DELIVERY ENTREPRENEUR

An Honors Project Manuscript

Presented by

Yueying Liu

Completion Date:
April 2018

Approved By:

____
Timothy Richards, College Of Information And Computer Science

____
Shane Mecklenburger, College of Humanities and Fine Arts

ABSTRACT

Title: **Delivery Entrepreneur**
Author: **Yueying Liu**
Thesis/Project Type: **Independent Honors Project**
Approved By: **Timothy Richards, Computer Science**
Approved By: **Shane Mecklenburger, Art**

This is a Commonwealth Honors Interdisciplinary Project which integrates knowledge from Computer Science and Art. The goals of the project are to create an engaging user experience and convey social messages about labor and thinking through a Virtual Reality game to be played on a mobile platform using Google Cardboard. In particular, these goals are achieved through the implementation of a game that has the user take the role of a delivery entrepreneur to survive in a highly competitive world. A delivery entrepreneur is someone who invents new ways for food delivery. Only through thinking and hard work can the user make progress in the game. The game is developed using Unity, a game engine framework, and the C# programming language and Maya as the Art creation tool. Rounds of playtests were conducted to get feedback on how well these goals were achieved. During the development process, the thoughts and efforts in achieving these goals and the differences between developing mobile games versus desktop applications are noted for analysis and future references. The game turns out to be successful on the technical level. The delivery of social messages still needs improvements. The experience of developing a mobile VR application can be applied for future application developments.

1. Introduction

This is a Virtual Reality (VR) game project that is run on mobile phones using Google Cardboard. The game is set in a future time when humans need to compete with artificial intelligence for jobs. In the game, the player collects and sells food as a delivery person, who used to be an accounting supervisor but has lost his/her job. The game proceeds through multiple levels and each level is limited to 60 seconds. At each level the player is given a target amount of money to be earned. The player must collect and sell food to achieve this amount to enter the next level, otherwise the game is over.

The overall goal of this project can be divided into three components:

The first being on the technical level involving this researcher: I had to learn to develop a 3D mobile game on the Android platform using the Unity game engine. For this I used Google Cardboard to create a VR experience with a mobile phone. Research was completed on the following: how to use the program Unity, the understanding of the mobile platform, enhancing real-time rendering, and game design. The goal of this part of the project was to have the game built and run on an Android phone with Google Cardboard.

The second goal focused on the artistic component. This project used artistic techniques, including camera framing, 3D modeling and lighting to let players put themselves in the position of a delivery person and experience his/her struggle in the VR environment. The goal of this section was to allow the players to experience a delivery entrepreneur’s life, feel the pressure from life and reflect on matters like artificial intelligence, identity politics and laborer.

The final goal focused on two aspects: one, relates to the programmer. My goal has been to compare and contrast the writing of a desktop application and a game application. The second goal relates to the user experience of the game. This includes my reflections on how to improve the experience of this game using the players' verbal and non-verbal feedback.

This project was developed using Unity as the game engine. Through this engine the game can be easily built and run on different platforms, including Windows and iOS desktops, Android and Apple mobile phones, and also VR devices like PlayStation4 VR headset[1]. But since the goal of this project was to focus on the mobile platform which can reach the widest audience to better connect social messages and the perspective of a laborer in an environment that more closely resembles reality, only Android mobile phones were used for play-testing.

At first a runnable prototype of the game was developed for preliminary playtest. The subsequent refinement of the game was done by taking into account the user feedback from the playtest. Throughout this process, the similarities and differences between developing traditional desktop programs and developing mobile VR games were noted for future analysis.

The game was successfully designed, implemented, and tested. Users' feedback were seriously considered and used to make adjustments to the game. During the user-testing process, users read the story of the delivery entrepreneur first, but once the game starts, they pay full attention to survive in the game and spared no attention to reflect on the life of the delivery entrepreneur. They can only reflect after the game is over. This indicates the success of the entertainment value of the game. However, the social message delivery of the game needs improvements, probably by presenting the backstory in a more succinct way, like in a short video or anime.

The differences between developing desktop applications and mobile VR games are the following:

1. the structures of the two applications are very different. In developing desktop applications, programmers tend to put all the data in one place; whereas due to the design of game engines, it is more intuitive and easier to save data as attributes in separate objects.

2. for desktop applications, there are more than enough ways for a user to interact with the machine via various input/output devices; whereas in mobile VR games, due to the short distance between the screen and the eyes, the ways of interaction are restricted to the gaze and/or single taps. The design of interactions in VR games must be very carefully thought through to make full use of the available ways of user input.

3. cooperation is harder for programmers using game engines because the game objects are stored in numbers which do not make sense for non-developers of the game engine. When two game programmers both modify the same game object of the same copy of a game, it is almost impossible to keep both modifications. Only one modification can be preserved and the other one must be made again.

The rest of this thesis is organized as follows: Section 2 discusses the significance of this project. Section 3 covers the technical, artistic and social backgrounds of the game. Section 4 provides an overview of the methodology I have used, including choice of tools, choice of project design, artistic choice and playtest method. Section 5 presents the results of this work.

2. Significance

This socially conscious mobile VR game has significance in both technological and social aspects.

Improving VR technology, including hardware, software and tricks to improve user experience through games is common. Actually gaming industry is expected to drive the development of VR just as it drove the development of graphic displays.

A variety of hardwares and softwares are developed to support VR these days. Some may become the future of VR industry, whereas some may never be shown to consumers. Of all these, mobile VR is kind of "known" to survive this technological revolution[2]. However, since mobile phones are designed to do many things, its user experience on VR is innately not as good as specialized devices. Many technological and design tricks such as dynamically reducing and increasing the level of detail of rendered objects depending on their distance from the camera, must be deployed to so as to ensure superior user experience, including a good frame rate which is critical to have smooth animations in the game[3].

However, the fast growing world of technological advancement is a threat to many traditional industries. Since the Industrial Revolution in the nineteenth century in England, new industries have emerged and replaced some traditional industries. Many human jobs have also been replaced by machines. The number of traditional industries threatened by technologies is growing everyday[5]. I want people to feel the threat of technology and be alerted to the world where technology takes over our lives. Playing a game is always an escape from the reality. Many people play games for relaxation. I would like to put my message out that "people need to be constantly thinking so as to not be outed by technological games in the game of life" on my VR game. The VR game also gets increasingly hard which simulates the hardships in our lives. The causes of the increasing hardship in the game are present in the backstory of the game: because of artificial intelligence and competition from other people, life gets harder for every body. This reflects what can happen in society if artificial intelligence takes control of so many tasks performed by human beings right now. It would be most gratifying for me if some of the players one day realize this message, feel the pressures of life, reflect upon artificial intelligence and think about the lives of laborers who would be on the frontline of losing their jobs to artificial intelligence.

Although there are many socially conscious games in many forms, they are less available in the form of VR. I believe that since VR provides better immersion into the game environment, VR games help the users to feel more and think deeper. My project is a novel trial on integrating social consciousness to VR games.

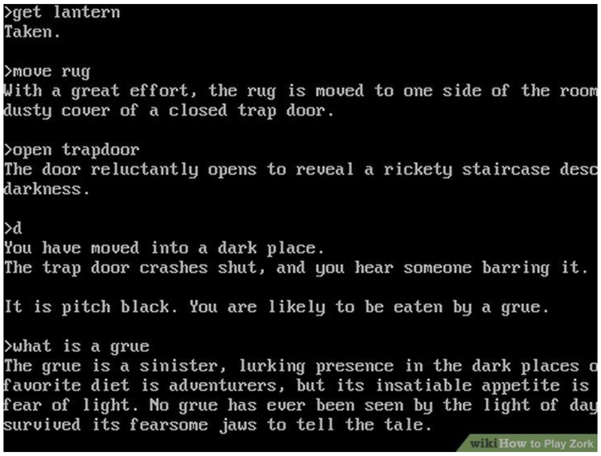
In my project, I explored a variety of techniques and designs to improve user experience and convey social messages. There are abundance of writings available on improving the experience of VR users[6]. It is important that we filter the research that is useful for understanding social consciousness and VR experience. My goal in this study has been to develop applications that will make mobile VR experience better while efficiently deliver my social messages. The process of developing a more satisfying VR experience can be replicated to many different fields, like construction prototypes, military simulation to improve the experience of VR in these different fields[7] . The experiences in delivering social messages through VR games could be applied for future socially conscious VR game development to expand this market. Hopefully with the accumulation of knowledge on how to make good socially conscious VR games, we would get popular VR games that discuss a variety of social issues in the future.

3. Background

3.1 Technological Background

It is a trend for some time now that computer programs are becoming more accessible to the common persons. Although earlier on, computer programs were executed using the console where people typed in a command which was usually hard to understand and prone to errors, hit the enter key and then got the results in texts. Consoles are still frequently used by programmers. This kind of user interface is the most inaccessible for humans as users must memorize the computer commands in order to communicate with the computer. Gradually, with the introduction of Graphical User Interface(GUI) since 1963[8], desktop programs, whether running on a web browser or on its own as an executable program, transformed from text-only to using graphics and icons. GUI makes visualization easier and makes softwares more accessible to people of all backgrounds including gender, race, language, religion and those with physical and cognitive impairments.

