



UNIVERSITÀ DEGLI STUDI DI PADOVA

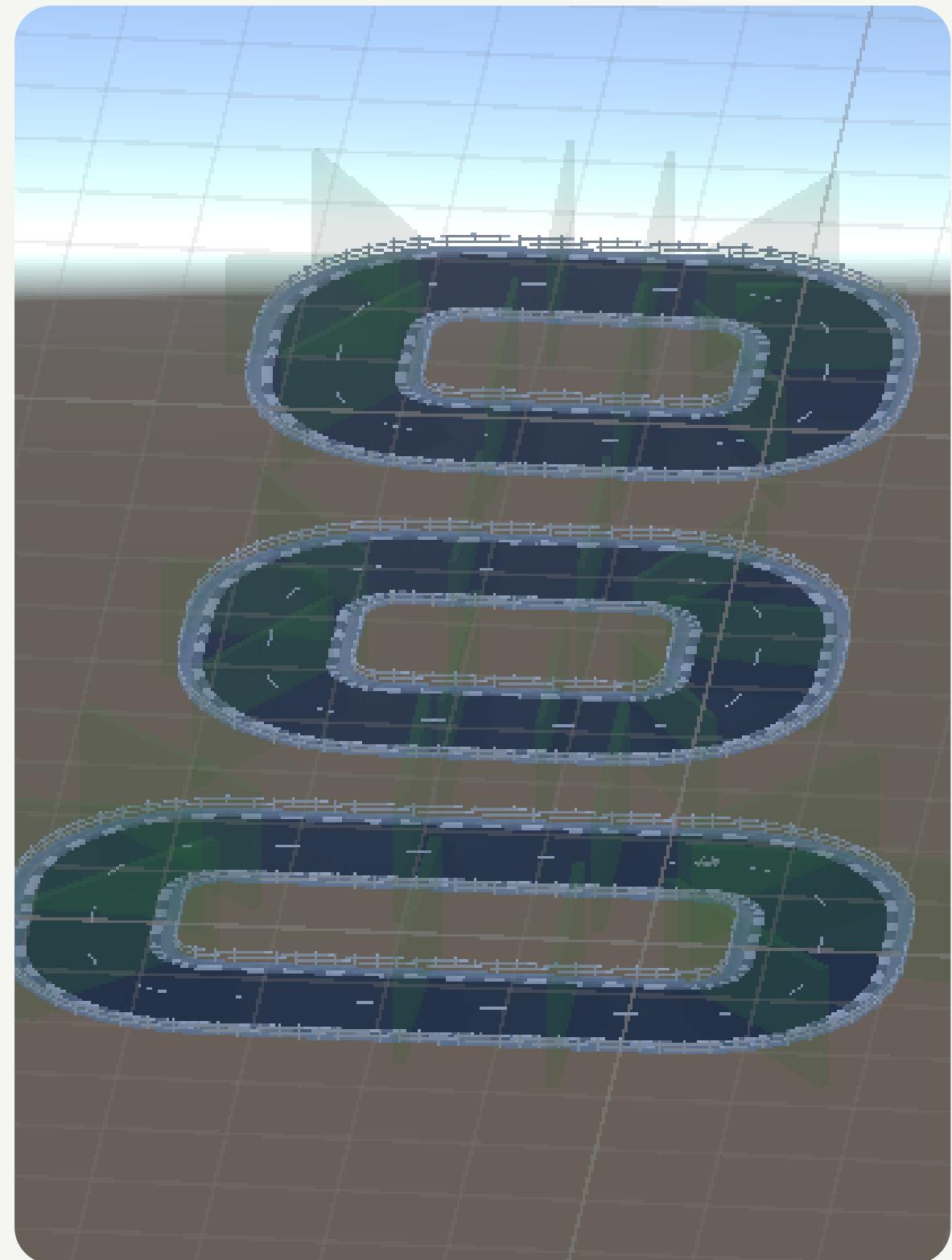
# Racing Game With ML Agents

Project presentation  
3D Vision and Extended Reality (2023-2024)

