universidade
de aveiro

Pathfinder solver using the **CiberRato** simulation environment

Diogo Silva
dbtds@ua.pt
60337

Intelligent And Mobile Robotics

2016/11/24

Pathfinder solver using the CiberRato simulation environment

What is the objective?

Objective

- ➔ Robot must find its way to the cheese (target area) and return using the best path, without colliding.

The scenario used contains the following characteristics:

- ➔ Only vertical and horizontal walls;
- ➔ Walls can only be in the margins of every cells (cell size has 2 robot units)

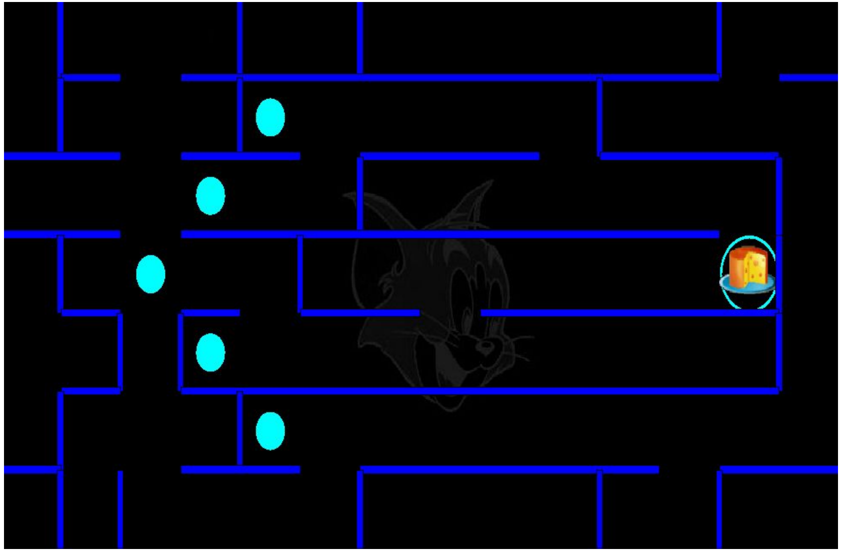


Fig 1. Scenario that loads with cibertools without any robot running.

Robot Schema

Sensors

- 4 obstacle sensors (0, 60, -60 and 180 degrees)
- Ground sensor that is able to detect the target area
- Compass
- Bumper that detects collisions
- Time left for the simulation to end

Actuators

- 2 motors (left and right)

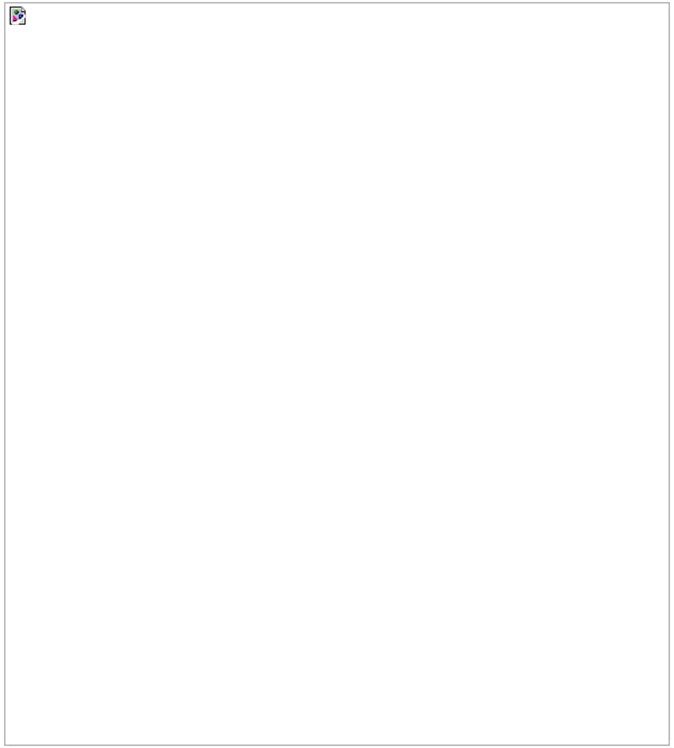


Fig 2. Robot simulated has the following actuators and sensors.

