LAB#3

VISUAL RECOGNITION AND OBJECT RECOGNITION

I. OBJECTIVE

To introduce students to visual recognition and computer vision as applications to AI using Cozmo.

II. DUE DATE

During the class

III. EQUIPMENT

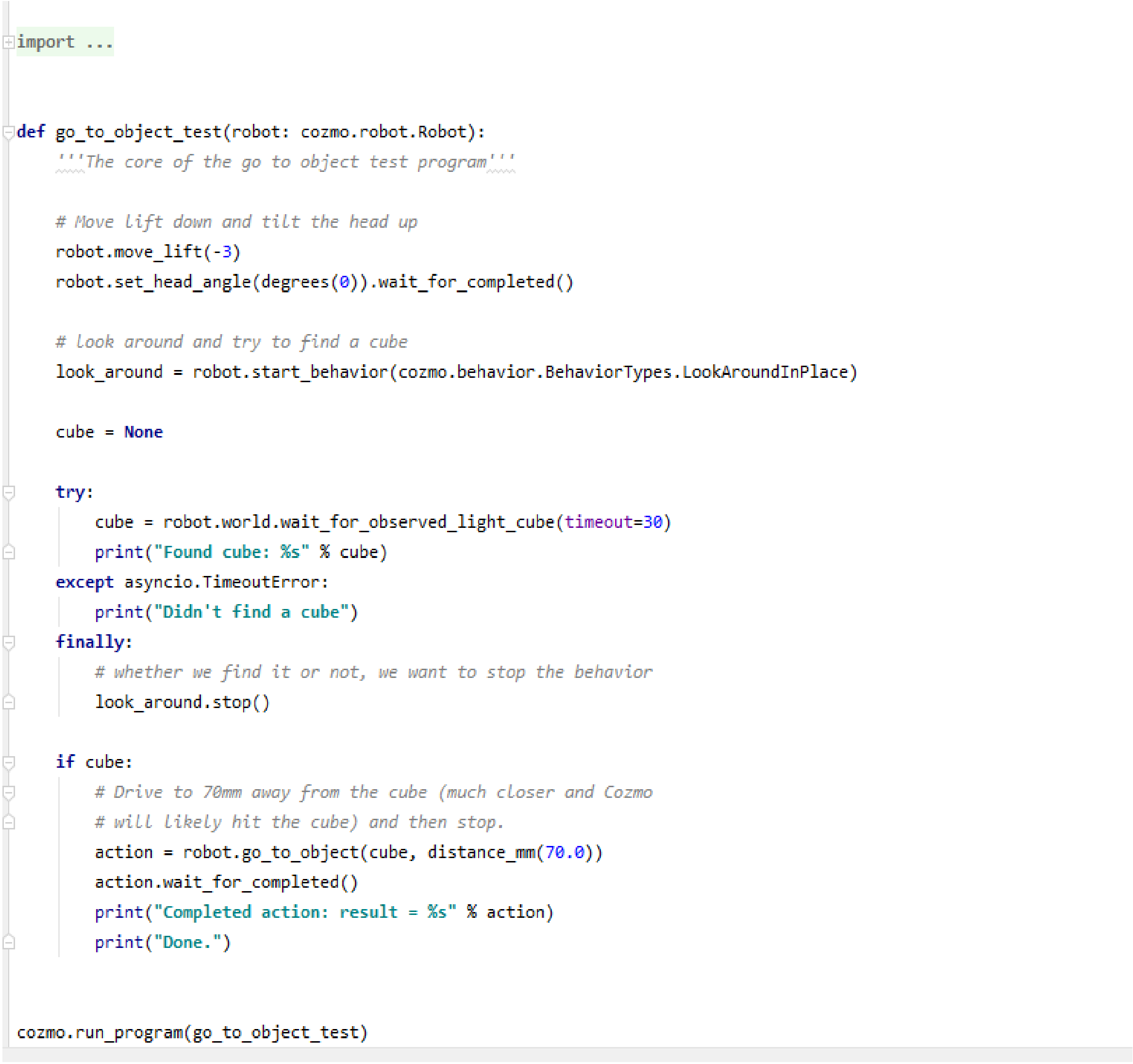
1. A Cozmo Robot along with his charger
2. An phone or tablet to run free Cozmo app along with the connecting wire
3. Laptop or desktop to code.
4. Robot's cubes

IV. PRE-CLASS EXERCISE

1. Watch the video on object recognition and object manipulation here.
2. Generate your own Aruco marker from here. Bring a printout to the class.

V. LAB EXERCISE

1. Navigate to '*cubes and objects'* folder in the tutorials section of the downloaded examples
2. Connect to Cozmo with SDK mode enabled in the app.
3. In the command prompt, execute the stack_or_roll python script
4. Observe Cozmo picking and stacking the cubes.
5. Analyze the source code in the Python IDE of your choice and discuss that with your partner.
6. Execute and analyze the sample code for pickup furthest, and go_to_object_test



```
import ...


def go_to_object_test(robot: cozmo.robot.Robot):
    '''The core of the go to object test program'''

    # Move lift down and tilt the head up
    robot.move_lift(-3)
    robot.set_head_angle(degrees(0)).wait_for_completed()

    # look around and try to find a cube
    look_around = robot.start_behavior(cozmo.behavior.BehaviorTypes.LookAroundInPlace)

    cube = None

    try:
        cube = robot.world.wait_for_observed_light_cube(timeout=30)
        print("Found cube: %s" % cube)
    except asyncio.TimeoutError:
        print("Didn't find a cube")
    finally:
        # whether we find it or not, we want to stop the behavior
        look_around.stop()

    if cube:
        # Drive to 70mm away from the cube (much closer and Cozmo
        # will likely hit the cube) and then stop.
        action = robot.go_to_object(cube, distance_mm(70.0))
        action.wait_for_completed()
        print("Completed action: result = %s" % action)
        print("Done.")


cozmo.run_program(go_to_object_test)
```

Figure 1: go_to_object_test sample code

7. Each Aruco marker has a unique identification inside of it. Write Python code to program Cozmo to go towards your team's marker when it sees it.
8. Place the Aruco marker near the wall/PC and present a demo of the functionality.

VI. POST-LAB SUBMISSION

Submit your source code along with your Aruco marker image before next class.