Introduction to Programming

CMPT 120

Project 4 – Customization and Enhancements – 100 points

Deadline

Final push due before Friday 15 December 2023

Goals

To finish work on your "robot" simulator term project by adding your own personal flair and choice of enhancement to the program. To apply coding and design best practices as part of your development process.

Preparation

For this assignment, you must have already completed the second version of your project hosted in the *{YourLastname}-Work* repository of our class organization on GitHub.com. Open your Python script in the VSCode IDE to continue editing your code.

Instructions

Create the final version of your project by implementing one of the enhancements below:

★ *Option #1 – Replace the console output with a graphical view in a window.*
- Use the author's graphics.py module to display the room in a graphical window.
- Also provide controls on the window for interacting with the robot.

★ *Option #2 – Redesign your code using object-orientation.*
- Make classes to represent the entities in your program, such as Robot, Room, Obstacle, and Goal.
- Rewrite the simulation to rely on instances of your new classes and methods.

★ *Option #3 – Add other objects to the room and more commands for your robot.*
- For example, perhaps the room contains various "doodads" and matching "bins", and the robot can pick up a doodad and then drop it at a bin.
- The user earns points for every doodad that correctly sort among the bins.

★ *Option #4 – Turn the simulation into a race against the "clock."*
- For example, after reaching the goal, maybe the goal disappears and one of the obstacles turns into a second goal. Then upon reaching that goal, it disappears and another obstacle turns into a goal. This continues until all the obstacles are gone.
- Keep track of the number of commands it takes to finish, and display the final "time" (i.e., umber of moves) to the user.

★ *Option #5 – Batch commands and undo history.*
- Keep a record of the moves the robot has made. Add a command that allows the user to "undo" a move, reversing it an removing it from the history.
- Then, allow the user to enter multiple commands at once, and the robot then executes all the commands one after the other until the batch is finished.

Submitting

You must push your changes to GitHub before the due date.

Note: Pushing regularly will reduce the risk of losing your work, so do not wait until after you have made all changes and commits before pushing.

Advice

Follow coding best practices with mnemonic naming, consistent indentation and spacing, frequent comments., and appropriate use of user-defined functions.

Be sure to stage your changes a little at a time and commit frequently. Push any local work to your remote GitHub repository regularly. Don't forget to:

- Write short but meaningful messages for every commit. (Tip: Consider using the instructions themselves as your commit messages.)
- Look at the differences between successive versions of your code. You can do this on the GitHub website as well as in GitHub Desktop app itself.

Test, test, test… and test again. Then test some more. When you think you've tested enough, go back and test yet again. Then get someone else to test for you while you test theirs. Etc.