

## Wizards Unite

### 1. Overview

In this project, you will:

- Practice basic C++ syntax including branching structures
- Write classes and instantiate those classes using a constructor
- Use arrays to hold objects
- Use simple file input
- Practice breaking projects into multiple files
- Use a makefile to compile a project

### 2. Background

Harry Potter: Wizards Unite is an augmented reality game released to iOS and Android in June 2019. Like Pokémon Go, Wizards Unite follows an Auror, Professor, or Magizoologist wandering around fighting against mythical beasts from the Harry Potter and Fantastic Beasts franchises.

For this project, you will be required to create a text-based version of this game. You may see a videos and various details of the game online here: <https://www.harrypotterwizardsunite.com/>

In this implementation of Wizards Unite, you will be implementing the role of the Wizard. For our purposes, the Wizard will be gathering Foundables which are a variety of people and artifacts.

### 3. Assignment Description

Initially, you will need to read in a list of Foundables from a file and load them into an array. The list of Foundables is static, and you can assume that the size can be stored in a constant. The Wizard starts by having collected no Foundables although the goal is to collect all the Foundables in the game. The goal of the game is to collect all unique Foundables available across 5 different rarities. The more rare the Foundable, the higher the toughness!

### 4. Requirements:

This is a list of the requirements of this application. For this project, you will be provided with header files to start you in the right direction. For you to earn all the points, however, you will need to meet all the defined requirements.

- The project must be completed in C++. You may not use any libraries or data structures that we have not learned in class. Libraries we have learned include `<iostream>`, `<fstream>`, `<iomanip>`, ~~`<vector>`~~, `<cmath>`, `<ctime>`, `<cstdlib>`, and `<string>`. You may use [stod](#) and [stoi](#) in this project to convert strings to doubles or strings to integers. You may not use vectors – everything must be implemented in arrays. You should only use `namespace std`.

- You **must** use the provided header files (**Foundable.h**, **Wizard.h** and **Game.h**). You may only add HELPER functions and global constants to these files. Do not add member variables to any class.
- All user input must be validated. For example, if a menu allows for 1, 2, or 3 to be entered and the user enters a 4, it will re-prompt the user. However, the user is expected to always enter the correct data type. i.e. If the user is asked to enter an integer, they will. If they are asked to enter a character, they will. You do not need to worry about checking for correct data types.

There is a single input files for this project named, “**proj2.txt**”. The file name can be stored as a constant.

- The player’s name can have a space. Additionally, the names of the Foundables may have a space.
- A Wizard starts with a toughness somewhere between 150 and 300. There are constants in Wizard.h to help **MIN\_START\_TOUGH** and **MAX\_START\_TOUGH**.
- A Foundable is defeated when the Wizard has a higher toughness. Ties go to the Foundable. The game continues if the Wizard is defeated but that Foundable will still need defeating in order to win the game.
- A Wizard goes up in level when they successfully defeat another Foundable.
- The game is won when all Foundables have been defeated.
- When a Wizard goes up in level, their toughness increases from 0 – 99. There is a constant in Wizard.h for this called **LEVEL\_TOUGH\_INCREASE**
- Have a main menu that asks if the user wants to:
  - Display information about the Wizard including name, level, toughness, wins, losses, win rate (2 decimal place percentage) and number of Foundables found.
  - Display defeated Foundables in a numbered list.
  - Display all Foundables in the game in a numbered list.
  - Attack Foundables
    - Asks which rarity of the Foundable you would like to attack. Each rarity generally has a higher toughness.
    - When attacking, the Foundable is randomly chosen from the list of all Foundables. There are two rules though: Each Foundable can only be defeated once and the Foundable fought will always be the rarity chosen (unless all Foundables of that rarity have been defeated).
    - If the Wizard wins, the Wizard win count is incremented. If the Wizard loses, the Wizard loss count is incremented. The win rate is displayed as a percentage (with exactly 2 decimal places).
    - Ties are considered a loss for the Wizard.
    - If all Foundables *of a specific rarity* have been defeated then the game tells the user that has happened.
  - Exit