VRIJE UNIVERSITEIT AMSTERDAM

TASK 1 - OBSTACLE AVOIDANCE

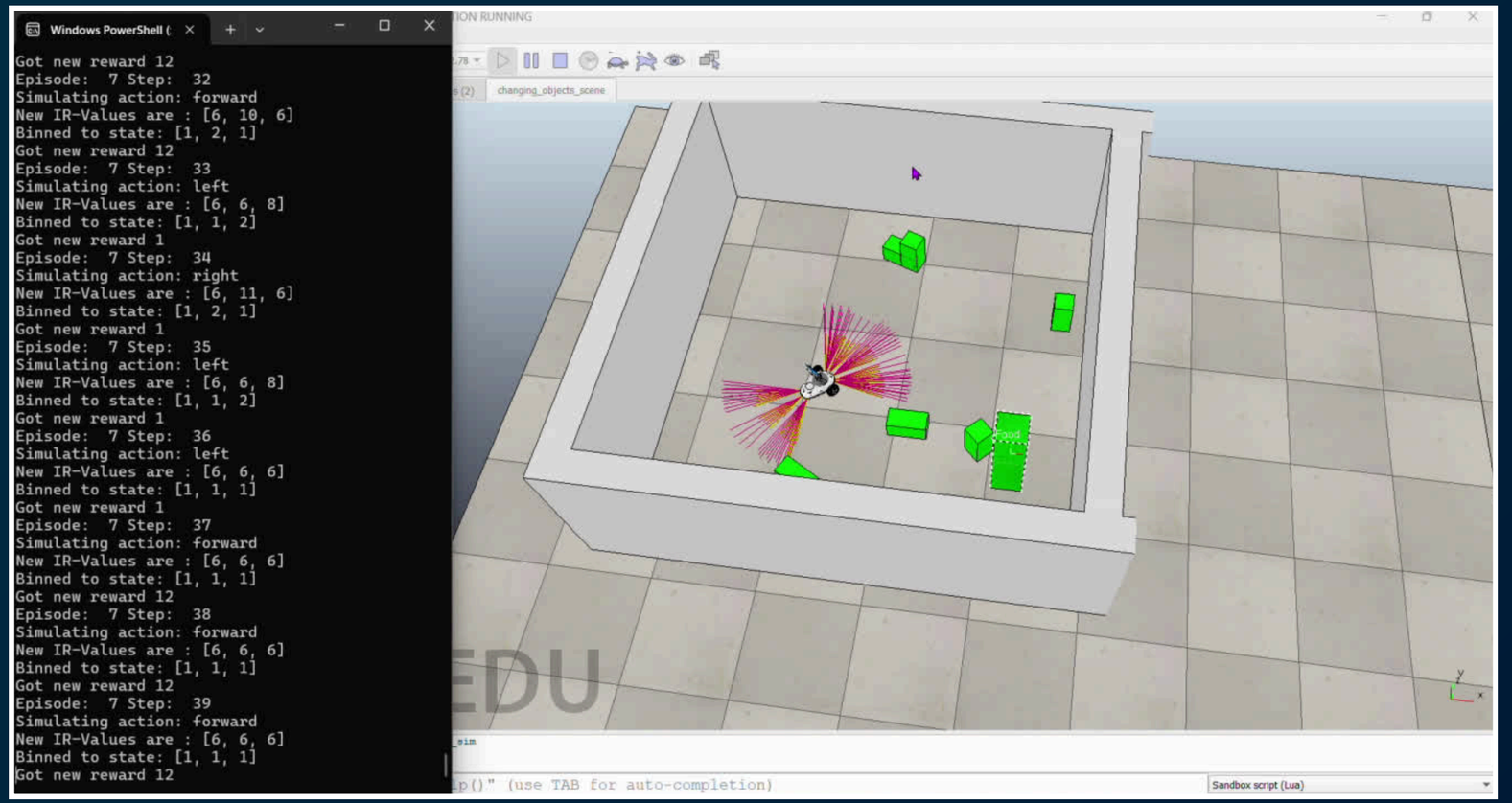
LEARNING MACHINES Robobo

GROUP 14

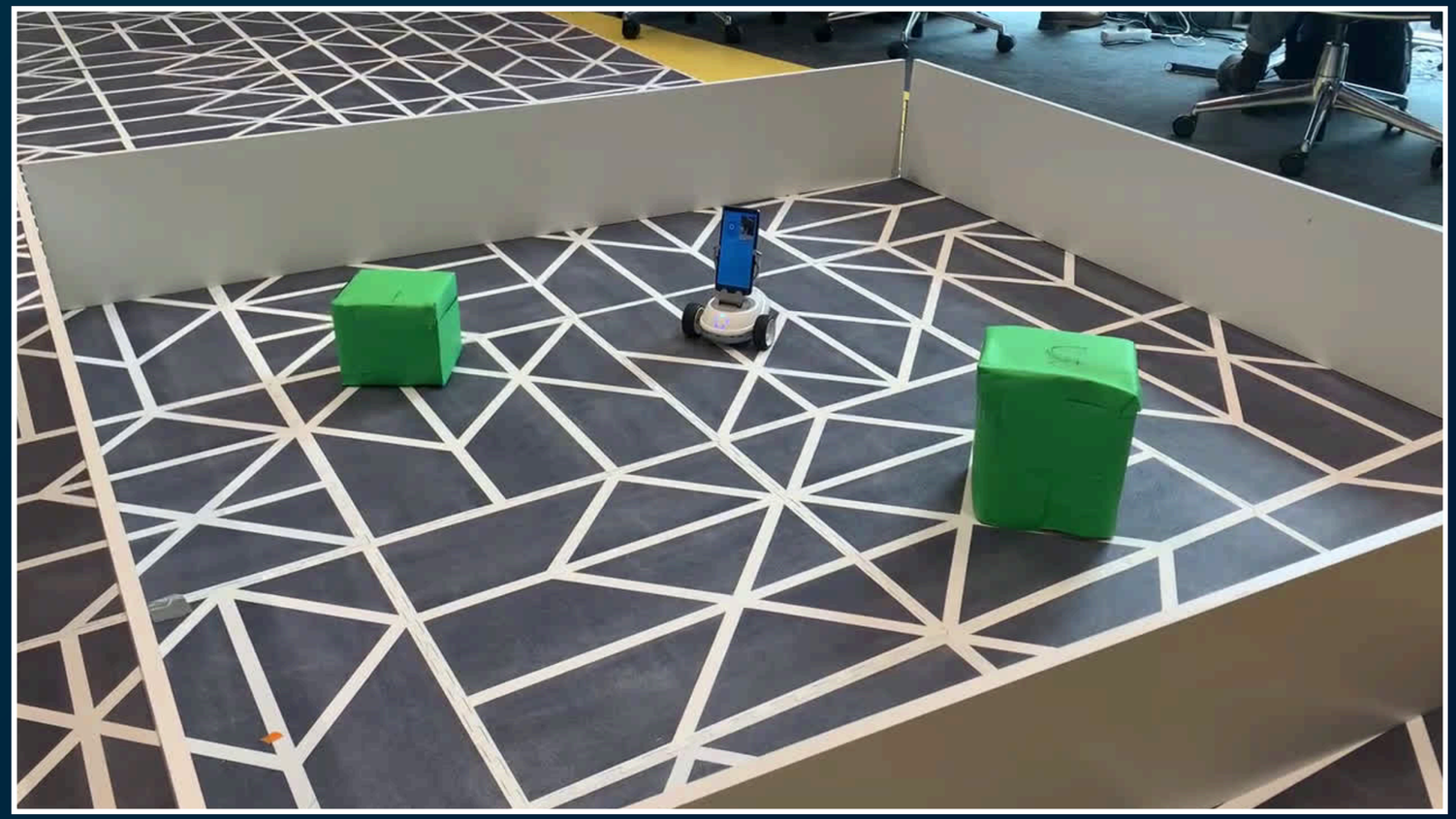
Stergios Ntanavaras
Thijs Grootjans
Quirinus de Ruijter

VRIJE UNIVERSITEIT AMSTERDAM

SIMULATION DEMO



HARDWARE DEMO



PROBLEM & MOTIVATION

- **Objective:** Train a robot to navigate an environment filled with static obstacles efficiently and with minimal collisions.

- **Challenges:**
  - Navigating cluttered spaces demands advanced sensory and processing skills.
  - Requires quick-decision algorithms to prevent collisions and enable real-time reactions.
  - Ensure robust performance despite hardware and software constraints.
  - Operates within sensory and computational limits.

