Unity ML-Agents: Revolutionizing Gaming through Reinforcement Learning

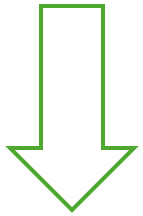
Authors: Umesh Raut, Prathmesh Galchhaniya, Anish Nehete, Rohan Shinde, Avaneesh Bhoite

Published in: 2024 2nd World Conference on Communication & Computing (WCONF)

Presenter: s1310115 OtoRyo

Introduction

Game‘s replay value, general appeal, and quality are determined by the actions, intelligence, and flexibility of NPC



pursuit of improving NPC’s skill

In addition to static pre-programmed creatures, creating adaptable NPC that can react to the actions of players and even learn from their experiences by experience with AI algorithms

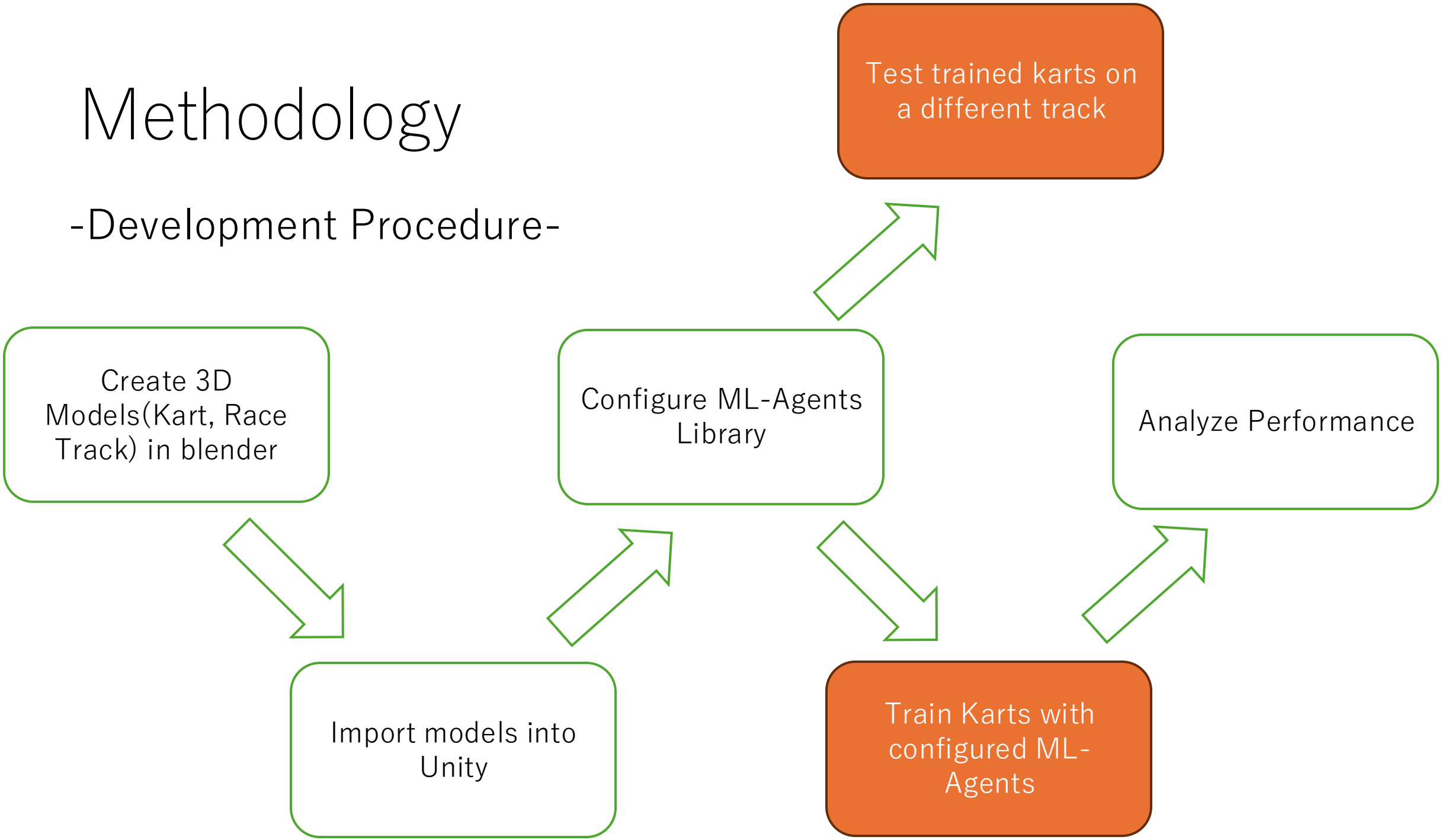
Target and Objective

Target: Particular use of AI in “Racing Game”

