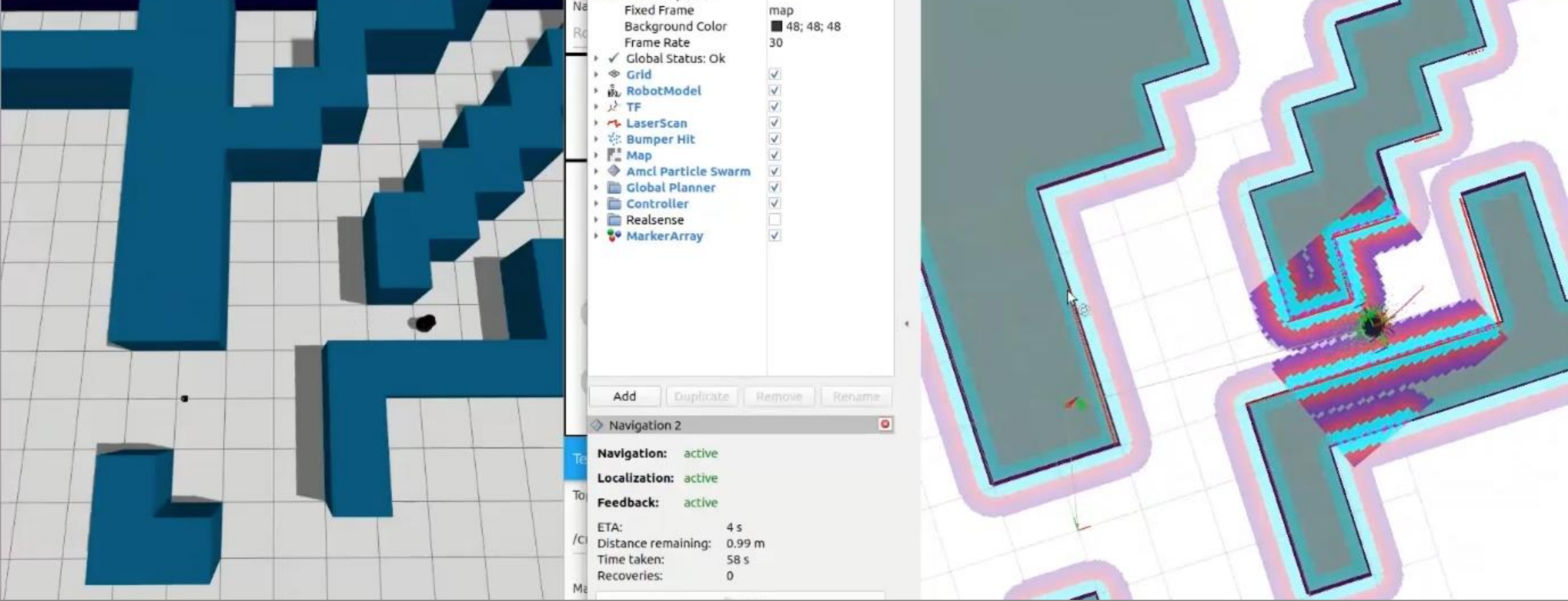


Robotics & XR

5 ECTS
Ilkka Jormanainen and Samuel Yigzaw
{firstname.lastname}@uef.fi

Navigation
November 18, 2024

Course contents

- Theme 1: The role of robotics in society and an introduction to ethical issues in robotics
- Theme 2: Different types of robots and their application to real-world problems
- Theme 3: Robot control theory
- **Theme 4: Navigation**
- Theme 5: Robotics & AI
- Theme 6: Extended Reality applications in robotics

Theme 4: Navigation

- *Week 47 (November 18 – 24)*
- Local and global controllers
- Route planning and most common search and routing algorithms
- Mapping and localization
- 2D and 3D SLAM

Core concepts

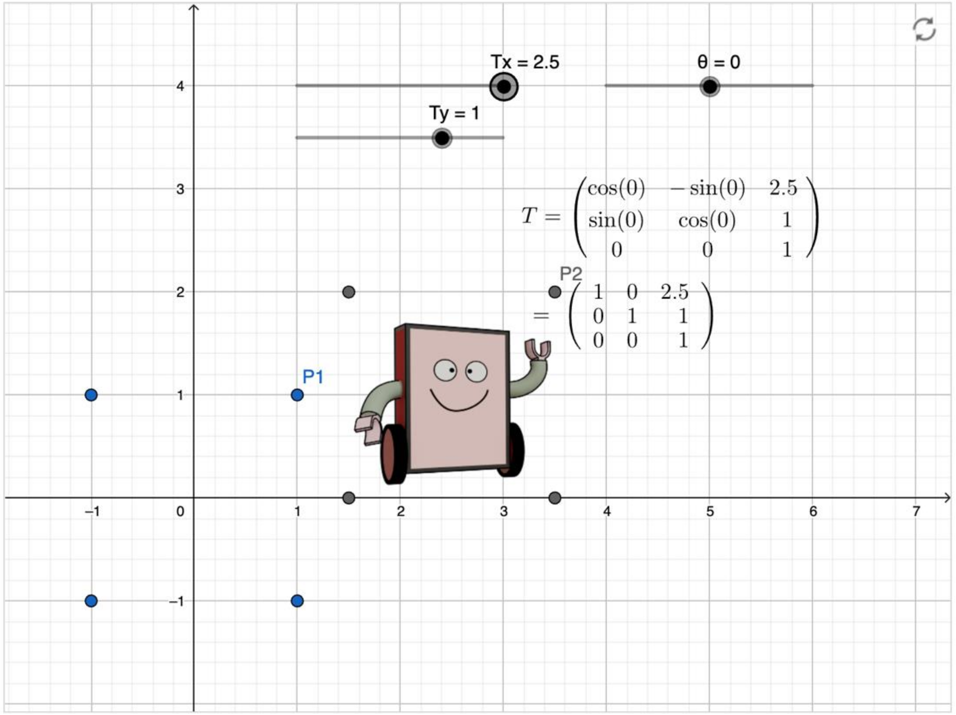
- Transformations (tf)
- Frames
- Odometry
- **Mapping**
- **Localization**
- **Navigation**

