COM S 127 - Assignment #6 Grading Rubric

This assignment was assigned on 11-14-2022

This assignment is due by 11:59 p.m. Monday, November the 28th (11-28-2022). It will be considered 'late' if turned in after this time. However, there will be a two (2) day 'grace period' extended until 11:59 p.m. Wednesday, November the 30th (11-30-2022). If the assignment is turned in after 11:59 p.m. Friday, November the 28th, but before 11:59 p.m. Wednesday, November the 30th, it will suffer at 10% 'late penalty' to the grading. The assignment will not be accepted after 11:59 p.m. Wednesday, November the 30th.

Assignment Objective

The purpose of this assignment is to allow students to become more familiar with the use of 'modules.' The student will be tasked with creating their own version of the classic game 'Battleship' - here called 'Naval Battle.'

Please carefully read the following link for details about the game: https://en.wikipedia.org/wiki/Battleship_(game)

This assignment will require the use of several 'list of lists' which will contain the data for the game. It will also require the student to make use of several modules to segregate the game's functionality into coherent groups. The student will be given pre-coded 'start files,' to which they will have to add core functionality. This assignment will also require substantial use of logical step-by-step thinking.

Optionally, for this assignment, the student may completely forego the 'start files' and code their own game 'from scratch.' However, if a student chooses to do this, their submission **must** contain **all** of the modules and functions seen in the 'start' files. Meaning - there should be a gamePlay module which contains all the functions seen in that particular 'start file.' There should be a gameInput module which contains all the functions seen in that particular 'start file.' Etc.

Instructions

Students should study the file provided, and notice the various 'TODO' comments. These indicate tasks in the file that students should complete by typing in their own original code. These 'TODO' comments indicate items in the script which will be evaluated for the final grade on the assignment. The files the student submits **must** be named **navalBattle.py**, **gameBoard.py**, **gameInput.py**, **gamePlay.py**, **targetBoard.txt**, **board1.txt**, **board2.txt**. Further, all of the files the student submits must be included in a .zip file named **navalBattle.zip**.

NOTE: All of the 'TODO' statements are contained in the **navalBattle.py**, **gamePlay.py** and **gameInput.py** 'start files.'

Optionally, the student may add additional 'boardX.txt' files. However, when doing so, the appropriate global variables will need to be updated, and any additional files will need to be included in the final submission. These files should be numbered sequentially, with no gaps in the sequence. For example, submitting board1.txt, board2.txt, and board4.txt, without submitting board3.txt, could cause the script to crash when the user attempts to select board3.txt.

Files Provided

Students will have access to the following 'start files':

- **navalBattle.py**
- **gameBoard.py**
- **gameInput.py**
- **gamePlay.py**
- **targetBoard.txt**
- **board1.txt**
- **board2.txt**

Example Output

```
By: Matthew Holman
[COM S 127 A]

[p]lay, [i]nstructions, or [q]uit?: i


Enter a row and a column to fire on the COMPUTER's ships!
Sink all the COMPUTER's vessels before they sink yours!

[p]lay, [i]nstructions, or [q]uit?: p

Which gameboard to use for the player (1 - 2)?: 1

HUMAN TURN:

 0123456789
0##########
1##########
2##########
3##########
4##########
5##########
6##########
7##########
8##########
9##########

 0123456789
0..........
1.@........
2.@....@...
3.@....@...
4.@..@.@...
5.@..@.....
6....@.....
7..........
8.....@@@@.
9@@........
Select Row (0 - 9): 3
Select Col (0 - 9): 3
The HUMAN targets coordinates (3, 3)
MISS!
```

```
COMPUTER TURN:

The COMPUTER targets coordinates (3, 9)
MISS!

HUMAN TURN:

 0123456789
0##########
1##########
2##########
3###O######
4##########
5##########
6##########
7##########
8##########
9##########

 0123456789
0..........
1.@........
2.@....@...
3.@....@..O
4.@..@.@...
5.@..@.....
6....@.....
7..........
8.....@@@@.
9@@........
Select Row (0 - 9):
```

etc.

