ESC190 PROJECT TOPICS

There are 3 project topics to choose from. **You must select 1 of the 3 available projects**. The weighting has been revised as follows:

> **Interim Report:** submission only. You will be provided with feedback on your Interim report. If you do not submit an interim report, you are assigned a grade of 0 on the project. No late submissions will be accepted.
>
> **Final Report and Code:** 15% of your final grade. Late submissions will be accepted up to 3 days (72 hours) after the stated deadline, with a penalty of 1.5% applied per day. I.e. if you submit your final report and code within 24 hours, the maximum grade you can get is 13.5/15 %. Within 72 hours, this becomes 10.5/15 %.

ACADEMIC INTEGRITY

As an ESC190 student, you are responsible for ensuring the integrity of your work and for understanding what constitutes an academic offence. The work submitted for this project must be your own individual work. If you are unsure whether what you are doing is considered an academic offence, ask an instructor or TA or review the Code of Behaviour on Academic Matters (www.governingcouncil.utoronto.ca/policies/behaveac.htm) To avoid any possibilities of academic infringement, do not reference code from your peers or online; conversely, do not post your code on online forums or share your assignment solutions with your peers.

PROJECT DESCRIPTIONS

PROJECT 1: SNAKE AI

Your name is Voldy. You have a baby pet snake (snek) named Magini, and you both want to play a game with the Moogles.

The game starts off in a playing field where Magini occupies a single cell. You can consider the playing field to be a coordinate system, where the top-most left-most cell has coordinate (0, 0), and cells to the right increase in x value, cells to the bottom increase in y value. Magini has a sort of inertia; until commanded to change direction, she will slither onwards, undisturbed. Magini can move in one of four directions: {UP, DOWN, RIGHT, LEFT}, creating {+1, -1} y and {+1, -1} x motions. You can command Magini once as she changes cells. As soon as Magini moves a single cell, the field refreshes and a TARGET (a Moogle or Harry Pooter) may appear. Only one target will be on the game board at any moment. Once Magini eats the target, the cell resets to a value of zero, and the user's score increments by that score.

Magini becomes angry when she: (1) bumps into a "wall" or (2) bumps into herself. The game ends if this happens or she covers the entire screen.

You will be provided with a game API in C, which you may choose to interface with in C or import into Python.

OPERATIONS

1. Pass commands to change the direction of Magini's trajectory.
2. Augment the API such that it can accept a sequence of moves in a data structure of your choice and execute these moves in the game.
3. Design and implement an algorithm to maximize the points Magini acquires before the game ends.

PROJECT 2: EXAM SCHEDULER

The Registrar's Office (RO) is working on scheduling the Engineering students' final exams. They have approached the class of Engineering Science (2T3) to help them out. They will be supplying you with course exam information on a rolling basis (i.e. not all at once). You are asked to write an interface which performs the functions indicated in the *Operations* section. You are required to tell the RO what format they should provide exam data to you, but at the minimum, it should include the course code, instructor(s), and exam type. You may choose whether the RO provides you with exam location and time, but you cannot enforce any constraints on the data they provide you (i.e. they may provide you with an exam that creates a scheduling conflict).

Your task is to design an efficient data structure (e.g. heap) to implement a priority queue ADT that is capable of performing the following functions under *Operations*.

OPERATIONS

1. Poll whether an exam is ongoing in a given room at a given time.
2. Schedule in one or more exams.
3. Delete one or more exams.
4. Detect exam scheduling conflicts and suggest an alternative time and location for the conflicting exam.
5. Visualize the schedule in real time as the RO gives more exams to schedule.

PROJECT 3: ENGSCI PRESS

The ESC190 teaching team was reminiscing about Lab 4 and dictionaries and couldn't find the right words to describe their feelings in that moment. They ended up sitting in awkward silence for much too long and decided to change topics to the ESC190 projects. It occurred to them to seek the bright minds of the EngSci 2T3 class to create a software to help with their loss of words in critical moments.

You will work with a corpus of English words: https://github.com/dwyl/english-words
Your task is to design an efficient data structure (e.g. BST, hash table) to implement a dictionary ADT that has the following operations.

OPERATIONS

1. Populate a dictionary with a given text file.
2. Look up a word's definition.
3. Insert a new word and its definition.
4. Delete a word and its definition.
5. Suggest another word given an incorrectly spelled word.
6. Generate a specified length story given an initial seed in the dictionary by only using the words in the current dictionary.