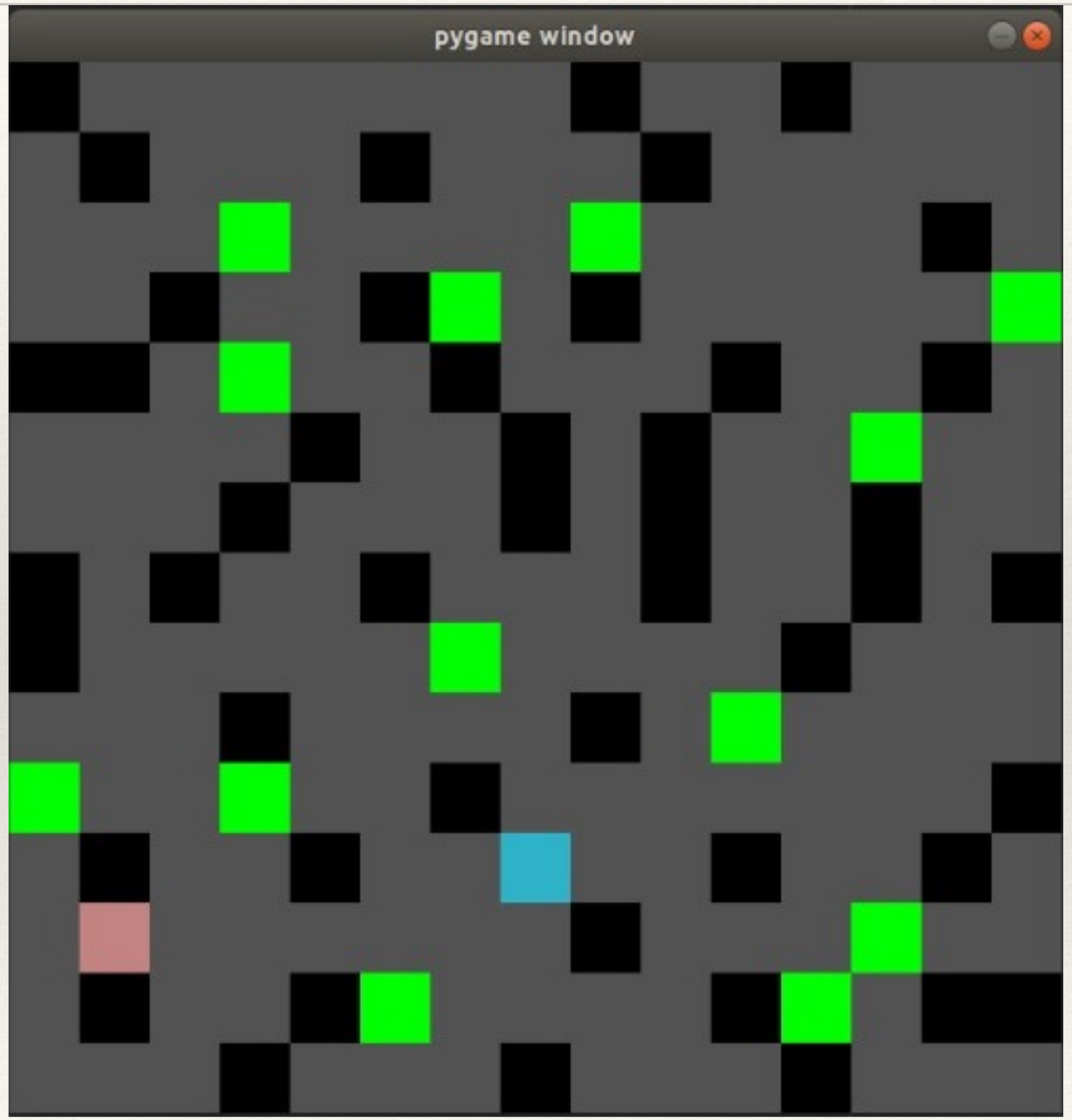
Path Finding by Color Detection

Intelligence Systems and Robotics Laboratory
UNIVAQ - 2020
Dila Aslan
Mustafa Yumurtacı

Introduction

- We aimed to develop a real-world robot that finds all foods in the given random map by using color detection.
- The idea of project come from the 2D Robot World game that we developed for first homework.

Introduction



Why we chose Lego Robot?

- We chose Lego Mindstorms EV3 robot; because it provides easy and powerful API for development.
- It provides flexibility to create any design that we want.
- It has a good dev-community which makes easy to find solutions for problems that we faced.

Robot Specification

- 3 color sensors
- 2 motors
- 1 gyro sensor

Sense

- We used 3 color sensors for detecting the colors in three sides of robot: *Front, Right and Left*.
- In our structure:
- Red color means obstacle
- Yellow color means path
- Green color means food

Plan

- The robot performs reasoning by Prolog. The Prothonics is used to interact with SWI-Prolog.
- *Prothonics is a Python library to create proactive agents, that perform reasoning by using SWI-Prolog.*
- Robot's first priority is food. If it sees a food (green), it goes for that way. It also starts to check sensors by Front-Right-Left order.
- If there is no food in any of three sides, then it checks for path (yellow) and go for it.
- If all sides are obstacle (red), it turns backwards.

Act

- The robot performs 4 type of action; **move forward, rotate right, rotate left and turn backward**.
- The problem in acting part is rotation angle. Because we need certain rotations with specific angles, we had to use the gyro sensor (measures rotation angle) to control if robot rotates exactly the angle we want.
- The gyro sensor measures the angle with 2-3 degree loss. That's why, if we want robot to turn 90 degree, we set it to 87 degree.

Act

- Another problem that we faced while working on the project, if robot rotate right or left, it stays on the wrong square. That's why, if robot is going to perform rotate right or left action, before that, it performs move backward action to take position on the correct point.

Simulation

- To simulate our robot to make sure the logic performs good, we used V-REP with Python API.
- In V-REP, we used same sense and acting code that we used for Lego Robot.
- Only the acting part is different, which sends commands to V-REP using Python API to simulate actions of robot.

Simulation

- The problem with V-REP is, in our case, we used velocity to control the robot. Because of this, we need to set delays until the robot finishes it's movement.
- But the process time of robot action in V-REP is not consistent. Sometimes move forward action with velocity 2.0 takes 0.5 secs, and sometimes it takes 0.45 secs. This problem probably occurs because there is always processes running in OS with different resource usages.
- Because of that problem, the robot actions start to delay after some time, and it causes robot to take incorrect actions.

GRAZIE MILLE!

- DİLA ASLAN
- MUSTAFA YUMURTACI