Test plan

Introduction

This section describes the nature of the system under test (SUT), the goals and scope of the testing, and the types of the testing that will be performed. The testing approach and the features to be tested are also included in this section.

Nature of the Project

The Water Cycle VR guide is being developed for students, teachers and for 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 user’s knowledge in the water cycle, making the learning process entertaining as well as useful. The system is expected to have features which are currently not present in much 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 Bluetooth controller is mainly used to control the player in the app.

Testing objectives and scope

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.

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.
* Performance testing: to test that the software meets its non-functional requirements.

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 and designed.
* Game: testing that it performs as designed.

Approach

In order to ensure the delivery of a high-quality application, several tests will be performed on Water Cycle VR guide prior to 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 in order to ascertain that they meet user requirements. All requirements must be covered by at least one test case.

Staff

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

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.

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.

Testing techniques:

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

Equivalence class partitioning (ECP)

State transition testing

ECP will be mainly used in unit testing to insure 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 types 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. State transition testing will also be used to cover all paths, to enter all states and to examine every possible state transition, for example, the state transition in our app from main menu to the game. This will help us to make certain that the flow between the states is on the mark. Many transitions that look ambiguous or haven’t been tried before will be tested in order to ensure that the system works as designed.