Virtual Reality App

Mohammadreza Farhaditoumaj

C1670504

Supervisor: Dr Yukun Lai

MSc Advanced Computer Science

School of computer science and informatics

September 2018

NOTES OF GUIDANCE ON THE COMPLETION OF THE NOTICE OF SUBMISSION FORM

You should complete both sides of the form in typescript or black ink and use block capitals. The completed form must be handed to your Head/Dean of School (or his/her nominee) with two copies of your dissertation and (where appropriate) the examination fee.

Each copy of the dissertation shall contain:

1. a summary not exceeding three hundred words;
2. a statement signed by the candidate showing the extent to which the work submitted is the result of the candidate's own investigation, and an explicit acknowledgement (with references) of any other sources used;
3. a full bibliography;
4. a signed declaration to certify that the work submitted has not been accepted in substance for any degree or award, and is not being submitted concurrently in candidature for any degree or other award;
5. a signed statement regarding availability of the Dissertation;
6. a completed 'Notice of Submission' form.

The declaration and statements referred to in 2 to 5 above should be incorporated at the beginning of the dissertation, as shown in Appendix 1.

Where this is in accordance with the School's policy on the submission of Taught Master's Dissertations:

- candidates may submit their work for examination in temporary binding;
- candidates may be instructed by the Schools to submit one of the two required copies in an approved electronic format.

APPENDIX 1: Specimen Layout for Declaration/Statements page to be included in Taught Master's Degree Dissertations

CANDIDATE'S ID NUMBER	c1670504
CANDIDATE'S SURNAME	Please circle appropriate value Mr/ Miss / Ms / Mrs / Rev / Dr / Other please specify ..... Farhaditoumaj
CANDIDATE'S FULL FORENAMES	Mohammadreza

DECLARATION

This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed ..... (candidate) Date 7/9/2018

STATEMENT 1

MSc Advanced Computer Science

This dissertation is being submitted in partial fulfillment of the requirements for the degree of .....(insert MA, MSc, MBA, MScD, LLM etc, as appropriate)

Signed ..... (candidate) Date 7/9/2018

STATEMENT 2

This dissertation is the result of my own independent work/investigation, except where otherwise stated.
Other sources are acknowledged by footnotes giving explicit references. A Bibliography is appended.

Signed ..... (candidate) Date 7/9/2018

STATEMENT 3 – TO BE COMPLETED WHERE THE SECOND COPY OF THE DISSERTATION IS SUBMITTED IN AN APPROVED ELECTRONIC FORMAT

I confirm that the electronic copy is identical to the bound copy of the dissertation

Signed ..... (candidate) Date 7/9/2018

STATEMENT 4

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... (candidate) Date 7/9/2018

STATEMENT 5 - BAR ON ACCESS APPROVED

I hereby give consent for my dissertation, if accepted, to be available for photocopying and for inter-library loans **after expiry of a bar on access approved by the Graduate Development Committee.**

Signed ..... (candidate) Date 7/9/2018

Abstract

Virtual Reality (VR) has been used in many different fields, as there are available cheap solutions for users to experience it. In order to explore the navigation and interaction in the virtual reality 3D world in an affordable way, this project aimed to create the scenario using solutions with cheap and accessible equipment. This project focused on developing a Virtual Reality game in shape of the maze with Google VR and Google Cardboard, using Unity3D as the Game Engine. The study illustrates the design, development process, and the logical theory behind the idea of virtual reality maze game, and the stages where testing the application. Seven healthy participants (20-61 years old) were recruited to test this game and to answer the feedback questionnaire. As a result, participants had positive experience during the game in general, and this application is suitable for several Android devices.

Acknowledgment

I am so grateful that I had a chance to study in Cardiff Universty and I appreciate the helps and guidance of all professors and staffs during my studies.

I would like to thank my supervisor, Dr Yukun Lai. He gave me valuable guidance and support throughout the process of my dissertation.

I sincerely thank my family that they support me mentally and financially and gave me this opportunity to follow my dreams.

Table of Contents

Table of Figures and Tables:

Table 7: question 7 **43**

1.Introduction:

With the daily improvement and development of technology, every day we are observing the new solutions to reduce the cost of the productions that make it affordable for users to have access to them. This project looks at the Virtual Reality that it has been used in different fields of industries, such as health and education, and there are low-cost solutions available for it. This project demonstrates the experience of developing a Virtual Reality application using Google VR SDK and Google Cardboard and using free programs to achieve this goal.

It demonstrates that how it is possible regarding the scenario of the application how a user can navigate in the 3D environment and it demonstrates the different ways that can use in the Virtual Reality to interact with the 3D world.

The objectives of this project demonstrate in a Virtual Reality Game in a shape of a maze that developed with Unity3D, that user can experience the mechanics by solving the puzzles and trying to reach the end of the maze while using the Google Cardboard, that it has limited options for the user to interact with the application.

There are different types of interactions between the user and the application, especially in the VR, such as objects that user can grab them and place them in different areas, or the interaction that user can have his/her senses, such as hearing senses.

The targeted audience for this project, are the beginner developers and the students and researchers in the university that are willing to produce Virtual Reality and they are considering the platforms that they want to develop.

2.Background:

2.1. What is Virtual Reality?

Virtual Reality has different definitions one group believe it is a combination of particular technologies: Display that that mount on the head and audio and glove for input data and other group define Virtual Reality as imagination and fantasies that used in books and movies (Isdale 1993).

By looking at the words, “Virtual” means almost or nearly described and “Reality” which it means the things that actually exist, so it is an environment that creates a world that is close to similar to an actual real world.

2.2. Technical view:

Virtual reality is a three-dimensional environment that built by a computer, Which the user by using relevant devices can interact with that three-dimensional world.

2.3.Types of Virtual Reality:

At the moment they are four types of the VR:

1.Textural VR (interaction, no immersion)

2.Desktop VR (interaction, immersion)

3.Immersive VR (interaction, high immersion)

4.Augmented VR (interaction, no immersion)

Augmented reality or AR that is a non-immersion VR is the combination of the reality and the computer data, nowadays can find the simple examples of it like the filters on the mobile apps that effects the user’s face image or adds a 3d model to the real environment.

The Immersive VR is that user become deeply interactive with the virtual world and it uses the computer components (Burdea and Coiffet 2011).This project is implemented in the form of Immersive VR.

2.4.Use of Virtual Reality:

There are different fields that use Virtual Reality, such as medicine, education, military, Architecture, entertainment such as gaming, movies. And in each of them it has its own important effect, for example in medicine can use it as Virtual Reality Therapy where the patients experience the VR environment in immerse or non-immersed that recreate the responses similar to the ones in real world that helps the patients to control it based on difficulty and complexity of the program by engaging in lots of activities patients become interested in the entertaining environment and helps them to improve their focus and compliance in the therapy. Or in a different method, VR helps as a pain relief for the patients by distracting them (Romano 2005).

VR has an important role in future of the education because in education, students can learn the subject by learning the theory and exploring it in a cheaper environment and that gives lots of opportunities to schools to find innovative ways to improve the student's skills.

In this all different fields, we can see that the reasons they are using the VR is because it is more affordable, in medical uses it reduces a considerable amount of fees. However, these fees can be much cheaper especially if we are looking at education, not all schools can afford VR devices and applications to use for their learning and teaching purposes.

2.5.VR devices:

Virtual reality requires a device or combination of devices to provide the experiment to the user, In the market can find a variety of models from different brands and most of them are quite expensive. Normal VR device is a headset that contains a display and lenses and controllers

2.6.Google Cardboard:

The aim of the project is implementing the VR app on a device using the cheapest solution, using an android mobile phone for this project gives options like Google Daydream and Google Cardboard. Google Cardboard is the basic and cheapest model to use, and it follows the purpose of the project which is finding low-cost solution.

Google Cardboard as it comes from its name, is built with cardboard, it uses two lenses and a button.

2.6.1.How does it work?

Cardboard uses the screen and all related functions of the smartphone such as sensors, sound, a microphone for running the application in Android and iOS platforms. It uses Biconvex lenses and a magnet button as a controller. When the button moves, it is equal to touching on the touch screen, the magnet button works with the magnetometer of the phone (gearbrain, 2015)

2.6.2.The difference of the Daydream and Cardboard:

Daydream is the other device that Google provides for VR, it provides more quality comparing to the Cardboard, it has a headset and a controller, it tracks the rotation and orientation in higher accuracy comparing to Cardboard, the headset has the adjustable band that makes the user be more comfortable(Unity Technologies, 2018). And by comparing the price, based on the Google store it is about 99 pounds. But regarding accessibility, the Cardboard is easier to access, the user can download the plan of the Cardboard and build by him/her self.

the photos in below demonstrate the Cardboard that used in this project, it is suitable for a mobile device with a size up to 6.1 inches, it does not have the support for AR but with some adjustment on it, it is possible to create a place to uncover the mobile phone's camera to use for AR purposes.

Figure 1: Google Cardboard

Figure 2: Goolge Cardboard from side

2.7.Tools for creating Virtual Reality:

Creating a VR application needs a framework, there are two ways to create Virtual Reality, one method is using a 2d professional platform like ArcGIS. And the second method is using the 3d or 2.5d software as a platform, Skyline software is one of the available programs that can create VR with it. It is possible to create to create VR application with Game Engines too. The game engine is a combination of frameworks that provide the developer's different resources that they need regarding the product that they are producing. In a VR application for creating the interactions there are different sources that

must be considered, and a game engine provides them all. There are some different game engines available, the most common game engines that they are free, and developers use it are Unity3d and Unreal Engine 4(Unreal Engine, 2018), both provide incredible supports in the cases that developer needs to build.

2.7.1.Google VR SDK:

Google provides the Daydream and Cardboard for using and experiencing the Virtual Reality, also they provided SDK which it is the solution for creating VR applications for Android and iOS devices, it is compatible to use with Unity3D and Unreal engine.

2.7.2.Unity3d:

Unity is a cross-platform Game engine that it supports development for Windows, MAC OSX, Xbox One, Play Station 4, Android and iOS and some others. it is possible to script C#, JavaScript before 2014 could use Boo language too but they removed it from the framework. it supports .NET libraries and the compiling time of the scripting languages are as fast as C++ (Wang et al. 2010).

2.7.3.Which version of the Uity3D?

In this project, Unity3d is used to create the application and C# used as the scripting language but it was a huge challenge to find the correct version that is compatible with Google VR and the Android SDK and JDK.

First Google VR 06 tested with the latest version of Unity 2017, until the time that I tried to build the application, and Unity gave more than 30 errors in the different fields. After researching and testing, downgraded the Unity to 2017.2.3 f1 version and use the GVR SDK for Unity v1.150.0 and downgraded the JDK to jdk1.8.0_181 managed to build the application.

For the scripting, Visual Studio 2017 used which it is compatible with the Unity in the Windows Operating system.

