

Swarm Built Pad



Group 28

Camry Artalona
Hung Nguyen
Richard Malcolm
Stanley Minervini
Coy Torreblanca
Sponsor
Mike Conroy

NASA's Artemis Program



The Problem

- Regolith (lunar dust) in the atmosphere of the moon is harmful to astronauts and spacecraft.
- Need a level surface on the moon before a landing pad can be constructed.

EZ-RASSOR



Goal of the Project

The goal of our project is to successfully have a swarm of rovers level an area of the lunar surface for the creation of a landing pad. We will do this using a framework called ROS (Robot Operating System) and simulate it in Unity.

Requirements

- Create a leveling algorithm which handles dig and dump instructions for each rover.
- Efficiently manage all rovers while algorithm instructions are performed.

Stretch Goals

- Merge with ARM team to place landing pads in the Gazebo simulation
- Demo Sim Stretch Goals:
 - Creating a Unity simulation in which the rovers edit the terrain
 - Decoupling of Gazebo and Unity

Objectives

- Convert surface map image into 2 dimensional matrix.
- Implement leveling algorithm to generate rover instructions.
- Implement central task manager to assign tasks to rovers.
- Create simulation in Unity.

Meeting/Schedule

- Scrum Framework.
- 2-Week Sprints.
- Daily Standups.
- Sprint Planning/Grooming.
- Sprint Retrospectives.
- Daily Sync-ups.
- Weekly meetings with sponsor, Mike Conroy.

Technologies / Languages

- Discord
- Jira
- Github
- ROS1
- Gazebo
- Python2
- Unity
- C#

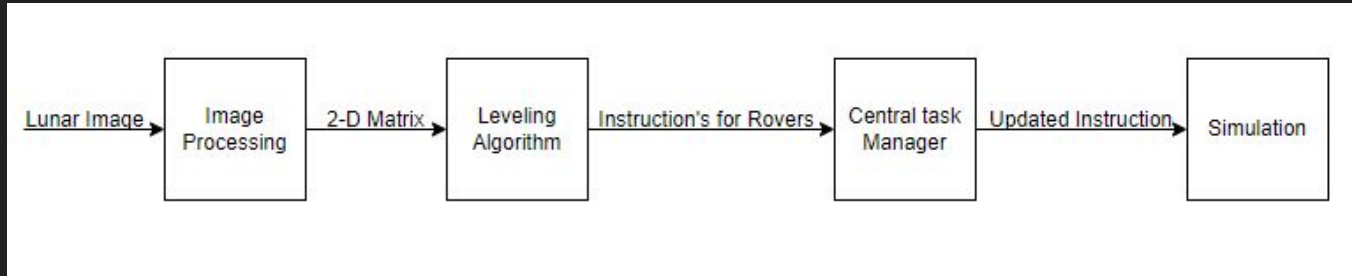


unity :: ROS



Jira Software

Block Diagram of Solution



The Area



Image Processing

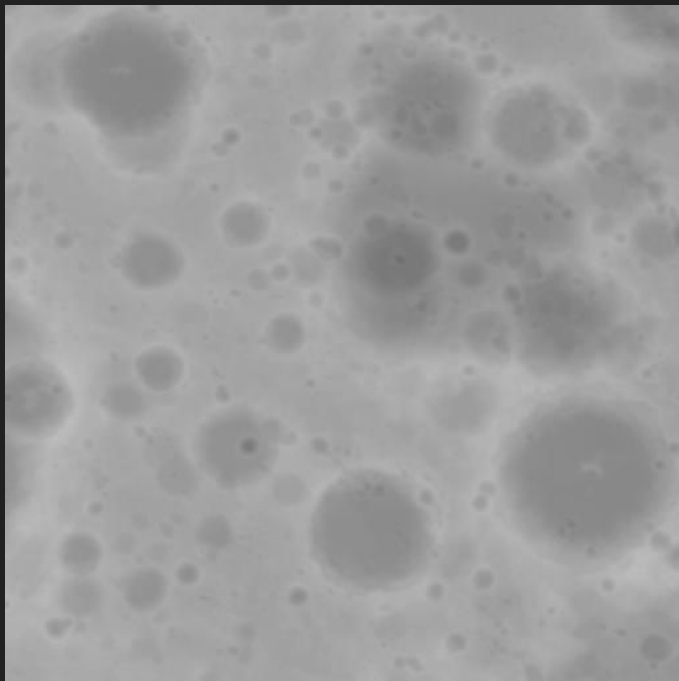
Input - A topographical image of the moon imported into the Gazebo simulation.

Output - A scaled, 2d array of the landing pad area where each value in the array represents the number of actions required to level a pixel.

