

USER MANUAL



Automated
Kiwifruit Picking

Test Drive

CONTENTS

Table of Contents

Start-up.....2

Status.....3

Stage.....4

Stage - Perspectives.....5

Stage - Toggle Laser.....6

Testing.....7

Table of Figures

Figure 1.0: Initial GUI.....2

Figure 2.0: GUI Showing Status.....3

Figure 3.0: Stage Window.....4

Figure 4.0: Stage 3D View.....5

Figure 5.0:

Figure 6.0: Test Results Terminal.....7

START-UP

The ROS simulation can be initialised by running the start-up script provided.

To do so, please navigate to the “se306project” folder in the linux terminal and enter the following command -
`./startup.sh`

The screenshot shows a window titled 'MainWindow' with a light gray background. It contains three large, empty rectangular boxes labeled 'Robots', 'People', and 'Animal' from top to bottom. Below these is a 'Configuration' section with a grid of input fields and buttons. The grid has four columns: the first column lists configuration items, the next three columns contain input fields with numerical values or selected options. At the bottom right of the configuration section are four buttons: 'Test Drive!', 'Close', 'Launch' (highlighted in blue), and 'Display Status'.

Configuration			
Picker Robots	1	Row Number	7
Carrier Robots	1	Gardener	1
Row width	3.50	Dog	1
Pole/Trunk spacing	2.50	Cat	1
		Worker	1
		Blind Person	1
		Neighbor	1
		Tractor	1

Buttons: Test Drive! Close Launch Display Status

Figure - 1.0

The script will compile & launch the project. Once the project has been successfully compiled the following interface will be presented.

The world of the simulation can be configured on this interface. Simply, enter the desired number of each world component and select the “Launch” button.