3.Approach:

3.1.Design:

This virtual reality application designed to demonstrate the navigation and interaction in a 3D environment, for demonstrating it, it needs to have a scenario and a story, for this purpose whole project designed in a VR game application which it has one main level with starting a scene and a finishing scene.

The maze designed to make the player experience different interactions while using the movement method in the 3d environment using the Google Cardboard, the player has to solve the puzzles and face the challenges to reach the final scene.

For developing this application must consider, the stages of the development, the main stages are the parts that created in Unity3D and the parts outside of it (Wang et al. 2010).

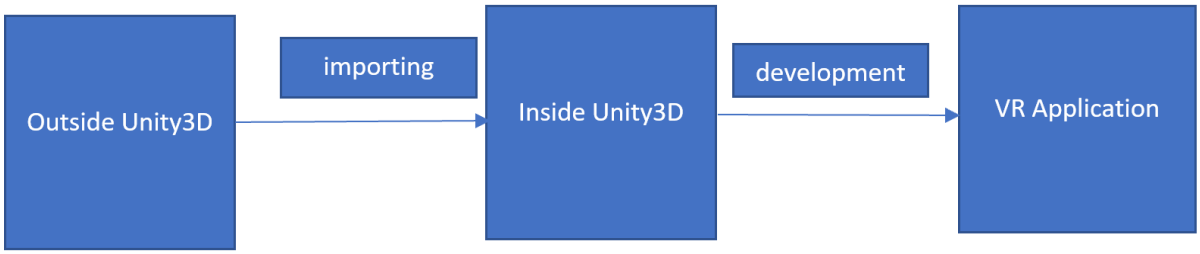

Figure 3: design based on Outside and Inside of Unity3D

3.1.1.Outside of Unity3D:

There are different parts that designed and created outside the Unity3D, for example I some projects they use 3D studio max and photoshop to create the art assets such as textures and 3D models, for this project there are some textures used and a 3D model that they are built for Unity3D and they were available in the Unity Store, Unity Store provides the models and the assets need for creating any application related to Unity3D with different prices. These assets that used for this project are one snowy mountain and the sky, that the designer designed the shaders for them too, these assets used for starting scene and finishing scene, and a 3D model for a key with the related texture skin, and texture for the grass and walls used for the maze level. These assets were all free.

Another asset that created outside of the Unity3D is the soundtrack that created and recorded in the Music Maker Jam in Android operating system with the Ogg format.

3.1.2.Inside the Unity3D:

For creating some parts, for example, the movement, it needs to create scripts with C#, and Unity3D in Windows operating system works with the Microsoft Visual Studio 2017 for creating the scripts. Scripts and animations and the GameObjects like walls, door, plane, buttons are created inside the Unity3D, and some of them like the plane they got merged with the grass texture from outside the Unity3D.

The map of the maze initially designed with paper and pen that eventually created and shaped inside the Unity3D.

3.2.The concept of the application:

There are three scenes in this application:

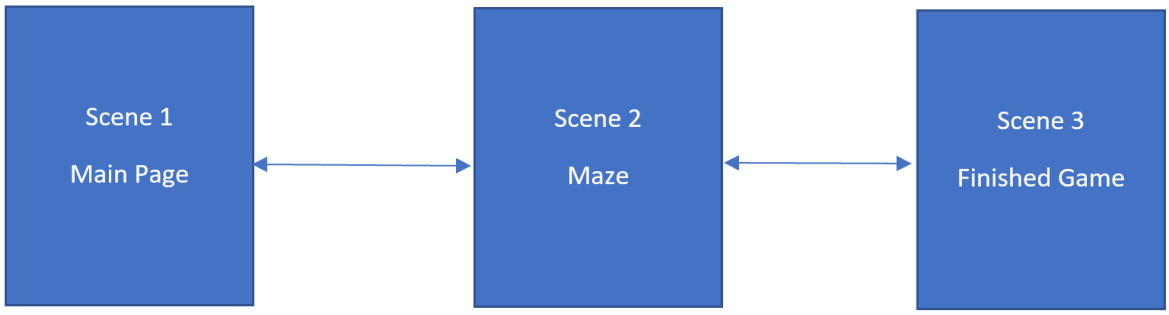

Figure 4: Concept of Application

This application has three scenes, the first scene is the main page that user chooses the level that wants to play, the second scene is the playable part, and the final scene shows that user finished the game.

3.2.1.Scene 1:

This scene is the Main page that allows the user to choose the level that he/she wants to play, and it gives a tutorial to make the player to be familiar with the mechanics of the application such as how to interact with options and how to move forward.

3.2.2.Scene 2:

In the second scene, all mechanics such as movement and drag and drop are used, the player has to walk and find the way out, the instructions that provided in the environment helps the player to find the solutions and the way out.

As mentioned before a maze designed for this scene that this maze contains the puzzles and obstacles such as:

doors that user has to open them, one needs a key to open which player has to find the key. Cubes that blocks the way or by moving them can pass the obstacle. and there is the soundtrack that player must listen to it to find the direction. Use of Soundtrack can be an interesting point for users that have issues with eyesight so can create the applications or games that they are just based on the sounds.

3.2.2.1.Why use the soundtrack to find the direction?

In lots of computer games such as first-person shooters game, the sound plays an important rule, that helps the player to locates and predicts the movements around him, for example in the computer game called Counter-Strike: Global Offence, players can hear each other's movements so if the enemy is shooting in one side of the map, based on the volume of the sound, player can find out, how far is the enemy and which direction is he/she is coming from.

So, use of soundtrack in the VR for finding the direction brings the idea of creating applications that helps the people that who has the disability in the vision, like if a person suffers from blindness, in the entertainment field, can create a game that challenges the player to choose the right sounds from different directions and turns around based on it, and with VR user does not need to use keyboards or gamepads and while player is standing up, can turn around by facing the right direction that sounds come from can score points.

Overview of the maze:

The maze has a start point and a finish point, and in the middle, the obstacles are positioned that player must pass them.

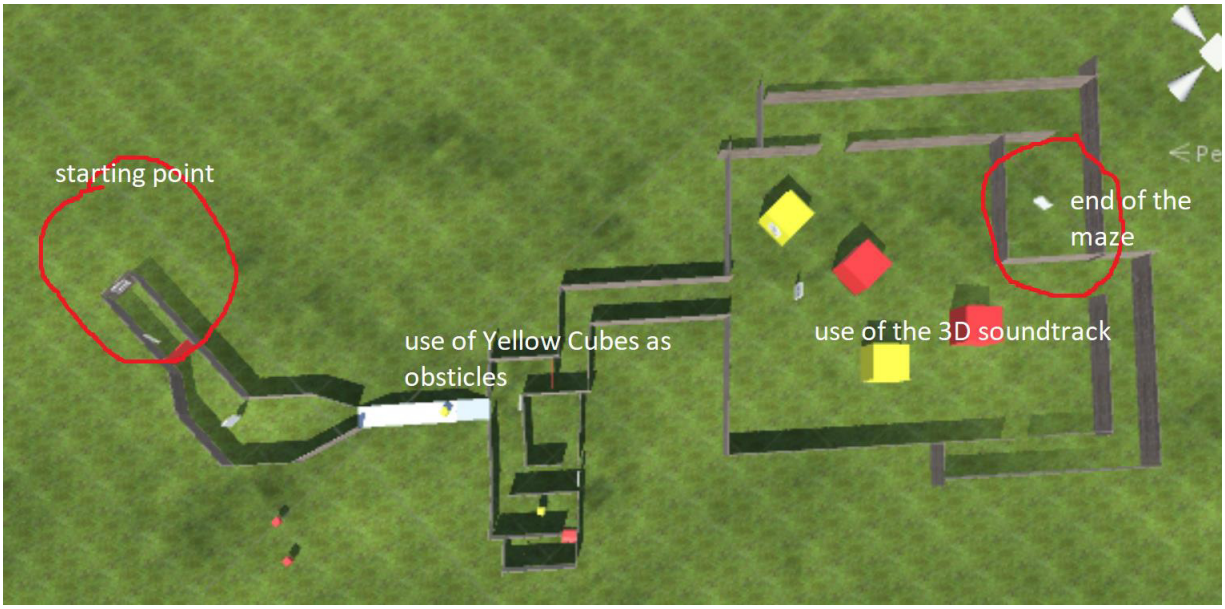

Figure 5: view of the maze

3.2.3.Scene 3:

This scene is the final scene that gives the message to the player that he/she finished the game and it gives the option to restart the scene 2 and plays it again.

4.Implementation:

4.1.Navigating or movement:

Moving around in the 3d world especially in computer games happens with using WASD keys or arrow keys that user by using the mouse can choose the path and move around the environment or on a gaming console using the analogs on the controllers. In the virtual reality, the scenario is different. We do not have access to the keyboard and while we are using the Google Cardboard we only have the magnet button. Navigating in the Unity is moving the camera around the environment, it is like the camera is fixed but the environment moves and gives this feeling that we are moving the camera. For making movement there are two methods, that can call them automatic movement and angle movement, with the automatic movement each time user presses the magnet button camera starts to move and whenever press again it stops, the other one instead of using the button user uses his own head, it means that when the user holds his/her head at a specified angle, he/she can move; and when he/she puts the head in the normal position the camera stops moving (NurFACEGAMES, 2016).

By looking at the design which it is a maze and must interact with other objects. It would be too much to use the same button to do everything. And it will cause confusion in the player and have to press the button several times for different purposes. The second method which using the specific angle for head that makes the camera moves gives the user opportunity and time to look around and then moves to the decided direction while still have vision on front and by playing for few times user can understand the mechanic better and perform better.

4.1.1.Creating the movement:

In the real life when we look at the objects we are using our eyes that they are located on our head, so in the Unity we use the Camera as the eyes, an empty game object created for it and named head and it became parent of the main camera, and it locates on the top of the head.

Figure 6: head

And the main camera is the parent of the GvrReticlePointer and hand, the hand is related to the interaction will discuss it later.

GvrReticlePointer is a function from Google VR SDK, it provides the dot in the middle of the screen that is the gaze of the player, so where ever it points means player is looking at that direction and would interact with the object that is on the way of it.

Head is the main game object that deals with the interactions and another event in the whole application. On below it demonstrates the components that are related to the movement.