PROBLEM & MOTIVATION

- **Utility and Importance to AI Community:**
  - Pushes forward the capabilities of autonomous robots, in navigation and real-time processing.
  - Promotes research in AI on learning efficient navigation strategies, sensory integration, and obstacle avoidance.
  - Development of more sophisticated decision-making models that can generalize from simulation to real-world scenarios.
  - Insights gained can benefit other fields such as autonomous driving, drone operation, and even video game AI.

METHODOLOGY

- **Q-learning :** A model-free Reinforcement Learning algorithm
  - *Purpose:* It helps the agent/robot to learn making optimal decisions by estimating the utility of actions in given states.
  - *Key Components:*
    - Q-values: state-action value function.
    - Q-table: Represents the expected future rewards for an action taken in a given state.
    - State: The current conditions under which the robot is operating.
    - Action: A set of all possible actions the robot can take.
  - *Learning Process:*
    - Initializes Q-values : In our case 0.0
    - Uses the Q-values to guide actions and update them based on rewards received and future ones.

METHODOLOGY

- The Q-learning algorithm updates the Q-values with the following formula:

$$Q(s, a) = Q(s, a) + \alpha [R(s, a) + \gamma \max Q'(s', a') - Q(s, a)]$$

- *Sensing and Reality Gap:*
  - 3 Infrared (IR) sensors were used to perceive the environment : Front-Left, Front-Center, Front-Right
  - Binning Logic was used to simplify the state space for Q-learning:
    - Sensor readings: [15, 5, 5] → Discrete bins: [2, 1, 1]
    - Sensor readings: [50, 20, 5] → Discrete bins: [3, 2, 1]
  - Bin-values: 0 - ‘no detection or sensor error’, 1 - ‘detected far’, 2 - ‘detected close’, 3 - ‘collision’.
  - Different thresholds for simulation and hardware, due to Reality Gap

EXPERIMENTAL SETUP

- Three actions *(‘forward’*, *‘left’*, and *‘right’)* were used determined by sensor-based state readings.
- The *‘Epsilon-Greedy Algorithm’* was used for action selection.
- 3 sensors were used: Front-far-Left, Front-Center, Front-far-Right.
- Different bin-thresholds were used for the simulation and the hardware ([4, 7, 25] [-1, 15, 100]).
- 4-6 bins were used per sensor. ([4, 7, 15, 30] [4, 7, 10, 15, 25])
- Up to 40 steps were used per episode.
- 100 episodes were used for training.
- An *epsilon* of 0.1 was used, allowing for 10% random action selection.
- A forward reward of 11, and a hitting penalty of -50 were used.
- *Training:*
  - Conducted in simulation.
- *Evaluation:*
  - Total rewards collected per episode to measure the effectiveness of navigational decisions.

