Capstone Python Project – ***Features***

CSSE 120, Introduction to Software Development – Summer, 2016

General instructions:

This document lists Features that you can implement. As a rule of thumb, you should implement:

- An easy (green) Feature during Sprint 1.
- One or two harder Features during Sprint 2.
- One or two harder Features during Sprint 3.

Important: Read the document on "How you will be graded." ***Half of your grade involves following the specified process*** and working well with teammates. Follow the process exactly – those are very easy points to earn. **Don't get so carried away with implementing Features that you fail to earn the easy points on process!**

You can implement almost ANYTHING that interests you. Some of you will concentrate of doing fancy things with the Graphical User Interface. Some will make the robot do things. Some will write code for a robot simulation. The list below contains ideas for what you might implement. **YOU should choose what *interests* you. It is perfectly OK to do things NOT on the list (see Feature 30).**

The colors are ROUGH indications of difficulty. From easiest to hardest: Green, Blue, Yellow, Dark Yellow. The uncolored Features vary wildly in their difficulty, depending on what you do with them.

Features:

1. [Team-coded, with your instructor] The user can ***connect to the robot,*** as well as ***shut it down cleanly***, with a way to specify the robot's IP address.
2. ***The robot can GO STRAIGHT*** in a specified ***direction*** (***forward or backward***) for a specified ***distance (in inches or centimeters)*** at a specified ***speed***, at some reasonable degree of accuracy.

   Additionally, the code contains ***functions that the entire team can use*** whenever the robot needs to go a specified distance.
3. ***The robot can SPIN (in place)*** in a specified ***direction*** (***left/right***) for a specified ***number of degrees*** at a specified ***speed***, at some reasonable degree of accuracy.

   Additionally, the code contains ***functions that the entire team can use*** whenever the robot needs to spin a specified number of degrees.
4. The ***GUI indicates,*** for ***each Sprint*** and ***each team member***:
   - The ***total hours*** that the team member worked during that Sprint.
   - The ***cumulative hours*** that the team member has worked up to and including that Sprint.
5. The robot can be ***tele-operated*** (i.e., remote-controlled, like a remote-control car) with **keyboard keys**.
6. The robot can ***spin and move forward/backward to track an object*** using its ***camera*** (for determining what spinning to do) and its ***front proximity sensor*** (for determining what forward/backward motion to make).

7. The robot can ***search for an object by moving its head left/right and up/down***, and then ***move so that its head is pointing straight at the object***.
8. [Team-coded, with your instructor] ***Long-running loops can be interrupted.***
9. ***The robot can use its reflectance sensors to follow a curvy black line,*** using ***Bang-Bang control*** as well as using ***PID control***.
10. ***The robot can use its encoders to go straight,*** using ***Bang-Bang control*** as well as using ***PID control***.
11. ***The robot can use its camera and front proximity sensor to follow an object,*** using ***Bang-Bang control*** as well as using ***PID control***.
12. Given a list of x/y coordinates (***waypoints***), the robot can ***move to each***, pausing briefly at each waypoint, using Manhatten movement (i.e., all movement is along the x-axis or y-axis).
13. The robot can ***"talk" to another*** robot using head movements and other forms of communication.
14. The robot can ***parse files with songs and play the songs*** using its buzzer.
15. The robot can move quasi-randomly, using its sensors to ***avoid objects, stop at lines***, and more.
16. The robot can ***display emotion***.
17. The robot can ***compose*** music, and then play its compositions.
18. The robot can ***watch a conductor and play music accordingly***.
19. The robot can ***compose a fictitious bio*** for itself and/or for you.
20. The robot can do ***sophisticated movements***, e.g. trace a regular polygon, parallel park, and more.
21. The robot can ***tweet***!
22. Use a ***Leap Motion device*** (and accompanying Python software) to control the robot with hand movements.
23. Do ***interesting things*** (beyond those already listed) with the robot's ***camera***.
24. Do ***interesting things*** (beyond those already listed) with the robot's **standard sensors**.
25. Do interesting things with ***additional motors, servos and/or sensors.***
26. Use ***swarm techniques*** and/or distributed algorithms to accomplish interesting things.
27. Use ***parallel algorithms*** (in processes and/or threads, in a single processor or across cores) to accomplish interesting things.
28. Use ***internet communication*** and/or ***files*** to do interesting things.
29. ***Interact with a different kind of robot***, e.g. a quadcopter or BERO robot.
30. Do something interesting… ***[You suggest what!]***