LAB 2: INTRODUCTION TO COZMO/VECTOR
Part 1 Due: Thursday, September 10th, 11:59pm EST
Part 2 Due: Tuesday, September 15th, 11:59pm EST

The objective of this lab is to get familiar with the functionality provided by the Cozmo/Vector SDK. This lab has 2 parts: Part I (Checkpoint) is an individual assignment (25 points) and Part II can be done either independently or with a partner (75 points).

Lab Part 1 [25 points] (complete individually): This checkpoint is designed to verify that you are able to run the robot and that it's operating correctly.

1. Print and cut out the symbols you saw in the Lab 1 images, we will use them in this and other labs. The symbols can be found in an accompanying PDF. Although they are in color, you can print them in black and white if that's easier. Our Lab 1 classifier is trained on grayscale images. The symbols are named as follows:

drone   inspection   order   plane   truck   hands   place

2. Setup your robot!

   - Complete the installation of the Cozmo/Vector SDK on your computer.
   - Download the Cozmo/Vector app from the app/play store.
   - Go through the connect process. This requires you to connect to the robot using wifi.
   - Connect your phone to your computer using USB.
   - Go to settings on the app and click "Enable SDK".
   - Feel free to test out the example code that comes with the SDK.
   - *Note: Turn off Cozmo by placing him on his charger! Turn off Vector by pressing the black back button with the LED strip.*

3. Download the **checker_cozmo.py** or **checker_vector.py** file provided with this lab. This code will serve two functions:

   - *Verify basic robot functions:* The first part of the checker file will test the selected functionalities of your robot: audio, motors, gyroscopes, and accelerometers. You will not have to do anything for the audio and motors test. For the gyro and accelerometer test, please pick up and gently shake the robot. Listen to the robot for when to do this.

- *Verify the camera is working properly*: The second part will collect three images with the robot to ensure your camera is working properly. When prompted by the robot, hold up the required symbol (inspection, drone, or order) a few inches in front of the robot. There will be a total 5 second pause in between each image capture: a 3 second pause once the robot says the name of the symbol to be shown, then another 3 second pause. The robot will take the pictures, and the resulting images will be automatically assigned a unique filename; do not change the filename.

To run the code, execute one of the following, depending on your robot model:

```
python3 checker_cozmo.py 1 inspection drone order
```

```
python3 checker_vector.py yourSerialNumber 1 inspection drone order
```

*Note: An **/outputs** folder should be included with the lab files, but make sure one exists in your directory. **yourSerialNumber** is the serial number of your Vector found on the bottom of your robot.*

4. Open the images captured by your robot which will be found in the outputs folder. If there is any unexpected noise (e.g. random lines across the image, extreme blurriness, etc.), please reach out on Piazza or directly contact a TA (if in doubt whether the images are normal or not, it is better just to ask).

5. **Submission:** Upload the outputted text file **test_output.txt** and the three captured images located under **/outputs** to show that your robot is functioning properly. Submit the **/outputs** folder as a zip file. Lab 2 Checkpoint is submitted separately and individually. To receive full credit for Part I, each student must upload the files to GradesScope under the Lab 2 Checkpoint assignment by September 10th at 11:59pm.

Lab Part II [75 points] (complete individually or with a partner): Write a Finite State Machine that encodes the following robot states and behavior:

Welcome Agent Yellow Jacket to the CS3630 Intelligence Agency where we like to build tiny robots to do our spy work for us! Robot Cozmo and Robot Vector are our top functioning spies and this week you have the opportunity to work with them.

Your mission, Agent Yellow Jacket, should you choose to accept it, is to complete the following four missions with your assigned robot (and lab partner if you choose) and successfully upload your mission report (aka a demo video) to our secret records (aka Gradescope). This pdf will self-destruct after you submit your assignment.

Mission Field:
This class usually uses an "arena" that the robot drives around in. However, since we weren't able to distribute those because of COVID-19, please try to make a make-shift arena with the perimeter of the arena being 26in x 18in (as seen in the figure below) and the walls being 4 inches or higher. You will only need to make the walls of the arena to enclose the robot, as any hard and ideally flat surface will serve well as the bottom of the arena. This assignment, and all future assignments, will make use of this arena.

> *Mission Note:* Keeping all parts of the arena wall one solid color may help in avoiding creating additional noise in your Cosmo's/Vector's vision.