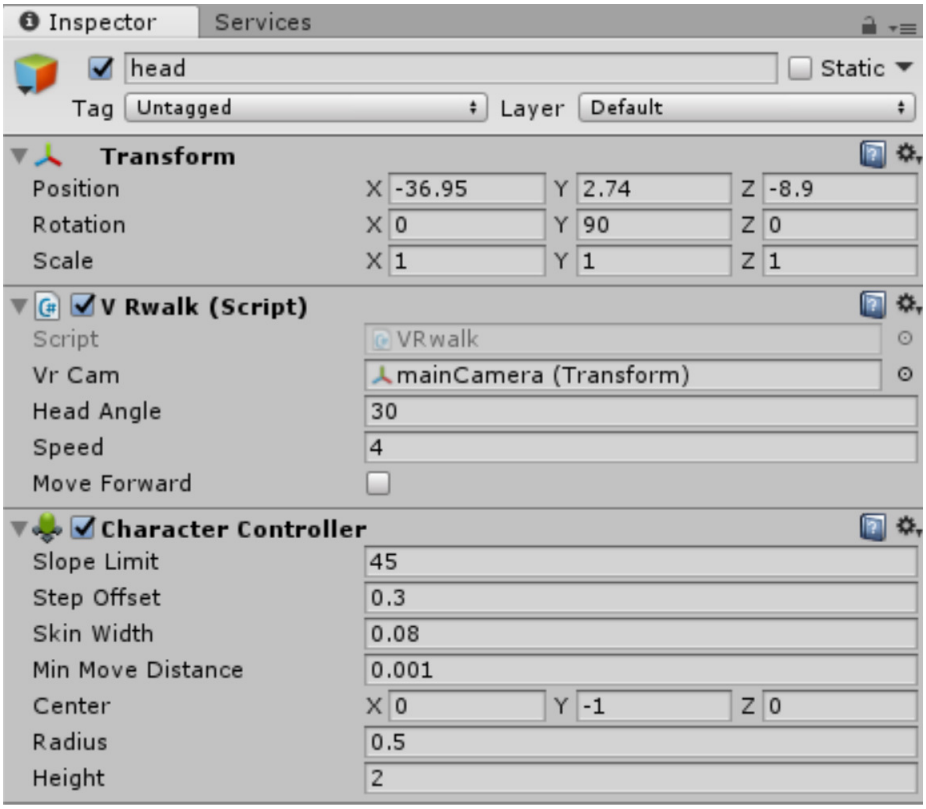

Figure 7: head's components

As you can see the for the movement head has two components, VRwalk which is the script and the Character Controller from Unity, Character Controller is mainly used for third person and first person controller in the games, and while in Virtual Reality our vision is from the main camera, it is the first person. Character Controller has some options that can help to optimize the movement, for example here the Slope Limit is 45 degrees, it means the head can climb the Slope or ramp that the degree of them are less than 45. Or Step Offset is the height that the character can climb. Higher than 0.3 is impossible to climb. Radius is the amount of area around the character that can collide with the objects. And height is the height of the capsule which camera is on top of it that gives the feeling of the human body that the head is on top. The height here is set to two.

Before we discuss the script, Google VR provides another function that by using that character can start looking around by turning his/her head around. This function is GvrControllerMain and can use anywhere in the scene. The other controller that used in this project is the GvrEditorEmulator that helps the developer to test the application with a mouse and keyboard with the computer that he/she is using for creating the application. So for the basic controlling in the environment which it is just looking around, we can use GvrControllerMain.

In the components, the VRwalk is the name of the script that makes it possible to walk around in the environment. In Unity, in the current version, it is possible to use C# and JavaScript. All scripts used in this project are written with C#. in Unity3d, MonoBehaviour is the main class that is the parent of all components. The default C# script uses three namespaces:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

Figure 8: namespaces

UnityEngine is one main namespace that contains most of the useful classes such as gameObject. A default empty script contains two functions:

void Start() and void Update()

Start: when the user reached to the related frame, and the script got triggered, Start runs before the Update function and it only runs one time (Unity Technologies, 2018).

Update: this function after the script got triggered, would run every frame.

4.1.1.1.How does the VRwalk work?

The logic for the movement is by looking at the angle of the camera to walk and stop, so for this part, we need to check the camera's Euler angles in x-axis constantly. a statement in the Update function that checks the Euler angles between two degrees, in this project used 30 and 55, so while the angle of the camera is between 30 and 55 degrees it starts to move forward.

```
if(vrCam.eulerAngles.x >= headAngle && vrCam.eulerAngles.x<=55.0f )
{
    movement = true;
}
```

Figure 9: VRwalk eularAngles statement script

In Unity, there are 3 types of vectors, vector2 that can store the value of X and Y in the 2d environment, and vector3 that stores x, y, and z. vector2 and vector3 have more use comparing to the vector4, vector4 store x, y, z, and w. in 3d math fourth component uses for position or represent a vector (hou.Vector4. 2018). for making the movement possible vector3 is enough.

VRCam is a Transform, the Transform makes it possible to store and make changes in the position, rotation, and scale of objects (Unity Technologies, 2018). The purpose here is to find the direction that leads the user to move forward, so for finding the forward direction have to use Transform.TransformDirection(Vector3.forward)

It means from local space it converts the direction to world space.

And VRCam is the Transform so it would be like this:

```
Vector3 moving = vrCam.TransformDirection(Vector3.forward);
```

Figure 10: vector3 moving script

Vector3.forward represents Vector3(0,0,1).

After finding the location the only thing is left is making it move. By using a character controller and SimpleMove function where we use the initialized speed (4) and VRCam. Character Controller only moves when a move function called, so move function here is SimpleMove.

```
if (movement == true)
{
    Vector3 moving = vrCam.TransformDirection(Vector3.forward);

    characterC.SimpleMove(moving * speed);
}
```

Figure 11: Simple Move script

All these stages implement in the Update() function which needs constantly to rerun to check the status of the camera's angle to make it to move or not.

As you can see by using a script and other components inside the Unity3d managed to start moving around in the 3D world in the first-person shooter mode. Can walk around and stop observing the area when decided the direction walk toward it just by toggling the head without using other tools.

4.2.Interaction:

In this environment, user looks around and observes the objects and hear the voices inside the environment and make decisions based on it. Some objects can be moved and help to solve some mini puzzles and move forward or hears the voice and find the direction to it, or looking for the hints and read the messages that give guidance to reach to the goal.

In the playable level there are movable cubes and doors and the menus, player starts interacting with them when the small dot that acts as a pointer in the middle of the screen is on the object, if the object is interactable the pointer become bigger and by pressing the button on the Google Cardboard it starts the interaction.

4.2.1.Drag and Drop:

One main interaction in the environment is when a player can effect on the status of an object in the 3D world, by changing the status means changing the position of it, it can happen by picking the object and move or drag it around the place.

In this application the only object that can move and carry around the maze is a yellow cube, they are two cubes in the maze that one of them can use as a step and another one is blocking the way that needs to put it out of the way. For making this possible let's look at the concept of picking up an object by a human in the real world:

1. First, human observes the object

2. Then decides to pick up the object
3. Approaches to the object
4. By using the hand, he picks the object up

By looking at the Figure 12 and 13, can see that in a 3D world when the character picks up the yellow box how does it look like:

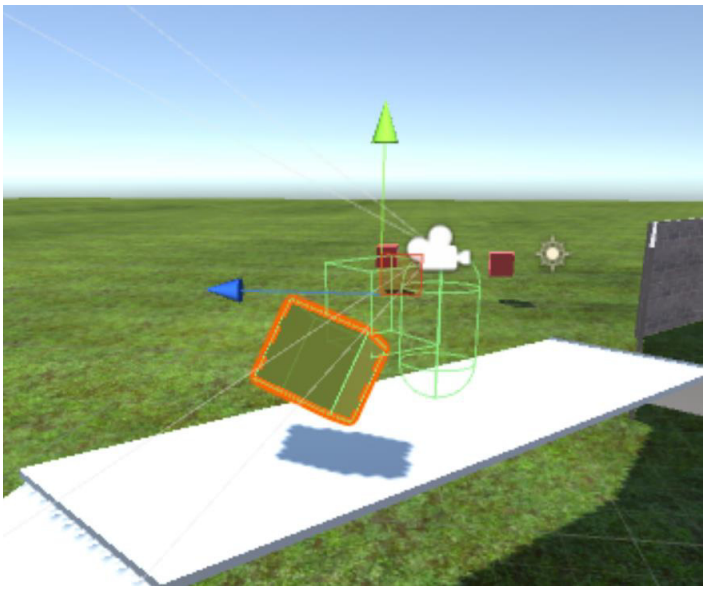

Figure 12: grabbing yellow box 1

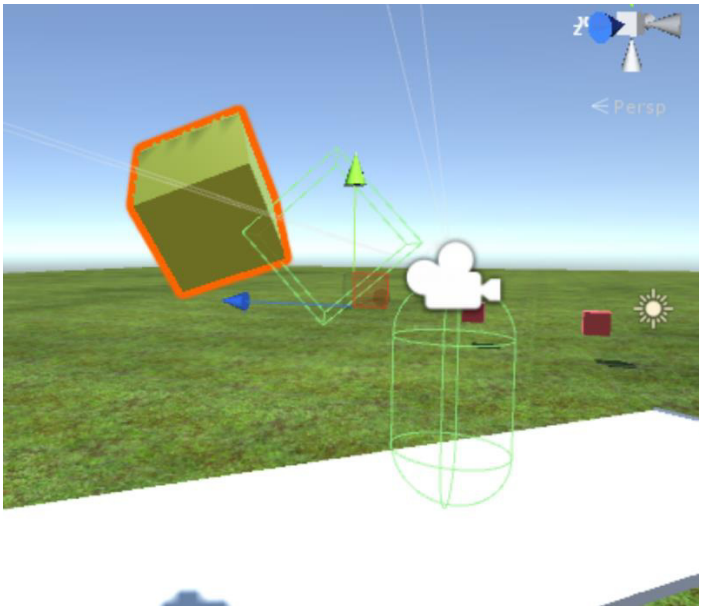

Figure 13: grabbing yellow box 2

The character has a body that camera is on top of it as the head and a cube front of it works as the hands and it holds the yellow cube.

As you can see it goes through interaction between GameObject that named “head” and the yellow cube.

4.2.1.1.Yellow Cube:

Yellow Cube made from two cubes, one is transparent and another one has the colour of the yellow.

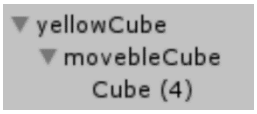

Figure 14: YellowCube

In both movableCube and Cube, there is some component that used in both cubes, Box Collider and Rigidbody.

4.2.1.2.What is Box Collider?

Collider components are the shapes that are invisible, depending on the use of them, they can be in same size and shape of the object’s mesh or they might be bigger than the size of the actual size of the object, these components are used for the physical collision in the 3D world. Box Collider contains Physical material and Triggers (Unity Technologies,2018).

Physical material:

Physical material instead of effecting on the appearance of the game object it effects on the way game object reacts to the physics engine, for example if there is a car moving on the road, it would have different reaction when it moves on a road covered with ice, like it has different frictions between those surfaces, or when a ball fall on two different surfaces it can bounce differently.

Triggers:

With the function OnCollisionEnter on the Unity it makes possible that with scripts detects the collisions, by using Is Trigger option when the main character or other related objects enters the collision area it acts as trigger and makes the action or response happen, by activating the Is Trigger the object loses the physical behavior, it means it does not act as a solid object.

