SIMULATED SWARM

Team: Black Team
Sponsor: Lockheed Martin

Group Members & Division of Labor

Jakob Germann
Swarm Intelligence,
Phase 2 Navigation,
Team Leader

Keegan Kerns
Simultaneous
Localization & Mapping
Navigation

Akash Samlal
Simultaneous
Localization & Mapping
Navigation

JJ Chan
Object Detection/
Recognition

Caleb McCown
Object Detection/
Recognition

Project Overview

Goal: Implement an Autonomous Swarm of Drones that searches for a Target in an Urban Environment under given time constraints.

Project Justification

- Search & Rescue Operations.
- Disaster Relief Resource Delivery.
  - Hurricane Ida & Louisiana evacuation efforts.
- High Speed Distribution.
  - Amazon Prime Air.

Requirements

1. *Swarm will consist of 4 - 5 drones.
2. *Swarm must find target within 3 minutes.
3. *Object detection model must have a minimum 70% accuracy.
4. *Object detection model must have Intersection of Union greater than 0.5.
5. Object detection model will detect a custom target. (BB8)
6. Each drone uses Lidar to accurately localize itself and avoid obstacles.
7. Must use ROS and Gazebo.

*Sponsor Specified

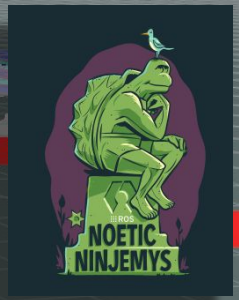
Technologies Used

Implementation

- Operating System - Ubuntu 20.04.
- Robotics Middleware - ROS Noetic.
- Sim Environment - Gazebo 11.
- Object Detection - You Only Look Once (YOLOv5).

Administration and communication

- Discord.
- GitHub.
- Google Drive.
- Jira.
- Zoom.



Jira

GAZEBO

Test World

Overview of Urban Environment

Top-Down view of Suburban Environment

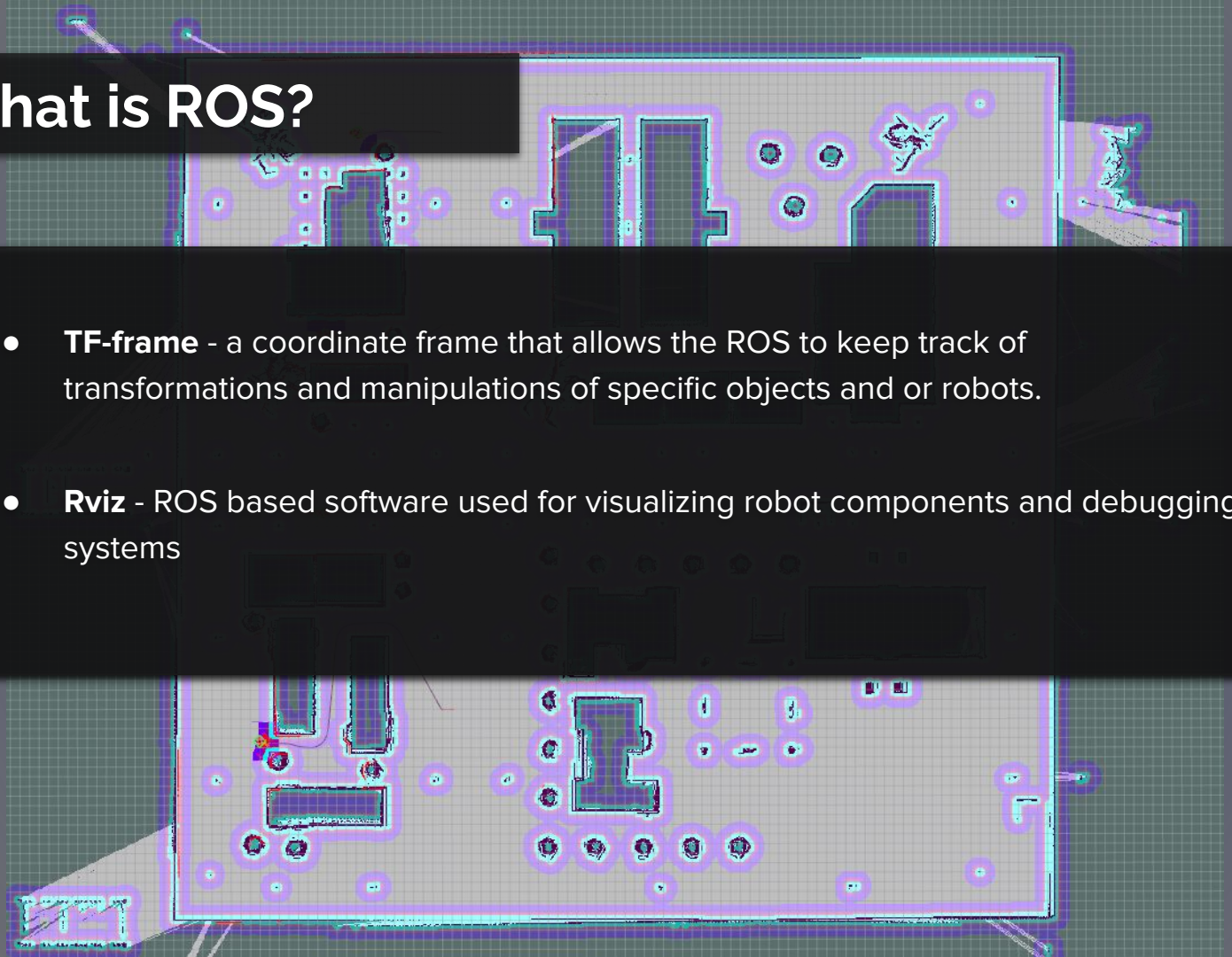
- Both sections are roughly 172 x 172 meters.
- Constraint: Target will not spawn inside or atop any buildings.

What is ROS?

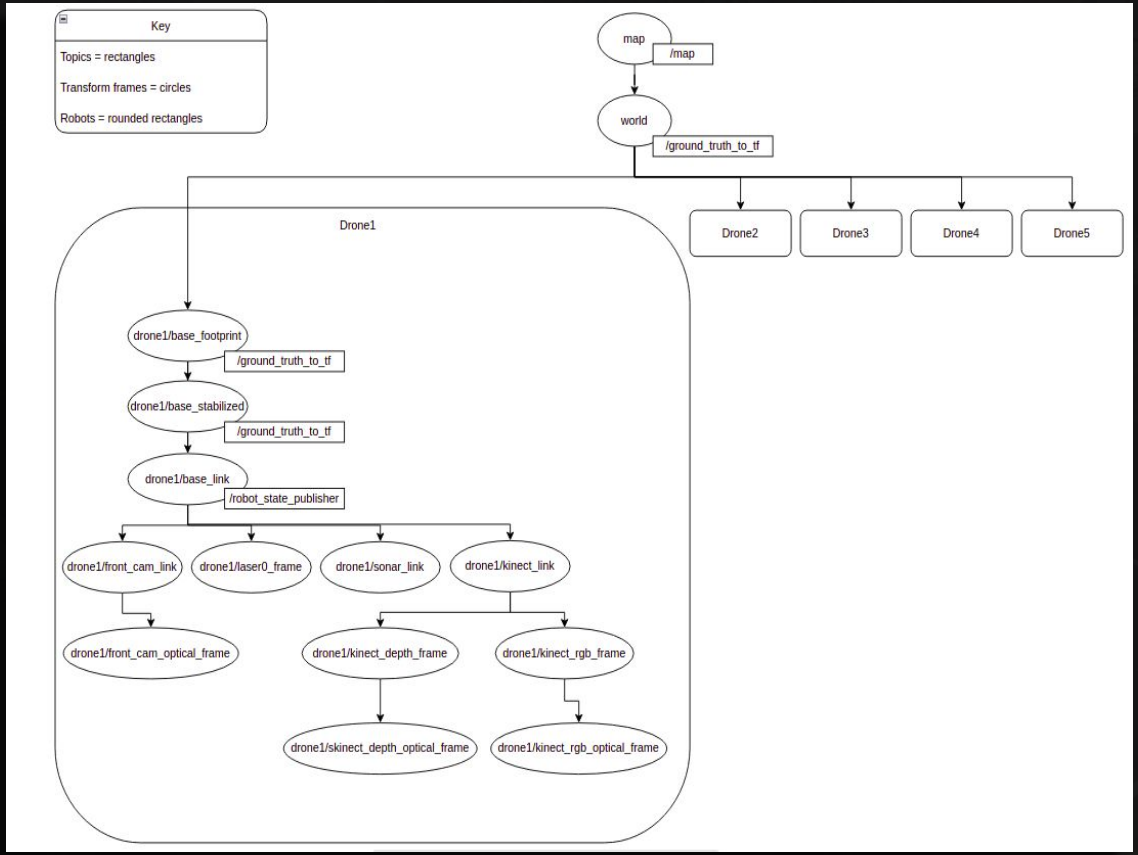
- **ROS** - Open source software that defines the components, interfaces and tools for advanced robots.
- **Nodes** - a process that performs computation.
- **Topics** - named buses that nodes use to exchange messages via a continuous stream.
- **Messages** - a data structure that has typed fields.
- **Services** - like topics, but only provide data upon request.
- **Actions** - communication method combining topics & services to allow for sophisticated systems

What is ROS?

