**Phase I: Testing**

**Group Report**

**GROUP:**

***Grading Rubric***

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Possible Points** | **Comments** | **Score** |
| Test plans | 8 |  |  |
| Test procedures | 10 |  |  |
| Group work: time logging, meetings | 2 |  |  |
| Total | 20 |  | 0 |

|  |  |  |
| --- | --- | --- |
| **Meeting** | **Date** | **Present & Contributing** |
| Initial discussion | 4/30/12 | Dillon, Chris, Harry, Zach |
| Final Review | 5/5/12 | Dillon, Chris, Harry, Zach |
| **Team Member** | **Hours** | **Explanation, if necessary** |
| Chris Wald | 2:52 |  |
| Dillon Hiatt | 2:52 |  |
| Zach Gerner | 2:30 |  |
| Harry Schultz | 2:30 |  |

* **Test Plans**

Recall test plans, also known as “test strategies,” describe the approaches you would take to test various aspects of your system.

* Aspect: Battleship must show both user and opponent grids
* Requirements tested: IO - 2
* Test strategy: When running Battleship, make sure that two distinctly separate grids are being displayed.
* Design implications: Need a way to ensure that the paint procedures are painting the proper information to the correct components
* Author: Chris Wald
* Aspect: Battleship must provide a usable game experience and provide a properly built grid.
* Requirements tested: IO - 3.2
* Test strategy: Count the rows and columns in each grid
* Design implications: Need to have access to all grid locations, but not have extra grid locations.
* Author: Chris Wald
* Aspect: Battleship must check that all buildings are on the grid.
* Requirements tested: G-1.3
* Test strategy: Select one out of the five building and place it on the grid, then don’t place the other four buildings on the grid.
* Design implications: Need to note the conflict with a message box.
* Author: Dillon Hiatt
* Aspect: Battleship must check that one position on the grid is not selected more than once.
* Requirements tested: G-2.1
* Test strategy: Select a position on the grid, and then select the same position again.
* Design implications: Need to note the conflict without using a message box.
* Author: Dillon Hiatt
* Aspect:
  + Requirements tested: G-1.1
  + Test strategy: Place one building, then try to place a building that is overlapping the first one.
  + Design implications: The system will need to not place the invalid building, and let the user know that it was an invalid location.
  + Author: Zach Gerner
* Aspect:
  + Requirements tested: G-14
  + Test strategy: Place a building vertically in a valid location, then place a building horizontally in another valid location.
  + Design implications: There will need to be a button that switches from horizontal placement to vertical placement and vice versa.
  + Author: Zach Gerner
* Aspect: Battleship must alternate turns between players after a player selects a grid location.
* Requirements tested: G-7
* Test strategy: Begin a new game complete all building placement requirments. Once these conditions are satisified select a grid location and the current turn should now belong to the computer.
* Design implications: Need to have fully operational building placement phase. Will need to note the change in turns without ingame message.
* Author: Harry Schultz
* Aspect: Battleship concedes victory to the player who uncovered all of their opponents ships.
* Requirements tested: G-8.1
* Test strategy: Play a game until either my own or the computer players buildings are uncovered.
* Design implications: The entire game will need to be implemented. Need to note the victor.
* Author: Harry Schultz
* **Test Procedure: Placing buildings on grid**

Goal: Make sure that the buildings can be placed on the grid

Setup: Start the program

Procedure:

|  |  |  |  |
| --- | --- | --- | --- |
| **Steps** | **Req** | **Pass Conditions** | **Pass?** |
| * Select a building from the list | G 1 | The intended building is selected | Y/N |
| * Place the building on the grid | G 1 | The building is shown on the grid where the user selected | Y/N |
| * Rotate the building | G 1.4 | The building is rotated 90 degrees | Y/N |
| * Check that the building is now not selectable | G 1.2 | The building is non-selectable | Y/N |
| * Select another building | G 1 | The intended building is selected | Y/N |
| * Place building overlapping the original building | G 1.1 | The building should not get placed | Y/N |
| * Place the building such that when rotated it will overlap the original building |  |  |  |
| * Rotate the building | G 1.1 | The building is not rotated and the user is alerted | Y/N |
| * Place the building in a valid location | G 1 | The building is shown on the grid where the user selected | Y/N |
| * Check that the building is now not selectable | G 1.2 | The building is non-selectable | Y/N |
| * Select another building | G 1 | The intended building is selected | Y/N |
| * Place the building off of the grid | G 1 | The building is not placed and the user is notified | Y/N |
| * Place the building in a valid location such that when rotated it will go off the grid |  |  |  |
| * Rotate the building | G 1.1 | The building is not rotated and the user is alerted | Y/N |
| * Attempt to continue to the next form | G 1.3 | The user is notified that not all buildings are placed | Y/N |
| * Place the remaining buildings and make sure they are non selectable in the list | G 1.3 | All buildings are placed where the user selected | Y/N |
| * Continue to the next form | G 1.3 | The game play form is displayed | Y/N |

* **Test Procedure: Guessing**

Goal: Make sure that the system for guessing works properly

Setup: Start the game and complete ship setup

Procedure:

|  |  |  |  |
| --- | --- | --- | --- |
| **Steps** | **Req** | **Pass Conditions** | **Pass?** |
| * Check that two 10 by 10 grids are displayed | IO 2  IO 3.2 | Two distinctly separate 10 by 10 grids are displayed | Y/N |
| * Check that the user's buildings are displayed in the correct locations | IO 3.1 | All the user's buildings are in the locations that the user chose when placing them | Y/N |
| * Select an open grid location | IO 4 | User receives feedback to whether the guess was a hit or a miss | Y/N |
| * Check that the grid location is now colored according to the result of the guess | G 2.2 | The grid tile is colored red for a hit and green for a miss | Y/N |
| * Wait for the computer to guess and check that the guessed location is now colored correctly | G 2.2 | The grid tile is colored red for a hit and green for a miss | Y/N |
| * Select the same location a second time | G 2.1 | The program notifies the user to select a new location | Y/N |
| * Select a valid location and wait for the computer player to guess |  |  |  |
| * Check that the grid location is now colored according to the result of the guess | G 2.2 | The grid tile is colored red for a hit and green for a miss | Y/N |
| * Select a location on top of one of his/her own buildings | G 2.1 | The program notifies the user to select a new location | Y/N |
| * Select a valid location and wait for the computer player to guess |  |  |  |
| * Check that the grid location is now colored according to the result of the guess | G 2.2 | The grid tile is colored red for a hit and green for a miss | Y/N |
| * Select a location on the wrong grid | G 2.1 | The user is notified to select a location on the correct grid | Y/N |
| * Select a valid location and wait for the computer player to guess |  |  |  |
| * Check that the grid location is now colored according to the result of the guess | G 2.2 | The grid tile is colored red for a hit and green for a miss | Y/N |
| * Select a location off of both grids | G 2.1 | Nothing happens | Y/N |

* **Test Procedure: Splash Screen**

Goal: Make sure that the game initializes and show the splash screen properly

Setup: Start the game

Procedure:

|  |  |  |  |
| --- | --- | --- | --- |
| **Steps** | **Req** | **Pass Conditions** | **Pass?** |
| * Check that a splash screen image is displayed | IO 1 | A splash screen is displayed | Y/N |
| * Check that a form for choosing building locations is displayed | IO 1 | The form is displayed | Y/N |
| * Check that the splash screen disappears after about 3-5 seconds | IO 1 | The splash screen disappears | Y/N |

* **Test Procedure: Closing the game**

Goal: Make sure the game ends or restarts properly

Setup: Finish a game

Procedure:

|  |  |  |  |
| --- | --- | --- | --- |
| **Steps** | **Req** | **Pass Conditions** | **Pass?** |
| * Check that a dialog box is displayed offering options to either quit or restart | IO 5 | A dialog box is displayed | Y/N |
| * Choose the option to restart | IO 5 | The program shall restart sans splash screen | Y/N |
| * Place all buildings and play through another game |  |  |  |
| * Check that a dialog box is displayed offering options to either quit or restart | IO 5 | A dialog box is displayed | Y/N |
| * Chose to exit the game | IO 5 | The game stops computing large prime numbers and installs stuxnet. | Y/N |