Group 10: Factory Simulation

Devin Shingadia | 170036317

Jacob Williams | 170050173

Mohammed Hamza Zaman | 170107477

Miraj Shah | 160050110

Christos Dolopikos | 170116343

Vivek Bhukhan | 159087932

Abstract:

This report will provide an in-depth view of the factory simulation that we have developed, highlighting, how the behaviour has been implemented, and the reaction of the simulation to various parameters, inputted through a ‘.SIM’ file. It will be structured as follows: there will be an outline of the various behaviours that have been implemented, followed by a conclusion which summarises the behaviour of the entities that we have created.

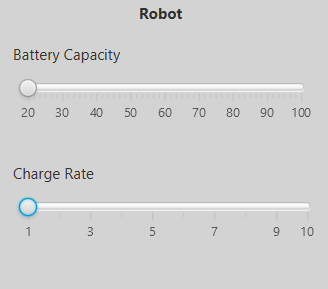
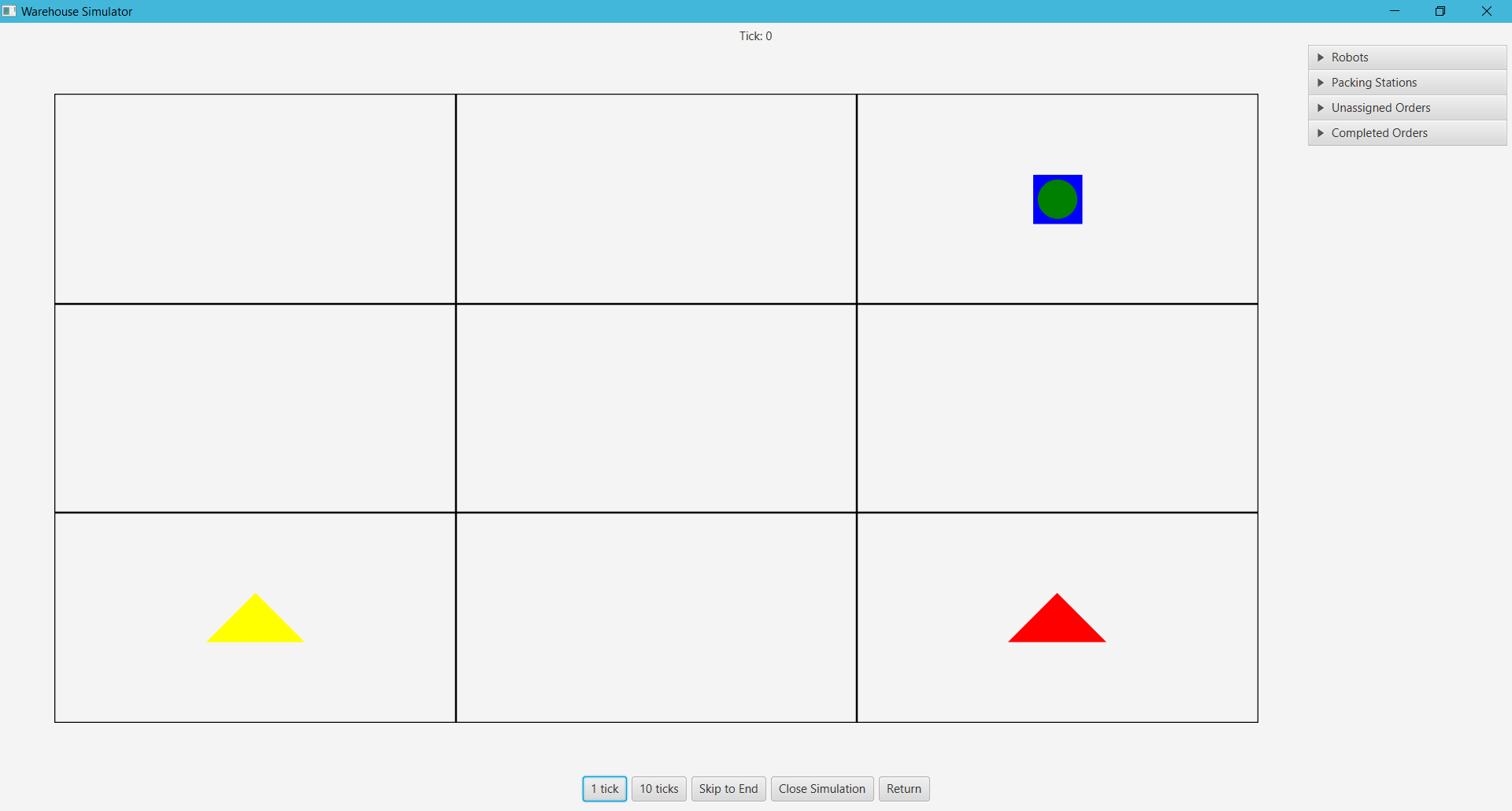
A report on what happens in the simulation as you change the different parameters. For instance, how do your robots react to configurations that have more overlap between their usual paths? How much of a safety margin do you need for the batteries in your robots?

