David Dang
djd122
12/7/18

EECS 473 – Grad Student Project

Link to Github repository containing all relevant files: https://github.com/DangD96/eecs473_project.git
Video: djd122_grad_project.mp4

Protocol:

1. roslaunch irb140_description irb140.launch (sometimes the environment launches with an error or some of the toy blocks do not spawn. If this happens, CTRL+C to close gazebo and retry until the environment launches without issues)
2. rosrun magic_object_finder magic_object_finder
3. then run my node - rosrun cartesian_motion_commander jenga_tower_builder

Packages Required Under Src Directory (take these from the linked Github repo)

1. irb140 (contains edited .launch, .h, etc files)
2. magic_object_finder
3. cwru_stickyfingers

Background

For my EECS 473 project, I plan on using the IRB140 robot to pick up various toy blocks in Gazebo and use them to assemble a structure. I intend on assembling a small Jenga tower consisting of nine total blocks.

Since I am doing this project primarily on my Virtual Machine, I had to make sure that the robot environment would not cause my virtual machine to crash. I went to Dr. Newman for some help and he made edits to both the irb140.launch and irb140.xacro files.

This disabled the simulated Gazebo camera that accompanied the IRB140. Rviz was disabled as well. The stream of information from the camera was too much for my processor to handle, and it caused Gazebo to crash whenever I would try launching the robot in the past. With both the camera and Rviz disabled, I was able to launch the IRB140 robot without any issues with my processors. Dr. Newman also showed me how to add more toy blocks into the Gazebo environment through editing the irb140.launch file. This was important because my project relies on having toy blocks present so the robot can manipulate them.

My Approach

I first wrote a test node that made sure the magic_object_finder action server could identify toy blocks that spawned in Gazebo. Once I was convinced that the magic_object_finder was able to identify my blocks, I started working on integrating it into my main node.

My main node is named, “jenga_tower_builder.cpp” and is located in the irb140/cartesian_motion_commander package. I included a link above to a newly created Github repository that contains all of the relevant files for this project.

I further edited the irb140.launch file so it would spawn more toy blocks, up to nine. This was an iterative and painstaking process that involved editing the .launch file, checking to see where the blocks spawned, and repeating to make sure the blocks did not spawn on top of each other or out of the robot's reach.

In my node, I hard-coded the names of all nine toy blocks so the magic_object_finder could properly identify them. Once all nine toy blocks were spawned in an area that was pretty close to the robot, I wrote some test code that told the robot to go pick up the block furthest away from it (toy_block9). If the robot could pick up the furthest block without issue, then the other blocks would be no problem.

Once that was confirmed, I had to spend some time manually rearranging the blocks into the Jenga tower I wanted the robot to build. Once I was satisfied with the positions of the blocks, I jotted down their (x,y) coordinates. These coordinates were then hard-coded in my main node as the (x,y) destinations of where I wanted each block to be placed.

My main node basically tells the robot to look for each block in sequential order. When the robot finds the specified block, it goes over to it, picks it up using the vacuum gripper, moves the block over to its target location, and drops it down. This is repeated for all nine blocks until the small Jenga tower is completed (see video, "djd122_grad_project.mp4").

Limitations

I attempted to have the robot place the blocks down in hard-coded orientations [x,y,z,w]. However, I had a hard time getting this to work. Whenever I would try hard-coding a quaternion for a block when it arrived at its destination, the robot's action interface would display a message about how no IK solution was found and then proceed to glitch.