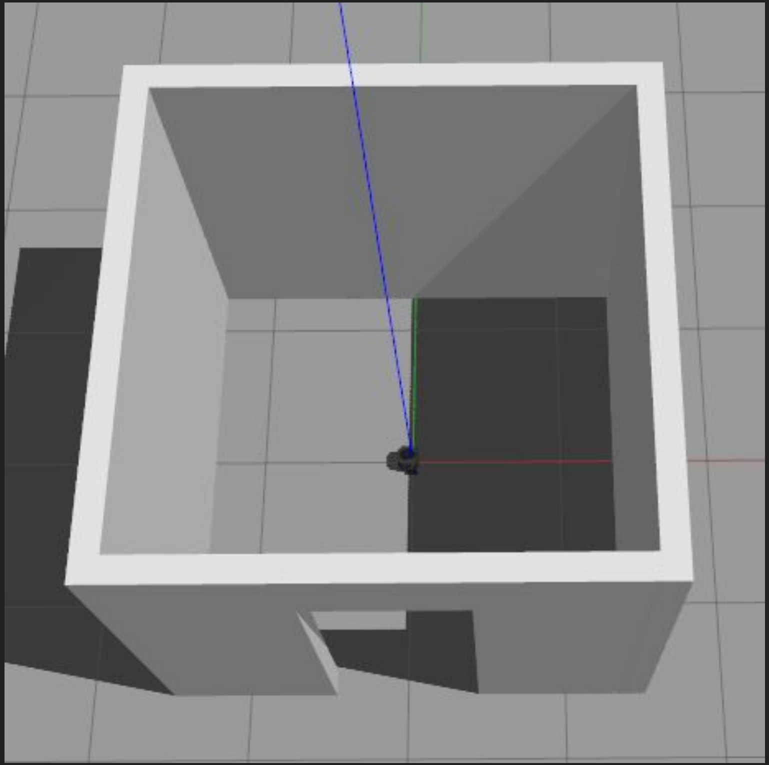
# Escape Room

Yashasvi Sriram, Levi Mathwig, Yaseen Khan

# What?

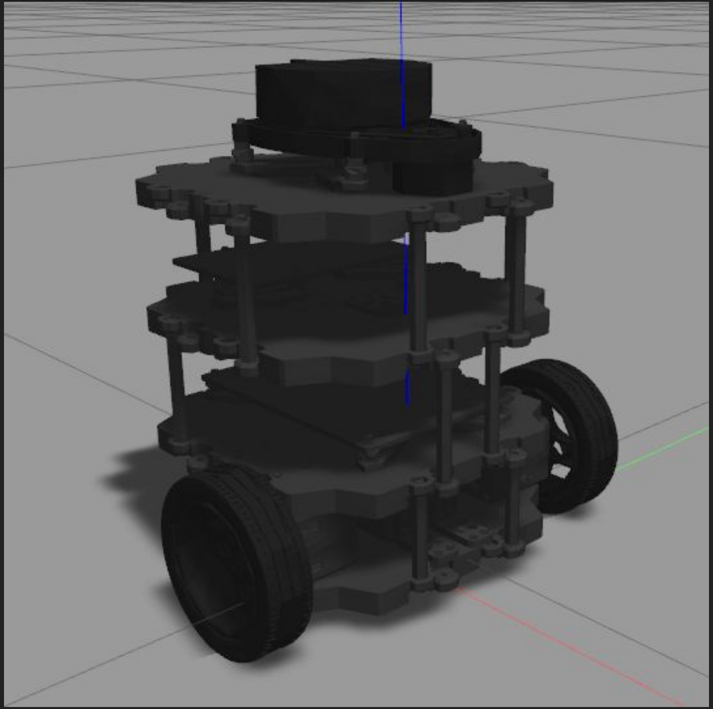
Given a robot placed in a room the task is to

1. Escape the room
2. Sense and avoid hitting **static** obstacles (viz. walls)
3. Sense and avoid hitting **dynamic** obstacles (viz. humans)



# Why?

1. Has sensing, planning and acting components.
2. Building blocks for building more sophisticated robots.
3. Sort of like a “basis set” for the “robot problem space”. :)



# CSci 5551 relevance

1. Deals with the basic math behind robots in the aspects of sensing, planning and acting, which is the essence of CSci 5551
2. Uses ROS