Transformations (tf)

- Provide information on how a robot could move in *n*-dimensional space
- A mobile robot typically moves in 2D space and is able to rotate (3 DOF)
- Translation: Changing position in Cartesian 2D coordinate system
- Rotation: Rotating the object in the space

Transformations

- Transformation matrix – rotate and translate



https://articulatedrobotics.xyz/tutorials/coordinate-transforms/transformation-matrices

Navigation

- Very common problem in robotics
- Can be formalized as…
  - Universe *U*
  - Robot configuration *R = <…>*
  - Starting point *s*
  - Target point *t*
- *U* consists of layers, where the robot…
  - can (C_+), or
  - cannot (C_-) move

Robot configurations

- Movement (translate)
  - How can the robot move?
- Rotations
  - How can the robot rotate?
- If a robot is a point in Euclidean space R^d, $d = 2$ and the robot cannot rotate, then the configurations are $<x, y>$
- If a robot is a two-dimensional object, U is two-dimensional plane and the robot can rotate, then the configuration space is $<x, y, q>$, where q is the rotation: 3 degrees of freedom (DOF)

=> The problem becomes very difficult for general R^d

Robot navigation

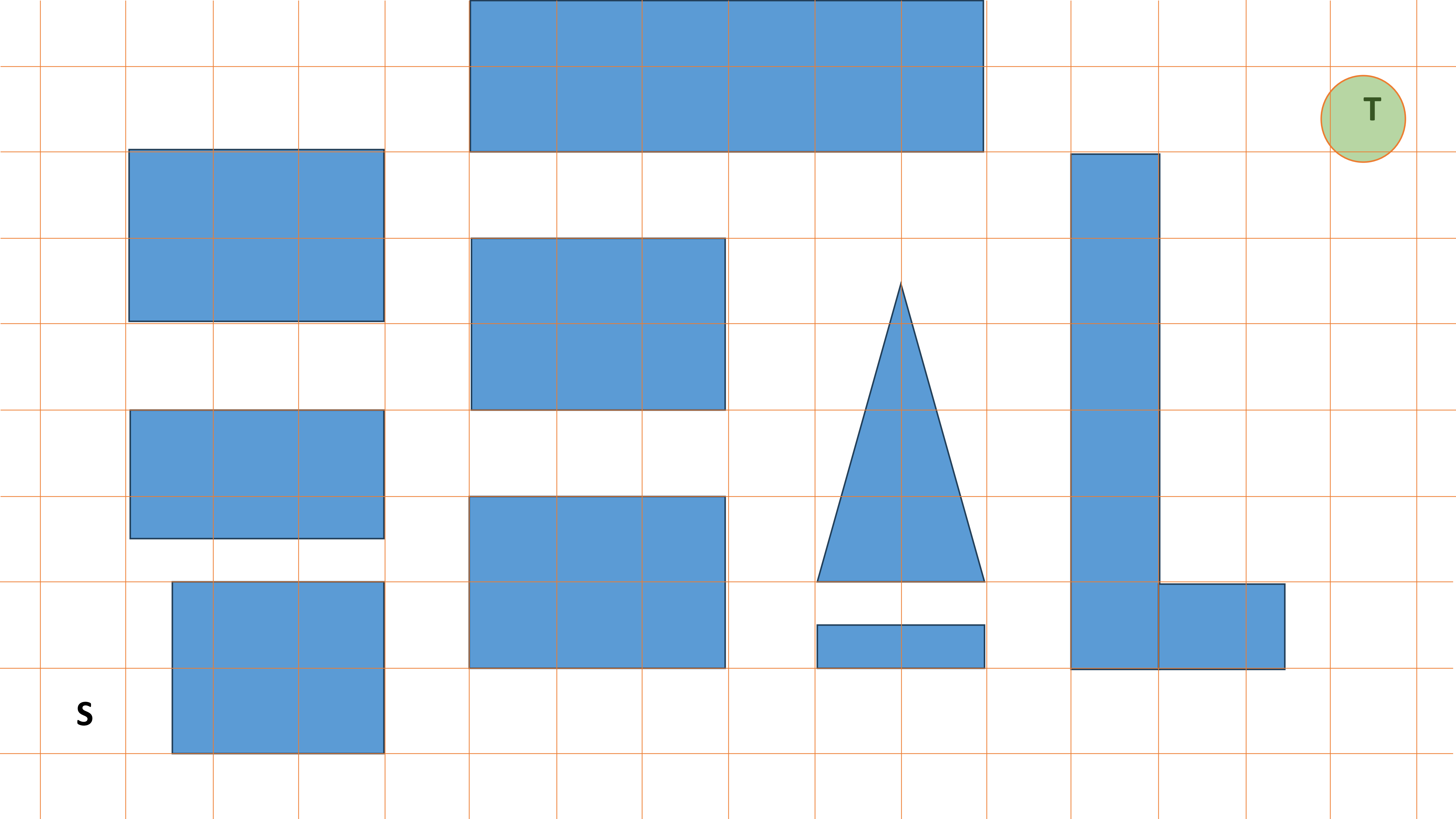
- Two different problems
- Optimization problem
  - Finding a path between two points *s, t*
- Decision problem
  - Is there a path between two points *s, t*?

