EECE 2140: Computing Fundamentals for Engineers (Spring 2022) Roi Yehoshua
Student name: (Due) April 27, 2022

Final Project

1 Project Description

The final project should demonstrate your proficiency in the concepts learned in class by developing a medium-size application in Python. You can write a program on whatever interests you. Below you will find a list of sample projects you may choose from, though you can propose your own system. Completion of the project will include a project proposal, a presentation and a final report.

2 Teams

You can work with a partner or alone (so a group of 2 or 1). If working with a partner, you will be graded accordingly, with the expectation that the project is more substantial than a single person project.

3 Sample Project Ideas

This section provides some examples for applications that you may develop.

1. Design and implement a spell-checker application.

   Create a text file that has some words spelled correctly and some misspelled. Your script should look up each word in the dictionary, and then point out each incorrect word and suggest some correct alternatives that might have been what was intended.

   For example, you can try all possible single transpositions of adjacent letters to discover that the word "default" is a direct match for "default." Of course, this implies that your application will check all other single transpositions, such as "edfault", "dfeault", "deafult", "defalut" and "defautl". When you find a new transposition that matches a word in the dictionary, print it in a message, such as "Did you mean default?" Implement other tests, such as replacing each double letter with a single letter, and any other tests you can develop to improve the value of your spell checker.

2. Build a self tutor for learning a new language.

   Write a script that allows you to indicate the language that you wish to learn. The script should then allow you to say something in English, transcribe your speech to text, translate it to the selected language and use text-to-speech to speak the translated text back to you so you can hear it. Try your script with words, colors, numbers, sentences, people, places and things.

   For language translation, you can use Python's TextBlob library which relies on Google Translate (`https://textblob.readthedocs.io/en/dev`).

3. Build an automatic grader for students' assignments.

   Write a program that goes over a folder containing student' assignments in a Python programming course (in the format of .py files). The program should execute each Python script on a set of predefined inputs, and verify that it generates the expected outputs. The set of test inputs and their expected outputs should be defined in an external text file. The program would then grade each of the assignments based on the number of correct outputs, and generate a file with the final grades of the students.

4. Build a Sudoku generator and solver.

   A Sudoku game is a number-placement puzzle. The objective is to fill a 9×9 grid with digits so that each column, each row, and each of its nine 3×3 blocks contain all of the digits from 1 to 9.

   Your task is to design an algorithm to create a Sudoku grid. The generated grid should have enough clues (numbers in cells) to be solvable resulting in a unique solution.

   Your Sudoku generator algorithm may need to use a Sudoku solver algorithm in order to test whether a generated grid is solvable and to check and that it gives a single solution. The most common type of Sudoku solver algorithm is based on a backtracking algorithm used to investigate all possible solutions of a given grid.

5. Develop a board game, such as checkers or reversi.

   The game can use a command-line interface or a graphical user interface (GUI). For building a graphical interface, I recommend using Python's tkinter library (`https://docs.python.org/3/library/tkinter.html`).

6. Create a twitter bot.

   Part of having a great Twitter presence involves keeping your account active with new tweets and retweets, following interesting accounts, and quickly replying to your followers' messages. You can do all this work manually, but that can take a lot of time. Instead, you can rely on a Twitter Bot, a program that automates all or part of your Twitter activity. The bot can do many tasks on its own such as like/reteweet tweets with particular hastags, follow users who tweets with particular hashtags, etc.

4 Project Guidelines

- Design and plan the structure of your program before writing any code.
- Separate the user interface from the logic of the application by using different modules/classes.
- The source code should be well commented and organized, and run without errors.
- All classes and methods should be documented with docstrings.
- Follow python coding standards regarding variable names, indentation, spaces, etc.
- Use github to manage your code (from day one).

5 Academic Integrity

WARNING: You are not allowed to use any code from an online resource without explicit permission from the instructor. Your project should be built entirely by yourself from the ground up. Any project suspected of plagiarism will automatically receive a grade of 0 and report to the university administration (OSCCR).

6 Project Proposal

The project proposal is a short document (1-2 pages) that describes what you plan to do for the project. It should specify the following items:

1. Project title.
2. Project participants: Who is on the team? If you are working with a partner, is there a clear division of labor?
3. Application description: Describe the main features of the system you are going to develop. Why is your application useful? Who are the potential users of the application?
4. Application design: What will be the structure of your program? What will be its main classes/compnents?
5. Libraries and tools: What libraries/platforms, if any, will you have to learn in order to undertake your project? Provide references where applicable.
6. Results: What is the ideal outcome of the project? Are there risks for not getting your project done on time? If so, what will you do about it?

We will review all project proposals, and they will be part of your grade on the project.

7 Project Presentation

In the final class, you will present your application to the class. The presentation should include:

1. Overview of the application, its purpose and main features.
2. Demo of the system and how it runs for different inputs.
3. Display the main parts of your code.

The presentation should be 5 minutes long, with an additional 1-2 minutes for questions.

8 Project Report

The report should be written in a PDF format and include the following items:

1. Introduction: a brief presentation of the system you have developed, the methodology adopted and the results obtained.
2. Application design: What are the main classes in the system? What are the main features provided by each class? How the classes interact with each other?
3. A class diagram of your system.
4. A link to your github repository and history of your commits.
5. Instructions on how to download/run your system.
6. Libraries and tools: What external libraries and tools did you use in your project? Provide references where applicable.
7. Lessons learned: What did you learn from conducting the project? If you had more time to spend on the project, what would you have liked to do next? What advice about the project would you give to future EECE 2140 students?

9 Grade Breakdown

10% proposal, 20% presentation, 70% final report and code.

10 Timeline

All deadlines, except for the project presentation, are at 11:59 PM. If you are a group of 2, both students should submit their project to Canvas.

- The project proposal is due 3/21.
- Project presentations will happen during the final week of the semester.
- **The final project report / deliverables is due on 4/27.**

There can not be any extensions on the final report, because final grades are due soon after that, and we need time to ensure every project receives a proper assessment.