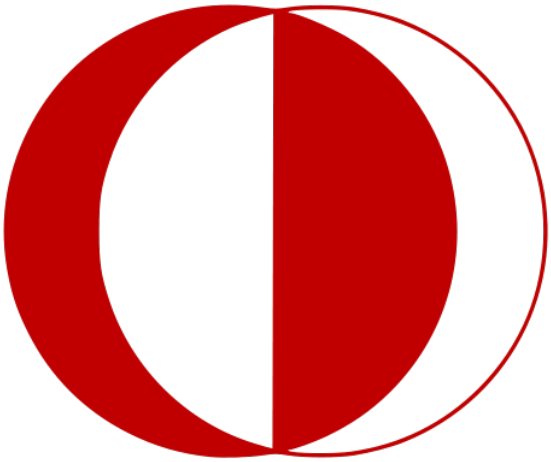
Middle East Technical University

Department of Computer Engineering



CENG 492

SOFTWARE TEST DOCUMENT

v1.0

DYNADRAW

Date : 03 June 2016

Prepared By : Barış Özçelik - 1881804

Bünyamin Sarıgül - 1881440

Turan Soyuer - 1881481

Cemil Kocaman - 1881333

Contents

1. Introduction

1.1 Document Identifier

"DynaDraw" team members prepared this document for testing all functionalities and properties of "Drawing Based Platform Game". This Software Test Document is the first document for explaining all of test cases which done to "Drawing Based Platform Game" so the version of the document is "Version 1.0". In addition to this, "IEEE Std 829-2008 Standard" was the reference of this Software Test Document. Notice that this document includes all of test cases and outcomes of functionalities.

1.2 Scope

This document was prepared to test all of functionalities and outcomes of "Drawing Based Platform Game". Moreover, the main purpose of the Software Test Document is evaluating the test cases that was done, and deciding whether outcomes are reliable (true) or not. In this system, there are two types of user, which are normal people, and visually impaired people so there are two types of interfaces. (The only interface that system is controlled voice and normal user interface). Because of that reasons, these interfaces and their functionalities are tested. Moreover, the software will be tested and evaluated using guidance. In addition to this information, inspection, analysis, demonstration and validation will be considered while test phase will be completed.

1.3 References

- DynaDraw SRS
- DynaDraw SDD
- IEEE 829-2008 Standard

2. Details for System Test Plan

This section describes the specific items to be tested at different levels and provides a Test Traceability Matrix that links the items to be tested with the requirements.

2.1 Test Items And Their Identifiers

Since Game is includes different scenes, and all scenes have different components, all of these scenes is different case of testing. In addition to that, because of Unity is based on scripting language

programming, all scripts and their compatibility with game components should be included to tests as well.

Test cases were done in both Android, Web and PC Platform with visual test. This means that any test case does not try to black-box method. All of these methods are listed below.

Start Scene

- Case 1 : Do Visual Components Work Properly on All Platforms?
- Case 2: Do Visual Components Look Well on Different Resolutions?
- Case 3: Does Start Button Works?

Drawing Scene

- Case 1: Do Drawing Components Work Properly?
- Case 2: Does Pencil Button Work?
- Case 3: Does Eraser Button Work?
- Case 4: Do Color Buttons Work?
- Case 5: Outlining and Shading Over Color Buttons
- Case 6: Does Undo Button Work?
- Case 7: Does Clear Button Work?

Loading Scene

- Case 1: Are Animation Sprite Sheets Produced Properly?
- Case 2: Does Play Button Work?

First Level Scene

- Case 1: Does Character Animation Work Properly?
- Case 2: Does Character Movement Work Properly?
- Case 3: Does Animal Animation Work Properly?
- Case 4: Does Animal AI Work?
- Case 5: Does Drawing/ Movement Button Work?
- Case 6: Does Restart Work?
- Case 7: Do Colliders of the Game Objects Work Properly?
- Case 8: Do RigidBodies of the Game Objects Work Properly?
- Case 9: Does PickUpItem Function Work?

Second Level Scene

All of Test Cases in First Level Scene is also tested for Second Level Scene.

2.2 Features to be Tested

In this document, all of the features to be tested were listed section 2.1. Notice that all of the features, mentioned above, were tested with user interface. In addition to this, all of the information about features to be tested was found in “Software Requirements Specification” document of “Drawing Based Platform Game”.

