Project

Autonomous Software Agents

A.A. 2022-2023

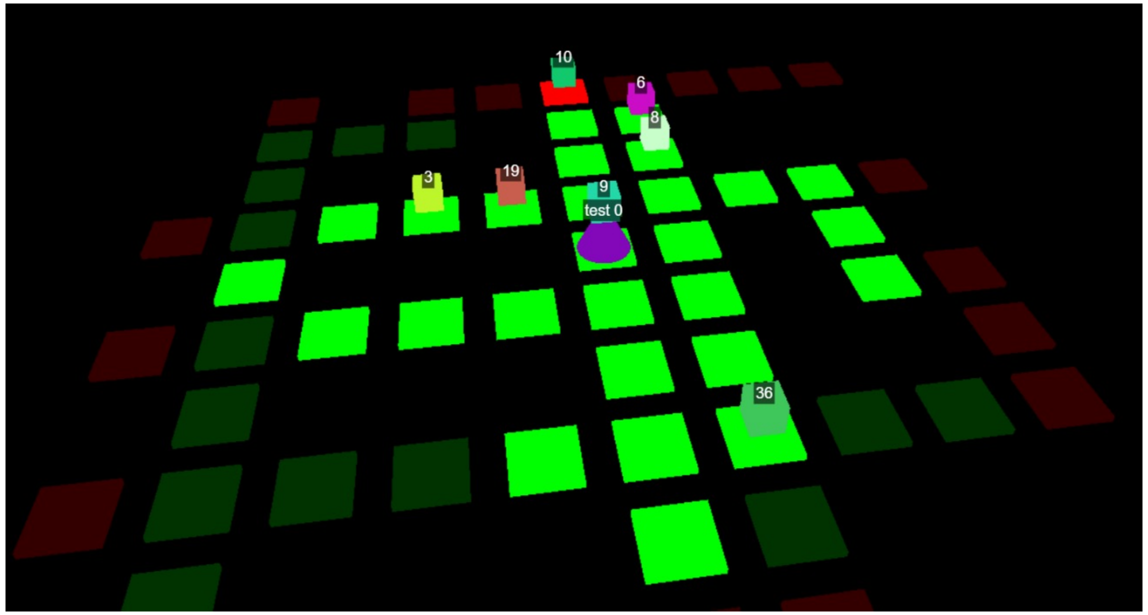
Prof. Paolo Giorgini

Dr. Marco Robol

UNIVERSITY OF TRENTO - Italy

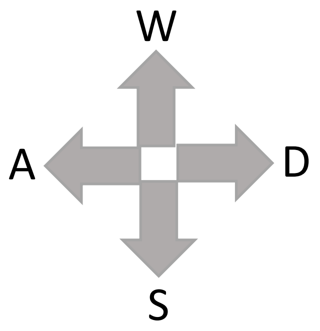
Department of Information and Communication Technology

Deliveroo



Deliveroo.js is minimalistic parcels delivering web-based game developed specifically for this course and for educational purposes

