



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ**

**UNIVERSITY OF PIRAEUS**

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ  
ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΥΦΥΕΙΣ ΠΡΑΚΤΟΡΕΣ

Ομάδα:

Επώνυμο	Όνομα	Αριθμός Μητρώου	Email
Κικίδης	Δημήτριος	Π18073	dimitriskikidismg@gmail.com

## Περιεχόμενα

Εισαγωγή .....	3
Παραδοτέα αρχεία.....	3
Περιγραφή προβλήματος.....	4
Προαπαιτούμενα και εγκατάσταση.....	7
Οδηγίες εκτέλεσης εφαρμογής .....	12
Τεκμηρίωση κώδικα.....	15
Εκπαίδευση μοντέλου.....	20
Συμπεράσματα και αποτελέσματα.....	26

## Εισαγωγή

Η εργασία υλοποιήθηκε σε **Unity 2020.3.37f1**. Υλοποιήθηκε το θέμα 5 της εκφώνησης (ML-Agents). Η έκδοση ML-Agents που χρησιμοποιήθηκε είναι η **ML-Agents 2.0.1**. Ο επίσημος σύνδεσμος για την έκδοση αυτή, είναι ο εξής: [https://github.com/Unity-Technologies/ml-agents/blob/release\\_17\\_docs/docs/Readme.md](https://github.com/Unity-Technologies/ml-agents/blob/release_17_docs/docs/Readme.md).

Σκοπός της εργασίας είναι η εκπαίδευση ενός πράκτορα προκειμένου να μπορεί να εκτελέσει κάποιες συγκεκριμένες ενέργειες σε ένα δοσμένο περιβάλλον, προκειμένου να πετύχει κάποιον στόχο. Η εκπαίδευση γίνεται με το πακέτο ML-Agents.

Η ιδέα για την εργασία, καθώς και η ίδια η υλοποίηση, βασίστηκαν σε κάποιον βαθμό στα εξής βίντεο του YouTube:

- <https://www.youtube.com/watch?v=zPFU30tbyKs>
- <https://www.youtube.com/watch?v=supqT7kqpEI>.

## Παραδοτέα αρχεία

- Βίντεο παρουσίασης: <https://www.youtube.com/watch?v=6dkVFMB3ukE>
- PowerPoint παρουσίασης: «presentation.pptx»

Σημείωση: Στο build εκτελείται προσομοίωση με τον πράκτορα να έχει το μοντέλο **FetchBall3**. Η έξοδος από το εκτελέσιμο αρχείο του build γίνεται με Alt + F4.

## Περιγραφή προβλήματος

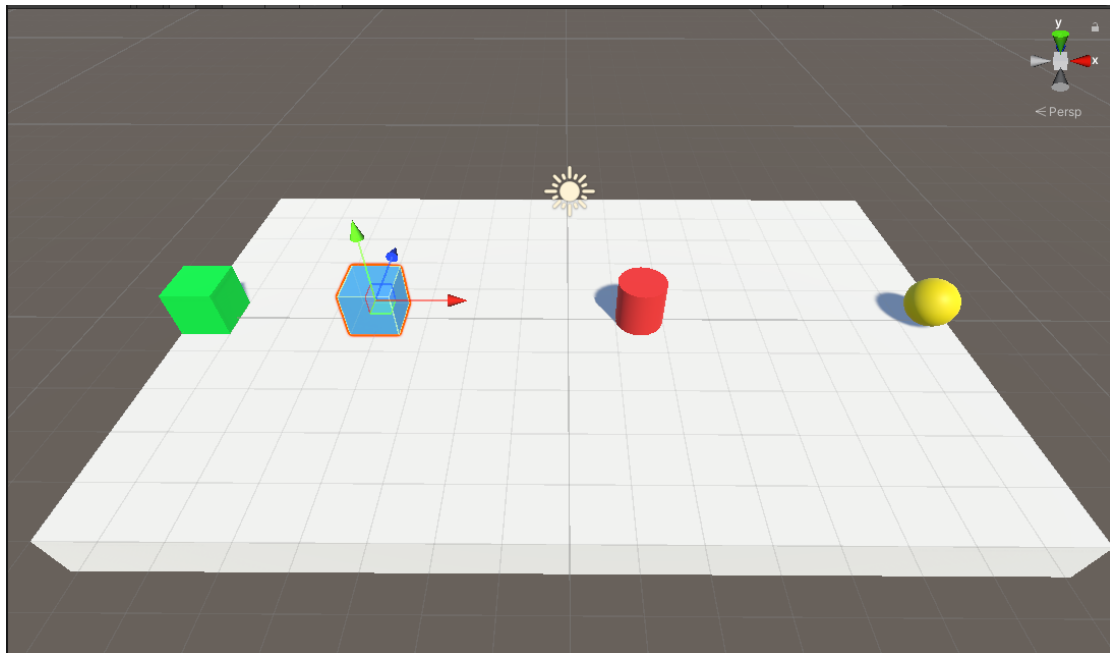
Στην παρούσα εφαρμογή υπάρχει ένας πράκτορας στο περιβάλλον και αντικείμενα με τα οποία έχει αλληλεπίδραση. Στόχος του πράκτορα είναι να συλλέξει μια μπάλα και να επιστρέψει σε ένα σημείο-στόχο, αποφεύγοντας παράλληλα έναν εχθρό που κατευθύνεται συνεχώς προς αυτόν.

