PHY 410 / 505 Midterm Exam
Fall 2022

Instructions:
PHY 410: Do problems 1-3. You can do Problem 4 for extra credit.

PHY 505: Do all 4 problems.

For undergrads: The extra credit problem will count as an optional homework grade and will replace your lowest homework score for the semester.

Assignment file on github: https://classroom.github.com/a/W8JUmpX8

For setup instructions, be sure to follow instructions here:
https://github.com/ubsuny/WorldOfTextCraft

Then your git commands (on your host OS) will be like:

```
git clone git@github.com:ubsuny/midterm2022-rappoccio.git
cd midterm2022-rappoccio
git add *.cpp *.h *.ipynb *.txt
git commit -m"I hope I passed"
git push origin main
```

Do not work with other students. Submit your solutions in a PDF or HTML file, and your code to github classroom. **I will not accept any submissions without a writeup.**

Problem 1 (25 points): The Three Stooges.

a. **(10 points)** Run the Three Stooges scenario:

```
./WorldOfTextCraft ThreeStooges.txt Shemp.txt ShempAttacks.txt
problem1a.txt
```

The text file for the battle log should be called "`problem1a.txt`". Be sure to post it. Did you win?

b. **(15 points)** You will now update the "Rogue.h" file. This class currently overloads the "attack" method, and calls a function called "defaultAttack", which is defined in the base class in "Entity.h" and "Entity.cc". Instead of calling the "defaultAttack" method, write a different "attack" method to deal 20 additional damage (on top of the base attack power) every third turn (i.e. don't do it on turn 0). Ensure that the "myAttacks_" action vector is up to date!

Run like:

```
./WorldOfTextCraft ThreeStooges.txt Shemp.txt ShempAttacks.txt
problem1b.txt
```

Hint: The "getTurn()" method in the "Entity" base class can be used to determine which turn the Rogue is on.

Be sure to upload the battle log output in the file called "`problem1b.txt`". Did you win?

Problem 2 (25 points) : The Lich King.
Run the Lich King scenario, including the "Rogue.h" update from problem 1 (if you did not get the correct answer in problem 1, you will not have points deducted for that in this problem).

```
./WorldOfTextCraft AshenVerdict.txt Arthas.txt ArthasAttacks.txt problem2.txt
```

Be sure to post the "`problem2.txt`" file.

The problem will be done in the "AnalyzeBattle" python notebook. Export the output as "Problem2.html" from jupyter and be sure to upload it to UBLearns.

a. **(5 points)** Make plots of all action types for each character: `['Attacks', 'Defends', 'Heals', 'DamageReceived', 'HealingRecieved']`. Label your axes and make a legend.

b. **(10 points)** Using the matplotlib hist function, create a single plot showing histograms of the `Attacks` for each character. Label your axes and make a legend.

c. **(10 points)** Repeat part b, but this time plot the `Attacks` only for turns where the character was healed.

Problem 3 (25 points): Enter the Mage.

a. **(15 points)** Create a new class called “Mage” in a file called “Mage.h”. This is a damage dealing class, similar to the Rogue, so you will need to implement a new “attack” method. The mage has 18 attack power. However, unlike the Rogue, the Mage starts with 100 mana, and each attack costs 10 mana. When the Mage runs out of mana, they cannot attack any more. They also deal 14 extra damage every 4 turns, unlike the Rogue. Implement the “Mage” class with those specifications. You will also need to modify the “Battle” class to handle your new Mage.

   Now copy the “AshenVerdict.txt” file to a file named “AshenVerdictWithJaina.txt” and include a mage named “Jaina” by adding the appropriate line (remember she has 18 attack power, 0 defense power, and 0 heal power). Re-run the Lich King scenario as in Problem 2.

   ```
   ./WorldOfTextCraft AshenVerdictWithJaina.txt Arthas.txt ArthasAttacks.txt problem3.txt
   ```

   Record the entire output in a text file called “`problem3.txt`”.
   Did you win?

b. **(10 points)** Repeat problem 2, making a new python notebook called “Problem3.ipynb”, with the new file `problem3.txt`. Remember to update the character list! Export the output as “Problem3.html” from jupyter and be sure to upload it to UBLearns.

Problem 4 (25 points) Heroic Mode (Extra Credit for undergrads).

a. **(20 points)** Adjust the "Boss" class to make a configurable "heroic" mode for the Lich King encounter. This means the boss is stronger than usual. In this case, adjust the Boss class to have a configurable amount of mana, and then for this encounter, give him 30 mana in a new configuration file called "ArthasHeroic.txt", that is otherwise a copy of "Arthas.txt" (except for the 30 mana).

   Give the Lich King a stronger, magical, multi-attack, which is OFF by default and must be enabled by checking if the word "Heroic" is in the boss configuration. On every fifth "multi-attack", he deals 50% more damage, at the cost of 10 mana. When the Lich King runs out of mana, he reverts back to the standard multi-attack. You will need to also modify the Battle.cc class's "readNPCConfiguration" function to update the boss to heroic mode. Hint: You may need to use a dynamic cast from the pointer to the base class to be a pointer to the derived class!

   Be careful! Make sure the "default" behavior of the Boss class is still as in Problem 1,2,3 for "non-heroic" mode!

   Re-run the Lich King scenario as in Problem 3a, but now using "ArthasHeroic.txt" (including the enhanced Rogue, and the Mage).

```
./WorldOfTextCraft AshenVerdictWithJaina.txt ArthasHeroic.txt
ArthasAttacks.txt problem4.txt
```

   Did you win?

b. **(5 points)** Repeat problem 2, making a new python notebook called "Problem4.ipynb", with the new file problem4.txt. Export the output as "Problem4.html" from jupyter and be sure to upload it to UBLearns.