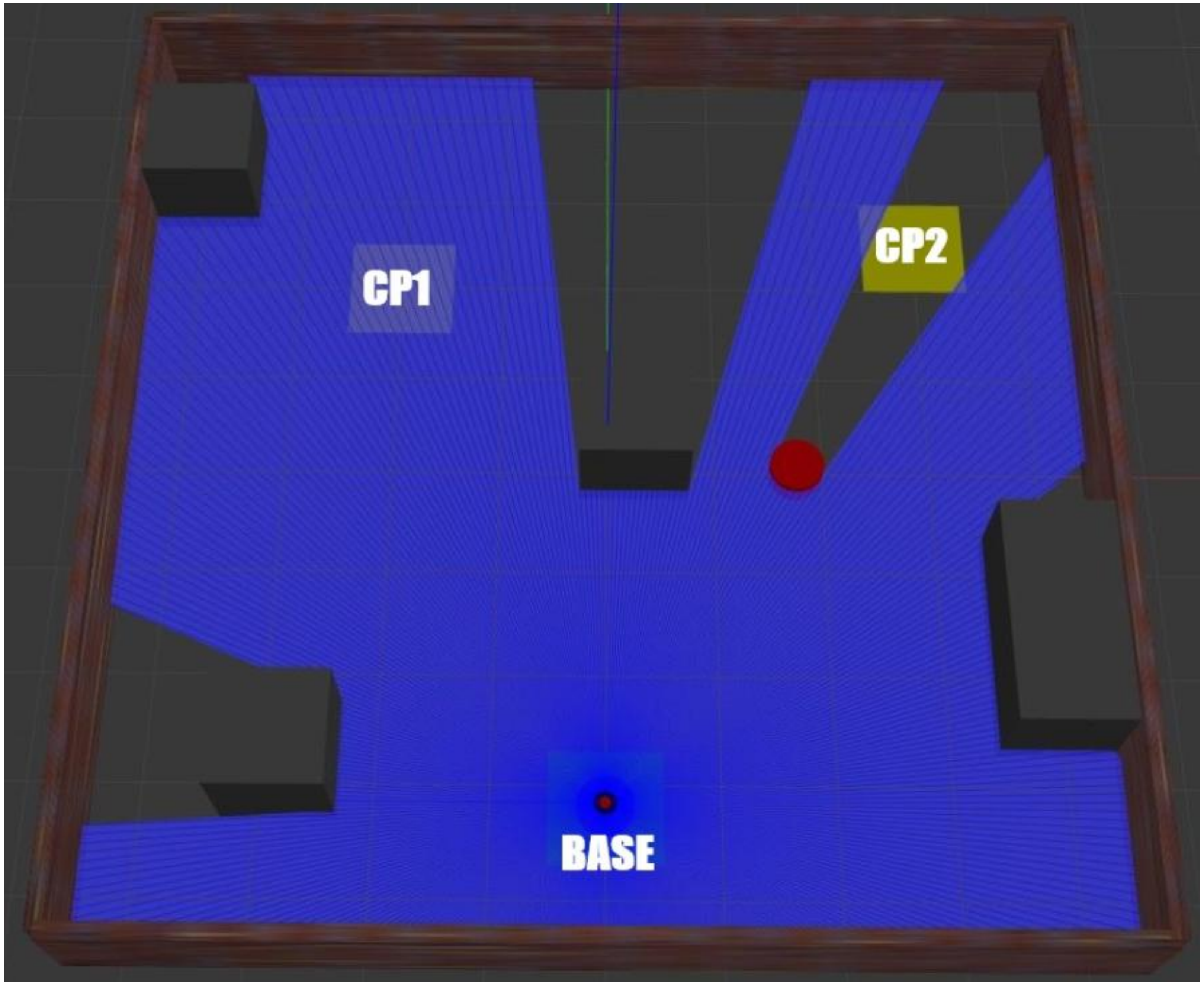
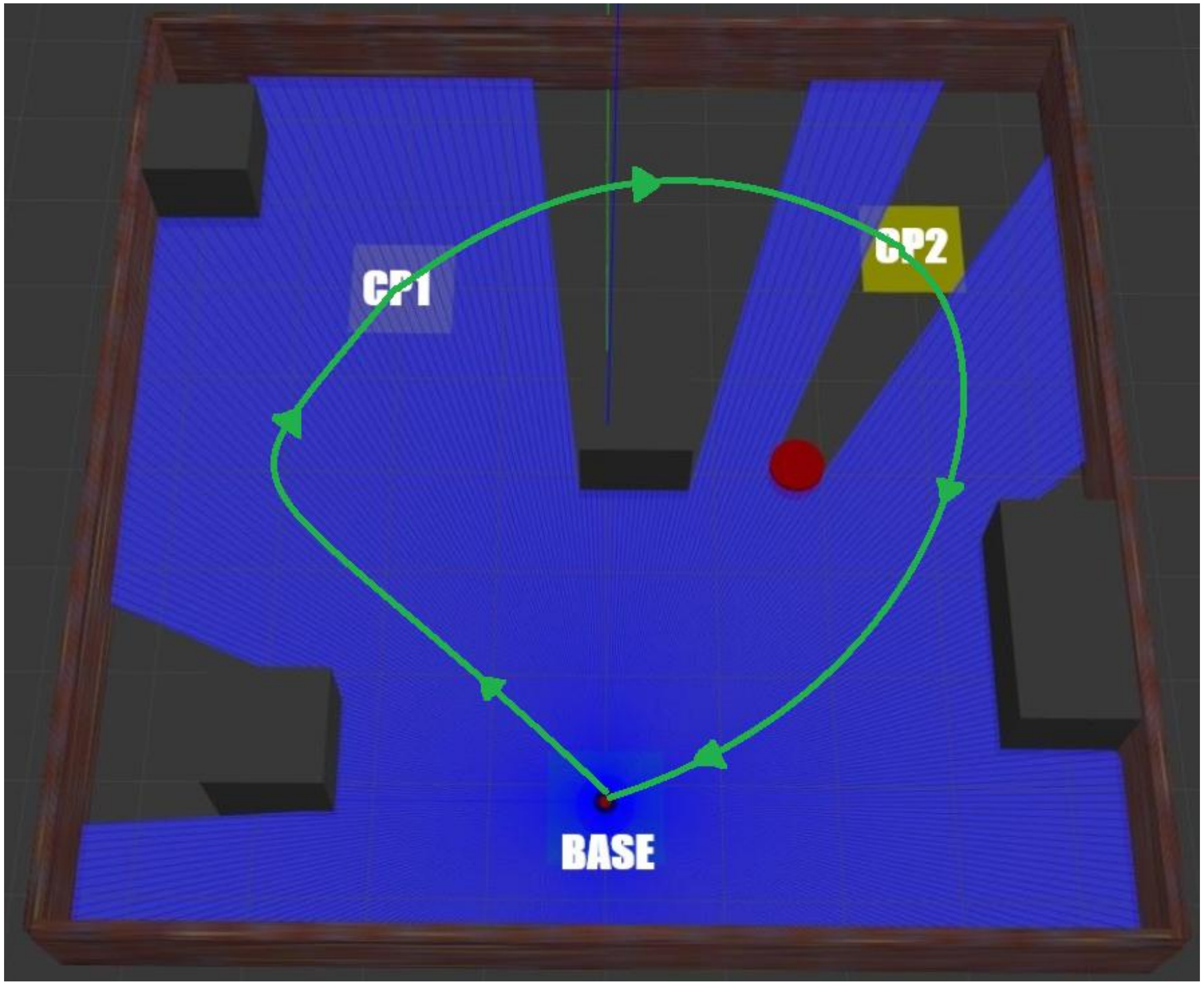
Robot Navigation with Reinforcement Learning (SAC) & Dynamic Obstacle Avoidance

