**Module Code:** CS2PJ20

**Assignment report Title:** Robot Simulation + GUI

**Date (when the work completed):** Thursday 14th of December ( extended Deadline)

**Actual hrs spent for the assignment:** 50 Hours more or less

(REPORT TEMPLATE)

Robot GUI Simulation

# Introduction and Showcase

This is a basic simulation of Robots. This simulation is created with javafx, which is helping us to create an interface for the arena of the robots. Added buttons such as start and stop of the simulation and menu bars so the user can save, load or create a new arena with their desire dimensions. Each robot on the arena have their own different characteristics.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Fig. 1:** Start of the robots simulation |  | **Fig. 2:** Details of th robots on the right |
|  |  |  |
| **Fig. 3:** Menu Bar options |  | **Fig. 4:** About information |

# OOP design diagram

|  |
| --- |
|  |
| **Fig 5.** Class Diagram of Robots |

This is the diagram of the Robot\_GUI\_Simulation, is an application written in java which provides a interface to simulate and interact with robots in the arena. This project works with OOP ( Object-oriented principles) .

* MainRobotSumGUI – Application Folder wher the application launches.
* RobotArena – Manage the arena, robots and interactions.
* Robot (Abstract) – it is the skeleton of all the robots, common properties like drawing, checking collisions and adjustments.
* GameRobot – The robot game, which extends robots
* PacManBot – Robot that consumes Robots within a certain range.
* MyCanvas – Graphical circles, lines, and text
* RobotInterface - User GUI with menus, and buttons to interact with RobotArena.

This are the classes that creates a simulation where the robot move, interact and displays their data on a canvas. GUI allows the user to use buttons such as:

* Load / Save
* Play / Stop
* Create New Arena
* Add Robots

Then the robot that is special is PacManBot which is eating GameRobots when they are in front of it.

# Discussion

Robot Simulation GUI is an application where the user can visualize and interacts with robots in the arena. The simulation presents the behavious of the robot such as collision detection, a unique feature like PacManBot eating GameRobots. The application offers a graphical user interface (GUI) for an interactive user experience. The simulation includes various types of robots, each with distinct behaviors, and provides features for loading/saving arena configurations.Tried to add new feature for the robots but the testing were failure ex:

The application follows an ood (object-oriented design), including classes like Robot and RobotArena, with different types of robots:

* GameRobot
* TargetRobot – is a targer where the robot need to touch to make the counter go up
* BeamRobot – Is the robot that meant to have beams and move when the beams detects any obstacle in front of them
* WhiskersBot – is same as beam robot but using whiskers. This one has the design but not the fuction.
* PacManBot – it is the New Created Robot. It is successfully created but missing the function of following the robot that is going to eat.
* BrokenRobot – represents the obstacle,

The Robot class is the parent class, which provides the common attributes and methods of all the robots, each Robot has specific types of inherits from the class, with checkRobot and adjustRobot. The checkRobot determines if a robot should change its angle based on the collision with walls and robots within the arena. Then adjustRobot handles the movement of the robot, this methos enable dynamic and responsive robot behaviors on the simulation.

RobotArena is the container for the robots in the arena, this class provides the draw of the entire arena, the collision of the robots and the adjusting of their positions. Also , includes the loading and save of the arena for the user to use the current state for future use.

For the last we have the GUI feature where it holds the menubar,and all the options for the application, the design of this is aim for a user-friendly interaction and real-time visualization of the robots movements.

# Reflection

After this project of balls transforming it in to robots, I gain some knowledge in java structure and the way it needs to be wrote. This simulation was very interesting in a way of me learning a new language of coding. As a newbie in java I was a bit scared at the beggining but slowly I took the corauge of keep adding new lines and new pieces of code to the application, with still a fail result, some were successfully but I know, that I need still to learn more about it. The cleaning of the code still on work as seen on this indents and comments are still lucking, but with more practice I am sure I can get to at lease work with java.

Java in compare to the other languages I have learnt or/and work with, I see it is a bit harder, all the syntax, the way it needs to be organize and all the functions.

In conclusion, I really love this project and with the output of my application. I know is not the best one but for me and all the time I put in to it I iam very happy, i did try the hardes ways to implements new functions or new features for robots. As a beginner and with what I have I am very happy to finish like this. Next time I would be ready for it.

THIS IS PAGE 4

END OF YOUR COURSEWORK PAGE LIMIT, OTHER THAN APPENDIX WITH YOUR EVALUATION

# Appendix – Self Evaluation

Fill in the table below, indicating in column 3 the marks you feel you should get for the different categories, a brief justification in column 4, and the total marks for the section in the Total column.

The marker will read the report, check code and watch the video and determine the actual mark

|  | Student’s own assessment | Mark | Brief justification | Total |
| --- | --- | --- | --- | --- |
| 1. | **Appearance**   * Professional looking 0-2  * Has About and Help 0-2 * Understandable 0-2  * Toolbar buttons 0-2 * Menu 0-2 |  | Comments | 0-10 |
| 2. | **Interaction / Animation.**   * Animation works well 0-1  * Animation controlled by menu 0-1  * Stop/Start by buttons 0-2  * User can add new items 0-2 * User can select item to move 0-2 * User can select item to delete 0-2 |  | Comments | 0-10 |
| 3. | **File Handling etc**   * New arena works 0-2  * Default has populated arena 0-2 * File Load works 0-3  * File Save works 0-3 |  | Comments | 0-10 |
| 4. | **Arena**: does it support different entities   * Robots shown with wheels etc 0-2 * Robots move 0-2  * Has obstacles 0-2 * Has other non moving objects 0-2  * Robot with ‘bump’ type sensor 0-2  * Robot with ‘whisker’ or similar 0-2 * Robot with ‘beam’ sensors 0-2  * Other (specify) 0-2 * Other (specify) 0-2 * Other (specify) 0-2 |  | Number of different types in arena:   |  | | --- | |  |   Comments: | 0-20 |
| 5 | **On Code**   * Appropriate use of names 0-3 * Consistent indentation 0-2 * Has inline comments 0-2  * Has Javadoc comments 0-3 * Correct use of access modifiers 0-3 * Has abstract class for arena items 0-2 * Has good use of inheritance 0-5 |  | Comments | 0-20 |
|  | Section below is assessed by marker of work |  |  |  |
| 6 | Quality of the submitted **report, etc**.  Tests and discussion in report  Video, Javadoc, code, etc., submitted.  Novel features |  | **Evaluation based on submitted work.** | 0-30 |