Both movableCube and Cube use the Box Collider, but they use it differently from each other, movableCube is the cube that has the yellow colour on it, and Cube is transparent, it is like movableCube is the cover or dress and Cube is the inner Cube.

4.2.1.3.Why use two Cubes instead of one?

as mentioned before the yellow cube is the object that the player can pick it up and hold and move around, so need to make it possible that it collides with the player. By adding the Box Collider component to the cube, it makes it collide as a solid object to the player, but for making it possible to pick it up, it has to be Is Trigger. By Using Is Trigger it is possible to make the script the cube loses the solid shape of it and for example when the user wants to walk pass without interaction, cube instead of blocking it, it lets the user go through it. So, for fixing this issue solution was to use two cubes one as a trigger and another one to give solidity.

4.2.1.4.What is Rigidbody?

This component allows the game objects to react to the physics, for example, if game objects want to move it needs a Rigidbody to mention if gravity effects on it or it wants to float in the air.

Rigidbody has Mass and Drag and Angular Drag, which can choose the proper one regarding to the need of the project, They are two options, Use Gravity and Is Kinematic, Use Gravity give gravity to the object it means if the object is floating in the air it will fall and stand on the surface, Is Kinematic cancels all forces, here in this project for making it possible to use the yellow cube as the step, Is Kinematic used, and position of it is freeze in x-axis (Unity Technologies, 2018).

4.2.1.5.Script:

playerGrab.cs is the script for this process of picking up the box and moving around, as mentioned before head which is the player has a hand, and hand is going to grab the yellow box, the logic for the script here is that by clicking on the yellow box, yellow box become child of the hand, and hand is child of the main camera, so when the main camera moves, yellow box will have the same position as hand as a child of the main camera and moves around with the hand.

The script uses the normal Unity namespaces and does not need any additional namespace for this scenario. The namespaces are:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

Figure 15: namespaces for Drag and drop script

In the script, considered two game objects that they called oBject and hAnd, so oBject is for yellow cube and hAnd for the hand and used a Boolean for checking that if the object is grabbed or not and called it grabbing.

```
public GameObject oBject;
public GameObject hAnd;
bool grabbing = false;
```

Figure 16: GamObjects scripts, drag and drop

Then also needs to connect the game objects inside the Unity itself, this script gets placed for the head(the player character).

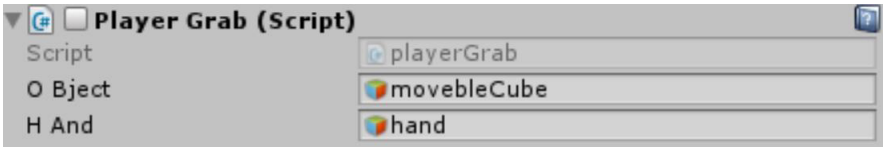

Figure 17: Player Grab in Unity

for grabbing the cube, an interaction between user and device needed, and this interaction happens by pressing the magnet button, which it is equal to touching the screen, name of the button in Unity scripts calls “Fire1” for managing to call this have to use the Input class, this class is an interface for the input system, while the interaction is by pressing the key down so need to choose a function that does it, GetButtonDown makes it possible to use the magnet button to use and use this a statement that while the button is pressed the other processes happen and while it has to check in every frame that if the button is pressed or not, it has to be used in the Update() function (Unity Technologies, 2018).

```
if (Input.GetButtonDown("Fire1"))
```

Figure 18: GetbuttonDown script

So, based on the logic that the parent of the yellow cube changes to the hand, the method transform.SetParent, and between global orientation and local orientation it keeps the local orientation, and this method uses the worldPositionStays which it is a Bool and sets to false, by calling the SetParent the worldPositionStays it becomes true, the parent-relative details(position, scale, rotation) get modified and the world space details of the object stays the same(Unity Technologies, 2018).

Then set the transform.localPosition of the yellow cube equal to hand’s transform.localPosition.

```
if (!grabbing)
{
    oBject.transform.SetParent(hAnd.transform);
    oBject.transform.localPosition = hAnd.transform.localPosition;
    grabbing = true;
}
```

Figure 19: changing parent of the Yellow Box scripts

At this point, the yellow cube can be grabbed and held by the hand of character, after this it needs to have an option to release the cube in the area that player wants it to drop. The logic of this part is to get the last position of the yellow cube and release the cube at that point. for this part, the solution is to disable the component which is the script by pressing the button on Google Cardboard. So, need to call the component and set it to false.

```
this.GetComponent<playerGrab>().enabled = false;
```

Figure 20: this.GetComponent script

And after that set the yellow cube’s parent to null that release the cube on the area that player wants to be.

```
void Update () {
    if (Input.GetButtonDown("Fire1"))
    {
        if (!grabbing)
        {
            oBject.transform.SetParent(hAnd.transform);
            oBject.transform.localPosition = hAnd.transform.localPosition;
            grabbing = true;
        }
        else if (grabbing)
        {
            this.GetComponent<playerGrab>().enabled = false;
            oBject.transform.SetParent(null);

             grabbing = false;
        }
    }
}
```

Figure 21: Drag and Drop Update function scripts

GvrEventSystem:

In Unity3d uses Event System that uses inputs from input devices (Keyboard, mouse and …), Event System as the manager has some roles that they are:

1.Manage which GameObject is considered selected

2.Manage which input module is in use

3.Manage Raycasting

4.Updating all Input Modules as required

Event modules are for dealing with the input and managing the event state and sending the events to scene objects(Google VR, 2018).

And the use of Rayscasters is that to find out where the pointer is looking at or which GameObject is it pointing.

In the Google VR instead of using Event System, the GvrEventSystem is available which it has the Event System's components and the GvrPointerInputModule. Without GvrEventSystem it is not possible to make all the interactions that designed and scripted on above to work. For making it work properly need to use event trigger, which it activates the whole interactions with the objects when the pointer goes on the object or exit the object, after setting the related GameObject to it, it gives the option to enable the

desired the function that makes the wanted process working.

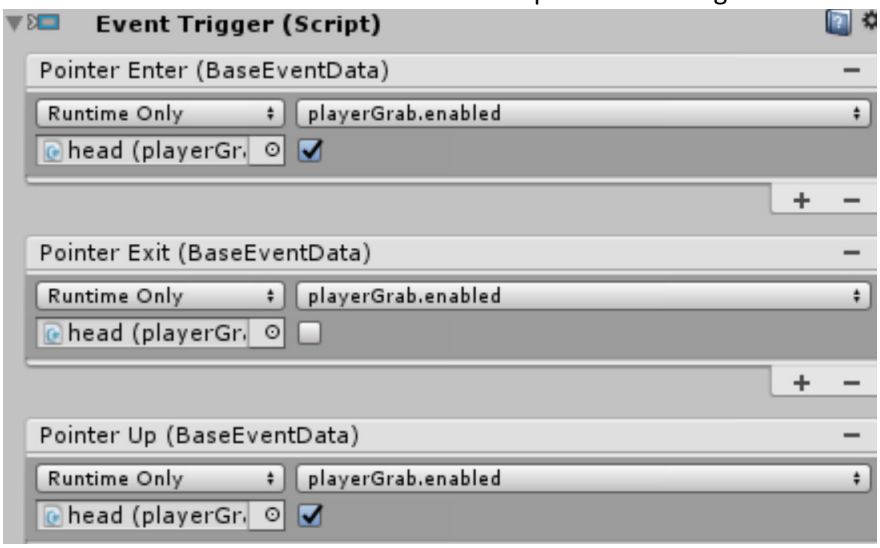

Figure 22: Event Trigger

GvrReticlePointer sends the signal as the input and with GveEventSystem and the playerGrab.cs script can drag and drop the yellow cube.

4.2.2.Key and the door:

This interaction is between the main character and two other objects. A key and a door, which player has to find the key inside the maze, and after collecting it, can open the door, to access other parts of the maze. The components that used in this part are, two scripts one for the process of picking up the key and another one is for opening the door, and for opening the door the component that used is the animation. Then box collider which it used on both key and door.

Door:

DoorHinge is the parent of the door, Doorhinge contains the animation and the Box Collider.

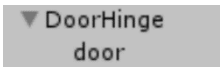

Figure 23: DoorHinge

Key:

the key is the object that the player needs it for opening the door, it uses a 3D model in shape of key and it called pPlane3, it has its own textures and colour settings.

Figure 24: key

Key has a Box Collider and Event trigger as the components, the Event trigger uses the pickUpkey script which is the component of the head, and whenever the player points at the key in an accessible distance to it, the script becomes activated.

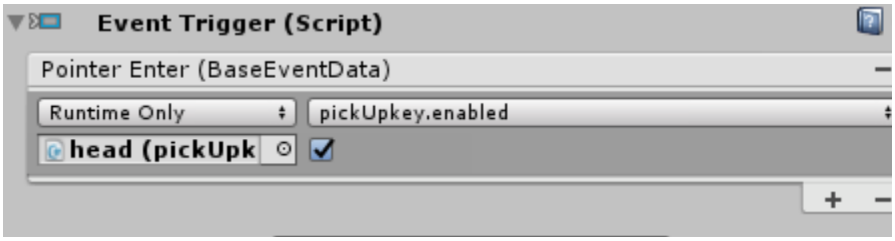

Figure 24: event trigger of key

4.2.2.1.Animation:

Unity3d has Animation Window Guide(Unity Technologies, 2018) that helps to create animation with the GameObjects and modify other available animations (Unity Technologies, 2018). For example, in some games, there is an animation that they make for showing the whole level at first, which it's by moving a camera around the level and recording it. For the door, the animation is simple so, with the Animation Window Guide just recorded the rotation of the door and with the timeline chose the proper length for the animation time.

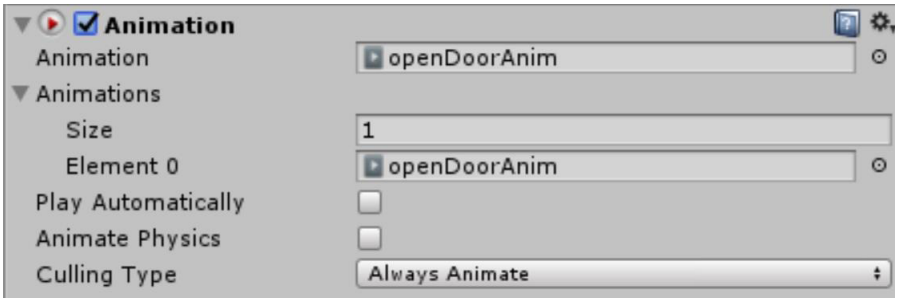

Figure 25: Animation

The animation uses the openDoorAnim that created with the Animation Window Guide, it would not play automatically, because if it plays automatically then it would not wait for the key to being collected and opens by itself and it has to be activated with the script.