DONE BY: MAKOWSKI MICHAŁ

Map Visualization



Map Visualization



The Environment

<u>Arena Configuration:</u>

a) Size: 9×9 meter enclosed room with brick walls

b) Static Obstacles: 4 boxes

c) Base (green), CP1 (yellow), CP2 (yellow)

<u>Robots</u>

a) Avoider (Main Agent)

Differential drive robot

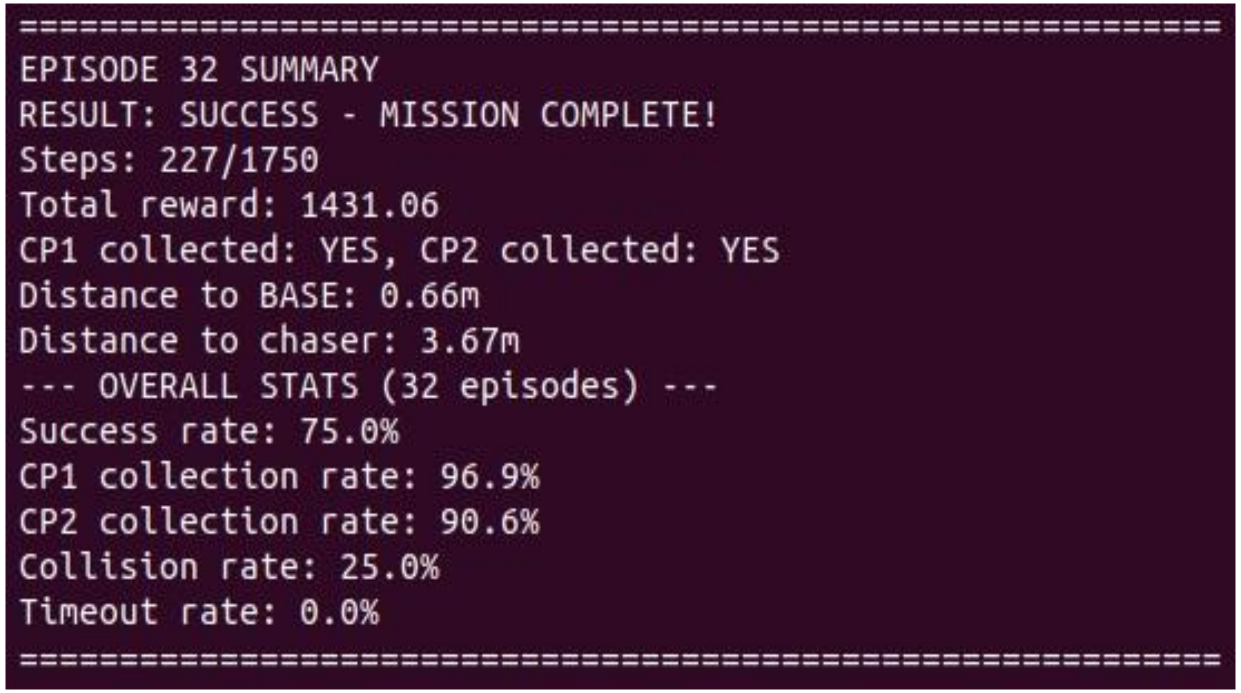
360° Lidar sensor (72 rays)

b) Chaser (Dynamic Obstacle)

Moves in circular pattern

Radius: 1.75m, Angular velocity: -0.12 rad/s

Mission Result



```
==============================================================
EPISODE 32 SUMMARY
RESULT: SUCCESS - MISSION COMPLETE!
Steps: 227/1750
Total reward: 1431.06
CP1 collected: YES, CP2 collected: YES
Distance to BASE: 0.66m
Distance to chaser: 3.67m
--- OVERALL STATS (32 episodes) ---
Success rate: 75.0%
CP1 collection rate: 96.9%
CP2 collection rate: 90.6%
Collision rate: 25.0%
Timeout rate: 0.0%
==============================================================
```

Each „episode" consists of 1750 time steps. In this period of time the mission has to end. One of three outputs occur:

a) mission success,

b) main robot collision,

c) timeout.

