

**Department of Computer Science, Faculty of
Engineering, The University of Hong Kong**

**COMP4801 Final Year Project
A First-Person VR Puzzle Game**

Supervised by Dr. T. W. Chim

FYP18015 Interim Report

Wong Ka Wing (3035124269)

Wong Yu Hin (3035124324)

Date of submission: 20/01/2019

Abstract

The virtual reality technology has brought a significant impulse towards the game industry. It allows the players to have the feeling of virtually being there inside the game. However, the freedom brings by current VR games are quite restricted in terms of the availability of interaction between the virtual world and the player. This paper demonstrates the preparation work, design procedure and implementation method of “Dream Driver”, a VR puzzle game which emphasizes the freedom, unreality and creativity by integrating the concept of dream into the game. While the design of the game is close to being finished, our next major step would be to focus on the implementation of the game logic and mechanics in the coming months. The features of the game will hopefully be achieved in the code level and the core design of the game.

Acknowledgement

We would like to extend our sincere thanks to all individuals and organizations who have kindly assisted and offered help to this project.

We are highly indebted to our supervisor Dr. T. W. Chim for his regular supervision, sharing of his experience and ideas for this project. He has always conducted meetings with us and giving us useful advices and feedbacks.

We would like to give special thanks to FIGHT 4 DREAM LIMITED for their expert suggestion. They have aggressively provided us advices and answered our questions regarding game development.

Table of Contents

Abstract	1
Acknowledgement	2
Table of Contents	3
List of Figures	5
Abbreviations	5
1. Introduction	6
1.1 Background	6
1.2 Scope	7
1.3 Objectives	7
1.4 Outline	8
2. Methodology	9
2.1 Game Design Document	9
2.2 Unity Game Engine	9
2.3 Design Principle	10
3. Current Progress	11
3.1 Game Design	11
3.2 Hardware Testing	11
3.3 Technical Study	12
3.4 Version Control	13
3.5 Feature Implementation	13
3.5.1 Locomotion with Gesture	13
3.5.2 Dream Driver	15
3.5.2.1 Energy Absorption	16
3.5.2.2 Summon	18
3.6 Artistic Evaluation	19
3.7 Post-processing	20
4. Challenges Encountered	21
4.1 Design Decision	21
4.2 Artistic Work	21

5. Future Planning	22
5.1 In-game Features	22
5.2 Stage Design	23
5.3 System Involved	23
Conclusion	24
References	25

List of Figures

Figure 1	HTC Vive Headset, Controllers and Base Stations	6
Figure 2	Setting Up the Hardwares	11
Figure 3	Hardware Testing	11
Figure 4	VIU Scripting	11
Figure 5	Ninja Run Locomotion	13
Figure 6	Ray Casting From Controllers	13
Figure 7	Dream Driver	14
Figure 8	Specific Dialogue System	14
Figure 9	Energy Cube and Power Station	15
Figure 10	Power Station with Absorbed Energy Order From Blue to Yellow	16
Figure 11	Animation of Successfully Absorption	16
Figure 12	Sword with Cyan Energy	17
Figure 13	Experiment laboratory scene in game	18
Figure 14	Keys and Clues without Post-processing	19
Figure 15	Keys and Clues with Post-processing	19

Abbreviations

VR	Virtual Reality
GDD	Game Design Document
SDK	Software Development Kit
VIU	Vive Input Utility
API	Application Programming Interface

1. Introduction

In this chapter, the background information of the topic will be introduced by evaluating the status of different industries. Following with the scope and objectives, the motivation of developing a VR puzzle game will also be addressed. Lastly, the summary of this paper is located at the end for references.

1.1 Background

The rapid growth of VR technology has brought up a numerous of applications in different industries, like medical [1], logistics and business [2]. Among all, gaming is one of the field that has been substantially promoted because this technology absolutely brings the user experience to a next level [3]. Undoubtedly, VR games are being releasing onto the market. Game developers attempt aggressively to explore more innovative idea to fully utilize the experience that VR technology provides. Some work on various game types and some focus on the hardware support [4] which collapse into the same goal of enhancing the human senses in the virtual world.

At this stage in the VR game industry, the leading companies in the market like HTC, Samsung and Sony have published a similar model of VR components to all around the world (see Figure 1) [5][6][7]. These standard equipments are constituted by a headset, a pair of controller and a pair of base station. Providing with a set of these devices, players could have their visual and interactive illusion of being in the virtual world by wearing the headset and playing the object around with the controllers. However, the generality of the VR-related hardware brings a huge limitation to game developers that they have to stick to the functions have been given unless a further hardware support can be globally circulated. Luckily, there is always possibility and potential in the implementation level of a VR game.



Figure 1. HTC Vive Headset, Controllers and Base Stations [5].

