Battle of the Ships

Dax Valdez

104759064

3/21/2019

Description This is a copy of the board game Battleship. In this game, both the player and the opponent will have their ships on a grid. It is up to the player to guess where the ships of the opponent are located. The person to correctly guess and destroy the other players ships first will win.

1. Input RequirementsThe project will use a csv file in order to take in the inputs in order to know the locations of the battleships. The first value of the file will be the ship name, followed by the location of the ship on the grid, followed by whether the ships will be placed horizontally or vertically. In order to place the ships into a grid, a grid class must be created with either a 2d array or 2d vector of size 10. The program also requires the player to use the keyboard to select the location of the battle ship by entering the integer followed by a character value to guess the location of the opponents ship when prompted.
2. Output RequirementsThe screen will show both your grid and the opponent’s grid of possible locations for the ships. If there is miss for the given location, a m will appear. If there is a hit there will appear an x in the grid. Otherwise, the grid will appear empty for the enemy’s grid. The player’s ship’s locations will appear with an s on it.
3. Problem Solution Discussion  
   Reading from a file: The reading from a file function will store the names, locations and type of placement on the grid. It will then store them into 3 sets of vectors.

Set players field: This function takes information gathered from a set of vectors to place the ships locations on the grid. It checks to see if the ships are about to placed out of the grid and inverts them if required.

1. Classes, Inheritance, and Data Structures This program utilitlizes two different classes. One is a player class that inherits a grid class while storing values for a ship’s location, placement, and if it has been sunk or not. The grid class stores a 2d vector and some member functions to configure it.

1. Overall Software ArchitectureDisplay grid: Displays the grid and its contents for each specific index. It also displays the specific horizontal and vertical index for ease of use.

Convert from character to integer: Takes in a specific character A to J and converts it to an integer from 0 to 9 respectively.

Convert from integer to character: Takes in a specific integer from 0 to 9 and returns an integer from 0 to 9 respectively.

Convert from string to integer: Takes in a string value from 0 to 9 and returns an integer value of itself.

Check for intersection: This function checks to make sure that a ship does not intersect with another ship. If an intersection is found, the ship is moved first once to the left, then once to the right, then once up, then once down until a suitable match is found.

Check for boat sunk: Checks where the boat should be and returns weather or not if all its locations have been hit or not.