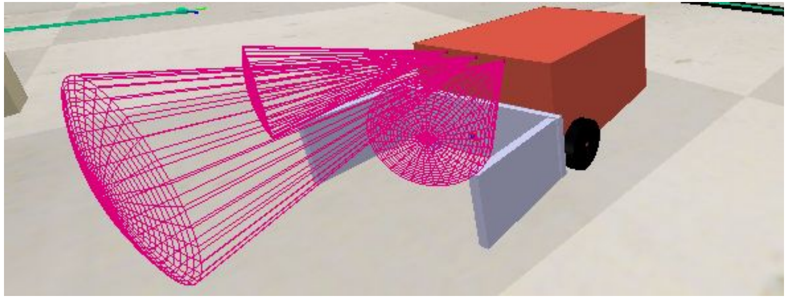
SYSC 4805 Presentation



Group 10 Project: **Fire Opal**

Chase Fridgen - 101077379

Israel Okonkwo - 101100113

Sunjeevani Pujari - 101110032

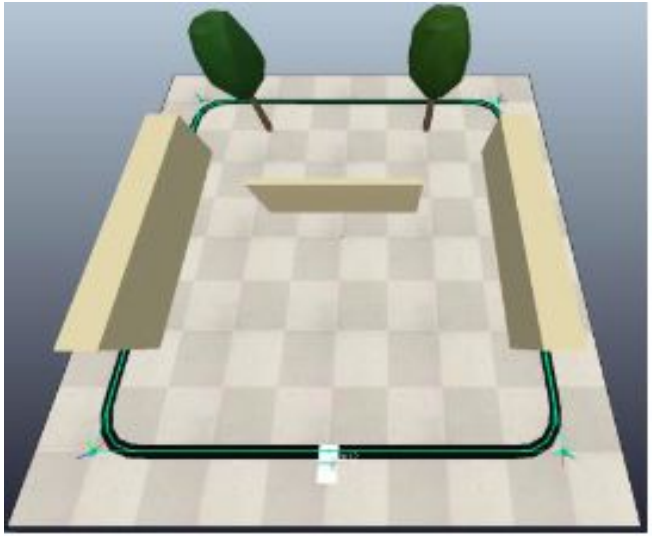
Yunas Magsi – 101115159

Content

- Objective
- Requirements
- Solution
- Implementation
- Demonstration
- Conclusion

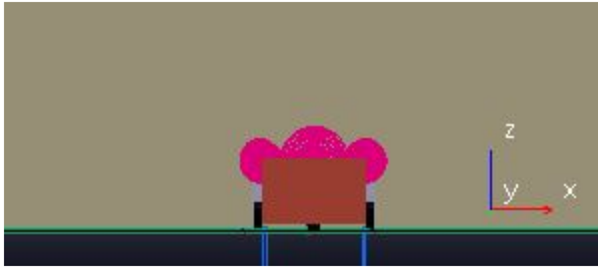
Objective

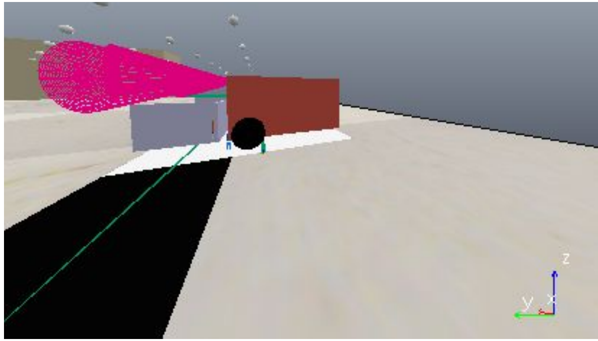
- Efficiently clear out snow off the path
- Overcome different challenges on the path



Requirements

- Use Coppeliasim
- Maximum Robot Size: 0.5 x 0.8 x 1 m
- Use Realistic Sensors
- Design a unique plow to clear out snow
- Starting Position: (0, -6.25)
- Starting Orientation: (90, 0, 90)
- Simulation time ≤ 5 min