**Presented By:**

Alessandro Canel

Fateme Baghaei Saryazdi

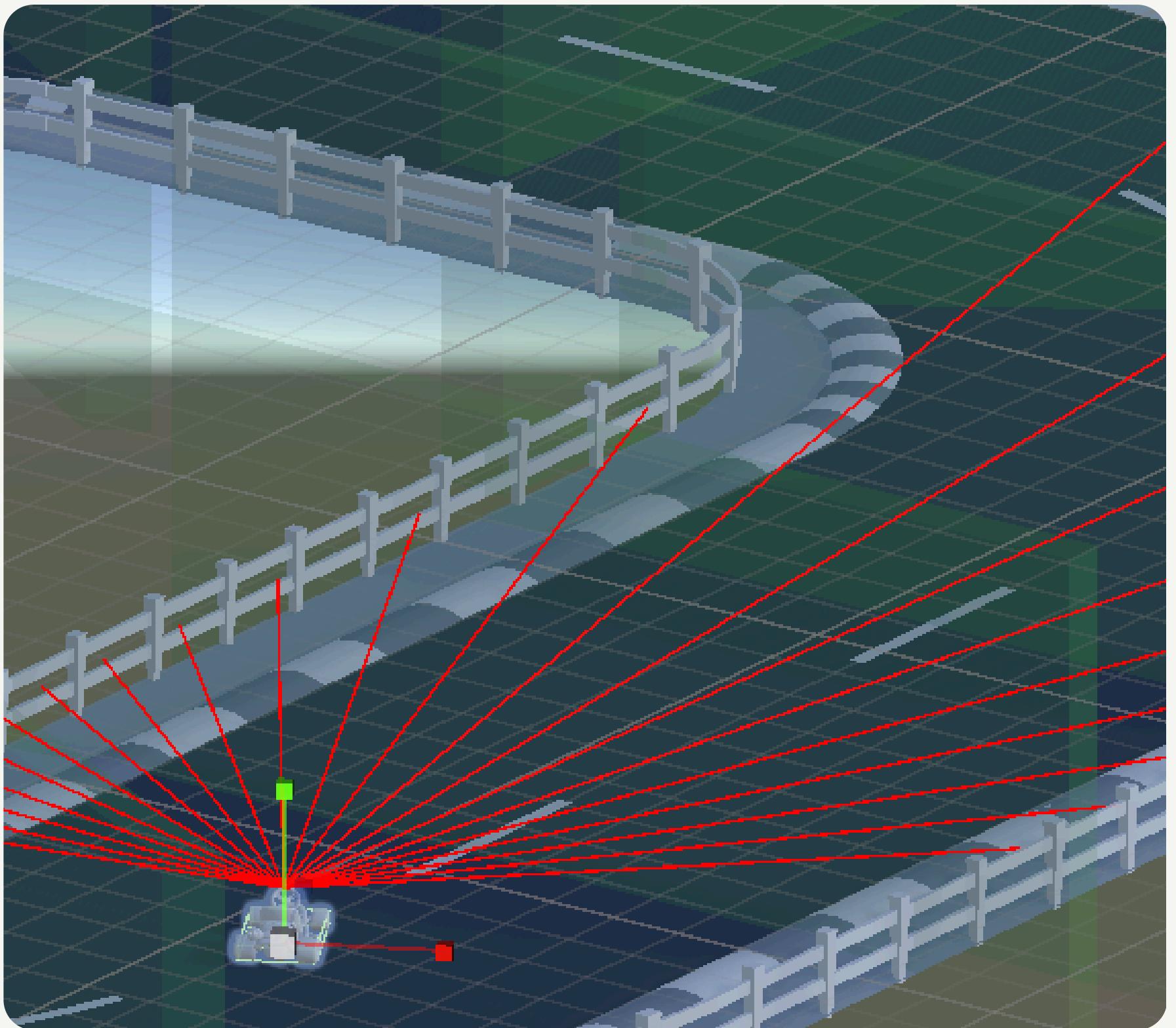


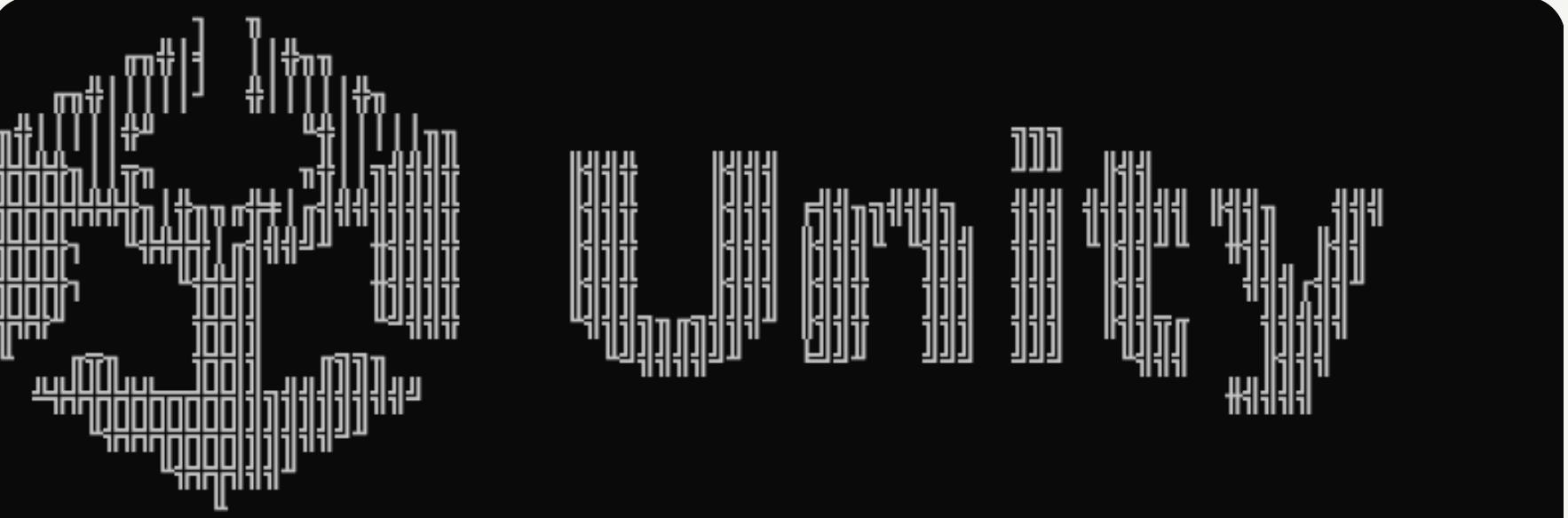
# Racing circuit

- Checkpoints
- Track
- Shape

# The Kart

- The handling
- 3D RayCasting
- The vision



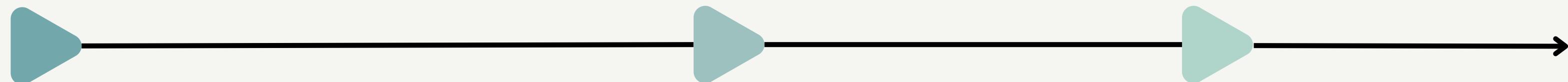


# The Training

- The specs
- YAML File
- Configuration

```
Version information:  
ml-agents: 0.30.0,  
ml-agents-envs: 0.30.0,  
Communicator API: 1.5.0,  
PyTorch: 2.3.1+cu118  
C:\Users\aless\venv\lib\site-packages\torch\__init__.py:749: UserWarning: torch.set_defau  
ecated as of PyTorch 2.1, please use torch.set_default_dtype() and torch.set_default_devi  
triggered internally at C:\actions-runner\_work\pytorch\pytorch\builder\windows\pytorch\to  
ensor.cpp:433.)  
    _C_.set_default_tensor_type(t)  
2024-07-14 19:43:32.611201: I tensorflow/core/util/port.cc:153] oneDNN custom operations  
ntly different numerical results due to floating-point round-off errors from different co  
n them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.  
2024-07-14 19:43:33.636142: I tensorflow/core/util/port.cc:153] oneDNN custom operations  
ntly different numerical results due to floating-point round-off errors from different co  
n them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.  
[INFO] Listening on port 5004. Start training by pressing the Play button in the Unity Ed  