> *Mission Note:* One way you could set up the arena is by taking some tape to mark the perimeter of the arena and cut out some cardboard to enclose the perimeter with 4 in or higher walls. Alternatively, you could line up books around the perimeter to create the walls.
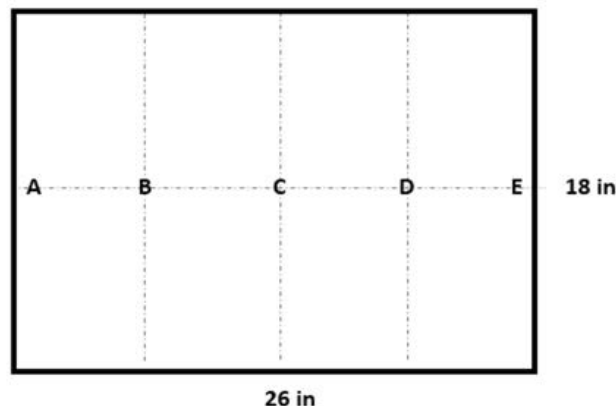
Mission 1: Surveillance (aka the Idle starting state)
Background: It's time to get in the field.
Activity: Look out for the secret symbols by monitoring the stream of images from the camera. Classify each symbol (image) using the model you developed in Lab1. If one of the symbols is recognized (i.e. not "none"), use the built-in text-to-speech functionality to have the robot say the name of the recognized symbol, then switch to the appropriate state (see below).

Mission Note: The Cozmo SDK can provide both grayscale and color images, at resolutions of 320×240 and 160×240, respectively. More specifically, the SDK reduces the width resolution of color images by half, transfers them from the robot, then resizes color to match the grayscale at 320×240. You are welcome to use the color images if you find it helpful, but be aware that as a result of rescaling they are not as detailed as the grayscale images. The Vector SDK is similar and you can reference the **annotate_image** function to resize the image by a scale factor or a specific width and height (the height and width are bound to the original aspect ratio). The original image size is 1280 x 720 or 640 x 360 depending on the function used. You may also find it conducive to image classification to manually crop/manipulate the image arrays.

Mission 2: Defuse the Bomb (aka state activated by showing "order" symbol)
Background: There is a bomb (oh no!). Defuse it.
Activity: Place a cube at point C on the arena. Start your robot at point D on the arena and directly face the cube. The robot should locate the cube (any cube if you have more than one), pick up the cube, drive forward with the cube to the end of the arena (point A), put down the cube, and drive backward to the robot's starting location. Then return to the Idle state.

Mission 3: In the Heights (aka state activated by showing "drone" symbol)
Background: SOS! Someone's spying on us! Inform headquarters.
Activity: Have the robot drive in an "S" formation. Show an animation of your choice on the robot's face. Then return to the Idle state.

> *Mission Note*: Look into the **drive_wheels** function for cozmo or the **set_wheel_motors** function for Vector.

Mission 4: Burn Notice (aka state activated by showing "inspection" symbol)
Background: You've been given a burn notice and must be disavowed.
Activity: Have the robot drive in a square, where each side of the square is approximately 20 cm. While driving, the robot must continuously raise and lower the lift, but do so slowly (2-3 seconds to complete lowering or raising the lift). Simultaneously, the robot must say, "I am not a spy". Lower the lift at the end of the behavior, and return to the Idle state.

Mission Report: (aka your submission to gradescope)
Part II will be graded based on a video you submit of your robot executing all four of these tasks without resetting your code. Start the video by displaying your and your partner's GT username (e.g. gburdell9) on a piece of paper in the middle of your arena (you can remove the paper afterwards) and then run your code. Your robot should begin in the idle state and you should sequentially place the appropriate image in front of the robot to complete the desired tasks. <u>You may physically move the robot back to a good starting position after completing a task, but you should not have to restart your code.</u> The video must be continuous (no stitching clips together) and 3 minutes or under total (all states visited).

Please also submit your code as a zip file named **Last1First1_Last2First2.zip**, corresponding to the first and last names of partner 1 and 2, respectively. The zip file can contain just a single python file, but turning it into a zip file will prevent Gradescope from renaming the file. Also make sure you enter the names of both partners in a comment at the top of the Python file. Only one partner needs to upload the submission on Gradescope. If you are working without a partner, only one name needs to be listed in the file header and file name.

Grading Rubric:
[5 pts] Say recognized symbol
[15 pts] Pick up the cube when order symbol picture is shown
[15 pts] Drive the cube forward, put it down, and return to starting location
[15 pts] Drive in an "S" formation when drone picture is shown and show animation
[15 pts] Drive in a square when inspection symbol picture is shown
[15 pts] Raise lift up and down while driving in square