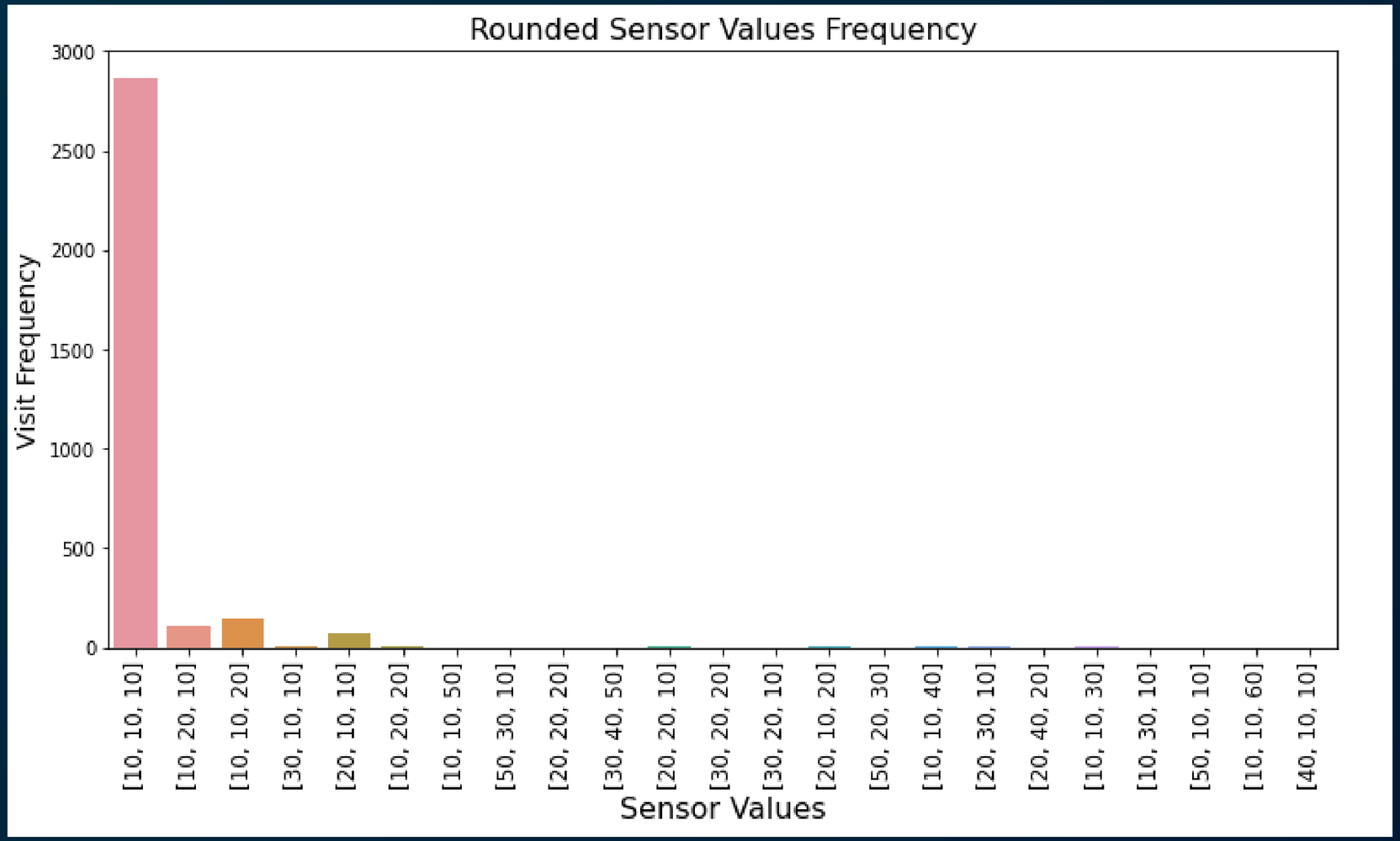
EXPERIMENTAL SETUP

- Challenges Encountered: The robot always moved to the right, avoiding obstacles, due to a low default reward of 1, which encourages actions that prolong survival without hitting objects.

- Solution Implemented: We increased the reward for moving forward, pushing the robot to choose forward movements over other directions.

