

WiRom Exercise

13/11-22 Simen Farbu Swensen 595487 Link to Gitrepo:

<https://github.com/H595487/WiRoM2.0>

One of my biggest hobbies is playing videogames, and as I have continued my studies in programming I have definitively gotten more interested in how automation of npc's etc work, as I have also understood more of how code works. I often find it interesting when playing games, to try to test the limits of physics engine that games run. The easiest way of doing this is often to see what happens when 2 movable objects interact (crash into each other etc). Sometimes the effect simulates reality pretty good, sometimes not. I noticed a lot of room for creativity in this task, so I wanted my custom missions to (in addition to what's required in the assignment) test this in the WiRom sim.

The first mission just made the bots crash together, and tested simple actions like move to point and go forward. The second mission added more tasks for each robot and I played around with the code in "original_task_allocation.py" to try to avoid the robots getting stuck. Video of execution of both missions (mission 2 had to be split in 2 vids) are available in git repo.

By editing the "calculate_utility" method in the "OriginalTaskAllocation" class it was interesting to see changes in task allocation as well.

The screenshot displays the WiRom web interface for configuring missions. It features several panels: 'Missions' at the top with tabs for 'Quarantine delivery', 'Scout area and deliver item', 'Test All Forward', 'Package Delivery (!)', 'Giga Mission (!)', and a 'Test' button; 'Tasks' on the left with a list of tasks like 'Drive to barrel', 'Drive to Moose path', and 'Crash into Per & bb8', each with a dropdown for robot assignment and a red 'X' icon; 'Simpleactions' in the center with a list of actions like 'Go to location' and a search bar; and 'Simpleactions' on the right with a list of actions like 'Go backward' and a search bar. The interface is designed for creating and testing custom missions in the WiRom simulation.

I did add another moose robot that was named "Per", and I found this operation to be very intuitive when I used the DSL editor in the web interface.