CSCI 2312 - Design Document

Battle Ship

Michael Zabawa

107522040

4/4/2018

1. Problem Description
   A program that models the classic game of battle ship where players place ship pieces on a grid. Then the players take turns guessing at the location of the other players ship.

2. Overall Software Architecture

   Class Board
   responsible for the game board, functions include:
   - checkMark returns true if the provided mark is at the provided indices
   - placeMark replaces the mark at the provided indices
   - getMark returns the mark at the provided indices

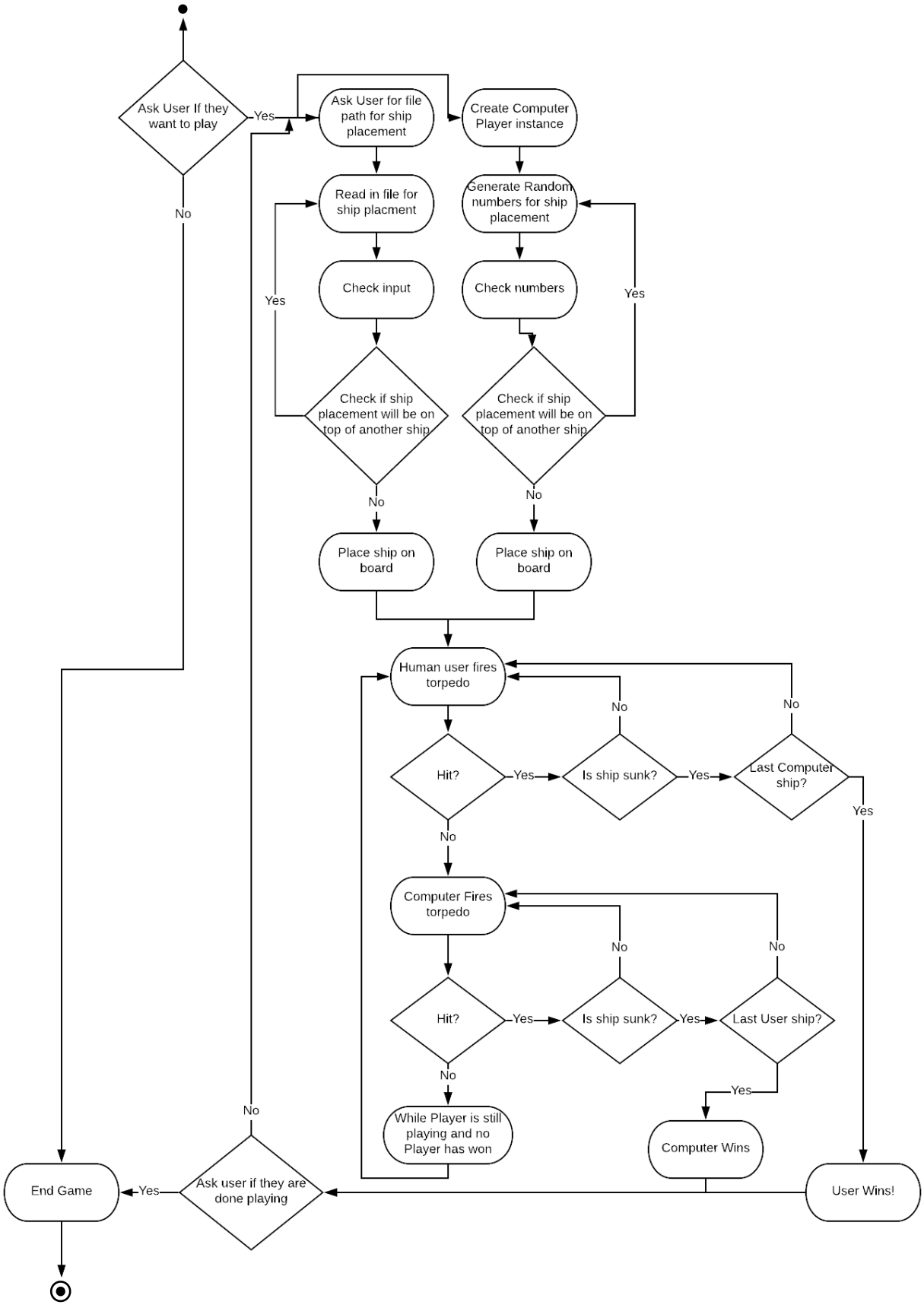
   Class WaterVehicle
   responsible for the ships, functions include:
   - getShipPlacement reads input and mutates the starting location and orientation of ship
   - gotHit increments the hit counter

