

# Hide-and-Seek Agent Training with Unity ML-Agents

*Step-by-step build, training, and export model & results*

Student: Zakria (Matricola: S6670951)

Course: Applied Reinforcement Learning

Professor: Riccardo Berta

Date: January 17, 2026

## 1. Objective

The goal of this project is to train a reinforcement learning agent in a Hide-and-Seek environment using Unity ML-Agents. Training is performed locally on a Windows machine, producing an exported ONNX policy that can be executed inside Unity (inference mode) without Python.

## 2. Requirements and Versions

Component	Version / Notes
Unity Editor	2022.3.62f3 (LTS)
Unity ML-Agents (C#)	2.3.0-exp.2 (package detected at runtime)
Python (Conda env)	3.8.18
mlagents	0.30.0
mlagents_envs	0.30.0
PyTorch	1.7.1 + CUDA 11.0 build
protobuf	3.20.3 (required for ML-Agents 0.30.0)

**Note:** A dedicated Conda environment was used to isolate dependencies and avoid version conflicts.

## 3. Project and Scenes

The Hide-and-Seek environment is provided as a standalone Unity project (unity-ml-agents\_hide-and-seek). Two scenes are important for this workflow:

- Training scene: Assets/Scenes/Training.unity
- Test (inference) scene: Assets/Scenes/Test.unity

## 4. Python Environment Setup

A Conda environment was created and activated before installing ML-Agents. The following steps were executed from the terminal.

```
conda create -n ml_agents python=3.8.18 pip -y
conda activate ml_agents
python --version
```

Package installation:

```
python -m pip install --upgrade pip setuptools wheel
pip install torch~=1.7.1 -f https://download.pytorch.org/whl/torch_stable.html
python -m pip install mlagents==0.30.0
```

## 4.1 Dependency Fixes

During setup, ML-Agents failed due to an incompatible protobuf version and a missing package required by TensorBoard logging. The following fixes were applied:

```
python -m pip uninstall -y protobuf
python -m pip install protobuf==3.20.3
python -m pip install six
mlagents-learn --help
```

## 5. Building the Training Executable (Windows)

Training was performed with a built Unity executable instead of training in the Editor. Executable training is faster, supports headless mode (--no-graphics), and avoids Editor overhead.

Unity build steps:

1. Open the Hide-and-Seek Unity project.
2. File -> Build Settings...
3. Add Assets/Scenes/Training.unity to "Scenes In Build" and ensure it is checked.
4. Select "PC, Mac & Linux Standalone" -> Target: Windows, Architecture: x86\_64.
5. Click Build and select an output folder: Builds/Training/

Expected build output:

```
Builds\Training\unity-ml-agents_hide-and-seek.exe
Builds\Training\unity-ml-agents_hide-and-seek_Data\
```

### 5.1 Build Error and Resolution

The first build attempt failed due to a corrupted Unity package cache (Visual Scripting DLL missing in PackageCache). After rebuilding and allowing Unity to refresh packages, the build completed successfully and only warnings remained.

## 6. Discovering Behavior Names and Specs

The trainer configuration must use the correct behavior key(s). In this environment, behavior names were extracted directly from the built executable using the ML-Agents Python API.

```
python -c "from mlagents_envs.environment import UnityEnvironment;
env=UnityEnvironment(file_name=r'Builds\Training\unity-ml-agents_hide-and-seek.exe',
no_graphics=True); env.reset(); print('Behaviors:', list(env.behavior_specs.keys()));
[print(f'\n{name}\n action_spec: {spec.action_spec}\n obs_shapes: {[o.shape for o in
spec.observation_specs]}\n') for name,spec in env.behavior_specs.items()]; env.close()"
```

Observed behaviors (two teams):

- HideAndSeekAgent?team=0
- HideAndSeekAgent?team=1

## 7. Creating the Trainer YAML (ml-agents\_config.yaml)

The repository did not include a trainer YAML, so a new configuration file was created in the project root. A common mistake is to use team-suffixed behavior names as YAML keys. ML-Agents expects the base behavior name (HideAndSeekAgent).