Πιο συγκεκριμένα, το περιβάλλον αποτελείται από μια πλατφόρμα πλάτους 16 και μήκους 10. Σε αυτήν, υπάρχουν τα εξής:

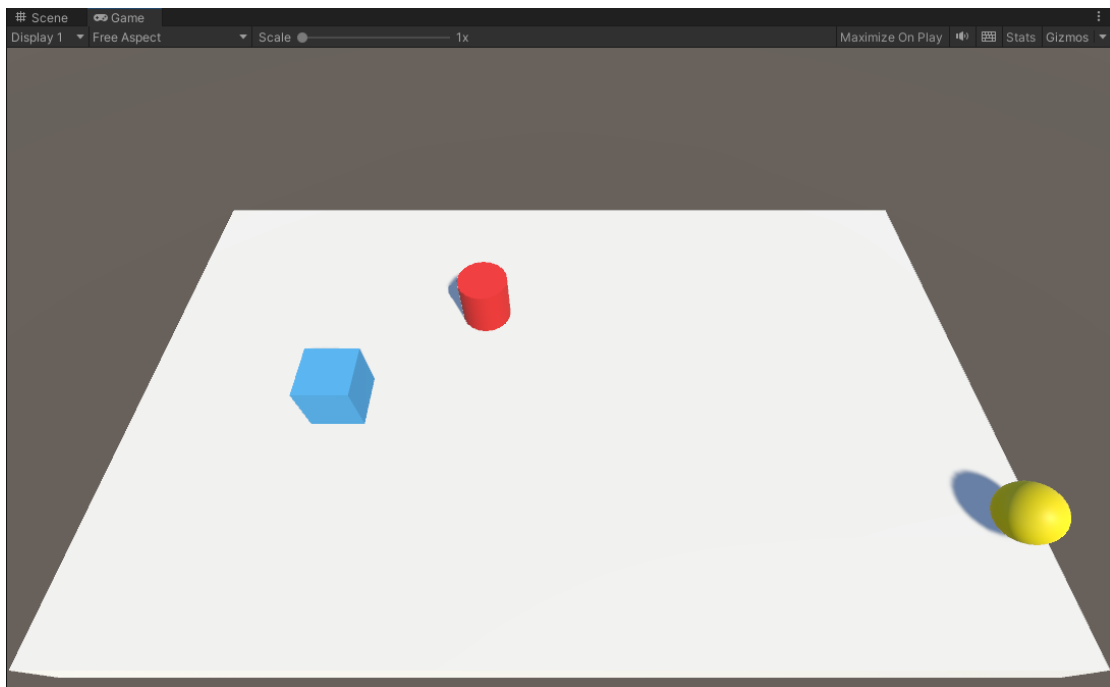
- Πράκτορας (Agent): Ένας μπλε κύβος ο οποίος αντιπροσωπεύει τον πράκτορας που εκπαιδεύτηκε με το πακέτο ML-Agents.
- Μπάλα (Ball): Μια κίτρινη μπάλα την οποία πρέπει να συλλέξει ο πράκτορας.
- Σημείο-Στόχος (Goal): Ένας πράσινος κύβος ο οποίος αντιπροσωπεύει τη θέση στην οποία πρέπει να μεταβεί ο πράκτορας, εφόσον συλλέξει την μπάλα.
- Εχθρός (Enemy): Ένας κόκκινος κύλινδρος ο οποίος προσπαθεί να φτάσει στον πράκτορα.

