**Water Cycle Guide using Virtual Reality**

**ADNAN ALI - 201520023**

**ZAINAB SHAHID - 201520629**

**A project report submitted in partial fulfillment of the**

**requirements for the award of the degree of**

**Bachelor of Software Engineering**

****

**College of Engineering**

**Al Ain University, Al Ain, U. A. E.**

**December 2019**

## APPROVAL FOR SUBMISSION

I certify that this project report entitled **“Water Cycle Guide using Virtual Reality”**

was prepared by **ADNAN ALI, ZAINAB SHAHID** has met the required standard for submission in partial fulfillment of the requirements for the award of Bachelor of Software Engineering at Al Ain University of Science and Technology, Al Ain.

Approved by,

Signature : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supervisor : Dr. Saqib Iqbal

Date : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Water Cycle Guide using Virtual Reality**

## ABSTRACT

This project is a Virtual Reality application which is made to educate students about the earth water cycle. It will allow users to have a visual view of the complete water cycle and its stages. The water cycle is a very important concept in our life and due to its importance, several applications have been made in this domain to teach students about the water cycle. Earlier, students were taught these concepts using images and texts, which is not very elaborative, making it more complicated for the little kids to understand it. Furthermore, not many Virtual Reality applications have been made in this domain and using this technology, therefore this application is beneficial for students to gain knowledge and for any other user who is interested in familiarizing himself/herself with the water cycle. Our application is divided into two major roles; one is to allow the user to take a tour of the water cycle and visualizing every stage. Second is a mini game that will polish up user’s knowledge in the water cycle, making the learning process entertaining as well as useful. The methodology used to fulfill this project is Rapid Application Development (RAD) and the expected outcome will be an informative and interactive game and a tour of the water cycle in which there will be a robot companion to explain every stage.

Table of Content

[APPROVAL FOR SUBMISSION ii](#_Toc25090344)

[ABSTRACT iii](#_Toc25090345)

[LIST OF FIGURES vii](#_Toc25090346)

[LIST OF TABLES ix](#_Toc25090347)

[LIST OF SYMBOLS / ABBREVIATIONS x](#_Toc25090348)

[1: INTRODUCTION 1](#_Toc25090349)

[1.1 Introduction 1](#_Toc25090350)

[1.2 Problem Statement 1](#_Toc25090351)

[1.3 Objective 1](#_Toc25090352)

[1.4 Methodology 2](#_Toc25090353)

[1.5 Robot Companion 2](#_Toc25090354)

[1.6 Expected outcome 2](#_Toc25090355)

[1.7 Plan-Gantt chart 2](#_Toc25090356)

[1.8 Structure of report 3](#_Toc25090357)

[1.9 Conclusion 3](#_Toc25090358)

[2: Literature Review 4](#_Toc25090359)

[2.1 Introduction 4](#_Toc25090360)

[2.2 Similar Apps 4](#_Toc25090361)

[2.3 Apps Comparison 8](#_Toc25090362)

[2.4 Conclusion 8](#_Toc25090363)

[3: Requirements Analysis 9](#_Toc25090364)

[3.1 Introduction 9](#_Toc25090365)

[3.2 Functional Requirements 9](#_Toc25090366)

[3.3 Non-Functional Requirements 9](#_Toc25090367)

[3.4 Use Case Diagram 10](#_Toc25090368)

[3.5 System Sequence Diagram: 15](#_Toc25090369)

[3.6 Conclusion 15](#_Toc25090370)

[4: Design 16](#_Toc25090371)

[4.1 Introduction 16](#_Toc25090372)

[4.2 Component Diagram 16](#_Toc25090373)

[4.3 Deployment Diagram 20](#_Toc25090374)

[4.4 Conclusion 20](#_Toc25090375)

[5: Implementation 21](#_Toc25090376)

[5.1 Introduction 21](#_Toc25090377)

[5.2 Software Description 21](#_Toc25090378)

[5.3 Programming Language 21](#_Toc25090379)

[5.4 Unity 21](#_Toc25090380)

[5.5 Microsoft Visual Studio 21](#_Toc25090381)

[5.6 Hardware Description 21](#_Toc25090382)

[5.7 User Interface Description 22](#_Toc25090383)

[5.8 Conclusion 24](#_Toc25090384)

[6: Software Testing 25](#_Toc25090385)

[6.1 Introduction 25](#_Toc25090386)

[6.1.1 Nature of the Project 25](#_Toc25090387)

[6.1.2 Testing Objectives and Scope 25](#_Toc25090388)

[6.1.3 Types of System Tests 25](#_Toc25090389)

[6.2 Features to be Tested 26](#_Toc25090390)

[6.3 Approach 26](#_Toc25090391)

[6.3.1 Staff 26](#_Toc25090392)

[6.3.2 Record Keeping 26](#_Toc25090393)

[6.3.3 Stop-Test Criteria 26](#_Toc25090394)

[6.4 Testing Techniques 26](#_Toc25090395)

[6.5 Test Cases 27](#_Toc25090396)

[6.5.1 Unit Testing 27](#_Toc25090397)

[6.5.2 System Testing 31](#_Toc25090398)

[6.6 Usability Testing 33](#_Toc25090399)

[6.6.1 Methodology 33](#_Toc25090400)

[6.6.2 Usability Testing 33](#_Toc25090401)

[6.6.3 Test Results 35](#_Toc25090402)

[6.7 Conclusion 36](#_Toc25090403)

[7: Conclusion & Future Work 37](#_Toc25090404)

[7.1 Conclusion 37](#_Toc25090405)

[7.2 Future Work 37](#_Toc25090406)

[References 38](#_Toc25090407)

## LIST OF FIGURES

**FIGURE TITLE PAGE**

[Figure 1.1: Plan-Gantt Chart 2](#_Toc25092093)

[Figure 2.1: Splash Sim app 5](file:///C:\Users\Ali\Documents\GitHub\Capstone\WATERCYCLE-VR%20guide%20(final).docx#_Toc25092094)

[Figure 2.2: Cicle de l’Aigue VR app 6](file:///C:\Users\Ali\Documents\GitHub\Capstone\WATERCYCLE-VR%20guide%20(final).docx#_Toc25092095)

[Figure 2.3: TABI the water cycle 6](file:///C:\Users\Ali\Documents\GitHub\Capstone\WATERCYCLE-VR%20guide%20(final).docx#_Toc25092096)

[Figure 2.4: วัฏจักรน้ำ (Water Cycle) 7](file:///C:\Users\Ali\Documents\GitHub\Capstone\WATERCYCLE-VR%20guide%20(final).docx#_Toc25092097)

[Figure3.1: Main Menu Use Case Diagram 10](file:///C:\Users\Ali\Documents\GitHub\Capstone\WATERCYCLE-VR%20guide%20(final).docx#_Toc25092099)

