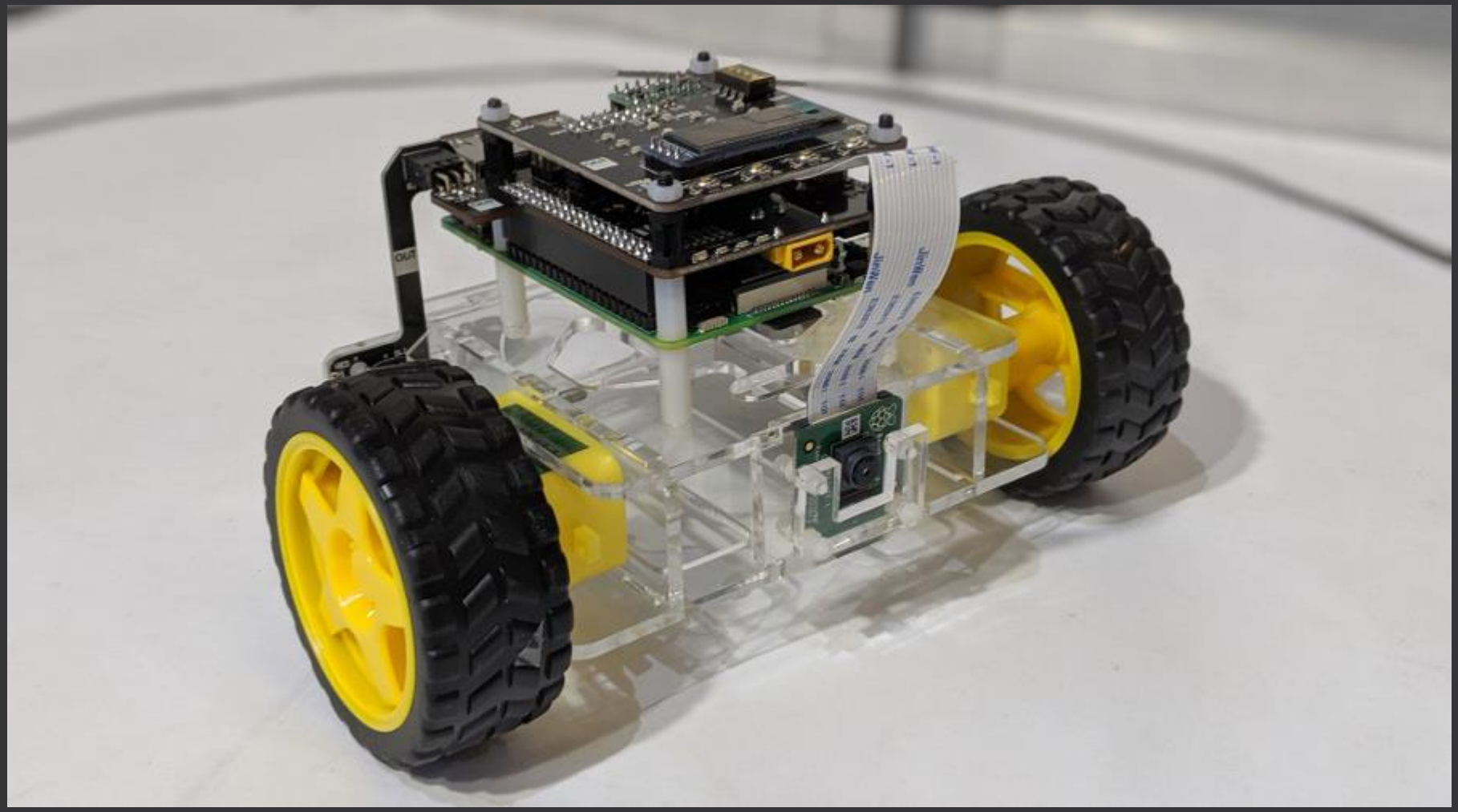
ECE4078 Intelligent robotics



Lab 4-1: Kal Backman & Shujie Zhou

ECE4078 Intelligent robotics

Week	Objectives	Milestones
~~2: M1-1~~	~~Introduction and setup~~	
~~3: M2-1~~	~~Calibration, ARUCO markers~~	~~M1 due~~
~~4: M2-2~~	~~SLAM~~	
~~5: M2-3~~	~~SLAM~~	
~~6: M3-1~~	~~Object recognition & localisation~~	~~M2 due~~
~~7: M3-2~~	~~Object recognition & localisation~~	
* 8: M4-1	Navigation & Planning	M3 due
9: M4-2	Navigation & Planning	
10: M5-1	Integration	M4 due
11: Final	Improvement	M5 due
12: Final	Final demo & competition	Final demo

Milestone 4

Milestone is focused on navigation and planning

Your task will be to semi OR fully autonomously navigate and path plan around the arena to “pick up” various fruit in a specified order

