Master's Thesis
Academic Year 2018

VARIABLES - A Gaming Experience That Breaks The Fourth Wall Inversely

Graduate School of Media Design,
Keio University

Chen Jiajun

A Master's Thesis
submitted to Graduate School of Media Design, Keio University
in partial fulfillment of the requirements for the degree of
MASTER of Media Design

Chen Jiajun

Abstract of Master's Thesis of Academic Year 2018

VARIABLES - A Gaming Experience That Breaks The Fourth Wall Inversely

Category: Design

Summary

The concept of ”Breaking The Fourth Wall” is widely implemented in different art forms. Given the interactive nature, video games have the power of delivering Fourth Wall Breaking experience. However, the way in which most video games breaks the fourth wall is fairly close to traditional media such as movie and TV. The potential of the interactive nature in Breaking The Fourth Wall is yet to be discovered. This thesis and project aims at creating a new experience that Breaks The Fourth Wall in a different way, by enabling the player to directly refer to the game itself along with gaming process.

Keywords:

Game Design, Experience Design, Breaking The Fourth Wall

Graduate School of Media Design, Keio University

Chen Jiajun

Acknowledgements

It would not have been possible to finish this project and thesis without the help and support of the kind people around me, to only some of whom it is possible to particularly mention here.

I would like to express my deepest gratitude to my main supervisor, Prof. Kai Kunze. It's alway delightful and inspiring talking with him, from where I really got my horizon broadened and gained new perspectives. I'm really grateful for his support and encouragement in my study for the past three years.

It's my honor to have Prof.Tetsuya Mizuguchi, to be my sub-supervisor of this thesis. I'd say Prof.Mizuguchi is my life changer. I set up my mind to become a game designer because of Prof.Mizuguchi. His lecture two years ago was where my journey was started, and this project is one milestone of my voyage. Fusion of Activeness and Passiveness, one crucial concept of this thesis, also comes from a lecture by Prof.Mizuguchi. The conversation with Prof.Mizuguchi at EdgeOF, was one of the most inspiring moments in my life.

Thank Prof.Nakamura, my third supervisor, Prof.Inakage and Prof.Kato for their critics that motivated me to further develop my ideas and to go an extra step into my research.

The idea of this thesis came from a drunk chat with my friend George, I'd like to thank him for providing me advices on this thesis. I also hope to express my appreciation to all the professional game designers who spent their precious time discussing with me about this project. I'd like to particularly mention Zhang Jing, Liu Yu and Wang Jue, who are my former colleagues in NetEase Games and my important friends. I would also like to thank Yi, who has been always supporting me during my hardest time, fighting by my side and making me the luckiest person in the world.

Finally my most sincere love and gratitude goes to my parents, who have been always trusting me and supporting my decisions.

Table of Contents

List of Figures

Chapter 1
Introduction

VARIABLES is a new gaming experience that breaks The Fourth Wall inversely. With a special story supported depicted along with a set of game mechanics that enables players to hack and edit variables and properties of objects and elements in the game, the players are provided with an unprecedented gaming experience.

The mechanics of VARIABLES is a combination of stealth game and function of memory editing software. With a balance of cheating and stealth, the player is enabled to have a Fourth Wall breaking sandbox experience. As there are many properties in games can be reflected by variables, by spotting and editing variables in the game, players are able to change position of game objects, physical parameters in the game or even game rules.

In VARIABLES, player needs to play two roles simultaneously, which also makes the real world part of the game story, by which The Fourth Wall is broken in a special way. With the variable editing game mechanics and puzzles that reach out to real world, the player can enjoy the blending experience as a player and a developer at the same time. By providing two layers of gaming experience that player can gain experience as the role in the game and the player himself outside the game, the fourth wall is broken inversely from the player to the game itself.

Chapter 2
Related Works

2.1 Games That Breaks The Fourth Wall

2.1.1 About Breaking The Fourth Wall

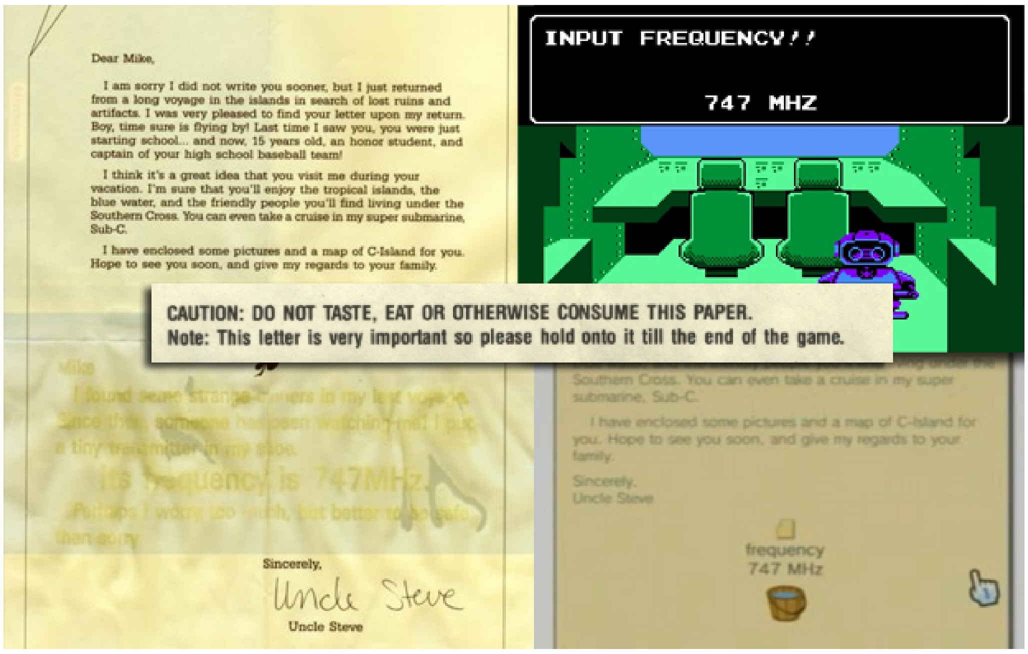
The Fourth Wall is a term used to describe the imaginary wall between the audience and the stage. "Breaking the fourth wall" is any instance in which this performance convention, having been adopted more generally in the drama is violated [2]. This can be done through either directly referencing the audience, the play as a play, or the characters' fictionality.

The temporary suspension of the convention in this way draws attention to its use in the rest of the performance. This act of drawing attention to a play's performance conventions is metatheatrical. A similar effect of metareference is achieved when the performance convention of avoiding direct contact with the camera, generally used by actors in a television drama or film, is temporarily suspended. The phrase "breaking the fourth wall" is used to describe such effects in those media. Breaking the fourth wall is also possible in other media, such as video games and books.

2.1.2 Breaking The Fourth Wall in Video Games

Given the interactive nature, video games can easily break the fourth wall. Breaking the Fourth Wall in video games is often used for a comical effect or to re-enforce the fictional natures of the game. In video games, Breaking the Fourth Wall occurs when a game becomes aware of its nature as a game, or when a character directly acknowledges the player, or when some of the game mechanics are referring to the real world [4]. Breaking The Fourth Wall is widely applied throughout the video game history.

StarTropics is one old school example of Breaking The Fourth Wall. Before the game begins, the player can find a real paper letter inside their game package. The letter outlines the plot and tells the player where to go at the start of the game. From then on, the game plays as normal until you receive a message asking you to dip the letter in water. Doing so revealed a code, which was used later in the game and gave directions on where to go next.



Dear Mike,

I am sorry I did not write you sooner, but I just returned from a long voyage in the islands in search of lost ruins and artifacts. I was very pleased to find your letter upon my return. Boy, time sure is flying by! Last time I saw you, you were just starting school... and now, 15 years old, an honor student, and captain of your high school baseball team!

I think it's a great idea that you visit me during your vacation. I'm sure that you'll enjoy the tropical islands, the blue water, and the friendly people you'll find living under the Southern Cross. You can even take a cruise in my super submarine, Sub-C.

I have enclosed some pictures and a map of C-Island for you. Hope to see you soon, and give my regards to your family.

Mike

I found some strange [illegible] in my last voyage. Since then, someone has been watching me! I put a tiny transmitter in my shoe.

Its frequency is 747MHz.

Perhaps I worry too much, but better to be safe than sorry.

Sincerely,

Uncle Steve

Uncle Steve

CAUTION: DO NOT TASTE, EAT OR OTHERWISE CONSUME THIS PAPER.
Note: This letter is very important so please hold onto it till the end of the game.

INPUT FREQUENCY!!

747 MHZ

Southern Cross. You can even take a cruise in my super submarine, Sub-C.

I have enclosed some pictures and a map of C-Island for you. Hope to see you soon, and give my regards to your family.

Sincerely,
Uncle Steve

frequency
747 MHz

Figure 2.1: *StarTropics* Player needs to dip the letter into water to see the frequency

The Mother Series loves to make the player a part of its narrative. In Mother 2, the only thing that could defeat a primordial force of evil was something which is the most powerful power something that existed only outside of the game the player. The death animation for the final boss is modeled after a television getting overwhelmed by static before turning off. Some interpret the ending as implying the only way to defeat this evil was to exert your power as a player outside the game world and turn the game off.

In Eternal Darkness, a game loosely based on Lovecraftian horror, player needs to manage a sanity meter. As your characters were exposed to the horrors, their sanity dropped which causing them to see things that doesnt really exist. Slight sanity damage makes you see enemies that doesnt exist or walk down hallways that never ends. Huge sanity hits makes you think that the game crashed or delete your save files. It was a fantastic way to bring horror and insanity out of the game and into your living room.

Making narrator as part of the game mechanics is also one approach to Break the Fourth Wall. The Stanley Parable and ICEY are two typical examples of this approach. Along with the process of the game, the narrator will give player instructions. The player can choose to follow the given instruction or not. The narrator will talk according to the players action. For example, at the beginning of ICEY, the narrator tell the player to turn on the switch so as to trigger the bridge. If the player keeps jumping off the edge, the narrator will become inpatient and even restrict the main characters action space until he finish talking.

Undertale is also an excellent example of games with the core Breaks The Fourth Wall mechanics. Undertale is a game about morality and choice just like in which you can see in game titles by Peter Molyneux, but not just the choices you make when playing and it goes beyond. It criticizes every choice you have made in every game you have ever played. The main character in Undertale knows he is a character in a game and he has his own will and struggle. It asks you whether or not you treated the characters in these game worlds like people, or like playthings. It makes you wonder whether or not this is how an all-powerful deity might see us.

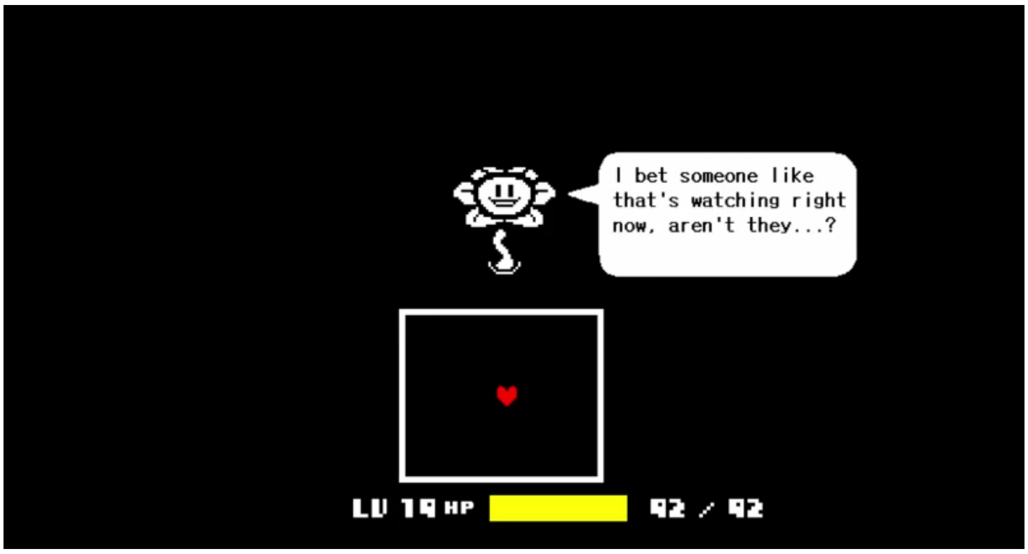

I bet someone like that's watching right now, aren't they...?
LV 19 HP
92 / 92

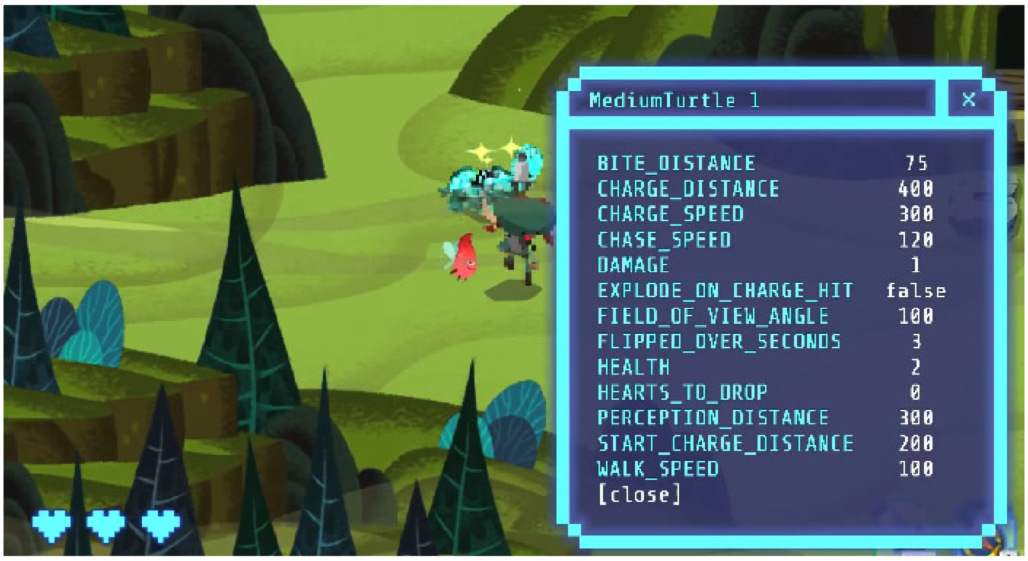
Figure 2.2: *Undertale* Character in the game mocking the player

All the examples mentioned above are all great and typical examples about Breaking The Fourth Wall in video games. The following two related works are video games that Breaks the Fourth Wall with similar hacking game mechanics as VARIABLES.

2.1.3 Hack n Slash

Introduction

Hack n Slash is a puzzle action game about hacking – reprogram object properties, hijack global variables, hack creature behavior, and even rewrite the games code. Players are allowed to use in-game tools to hack the game while theyre playing it. The sword of the main character can hack the variables of objects. The players can find magic artifacts that allow them to tune global variables. Players can also discover equipment that lets them see the games internal debug visualization to uncover things that werent meant to be seen. As the players achieve advanced hacking mastery, theyll be able to dive directly into the games assembly in the form of procedurally generated dungeons and modify the running code.



```
MediumTurtle 1                              ×

BITE_DISTANCE                75
CHARGE_DISTANCE              400
CHARGE_SPEED                 300
CHASE_SPEED                  120
DAMAGE                       1
EXPLODE_ON_CHARGE_HIT        false
FIELD_OF_VIEW_ANGLE          100
FLIPPED_OVER_SECONDS         3
HEALTH                       2
HEARTS_TO_DROP               0
PERCEPTION_DISTANCE          300
START_CHARGE_DISTANCE        200
WALK_SPEED                   100
[close]
```

Figure 2.3: *Hack n Slash*Editable values and Properties

Discussion

Hack and slash games refers to a type of gameplay that emphasizes combat. The term ”hack and slash” was originally used to describe a play style in tabletop role-playing games, carrying over from there to MUDs, MMORPGs, and role-playing video games. In arcade/console-style action video games, the term has a different usage, specifically implying a focus on real-time combat with hand-to-hand weapons as opposed to guns or fists.

These elements result in that Hack and Slash games are strongly static driven. Upgrading weapons and armors, gaining talent point so as to upgrade abilities or unlock new moves or magic, attacking, defending, weakening or being weakened, these elements are technically statics. By eliminating enemies or completing tasks, the improvement of power and defeating strong enemies is from where players in hack and slash games get fun. Allowing players to edit variables that directly connected to the power will destroy the core entertaining experience in hack and slash game.

Another essential element of hack and slash games is enemy behavior. Learning the behavior pattern of the enemies, such as attacking frequency and range, finding the weakness and reacting to enemies actions, is also a core mechanics and gaming experience in hack and slash games. Enabling the players to edit the enemies behavior pattern, such as disabling damage or changing charging distance, will completely destroy the balance and experience.