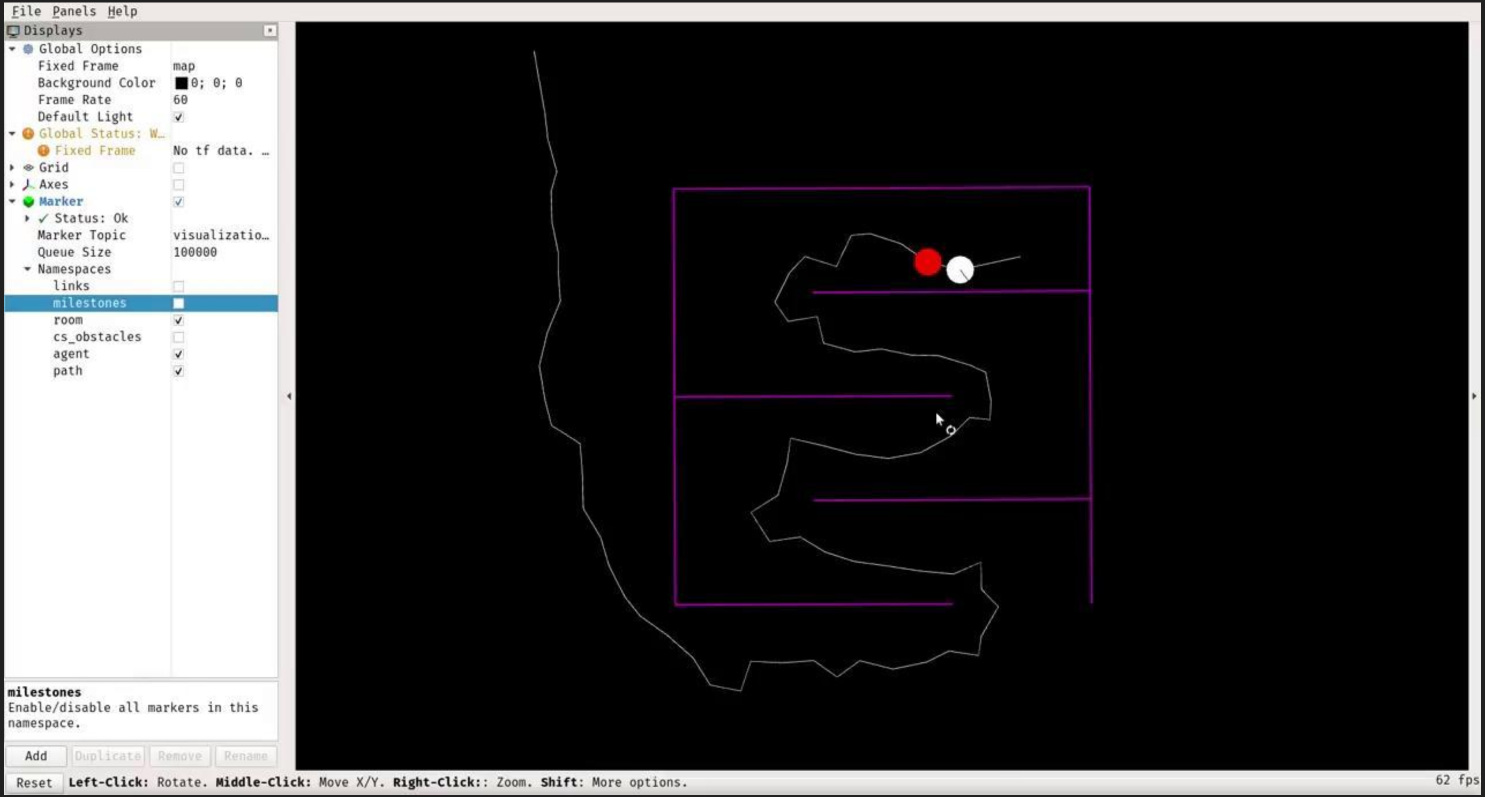
# Roadmap

- Assume static obstacles are known and no dynamic obstacles and plan for this
  - Build configuration space with obstacles
  - Build a Probabilistic Roadmap
  - Use A* like search to find a path from start to finish positions
  - Move the Turtlebot along the path
- Get static obstacle information from sensors
  - Use laser scan for detecting obstacles
  - Keep track of them
  - Plan using currently known obstacles, replan if new ones found on path (spirit of D*Lite)
- Have dynamic obstacles
  - Assume friendly behaviour
  - Use repulsive force methods to avoid collision from dynamic obstacles