Besides that, the simulation has CP1 (checkpoint 1) and CP2 flags. Therefore, we are actively measuring progress made in each „phase" of the mission.

Key Features

1. Continuous Control

The robot smoothly controls speed and turning angle (not discrete moves like "turn left", "go forward")

2. Dynamic Avoidance

Learns to predict and avoid a moving chaser in real-time

3. Multi-Objective Mission

State machine handles sequential goals: CP1 → CP2 → Base

4. Custom Gym Environment

Handles sensor processing, reward calculation, and simulation resetting

Observation & Action Space

Input (Observation): 76 dimensions

[72 Lidar rays] + [Distance to target, Angle to target] + [Relative X to Chaser, Relative Y to Chaser] = 76-dimensional state vector

Output (Action): 2 continuous values

Linear Velocity	Angular Velocity
Range: -0.2 to 0.5 m/s	Range: -1.5 to 1.5 rad/s
Forward/Backward movement	Turning left/right

Reward System

- Main Rewards (Major Events):

  Checkpoint collected: 100.0

  Mission complete: 600.0

  Collision penalty: -100.0

- Additional Rewards (Continuous Feedback):

  Progress Reward: +/- (distance_change × 4.0)

  Encourages moving toward the current target

  Provides constant guidance during navigation

  Time Penalty: -0.05 per step to encourage efficiency

Soft Actor-Critic (SAC) Algorithm

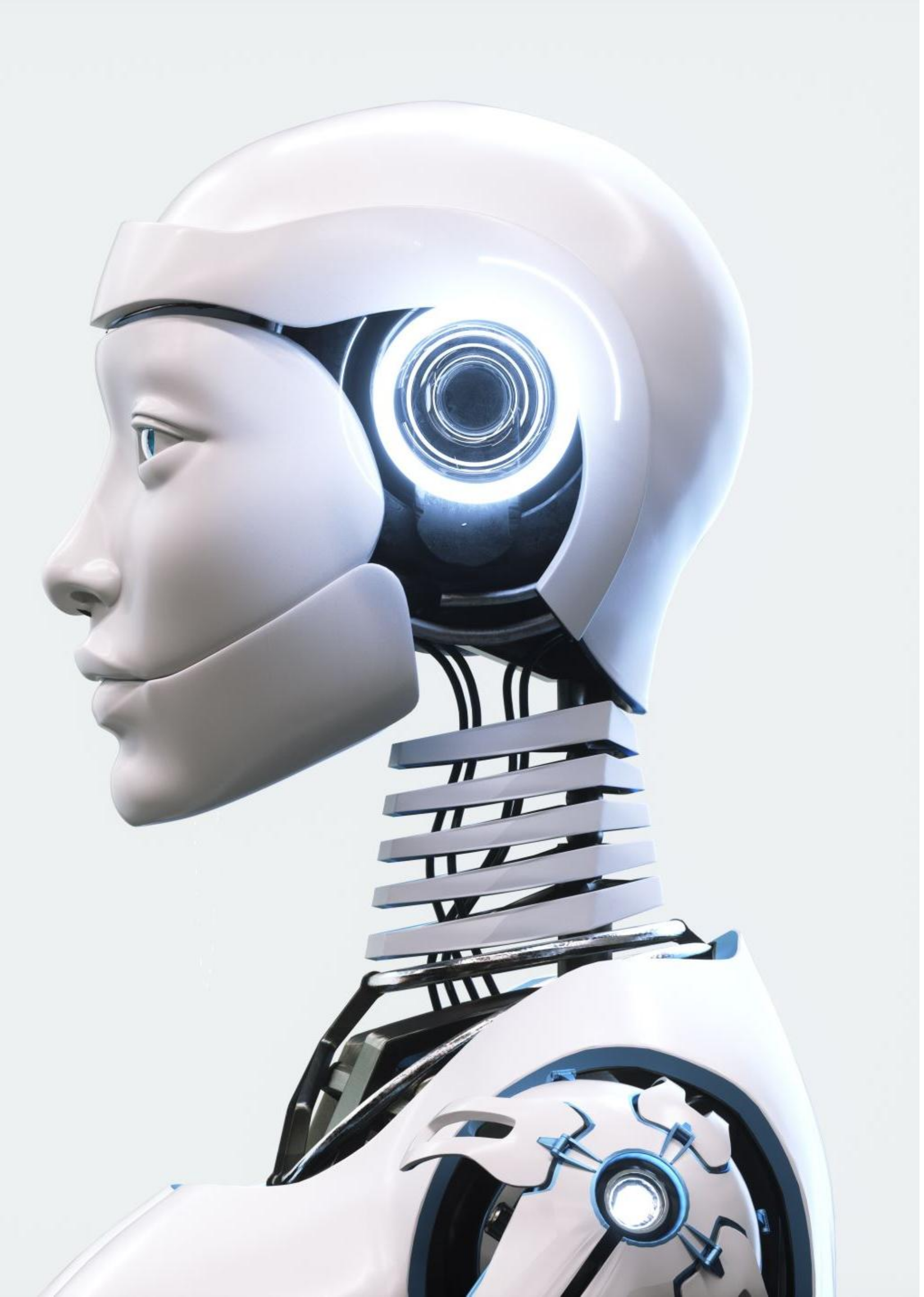
- Why SAC? Off-policy actor-critic algorithm that maximizes both reward AND entropy (exploration)