STATUS

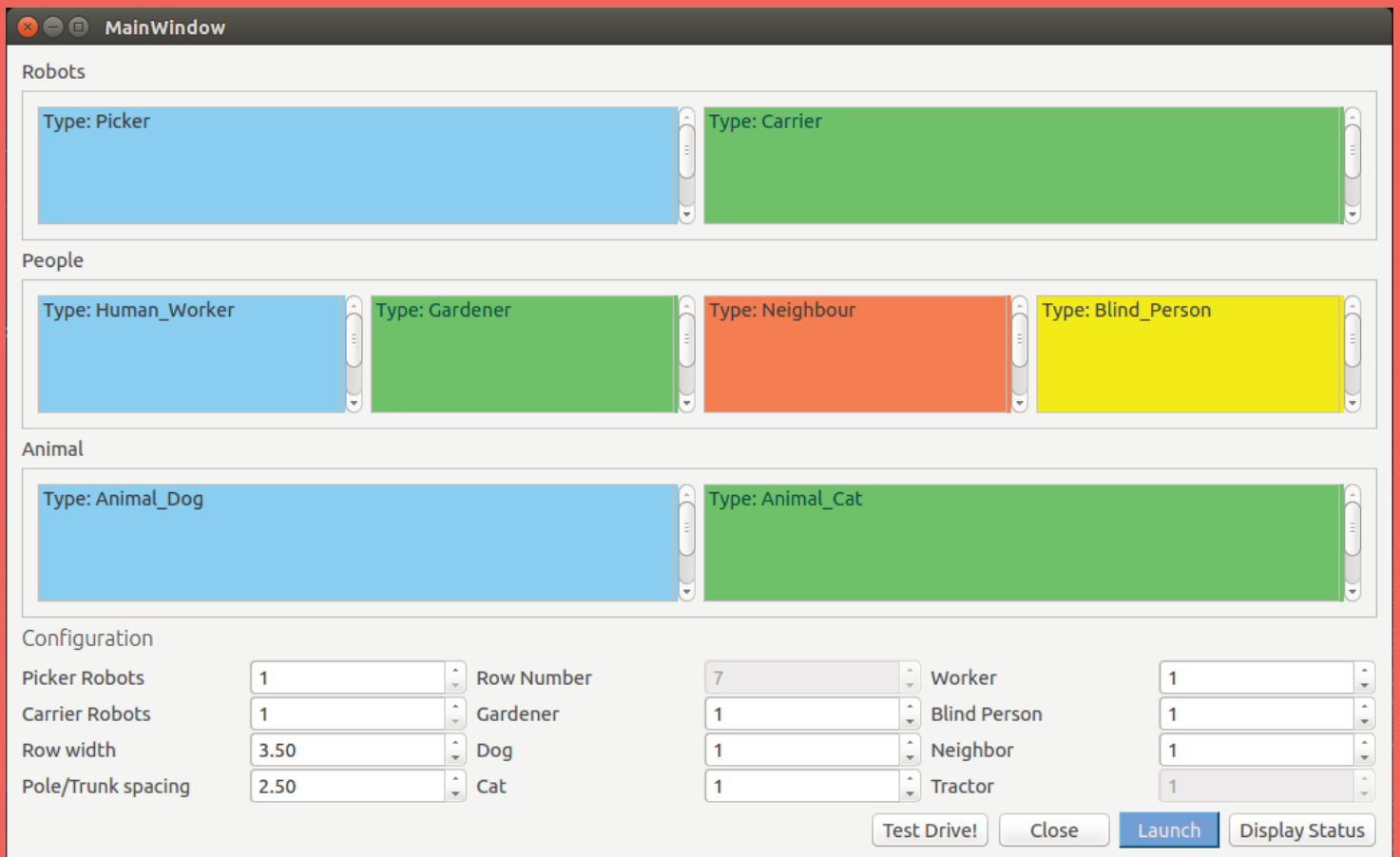


Figure - 2.0

During the simulation the statuses of the world components, such as the picker robot, can be followed on the interface.

For the picker and carrier robots the colour of the panel represents the colour of the robot in Stage.

Status Fields:

Type: What the world component is.

pos_x: The component's x position.

pos_y: The component's y position.

pos_theta: The component's facing angle.

status: Current action, such as moving or idle.

obstacle: Whether the component has encountered an obstacle.