Pathfinder solver using the CiberRato simulation environment

How does the robot behave?

Agent localization & mapping

- It uses the following **movement model:**

$$lin=\frac{out_{\text{right}}+out_{\text{left}}}{2}$$

$$x_t=x_{t-1}+lin*\cos(\theta_{t-1}) \quad y_t=y_{t-1}+lin*\sin(\theta_{t-1})$$

- But it does not use rotation part, it uses **compass** instead

- When **mapping,** it creates a **line of points** (0, -30 and 30 degrees) on the **obstacle sensors** and increases specific counters if it is wall or ground.

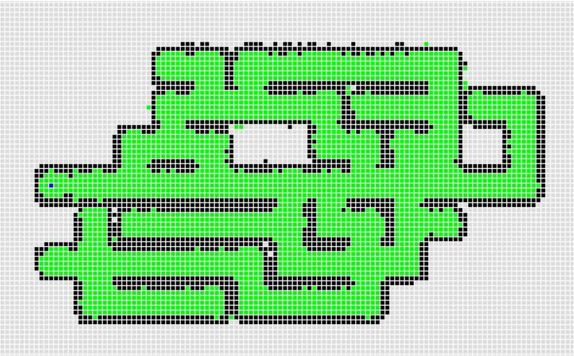


Fig 3. Mapping created by the agent using SDL2

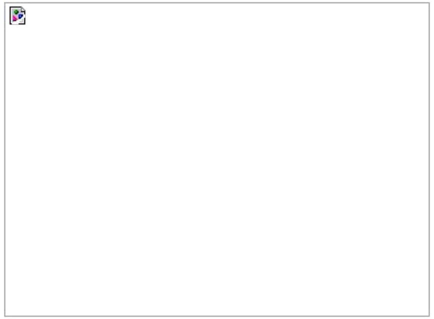


Fig 4. Obstacle sensor schema

Main flow of the agent

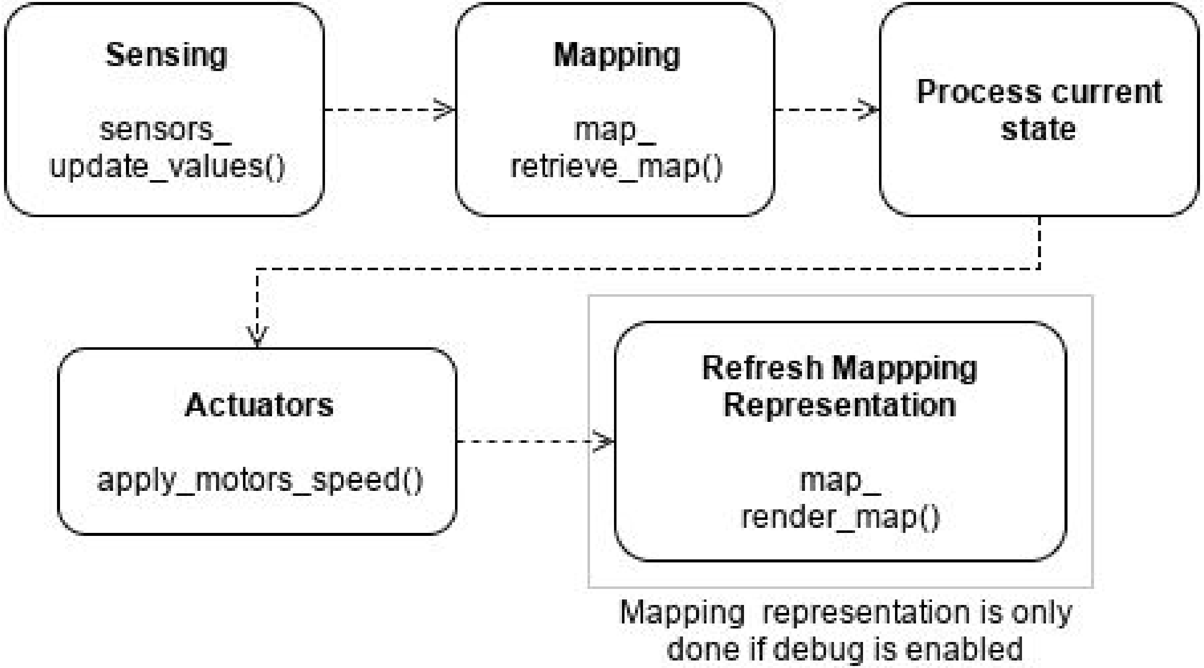


Fig 5. Main flow of the agent

States of the system

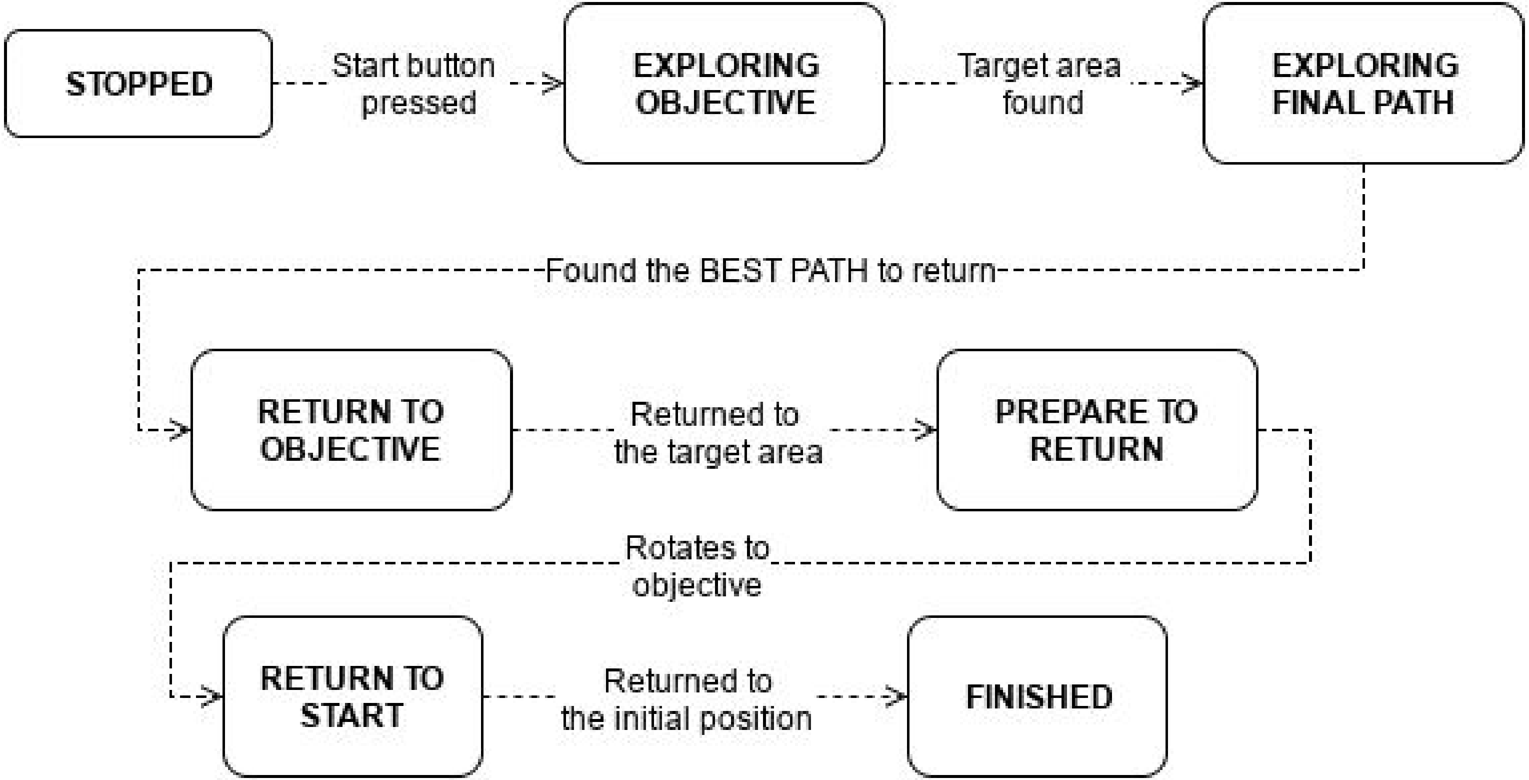


Fig 6. States that the agent uses to fulfill the challenge

Agent state - EXPLORING_OBJECTIVE

- Tries to find the closest exit
  - That exit must contain an **unknown** cell for the robot to explore
  - It uses an **algorithm** that **spread over the known** cells until it finds an **unknown** cell
