Capstone Python Project – *Features -- DRAFT*

CSSE 120, Introduction to Software Development

General instructions:

The following assumes a 3-person team. If you are a 2-person or 4-person team, see your instructor for how to deal with that.

General Project Requirements:

- All features MUST be implemented in a ***Graphical User Interface (GUI)***. This means that you ***should NOT be using the Console for any of your input or output.***
  - All team members ***must*** contribute to the GUI.
- Each team member must complete one green feature, one blue feature, and one yellow feature.
  - This is a ***BARE MINIMUM*** and will result in a C for the project.
  - Green and blue features are simpler, yellow are more sophisticated.
  - Uncolored features are open ended and provide room for creativity.
  - Several features contain "Advanced Options" that are not required, but are recommended for a higher grade.
- To receive an A on this project, you must complete the minimum requirements above, some of the uncolored features, and some of the advanced features. There is no set number for this, do the best your team can.
- Use as many different GUI widgets as you can.

The best projects will take care to re-use each other's GUI, functions and data wherever practical.

Grading and demos:

The grading is based on:

- the quality of your individual contributions to the project;
- your attention to detail to the Trello board, SVN, and your code.
- completeness of each feature based on the specification below.

On Friday of 10th week, your instructor will require that each team give a brief (5-10 minute) demo of all of the features that were implemented.

Due date:

The final project code is due at the start of class on Friday of 10th week.

Features (brief version – longer version coming soon):

Each of the following features will have a longer, more complete, description. The following descriptions convey the basic idea of the feature, but not its details. ***THIS IS A DRAFT – detail may change.***

1. The user can ***connect to the robot*** , with a way to specify the robot's IP address.
2. The ***GUI indicates,*** for ***each Sprint*** and ***each team member***, ***the total hours that the team member worked during that Sprint.***
3. Play ***N random notes***, where the user specifies N. The notes must not be "clipped".
4. ***Move autonomously***, by going a specified ***distance*** in a specified ***direction*** at a specified ***speed***. That is, the user can set the direction (forward, backward, spin left or spin right) and the distance and speed (each in some reasonable units). Then, the user can make the robot go (e.g. by pressing a *Go* button) and the robot should move the specified direction for the specified distance at the specified speed, with some reasonable accuracy.

   An important by-product of Feature #4 is to provide a good set of functions that teammates will use for most of the movements that they ask of the robot.

5. ***Move autonomously***, by going until a specified ***sensor*** reaches a specified ***threshold***. Sensors should include the bump sensors and the line-following sensors, at the least.
6. Be ***tele-operated*** (i.e., remote-controlled, like a remote-control car) by using buttons and other GUI widgets.
7. ***Follow a curvy black line*** using PID control.
8. Move through a sequence of user-specified ***waypoints***.
9. ***Use the camera to follow a moving object.***
10. Robot ***composes*** new music.
11. Do ***sophisticated movements***, e.g. trace a regular polygon, parallel park.
12. Do interesting things with its ***internal*** sensors.
13. Do interesting things with ***external motors and/or servos.***
14. Use ***swarm techniques*** and/or distributed algorithms to accomplish interesting things.
15. Use ***parallel algorithms*** (in processes and/or threads, in a single processor or across cores) to accomplish interesting things.
16. Use ***internet communication*** and/or ***files*** to do interesting things.
17. ***Compose a fictitious bio*** for itself and/or for you.
18. Use a ***Leap Motion device*** (and accompanying Python software) to control the robot with hand movements.
19. ***Interact with a different kind of robot***, e.g. a quadcopter or BERO robot.
20. Do something interesting… ***[You suggest what!]***