1.2 Scope

Puzzle games are a popular game type on different platforms. When it comes to VR, game developers have suggested some great idea like virtual escape room [8]. However, lack of interaction related implementation turns out to be the common failure of VR puzzle games upon our observation and experience. Puzzle game should be solvable within a reasonable logic, any restrictions that does not allows players to demonstrate their common sense are a bad practice for a puzzle game design [9]. Especially for such a VR puzzle game that highlighting the importance of interaction, it becomes the best choice for a VR game project aiming to enhance the player's visual feeling and experience in the virtual world.

1.3 Objectives

The primary objective of our project is to develop a VR puzzle game in Unity and coding with C# languages. Players can play the game with a PC and a set of HTC Vive VR kit. In order to reinforce the virtual experience for players, the game aims to mitigate all the unnatural and weird feeling brings up by the unresponsive reaction for every object interaction. With the idea of open-ended puzzle integrated, players will have even more freedom to explore the virtual world and adopt the best way they find to pass a level in game. Supporting with the logic scripts implemented, this feature will probably provide another excitement to our players while they are being in the virtual world.

The secondary objective of developing the game is for commercial purpose. As this project is supervised and cooperated with a local game company, the Fight4dream Limited [10], we are working together towards the direction of publishing the game onto the Steam platform. However, this is nearly an impossible task to finish a complete version of the game with the quality assurance within a short period of time. As a result, the final deliverable will only be the first phase of the entire game. In which it leads to the final objective of this project, that is to get ready for the future implementation. To achieve this, we have to maintain our code its readability, modifiability and sustainability for adapting any possible structural change afterward.

1.4 Outline

This paper demonstrates the processes to build up a VR puzzle game from scratch, which includes the preparation work for setting up a VR ready environment, designing procedure for pointing out the details of workflow and implementation method for achieving the desired outcome and effect in the game.

The remainder of this progress report proceeds as follows. First, we introduce the methodology for implementing a VR game including the justification of software choices. Followed with our current working progress on this project, the difficulties encountered and the mitigate solutions have also been discussed. the scheduled work in the coming months. Lastly, we close with the conclusion of this paper.

2. Methodology

This chapter introduces two main tools for developing a VR game. The first one is game design document where it act as a implementation guideline for programmer to follow with. The second one is Unity, one of the most well known game development platform. Design principles of the game have also included at the end to pinpoint the importance of the above methods.

2.1 Game Design Document

Before starting the project at once by digging into the coding part, a game design document (GDD) is employed to present the idea in a systematic manner after brainstorming [11]. The GDD generally includes the game identity, mechanics summary, features, interface and art style etc. These kind of keyword allows the developer to focus on the scope of the game. Some detailed GDD also provides several use cases to indicate every details coming from an idea. As the routine is explained step by step, developers can easily follow the instruction and implement the required logic accordingly. It boosts the efficiency of development process by making everything clear.

2.2 Unity Game Engine

Unity is a famous game development platform which provides a clear user interface and a set of simple commands and operations built-in [12]. The main reason for choosing Unity as the game engine is we have had experience to play around it in our previous game development course, we are more familiar with it. In addition, comparing to another game engine, Unity supports high level javascript and C# languages for development. These two languages are rather handy especially for beginners as everything can be defined like an English sentence [13]. The asset store associated along with Unity is also intrinsic to an efficient development. As a amount of high quality assets can be obtained free from charge on the asset store, we as an engineer can have no worry regarding the art stuff too.

2.3 Design Principle

As being a beginner in game development, most of our work can be done are highly relying on the SDK provided by the publisher, especially for those newly released VR-related package where only a limited modification can be applied to avoid the faulty deconstruction of content structure, the design principle is then mostly depending on the practicality of the implementation. While the feasibility is being taken into account, the time cost is also a concern in the design process. Therefore, the scale of project has to be always under control. To conclude, the first priority of our design principle is to be practical.

For the general in-game ideas, the most efficient way for us is referring to the game ideas on the current market. As puzzle game is a typical game type, we have concluded three main design principles after conducting the research. They are common sense, brute force and relativity. The first one means the solution to the puzzle can always has a trivial answer like a natural phenomenon, the second one means the puzzle can always be solved within a reasonable trials and the last one means the flawless logic between puzzle should always be presented. And the game ideas are generated by these principles to ensure there is enjoyment provided to every player inside the game.

3. Current Progress

In this chapter, the current progress for our project is presented. It includes the design progress of the game ideas, a successful hardware testing result, and the method we have adopted to help with version controlling. In the meantime, some basic features of our game has been implemented and we have also done part of the artistic evaluation for building the game scene.

3.1 Game Design

After conducting research on the trending style of puzzle game available on the market, we have settled down with the topic and criteria of our game. We have finished the majority part of our GDD with writing down all the unique features, systems, art style and level design of our game. As developing a puzzle game requires a lot more innovative idea, each stage need to have its own characteristic and be a much more different one than another. These tasks are pending to be designed and added into our to-do list in Trello, an application that allows users to manage their projects and organize things in a easy, flexible and visual way. With the help of Trello, we can keep track of our process of development and achieve division of work easily. It will be a convenient tool when we come to the stage of implementation.