As increasingly more people have access to some type of mobile devices which are getting cheaper, mobile applications have become popular, and their market is still expanding rapidly. Mobile applications increase the human accessibility by its increased portability and the ability to sense the direction and orientation of the device with accelerometers which measures accelerations from gravity, vibrations and movements[9]. Nevertheless these programs all require the user to view a two-dimensional(2D) interface in this three-dimensional(3D) world. Although with years of using the existing GUI, we are familiar with this form of interaction, the most intuitive interaction for humans should be with a virtual 3D environment. This would have the same dimension as the real word. VR, whose foundation was laid down in 1838[8] but developed slowly through the years due to hardware constraints, is one of the hottest research areas in computer science[4]. It aims to allow the users to view the contents of the program in 3D directly. This is achieved by simulating how the 3D world is projected onto human retinas and interpreted into a 3D setting by the brain. For this to happen the user puts on a VR headset that displays two slightly different images which belong to the same scene but are viewed from angles that simulate what the two eyes see: the left and right eye sees separately. The two images are called stereoscopic images[10]. The human brain interprets these two images into a 3D setting.



```
>get lantern
Taken.

>move rug
With a great effort, the rug is moved to one side of the room
dusty cover of a closed trap door.

>open trapdoor
The door reluctantly opens to reveal a rickety staircase desc
darkness.

>d
You have moved into a dark place.
The trap door crashes shut, and you hear someone barring it.

It is pitch black. You are likely to be eaten by a grue.

>what is a grue
The grue is a sinister, lurking presence in the dark places o
favorite diet is adventurers, but its insatiable appetite is
fear of light. No grue has ever been seen by the light of day
survived its fearsome jaws to tell the tale.
```

Image of *Zork*, a console game

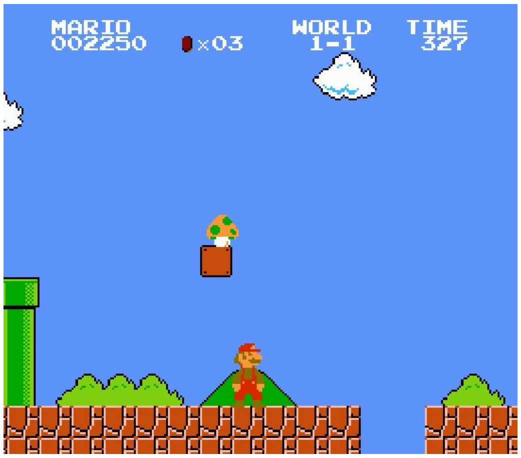


Image of *Mario Brothers*, a 2D GUI game

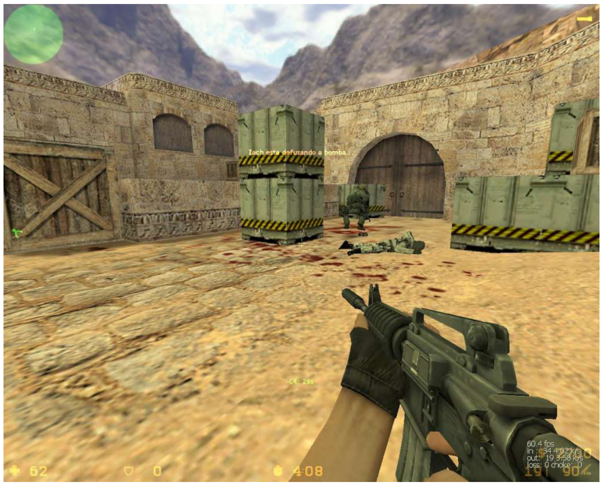
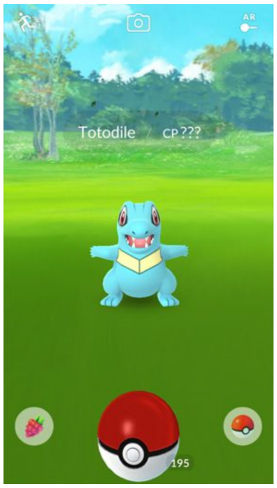
	

Image of *Counter-Strike*, a 3D game displayed in 2D	Image of *Pokemon Go*, an augmented reality game

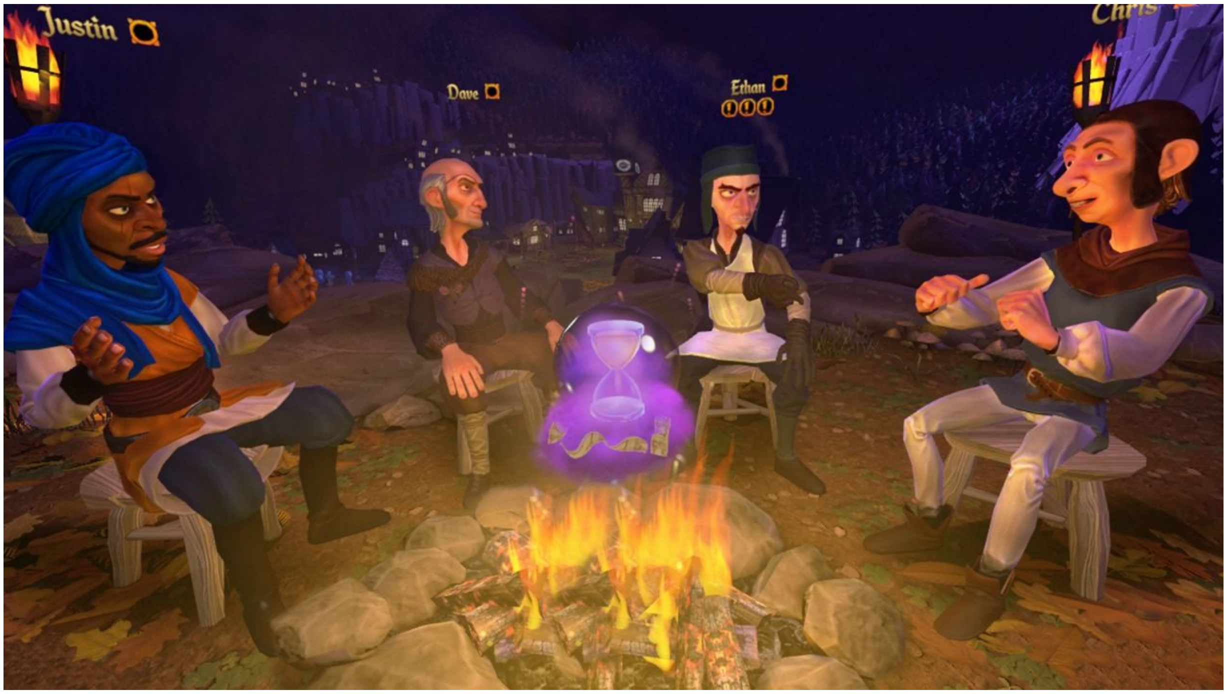


Image of *Werewolves Within,* a VR game

Figure 1: Evolution of Game Interfaces

Computer games, a subcategory of computer programs, follow this trend of making the games more accessible to the public. Early console games like *Zork*[11] have the same human-computer interactions as any other console programs. For example, in Zork the user explores a virtual world using commands such as “go west”, “climb ladder”, etc. These movements are then followed by the display through the text, of the following of these commands. There are no images in the game *Zork* - only texts. This is typically referred to as a “text adventure”. With GUI and a mouse which can locate any point on the screen, desktop games have a large variety, ranging from 2D games like *Mario Brothers*[12] which is an abstract the world on a 2D plane, to 3D model games like *Counter-Strike*[13] which projects a 3D world onto a 2D screen. In *Mario Brothers*, players mainly use the direction arrows to control the movement of the character, Mario. His movements are directly shown on the 2D screen. The camera is fixed at an angel and all the objects on the screen are flat. In *Counter-Strike*, players still use keyboard to control the movement of the character. But the player moves the camera instead. The 3D environment and everything going on in the environment is projected to the 2D screen. Players still see a 2D display, but the objects are rendered through the perspective of the moving camera. Thus players can get a fake sense of a 3D world. With time, as technology developed, the multitouch screen and accelerometers of mobile devices give the players more control of the games. *Pokemon Go*[14], for example, is an augmented reality game which adds components to the real world to the game. What the game displays is a cartoon version of the physical location that the player of the game is at. In addition to moving around to explore the world and encounter pokemons, the player can control the game by touching the icons on the screen. Although this game does not require a keyboard, it has a versatile control since players can touch many things on the screen. Moving around in the physical world to trigger change of location and touching icons which shows its function to trigger events are more similar to what we can do in reality which amounts to more than the use of a keyboard as a control. Therefore, augmented reality games such as *Pokemon Go* become more accessible to humans. Again, all these aforementioned games are limited by the hardware used to display the game, that even if the game creates a 3D world, the player could only see the 2D projection of the world. VR games, on the other hand, allow the players to immerse into a virtual 3D world that is meant to reflect the real world they are part of. For example the players of the game *Werewolves Within*[15] would find themselves gathered around a campfire in the medieval town of Gallowston. Whether the player is standing-up, or turning their heads in a different direction or lying down, would directly change what they see in the game. The players need to move as per the real world to trigger changes in the virtual world.

Figure 2 summarizes the spectrum of human accessibility for each of these different "virtual" worlds that have been discussed:

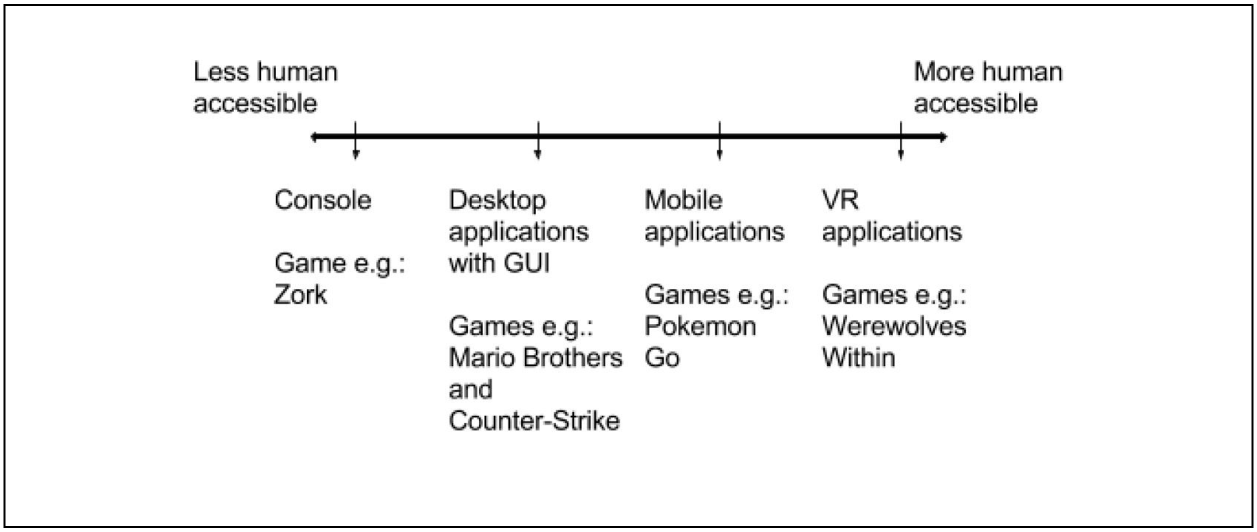


Figure 2: Spectrum of Computer Programs and Example of Games

Admittedly, applications using new technologies are only available after the existing technology has been developed enough that it can be brought to use in new, innovative ways. We cannot deny that the design of applications are moving towards the more human accessible end. One evidence is that after GUI technologies are mature, console applications are rarely seen by the wide audience. Now mobile applications are gaining market share and people spend more than 50% on mobile applications among all applications[16]. This trend is well-noted.