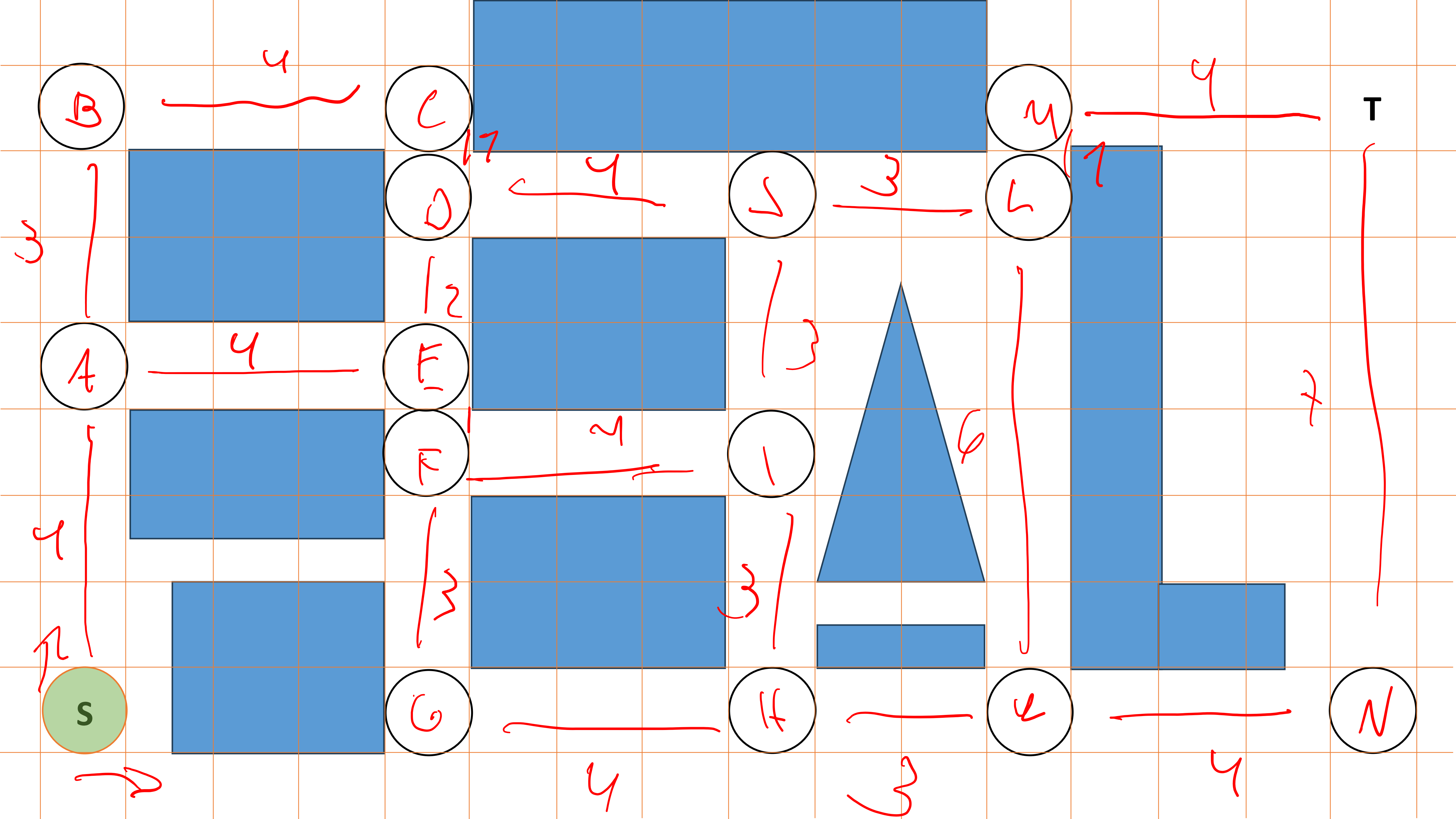
- **The importance of the optimization problem is trivial and clear, but when the decision problem is important?**

Building the navigation

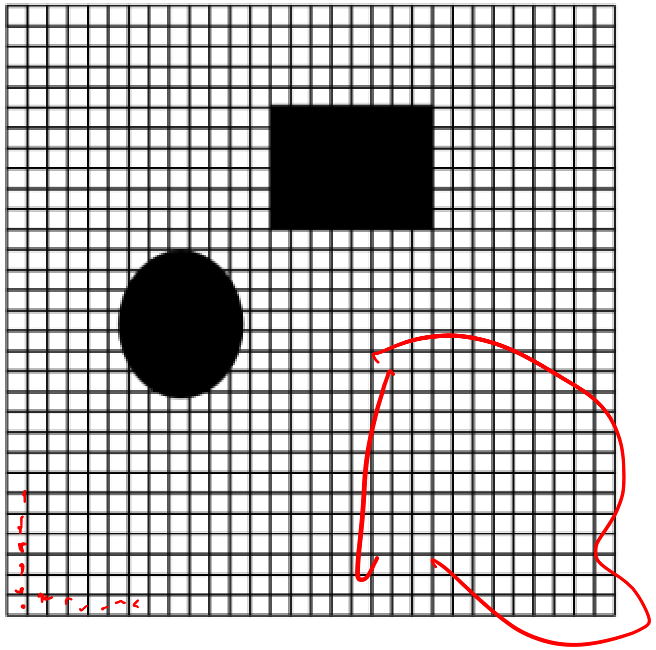
- Create a multidimensional grid on the top of the universe U
- Remove the cells where the robot cannot move to
- Set neighboring cells as adjacent vertices of in an undirected graph
- Set weights to the edges
  - For example, weight of moving from (0,0) to (0,1) is 1
  - Weight of moving from (0,0) to (1,1) is 2 or $\sqrt{2}$, depending on the robot's rotation