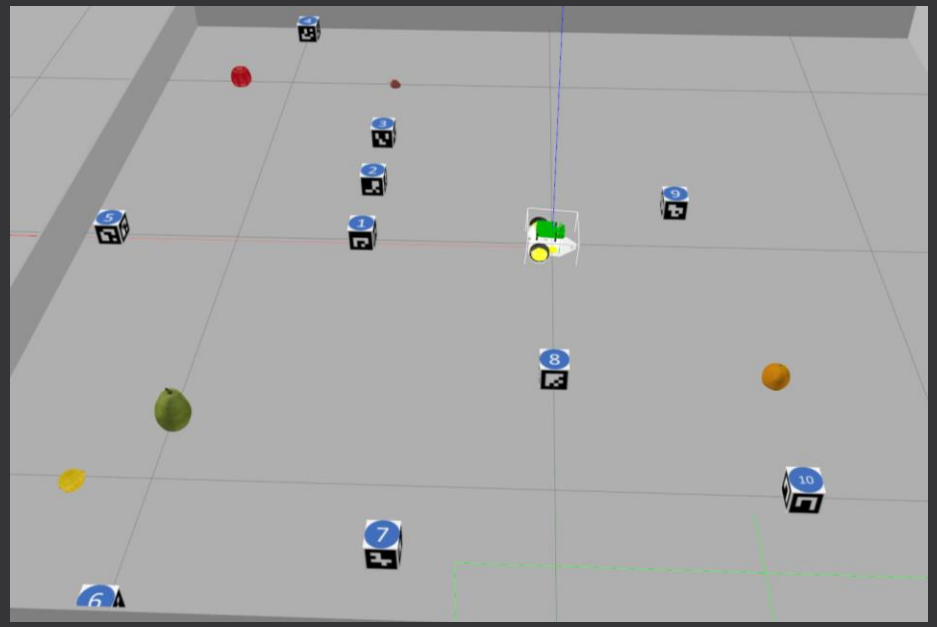
The arena will always contain:

10 ArUco markers

1 of each fruit (apple, lemon, pear, orange & strawberry)

You will given a text file “search_list” that contains 3 fruits in the order to pick them up in

To pick up a fruit you must stop the robot for 3 seconds whilst being within 0.5m of the fruit



Milestone 4

There are 3 levels of difficulties which you can choose to complete the task in:

Level 1: Semi-auto navigation + known map

- You will be provided with the location of all objects within the arena
- You may control the robot by inputting “waypoints” for the robot to travel to
- You can NOT teleoperate the robot (you can’t use keyboard inputs to drive the robot like in M2)

Level 2: Fully autonomous navigation + known map

- You will be provided with the location of all objects within the arena
- You are only allowed to enter a single command to start the robot
- The robot should autonomously complete the task without human intervention

Level 3: Fully autonomous navigation + partially known map

- You will be provided with the location of all objects EXCEPT the 2 unlisted fruit
- These unknown fruit will be strategically placed such that there is a high probability that your generated path will intersect these fruit
  You will have to use your M3 work to detect the obstacle fruit
- You are only allowed to enter a single command to start the robot
- The robot should autonomously complete the task without human intervention

Milestone 4

Each of the 3 levels have different amount of marks associated with picking up the fruits

Level 1: (Total marks: 60)
For each fruit you correctly pick up in order: 20 marks

Level 2: (Total marks: 20)
For the first fruit: 8 marks
For the second & third fruit: 6 marks

Level 3: (Total marks: 20)
For the first fruit: 8 marks
For the second & third fruit: 6 marks

You will also be given penalties based on:
Robot colliding with fruit (-2 marks)
Robot colliding with ArUco marker (-5 marks)
Robot going out of bounds / hitting boundary wall (-5 marks)

You can choose to forfeit the run if you do not want to incur the penalties, but marks gained for the forfeited run will also be lost

Milestone 4

Correctly picking up a fruit during a later level will automatically give you the marks for the prior levels

You may attempt any level however many times you'd like
When you start a new attempt you must start collecting the fruit from the beginning of the list

Example marking scenarios:

Level 1: Collected first 2 fruits
Level 2: Collected first fruit
Mark: 20 + 20 + 8 = 48

Level 1: Collected first 2 fruits
Level 1: Collected last fruit ONLY
Mark: 20 + 20 = 40

Level 3: Collected first fruit
Level 2: Collected first 2 fruits
Mark: 20 + 8 + 8 + 20 + 4 = 60

Level 1: Collected all 3 fruits but collided with 6 ArUco markers (forfeited)
Level 1: Collected first 2 fruits
Mark: (20 + 20 + 20 – 6 * 5) * 0 + 20 + 20 = 40

Milestone 4

Milestone is focused on navigation and planning

Your task will be to semi OR fully autonomously navigate and path plan around the arena to “pick up” various fruit in a specified order

You can do this by modifying the skeleton code “auto_fruit_search.py” provided to you