# Tests

## Comparing the number of ticks required to carry out the orders (File: One of Everything)

**Battery Capacity: 20**

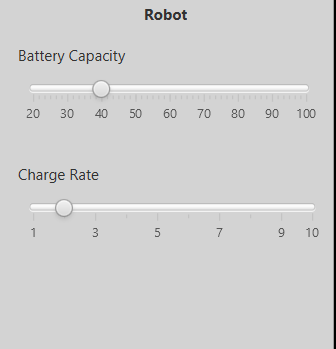
**Charge Rate: 1**



.SIM file used: OneOfEverything.SIM

Result : With the lowest battery capacity and charge rate, the number of ticks required to carry out this activity was 149.

## Modifying the battery capacity and charge rate (File : OneOfEverything.SIM)

****

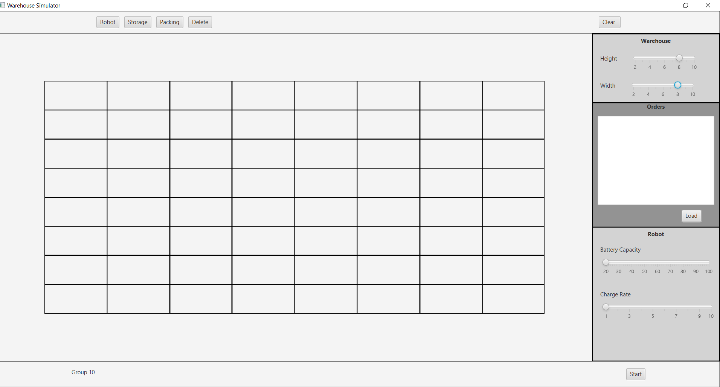
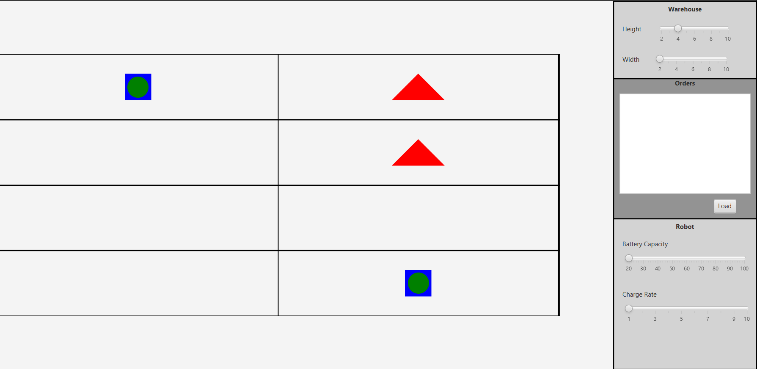
**Battery Capacity: 40**

**Charge rate: 2**

In theory, the robot should be able to carry out work in fewer ticks as it does not require to charge as often. Furthermore, whenever it does need to charge, the amount of time required to charge should be halved, as a result of doubling the charge rate.

Result: The number of ticks required to carry out the orders specified in ‘OneOfEverything.SIM’ is 149. This indicates that after loading a file, the modification of the battery capacity and the charge rate does little to affect the performance of a robot when it is carrying out the orders.