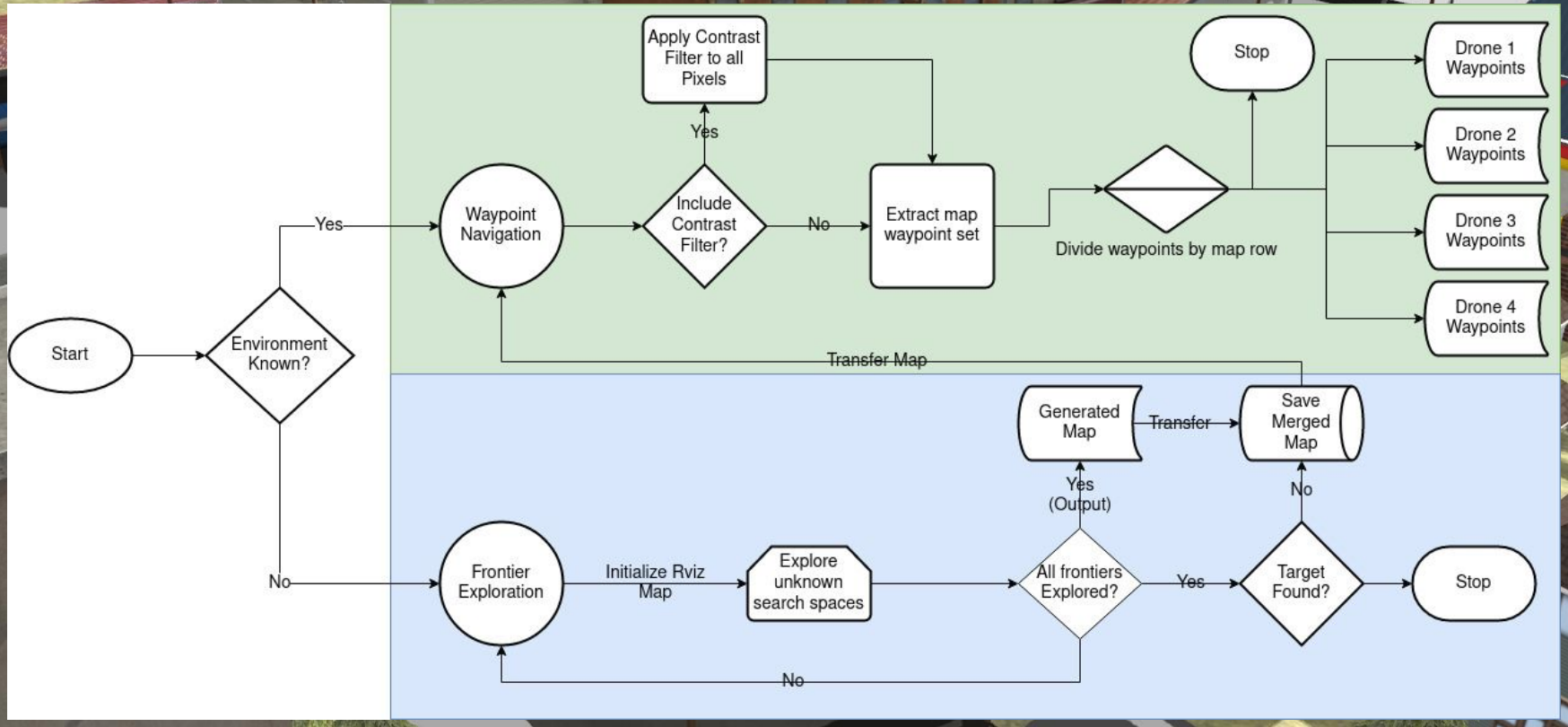
- **TF-frame** - a coordinate frame that allows the ROS to keep track of transformations and manipulations of specific objects and or robots.
- **Rviz** - ROS based software used for visualizing robot components and debugging systems



TF Frame Block Diagram



Project Hierarchy



Object Detection

- JJ Chan & Caleb McCown

Training Dataset

- BB8 target.
- Trained on 890 images and validated on 100 images post-augmentation.
  - Image augmentations and resizes done on Roboflow.
  - 1920 x 1080 fit to 960 x 960.
  - Null images.
  - Noise augmentation.

YOLOv5 Integration

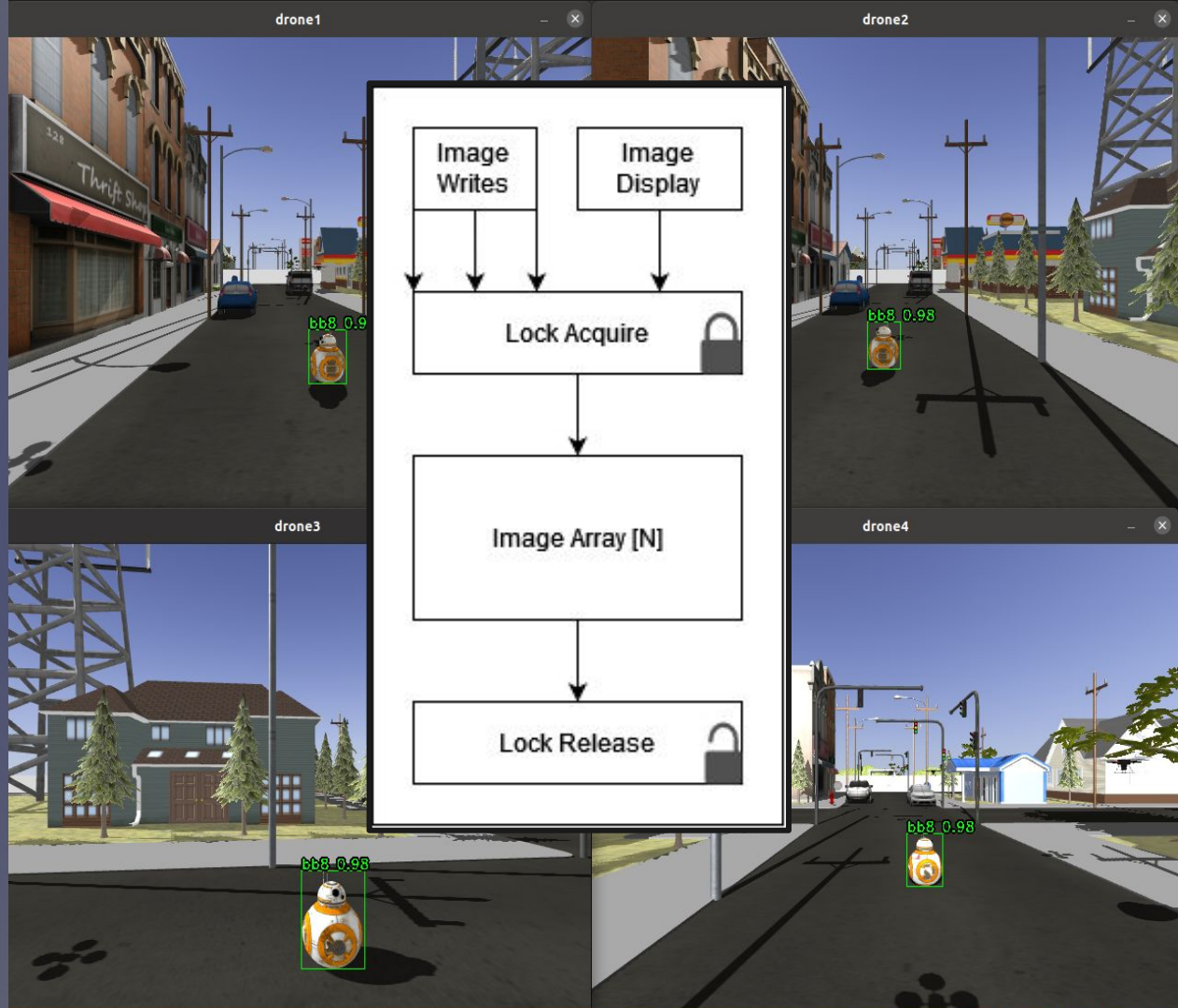
- YOLOv5 is a model that takes in weights and pictures, and it returns corners of a bounding box and confidence.
- ROS sends camera feeds to the YOLO program, which infers and annotates the image.
- The annotated image gets displayed to the screen.

YOLOv5 Code Flexibility

- A single instance of YOLOv5 can theoretically handle N drones, given the hardware is fast enough.
- Otherwise, multiple instances can be ran with less drones to divide the load.

Thread Safety

- YOLOv5 is thread-safe which simplifies implementation.
- The image array and OpenCV's display function are not thread-safe.
- A lock protects the image array, and the display function gets its own thread.

