1. Questions:

**Simulation Environment: Robot Virtual Worlds>Utility Tables>Metric Distance Utility**

**1. Animate a bouncing ball on the LCD screen.**

1

Use the drawEllipse and eraseEllipse functions to draw and erase the circle. The circle should move in a random direction (use a random value from the function rand()%7-3) generates a random 𝑣! and 𝑣" velocities. See Figure 1-a. Upon reaching the limits of the LCD screen (LCD screen is 178x128 pixels in size), the ball should bounce (if you hit in the X direction, the velocity reverses and the, same is true for the Y direction). See Figure

*Figure 1: (a) Starting position (b) the ball bounces*

1-b. The program should continue until a user terminates.

**Deliverable**: Complete code for this part of the question. Screenshot of the ball hitting one of the walls and then bouncing from it (one before and one after)

**2. Add motor motion**

Now program the robot to move in the following way:

* + -  The robot remains at its starting position.
  + -  The robot makes a right (left) point turn when the ball bounces on the right (left) edge and

go forward (backward) when the ball bounces on the top (bottom) edge.

* + -  During all the robot movements, the motor power is 10 times that of the speed of the ball

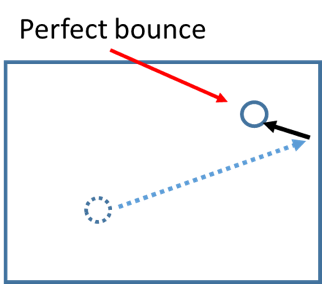
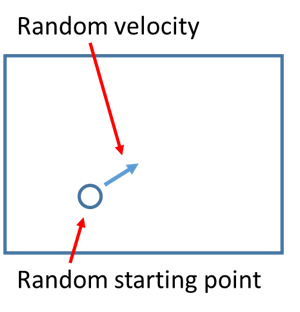
given by 𝑠𝑝𝑒𝑒𝑑 = √(𝑣# + 𝑣#\* where 𝑣 and 𝑣 are velocities in the X and Y directions. !"!"

Make use of the functions for turning and moving forward in Lab 1. **Deliverable**: Complete code for this part of the question.

3. **Measuring the distance traveled by one wheel**Assume an EV3 wheel has a diameter of 56 mm. Therefore, for a 360 degrees rotation, a wheel travels 176 mm (from the formula of circumference: 𝜋𝐷). Using the method getMotorEncoder(), for Q2, calculate how far each wheel travels in Q2 and display the distances traveled on the bottom of the LCD. For Q2, you may want to reduce the height of the ball bouncing to be able to display the distances.

**Note:** Use resetMotorEncoder method to start the encoder counting from 0 each time you run the program. Refer to the help document for reading motor encoder values: https://www.robotc.net/WebHelpMindstorms/index.htm#Resources/topics/LEGO\_EV3/R OBOTC/Motors/getMotorEncoder.htm?Highlight=getMotorEncoder and

2

page2image63504576page2image63505728

https://www.robotc.net/WebHelpMindstorms/index.htm#Resources/topics/LEGO\_EV3/R OBOTC/Motors/resetMotorEncoder.htm

**Deliverable**: Complete code for this part of the question. Snapshots of LCD screen with the displayed speed.

4. **(Only for CSC-475 students)** Using motor encoder, now we have a way to check if we have covered a specific distance. Now, use motor encoder to traverse 1 m in a straight line. Use **setMotorSync()** to run both the motors at the same time. Discuss if there is any difference in the robot’s performance compared with the distance traveled in Q6 of last lab. Point out explicitly in what way the two methods are different for moving the robot.

**Deliverable:** Complete code and discussion. A snapshot of the robot at 1 m mark.