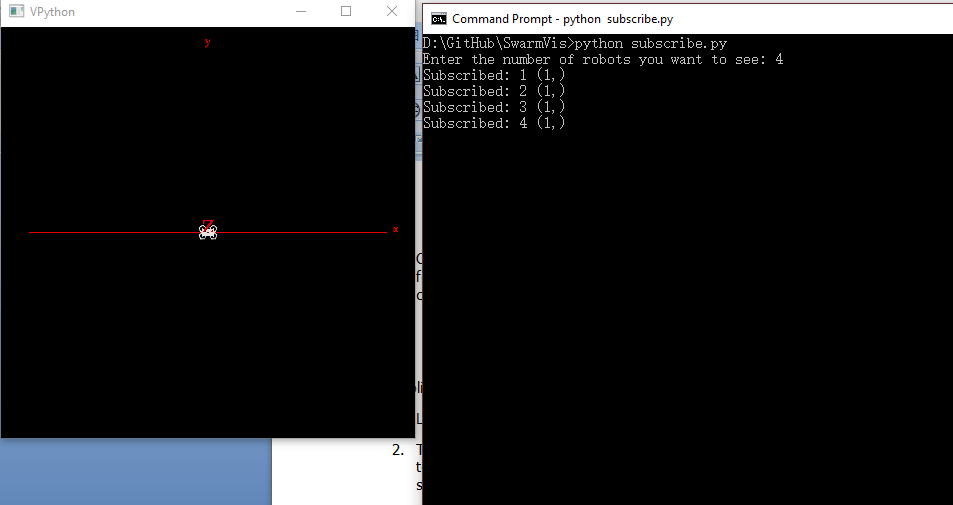
**Installation:**

Paho, Vpython and backend packages if needed

//words in boldface are the procedures to run the simulation//

**Subscribe:**

1. **Open cmd, and run “subscribe.py”, it will automatically listen to incoming messages to “subscribe”, and then subscribe to the appropriate topic.**

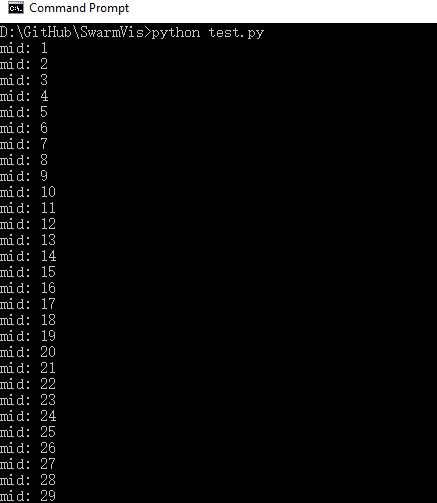
**You are done, don’t close this!**

**Publish:**

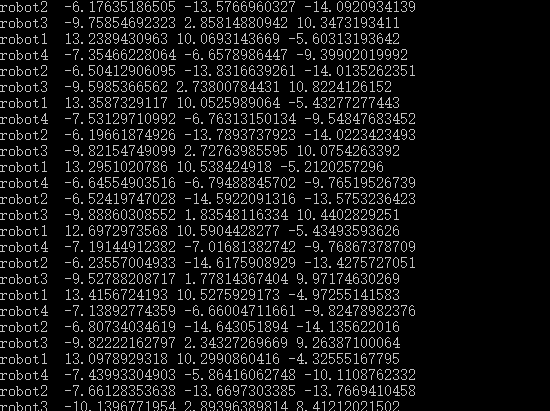
1. Look into “publish.py” and choose the method of publishing you want to use
2. Then edit the “runpublish.py” to utilize the methods in “publish.py”; each robot publish to an unique topic(robot1, robot2, ….), we will first publish to “subscribe” to inform the robot’s type and its topic; we use threads to let data published at the same time
3. For the simulation, we are using rbtpublishseperate function; for later real experiments, probably simplepublish function will be used
4. **Open a new cmd other than the one for subscribe and draw; Run “runpublish.py” in cmd, note that if the Internet is not stable or the server of neptune.usc.edu crashed, it may not publish; when encountered this problem, try again with a new cmd later or change ports in “subscribe.py” and “publish.py” into local ports to simulate**
5. If you want to increase the number of robots, in “runpublish.py” publish to “subscribe” in the form of “topic type” like this: simplepublish("subscribe", "robot1 car"); then create a new thread in the form of threadname, topic like this : thread.start\_new\_thread( rbtpublishseperate, ("thread1", "robot1", ) )