File created:

```
unity-ml-agents_hide-and-seek\ml-agents_config.yaml
```

Final YAML content used:

```
behaviors:
  HideAndSeekAgent:
    trainer_type: ppo
    hyperparameters:
      batch_size: 2048
      buffer_size: 20480
      learning_rate: 3.0e-4
      beta: 1.0e-3
      epsilon: 0.2
      lambda: 0.95
      num_epoch: 3
      learning_rate_schedule: linear
    network_settings:
      normalize: true
      hidden_units: 256
      num_layers: 2
    reward_signals:
      extrinsic:
        gamma: 0.99
        strength: 1.0
    max_steps: 2000000
    time_horizon: 256
    summary_freq: 20000
    checkpoint_interval: 100000
```

## 8. Training Command

Training was launched from the Conda environment using mlagents-learn. Headless mode was enabled for speed (--no-graphics) and the Unity time scale was increased.

```
mlagents-learn "ml-agents_config.yaml" --env "Builds\Training\unity-ml-agents_hide-and-seek.exe" -  
-run-id hns_run1 --force --no-graphics --time-scale 20
```

Successful startup indicators:

- Connected new brain: HideAndSeekAgent?team=0
- Connected new brain: HideAndSeekAgent?team=1
- HideAndSeekAgent. Step: ... Training.

## 8.1 Notes on Training Warnings

During training, ML-Agents reported that multiple teams were present and suggested enabling self-play for adversarial games. This run (hns\_run1) was treated as a baseline training run.

## 9. Exported Models and Results Directory

ML-Agents periodically exports ONNX policies during training at the checkpoint interval. Export messages appeared in the console (e.g., Exported ...HideAndSeekAgent-999900.onnx).

Results directory:

```
results\hns_run1\HideAndSeekAgent
```

Example exported files:

- HideAndSeekAgent-99900.onnx
- HideAndSeekAgent-199900.onnx
- HideAndSeekAgent-999900.onnx
- HideAndSeekAgent-1599900.onnx

**Note:** The latest trained policy is the ONNX file with the highest step number in its filename.

## 10. Running the Trained Model in Unity (Inference)

After training, the exported ONNX model can be assigned to the agents in the Test scene to run without Python.

- Stop training from the terminal with Ctrl + C.
- Open Assets/Scenes/Test.unity.
- Select each agent GameObject and open Behavior Parameters.
- Set Behavior Type to Inference Only.
- Assign the chosen .onnx model file to the Model field.
- Press Play to observe behavior in Unity.

## 11. Reproducibility Checklist

A run is reproducible if the following items are present:

- Build executable: Builds/Training/unity-ml-agents\_hide-and-seek.exe
- Trainer config: ml-agents\_config.yaml
- Training command (mlagents-learn) used for hns\_run1
- Exported ONNX policy file(s) under results/hns\_run1/HideAndSeekAgent/

## Appendix: Useful Commands

Verify GPU availability (optional):

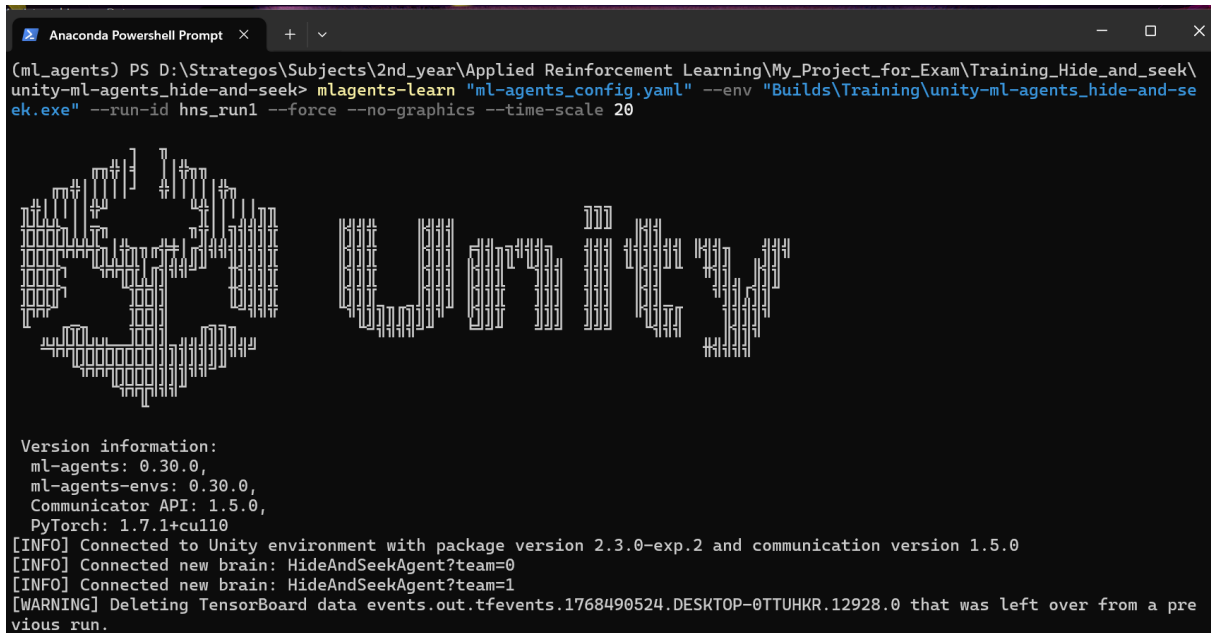
```
python -c "import torch; print(torch.cuda.is_available());  
print(torch.cuda.get_device_name(0) if torch.cuda.is_available() else 'N/A')"
```