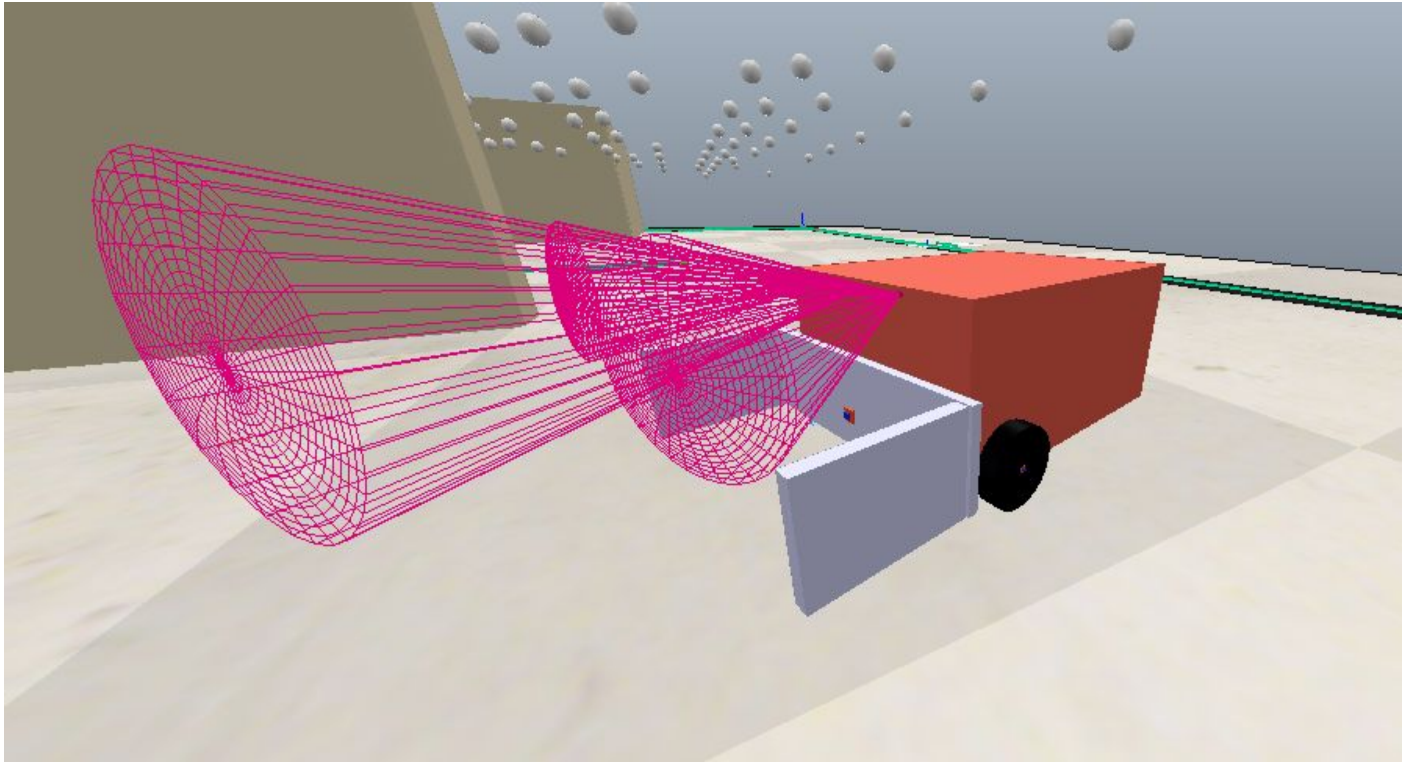
Implementation

- Build Robot Body
- Identify drive mechanism
- Identify different challenges on the path
- Pick suitable sensors
- Identify proper sensor orientation on robot body
- Design plow to work with sensors and robot body

