



## Project Proposal - SYSC 4805



### Team Blue Jeans

Angela Byun 101004764

Tarun Kalikivaya 101056745

Martin Klamrowski 101009909

Matthew Wesley-James 100944007

1023 Words

**Keywords:** Mobile robot, maze, path-finding

11<sup>th</sup> February, 2021

---

#### Abstract

Team Blue Jeans proposes a mobile robot with maze navigating capabilities. The robot will enter a maze with the purpose of locating an objective. The robot must locate the objective, load it into its carriage, and exit the maze within the allowable time. This robot would have applications in search and rescue, and exploration. The physical elements of the robot will be simulated in CoppeliaSim. The robot's underlying intelligence will be implemented in Python and integrated with CoppeliaSim using the provided CoppeliaSim Python API.

---

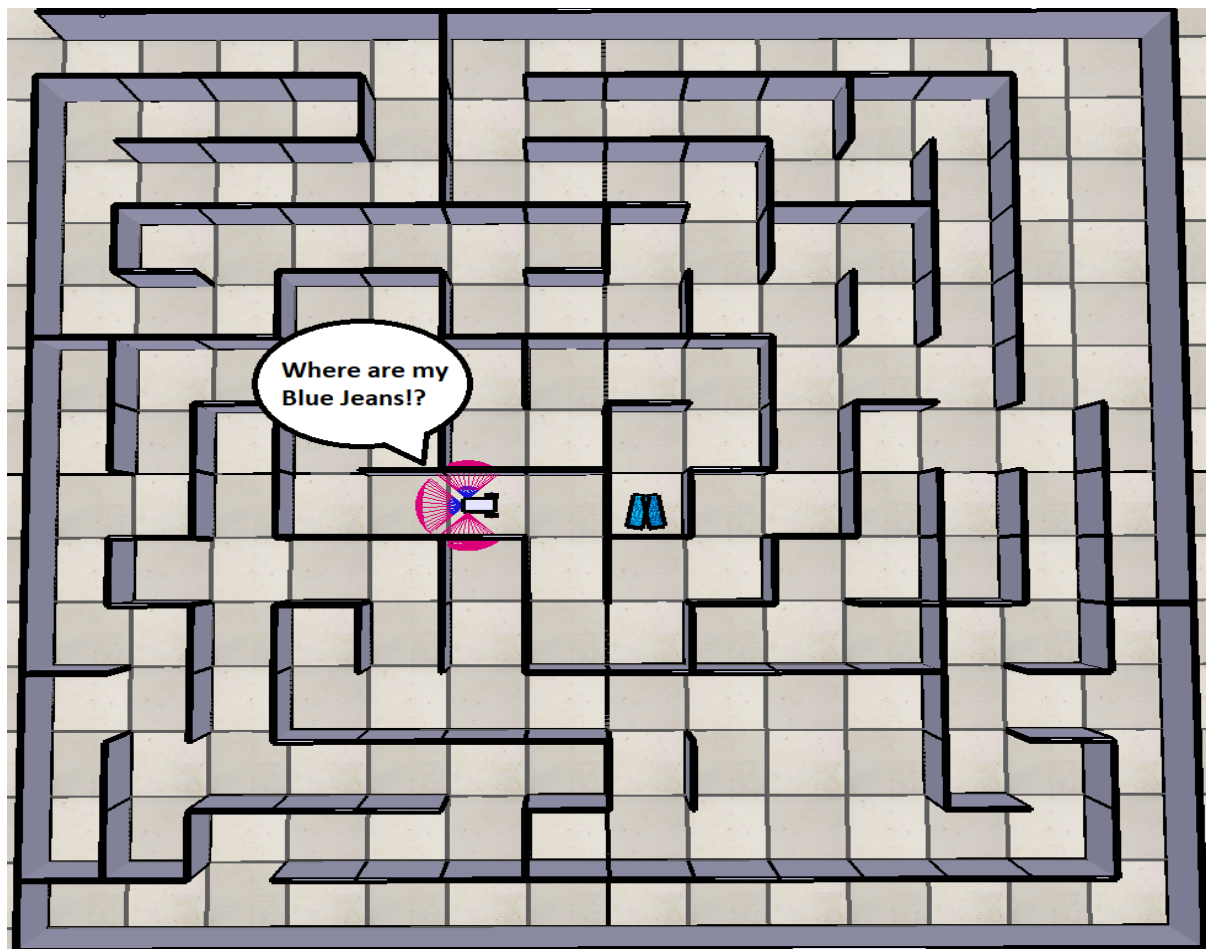
## Contents

<b>1</b>	<b>Charter</b>	<b>3</b>
1.1	Objective . . . . .	3
1.2	Deliverables . . . . .	3
<b>2</b>	<b>Scope</b>	<b>4</b>
2.1	Requirements . . . . .	4
2.2	Work Breakdown Schedule (WBS) . . . . .	7
2.3	Testing . . . . .	7
<b>3</b>	<b>Schedule</b>	<b>9</b>
3.1	Schedule Network Diagram . . . . .	9
3.2	Gantt Chart . . . . .	9
<b>4</b>	<b>Human Resources (HR)</b>	<b>10</b>
4.1	Responsibility Assignment Matrix . . . . .	10