- https://deliveroojs.onrender.com/



Q: pick_up
E: put_down

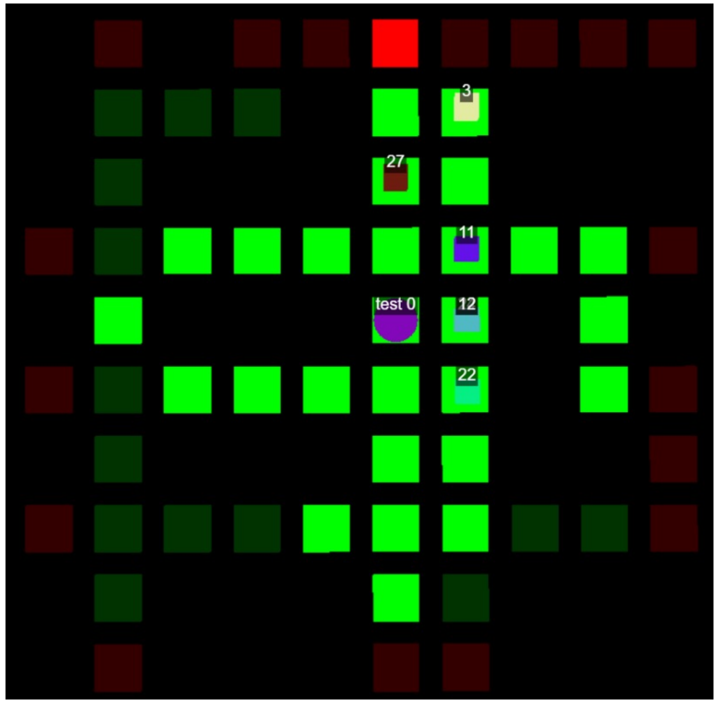
The objective of the project

- Develop an autonomous software that will play on your behalf
  - earning points by collecting as many parcels as possible and delivering them in the delivery zone
  - Using a BDI architecture
    - Sensing the environment and managing Beliefs
    - Deliberating Intentions
    - Select plans from a library
    - Using an external planner component
    - Execute a plan (actions)
    - Defining strategies and behaviours
    - Replanning and redeliberating

© P. Giorgini and M. Robol

The Game – environment

- Grid MxN + Blocked Tiles
- Borders of the grid are delivering zones
- Parcel
  - Located in a tile (x,y)
  - Appear in the grid with a Timer
  - More than one parcel at a time can appear during the game
  - Disappear when the timer expires or when it has been delivered
  - Can be picked up or put down in any free tile by a player
  - Can be carried by a player



Example map on a 10x10 grid as sensed by the 'test' player in the middle

The Game - players

- **Players** can do:
  - Actions: {`move_right,move_left,move_down,move_up,pick_up, put_down`}
  - A player can pick up and carrying more than one parcel at a time
  - Sensing: acquiring information within a limited range (max distance: x_offset + y_offset < 5; shown in the game as light/shaded tiles)
    - `Parcel sensing: {id, x, y, carriedBy, reward}`
    - `Player sensing: {id, name, x, y, score}`
    - `myPos(X,Y), myScore(S) => You: {id, name, x, y, score}`
- **The Game**
  - The objective for a player is to gain points picking up and delivering parcels
  - Each delivered parcel counts as much as its remaining time
  - Game Clock is used to update parcels' timer and duration of actions

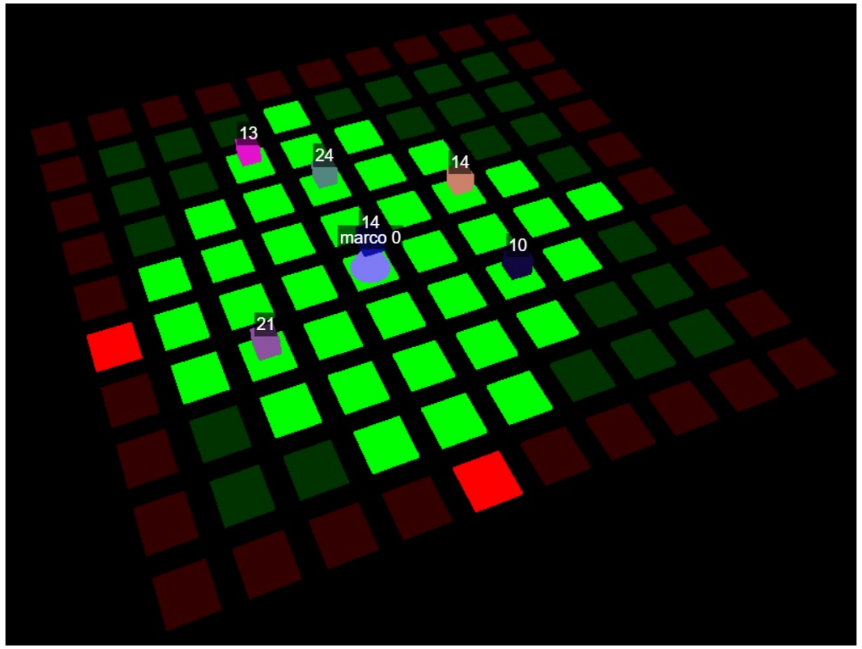
Actions

- Moving actions are not instantaneously
  - They have a fixed duration
    - When moving action starts, player is moved in between the initial and the target tile
      - By updating (increasing or decreasing) players’ coordinate (X or Y) of 0.6
    - Then, when moving action ends, movement on the target tile is completed
      - By moving player of another 0.4
    - Therefore, it is possible to know whether a player is moving in one direction or another.
  - During the execution, starting and ending tiles are locked (they cannot be occupied by other players)
  - A player cannot move in a tile occupied by another player
    - The moving action will fail
  - Starting from (x,y) - right: x+1, left: x-1, up: y+1, down: y-1
- Pick up and put down are instantaneously
  - A player can pick up and put down a single parcel at a time
  - To pick up a parcel, the player has to be in the same tile of the parcel
  - A player can put down a parcel everywhere
    - The player gains points only putting down a parcel in a delivering area