Κατά την εκκίνηση του περιβάλλοντος, εμφανίζονται ο πράκτορας, η μπάλα και ο εχθρός. Ο πράκτορας πρέπει να φτάσει στην μπάλα για να τη συλλέξει. Όταν συλλέξει την μπάλα, εμφανίζεται το σημείο-στόχος στο οποίο πρέπει να φτάσει ο πράκτορας. Στόχος του πράκτορα είναι να καταφέρει να φτάσει στο σημείο-στόχο χωρίς να χάσει. Ο πράκτορας μπορεί να χάσει αν έρθει σε επαφή με τον εχθρό ή αν καθυστερήσει να φτάσει στο σημείο-στόχο. Από την άλλη, ο εχθρός κατευθύνεται συνεχώς προς την κατεύθυνση του πράκτορα. Επομένως, για να μπορέσει ο πράκτορας να πετύχει τον στόχο του, πρέπει πρώτα να φτάσει στην μπάλα, και έπειτα στο σημείο-στόχο. Αυτό όμως πρέπει να γίνει σε έναν αριθμό βημάτων και ενώ ταυτόχρονα ο πράκτορας αποφεύγει τον εχθρό που κατευθύνεται συνεχώς προς αυτόν. Ο μέγιστος αριθμός βημάτων έχει οριστεί να είναι το 1000, ενώ κάθε δευτερόλεπτο εκτελούνται 50 βήματα.

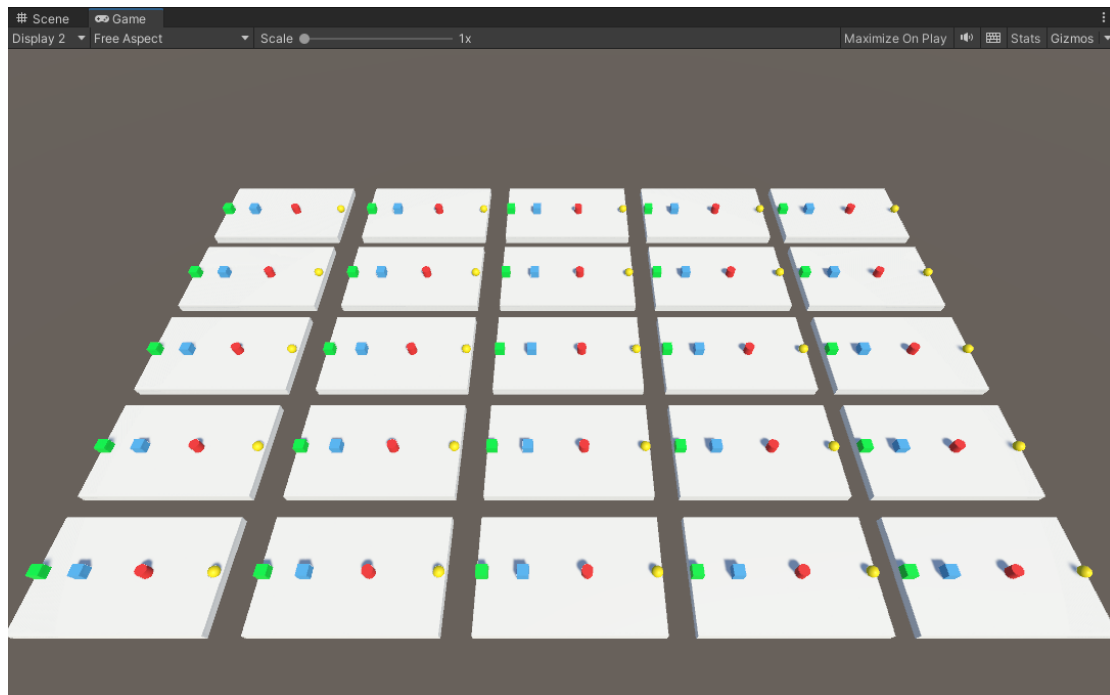
Ακολουθεί ένα screenshot της εφαρμογής πριν την εκτέλεση.



