Robotic Mining Competition - Software Testing Plan

Liam Sapper (lsapper2020@my.fit.edu)

Faculty Advisor: Dr. Marius Silaghi (msilaghi@fit.edu)
Client: Robotic Mining Competition team, NASA
Meeting Times: Wednesdays, 4:00pm - 5:00pm; Fridays, 3:00pm - 3:30pm

Simulation Software:

For most of the software testing, a simulation program will be used: Webots, by Cybernetics. Webots is a virtual simulation environment that gives the user a real-time graphic of the robot being tested to show how it acts. This software was the most accessible and cost-efficient option for the RMC team. However, since the team will be using Fusion 360 as their modeling software for creating the robot's design, if the education edition allows for simulation within the Fusion 360 software, we may switch to using that instead.

Requirements and functions will be done with unit testing. This means that smaller parts of functions are broken down and tested before they are put together, and then tested together. Once all functions are tested, the whole class will be tested. Finally, classes will be tested together to double check their compatibility with each other. The size of units of each test may vary depending on the complexity of the functions. A function may not need to be broken up at all.

The terms waypoint and navigation point will be used interchangeably.

Functional Requirements Testing

1. Test cases: Prolonged button pressing; quick repeated presses; pressing different buttons; pressing every input all at once.
2. Test cases: Create sample navigation points for the simulation to traverse to, observe where it goes; introduce different obstacles in different places; send incorrectly formatted navigation points.
3. Test cases: Use different number of waypoints on each run; use different waypoint locations; have two waypoints at the same location; give incorrectly formatted points.
4. Test cases: Prolonged button pressing; quick repeated presses; pressing different buttons; press buttons for movement and digging at same time; try to dig when full; unload at any time;
5. Test cases: Multiple button presses; multiple dig zone waypoints; don't include any dig waypoint; don't include unload waypoint and have go to dig waypoint again.
6. Test cases: Press button multiple times; Press and try to control robot again; Press while robot is working autonomously.

Interface Requirements Testing

1. Test cases: Turn on and off.
2. Test cases: Set starting point at 0,0; set starting point somewhere other than 0,0;
3. Test cases: Give 1 waypoint too track; give multiple waypoints to track; Give incorrectly formatted waypoints.

Performance Requirements Testing

1. Test cases: Test bandwidth without any limiters; test bandwidth with limiters;
2. Test cases: Use as much as possible then try to go over bandwidth;