# 1 Charter

## 1.1 Objective

The objective of this project is to design a robot that will enter a maze to find an objective then retrieve it by carrying it out of the maze. The concept is visualized in Fig. 1 below.

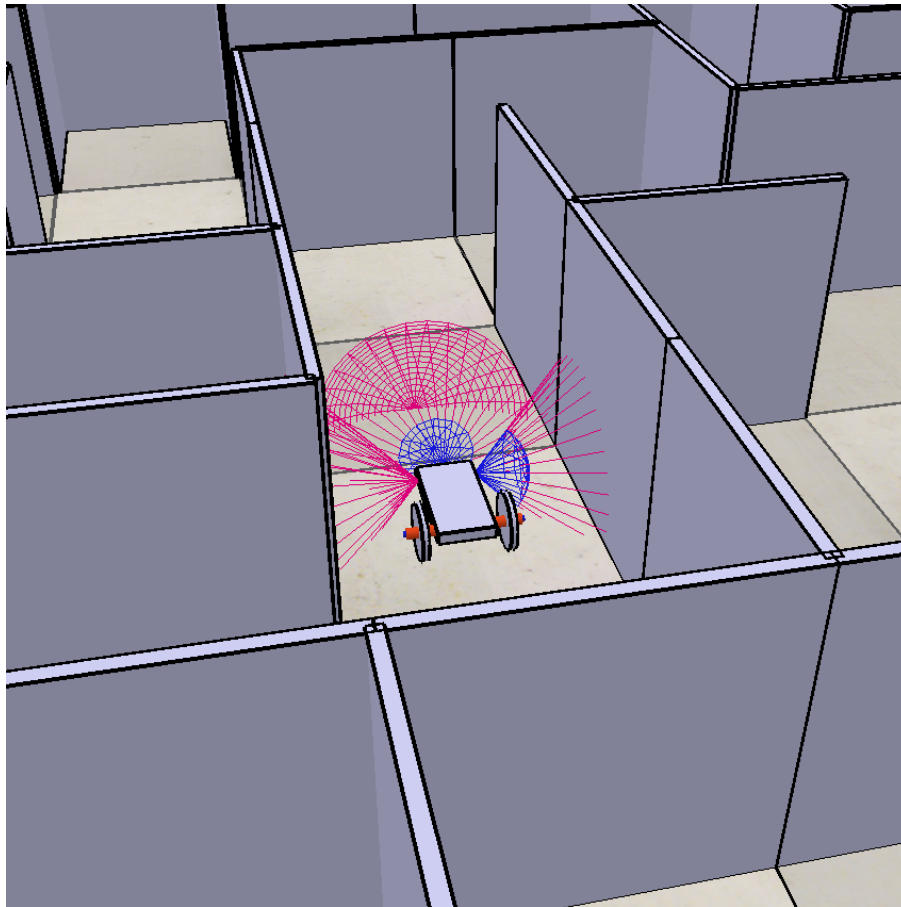


**Fig. 1.** Project Concept.

## 1.2 Deliverables

The purpose of this project is to attain 2 deliverable; first being the project management based on expectations for this course and the second being the final robot system, which will be the maze robot. Project management will be applied through the proposal, progress report, presentation, demonstration, and the final report by monitoring the progress. The proposal will consist of how we plan on achieving the project goal, and the basic layout of the maze and the robot, including the milestone. The maze robot system will consist of the maze, the robot, and the objective. The robot will be programmed to find its way by using different sensors, and a path finding algorithm. We believe that our challenge would be programming the robot because it will be required to move through and detect possible paths in the maze, find and carry the objective then find its way back out of the maze again.