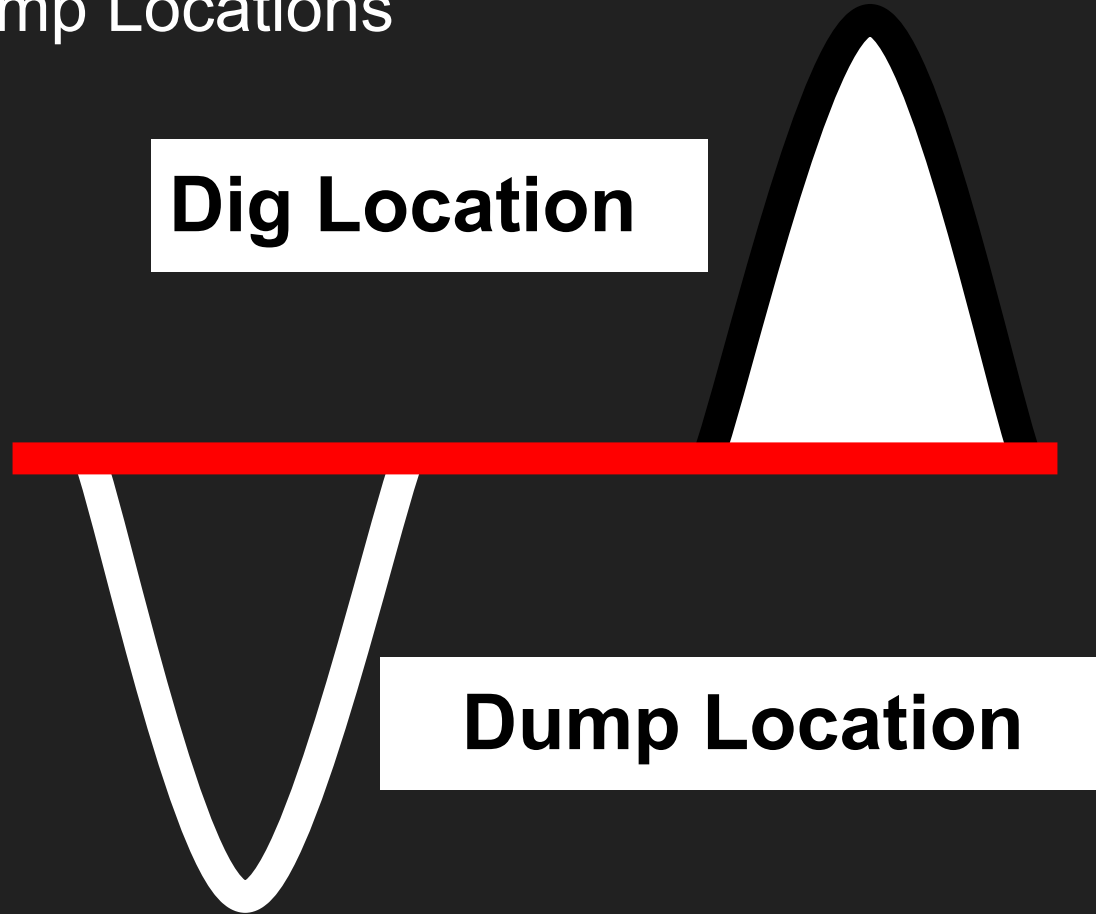
Implementation

- Gray scaled
- Zero scaled
- Pixel expand

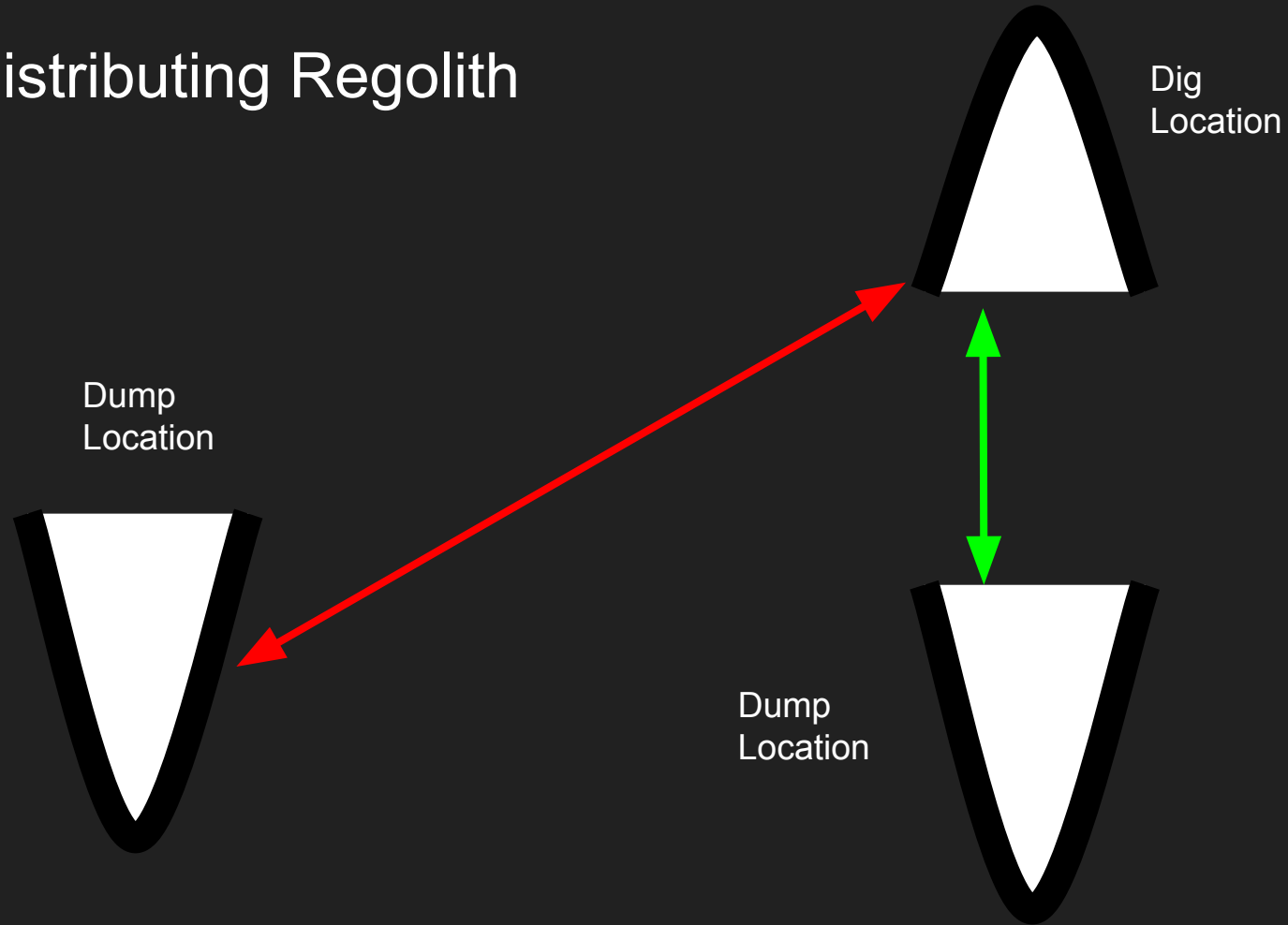
Image Processing



Dig and Dump Locations



Redistributing Regolith



Leveling Algorithm

Leveling algorithm creates a set of instructions which rovers use to level the land efficiently.