4.2.2.2.Scripts:

As mentioned before there are two scripts for this part, pickUpkey.cs and OpenDoor.

pickUpkey:

this script is in relation with the other script which is the OpenDoor, the logic of this script is when the player reaches the key after collecting it, by pressing the magnet button, the key disappears, and it activates the OpenDoor.cs which it lets the player open the door.

It uses the standard namespaces:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

Figure 26: key and door namespaces scripts

There is only one public GameObject used and it named DisapearKey.

```
public GameObject DisapearKey;
```

Figure 27: key and door GameObject scripts

The rest of the script was used in the Update() function one reason is that the method GetButtonDown used in it, and it only works in the Update() function and also it keeps need to check in every frame that if the player has the key to open the door or not.

There are two statements one for enabling the OpenDoor.cs which it is:

```
if (Input.GetButtonDown("Fire1"))
{
    this.GetComponent<OpenDoor>().enabled = true;

}
```

Figure 28: Key and door statement 1 scripts

It uses the GetButtonDown which it allows to use the magnet button or the touch screen of the phone, so, if the button is pressed then it calls the other component which it is OpenDoor class and it enables it.

The other statement makes the key disappear after collecting it from the area.

```
if (Input.GetButtonDown("Fire1"))
{
    DisapearKey.SetActive(false);
}
```

Figure 29: Key and door statement 2 scripts

It uses the same approach as the previous statement for using the button, and it follows the GameObject.SetActive(false) which GameObject here is DisapearKey and the method SetActive it uses for activating and deactivating GameObjects in the Unity3d by setting it to false it makes the key to being disappear after interaction with it.

This script is the component of the main character which is head and the GameObject DisapearKey is matched with the actual GameObject in the maze which is the key.

Figure 30: head's component, pickUpkey

4.2.3.OpenDoor:

This script is responsible for playing the animation of the door, so when the animation plays, the door opens and stays open to make it possible to go through.

It uses the standard namespaces:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

Figure 31: OpenDoor namespaces scripts

There is one component used here which it is an Animation and it called HingeHere:

```
public Animation HingeHere;
```

Figure 32: OpenDoor GameObject scripts

As mentioned before the Animation component can manage the animations like playing them with the script.

Rest of the script is based on the Update() function and it uses one statement that it checks if the button is pressed it will play the animation.

```
void Update()
{
    if (Input.GetButtonDown("Fire1"))
    {

        HingeHere.Play();
    }

}
```

Figure 33: OpenDoor Update Function scripts

The Play is a Bool that uses in the form of the Animation.Play. It plays the animation without mixing it with other components.

This script is the component of the DoorHinge and the HingeHere is matched with the animation called openDoorAnim.

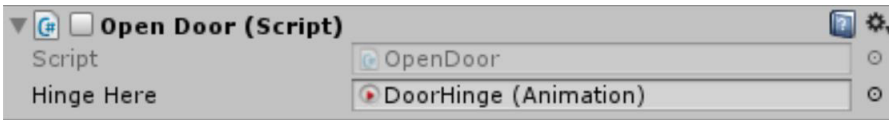

Figure 34: head's component OpenDoor

This part had the interaction between three GameObjects and used animation and scripts, in the maze, there is another door in the beginning that it uses a different animation but it uses the same script which is the OpenDoor.cs that it opens the door by pressing the button without using a key.

4.2.4.Soundtrack:

the soundtrack designed and recorded in another application called Music Maker Jam on android, length of the track is 1 minutes and 17 seconds and it is in Ogg format. After importing inside the Unity, it used as a 3D sound in the level two. The way it works is it has a radius that is possible to hear, while the user goes closer to the sound, it become louder by looking in a different direction while using headphones is it clear that which direction the voice is coming from.

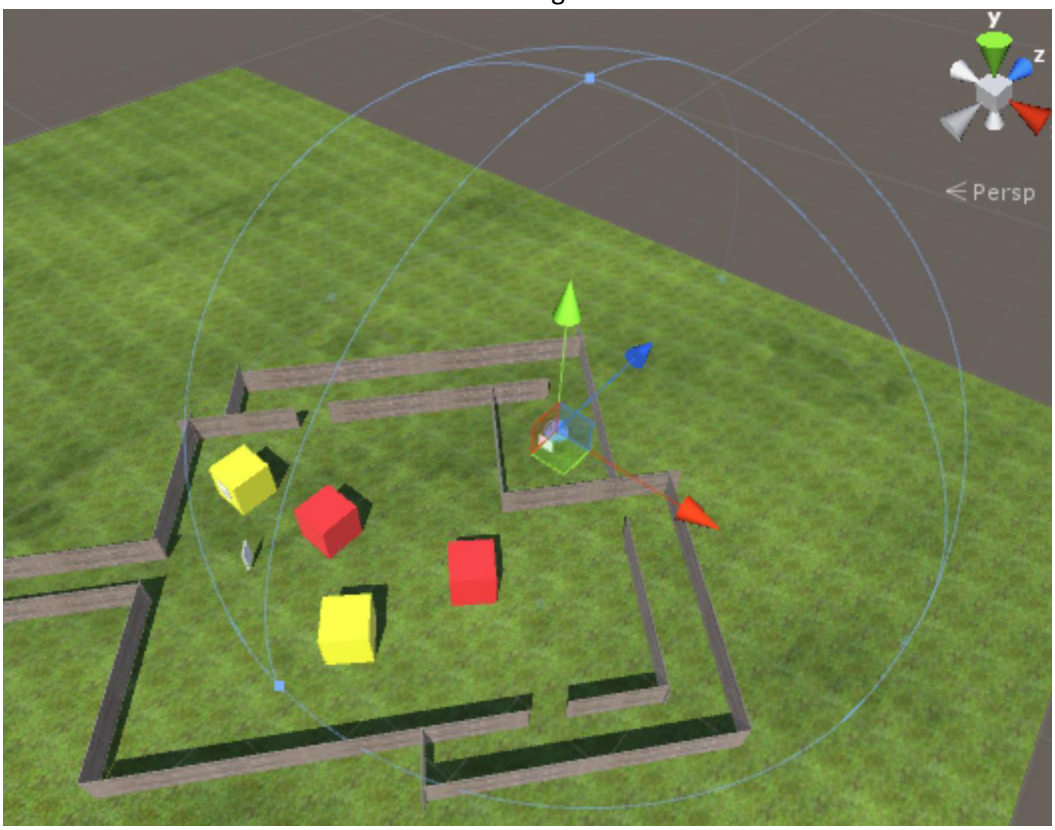

Figure 35: Soundtrack

based on the radius for making the sound has a different amount of volume Unity has an option called Volume rolloff that has to set to Linear Rolloff.

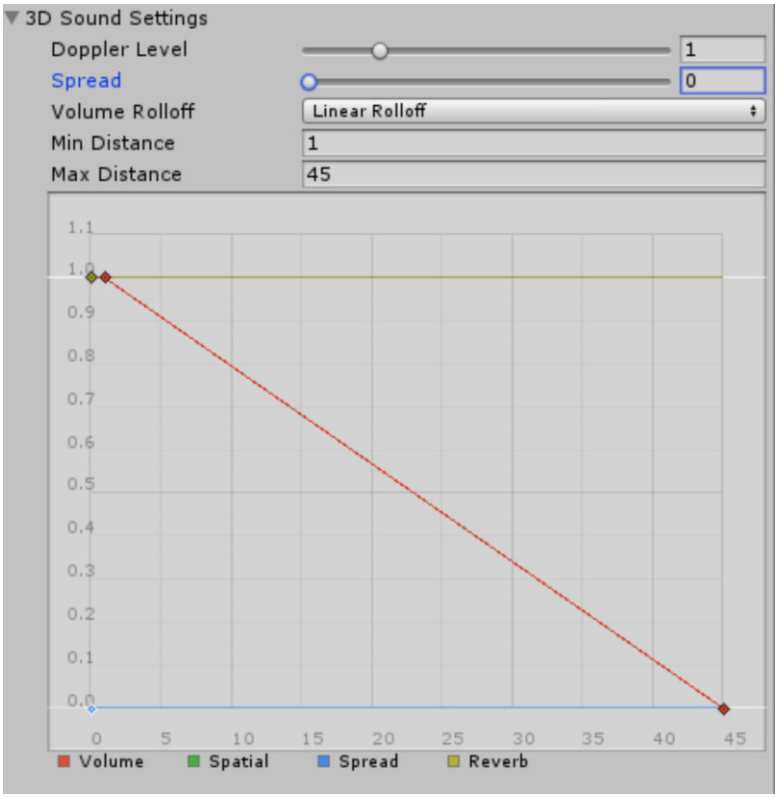

Figure 36: Soundtrack setting

As you can see it is possible to hear the sound from a distance of 45 and it becomes louder when user become closer.

UI:

User interface is another way of the interaction between user and the application, in computer games we usually have HUD which is located on top of the screen, with Unity3d for android the story is different, at the moment with the latest Google VR that used for this application, Android devices can not have a HUD, however by using the Google Cardboard the focus on the screen is on more on the middle of the screen and if there is a note on top of the screen it is going to be blurry for the user, so, tutorials and instructions that usually place on top of the screen in most applications are in the world space for this application instead of screen space.

4.2.5.Menu:

In Virtual reality application decided that to put the menu in the actual environment instead of separating it from the scene, it means for using the menu no pause would happen and while the user is wandering around the environment can interact with the menu.

How does it work?

In the menu, it gives the options to the user that can decide to quit to the main menu or restart the level.

The menus are located in the different locations in the maze, for example at the beginning of the level if the player wants to quit the level he/she needs to look behind and see the menu, the idea of it comes from that when in the real world a person enters a room if he/she wants to go out of the room has to turn around and open the door and leaves. Other ones are in areas that player might need to restart the level.

The process of creating the menus are so simple for instance the scripting is just a few lines but in the menu there are buttons each button has its own script for example in the first menu in scene two that it just has quit button which by pressing the button user goes to scene one, it uses the script called QuitButton.cs inside this script uses UnityEngine.SceneManagment namespace.

```
using UnityEngine.SceneManagement;
```

Figure 37: Menu namespaces script

Then for selecting the scene must use the class SceneManager, this class has different functions that help in loading and creating the scenes, for example, there is one function called GetSceneByName that it searches the list of scenes based on the given name. the solution for selecting the scene which at this point is the scene one which it named menu, function LoadScene is used for (Unity Technologies, 2018).

```
void Start () {
    SceneManager.LoadScene("menu");
}
```

Figure 38: menu, start function scripts

Because of loading the scene, the function just needs to execute one time, which locates in Start() function and uses the name of the actual scene to load it(“menu”).

The creating of the physical shape of this menu that you can find it in figure 39 in below, uses mix of the game objects.