3.2 Hardware Testing

The HTC Vive VR components have been successfully set up and tested in our own place followed by the step by step instructions (see Figure 2 and 3). We have gone through the connection of the headset and computer, firmware update of the controller and room scale measuring performed by the two paired up base stations for motion tracking. We finished and passed also a built in tutorial associated in Steam VR and a Unity version testing scene to ensure the functionality of all the hardware.

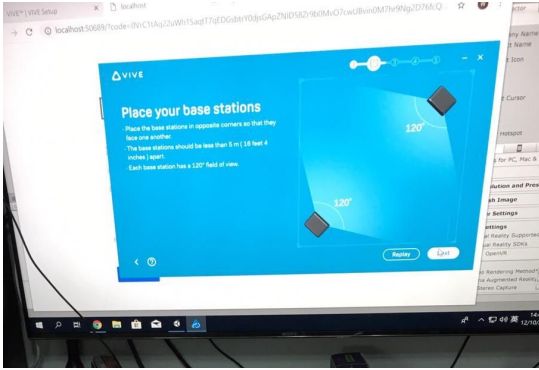


Figure 2. Setting Up the Hardwares.



Figure 3. Hardware Testing.

3.3 Technical Study

HTC Vive has recently released their exclusive SDK package, “Vive Input Utility” onto the Unity asset store. We downloaded and imported the package into our project free of charge. The SDK consists of the aforementioned testing scene, a detailed tutorial and guideline explaining the relationship and interaction method between the VR components and the virtual world. We have briefly studied the API and function call across different testing scripts provided to get ourselves ready for the coming task (see Figure 4). Based on all those given materials, we will have to work with the basic and do a lot more advanced modification in the coming tasks.

Class **ViveInput** under **HTC.UnityPlugin.Vive** provides a simpler API to achieve that.

```
using UnityEngine;
using HTC.UnityPlugin.Vive;

public class GetPressDown_ViveInput : MonoBehaviour
{
    private void Update()
    {
        // get trigger down
        if (ViveInput.GetPressDownEx(HandRole.RightHand, ControllerButton.Trigger))
        {
            // ...
        }
    }
}
```

Figure 4. VIU Scripting [14].

3.4 Version Control

Upon the study on the newly released official API, we started our very first project for testing purpose. The characterised summon feature in our game has been successfully implemented by working around mainly the trigger checking and the event handler to response to an event while the collision between the controller and the virtual object has been detected. This project progress has been located on Github, an online repository for developers to store their work and monitor the working schedule. Github also serves another function, which is to help different contributors to maintain a different version of code by cooperating with a git client software. We adopted Sourcetree for this purpose.

3.5 Feature Implementation

In order to reinforce the uniqueness of Dream Driver, two features will be employed in Dream Driver. They are locomotion with gesture and Dream Driver. The remaining subsections introduce the detail of each feature.

3.5.1 Locomotion with Gesture

To further facilitate the future implementation, another testing scene has been built for evaluating different functionalities of the predefined logic scripts provided in the VR SDK. It includes the basic grabbable logic which once the script has been attached and applied on to an object, the object will react to the VR controller trigger button while the controller is hovering the object. And the follow up response of the object like sticky and unblockable state could also be controlled by other advanced script inheriting from the basic one. Based on these evaluations, the common grab and drag mechanism in most of the VR game could be easily achieved and applied into our puzzle game.

In addition, our originated implementation for the locomotions in VR has been successfully done. To achieve the simulation of floating in the air that people would possibly have their experience in the dream, we found out that ninja run mechanism (see Figure 5) is the most suitable case to accomplish this fantasy as it is a rather natural gesture for people to realise their forward movement without struggling with their hands or feet around the same position. And the work around is theoretically simple. Only the raycaster has been used to capture the direction pointing by the controller (see Figure 6). After that, the players will be translated forward according to the negative value of the ray casting direction.



Figure 5. Ninja Run Locomotion.

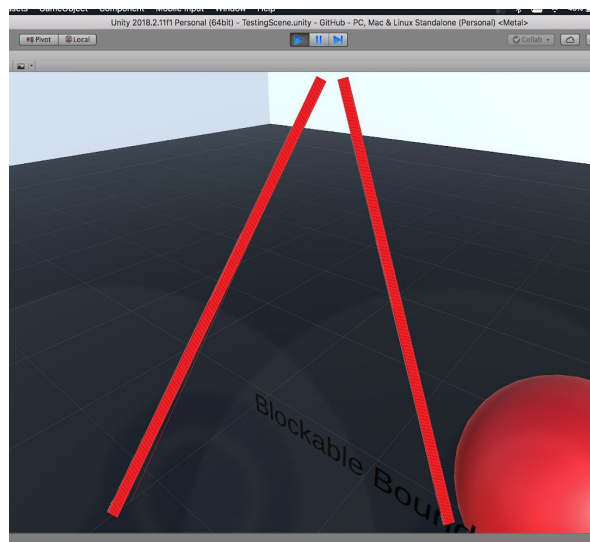


Figure 6. Ray Casting From Controllers.