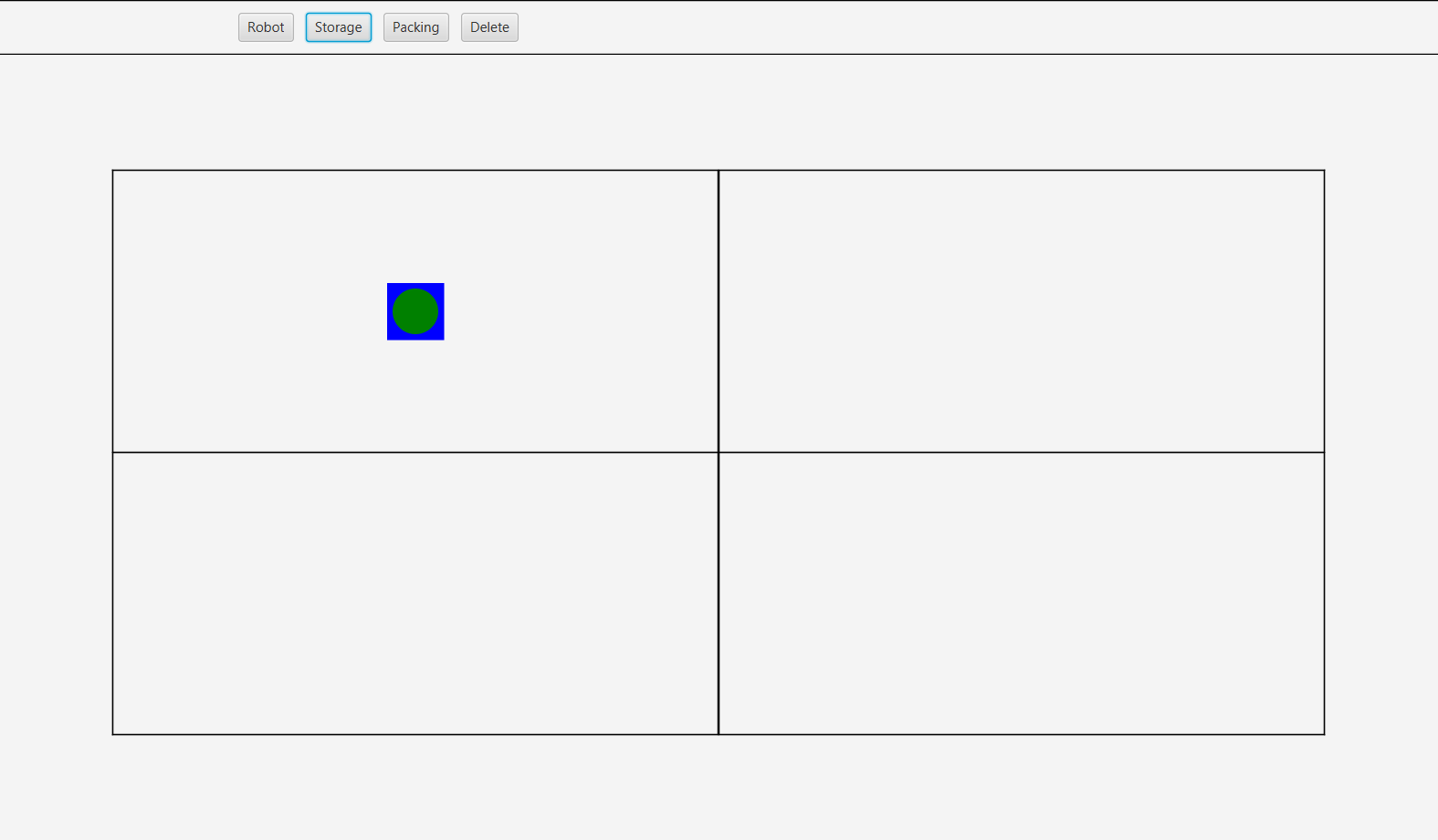
## Attempting to resize the grid with entities on the grid

Prior to resizing: After resizing:

This test shows that when resizing the grid, the entities that have been placed on the grid are removed. The implications of this is that a user may need to redraw the entities on the grid if they accidentally change the

## Attempting to place two entities on the same cell in the GridPane.

In this test, I select the Storage shelf and attempt to place it into a cell that current contains a robot entity.



Result: The storage shelf would not be placed into the same cell as the robot entity. This shows that the simulation does not allow two entities to be placed into the same cell in the grid. As a result the simulation works as intended, and some user error has been taken into account.

