Project 3: Build a Prototype

Requirements

- *Your team must all agree on the application, language, and platform.*
- *Use one of the Requirements Engineering artifacts to document the requirements of your application and identify which requirements you intend to include in the prototype.*
- *Approval from your GTA is required before moving forward.*
- *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.*

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*

Application Ideas

- Jeopardy
- Weather App
- Choose your own adventure
- Chess

Language Ideas

- C++
- JavaScript
- Python

Platform Ideas

- Tkinter via python
- Web based application
- Pygame via python.
- Terminal Based

Put Together Ideas

- Create the game **Jeopardy**, using a **C++ terminal based application**. The game would be played with two players. Categories could be file names and the questions (and answers obviously) for each category could be stored in the files. When the game begins 5 random files could be chosen randomly and 5 random questions could be chosen from the file. Before the game starts the user would have an option of adding a category/question thus creating/adding to a file.
    - Difficulty - **Easy**
    - Would be a decent amount of file management and a good amount of code but since we are all familiar with c++ it shouldn't be too bad.

- Create a **weather app** using **python/tkinter**. We would create a good looking GUI that would get real time weather updates for Lawrence, KS using a weather API. We could give the user the option to add a location that they want the weather from and update the GUI accordingly.
    - Difficulty - **Medium**
    - It would take a little bit of time to understand how tkinter and weather APIs work but as soon as we figured this out it would be easy. This might not be a big enough project though.

- Create a **terminal based choose your own adventure** story using **C++.** The options are endless for this. This would require some story structure before we start coding and A LOT of conditional statements that might be hard to keep track of.
    - Difficulty - **Medium**
    - This would take a lot of organization before we start coding and some game/software development ideas that might be difficult to grasp.
    - This idea might not fit the scope for this project as well.

- A **terminal based Chess** implementation using **C++**. This will be similar to the battleship project where we create a board and have different piece classes. The board will be reprinted every time and the game will loop through player one and player two. We could also add an AI for this but it would be more complex than the Battleship AI.
    - Difficulty - **Medium**
    - This probably will be a little more difficult as move validation might be a little harder.
- **Time management app,** add classes, calendars, plan the week/month.Could be **python GUI / c++ terminal based**.
    - Difficulty - **Easy**

    - Depending on the platform we use this still should not be too hard.
- **Fantasy games**, similar to fantasy football, big projects, using **Python** and APIs it could be implemented using real information.
    - Difficulty - **Medium**
    - It would be difficult at first, but once the APIs are understood / taken care of then it shouldn't be too bad.
- Arcade games, **Pac-Man** / Galaga using **pygame** / **javascript**.
    - Difficulty - **Medium**
    - Figuring out how to use pygame would be the hardest part of this project. The logic of the game is pretty simple once we figure it out.