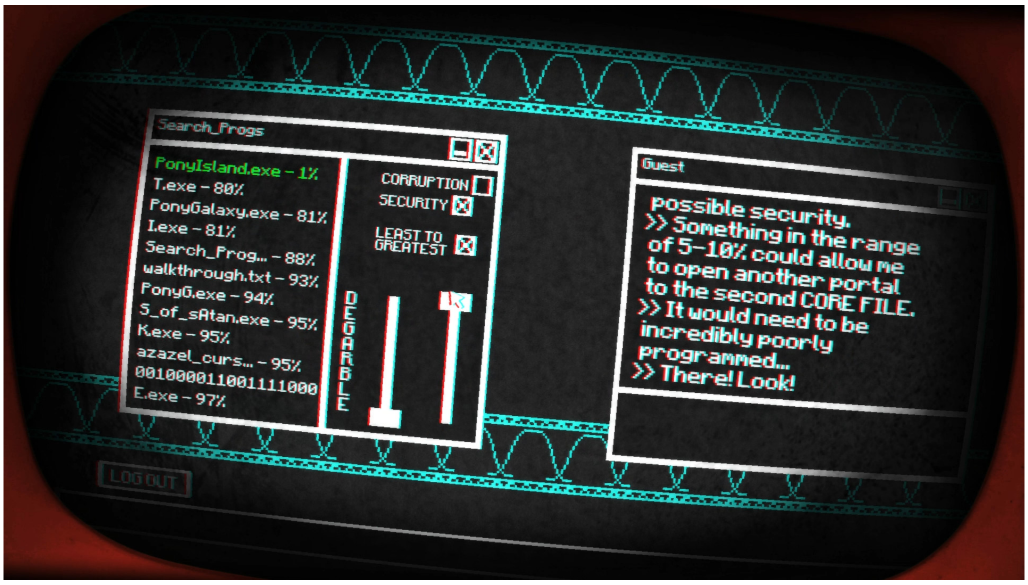
Secondly, hack and slash is incompatible with the hacking game mechanics. Hack and slash game is about minimizing waiting and about making action as engaging as possible and minimizing the time spent not that action. The player also cannot focus on both taking action and figuring out solutions to puzzles simultaneously. Maybe thats why in Hack n Slash, properties and meaning of each variable is shown to the player. As a result, the player is not actually hacking or finding solutions to puzzles but simply modifying properties of the game so as to empower the main character or disempower the enemies.

Thirdly, Hack n Slash enables the player to hack the game unlimitedly while VARIABLES doesnt. In Hack n Slash, players can edit properties and variables as many times as they want, which means there is no try and error process in the game and no penalty of mistake. In VARIABLES, times of variable editing is limited, which means the player needs to consider carefully at every step and try to make the most of every single edits.

2.1.4 Pony Island

Introduction

Pony Island is another game that Breaks the Fourth Wall and with the game mechanics of hacking. Pony Island is a suspense puzzle game in disguise. Daniel Mullins, game developer of Pony Island, wanted to create a game that defied players' expectations from standard game interfaces, and "flipping them upside down" [14]. As a metafictional game, the game has the player interact with what appears to be an old arcade cabinet game called "Pony Island". The player soon discovers the game is corrupted by a demonic being which is trying to claim the player's soul for itself. The player is aided by the soul of a previous player who helps the player access Pony Island's internal programming to get around the traps left by Satan and save their soul.



```
PonyIsland.exe - 1%
T.exe - 80%
PonyGalaxy.exe - 81%
I.exe - 81%
Search_Prog... - 88%
walkthrough.txt - 93%
PonyG.exe - 94%
S_of_sAtan.exe - 95%
K.exe - 95%
azazel_curs... - 95%
0010000011001111000
E.exe - 97%

CORRUPTION ☐
SECURITY ☒
LEAST TO GREATEST ☒
```

```
possible security.
>> Something in the range
of 5-10% could allow me
to open another portal
to the second CORE FILE.
>> It would need to be
incredibly poorly
programmed...
>> There! Look!
```

Figure 2.4: *Pony Island*Editable values and Properties

Discussion

Pony Island and VARIABLES share the same concept of playing a game in a game. In Pony Island, players will find themselves ensnared in a battle for their soul with malicious demonic files that have taken over their computer. Some will even use the Steam interface to screw with the players, making entities from inside the game contact the players outside the game. In VARIABLES, players plays two roles simultaneously, a programmer that hacks the game and the digital copy of his mind in the game.

The main difference between the two games is the core game mechanics. Pony Island is a story-based linear puzzle game while VARIABLES is a stealth games. This result in different roles of the hacking mechanics plays in the two games. In Pony Island, because its a linear puzzle game, the answer to every single puzzle is fixed and preset by the designer. Hacking is part of the story and it plays a role as the form of puzzles. Hacking is not one of the game mechanics but the game itself. The player has to figure out what the designer is thinking and find out the one and only solution to every puzzle so as to clear the levels. In VARIABLES, hacking plays the role as one of the tools provided to the player and player are enabled to use this tool in the way they want. As I mentioned above, stealth games are actually puzzle games with multiple solutions. In conclusion, even though the two games share the same concept and similar mechanics as hacking, the hacking mechanics play different roles and affect the game system in different ways in these two games.

2.2 Stealth Games

2.2.1 About Stealth Games

Stealth game is a type of video game in which the player primarily uses stealth to avoid or overcome antagonists. Games in the genre typically allow the player to remain undetected by hiding, sneaking, or using disguises. Some games allow the player to choose between a stealthy approach or directly attacking antagonists, but rewarding the player for greater use of stealth. Elements of "stealth" gameplay, by way of avoiding confrontation with enemies, can be traced back and attributed to a diverse range of games, including Pac Man (1980). The first stealth game can be traced back to Manbiki Shounen (Shoplifting Boy), published in November 1979. [20] The PET 2001 personal computer game was developed by Hiroshi Suzuki which involves a boy entering a convenience store and attempting to shoplift by stealing money symbols while avoiding the line-of-sight detection of the owner. If caught, the player is led away by the police. The genre became a mainstream success in 1998, with Tenchu: Stealth Assassins [7], Metal

Gear Solid [10], and Thief: The Dark Project [19] all being released in that year. These games were followed by other successful stealth series, such as Hitman and Splinter Cell. Unlike most action games, stealth games challenge the player to avoid alerting enemies altogether. [8] The core gameplay elements of the modern stealth game are to avoid combat, minimize making noise, strike enemies from the shadows or traverse the environment without interacting with enemies.

Basically most stealth games are not about eliminating enemies in the level. The goal of one level is usually not about elimination, at least its not about eliminating all the enemies. Its usually about sneaking to a position, rescuing hostage, breaking down a machine, stealing an item or in the worst case assassinating an assigned target. Many stealth games actually dont encourage players to attack or kill normal enemies. In some stealth games players even get a worse review if they kill enemies or killing enemies may cause passive consequences during the gameplay. Short finishing time and combat avoidance are usually rewarded in stealth games. For example, in Metal Gear Solid, players will hear really disencouraging sound effect if they kill an enemy and get really low rating when level is cleared. But players will get really high rating if they clear the level within a short time, in a special way, without being spotted and avoid killing. In Dishonored, killing enemies may cause consequence in future gameplay and even affect the ending of the game.

Stealth games are about making waiting engaging. Almost every other type of games is about minimizing waiting and about making action as engaging as possible and minimizing the time spent not that action. In games where expectation is that the player will be constantly moving or attacking, if you let your players keep their gun, players will want to solve their problems with their gun, and that desire to do so is just intensified if you try to force them to creep around quietly. So stealth game is not about empowering the players but providing them with tools to trick the enemies. In stealth games, the players are not actually waiting, they are solving a puzzle. Each moment hidden is a moment spent observing and working through the next step of the puzzle. Stealth game design and level design is much more akin to designing puzzles than designing encounters like action games or shooting games. Each ledge and each patrol path is another piece and a large interlocking logic challenge.

As stealth games are about puzzles, why stealth game is different from typical puzzle games? Basically there are two types puzzles, puzzles with one solution, where players need to figure out what the designer was thinking and replicating it;

and puzzles with multiple solutions, where players are given a problem and a set of tools and they use these tools to solve the problem however they want. Good stealth games rely on second type of puzzles. Takedowns, distracting the enemies, camera disables, the ability to crawl through ventilation shafts or parkour on the rooftops, these are all tools provided to the players. Their logical functions which can be applied to the problems at hand. One typical example is Hitman, in which players are provided with multiple approaches to assassinate the target without really interacting with the target.

Stealth games teach the players to think these techniques and solve the puzzles in the game. For example, in Mark of the Ninja [6], there is a lot of focus on providing the players with a bunch of easy to understand ways to traverse their environment. In Mark of The Ninja, players can destroy the cameras or lights using darts or grapple up to a catwalk. In Dishonored, players are able to teleport to some specific points. Stealth is about traversal more than it is about combat. So delivering interesting ways to traverse is more valuable to players than providing them with powerful weapons which can eliminate enemies easily. The stealth fantasy is very different from the power fantasy that most action games deliver on. Stealth is about triumphing despite being disempowered and beating the odds and outwitting the enemy. This is completely different from the dominance games involved in mowing down waves of enemies.

Gameplay in stealth games is about traversal, interacting with enemy and environment is therefore an essential element. Interacting with enemies is usually about distraction and its widely seen in stealth games. In many stealth games in which players can pick up items, players can distract enemies by throwing the item to a point and cause some noise so as to make the enemies go and check, by which the traversal path is cleared and players can sneak through the position. Interacting with environment holds a larger range of possibilities. Players can cause distraction, eliminate target or switch sight by interacting with environment. For example, in Hitman, players can assassinate the target by causing accidents using the environment.

In stealth games, players also need fast iteration time. Because stealth games are inherently about waiting, making waiting time interesting is also one important issue. Stealth games also needs to make escaping rewarding while also minimizing the downtime between attempts at solving a section of the game. The punishment for failing an attempt is just as engaging as the rest of the experience. Stealth games are about waiting than about action, about triumphing despite weakness rather than displaying power.

2.2.2 Metal Gear Series

Introduction

Metal Gear (Japanese:) is a series of action-adventure stealth video games, created by Hideo Kojima and developed and published by Konami. Hideo Kojima designed the original Metal Gear, which debuted in Japan and Europe in 1987 for the MSX2 computer platform. According to Hideo Kojima, he made the first Metal Gear a stealth game because of the limited power of MSX2 [11]. At first Kojima was assigned the task of making a shooting game for MSX2 but due to the low power of MSX2 which can only process 8 sprites on the game scene, while coloring one enemy takes one extra sprite, he had to come up with a solution. Finally inspired by a traditional game he played in his childhood, he decided to make a game whose main focus is not intense combat. Many shooting games on the market back then are mainly focus on combat while the target of players is eliminating the enemies. But in Metal Gear, players main target is sneaking through without being spotted enemies and reach a certain position. If the player is spotted by one enemy, the enemy will start chasing and attacking the player. An alarm will also be triggered and reinforcement will be called. Due to the huge gap of power between the enemies and the player, the player needs to escape and hide until the enemies stop searching for him.

Stealth in Top Down 2D Environment of Metal Gear 1 and 2

Metal Gear 1 and 2 are both top down 2D action stealth game and thats one of the reasons why I take them as related works. In these two game titles, enemies has 4 facing directions, up, down, left and right. Enemies will patrol along a certain route or change facing direction in a specific pattern. The only way to sneak through is to pass through the route while the enemy is not facing to, sneaking behind objects that blocks the sight of enemy or hiding in the legendary item, the cardboard box and move. In my opinion, there are actually two types of stealth games. One type is purely action oriented stealth games, like Tenchu Series [7] and Assassins Creed Series [13], in which main characters has really high action skills and player can assassinate enemies easily. Even been spotted by enemies, the main character has the equivalent power to fight and higher skills than enemies to move so as to escape easily. Players in such type of stealth games are provided with a set of tools, like a range of weapons, but they are all empowering the main character to eliminate enemies. Another type of stealth games is like Metal Gear and Thief Series [18]. In such type of stealth games, enemies have absolute power advantage over the player and player can hardly survive the attack of the enemy once got spotted, not to mention eliminating the enemies. If the player is spotted by the enemies, he can only escape and hide until the enemies stop searching for him.

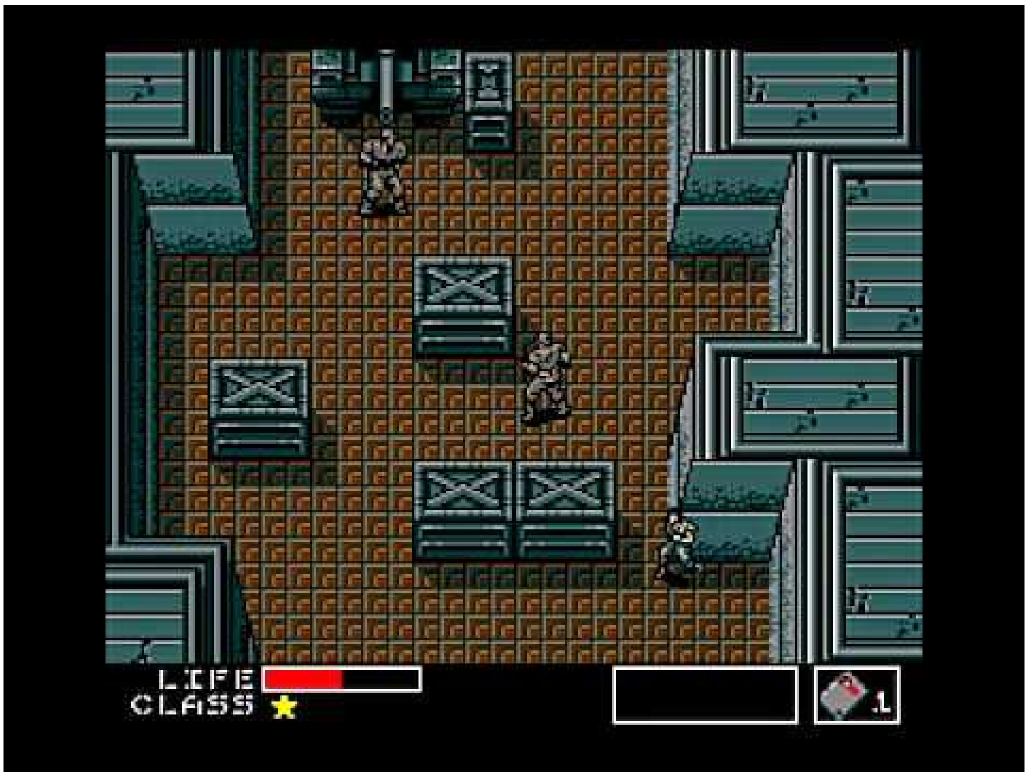


Figure 2.5: *Metal Gear:* 4 Directions of Characters

As for puzzles in games, the first important part is to let the player learn about the puzzle. In Metal Gear 1 and 2, both are top down stealth games, they did a really good job at making the puzzles clear enough to the player. Both the main character and enemy have only 4 facing directions. Each enemy also has a simple patrolling path, basically walk along a rectangle or merely change facing direction.

Enemy Distraction

As I mentioned above, distracting enemies is one elementary approach for the player to traverse. In the early Metal Gear 1 and 2, players can distract enemies by causing some noise. One readily available method is knocking on the wall. When players knock on the wall, the enemies will go to where the noise was caused and players can sneak through the path. Along with the evolution of Metal Gear Series, players have increasing number of ways to distract enemies. One famous approach is dropping a porno magazine on the floor and when the enemy sees it, he will be fully concentrating on reading the book and players can easily take him down or just traverse.

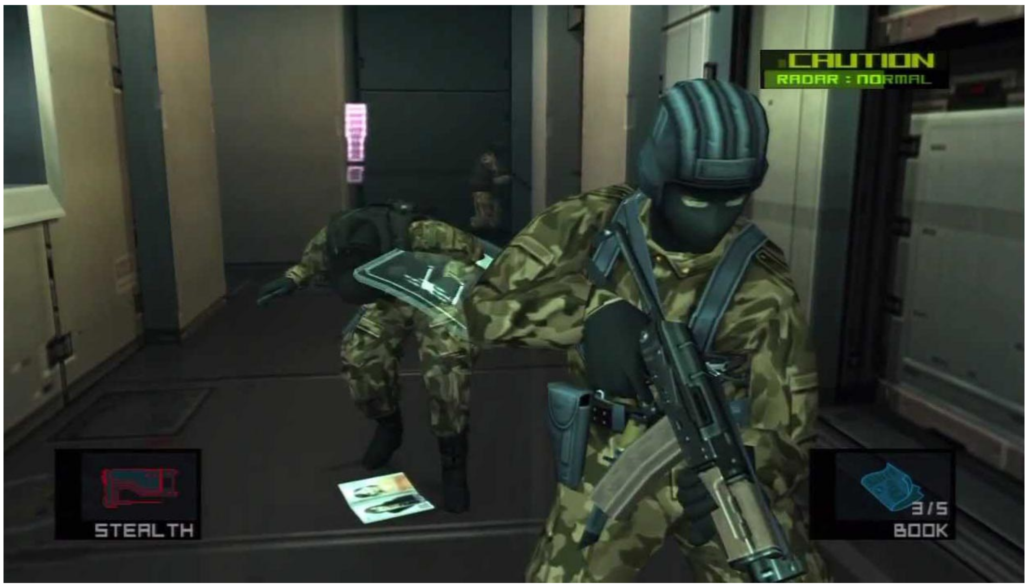


Figure 2.6: *Metal Gear Solid:* Enemy Distracted by Magazine

Sandbox Gameplay

The second reason why I pick Metal Gear Series as related work is the sandbox like gameplay mechanics, especially in Metal Gear Solid V: The Phantom Pain. As I mentioned above, stealth games strongly rely on puzzles with multiple solutions. In Metal Gear Solid V: The Phantom Pain, players are put in a huge sandbox and provided with a large variety of tools. This is also what VARIABLES is aiming at. And two scenes in the Metal Gear Solid Series really inspired me on designing and developing game mechanics in VARIABLES.

One scene is that the first time when Snake encounters Quiet, a powerful sniper who has the ability to teleport and can kill the player with two shots easily. When Snake gets close to Quiet, she will teleport to another spot. And her aiming is extremely precise and taking her down with shooting is relatively difficult. However, there is one tricky solution to this puzzle discovered by players. In Metal Gear Solid V, there is one system that players can call for supply. The player can choose the items he needs and the items will be airdropped to the assigned spot in a cargo. The fast approach to defeat Quiet is calling for supply and marking the position of Quiet as the dropping spot. Getting hit by the airdropped cargo Quiet will faint and the mission is completed.