Actor & Agent

Critic

Entropy

Replay Buffer



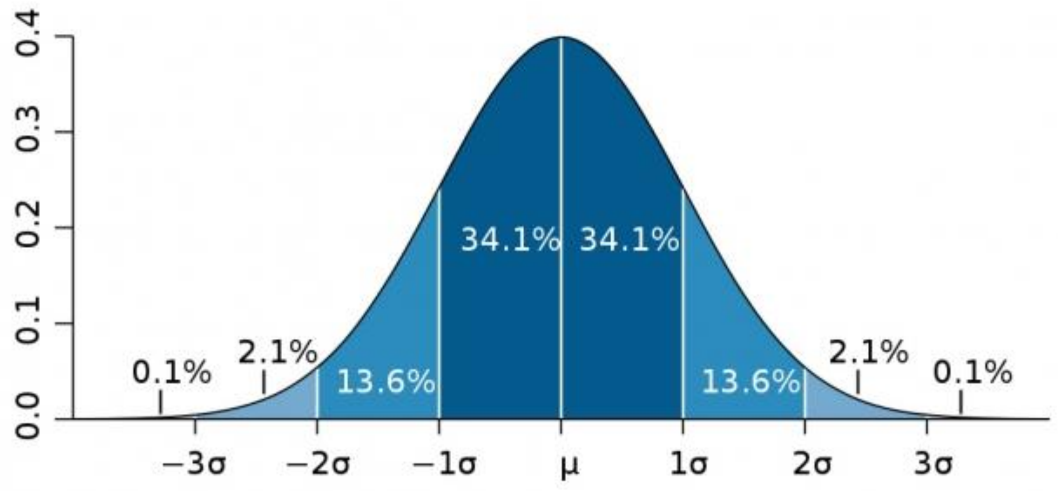
Actor & Agent

The Actor is the 'brain' that processes Lidar and target data. It defines a **probability distribution** (a range of potential moves) rather than a single fixed action. During training, the **Agent samples** a specific action from this distribution. This randomness allows the robot to explore the environment and find better strategies to avoid the Chaser.

How it works?

•The Actor uses a Gaussian (Normal) Distribution to propose moves:

- **Mean (μ):** The Actor suggests the "best" action based on its current knowledge.
- **Standard Deviation (σ):** This represents the Actor's "confidence" or uncertainty.
- **Sampling:** During training, the Agent samples from this "bell curve." Most actions will be near the mean, but some will be different, allowing for exploration.

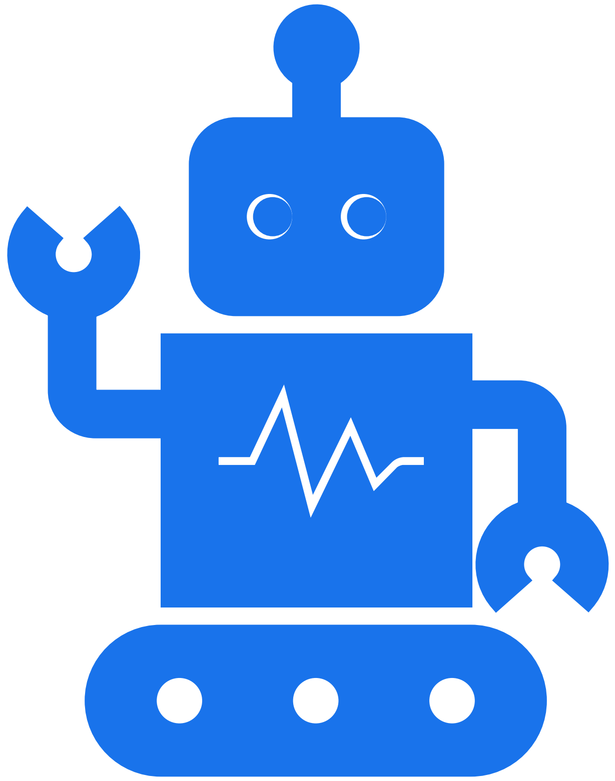


Critic

The Critic's job is to evaluate the quality of the actions proposed by the Actor in a given state. In SAC, we use two Critics to prevent the robot from being too "overconfident" about its actions. They predict how much total reward the robot will get in the future for a specific move. **The Actor** updates its strategy based on the Critics' feedback.

Entropy

Entropy measures how random or "uncertain" the robot's behavior is. SAC is unique because it rewards the robot for both getting points **and** for being curious (having high entropy). This prevents the robot from getting stuck in a boring routine and encourages it to find the best possible paths around the Chaser.



Replay Buffer

The Replay Buffer is a giant storage of the robot's past experiences. It stores what the robot saw, what it did, and what happened next. During training, the Agent picks random "memories" from this buffer to learn from. This way, the robot can learn from old mistakes and successes without having to repeat them in real-time.