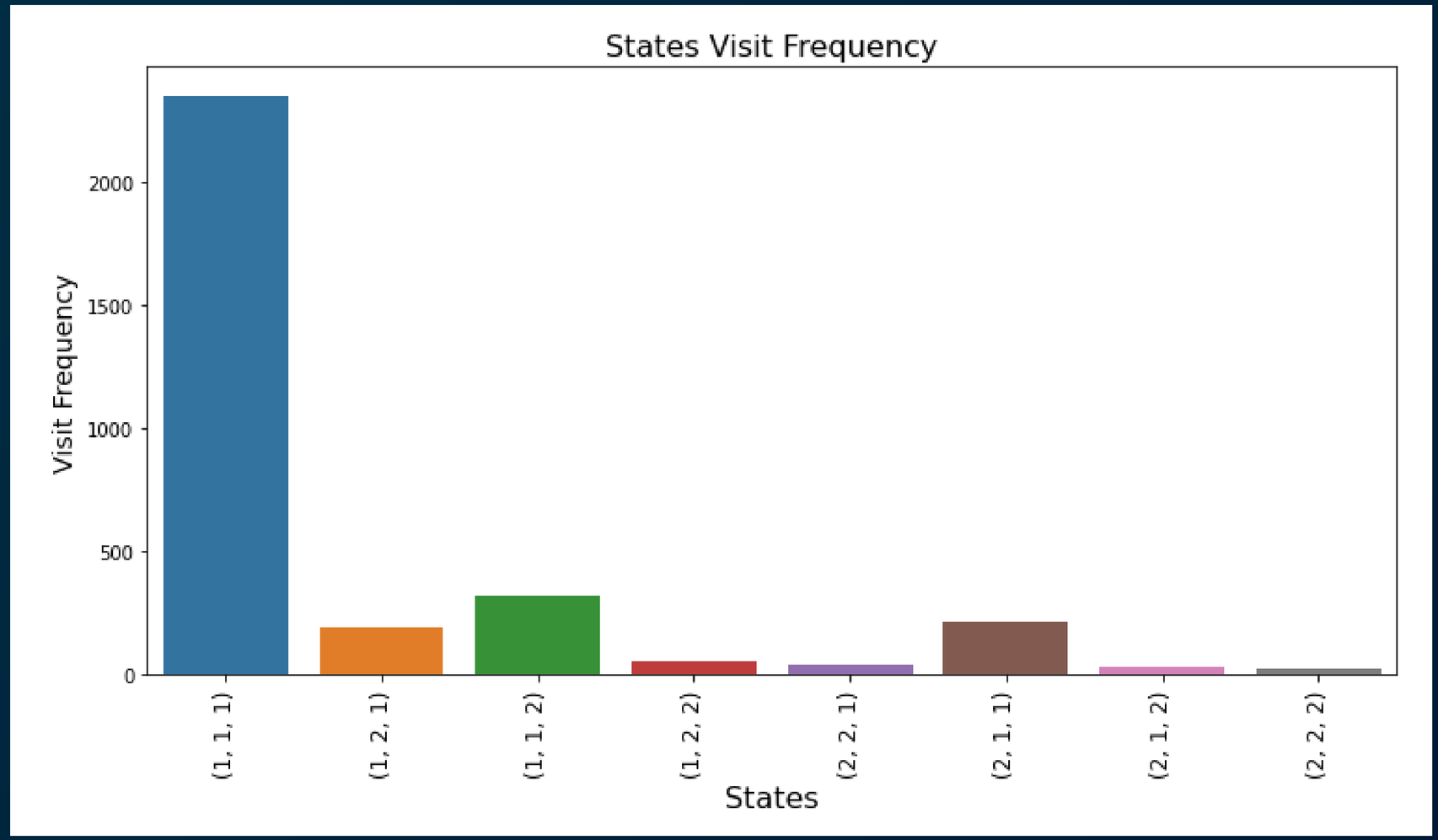
RESULTS: WHAT DO THE SENSOR DETECT?