One of the most avant-garde types of computer programs today are VR and mobile, due to their novelty and wide application. Actually the term VR was coined only in 1987[4], and the first mobile application was released in early 90's[4]. Being relatively new areas of research, there are many things to learn and explore in VR and mobile.

Currently the most prominent challenge for VR is ensuring fast real-time rendering and high resolution display, which contradict each other. In other words, when a person enters a virtual world, the person expects to see the world as clear and smooth as the real world. To see things clearly requires all objects in the virtual world to have high resolution. To see things smoothly, the hardware needs to finish calculating the next frame to be displayed for both eyes within 1/60 seconds which is the frame rate for most of the games. The higher resolution of the objects, the slower the calculation. All VR projects face this challenge and must find the right balance to provide the most realistic experience for the user.

Coincidentally, mobile development research is leaning towards VR and something slightly different: Augmented Reality(AR) which is displaying on the screen the reality with computer-generated data superimposed. For example, having a pair of rabbit ears during video chat (Figure 3), and *Pokemon Go* as mentioned before. AR has the same calculation-resolution dilemma as VR. On mobile platforms, especially, the dilemma is intensified because mobile devices cannot have as good hardware as bulkier devices.

Technologies are developed to solve this problem. Various game engines, such as Unity, Castle[28], are developed with functions to optimise the displays on respective target devices. Game developers just need to turn on more computationally expensive functions to achieve better visual effect for better hardware platforms, while disabling these functions for slower hardware platforms. For example, good hardware can render objects from further away while lower quality hardware should only render objects close to the camera.

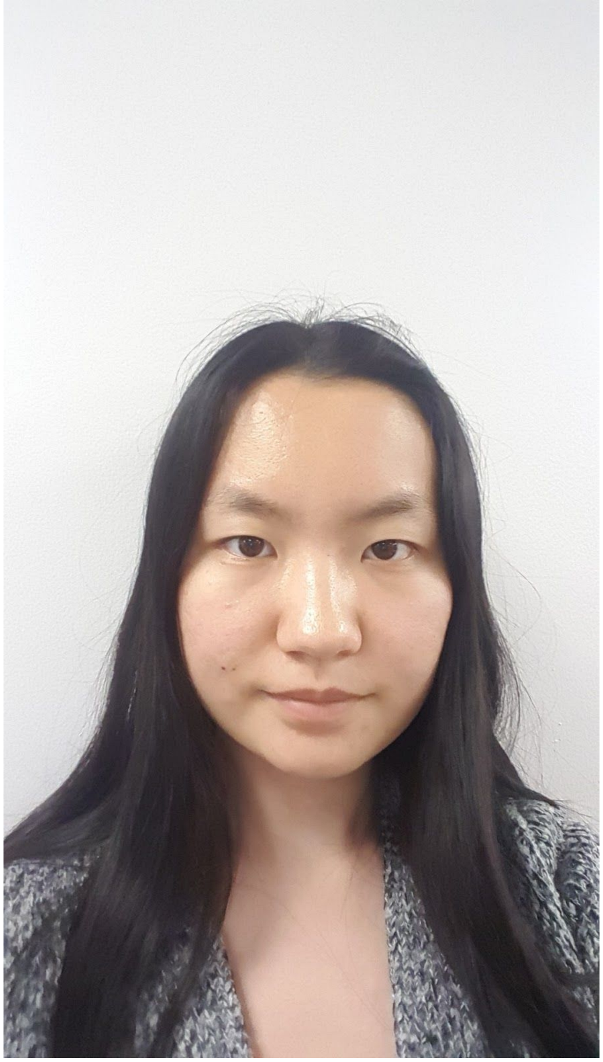
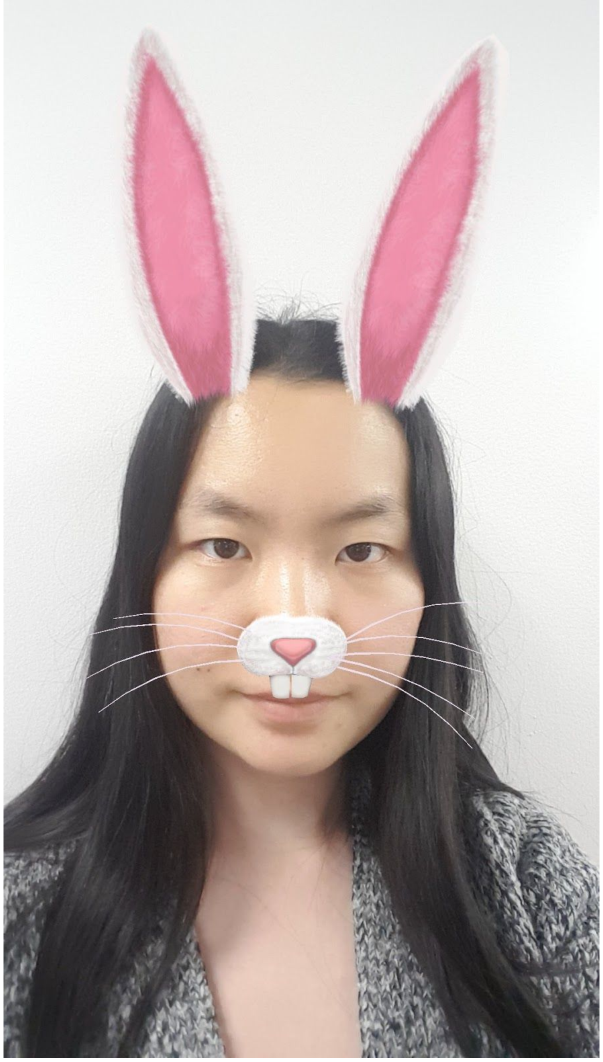
	
Before AR	After AR

Figure 3: Augmented Reality on Video Chat

3.2 Artistic Background

Through the long history of art, humans are trained or influenced to have different interpretations to different artistic styles: abstract - open to interpretation; realistic - reflecting the reality, etc. Every new art form requires significant development time to have mature forms of abstract and realism. For example, 2D paintings developed from the abstract cave paintings of the ancient civilizations, to very photographic realistic, then to various ways of abstract painting again (Figure 4). 3D modeling, as a relatively new art form, is probably in the stage of realistic appreciation. This can be seen through the increasing high expectations of realistic special effects in movies (Figure 5). This is understandable because 3D modeling has the same dimension of the real world. People innately expect models to be realistic at the beginning stage of this art form. However, efforts are made to abstract 3D models to open their interpretations. For example, the artwork of Zeitguised Studio, Nikopicto Studio, artist Mizter H (Christophe Heughe), and Antoine Magnien create inspiring abstract 3D artworks (Figure 6).

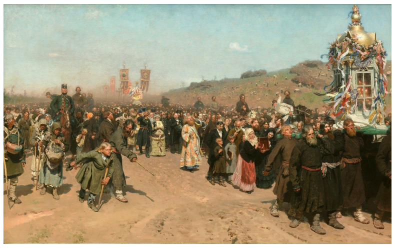
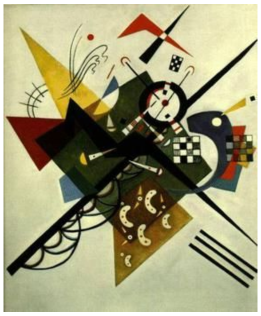
		
Cave painting in Bhimbetka Rock Shelters, India (10000-5000 B.C.)	Ilya Repin, *Religious Procession in Kursk Province*, 1880–83	Wassily Kandinsky, *On White 2*, 1923