each robot should have its own threadName and topic

1. When there are four robots, the output should look like this for publish cmd:



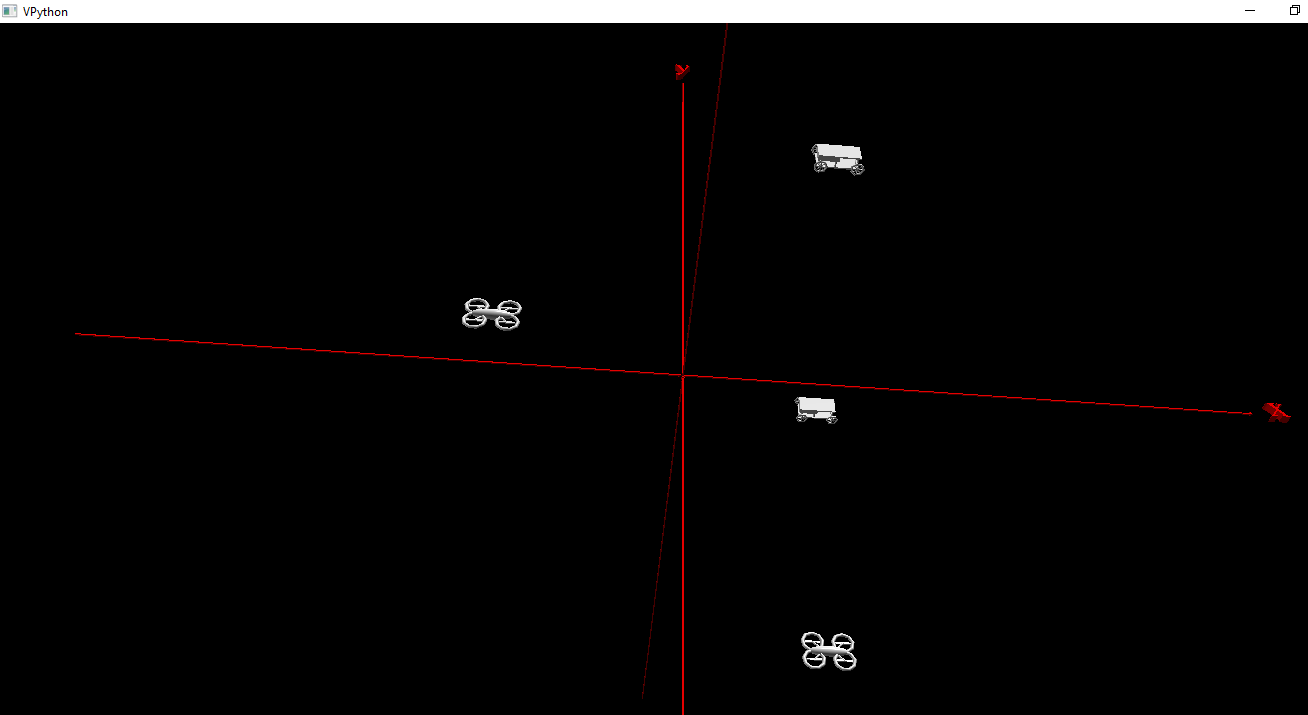
The output should look like this for subscribe cmd:



And now the drones and cars should be moving in the canvas now. to change the rate of the movements, change the sleep time in “publish.py”

Canvas:

1. Now the robots are moving, you can move the visual angles and can zoom in or out: Right button drag or Ctrl-drag to rotate "camera" to view scene. Middle button or Alt-drag to drag up or down to zoom in or out. On a two-button mouse, middle is left + right
2. To change the look of canvas or the models of robots, look into “construct.py”, in which has a drone class to create drone, a class to create car, a create\_coordinate function to create xyz axis
3. The simulation should look like this:



1. You can add trails by set trail = True in “construct.py”, and you can add texts showing the current xyz location of each robot; but that may cause the simulation to be slower