[Figure 3.2: Game Mode Use Case Diagram 11](#_Toc25092103)

[Figure 3.3: Tour menu Use Case Diagram 13](#_Toc25092107)

[Figure 3.4: System Sequence Diagram 15](#_Toc25092110)

[Figure 4.1 Main Menu Component 16](#_Toc25092111)

[Figure 4.2 Game Mode Component 17](#_Toc25092112)

[Figure 4.3 Tour Guide Component 18](#_Toc25092113)

[Figure 4.4 Module for Animator 19](#_Toc25092114)

[Figure 4.5 Deployment Diagram 20](#_Toc25092115)

[Figure 5.1 Screenshot of Main Menu 22](#_Toc25092116)

[Figure 5.2 Screenshot of Tour Mode 23](#_Toc25092117)

[Figure 5.3 Screenshot of Game Mode 23](#_Toc25092118)

[Figure 6.1 Survey 34](#_Toc25092127)

[Figure 6.2 Bar graph for response of I saw or heard something entertaining 35](file:///C:\Users\Ali\Documents\GitHub\Capstone\WATERCYCLE-VR%20guide%20(final).docx#_Toc25092128)

[Figure 6.3 Bar graph for response of I had Fun 35](file:///C:\Users\Ali\Documents\GitHub\Capstone\WATERCYCLE-VR%20guide%20(final).docx#_Toc25092129)

[Figure 6.4 Bar graph for response of I found the text clear 35](file:///C:\Users\Ali\Documents\GitHub\Capstone\WATERCYCLE-VR%20guide%20(final).docx#_Toc25092130)

[Figure 6.5 Bar graph for response of I will use the app again 35](file:///C:\Users\Ali\Documents\GitHub\Capstone\WATERCYCLE-VR%20guide%20(final).docx#_Toc25092131)

[Figure 6.6 Bar graph for response of I felt comfortable using the app 35](file:///C:\Users\Ali\Documents\GitHub\Capstone\WATERCYCLE-VR%20guide%20(final).docx#_Toc25092132)

## LIST OF TABLES

**TABLE TITLE PAGE**

[Table 2.1: Comparison table between our app and similar apps 8](#_Toc25092058)

[Table 3.1: Tour Guide Use-Case 10](#_Toc25092060)

[Table 3.2: Start Game Use-Case 11](#_Toc25092061)

[Table 3.3: Exit Use-Case 11](#_Toc25092062)

[Table 3.4: Move Player Use-Case 12](#_Toc25092064)

[Table 3.5: Play Again Use-Case 12](#_Toc25092065)

[Table 3.6: Exit to Main Menu Use-Case 13](#_Toc25092066)

[Table 3.7: Repeat tour Use-Case 14](#_Toc25092068)

[Table 3.8: Exit to main menu Use-Case 14](#_Toc25092069)

[Table 6.1 Test Cases for Start Tour Guide 27](#_Toc25092079)

[Table 6.2 Test cases for Start Game 28](#_Toc25092080)

[Table 6.3 Test Cases for Exit 29](#_Toc25092081)

[Table 6.4 Test cases for Repeat Tour 29](#_Toc25092082)

[Table 6.5 Test cases for Exit to Main Menu 30](#_Toc25092083)

[Table 6.6 Test Cases for Move player 31](#_Toc25092084)

[Table 6.7 test cases for Play again 31](#_Toc25092085)

[Table 6.8 Test cases for the flow of the user tasks 33](#_Toc25092086)

## LIST OF SYMBOLS / ABBREVIATIONS

Index Title Description

1 Virtual Reality VR

2 Application App

3 Software Development Life Cycle SD

## 1: INTRODUCTION

## 1.1 Introduction

Virtual reality (VR) is an innovation that enables a client to connect with a PC reproduced condition, regardless of whether that condition is a recreation of this present reality or a conjured-up universe. It is seen as one of the key technologies that are going to shape the future of computing. It is the way to encounter, feeling and contacting the past, present and what's to come. It is the mode of making our reality, our very own redid reality. It could go from making a computer game to having a virtual walk around the universe, from strolling through our own fantasy house to encountering a stroll on an outsider planet. With computer-generated reality, we can encounter the scariest and difficult circumstances by playing safe and with a learning point of view [1].

The water cycle is a very important process and essential to all living organisms. It ensures that the water is available for all the creatures and it helps to regulate weather patterns on the planet [2]. A water cycle describes the continuous movement of the water on the surface of the earth. Its main stages are evaporation, transpiration, condensation, and precipitation. In evaporation, the water is transferred from the surface of the earth to the atmosphere. The main sources of evaporation are the oceans, the rivers, the seas, and the lakes. Transpiration is known as evaporation of water from the leaves of plants. After rising, the water vapor cools down and forms into tiny particles of water or ice because of the low temperature at high altitudes. When these tiny droplets combine, they create a bigger droplet, the air then cannot hold any more water, so it pours it down as rain, and this stage is called precipitation [3].

As to VR, we can develop an application that will help kids have a better view of the water cycle through this technology. As kids will be able to see all the stages that take place in the water cycle. Moreover, an added game for the kids will help them understand the water cycle in a better way as kids love to play games and that helps them understand better than just have an explanation/imagination of the different stages of the water cycle.

## 1.2 Problem Statement

As of now, there are few VR app that takes the user through a virtual tour of the water cycle which is difficult for kids when they hear about the water cycle, it is explained to them through text/pictures that make it difficult for them to imagine how it exactly works. But if they have a chance to have a visual view of the water cycle, they will understand it easily as they will have a visual of all the stage taking place and how these stages are linked to each other.

## 1.3 Objective

The purpose of this project is to design a Virtual Reality application for school students to learn about earth’s water cycle.

The objects of the project are:

* To develop a virtual reality application for water cycle education.
* To provide users with a game to understand water cycle
* To provide users with different modes of interaction.
* To provide users with real model view of water cycle.

## 1.4 Methodology

In this project, we will use Rapid Application Development (RAD) [4].

RAD is a type of agile methodology. Unlike other methods that are designed to be a one-way process that must go through all the phases before coming back to the previous stage if needed. RAD is less strict on the phase model, focus moreover user feedback and working software. RAD soul is based on more action rather than talks. Even though RAD doesn’t work based on strict planning techniques, many other helpful things make this methodology more successful.

Other possible methodologies are Prototyping, waterfall, but we prefer to stick to RAD as we are on tight deadline and don’t have enough time to develop a non-working model and then work on its functionality or in waterfall model we will have to go through all phases and wait to add/remove certain elements that will harshly affect our time for project completion.

## 1.5 Robot Companion

Robot Companion is like a tour guide that explains different phases of the water cycle to the user and guides them along the way. He will do certain action as new text appear as to make sure the user reads it and doesn’t lose its way during the transition of phases.