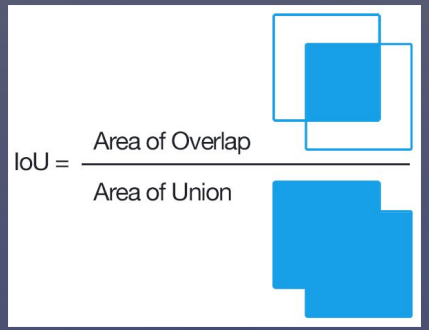


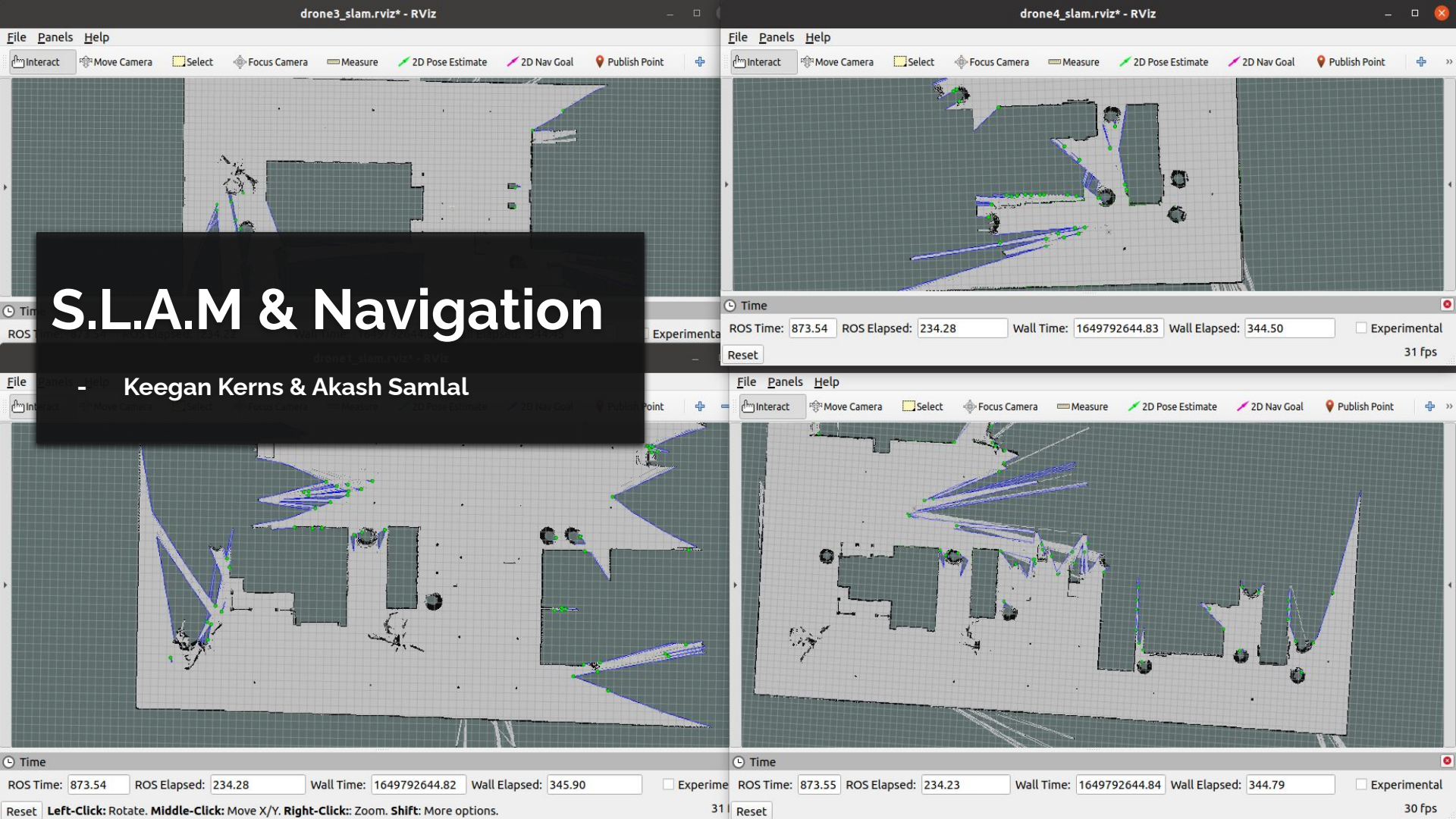
Accuracy Performance

- Stop condition triggers when confidence > 0.96.
- Model has not triggered stop condition with anything other than BB8.

IoU Performance

- In every case where confidence > 0.90, the bounding box is extremely tight.





S.L.A.M & Navigation

- Keegan Kerns & Akash Samlal

Phase 1 - Exploration

SLAM is an algorithm that allows robot to construct a map from an unknown environment while simultaneously keeping track of the robot's location.

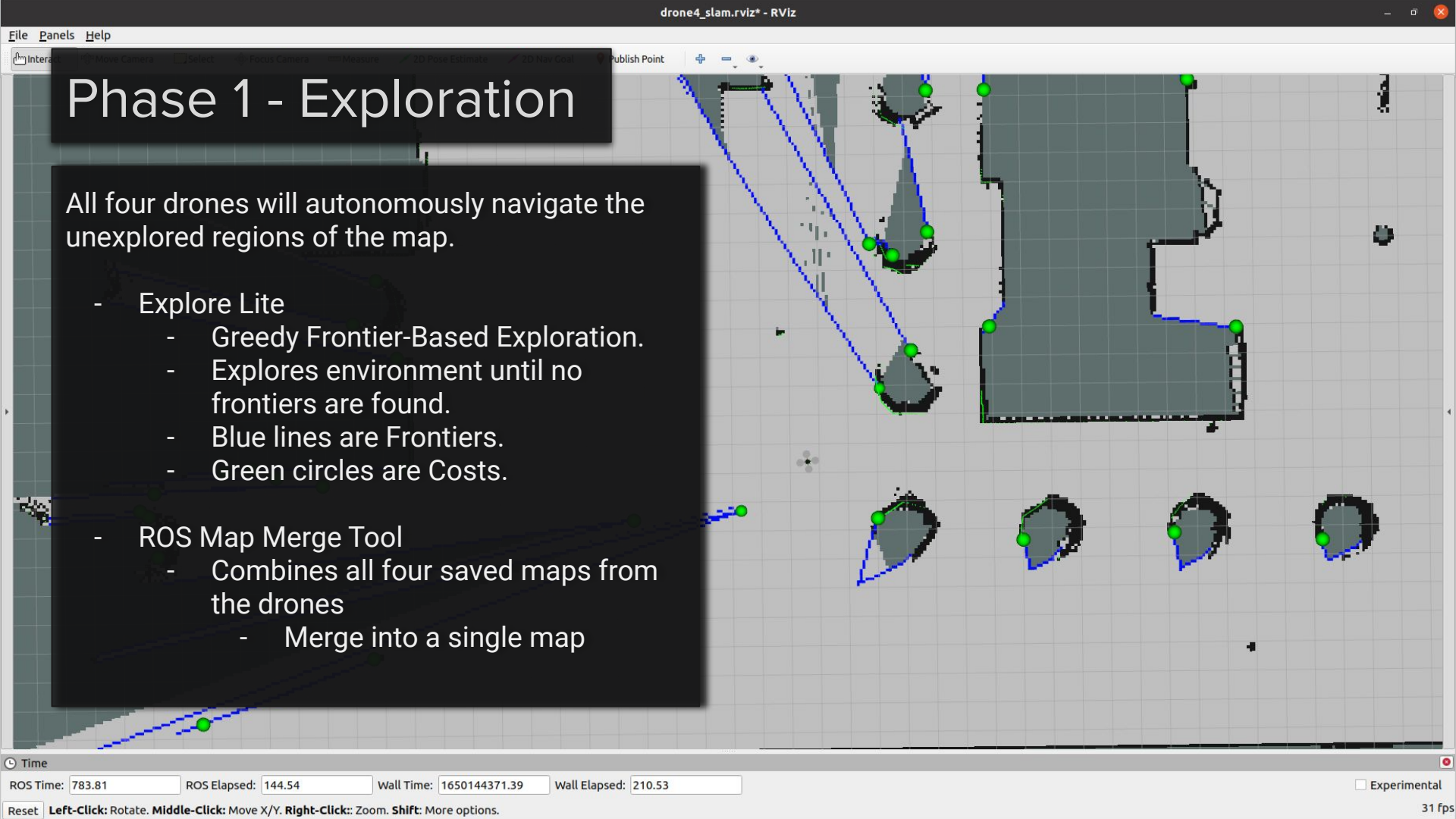
Requirement before autonomous navigation of a known map:

Steps:

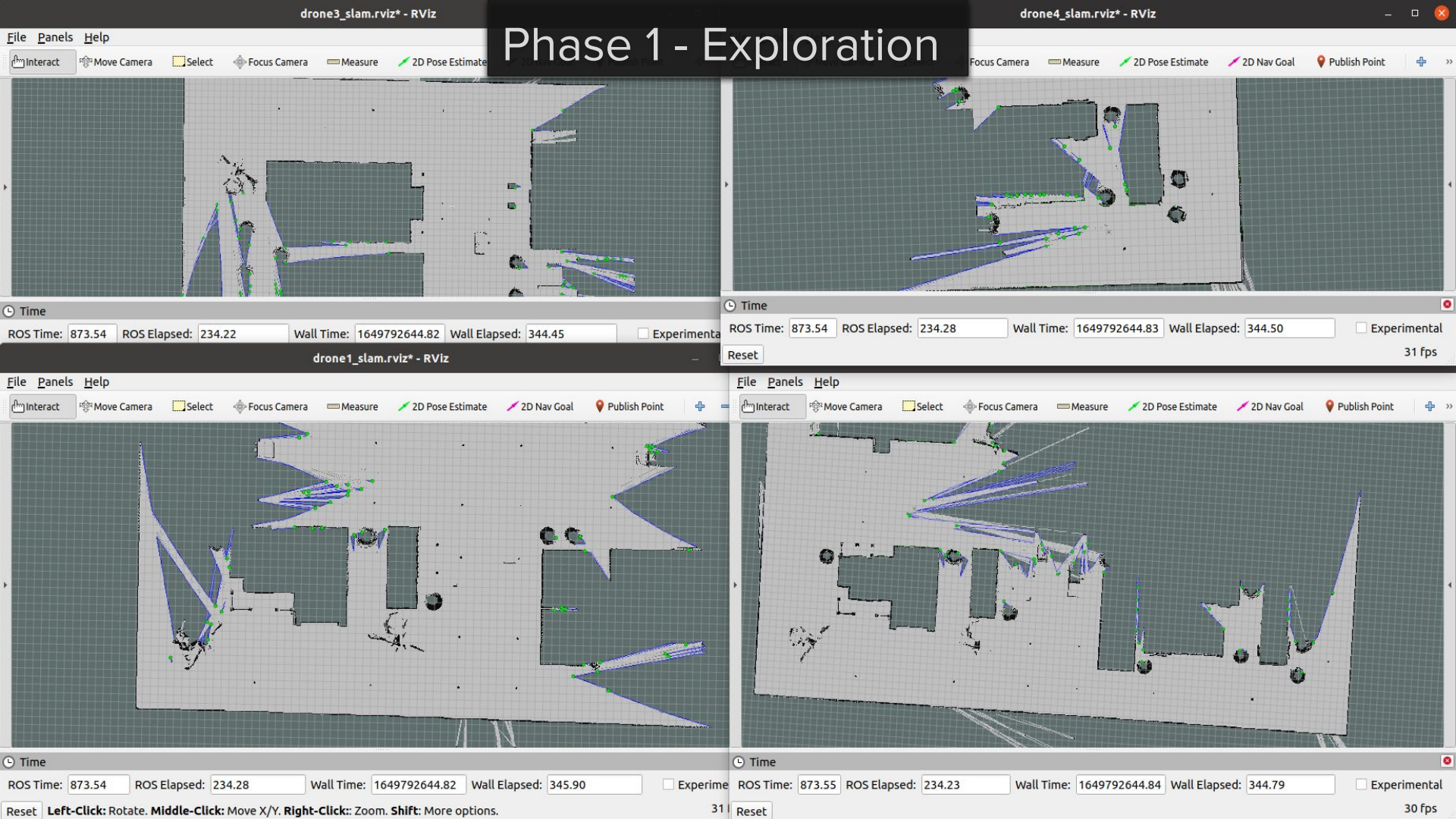
1. Map out the unknown environment.
2. Load the mapped map and localized the drone.

We use:

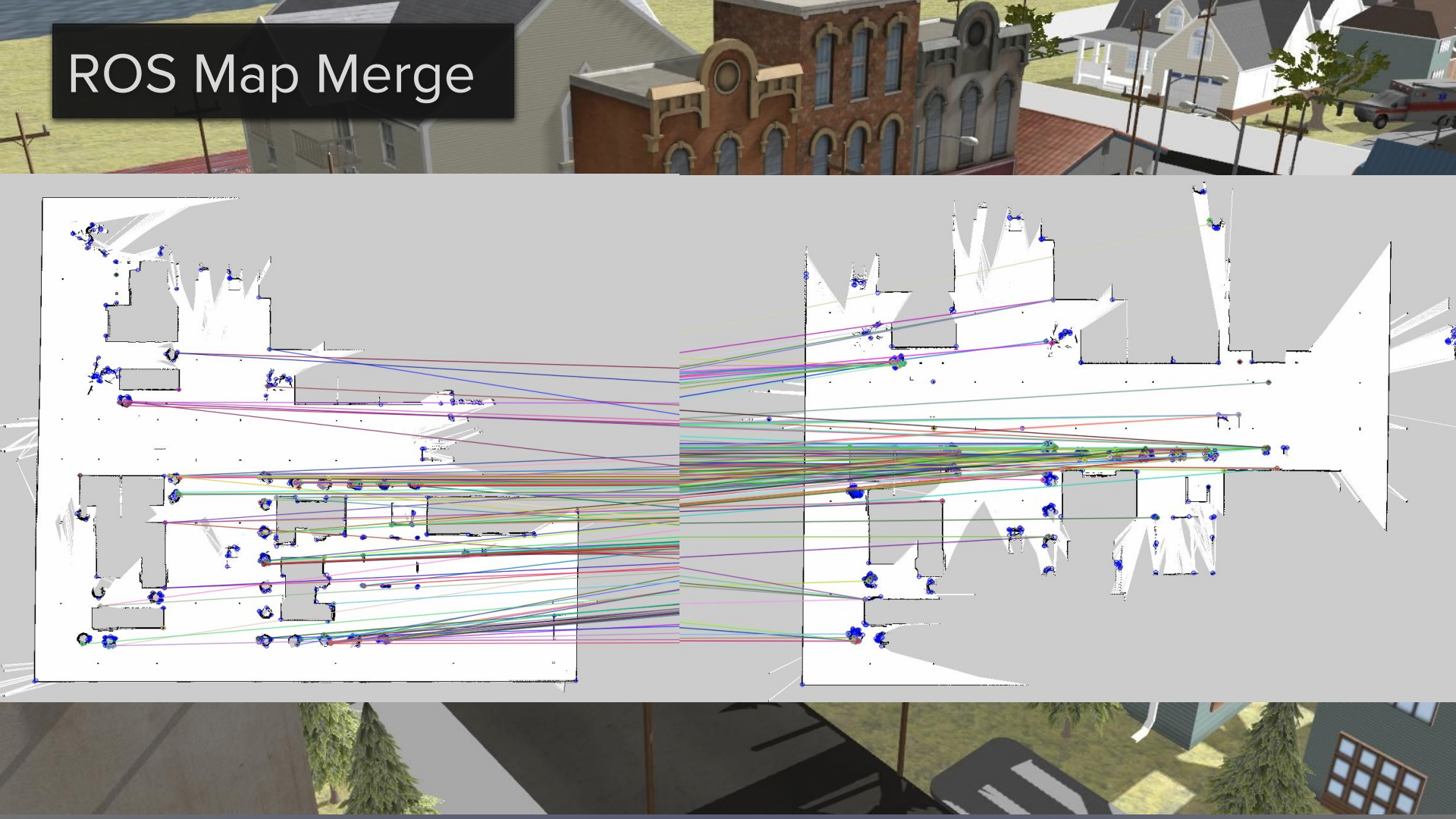
- Hector SLAM
  - 2D SLAM System.


drone4_slam.rviz* - RViz
File Panels Help
Publish Point
Phase 1 - Exploration
All four drones will autonomously navigate the unexplored regions of the map.
- Explore Lite
  - Greedy Frontier-Based Exploration.
  - Explores environment until no frontiers are found.
  - Blue lines are Frontiers.
  - Green circles are Costs.
- ROS Map Merge Tool
  - Combines all four saved maps from the drones
    - Merge into a single map
Time
ROS Time: 783.81
ROS Elapsed: 144.54
Wall Time: 1650144371.39
Wall Elapsed: 210.53
Experimental
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click:: Zoom. Shift: More options.
31 fps