[INFO] Connected to Unity environment with package version 2.0.1 and communication versio  
[INFO] Connected new brain: MoveToCheckpoint?team=0  
[WARNING] Deleting TensorBoard data events.out.tfevents.1720978761.AlexPC.81968.0 that wa  
us run.  
[INFO] Hyperparameters for behavior name MoveToCheckpoint:
```

# Unity Machine Learning Agents



## Definition:

- Open-source project for Unity 3D
- Enables games and simulations to serve as environments for training intelligent agents using machine learning algorithms

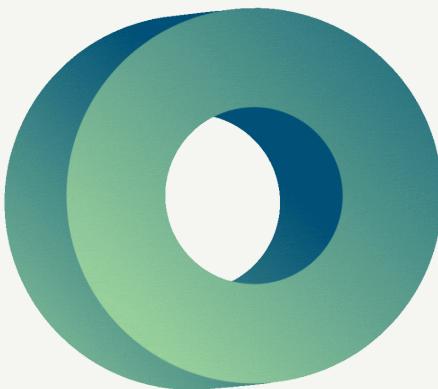
## Learning methods:

- Reinforcement Learning
- Imitation Learning

## Core components:

- Agents
- Environment
- Brains
- Rewards

# Reinforcement Learning



## Definition:

- solving control tasks by building agents that interact with the environment by performing actions, learn through trials and errors and receive rewards (positive or negative) as unique feedback.

## Goal:

- learn a policy that will maximize the sum of rewards it receives over the long run.
- This involves balancing short-term rewards with long-term benefits.



## Methods:

- Policy-based
- Value-based Methods

# Reward System

- Crucial component that guides the agents to learn desirable behaviors through reinforcement learning
- Provides feedback to the agents based on their actions
- Helping them understand what behaviors lead to success and which ones do not.