T
S


T
S


B
C
M
T
D
S
L
A
E
F
I
G
H
N
S

Occupancy grid maps



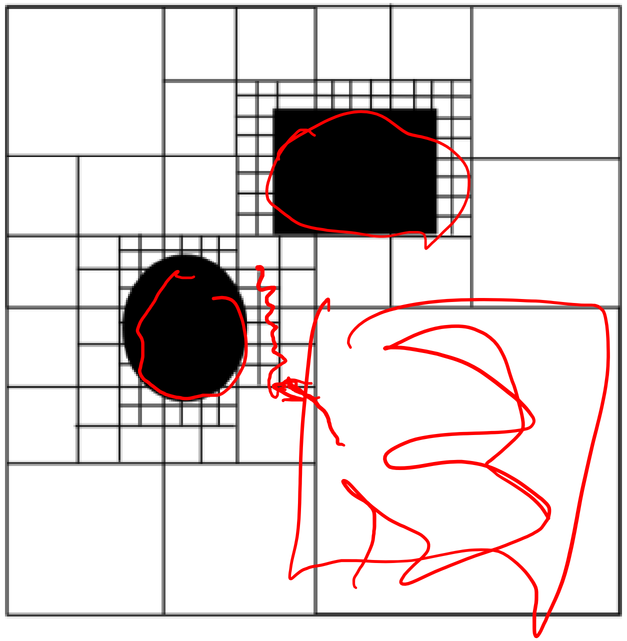


Image source: https://fab.cba.mit.edu/classes/865.21/topics/path_planning/robotic.html

- Arbitrary resolution to map the surrounding
- The map can be stored in *k-d* tree for memory optimization

Configuration spaces

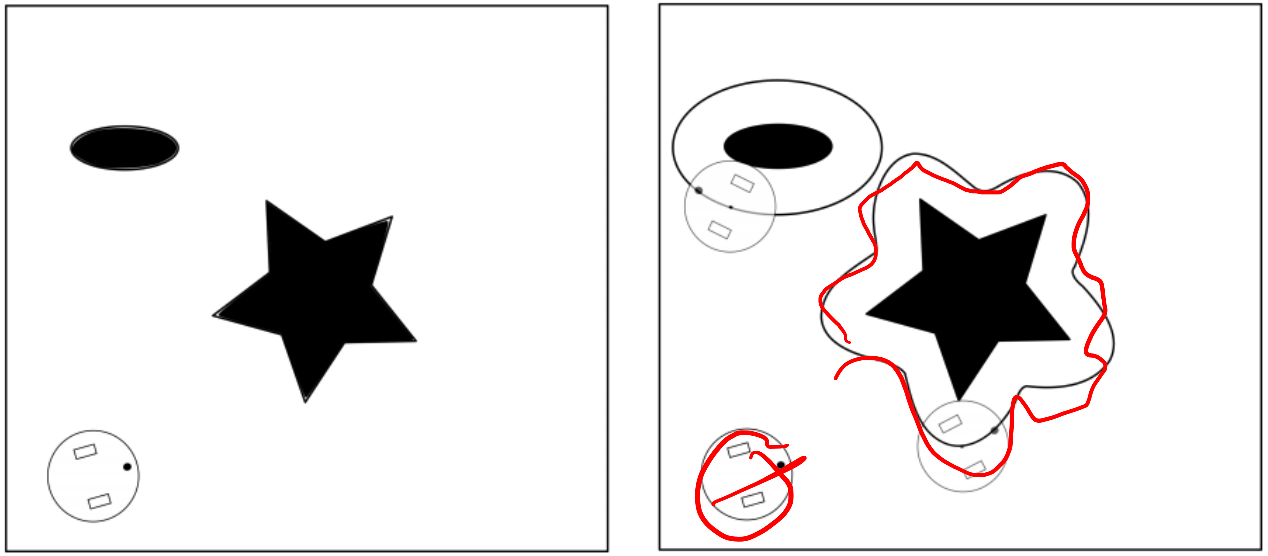


Image source: https://fab.cba.mit.edu/classes/865.21/topics/path_planning/robotic.html

