CHE 120 Project

1 Introduction

The project for this course will be creating a python game in a group of 2-4 people. The due date is end of the day (11:59 PM Waterloo time) on Dec 2nd.

2 Picking the game and your group

You will need to choose a game from one of the references mentioned in the section 6. The games vary in complexity but the more challenging they are, the more fun the project will be!

Alternatively, if you would like a harder challenge (and a chance at bonus marks) you can also create a game from scratch. You have to come talk to me before choosing this option.

Groups can be 2-4 people. Each of you will need to contribute equally and will need to show proof towards this.

Send me an email with the following info by the end of the day on **Nov 1st**:

- Email title : CHE 120 Project
- Email content : Group members, all student IDs, and the game you have chosen.

3 But what do we need to do?

If you are starting with an existing game:

1. Run the code, read the documentation, and build an understanding of how it works.
2. Add comments to the code, describing what it does and **why** it does what it does.
    - Initial the comments to describe who has been working on that section.
    - Add all the full names and initials of the group members at the top of the code as comments so we know who is who!
3. Save the original file with the comments added in and submit it as one file.
4. Once you have these comments, you should have a good understanding of the code. Now it's time to make the game your own!
    - The game rules must be changed from the original (at least one or two fundamental rules).
    - You may want to change the graphics (if it has any).

If you are writing your own game from scratch:

1. You will likely find it useful to follow the same steps as when designing a function; conceptualize, write out structure with comments, write the code, test.
2. I recommend that you test and play your game as earlier as possible; test it and fix bugs as soon as you can rather than waiting until you have the full game implemented.

4 Presenting your work

To showcase your work, you will need to make a small presentation (max 10 minutes) with all group members to describe both the original code and the changes you have made.

Think of this video as an elevator pitch to convince other students to play your game rather than going through code.

This can either be a video or an audio recording of you talking over a slide deck.

4.1 Sections your presentation must include

- Breakdown of roles (who did what).
- Pitch the game and how you changed the rules.
- Challenges you faced.
- Reflection on lifelong learning skills.

5 What to submit

You will need to submit:

- The presentation file,
- your recording of the presentation,
- all the python files (both the original with the comments as well as your changed version),
- any images used in the game,
- an official one page pdf of the changed game's rules.

Submit all files as a zip file to the dropbox on Learn. Mention all group members in any files so that you all get credit!

Remember, the due date is **Dec 2**nd and **no exceptions can (or will be) made.**

6 Sources

There are two references you can use :

6.1 Invent Your Own Computer Games with Python

This books walks you through how to develop several games of increasing complexity. A copy of the games' full source code is available online. To download them, follow the first link "Download the book's resources" from the link above; it's at the bottom of the page, see the screenshot below.

There are also 11 different games from an older version of the book online here.

6.2 Free Python Games

This site has many simplified version of classic games.

6.3 Other sources

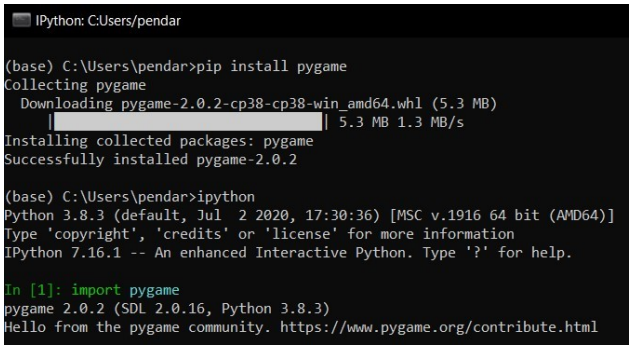
You may use other sources, but cite them and give credit to the original author(s).

7 Adding Pygame to Anaconda

The `pygame` module is needed for many of the projects). To install it, open the Anaconda prompt and type:

```
pip install pygame
```

To check that it has been installed, open `ipython` and run the command `import pygame`. It should work without producing an error.



8 GitHub (optional but highly recommended!)

Since you are working on this amazing project, you should definitely add it to your GitHub! Let's go through the basic setup.

- Create an account on GitHub.
- A useful feature of GitHub is version control (ask me what that is!). To be able to use this for your own projects, I have found a starter article for both windows and mac users. Follow the steps mentioned to create a new repository and commit your first file!

  Note
  If you have never used git before, you will need to define the user before you commit. To do this, type

  ```
  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"
  ```

  in GitBash (windows) or Xcode(mac) with your email and name filled in the strings.
- Once you are done with the last step, you can add your project to your repository to be able to showcase it to the world once your are done!

Here is a Git cheat sheet that might come in handy.

Create a repository with the course name and include your GitHub username as part of your project to get a bonus! You will have to demonstrate that all group members have been actively contributing to the repository.

9 Last note

Any outstanding projects will be considered for additional bonus points (and possible prizes!). Have fun!