CS 135 @ UNR Fall 2014 11/25/2014

Project Title: !Battleship…Kind Of

Group Name: The Old-Timers

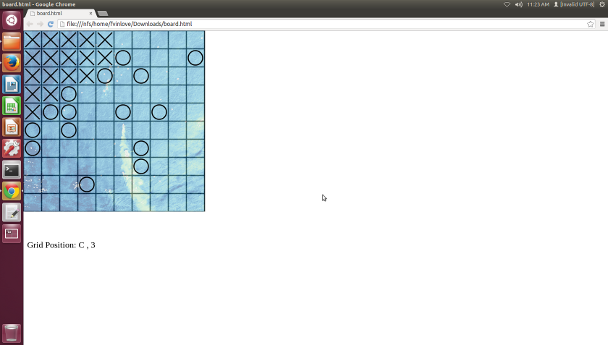
* Kurt Andersen
  + (775)225-3252
  + [kurt.t.andersen@gmail.com](mailto:kurt.t.andersen@gmail.com)
* Frances Vinlove
  + (775)384-7281
  + [f.vinlove@yahoo.com](mailto:f.vinlove@yahoo.com)

* Ryan Battistone
  + (775)303-9674
  + [battistone@nevada.unr.edu](mailto:battistone@nevada.unr.edu)

Project Description:

We are proposing to build a 10x10 single-player Battleship game. The boats that the player will attempt to find will be randomly generated and placed on the grid, without boats overlapping or going through the edges of the canvas. The player will click where they want to shoot, and if it is a miss, a splash sound will play, and a circle will show up on the grid indicating a miss. If it is a hit, there will be an exploding sound, an X will be placed on that spot indicating a hit. Player has X amount of turns to find all five of the enemy boats, which are five, four, three, three and two squares in length. If the player finds all the boats before his/her amount of turns is up, a winning message will be displayed, if turn counter gets to zero, a losing message will display.

Features:



Project Specification:

1. This is a one player game and the player clicks where they want on the game board. The board is a 10x10 grid with coordinates, so the player can keep track of where they are clicking. An X will appear when part of a boat is hit, and an O will appear when an open space is hit.
2. Random Boat Generator. This will generate the boats in random positions each time the game is played. There are five different boats, of size 2, 3, 3, 4, and 5.
3. Hit/Miss Check Function. This function will check the position the player clicks on the board to see if there is already something there, or if the player previously choose that spot. A pop-up message will appear telling the player that they can’t play on a previously played spot.
4. Scoreboard Function. This function will store the players name and their score every time they play, and then output the results at the end of the game (see Top Ten List Of Scores for more detail).
5. Turn Calculator. This will display the amount of turns the player has left, decrementing each time the player clicks on the game board.
6. Images & Sounds. These will be items such as the bang and splash for hits and misses, and sprites for the boats (once an entire boat is discovered, the X’s will turn into an image of a boat).
7. Top Ten List Of Scores. We will track the top ten scores using the Scoreboard Function previously written. This means the player will choose their desired difficulty at the beginning of the game, and also input their name. This has three subparts.
   1. The file needs to sorted in order of highest score to lowest score.
   2. If a player plays pong more than once, we will need to update their score, but only if it improves. This means, we have to search the file for the player handle, if the player name exists, then we need to check the stored score against the current score and replace the stored score with the current score, if the current score is greater than the stored score.
   3. We will display the top ten list at the end of every game.
8. Game End. The game ends when a player finds all the ships, or runs out of turns. We will have the turn counter displayed so they can keep track of their turns.
9. Documentation. Ensure that the documentation is well written, and that instructions for game play and use are well written and easy to read.
10. Testing. We will be testing the program for bugs and other issues extensively during play testing and tuning the various parameters that affect game play. In addition, we will be making sure the details (such as sounds and images) are all working correctly.

Schedule and percentage effort are on the next page. The gray part of the table indicates the features to be demonstrated at the project prototype demonstration in lab.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feature** | **Implementor** | **%** | **Due** | **Grade** |
| 1. 1. Game Board | Kurt | 7% | 11/25/14 |  |
| 2. Boat Generating Function (Static) | Kurt | 10% | 12/2/14 |  |
| 1. Hit/Miss Check Function (Static) | Frances | 10% | 12/2/14 |  |
| 1. Scoreboard Function (Array) | Ryan | 10% | 12/2/14 |  |
| 1. Boat Generating Function (Random) | Kurt | 10% | 12/6/14 |  |
| 1. Hit/Miss Check Function (Random) | Frances | 10% | 12/6/14 |  |
| 1. 7. Score Board (Output) | Ryan | 10% | 12/7/14 |  |
| 1. Turn Calculator | Kurt | 8% | 12/8/14 |  |
| 9. Images & Sounds | Ryan | 10% | 12/8/14 |  |
| 1. Documentation | Frances | 10% | 12/8/14 |  |
| 1. Testing | Ryan & Frances | 5% | 12/8/14 |  |
| Project release |  | 100% | 12/9/14 |  |

# Sample Input and Output

The screenshot above provides an idea of what the game board should look like. As the user clicks on the desired coordinates on the grid, X and O will appear, indicating a hit or a miss. Once a ship is entirely discovered, the X’s will convert to an image of a boat.

# Running the application

Players would go to the specific website and click OK on a pop-up window to start the game. The game would then start, players then choose their difficulty and enter their name, then proceed to the game would play by choosing their coordinates.