## 1.6 Expected Outcome

The developed application will allow students to have a tour of different phases of the water cycle with help of a robot companion that will provide them with information about the different stages of the water cycle tour as the go through them. The other part which is the mini-game will be where the student uses the Bluetooth controller to play the game by controlling the player’s movement and achieving the task given to them within a certain time limit.

## 1.7 Plan-Gantt Chart

The plan of the project is provided in a Gantt Chart below:



Figure 1.1: Plan-Gantt Chart

## 1.8 **Structure of Report**

The rest of the report is structured as follows: Chapter 2 is about literature review that will include most of the similar Apps related to our project and a comparison table to give a better idea of the differences between the VR Apps. Chapter 3 will be requirements analysis that contains the functional and non-functional requirements leading to the use-case diagram and system sequence diagram. Chapter 4 is the conclusion with references included as well.

## 2: Literature Review

## 2.1 Introduction

The water cycle has always been an important topic in science. Teachers sometimes struggle to explain such concepts briefly as they require imagination and a complete picture of how such processes go on. So, in this domain, many applications are created to ease the understanding of the students as well as the teachers.

Virtual Reality (VR) is seen to be playing a great role in education. Using its various features, it has made learning of many concepts easier. Like our app, there are many VR applications created to widely expose these concepts to students and allow them to virtually visit places that are distant and physically inaccessible to them. In this chapter, we will be discussing applications that are already made for users to educate them about the water cycle and comparing our application and its features with them.

## 2.2 Similar Apps

Below are some applications that are like our application and are meant to educate students:

1. **Splash Sim: A VR Experience in the Water Cycle [5]**

**App Description:**

This app is to teach students the water cycle by engaging them in all the cycles using virtual reality. It requires the virtual reality headset to be viewed properly. It has also been optimized for Google cardboard [3]. This app explains only the three phases of the water cycle (evaporation, condensation, and precipitation) by a guided tour along with a text appearing on the screen.

**Features:**

* User is taken to a tour to see different phases of water cycle.
* Text is appeared to explain user about the phases.
* Audio effect of rain, wind and birds.

**Weaknesses:**

* Lack of animation in evaporation stage.
* No text to speech implementation.
* The tour gets on repeated by itself.



Figure 2.1: Splash Sim app

1. **Cicle de l’Aigue VR (Water cycle VR) [6]**

**App Description:**

This app, which is in the Catalan language, allows the user to experience the natural cycle of water from the perspective of a drop. Water cycle VR explains and shows each phase separately with a smooth shift between. The phases are explained with the help of audio speakers in the Catalan language. A piece of slight music is played in the background along with the sound effects of rain, wind, and ocean to make the user engage in the environment.

**Features**:

* A guided tour to allow the user to understand all the phases of water cycle.
* Text to speech implementation to explain user the water cycle stages.
* Audio effect for rain and wind.
* Main menu to allow user to choose the option they would like to go for.
* Bluetooth control.

**Weaknesses:**

* Hanging occurs while the scene is viewed from different angles.
* The app has no other language support other than Catalan language.



Figure 2.2: Cicle de l’Aigue VR app

1. **TABI the water cycle [7]**

**App Description:**

This app teaches the user how the water behaves in our world. Basically, it’s a game-based app to allow the user to interact and observe the different phases of the water cycle. The user is asked to perform a function to see the result. The result is different phases of the water cycle. It teaches the phases precipitation, condensation, and evaporation and how one phase changes to another.

**Features:**

* User plays a game to create evaporation, condensation and precipitation.
* Text shown at the beginning to guide the user to play the game.

**Weaknesses:**

* Not a VR app.
* It doesn’t show the complete water cycle at once and how a stage transforms into another.

Figure 2.3: TABI the water cycle

1. **วัฏจักรน้ำ (Water Cycle) [8]**

**App Description:**

This app is based on the Thai language, contains multimedia lessons of the water cycle to help students learn the water cycle. These lessons explain the meaning of the water cycle, the processes and factors of the water cycle. It also includes a game to allow students to interact with the app and understand the phases more clearly through playing the game.

**Features:**

* Main menu to allow user to choose where they want to go.
* Lessons about water cycle explaining the phases.
* Game to educate user more about water and to allow them to understand it more.
* Audio speech to engage then user more in the app and explain each step clearly.

**Weaknesses:**

* No language support other than Thai language.
* Not a VR app.
* All phases are not shown continually to explain the transformation between them.

****

Figure 2.4: วัฏจักรน้ำ (Water Cycle)

## 2.3 Apps Comparison

Here is comparison table of our app with the already existing water cycle apps showing the similarities and differences between the apps and distinguishing the unique features of our app.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Features\ Apps | Splash Sim | Cicle de l’Aigue VR | TABI the water cycle | วัฏจักรน้ำ (Water Cycle ) | Water Cycle VR Guide (Expected app) |
| VR app | ✓ | ✓ | x | x | ✓ |
| Interactivity implementation | ✓ | ✓ | x | x | ✓ |
| Main menu | x | ✓ | ✓ | ✓ | ✓ |
| Phase Explanatory text | ✓ | x | ✓ | ✓ | ✓ |
| Guided tour | ✓ | ✓ | x | x | ✓ |
| Educative Game | x | x | ✓ | ✓ | ✓ |
| Animation | ✓ | ✓ | ✓ | ✓ | ✓ |
| Robot companion | x | x | x | x | ✓ |
| Text to speech implementation | x | ✓ | x | ✓ | x |

Table 2.1: Comparison table between our app and similar apps

The table shows that our application is expected to be unique in some aspects. We will be implementing the robot companion which is a robot guide within the app that guides the user throughout the tour. Another distinguishing feature of the app is that it contains both a tour guide and a game, which are not available together in any other app. However, the features which are present in most of the applications are the Main menu, Animation, and phase explanatory text, which is basically a text that appears to the user as they go through the tour to explain them about every phase of the water cycle.

## 2.4 Conclusion

This chapter described the applications that are already been built in the domain, highlighting the features present and absent in those apps. Clearly, we have seen that multiple apps like Water Cycle VR Guide app were developed in this field of technology. Furthermore, this chapter focused on the similarities and differences between the existing apps and Water Cycle VR Guide app, concluding how different the water cycle VR Guide is expected to be.

## 3: Requirements Analysis

## 3.1 Introduction

In this chapter, we will discuss the functional and nonfunctional requirements of the application. Use case diagrams along with a system sequence diagram that will give a brief detail of the requirements and how it will be further implemented.

## 3.2 Functional Requirements

In this section we will list out the functional requirements of our application.

**R1**: The application shall allow user to select Tour Guide/Game or Exit the application

**R2**: The application shall guide the user with robot companion.

**R3**: The application shall pass the user through all the water cycle phases.

**R4**: The application shall allow user to look around (360° view).

**R5**: The application shall allow user to control player movement.

**R6**: The application shall limit user gameplay time.

**R7**: The application must record user gameplay time for high score.

## 3.3 Non-Functional Requirements

In this section we will list out the non-functional requirements of our application.

