King of Tokyo Test Plan

Version 0.1
October 28, 2019

Project Team:
Peter Bui, Software Engineer
Taylor Bui, Software Engineer

Document Authors:
Peter Bui
Taylor Bui

Project Sponsor:
CSULB

I. Introduction

This document serves as a plan for testing all software artifacts as well as the reporting of test results. This test plan will attempt to provide a solid footing for ensuring that the test cases will be properly evaluated. Test cases will make it easier to ensure a high quality product. Additionally, it will make coding the game substantially easier as unit tests will allow the team members to pinpoint problematic sections of the code.

II. Test Plan

Please refer to the Test Case Specification spreadsheet for detailed information on the test cases.

III. Testing Deliverables

1. Test Plan Specification - document outlining the testing procedures
2. Test Case Specification - spreadsheet that details test case procedure
3. Test Procedure Specification - how the program is tested (input/output, state, etc)
4. Test Log - text document outlining the specific errors encountered by the unit test
5. Test Incident Report - team members will report on testing incidents
6. Test Summary Report - team members collaborate to give reports on testing

IV. Environmental Requirements

The team members will be using their personal devices to conduct the tests. Communication will primarily be through Discord. The Unity suite will need to be installed on the personal devices and the inbuilt testing tool will be used. Little to no security is necessary for this environment because there will be no sensitive information exchanged in the final product nor during testing.

V. Staffing

Ata minimum, tests will be performed weekly by a team member who will alternate each week. The team members will discuss the breadth of the testing, but it is likely that the majority of it will be automated unit testing through Unity. Ultimately, the engineers will decide on how often and how deeply they want to test the program. The team members will self-learn how to write tests and will assist each other if further training is necessary.

VI. Schedule

Testing will likely begin in early November as soon as the first code is demoable. Unit tests will be run as soon as possible to ensure that each component is working properly by itself. Throughout development, unit tests will likely be run at least once every week or on an as needed basis. When possible, larger scale tests will be created to ensure that components will properly work with each other. For example, a large scale test might involve a simple run through of a game where several things will be tested such as resource changes and how it affects the players. In other words, integration testing will be performed when unit tests are at a satisfactory level.

VII. Risks and Contingencies

A major risk of test planning is in spending too much time on testing and falling behind schedule. This is why automated unit tests are an attractive candidate to ensure quality while maintaining efficiency. In the worst case, extensive testing and/or manual testing may be forgone in lieu of having a working game by the deadline. Time to market is essential for success as long as there are no game breaking bugs.

VIII. Approvals

The team will be self-guiding so most approvals will be done by the engineer performing the tests. If necessary, team members may request a meeting if any uncertainties arise.

IX. Document Revision History:

Version	0.1
Name(s)	Peter Bui
Date	October 28, 2019
Change Description	File created