An instruction is a hill paired with a hole.

The leveling algorithm uses a greedy approach.

Distance Matrix

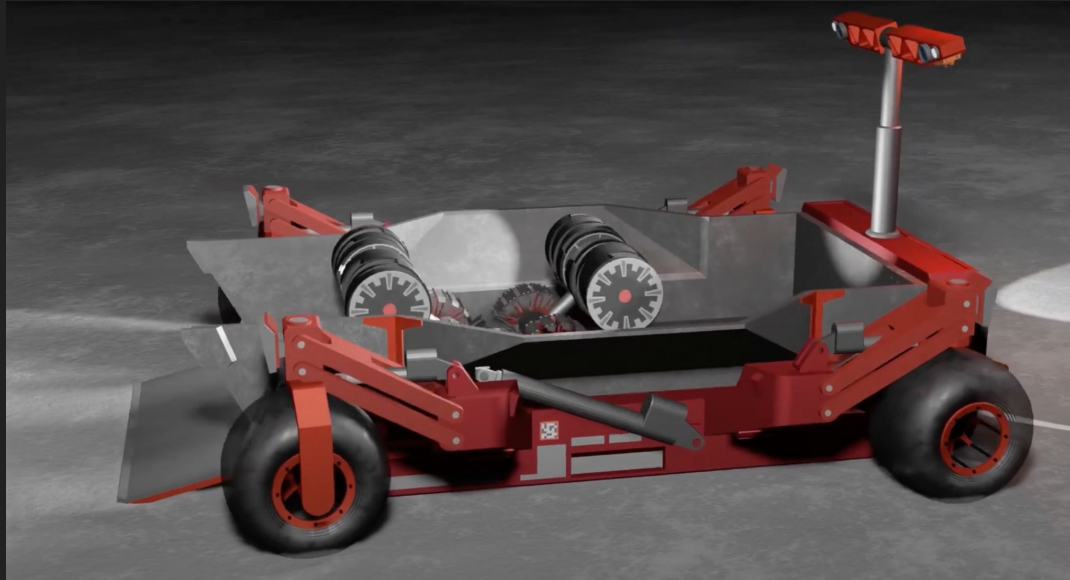
Dig Location 1	Dump Location 1	Dump Location 2	Dump Location 3
Dig Location 2	Dump Location 3	Dump Location 1	Dump Location 2
Dig Location 3	Dump Location 3	Dump Location 1	Dump Location 2