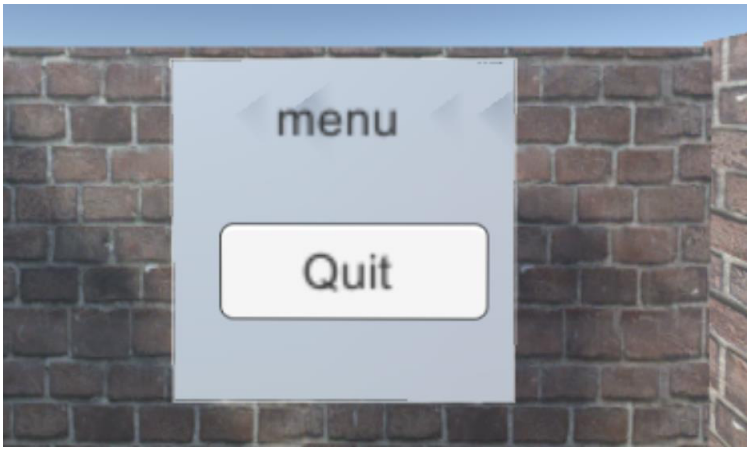

Figure 39: menu 1

Figure below demonstrate the GameObjects that uses in the menu.

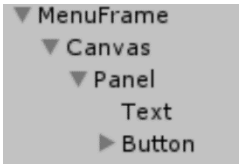

Figure 40: MenuFrame

The frame which its called MenuFrame is a normal cube in Unity. Then there are the UI game objects. Canvas is the type of the Game object that all the UI components such as text, Button and others have to be Canvas's children. Then it is the Panel that can give the background colour to it, which is Grey here. The Panel has two children: one is the Text which it contains text "menu"; the other is Button. The script QuitButton.cs is assigned to it.

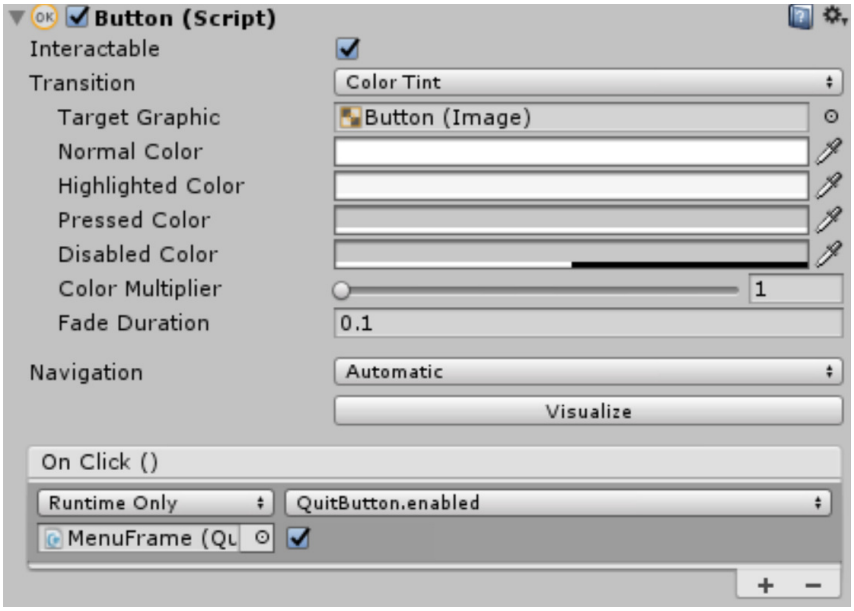

Figure 41: Button

The other menu follows the same pattern, but it has an additional button that gives the option to restart the level, the script for restarting is a separate file which it named RestartButton.cs but follows the same way as the quit button. It uses the same namespace UnityEngine.SceneManagment and same way for calling the scene, it calls the same scene that it is currently in it and makes it start again.

```
void Start () {
    SceneManager.LoadScene("dissertation");
}
```

Figure 42: restart, start function script

In the figure below (Figure 43), you can see that the menu is in different location, and it has two buttons which are restart and Quit.

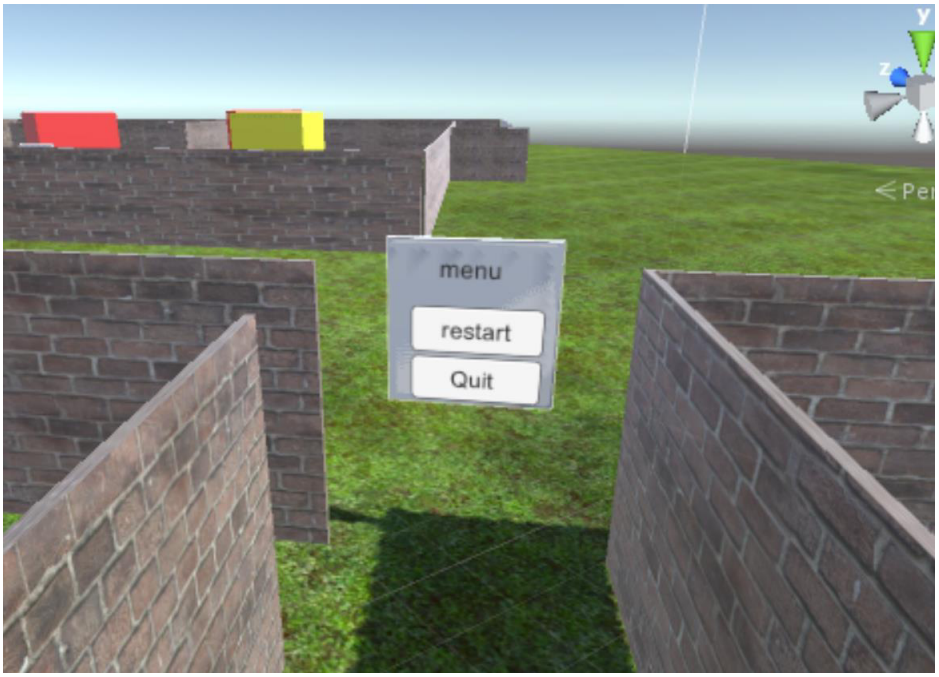

Figure 43: menu 2

4.2.6.Hints:

The other way of interaction is the hints that they are available in the maze, they are the message boxes that are giving hints about the mission that the user has and helps the user to achieve the goal.

Figure 44: hints

4.2.7.Tutorials and Instructions:

For making the users be familiar with the movements and the mechanics in the games and applications, some notes have to be provided for them to guide them how to use this application. In this VR application, in scene one and scene two, tutorial and instructions provided that helps and remind the player how the movements and interactions work.

The same way used in both scenes for creating this interface, the GameObjects that used are the UI objects that there mentioned below in the figures():

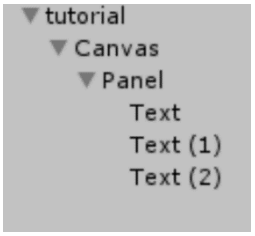

Figure 45: tutorial

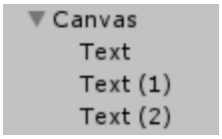

Figure 46: instruction

The first one is from scene one and the second one is for scene two. As you can see, Canvas is used for both of them. In the first scene, because of the colour of the background, the panel is used to help the texts to be more readable for user.

The Canvas Render Mode is set to world space, and the text is available whole the time for the player that by looking at the sky player can observe it.

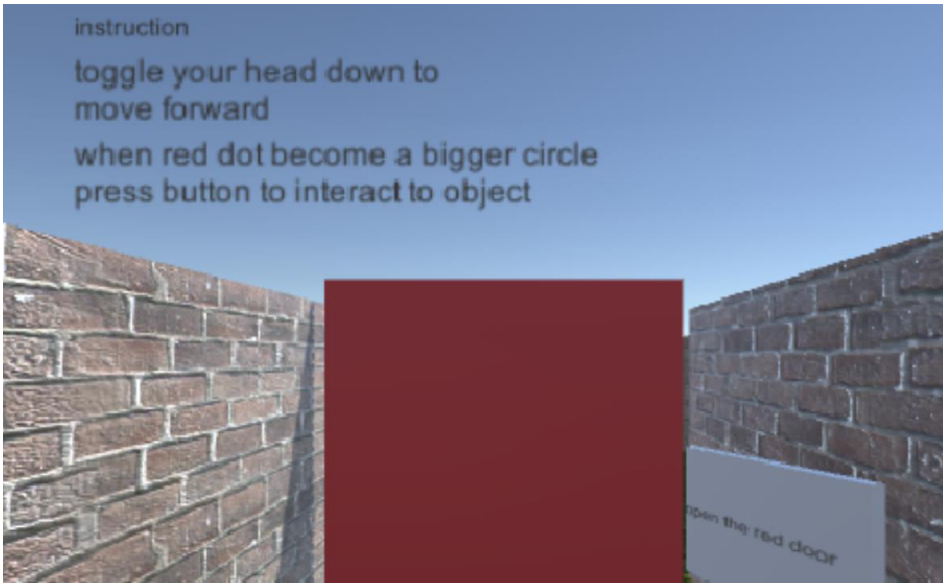

Figure 47: Scene 2 instructions

Figure 48: Scene 1 tutorials

5.Results and Evaluation:

For reaching the final version of the application for this project, testing was one of the important aspects of the progress in different stages, and it helps to measure the functionality of the application. The total time that spent on the testing was equal to 50 percent of the time spent on the whole project.

The diagram below demonstrates the process of the building the application:

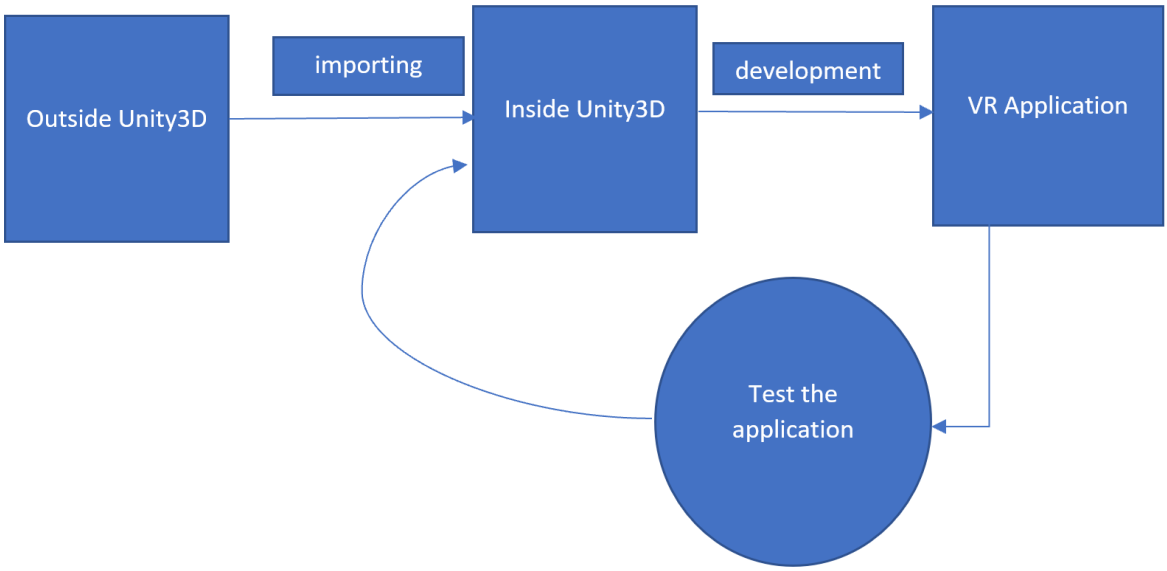

Figure 49: development diagram

5.1.Testing:

Application test with the emulator:

This stage was during the production, in which Unity provides the emulator that could test the mechanics to see how they are performing, however there were lots of differences in the performance of the mobile phone and the PC, for example it could test the frame per seconds with the emulator but it would not work on the android device especially for Cardboard.