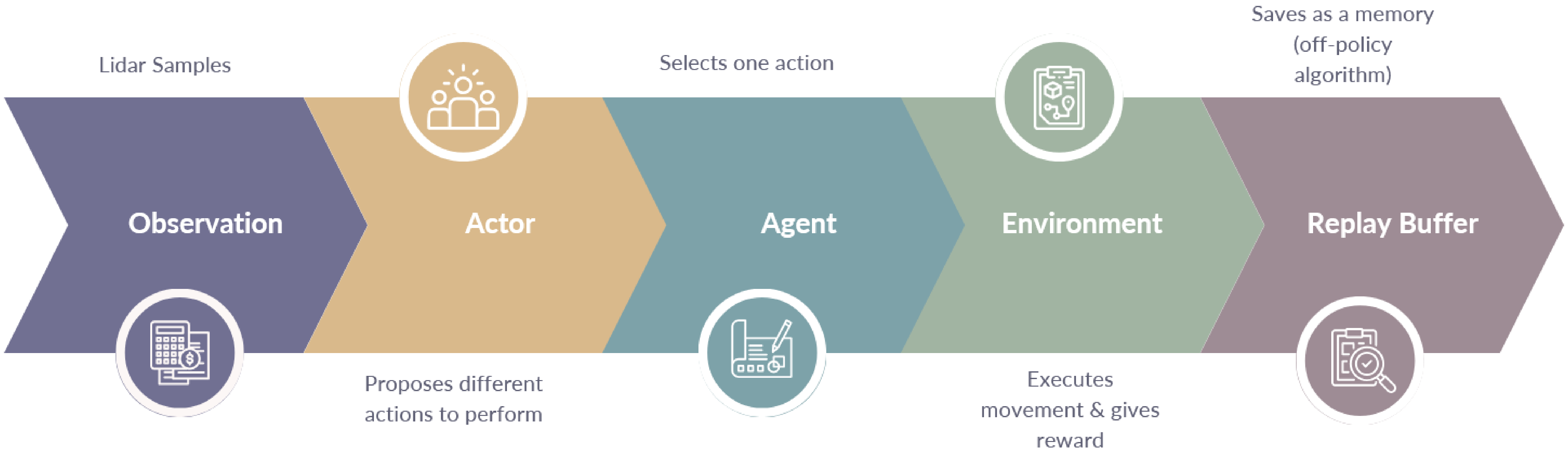

Lidar Samples
Observation
Actor
Proposes different actions to perform
Selects one action
Agent
Environment
Executes movement & gives reward
Saves as a memory (off-policy algorithm)