2.3 Approach

The approach that is applied in this document for testing features can be described as manual testing. This means that there is no any test method that can be associated with black-box method. For completing all of test case and evaluating whether their outcomes are reliable or not, manual tools are used. This means that if system gives continuously the same results, then test was true. Otherwise it was false. The main approach of this document is trying the same things in different platforms as Web, PC, Android to take same results.

2.4 Item Pass/Fail Criteria

As we mentioned above, the main criteria of the project is whether the system gives same results or not in different platforms as Web, PC, Android. If outcomes that are taken under these conditions are same, then item pass. Otherwise items fail. Notice that fail test case examined and fixed before test start over again.

3. Test Management

3.1 Planned Activities and Tasks/ Test Progression

In our test progression, all of test cases are tested in terms of systems and alone. This means that all of test cases are tested alone and if this test case is reliable for all trials then related test case referred to reliable test. After this process completed successfully, another test approach are done. This is the most important part of test progression to “Drawing Based Platform Game”. Another test approach is that referred use case is tested in all of system. This is the true approach for “Drawing Based Platform

Game" because whether this test case breaks other features or not is the important issue. If related use case works properly in all system and does not break any other features then it is referred as reliable test case. Reliable test case means that related use case passes the entire test. This situation are tried to all of functionalities more than once because we are sure that related use case literally works properly in different platforms as Web, PC, Android. Notice that if tested test case becomes fail when trying alone or in all system, then appropriate error is tried to find and fix; and all steps of test progression start anew. If all of tested steps are completed and all of steps give successful report then task is referred as reliable. Notice that all of different task is tested before integration the system. Thus, the test progression applies double testing. One the double testing is unit testing which is explained above and another is system testing which testing tasks after implementation of each module of the system. In addition to this, this test progression provides us to maximum information about tested tasks with minimum effort.

3.2 Environment/ Infrastructure

As mentioned above, we used an Unity Platform. Moreover, we used different platforms as Web, PC, Android. This means that we make ensure that the application works properly in different infrastructures.

4. Test Case Details

This section will explain the detailed information about the test case for each function requirement. Each test case includes identifier, objective, inputs, outcomes, environmental needs, special procedural requirements and intercase dependencies.

4.1 Start Scene

4.1.1 Do Visual Components Work Properly on All Platforms? (This Test is Applied to All Scenes)

Test Case Identifier	Do Visual Components WorkProperly on All Platforms
Objective	To display all visual objects properly in all platforms
Input	Change the platform and run the game
Outcome	All objects look well on all platforms

4.1.2 Do Visual Components Look Well on Different Resolutions? (This Test is Applied to All Scenes)

Test Case Identifier	Do Visual Components LookWell on Different Resolutions
Objective	To display all visual objects well-proportioned in different resolutions
Input	Change the resolutions to observe changes on the transformaiton of objects
Outcome	All objects look well on different resolutions

4.1.3 Does Start Button Work?

Test Case Identifier	Does Start Button Work
Objective	To change the scene when start button is clicked
Input	Click the start button
Outcome	The scene is changed to the next scene

4.2 Character Drawing Scene

4.2.1 Do Drawing Components Work Properly?

4.2.1.1 Does Pencil Button Work?

Test Case Identifier	Does Pencil Button Work
Objective	To swich drawing mode when pencil button is clicked

Input	Click the pencil button
Outcome	The mode is switched to the drawing mode

4.2.1.2 Does Eraser Button Work?

Test Case Identifier	Does Eraser Button Work
Objective	To swich erasing mode when eraser button is clicked
Input	Click the eraser button
Outcome	The mode is switched to the erasing mode

4.2.1.3 Do Color Buttons Work?

Test Case Identifier	Do Color Buttons Work
Objective	To change the brush color when a color button is clicked
Input	Click red, yellow, green, etc. color buttons
Outcome	The brush color is changed to the selected color

4.2.1.4 Outlining and Shading Over Color Buttons

Test Case Identifier	Outlining and Shadowing Over Color Buttons
Objective	To add outline on the selected color button and shadow on the other buttons.

