**Sinister Transistor**

**Test Plan**

**COP 4331, Spring 2016**

Team Name: <your team name here>

Team Members:

* Greg Kelso
* Mark Boutwell
* Joel Gardyasz

Modification history:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Who | Comment |
| v0.0 | 03/25/16 | Greg | Template |
| v1.1 | 03/25/16 | Joel | Introduction and Description of Test Environment |
| ... |  |  |  |

**Contents of this Document**

Introduction:  
    Overall Objective for Software Test Activity  
    Reference Documents

Description of Test Environment

Overall Stopping Criteria

Description of Individual Test Cases

**Introduction**

**Overall Objective for Software Test Activity**

The software testing effort should determine if all aspects of the game and code are running as expected. The game should be thoroughly tested in order to make an attempt to catch all bugs or things that could go wrong during a showcase of the game. Also, during the software testing, a short video of the game working should be made in order to use if the game does not work correctly during the final showcase. This will act as a fail-safe for if something goes wrong and can be used to show the game in a working state.

**Reference Documents**

* Concept of Operations
* Project Plan
* SRS
* <any other relevant documents>

**Description of Test Environment**

The hardware environment that will be used to test will primarily be the personal computers of the developers. The software environment that will be used to run the tests will be the Unity game engine in which the software will be made. Most testing will be done in the development stages of the design, while different aspects of the game are being developed. Hopefully, the game reaches a state to where it can be exported and sent to many people for secondary testing and debugging. In this case the test environment would be the computers of the people playing the game.

The test environment will be the same environment that the software operates in. The testers will most likely be the developers of the software, however, there may be opportunities to have other users test the system. Having actual users test the game is an excellent way to test for bugs or flaws in the game. Users who did not create the game have no prior cognition on how the game is supposed to operate, therefore, will play how they think they game should be played. This in itself can lead to the game being stressed in ways that the developers may not have thought of which is an excellent debugging tool.

**Stopping Criteria**

<How will you determine when to stop testing the software and either deliver it or send it "back" to development? Things to consider:

* If you find errors during testing: Will you stop testing each time you find a problem and immediately fix that problem? Will you continue testing and recording errors until you find a fatal error that won't allow you to continue? Will you test for 2 hours and then fix whatever errors have been found? Will you test for 2 hours and then hold a group meeting to decide whether to continue to test? ...
* If you find no errors during testing: How many test cases of what sort will you run before you declare the software to be "good enough to deliver"?
* How do you define "good enough to deliver"? Does it require that there are no known errors? Or no known errors other than cosmetic errors? Or no known errors other than cosmetic errors and errors for which there is a well-defined workaround? .... >

**Description of Individual Test Cases**

Describe EACH individual test to be run: (If you plan to run 20 test cases, you would answer the following questions for each of them. A table format or a bulleted format is acceptable.)

* Test Objective: <exactly what are does this specific test demonstrate?>
* Test Description: <exactly what will you test? What test data will be used (specifically -- what data values, what data files? This data must be determined in advance. So if you plan to use a test file, include that file in an appendix. If you plan to key in specific data, include the data here. Make sure the data you choose will allow you to achieve your stated objective for this test.>
* Test Conditions: <Under what conditions will you run this test? This is relevant for software for which there are multiple "modes". For some systems, the test conditions are totally described in the above test environment section -- in this case, the response to this would be "See Test Environment".
* Expected Results: <If the test executes correctly, what will be the result -- i.e., exactly what will the output look like; what will be the resulting data in the database, etc...>