Replay Buffer

Illustration: This is how the robot moves in the simulation – *Action Cycle (Execution)*

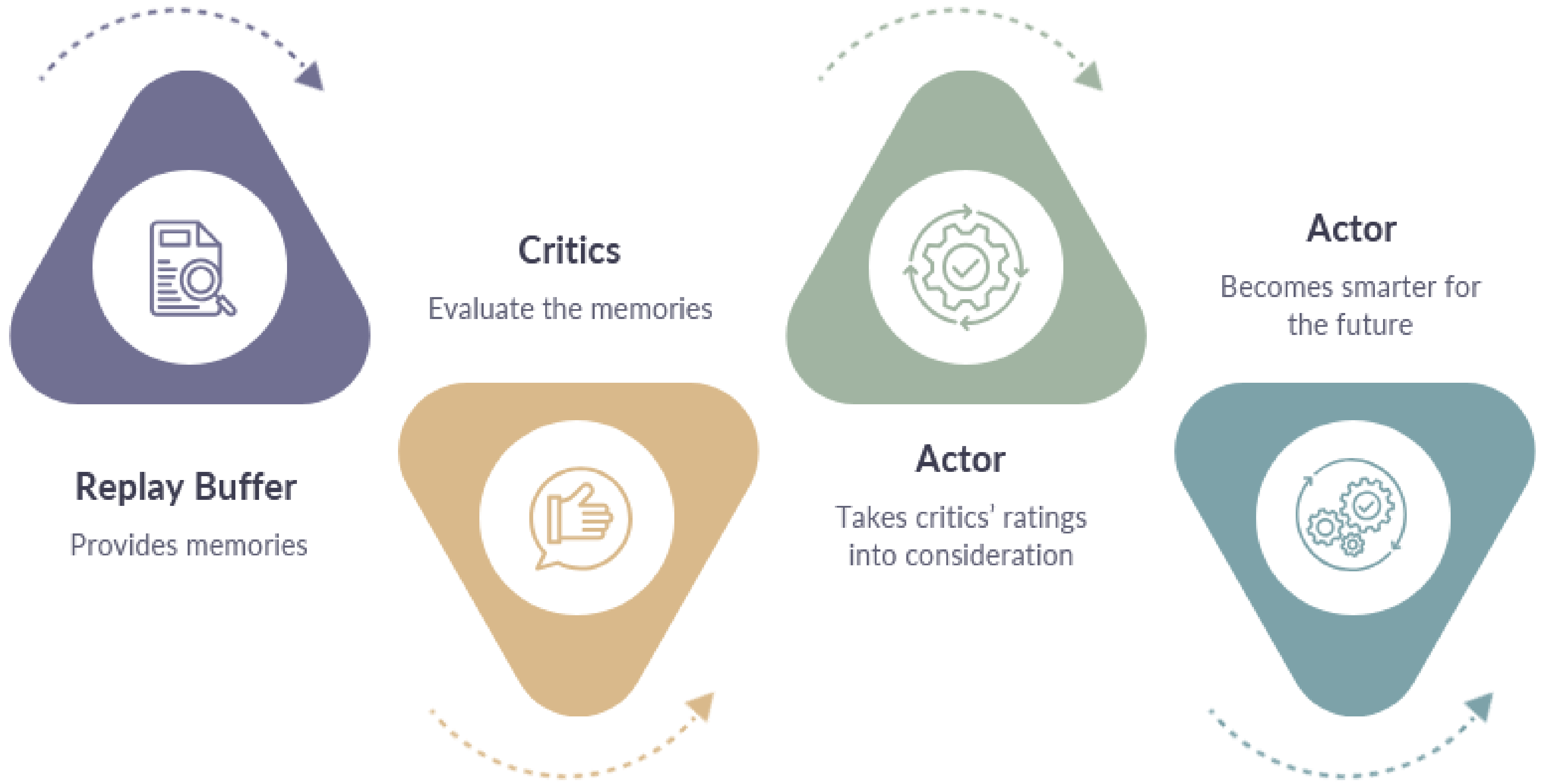
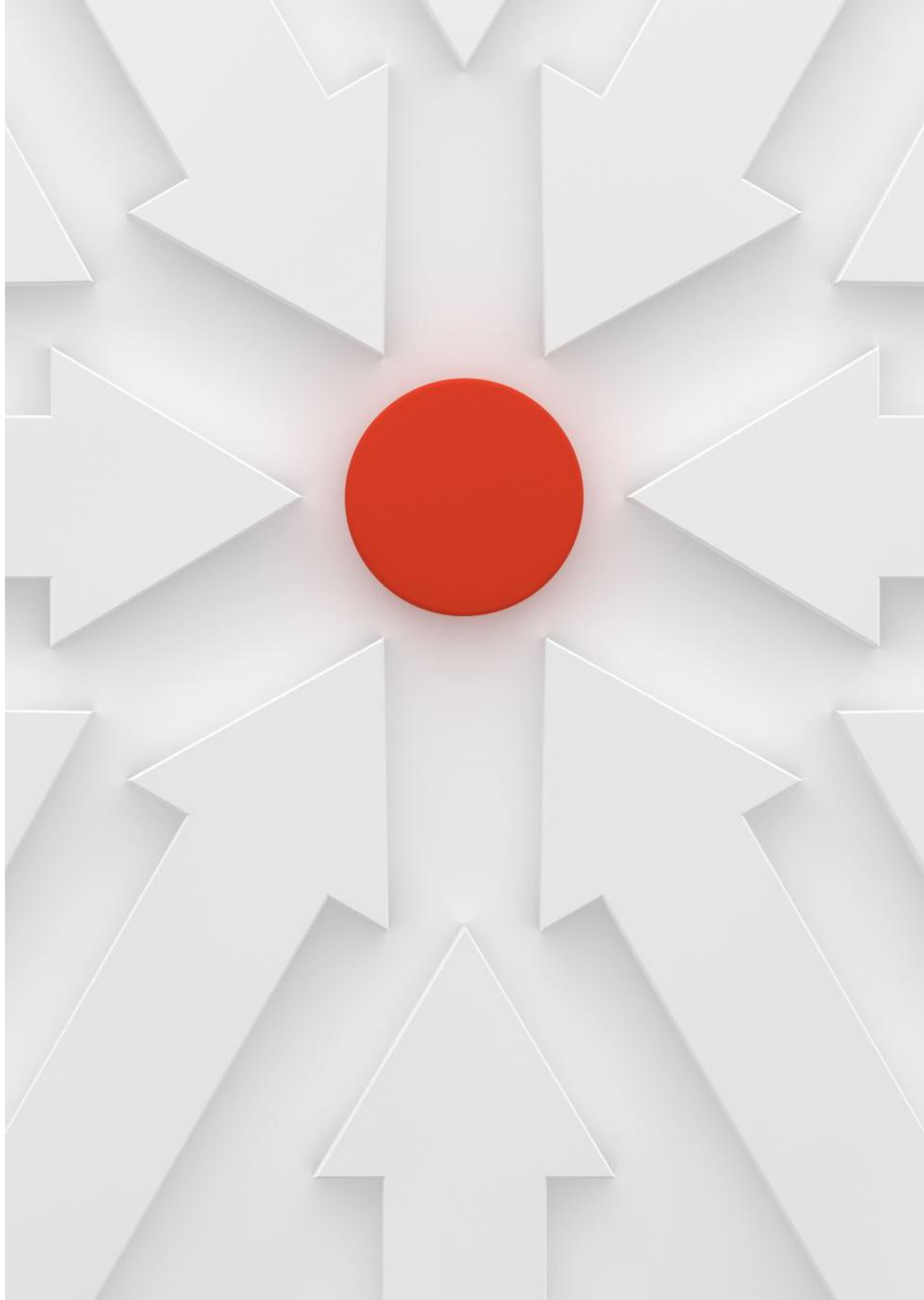