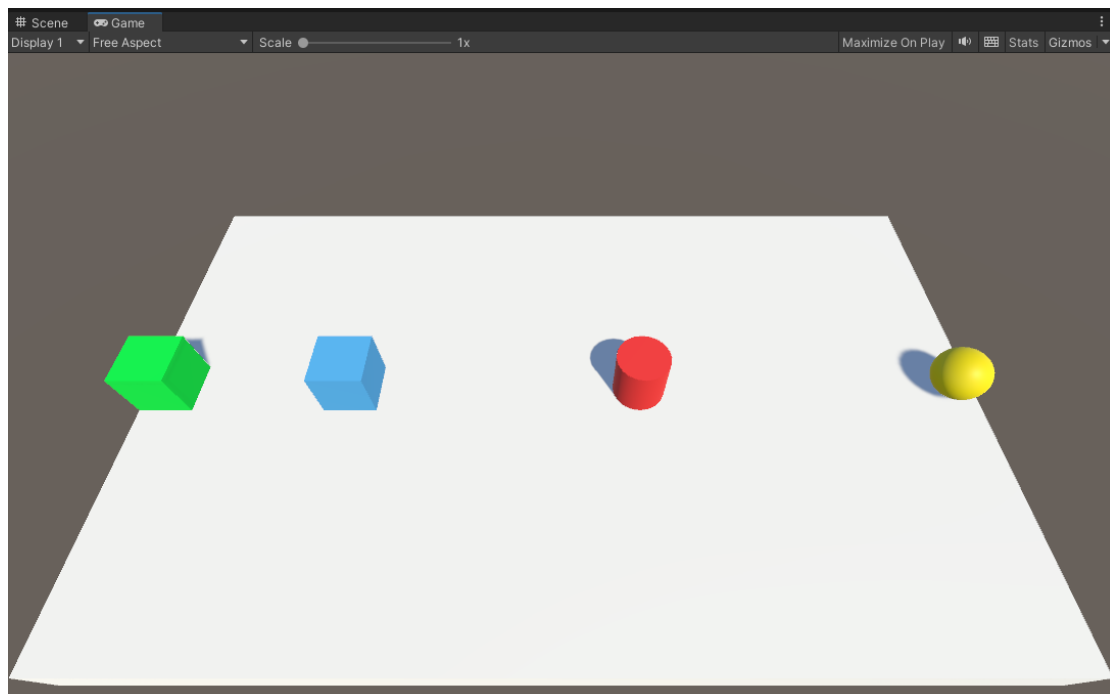
Ακολουθεί ένα screenshot της εφαρμογής κατά τη διάρκεια της εκτέλεσης.



Αλλάζοντας από Display 1 σε Display 2, όταν είναι ενεργοποιημένο το **TrainingEnvironments** και απενεργοποιημένο το **Environment**, φαίνονται όλα τα περιβάλλοντα εκπαίδευσης



Αλλάζοντας πάλι σε Display 1, ενεργοποιώντας το **Environment** και απενεργοποιώντας το **TrainingEnvironments** γίνεται επιστροφή στην προηγούμενη κατάσταση θέασης.



## Προαπαιτούμενα και εγκατάσταση

Μοναδικό προαπαιτούμενο για την εγκατάσταση του project και την εκτέλεσή του με κάποιο ήδη εκπαιδευμένο μοντέλο, είναι η **Unity 2020.3.x**. Για το project χρησιμοποιήθηκε **Unity 2020.3.37f1**. Το πακέτο **ML-Agents 2.0.1** της Unity είναι ήδη εγκατεστημένο στο project. Στο project, υπάρχουν ήδη 5 εκπαιδευμένα μοντέλα, τα οποία μπορούν να χρησιμοποιηθούν απευθείας από τον πράκτορα, χωρίς να γίνει εκπαίδευση ενός μοντέλου από την αρχή.

Για να είναι επιπλέον δυνατή και η εκπαίδευση ενός μοντέλου από την αρχή, χρειάζεται να εγκατασταθεί το πακέτο **mlagents** της Python. Προαπαιτούμενο για την εγκατάσταση αυτού του πακέτου, είναι το **conda** package manager. Για το project χρησιμοποιήθηκε το **conda 4.13.0** από το Miniconda, ενώ στο virtual environment που δημιουργήθηκε χρησιμοποιήθηκε η **Python 3.7.9**.

Ακολουθούν οδηγίες εγκατάστασης του πακέτου **mlagents** της Python με **conda 4.13.0** και **Python 3.7.9**. Οι οδηγίες αυτές προορίζονται για την έκδοση **ML-Agents 2.0.1** της Unity, όπως αναφέρεται στον σύνδεσμο [https://github.com/Unity-Technologies/ml-agents/blob/release\\_17\\_docs/docs/Installation.md](https://github.com/Unity-Technologies/ml-agents/blob/release_17_docs/docs/Installation.md).