**R8**: Usability

Application is easy to use.

## 3.4 Use Case Diagram

## 

Figure3.1: Use Case Diagram

|  |  |
| --- | --- |
| Use Case ID | 1 |
| Use Case Name | Start Tour |
| Actor | User |
| Description | This describe the process that allows user to start tour |
| Precondition | Application launched |
| Flow of Events | 1. Main Menu loaded. 2. User gaze at Start Tour button for 2 seconds. |
| Post-Condition | The application will start the tour |
| Alternative Flow | - |
| Assumptions | - |

Table 3.1: Tour Guide Use-Case

|  |  |
| --- | --- |
| Use Case ID | 2 |
| Use Case Name | Start Game |
| Actor | User |
| Description | This describe the process that allows user to start tour |
| Precondition | Application launched |
| Flow of Events | 1. Main Menu loaded 2. User gaze at Start Game button for 2 seconds. |
| Post-Condition | The application will start the game |
| Alternative Flow | - |
| Assumptions | - |

Table 3.2: Start Game Use-Case

|  |  |
| --- | --- |
| Use Case ID | 3 |
| Use Case Name | Exit |
| Actor | User |
| Description | This describe the process that allows user to exit application |
| Precondition | Application launched |
| Flow of Events | 1. Main Menu loaded 2. User gaze at Exit button for.2 seconds. |
| Post-Condition | The application will exit |
| Alternative Flow | - |
| Assumptions | - |

Table 3.3: Exit Use-Case

|  |  |
| --- | --- |
| Use Case ID | 4 |
| Use Case Name | Move Player |
| Actor | User |
| Description | This describe the process that allows user to move player to complete mission |
| Precondition | Application launched and user must be in-game |
| Flow of Events | 1. Application loaded 2. Main Menu loaded 3. User gaze at Start Game button for 2 seconds 4. User controls player movement |
| Post-Condition | The application will display time taken (high score) to complete task |
| Alternative Flow | 1. User fails to complete mission in time provided |
| Assumptions | - |

Table 3.4: Move Player Use-Case

|  |  |
| --- | --- |
| Use Case ID | 6 |
| Use Case Name | Play Again |
| Actor | User |
| Description | This describe the process that allows user to Play Game after first try |
| Input | Button C |
| Precondition | User fails in first attempt |
| Flow of Events | 1. Application loaded 2. Main Menu loaded 3. User gaze at Start Game button for 2 seconds 4. User controls player movement 5. User press C button to Play Again. |
| Post-Condition | The Game will restart |
| Alternative Flow | - |
| Assumptions | - |

Table 3.5: Play Again Use-Case

|  |  |
| --- | --- |
| Use Case ID | 7 |
| Use Case Name | Exit to main menu |
| Actor | User |
| Description | This describe the process that allows user to exit game and load main menu |
| Input | Button A |
| Precondition | User fails in first attempt |
| Flow of Events | 1. Application loaded 2. Main Menu loaded 3. User gaze at Start Game button for 2 seconds 4. User controls player movement 5. User press A button to return to Main Menu |
| Post-Condition | The Game will exit and load main menu |
| Alternative Flow | - |
| Assumptions | - |

Table 3.6: Exit to Main Menu Use-Case

|  |  |
| --- | --- |
| Use Case ID | 8 |
| Use Case Name | Repeat tour |
| Actor | User |
| Description | This describe the process that allows user to restart tour |
| Precondition | Application launched and user must be in-tour |
| Input | Button C |
| Flow of Events | 1. Application loaded 2. Main Menu loaded 3. User gaze at Start Tour button for 2 second 4. User press C button to return to Main Menu |
| Post-Condition | The tour will restart. |
| Alternative Flow | - |
| Assumptions | - |

Table 3.7: Repeat tour Use-Case

|  |  |
| --- | --- |
| Use Case ID | 9 |
| Use Case Name | Exit to Main Menu |
| Actor | User |
| Description | This describe the process that allows user to exit tour and load main menu |
| Precondition | Application launched and user must be in-tour |
| Input | Button A |
| Flow of Events | 1. Application loaded 2. Main Menu loaded 3. User gaze at Start Tour button for 2 seconds   4.User press A button for exit to Main Menu |
| Post-Condition | The application will exit tour and load main menu |
| Alternative Flow | - |
| Assumptions | - |

Table 3.8: Exit to main menu Use-Case

## 

## 3.5 System Sequence Diagram:

The system sequence diagram shown below reveals the sequence of actions, their order and events generated by external actors, to fulfill all the tasks of our water cycle VR app.

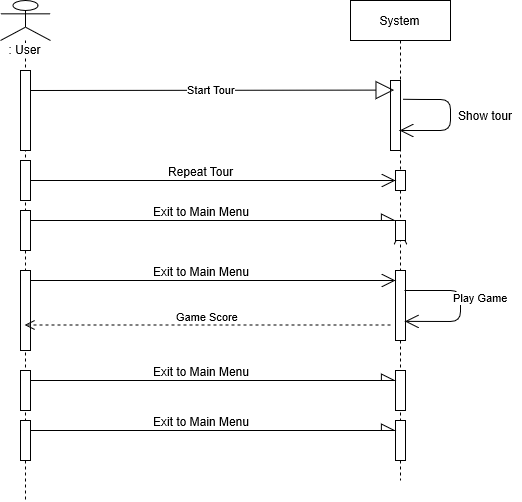


Figure 3.4: System Sequence Diagram

## 3.6 Conclusion

In this chapter, we discussed the functional and non-functional requirements. Use case diagram was developed to provide the high-level view of the system and to model the functionality of the system using actors and use cases. Moreover, System sequence diagram was developed to describe the interaction between the actor and system.

## 4: Design

## 4.1 Introduction

In this chapter, we will be talking about the design phase of the software development life cycle (SDLC). After the requirements are captured accurately, the requirements in this phase will be broken down further to estimate the amount of effort and resources needed. In this phase, a software architecture that is used for implementing system development is derived from the software requirement specification document. Therefore, the architecture of the system will be established, component and deployment diagrams will be made, and all supporting diagrams will also be included in this chapter.

## 4.2 Component Diagram [9]

The component diagram is best suited for our app. The whole system is decomposed into subsystems. Each subsystem has its classes and objects. It makes it easy for implementation as all the subsystems can be developed simultaneously as they are not connected, and classes of those subsystems do not depend on other subsystems to work perfectly. It also allows planning the tasks of each subsystem. As it allows us to decompose all the related class to a single subsystem that provides a better idea for the developers a higher level of detail.

Component in the component diagram is denoted by stereotype “<<component>>”, classes are denoted by stereotype “<<source>>”. Classes related to the module are displayed with the module for better understanding.

