**C951**

**WGU**

**Introduction to Artificial Intelligence**

**Task 2**

**Daniel Roberts**

**Student ID: 009198202**

1. **Environment and Obstacles**

* A tornado has hit a small town in middle Tennessee. My disaster recovery environment is a room inside a building and part of the room has been blocked by debris. This prevents first responders from entering the room and locating any people inside. The room has a total of five obstacles including two bookcases, a table, and two pillars.

1. **Goal**

* Our robot’s mission is to navigate through the debris and to locate any people remaining inside. Since first responders cannot readily make it through the debris safely, deploying our robot will allow first responders to determine if anyone remains inside, determine the extent of debris inside the room, and help plan a recovery mission from the information gained.

1. **Modifications**

* Our robot has front and rear-facing proximity sensors and a front-facing visual sensor. The front of our robot has 2 proximity sensors, one to assist the robot in movement and the other to detect people inside. We added a rear proximity sensor to keep the robot from backing into objects and falling over. Both the detection and visual sensor will help first responders to better understand the environment inside of the room. This will help during the recovery mission. Finally, we added a front-facing spotlight to help with lowlight conditions.

1. **Internal representation of the environment.**

Our robot uses its detection sensors to maintain an internal representation of the environment of the office. When our sensor detects a debris object, it turns accordingly until the sensor does not detect any objects and then proceeds with its’ navigation through the room. Our robot has a detection sensor that allows it to recognize people that may be trapped inside the room. Our robot recognizes the difference between debris objects and a person. It then flashes its detection sensor light from blue to green and prints a notification that a person has been detected. We have also added a graph to our robot that allows the robot to read the distance between it and nearby objects. This can be used to collect and plot data about the environment inside the room.

1. **Four concepts to achieve its goal:**

* **reasoning**

Our robot utilizes reasoning by changing direction whenever a piece of debris is detected. This ensures our robot does not get stuck on debris or in one location. The code also allows our robot to adjust its speed when turning and changing the direction of travel. When our robot detects a person, it uses reasoning to understand that it needs to change its detection sensor color and print a notification that a person has been located.

* **knowledge representation:**

Our robot utilizes sensors, both proximity and visual, to gather knowledge about the environment inside the room. The visual representation will allow first responders to map the room prior to conducting a recovery mission for the survivors inside. Our robot utilizes knowledge from two proximity sensors to maneuver throughout the room and another to detect survivors inside the room.

* **uncertainty**

During any disaster, there can be unknown factors prior to deploying our robot. One of those factors may be lowlight conditions. Our robot utilizes a front-facing spotlight to help combat that potential issue. Lowlight conditions could tamper visual feeds, causing first responders to be unable to properly map the area before a rescue mission. There is a chance someone may be injured inside the room and hidden in small areas or behind objects. We ensured that our detection sensor exceeded the distance covered when compared to our maneuvering sensor. This helps our robot detect people prior to the maneuvering sensor detecting the debris and causing our robot to change directions.

* **intelligence**

Our robot utilizes information gathered through the sensors to adjust its direction. This keeps our robot from getting stuck in one location and allows it to completely cover the entire area. Whenever our robot’s detection sensor detects a person, it will print a notification that a person has been found and the detection sensor will flash green. Finally, our robot uses a graph to detect the distance between itself and debris objects. This will allow first responders to gather data about the inside of the room. It will particularly help them determine how tight spaces may be between debris objects.

**F. Improvements**

* Many improvements can still be made to our robot. We would want to add a location feature almost immediately. We could utilize a sensor to track our robot’s location inside the environment. We could utilize a machine learning algorithm such as SLAM to achieve that goal. We could use that feature to give precise locations of people or objects inside. We could add a graph that can map the environment inside the room. That can be utilized to ensure our robot doesn’t continuously search the same area repeatedly. Seconds are crucial in a disaster situation. Ensuring our robot conducts its mission in the most efficient way possible is crucial to saving lives. We could apply a reinforced learning technique by using a reward system. This would help motivate our robot to perform in a more efficient and desirable way. We could utilize a positive and negative reward system by giving our robot a positive reward as it moves closer to our target and a negative reward if it moves further away from the target. We would then need to train our robot by giving it feedback on the decision that was made in a test case. Using this approach would help to reduce the time wasted during a search.