N = Number of Dig Locations
M = Number of Dump Locations

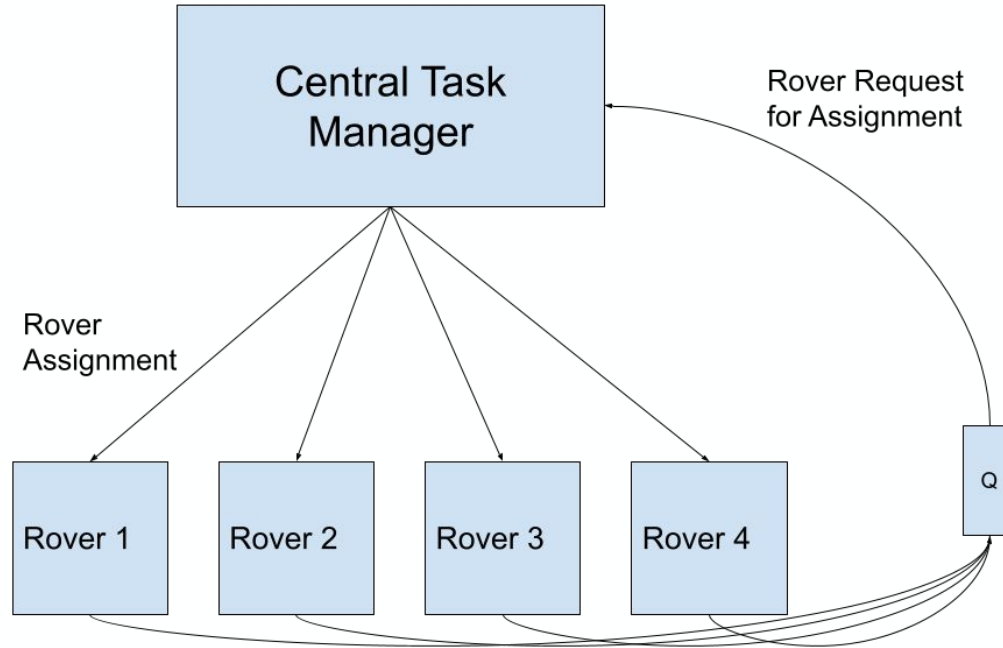
$$O(N * (M + \text{Log } M))$$

The Central Task Manager

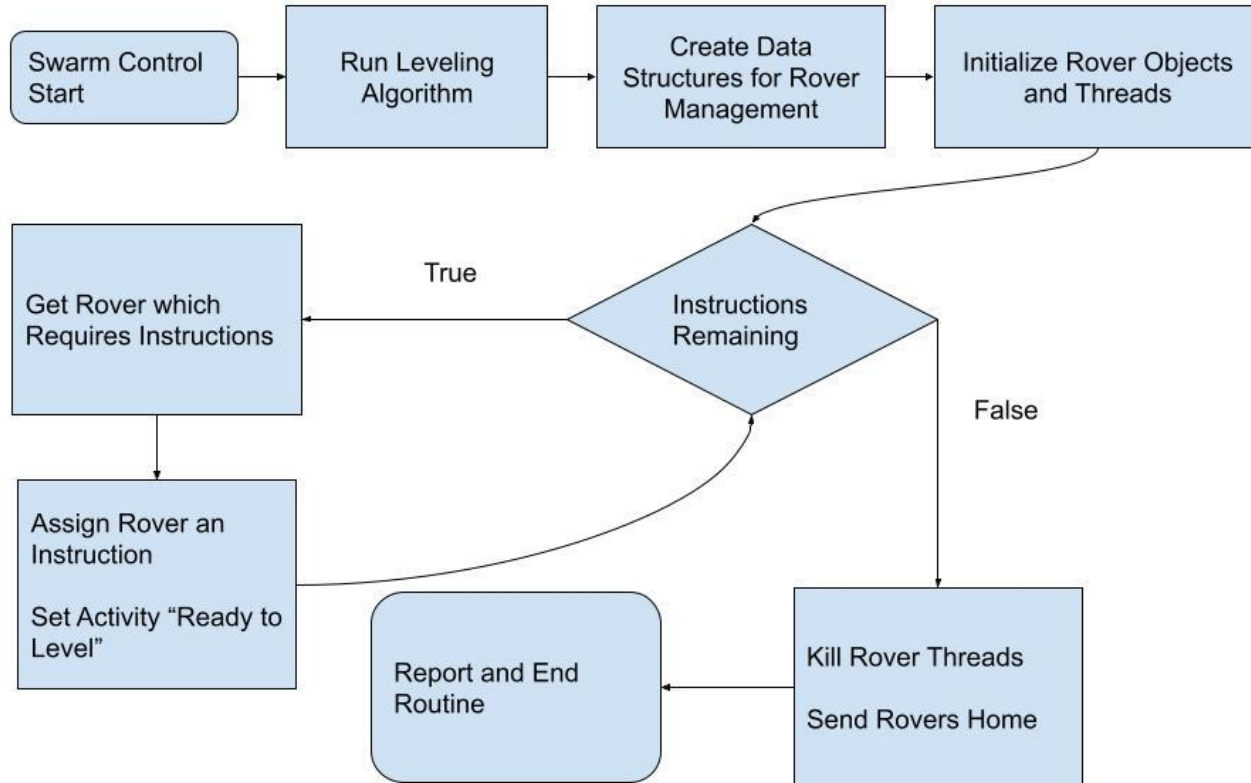
- The Central Task Manager gives rovers instructions created by the Leveling Algorithm.
- The rovers execute the instructions independently.