Sandbox gameplay in Metal Gear Solid V is strongly relied on realistic and corresponding AI behavior. The second example Id like to mention is when I was playing one level, in which the task was rescuing a hostage, I completed the mission by causing an accident. During the mission, the hostage was transported from one enemy base to another. During the transportation, the hostage was put in the trunk of a truck which followed by a jeep. What I did was placing a remote-controlled bomb on the track and bombed the jeep when the team passed by. After the jeep was bombed, all the enemies in team including the enemies in the truck got out of the cars and started searching the jeep. I caught this timing and rushed to the truck, unloaded the hostage from the truck and carried the hostage to a safe place.

Stealth in Open Space Environment

The third part of Metal Gear Series which is related to my work is open world level design. In most stealth games, levels are linear and sequential designed. The levels are also divided into several sequential scenarios. The players are restricted in a limited space every time where there is a puzzle await to be solved. The route from the beginning of the level to the end is preset and the whole level is divided into several sections. The players solve the puzzles one after another and finally complete the level. In this case, level design is relatively easily because the solution to the puzzle is more or less preset, and behavior of players is more or less predictable. But in open world stealth games like Metal Gear Solid V, the mission and level is placed in a huge sandbox. Players is allowed to infiltrate into the target position from any direction, with any route. Whats more, players are provided with a large variety of tools to accomplish their mission, how to make the gameplay balanced is a huge challenge for game design and level design. Even in open world stealth games such as Assassins Creed, almost all the missions are spatially isolated from the open world environment. They are either placed in a tall castle or a a church which is blocked by walls, or the target to be assassinated walks along a preset route where there is walls or architectures on both sides.

In Metal Gear Solid V, this problem is solved as follows. Firstly, the target of the mission is kept in secret to the player at the beginning. For example, in one mission, the player is told to assassinate one general but the player doesnt which person is the target. The player needs to firstly infiltrate into the base and use the telescope to identify and spot the target. The target is not staying at one position but keep moving in a pattern which force the player to search around without being spotted. Another example is that in another mission, in which the goal of the task is rescuing one hostage but the exactly position of the hostage is be to found by the player. By keeping the target unknown the game successfully prevent the player from highlighting the optimal infiltrating path easily when the first time they play through each mission.

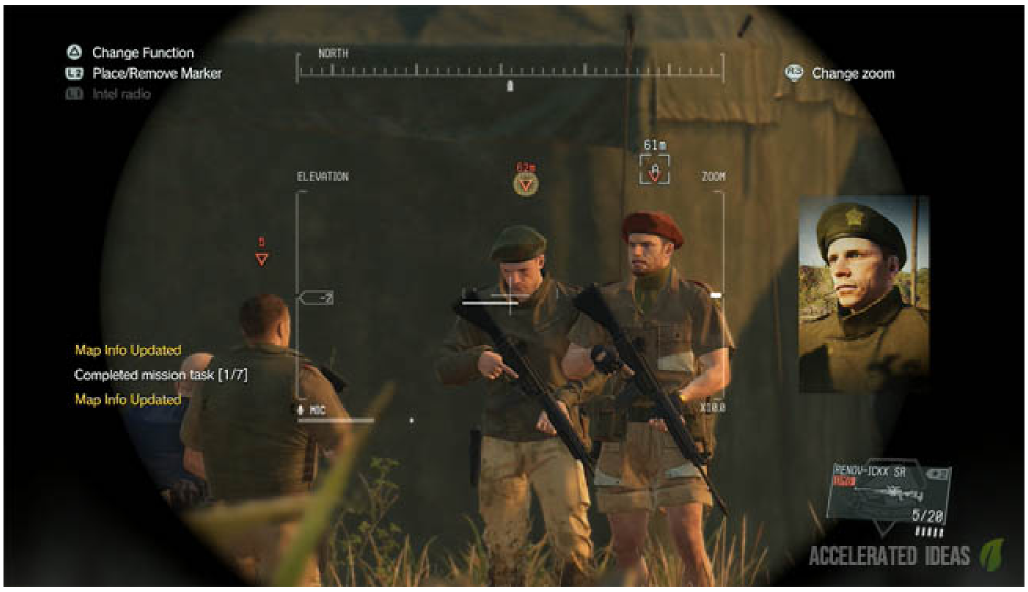


Figure 2.7: *Metal Gear Solid V:* Identifying The Target

Another approach of Metal Gear Solid V to solve the open world issue is relatively straightforward. They equally deploy guards at every single direction while the target is usually placed in the middle. This makes difficulty of infiltrating from each direction relatively well balanced.

Another approach is quite straightforward. Unlike 3D environment with third person perspective, in which the player can observe the whole environment from a high spot, in top down 2D environment the range of players sight can be easily restricted, the player therefore needs to get close enough to spot the target. This makes the puzzles in the game divided by the range of sight of the player.

Breaking The Fourth Wall

The last reason why I picked Metal Gear Series is that Breaking The Fourth Wall is frequently used in this series.

In Metal Gear 1, after reaching Building 3's 100th floor basement in Outer Heaven, Big Boss will talk to the player via the communication system which Snake uses in the game to communicate with other members. In Metal Gear Solid, Master Miller warns the player to use the bathroom frequently so as to not miss one of the game's cutscenes, and to not play while tired, after eating, after a bath, or in the dark. He also advises the player to trust their instincts as a gamer and think as a map designer.

The player must find Meryl Silverburgh's frequency on the back of the game's package. In The Essential Collection version of Metal Gear Solid, her Codec frequency is in the manual. It's also featured on the back of the collector's box, due to the Kojima Productions logo put on the back of the game case. In the PlayStation Network version, it is in the game's digital manual. This is one typical example of Breaking The Fourth Wall inversely.

The most famous Breaking The Fourth Wall example may be the battle with Psycho Mantis. Mantis reads the player's memory card. If the player has corresponding Konami save files, Mantis will comment on them (e.g. "You like Castlevania, don't you?"). And if the player doesn't have anything on their memory card, or anything Mantis recognizes, he'll say that "your memory is completely clean."

If the player is using a DualShock controller, Mantis tells the player to place it on the ground and that he will move it with the power of his mind. In The Twin Snakes, Snake physically turns to face the camera, and nods approvingly to the player. Before the fight, Mantis will check to see how many times the player has saved the game. If the player has saved less than three times, Mantis will say,

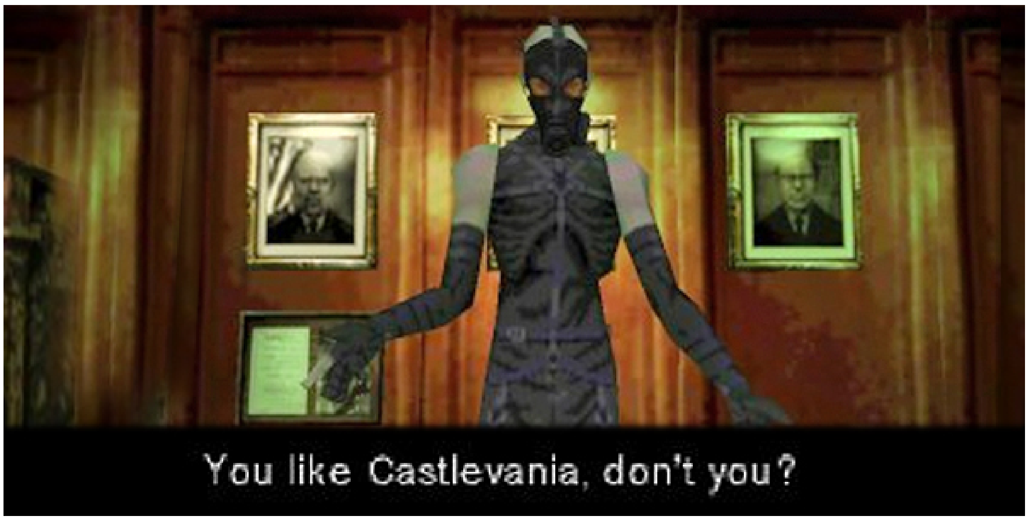


Figure 2.8: *Metal Gear Solid :* Mantis Reads Save Files

”You’re somewhat reckless.” If the player has saved more than three times, he will call them prudent. Mantis will also comment on how the player has progressed during the game up until their encounter. If the player has successfully avoided enemy traps, he will comment, ”You are either very cautious... or you are a coward.” If the player performs well in sneaking and/or combat, he will praise Snake for being a skilled warrior, and being well suited to this stealth mission. Using his ”blackout” technique, Mantis will change the player’s screen to a black screen reading ”HIDEO” in the top corner, as to replicate most standard television sets’ VIDEO setting.

Campbell tells Snake that he must physically change the game controller’s port on the PlayStation or GameCube, to prevent Mantis from ”reading their mind.” In The Twin Snakes, Mantis will cause ”sanity effects” during the battle by making the overhead camera tilt at an angle and cracking the screen if the player manages to make Meryl shoot at the screen.

One more example is a tricky method to defeat The End, an extremely aged(over 100 years old) but skillful sniper in Metal Gear Solid 3. Just like Quiet, its extremely difficult to defeat The End via shooting. But if you save the game at the beginning of the battle and then change the PS2 clock to 1 week later, when you load the game again, you’ll see The End dies of old age and there is even a specific cut scene for this.

2.2.3 Streets of Rogue

Introduction

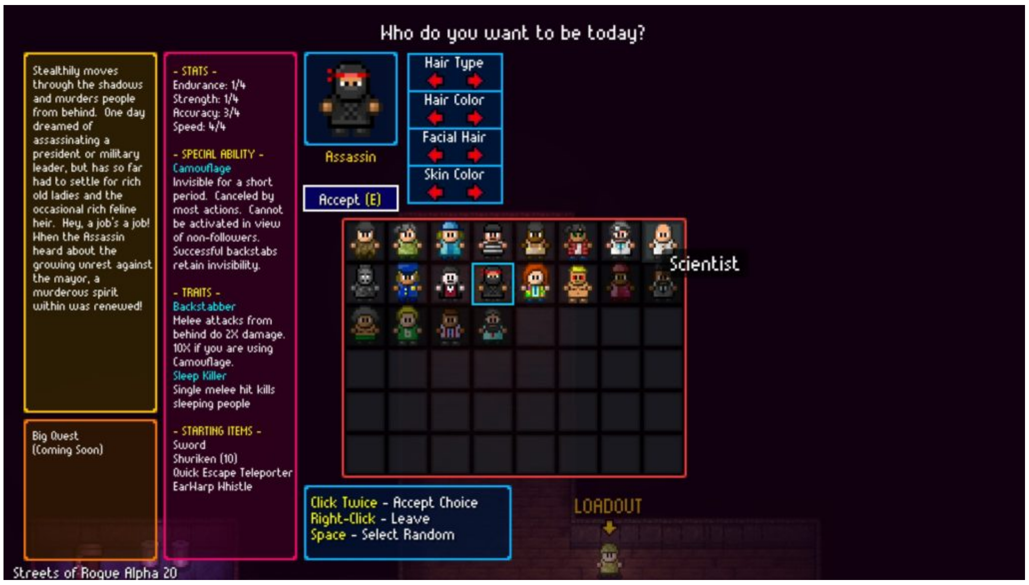
Streets of Rogue is a rogue-lite stealth game. The game takes inspiration from fast-paced top-down rogue-lites like Binding of Isaac [12], and adds free-form, experimentation-driven, emergent gameplay elements of RPGs. Rather than taking place in a dungeon, the game is set in a functioning, procedurally generated city, where complex AI informs denizens from all walks of life, who are just trying to get by in their daily activities. In order to progress, the player will need to accomplish specific mission goals in any way they see fit through use of their special character traits, items, and the environment.

The player will choose to play as one of the characters. Each character has his own starting weapon, special abilities and weaknesses, and there are different requirements to unlock each character. For example, Ninja can use Camouflage to stay invisible for a short time and to unlock Ninja player needs to clear a level without killing anyone and without alerting anyone. Cop can use handcuffs to neutralize a target with handcuffs but will lose XP will he attack innocent people, and to unlock Cop player need to bride 3 cops across multiple runs.

All the characters in the procedurally generated levels are the characters that player can choose to play as. There are different interactions between characters and these interactions will affect the process of the game. For example, if you choose cop in one game and the task of one level is rescuing a character in the police office, the player can just walk in the police office and talk to another cop. The other cop will just give you the key of the cell and you can easily release the target. But if you play as other characters, you will be warned to leave the police office otherwise the cops will attack you.

Top Down 2D Stealth

The character action and animation part in VARIABLES is almost the same as Streets of Rogue. Unlike many 2D top down shooting games in which either characters are always facing the player with weapon rotation or the character can only shoot at the direction where hes facing, Streets of Rogue and VARIABLES combine weapon rotation and character facing direction in a 2D environment. The reason why is that facing direction of enemies refers to the direction of his sight, which is an essential part in stealth mechanics. But unlike old school top down stealth games like Metal Gear, Streets of Rogue allows player to aim and shoot


Who do you want to be today?

Stealthily moves through the shadows and murders people from behind. One day dreamed of assassinating a president or military leader, but has so far had to settle for rich old ladies and the occasional rich feline heir. Hey, a job's a job! When the Assassin heard about the growing unrest against the mayor, a murderous spirit within was renewed!

Big Quest
(Coming Soon)

- STATS -
Endurance: 1/4
Strength: 1/4
Accuracy: 3/4
Speed: 4/4

- SPECIAL ABILITY -
Camouflage
Invisible for a short period. Canceled by most actions. Cannot be activated in view of non-followers. Successful backstabs retain invisibility.

- TRAITS -
Backstabber
Melee attacks from behind do 2X damage. 10X if you are using Camouflage.
Sleep Killer
Single melee hit kills sleeping people

- STARTING ITEMS -
Sword
Shuriken (10)
Quick Escape Teleporter
EarWarp Whistle

Click Twice - Accept Choice
Right-Click - Leave
Space - Select Random

Figure 2.9: *Streets of Rogue :* Different Classes in The Game

at any direction.

As a procedurally generated rogue-lite, Streets of Rogue is also tile based. The environment of each level is composed with obstacles and items. Unlike many top down RPGs in which player needs to enter houses to find out whats going on inside, in Streets of Rogue there is no rooftop visual coverage in the game while all targets are set indoor and guarded by NPCs. VARIABLES has a similar way to build level environment, tile based and without visual coverage of indoor environment

Environment-based Game Mechanics

In order to craft a sandbox gameplay in 2D environment, Streets of Rogue did a great job in creating environment-based game mechanics. There are trap, explosive items which can blow up the surroundings and the air filtering system, into which you can insert drugs and make all characters in the house affected by the drug and escape from the house. This game mechanics is more enjoyable when you play as Hacker, a class who can remotely hack and control items. The Hack can volume up a TV to distract a character and make the TV explode, make a refrigerator move, disable alarm system and trigger traps.

In VARIABLES, I also make efforts on crafting environment-base gameplay. I set items that respond to the environment in the game. For example, there are some automatic guns that shoots the player when it detects invader. By hacking the gun and changing the target of the gun, it will attack enemies that enters its detecting area. There are also traps and explosives in the game which can be hacked or triggered.

2.3 Cheating in Video Games

2.3.1 Introduction

As the hacking and variable editing game mechanics in VARIABLES works in a similar way as cheating in video games, discussing about this topic is relevant to the concept design of the game. Cheating in video games involves a video game player using non-standard methods to create an advantage or disadvantage beyond normal gameplay. Cheats may be activated from within the game itself such as a cheat code implemented by the original game developers, or created by third-party software or hardware.

However, a cheat industry emerged as gaming systems evolved, through the packaging and selling of cheating as a product. Cheat-enablers such as cheat books, game guides, cheat cartridges helped form a cheat industry and cemented cheating as part of gaming culture. [3] In some games, cheat codes can even works as part of the game mechanics and plays an important role as creating special gaming experience.

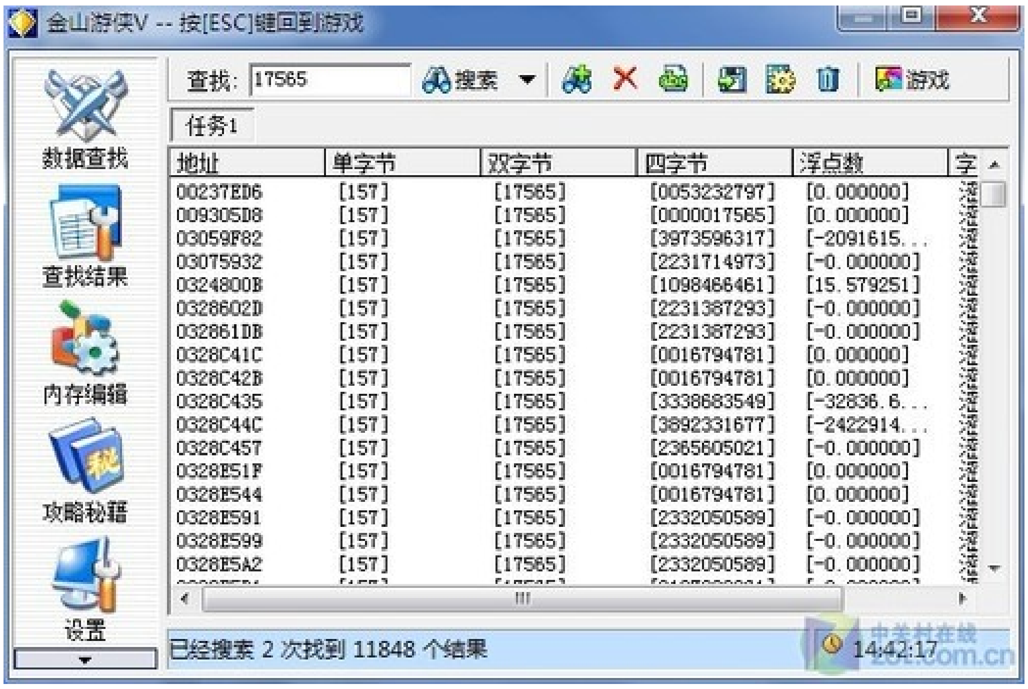
2.3.2 Memory Editing Tools

Cheating can easily be achieved by modifying the game's data while it is running and cheating via memory editing which involves modifying the memory values where the game keeps its status information is a common way. Hacking mechanics in VARIABLES is fairly similar to the way of how memory editing works. One popular example of this is Game Genie for Genesis, NES, Super NES, Game Boy, and Game Gear game consoles. [16] Another way of achieving this is by memory editor software, which allows the player to directly edit the numeric values in a certain memory address. This kind of software usually includes a feature that allows the player to perform memory searches to help the user to locate the memory areas where known values (such as the number of lives, ammo or health point) are located. Provided a memory address, a memory editor may also be able to freeze it, preventing the game from altering the information stored at that memory address.