- Obstacles are enlarged by half of the longest extension of the robot

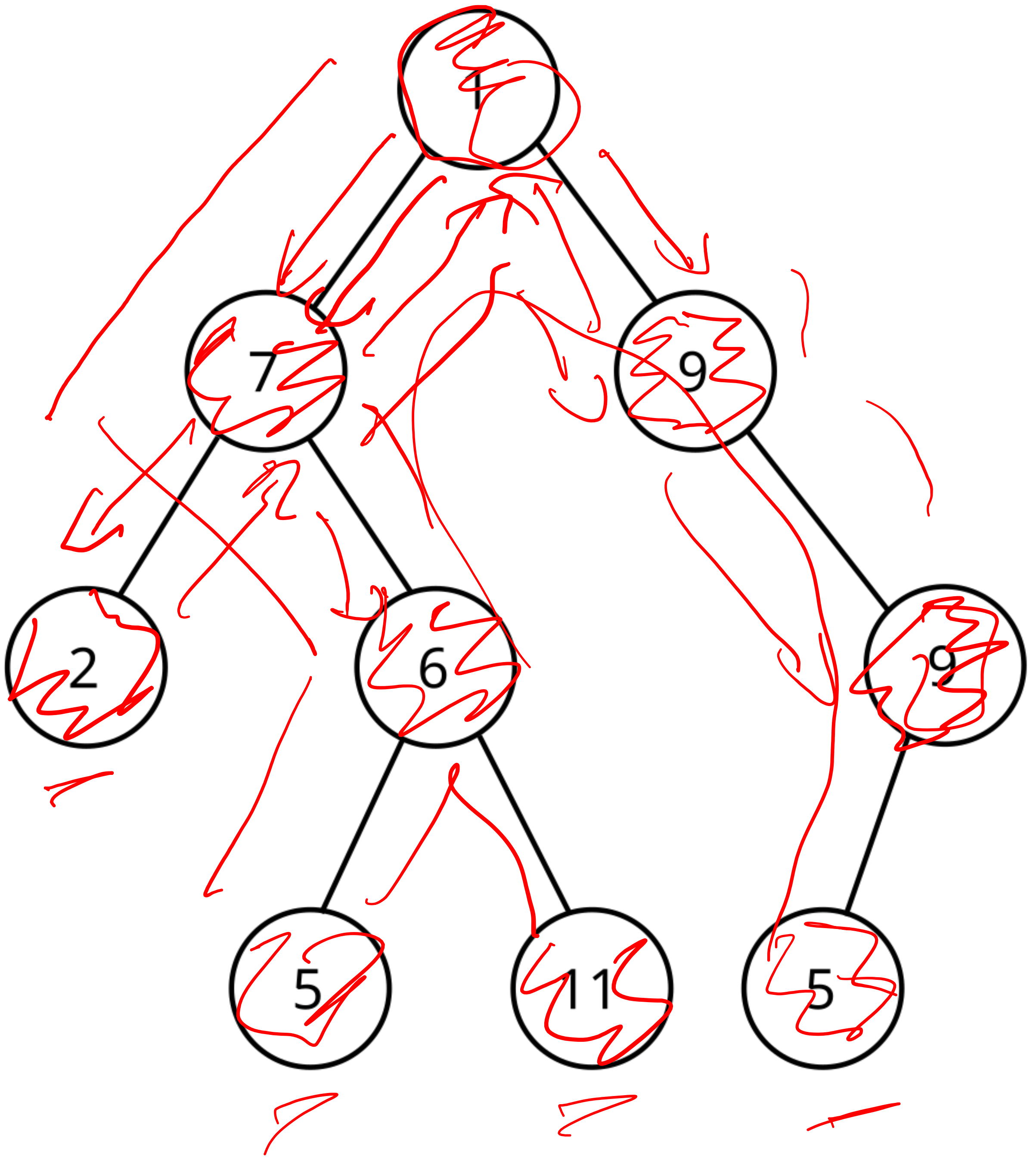
Finding the path

- Solving decision problem or finding the shortest path is relatively easy (at least when the search space is small such as $d=2$)
- Finding the longest path between s, t, instead, is NP-hard problem even in small search spaces (unless it is a directed acyclic graph)
- Is there a path?
  - Depth-first search (DFS), Breadth-first search (BFS)
- Is there a path with the length of less than k?
  - Shortest path problem, Dijkstra, A*

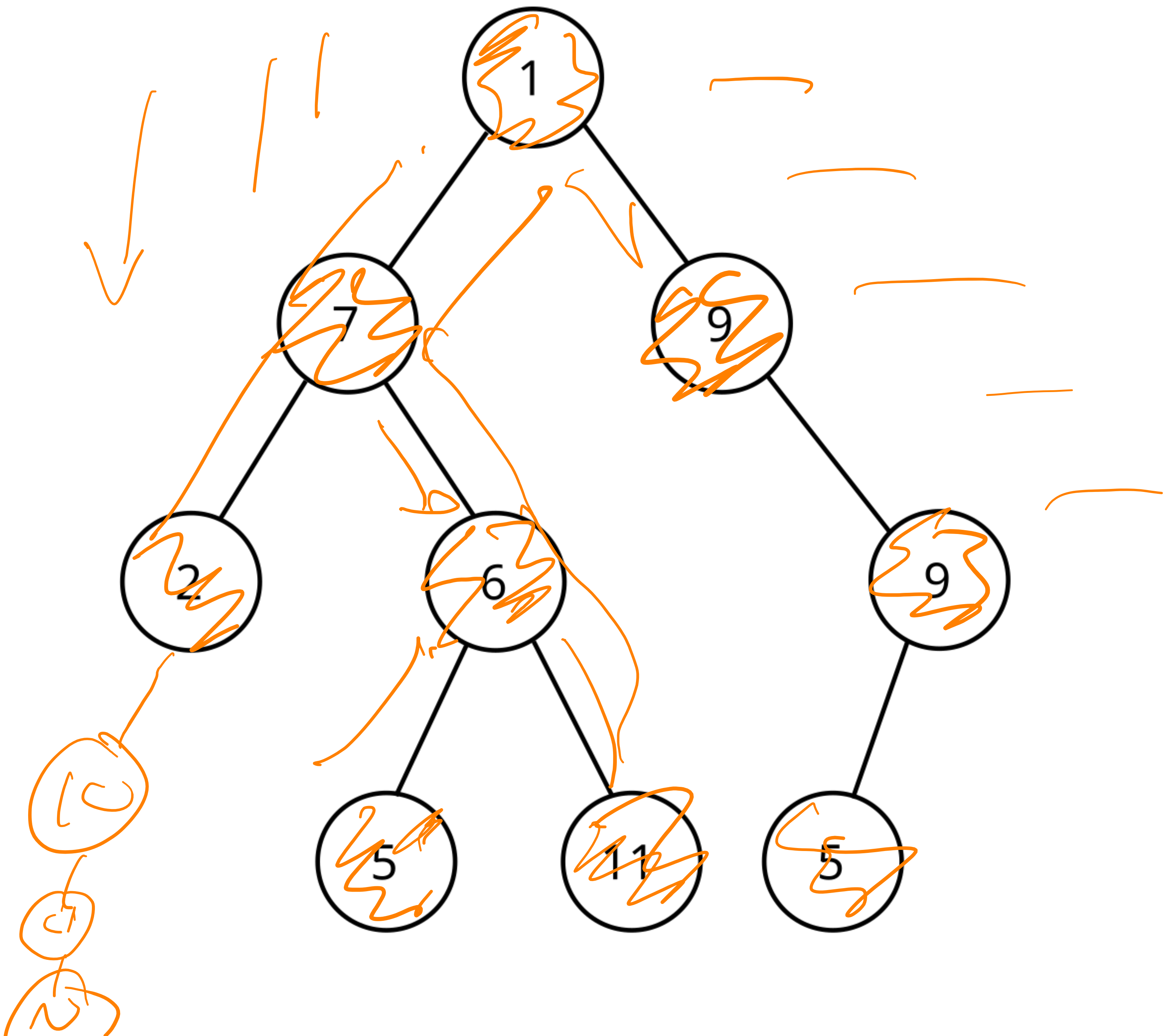
BFS and DFS

- BFS always finds the target
- DFS is not *complete search algorithm*
  - If the graph is infinite, then DFS might not find the target vertex