## 2 Scope

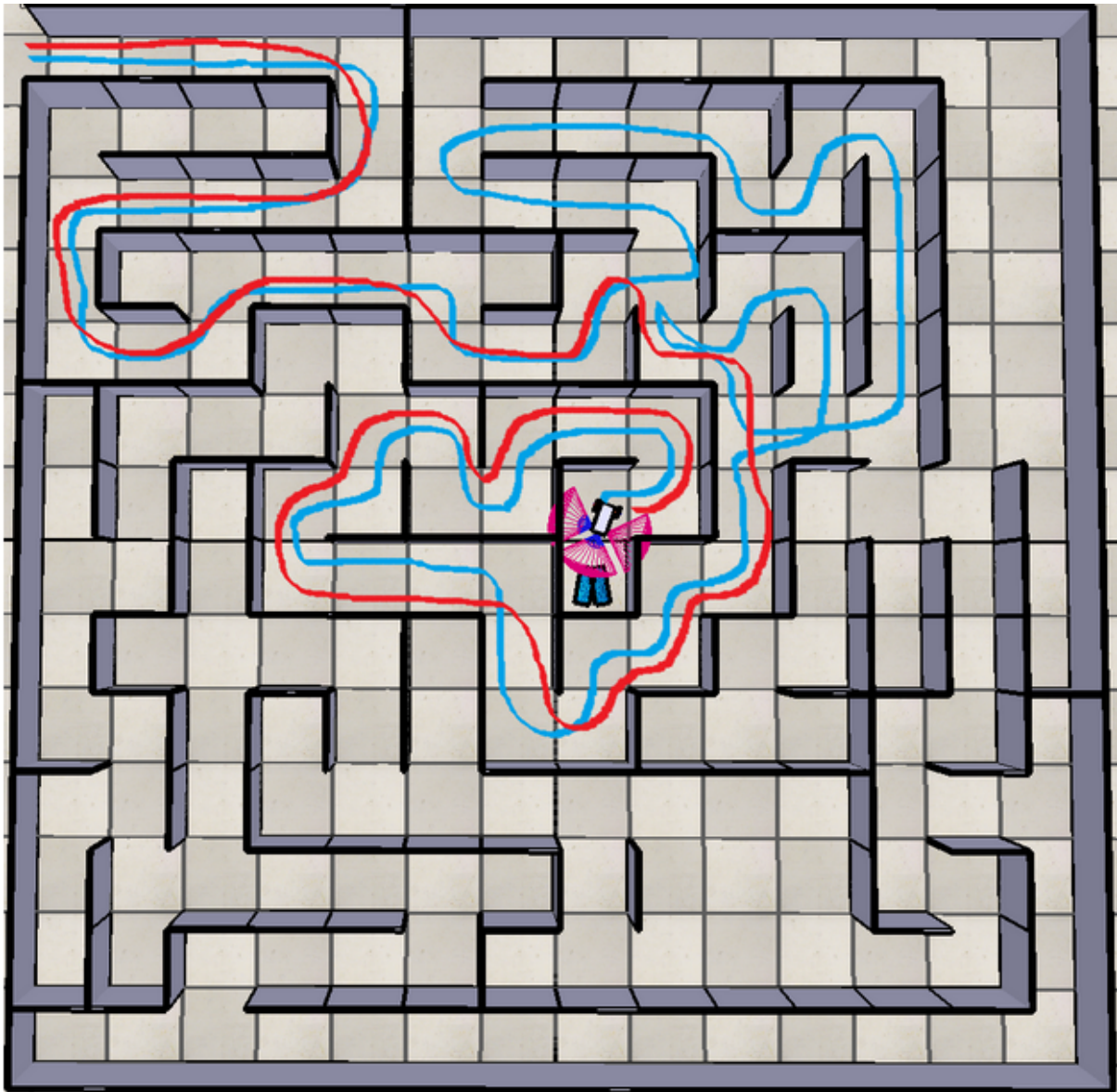


**Fig. 2.** Robot navigating a maze corridor.

### 2.1 Requirements

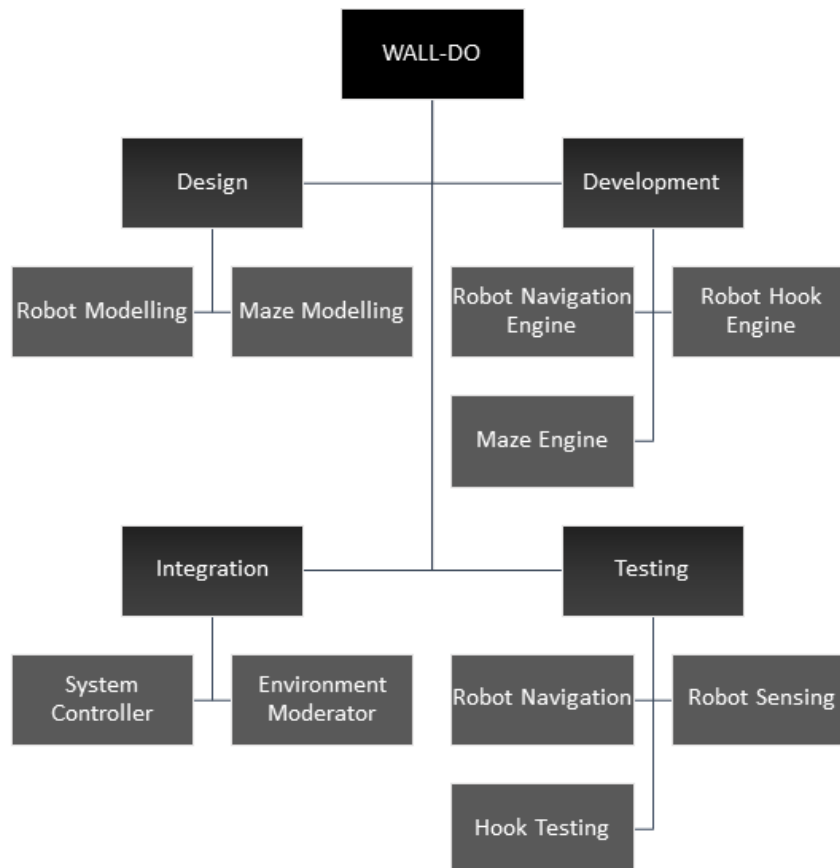
1. The robot must be able to maintain constant and smooth forward and backwards motion at a speed that ensures reqs. 2, 3, 4 are also met.
2. The robot must be able to perform turns up to  $180^\circ$  in sharpness.
3. The robot's forward sensors must detect walls in the robot's path at least 0.5m away (the length of a single wall segment).
4. The robot's lateral sensors must detect walls immediately adjacent and parallel to the robot's path at least 0.25m away (half the length of a single wall segment).
  - 4.1. The robot must maintain 0.25m of distance from any detected wall segments, unless doing so conflicts with reqs. 5.1, 5.2.
5. The robot's sensors must be able to detect the objective in a  $180^\circ$  forward-facing arc of radius 0.5m.
  - 5.1. When the robot detects the objective, it must turn and move to the objective.
  - 5.2. The robot must stop when less than 0.05m away and facing the objective.

6. The robot must move forward until a blocking wall segment is detected or req. 5.1 and 5.2 are met.
  - 6.1. When the robot's sensors detect a wall segment in the robot's path, and only the robot's right lateral sensor detects a wall segment, the robot must perform a 90° turn to the left.
  - 6.2. When the robot's sensors detect a wall segment in the robot's path, and only the robot's left lateral sensor detects a wall segment, The Robot must perform a 90° turn to the right.
  - 6.3. When the robot's sensors detect a wall segment in the robot's path, and both the left and right lateral sensors of the robot detect a wall segment, the robot must perform a 180° turn in place.
7. The robot must keep track of the path it takes through the maze.
  - 7.1. The robot must know what direction (in °) it is heading.
