Test Specification

For

Project R

|  |  |
| --- | --- |
| **Instructor: Professor Char** |  |
| **Team Members: Min Dye, Ryan Do, Peter Wainwright, Scott Hargrove** |  |
| **Cycle: 5** |  |
| **Date Submitted:** |  |

Document template copyright 2005-2015, CCI Instructors. Version 2.2. Use permitted under Creative Commons license CC-BY-NC-SA. See http://creativecommons.org/licenses/by-nc-sa/3.0/.

Grading Rubric – Test Specification

This rubric outlines the grading criteria for this document. Note that the criteria represent a plan for grading. Change is possible, especially given the dynamic nature of this course. Any change will be applied consistently for the entire class.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Achievement** | **Minimal** | **Exemplary** | **Pts** | **Score** |
| **Content** | Section(s) missing, not useful, inconsistent, or wrong. | Provides all relevant information correctly and with appropriate detail |  |  |
| Introduction |  |  | 5 |  |
| Test Specs |  |  |  |  |
| Selection | Aspects tested are trivial | Tests clearly address core system functions | 20 |  |
| Organization | Tests are disorganized, IDs or Objectives are not meaningful | Tests are well-organized with structured IDs and clear objectives | 20 |  |
| Set-up | Steps are unclear or incomplete | Complete, easy to follow conditions and steps | 20 |  |
| Results | Unclear or incomplete | Complete and clear | 20 |  |
| **Grammar and Spelling** | Many serious mistakes in grammar or spelling | Grammar, punctuation, and spelling all correct | 5 |  |
| **Expression** | Hard to follow or poor word choices | Clear and concise. A pleasure to read | 5 |  |
| **Tone** | Tone not appropriate for technical writing | Tone is consistently professional |  |  |
| **Organization** | Information difficult to locate | All information is easy to find and important points stand out | 5 |  |
| **Layout** | Layout is inconsistent, visually distracting, or hinders use | Layout is attractive, consistent, and helps guide the reader |  |  |
| **Late Submission** |  |  |  |  |
| **Total** |  |  | 100 |  |

Test Specifications[Instr3]

## 

**Debugging**

|  |  |
| --- | --- |
| **Objective** | Go through the game and find any bugs, then resolve them. |
| **Set-up** | Find out the bugs through either self testing or through play testers. |
| **Expected Results** | Plan on spending 3-4 hours going through entire game and finding any bugs that might hurt the game, and fixing them all. This will be the main focus of this week so much work will be done on it. |
| **Actual Results** |  |

[Instr1]Please use this report template as provided. Do not change template elements such as section numbers, headings, or stock text. Do not reorganize the report or delete sections.

You may add sections at the end or add sub-sections if needed. You should also adjust page breaks so major headings are placed appropriately in the final document.

Items in angle brackets, < >, are parts of the report that you must replace with appropriate information. All the angle brackets should be gone in your final submission.

All comments (like this one) should be DELETED before submitting the final version of the report. Comments in a draft version are fine.

The approach to test specification is loosely patterned after the Personal Software Process (Humphrey).

[Instr2]If you submit drafts or revisions, change this date for each submission.

[Instr3]Repeat the subsections and tables below as needed. Put subsections in order based on the Design Entity ID.

[Instr4]Identify the design entity to which the tests in this section relate. The ID and Name MUST match entries in your design specification

[Instr5]Create a unique ID for each test. It may be useful to adopt a naming scheme that incorporates the ID of the design entity being tested

[Instr6]Define a short name for the test. The name should fit the objective.

[Instr7]Enter the initials of the team member who wrote this test

[Instr8]Provide a short statement that defines what the test will prove. Use the terminology of the design entity being tested.

[Instr9]Define what conditions are needed for the test, e.g., state of the system, actions by a tester, contents of data files, values of data elements, input values, etc.

[Instr10]Define what should happen when the test executes. That is, define how the tester will recognize correct behavior by the system. Be specific about where the tester should look and what they should see.

[Instr11]This field provides a place to record the results of the test. It should be left blank when developing the specification