Figure 4: Painting Style Changes from Abstract to Realistic, Back to Abstract

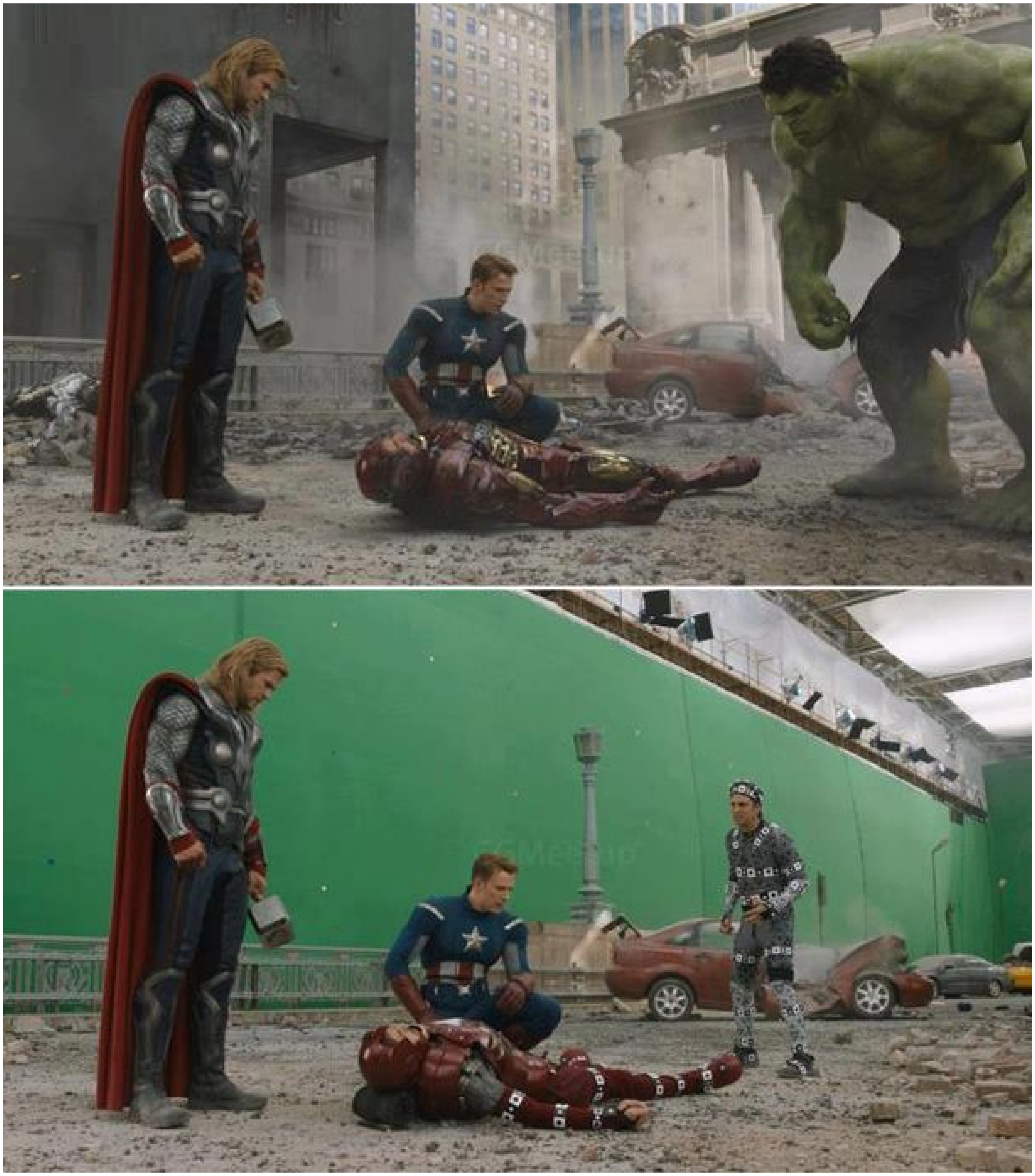
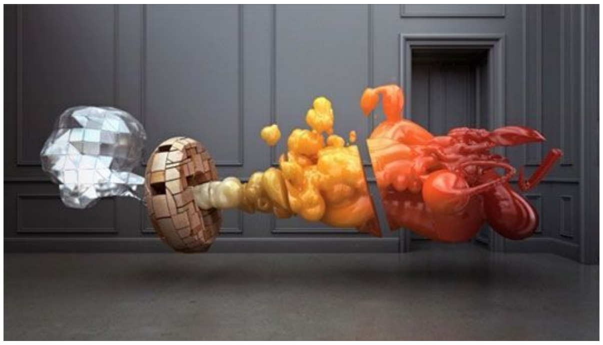
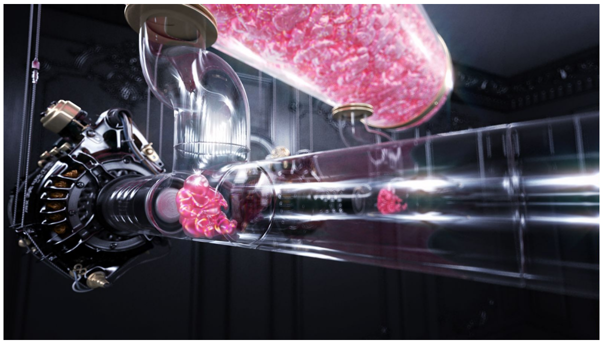


Figure 5: Special Effect in *The Avengers*

	
Work by Zeitguised Studio	Work by Nikopicto Studio

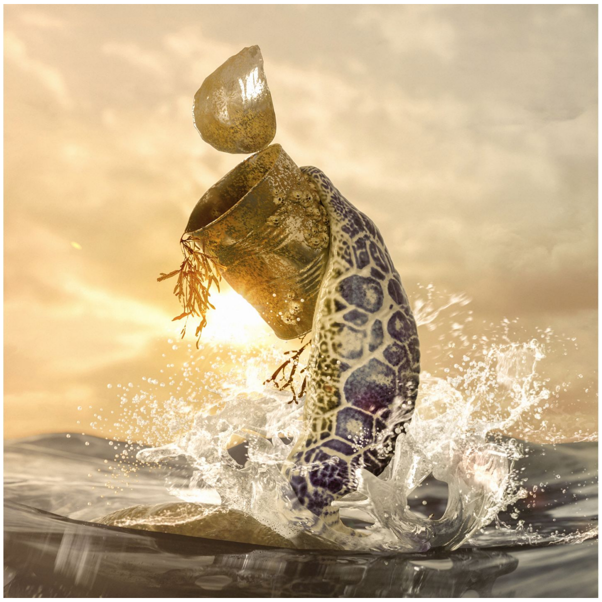
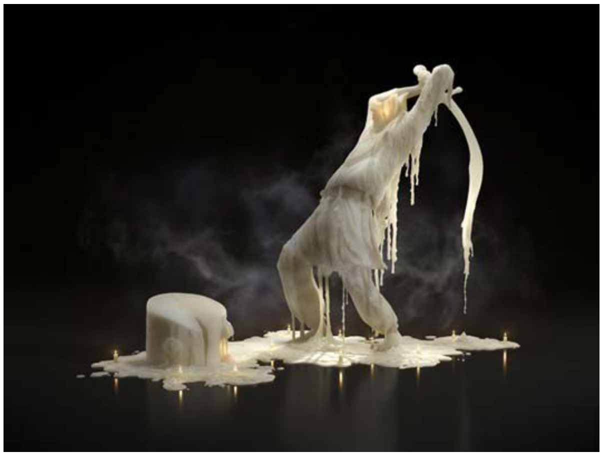
	
Work by Mizter H	Work by Antoine Magnien

Figure 6: Abstract Art Work by Various Artists

Tools for making 3D objects are getting better and easier to use. There are many tools for 3D modeling, including Sculptris[17], Maya[18], Blender[19], etc. The new tools push the innovation of styles.

Although games are usually made for entertainment purposes, socially conscious games always exist. Various social issues can be simulated in games to raise people's attention. For example, *Watch Dog 2*[20] is a game raising the concerns of big data, privacy and government overwatch. In the game the player hacks into the government systems made by irresponsible companies which monitor and manipulate citizens. While players enjoy the empowerment in the game of being able to hack, the social issues are presented blatantly to them in the game. The game I created is similar to this. I let the players to take the role of a delivery entrepreneur. In order to survive, players need to strategize seriously. By playing the game, players can not only enjoy the achievements of reaching the target, but also experience the mental and physical work needed for a future laborer even though the physical labor has been minimized by a lot in the game environment.

3.3 Social Background

Accompanying technological advancement is always the loss of jobs and the replacement of human laborers by machines. A study by Oxford University estimates that in the U.S. 47% of jobs could be replaced by robots and automated technology within the next two decades[21]. At the same time, cost of living is always increasing. The education cost increases about 3% annually[22] and modest projection of inflation rate is 2-3% in the U.S[23]. Threats are always present and lives are getting harder.

This project addresses the result of the technological advancement in its narrative about being a delivery person - the delivery entrepreneur and his/her spouse lost his/her job because machines are taking the jobs of humans. Other than making people lose their jobs, technological advancement creates new opportunities: as people need to work harder to compete with machines, they even save the time for ordering food - they take whatever the delivery person brings to them but are not willing to pay much for food that they do not like. The delivery entrepreneur needs to think hard on how to maximize earning through limited time and energy so as to survive in the society where cost of living is always increasing. This project simulates this entire situation and presents the issues right in front of the players: future laborers which is represented by the delivery entrepreneur the user plays; artificial intelligence and technology which are discussed in the backstory of the delivery entrepreneur and the entire virtual society; hardships in lives which is simulated through the increased difficulties in the game and the amount of work the user needs to do in the game; and identity politics which requires the user to ponder on the transition of the delivery entrepreneur from doing a sedentary job to a labor-intensive job.

With the widespread of smartphones, the proportion of mobile gamers has become the largest among all the gaming platforms in the global game market[5]. Mobile games thus reaches the most audience and is the most influential gaming platform.

4. Methodology

4.1 Choice of Tools

I chose Unity[24] as the game engine, Maya as the 3D modeling software and Android as the mobile operating system.

Unity developed by Google is targeted at mobile devices which paired with their head-mounted devices could be used to have VR experiences. It allows the game to be built for various devices, including desktops (supports both Windows and Mac platforms), mobile phones (supports both Android and iOS platforms), and most of the popular VR headsets such as VR for

PlayStation. Google Cardboard is the cheapest headset for mobile VR. Figure 7 is an example of what Google Cardboard looks like. A user places his mobile device inside the cardboard and looks through the lenses to see the VR environment. When I developed the game, I have built in mouse and keyboard controls to simulate VR controls in order to test the game on my laptop for simplicity. With these controls, the game can be easily modified to play on other platforms in the future using Unity as the game engine.

