Cambridge, MA: MIT Press.

Genette, G. (1980). *Narrative Discourse: An Essay in Method.* Ithaca, NY: Cornell University Press.

Hartsook, K., Zook, A., Das, S. & Riedl, M.O. (2011). Toward Supporting Stories with Procedurally Generated Game Worlds. In *Proceedings of 2011 IEEE Conference on Computational Intelligence and Games*, (pp. 297-304). US: IEEE.

Herman, D. (2002). *Story Logic: Problems and Possibilities of Narrative.* Lincoln, NEB: University of Nebraska Press.

Isla, D. (2005). *GDC 2005 Proceeding: Handling Complexity in the Halo 2 AI.* Retrieved from: https://www.gamasutra.com/view/feature/130663/gdc_2005_proceeding_handling_.php

James, P.D. (1997). *Murder and mystery: the craft of the detective story.* Stockholm, Sweden: Svenska förläggarefören.

Jenkins, H. (2004). Game Design as Narrative Architecture. In Wardrip-Fruin, et al. (Eds.), *First Person: New Media as Story, Performance and Game* (pp. 118-130). Cambridge, MA: MIT Press.

Juul, J (2005). *Half-real*. Cambridge, MA: MIT Press.

Kumar, D. and Ramakrishnan, N. (2008). Algorithms for Storytelling. Knowledge and Data Engineering. In *IEEE Transactions 20*(6)*,* 736-715.

Lessard, J. (2012). Adventure before adventure games: A New Look at Crowther and Woods's Seminal Program. In *Games and Culture, 8*(3), 119-135.

Manovich, L. (2000). What is New Media? In *The Language of the New Media,* (pp. 27-48). Cambridge, MA: MIT Press.

Manovich, L. (2000). The Interface. In *The Language of the New Media*, (pp. 78-88). Cambridge, MA: MIT Press.

Manovich, L. (2000). The Illusions. In *The Language of the New Media*, (pp. 205-211). Cambridge, MA: MIT Press.

Manovich, L. (2000). The Forms. In *The Language of the New Media*, (pp. 212-281). Cambridge, MA: MIT Press.

Miller, J.H. (1995). Narrative. In Lentricchia, F. and McLauglin, T. (Eds.), *Critical Terms for Literary Study* (2nd ed.) (pp. 66-79). Chicago, IL: University of Chicago Press.

Moss, R. (2011). *A truly graphic adventure: the 25-year rise and fall of a beloved genre*. Retrieved at: https://arstechnica.com/gaming/2011/01/history-of-graphic-adventures/

Murray, J. H. (1998). Immersion, Agency. In *Hamlet on the Holodeck* (pp. 97-153). Cambridge, MA: MIT Press.

Murray, J. H. (2011). Affordances of the Digital Medium. In *Inventing the Medium: Principles of Interaction Design as a Cultural Practice* (pp. 56-92). Cambridge, MA: MIT Press.

Nilsson, N. J. (1998). STRIPS Planning Systems. In *Artificial Intelligence: A New Synthesis* (pp. 373-400). Burlington, MA: Morgan Kaufmann Publishers, Inc.

Orkin, J. (2006). Three States and a Plan: The A.I. of F.E.A.R. In *Proceedings of the Game Developer's Conference (GDC), 2006*. San Jose, CA: GDC 2006.

Queneau, R. (1961). *Cent mille milliards de poèmes (A hundred thousand billion poems)*. France, FR: GALLIMARD.

Ramirez, A. and Bulitko, V. (2014). Automated Planning and Player Modelling for Interactive Storytelling. Computational Intelligence and AI in Games*. IEEE Transactions pp*(99), 1-12.

Rouner, J. (2014). *10 Games with Guinness World Records*.

Retrieved from http://www.houstonpress.com/arts/10-games-with-guinness-world-records-6392480

Russell, S. & Norvig, P. (2010). A* Search: Minimizing the total estimated solution cost. In *Artificial Intelligence: A Modern Approach* (pp. 93-99). London, UK: Pearson Education, Inc.

Russell, S. and Norvig, P. (2010). Classic Planning. In *Artificial Intelligence: A Modern Approach* (pp. 366-393). London, UK: Pearson Education, Inc.

Ryan, M. (2004). Will New Media Produce New Narratives. In *Narrative across Media:*

the languages of storytelling (pp. 337-359). Lincoln, NEB: University of Nebraska Press.

SteamSpy (2017). *Left 4 Dead & Left 4 Dead 2 Online Data Page.*

Retrieved from https://steamspy.com/app/500 and https://steamspy.com/app/550

Ryan, M. (2006). Narrative, Media, and modes. In *Avatar of story.* Minneapolis, MN: University of Minnesota Press.

Ware, S. G., Young, M.R., Harrison, B., & Roberts, D. L. (2014). A Computational Model of Plan-Based Narrative Conflict at the Fabula Level. In *Computational Intelligence and AI in Games, IEEE Trasactions 6*(3), 271-288.

Wei, H. (2011). Time and Space in Digital Game Storytelling. In *International Journal of Computer Games Technology*, *2010*(Article ID 897217), 23 pages. Egypt: Hindawi Publishing Corporation.

Wichman, G. R. (1997). *A Brief History of Rogue*. Retrieved from http://web.archive.org/web/20150217024917/http://www.wichman.org/roguehistory.html

Wolf, Mark J. P. (2008). *Genre Profile: Adventure Games*. In *The Video Game Explosion: A History from Pong to Playstation and Beyond.* Westport, CT: Greenwood Press.

Young, M. R. (2007). Story and Discourse: A Bipartite Model of Narrative Generation in Virtual Worlds. In *Interaction Studies 8*(2), 177-208. Netherlands, NE: John Benjamins Publishing Company.

Zimmerman, E. (2000). *Against Hypertext*. Retrieved from: https://static1.squarespace.com/static/579b8aa26b8f5b8f49605c96/t/59925c2f37c581f4bdfa7649/1502764079951/Against_Hypertext.pdf

Zimmerman, E. (2004). Narrative, Interactivity, Play and Games. In Wardrip-Fruin, et al. (Eds.), *First Person: New Media as Story, Performance and Game* (pp. 154-164). Cambridge, MA: MIT Press.

Zimmerman, E. & Salen, K. (2004). *Rules of Play: Game Design Fundamentals*. Cambridge, MA: MIT Press.

Anonymous (n.d.). *About Exquisite Corpse*. Retrieved from: http://www.exquisitecorpse.com/about.html

Anonymous (1982), Space Invaders vs. Star Wars. In *Executive, 24*, 9. Don Mills, ON: Southam Business Publications.

User's manual for Legend of Zelda (1987/1989). Redmond, WA: Nintendo of America.

Retrieved from: http://legendsoflocalization.com/media/the-legend-of-zelda/manuals/ZeldaNESManual.pdf