**Main Menu** **Component**



Figure 4.1 Main Menu Component

In the figure above the main menu module has two classes PlayerWalk that allows user to walk around and the GvrHeadset is used for gaze control that click the button when user gaze at them.

**Game Mode Component**



Figure 4.2 Game Mode Component

In the figure above the game module has several classes that execute different task that are needed by this module to run properly. SunHit script counts the number of sun rays that hit the water and trigger objects execution. PlayerWalk script allows user to walk around to complete his task. RainSound script triggers the audio of rain when the user completes the task. LevelChan script allows user either to play game again or go to the main menu. Timer script counts the time given to the user to complete the task. HighScore script calculates the score of the user. Clocksound script runs when remaining time is less than 10 seconds. Buttontest script is used to enable the button on controller only if the user could not finish the task in 60 seconds.

**Tour Guide Component**



Figure 4.3 Tour Guide Component

In the figure above the tour module has three classes. characterTour script provides co-ordinates for the robot’s movement as it takes the tour of the complete water cycle. cameraController script ensures that camera follows the robot and moves with it. theEnd script display user with button to either play tour again or exit to main menu.

**Animator Module**



Figure 4.4 Module for Animator

In the figure above the animator module has classes that are responsible to load scenes. IntroGame script explains the task user must complete. LoadExit script exits the app. loadGame script loads the game. SceneControl scripts ensures that user is loaded into the required scene. LoadSceneButton script tells the animator which button is clicked through gaze.

## 4.3 Deployment Diagram [10]

The deployment diagrams show the physical relationship between the hardware and software of the system. The deployment diagram shown below describes the connection between the Android mobile device and the controller, which has a Bluetooth connection between them, stating that the devices communicate through Bluetooth. Furthermore, it shows that the Water Cycle VR Guide app resides in the android device and shows the components of the app.



Figure 4.5 Deployment Diagram

## 4.4 Conclusion

This chapter concludes the architecture of our system. It shows what components are present in the system and the classes associated with those components. Furthermore, it described the communication between the hardware and software, as our system includes the interaction between software and external hardware as well.

## 5: Implementation

## 5.1 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.

## 5.2 Software Description [11]

To develop this application, we had two main software development tools that could be used: Unity or Unreal Engine. We opted for Unity as it is having 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.

## 5.3 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#. One of the crucial 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.

## 5.4 Unity [12]

The main tool used for our app development as it allows creating objects, animations, sound editing, creating custom design and object, monitoring system usage, rendering, the 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 make it more efficient. An object can have many scripts attached to it and can be executed without any issue.

Framework

Unity was built by keeping component based development in mind. It has divided every single element of it into components as to provide high reusability. It separates all the scenes to separate components that are integrated by unity to develop the app. These scenes can have many objects, scripts, particles and work on their own without conflicting with other components. It removes the need for classes to interact with each other as each scene can have multiple classes attached to single object and do the task instead of relying on classes in other component to provide the required data. Since any element in unity is treated as components it’s much easier to add/ remove elements.

## 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.

## 5.6 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 the user’s device to be 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

## 5.7 User Interface Description

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

**Main Menu**



Figure 5.1 Screenshot of 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**



Figure 5.2 Screenshot of 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**



Figure 5.3 Screenshot of 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.

## 5.8 Conclusion

This chapter discussed how the app is developed, options available before development and what are the required software for development. Then we decided to set some minimum requirements for the app. A brief detail is provided for the required hardware components that are needed to run the application as expected. All the hardware requirements were analyzed and made sure those were easily accessible. Later, we described the different app phase and how they are expected to work along with the sample screenshot of the user interface.

## 6: Software Testing

## 6.1 Introduction

This section describes the nature of the system under test (SUT), the goals and scope of the testing, and the types of testing that will be performed. The testing approach and the features to be tested are also included in this section. The chapter follows the test plan from College Course Scheduler (CCS) system test plan for South Central College [13].

### **6.1.1 Nature of the Project**

The Water Cycle VR guide is being developed for students, teachers, and people who are interested in learning the earth’s water cycle. This virtual reality application will allow the user to have a visual view of the complete water cycle and its stages. The application is divided into two main features; one is the tour guide which shows the user all the stages completely and with details. Second is the game which will enhance the user’s knowledge in the water cycle, making the learning process entertaining as well as useful. The system is expected to have features that are currently not present in many applications.

The user will be interacting with the application through the gaze control and Bluetooth controller. The gaze control will allow the user to perform many functionalities such as looking around in 360o and selecting options, whereas the Bluetooth controller is mainly used to control the player in the app.

### **6.1.2 Testing Objectives and Scope [14]**

The main objectives to test the Water Cycle VR guide are to ensure that:

* All the required functionalities of the Water Cycle VR guide are well established and working correctly according to the requirements.
* The software meets its non-functional requirements.
* To identify and reveal as many errors as possible in the tested software.
* To perform the required tests efficiently and effectively, within budgetary and scheduling limitations.

### **6.1.3 Types of System Tests**

The set of system tests that will be performed in this test plan includes:

* Functional testing: all the main functions of the application will be testing. Unit testing technique will be used to test each module individually to ensure that they perform as designed.
* Non- functional testing: to test that the software meets its non-functional requirements, usability testing will be conducted, to ensure that the app is user friendly and easy to use.

## 6.2 Features to be Tested

The following features will be tested to ensure that they work as expected:

* Main menu: testing that the main menu performs its operations correctly such as proceeding to game or tour.
* Tour guide: testing that it performs as designed.
* Game: testing that it performs as designed.

## 6.3 Approach

To ensure the delivery of a high-quality application, several tests will be performed on Water Cycle VR guide before its releasing to the public to make certain that the app runs as smoothly as possible and to provide a good user experience. System tests will be performed extensively. The app will undergo functional testing to test requirements and features of the app to ascertain that they meet user requirements. All requirements must be covered by at least one test case. Non-functional testing will also be conducted to verify the non-functional requirements of the application.

### **6.3.1 Staff**

The team will be responsible for the testing phase of the project. They will design test cases and execute the tests. The students will be supervised by project supervisor.

### **6.3.2 Record Keeping**

All tests that are performed in the testing phase will be recorded. The test failures will be kept separately to be fixed and tested again.

### **6.3.3 Stop-Test Criteria**

The system testing will be concluded after covering every user requirement by at least one test case and making sure that there are no incidents or failures in the application. We acknowledge the fact that the system will still not be 100% defect free.

## 6.4 Testing Techniques

The functional requirements will be evaluated using the following black box testing technique:

Equivalence class partitioning (ECP) [15]

ECP will be mainly used in unit testing to ensure that all classes of legal inputs are accepted, for example, gazing at tour guide for 2 seconds will take the user to the tour, and all classes of illegal input are rejected by the system. This type of black box testing eliminates the need for exhaustive testing, which is not feasible. Furthermore, it allows the tester to cover a large domain of input or output with a smaller subset that is selected from an equivalence class.