  - 7.2. The robot must have an odometer to track the distance it travels in each direction.
  - 7.3. The robot must prioritize turns into unexplored areas higher than turns into explored areas.
  - 7.4. When presented with multiple turns of differing priorities, the robot must take the higher priority turn.
  - 7.5. When presented with multiple turns of the same priority, the robot must randomly choose a turn to take.
  - 7.6. The robot must take the shortest path it knows when exiting the maze.
8. The robot must have a retractable hook on both the front and back of its body.
  - 8.1. The robot will line up one of the retractable hooks with a hitch on the objective when req. 5.2 is active.
  - 8.2. The robot will secure the objective by hooking a protruding hitch on the objective when req. 5.2 is active.
  - 8.3. When the objective is secured, the robot must pull the objective while moving out through the maze towards the exit.



**Fig. 3.** Path taken by the robot in (Blue) and out (Red) of the maze.

## 2.2 Work Breakdown Schedule (WBS)



**Fig. 4.** WBS Diagram

## 2.3 Testing

- **Unit Testing**
- **Robot Navigation**
  - Robot (without Sensors)
    - \* Robot will be tested to move straight at a constant speed, turn 90°, and 180°, and stop. This will be to confirm requirements 1 and 2. These will be verified by calling getObjectPosition() in a loop.
  - Robot Sensors
    - \* Each sensor will be tested by having the sensor move toward a flat wall, the position of the sensor will be tracked to verify the condition is met. This will confirm requirements 3 and 4 (using get/setObjectPosition).
    - \* The sensor will be placed with a wall on one side and the objective on the other side (0.5m away), it will rotate in place and detect what is in front of it. This will confirm requirement 5 (using get/setObjectPosition).

