**Project 2: Enhance another team’s system from Project 1**

**Due Date: 11:59 PM, Sunday, March 14, 2021**

* All teams must have a code freeze as of 11:59 PM, Sunday, March 14, 2021.
* 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 the week of March 15 in your lab 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 CATME Peer Reviews for each team member are also due at this time.

**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 is posted on BlackBoard.
* The emails of each team member are also posted.
* 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 is allowed.
  + Refactoring is restructuring an existing body of code, altering its internal structure without changing its external behavior.
* Adding a GUI to a terminal based implementation is fine.
* Other drastic changes need approval by your Lab GTAs.

**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:
   1. Three difficulty levels:
      1. Easy: It just fires randomly every turn.
      2. Medium: It fires randomly until it hits a ship then fires in orthogonally adjacent spaces to find other hits until a ship is sunk.
      3. Hard: Cheater, cheater pumpkin eater! This mode knows where all your ships are and lands a hit every turn.
   2. Setup: all AI difficulty levels place their ships randomly (must still be legal placement).
3. Custom Addition:
   1. Your team must decide upon a new addition to the game.
   2. Approval from your GTA is required before moving forward.
   3. The ceiling on scope and difficulty is up to your team, but your GTA has the right to increase the difficulty if he or she feels it is necessary.
   4. Here are some ideas:
      1. Special shot (e.g. a limited number of 3x3 giant shots)
      2. Animations
      3. Sound effects
      4. Scoreboards
      5. 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.

**Plagiarism**

* You can use existing libraries, but you must cite all sources of code you did not author in your documentation.
* This assignment, or variations of it, has been done in other EECS 448 classes.
* **You may not use code from those classes (even if you cite it)!**
* Like any good programmer would do, you may certainly look at these projects for ideas, but you may not use any of the code.

**Grading Rubric**

* Team Score (80% of grade – 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%)
  + You will demo on a device of your choice in the lab during the week of March 15.
  + 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 (40%)
  + GTA approved Requirement Engineering artifact describing the features of your Custom Addition to Project 1 (10%)
  + Estimate of person-hours for completing Project 2 (this should be done as soon as possible after your GTA signs off on your Custom Addition – 10%)
    - Provide the details of how you arrived at the estimate (e.g., LOC, using the Agile user story method where Project 1 represents your inventory of projects)
  + Actual accounting of the person-hours required to complete Project 2 (10%)
    - 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 448 lectures or 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.
  + Identify the design paradigm you think the Project 1 team used and explain why you think that (250-350 words – 10%).
  + On your github repository, have a folder called "documentation" that contains all the system documentation described above.
* Team Evaluations (20% of grade – individual based)
* This part will be determined by CATME Peer Review.