```
mirror object to mirror
mirror_mod.mirror_object
peration == "MIRROR_X":
irror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
 _operation == "MIRROR_Y"
Irror_mod.use_x = False
 !rror_mod.use_y = True
 lrror_mod.use_z = False
  operation == "MIRROR_Z";
  rror_mod.use_x = False
  rror_mod.use_y = False
  rror_mod.use_z = True
 melection at the end -add
   ob.select= 1
   er ob.select=1
   ntext.scene.objects.action
  "Selected" + str(modification
   rror ob.select = 0
  bpy.context.selected_obj
  lata.objects[one.name].sel
  Int("please select exaction
  OPERATOR CLASSES ----
    vpes.Operator):
    X mirror to the selecte
   ject.mirror_mirror_x"
  ext.active_object is not
```

Advanced
Software
Development for
Autonomous
Mobile Robots

Final Project - Presentation



### Team Name

Potterheads

## Members

Adithya Sabarish Saravanan (32979) Arulious Jora Antony Raj Sathish (33048)

## Turtlebot

Harry\_potter

## Introduction



### **Problem Statement**

To navigate Harry\_potter to a given goal point in a predefined environment collecting the rewards while avoiding obstacles at the same time.



#### Resources

Move base navigation stack as the server.

Goal publisher.

Obstacle avoidance.

A Navigation client to control Harry\_potter.

## Algorithm – Client node

Receive

 Receive the goal from goal publisher; Sort goals based on rewards; Choose the first goal.

Send

• Send the goal to the move base server; Monitor the state of that goal.

Repeat

• If the goal is reached, repeat step1; Continue until all goals are reached.

# Special cases – Case 1:

After 15 seconds, if Harry\_potter stops or if move base responds that it cannot reach the goal

Cancel the goal for move base.

Navigate Harry to the 10th way point (0.25m) on the path planned by move base

• The planned path is obtained from move base as waypoints using the ROS service Getplan().

Resend the goal to move base.

# Special cases- Case 2:

For goals with very high rewards I.e., if the path to reach the goal is very narrow

Cancel the goal for move base.

1

Get the plan from move base

2

Using the plan and obstacle avoidance algorithm, make Harry\_potter follow the waypoints to reach the goal.

# Special cases - Case 3:

If there is a timeout of 120s

Abort the goal; Move Harry\_potter to the origin using move base.

The next goal is updated to move base.

The aborted goals are stored and reattempted at the end.

1

2

## Obstacle avoidance

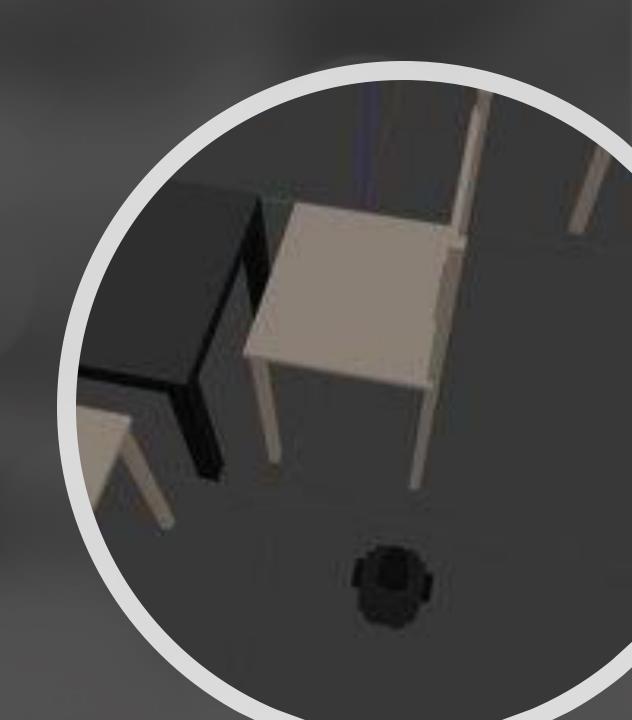
#### Turn

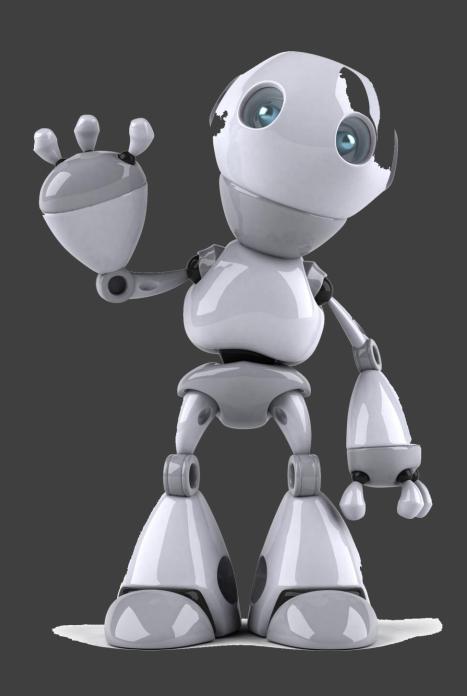
- If an obstacle is observed in the front 36 degree region
  - turn Harry\_potter by 36 degrees to the left or to the right; whichever side has no obstacle in the 36 degree span.
  - Rotate Harry\_potter until there is no obstacle in the front

1

#### Move

- Move one meter in the same direction as long as there are no obstacles in its path.
- No obstacle => Free space > 0.2 m





# Thank You