SE 3XA3: Test Plan
Staroids

Team 20, Staroids
Eoin Lynagh, lynaghe
Jason Nagy, nagyj2
Moziah San Vicente, sanvicem

October 24, 2018

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Oct 22 2018	1.0	Added Purpose, Test Team, Scope, a couple Acronyms, abbreviations, and symbols
Oct 24	1.1	Added Software Description, Overview of Document, Automated Testing Approach

1 General Information

1.1 Purpose

This document is designed to show the detailed test plan for the Staroids game. This will include a description of the testing, validation, and verification procedures that will be implemented. All the tests in this document have been created before the final implementation has been completed and any tests have actually occured, so it will be the guide followed during the testing phase of the project.

1.2 Scope

The scope of the test plan is to provides a basis for testing the functionality of this re-implementation of asteroids. It has the objective of proving all the functional and non functional requirements listed in the SRS document.

1.3 Acronyms, Abbreviations, and Symbols

Table 2: **Table of Abbreviations**

Abbreviation	**Definition**
POC	Proof of Concept
SRS	Software Requirements Specification

1.4 Overview of Document

This Test Plan's main goal is to inform on how Staroids is tested for correctness about its various objectives and requirements. All tools are stated and their use case is explained. All planned test cases are also listed that are used to verify the correctness of Staroids with regards to its functional and non-functional requirements.

Table 3: **Table of Definitions**

Term	**Definition**
Functional Testing	Input-Output type of testing apporach known Input, expexted Output
Static Testing	Just looking at code, no actual execution
Dynamic Testing	Testing that requries code execution
Structural Testing	A whitebox type of testing Approach so cases are derived from internal structure of the software
Automated Testing	Testing is handeled by the testing framework (JUnit) (testing done by software)
Manual Testing	Manual individually written test cases. (testing done by people)
Stress Test	Testing the limits of a system, usually refers to amounts of data the system can handle

2 Plan

2.1 Software Description

Staroids is a recreation of the HTML-5 Asteroids created by Doug McInnes which itself is a recreation of the Asteroids arcade game. It allows the user to pilot a space ship through a rectangular piece of space with wrapping edges. Cohabiting with the space ship, there are several asteroids and an alien. These entities are considered hostile to the space ship and will damage it if they come into contact. The space ship and alien can defend themselves from any hostiles by firing at them with a laser bullet. This will damage anything (except the shooter) that the bullet comes into contact with. Staroids allows the user to have 3 lives and keeps a running score for the current game that is based off of how many asteroids and aliens have been destroyed.

2.2 Test Team

The test team for this project consists of the following members who are each responsible for writing and executing tests for modules later to be specified:
- Moziah San Vicente

- Eoin Lynagh
- Jason Nagy

2.3 Automated Testing Approach

Automated tests are to be used for all game situations where the situation result is expected to be the same for every situation. Since the result is always true, these tests can be quickly run after every Staroids edit to ensure that no game functionalities have been broken by the edits. These tests are not done automatically at game for several reasons. Firstly, if there is an assertation error, it would be meaningless to the user and there would be no action done on the user's part. Additionally there would be time added to the startup with no benefit to the user. Therefore, all major versions of Staroids will run through the test cases before they are pushed as a final Staroids build.

2.4 Testing Tools

2.5 Testing Schedule

See Gantt Chart at the following url ...

3 System Test Description

3.1 Tests for Functional Requirements

3.1.1 Area of Testing1

Title for Test

1. test-id1

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.
   Initial State:
   Input:
   Output:
   How test will be performed:

3.1.2 Area of Testing2

...

3.2 Tests for Nonfunctional Requirements

3.2.1 Area of Testing1

Title for Test

1. test-id1

   Type:
   Initial State:
   Input/Condition:
   Output/Result:
   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.
   Initial State:
   Input:
   Output:
   How test will be performed:

3.2.2 Area of Testing2

...

3.3 Traceability Between Test Cases and Requirements

4 Tests for Proof of Concept

4.1 Area of Testing1

Title for Test

1. test-id1

   Type: Functional, Dynamic, Manual, Static etc.
   Initial State:
   Input:
   Output:
   How test will be performed:

2. test-id2

   Type: Functional, Dynamic, Manual, Static etc.
   Initial State:
   Input:
   Output:
   How test will be performed:

4.2 Area of Testing2

...

5 Comparison to Existing Implementation

6 Unit Testing Plan

6.1 Unit testing of internal functions

6.2 Unit testing of output files

7 Appendix

This is where you can place additional information.

7.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance. **Constants**
FPS = 30; The curent frames per second
SHIP_SIZE = 30; The ship size in pixels
TURN_SPEED = 180; Player turn speed in degrees per second
SHIP_THRUST = .2; Player thrust power in pixels per second squared
SHIP_BRAKE = 0.98; player airbrake power (¡0.9 = full stop 1 = no brake)
MIN_SPEED = 0.1; minimum speed
MAX_ACC = 2; maximum ship acceleration
MAX_SPEED = 20; Maximum ship speed (velocity)
CVS_WIDTH = 500; canvas width
CVS_HEIGHT = 400; canvas height
BULLET_EXTRA = 5; Extra velocity on bullet on top of ship's velocity
KILLABLE = true; Testing invulnerability
MAX_ASTEROIDS = 2; Maximum amount of asteroids
TEST=false; experimental features

7.2 Usability Survey Questions?

This is a section that would be appropriate for some teams.