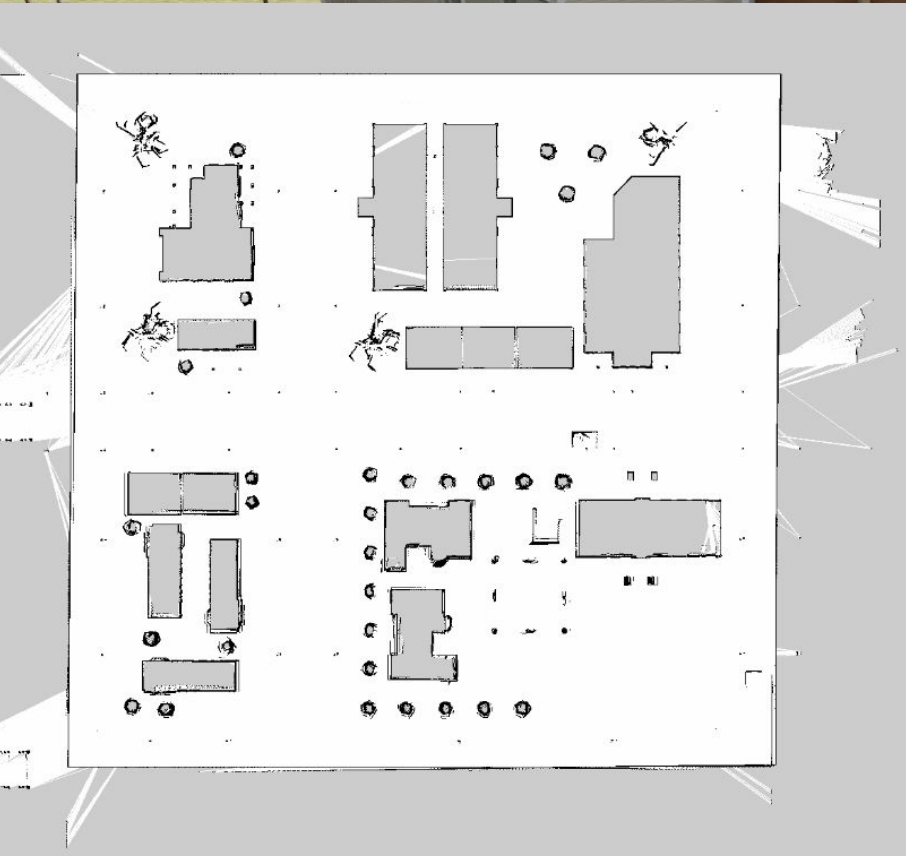
Phase 1 - Exploration



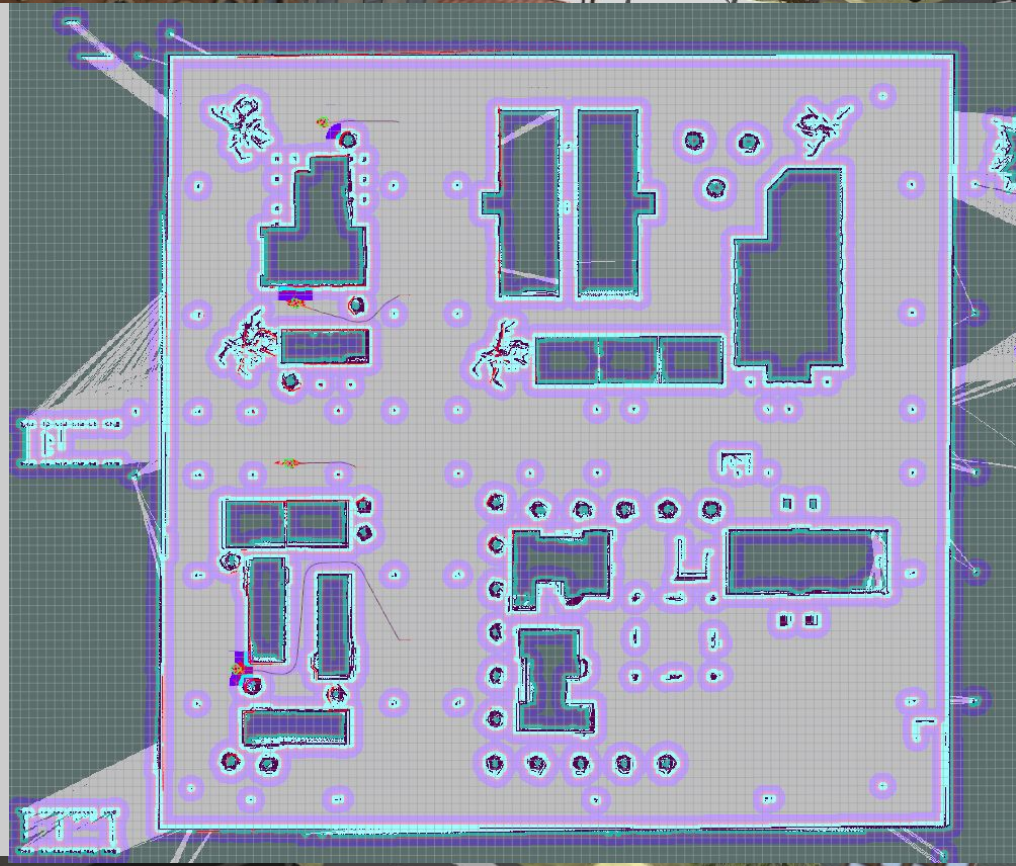
ROS Map Merge



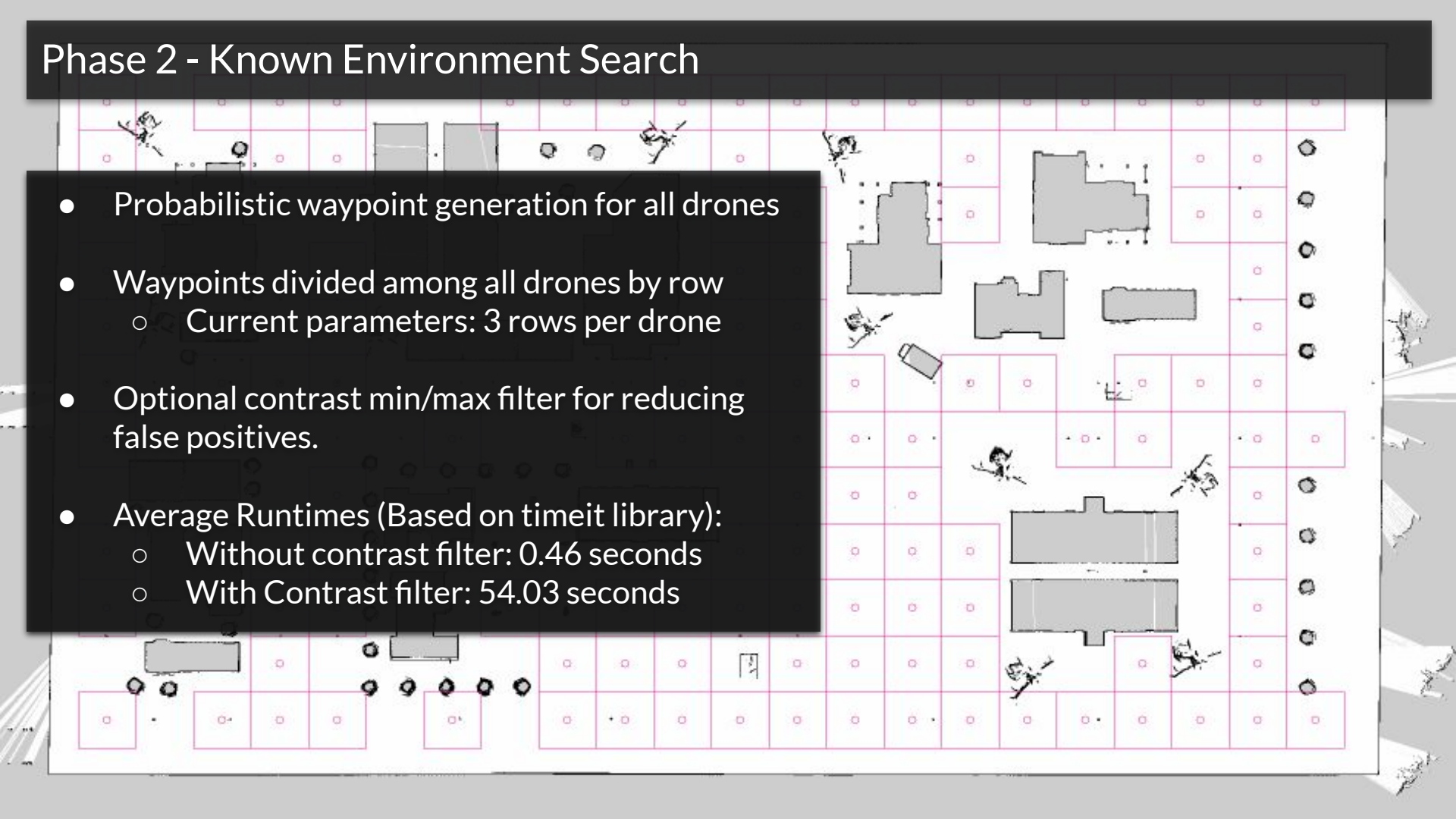
Final Map Generated



Loaded into Rviz



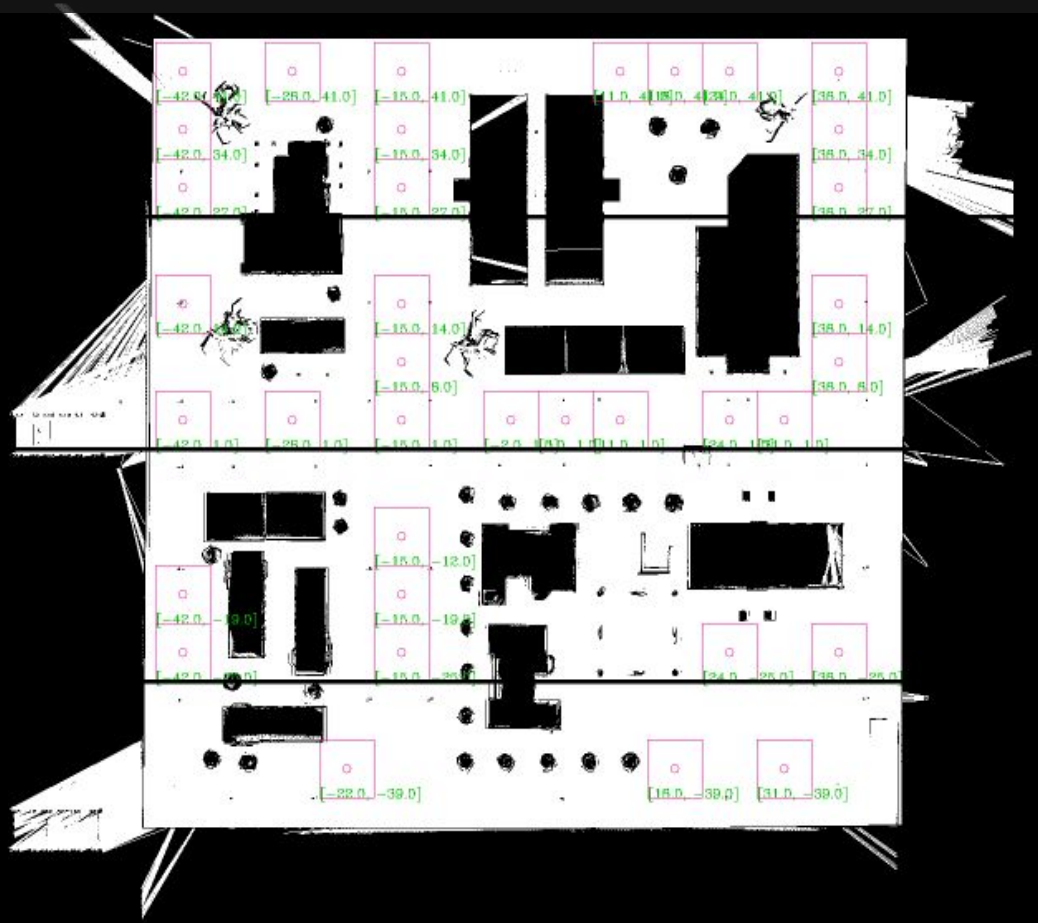
Phase 2 - Known Environment Search



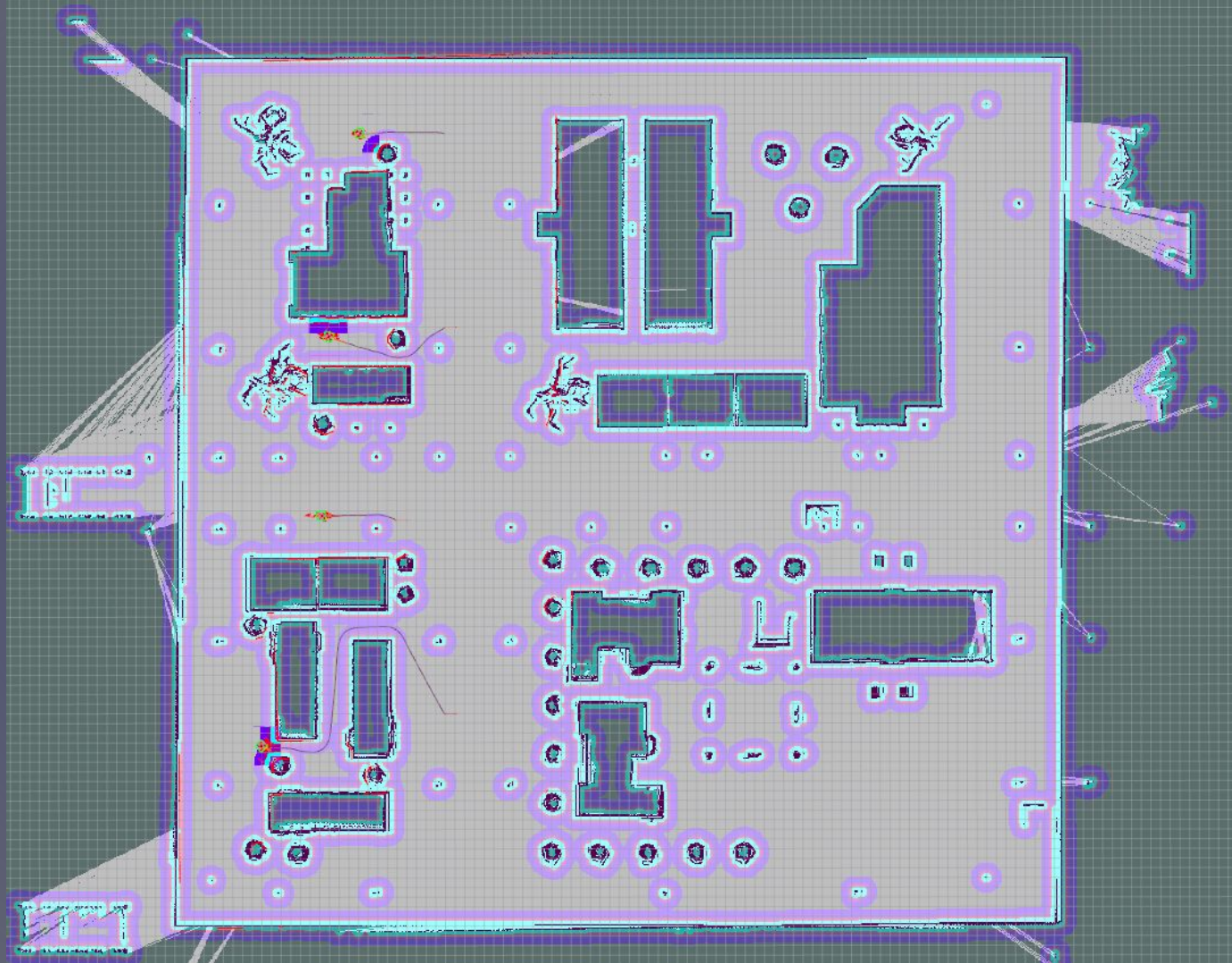
- Probabilistic waypoint generation for all drones
- Waypoints divided among all drones by row
  - Current parameters: 3 rows per drone
- Optional contrast min/max filter for reducing false positives.
- Average Runtimes (Based on timeit library):
  - Without contrast filter: 0.46 seconds
  - With Contrast filter: 54.03 seconds