Replay Buffer
Provides memories
Critics
Evaluate the memories
Actor
Takes critics' ratings into consideration
Actor
Becomes smarter for the future

Illustration: This is how the robot moves in the simulation – *Learning Cycle (Optimization)*

How to summarize SAC?

SAC is an algorithm that balances getting the highest reward with being as curious as possible to find the safest path.



Training Results (after 2500 episodes)

LEARNING PROGRESSION:

Episodes 0–800: Early Exploration

Frequent collisions, learning basic movement and obstacle avoidance

Episodes 800–1000: Checkpoint Understanding

Robot begins to consistently reach and collect checkpoints

Episodes 1000–2200: Strategy Refinement

Improving path efficiency and chaser avoidance tactics

Episodes 2200+: Mastery

High success rate with optimized paths balancing speed and safety

Final Performance Metrics

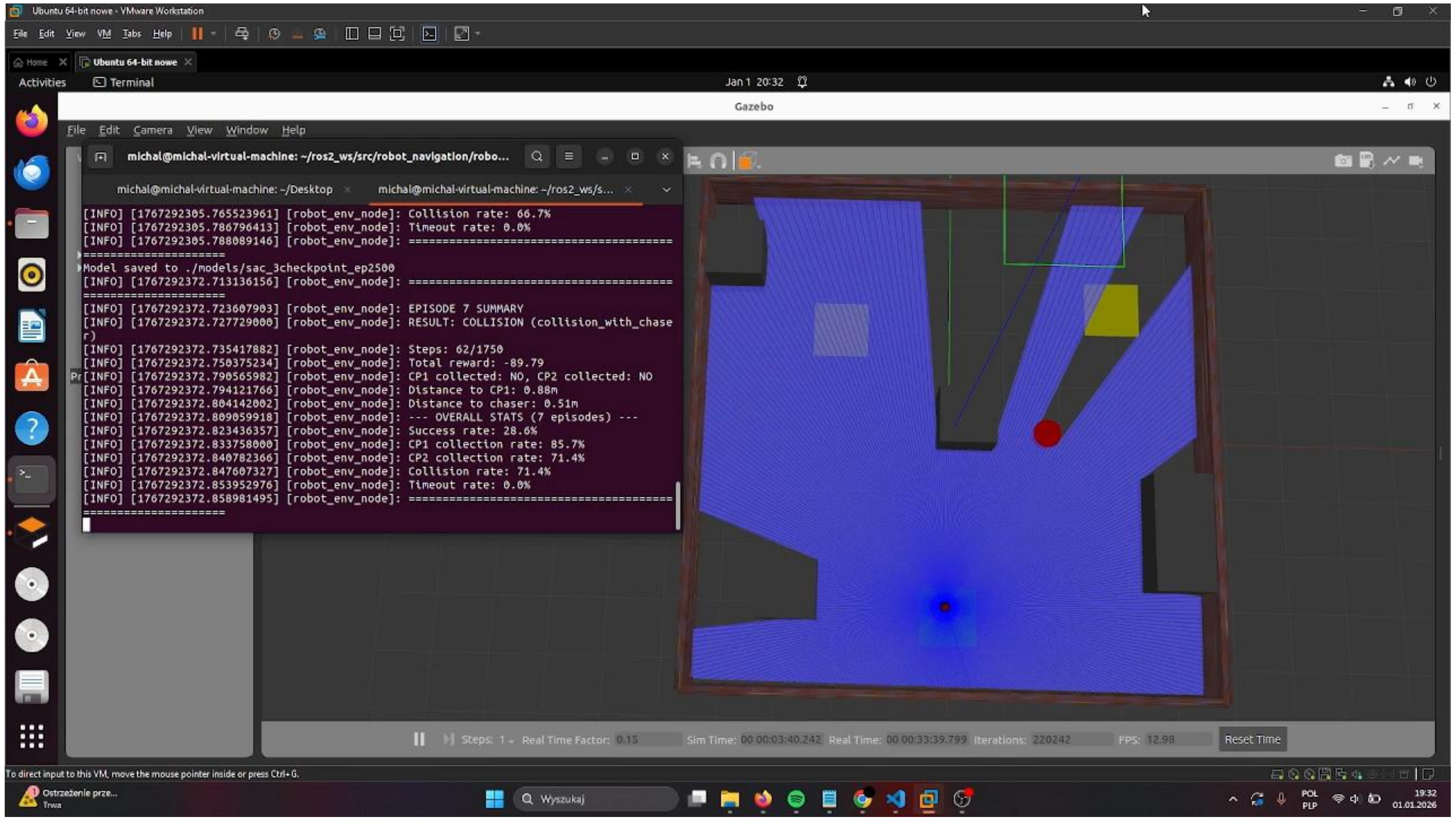
Mission Success Rate

60-68%

Successfully completes full mission loop

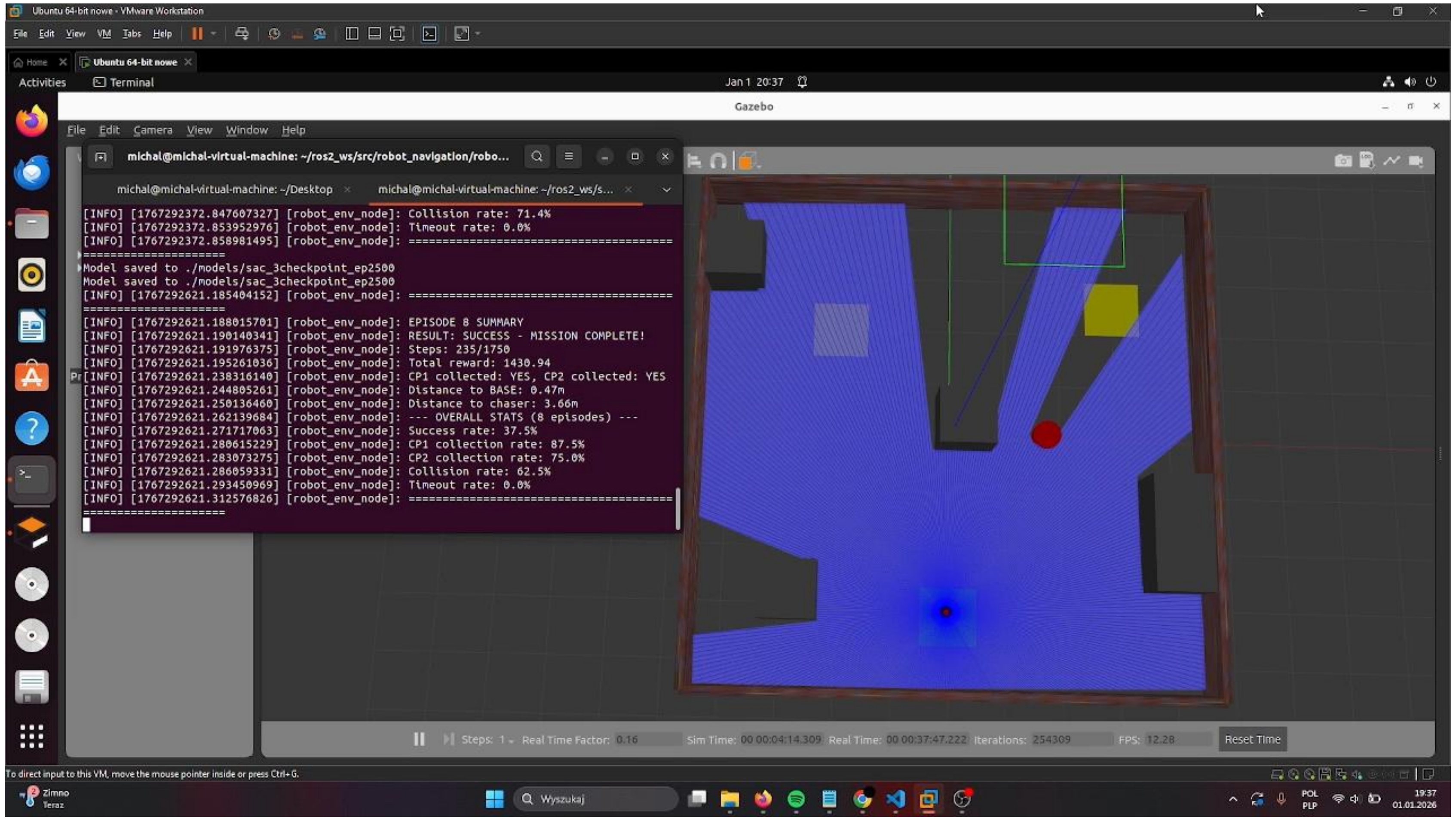
CP1 Collection Rate	CP2 Collection Rate	Collision Rate
~85%	**~70%**	**~25%**

