Classes for modules chapter 4

The classes in unity are attached to the game objects, therefore every module has its own classes which are not connected to any other module. The classes that are linked to each module are described below.

Main menu:

We use the animator’s built-in functions to click buttons through gaze method

|  |  |
| --- | --- |
| Index | Class |
| 1 | Player Walk |

Game Mode

We use the animator’s built-in functions to trigger events based on user input at the end of level

|  |  |
| --- | --- |
| Index | Class |
| 1 | Timer |
| 2 | Level Chan |
| 3 | SunHit |
| 4 | Rainsound |
| 5 | Low Poly Water |

Tour Guide

We use the animator’s built-in functions to trigger events based on user input at the end of level

|  |  |
| --- | --- |
| Index | Class |
| 1 | Simple Character Con |
| 2 | Character Tour |
| 3 | Low Poly Water |

Chapter 5 implementation

Introduction

In this chapter we will give brief details about how we will be implementing our application, hardware and software description along with the interface. In this chapter we will give a basic idea of how the application can be used.

This section includes software’s used for app development, hardware components used, then a brief explanation about the app user interface.

Software Description

To develop this application, we had two main software development tool that could be used: Unity or Unreal Engine. We opted Unity as it is has few features that meet our need such as better particle system management, pro-builder that allows custom object creation, different rendering options. Unity supports C# as its standard language. Therefore, Unity is the main tool along with Microsoft Visual Studio that is integrated with Unity for scripting.

Programming Language

For this app development we used C# as it is integrated with unity and works well along with many features making it easy to implement in C#. Benefits of C# is that it does memory management as we want this app to work on low end device where memory management is important.

Unity

Main tool used for our app development as it allows creating objects, animations, sound editing, creating custom design and object, monitoring system usage, rendering, app can be deployed to several operating systems and particle system management. Most of these functions are built-in features. It allows scripts to be connected directly to objects instead of classes that makes it more efficient. An object can have many scripts attached to it and can be executed without any issue.

Microsoft Visual Studio:

Mainly used for script as it in has better connectivity with UNITY. All scripts are written in this tool and it will update UNITY if the scripts are being edited or not. Locks scripts that are being edited and unlock once they are saved.

Hardware Description

Our application requires more than one device to work as expected, list of devices required are as follows:

VR headset

Head worn apparatus that provides an impressive 3D experience by covering around the eyes and making the device placed inside as the main screen. We will be using a low-cost VR headset as it requires user device to placed inside the VR.

Android Device

Any android device can be used that runs on version KitKat or later, minimum 1 gb ram, built in gyroscope, accelerometer, proximity sensor as all devices come pre-equipped with of the required feature.

Bluetooth controller

A controller that has built-in 360o rocker along with fire 1 and fire 2 button. It will allow the user to control the player in app as needed

User Interface Description

We will discuss about how the user interface looks at different levels within the game. Each element that interacts with the user is made in a simple way as users will be easy able to navigate throughout the app without any issue. Overall, we have implemented few buttons as to make sure we don’t confuse the user and keep it as simple as possible.

Main Menu



The figure above is the main menu of the app where user gets three options: Start Tour, Start Game and Exit. Having a look at these buttons easily allows user to get an idea of what these buttons means and where they will be navigated to.

Tour Mode



The figure above is taken during the user is going through the tour. User will be able to look around and see how the water cycle works and a robot companion is also seen guiding the user about the different phases of the water cycle.

Game Mode



The figure above is taken during user playing the game. User is assigned to hit the yellow rays towards the water as it increases the water temperature and thus leading to condensation that leads to rainfall. Count down timer is seen on the top right which indicates how much time is left for the user. Below the timer the user can look at the water temperature and see how it increases as the rays hit the water and on the top left of the screen the user can see the highest score that has been achieved by any of the previous users.

Conclusion

This chapter discussed how the app is developed along with details of the hardware and software requirements. Also, a detail description about the user interface and how the user will be able to use it in different phase of the app