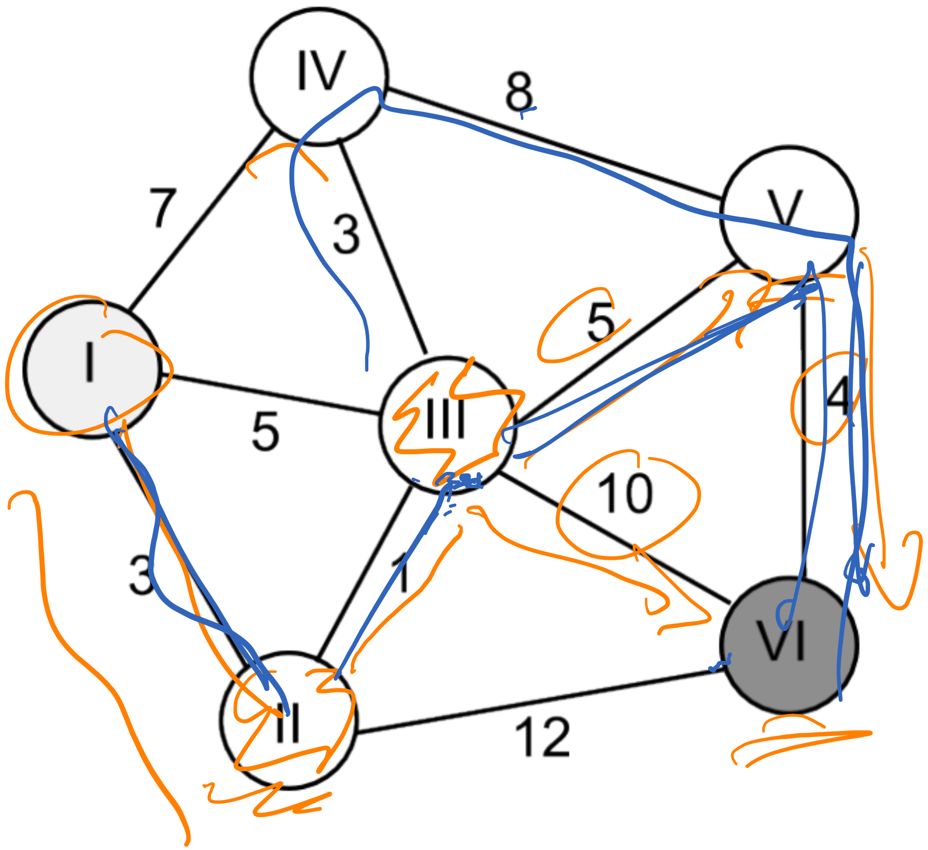
BFS



DFS

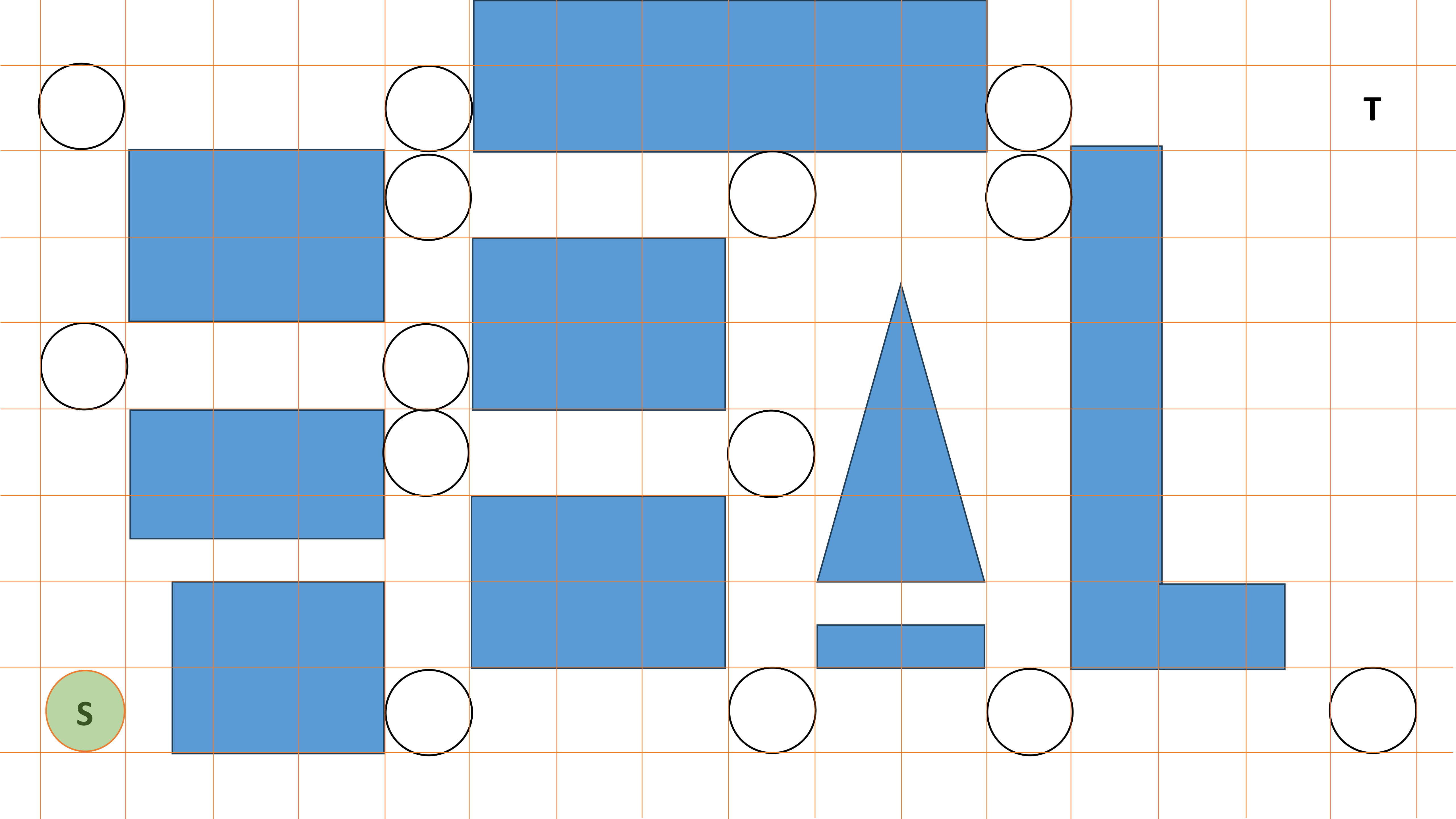


Dijkstra's algorithm



A*, D*

- Commonly used in robotics solutions to find the shortest path
- Uses heuristic to estimate the direction of the search
- Weighting the cost of nodes by their Euclidian distance from the target point
  - Only the paths that are generally directed towards the target will be searched
- D* is incremental search algorithm