Objective: Developing an autonomous agent that can navigate a challenging racecourse, arrive at certain checkpoints, and use reinforcement learning to acquire the best driving practices

Methodology

-Development Procedure-



Methodology

-Game Engine-

Unity: provides a comprehensive framework for producing simulations and games, allowing for the integration of complicated physics, graphics, and AI components

-Game Architecture-

KartController: Responsible for handling the kart‘s movement, acceleration, steering, and collision detection.

Checkpoint and CheckpointManager: Define the checkpoints in the racing circuit

AutomaticCameraSystem: Manages the game cameras, switching viewpoints

KartAgent: This applies reinforcement learning using the Unity ML-Agents framework.

Methodology

-Reinforcement Learning Framework-

Initialise() Configures the agent at the start of every training session

OnEpisodeBegin() Starts the kart at the beginning location and resets the surroundings

CollectObservations() This function compiles pertinent environmental data, like the distance to the next checkpoint, and feeds it to the neural network so that it can make decisions

OnActionReceived() Controls the steering and acceleration of the kart by processing commands from the neural network

Heuristic() Enables the agent to be manually controlled for testing purposes

Methodology

-Sensors and Observations-

The agent sees the surroundings via vector observations and is able to make decisions more easily by feeding the neural network with the observations

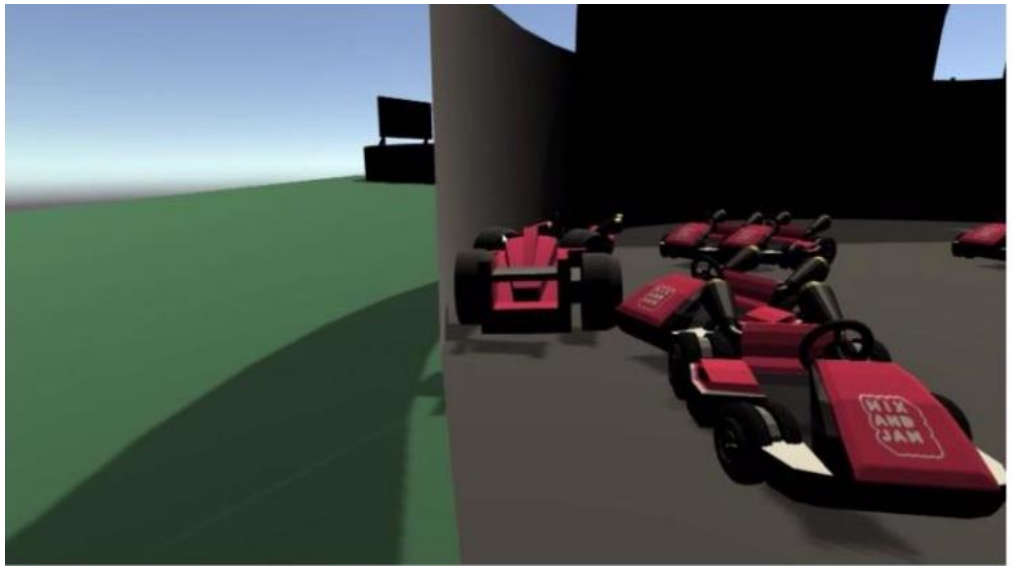
-Reward System-

Positive incentives are given to the agent for finishing the race and getting to checkpoints in the allotted time. When players deviate from the path or take longer than expected to reach a checkpoint, they receive negative rewards

-Training Process-

During the training phase, episodes are run iteratively so that the agent can pick up knowledge from its interactions with the surroundings

