***INFO6205 Final Project-Genetic Algorithms***

*Robot Controller*

*Group 525*

***Group Members:***

*Tse-Hsi Lin*

*Yu-Chiao Huang*

***Instructor:***

*Prof. Robin Hillyard*

***Implements***

*Problem Statement:*

The problem we are going to solve is designing a robotic controller that can use the sensors to walk through the maze and arrive the destination without crashing into the wall. The robot has six different direction sensors, three on the front, one on the left, one on the right and one on the back. The robot can take four actions: move one-step forward, turn left, turn right, or stay. Each route can be one solution and can be scored, we can find the fittest solution by comparing the scores we get. The purpose of this project is not to train a robot to solve the maze, our purpose is to automatically program a robot controller with six sensors so the robot won't crash into the wall, and the maze in this case represent the complicated environment in the world.

*Genotype:*

*Expression:*

*Phenotype:*

*Fitness:*

*Initialization:*

***Execution***

We use *IntelliJ* as our IDE for this project, and the framework we use is *Maven*, we have five different packages: *controller, helper, impl, repository and robot\_main*. We put the robot controller under *controller* package, put maze generator under *helper* package, put our Interfaces under *repository* package, put our implements for Interfaces under *impl* package and put our Main class under *robot\_main* package, we also have a maze folder to store our maze and a test package to store our tests.

**To execute our project**, you need to run the Main class under *robot\_main* package, then it will call the controller class to run our program, you also need to input the Maze name in the console, for example, input “*Maze1”*, we have 4 different Mazes, if you want to run the default Maze, you can just press *Enter*.

***Code***

***Experiment***

***Conclusion***