VRIJE UNIVERSITEIT AMSTERDAM



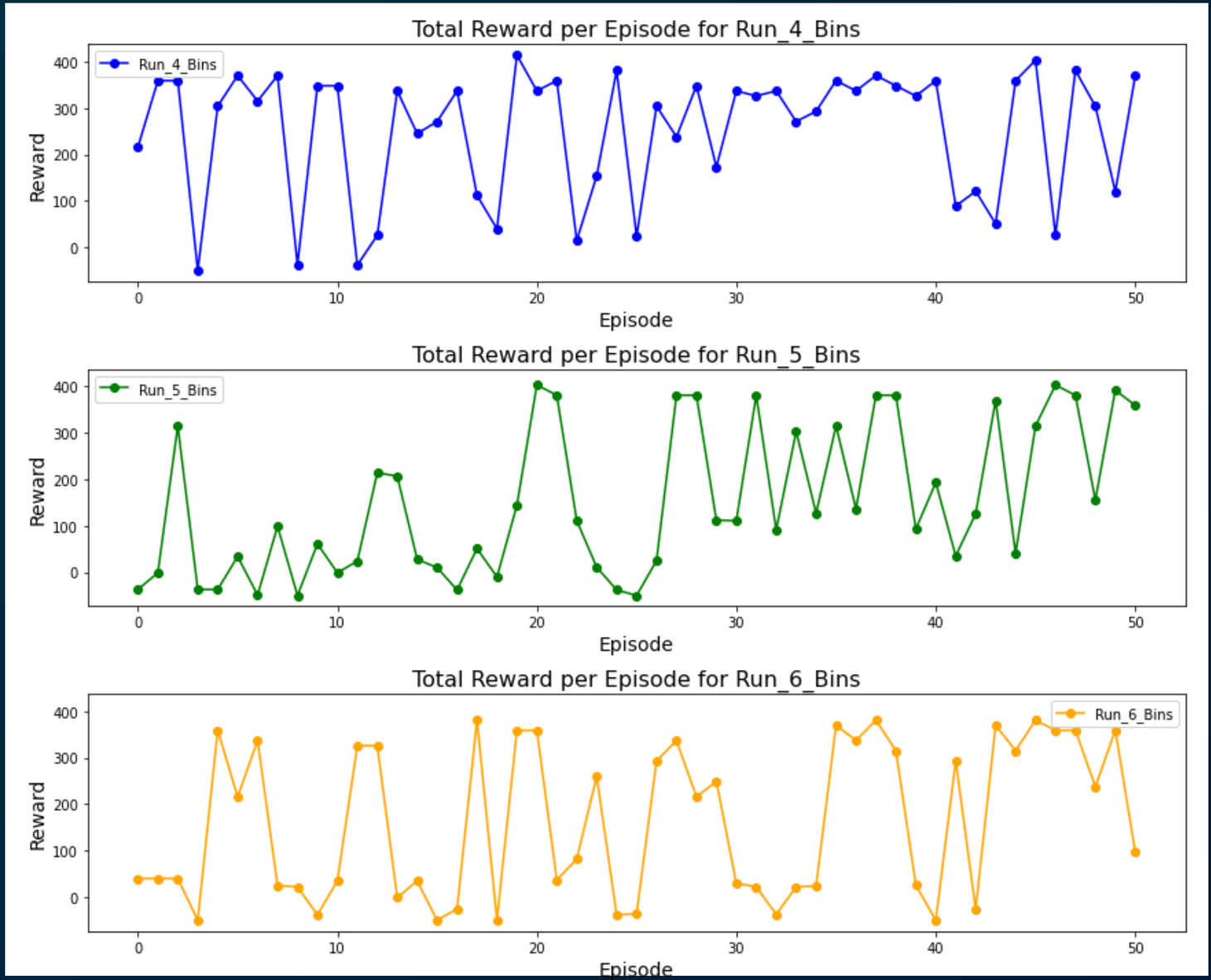
RESULTS: WHAT DO THE SENSOR DETECT?

VRIJE UNIVERSITEIT AMSTERDAM



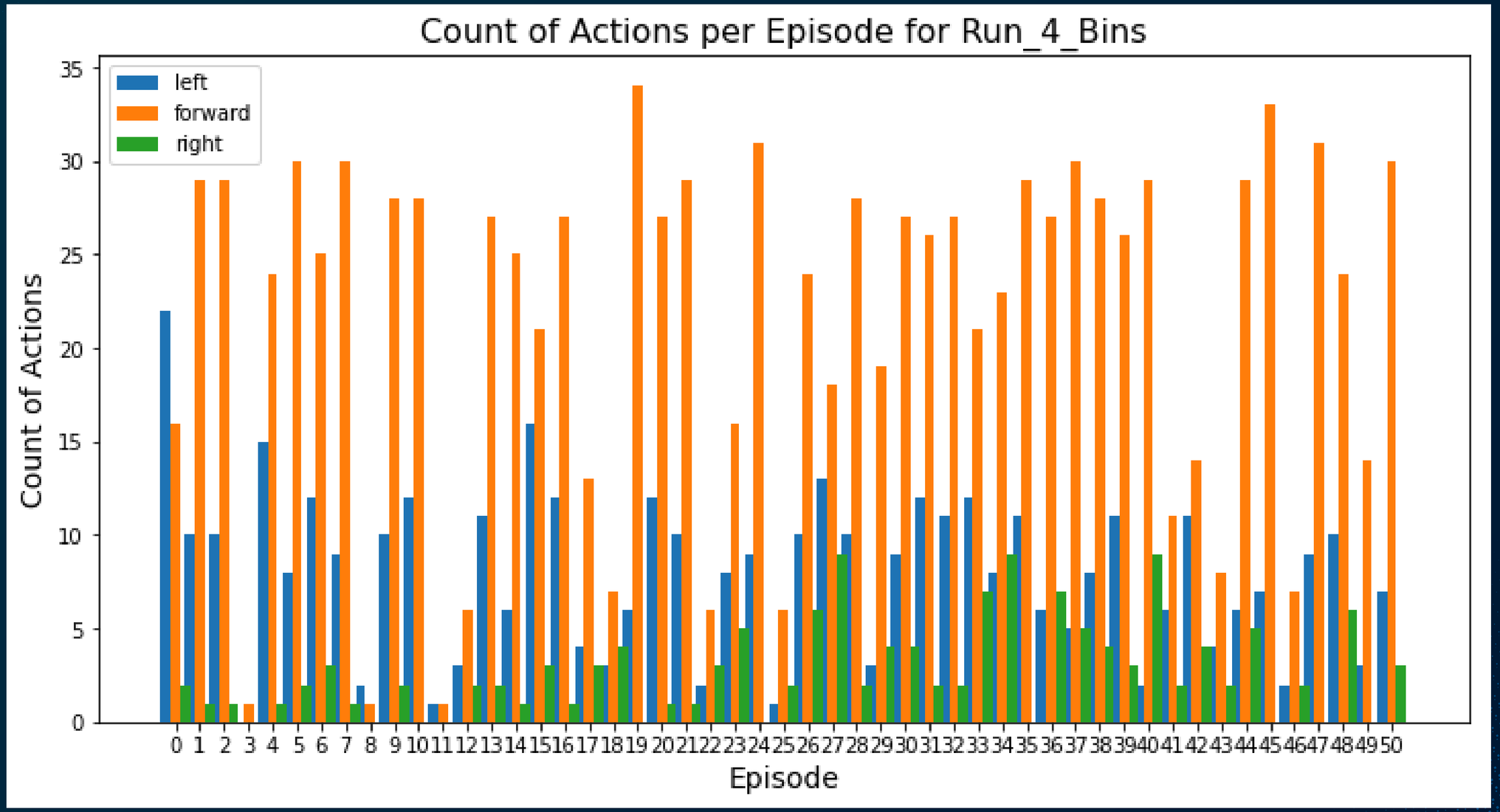
RESULTS: HOW FAST DO WE LEARN?