Below are the equivalence classes for the following input conditions:

|  |  |
| --- | --- |
| **Condition** | **Equivalence classes** |
| The user must gaze at the start tour button for 2 seconds. | EC1: user gazes at start tour button for 2 secs. (Valid)  EC2: user gazes anywhere except the start tour button. (Invalid).  EC3: Gaze at the start game button for 2 seconds. (Invalid) |
| The user must gaze at the exit button for 2 seconds | EC4: Gaze at the Exit button for 2 sec. (Valid)  EC5: Gaze up the Exit button. (Invalid) |
| The user must press button C to repeat tour | EC6: the user presses button C. (Valid)  EC7: the user presses button B.(Invalid) |
| The Move the 360 ˚ Rocker to right to move right | EC8: the user moves the 360 ˚ Rocker to right. (Valid)  EC9: the user moves the 360 ˚ Rocker to left. (Invalid) |
| The user must press button A to return to main menu | EC6: the user presses button A. (Valid)  EC7: the user presses button D.(Invalid) |

The non-functional requirement, usability, will be tested using surveys, these surveys will be distributed to the selected users, to obtain their feedback. The feedbacks will then be analyzed and concluded.

## 6.5 Test Cases

Test cases designed for the functional testing are divided into two types; unit testing and module testing. Each type of testing has appropriate amount of test cases to ensure that the functionality and the features behave according to the requirements.

### **6.5.1 Unit Testing:**

In this test, we will be testing individual units of the system separately to ensure that each unit performs as designed. This system has been divided into three modules, so in unit testing each module will be tested individually.

**Main Menu (Module 1) Testing:**

In this test, we will perform scenario-based test case design for Start Tour Guide Use-Case:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Case Description** | **Precondition** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Result** | **Test**  **Result** |
| TC\_001 | Starting the tour guide for the water cycle | Testing the start tour guide functionality by looking at the button for 2 seconds | Launch the app. | Look at the Tour Guide button for 2 seconds. | Gaze at the start tour button for 2 sec. | Tour guide Started | Tour Guide started | Pass |
| TC\_002 | Starting the tour guide for the water cycle | Testing the start tour guide functionality by not looking at the tour guide button | Launch the app | Look anywhere but not on the Tour Guide button | Gaze anywhere expect the start tour button. | Tour Guide does not start. i.e. no action taken | Tour guide does not start | Pass |
| TC\_003 | Starting the tour guide for the water cycle | Testing the Start tour guide functionality by looking at start Game button | Launch the app. | Looks at the start game button for 2 seconds. | Gaze at the start game button for 2 seconds. | Game starts | Game starts and tour guide does not start | Pass |

Table 6.1 Test Cases for Start Tour Guide

In this test, we will perform scenario-based test case design for Start Game Use-Case:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Case Description** | **Precondition** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Test**  **Result** |
| TC\_004 | Starting the Game. | Testing the Start Game functionality by looking at the start Game button for 2 seconds. | Launch the app. | Look at the Start Game button for 2 seconds. | Gaze at the Start Game button for 2 sec. | Game is started | Game is started | Pass |
| TC\_005 | Starting the Game. | Testing the Start Game functionality by not looking at the Start Game button | Launch the app | Look anywhere but not on the Start Game button | Gaze anywhere expect the Start Game button. | Game does not start. i.e. no action taken | Game does not start | Pass |
| TC\_006 | Starting the Game. | Testing Start Game functionality by looking at Tour Guide button. | Launch the app. | Looks at the Start Tour Guide button for 2 seconds. | Gaze at the Start Tour button for 2 sec. | Tour Guide starts | Game does not start but instead tour guide starts | Pass |

Table 6.2 Test cases for Start Game

In this test, we will perform scenario-based test case design for Exit Use-Case:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Case Description** | **Precondition** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Result** | **Test**  **Result** |
| TC\_007 | Exiting the app. | Testing the Exit functionality by looking at the Exit button for 2 seconds. | Launch the app. | Look at the Exit button for 2 seconds. | Gaze at the Exit button for 2 sec. | App terminated | App terminated. | Pass |
| TC\_008 | Exiting the app. | Testing the Exit functionality by not looking at the Exit button | Launch the app. | Look anywhere but not on the Exit button | Gaze anywhere expect the Exit button. | App does not terminate. i.e. no action taken | App does not terminate | Pass |

Table 6.3 Test Cases for Exit

**Tour Guide (Module 2) Testing**:

In this test, we will perform scenario-based test case design for Repeat Tour Use-Case:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Case Description** | **Precondition** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Result** | **Test**  **Result** |
| TC\_009 | Restarting the tour | Testing the Restarting the tour functionality by pressing button C. | The user is in the tour. | The user presses Button C. | Button C | The tour is repeated. | The tour is repeated. | Pass |
| TC\_0010 | Restarting the tour | Testing the Restarting the tour functionality by pressing button A. | The user is in the tour. | The user presses button A. | Button A | The main menu appears. | The main menu appears. | Pass |
| TC\_0011 | Restarting the tour | Testing the Restarting the tour functionality by pressing button B. | The user is in the tour. | The user presses Button B. | Button B | No action taken. | No action taken. | Pass |
| TC\_0012 | Restarting the tour | Testing Restarting the tour functionality by pressing button D. | The user is in the tour. | The user presses Button D. | Button D | No action taken. | No action taken. | Pass |

Table 6.4 Test cases for Repeat Tour

In this test, we will perform scenario-based test case design for Exit to Main Menu Use-Case:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Case Description** | **Precondition** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Result** | **Test**  **Result** |
| TC\_0013 | Exiting to main menu from tour. | Testing the exiting to main menu functionality by pressing button A. | The user is in the tour. | The user presses button A. | Button A | The main menu appears. | The main menu appears. | Pass |
| TC\_0014 | Exiting to main menu from tour. | Testing the exiting to main menu functionality by pressing button C. | The user is in the tour. | The user presses Button C. | Button C | The tour is repeated. | The tour is repeated. | Pass |
| TC\_0015 | Exiting to main menu from tour. | Testing the exiting to main menu functionality by pressing button B. | The user is in the tour. | The user presses Button B. | Button B | No action taken. | No action taken. | Pass |
| TC\_0016 | Exiting to main menu from tour. | Testing the exiting to main menu functionality by pressing button D. | The user is in the tour. | The user presses Button D. | Button D | No action taken. | No action taken. | Pass |