1. **Submit the robot code that you created.**

function sysCall\_init()

num\_To\_Detect = 0

noseSensor\_To\_Detect=sim.getObject("./sensingNose\_To\_Detect")

bubbleRobBase=sim.getObject('.')

leftMotor=sim.getObject("./leftMotor")

rightMotor=sim.getObject("./rightMotor")

noseSensor=sim.getObject("./sensingNose")

minMaxSpeed={50\*math.pi/180,300\*math.pi/180}

backUntilTime=-1

robotCollection=sim.createCollection(0)

sim.addItemToCollection(robotCollection,sim.handle\_tree,bubbleRobBase,0)

distanceSegment=sim.addDrawingObject(sim.drawing\_lines,4,0,-1,1,{0,1,0})

robotTrace=sim.addDrawingObject(sim.drawing\_linestrip+sim.drawing\_cyclic,2,0,-1,200,{1,1,0},nil,nil,{1,1,0})

graph=sim.getObject('./graph')

distStream=sim.addGraphStream(graph,'bubbleRob clearance','m',0,{1,0,0})

-- Create the custom UI:

xml = '<ui title="'..sim.getObjectAlias(bubbleRobBase,1)..' speed" closeable="false" resizeable="false" activate="false">'..[[

<hslider minimum="0" maximum="100" on-change="speedChange\_callback" id="1"/>

<label text="" style="\* {margin-left: 300px;}"/>

</ui>

]]

ui=simUI.create(xml)

speed=(minMaxSpeed[1]+minMaxSpeed[2])\*0.5

simUI.setSliderValue(ui,1,100\*(speed-minMaxSpeed[1])/(minMaxSpeed[2]-minMaxSpeed[1]))

end

function sysCall\_sensing()

local result,distData=sim.checkDistance(robotCollection,sim.handle\_all)

if result>0 then

sim.addDrawingObjectItem(distanceSegment,nil)

sim.addDrawingObjectItem(distanceSegment,distData)

sim.setGraphStreamValue(graph,distStream,distData[7])

end

local p=sim.getObjectPosition(bubbleRobBase,-1)

sim.addDrawingObjectItem(robotTrace,p)

end

function speedChange\_callback(ui,id,newVal)

speed=minMaxSpeed[1]+(minMaxSpeed[2]-minMaxSpeed[1])\*newVal/100

end

function sysCall\_actuation()

result=sim.readProximitySensor(noseSensor)

if (result>0)

then backUntilTime=sim.getSimulationTime()+3 end

if (backUntilTime<sim.getSimulationTime()) then

sim.setJointTargetVelocity(leftMotor,speed)

sim.setJointTargetVelocity(rightMotor,speed)

else

sim.setJointTargetVelocity(leftMotor,-speed/2)

sim.setJointTargetVelocity(rightMotor,-speed/8)

end

local result\_To\_Detect,distance,detectedPoint,detectedObjectHandle=sim.readProximitySensor(noseSensor\_To\_Detect)

if (result\_To\_Detect>0) then

if detectedObjectHandle then

if sim.getObjectAlias(detectedObjectHandle) == 'Person\_To\_Detect' then

num\_To\_Detect = num\_To\_Detect + 1

print("Person detected!")

-- simInt simGetObjectColor(simInt objectHandle,simInt index,simInt colorComponent,simFloat\* rgbData)

-- Colors: Red: {1,0,0}, Green = {0,1,0}, Blue = {0,0,1}

sim.setObjectColor(noseSensor\_To\_Detect,0,sim.colorcomponent\_ambient\_diffuse,{0,1,0}) -- Green = {0,1,0}

end

end

else

sim.setObjectColor(noseSensor\_To\_Detect,0,sim.colorcomponent\_ambient\_diffuse,{0,0,1}) -- Blue = {0,0,1}

end

end

function sysCall\_cleanup()

simUI.destroy(ui)

end

1. **Panopto video recording:**

**A statement about disaster recovery problem.**

* A tornado has hit a small town in middle Tennessee. There are an unknown number of people inside an office building.

**A summary of environment and obstacles.**

* My disaster recovery environment is a room inside a building and part of the room has been blocked by debris. This prevents first responders from entering the room and locating any people inside. The room has a total of five obstacles all of which are a variety of shapes and sizes.

**A summary of goal and objectives.**

* Our goal is for our robot to locate people remaining inside of the room and trapped by debris. Our robot’s objectives include navigating the debris and searching the entire area of the room.

**A description of robot and its architecture.**

* Our robot has front and rear-facing proximity sensors and a front-facing visual sensor. The front of our robot has 2 proximity sensors, one to assist the robot in movement and the other to detect people inside. We added a rear proximity sensor to keep the robot from backing into objects and falling over. Both the detection and visual sensor will help first responders to better understand the environment inside of the room. This will help during the recovery mission. Finally, we added a front-facing spotlight to help with lowlight conditions.

**A demonstration.**

* See attached video file.

**Robot’s capabilities.**

* Overall, I believe our robot is capable of efficiently completing its objectives and reaching its goal.

**Improve the prototype.**

* Many improvements can still be made to our robot. We would want to add a location feature almost immediately. We could utilize a GPS sensor to track our robot’s location inside the environment. We could utilize a machine learning algorithm such as SLAM to achieve that goal. We could use that feature to give precise locations of people or objects inside. We could add a graph that can map the environment inside the room. That can be utilized to ensure our robot doesn’t continuously search the same area repeatedly.