- It follows a list of points that correspond to the middle of each cell until it reaches the exit
- Uses a controller to follow each point
  - Error is the angle between the points

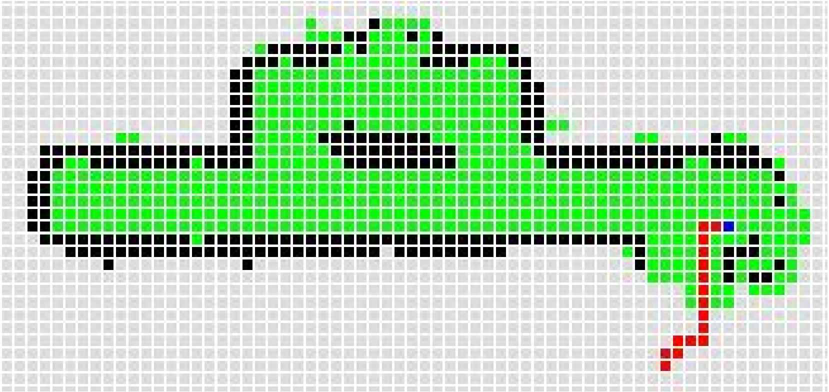


Fig 7. It shows the next step that the agent wants to take

Agent state - EXPLORING_FINAL_PATH

Compares the size of two paths:

- ➔ A-star where it may use **known** and **unknown** cells;
- ➔ A-star where it may only use **known** cells

If they have the **same size**, agent already **knows the best way**. Otherwise, it explores it.

Time left was taken into consideration.