3.5.2 Dream Driver

Dream Driver is one of our feature system. In the storyline, It acts as a tool for communication, storing magical power and releasing sword skill by consuming magical power. Dream Driver's appearance similar to a watch. It is attached on the player's left hand throughout the whole game (see Figure 7). There is a specific dialogue system mainly focus on presentation of communication (see Figure 8). Before able to release a sword skill, player needs to absorb magical power. In order to achieve this particular function, there are three main logics cooperating with each other. Among these three main logics, energy absorption and summon are being implemented while sword skill will be implemented in the future.



Figure 7. Dream Driver.



Figure 8. Specific Dialogue System.

3.5.2.1 Energy Absorption

In order to activate a particular sword skill, the player has to first collect the corresponding colored energy by placing the colored energy cube inside the corresponding power station (see Figure 9), while different energy will have different ways to obtain. Next, Generator will be charged with energy and activated for absorption (see Figure 10). When the player get close enough to the Charged Generator and hold down the grip button of the left controller for several seconds, the energy will be successfully stored into the Driver and the animation of successfully absorption will be played (see Figure 11).

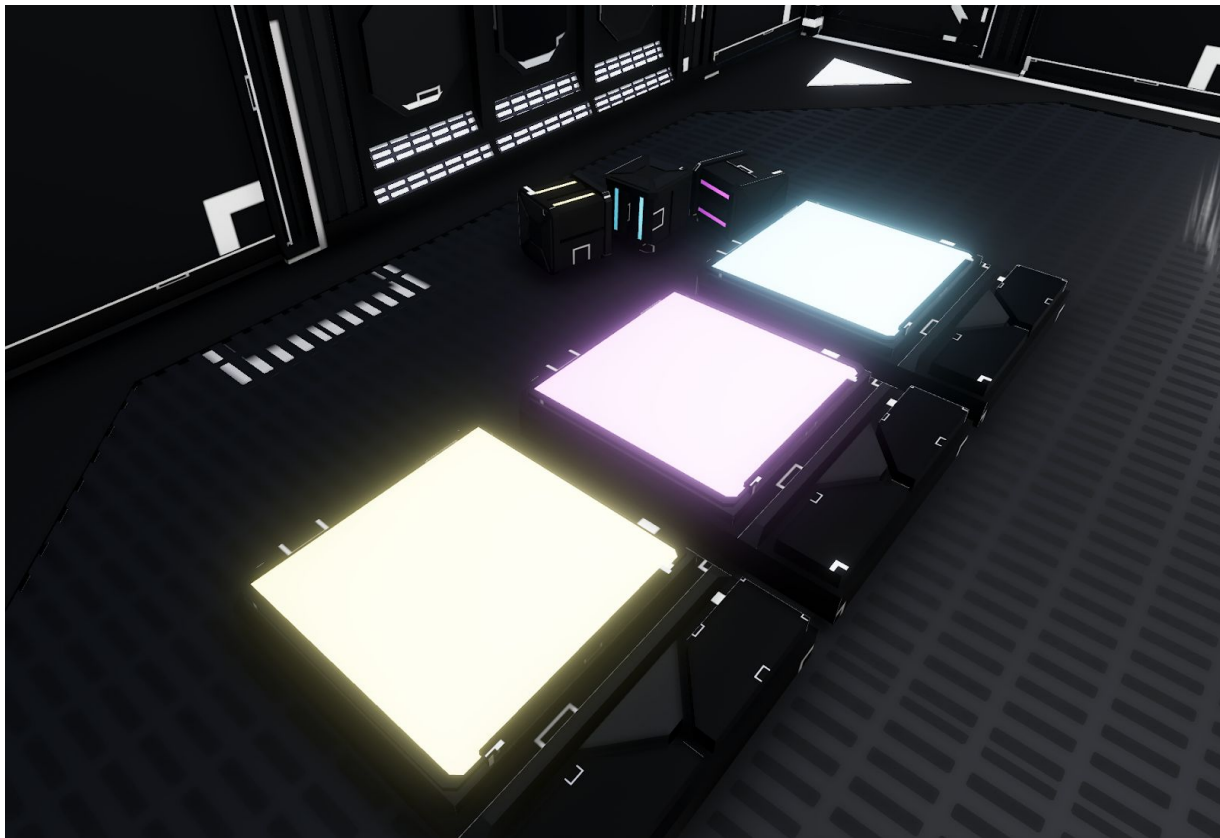


Figure 9. Energy Cube and Power Station.