# Progress

- Assume static obstacles are known and no dynamic obstacles and plan for this
  - Build configuration space with obstacles
  - Build a Probabilistic Roadmap
  - Use A* like search to find a path from start to finish positions
  - Move the Turtlebot along the path
- Get static obstacle information from sensors
  - Use laser scan for detecting obstacles
  - Keep track of them
  - Plan using currently known obstacles, replan if new ones found on path (spirit of D*Lite)
- Have dynamic obstacles
  - Assume friendly behaviour
  - Use repulsive force methods to avoid collision from dynamic obstacles

# Results: Path generation for known obstacles



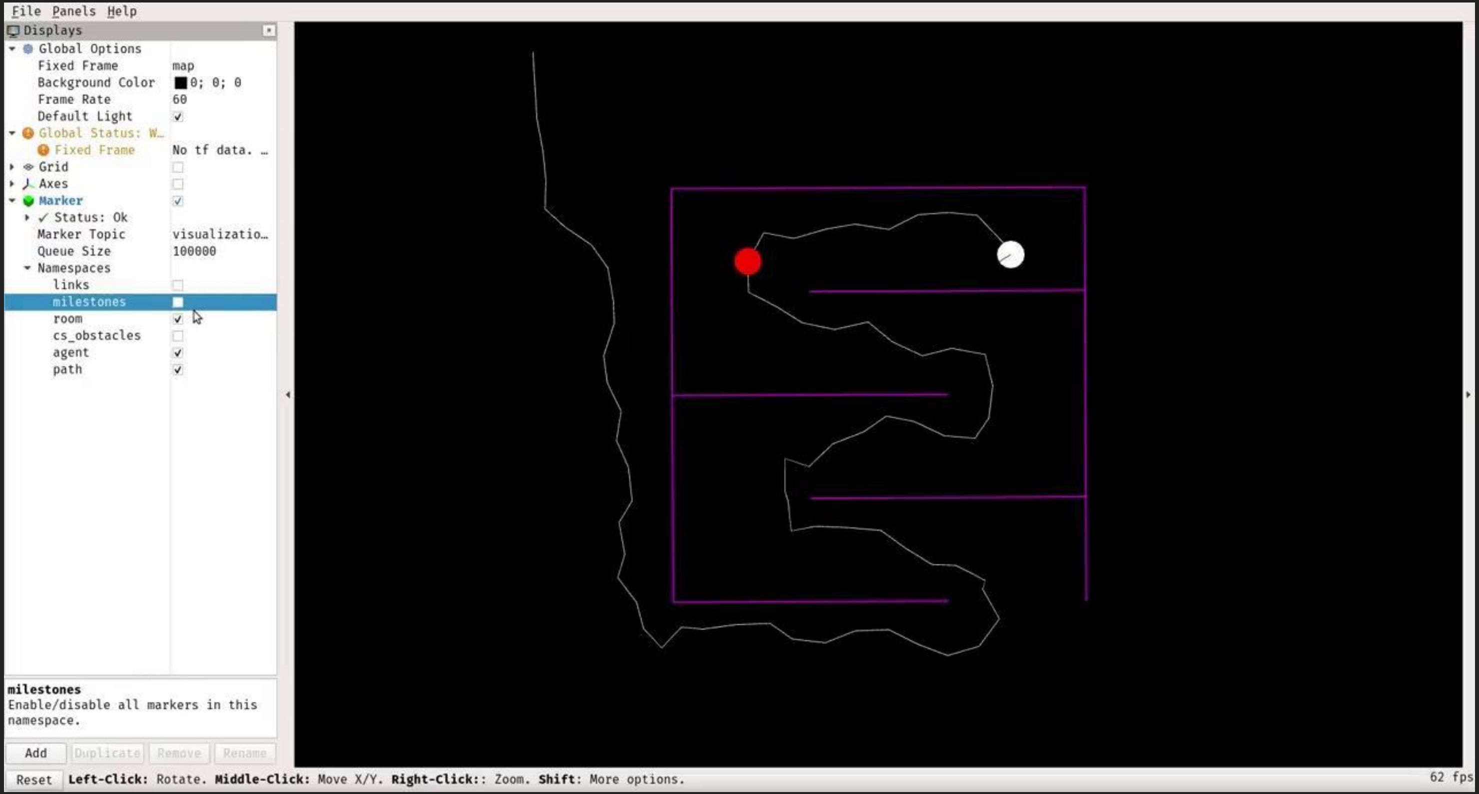
Global Options
Fixed Frame map
Background Color ■ 0; 0; 0
Frame Rate 60
Default Light ☑
Global Status: W...
Fixed Frame No tf data. ...
Grid □
Axes □
Marker ☑
✓ Status: Ok
Marker Topic visualizatio...
Queue Size 100000
Namespaces
links □
milestones □
room ☑
cs_obstacles □
agent ☑
path ☑

