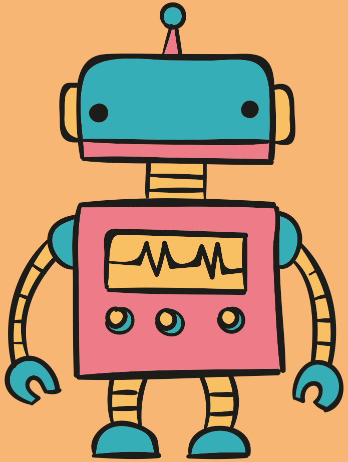
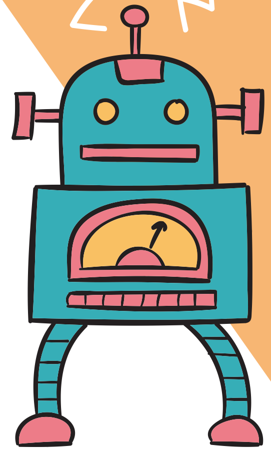
C.A.N.D.I FINAL PROJECT



By: Dawei Wang, Meet Brahmbhatt, Michael Nunes, Shourya Vuddemarri

Table of Contents



Problem Statement

Develop a virtual autonomous rover capable of delivering candy from home state to boost productivity within workspace.

C.A.N.D.I

Compact

Autonomous

Navigating

Delivery

Intelligence

Objective

Level 1

Design an autonomous rover and develop a virtual rover.

Level 2

Successfully incorporate driving and attempt to reach end location

Level 3

Enhance rovers ability to get from home state to end location more efficiently!

Current Progress

Level 1

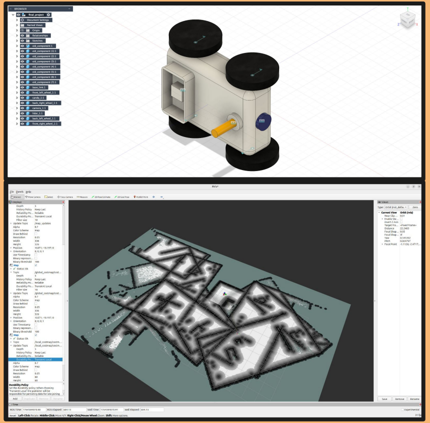
- Rover model is created from CAD.
- Rover is fully simulated in Gazebo.

Level 2

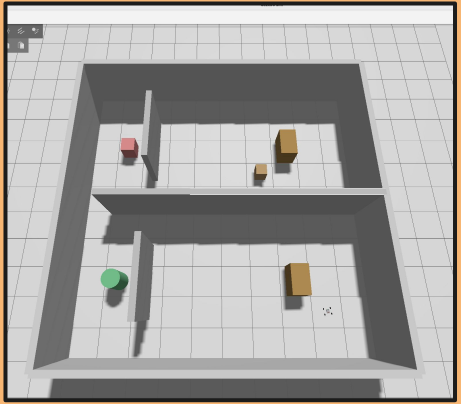
- Mapping stack (SLAM) is operational.
- Sensor fusion (LiDAR scans + Camera) is integrated

Level 3

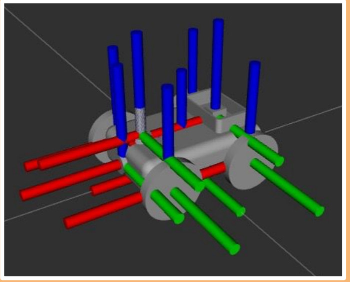
- Optimization



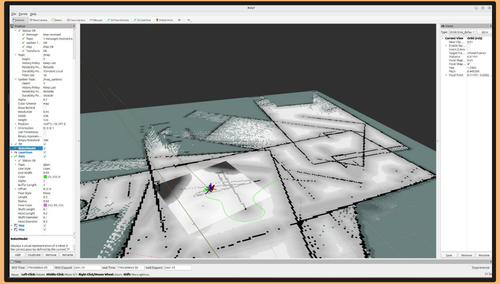
Model & SLAM Mapping [1]



Gazebo Display [2]