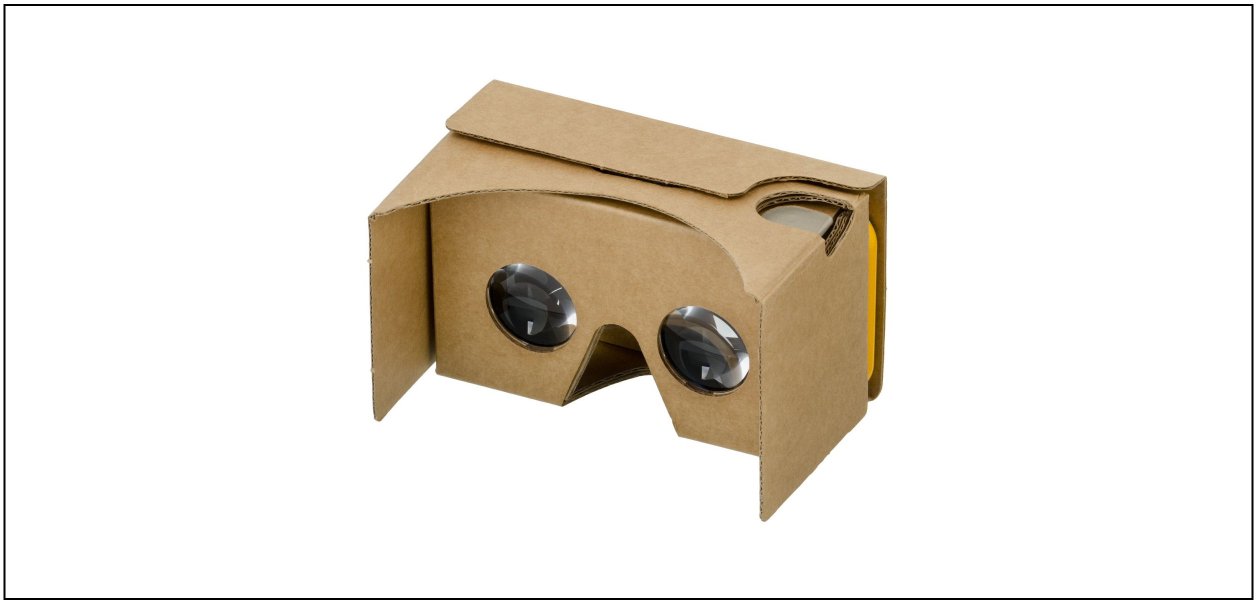


Figure 7: Google VR Cardboard

Another reason for choosing Unity is that it is compatible with the models made using Maya, which is the modeling tool I am most familiar with. Maya developed by Autodesk has many functionalities on modeling and rendering that are similar to those on Unity, with just more sophisticated features. It is one of the sophisticated softwares used in industries, thus is more than enough for my purpose.

Although the game can be built for different platforms, for the purpose of this project, I need a mobile platform to work with. Android platform was chosen because it is more compatible to Unity as both are developed by Google. In addition, this mobile phone operating system has the majority of the market share[25] so presumably it can reach the largest audience.

4.2 Game Architecture

The game has two scenes: the Home scene and the Game scene.

The Home scene contains

1. the title of the game
2. backstory
3. player directions
4. start button with countdown.

The first three objects are static and do not change. The start button leverages a technique known as raytracing. Raytracing allows a user to gaze at a particular object of interest. While doing so, it causes an event that can be captured by the VR code. This replaces the action of a user clicking a button in a traditional desktop environment. In my game, if the ray from the camera focuses on the button, a countdown of five seconds appears and the game begins when the timer reaches zero. When the ray moves away from the button, countdown stops. This feature removes the necessity of pressing a button which is not easy for a mobile VR game. It also allows the player to get ready during countdown so that they can start immediately when the scene switches to the Game scene.

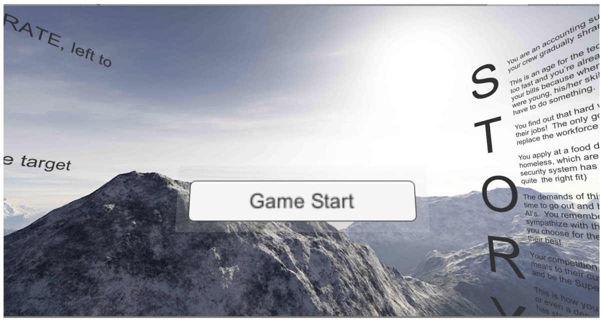
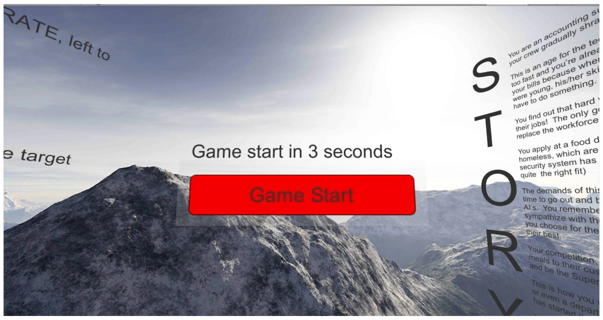
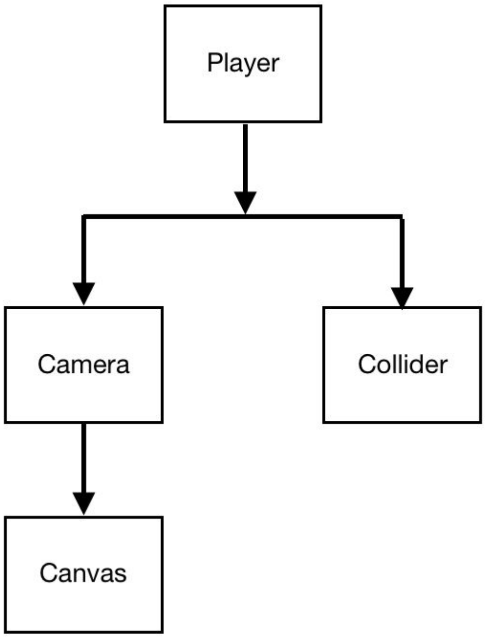
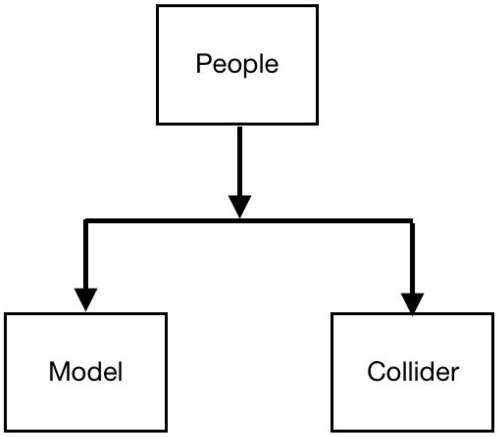
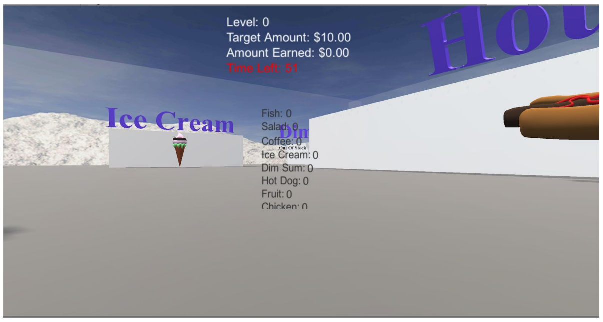
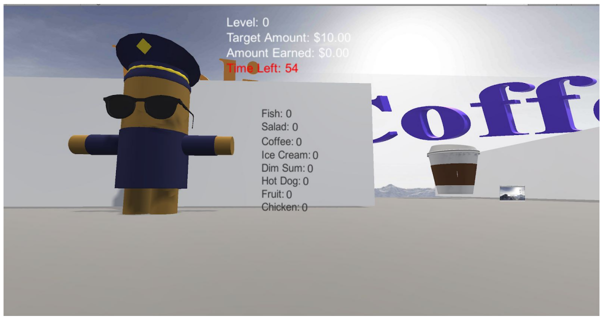
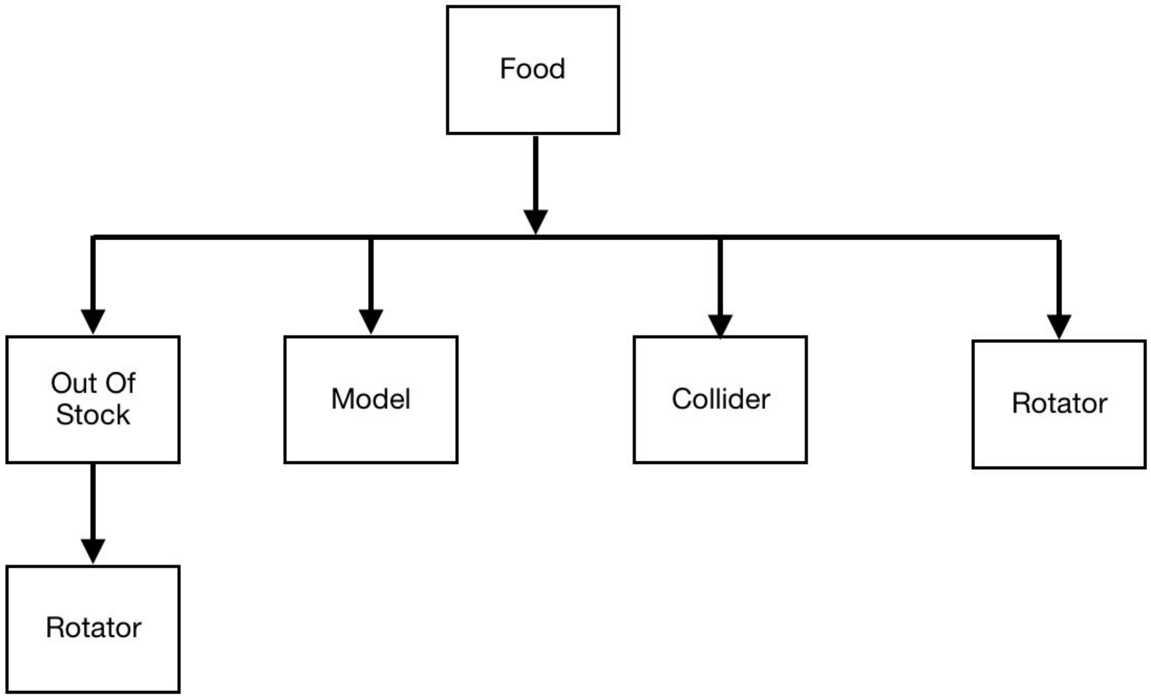
	 RATE, left to e target Game start in 3 seconds Game Start STORY
Gaze not on the button	Gaze on the button