**milestones**
Enable/disable all markers in this namespace.

**Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click:**: Zoom. **Shift:** More options.

62 fps

# Results: With trajectory optimization



Global Options
Fixed Frame map
Background Color ■ 0; 0; 0
Frame Rate 60
Default Light ☑
Global Status: W…
Fixed Frame No tf data. …
Grid □
Axes □
Marker ☑
✓ Status: Ok
Marker Topic visualizatio…
Queue Size 100000
Namespaces
links □
milestones □
room ☑
cs_obstacles □
agent ☑
path ☑

**milestones**
Enable/disable all markers in this namespace.

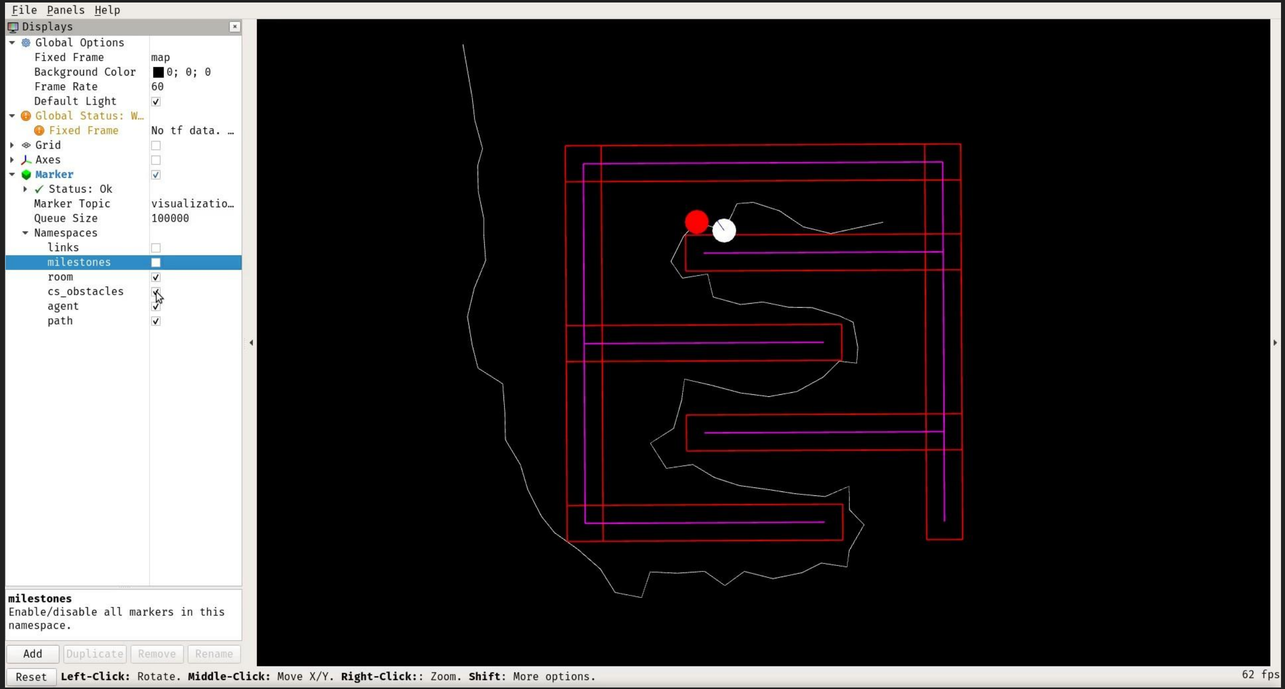
**Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click:**: Zoom. **Shift:** More options.