Table 6.5 Test cases for Exit to Main Menu

**Game Mode (Module 3) Testing**:

In this test, we will perform scenario-based test case design for Move Player Use-Case:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Case Description** | **Precondition** | **Test Steps** | **Expected Results** | **Actual Result** | **Test**  **Result** |
| TC\_0017 | Moving the player. | Testing the functionality of moving the player by moving 360˚ rocker to right | The user must be in game. | Move the 360 ˚ Rocker to right | Player moved to right. | Player moved to right. | Pass |
| TC\_0018 | Moving the player. | Testing the functionality of moving the player by moving 360˚ rocker to right | The user must be in game. | Move the 360 ˚ Rocker to left | Player moved to left. | Player moved to left. | Pass |
| TC\_0019 | Moving the player. | Testing the functionality of moving the player by moving 360˚ rocker to right | The user must be in the game. | Move the 360 ˚ Rocker upwards. | Player moved to forward. | Player moved forward. | Pass |
| TC\_0020 | Moving the player. | Testing the functionality of moving the player by moving 360˚ rocker to right | The user must be in the game. | Move the 360 ˚ Rocker downward. | Player moved to backward. | Player moved to forward. | Fail |

Table 6.6 Test Cases for Move player

In this test, we will perform scenario-based test case design for Play Again Use-Case:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Case Description** | **Precondition** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Result** | **Test**  **Result** |
| TC\_0021 | Playing the game again. | Testing the play again functionality by pressing button C. | The user must be in game. | Pressing the Button C. | Button C. | Game is loaded again. | Game is loaded again. | Pass |
| TC\_0022 | Playing the game again. | Testing the play again functionality by pressing button A. | The user must be in game. | Pressing the Button A. | Button A. | User returned to main menu. | User returned to main menu. | Pass |

Table 6.7 test cases for Play again

### **6.5.2 System Testing [17]**

After performing unit testing, in which all the modules were tested one by one, we will perform system testing. System testing is the testing conducted on the system that is integrated and completed. This type of testing allows us to determine whether the system’s modules interact with each other properly or not and whether the inputs given to the system result in the desired output.

The first test case in the following table is to test the flow of the user entering the tour and exiting it.

The second test case in the following table is to test the flow of the user entering the game, playing the game and exiting it.

The third use case in the following table is to test the flow of the user entering the tour, exiting the tour, entering the game and exiting the game and then exiting the app.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Case Description** | **Precondition** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Result** | **Test**  **Result** |
| STC\_001 | Testing the flow of the user entering the tour and exiting it | The user will enter the tour by gazing at the start tour button for 2 seconds, then the user will the tour with button A. | App is launched. | User gazes at the start tour button for 2 seconds. | Gaze at the start tour button for 2 seconds. | User enters the tour. | User enters tour. | Pass |
| User presses Button A to exit the tour | Button A | User exits the tour. | User exits the tour. |
| STC\_002 | Testing the flow of the user entering the game, playing the game and exiting it. | The user will enter the game by gazing at the Start Game for 2 seconds, plays the game and then the user exits the Game with button A. | App is launched. | User gazes at the Start Game button for 2 seconds. | Gaze at the start game button for 2 seconds. | User enters the game. | User enters the game. | Pass |
| User moves the player right. | Move the 360 ˚ Rocker to right. | Player moves right. | Player moves right. |
| User moves the player left. | Move the 360 ˚ Rocker to left. | Player moves left. | Player moves left. |
| User exits the Game. | Button A | Player exits the game. | Player exits the game. |
| STC\_003 | Testing the flow of the user entering the tour, exiting the tour, entering the game and exiting the game and then exiting the app. | The user will enter the tour by gazing at the start tour button for 2 seconds, then the user will exit the tour by pressing button A, then the user will enter the game by gazing at the start game button, then the user will exit the game by pressing button A, then the user will exit the app by gazing at the exit button for 2 seconds | App is launched | User gazes at the start tour button for 2 seconds. | Gaze at the start tour button for 2 seconds | User enters the tour. | User enters the tour. | Pass |
| User presses button A to exit the tour. | Button A | User exits the tour | User exits the tour |
| The user gazes at start game button for 2 seconds. | Gaze at the start game button for 2 seconds. | User enters the game. | User enters the game. |
| The user exits the game by pressing button A. | Button A | User exits the game. | User exits the game. |
| The user Exits the app by gazing exit button for 2 seconds. | Gaze at the exit button for 2 seconds. | App is terminated | App is terminated |

Table 6.8 Test cases for the flow of the user tasks

## 6.6 Usability Testing [18]

Usability testing is conducted to determine the user’s ease of using the application and flexibility in performing tasks. The main parameters of this testing is to ensure the effectiveness of the software, if the software is easy to learn, the accuracy of the software, and the user-friendliness of the app, which makes certain that user must not undergo any training to operate the software.

### **6.6.1 Methodology**

Surveys will be used to perform usability testing. A group of 6 users will be selected and will be provided with an overview of the test procedure, surveys, and software. The participant’s responsibility is to use the app individually and provide feedback by filling in the surveys regarding the usability and acceptability of the user interface. The selected users for the testing are mostly children, aged 7-16. The survey will have questions based on user satisfaction while using the app, the ease of use, and suggestions for improvement.

### **6.6.2 Usability Testing**

The tool used, is the survey. The surveys are made in such a manner, that it’s easy for kids to understand and fill it. The questions are kept as straightforward as possible to make certain that the participants can answer accurately. The number of questions and the format of the questions are written after taking into consideration the kids’ nature, as kids do not like large amount of text and complicated language.

Below is the sample of the survey that was distributed to the participants.



Figure 6.1 Survey

### 

### **6.6.3 Test Results**

I saw or heard something entertaining



I Had Fun

Figure 6.4 Bar graph for response of I had Fun

Figure 6.3 Bar graph for response of I found the text clear

I found the text clear and easy to understand

Figure 6.2 Bar graph for response of I saw or heard something entertaining



I will use this app again

I felt comfortable using the app.

Figure 6.5 Bar graph for response of I will use the app again

Figure 6.6 Bar graph for response of I felt comfortable using the app

After analyzing the test results, we have come up to the conclusion that participants had fun and they heard or saw something entertaining based on their answers. While conducting the survey, a kid was asked to mention the entertaining part of the app, the rain and thunder sound was the entertaining part according to him. The response given to the clarity of the text was not 100% “yes” compared to the responses of other questions, rather it was 33% “I don’t know” and 19% “no”. However, the feedback is given to “I will use the app again” had a higher percentage of “yes” than “I don’t know” and “no”. Furthermore, some responses were also given for the open-ended question “Did you face any problem”, an example of the response is “yes, time-consuming”, but most of the answers were “no” to this question. Based on the results we have concluded that the app passed the usability test as the overall percentage of favorable feedbacks is more than the percentages of unfavorable feedbacks.

## 6.7 Conclusion

This chapter described the system test plan to test our system which included detailed information on the types of testing used to test the functional requirements of the software. The application was then tested using the test cases designed to find the maximum number of defects possible and to ensure that the app being delivered is bug-free.