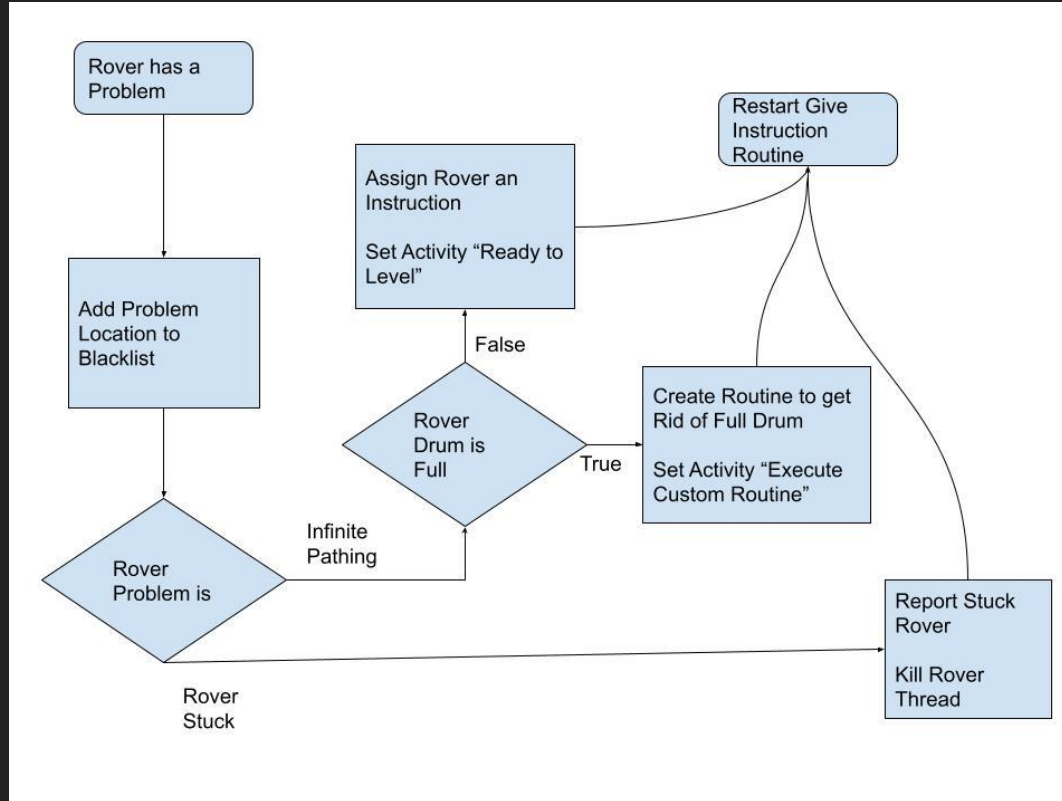
Swarm Controller and Rovers Interface



Central Task Manager

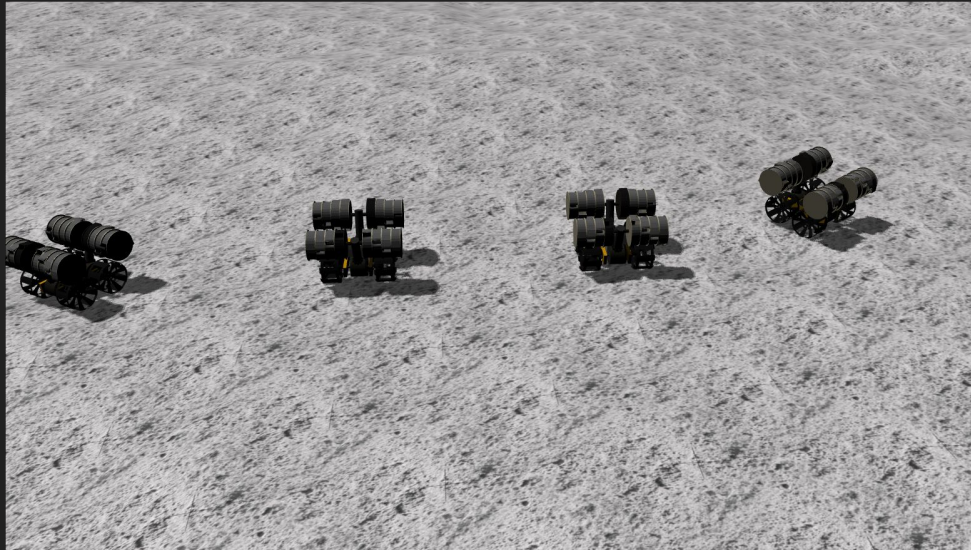


Rover has a Problem

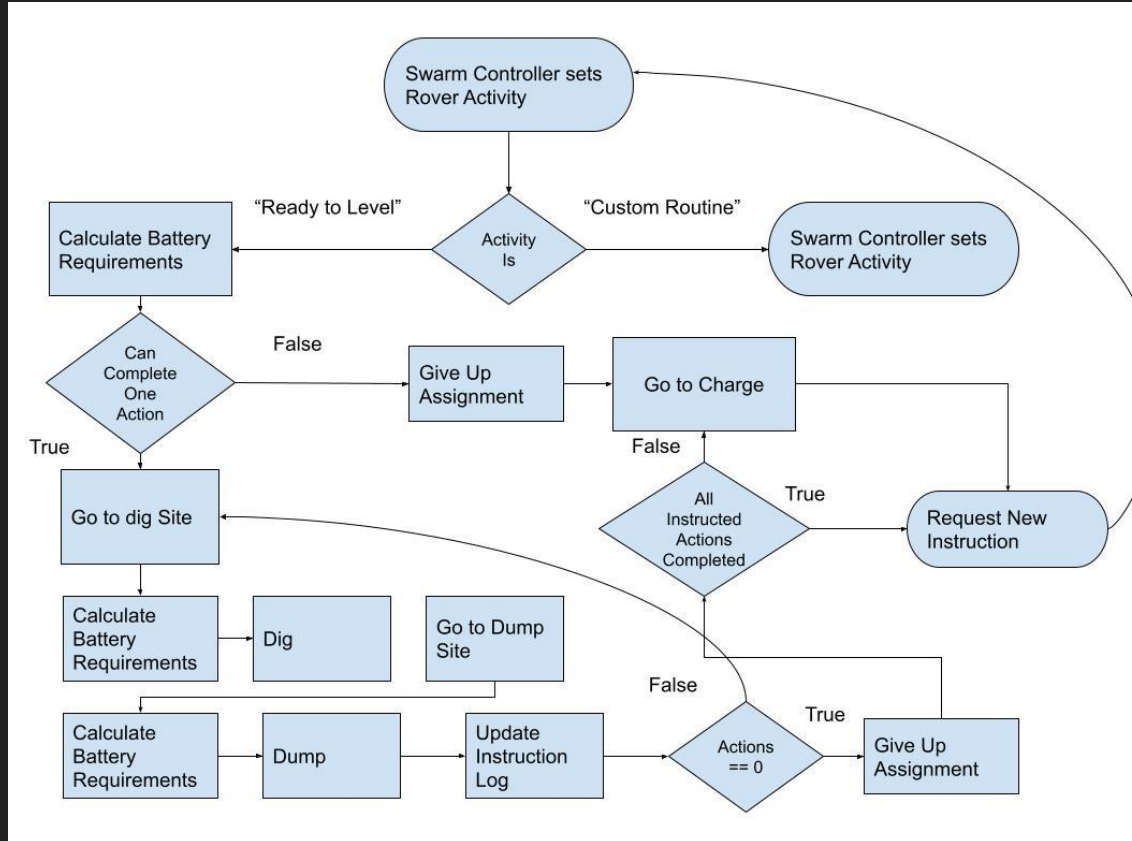


The Rover Logic

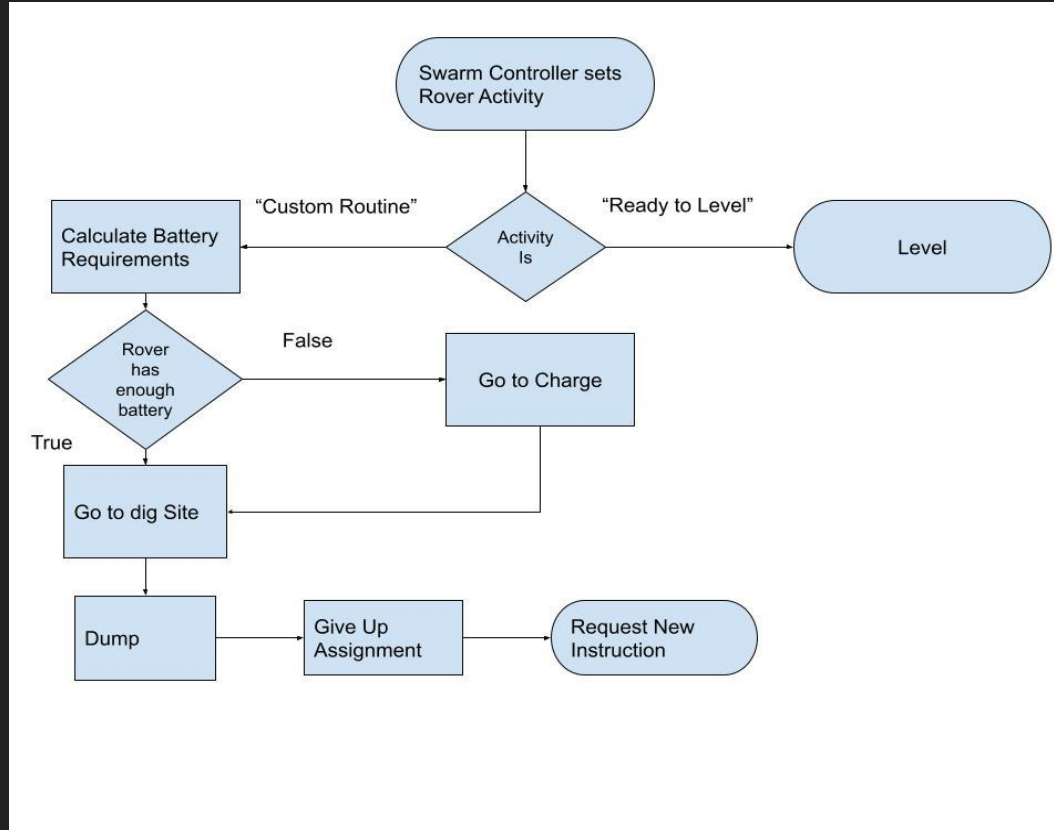
- Logic required to execute rover instructions is encapsulated into a rover class.
- Rover logic can solve problems which may arise at runtime.



Rover Leveling Logic



Infinite Pathing with Full Drum



Simulation

Map: Imported from the EZ-Rassor github, converted to a .raw file format, then used to create terrain in Unity.

Rover Logic: Create Subscribers for each component needed, parse data and handle information properly.

Terrain Manipulation: Rovers are able to edit terrain object's height map in scene once they receive a command via ros to do so.

Division of Tasks

- Camry Artalona - Project lead/Simulation
- Richard Malcolm - Simulation
- Coy Torreblanca - Leveling Algorithm/Central Task Manager
- Hung Nguyen - Scrum Master/Image Processing/Central Task Manager
- Stanley Minervini - Image Processing/Central Task Manager

Successes and Difficulties

Successes

- Team work
- Good communication with FSI teams
- Completed Objectives.
- Completed stretch goal of terrain manipulation in Unity.

