The “NavyDaddys”

**EC327: Project Timeline for “Battleship”**

|  |  |  |
| --- | --- | --- |
| **Time** | **Plan** | **Code Implementation** |
| 11/15 | * “NavyDaddys” formed! * Formulation of ideas, vote on a final idea to commence the project * Initial plan was to create a game based on the board game “Battleship” using the software “GameMaker” | * GitHub repo created |
| 11/16 -> 11/28 | * Independent research for coding similar projects and coding references * Changed plan to making the game on the Terminal using Bash Scripts to output graphics |  |
| 12/5 | * Coding commenced! * Goal: Create a foundation for the product by creating a simplified version of the final game | * main.cpp * test.cpp   Features implemented:   1. The game involves the A.I randomly choose the location of 10 different ships (all with size 1) and the player guesses the location of these ships to attack them. The game ends when either the player quits, or all the ships are sunk. 2. Added two ship types Cruisers (size 3) and Frigates (size 1) 3. Added functions to display plots for players |
| 12/6 | * Starts implementing object-oriented programming | * console.cpp/.h * coordinate.cpp/.h * cruiser.cpp/.h * frigate.cpp/.h * ship.cpp/.h   Features implemented:   1. Different types of ships will have their own object class to store their relevant member variables and functions 2. Functions to displayed plots to players is now implemented in console.cpp/h 3. Coordinates also has its own class type with its relevant member variables and functions |
| 12/8 | * Improves visual features of the game | * menu.cpp   Features implemented:   1. Game now starts with a menu displaying the game information, along with options to either start playing, view credits or exit the game |
| 12/9 | * Works on player input when starting the game | * main.cpp * console.cpp * ship.cpp/.h   Features implemented:   1. Continues working on these files 2. Game now starts up asking player to input coordinate and verify the validity of these inputs. 3. Game will not display all information at once, instead waiting for input from players to process |
| 12/10 | * Refined existing files and bugs fixing | * menu.cpp * coordinate.cpp/.h * makefile * main.cpp * functions.cpp/.h   Features implemented:   1. Fixed bugs that make the game quits after asking difficulty level 2. Update bugs in main.cpp and functions.cpp/.h that gives an extra dimension to ship types 3. Coordinates.cpp/.h and its relevant functions in functions.cpp/h works fully with main.cpp 4. Functions are now moved to separate cpp/.h, for clarification and easier documentation 5. Plot now displays marked location for ship hit or missed |
| 12/11 | * Refining existing features and finalizing the game * Game will now feature 3 complete difficulties: Easy, Medium, and Hard difficulty * Removes features where player can choose coordinates, now the AI makes the move, and the players choose target to attack. | * main.cpp * battleshipgame.cpp/.h   Features implemented:   * Plot now displays the type of ships that is being hit at the chosen location by player * Player will be represented with the board game (plot) after every move, improves player experience * Complete game is available: Game now involves player choosing location to attack enemy ships placed by A.I. Constraints on player involves a limited number of missiles available and the size of the board. The harder the difficulty, the less the available number of missiles and the larger the size of the board. * Bugs fixed where game exits abruptly when asks for retry * Separate each instance of board to improve visual presentation for players |