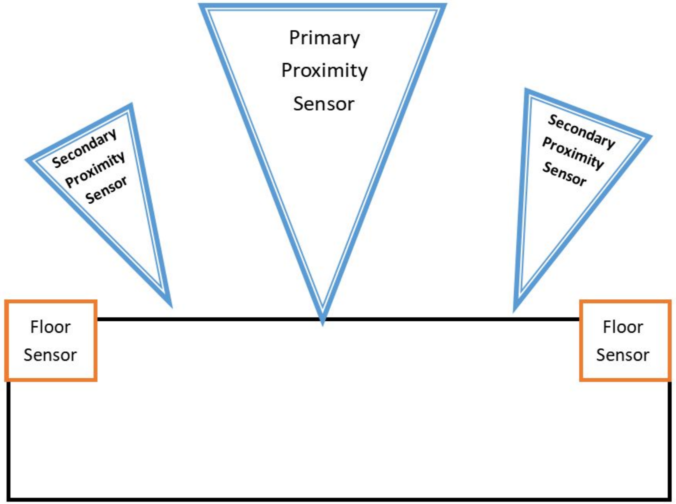
Components

- 3 Proximity Sensors(1 long range & 2 short Range)
- 2 Vision Sensors(Line Detection)
- 2 Wheels + Motors
- Castor Wheel in the rear
- Robot Frame
- Plow

