NASA Robotic Mining Competition

Taylor Ertrachter Bailey Hamant James Spies

Faculty Advisor: Dr. Marius Silaghi

Client

Our client is the Aerospace NASA Lunabotics Engineering Competition(LEC) team division.

Our boss is Dr. Kimberly Demoret, the Aerospace team advisor.

Meeting Schedule

We meet once a week on Mondays at 5:00pm. At 5:30pm Dr. Demoret joins our meetings.

We may also have Software subteam meetings to discuss software specific questions. These meeting include the 3 of us, as well as Steven, our subteam lead, and sometimes Nathaniel, the overall NASA LEC team lead.

Goals and Motivations

Partnered with Aerospace majors, our goal is to work on developing the software of the robot to be entered in NASA's Lunabotics Engineering Competition.

Approach

We intend to first create a virtual simulation and test environment to run the software on, both to avoid damaging the actual hardware and to be prepared in the event that the hardware is not finished by the date of the showcase.

Approach

Our system will implement these key features:

1. Manually controllable robot actions for all major motors (navigation, claw, conveyor belt) through Xbox wireless controller
2. Object detection for future autonomous navigation
3. Fully autonomous sample collection, navigation, and deposition system using manual controls as backup

Novelty Feature

Having an autonomous robot is the key feature of this project. Implementing an AI to a robot designed to navigate an obstacle course and excavate minerals will be very a very interesting and challenging goal.

Technical Challenges

- Client has requested that the code be written in Python for readability purposes, meaning that all of our controls for the robot will require libraries that we are unfamiliar with
- We are using APIs that we are unfamiliar with for the object detection camera and motor controls
- We have limited experience with object detection and automation
- Online learning

Milestone One

- Create a virtual environment in which we can test robot controls on a simulated level.
- We will be able to work on code for specific functions even if the hardware has not been finished for it.
- We want this simulation completed ASAP so we can get visuals on how the code is running.

Milestone Two

- Implementing Python controller for robot.
- Implementing an operable arm controller based on controller input.
- Implementing a moving conveyor belt.

Milestone Three

- Implementing the camera's vision into the code.
- Implementing the camera code into the robot's actions.
- Implementing different actions based on different objects.