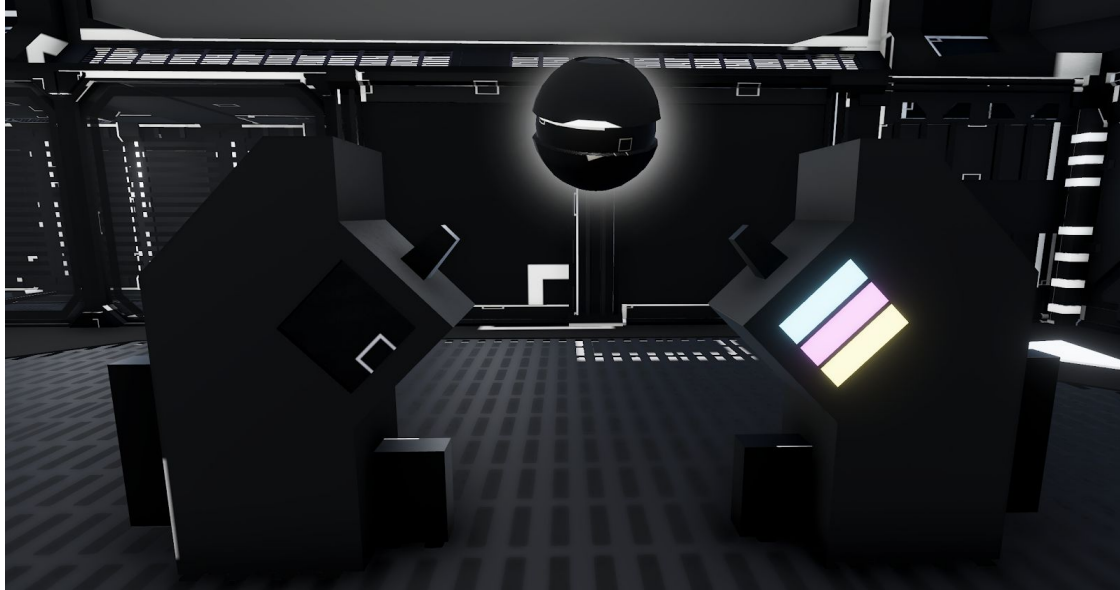


Figure 10. Power Station with Absorbed Energy Order From Blue to Yellow.



Figure 11. Animation of Successfully Absorption.

3.5.2.2 Summon

After obtained the energy, the player will be able to summon a sword with emitting color as same as the color of absorbed energy (see Figure 12). The player can summon the sword by hold down the grip button of the right controller for several seconds. After summoning the sword, the system will activate the corresponding sword skill to release the seal of the room and escape out of here.