1. Δημιουργία ενός Python virtual environment με όνομα **ml-agents**, με την εκτέλεση της εντολής

**conda create -n ml-agents python=3.7.9**

2. Απάντηση με «γ» στην εμφανιζόμενη ερώτηση

```
Anaconda Prompt (miniconda3)

(base) C:\Users\Dimitris>conda create -n ml-agents python=3.7.9
Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\Dimitris\miniconda3\envs\ml-agents

  added / updated specs:
    - python=3.7.9

The following NEW packages will be INSTALLED:

ca-certificates      pkgs/main/win-64::ca-certificates-2022.07.19-haa95532_0
certifi              pkgs/main/win-64::certifi-2022.6.15-py37haa95532_0
openssl              pkgs/main/win-64::openssl-1.1.1q-h2bbff1b_0
pip                  pkgs/main/win-64::pip-22.1.2-py37haa95532_0
python               pkgs/main/win-64::python-3.7.9-h60c2a47_0
setuptools           pkgs/main/win-64::setuptools-61.2.0-py37haa95532_0
sqlite               pkgs/main/win-64::sqlite-3.38.5-h2bbff1b_0
vc                   pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime       pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel                pkgs/main/noarch::wheel-0.37.1-pyhd3eb1b0_0
wincertstore         pkgs/main/win-64::wincertstore-0.2-py37haa95532_2

Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate ml-agents
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

3. Ενεργοποίηση του virtual environment που μόλις δημιουργήθηκε, με την εκτέλεση της εντολής

**conda activate ml-agents**

4. Εγκατάσταση του πακέτου **PyTorch** της Python, με την εκτέλεση της εντολής

**pip3 install torch~=1.7.1 -f https://download.pytorch.org/whl/torch\_stable.html**

```
Anaconda Prompt (miniconda3)

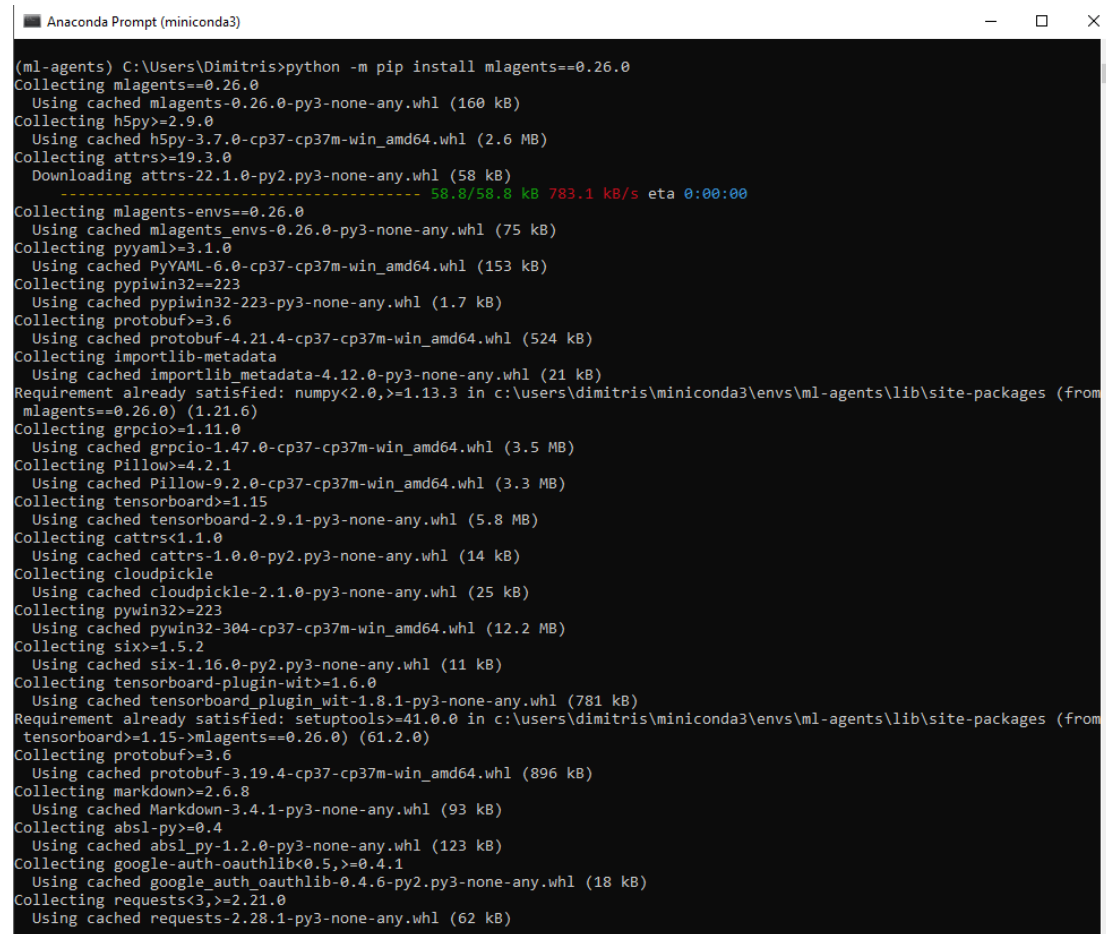
(ml-agents) C:\Users\Dimitris>conda activate ml-agents