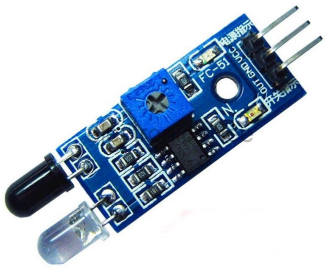
The Robot



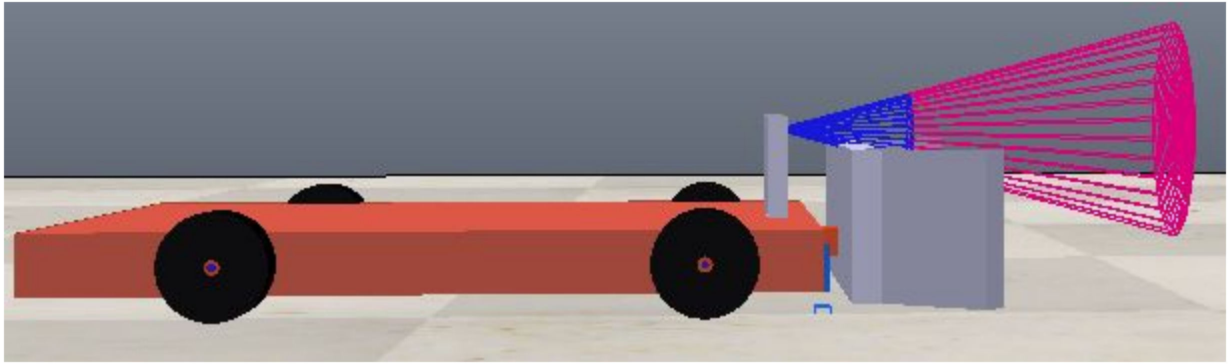
Sensors Orientation

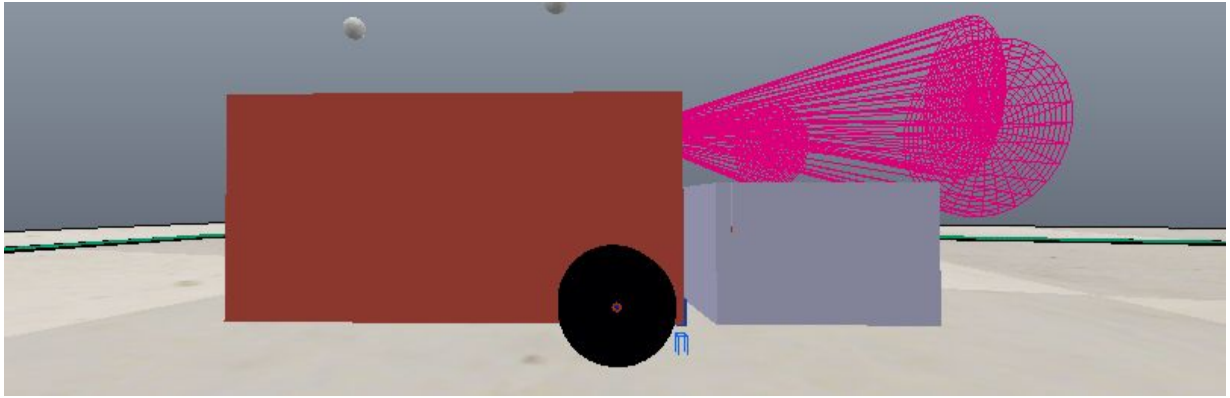


- 2 for line detection
- 3 for obstacle avoidance



Drive Configuration

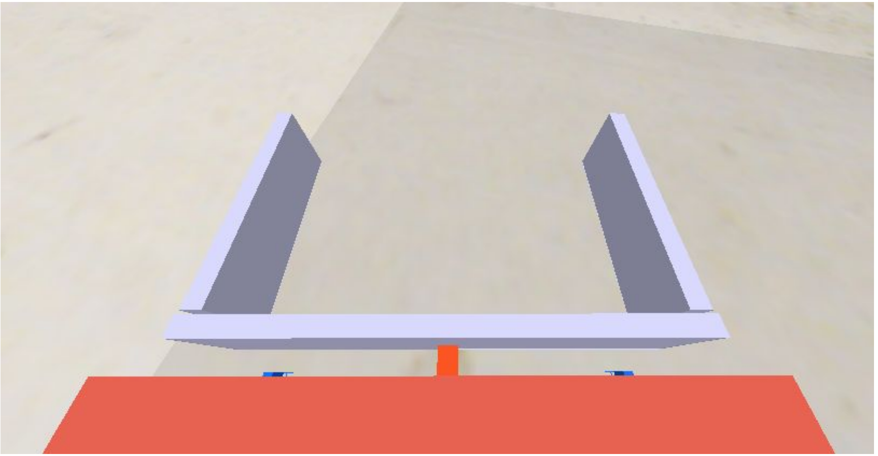




Plow Design

- Simple U Design
- + Great Capacity
- + Holds Snow

  Turning
- - Large in Size



Angular Velocity

0.0775 m = diameter

2 m/s

$C = 2\pi r = \pi d = 0.24347\,m$

rev/m = Speed in m/m / Circumfrace in meters

$2\,m/s \cdot 60s/1m = 120\,m/m$

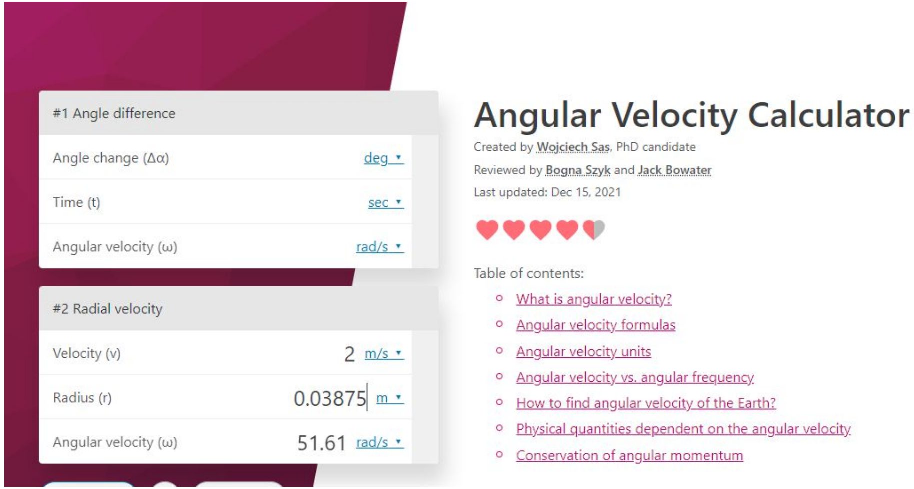
$rev/min = \dfrac{120m/min}{0.24347m}$

492.9 rev/min want in Rad/s

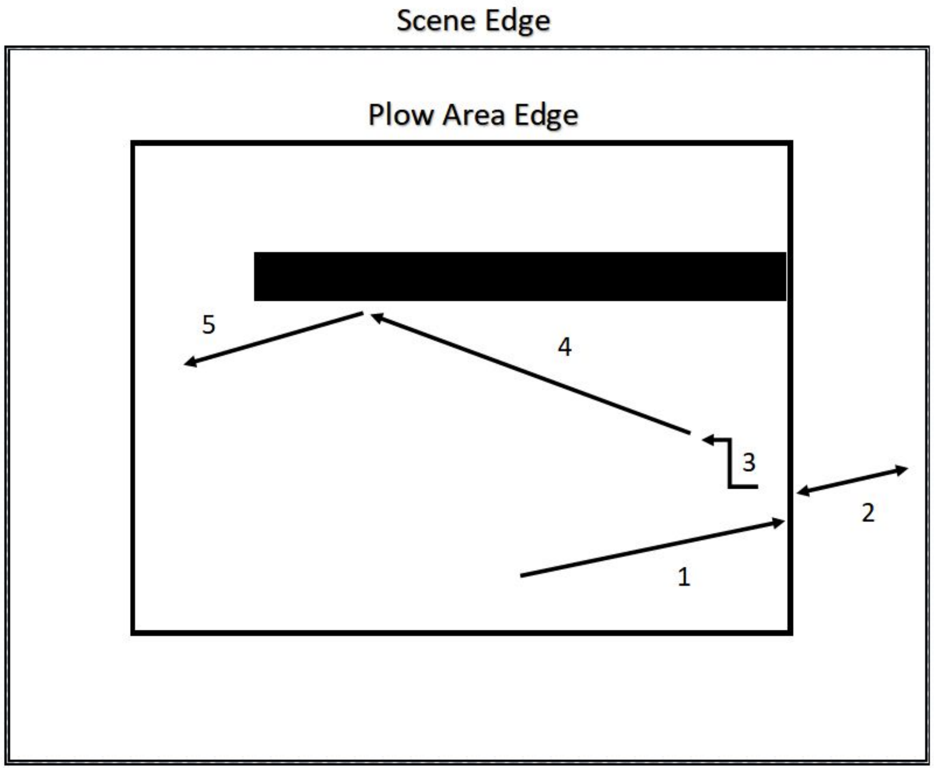
1 rev = 2π Radians

$492.9\,rev/min \cdot 1min/60s \cdot 2\pi\,Rad/1\,rev$

angular velocity = 51.6 Rad/s



Sample Algorithm



Robot Behaviour Algorithm: Sensor Hierarchy

```python
sensorArray[1] = proximdetect
sensorArray[2] = left_proximdetect
sensorArray[3] = right_proximdetect
count = 0

for i in range(len(sensorArray)):
    if sensorArray[i] == True:
        count += 1

if (count > 1):

    if (floorReading[0] == 1 or floorReading[1] == 1):
        #print("both sensors detected line")
        dumpSnow()
        #rotateRobot(turnRight)
        debounceFloorSensorCounter = 0
    else:
        doa180()

    if (floorReading[0] == 1):
        rotateRobot(turnLeft)
    elif (floorReading[1] == 1):
        rotateRobot(turnRight)
```