Force training on GPU (optional):

```
mlagents-learn "ml-agents_config.yaml" --env "Builds\Training\unity-ml-agents_hide-and-seek.exe" -  
-run-id hns_gpu1 --force --no-graphics --time-scale 20 --torch-device cuda
```

## Appendix A – TensorBoard Training Logs (Screenshots)

This appendix contains selected screenshots from the TensorBoard dashboard and command-line output to document the training process and key metrics observed during the Hide-and-Seek experiment.



```
(ml_agents) PS D:\Strategos\Subjects\2nd_year\Applied Reinforcement Learning\My_Project_for_Exam\Training_Hide_and_seek\
unity-ml-agents_hide-and-seek> mlagents-learn "ml-agents_config.yaml" --env "Builds\Training\unity-ml-agents_hide-and-se
ek.exe" --run-id hns_run1 --force --no-graphics --time-scale 20

          H
          I
          D
          E
          A
          N
          D
          S
          E
          E
          K

Version information:
ml-agents: 0.30.0,
ml-agents-envs: 0.30.0,
Communicator API: 1.5.0,
PyTorch: 1.7.1+cu110
[INFO] Connected to Unity environment with package version 2.3.0-exp.2 and communication version 1.5.0
[INFO] Connected new brain: HideAndSeekAgent?team=0
[INFO] Connected new brain: HideAndSeekAgent?team=1
[WARNING] Deleting TensorBoard data events.out.tfevents.1768490524.DESKTOP-0TTUHKR.12928.0 that was left over from a pre
vious run.
```

Figure A1. Command used to start the PPO training run (*mlagents-learn*) and environment connection logs.

```
Anaconda Powershell Prompt x + v
[INFO] Hyperparameters for behavior name HideAndSeekAgent:
trainer_type: ppo
hyperparameters:
  batch_size: 2048
  buffer_size: 20480
  learning_rate: 0.0003
  beta: 0.001
  epsilon: 0.2
  lambda: 0.95
  num_epoch: 3
  shared_critic: False
  learning_rate_schedule: linear
  beta_schedule: linear
  epsilon_schedule: linear
network_settings:
  normalize: True
  hidden_units: 256
  num_layers: 2
  vis_encode_type: simple
  memory: None
  goal_conditioning_type: hyper
  deterministic: False
reward_signals:
  extrinsic:
    gamma: 0.99
    strength: 1.0
    network_settings:
      normalize: False
      hidden_units: 128
      num_layers: 2
      vis_encode_type: simple
      memory: None
      goal_conditioning_type: hyper
      deterministic: False
  init_path: None
keep_checkpoints: 5
checkpoint_interval: 100000
max_steps: 2000000
time_horizon: 256
summary_freq: 20000
threaded: False
self_play: None
behavioral_cloning: None
```

Figure A2. Printed hyperparameters and network settings used for the HideAndSeekAgent behavior.