STAGE

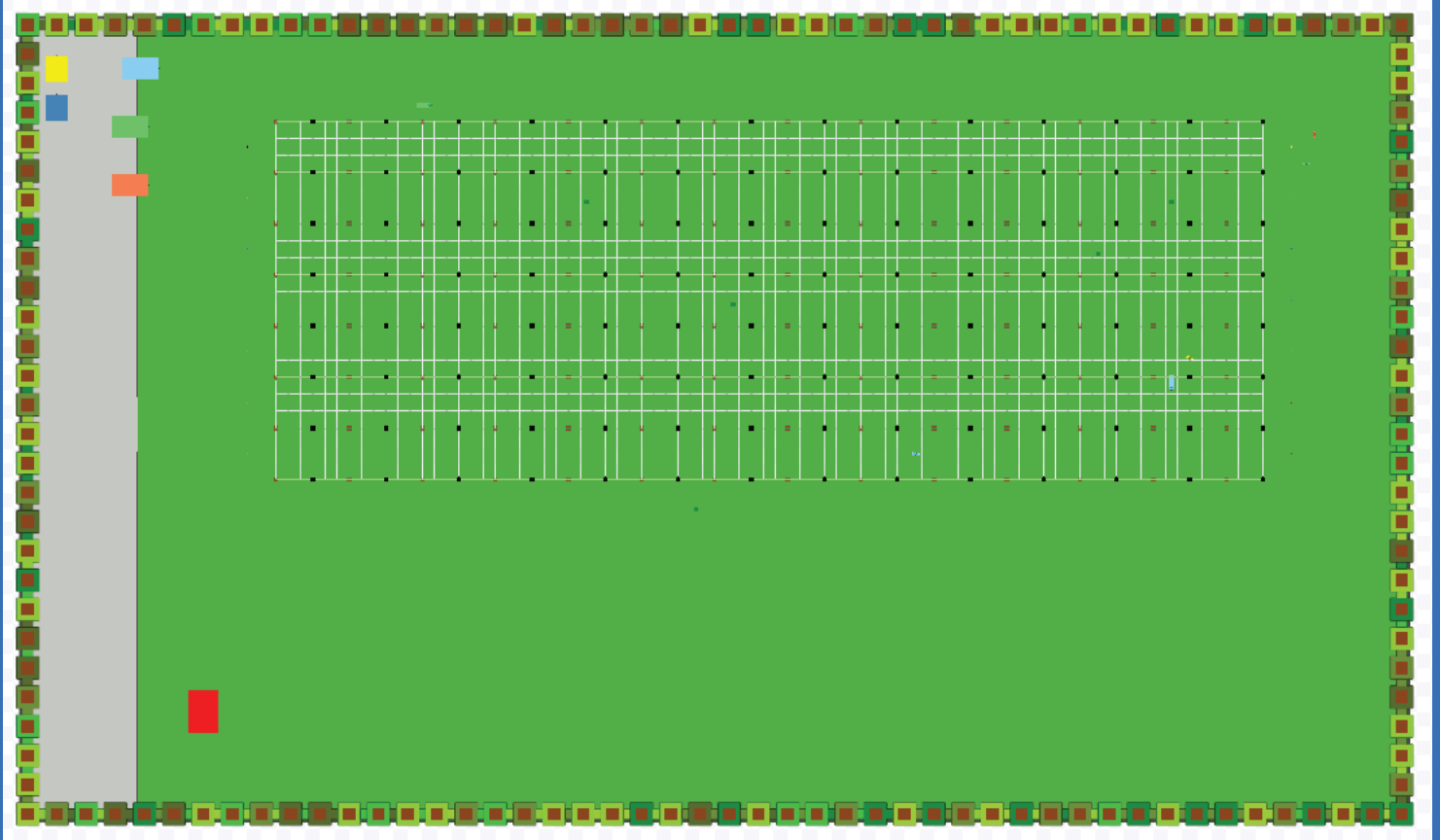


Figure - 3.0

The simulation can be viewed on the Stage window once the “Launch” button on the interface has been pressed. Here the simulation of the robots picking kiwifruits can be viewed; combined with the environment components, such as the worker and dog components.

To close the Stage window press the “Close” button on the interface.

To change the view perspective in Stage press the “ctrl” key on the keyboard and hold the left mouse button. Then, move the mouse in the direction for the desired change in perspective.