Figure 8: Start the Game Through Gaze

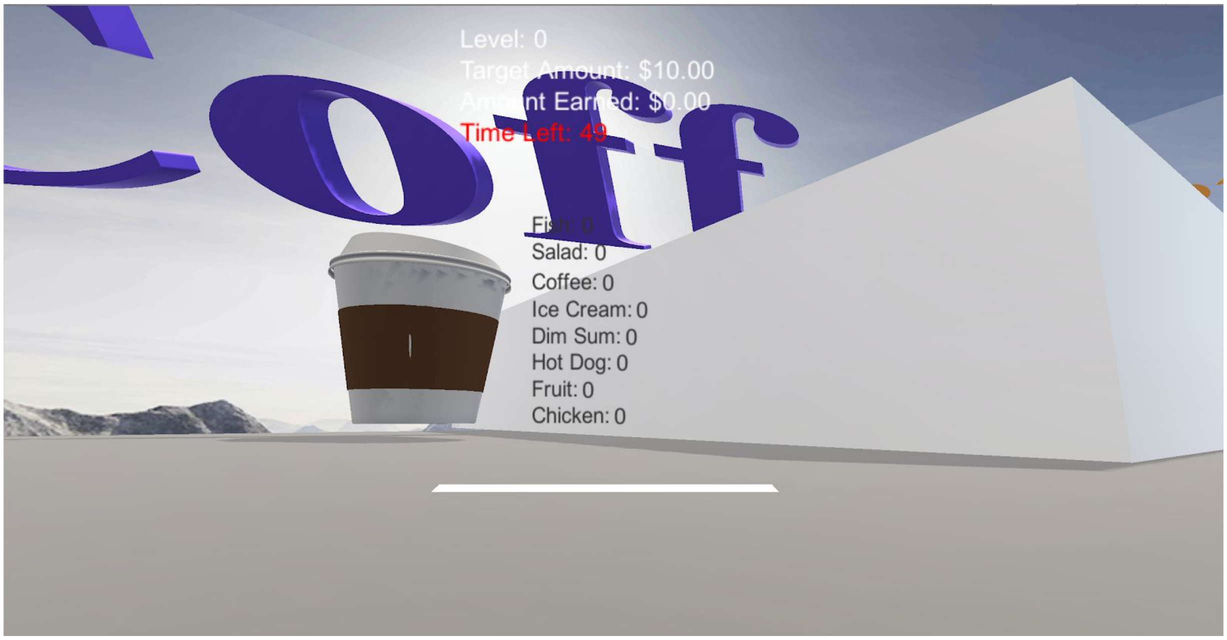
The Game scene is more complex. There are three main entities: Player, Food and People.

	
Player entity in the game. This entity is linked to the camera and performs the functions of collecting food, selling food, moving and	People entity in the game. This entity is linked to every buyer object in the game. It performs the function of saving the amount a

displaying any results.	particular buyer is willing to pay for different food and the amount of time a buyer appears and disappears.
 Level: 0 Target Amount: $10.00 Amount Earned: $0.00 Time Left: 51 Fish: 0 Salad: 0 Coffee: 0 Ice Cream: 0 Dim Sum: 0 Hot Dog: 0 Fruit: 0 Chicken: 0 	 Level: 0 Target Amount: $10.00 Amount Earned: $0.00 Time Left: 54 Fish: 0 Salad: 0 Coffee: 0 Ice Cream: 0 Dim Sum: 0 Hot Dog: 0 Fruit: 0 Chicken: 0
Player entity in the game - you cannot see it because it is attached to the camera.	An instance of the People entity in the game: policeman.



Food entity in the game. This entity is linked to every food object in the game. It performs the function of saving the appear and disappear time of a food.



An instance of the Food entity in the game: coffee.

Figure 9: main entities in the game

Player is the entity to which the user has direct control. It will move in the forward direction of the camera. It collides with food to buy food and collides with people to sell food. There is a canvas attached to the camera to display 2D information.

- **Camera** represents the first person point of view of the user. Its forward direction is the direction of the player movement. If the user tilts his head to the left, the speed of movement decreases until 0; if he tilts to the right, the speed of movement increases until 4. The reasons for the choice will be discussed in Section 5.1. The camera emits rays for raytracing. If the ray hits the ground (which means the user looks down), the player movement stops. These simulate different speeds the player "walks" through the simulated environment.

- **Canvas** displays 2D information which includes the number of each food the user has collected, the game level, the target amount of money, earned amount and a countdown. All the colliders report to Canvas whenever a collision happens because each collision represent a change on canvas - collect food and sell food both are detected via collision. Canvas also keeps the time for each level. When 60 seconds are used up, Canvas determines if the earned amount exceeds target amount: if yes, proceed to the next level. The target amount increases by 0.26% to represent the increase in education fee, inflation and improvement of life; if no, switch to the Home Scene. Game over.
- whenever a **Collider** overlaps with another Collider, the collider with a script specifying collision events determines which object it collides to and triggers subsequent events. In this game, if the Player Collider collides with a Food object, it updates the number of that type of food in the Canvas; if it collides with a People object, it gets the amount of money the person is willing to pay, clear the food in the Canvas and update the earned amount.

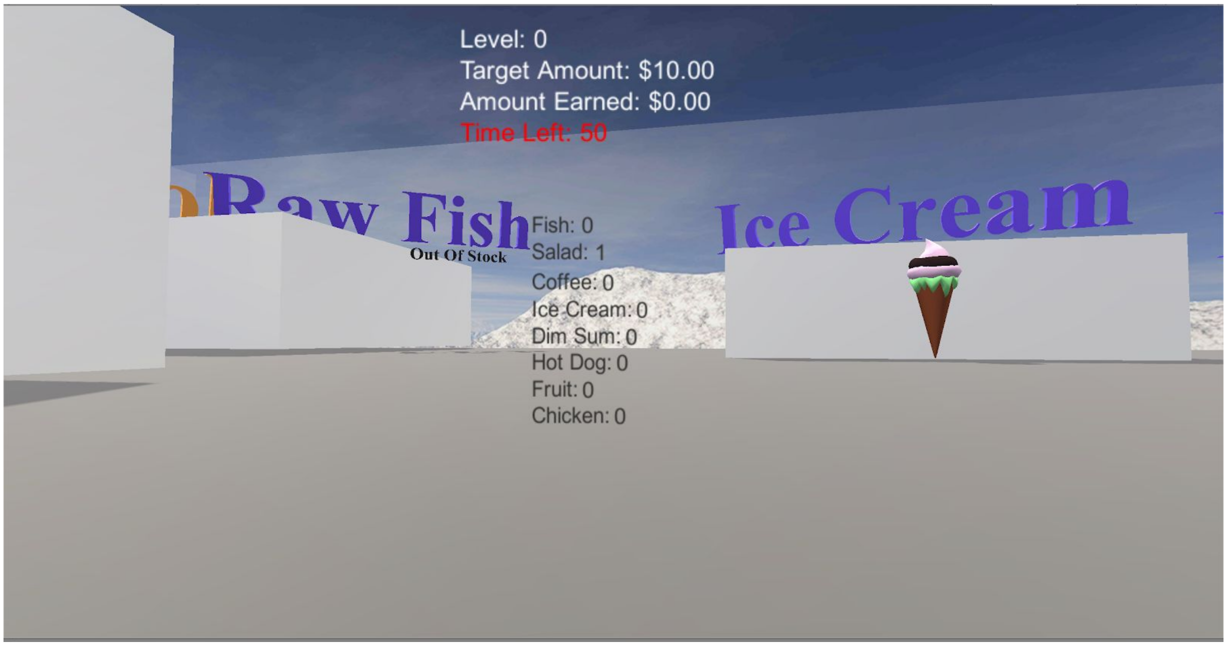
People are buyers of food in this game. Each People object contains its customized information: how much this person values each food and how long the model appears and disappears in the scene. People value different food differently based on their preferences and income level. Users should learn how much they value each food through playing to maximize income. People appear and disappear at fixed time intervals to simulate that people do not always buy food in real life.

- the **Model** of a People object is the 3D model. They have different appearances to represent different occupations. The details of the artistic design will be discussed in Section 4.3.

- the **Collider** of a People object triggers sell food event when it overlaps with the Collider of a Player object. Selling food event provides the information of how much this People object value each food to the Player object.

Food objects themselves are the name of the food in 3D text. They represent restaurants which offer take-out services in reality. Each Food object contains its customized information of how long the Model appears and disappears.

- a **Rotator** keeps the text always facing the Camera. This is important in a 3D environment because it compromises the user experience very much when the user cannot read the text easily.
- the **Collider** of a Food object triggers food collection event when it overlaps with the Collider of Player object. The food collection event updates the number of this particular food on Canvas.
- the **Model** of a Food object is the 3D model. The details of the artistic design will be discussed in Section 4.3.
- **Out Of Stock** is a 3D text which appears when the Model disappears. It shows up when the Food is collected or when the disappear time of the Model starts. It also has a Rotator to keep its orientation towards the user.



Level: 0
Target Amount: $10.00
Amount Earned: $0.00
Time Left: 50

Fish: 0
Salad: 1
Coffee: 0
Ice Cream: 0
Dim Sum: 0
Hot Dog: 0
Fruit: 0
Chicken: 0

Figure 10: Out of Stock in The Game: Raw Fish is Out of Stock whereas Ice Cream is not. Out of Stock disappears when the object is present, and vice versa.

4.3 Artistic Choices

The design of the People and Food goes for “recognizable abstract” - once users see the objects, they know what these objects represent in real life, but are certain that these are made up in the computer, rather than real things. The reason for this choice is mainly because there are People with various backgrounds and appearances; the same type of food has many variations. It is impractical to cover all the possibilities.