62 fps

# Results: With trajectory optimization

# Challenges: Building a configuration space

- Taking extent of walls and turtlebot into account



Displays

- Global Options
  - Fixed Frame: map
  - Background Color: ■ 0; 0; 0
  - Frame Rate: 60
  - Default Light: ☑
- Global Status: W…
  - Fixed Frame: No tf data. …
- Grid: □
- Axes: □
- Marker: ☑
  - ✓ Status: Ok
  - Marker Topic: visualizatio…
  - Queue Size: 100000
  - Namespaces
    - links: □
    - milestones: □
    - room: ☑
    - cs_obstacles: ☑
    - agent: ☑
    - path: ☑

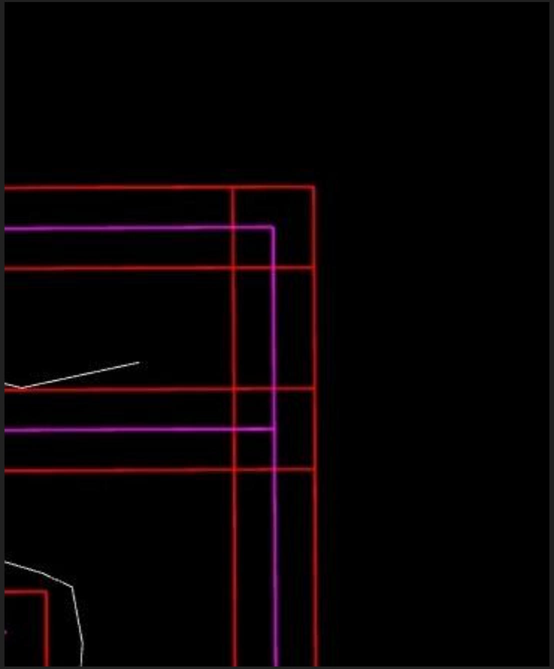
**milestones**
Enable/disable all markers in this namespace.

**Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click::** Zoom. **Shift:** More options.

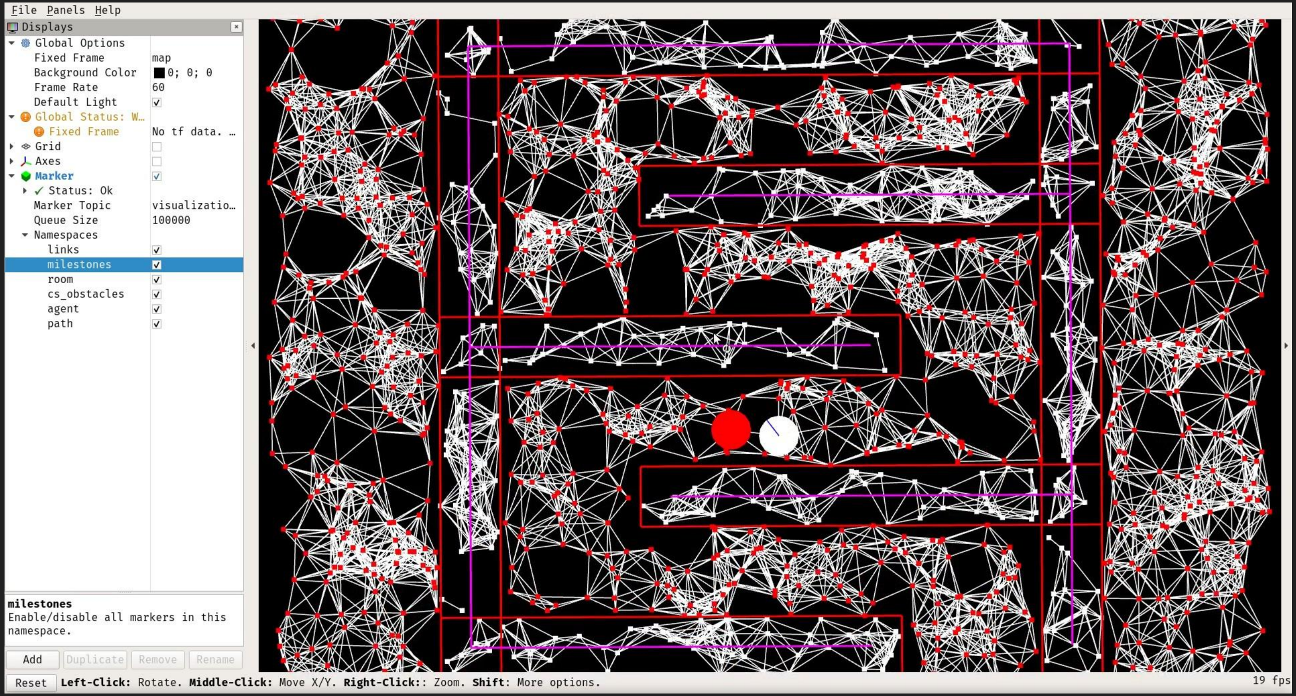
62 fps

# Challenges: CSpace edges of walls

- Simplifying assumption - rectangular CSpace obstacles



# Challenges: Detecting edges intersect that obstacles



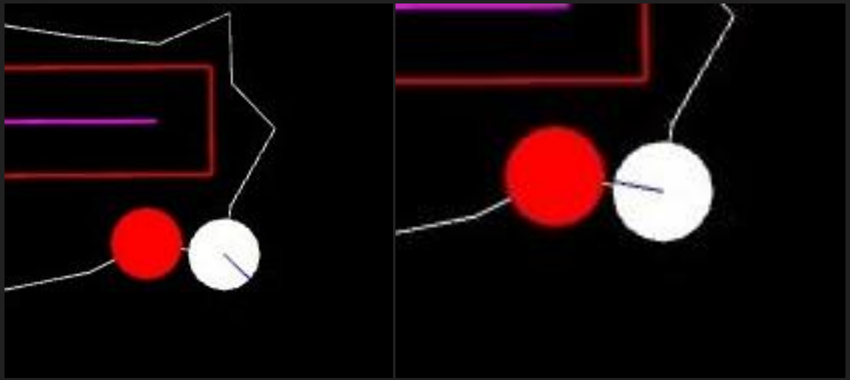
```
▾ Global Options
    Fixed Frame          map
    Background Color     ■ 0; 0; 0
    Frame Rate           60
    Default Light        ☑
▾ Global Status: W…
    Fixed Frame          No tf data. …
▸ Grid                   □
▸ Axes                   □
▾ Marker                 ☑
  ▸ ✓ Status: Ok
    Marker Topic         visualizatio…
    Queue Size           100000
  ▾ Namespaces
      links              ☑
      milestones         ☑
      room               ☑
      cs_obstacles       ☑
      agent              ☑
      path               ☑
```