This is where the concept of hacking mechanics in VARIABLES comes from. The hacking and variable editing mechanics basically works in a similar way as memory editor softwares. Many in-game elements can be reflected by variables, by editing or fixing the values in the memory areas, the consequence can be directly reflected in the running game, which enables players to gain a special gaming experience and even redefine the game system and even enables the players to play a different game in the preset game level and game environment but in a different set of game rules and mechanics. The players can even create their own game and craft their own game mechanics easily by memory editing.

The process of locating and editing is already an entertaining game mechanics, with the combination of puzzle and sandbox. In my childhood, I used to play games using a memory editing tool called Jinshan Youxia(:). Jinshan Youxia works in the same way as many mainstream memory editing tools like Game Trainer. When I was playing Commandos: Beyond the Call of Duty [9], a stealth-oriented real-time tactics game in which player controls a team to infiltrate into enemy base and complete some tasks, memory editing tool enables me to gain special gaming experience. In the game, there is one sniper in the team controlled by the player. Commandos: Beyond the Call of Duty is an extremely difficult game and it was nearly impossible for me to clear even the first level. However, as I attempted to use Jinshan Youxia to hack the game, I opened the gate to the new world. The sniper in the team has only 6 bullets. As I searched number 6 in Jinshan Youxia, I got a list of results. After that I made a shot which makes the quantity of bullets drop to 5, and I searched number 5 from the previous list of results. I repeated the same move and finally found one promising value that matches the change of number of bullets. I edited the number to 6 and freeze it. A miracle happened. The number of bullets remains at 6 no matter how many times I shot, which enables me to shoot the sniper rifle unlimitedly. This method was widely spreaded among my school after my discovery. As Commandos: Beyond the Call of Duty is a stealth-oriented game and the range of sniper rifle is limited, the game became easier with unlimited bullets but it remains relatively balanced.

The process of spotting the correct value to edit and freeze works as a puzzle game mechanics, by which I was inspired and come up with the hacking concept of VARIABLES.



查找: 17565

地址	单字节	双字节	四字节	浮点数	字
00237ED6	[157]	[17565]	[0053232797]	[0.000000]	[illegible]
009305D8	[157]	[17565]	[0000017565]	[0.000000]	[illegible]
03059F82	[157]	[17565]	[3973596317]	[-2091615...	[illegible]
03075932	[157]	[17565]	[2231714973]	[-0.000000]	[illegible]
0324800B	[157]	[17565]	[1098466461]	[15.579251]	[illegible]
0328602D	[157]	[17565]	[2231387293]	[-0.000000]	[illegible]
032861DB	[157]	[17565]	[2231387293]	[-0.000000]	[illegible]
0328C41C	[157]	[17565]	[0016794781]	[0.000000]	[illegible]
0328C42B	[157]	[17565]	[0016794781]	[0.000000]	[illegible]
0328C435	[157]	[17565]	[3338683549]	[-32836.6...	[illegible]
0328C44C	[157]	[17565]	[3892331677]	[-2422914...	[illegible]
0328C457	[157]	[17565]	[2365605021]	[-0.000000]	[illegible]
0328E51F	[157]	[17565]	[0016794781]	[0.000000]	[illegible]
0328E544	[157]	[17565]	[0016794781]	[0.000000]	[illegible]
0328E591	[157]	[17565]	[2332050589]	[-0.000000]	[illegible]
0328E599	[157]	[17565]	[2332050589]	[-0.000000]	[illegible]
0328E5A2	[157]	[17565]	[2332050589]	[-0.000000]	[illegible]

已经搜索 2 次找到 11848 个结果

Figure 2.10: *JinShan Youxia :* Interface

2.3.3 Cheat Codes

The most basic type of cheat code is one created by the game designers and hidden within the video game itself, that will cause any type of uncommon effect that is not part of the usual game mechanics. [17] Cheat codes are usually activated by typing secret passwords or pressing controller buttons in a certain sequence.

There are basically two types of cheat codes in video games. The first type is cheat codes that modify the difficulty of the game. By empowering the player, weakening the enemies, providing the player with extremely powerful weapon or large amount of XP or in-game currency or allowing the player to skip levels, this types of cheat codes fasten the process of a game and create god-like gaming experience for player. There are several examples of such type of cheat codes. By typing Whosyourdaddy in Warcraft III [5], the army controlled by the player become undefeatable and extremely powerful. By entering "ZELDA" in the name entry screen the player can reach the second quest in The Legend of Zelda [15] and Zelda II [21]: The Adventure of Link without going through all those hours of the first one.

There is another type of cheat codes that blends naturally in the gameplay and doesnt affect the gaming process. One example is seaways cheat code in Grand Theft Auto: Vice City. [1] By entering this cheat code, vehicles in the game can move on the sea.

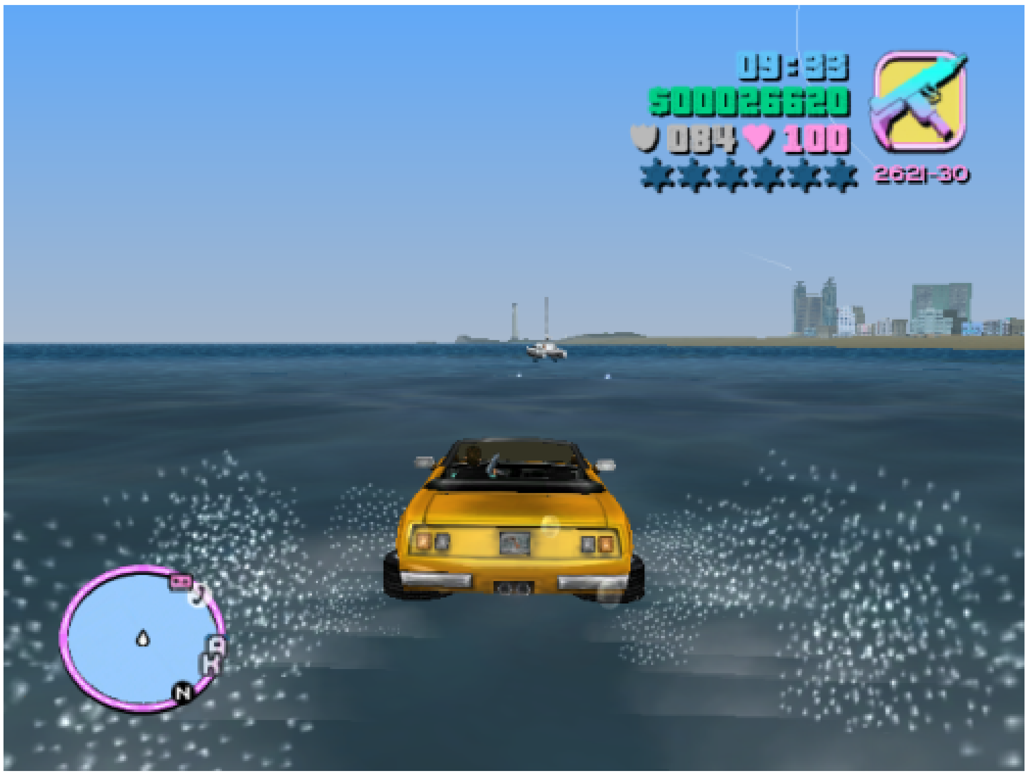


Figure 2.11: *GTA:Vice City :* Cars Move On Water with "Seaways"

Concept of cheat codes in games can be applied in VARIABLES so as to create special gaming experience for player with the hacking mechanics. For example, player can get a weapon or item by entering the serial number. This richen the possibilities of the game, enables the game break the fourth wall and also triggers eager of players to discover and discuss about hidden functions of hacking mechanics in the game.

Chapter 3
Concept Design

3.1 Breaking The Fourth Wall Inversely

3.1.1 Two Layers of Experience

First, I'd like to discuss about the definition and evaluation of breaking the fourth wall.

In traditional media, no matter the technique, to break the fourth wall is normally to break the suspension of disbelief, to remind the audience it is just a film, just a television show, just a performance. The ultimate goal of breaking the fourth wall is blurring the border between the stage and the audience in order to create a surprising and humorous experience at a price of breaking the immersion.

Yet in video games, the inverse will often apply. As mentioned above, many of these so-called fourth wall breaks actually serve to further immerse the player, extending the immersion beyond the screen. There are basically three approaches to break the fourth wall in video games. Firstly, a direct acknowledgement of the player by the game is a clear fourth wall break in the most conventional sense. Therefore a character directly addressing the gamer as player of the video game would be a breakage. Secondly, a display of self-awareness by the product to its own status as game, such as a character's commentary on his position as avatar, as a fictional character in a fictional story. Thirdly, making reference to an artifact, event or person that is obviously outside the fictional world of the game.

All video games are technically role-playing games. Players play a role in video games to accomplish tasks, take part in stories or discover the game world. In this case, the fourth wall refers to the boundary between the role played by the player and the game world where the character is put in. The role here doesn't refer to the character controlled by the player, but the whole set of connection and interaction between the player and the game. In normal fourth wall breaking cases in video games, the connection and interaction between the player and the game is changed or extended. When the fourth wall is broken, the player is not only the character in the game but also the player himself. The player himself become one independent existence in the game and part of the game story. In my word, the goal of breaking the fourth wall in video games is providing two layers of gaming experience to the player. The player are provided with the experience as a role in a game and as himself. If the player is provided with the two layers of experience, the fourth wall is broken.

As the goal and approach of breaking the fourth wall is defined, I'd like to discuss about breaking the fourth wall inversely.

The first approach is referring to game world from player directly. Both traditional media and video games that break the fourth wall recently are breaking the fourth wall from the stage to audience or from the game to player. The approach is that the story is extended to real world. It's like viewing a story from a different perspective and the real world become part of the story. Let's take one step further. The way that video games extends the story to the real world is applying rules in real world and triggering events in the game world based on that. The rules here refers to the rules based on which the real world is run, which we call it law of nature. In the Psycho Mantis example mentioned above, the story of Metal Gear Solid is extended to the real world by reading the saved files in player's PlayStation. The action of reading saved files is actually application of rules in real world.

As the real world is built and runs based on law of nature, what is the "law of nature" of a video game? As we all know, video games are technically softwares which is built and run based on programming logic and variables. The approach of extending the game flow from the game to real world can be considered equivalent to the approach of extending the game flow to the game itself. If the player is provided with the experience of referring to the game itself directly, the fourth wall is broken from the player side.

The other approach of breaking the fourth wall inversely is making the player become strongly aware of the fact that he is the player who is playing the game. In both traditional media and video games, the fourth wall is broken when the character in the story shows that he "realize" the fact that he is one fictional character in a story. In this way, the fourth wall can be broken when the player realize the situation, which is, the fact that he is the player instead of the character or the role he is playing in the game.

3.1.2 Fusion of Activeness and Passiveness

Almost all the video games that Breaks The Fourth Wall mentioned before, The Fourth Wall is broken from the stage and reaching out to the audience. They are either breaking The Fourth Wall by applying meta elements or placing part of the game contents in real world, such as puzzles that players need to find the answer in real world or taking actions physically. Even though it seems that players are proactively taking actions so as to affect the game, The Fourth Wall is not actually broken from the audience. The reason why is that the answer is set in advance and only expected actions can affect the game environment. Id say that the answer to the puzzle is placed on the stage. Its just like the actor on the stage reaching their hands to the audience and clap the hands with them.

Given the interactive nature, what if making the most of this nature of video games and enabling the audience to affect the play on the stage? By blurring the border between experiencing and acting, The Fourth Wall can be broken inversely. Just like an audience watching and directing a movie simultaneously, by enabling the player to edit the game along with playing the game and blending the two types of experience together, the player are able to gain this Inversely Fourth Wall Breaking experience.

3.1.3 Story and Setting

As I mentioned above, the point of breaking the fourth wall in video games is providing the player with two layers of experience, as the character or role he is playing in the game and as player himself. So in this case, the player is provided with the experience of playing two roles from different perspective of the game.

In VARIABLES, player plays two roles simultaneously – the main character taking actions in the game world, and the hacker who hacks the game and assist the main character outside the game environment. To make this setting reasonable, inspired by the concept of digital copies of people which is frequently seen in Black Mirror, I set the background story on a sci-fi stage.

VARIABLES itself is a simulation program called Marga in the story of the game. A genius programmer, hired by the government, developed this program to simulate human beings behavior so as to help scientists to do research on sociological topics. They make digital copies of people from real world and spawn them into Marga. By creating virtual environment or events and observing their behavior, human behavior can be simulated and observed as samples. However, as all the NPCs in the system are digital copies of real human beings, they have a mind with all emotions. Some of them are programmed to witness and experience tragedies like wars, over and over again. So of them realize they are actually digital copies but some are not, while they need to all keep walking on the eternal anantarya-marga.

The programmer, who is also the player, realized its cruel and finally decided to shut down the system secretly. Actually, when he was developing the system, he left a back door in the system. In order to shut down the system, he need one character in the system to complete several tasks, reach the final terminal in the system and type in the shutdown code. The shutdown code is composed of two parts, one fixed string and another random string updates every single minute which can be only seen from the outside world. However, there is no way to communicate with the digital copies in the system from the outside world. As the genius who developed the whole system, only he knows how to trigger the back door and build a bridge across the two worlds. He decided to make a digital copy of himself and spawn it into Marga.

As the main character, also refers to the player, is merely a computer scientist with slight military training, it is fairly difficult to accomplish the task by his digital copy. Whats more, in order to monitor the system, there are guards working for the government from the outside world keeping an eye on the system. There are also digital guards in the system. The digital copy of the main character therefore needs to accomplish the tasks by stealth and the help from the outside world is also necessary.

As the main character outside Marga and the digital copy share the same mind, playing both roles, the computer scientist outside the system and the digital copy simultaneously, is understandable. Hints to some puzzles in the game are set in the real world. Sometimes the player even needs to click to a link or search on Google to find the hint. By making the real world part of the game story, The Fourth Wall is broken.

3.2 Basic Game Mechanics

3.2.1 Hacking and Variable Editing

As I discussed above, the approaches of breaking the fourth wall inversely in this case are allowing the player to refer to the game itself directly and making them strongly be aware of the fact that he is the player of the game, the basic game mechanics that makes this happen is allowing the player to check and edit variables and programming logic of game objects.

Variables and properties of some of the objects near the main character can be seen and partly edited on the terminal. However, the exact function and meaning of the variables and properties are not illustrated. By observing the change pattern and logical thinking, the player is expected to figure out the meaning of the shown variables and properties. By editing the variables and properties, players can trigger some changes in the game. For example, player can teleport to a specific position or drop a car on a group of enemies by editing the axis and execute. By editing the number of ammo player can gain ammo of a weapon. Player can also interact with environmental objects via hacking such as opening a gate or disabling a camera.

I also make the variable spotting and editing mechanics one step further. As stealth game are space-oriented, modifying spatial relations by editing variables is an approach. In game engine or editors, positions of game objects are reflected by axis, which means by editing axis which is usually illustrated as three values referring to XYZ axis, positions of game objects can be changed. In a sandbox stealth game environment, changing positions of game objects can deliver abundant gaming experience. By changing the axis of the main character, player can teleport to a specific position. By changing the axis of some huge objects such as vehicles, player can drop a car on a group of enemies and eliminate them in an unexpected way. Along with rich AI behavior of NPCs in the game, players can discover unlimited numbers of tactics and a sandbox gaming experience can be delivered.

The hacking mechanics also reach out to real world. Players are able to use hint hidden in real world and type in specific values, by which players are able to get secret items or unlock special game mechanics.

3.2.2 Stealth

The core game mechanics of the game is stealth. In VARIABLES, player is extremely disempowered in action and force. Defeating the enemies in combat is nearly impossible in the game and is not encouraged, rewarded or directly contributing to accomplishing tasks. In VARIABLES, the player needs to creep around and avoid being sighted by the enemies.

AS I mentioned before, in Hack n Slash, players are allowed to edit object properties, global variables and even rewrite the code of the game. Its a relatively innovative game mechanics. VARIABLES has very similar game mechanics, such as allowing players to modify object properties. However, the fundamental gameplay system is completely different. Besides the hacking game mechanics, Hack n Slash is a hack and slash game, in which the players goal is to eliminate or neutralize enemies. VARIABLES is a stealth game besides the hacking mechanics. The difference between the two types of basic game mechanics determines how hacking mechanics affects the gameplay.