But if we go for realistic and not cover all the possibilities, a problem arises for the People object. There are people from various backgrounds, and my goal is to achieve inclusiveness in this virtual environment. Unintentional discrimination could easily happen if the style is too realistic. To avoid this issue, the People objects must be designed to be abstract. However, too much abstraction in a 3D environment may cause confusion because users may view the model from many possible directions. Hence the style “recognizable abstract” was chosen. The styles of the Food and People objects are similar and differentiable from other objects to hint to the users that these two are the same type of objects: collidable objects. Other entities in the scene have different styles. For example, the buildings are simply white. Other than reducing the work on modeling buildings, this choice was made to make sure Food and People objects stand out from the background thus are easily seen by the user: all the other objects in the game have vibrant colors and white gives the best contrast. In the game the reality should be modified to give users a sense of empowerment. Saving the trouble of looking for the target in a messy background makes the job of a delivery person easier and improves user experience.

For the same reason, the names of the buildings are unrealisticly large 3D texts to allow the users to see them from a distance in the scene.

Many artistic choices were made to achieve the social goal of this game:

- The game is in first person view for the user to immerse in the game as a future laborer: a delivery person. The first person view does not have a full vision of the entire game scene so that the user is more disoriented in identifying people and places, which is similar to what a real delivery person’s life.
- Buyers appear and disappear at different intervals. This is to simulate in real life that when people are working, they cannot come and pick up food. And people working for different jobs have different time flexibility to pick up food. A delivery person always need to meet the clients’ time. Similarly, food appear and disappear because other people are buying them as well. A delivery person need to rush for them when they appear,

especially for popular food. This design lets the users to experience the hardship of a delivery entrepreneur.

- Different buyers pay different amounts for different products, but these pieces of information are hidden from users. Users have to learn these to earn enough later in the game as the target amount increases. This allows the user to strategize when they play. This simulates the mental work that a future laborer needs to do. By playing this game, users get to know how much a laborer needs to think.
- The game gets progressively harder as the target amount increases each level. This is because in real life, if a person remains the same, the always improving society kicks the person out of the game. A person must always improve to survive in the highly competitive world. By playing this game, users can experience how much more work a future laborer need to do in order to keep competitive.

4.4 Playtests

The knowledge elicitation method that was chosen for the playtests was informal interviews. This method was chosen so as to keep the participants in the most relaxed state, which is similar to the state when they play a game. There were twelve people participated.

Participants tried the game using a Samsung Galaxy Note 8 mobile device which fits perfectly in a Google Cardboard to play the game. They played the game at least twice, once standing and once sitting on a 360-degrees rotating chair. While playing, the participants are asked to express what they see and what they think. Any non-verbal sounds made, like “Oh!”, “Ah!”, “Wow!” were noted and asked about after the gameplay because these are important emotional cues.

5. Results

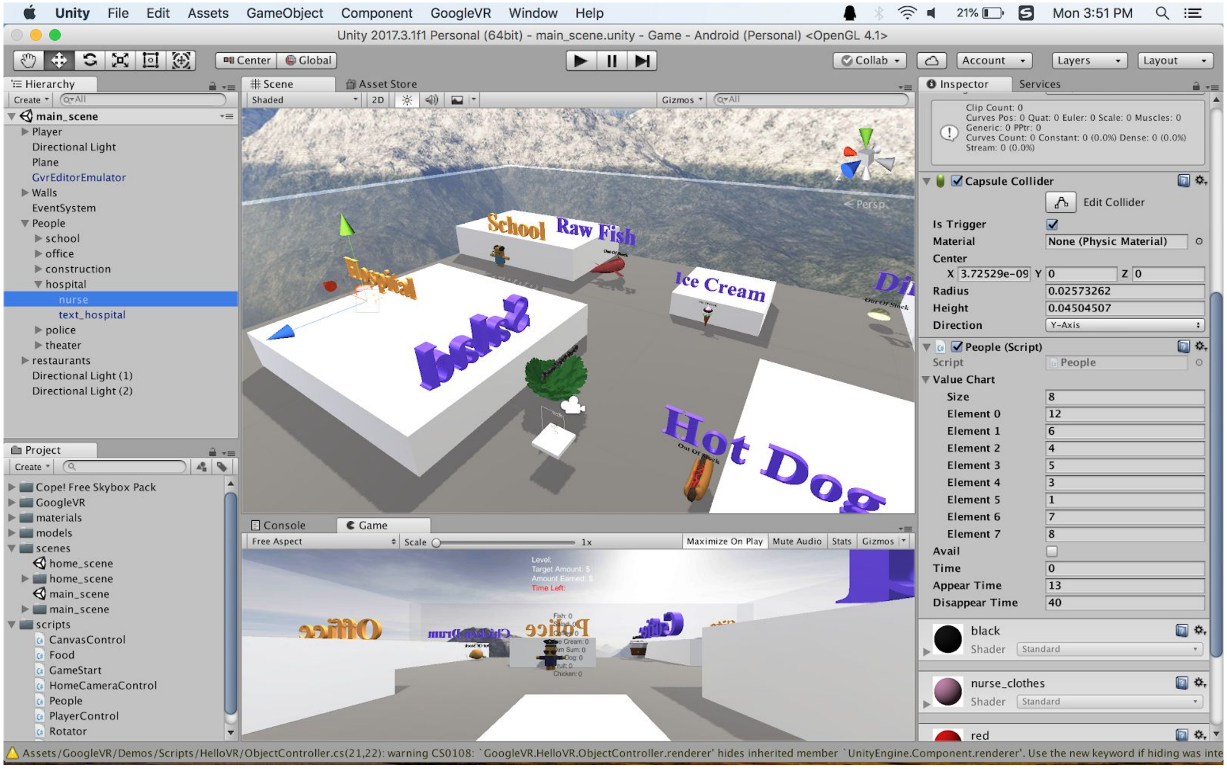
5.1 Coding Evaluation

The coding of this game is evaluated in three aspects: game running performance, scalability and readability.

The running performance of the game is satisfactory. During all the playtests, the game runs smoothly with no unexpected errors[27]. From the output of the console, the frame rate varies a bit depending on the number of things to be rendered. However, frame rate changes are not noticeable for various reasons. Firstly, the scene is relatively small. Thus the number of static components in the scene is capped at a small number. Less computing power is required for the calculation related to the static components. Furthermore, less calculations are required to render further away models which are not in the interest of the user, or apply mist effect to hide further away models. Secondly, all the changes in update function (the function called every frame) are scaled by Time.deltaTime which is the time between each frame is rendered. This ensures that at a constant speed, the camera moves the same distance per unit time, no matter how much time is used to render a frame. Therefore the camera appears to move smoothly.

This game was designed with the consideration of future scaling. Therefore it is very easy to expand the game. This game is mainly scaled up by adding more restaurants and buyers. The codes for all restaurants are the same, so do the codes for buyers. The unique attributes of the new restaurants/buyers are public, hence can be easily set in the Unity properties editor. To scale the game up, a programmer simply needs to copy one of the existing restaurants/buyers, change the model and change the public attributes. Other than that, if a new restaurant is added, the Player's food array and the canvas food count must be updated to accommodate the new food, and all the buyers' food array must be updated to set the perceived value of the new food. Almost all the changes can be easily done right in the Unity interface. The only change of code is in the Player class where a new type of collider must be recognized when the Player collides with the new Food. The code is the same for all the Food colliders. We only need to copy the code and replace the collider type to the new one.

Other than the standard ways to improve code readability, like commenting, naming, non-repeating, this game makes use of the game engine interfaces to make any code and object referencing clear and simple. All the codes are made as a component of the object it controls. Whenever there needs a reference of an object in the code, a public attribute for a GameObject is created so that the referenced object can be dragged in the Unity interface to be linked to the code. This saves the trouble of finding and attaching objects in the code. This improves the readability because all the references to objects are directly shown.



Unity 2017.3.1f1 Personal (64bit) - main_scene.unity - Game - Android (Personal) <OpenGL 4.1>

Clip Count: 0
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0 Muscles: 0
Generic: 0 PPtr: 0
Curves Count: 0 Constant: 0 (0.0%) Dense: 0 (0.0%)
Stream: 0 (0.0%)

☑ Capsule Collider

Edit Collider

Is Trigger	☑
Material	None (Physic Material)
Center	X 3.72529e-09 Y 0 Z 0
Radius	0.02573262
Height	0.04504507
Direction	Y-Axis

☑ People (Script)

Script	People
Value Chart	
Size	8
Element 0	12
Element 1	6
Element 2	4
Element 3	5
Element 4	3
Element 5	1
Element 6	7
Element 7	8
Avail	□
Time	0
Appear Time	13
Disappear Time	40

```
Assets/GoogleVR/Demos/Scripts/HelloVR/ObjectController.cs(21,22): warning CS0108: `GoogleVR.HelloVR.ObjectController.renderer' hides inherited member `UnityEngine.Component.renderer'. Use the new keyword if hiding was inte
```

The game engine interface. The game objects are arranged in hierarchy on the left (the Hierarchy panel). The code can be dragged to the inspector of an object on the right: in this case, the People(Script) component in the Inspector panel. All the public attributes can be set right in the People(Script) component.

Figure 11: The Game Engine Interface

5.2 User Experience

Most of the design choices were unconscious to the users. All of the users did not mind the lack of realism in the scene because things they need to know and care about are clearly presented. And the intensity of the game is high enough to catch their attention. Probably if the scene gets more colorful, it can be annoying and more dizzy when people rush through the scene in a VR environment. It is believed that the content of the game is more important than visual pleasure to keep people playing this game.

The designs that improve user experience the most tend to be those not noticed during the process of playing. For example, none of the users noticed that all the 3D texts are always facing the camera. But all of them agree that this improves the user experience a lot. Another example is that all the users noticed the minimum speed is 0 but none of them noticed that there is a maximum speed limit. The maximum speed is actually 4 units of displacement per frame. This is the speed that most of people get cybersickness, which is the headache and nausea when people see movements but do not feel it[26]. This was found out during preliminary playtests. Users believe when they reach the maximum speed, the changes on the peripheral vision are too fast for them to notice if the speed changes at all. At that point, they tend to slow down to prevent sickness. This was expected when the speed was determined.