Video of a Successful Mission Run


Ubuntu 64-bit nowe - VMware Workstation
Gazebo
michal@michal-virtual-machine: ~/ros2_ws/src/robot_navigation/robo...
michal@michal-virtual-machine: ~/Desktop
michal@michal-virtual-machine: ~/ros2_ws/s...
Steps: 1
Real Time Factor: 0.15
Sim Time: 00 00:03:40.242
Real Time: 00 00:33:39.799
Iterations: 220242
FPS: 12.98
Reset Time

```
[INFO] [1767292305.765523961] [robot_env_node]: Collision rate: 66.7%
[INFO] [1767292305.786796413] [robot_env_node]: Timeout rate: 0.0%
[INFO] [1767292305.788089146] [robot_env_node]: ========================================
=====================
Model saved to ./models/sac_3checkpoint_ep2500
[INFO] [1767292372.713136156] [robot_env_node]: ========================================
=====================
[INFO] [1767292372.723607903] [robot_env_node]: EPISODE 7 SUMMARY
[INFO] [1767292372.727729000] [robot_env_node]: RESULT: COLLISION (collision_with_chase
r)
[INFO] [1767292372.735417882] [robot_env_node]: Steps: 62/1750
[INFO] [1767292372.750375234] [robot_env_node]: Total reward: -89.79
[INFO] [1767292372.790565982] [robot_env_node]: CP1 collected: NO, CP2 collected: NO
[INFO] [1767292372.794121766] [robot_env_node]: Distance to CP1: 0.88m
[INFO] [1767292372.804142002] [robot_env_node]: Distance to chaser: 0.51m
[INFO] [1767292372.809059918] [robot_env_node]: --- OVERALL STATS (7 episodes) ---
[INFO] [1767292372.823436357] [robot_env_node]: Success rate: 28.6%
[INFO] [1767292372.833758000] [robot_env_node]: CP1 collection rate: 85.7%
[INFO] [1767292372.840782366] [robot_env_node]: CP2 collection rate: 71.4%
[INFO] [1767292372.847607327] [robot_env_node]: Collision rate: 71.4%
[INFO] [1767292372.853952976] [robot_env_node]: Timeout rate: 0.0%
[INFO] [1767292372.858981495] [robot_env_node]: ========================================
=====================
```

Video of Robot's Crash and Mission Restart


Ubuntu 64-bit nowe - VMware Workstation
Activities
Terminal
Jan 1 20:37
Gazebo
File Edit Camera View Window Help
michal@michal-virtual-machine: ~/ros2_ws/src/robot_navigation/robo...
michal@michal-virtual-machine: ~/Desktop
michal@michal-virtual-machine: ~/ros2_ws/s...
Steps: 1
Real Time Factor: 0.16
Sim Time: 00 00:04:14.309
Real Time: 00 00:37:47.222
Iterations: 254309
FPS: 12.28
Reset Time
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
Wyszukaj

```
[INFO] [1767292372.847607327] [robot_env_node]: Collision rate: 71.4%
[INFO] [1767292372.853952976] [robot_env_node]: Timeout rate: 0.0%
[INFO] [1767292372.858981495] [robot_env_node]: ========================================
=====================
Model saved to ./models/sac_3checkpoint_ep2500
Model saved to ./models/sac_3checkpoint_ep2500
[INFO] [1767292621.185404152] [robot_env_node]: ========================================
=====================
[INFO] [1767292621.188015701] [robot_env_node]: EPISODE 8 SUMMARY
[INFO] [1767292621.190140341] [robot_env_node]: RESULT: SUCCESS - MISSION COMPLETE!
[INFO] [1767292621.191976375] [robot_env_node]: Steps: 235/1750
[INFO] [1767292621.195261036] [robot_env_node]: Total reward: 1430.94
[INFO] [1767292621.238316140] [robot_env_node]: CP1 collected: YES, CP2 collected: YES
[INFO] [1767292621.244805261] [robot_env_node]: Distance to BASE: 0.47m
[INFO] [1767292621.250136460] [robot_env_node]: Distance to chaser: 3.66m
[INFO] [1767292621.262139684] [robot_env_node]: --- OVERALL STATS (8 episodes) ---
[INFO] [1767292621.271717063] [robot_env_node]: Success rate: 37.5%
[INFO] [1767292621.280615229] [robot_env_node]: CP1 collection rate: 87.5%
[INFO] [1767292621.283073275] [robot_env_node]: CP2 collection rate: 75.0%
[INFO] [1767292621.286059331] [robot_env_node]: Collision rate: 62.5%
[INFO] [1767292621.293450969] [robot_env_node]: Timeout rate: 0.0%
[INFO] [1767292621.312576826] [robot_env_node]: ========================================
=====================
```

Conclusions & Future Work

ACHIEVEMENTS:

+ Successfully trained an autonomous navigation agent using SAC

+ Robot learned complex multi-stage mission planning

+ Demonstrated dynamic obstacle avoidance capabilities

+ Achieved 60-75% mission success rate

POTENTIAL IMPROVEMENTS:

1. Training: More training time to improve the success rate to 80-90%.
2. Complex Chaser AI: Currently, the chaser moves in a circle. Future versions could implement a chaser that actively hunts the player.
3. Tougher Area: Adding second chaser robot.
4. Different Scenarios: Implementing different maps, with different placements of static boxes and different chasers' fixed paths.
5. Experimenting with SAC settings.