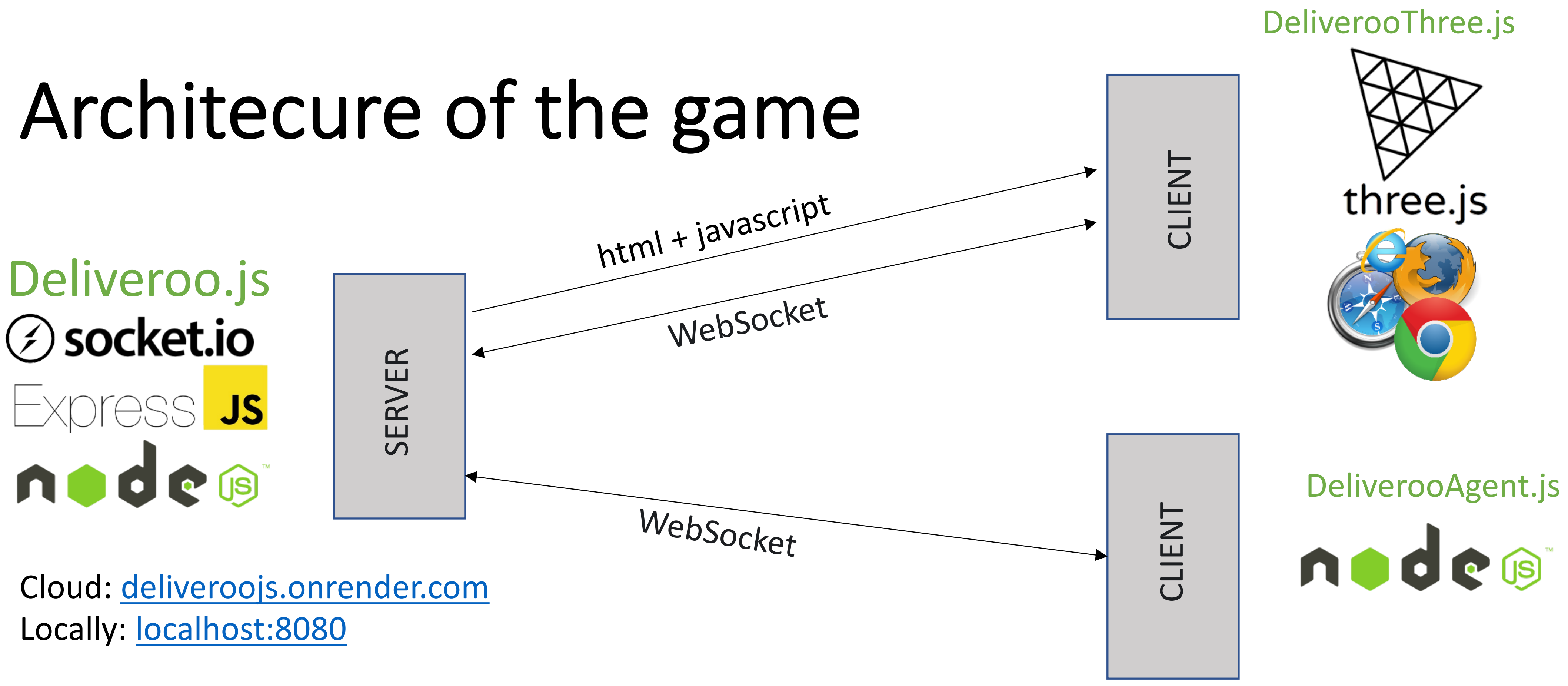
Sensing

- On connection, the client receives game map as a list of tiles
  - `tile: {x, y, blocked}`
- Then, every time something happens (parcel timer decades, player moves,…) the client receives the list of players and parcels sensed with position and reward/score information
  - `Parcel sensing: {id, x, y, carriedBy, reward}`
  - `Player sensing: {id, name, x, y, score}`
  - `You: {id, name, x, y, score}`
- Everything outside from the sensed area cannot be known
  - Position of previously observed players can be guessed
  - Reward timer of parcels can be computed locally