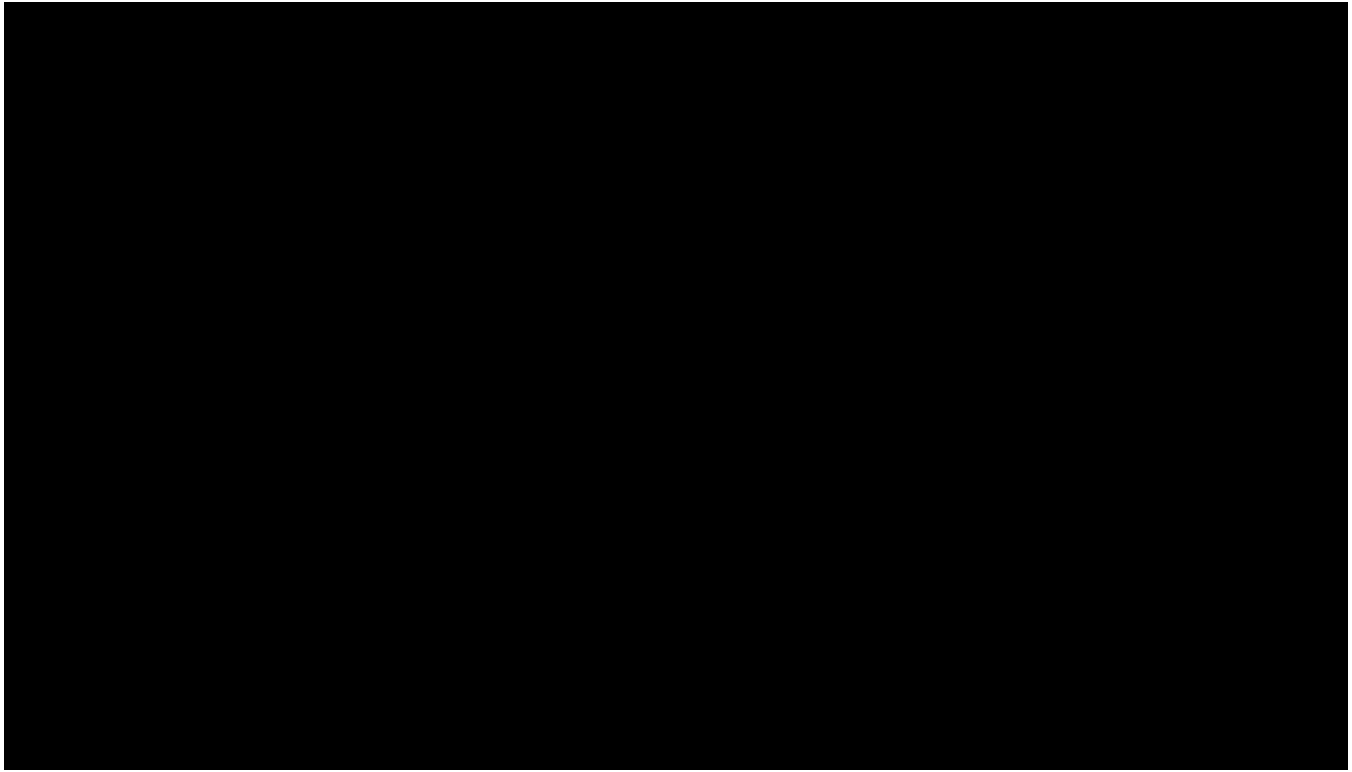
```python
else:
    if proximdetect:
        print("prox detected")
        rightSideVelocity = -velocity * adjustSpeedBy
        leftSideVelocity = velocity * adjustSpeedBy

    if left_proximdetect:
        print("left front sensor detect")
        rightSideVelocity = velocity * adjustSpeedBy
        leftSideVelocity = -velocity * adjustSpeedBy

    if right_proximdetect:
        print("right front sensor detect")
        rightSideVelocity = velocity * adjustSpeedBy
        leftSideVelocity = -velocity * adjustSpeedBy

    if (floorReading[0] == 1 or floorReading[1] == 1):
        print("both sensors detected line")
        dumpSnow()
        rotateRobot(turnRight)
```