Implementation



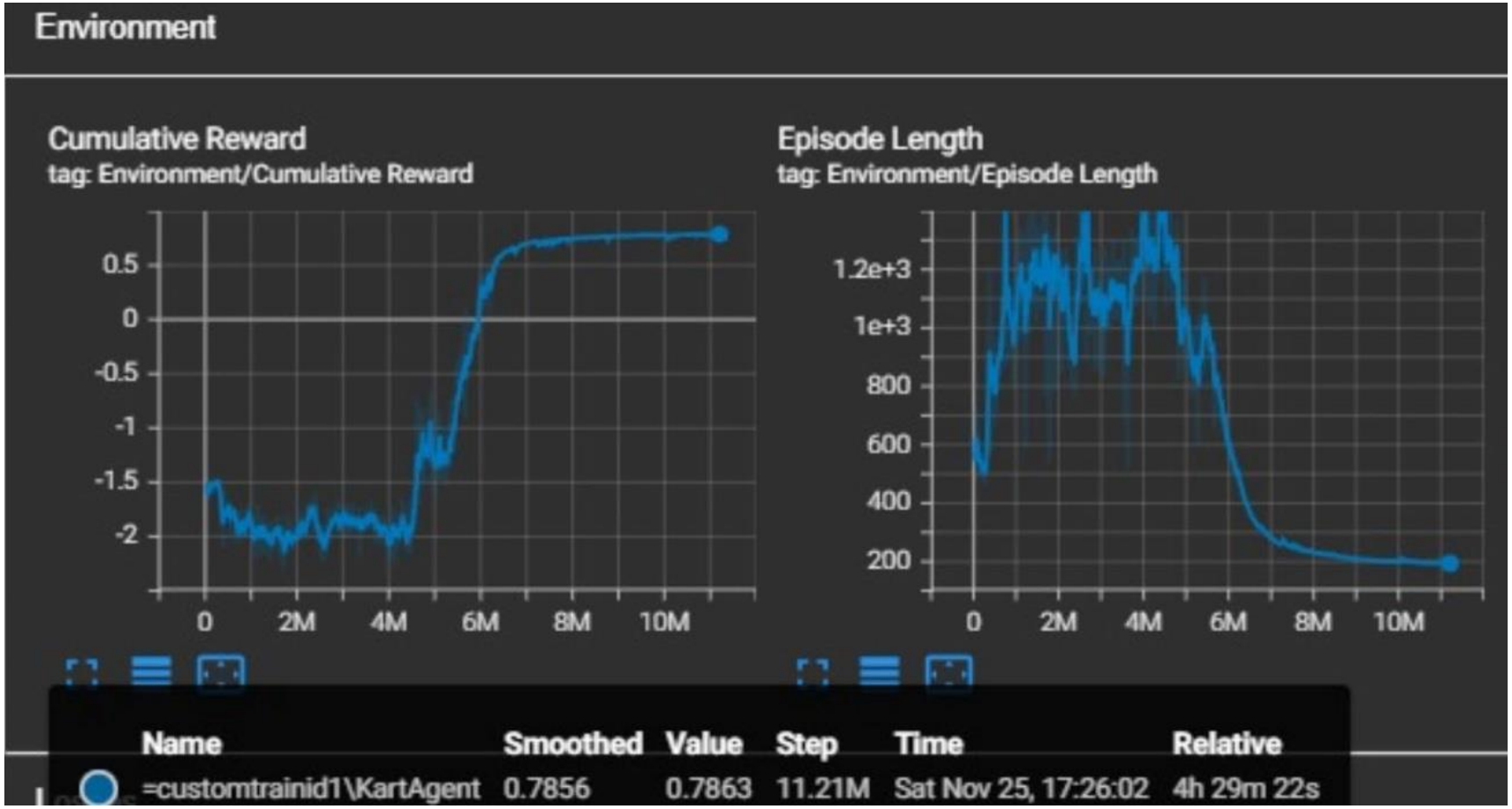
※Screenshot of game during training

Result

With just 4.5 hours of training, the model achieved stellar results, demonstrating both accuracy and incredible speed.

The model gained significant intelligence after approximately 4 million iterations and reached optimal driving performance after about 10 million iterations

Result


Environment
Cumulative Reward
tag: Environment/Cumulative Reward
0.5
0
-0.5
-1
-1.5
-2
0
2M
4M
6M
8M
10M
Episode Length
tag: Environment/Episode Length
1.2e+3
1e+3
800
600
400
200
0
2M
4M
6M
8M
10M
Name
Smoothed
Value
Step
Time
Relative
=customtrainid1\KartAgent
0.7856
0.7863
11.21M
Sat Nov 25, 17:26:02
4h 29m 22s

Conclusion

It has been demonstrated that by using reinforcement learning, dynamic and human-like NPCs can be created that adapt to player tactics, rather than relying on pre-programmed movements.

This autonomous driving configuration may also be applicable to VR and AR racing games.

References

Raut, Umesh, et al. "Unity ML-Agents: Revolutionizing Gaming Through Reinforcement Learning." 2024 2nd World Conference on Communication & Computing (WCONF). IEEE, 2024.

“Unity ML-Agents 実践ゲームプログラミング v2.2対応版”

Thank you for listening.