Project 2: Enhance another team's system from Project 1

Due Date: 11:59 PM, Sunday, September 29, 2024

- All teams must have a code freeze as of 11:59 PM, Sunday, September 29, 2024.
- The timestamp will be judged by the final commit on your master branch of the team's Github repository.
- You can continue to work on other branches but cannot update your master branch after the freeze date.
- You must demo your code during your GTA/Team meeting based on that master branch as of the code freeze.
- All artifacts (code and documentation) must be in your repository on your master branch.
- Your Team Peer Reviews for each team member are also due at this time in Canvas.

Overview

- You're all fired!
- You are no longer working on your projects.
- Luckily, there are a lot of job opening at other companies.
- In fact, the market is booming right now and finding a new job for your team should be easy.
- You will fork the project of another team's Project 1 and complete the requirements for Project 2.
- The table showing which team gets which team's Project 1 will be posted on Canvas by 11:59 PM, Sunday, September 15.
- As the former Project 1 owners, you link the new team your GitHub repo and any documentation.
- **Do not to provide support beyond giving them your work.**
- The new team should make changes to the repository on a separate branch (not master).
- **You cannot contact the team you inherited from for help of any kind.**
- **You must program in the same language(s) and support the same platform as the original team.**
- Refactoring and bug fixing are allowed.
    - Refactoring is restructuring an existing body of code, altering its internal structure without changing its external behavior.

Requirements

1. You must get all required functionality from Project 1 working, even if the team you inherited it from did not.
2. In addition to playing against a human, you will create an Artificial Intelligence (AI) opponent to play against. Requirements for the AI are:
    a. Three difficulty levels:
        i. Easy: It just fires randomly every turn.

ii. Medium: It fires randomly until it hits a ship then fires in orthogonally adjacent spaces to find other hits until a ship is sunk.
iii. Hard: Cheater, cheater pumpkin eater! This mode knows where all your ships are and lands a hit every turn.

b. Setup: all AI difficulty levels place their ships randomly (must still be legal placement).

3. Custom Addition:
    a. Your team must decide upon a new addition to the game.
    b. Your team will describe the Custom Addition using one of the UML diagrams you learned about in EECS 348. You may choose to use any of the UML diagrams.
    c. Approval from your GTA of the UML diagram is required before moving forward.
    d. The ceiling on scope and difficulty is up to your team, but your GTA has the right to increase the difficulty if they feel it is necessary.
    e. Here are some ideas:
        i. Special shot (e.g. a limited number of 3x3 giant shots)
        ii. Animations
        iii. Sound effects
        iv. Scoreboards
        v. Ability to move ships after setup

Language and Platform

- You must program in the same language(s) and support the same platform as the original team.

Grading Rubric

- Team Score (75 points – team based)
    - This portion of the project will be graded by your GTA.
    - The project points are broken down into the following sections.
    - Demo (40 points)
        - You will demo on a device of your choice in your weekly GTA/Team meeting.
        - All of the Project 1 features, the AI feature, and the Custom Addition approved by your GTA must work to the GTA's satisfaction.
    - System Documentation (25 points)
        - On your GitHub repository, have a folder called "documentation" that contains all the system documentation described below.
        - Estimate of person-hours for completing the project (this should be done as soon as possible – 10 points)
            - Provide the details of how you arrived at the estimate
        - Actual accounting of the person-hours required to complete the project (10 points)
            - This needs to be a day-by-day accounting from each team member on how many hours they spent on the project, including team and GTA meetings, coding, testing, documenting, etc. Do not include time attending 581 lectures, working on in-class problems.

        - You WILL NOT be penalized for taking longer (or shorter) than your estimate.
        - You WILL be penalized, if you do not have a day-to-day accounting or it looks like to the GTA that you created the accounting just to meet this requirement.
    - Documentation of your code (5 points)
        - This can be in any format you choose, but it should be detailed enough that the GTA can understand what you did.
        - This documentation should supplement the comments in the code.
- Comments (10 points)
    - The software must be adequately commented with prologue comments, comments summarizing major blocks of code, and comments on every line.
    - Adequate prologue comments include:
        - Name of program contained in the file
        - Brief description of the program
        - Inputs
        - Output
        - Other sources for the code ChatGPT, stackOverflow, etc.
        - Author's full name
        - Creation date: The date you first create the file, i.e., the date you write this comment
    - Adequate comments summarizing major blocks of code and comments on every line:
        - Provide comments that explain what each line of code is doing.
        - You may comment each line of code and/or provide a multi-line comment that explains what a group of lines does.
        - Multi-line comments should be detailed enough that it is clear what each line of code is doing.
        - Each block of code must indicate whether you authored the code or you obtained it from one of the sources listed in the prologue, or if it was a combination of both.
    - Other sources for code:
        - When you use other sources for the code (e.g., ChatGPT, stackOverflow):
        - Your comments must be significantly different from the source's comments.
        - More scrutiny will be applied to grading your comments in particular explaining the code "in your own words", not the source's comments (e.g., ChatGPT's comments).
        - Failure to identify other sources of code will not only result in a 0 on the assignment but will be considered an act of Academic Misconduct.
        - Students who violate conduct policies will be subject to severe penalties, up through and including dismissal from the School of Engineering.
    - The comments should be detailed enough that the GTA can understand what you did.
- Team Peer Evaluations (25 points – individual based)
    - This part will be based on the Team Peer Evaluation forms as described in the first lecture.

- You may receive more than 25 points if your teammates think you did more than your fair share of the work.
- If you turn the Team Peer Evaluation form in late or do not turn it in:
  - You will still get a Team Peer Evaluation score based on your teammate's evaluation.
  - You will lose 25 points, which could result in a negative score.
  - The $10,000 bonus you were supposed to divide between your teammates will be divided equally among your teammates.