  - when combined with A*, the D* allows faster replanning of the route for example around obstacles
- Visualization of A*



T
S

Global planners

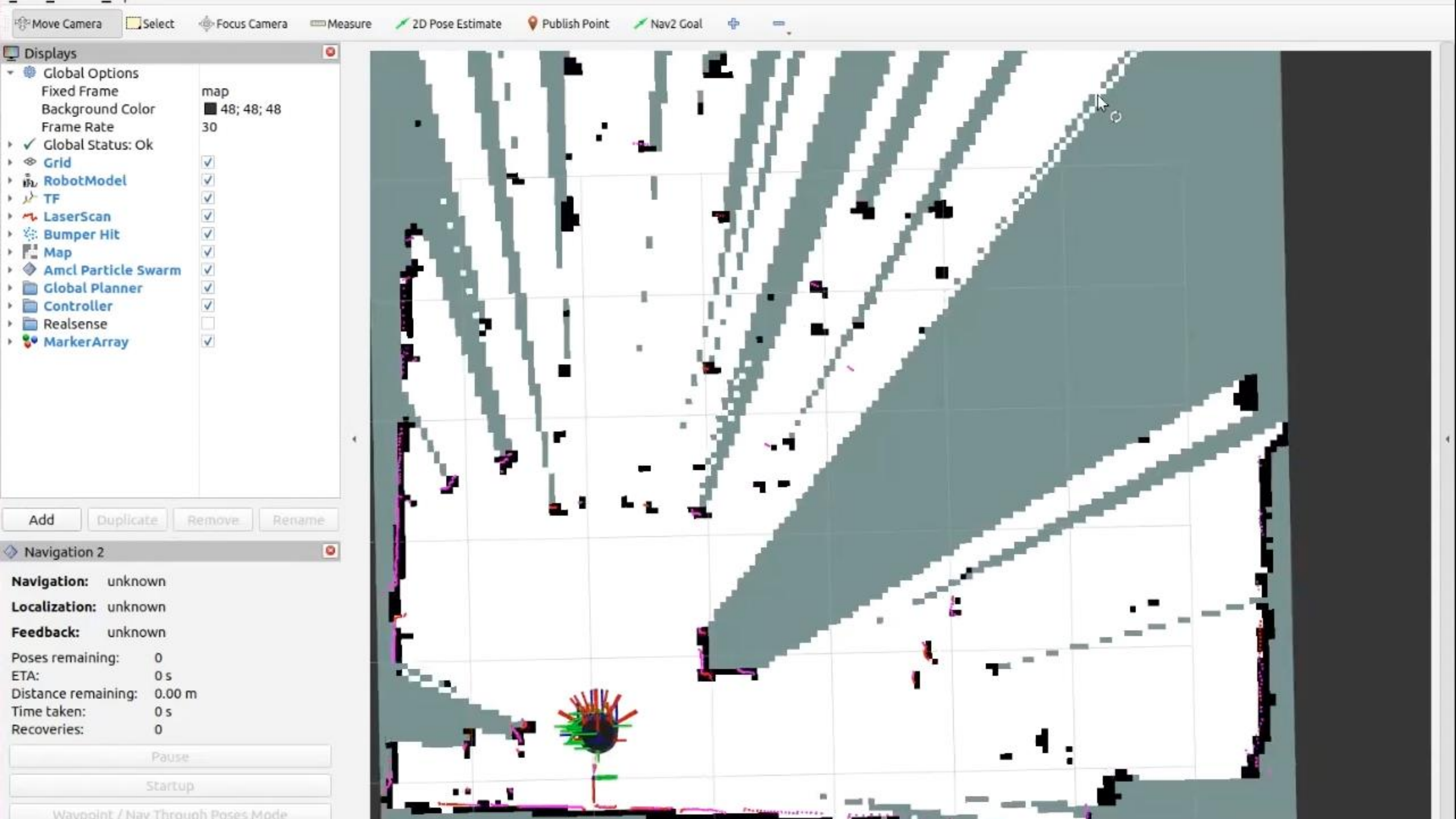
- Designing a ”big picture” of the navigation task – roadmap to the destination
- Avoid known large-scale objects and find optimal or near-optimal path
- Uses static or semi-static maps for path planning
- Dijkstra, A* algorithms (graph-based methods) on previously mapped environment and saved maps (occupancy grids)