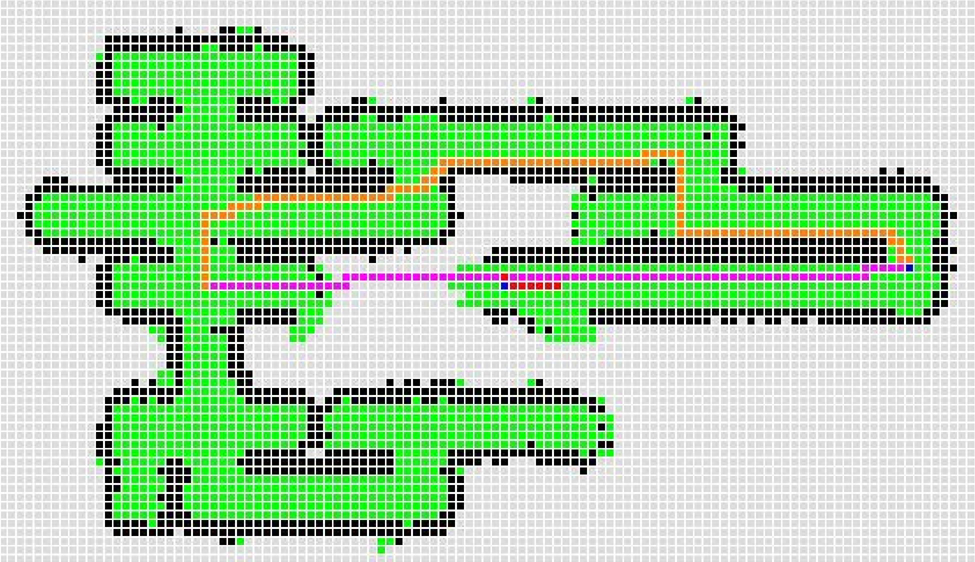
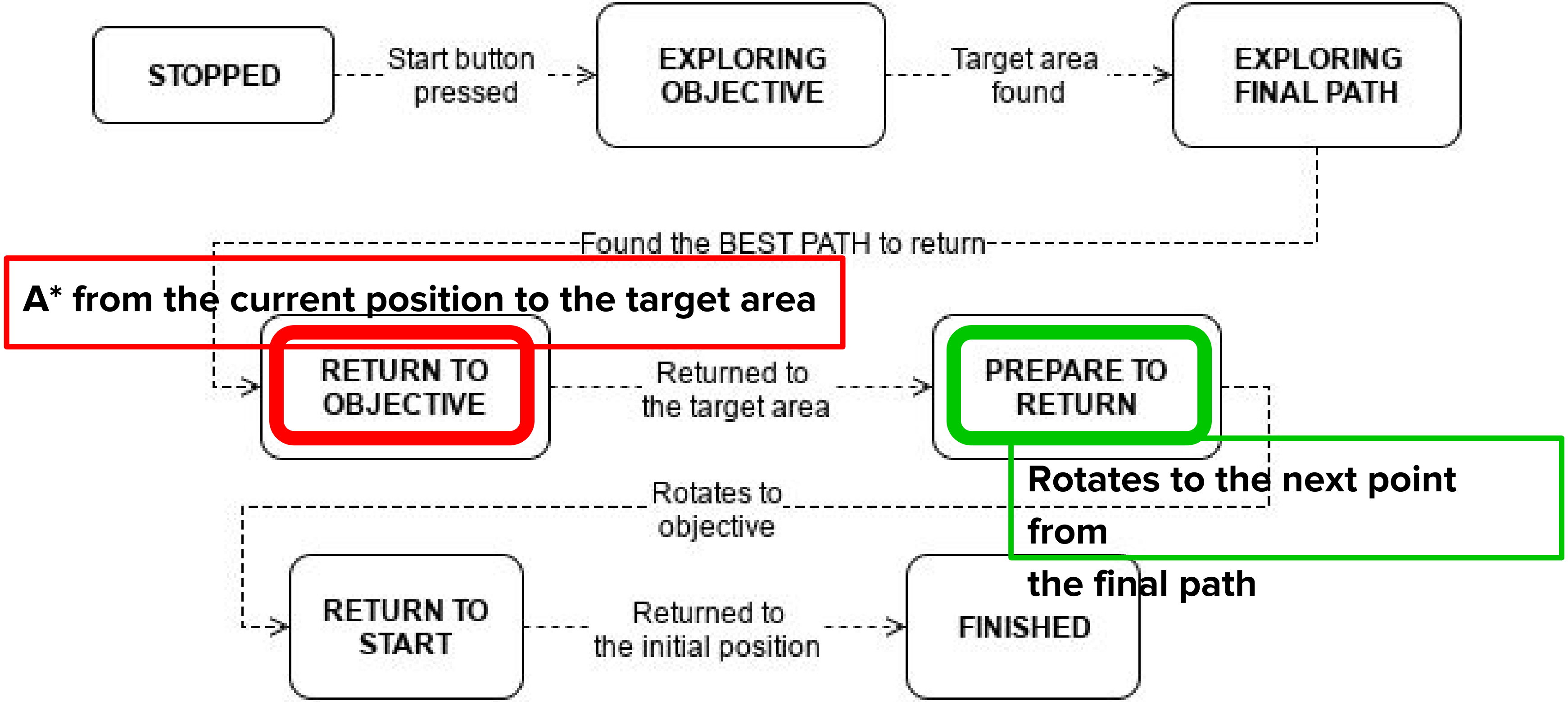


Fig 8. Map showing an example where the agent does not know the best path yet.

States of the system - Review



STOPPED

Start button pressed

EXPLORING OBJECTIVE

Target area found

EXPLORING FINAL PATH

Found the BEST PATH to return

RETURN TO OBJECTIVE

Returned to the target area

PREPARE TO RETURN

Rotates to objective

RETURN TO START

Returned to the initial position

FINISHED

A* from the current position to the target area

Rotates to the next point from the final path

Fig 9. States that the agent uses to fulfill the challenge - Review

Pathfinder solver using the CiberRato simulation environment

Questions?