**milestones**
Enable/disable all markers in this namespace.

**Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click:**: Zoom. **Shift:** More options.

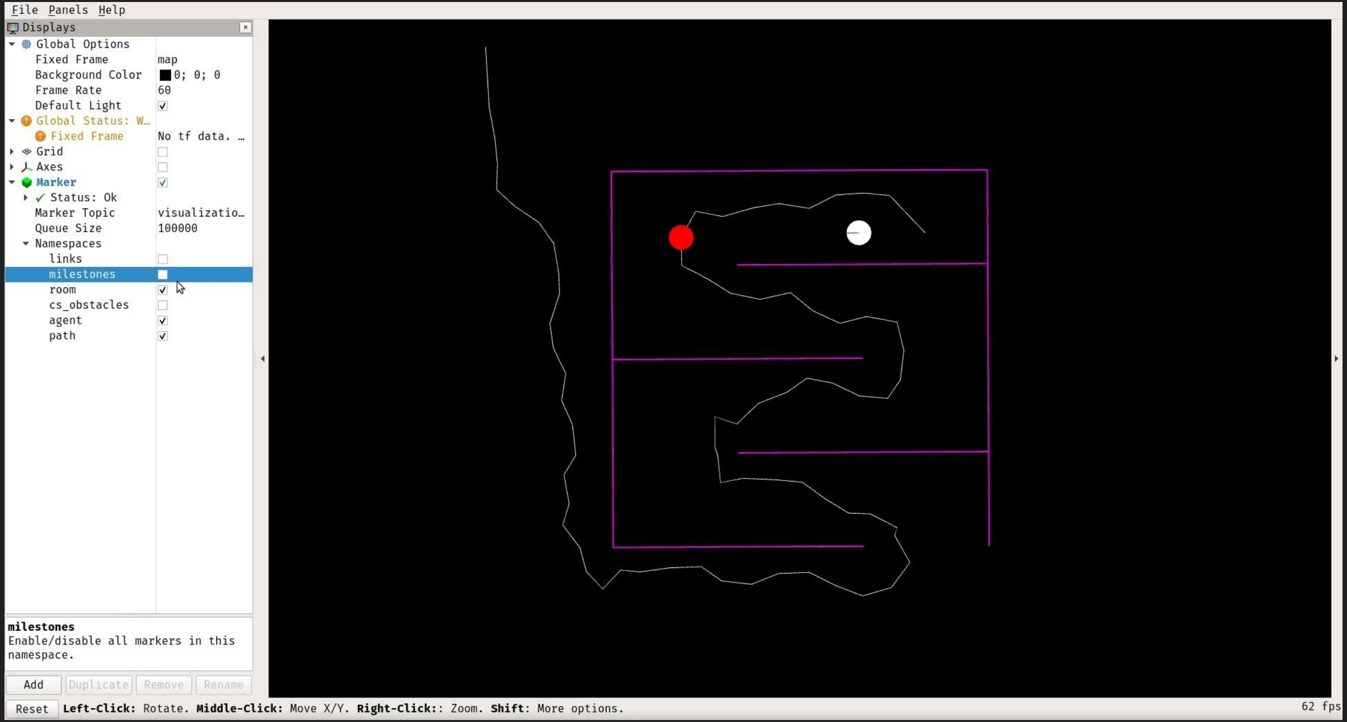
# Challenges: Moving the robot on path

- Simple strategy orient and move



# Challenges: Too many maneuvers

- Use trajectory optimization



```
Global Options
  Fixed Frame          map
  Background Color     ■ 0; 0; 0
  Frame Rate           60
  Default Light        ☑
Global Status: W…
  Fixed Frame          No tf data. …
Grid                   □
Axes                   □
Marker                 ☑
  ✓ Status: Ok
  Marker Topic         visualizatio…
  Queue Size           100000
  Namespaces
    links              □
    milestones         □
    room               ☑
    cs_obstacles       □
    agent              ☑
    path               ☑
```

**milestones**
Enable/disable all markers in this namespace.

**Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click:**: Zoom. **Shift:** More options.

62 fps

# Challenges - ROS/Gazebo

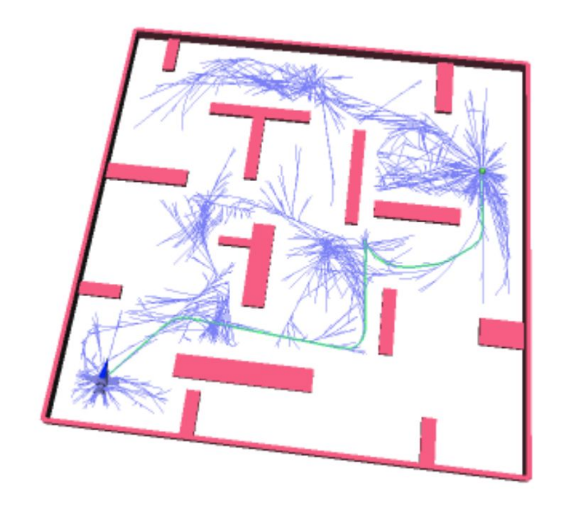
- Installing ROS and Gazebo
- Getting comfortable with ROS/rviz/Gazebo
- Gazebo is a resource hog!!

# Final project should...

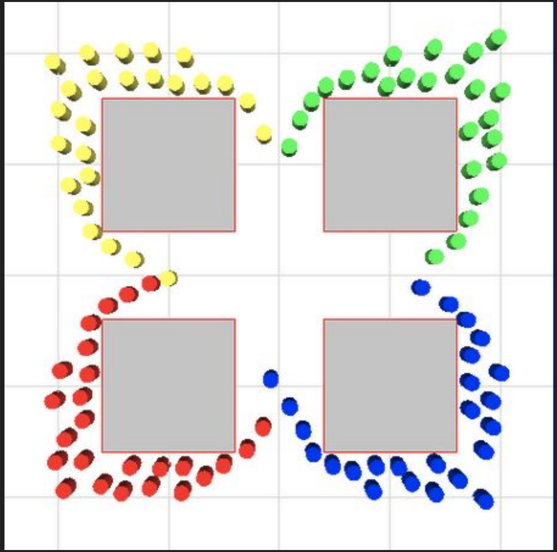
- Break “known” static obstacles assumption
  - Use laser scan for detecting obstacles
  - Keep track of them
  - Plan using currently known obstacles, replan if new ones found on path (spirit of D*Lite)
- Plan with dynamic obstacles
  - **May have to assume that the robot knows the position of humans**
  - Assume friendly behaviour
  - Use repulsive force methods to avoid collision from dynamic obstacles

# The potential

- Use better explorative techniques like RRTs
- Use better dynamic obstacle collision avoidance like TTC or RVO



RRT example



RVO example

Thank you!