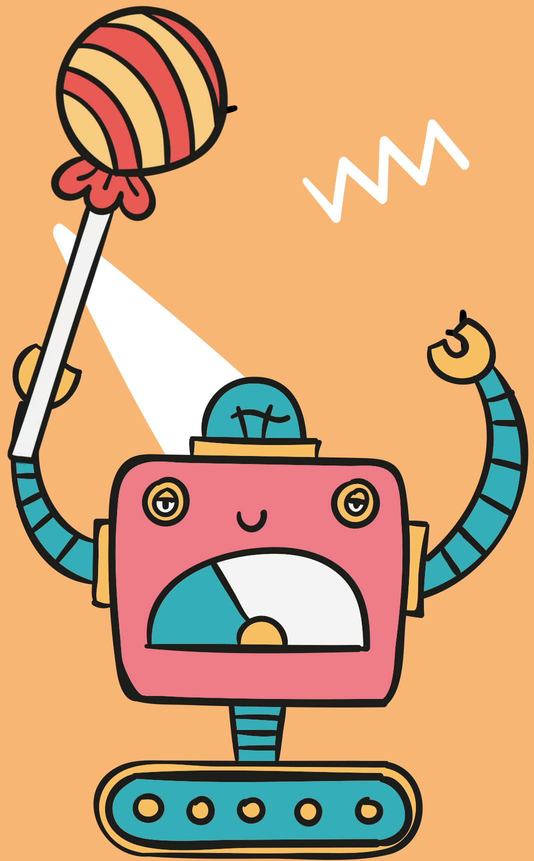


RViz [3]

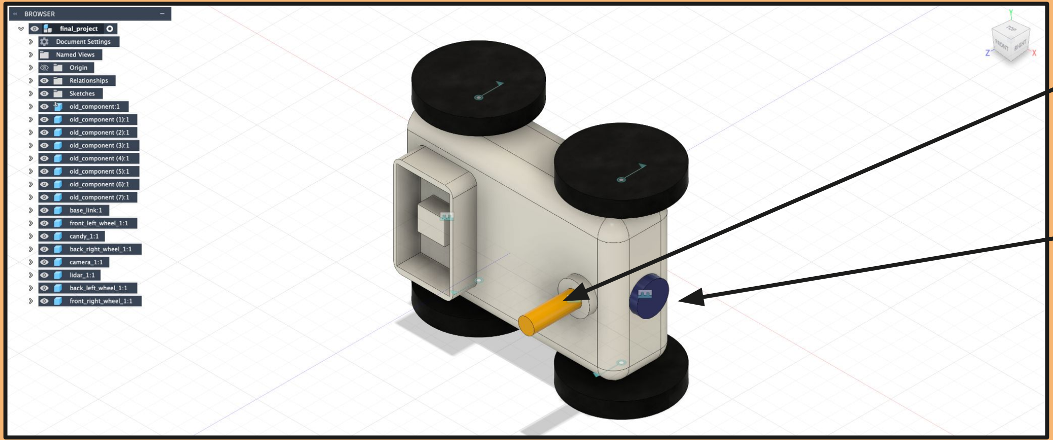


Navigation [4]

Approach & Results



Hardware



C.A.N.D.I Fusion 360 Model [5]

LiDAR = Used to develop 2D map of our virtual environment for navigation.

Camera = "Eyes" of rover. Will serve to mark obstacles and navigate our workspace.

Additional Components: 4x Wheels, Baselink, Candy

Conversion: Fusion360 → URDF ROS2

- **.xacro** = Stores macros and variables for rover description
- **.trans** = Stores coordinate frame transformation
- **.gazebo** = Stores simulation elements (physics, dynamics, sensor models, actuation configs)

```
<joint name="Revolute 15" type="continuous">
  <origin xyz="-0.045 -0.05 0.015" rpy="0 0 0"/>
  <parent link="base_link"/>
  <child link="right_back_wheel_1"/>
  <axis xyz="-0.0 -1.0 0.0"/>
</joint>
```

.xacro snipit [6]

