Test Plan and Cases (TPC)

Doodle Dash

Candice Academia

Johnny Chi

Nicholas Ruiz

Robert Steiminger

Stephen Sing

Table of Contents

Table of Figures

Introduction

The purpose of our project is to deliver a game that will be successful in a competitive market. We desire to create something that will be fun for all ages. We will be testing the game extensively and making adjustments for maximum desirability to the user.

To test, we will use all the menus of the game as well as simulate the possible states. The intention will be to find any possible bugs or game-breaking aspects.

The focus of the testing will be in regards to the user's experience. In order to put ourselves in the mind of the end user, we will be playing the game with the mindset of a player and rating the satisfaction we experience.

The testing will be of top scrutiny, so that we can locate and fix any problems before the game makes it to market. We will do extensive use-case testing, as well as worst-case scenario testing.

Test Strategy and Preparation

The strategy we will use to test is sequence of build. This means that testing will be done as more features are integrated into the game. This ensures that we know exactly what needs to be fixed at every iteration. It also means we will know immediately know what works and what does not, allowing us to better achieve our targeted goals.

Hardware preparation

In order to prepare the hardware for this project, we will install Unity on each team member's machines. We will also install graphics software for the team members producing art. Adobe Photoshop or Illustrator could be used for this purpose.

We also will need to make sure each team member has a machine capable of running the game at each iteration. When the game is complete, we will need a physical smartphone to test it on.

Software preparation

The software we will be using is Unity. The purpose of this is that it is great for building

games, and has the features that we desire.

Other pre-test preparations

We will need to make sure the testers know the controls and scope of the game, as well as the desired experience of the user.

Requirements Traceability

Table 1: Requirements Traceability Matrix

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

Test Cases

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 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

TC-01-02 - Finished game bugfixes

TC-01-03 - Public response gathering

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.

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

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

Test Cases

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

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

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

Resources and schedule

Since we will all be able to acquire Unity, budget for testing is essentially $0. We will be testing incrementally over the course of the semester and the development lifespan of the project. Testing will likely take less than an hour at a time but will be done frequently as more features are being built in.

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

The specific skill level needed for participation in testing is low to medium. For basic testing of the finished or nearly finished game, all that would be required is feedback from the tester. For this purpose, the tester could be anybody.

For the incremental testing, basic to moderate techinical knowledge is needed. For this, we will be using the team itself. This is due to the fact that we need to identify specific features or bugs going wrong in the game and exactly what is causing a problem and how to fix it.

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