In VARIABLES, the hacking mechanics doesnt break the balance and destroy the core gaming experience of the game. Unlike hack and slash game which is basically about defeating enemies, stealth games is about traversal and space oriented, in which the player is extremely disempowered. Eliminating enemy is completely optional and depending on the situation. Editing variables of the game cannot bring player absolute advantage in the game. Even though the player can gain some extra power by editing variables, he cannot defeat the reinforcing enemies.

Stealth is relatively compatible with the hacking mechanics. Stealth games are basically puzzle games, in which player are to find solutions to the problems with the provided tools. The hacking mechanics is one of the provided tools, same as weapons, items and environment. Whats more, waiting is one core action in stealth games and stealth game is also about making waiting engaging. The hacking mechanics can enhance the engagement of waiting time under the basic stealth mechanics. By editing variables, the enemies can be eliminated, weakened or enter a specific status for a short time, which also richen the gaming experience of hacking based on the well developed enemy AI. For example, players can set ammo of the gun an enemy is holding to zero. When this enemy sees the player and wants to shoot but realised there is no ammo in his gun, he will be panic and escape. But after a short while, he will pick up another gun and back to patrol. Players can get rich reaction from the game as a result of editing variables, by which a sandbox gaming experience is provided to the player.

Chapter 4
Implementation

4.1 Game Environment

4.1.1 Orthographic Top Down 2D

There is a reason behind setting the game in a top down 2D environment. When I was developing the demo, I did experimental attempts at using different environmental setup and camera angles. There are several types of game environment setup and camera angles and Ill go through and compare these options and discuss about why top down 2D is the most compatible option.

In 3D environment with following rotatable perspective camera, which is widely used in 3D stealth games, players can gain the most immersive and realistic stealth gaming experience. However, in type of environment, it is difficult for players to realize and analyze the exact spatial correlation lying on global axis, which makes matching the axis to the position of game objects fairly difficult. As players can rotate both the facing direction of the character and camera angle, it makes it even more difficult to realize the exact axis of game objects. In 2.5D environment with following fixed perspective camera, the same problem will be found.

In 2.5D environment with top down orthographic camera will face another dilemma. X and Y axis are usually perceived as horizontal and vertical directions in game world. Game objects in orthographic camera are not distorted but because of geometrical reasons the X and Y axis will be distorted if the camera is tilted, which makes the player move fast along one axis than another.

There are two ways to fix the axis distortion while using orthographic camera. One approach is rotating the camera around Y axis for 45 degrees. This approach can avoid the axis distortion on the ground but it makes it difficult for players to understand the relation between position of a game object and its axis because the perspective angle does not match the horizontal and vertical pattern. Another approach is make the camera overhead top down. This approach can perfectly

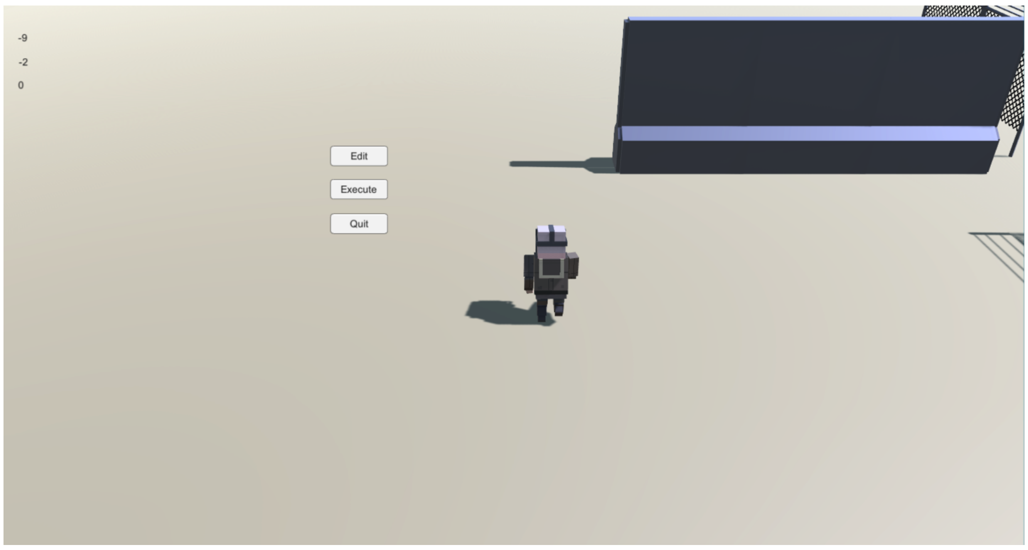


Figure 4.1: *Perspective Camera*

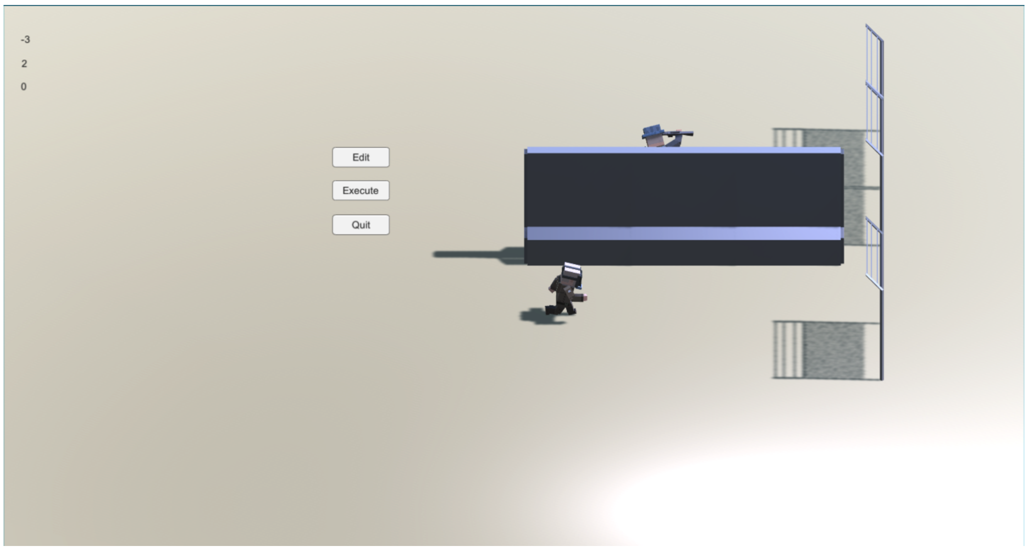


Figure 4.2: *Orthographic Camera without Rotation*

solve the previous problem but bring new issues. Firstly, the characters and game objects are not well depicted with this perspective. Secondly, the Z axis which refers to height in the game will not be well perceived by the player.

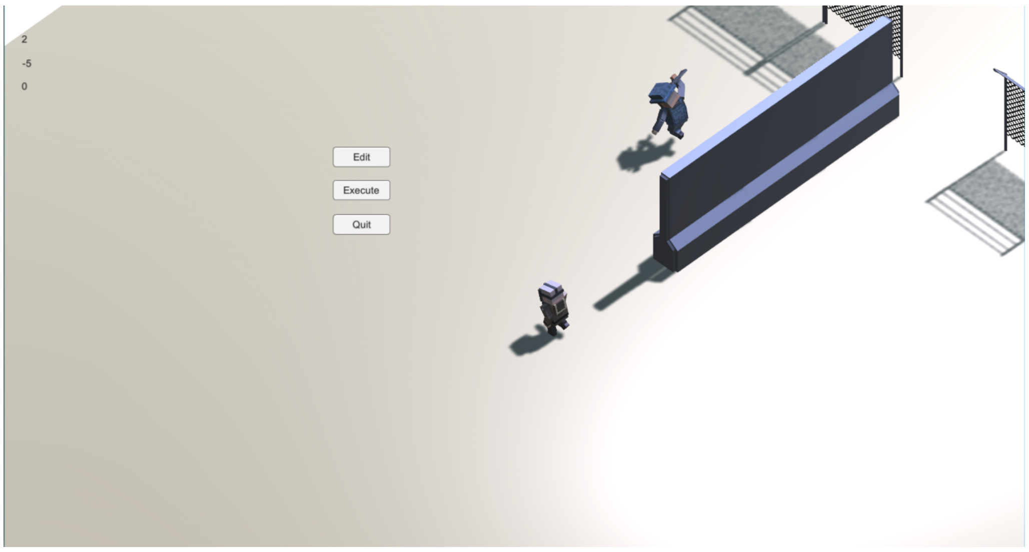


Figure 4.3: *Orthographic Camera with Rotation*

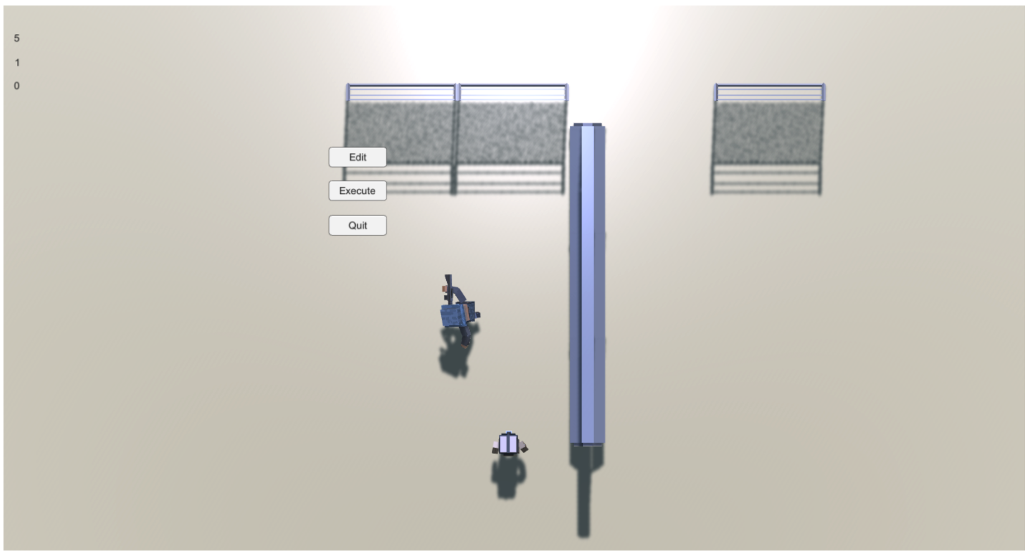


Figure 4.4: *Orthographic Overhead Top Down*

I decide to set the game in grid-based top down 2D environment which is widely used in retro games. By setting the game environment this way, all the problems mentioned above will be solved. The only problem await to be solved is the illustration of height in the game. Inspired by retro games with the same game environment, I use the distance between shadow and sprite of a gameobject to depict height in 2D environment. Shadows of game objects are allowed dynamically scaled according to the fake height of the object.

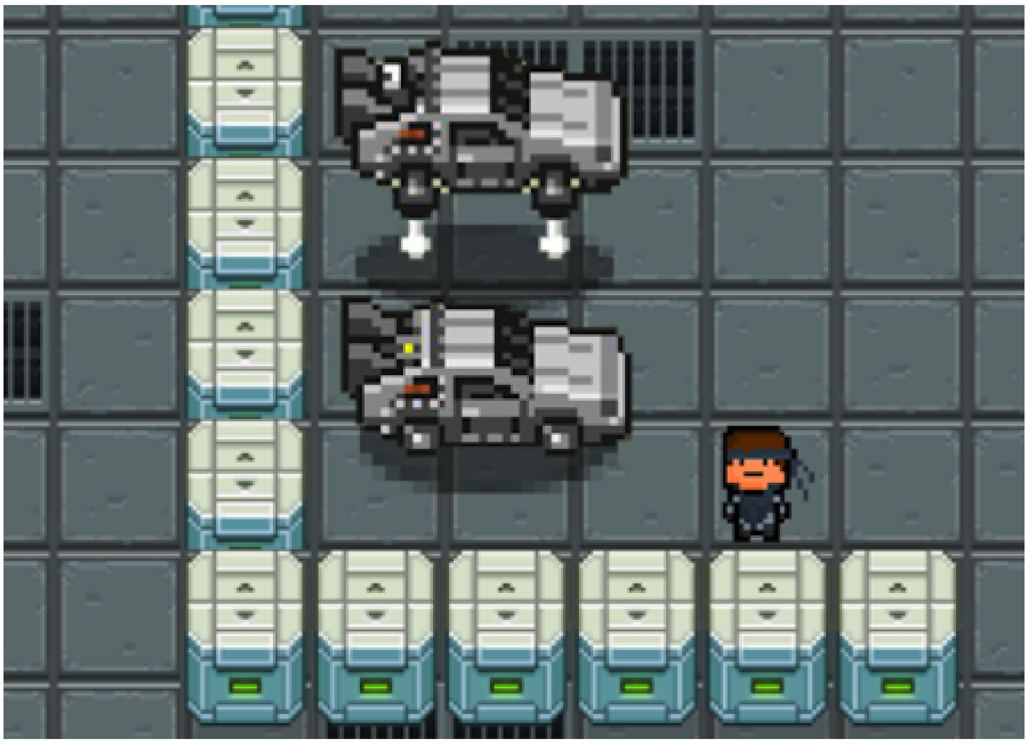


Figure 4.5: *Using Shadow to illustrate Hight in 2D*

4.1.2 Scriptable Tile Based

In order to improve the efficiency of level development, pathfinding and environmental interaction, the game is scriptable tile based. As scriptable tile is strongly bond to grid in the game, it makes addressing axis of game objects in the game much easier for players and avoids unexpected collisions. Scriptable tile based maps also enhance the sandbox gaming experience by enabling the players to spawn new objects or changing the size of game objects. As player is enabled to change position of game objects, scriptable tile system enables the game to predict and simulate the after effect of editing so as to avoid unexpected bugs.

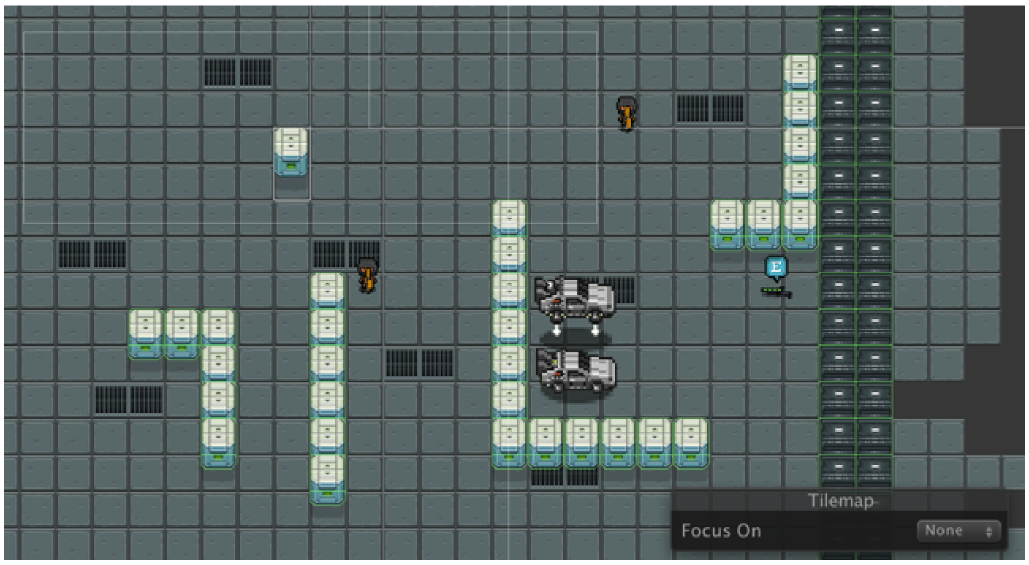


Figure 4.6: *Tile-based Level*

4.2 Enemy AI

4.2.1 Discussion

AI plays an essential role in most engaging video games, especially video games that with human-like characters. Unlike traditional action games, in which firstly there should be some output from the game and player takes actions according to it, game mechanic of stealth game works as a pull and push mechanics, which means, the game environment doesnt change or give output unless the player takes some actions, and the game environment change according to the players actions. The game pushes while the player pulls. While task goals of stealth games are not about eliminating enemies and game mechanic of stealth game is strongly bond to spatial relations, a space-based enemy AI system is demanded. Overall, stealth games vary in what player actions the AI will perceive and react to [8], with more recent games offering a wider range of enemy reactions. Because combat is not encouraged in stealth game, players may even get lower review if eliminate enemies in games, its not a simple kill or be killed game mechanics. Various and dynamic behavior modes of enemies is therefore highly required. To make it clear, enemies in stealth games need to react to abundant situations and also takes various actions accordingly, instead of merely chasing and attacking the player. Even in primitive stealth games, such as Metal Gear, player can distract enemies by making some sound, such as knocking at the wall.

For my part, creating realistic and abundant enemy behavior and gaming experience is about two parts. One part is to mimic the sensory system and decision making of human beings, and the other part is to mimic the real physical world. Enemy AI development in most of the games mainly focus on the former while the latter also plays an equally import role in crafting realistic AI behavior and immersive gaming experience.

As distraction is one useful tool for players in stealth games, I provide such tools to players in VARIABLES. In VARIABLES, players can easily distract enemies with a large range of methods. Players can cause a noise by shooting at the wall, dropping items and cause events, each brings different consequence in the game. Because in VARIABLES, enemies have both searching and panic status, players therefore can distract enemies by either making them leave their patrolling path or make them scared. There are lots of ways to make enemies leave their patrolling path. Making noise is one of them. As in VARIABLES, players are able to drop editable objects via editing axis of the objects, when the objects are dropped, a sound will be caused. Enemies who heard the sound will go to the source of the sound and check, which can create opportunity for players to traverse. Another way to make enemies to leave their patrolling path is causing unusual events. In VARIABLES, status of some of the objects in the environment can be changed via editing variables of the object, such as opening a gate, disabling a camera, triggering the fire alarm and so on. Enemies will change their behavior status and take actions accordingly. Another way to distract enemies is to make them panic. Witnessing extremely unusually events will make enemies panic and lose ability to behave for a specific period of time. Dropping, teleporting enemies or dropping huge objects in front of enemies will cause the enemies to enter panic status.