Difficulties

- Working with Python2
- Working with ROS (ROS1)
- Working with Gazebo
- Integrating a solution to an already established project

Landing Pads

Demo Simulation

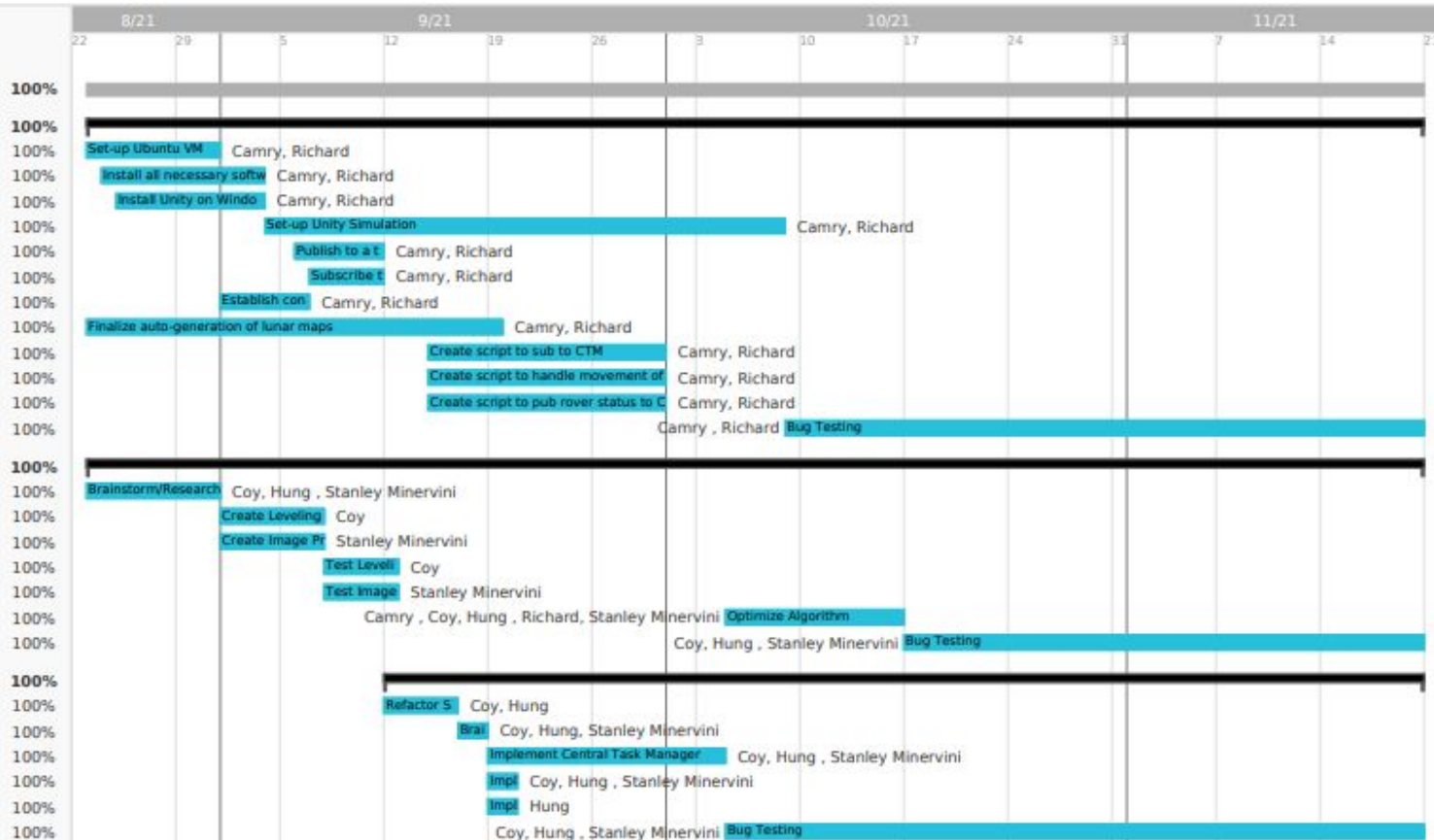
- Set-up Ubuntu VM
- Install all necessary software's on VM
- Install Unity on Windows
- Set-up Unity Simulation
- Publish to a topic from Unity to ROS
- Subscribe to a topic from Unity to R...
- Establish connection from ROS in vm...
- Finalize auto-generation of lunar ma...
- Create script to sub to CTM
- Create script to handle movement of...
- Create script to pub rover status to ...
- Bug Testing

Leveling Algorithm

- Brainstorm/Research Leveling Algori...
- Create Leveling Algorithm
- Create Image Processing
- Test Leveling Algorithm
- Test Image Processing
- Optimize Algorithm
- Bug Testing

Central Task Manager

- Refactor Swarm Control
- Brainstorm Swarm Management
- Implement Central Task Manager
- Implement Leveling Algorithm
- Implement Image Processing
- Bug Testing



Budget

- \$0.00

Summary

- Leveling Algorithm
 - The Leveling Algorithm creates the set of instructions which rovers use to level the area efficiently.
- Central Task Manager
 - The Central Task Manager assigns rovers instructions and handles rover problems if encountered.
- Rover Logic
 - Rover logic controls a rover to execute an instruction.
- Unity Simulation
 - Implemented terrain manipulation
 - Rovers receive instructions from ROS on Ubuntu
 - Map scaled correctly

/home/stanley/ezrassor_ws/install/share/ezrassor_launcher/launch/swarm_simu...