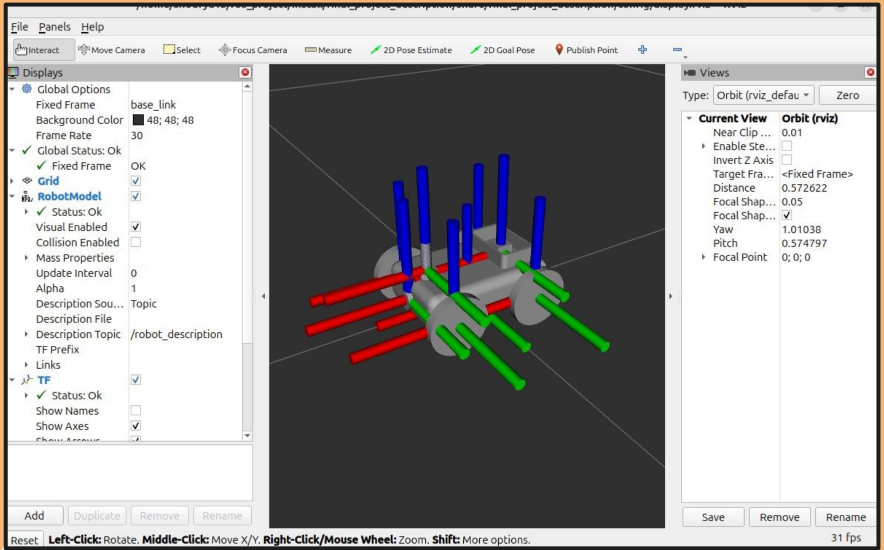
```
<gazebo reference="base_link">
  <material>${body_color}</material>
  <mu1>0.2</mu1>
  <mu2>0.2</mu2>
  <self_collide>true</self_collide>
  <gravity>true</gravity>
</gazebo>
```

.gazebo snipit [7]

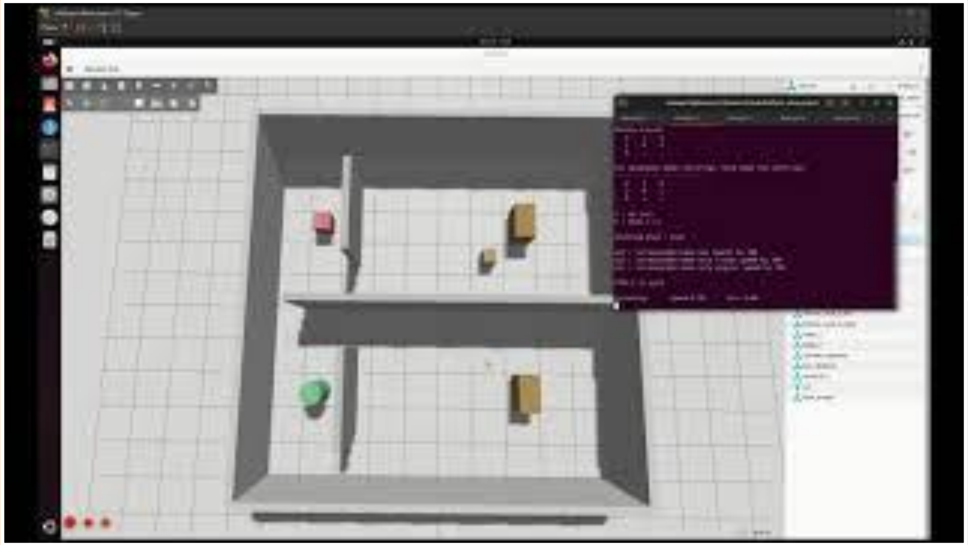
RVIZ and Gazebo Display

RViz = provides real-time visualization of all sensor data, the robot state, and navigation planning