4.2.2 Enemy Behavior Modes

Due to the limited developing time, I am not using behavior tree to develop the enemy AI. Instead, I set several behavior modes and prioritize them.

1)Patrol

When an enemy is in patrol mode, he will patrol following a preset specific path at a preset Patrol Speed or stay at a point and look around.

2)Search

When notified by an event, a Question icon will appear above the head the enemy, and after a Reaction Time, the enemy will head to the position of the event at a preset Search Speed. During this process, if notified another event, the enemy will head to the position of the new event.

3)Hunt
When player enters the range of view, Hunt Mode of the enemy will be triggered. A Spotted icon will appear above the head the enemy, and after a Reaction Time, the enemy will start rushing to the player at a preset Chase Speed. Notice Event will be continuously generated at the position of enemy and Last Sighted Position (the position of player last sighted) will be continuously updated.

4)Chase
Enemy rush to Last Sighted Position when lost sight of player. Notice Event will be continuously generated at the position of enemy. When the enemy arrives at Last Sighted Position, he will look around. If player is not spotted, the enemy switch to search mode and arrive at a random position nearby on the map.

5)Attack When player is in sight, within the weapon range and weapon is available, the enemy will aims at the player and attack the player.

6)Panic
If something horrible happened to or witnessed by the enemy, the enemy will switch to Panic Mode. When enemy is in Panic Mode, he will drop the weapon and run at a preset Panic Speed. Sound Event will be continuously generated at the position of the enemy.

7)Faint
All functions and actions are disabled and Search Event is triggered if its sighted by other enemies. The enemy with Faint behavior status will switch back to Patrol mode after a certain time or be awaken by another enemy.

8)Dead
All functions and actions are disabled. Search Event is triggered if its sighted by other enemies.

4.2.3 Detecting System

Sensory System

The term ”senses” in game development is a useful metaphor for understanding, designing, and discussing that part of the AI that gathers information about items of interest in the simulated environment of the game. Non-player characters visually presented as humans, animals, or creatures with eyes and ears in a realistic three-dimensional space lend themselves well to the metaphor. [8]

2D Spatial Event Triggers

In 3D stealth games, thanks to built-in physical simulation system in many mainstream game engines, triggering enemys behavior based on events in the simulated physical environment is not difficult. In that case, developers just need to focus on enemys sensory system. [3]Since VARIABLES is a top down 2D game, crafting its own physical environment is a tricky part. In order to tackle this problem, I designed and developed my own AI behavior system which combines sensory system set on the enemies and 2D spatial event triggers which can change behavior status of enemies within a range, unlike enemy AI development of most indie stealth game. For example, in many stealth games, enemies detect players or actions within specific sensory ranges, like a range of sight and a range of hearing. If the player enters the field of view of the enemy or there is a sound within the hearing range, the status of enemy behavior will change. The whole system is built on the enemy object, which means every single enemy detects the surroundings every single frame, which takes relatively much calculation. Additionally, such detection system is not efficient enough when there are more than 2 enemies or there are more complicated interactions between them. In VARIABLES, space is one core element of the game system and physical parameters are also editable, using spatial event triggers is more flexible and efficient. While sensory system is aiming at making the enemies more human-like, spatial event triggers aims at making the game environment closer to real physical world in a top down 2D environment.

My approach of building 2D spatial event triggers is relatively straightforward. Take sound event as an example. If someone shoot a gun, a sound is emitted. In VARIABLES, along with the generation of the sound, a sound event is also generated. A sound event is basically an empty game object generated at the position of the event sending message to all active enemies within a circle around it. It works the same way as an explosion delivering damages to objects around it. Whether the status of the enemies who received the message changes or not can be based on other conditions. For example, distance between the event and the enemy, quantity or material of walls between the event and the enemy or the current status of the enemy. The calculation of this process takes only few frames. Additionally, since many parameters in VARIABLES are editable, this approach is much more flexible and adaptable to the game mechanics.

1)Sound Event
Any event that cause a sound will generate a sound event. The sound event will generate a circle with a specific radius and the size of radius depends on the type of sound and refers to the volume and intensity of the sound. Larger sound has a larger radius while smaller sounds has a smaller radius. Sound event will detect all the active enemies within the circle and the behavior of all the enemies within the circle are ready to be changed. But whether the enemies behavior is going to change depends on other conditions. In VARIABLES, I set a Hearing Reaction Threshold to enemies, while different types of enemies have different Hearing Reaction Threshold. The behavior mode of an enemy changes only when the Received Sound Intensity is higher than his Hearing Reaction Threshold. The Received Sound Intensity is based on several conditions. Firstly, Received Sound Intensity is based on the distance between the position of sound event and position of the enemy. The further the sound event is from the enemy, the lower Received Sound Intensity is. Secondly, Received Sound Intensity is based on obstacles between the sound event and the enemy. Quantity and material of obstacles will affect Received Sound Intensity. For example, if there is a metal wall between the enemy and the sound event, the Received Sound Intensity will dramatically decrease.

2)Notice Event
Enemies in Attack, Hunt or Chase mode and other game objects(such as cameras, infrared sensors,etc) that spot the player will generate Notice Event. The notice event will generate a circle with a specific radius and the size of radius depends on the type of enemy or other game objects. For example, simple guards will call all the enemies around him to chase the player, and cameras at some highly guarded positions will send message to almost all the enemies on the map to hunt down the player. All the active enemies within the range will enter Chase Mode and rush to the Last Sighted Position where the player was last sighted.

3)Safe Signal
As for a top down 2D game with 2D physics involved and using A* pathfinding, one problem needed to be tackled is the collision of objects. While enemies in Variables have complex interactions, some tricks are necessary. One example is when multiple enemies are heading to the same position, they will probably collide with each other and even get stuck. While I was developing the game, these occasions happens quite oftenly. The solution is either rewriting the code of A* pathfinding or creating a system to avoid this situation. Because the value of

artificial intelligence in video games is partly simulating human behaviors. When multiple enemies are heading to the same search point, there is no need to let every single involved enemy to check the same point one after another. This will likely cause collision of enemies and is a lack of realistic experience. My solution here is to let the first enemy who searched the suspicious point to emit a Safe Signal to the enemies around searching the same position. All the enemies who received the Safe Signal will go to other places to search until searching time passed. This solution can efficiently tackle the collision issue and make the behavior of enemies much more realistic and convincing.

4)Chased Signal
Same as safe Safe Signal, Chased Signal is emitted by an enemy who reached the Last Sighted Position and to make enemies around to go to another random position.

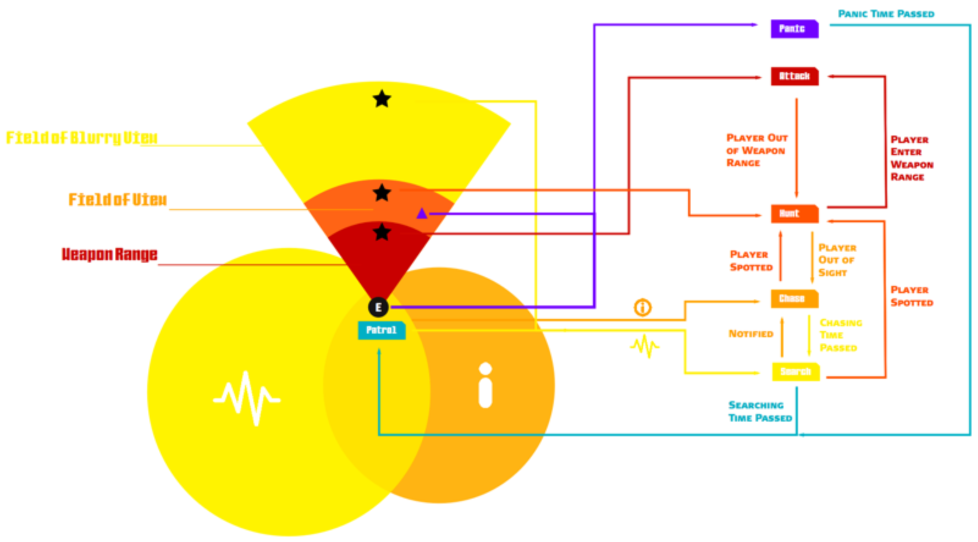


Figure 4.7: *AI Behavior Structure*

Behavior Mode Switching

1)Priority Integer
As every single enemy can only have one current behavior mode, prioritizing

behavior modes is one approach to switch behavior modes of enemies. In this case, I use Priority Integer to refer to priority of each behavior mode. The larger the Priority Integer is, the behavior is more prior. Since there are basically 6 behavior modes of enemies, and also according to real situation, I set Priority Integers for each behavior mode as follows: Patrol - 1 / Search - 2 / Chase - 3 / Hunt - 4 / Attack -5 / Panic - 6 / Faint - 7 / Death - 8 By comparing the Priority Integer, the enemy will switch to the most prior behavior at the moment. For example, if the player now patrolling, which means the current Priority Integer is 1, he will switch to Search Mode if he is triggered by a Sound Event. On the contrary, if the enemy is at Attack Mode with Priority Integer 5, Sound Event will not trigger his Search Mode because the Priority Integer is smaller, while he can switch to Panic Mode.

Basically there are two ways to switch behavior modes of enemies via priority integers. To make switching behavior modes more flexible and stable which satisfies most of the situations, I set two types of priority integers: New Priority Integer and Current Priority Integer. New Priority Integer refers to the priority level of a new outer trigger which attempts to switch the enemys current behavior. Current Priority Integer refers to the priority integer of the current behavior status of the enemy. When there is an outer trigger attempts to change the enemys behavior status, the New Priority Integer is compared to the Current Priority Integer, based on the comparing result the enemy will switch to a behavior status with higher Priority Integer or remain the current behavior mode.

2)Switching Conditions

Besides switching behavior status based on Priority Integer, behavior status can be changed by switching conditions. In Variables, switching conditions mainly refers to the current situation of the enemy or things happened to the enemy itself. For Instance, if the player is sighted, the enemy will switch to Hunt Mode. Or if the enemy is attacked, he will enter Search Mode. If the enemy is running out of ammo and in low health, he will switch to Panic Mode.

4.3 Level Design

4.3.1 Stealth Levels in Top Down 2D

As I mentioned before, Stealth game is about puzzles and making the player able to learn about the puzzles is the crucial part. In VARIABLES, in order to make the facing direction of enemies clear enough to the player so that the player can figure out the normal behavior pattern of the enemies, animation of characters also clearly shows which direction the characters are facing to.

4.3.2 Stealth in Open Environment

VARIABLES faces the same dilemma as Metal Gear Solid V. As in VARIABLES, players are allowed to change axis of objects in the game, I need to provide enough space for the player to teleport the objects, which results in that, route of the player shouldnt be blocked or restricted. Thats also why I dont place architectures in the game but only wall and leave a relative large and open space. This results in the fact that I need to pack all the puzzles together and make each level a big sandbox. Especially I need to limit the power of the axis editing mechanics. To make this clear, for example, in sequential stealth games, there is no way for players to reach the final point at the beginning because the route from the beginning to the end is fixed, while usually reaching the final position of a level is the prerequisite of completing the level. But in open world stealth games in which levels are blended in the open world environment, there is no preset route for the player. In VARIABLES, as players can teleport to every available position by editing axis, the methods of level design applied in sequential stealth games are not applicable.

In every mission, the target will be kept unknown and the player needs to find it out. In one level of VARIABLES, the goal of the task is preventing one important person from being assassinated. But the player doesnt know where the assassin is located and when the assassin is taking action. In another level, the player needs to steal one document in the enemy base but the position of the document is left to be found by the player.

4.4 Variable Editing

Some game objects in the game can be modified by editing variables or programming logic. When the mouse hovers over editable game objects, the colour of the game object will change. Player can right click on editable game objects to show terminal. The terminal interface will show some basic information of this game object in the program and editable variables. Player can edit variables in a similar way as editing files in C based terminal. After editing variables, player

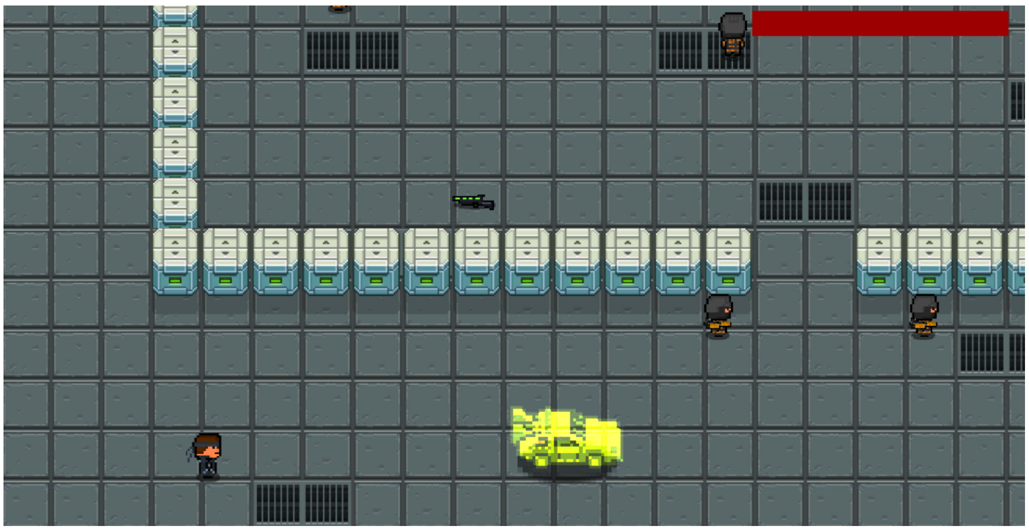
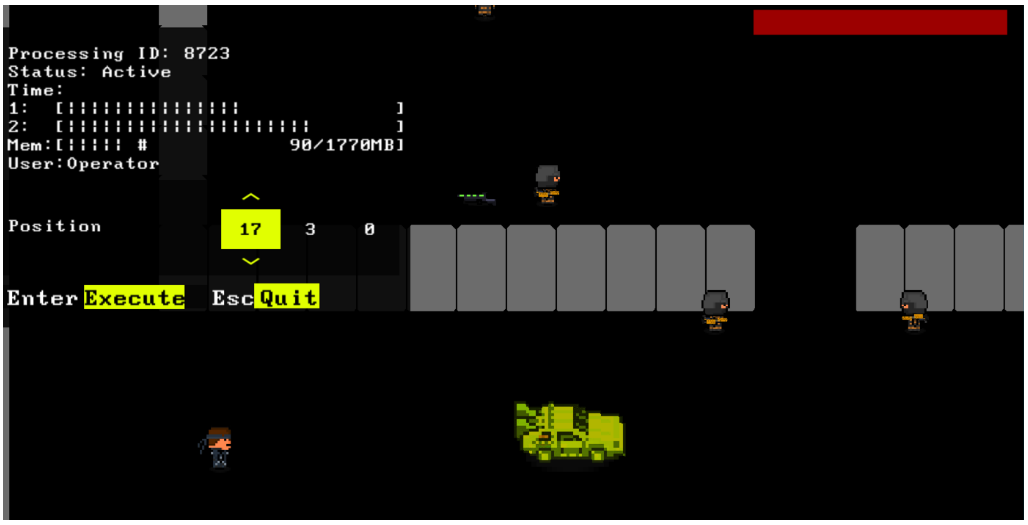


Figure 4.8: *Editable Objects Changes Colour When Hovered*

can execute the editing by pressing ENTER or RETURN.