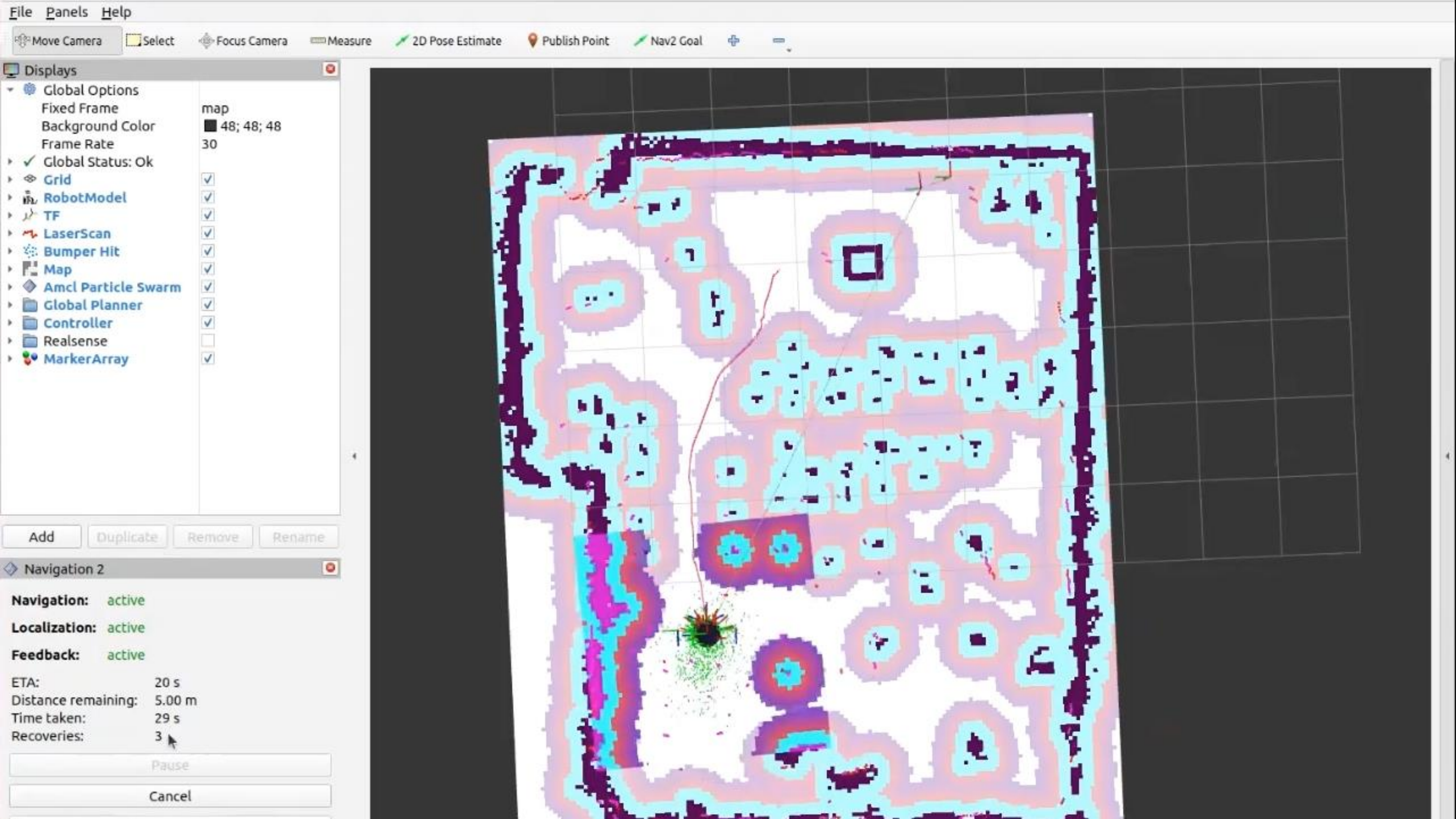
Local planners

- Refine and execute the path planned by the global planner
- Enable a robot to react and response to dynamic and unforeseen obstacles or other real-time constrains
- Responsiveness through real-time sensor data
- Focus on small area surrounding the robot

Simultaneous Localization and Mapping (SLAM)

- Perception
  - Aiming to measure robot’s motion
  - Sensors: LIDAR, cameras, IMU, etc
- Localization
  - Estimating robot’s position and orientation (pose) in relation to the map
  - Odometry, Kalman filter, particle filters
- Mapping
  - Representation of the environment in a structured way
  - Occupancy grid, landmark detection
- State estimation
  - Combining sensor data to estimate robot’s state and the map
  - EKF-SLAM, particle filters (FastSLAM)


Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
Publish Point
Nav2 Goal
Displays
Global Options
Fixed Frame
map
Background Color
48; 48; 48
Frame Rate
30
Global Status: Ok
Grid
RobotModel
TF
LaserScan
Bumper Hit
Map
Amcl Particle Swarm
Global Planner
Controller
Realsense
MarkerArray
Add
Duplicate
Remove
Rename
Navigation 2
Navigation: unknown
Localization: unknown
Feedback: unknown
Poses remaining: 0
ETA: 0 s
Distance remaining: 0.00 m
Time taken: 0 s
Recoveries: 0
Pause
Startup
Waypoint / Nav Through Poses Mode


File
Panels
Help
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
Publish Point
Nav2 Goal
Displays
Global Options
Fixed Frame
map
Background Color
48; 48; 48
Frame Rate
30
Global Status: Ok
Grid
RobotModel
TF
LaserScan
Bumper Hit
Map
Amcl Particle Swarm
Global Planner
Controller
Realsense
MarkerArray
Add
Duplicate
Remove
Rename
Navigation 2
Navigation: active
Localization: active
Feedback: active
ETA: 20 s
Distance remaining: 5.00 m
Time taken: 29 s
Recoveries: 3
Pause
Cancel