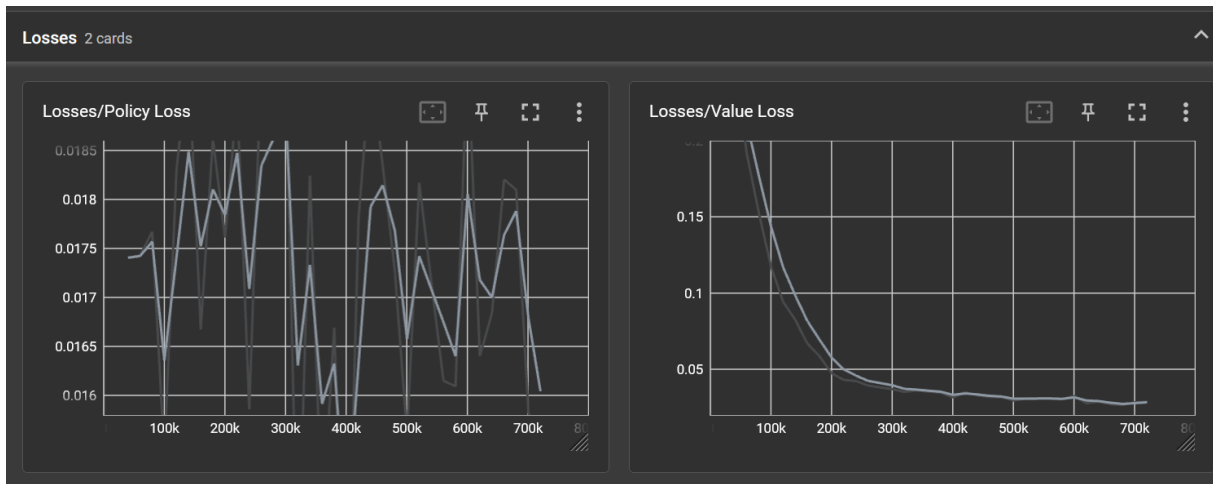


Figure A3. TensorBoard: Policy loss and Value loss during training.

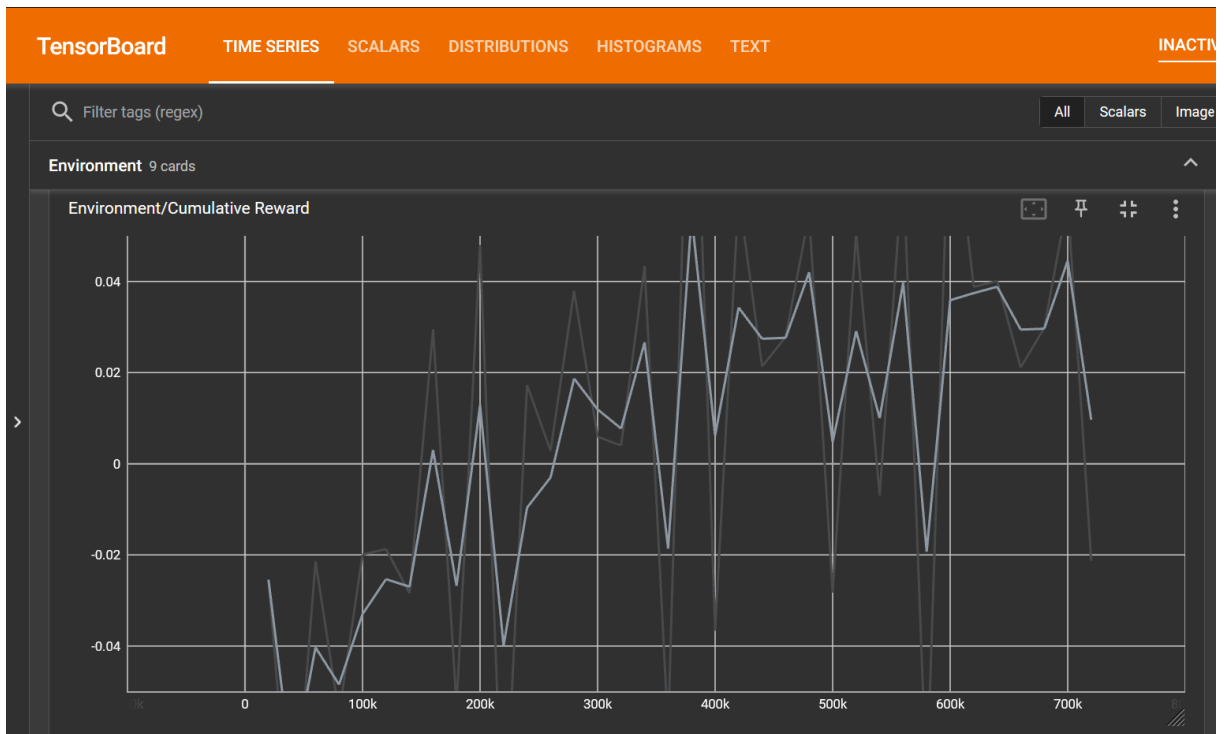


Figure A4. TensorBoard: Cumulative reward trend over training steps.

