# Updated **Test Plan and Cases (TPC)**

**Doodle Dash**

**Candice Academia**

**Johnny Chi**

**Nicholas Ruiz**

**Robert Steiminger**

**Stephen Sing**

**Table of Contents**

Cover………………………………………………………………………………………………………….1

Table of Contents………….………………………………………………………………………………….2

Table of Figures………….………………………………………………………………..………………….3

Introduction………….………………………………………………………………..……………..……….4

Test Strategy and Preparation………………….……….……………………………..……………..……….4

Hardware Preparation……………….………………………………………….……..……………..……….4

Software Preparation………….………………………………………….……….…..……………..……….5

Other Pre-test Preparation………….……………………………………….….……..……………..……….5

Requirements Traceability………….…………………………..………….….……..……….……..……….5

Test Cases………….……………………………………….….……………………...……………..……….5

Test Identifier………….………………………………….….……………………...……………..………...6

Test Level………….………………………………….….……………………...……………..………..…...7

Test Class………….………………………………….….……………………...……………..……..……...7

Test Completion Criteria………….………… ……….….……………………...……………..……..……...7

Test Cases………….………………………………….….……………………...……………..……..……...7

Resources and Schedule…………………………….….……………………...……………..……..…...…...8

Resources……………..…………………………….….……………………...……………..……..…...…...8

Staffing and Training Needs…………………………....……………………...……………..……..…...…...8

Schedule…………………………….……………….….……………………...……………..……..…….....8

# **Table of Figures**

Figure 1: Table 1: Requirements Traceability Matrix 5

Figure 2: TC-01-01 5

Figure 3: TC-01-02 6

Figure 4: Testing Schedule 8

### **Introduction**

We created a game that we feel appeals to all ages. The testing we used allowed us to incrementally make sure that each aspect of the game was up the par and ready to ship.

We thoroughly tested any game breaking bugs, and fixed them as soon as possible. Some of the bugs we encountered were a blue menu screen, disappearing buttons, and levels with infinite falls.

The main purpose of our testing was to view the game through the eyes of a user and see what was working and what felt “broken.” When we played the game, we imagined ourselves as new users experiencing it for the first time.

With our testing schedule, we were able to catch a multiple bugs before they showed up in the final version of the game. We caught worst case scenarios before they were able to present a fatal problem in the game.

### 

### **Test Strategy and Preparation**

We tested each feature as they were added into the game. This sequential approach allowed us to catch problems as they arose. We were able to implement this strategy due to the fact that for each increment in the completion of the game, new bugs arose and were taken care of as soon as they were found.

#### **Hardware preparation**

In order to test the project, team members were able to use the same software that was used to build the game Unity.

Not all bugs can be found in Unity, so to make sure we found mobile-specific bugs, we also loaded an apk onto various smartphones owned by the team members.

#### **Software preparation**

We used Unity for both building the game and the majority of testing. The simplicity of using the same software for implementation and testing allowed us a quick and seamless transition between the two modes of thought.

#### **Other pre-test preparations**

In order to ensure that we came into the game with the same mindset as a new user, we reviewed the advertised features of the game and tracked if they were successfully implemented.

#### **Requirements Traceability**

***Table 1: Requirements Traceability Matrix***

|  |  |  |
| --- | --- | --- |
| ***Requirement ID*** | ***Verification Type*** | ***Test Case ID (if applicable)*** |
| *Game starts: Complete* | *Demonstration* | TC-01-01 - Working Prototype |
| *Interface responds to user input: Complete* | *Testing* | TC-01-01 - Working Prototype |

##### **Test Cases**

Test 1: Successful

**Table 2: TC-01-01 Check report completeness**

|  |  |
| --- | --- |
| Test Case Number | TC-01-01 - Working Prototype |
| Test Item | *In this test we will test to make sure the game starts up at a minimum.* |
| Test Priority | *This test is critical to proceed in the project. It must have satisfactory results.* |
| Pre-conditions | *Prior to this test, the game needs to be constructed in unity.* |
| Post-conditions | *After this test, it must be established that the game runs.* |
| Input Specifications | *The input for this case will be the user starting the app.* |
| Expected Output Specifications | *The expected results are that the game starts up and displays a prototype splash screen.* |
| Pass/Fail Criteria | *A pass is the game starting without problems.*  *A fail would be the game crashing on startup or failing to load.* |
| Assumptions and Constraints | *We assume that the user has a compatible smartphone and operating system for the game.* |
| Dependencies | *This is a beginning test case.* |
| Traceability | *This traces to the origin of the game.* |