Gazebo Display = 3D Visualization from URDF Model and World file



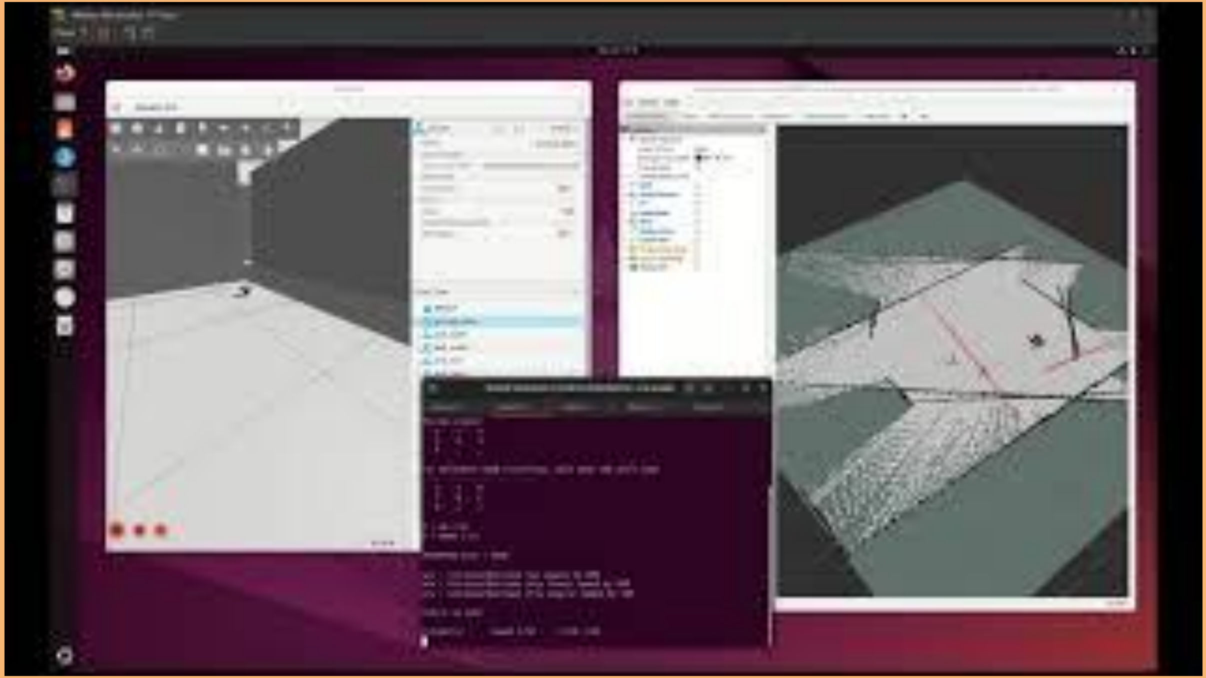
C.A.N.D.I RViz Model [8]



C.A.N.D.I Gazebo Display & Robot Control Test [9]

Simultaneous Localization and Mapping (SLAM)

SLAM = performs real-time mapping using LiDAR data while simultaneously estimating the robot's pose and updating a 2D occupancy grid.



C.A.N.D.I SLAM Model [10]

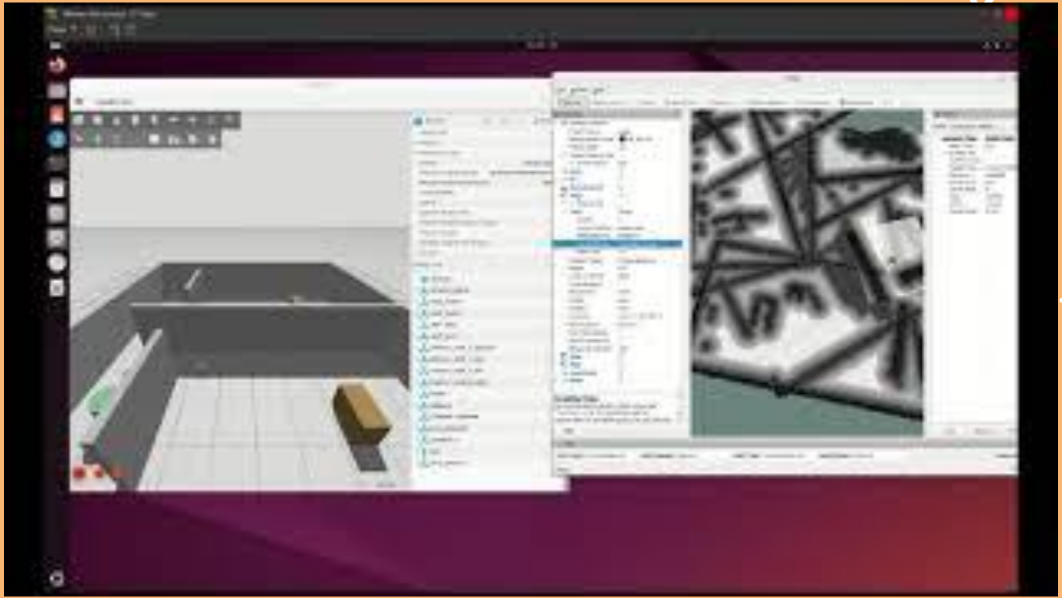
Navigation

Navigation2 stack = handles autonomous navigation from global planning to local obstacle avoidance.