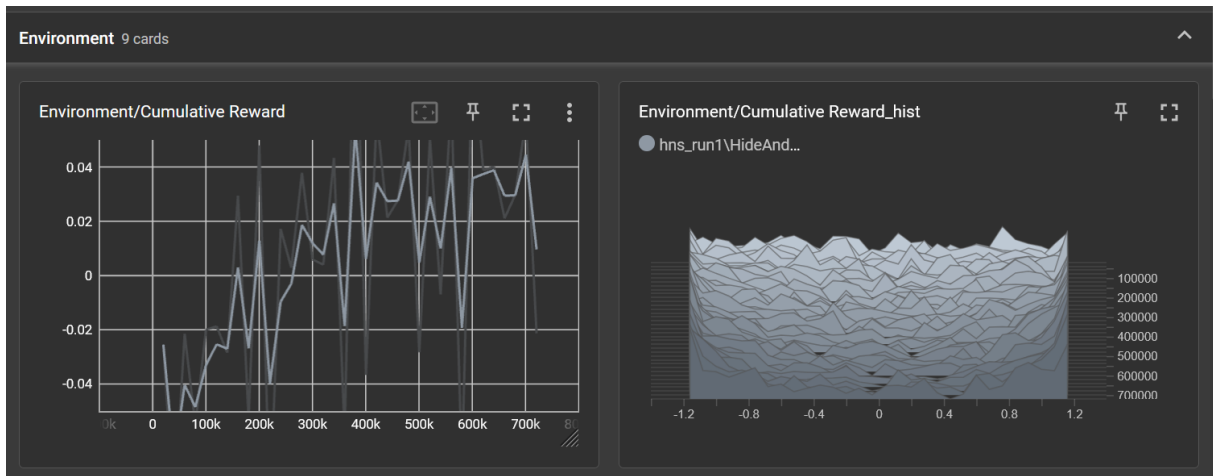


Figure A5. TensorBoard: Cumulative reward and its histogram (distribution across updates).