# Discussion of the results

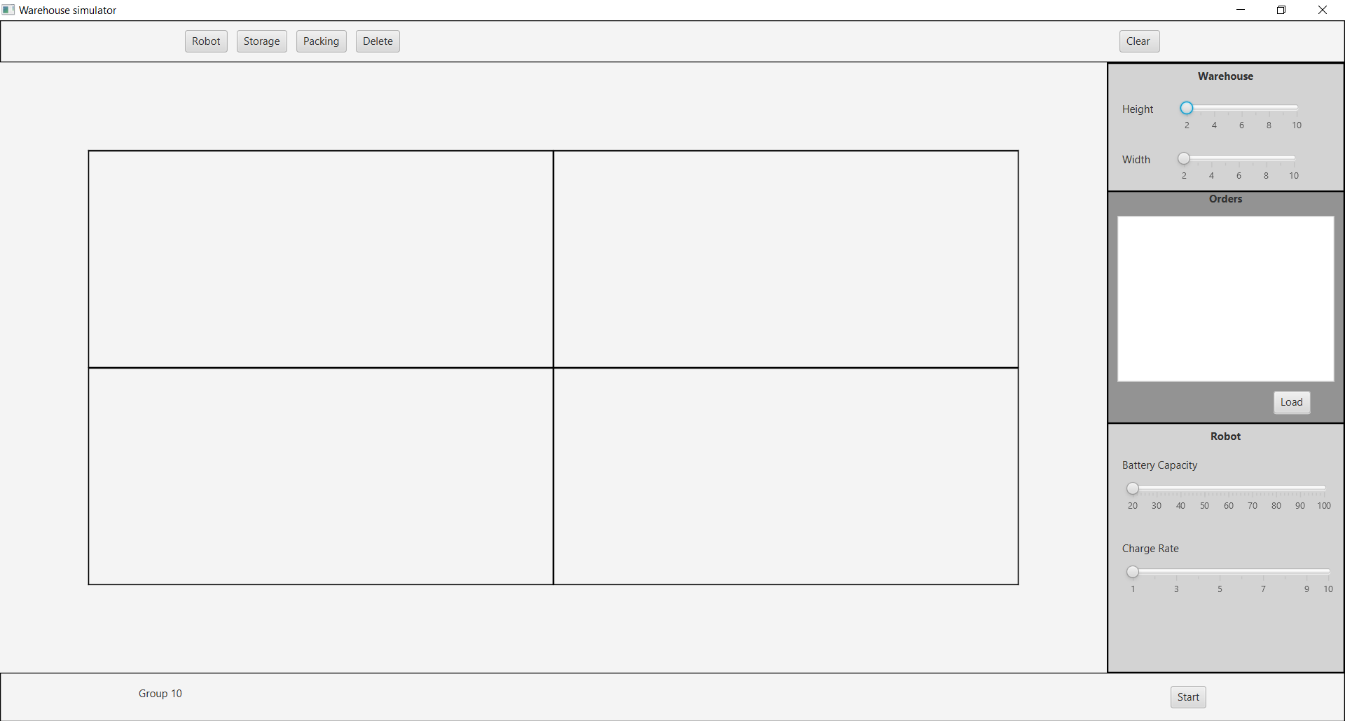
The results of the experiments show that most of our system is working as intended. Every feature that has been experimented works to some extent. Some features, such as the effect of the battery capacity and charge rate on the number of ticks required to carry out the simulation, do not work as intended, meaning that the charge rate and battery level does not have as much as an effect on the number of ticks consumed, as compared to the number of orders given to the order. The tests have shown us that there are some inadequacies in our code, as some parts do not perform the function that is intended, highlighting gaps in our knowledge.

# Main view

Add/Delete entities to the grid. Adding these entities will also create distinct objects.

Clear the grid

Adjust height and width of grid



‘.SIM’ File displayed here.

Load the ‘.SIM’ FIle

Set the battery capacity of the robot.

The grid is where entities can be added and removed. It can also show a preview of the simulation before it is run.