5.1.1.Testing with Android device:

By testing the application with the simulator on the computer found that there is a gap in performance between computer and Android device, So, after testing the mechanics at each stage on the emulator, a testing version of the application was built for the mobile device to find the errors and bugs.

5.1.1.1.Performance test on Android device:

Because computer has more power comparing to the mobile device, so the speed of performance and the quality is higher. For this project, the application tested on four different Android mobile phones to see is there any problem with different hardware.

After running the application on four Android phones: Google pixel 2 xl with android 8.1.0, Samsung Galaxy s9 and Samsung Galaxy s8, both were using the Android 8.0.0, and OnePlus 5T with Android 8.1.0.

The purpose of the first observation was to see if the application works smoothly on all these devices, regarding to their specs and installed Android version.

All the devices managed to run the application and quality was smooth after they loaded the application and there was no pause between switching the scenes and other mechanics. The only difference was the speed of the performance that it depends on the hardware of the smartphone.

Speed of performance:

A stopwatch used for this part and the stopwatch counts the time from running the application until the first scene appears on the screen, this test performed on the Oneplus 5T and Samsung Galaxy S8.

OnePlus 5T uses 6 gigabytes of ram and Samsung Galaxy S8 uses 4 gigabytes of Ram. The difference of the ram might have a significant impact on the performance. The application has been tested for five times on these two devices and the durations are recorded. The results are listed below:

OnePlus 5T:

1. 5.79
2. 5.73
3. 5.79
4. 5.75
5. 5.78

The average time of the performance is 5.768

Samsung Galaxy S8:

1. 9.3
2. 9.1
3. 8.80
4. 9.1
5. 8.86

The average time of the performance is 9.032

After viewing this result on OnePlus 5T and Samsung Galaxy S8, Performed another test with a different VR application called Gravity Pull VR that it built with Uity3D too on the OnePlus 5T.

Gravity Pull VR on Samsung 5T:

1. 5.7
2. 5.96
3. 5.94
4. 5.95
5. 5.86

The average time of the performance is 5.882

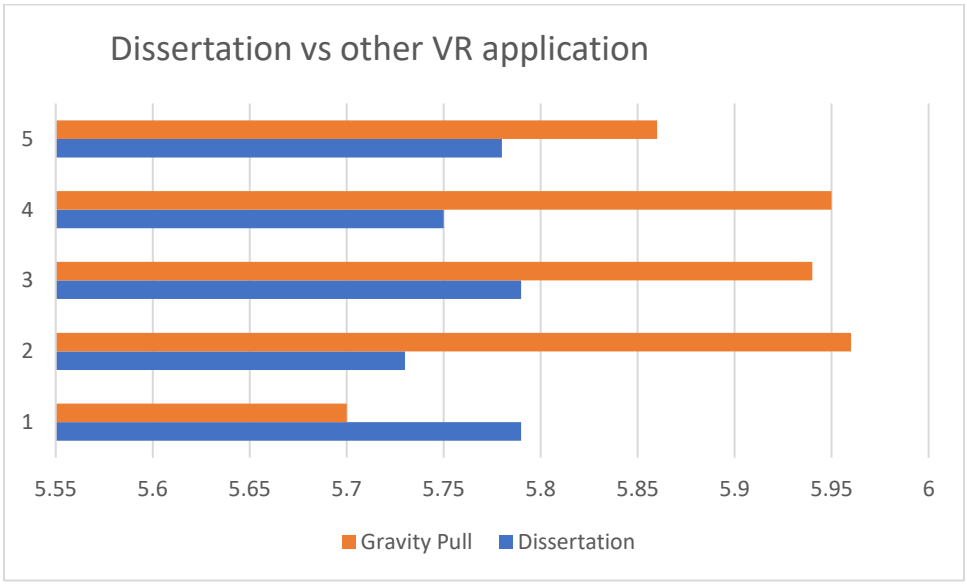

Figure 50: Dissertation vs other VR application

After observing the results, the conclusion was the application is running without any problem on the Android devices, that they are using the recent version of the Android operating system and performance is normal comparing to other applications that built with Unity3D.

5.1.2.Testing by participants:

This test performed by asking seven volunteers to test the application. For testing this application, all the movements in the game explained to them, such as how they have to interact and how they can move in it. The hardware used for this test were OnePlus 5T and Google Cardboard and a headphone.

Participant number seven used Google Daydream to test the application, and some parts of the application were not optimized for him.

This part has two parts one that monitored the participants' reactions and second part that participants answered the questioners and put their opinions in it.

5.1.2.1.Monitoring the participants:

Regarding to how much participants have experience in playing computer games, they had different backgrounds. Out of seven, four of them they do not play computer games that much and the other three mentioned they spend time on a weekly basis on playing computer games.

Three aspects examined while they were playing this application:

1. Length of the time that they manage to finish the game
2. Body gesture during the game
3. Participants' attention to the instructions

Length of the time that they manage to finish the game:

Out of seven people just one person could not find the key and asked for help and after giving a hint, managed to find the way out.

Other people managed to solve the puzzles and finish the game but the length of time that it took them to pass was so long, the anticipation for passing maze was around 3 minutes but the average of the time that users spend on the maze level was higher.

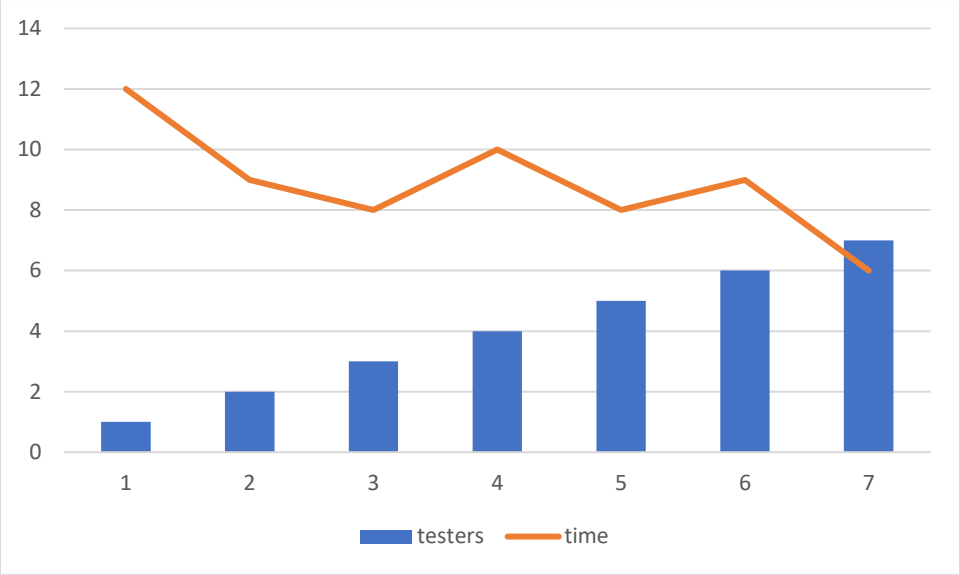

Figure 51: Length of the time that they manage to finish the game

The average time for finishing the game is: 8.857

The first participant took about 12 minutes to finish the game with help to find the key. This participant was the oldest one (61 years old) in the group without any gaming experience.

Comparing to the first participant, others were faster considering their ages. The last participant had the best performance.

Body gesture during the game:

In the beginning, two participants had difficulties, which are participant one and participant four. Both of them were still confused about the movement technique after the beginning tutoral session. Instead of toggling their head to move forward in the game, they started walking forward in real. Under this condition, they were stopped before injuring themselves and were instructed how to use the Cardboard

again. Participant one adopted the technique in a shorter time, while the participant four took about 4 minutes to familiarise with the procedure.

Another common phenomenon during this game experiment was that most of the participants became tired of holding the Cardboard on their head.

Participants' attention to the instructions:

The first four participants used the application which provided them with fewer instructions than the final version. Under this condition, In the first scene, they did not have tutorial; and on the second scene, there was an instruction note available but there were fewer hints in the maze.

The first participant read the instructions and had a problem in figuring out the movement in the beginning. Participant two figured out the mechanics and started wandering around the maze smoothly. Participant three demanded more hints about what to do at the start of the maze, and did not know how to deal with the door; but after this participant managed to open the door, he performed well in the following stages.

The performance of participant four was completely different. She did not follow any of the instructions and kept asking questions for help.

After observing these four participants, instructions were modified and a tutorial note in the first level was added according to their performance. More hints were added in the maze, for example, in the beginning of scene two, the hint that the players have to open the door was added. In addition, in the scene two, the instruction notes were enlarged comparing to the previous versions.

The version that provided for the last three participants had more instructions and the speed of the movement in the game was faster.

Participant five followed the instructions better and figured out the mechanics all by herself. The performance of participants six and seven were similar to participant five. Both of them followed the instructions without asking questions and managed to finish the game smoothly.

5.1.2.2Participants' opinion:

A questionnier was provided to all participants. It includes seven questions and one subjective comment part.

Question 1 asked for the age of the participants and answers are:

ANSWER CHOICES	RESPONSES	
Under 18	0.00%	0
18-24	28.57%	2
25-34	28.57%	2
35-44	28.57%	2
45-54	0.00%	0
55-64	14.29%	1
65+	0.00%	0

Table 1: question 1

Question 2 asked that about their experience in using the VR before:

ANSWER CHOICES	RESPONSES	
Yes	42.86%	3
No	57.14%	4
TOTAL		7

Table 2: question 2

That three of them have tried VR before and rest of them have not.

Question 3 asked about if they are interested to buy a VR device how much they prefer to spend on it and the responses are:

ANSWER CHOICES	RESPONSES	
1 to 20 pounds	14.29%	1
20 to 100 pounds	71.43%	5
100 to 300 pounds	14.29%	1
over 300 pounds	0.00%	0
TOTAL		7

Table 3: question 3

Five of them were preferred to spend between 20 to 100 pounds.