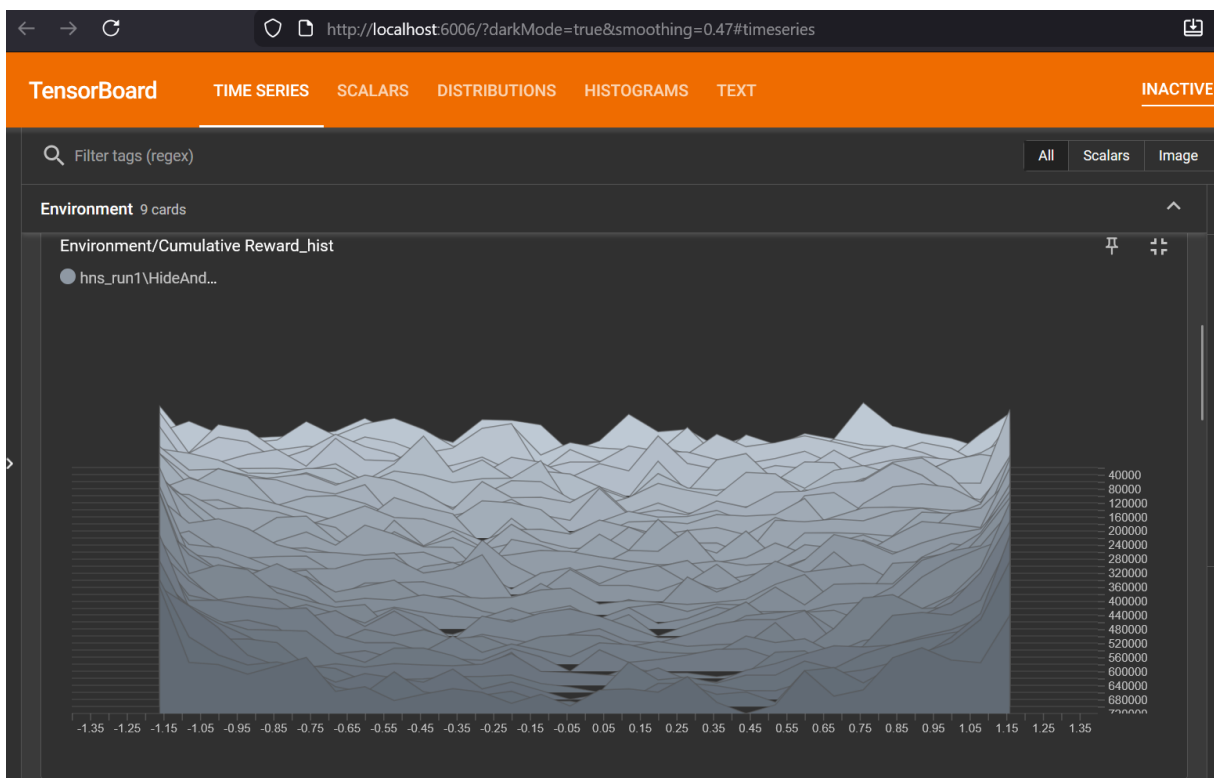


Figure A6. TensorBoard: Cumulative reward histogram view (distribution visualization).

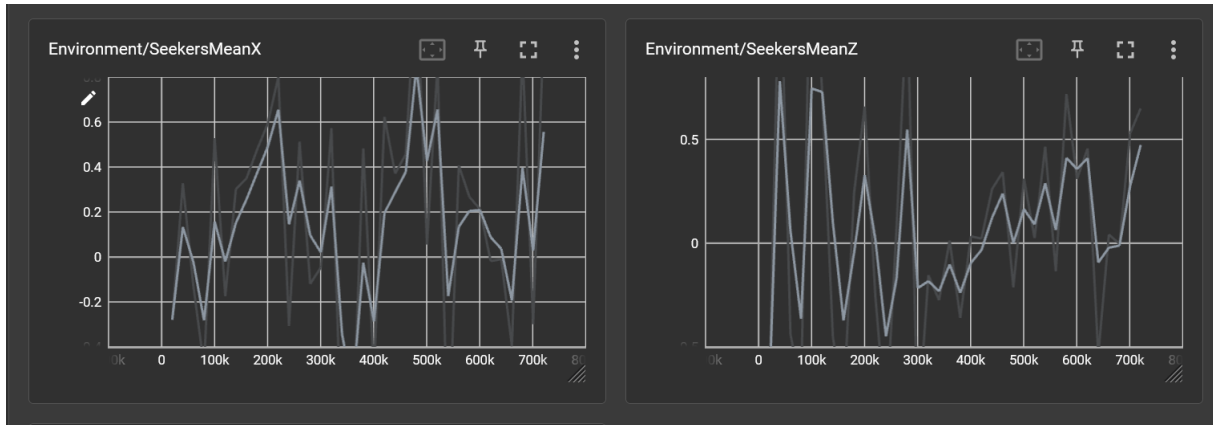


Figure A7. TensorBoard: Seeker agents mean position metrics (MeanX and MeanZ).