Users are almost only sensitive to things that cause physical uncomfort. For example, all of the users complain about the the Google Cardboard whose lenses do not exactly focus on the screen. And before the movement speed and the rate of speed change of the camera were determined through playtesting, most of the users stopped playing for cybersickness.

There were unexpected findings during the playtest. Users love the high-resolution mountain skybox used as the background. All of the users who tried that background invariantly looked down from the unmovable position in the air in the Home scene for the virtual experience. I have tried a panorama image of buildings as the background at a later stage of playtesting. Users in that batch did not express interests in the background. Since the users enjoy the mountains so much, I changed it back although it does not comply with the theme of the game that much.

The player's pose while playing also makes a difference. All of the users report that sitting on a rotating chair improves the playing experience as turning around is made easier. Standing also makes them feel insecure for falling down or knocking into surrounding objects in the real world.

5.3 Social Goals

The social goal is for the users experience the life of a future delivery entrepreneur, and reflect on matters like future laborers, artificial intelligence, society and competition. After a short time playing the game, this social goal seems not to be achieved.

All of the users saw the backstory before the game started. Only two of them finished reading. The rest of the users went back to read the backstory after they finished playing the game, and get distracted by the background easily. But the percentage of people going back to read the backstory is doubtful because users might not still read the long text if they were not in a playtest environment.

After playing the game, all of the users talked about the experience as a player. But only one of them related the game experience with any possible social messages. Other users only start to think of it when they are asked on related questions. Hence, the social message of this game might take some time for users to realize: an opportunity might be necessary for them to think in that direction.

There are some ideas on achieving the social goal better that could be implemented in the future. For example, I have added an image of the family above the game title, which is the first thing people would see when they lose the game and return to the Home scene. In the future, I can add in messages from the family members after the game ends. The message changes based on the user’s performance. This would remind the user to think about the family, and probably subsequently reflect on the issues in the social goal.

Also, a short animation of the backstory could be shown before switching to the Game scene from the Home scene. Another solution to let people know more about the backstory is that the backstory text could have a summary version and a longer version. The user would have a higher probability of watching an engaging animation or reading the shorter text. They can choose to read the longer text if interested.

5.4 Compare and Contrast with Desktop Applications

1. In developing desktop applications, usually there are not many instances of the same object. Even if there are, in many cases the information of each individual object are saved in the same place such as a file or an array for easy management. However, in a game engine an instance of an object is clearly in the hierarchy and its public attributes are easily seen on the GUI. The initialization of the public attribute values can be done in the game engine before the game starts. In a game engine, it is more intuitive and easier to save data as object attributes on each objects instead of saving them in a centralized place.

2. On a desktop, the choice of input devices is abundant. The default are the mouse and keyboard. For console games, the are many other control devices available for various games. For a mobile VR game, due to the short distance from the screen to the eyes, the usual multi-touch screen of a mobile phone is covered by Google Cardboard. Only one finger can stick to the screen. But sticking a finger on the screen affects the VR experience. Therefore, without professional VR gesture sensors for a mobile VR game, the only input is the head movement which is detected by the accelerometer of the phone. All the inputs of the game need

to be designed with simple head movements. This must be done carefully because fast head movements causes sickness.

3. Although I did not encounter this problem for this individual project, I can see that cooperation using version control is harder for programmers using game engines. The game objects are represented in the game engine in an indecipherable way for humans - the series of vertices of a geometry are only floating point numbers; the attributes of an object are saves with no descriptions but bare numbers. Once there are merge conflicts not on scripts, programmers can hardly resolve them. This is unlike the situation for programmers developing desktop applications. In that case, merge conflicts can be easily resolved changing the codes written by programmers themselves. Therefore careful distribution of tasks and the order of development for cooperation using a game engine is very important.

6. Conclusions

Overall, this project is an engaging game at least for the playtest participants. Other than the game itself, the experience of making the game gives much insights on how to create good gaming and VR experiences for the future.

For game development, the architecture of the game on a game engine should be designed to make use of the game engine features. There are more duplicates of the same object or class in a game than normal programs. The game engine displays all the public attributes for easy modification of different instances of an object or a class. This feature should be considered when the architecture is designed.

To achieve social goals of a game, sometimes we need to build things that people do not like. This time I chose to use the mountains as the background. Although this looks enticing for the audience, it makes the suppressing city life that I was trying to simulate less stressful. The conflict between what people like and what the game developer wants to communicate is always present in game development. Game developers sometimes need to sacrifice one for another. In the future, I might change the building design to be more realistic, and create a background image of city buildings of the same style.

Although VR can simulate experiences very similar to the reality, the level of realism of models may not be extreme. From my playtests, people can distinguish VR and reality, thus can tolerate models to be abstract, as long as abstraction does not compromise the messages that users care about. Any VR applications in the future can make use of this tolerance and make use of abstraction in various ways.

Despite of the completion of this project, the game has much room for improvement. For example, the simulation of food being picked up by others could be replaced by real players and let players to interact real time via internet. A scene of the player's home could be added to convey more social messages. Weather system could be added to affect the speed of movement. This is a project with great potential to be expanded.

7. References

1. Sony PlayStation. (n.d.). PlayStation VR - Over 100 games and counting. Feel them all. Retrieved from https://www.playstation.com/en-us/explore/playstation-vr/?emcid=pa-pe-97928
2. Sag, A. (2017, February 09). The Future Of Virtual Reality: Mobile VR Platforms In A Battle. Retrieved from https://www.forbes.com/sites/moorinsights/2017/02/09/the-future-of-virtual-reality-mobile-vr-platforms-in-a-battle/#6880fc913c9d

3. Unity. (n.d.). Optimisation for VR in Unity. Retrieved from https://unity3d.com/learn/tutorials/topics/virtual-reality/optimisation-vr-unity
4. Virtual Reality Society. (n.d.). History Of Virtual Reality. Retrieved from https://www.vrs.org.uk/virtual-reality/history.html
5. Frey, C. B., & Osborne, M. A. (2017). The future of employment: How susceptible are jobs to computerisation? *Technological Forecasting and Social Change, 114*, 254-280. doi:10.1016/j.techfore.2016.08.019
6. Glenister, M. (n.d.). The User Experience of Virtual Reality. Retrieved from https://www.uxofvr.com/
7. Virtual Reality Society. (2017, June 28). Applications Of Virtual Reality. Retrieved from https://www.vrs.org.uk/virtual-reality-applications/
8. Anthes, C., Hernandez, R., & Kranzlmuller, D. (n.d.). State of the Art of Virtual Reality Technologies. doi:10.1109/AERO.2016.7500674
9. TONI_K. (n.d.). SparkFun Triple Axis Accelerometer Breakout - ADXL335. Retrieved from https://learn.sparkfun.com/tutorials/accelerometer-basics
10. Instructables. (2017, October 23). 3D Stereoscopic Photography. Retrieved from http://www.instructables.com/id/3D-Stereoscopic-Photography/
11. Anderson, T., Blank, M., Daniels, B., & Lebling, D. (1977). Zork by Tim Anderson, Marc Blank, Bruce Daniels, and Dave Lebling. Retrieved from http://textadventures.co.uk/games/view/5zyoqrsugeopel3ffhz_vq/zork
12. Nintendo. (1985, September 13). Super Mario Bros. (NES) - online game. Retrieved from http://www.retrogames.cz/play_005-NES.php?language=EN
13. Valve Corporation, Sierra Entertainment, Namco, Nexon. (2000, November 9). Counter-Strike. Retrieved from http://blog.counter-strike.net/
14. Niantic. (2016, July 6). Catch Pokémon in the Real World with Pokémon GO! Retrieved from https://www.pokemongo.com/
15. Ubisoft. (2016, December 6). Werewolves Within. Retrieved from https://www.ubisoft.com/en-us/game/werewolves-within/
16. Raja. (2018, March 19). Infographic: Future of Mobile App Development 2018. Retrieved from https://www.dotcominfoway.com/blog/infographic-future-of-mobile-app-development-2018
17. Pixologic. (n.d.). Pixologic :: Sculptris: Enter a world of digital art without barriers. Retrieved from http://pixologic.com/sculptris/
18. Autodesk. (n.d.). Maya | Computer Animation & Modeling Software | Autodesk. Retrieved from https://www.autodesk.com/products/maya/overview
19. Blender Foundation. (n.d.). Home of the Blender project - Free and Open 3D Creation Software. Retrieved from https://www.blender.org/

20. Ubisoft. (2016, November 15). Watch Dogs 2. Retrieved from https://www.ubisoft.com/en-us/game/watch-dogs-2/
21. The Real Cost of Higher Education. (n.d.). Retrieved April 17, 2018, from https://www.savingforcollege.com/tutorial101/the_real_cost_of_higher_education.php
22. Statista. (n.d.). U.S. - projected inflation rate 2008-2022. Retrieved April 17, 2018, from https://www.statista.com/statistics/244983/projected-inflation-rate-in-the-united-states/
23. Emma McDonald. (2017, April 20). Retrieved April 17, 2018, from https://newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/
24. Unity Technologies. (2005, June 8). Unity. Retrieved from https://unity3d.com/
25. Statista. (n.d.). Mobile OS market share 2017. Retrieved April 17, 2018, from https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/
26. A. (2015, November 17). Cybersickness: The new 'illness' sweeping the nation. Retrieved from https://www.telegraph.co.uk/news/health/news/12001743/Cybersickness-The-new-illness-sweeping-the-nation.html
27. Liu, Y. (2018, April 30). Delivery Entrepreneur. Retrieved from http://v.youku.com/v_show/id_XMzU3NjUxMTc0OA==.html?spm=a2h3j.8428770.3416059.1
28. Kamburelis, M. (n.d.). Castle Game Engine. Retrieved from https://castle-engine.io/