Question 4 asked the participants to rate the application from 0 to 10 that how easy was using this application, 0 means extremely hard, and 10 means extremely easy.

10 EXTREMELY EASY	9	8	7	6	5	4	3	2	1	0 EXTREMELY DIFFICULT	TOTAL	WEIGHTED AVERAGE
0.00% 0	0.00% 0	0.00% 0	57.14% 4	14.29% 1	0.00% 0	28.57% 2	0.00% 0	0.00% 0	0.00% 0	0.00% 0	7	7.00

Table 4: question 4

The weighted average is 7, that means most of them found it easy to use.

Question 5 asked them to rate the movement technique from 0 to 10 that how easy was it? 0 means extremely hard, and 10 means extremely easy.

10 EXTREMELY EASY	9	8	7	6	5	4	3	2	1	0 EXTREMELY DIFFICULT	TOTAL	WEIGHTED AVERAGE
0.00% 0	14.29% 1	42.86% 3	14.29% 1	14.29% 1	14.29% 1	0.00% 0	0.00% 0	0.00% 0	0.00% 0	0.00% 0	7	8.29

Table 5: question 5

The weighted average is 8.29, that means they found it easy.

Question 6 asked them to rate from 0 to 10 how much they think is interesting to use the sound for finding directions. 0 means not interesting, and 10 means so interesting.

10 SO INTERESTING	9	8	7	6	5	4	3	2	1	0 NOT INTERESTING	TOTAL	WEIGHTED AVERAGE
14.29% 1	14.29% 1	28.57% 2	42.86% 3	0.00% 0	0.00% 0	0.00% 0	0.00% 0	0.00% 0	0.00% 0	0.00% 0	7	9.00

Table 6: question 6

The weighted average is 9, that means they found this interesting.

Question 7 asked them to rate how satisfied they were about using the Google Cardboard from 0 to 10, where 0 means very dissatisfied and 10 very satisfied.

10 VERY SATISFIED	9	8	7	6	5	4	3	2	1	0 VERY DISSATISFIED	TOTAL	WEIGHTED AVERAGE
28.57% 2	0.00% 0	0.00% 0	14.29% 1	14.29% 1	28.57% 2	0.00% 0	14.29% 1	0.00% 0	0.00% 0	0.00% 0	7	7.57

Table 7: question 7

The weighted average is 7.57, in total they were satisfied, however, some found it hard that it does not have the headband and makes the arms tired.

6.Functionality:

The purpose of this application is to demonstrate the movement and interactions in the 3D world using the cheap solution, it means the solution has to provide a comfortable and easy way for the users, based on the results that participants provide, application was easy to use and the movements inside the designed 3D world was comfortable and easy. And if they required any changes, some additional updates added to the application that demonstrated that the rest of the participants were satisfied with it.

So, by looking at the features that provided, the conclusion of it is that this application is completely functional, and easy to use and easy to interact with it.

7.Future work:

The Virtual Reality is an open field that, it is possible to test and research different experiences in the different fields of industry. The focus of this study was using cheap equipment and finding solutions to get the most out of it. The future work needs to find more mechanics that follows this pattern. Daydream and other devices are using controller that makes it more convenient for the user to use the VR applications. If we look at the smart phones, all of them have a camera. The users can use actual objects as a controller by using the camera following the Augmented Reality method.

Another interest is to look for the mechanics and techniques that can provide entertainment for disabled people who suffer from visual deficiency and other diseases. Virtual Reality provides a very comfortable way to control the character in the virtual environment, thus there is no need to use keyboard and mouse or a gamepad during a game. By using a 3D sound or other types of sounds, users can react to the voice in the left or right headphone and turn around to move in the virtual reality game, without using the vision. Therefore, it is possible to design game for visual deficiency people, by using 3D sound to guide the direction in the game. By pressing the single button of the controller, users can choose the right decision in the game easily.

8.Conclusion:

By looking at the results of the testing stage, this project is able to create user friendly application that provides a movement only with their head, without the need of using extra controller. The study demonstrates the ways of interactions in the 3D world, in the shape of the a VR game, which is a maze.

This project managed to achieve the purposes that it has, and it succeeded to use the cheap equipment in an optimize way, however in tech industry by cutting the budget, must sacrifice the quality in some areas. Based on the observation on the participants, using the Google Cardboard can cause excitement, however because of lack of the headband, the user must hold his/her hands up to hold the headset, which it makes it hard to use the application longer than a limited time regarding to the fitness of the user. There are some Cardboard headsets on the market that they provide the headbands with the headset like this:

Figure 52: Cardboard with headband

The idea looks interesting but after testing it, it puts lots of pressure on the nose and the forehead that makes it impossible to use it more than two or three minutes. So, best solution for this is to develop applications that last between 1 to 3 minutes and provide a break in it to help the user to rest his/her hands for some minutes.

For developing

But the main thing in developing the cheap tools for Virtual Reality, in the fields like education, it provides lots of opportunities for young students living in the poor areas, that can have access to the Virtual Reality with affordable budget. Or in medicine in some cases that it is applicable can reduce the fees in the treatment process.

9.Reflection:

During this project, there were different problems that I had to find a solution for it, and had to approach them from different prospects.

9.1.Virtual Reality:

The topic of this project is Virtual Reality, which it was a new reality to deal with, as a person who enjoys playing the computer games, during this project that I decided to create it like a game, I experienced lots of visions in the development of a game. Considering different tools that they are all working together in parallel, the hardware and software, using different resources at a time to achieve one goal.

9.2.Development:

The development of this project was different, it helps me to look at the problem from a different aspect, before finding the solution, I needed to design a scenario first, then analyze the concept of the design and put out the problems, and sort them, and plan to solve them in order. At that point the development started, figured out the testing is not necessarily needed to wait until the end of the production. Before finishing each mechanic, I tested it a lot in different scenarios, and checked it with different ways, for example, if I wanted to make the yellow box have the gravity and be solid to other GameObjects I could not use it as a step for finding the key.

9.3.Programming:

For the game development, developers follow the concept of the object-oriented, however in Unity it was using the C# as a scripting language. the concept of it was like C++ and it was a small boost for me to understand the logic, learning the classes and methods were the challenging part, it uses different names that I am not familiar with it and learned a lot how to research in Unity C# documents, however some descriptions looks vague, but it was helpful.

9.4.Finding participants:

This task was similar to the job of the salesman in a shop that has to convince the people to come and look at his products and sell them all. It puts me completely out of my comfort zone, I had this Idea to choose the participants from the people who had no background in gaming, to see while they are using the app how comfortable they are with application. But when I was approaching to them asking them to test, most of the times their answers were negative, because by seeing a VR headset and they had to play the game front of another person, it put them out of their comfort zone. Eventually, I managed to find the participants for my VR game.

9.5.Battle with the time:

The amount of the time that I had for this project, pushed me to my limits to finish this project less than five weeks, I scheduled an intense plan to manage to develop the application, for motivating myself and working extra and managed to change the habits of working. Every day has a different milestone and solving some parts of the application were going concurrently together.

References

Burdea, G. and Coiffet, P. 2011. Virtual Reality Technology. Wiley & Sons, Limited, John.

Isdale, J. 1993. What Is Virtual Reality?. 2.1.

Available at: *http://www.columbia.edu/~rk35/vr/vr.html*

[Accessed: 4 September 2018].

Gearbrain, 2015.Google Cardboard[Online]

Available at: *https://www.gearbrain.com/review-google-cardboard-v1-vs-v2-1622025361.html*

(Accessed 05 September 2018)

Romano, D. 2005. Virtual Reality Therapy. Developmental Medicine & Child Neurology 47(9), pp. 580-580. doi: 10.1111/j.1469-8749.2005.tb01206.x.

Unreal Engine, 2018. Available at: *https://docs.unrealengine.com/en-us/Platforms/GoogleVR*

(Accessed 05 September 2018)

Wang, S. et al. 2010. A new method of virtual reality based on Unity3D. In: *2010 18th International Conference on Geoinformatics.*

hou.Vector4. 2018. Available at: *http://www.sidefx.com/docs/houdini/hom/hou/Vector4.html* [Accessed: 7 September 2018].

NurFACEGAMES, Movement in Mobile VR: Look Walk,2016.

Available at: *https://www.youtube.com/watch?v=kBTn2pGwZUk*

[Accessed: 7 September 2018].

Unity Technologies, 2018. MonoBehaviour. [Online]

Available at: *https://docs.unity3d.com/ScriptReference/MonoBehaviour.html*

(Accessed 05 September 2018)

Unity Technologies, 2018. Rigidbody. [Online]

Available at: *https://docs.unity3d.com/Manual/RigidbodiesOverview.html*

(Accessed 05 September 2018)

Unity Technologies, 2018.SetParent . [Online]

Available at: *https://docs.unity3d.com/ScriptReference/Transform.SetParent.html*

(Accessed 05 September 2018)

Unity Technologies, 2018. Event Systems. [Online]

Available at: *https://docs.unity3d.com/Manual/EventSystem.html*

(Accessed 05 September 2018)

Unity Technologies, 2018. Transform. [Online]

Available at*: https://docs.unity3d.com/ScriptReference/Transform.html*

(Accessed 05 September 2018)

Unity Technologies, 2018. Box Collider. [Online]

Available at: *https://docs.unity3d.com/ScriptReference/BoxCollider.html*

(Accessed 05 September 2018)

Unity Technologies, 2018. GetButtonDown. [Online]

Available at*: https://docs.unity3d.com/ScriptReference/Input.GetButtonDown.html*

(Accessed 05 September 2018)

Google VR, 2018. Gvreventsystems.[Online]

Available at: *https://developers.google.com/vr/reference/unity/prefab/GvrEventSystem*

(Accessed 01 September 2018)

Unity Technologies, 2018. Colliders.[Online]

Available at: *https://docs.unity3d.com/Manual/CollidersOverview.html*

(Accessed 01 September 2018)

Unity Technologies, 2018. Animation.[Online]

Available at: *https://docs.unity3d.com/ScriptReference/Animation.html*

(Accessed 01 September 2018)

Unity Technologies, 2018. Animation Window Guide.[Online]

Available at: *https://docs.unity3d.com/Manual/AnimationEditorGuide.html*

(Accessed 01 September 2018)

Unity Technologies,Daydream 2018.[Online]

Available at: *https://docs.unity3d.com/2017.2/Documentation/Manual/VRDevices-GoogleVR.html*

(Accessed 05 September 2018)

Unity Technologies, 2018.SetActive. [Online]

Available at: *https://docs.unity3d.com/ScriptReference/GameObject.SetActive.html*

(Accessed 05 September 2018)

Unity Technologies, 2018.SceneManagment. [Online]

Available at:

https://docs.unity3d.com/ScriptReference/SceneManagement.SceneManager.LoadScene.html

(Accessed 05 September 2018)