Report of Assignment 2
Pathplanning on WeBots

Vinita Sao (2210705) - *PhD EECS*
Saswata Sarkar (19279) - *BS DSE*

Introduction

The objective of the problem is to plan a path for our point mass robot **e-puck** and help it reach the goal from the starting point. In the environment, four obstacles are shown as the yellow boxes and two other robots named **purplebots** that move to and fro, following a straight line with a velocity of 0.5 meters per second. In planning the path for the robot to reach the goal point, the collision with the obstacles and the other two robots.

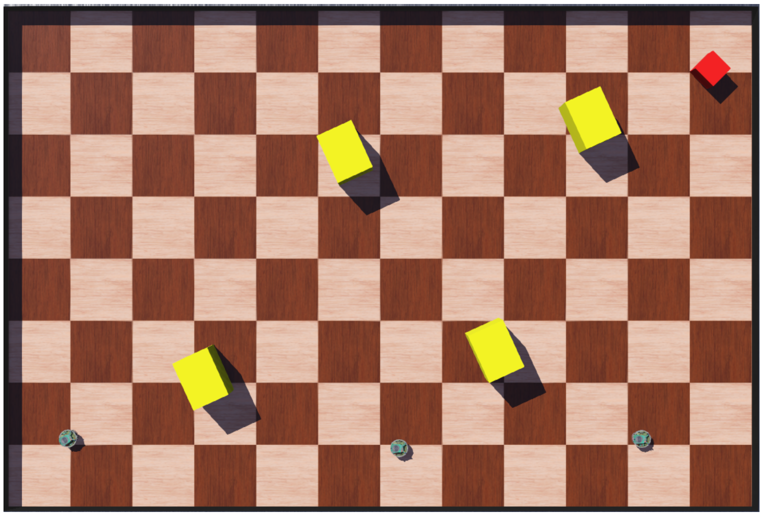
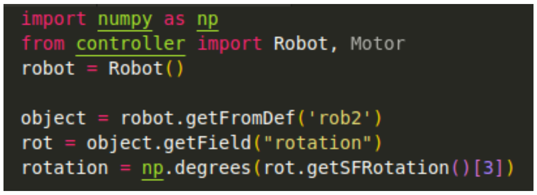


Figure 1: Simulation Environment

Steps

To move the two **purplebots** in a straight line with to and fro motion, the equation of the straight lines is first calculated, and then rotation of the **e-puck** is derived with **code**. The current location is obtained using the GPS of the



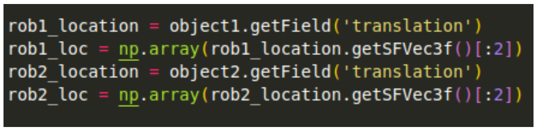
```
import numpy as np
from controller import Robot, Motor
robot = Robot()

object = robot.getFromDef('rob2')
rot = object.getField("rotation")
rotation = np.degrees(rot.getSFRotation()[3])
```

Figure 2: Code Snippet

robot, and some conditions on the heading angle to rotate the robot were set beforehand.

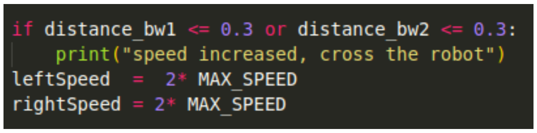
When the robot starts from the source location, it will move towards the goal, and there is a chance to collide with **purplebots**. To ignore this collision, we can increase the speed of **e-puck** two times its current speed whenever the distance between these two will be less than 3cm. Below **rob1** and **rob2** are **purplebots**. To get the location of **purplebots**, the following code is used:



```
rob1_location = object1.getField('translation')
rob1_loc = np.array(rob1_location.getSFVec3f()[:2])
rob2_location = object2.getField('translation')
rob2_loc = np.array(rob2_location.getSFVec3f()[:2])
```

Figure 3: Code Snippet

The following is the code to avoid a collision, where we used the **bug0** algorithm to reach the goal.



```
if distance_bw1 <= 0.3 or distance_bw2 <= 0.3:
    print("speed increased, cross the robot")
leftSpeed  =  2* MAX_SPEED
rightSpeed = 2* MAX_SPEED
```

Figure 4: Code Snippet

Please follow the github link to the simulation files Something Linky