VRIJE UNIVERSITEIT AMSTERDAM



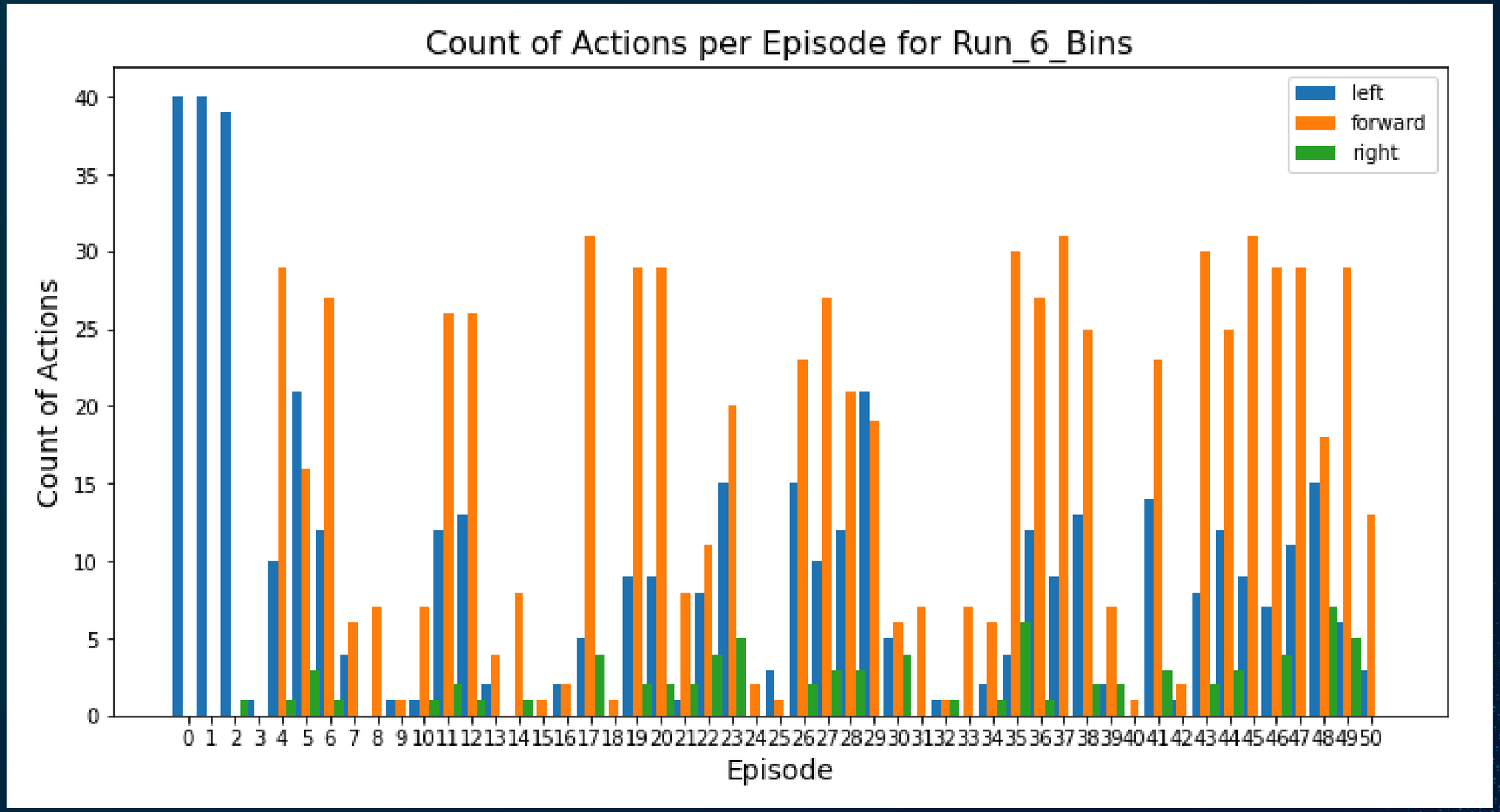
RESULTS: HOW FAST DO WE LEARN?