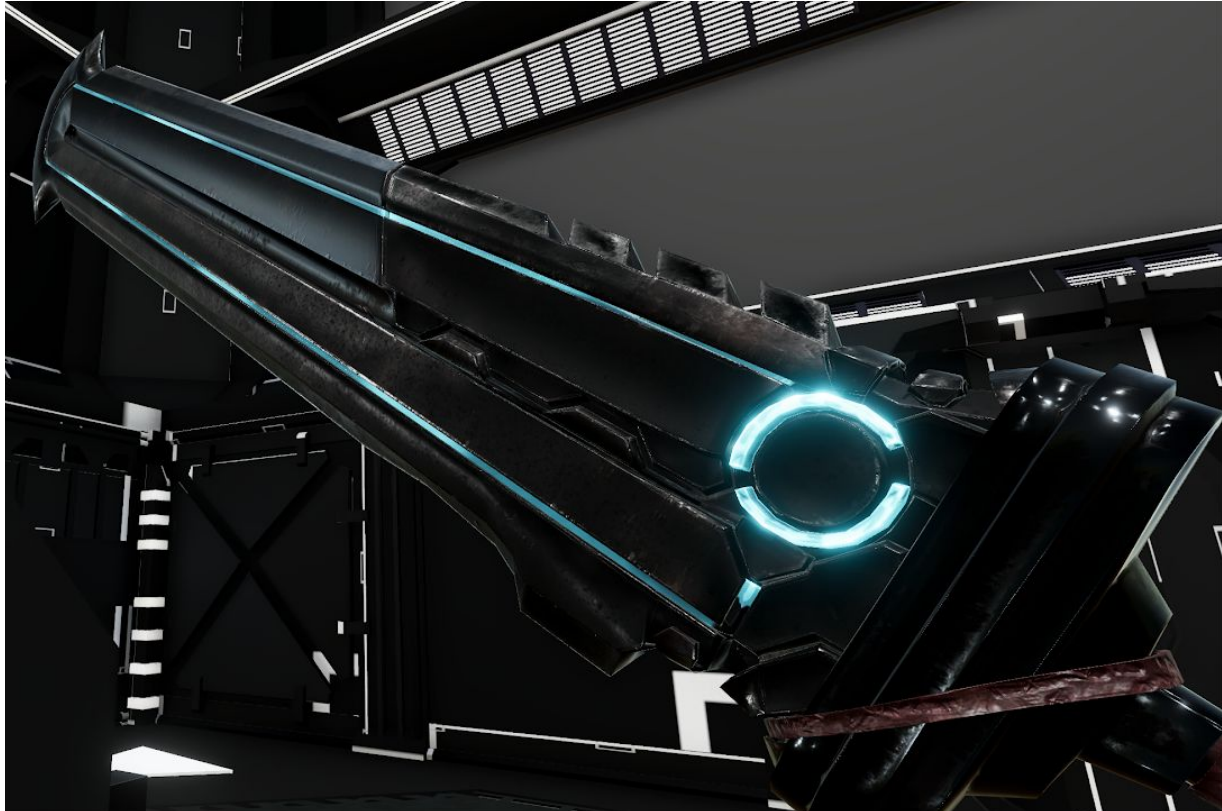


Figure 12. Sword with Cyan Energy.

3.6 Artistic Evaluation

In terms of the artistic work, the art style of our game has confirmed to be low poly. And as long as the asset is suitable, we will stick to what can be acquired in the Unity asset store. Unless for the featuring item in the game, we may have need to spare some more time to prototype the 3D model by ourselves in other modelling software like blender. For our current progress regarding the artistic work, a basic experiment laboratory scene has temporary built (see Figure 13). It will be repeatedly used in every stage throughout the game as to convince the player to follow the fundamental storyline where they are going to solve puzzles level by level to practice their problem solving skill.

Along with the scene building, the lighting of the environment, the technical scene transition, the door open-and-close and the fading animation in between scenes has also been implemented.

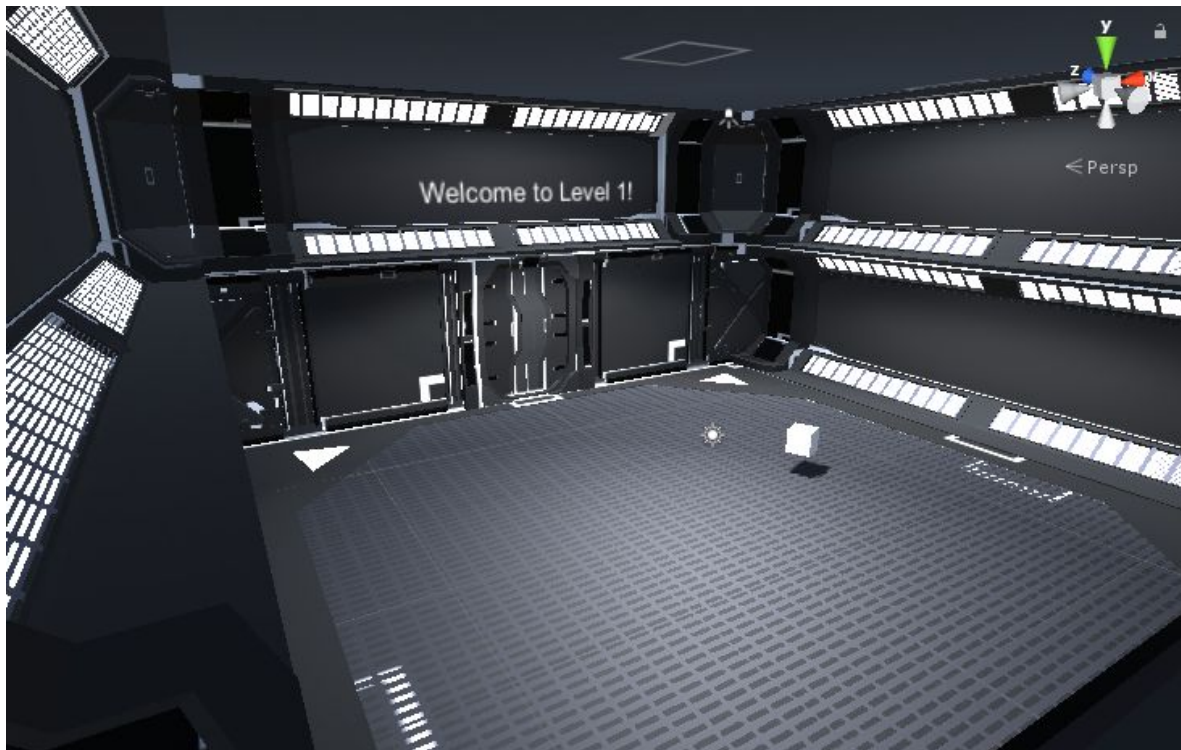


Figure 13. Experiment laboratory scene in game.

3.7 Post-processing

Post-processing is employed to enhance the quality of the visuals of Dream Driver's game environment. It is the process of applying full-screen filters and effects to a camera's image buffer before it is displayed to screen. It can drastically improve the visuals of the game environment with little setup time. We use the post-processing mainly for the color grading and the blooming of the brightness. It benefits Dream Driver because the pantone of our game is being so dark and dull. It acquires players' extra attention with the effect and helps to point out some keys and clues of the puzzle naturally (see Figure 14 and Figure 15).