# 

## 7: Conclusion & Future Work

## 7.1 Conclusion

The report talks about a Virtual Reality application that is made to educate students about the earth’s water cycle, allowing the users to have a visual view of the water cycle and its stages. Earlier students were taught this concept using texts and images, which isn’t very elaborative and easy to understand. Therefore, this type of technology was used to explain these concepts to students in a better way, as water cycle is a very crucial concept. Based on the domain study carried out in this report, there isn’t any such app made yet which combines the visual view of earth’s water cycle and a mini game. Whereas, this application has a tour guide of the complete water cycle and a mini game to polish up user’s knowledge in the field. The report also discloses the design of the application clearly, to minimize the effort in the implementation phase. Furthermore, a detailed test plan is provided in the report, to ensure the delivery of a high-quality application. The testing has identified and revealed as many errors as possible and made certain that the software met its functional and non- functional requirements. After providing a thorough report and putting in intensive work on the development of the app, we hope that the report provides absolute and clear instructions about our system from an abstract to a detailed level.

**Difficulties faced during development**

As most of the work done to develop the app was not in the course plan of our major. We had to start from scratch and look around for available resources, we had to look at the online community to rely on, as we had not background knowledge of game development, unity, C# language. At beginning it was quite difficult to get a grasp of all the information as we had no clue of how unity works, how the scripting is done or how to create model and animation. Another major challenge we faced was, that the unity changed its language support and the online community support was less reliable as most of the code available were outdated and we had to learn syntax along with the older functions used to created new scripts for same functionality. Once we overcame that challenge, we looked forward to creating custom animation and event trigger to take place at certain time as to point the user to notice.

**Outcomes/Learned**

During the project development we have learned a lot of things that we couldn’t achieve through courses, as most of the part of this project was our first time to develop in a completely different environment and different language. Other benefit was that we learned how to interact with online community to find the required information. Have real life experience of how a component-based software works and how good it can be if used in correct way. As if we had any other program we might have had a lot of issues in separating components, passing values between classes , creating animation whereas unity being a powerful software in many aspects led us to use it to our advantage as we developed different parts of our app simultaneously without any issue. Moreover, we learned how to modify objects’ details, animations, add custom objects to our assets, store data variables locally to save important information locally.

## 7.2 Future Work

As of the future, we plan to extend our efforts and improve the app graphics along with the better quality of hardware that is quite difficult to implement as of now. We would prefer to work more upon implementation text-to-speech as it will make it easy for user to understand the context instead of reading it and giving us a bit of extra room so that we can add few more animations and make it more appealing for the user when they see it as currently most of our space is taken away by the text displayed and robot. We would prefer having a better opportunity at improving the app by adding more language, different stages, different game mode as it will make it more interesting.

## References

|  |  |
| --- | --- |
|  | 1. Bardi, J. (2019). *What is Virtual Reality? VR Definition and Examples | Marxent*. [online] Marxent. Available at: https://www.marxentlabs.com/what-is-virtual-reality/?fbclid=IwAR1So2j0oR7WPzTQIX2Rozm3i3E\_5CXkqibiUICW6numh4uhW-3R-QLc2hw [Accessed: 21-Sep-2019]. 2. Pmm.nasa.gov. (2019). *Why are water cycle processes important? | Precipitation Measurement Missions*. [online] Available at: https://pmm.nasa.gov/resources/faq/why-are-water-cycle-processes-important. [Accessed: 19-Sep-2019]. 3. Freedrinkingwater.com. (2019). *The Water Cycle: A Guide For Student | APEC Water*. [online] Available at: https://www.freedrinkingwater.com/resource-water-cycle-student-guide.htm [Accessed: 19-Sep-2019]. 4. Lucidchart.com. (2019). *4 Phases of Rapid Application Development Methodology | Lucidchart Blog*. [online] Available at: https://www.lucidchart.com/blog/rapid-application-development-methodology?fbclid=IwAR3Jl9l6jUFhlIvRBCmb7q96mIDXEqvn-yBX-RC4gqdnL-c3eTbghGMcddc [Accessed: 10-Jul-2019]. 5. Sukhin, S. (2019). *Splash Sim: A VR Experience in the Water Cycle - Apps on Google Play*. [online] Play.google.com. Available at: https://play.google.com/store/apps/details?id=com.splashsim.splashsim&hl=en\_US [Accessed 28 7 2019]. |
|  | 1. A. V. Education (2019). *Cicle de l'Aigua VR - Apps on Google Play*. [online] Play.google.com. Available at: https://play.google.com/store/apps/details?id=com.allvreducation.cicledelaigua&hl=en. [Accessed 28 7 2019]. |
|  | 1. EUROCASE (2019). *TABI The Water Cycle - Apps on Google Play* [online] Play.google.com. Available at: https://play.google.com/store/apps/details?id=com.eurocase.hydrologic\_cycle&hl=en. [Accessed 28 7 2019]. |
|  | 1. S. Xpert (2019). *วัฏจักรน้ำ (Water Cycle) - Apps on Google Play*," [online] Play.google.com. Available at: https://play.google.com/store/apps/details?id=com.studioxpert.watercycle. [Accessed 28 7 2019]. 2. Visual-paradigm.com. (2019). What is Component Diagram?. [online] Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-component-diagram/ [Accessed 10 Sep. 2019]. 3. &rarr;, V. (2019). Deployment Diagram Tutorial | What is a Deployment Diagram. [online] Creately Blog. Available at: https://creately.com/blog/diagrams/deployment-diagram-tutorial/ [Accessed 15 Oct. 2019]. 4. Gamedesigning.org. (2019). [online] Available at: https://www.gamedesigning.org/engines/unity-vs-unreal/ [Accessed 18 Oct. 2019]. 5. Unity. (2019). Products - Unity. [online] Available at: https://unity3d.com/unity?\_ga=2.170247008.1388680148.1574509645-825276724.1574509645 [Accessed 7 May 2019]. 6. J. Smith, "College Course Scheduler (CCS) System Test Plan," Windy City Development Corporation, 2001. 7. GeeksforGeeks. (2019). Types of Software Testing - GeeksforGeeks. [online] Available at: https://www.geeksforgeeks.org/types-software-testing/ [Accessed 14 Aug. 2019]. 8. Software Testing Material. (2019). Equivalence Partitioning Test Case Design Technique. [online] Available at: https://www.softwaretestingmaterial.com/equivalence-partitioning-testing-technique/ [Accessed 20 Aug. 2019]. 9. The Economic Times. (2019). What is System Testing? Definition of System Testing, System Testing Meaning - The Economic Times. [online] Available at: https://economictimes.indiatimes.com/definition/system-testing [Accessed 20 Sep. 2019]. 10. Experience UX. (2019). What is usability testing?. [online] Available at: https://www.experienceux.co.uk/faqs/what-is-usability-testing/ [Accessed 19 Oct. 2019]. |
|  |  |