Sensing is limited to area within:
x_offset + y_offset < 5



Architecure of the game



Cloud: deliveroojs.onrender.com
Locally: localhost:8080

- Deliveroo.js is a web application based on Node.js and Socket.IO.
- Connection between client and server is based on Socket.IO and WebSocket.
- DeliverooThree.js is a 3D client that can be played directly from the browser.
  - Useful to test game strategies and see how the game is eveolving
  - No game logic is coded in the client, core of the game is on the server.

Game authentication

- A token is needed to connect to the game
  - Every unique token is associated to one player
  - Tokens can be easily created through the 3D client, by specifying a name for the player. Just follow the instructions appearing in the browser when connecting to the game for the first time. Different players with different tokens can still have the same name (the id is different)
    - Tokens are signed by the game using a passphrase and may not be usable across different instances
    - Tokens do not expires
- Use the same token on your agent and 3D client:
  - Follow your agent (script) playing from its very perspective
- No limit on the number of tokens, get as many as you want
  - When accessing the game with the 3D client, the browser store the token as a cookie, to facilitate log-in procedure. You can still remove or replace them
  - In the case of no clients connected for a given token, after 20 seconds the NPC is removed from the grid until reconnection
  - Multiplayer mode is possible over the network (and even from the same computer)
    - Connect to the same game host and use different tokens
    - If the same token is used, players will control the very same character

The project

- First part
  - Development of an autonomous agent that will play on your behalf
    - Represents and mange Beliefs from sensing data (including belief revision)
    - Activate goals / Intentions and act on the environment (including intentions revision)
    - Use predefined plans to achieve goals / intentions (parcels are known since the beginning)
  - Extend the above software with automated planning
    - Once an intention is activated the agent must call the planner to have the plan to execute
  - Validation and testing with predefined simulation runs (all together)
- Second part
  - Extend your software including a second autonomous software agent
    - The two agents have to communicate one another
    - They can exchanges beliefs (e.g., beliefs about the environment that other cannot see or beliefs about the intentions one agent is committed to, and so on)
    - They can coordinate (e.g., the closest agent will commit to pickup a new parcel)
    - They can negotiate (e.g., if you delivery this parcel for me I will go to pickup a new parcel)
  - Validation and testing with predefined simulation runs (all together)

© P. Giorgini and M. Robol

Project delivery

- Javascript code
- Final Report
    - Max 10 pages, which explains what you have done and how
- Both code and final report will be evaluated for the exam
    - Oral presentation of the project
- We will provide you the link where you can submit your project
    - At least 1 week before the exam (the exact deadlines will be provided to you later in the course)

Let’s play (on the cloud) – use Chrome!

- Open https://deliveroojs.onrender.com/
    - Or https://deliveroojs2.onrender.com/
    - Or https://deliveroojs3.onrender.com/
  - Put your name and retrieve your freshly generated token.
  - Play one against others, let’s see how it works.
- Try use the direct link to play with your own character
  - https://deliveroojs.onrender.com/?name=yourName

Running the game locally

- Download the game from Github
  - Git clone https://github.com/unitn-ASA/Deliveroo.js.git
- Install packages and run the server
  - npm install
  - npm start
- Play from the browser
  - http://localhost:8080 or the ip address of your colleague
- The file my_map.js can be modified to configure the game.
  - A different map can be defined
  - Parcel spawning logic
  - Actions duration

Getting started with scripting your NPC

- Have your local Deliveroo.js server running or play on the cloud
- Download the “draft” of scripted agent from Github
  - Git clone https://github.com/unitn-ASA/DeliverooAgent.js
- Install, setup connection parameters, and run the client agent
  - npm install
  - Edit ‘host’ and ‘token’ in config.js (get your token from the 3d client!)
  - node .lib\deliverooClient\examples\raw_socketio_agent.js
  - node .lib\deliverooClient\examples\client_agent.js
- Check in the browser what the agent is doing