**Test 2**

Test 2: Successful

|  |  |
| --- | --- |
| Test Case Number | TC-01-02 - Finished game bugfixes |
| Test Item | *We will use this test to identify and solve bugs added by all the features in the game.* |
| Test Priority | *This test is of high importance.* |
| Pre-conditions | *Before the test case, we assume that all desired features were implemented in unity.* |
| Post-conditions | *After the test case, we assume that the game is ready to be shipped to end users.* |
| Input Specifications | *The input necessary is the user playing the game normally.* |
| Expected Output Specifications | *The expected results are that the game runs smoothly with all features* |
| Pass/Fail Criteria | *The test passes if the game fulfills all requirements and features.*  *The test fails if one or more of the features are not able to be implemented.* |
| Assumptions and Constraints | *We assume that new features have been added to the game since its inception* |
| Dependencies | *This test depends on test case* TC-01-01 already having been successful. |
| Traceability | *This test maps to the end of the development cycle.* |

#### **Test Identifier**

TC-01-01 - Working game prototype: Completed 3/28/17

TC-01-02 - Finished game bugfixes: Completed 4/4/17

TC-01-03 - Public response gathering: Completed 4/13/17

##### **Test Level**

TC-01-01 - Basic level

TC-01-02 - Advanced level

TC-01-03 - Basic level

##### **Test Class**

TC-01-01 - Unity class

TC-01-02 - Unity class

TC-01-03 - End user experience class

##### **Test Completion Criteria**

TC-01-01 - Test completed once a working prototype is achieved.

Update: TC-01-01 Completion Criteria Achieved 3/28/17

TC-01-02 - Test completed when all found bugs are fixed.

Update: TC-01-02 Completion Criteria Achieved 4/4/17

TC-01-03 - Test completed when a viable sample of user’s opinions has been gathered.

Update: TC-01-3 Completion Criteria Achieved 4/13/17

##### **Test Cases**

TC-01-01 - The game starts up. User is able to perform basic action.

Update: TC-01-01 Test Case Completed 3/28/17

TC-01-02 - The game starts up, user is able to perform multiple complex actions and complete the game.

Update: TC-01-02 Test Case Completed 4/4/17

TC-01-03 - Outside user starts game and is able to play for an adequate amount of time.

Update: TC-01-02 Test Case Completed 4/13/17

### **Resources and schedule**

The teammates all supplied their regular laptops for Unity testing, eliminating the cost of any dedicated machines. For the Android testing, the team was able to use some old and current Andoid phones for a wide variety of model testing, ensuring a high compatibility.

#### **Resources**

We will need personal computers capable of running Unity as well as mobile phones for testing. Team members will each be bringing their currently owned machines to the resource pool.

#### **Staffing and Training Needs**

Because each of the team members had a hand in the project itself, testing was a natural step for each person. No additional training needed to be given as long as the team member had Unity.

Some of the specific problems in our game were fixed by the users working on that current portion of the project. Some specific technical problems that were fixed were disappearing menu buttons and the missing background screen.

#### **Schedule**

**Table 4: Testing Schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | ***Test Identifier*** | ***Responsible person*** | ***Resources*** | ***Training needs*** |
| 3/28/17 | *TC-01-01* | *Team members* | *Unity* | *Unity development* |
| 4/4/17 | *TC-01-02* | *Team members* | Unity | Unity bug-fixing |
| 4/13/17 | *TC-01-03* | Focus group | Mobile phone | Mobile phone operation |

|  |  |  |
| --- | --- | --- |
| Test Number | Complete/Not complete | Date Tested |
| *TC-01-01* | Complete | 3/28/17 |
| *TC-01-02* | Complete | 4/4/17 |
| *TC-01-03* | Complete | 4/13/17 |