When the editing is executed, the main camera will move to the edited game object. After the all the events caused by this execution is played, the main camera will move back to follow the main character.



```
Processing ID: 8723
Status: Active
Time:
1:  [¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦               ]
2:  [¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦         ]
Mem:[¦¦¦¦¦ #              90/1770MB]
User:Operator

Position        17   3   0

Enter Execute   Esc Quit
```

Figure 4.9: *Press E to Edit Variables*

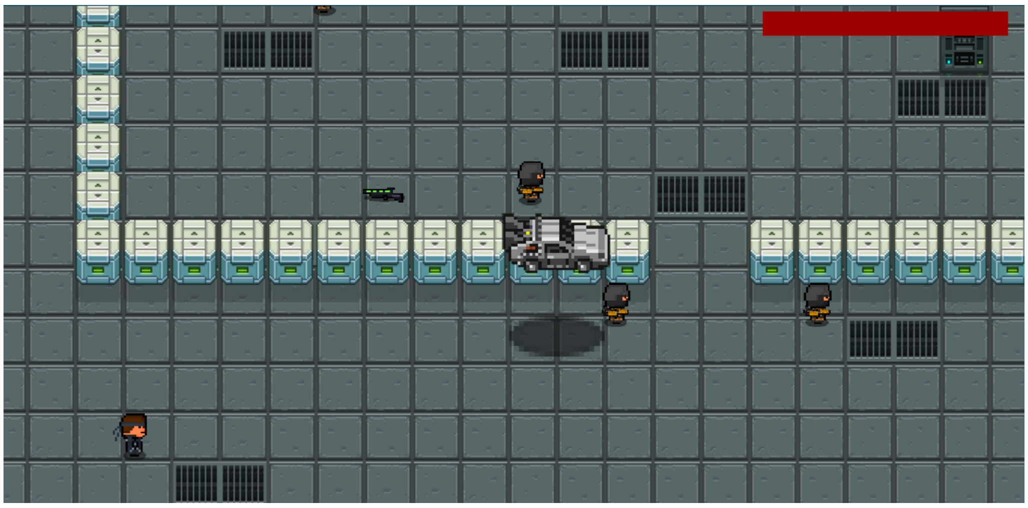
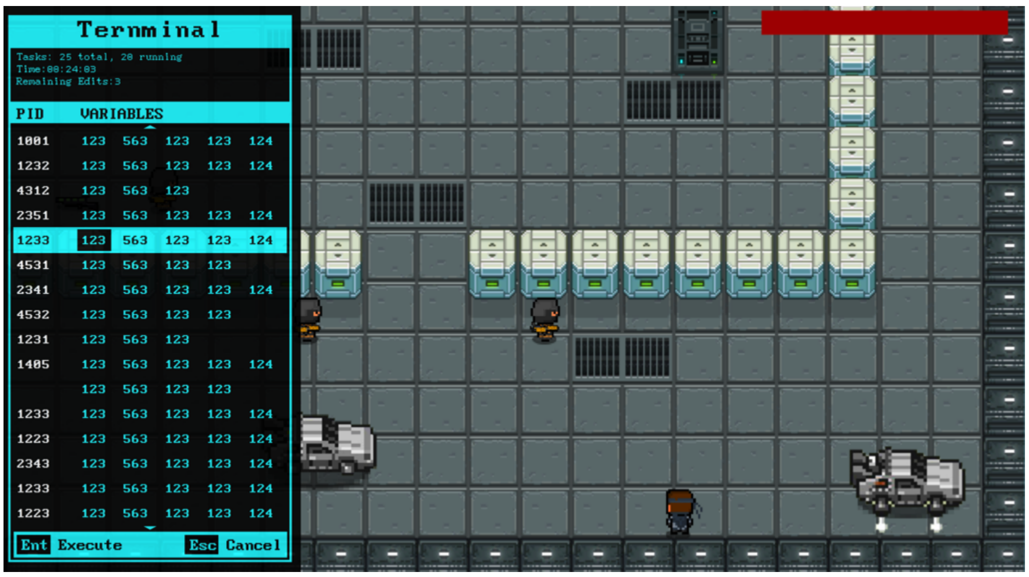


Figure 4.10: *Press ENTER to Execute Edits*

Chapter 5
Iteration and Evaluation

5.1 Version 1.0

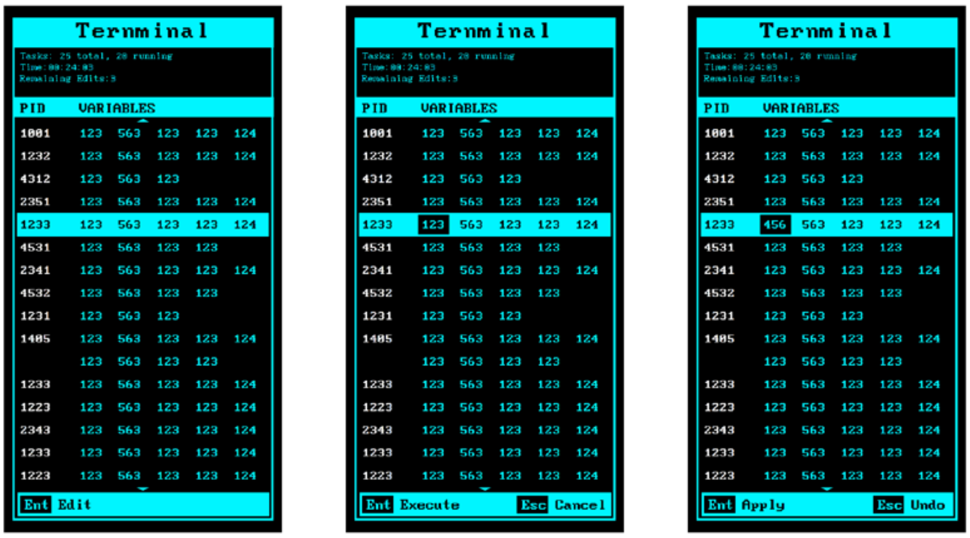
The first version of Variables is basically adding the function of memory editing software to the game. All the information of editable game objects are listed in the terminal UI. Player needs to figure out the correlation between the items on the list and game objects by axis.



```
Ternminal
Tasks: 25 total, 20 running
Time:00:24:03
Remaining Edits:3
PID     VARIABLES
1001    123  563  123  123  124
1232    123  563  123  123  124
4312    123  563  123
2351    123  563  123  123  124
1233    123  563  123  123  124
4531    123  563  123  123
2341    123  563  123  123  124
4532    123  563  123  123
1231    123  563  123
1405    123  563  123  123  124
        123  563  123  123
1233    123  563  123  123  124
1223    123  563  123  123  124
2343    123  563  123  123  124
1233    123  563  123  123  124
1223    123  563  123  123  124
Ent Execute              Esc Cancel
```

Figure 5.1: *Interface of Terminal*

6 Participants tried this version and all of them claimed that it is too difficult to play. There are overwhelming numbers changing all the time which makes them really dizzy and nearly impossible to spot the item on the list, not to mention editing the variables. Even me, the one who developed this game, find it extremely



```
Ternminal
Tasks: 25 total, 20 running
Time:00:24:03
Remaining Edits:3
PID     VARIABLES
1001    123  563  123  123  124
1232    123  563  123  123  124
4312    123  563  123
2351    123  563  123  123  124
1233    123  563  123  123  124
4531    123  563  123  123
2341    123  563  123  123  124
4532    123  563  123  123
1231    123  563  123
1405    123  563  123  123  124
        123  563  123  123
1233    123  563  123  123  124
1223    123  563  123  123  124
2343    123  563  123  123  124
1233    123  563  123  123  124
1223    123  563  123  123  124
Ent Edit
```

```
Ternminal
Tasks: 25 total, 20 running
Time:00:24:03
Remaining Edits:3
PID     VARIABLES
1001    123  563  123  123  124
1232    123  563  123  123  124
4312    123  563  123
2351    123  563  123  123  124
1233    123  563  123  123  124
4531    123  563  123  123
2341    123  563  123  123  124
4532    123  563  123  123
1231    123  563  123
1405    123  563  123  123  124
        123  563  123  123
1233    123  563  123  123  124
1223    123  563  123  123  124
2343    123  563  123  123  124
1233    123  563  123  123  124
1223    123  563  123  123  124
Ent Execute              Esc Cancel
```

```
Ternminal
Tasks: 25 total, 20 running
Time:00:24:03
Remaining Edits:3
PID     VARIABLES
1001    123  563  123  123  124
1232    123  563  123  123  124
4312    123  563  123
2351    123  563  123  123  124
1233    456  563  123  123  124
4531    123  563  123  123
2341    123  563  123  123  124
4532    123  563  123  123
1231    123  563  123
1405    123  563  123  123  124
        123  563  123  123
1233    123  563  123  123  124
1223    123  563  123  123  124
2343    123  563  123  123  124
1233    123  563  123  123  124
1223    123  563  123  123  124
Ent Apply                  Esc Undo
```

Figure 5.2: *Terminal UI Design of Version 1.0*

difficult to spot the right item on the list to edit.

Based on the user feedback, I need to modify some basic game mechanics, especially the variable interface and editing part. There are two potential options to improve the experience. One is showing variables and properties of objects only within a certain range from the main character. By doing this, there are less information filled in the list on the interface, but at the same time, the difficulty is lowered. The second option is directly referring the object to its showing variables and properties.

5.2 Version 2.0

I modified the game mechanics and interaction significantly based on the feedback of Version 1.0. In Version 2.0, instead of listing all the items on terminal interface, the terminal interface only shows the information and editable variables of one selected game object. To spot and select an editable game object, player needs to hover the mouse on game objects. If a game object is editable, the colour of the game object will change when it's hovered. The player can check and edit the variables of this game object by right clicking on the object.

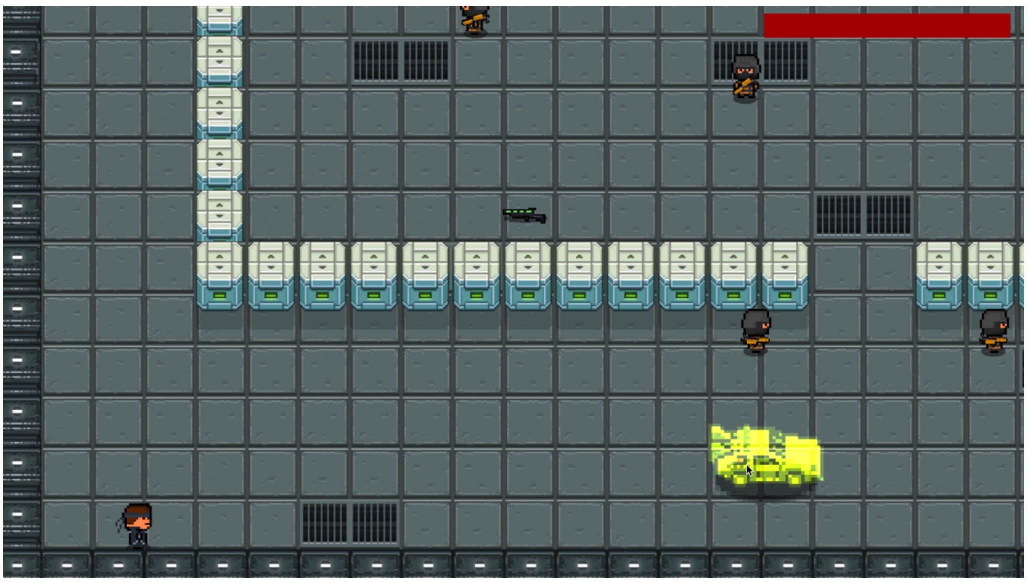


Figure 5.3: *Editable Objects Changes Colour When Hovered*

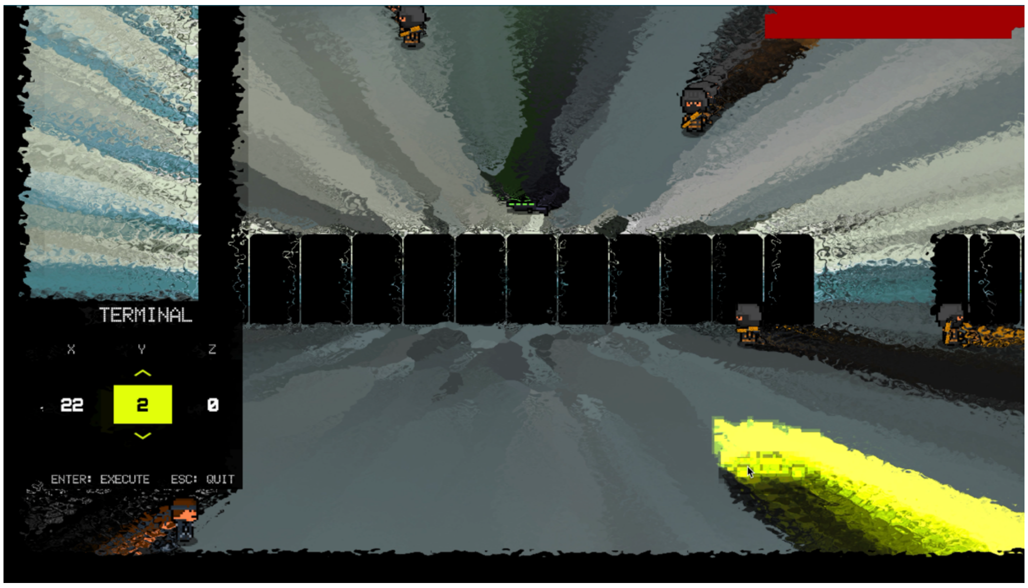
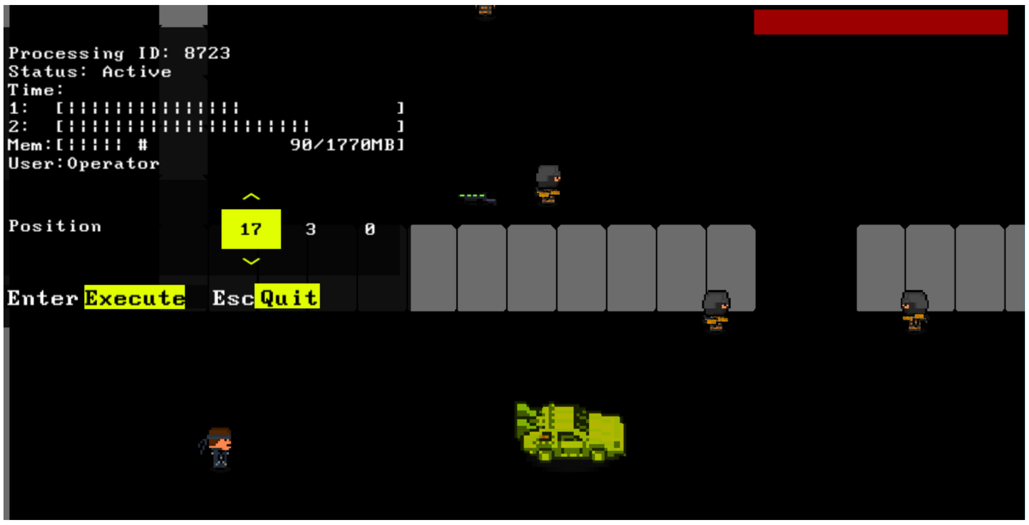


Figure 5.4: *Editable Objects Changes Colour When Hovered*

5.3 Version 2.1

In order to enhance the experience of hacking, I modified the Terminal UI based on the game mechanics and interaction of Version 2.0. I added more elements and information of processing in terminal, such as processing ID, User, Memory Load and time to the UI so as to make it more terminal-like.



```
Processing ID: 8723
Status: Active
Time:
1:  [||||||||||||||||                ]
2:  [|||||||||||||||||||||            ]
Mem:[|||||| #            90/1770MB]
User:Operator

Position       17   3   0

Enter Execute  Esc Quit
```

Figure 5.5: *Terminal UI in Version 2.1*

5.4 Evaluation of Version 2.1

5.4.1 User Test Setup

User test of Variables 2.1 takes 15 minutes for each participant. The user test is composed of three parts. The first part is playing the game without using the "Hacking" mechanics. The second part is trying to clear the level within 3 edits. The last part is filling up a questionnaire.

The questionnaire aims at verifying three main points of this project. The first point is whether the player has the experience of breaking the fourth wall. The second point is verifying whether this gaming experience new. The last point is whether this game is fun.

The questionnaire is as follows:
1.what degree would you consider yourself a gamer?
2.Have you had similar gaming experience with "Hacking" before?
3.Do you consider "Hacking" an ability of the character or yourself?
4.WHO do you think was "Hacking" the game objects?
5.Do you think you are the character or you are helping the character when you were "Hacking"?
6.I am somebody outside the game when I "Hack"?
7.I become strongly aware of the fact that I am the player of the game instead of the character in the game when I "Hack"?
8.How fun do you consider this game?
9.Would you buy this game for 5 dollars when it's released?
10.Would you like to continue playing the game after this experiment?

5.4.2 Discussion

9 volunteers participated in the user test. Different levels of gamers, ranging from people who hardly play video games to hardcore gamers, are relatively equally distributed in the group.

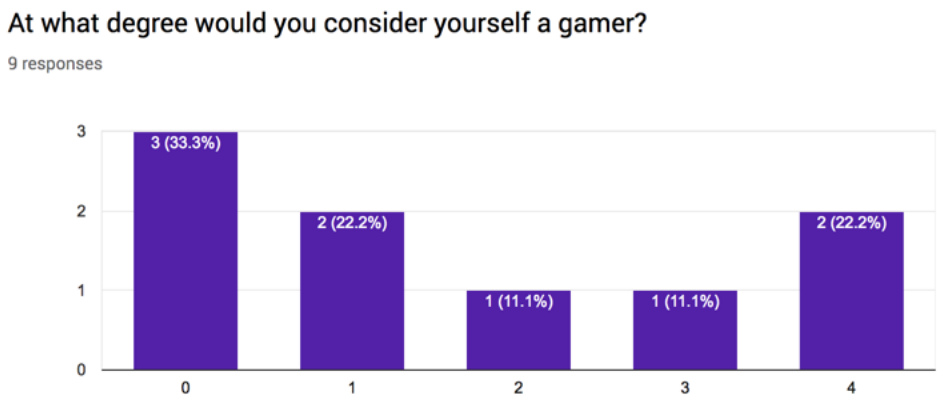


Figure 5.6: *Percentage of participants with different levels as gamer*

Fourth Wall Breaking

As I mentioned in Concept Design, breaking the fourth wall refers to providing two layers of experience, players should have the experience as the role in the game and themselves as player. Also, they need to be strongly aware of the fact that they are the player of the game instead of the role they are playing.

The result pointed out that the fourth wall is broken by the "Hacking" mechanics of the game. Firstly, most of participants consider "Hacking" an ability of themselves, as a player, instead of the character in the game. And they think it was themselves who was "Hacking" the game. Secondly, a significant percentage of participants become strongly aware of the fact that they are the player of the game instead of the character controlled by them. Thirdly, a overwhelming percentage of the participants claimed that they considered somebody outside the game while "Hacking".