Special Notes

NOTE: This assignment is *more* difficult than previous assignments:

- Completing this assignment may require the student to start their work 'before the last minute.' Please plan accordingly.
- The student's script **CANNOT** crash under any circumstances. Any portions of the student's code where the script crashes will receive a zero (0) for that aspect of the game.
    - For example, if a student's script crashes during the 'getHumanInput()' section, the student will not earn the point for implementing that particular feature.

NOTE: Assignments turned in in any other format other the specified types will not be accepted.

- Screenshots of code **will not** be accepted.
- `.sln` files are **not** code files - they contain **no** Python code and **will not** be accepted.
- `.zip`, `.rar`, `.tar.gz`, and other compressed files **will not** be accepted unless specified.
- If a student's submission is not in a `.py` file, when so specified, it will not be graded.
    - **THIS WILL LEAD TO THE STUDENT RECEIVING A ZERO (0) ON THE ASSIGNMENT.**
    - **Students will *NOT* be allowed to re-submit their work in this case.**

NOTE: Submitting the assignment on Canvas may result in either an error message or a 'spinning blue circle' on the submission page. This is the normal/ expected behavior.

- So long as the top of the page reads 'Submitted' in green text, there should not be a problem.
- The student can email a Graduate TA/ UGTA to confirm the status of their submission.
- Students may encounter diffficulties in submitting their assignment due to issues with the Canvas platform itself. However, this will **not** be an excuse to turn in work past the deadline.
    - **THIS MEANS THAT STUDENTS SHOULD NOT WAIT UNTIL THE 'LAST MINUTE' TO ATTEMPT TO SUBMIT THEIR ASSIGNMENT TO CANVAS. THUS, STUDENTS WILL NEED TO 'PLAN THEIR LIVES' IN REGARD TO THEIR TIMELY SUBMISSION.**
    - **STUDENTS SHOULD ATTEMPT TO SUBMIT THEIR WORK *BEFORE* THE FINAL DEADLINE.**

Grading Items

- (**Due Date *AND* File Names *AND* Zip File Submission**) Was the assignment turned in by the due date, **AND** are all the files named **navalBattle.py**, **gameBoard.py**, **gameInput.py**, **gamePlay.py**, **targetBoard.txt**, **board1.txt**, and **board2.txt**, **AND** did the student submit all the files in a .zip file named **navalBattle.zip**? ______ / 1
- (**Name In Script *AND* Delete First TODO *AND* Name In Output**) Has the student typed in their name/ date/ assignment number at the top of the **navalBattle.py** file, **AND** did the student delete the first 'TODO' comment in **navalBattle.py** such that the first thing in the file is their name/ date, **AND** Has the student added their name/ section number to the initial script output?: ______ / 1
- (**choice == "p" Section Tasks**) Has the student completed the choice == "p" section tasks in **navalBattle.py**?: ______ / 1
- (**choice == "i" Section Tasks**) Has the student completed the choice == "i" section tasks in **navalBattle.py**?: ______ / 1
- (**choice == "q" Section Tasks**) Has the student completed the choice == "q" section tasks in **navalBattle.py**?: ______ / 1
- (**getHumanInput() Function Tasks**) Has the student completed the getHumanInput() section tasks in **gameInput.py**?: ______ / 1
- (**getComputerInput() Function Tasks**) Has the student completed the getComputerInput() section tasks in **gameInput.py**?: ______ / 1
- (**_humanTurn() Function Tasks**) Has the student completed the _humanTurn() section tasks in **gamePlay.py**?: ______ / 1
- (**_computerTurn() Function Tasks**) Has the student completed the _computerTurn() section tasks in **gamePlay.py**?: ______ / 1
- (**_printWinner() Function Tasks**) Has the student completed the _printWinner() section tasks in **gamePlay.py**?: ______ / 1

TOTAL ______ / 10