Input	Click red, yellow, green, etc. color buttons
Outcome	The selected button is outlined andothers are shaded

4.2.2 Does Undo Button Work?

Test Case Identifier	Does Undo Button Works
Objective	To remove the last drawn line
Input	Click undo button
Outcome	The last line that the user drew is removed

4.2.3 Does Clear Button Work?

Test Case Identifier	Does Clear Button Works
Objective	To remove all lines drawn
Input	Click clear button
Outcome	All lines are removed

4.3 Loading Scene

4.3.1 Are Animation Sprite Sheets Produced Properly?

Test Case Identifier	Are Animation Sprite Sheets Produced Properly
Objective	To create animation sprite sheets for character

Input	The point data of the character drawn
Outcome	The sprites of character animation frames

4.3.2 Does Play Button Works?

Test Case Identifier	Does Play Button Work
Objective	To start the game when play button is clicked
Input	Click the play button
Outcome	The scene is changed to the tutorial level

4.4 Tuturial Level Scene

4.4.1 Does Character Animation Works Properly?

Test Case Identifier	Does Character Animation Works Properly
Objective	To look whether the character is animated well
Input	Move the character
Outcome	The character is animated

4.4.2 Does Character Movement Works Properly?

Test Case Identifier	Does Character Movement Works Properly
Objective	To move character

Input	Click the left side or right side of character
Outcome	The character is moved

4.4.3 Does Animal Animation Work?

Test Case Identifier	Does Animal Animation Work
Objective	To animateanimal well
Input	Move the character to make the animal follow it
Outcome	The animal is animated

4.4.4 Does Animal AI Work?

Test Case Identifier	Does Animal AI Work
Objective	To look whether the animal follows the character
Input	Move the character
Outcome	The animal follows the character

4.4.5 Does Drawing/Movement Button Work?

Test Case Identifier	Does Drawing/Movement Button Work
Objective	To switch the game mode from drawing/movement to other one
Input	Click Drawing/Movement Button

Outcome	The game mode is switched

4.4.6 Does Restart Work?

Test Case Identifier	Does Restart Work
Objective	To restart game when the character falls down
Input	Character falls down
Outcome	The game is restarted from latest points if there is any live of charcter OR if not, the game is restarted from the beginning of the level

4.4.7 Do Colliders of Game Objects Work Properly?

Test Case Identifier	Do Colliders of Game Objects Work Properly
Objective	To collide game objects when they touch each other
Input	1) Draw a line above of a cube 2) Draw a line above the character 3) Move the character on a cube 4) Move the character on a line
Outcome	The all of the colliders work properly

4.4.8 Do Rigid Bodies of Game Objects Work Properly?

Test Case Identifier	Do Rigid Bodies of Game Objects Work Properly
Objective	To add rigid bodies to game objects

Input	1) Draw a line at an empty point 2) Move the character to a space
Outcome	The character and drawn lines fall at an empty point, so rigid bodies work properly.

4.4.9 Pick Up Item Function Work Properly?

Test Case Identifier	Do Pick Up Item Function Work Properly
Objective	To make the character collect some items
Input	Move the character towards an item
Outcome	The character collects the items and the score increases

5. System Test Report Details

5.1 Overview of the test results

In test phase of all tasks that identified previous sections, it is observed that all of test cases are almost works properly. This means that all of tested tasks are applied and integrated system properly so the result of the test phase is as we expected. On the other hand, it is not possible to say that “Drawing Based Platform Game” literally works properly. On the other hand, if some cases that the system behaves unexpectedly are observed then, firstly, use case that has an error will be fixed and described test progression will be applied.

5.2 Detailed test results

Do Visual Components Work Properly on All Platforms	**Passed**
Do Visual Components LookWell on Different Resolutions	**Passed**
Does Start Button Work	**Passed**
Do Drawing Components Work Properly	**Passed**
Does Pencil Button Work	**Passed**
Does Eraser Button Work	**Passed**
Do Color Buttons Work	**Passed**
Outlining and Shadowing Over Color Buttons	**Passed**
Does Undo Button Work	**Passed**
Does Clear Button Work	**Passed**
Does Play Button Work	**Passed**
Are Animation Sprite Sheets Produced Properly	**Passed**
Does Character Animation Work Properly	**Passed**
Does Character Movement Work Properly	**Passed**
Does Animal Animation Work Properly	**Passed**
Does Animal AI Work	**Passed**
Does Drawing/ Movement Button Work	**Passed**

Does Restart Work	**Passed**
Do Colliders of the Game Objects Work Properly	**Passed**
Do RigidBodies of the Game Objects Work Properly	**Passed**
Does PickUpItem Function Work	**Passed**

5.3 Rationale for decisions

Rationale for decisions is the maximizing the sequence of tests with the minimum effort.

5.4 Conclusions and Recommendations

As a result, all of tasks and use cases are tested properly and tested tasks passed test progression. This means that it can be easily said that the steps up until now were developed properly. However, there may be some problems in such test phase that applied features not tested. After this process we should develop the remaining features carefully and all of these features should be tested in detail. Notice that all of these features should be tested with module test and integration test because of maintaining the integrity of the system.