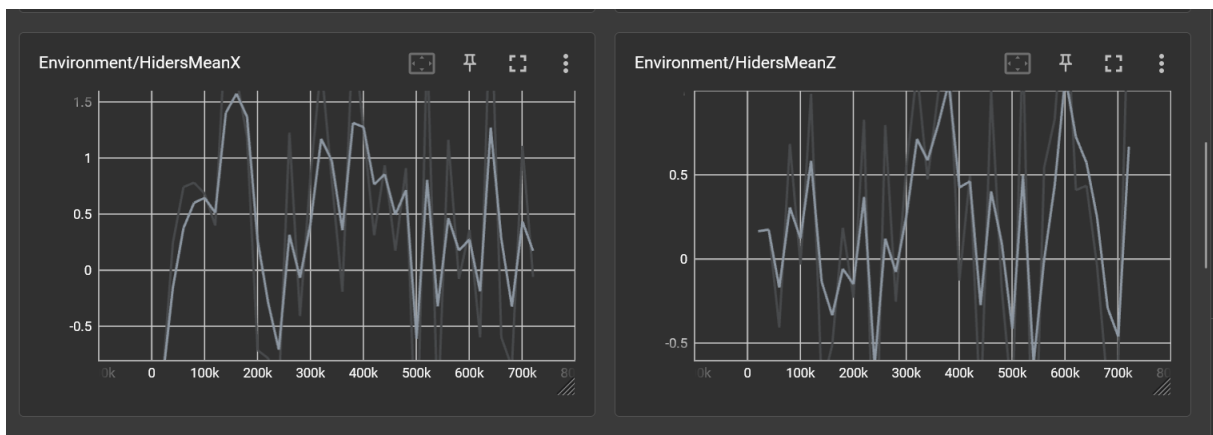


Figure A8. TensorBoard: Hider agents mean position metrics (MeanX and MeanZ).

```
Anaconda Powershell Prompt x + v
[INFO] Exported results\hns_run1\HideAndSeekAgent\HideAndSeekAgent-399900.onnx
[INFO] HideAndSeekAgent. Step: 420000. Time Elapsed: 1357.412 s. Mean Reward: 0.059. Std of Reward: 0.881. Training.
[INFO] HideAndSeekAgent. Step: 440000. Time Elapsed: 1435.106 s. Mean Reward: 0.021. Std of Reward: 0.881. Training.
[INFO] HideAndSeekAgent. Step: 460000. Time Elapsed: 1502.626 s. Mean Reward: 0.028. Std of Reward: 0.887. Training.
[INFO] HideAndSeekAgent. Step: 480000. Time Elapsed: 1572.866 s. Mean Reward: 0.055. Std of Reward: 0.908. Training.
[INFO] HideAndSeekAgent. Step: 500000. Time Elapsed: 1642.711 s. Mean Reward: -0.028. Std of Reward: 0.920. Training.
[WARNING] Trainer has multiple policies, but default behavior only saves the first.
[INFO] Exported results\hns_run1\HideAndSeekAgent\HideAndSeekAgent-499900.onnx
[INFO] HideAndSeekAgent. Step: 520000. Time Elapsed: 1719.975 s. Mean Reward: 0.051. Std of Reward: 0.870. Training.
[INFO] HideAndSeekAgent. Step: 540000. Time Elapsed: 1769.373 s. Mean Reward: -0.007. Std of Reward: 0.876. Training.
[INFO] HideAndSeekAgent. Step: 560000. Time Elapsed: 1838.160 s. Mean Reward: 0.066. Std of Reward: 0.875. Training.
[INFO] HideAndSeekAgent. Step: 580000. Time Elapsed: 1907.200 s. Mean Reward: -0.072. Std of Reward: 0.882. Training.
[INFO] HideAndSeekAgent. Step: 600000. Time Elapsed: 1977.287 s. Mean Reward: 0.085. Std of Reward: 0.932. Training.
[WARNING] Trainer has multiple policies, but default behavior only saves the first.
[INFO] Exported results\hns_run1\HideAndSeekAgent\HideAndSeekAgent-599900.onnx
[INFO] HideAndSeekAgent. Step: 620000. Time Elapsed: 2055.341 s. Mean Reward: 0.039. Std of Reward: 0.921. Training.
[INFO] HideAndSeekAgent. Step: 640000. Time Elapsed: 2121.798 s. Mean Reward: 0.040. Std of Reward: 0.900. Training.
[INFO] HideAndSeekAgent. Step: 660000. Time Elapsed: 2202.916 s. Mean Reward: 0.021. Std of Reward: 0.923. Training.
[INFO] HideAndSeekAgent. Step: 680000. Time Elapsed: 2270.624 s. Mean Reward: 0.030. Std of Reward: 0.943. Training.
[INFO] HideAndSeekAgent. Step: 700000. Time Elapsed: 2344.972 s. Mean Reward: 0.058. Std of Reward: 0.929. Training.
[WARNING] Trainer has multiple policies, but default behavior only saves the first.
[INFO] Exported results\hns_run1\HideAndSeekAgent\HideAndSeekAgent-699900.onnx
[INFO] HideAndSeekAgent. Step: 720000. Time Elapsed: 2414.388 s. Mean Reward: -0.021. Std of Reward: 0.867. Training.
[INFO] Learning was interrupted. Please wait while the graph is generated.
[WARNING] Trainer has multiple policies, but default behavior only saves the first.
[WARNING] Trainer has multiple policies, but default behavior only saves the first.
```

Figure A9. Training interruption example and periodic ONNX export messages produced by ML-Agents.