Robot Behaviour Algorithm: How we tell where the Robot is

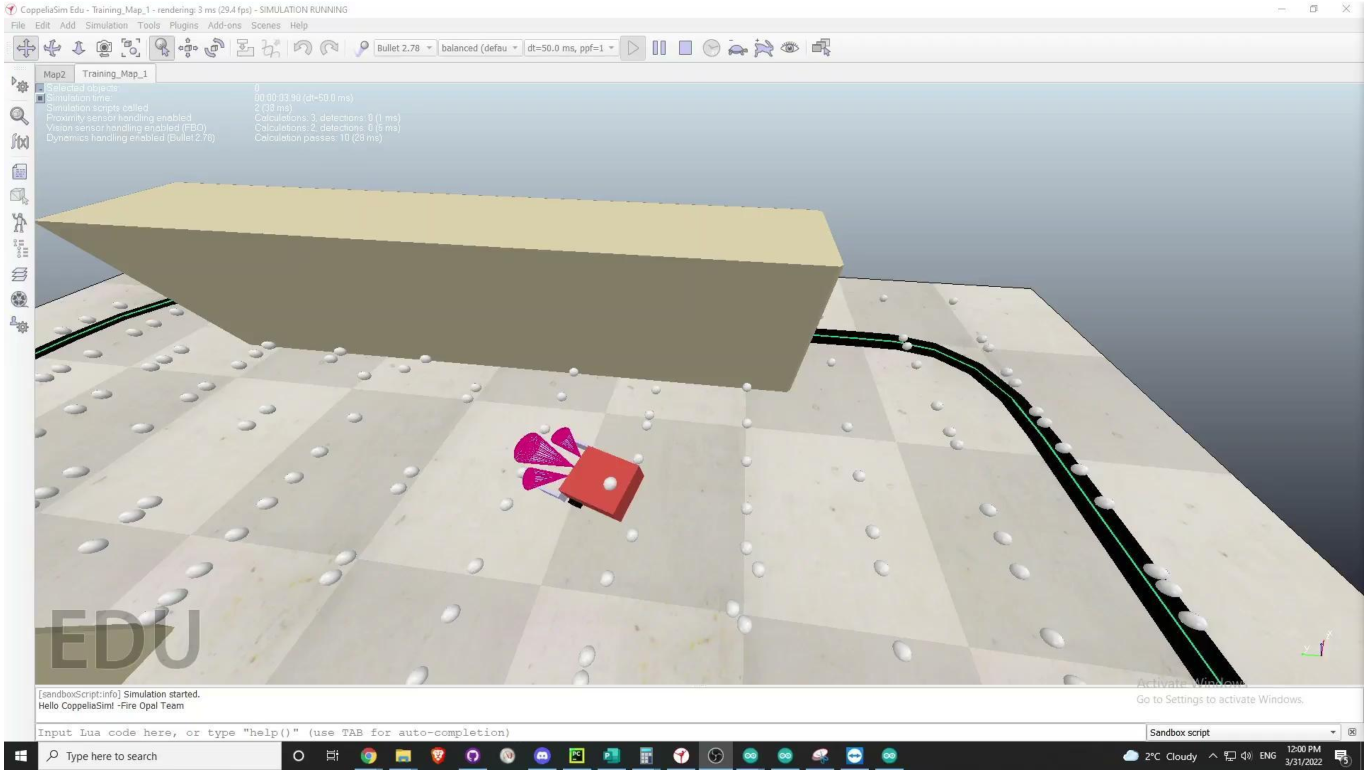
```
#Orientation of Robot Detect
RC, eulerAngles=sim.simxGetObjectOrientation(clientID,RobotBody, sim.sim_handle_parent, sim.simx_opmode_blocking)
```

We use a function call that gets the orientation of the robot so we know what part of the world the robot is facing and use that info to adjust how the robot turns so it doesnt stay in the same direction for too long to maximize snow collection.

Demonstration of Floor Sensor



Demonstration of Proximity Sensor



Project Cost Planning

Total Cost of Project

- Tires: $188 (2 front wheels and 1 castor wheel)
- Main frame plus plow: $300 (Frame is 3d-printed)
- Motors: $35 (2 motors, 1 castor joint)
- Sensor Cost: $160.27 (2 vision, 3 proximity)
- Engineer Cost: $1792 (4 Underpaid Engineers, 16$/hour, 4 hours per week, 7 weeks total)
- Total Cost: $683.27 + $1792 = $2 475.27

- The AC is $683.27
- The BAC is $2 475.27 and as far as EAC, no activity has been skipped
- EV > AC we are under budget
- EV > PV which shows we are on time
- There was extra cost for switching from LUA to python.

Conclusion

- Built a functional Robot that clears out snow off a path while overcoming challenges
- The current robot design will be the main and permanent design
- Our total (BAC) = $2 475.27 and so far we are on time and under budget
- Python is being used to program the robot in CoppeliaSim
- Satisfied both functional and non-functional requirements of the project

Thanks for listening and for your time

Are there an questions?