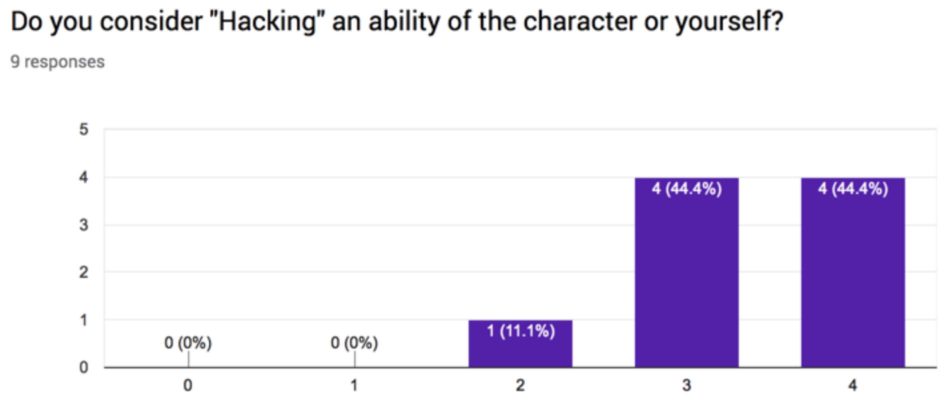


Figure 5.7: *88 percent of participants consider "Hacking" an ability of themselves at a high level*

Player Enjoyment

Firstly, 66.7 percent of participants claimed that they have never had similar gaming experience with "Hacking" and 22.2 percent picked maybe, which indicates that the "Hacking" mechanics is fairly new. 55.6 percent of participants picked the highest review and the rest picked the second highest review of the game, which means they are satisfied by this new gaming experience. 100 percent of

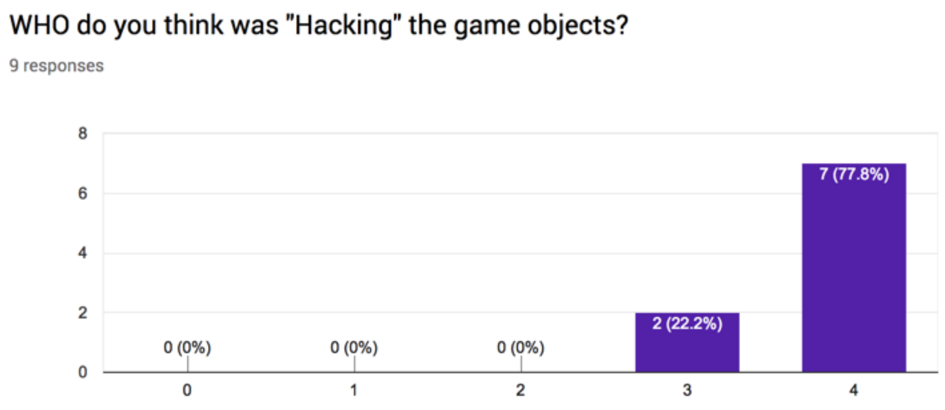


Figure 5.8: *Almost all participants think it's themselves instead of the character who was "Hacking" the game*

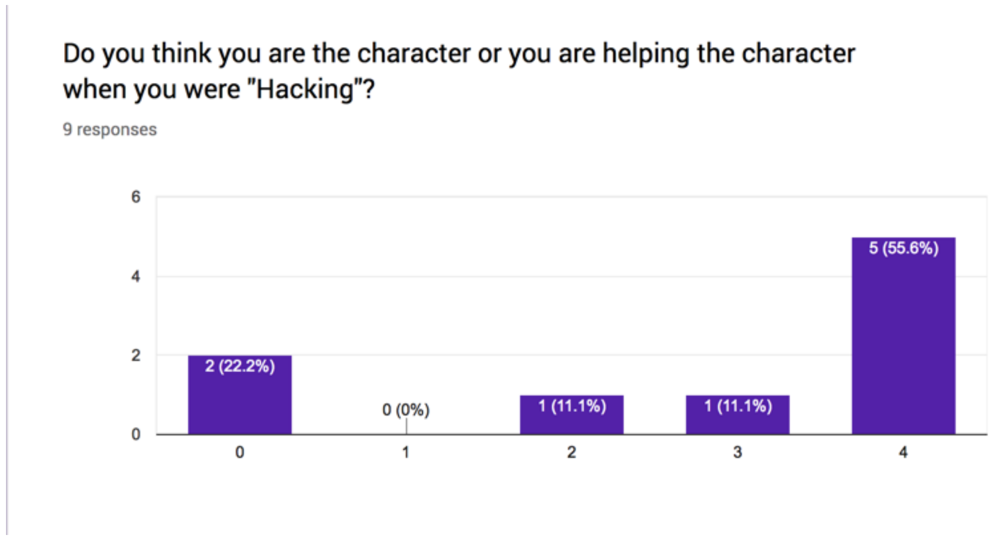


Figure 5.9: *Relatively more participants think they were helping the character instead of playing as the character while "Hacking"*

participants would like to continue playing the game even after the experiment can also clearly indicate player enjoyment of this game. 66.7 percent of participants

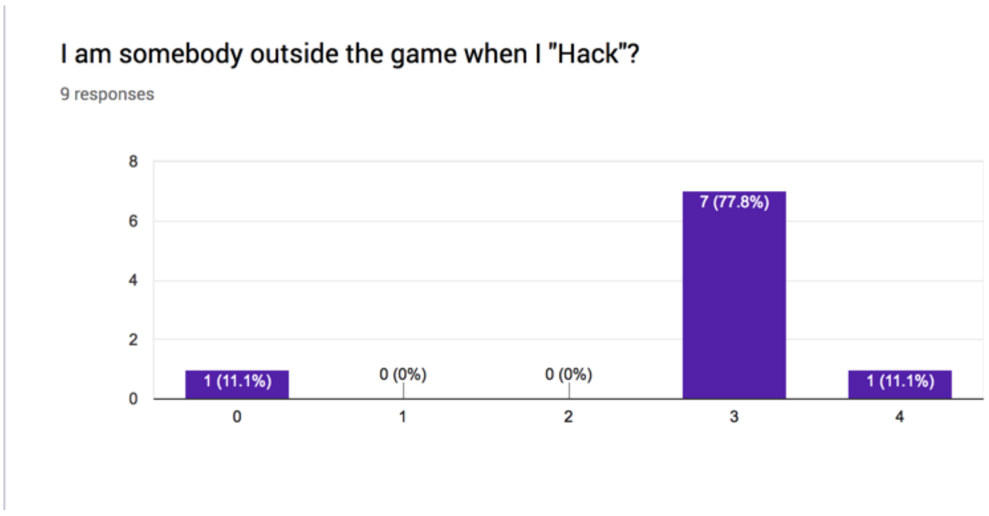


Figure 5.10: *Most of the participants think they were outside the game at a certain level while "Hacking"*

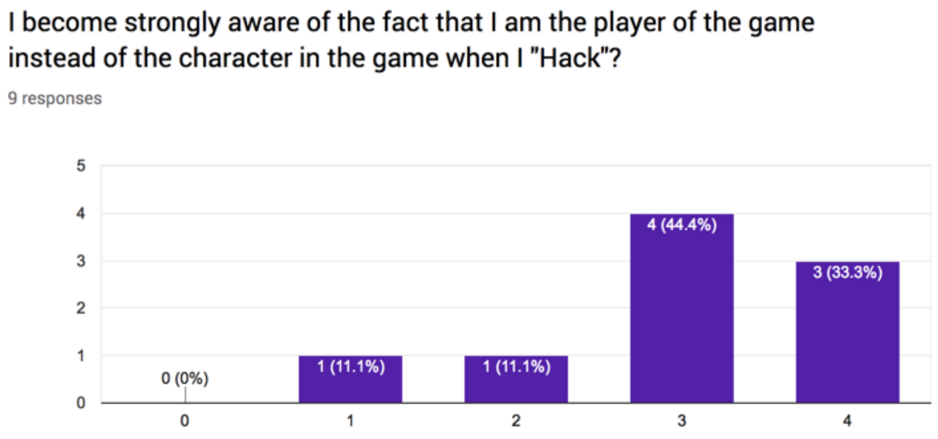


Figure 5.11: *88 percent of participants become strongly aware of the fact that they were the player while "Hacking"*

are surely going to buy this game for 5 dollars is also one strong evidence showing that this game is funny and satisfying.

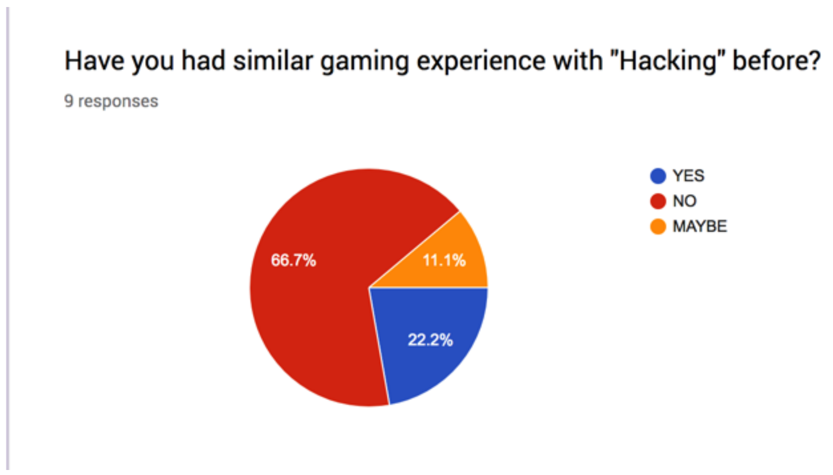


Figure 5.12: *88 percent of participants become strongly aware of the fact that they were the player while "Hacking"*

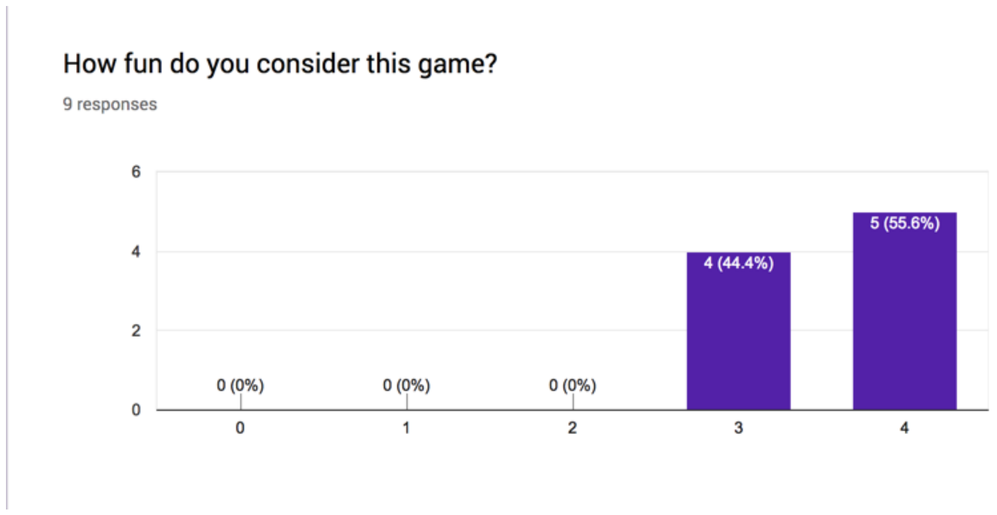


Figure 5.13: *88 percent of participants become strongly aware of the fact that they were the player while "Hacking"*

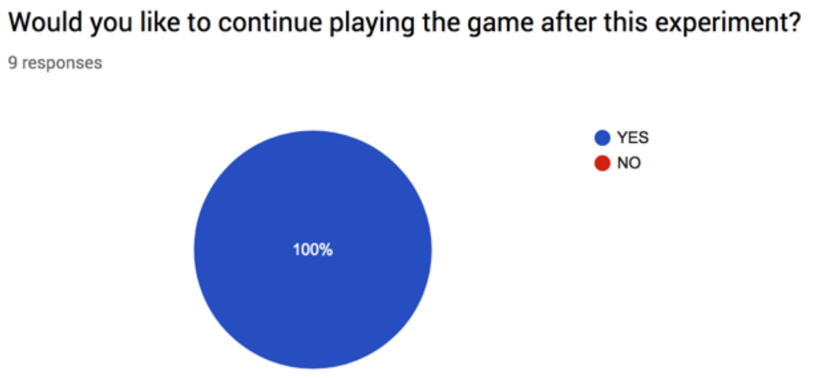


Figure 5.14: *88 percent of participants become strongly aware of the fact that they were the player while "Hacking"*

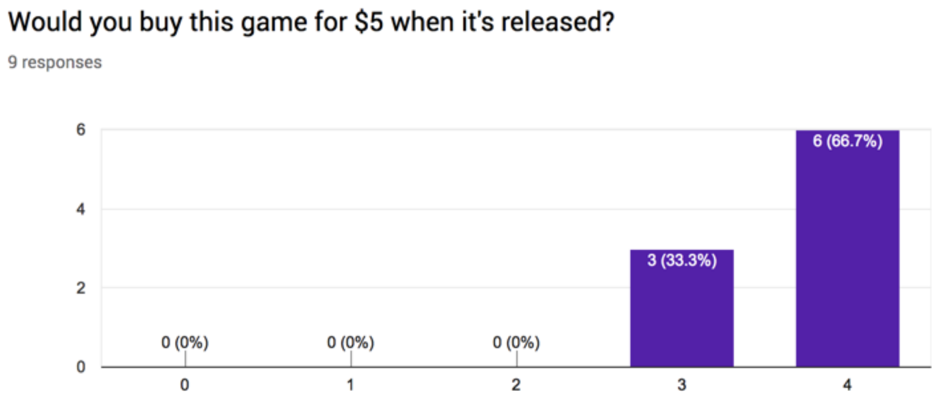


Figure 5.15: *88 percent of participants become strongly aware of the fact that they were the player while "Hacking"*

Chapter 6
Conclusion

6.1 Validation

This approach of breaking the fourth wall inversely in this project is proved to be feasible to a relatively large degree. The concept of breaking the fourth wall inversely is well implemented with the setting of player playing two roles, character in the game and the player himself simultaneously and the ”Hacking” mechanics. The gaming experience is new and also satisfying.

However, almost 90 percent of participants claimed that the game is too difficult. In order to provide the flow experience to the player, the difficulty of this game should be tuned.

6.2 Limitation

So far player can only edit axis of game objects so as to change the position of the game objects. Even though it already express the concept of breaking the fourth wall inversely in this project, it is just the first step and far from completion. Although changing the position of game objects by editing the axis is outside the basic game mechanics and game story setting, it is just one small part of the concept of referring to the game itself directly. In the original concept, player should be able to experience fully access to the game itself and have the freedom to hack and edit the game by modifying programming logic and variables, so as to craft a more sandbox-like gaming experience.

6.3 Future Work

As an ultimate goal, I plan to finish developing this game, release it on Steam and share this special gaming experience to the gamers all over the world. Since the project is so far a demo, there are a significant amount of points await to be improved. Among those points, three of them are reasonable to be prioritized.

6.3.1 More Editable Variables

Besides axis of game objects, there are plenty of properties that can be reflected by variables. Quantity, Object ID, the Character/item you control, size, status, physics parameters, BGM Soundtrack and even times of editing can be reflected by variables and be changed by editing variables. By providing more editable variables in the game, player can have a more profound, realistic and free gaming experience, which make the game more sandbox-like and leave a large space for player to explore new possibilities.

6.3.2 UI Optimization

UI design of terminal in the game plays a crucial role as providing the realistic hacking experience to the player, by which the fourth wall is more likely to be broken from the player side. The UI design of Terminal should be able to guide the player, make them feel like really hacking the game and also fit in the story setting.

6.3.3 Efficient Level Generating System

As this game is more like a sandbox gaming experience based than linear story driven, an efficient level generating system is in demand. Application of procedurally generated contents can be one approach of developing this project.

References

[1] Auto, G. T. Vice city (2002). *Rock Star Games*.

[2] Bell, E. *Theories of performance.* Sage, 2008.

[3] Consalvo, M. *Cheating: Gaining advantage in videogames.* Mit Press, 2009.

[4] Conway, S. A circular wall? reformulating the fourth wall for videogames. *Journal of Gaming & Virtual Worlds 2*, 2 (2010), 145–155.

[5] Entertainment, B. *Warcraft III: Reign of Chaos.* Blizzard Entertainment, 2002.

[6] Entertainment, K. Mark of the ninja, 2012.

[7] Farkas, B. G. *Tenchu 3: Wrath of Heaven (TM) Official Strategy Guide.* BradyGAMES, 2003.

[8] Hjorth, L., Milne, E., Gibbs, M., and Pisan, Y. *IE2007: Proceedings of the 4th Australasian Conference on Interactive Entertainment.* Yusuf Pisan, 2007.

[9] Knight, M., et al. *Commandos: Beyond the Call of Duty: Prima's Official Strategy Guide.* Prima Communications, Inc., 1999.

[10] Kojima, H. *Metal gear solid.* Orbit, 2008.

[11] Kojima, H. Solid game design: Making the impossible possible. In *Keynote at Game Developers Conference (GDC) 2009 presented at California, San Francisco, USA* (2009).

[12] McMillen, E., and Himsl, F. The binding of isaac. *Digital Game* (2011).

[13] Montreal, U. Assassins creed. *Rennes: Ubisoft* (2007).

[14] Muncy, J. The best new video games are all about video games. *Wired* (2016).

[15] Nintendo. *The Legend of Zelda.* 2008.

[16] Roskowski, S. G., and Perlman, S. G. Video game enhancer with intergral modem and smart card interface, Apr. 29 1997. US Patent 5,624,316.

[17] Sezen, T. I., and Isikoglu, D. From ozans to god-modes. In *International Conference, April*, vol. 27 (2007), 29.

[18] Storm, I. Thief: Deadly shadows. *Eidos Interactive* (2004).

[19] Studios, L. G. Thief: The dark project.

[20] Szczepaniak, J., and Yamamoto, Y. *The Untold History of Japanese Game Developers.* SMG Szczepaniak, 2014.

[21] Zelda, I. The adventure of link. *An Instruction Booklet from the Nintendo Entertainment System* (1988).