- Hook Testing

- \* To complete requirements 8 (and the subsequent requirements), the hook will be activated (after requirement 5.2 is met) then the hook will be activated. Once the hook has signaled it is finished. The robot will move forward. If the robot's velocity equals the objective's velocity the requirements will be satisfied.

- **Integration Testing**

- The robot will be placed facing towards a wall, and will run forward, if it stops 0.5m away from the wall requirement 6 is met.
- For requirements 6.1, 6.2, and 6.3 the robot will be placed in the center with walls in four different cases (as shown in Fig. 5 below.) Once the correct direction is chosen the requirement will be met.
- For case A the robot will be pre-programmed with each of the three directions having different unexplored area values, this will test the requirements 7.4, 7.5, and 7.6.



**Fig. 5.** A: Front-facing wall. B: Front-facing wall and adjoining left wall. C: Front-facing wall and adjoining right wall. D: Dead end.

- **System Controller Testing**

- For the requirements of 7 (and the subsequent requirements) a custom made testing maze will be made, with trackers for each requirement. These will be waiting until the robot goes to a certain positions that will be calculated based on the custom testing maze, then these requirements will be complete.

- **Full System Testing**

- Robot will enter the maze (assuming robot moves at 0.5m per second) the robot will have to exit the maze within the time limit to be successful. The time limit in seconds  $t_{max}$  is determined by Eq. (1), where  $l_M$ ,  $h_M$  represent the maze length and height respectively in meters.

$$t_{max} = 2l_M h_M + 5 \quad (1)$$



### 3 Schedule

#### 3.1 Schedule Network Diagram

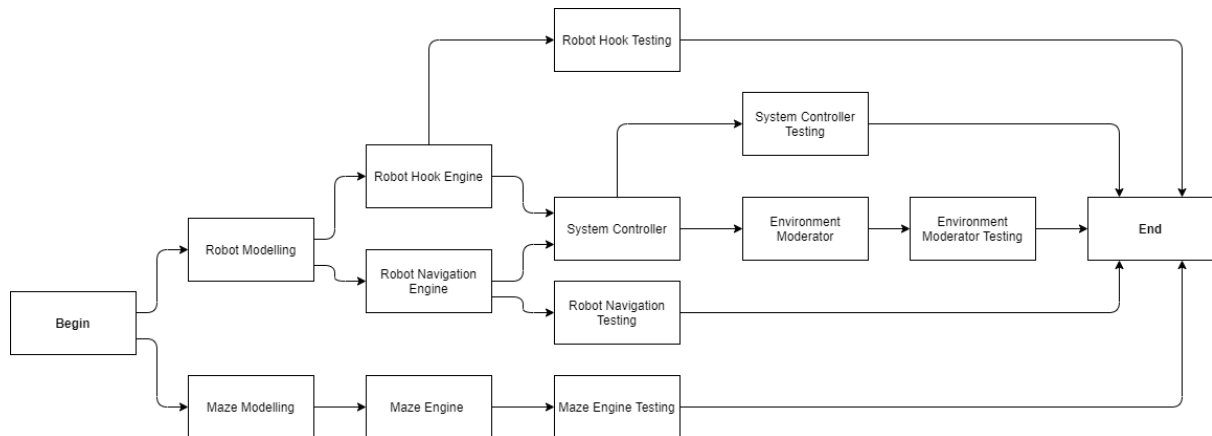


Fig. 6. Schedule Network Diagram


#### 3.2 Gantt Chart



Fig. 7. Gantt Chart for Activities

## 4 Human Resources (HR)

### 4.1 Responsibility Assignment Matrix

	<b>Angie</b>	<b>Tarun</b>	<b>Martin</b>	<b>Matthew</b> 
<b>Robot Modelling</b>	R	A	R	A
<b>Maze Modelling</b>	R	R	A	A
<b>Robot Navigation Engine</b>	A	A	R	R
<b>Robot Hook Engine</b>	A	R	A	R
<b>Maze Engine</b>	R	A	R	A
<b>System Controller</b>	R	R	A	A
<b>Environment Moderator</b>	A	A	R	R
<b>Robot Navigation Testing</b>	A	R	A	R
<b>Robot Hook Testing</b>	R	A	R	A
<b>Maze Engine Testing</b>	R	R	A	A
<b>System Controller Testing</b>	A	A	R	R
<b>Environment Moderator Testing</b>	A	R	A	R

Legend: R -- Responsible, A -- Approver

Fig. 8. Responsibilities.