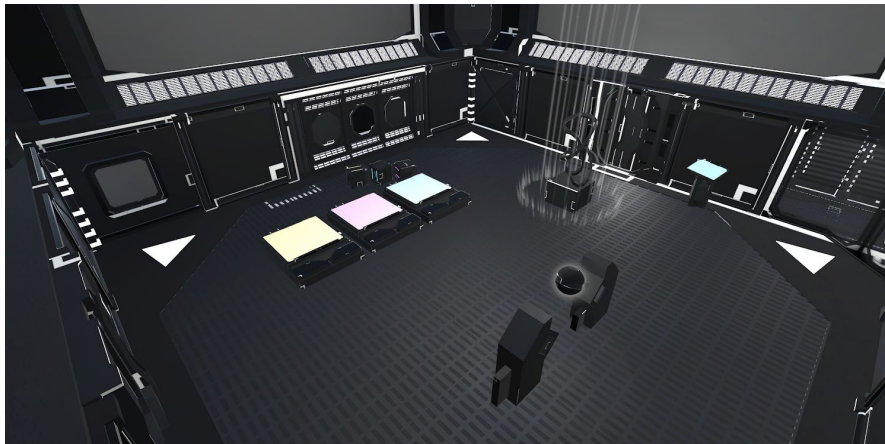


Figure 14. Keys and Clues without Post-processing.

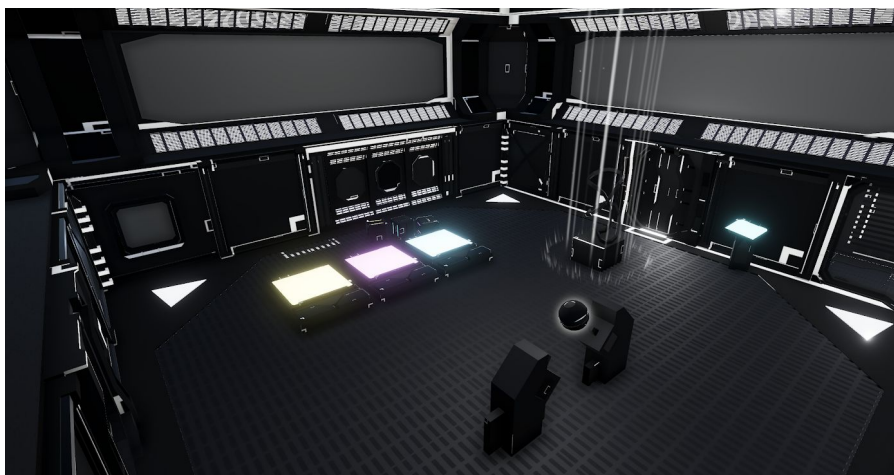


Figure 15. Keys and Clues with Post-processing.

4. Challenges Encountered

This chapter introduces the difficulties encountered while making design decision and dealing with the artistic work. The mitigation methods have also been addressed.

4.1 Design Decision

Gesture recognition is a typical example that we encountered as the major failure of our design decision. As most of our game idea is suggested by the actual experience of being inside a dream, the gesture recognition feature is adopted as a way to simulate the situation when one tries to summon something inside a dream. However, we foresee the implementation difficulties after the study of the API call. Without a further hardware support or fully implemented algorithm, it will be an expensive task to accomplish the recognition process which will probably affect the performance of the game. We decided to hold up the development of this feature until we get some spare time to work on. And the alternative way is to replace with a summon system to achieve this fantasy.

4.2 Artistic Work

A game is a combination of an innovative idea, implementation skill and artistic sense. Unfortunately as an engineer, we do not expert in art. We have to keep the scale of workload in respect of the artwork as small as possible. As a result, we decided to make use of what we could get from the asset store. The basic scene of every puzzle stages will therefore be the same and only alter slightly different by adding a reasonable amount of decorations to present the minimal visual enjoyment.

5. Future Planning

In this chapter, three major steps to finish the game will be discussed. They included the in game features, namely the remaining algorithm and visuals we have to deal with the dream driver, the design of remaining stages and all kind of system involved to constitute as an entire game.

5.1 In-game Features

As refer to our current progress, there are two remaining features of dream driver has to be implemented. The first one is the artistic presentation of the energy order that stored in the device. The ideal model for that will be worked like the following: the stored energy is presented in a diamond form, and arranged in a FIFO order in the Dream Driver. Players can scroll on the touchpad clockwise or anti-clockwise to choose a different energy to use.

And the second feature will be the sword skill. As Dream Driver is able to store numerous of energy, we would like to make use of a concept of combination in some of the level which players have to activate a sword skill with visual effects presented using a specific combination of energy for solving a puzzle. Thus this will involves some visuals and calculations, such as the angle, velocity, collision etc. after the shock wave is released.