Waypoint Generation on Small Urban Environment (No Filter)

Waypoint Generation on Small Urban Environment (With Filter)

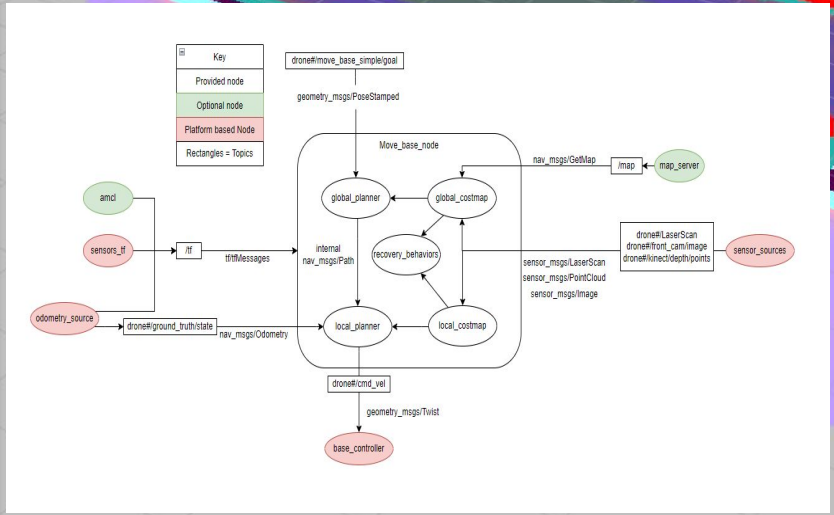


Generated
Waypoints in Rviz



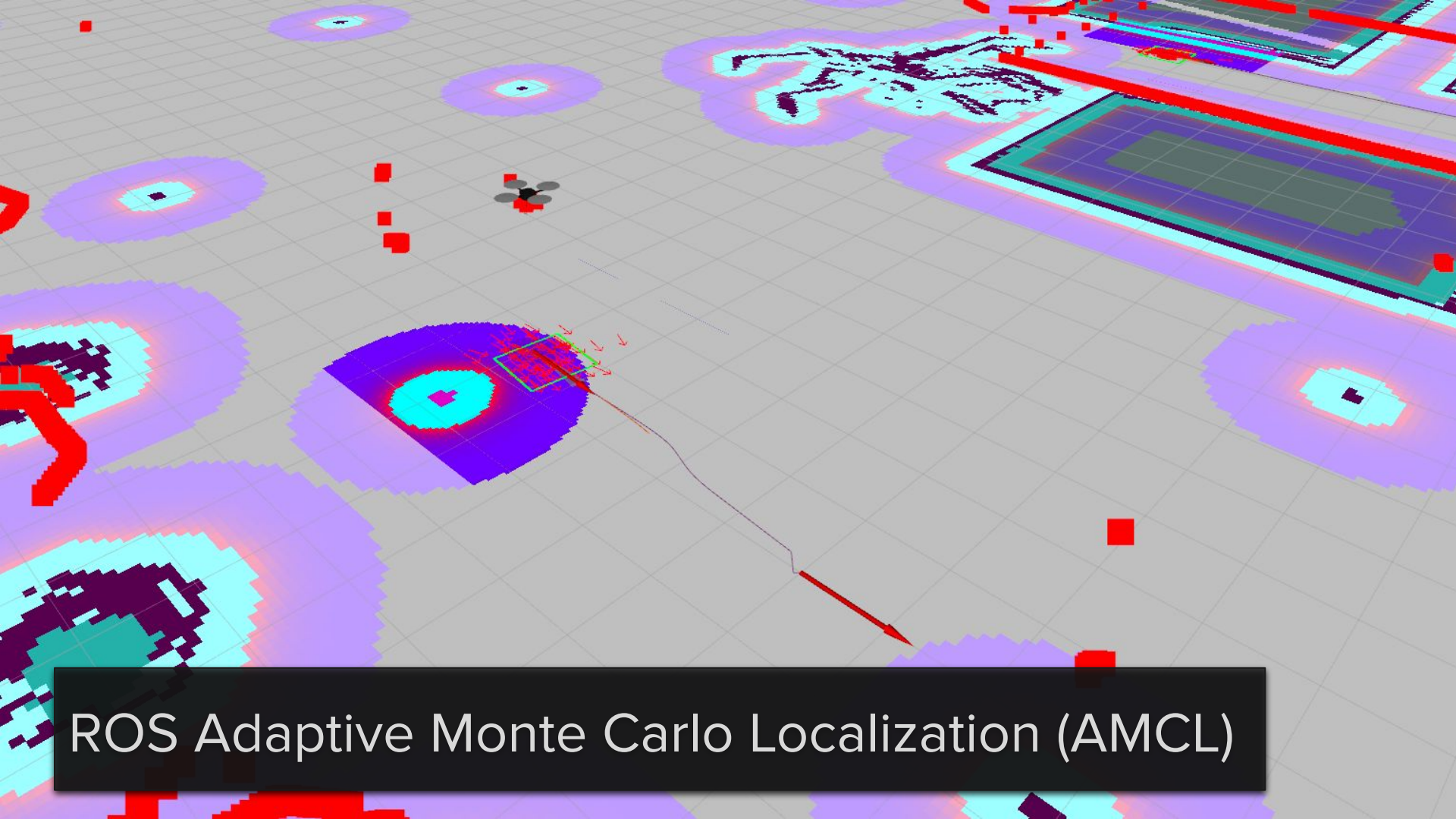
ROS Navigation Stack

- The official recommended setup that is native to ROS.
- Must contain: mapping algorithm, a localization algorithm to use the map created from mapping, and a move_base node.
- When combined costmaps, and local planners are generated to create a path based on dijkstra's algorithm and current odometry.
- Our current setup takes a map drawn via hector_slam and uses AMCL to perform SLAM operations.



