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?

# Experiments

|  |  |  |
| --- | --- | --- |
| **Experiment** | **Hypothesis** | **Result** |
| Display entities on the grid | Clicking the entity button and then clicking a location on the grid will get the shape of the entity to be displayed on the GridPane. | The entity is successfully displayed onto the GridPane. |
| Pressing ‘Delete’ will remove an entity from the grid | Clicking the delete button, followed by selecting an entity on the GridPane will remove the entity. | The entity is successfully removed from the GridPane. |
| Change grid height/width | Changing the values of the sliders will allow the grid to resize, with stackpanes being automatically created inside each cell in the grid. | The gridpane successfully resizes, and stackpanes are successfully created. |
| Load .SIM File | Clicking on the load button, followed by selecting a ‘.SIM’ file. The .SIM file should be read, and the entities should be created onto the gridpane accordingly and automatically. The grid should also be resized as specified in the .SIM file.  In this test, I will be loading the ‘Three of everything.SIM’ File. | Success: The .SIM file is loaded successfully, in the correct format. The GridPane is resized accordingly, and stackpanes are created automatically. |
| Display .SIM File | Clicking load, followed by the ‘SIM’ File contents being displayed in a Text Area. | Success: The .SIM file is read and displayed correctly in the text area. |
| Set Battery Capacity | Setting battery capacity from the slider. if the battery capacity is changed in the slider, then the Listview in the simulator will update accordingly. | Success: The battery capacity will change the parameter batteryLevel in the robot class. This will be changed in the listView, which is in the simulator view through the use of a listener. |
| Set Charge rate | Setting the charge rate will create a console output of the slider value. This means that the value that is holding the charge rate will be assigned to the value of the slider. By doing this the charge rate is set. | Success: The charge rate is set. It is assigned to the parameter chargeRate. The value of chargeRate is logged to the console. |
| Pressing ‘Start’ will display the simulator view | Pressing start will display the same grid as shown in the main scene, however, it will be in a separate window, and the user interface surrounding the grid will be changed. | Success, the simulator view, with the same grid, is shown. |
| Robots are animated when pressing ‘1 Tick’ | Pressing one tick will mean that the robots will shift cells, moving from the current cell in the gridpane to the next cell.  The shapes should be removed in the current cell, and after pressing the ‘1 Tick’ button, the same shape should be redrawn in the next cell. | **Prior to pressing ‘1 Tick’**    **After pressing ‘1 Tick’** |
| Show robot robot information in the robot listView. In the Simulator view. | The addition of robots in the simulator view should mean that the parameters holding the data regarding the robots are updated.  These parameters should be shown in the listView.  The parameters to display are:   * RobotID * Charge Rate * Coordinates   The list should be formatted in such a way that each robot’s information is shown separately. | Test failed. The list is not formatted correctly, and as a result, the information for each robot is not shown seperately, making it difficult to distinguish between the information regarding one robot and another.  The coordinates value is also not shown, meaning that the code that has been implemented to carry this action is not correct. |

# 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 listView shown in the simulator view, for the robot information, does not work as intended. This is a result of lacking the knowledge to update the Coordinates of the robot in real time. 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.