STAGE - TOGGLE LASER

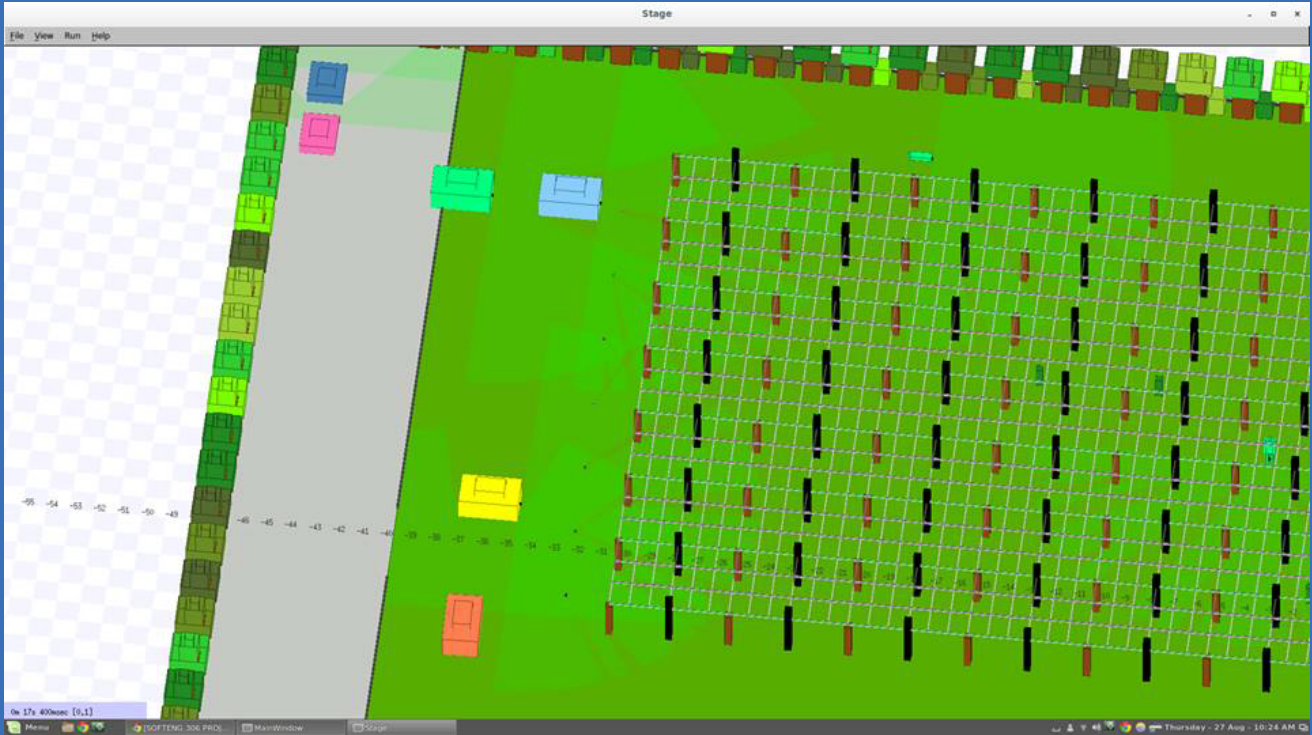


Figure - 4.0

To view the lasers emitted by each world component, such as the picker robots, press the “D” key on the keyboard.

This will show the lasers that are emitted and used by the world components for functions such as obstacle detection.

The lasers can be toggled off in the same manner as turning them on.

TESTING

To run the test scripts relating to the simulation please navigate to the “se306project” folder in the linux terminal. Then proceed to enter the following command - “./test.sh”.

```
ect --results-filename test rostest testGardenWorker.xml /afs/ec.auckland.ac.nz/
users/c/h/chan743/unixhome/ros_ws/src/se306project/test/rostest/testGardenWorker.
test
[-----] Running 5 tests from 1 test case.
[-----] Global test environment set-up.
[-----] 5 tests from Movement
[ RUN      ] Movement.constructMovementDefault
[       OK ] Movement.constructMovementDefault (0 ms)
[ RUN      ] Movement.constructMovementOverload
[       OK ] Movement.constructMovementOverload (0 ms)
[ RUN      ] Movement.setMovementType
[       OK ] Movement.setMovementType (0 ms)
[ RUN      ] Movement.setMovementPos
[       OK ] Movement.setMovementPos (0 ms)
[ RUN      ] Movement.setMovementVel
[       OK ] Movement.setMovementVel (0 ms)
[-----] 5 tests from Movement (0 ms total)

[-----] Global test environment tear-down
[-----] 5 tests from 1 test case ran. (1 ms total)
[ PASSED   ] 5 tests.
-- run_tests.py: verify result "/afs/ec.auckland.ac.nz/users/c/h/chan743/unixhom
e/ros_ws/build/test_results/se306project/gtest-testMovement.xml"
[100%] Built target _run_tests_se306project_gtest_testMovement
-- run_tests.py: execute commands
```

Figure - 6.0

The results of the tests will be displayed on the same terminal window.

There are tests for the world’s components, such as the carrier robots.

If a test has failed then the file and line number where the error occurred will be shown in the terminal window.