# Reward System

## Positive Rewards

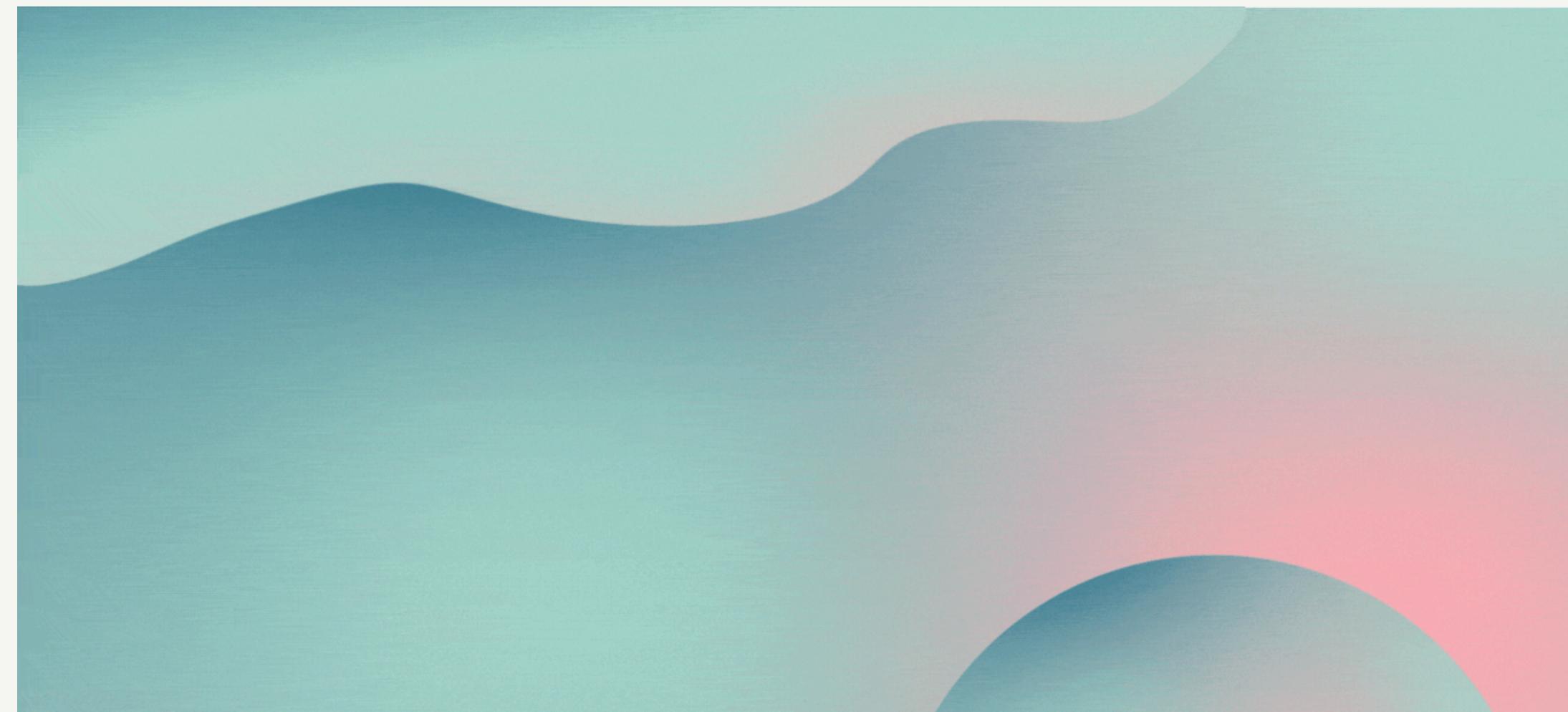
- Progress Along the Track
- Completing Laps
- Maintaining High Speed

## Negative Rewards

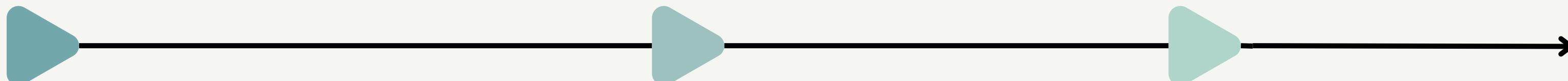
- Collisions
- Driving in Reverse
- Idle Time

## Shaping Rewards

- Smaller intermediate rewards
- ex: rewards for reaching checkpoints



# Imitation Learning:



## Definition:

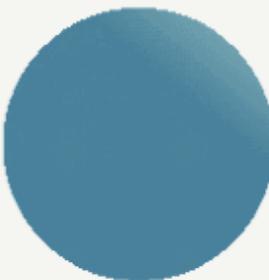
- Perform tasks by mimicking the behavior of expert demonstrations.
- The primary goal is for the agent to learn a policy that replicates the expert's actions in similar situations.

## Techniques:

- Generative Adversarial Imitation Learning (GAIL)
- Behavioral Cloning

## Demonstration Recorder:

- It should be enabled when you want to record your actions to be imitated





**Thank You!**