VRIJE UNIVERSITEIT AMSTERDAM



RESULTS: HOW FAST DO WE LEARN?

VRIJE UNIVERSITEIT AMSTERDAM



RESULTS: DO WE UNDERSTAND Q-TABLE?

(1, 1, 1)	[70.08844812904618, 83.85421854661962, 69.56212245613692]
(1, 1, 2)	[74.96237752355954, 28.122589864263055, 12.049834933493845]
(1, 2, 1)	[63.96763601338395, 6.562652192401206, 1.4003841897522982]
(1, 2, 2)	[60.73889384420776, 4.030099707645284, -0.06031122302687330 6]
(2, 1, 1)	[15.72534435158139, 37.157011168023246, 71.05373259919129]
(2, 1, 2)	[46.2882571489996, 9.941255095388183, 5.57402016717459]
(2, 2, 1)	[-0.4400359133515876 5, -4.691894377909125, 44.17709271334031]
(2, 2, 2)	[18.033189466463 17, -15.504303200000003, -2.510132534457830 5]

CONCLUSIONS

- The Robot learns to **not** hit anyting in the first +-50 episodes.
- There is not really a difference between the number of bins chosen, 4 is good enough.
- Hardware works with selecting right Bin thresholds.
  - Could be done automatically by first calibrating sensors, then mapping to bin.
  - Too much time consuming.
- Future work:
  - Try to implement a Deep RL algorithm.

THANK YOU

QUESTIONS ???

VRIJE UNIVERSITEIT AMSTERDAM

CONTRIBUTION SHEET

- **Stergios:**
  - Created the structure and content for the presentation slides.
  - Created a new obstacle scene in CoppeliaSim.
  - Experimented with the code for training the robot (action selection, binning logic, reward function).

- **Thijs:**
  - Set up for initial code.
  - Experimented with IR values.
  - Experimented with reward function.
  - Experimented with hardware.

- **Quirinus:**
  - Improved on initial code, especially the usage of the right sensors, introducing constans in code.
  - Created code for storing data while running.
  - Created plots from saved data.

VRIJE UNIVERSITEIT AMSTERDAM

BUDDY-GROUP SUMMARY

- **Stergios:**
  - Better understanding of how a team works.
  - Diverse perspectives improved my ability to see multiple sides of a situation before making decisions.
  - Learned how to implement a reinforcement learning algorithm to a real-world scenario.
  - Learned more about the functionalities of the robot (navigating and sensing).

- **Thijs:**
  - Diverse Perspectives.
  - Learned practical techniques for minimizing the reality gap in simulations.

- **Quirinus:**
  - Learned a lot from how to set up and work in coding projects like this.
  - Got better understanding of using Git (not there yet haha).
  - We started working together better with a plan and some task division.
  - Learned about group behavior and interactions.