ROS Adaptive Monte Carlo Localization (AMCL)

- AMCL uses sampling of a particle filter to track the known position of a robot in a 2D map.
- Essentially what this means is that if we provide an AMCL node a laser scan feed, it will localize the robot based on a known map.
- In Rviz, this node will create what is known as a poseArray. This poseArray is an approximation of all possible poses of the robot in a given local costmap.

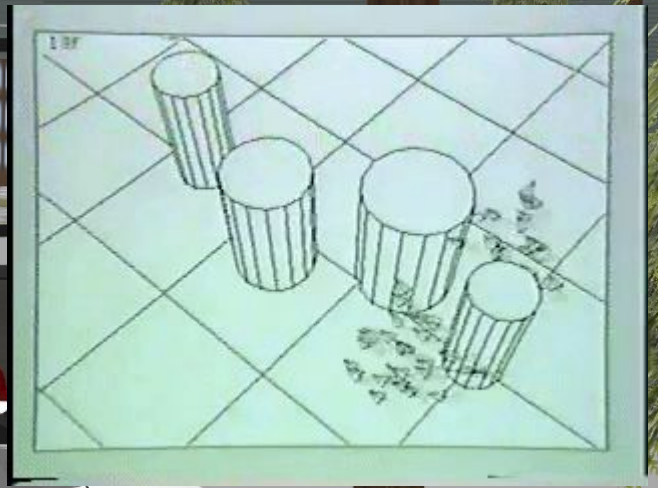

ROS Adaptive Monte Carlo Localization (AMCL)

Swarming Behaviors

- Jakob Germann

Boids

- **Boids:** Artificial life algorithm to emulate the flocking behavior of birds, cells, and other group based lifeforms.
- Developed by Craig Reynolds (1986).
- Defined by 3 rules:
  - Separation
  - Cohesion
  - Alignment



Implementation

- Direct drones to the drone that found BB8
- Used Boid Separation property to avoid collisions on approach

Test Plan

Level 1:

- Checking that each node functions independently.
- Check external API connections.

Level 2:

- Check that ROS nodes are able to connect to one another.
- Make sure topics and message streams are working.

Level 3:

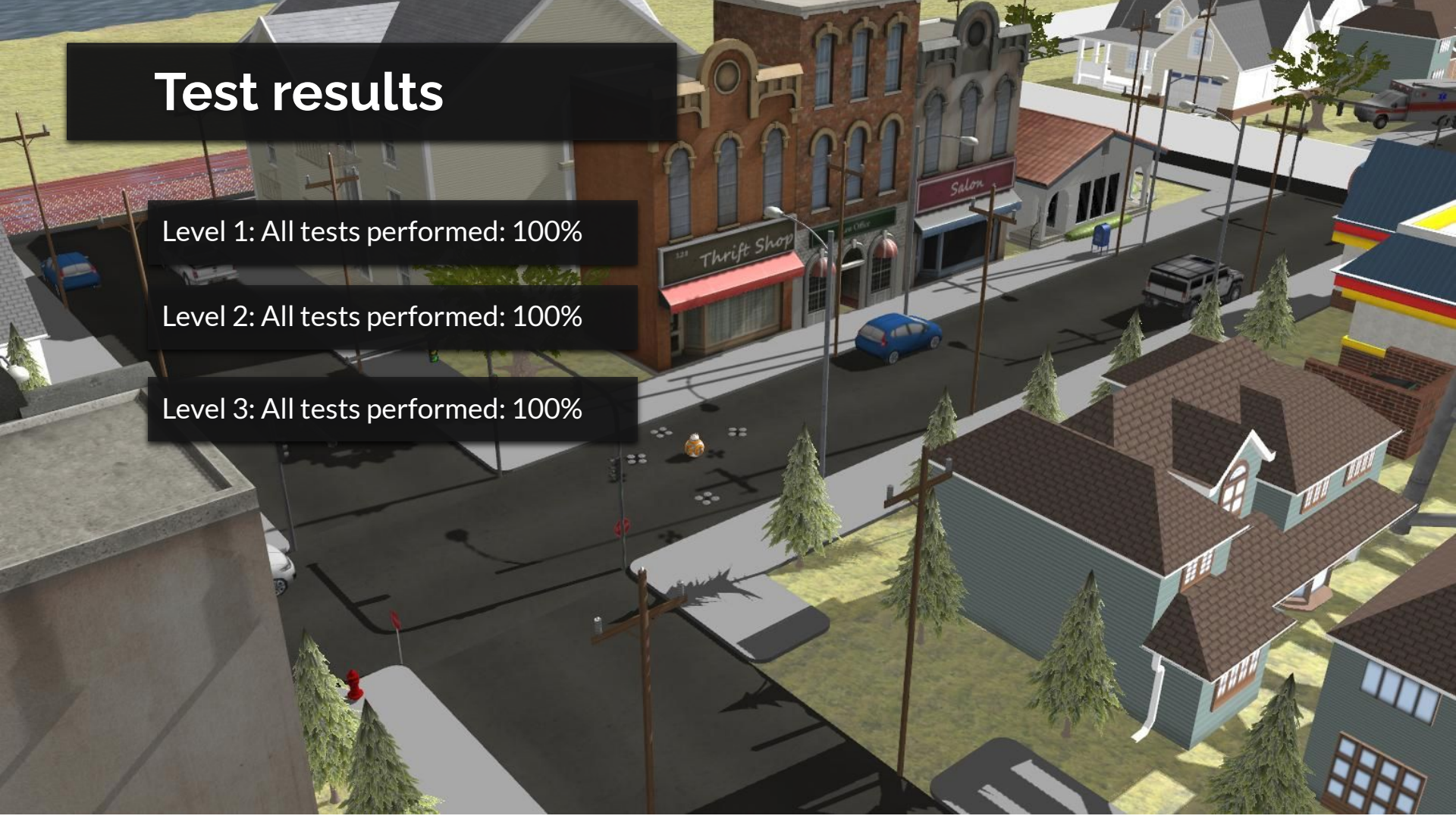
- ROS scenario testing.
- Make sure target is found in the correct amount of time.
- Make sure that drones don't collide.
- Make sure that all other sponsor requirements are met.