File Edit View Search Terminal Help

```
[INFO] [1637298802.697902, 611.200000]: Client /ezrassor1/waypoint finished sending waypoints!
[INFO] [1637298802.986933, 611.410000]: Rover 1 dumping.
[INFO] [1637298803.006587, 611.420000]: Client /ezrassor1/waypoint received dump command!
[INFO] [1637298803.024161, 611.430000]: Waypoint server /ezrassor1/waypoint executing dump command
[INFO] [1637298850.343855, 645.260000]: Client /ezrassor1/waypoint reached waypoint! (position: (2.92, -1.33) battery: 13)
[INFO] [1637298850.350914, 645.270000]: Client /ezrassor1/waypoint finished sending waypoints!
[INFO] [1637298850.581316, 645.440000]: Rover 1 has completed possible actions.
[INFO] [1637298850.586552, 645.440000]: Looking at tasks for Rover 1
[INFO] [1637298850.587233, 645.440000]: Rover 1 queued.
[INFO] [1637298850.642704, 645.480000]: Assigned locations to Rover 1.
[INFO] [1637298850.656300, 645.490000]: Rover 1 beginning leveling
[INFO] [1637298850.671943, 645.510000]: Rover 1 is calculating battery requirements.
[INFO] [1637298850.722944, 645.540000]: Searching for path from (3, -2) to (5, -2)
[INFO] [1637298850.731749, 645.550000]: Path found in 0.000750064849854
[INFO] [1637298850.740743, 645.550000]: Searching for path from (5, -2) to (4, -1)
[INFO] [1637298850.750055, 645.560000]: Path found in 0.000508069992065
[INFO] [1637298850.754824, 645.570000]: Searching for path from (4, -1) to (2, 0)
[INFO] [1637298850.767738, 645.580000]: Path found in 0.000523090362549
[INFO] [1637298850.833774, 645.610000]: Rover 1 cannot complete any actions. Going to charge.
[INFO] [1637298850.900860, 645.660000]: Searching for path from (3, -2) to (2, 0)
[INFO] [1637298850.909167, 645.660000]: Path found in 0.00119090080261
[INFO] [1637298850.917734, 645.670000]: Client /ezrassor1/waypoint received path from (3.0, -2.0) to (2.0, 0.0)
[INFO] [1637298850.921721, 645.680000]: Rover 1 driving to lander at (2, 0)
[INFO] [1637298850.946974, 645.680000]: Waypoint server /ezrassor1/waypoint moving rover to (3.0, -1.0)
[INFO] [1637298859.376570, 651.720000]: Client /ezrassor1/waypoint reached waypoint! (position: (3.16, -1.15) battery: 13)
[INFO] [1637298859.406062, 651.740000]: Waypoint server /ezrassor1/waypoint moving rover to (2.0, 0.0)
```

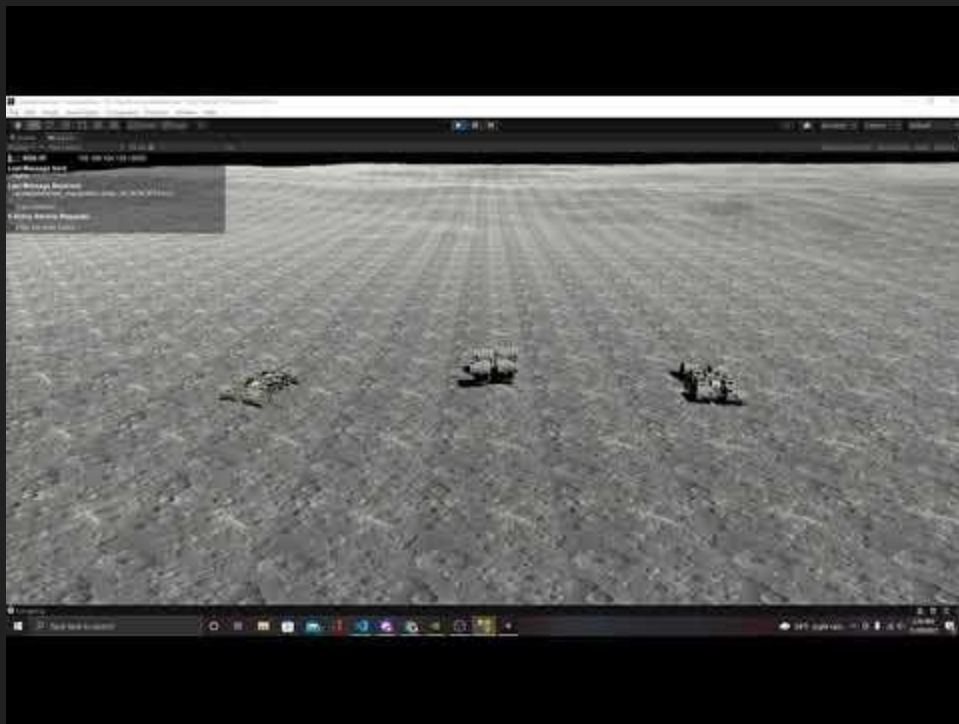


Sim Time: 00:10:51.770 Real Time: 00:07:25.835 Iterations: 25161

FPS: 25.43

Reset Time

Unity Demo



Unity Demo

