**CMSCH105 PROJECT PROPOSAL**

**Team Members:** Tazkia Afra, Nathan Wang, Caspian Macky

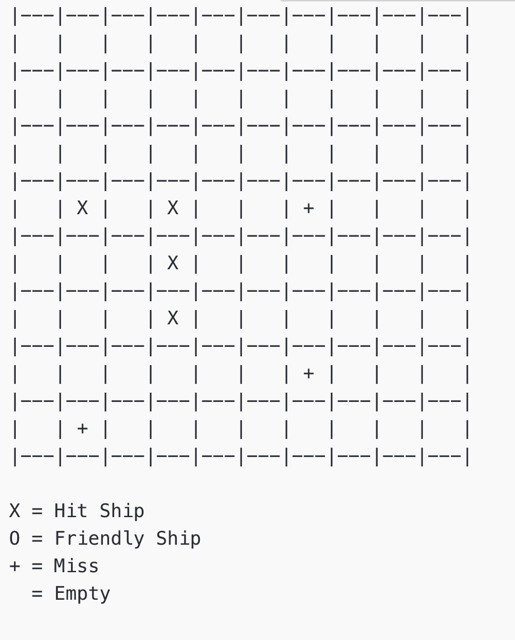
**Project Title:** Battleship

**Description of Project:**

We plan to produce a Python program that, when run, allows the user to play the board game Battleship against an AI. The user and AI will be able to place each of their five ships of different lengths on their (separate) boards. The first person to go will be chosen randomly. Each turn, the active player (user or AI) will fire a shot at one set of grid coordinates, with the goal of hitting every grid square containing an enemy ship before all of their own ships are sunk.

The game will need to detect and report when an individual ship has been sunk, as well as when the game is over. Furthermore, the AI will need to be able to tell when it has hit a ship in order to concentrate fire on that area until it has sunk the ship.

The program will output the two boards graphically. The following is a mock-up of a potential GUI:



**Suitability:**

We plan to implement the game using matrices to track the locations of different items. The game will need to reference and manipulate these matrices in various ways to accomplish its tasks, including displaying the content of the matrices in a legible GUI. Furthermore, multiple data structures will need to be constructed to track the position and status of the various ships used in the game. Finally, the game will require randomization to be combined with rigorous checking of the suitability of the random outcomes in order to function properly.

**Task Breakdown:**

* Create matrices: 1h, Nathan
  + Construct GUI
* Placing Ships: 4h, Caspian
  + Prevent overlapping
* Placing Enemy Ships: 3h, Caspian
  + Randomization of above process
* Shooting: 1h, Nathan
  + Disallow repeats
  + Check hit
  + Report hit
* Enemy shooting: 4h, Nathan
  + Random until hit
  + Recognition of hits
  + Strategic shooting based on past outcomes
* Check Sink and Win: 2h, Tazkia
  + Report sink
  + Check win
* User Interface: 4h, Tazkia, Nathan
  + Display
  + Start Game (including starting player)
  + Inputs
  + Outputs
* Playtesting: 2h, All
* Debugging: 6h, All

**Risks:**

We anticipate that preventing ship overlapping and coding the computer’s shooting strategy AI will be the most difficult parts of the project. Other problems may arise in the output GUI, specifically with relation to being able to display the full board on-screen.