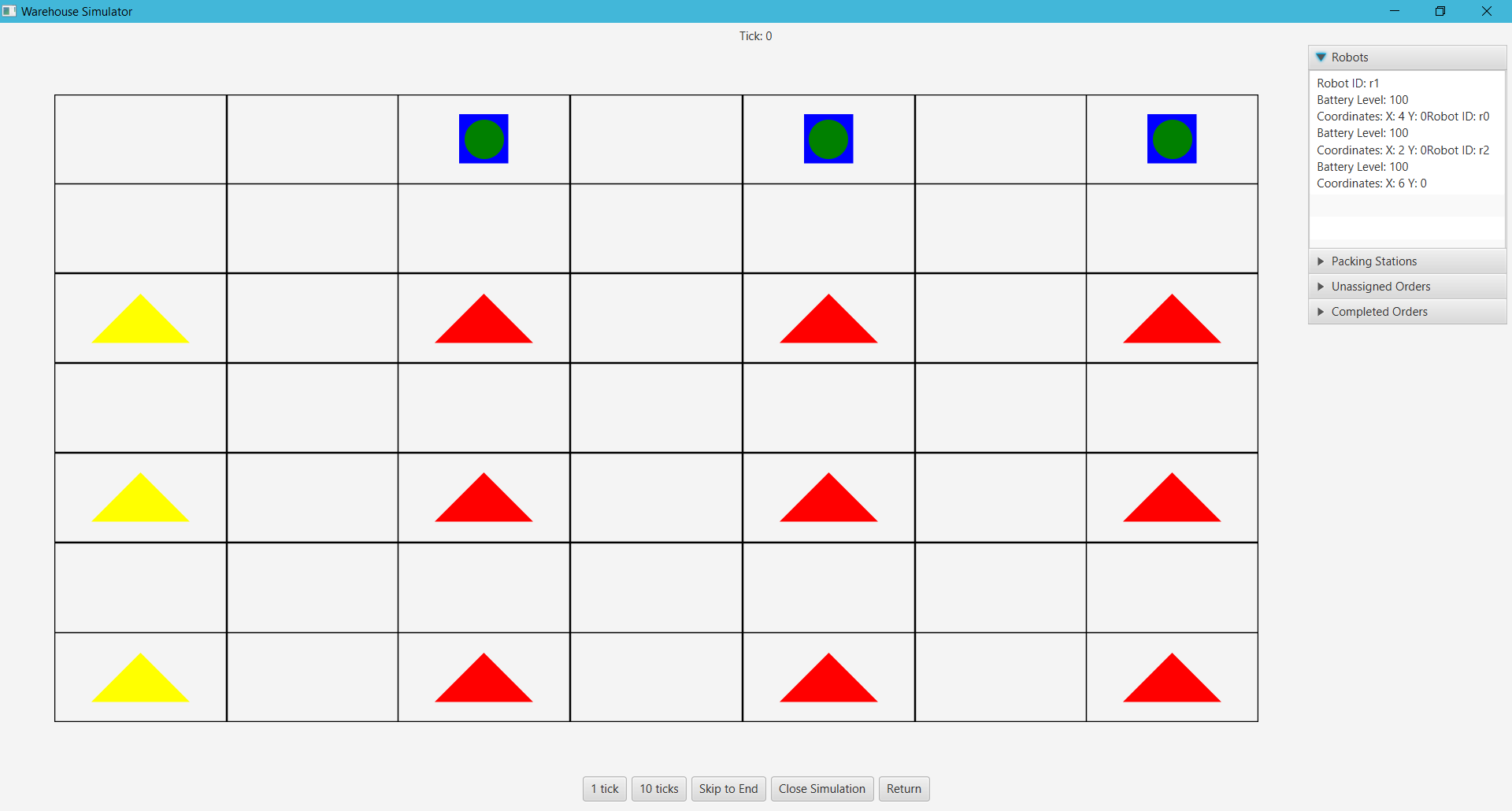
Open the simulator view

Set the charge rate of the robot

This is the first user interface that the user will see when starting the program. From here, the user can interact with the grid to create their own simulation, with their own defined, custom parameters, or they can load from a ‘.SIM’ File which predefines the parameters used in the simulation. After the user is satisfied with their simulation settings, they can press the ‘Start’ button and then view the simulation in action.

# Simulator View

Listviews representing the behaviour and entities.



Gridview showing the entities in the simulation. These entities will become animated as the user presses ‘1 Tick’, ’10 Ticks’

End the simulation or return to the main view.

Advance through the simulation; 1 tick at a time or 10 ticks at time

Skips to the end of the simulation.