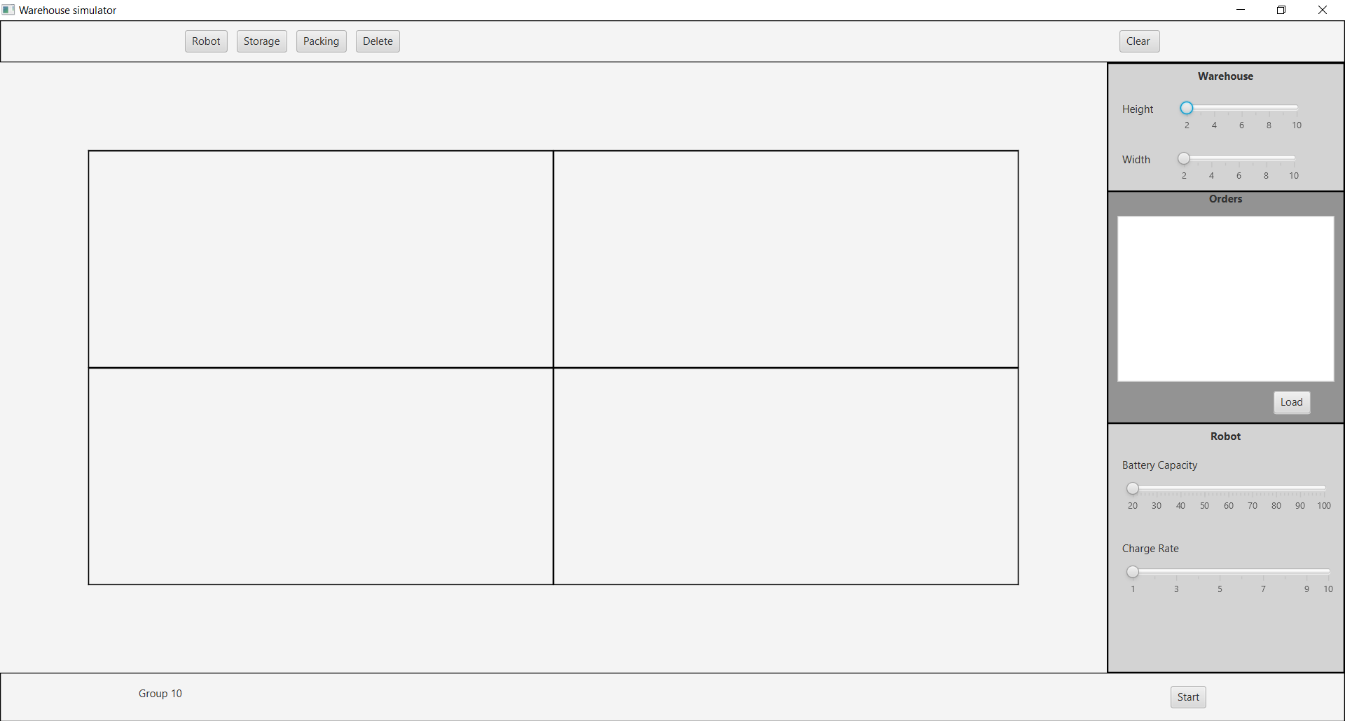
# The behaviour of the simulation

**The 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.

**The 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’

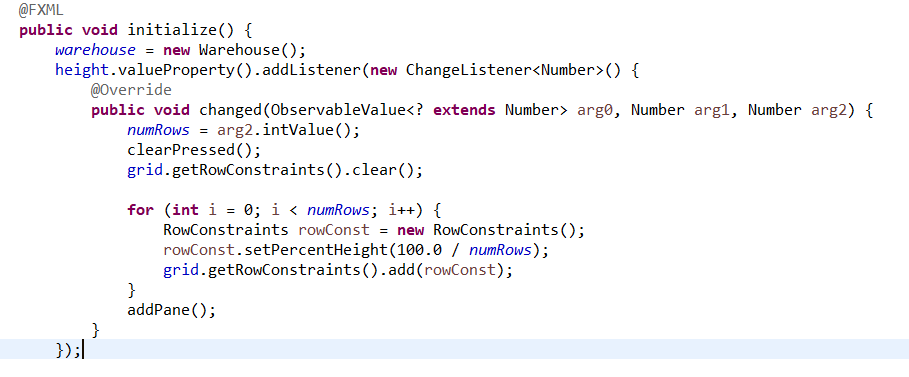
End the simulation or return to the main view.

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

**The slider behaviour**



The slider object, named height.



Call ValueProperty() This returns the current value that is set by the user for the grid height.

Call a listener on the valueProperty(), which is a changeListener**.** The changelistener detects whether the valueProperty of the height slider has been changed. When it does, the field called numRowswill be changed.

Furthermore, when the slide is adjusted, the clearPressed() method is called, which will remove all entities from the grid.

Call the getRowConstraints method, which clears the X and Y coordinates held in the grid, so that it can be redrawn.