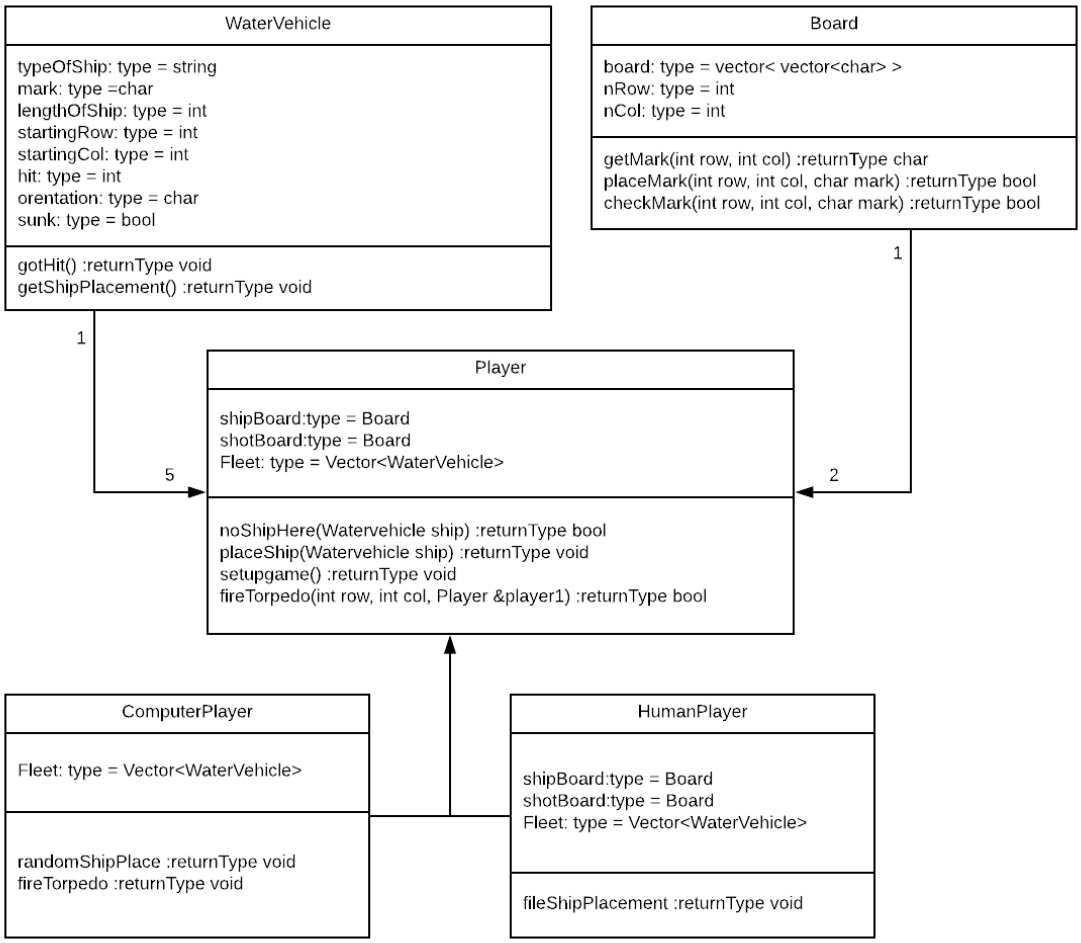
   Class Player
   responsible for player functions:
   - setUpBoard calls place ship and marks the board accordingly
   - shootTorpedo takes two ints and checks if shot was a hit

   Class ComputerPlayer
   - fileSetUpBoard reads in a file and mutates the ships accordingly

   Class HumanPlayer
   - randomSetUpBoard random set of the board
   - randomTorpedo random shot of the torpedo

   In the product description we have two players one is a computer and the other is a human user. Because there is two types of players with different functions I plan on creating the base class of Player and then the computerPlayer and the humanPlayer will inherit from the Player class.


Ask User If they want to play
Yes
No
Ask User for file path for ship placement
Create Computer Player instance
Read in file for ship placment
Generate Random numbers for ship placement
Check input
Check numbers
Yes
Yes
Check if ship placement will be on top of another ship
Check if ship placement will be on top of another ship
No
No
Place ship on board
Place ship on board
Human user fires torpedo
No
No
Hit?
Yes
Is ship sunk?
Yes
Last Computer ship?
Yes
No
Computer Fires torpedo
No
No
Hit?
Yes
Is ship sunk?
Yes
Last User ship?
No
Yes
While Player is still playing and no Player has won
Computer Wins
No
End Game
Yes
Ask user if they are done playing
User Wins!



3. Input Requirements

The description of the project requires that the user ships be placed by the use of an input file. The format of this file should read as follows: char, int, char. the first char is the column index within the range of 'A' - 'J' and the int is the row index within the range of 1 - 10 and the final char is for the orientation so it will have to be either an 'h' or 'v'. After the file is read in the case of the input will be changed to allow for the program to be case insensitive.

After the ships have been placed the user will have to fire a torpedo. Again the order will be char then int, the char for the col index with the same range as the ship placement and the int for the row index with the same range as the ship placement. The first input will be a char so on each attempt to fire the user could input 'q' or 'Q' to end the game.

4. Output Requirements

All of the output will be to the console for this program. There will be a message that the file was read correctly and display the ship board with the locations of the placed ships. The game will then commence with the user firing a torpedo at the computer player. A message will be displayed to indicate the result of the shot. The computer player will then randomly generate a shot and the ship board with the result will be displayed.

5. Problem Solution Discussion

Ask user if they want to play and then ask for the file path to read in the input. After the file is read in then the game will start. A user will input a shot and the shootTorpedo function will check if the shot was a hit. If it was a hit the gotHit will increment the hit counter and check if the hit counter is equal to the size of the ship. If it is then will report that the ship was sunk. If the shot was already called the function will report that and likewise with a miss. Then the computer will take a random shot. If the the shot was a hit it will store the hit and the preceding shots will be around then hit until ship is sunk then will return to random firing.

6. Classes, Inheritance, and Data Structures

I am choosing to model the game as each part that a player has. A player has a game board and ship pieces. I plan to keep the responsibilities separate so the responsibility of the board functions will only modify the board object as will the WaterVehicle class functions will only modify the WaterVehicle attributes. The player class will make the decisions with regards to whither the ships on top of each other and will return if there is a miss or hit on a ship and use public member functions in the WaterVehicle class to set those conditions.

I thought about having a more boundless scope with regards to allowing the WaterVehicle class to have the ability to modify the board but that structure appeared to be more disjointed and did not reflect the objects that is being modeled. The only class that modifies the board or ships in the “real” world is the player.

For the inheritance requirement I had initially and wrongly thought about having the player be derived from the board and WaterVehicle class. I now feel that the relation between the player and game objects are better represented with the planned structure.