Global Planning = generates a long-range path from the robot's current position to the goal position using the global map.

Obstacle Avoidance = Achieved through a Dynamic Window Approach

Costmap Generation = Static layer (SLAM), Obstacle layer (LiDAR), and inflation layer (safety radius)



C.A.N.D.I Navigation Framework [11]

Next Steps

1. Cleaned up 3D environment that resembles more a desk
2. Enhance autonomous navigation
3. Solve camera errors
4. Smoother turning and responsiveness

How will we get there?

1. Automatic Addison GitHub
   1.1. Similar project that will help guide us in further navigation development
2. Update /worlds 'sdf' file
   2.1. Update objects that are typical on a desk
   2.2. Better desk representation
3. Create PID gains for rover controller configuration to improve mobility

Our Learning Curve

1. Developing an autonomous rover
   1.1. How do we convert a CAD file to a readable file for ROS2?
   1.2. How do we create a workspace for the rover to operate within?
2. Controlling the rover
   2.1. How do we teleop the rover to listen to user commands?
   2.2. How do we ensure the rover is understanding and adapting to its environment?

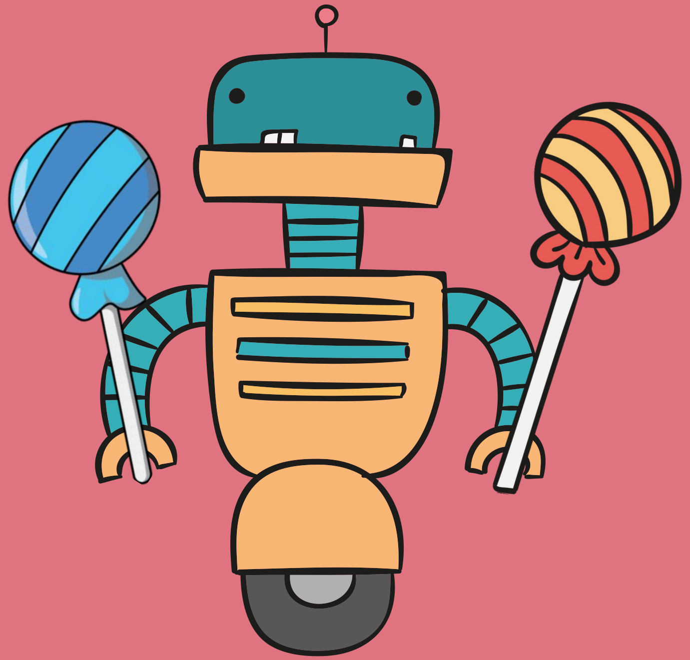
Our Approach

1. GitHub public repositories
   1.1. Understanding how to convert Fusion360 → URDF
2. Youtube Videos
   2.1. Joint Configuration for URDF
   2.2. Implementing Robot Navigation for ROS2
   2.3. ROS2 Navigation Tuning Guide
3. Claude.ai
   3.1. Code troubleshooting

Advice

- Time Management
- Understand the concepts first, then begin creating your rover
- Use the resources around you
    - Similar projects online through GitHub, Youtube, etc
    - AI models to clean up code, as well as explain concepts

Conclusion



- Built C.A.N.D.I from CAD to simulation
- Enabled mapping + SLAM vision
- Added autonomous navigation
- Refined with URDF, Gazebo, RViz
- Next: better vision, smoother motion

Thank you!

Work Cited

[1] “ROS 2 Navigation Tuning Guide (Nav2),” AutomaticAddison. Accessed: Nov. 30, 2025. [Online]. Available: https://automaticaddison.com/ros-2-navigation-tuning-guide-nav2/

[2] “Claude,” Claude.ai. Accessed: Nov. 30, 2025. [Online]. Available: https://claude.ai/

[3] “yahboom_rosmaster,” GitHub repository. Accessed: Nov. 30, 2025. [Online]. Available: https://github.com/automaticaddison/yahboom_rosmaster/tree/main

[4] “fusion360-urdf-ros2,” GitHub repository. Accessed: Nov. 30, 2025. [Online]. Available: https://github.com/runtimerobotics/fusion360-urdf-ros2