A restart level mechanism will be implemented too. As there is quite a few combination of solution but only one correct answer for players to choose from to solve a puzzle during the gameplay. It is possible to have a situation that player will fall into a deadlock, which means they can't have any progress no matter how they continue the game. And the restart level mechanism is good for this purpose, to help our players revert back to a new start level in a convenient manner. The way we are going to implement this is to let our players commit suicide in the game by pointing the sword towards his body.

5.2 Stage Design

The remaining stages left behind will also be scheduled into our future plan. It includes the construction of puzzle setting and solutions which mainly follow three main directions. The first one is our feature system, dream driver. We try to make things correlated on multiple layers. So that it will require our players to consider and think more before their action.

For example, in order to activate a particular sword skill, the player has to first collect the corresponding colored energy, while different energy will have different ways to obtain.

The second one is about space illusion where we will introduce a portal like door on level 2, where players can pass through to another room. And the two rooms are also correlated where some changes make in one room will also noticeable in another room in some sort of format. Just like a parallel world. And the last one is Time which we are still deciding how to express the idea on another level. Last but not least, the dialogue that driven the storyline will also be fully plotted in the coming months.

5.3 System Involved

After finishing up the entire game experience, the next thing to handle will be all kind of the game system, such as the save and load system for players to record his current progress of the game and user interface for players to interact with the game menu, etc. In the script, the function call that invoking sound effect should also be integrated into the game later in case to provide players extra enjoyment.

Conclusion

The final goal of this project is to develop a VR puzzle game named “Dream Driver” which utilize the experience of VR to recall player’s imagination and creativity inside their dream. This progress report introduces the basic background regarding the gaming industry and demonstrates the motivation of working on a new VR puzzle game from scratch. Following with the methodology mentions to develop a game, this paper also states our current progress mainly the preparation work have to be done before starting the implementation of the game, which includes the documentation of GDD, hardware testing and technical studies. We have also decided our mitigation method to the difficulties encountered to clarify our upcoming development decision.

In the coming months, we will focusing on the implementation of the game guided by the tasks specified in the GDD. We will also have the Trello application to record every decision making and milestones achievement. Lastly, the Sourcetree will also help with the version controlling of the implementation contributed by us.

References

1. Laver, K. E., George, S., Thomas, S., Deutsch, J. E., & Crotty, M. (2015). Virtual reality for stroke rehabilitation. *Cochrane database of systematic reviews*, (2).
2. Mujber, T. S., Szecsi, T., & Hashmi, M. S. (2004). Virtual reality applications in manufacturing process simulation. *Journal of materials processing technology*, 155, 1834-1838.
3. Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer*, 38(9), 25-32.
4. Lucas Matney. (2016, Nov 3). Inside Intels race to build a new reality. [Online]. Available: <https://techcrunch.com/2016/11/03/inside-intels-race-to-build-a-new-reality-2/> [Accessed: Jan 16, 2019]
5. HTC Corporation (2018). Vive virtual reality system. [Online]. Available: <https://www.vive.com/us/product/vive-virtual-reality-system/> [Accessed: Jan 16, 2019]
6. SAMSUNG ELECTRONICS CO., LTD. (2018). Samsung gear vr with controller. [Online]. Available: <https://www.samsung.com/global/galaxy/gear-vr/> [Accessed: Jan 17, 2019]
7. Sony Interactive Entertainment Inc.(2018) PlayStation vr. [Online]. Available: <https://asia.playstation.com/en-hk/psvr/> [Accessed: Jan 19, 2019]
8. VALVE Corporation. (2018) VR: Vacate the Room. [Online]. Available: https://store.steampowered.com/app/494810/VR_Vacate_the_Room/ [Accessed: Jan 17, 2019]
9. Smith, A. M., Andersen, E., Mateas, M., & Popović, Z. (2012, May). A case study of expressively constrainable level design automation tools for a puzzle game. In *Proceedings of the International Conference on the Foundations of Digital Games*(pp. 156-163). ACM.
10. FIGHT4DREAM LIMITED. (2016) FIGHT4DREAM. [Online]. Available: <https://fight4dream.com/> [Accessed: Jan 20, 2019]

11. Rouse, R., & Illustration-Ogden, S. (2000). *Game design theory and practice*. Wordware Publishing Inc..
12. Indraprastha, A., & Shinozaki, M. (2009). The investigation on using Unity3D game engine in urban design study. *Journal of ICT Research and Applications*, 3(1), 1-18.
13. Hejlsberg, A., Wiltamuth, S., & Golde, P. (2006). *The C# programming language*. Adobe Press.
14. HTC Corporation. (2018). HTC VIVE Tracker (2018) Developer Guidelines. [Online]. Available:
[https://dl.vive.com/Tracker/Guideline/HTC_Vive_Tracker\(2018\)_Developer+Guidelines_v1.0.pdf](https://dl.vive.com/Tracker/Guideline/HTC_Vive_Tracker(2018)_Developer+Guidelines_v1.0.pdf). [Accessed: Jan 20, 2019].