Test results

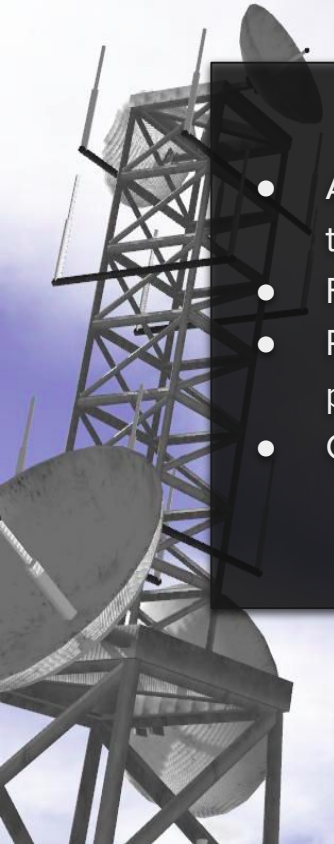
Level 1: All tests performed: 100%

Level 2: All tests performed: 100%

Level 3: All tests performed: 100%



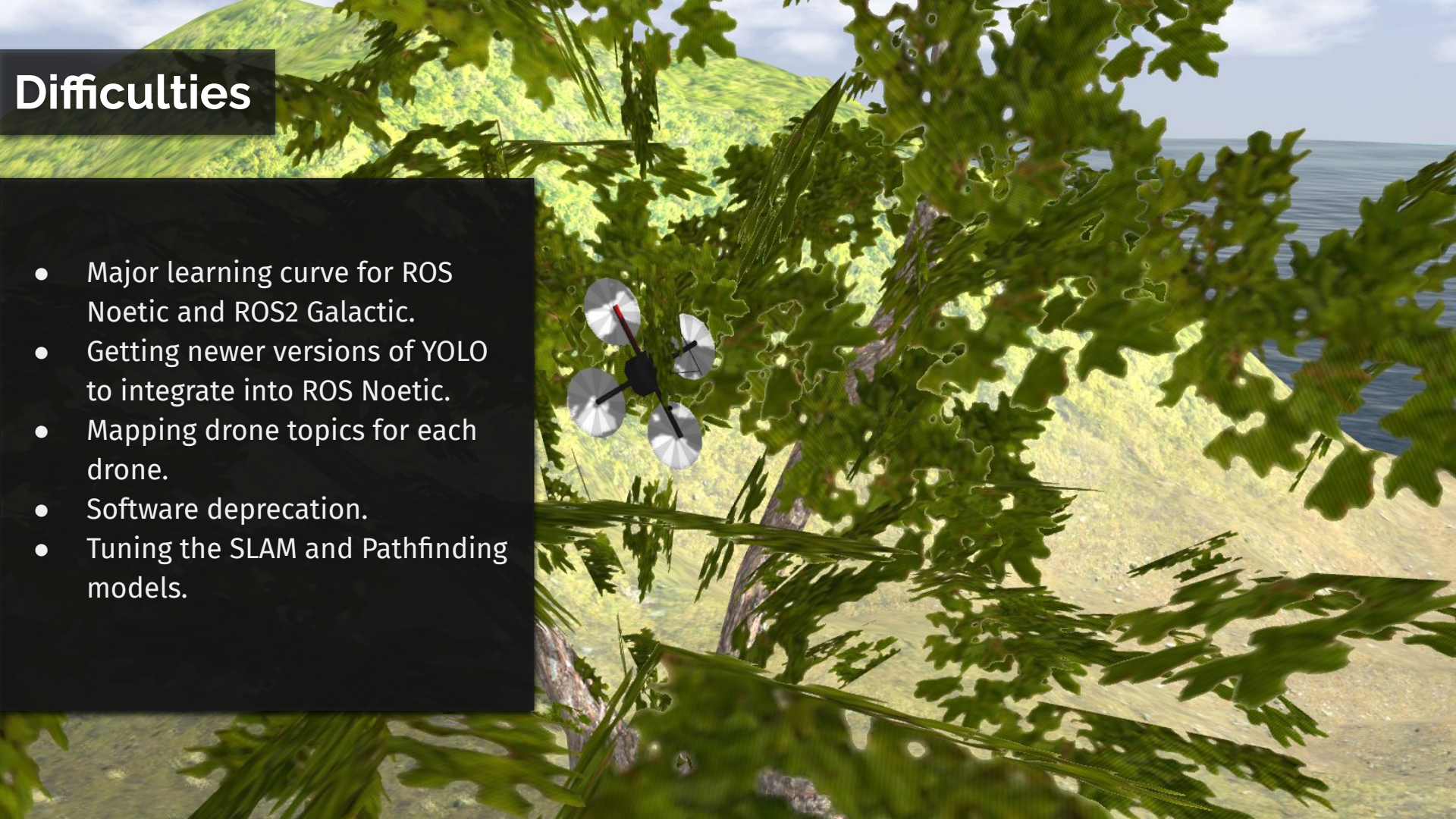
Successes



- Able to launch and navigate a small swarm of four drones throughout the environment
- Frontier exploration generates high detail map autonomously
- Post map generation, system automatically generates probabilistic path for secondary search
- Consistent confidence ratings > 96% for object detection model

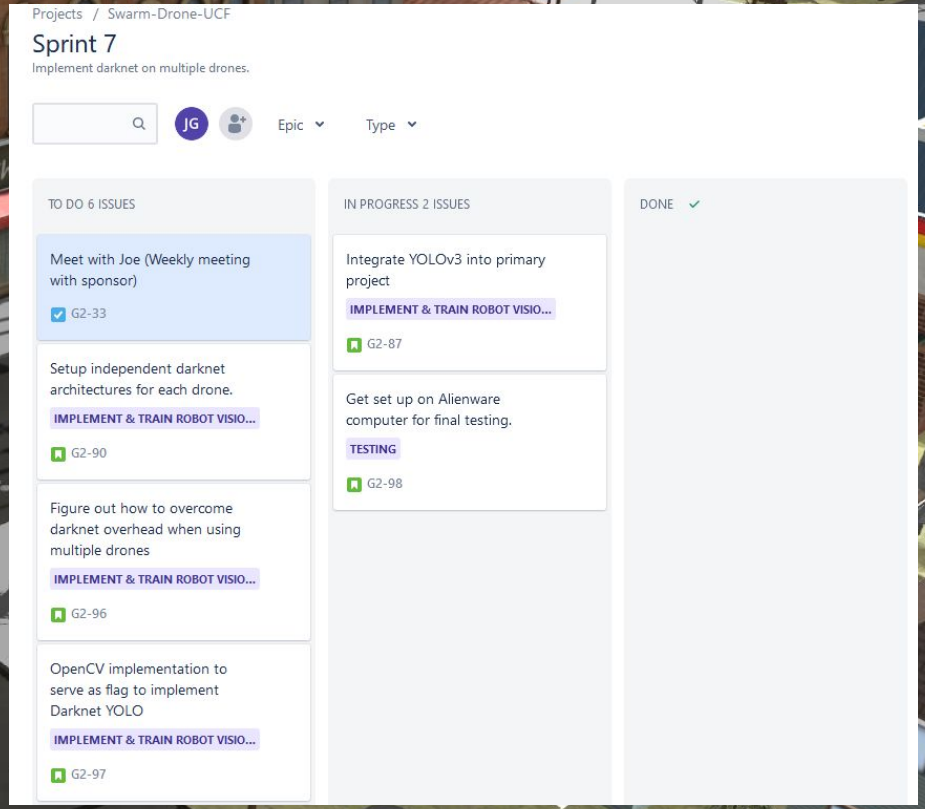
Difficulties

- Major learning curve for ROS Noetic and ROS2 Galactic.
- Getting newer versions of YOLO to integrate into ROS Noetic.
- Mapping drone topics for each drone.
- Software deprecation.
- Tuning the SLAM and Pathfinding models.



Management Methodology

- Paired & Group Programming
- Daily Standups
- Jira for project status upkeep
- Meetings held 1-2 times per week



RoadMap (Gantt Chart)

Cost & Budget

- Sponsor allocated ~~$0~~ $300
- Most software free
  - Construct Sim was $50 a month
- Simulations are running on team's computers/senior design lab's
- Alienware
  - No external hosting required

Construct Sim ROS Courses

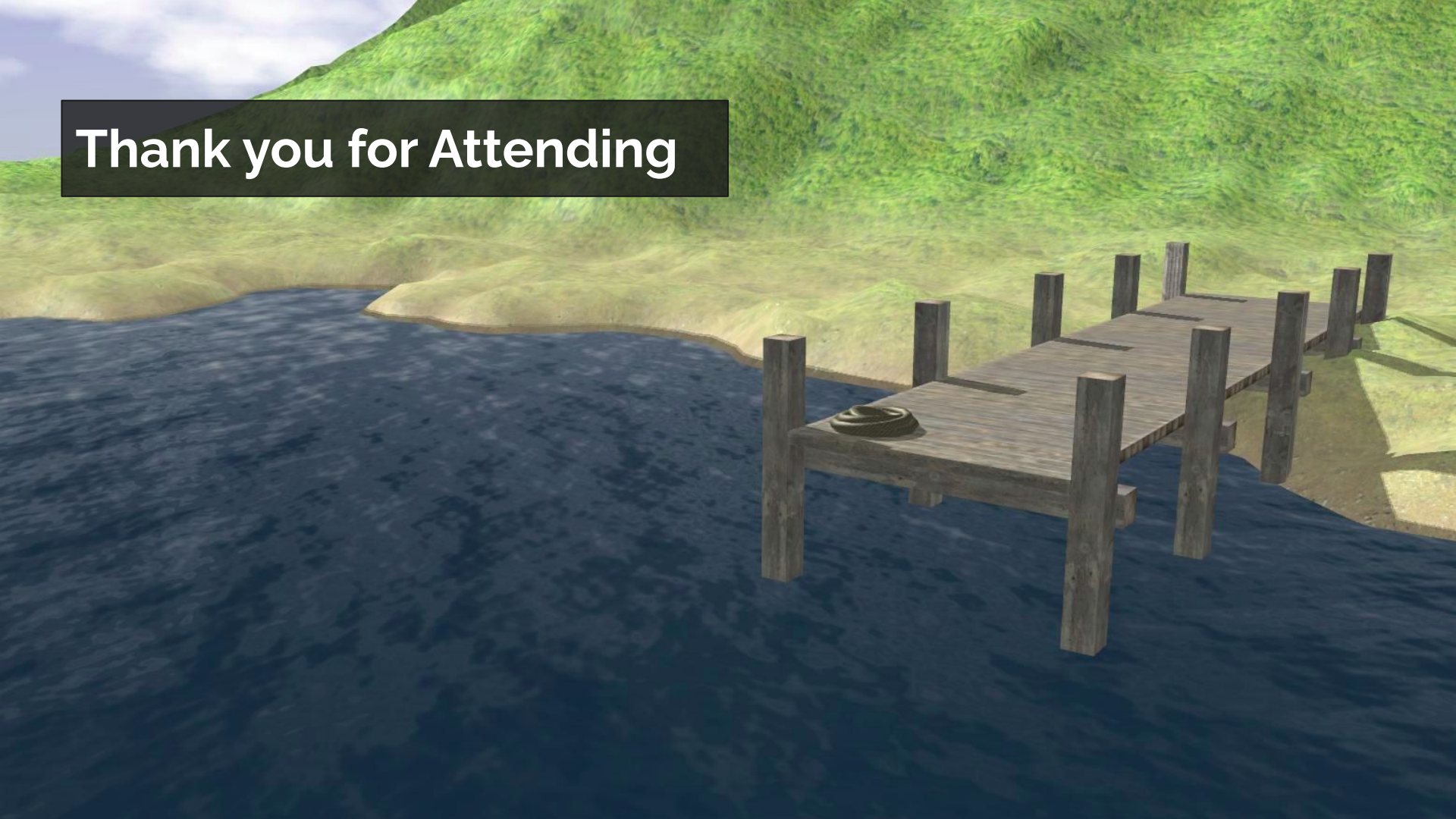
- All $300 used
  - x6 Courses

The Construct ROS

Stretch Goals

- Have BB8 move across the map
  - Incorporate classical Boid properties to track BB8 along its path once found
- Add gimball camera for 3 DOF camera rotation
  - Avoids having to turn entire drone when searching
- 3-d Navigation using Point Clouds instead of Lasers
  - Needs much more processing power
- Optimize waypoint generation when using filter

Project Demo


Thank you for Attending