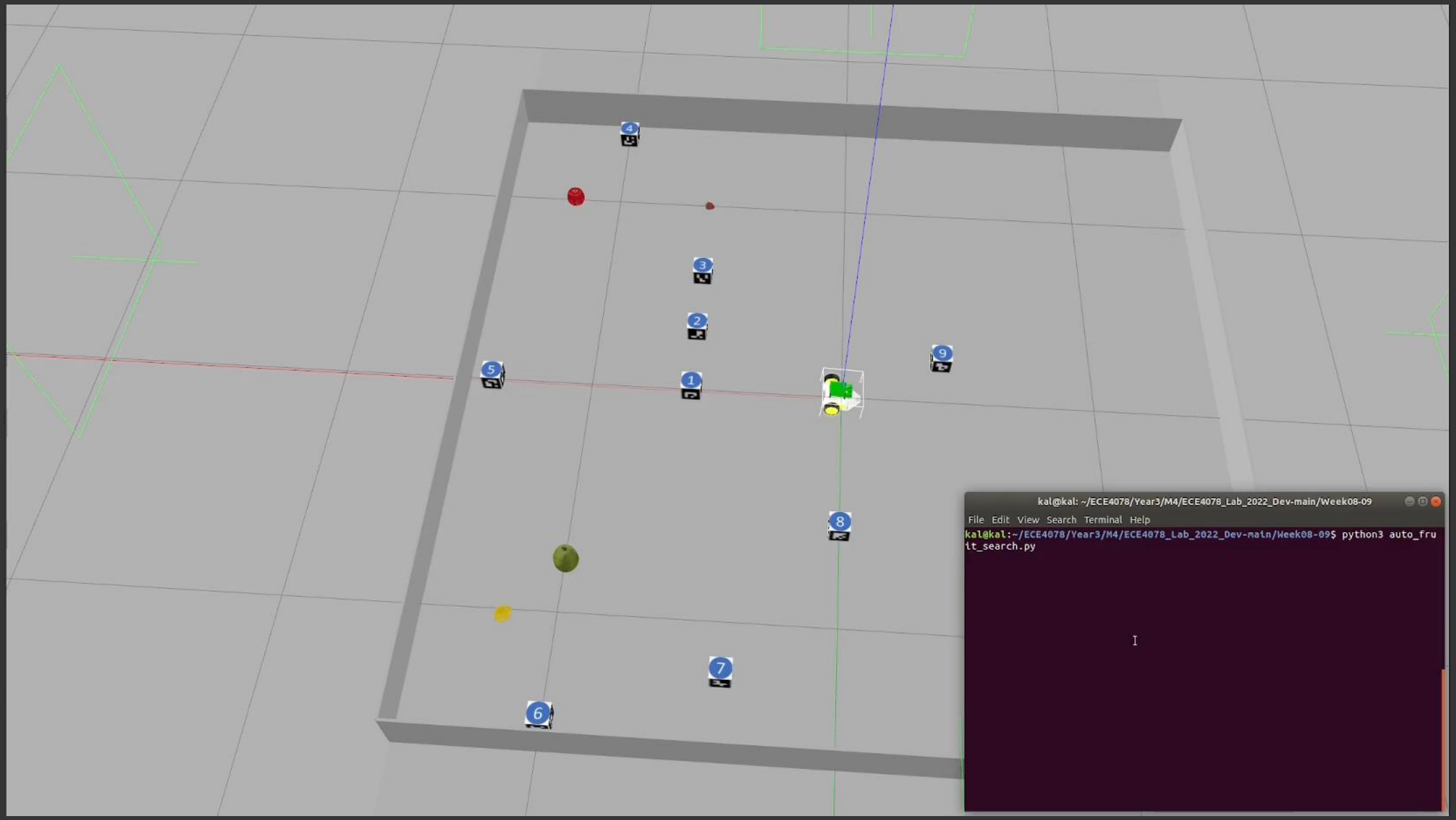
The skeleton code works by:

1: Receiving a waypoint that are inputted by the user
2: Spinning the robot based on the estimated time to face the waypoint
3: Moving the robot forward based on the estimated time to reach the waypoint
4: Repeat step 1 until task is complete

There are 3 things you need to change

1: Calculate the time required to turn
2: Calculate the time required to drive forward
3: Calculate the new pose of the robot

Milestone 4


kal@kal: ~/ECE4078/Year3/M4/ECE4078_Lab_2022_Dev-main/Week08-09
File Edit View Search Terminal Help
kal@kal:~/ECE4078/Year3/M4/ECE4078_Lab_2022_Dev-main/Week08-09$ python3 auto_fruit_search.py

Additional recommendations (optional)

- Visual localisation

  By default, the skeleton code does not use any visual features (ArUco markers) in estimating its current pose, relying solely on dynamics to estimate its current position in the arena.

  It is highly recommended that you implement a more robust localisation system that uses the ArUco markers like you did in M2. You do not have to generate / update the markers as the true map is given to you at the start

- Controller based waypoint navigation

  By default, the skeleton code travels to waypoints based on an initial time estimate.
  A small error in the initial rotation can cause a large error in the final position if traveling long distances

  You can improve this by using a proportional controller to gently steer the robot so that it is always moving towards the waypoint given visual feedback on the robot's pose

- Better GUI

  Manually inputting coordinates by hand is slow and prone to error, an interactive map to click on to create new waypoints would be more efficient
  It is also difficult to tell if the robot is within 0.5m of a fruit or if the robot's pose estimation has drifted without any graphical feedback, therefore this should be included in the GUI as well

Additional recommendations (optional)