RPAL ROS & Programming Skills Assessment

1 Objective:

Demonstrate understanding of basic ROS usage and Python (or C++) programming, as well as some basic robotics concepts. This project will be used to assess your readiness to begin working in RPAL as an undergraduate researcher.

2 Collaboration Policy:

This assignment **must be completed alone**. You are permitted to use any internet resources (short of asking for help on a forum, etc.), books, tools, what have you, but you may **not** collaborate with anyone and you may **not** ask any grad students or anyone else for assistance. This is an assessment of you and your current skills; if you violate this policy we cannot make an accurate assessment and will not be able to work with you.

Note that you can come to us if you are truly stuck. If the issue is a bug or simple misunderstanding with the provided code, we can help you out and you can get back to the assessment. **However**, if you come to us and the provided code is working correctly (i.e. the error is with your code), that will constitute termination of the project and we will assess your work in whatever state it is in.

3 Running the simulator framework

We have provided a simulation framework for you to develop and test your code. In order to use the framework, you'll need to carefully follow these instructions.

3.1 Setup

Do these steps *once*, before starting your work. We also recommend using Git to store your partial progress. Note that successfully completing the setup process is part of the assessment.

3.1.1 ROS

Follow the instructions here: `https://wiki.ros.org/melodic/Installation` to install ROS Melodic.

3.1.2 V-REP

The simulation framework is based around V-REP, which you can download at that link. Get the "Pro EDU" version for Linux and extract the files to some directory (`~/V-REP` is a good choice). Finally, copy the file `libv_repExtRosInterface.so` from this directory to wherever you extracted V-REP.

3.1.3 Catkin Workspace

Before starting the project, you need to set up a catkin workspace. To do this, follow the instructions here: `https://wiki.ros.org/catkin/Tutorials/create_a_workspace`. Afterwards, copy the `src` subdirectory from this package into `<YOUR CATKIN ROOT>/src/rpal-ros-test` and run `catkin build` to build the initial workspace.

3.2 Using V-REP

The second part of this assignment will require you to run V-REP. To this end, we have provided a V-REP scene file, `assessment.ttt`, in this directory. Use the following steps to run the simulator:

1. Start `roscore`.
2. (assuming you extracted V-REP to `~/V-REP`), run `~/V-REP/vrep.sh assessment.ttt` in this directory.
3. Press the "Play" button in the upper toolbar.
4. Run the commands you need for the particular problem.
5. Run the commands to execute your code.

To start the simulation, you must hit the play button.

4 Assignment:

RPAL primarily uses ROS and Python for research coding. Thus, you will need to write Python code using rospy to be an effective researcher in the lab. If you need a refresher for Python, we suggest this quick reference[1]. You should also use the numpy library for matrix calculations and linear algebra, which is common in robotics research.

If you need a reference for ROS, we suggest the ROS "Getting Started" guide, the rospy documentation, and Jason O'Kane's "A Gentle Introduction to ROS"[2].

Implement code to solve the following problems, using the provided simulator framework. Please also document any difficulties you encountered and how you overcame them in a file named `problems.txt`, which you should include with the final submission. When you're done, bundle up the *whole* `rpal-ros-test` directory into a `.tar.gz` and send it to us for assessment.

[1] Note that we use Python 3 in most research.

[2] Note that AGITR uses C++ instead of Python. The concepts will be the same, but the language used is different.