(ml-agents) C:\Users\Dimitris>pip3 install torch~=1.7.1 -f https://download.pytorch.org/whl/torch_stable.html
Looking in links: https://download.pytorch.org/whl/torch_stable.html
Collecting torch~=1.7.1
  Using cached https://download.pytorch.org/whl/cu110/torch-1.7.1%2Bcu110-cp37-cp37m-win_amd64.whl (2050.2 MB)
Collecting typing_extensions
  Using cached typing_extensions-4.3.0-py3-none-any.whl (25 kB)
Collecting numpy
  Using cached numpy-1.21.6-cp37-cp37m-win_amd64.whl (14.0 MB)
Installing collected packages: typing_extensions, numpy, torch
Successfully installed numpy-1.21.6 torch-1.7.1+cu110 typing_extensions-4.3.0
```



5. Εγκατάσταση του πακέτου **mlagents** της Python, με την εκτέλεση της εντολής

**python -m pip install mlagents==0.26.0**



```
Anaconda Prompt (miniconda3)

(ml-agents) C:\Users\Dimitris>python -m pip install mlagents==0.26.0
Collecting mlagents==0.26.0
  Using cached mlagents-0.26.0-py3-none-any.whl (160 kB)
Collecting h5py>=2.9.0
  Using cached h5py-3.7.0-cp37-cp37m-win_amd64.whl (2.6 MB)
Collecting attrs>=19.3.0
  Downloading attrs-22.1.0-py2.py3-none-any.whl (58 kB)
----- 58.8/58.8 kB 783.1 kB/s eta 0:00:00
Collecting mlagents-envs==0.26.0
  Using cached mlagents_envs-0.26.0-py3-none-any.whl (75 kB)
Collecting pyyaml>=3.1.0
  Using cached PyYAML-6.0-cp37-cp37m-win_amd64.whl (153 kB)
Collecting pywin32==223
  Using cached pywin32-223-py3-none-any.whl (1.7 kB)
Collecting protobuf>=3.6
  Using cached protobuf-4.21.4-cp37-cp37m-win_amd64.whl (524 kB)
Collecting importlib-metadata
  Using cached importlib_metadata-4.12.0-py3-none-any.whl (21 kB)
Requirement already satisfied: numpy<2.0,>=1.13.3 in c:\users\dimitris\miniconda3\envs\ml-agents\lib\site-packages (from mlagents==0.26.0) (1.21.6)
Collecting grpcio>=1.11.0
  Using cached grpcio-1.47.0-cp37-cp37m-win_amd64.whl (3.5 MB)
Collecting Pillow>=4.2.1
  Using cached Pillow-9.2.0-cp37-cp37m-win_amd64.whl (3.3 MB)
Collecting tensorboard>=1.15
  Using cached tensorboard-2.9.1-py3-none-any.whl (5.8 MB)
Collecting catrtts<1.1.0
  Using cached catrtts-1.0.0-py2.py3-none-any.whl (14 kB)
Collecting cloudpickle
  Using cached cloudpickle-2.1.0-py3-none-any.whl (25 kB)
Collecting pywin32==223
  Using cached pywin32-304-cp37-cp37m-win_amd64.whl (12.2 MB)
Collecting six>=1.5.2
  Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Collecting tensorboard-plugin-wit>=1.6.0
  Using cached tensorboard_plugin_wit-1.8.1-py3-none-any.whl (781 kB)
Requirement already satisfied: setuptools>=41.0.0 in c:\users\dimitris\miniconda3\envs\ml-agents\lib\site-packages (from tensorboard==1.15->mlagents==0.26.0) (61.2.0)
Collecting protobuf>=3.6
  Using cached protobuf-3.19.4-cp37-cp37m-win_amd64.whl (896 kB)
Collecting markdown>=2.6.8
  Using cached Markdown-3.4.1-py3-none-any.whl (93 kB)
Collecting absl-py>=0.4
  Using cached absl_py-1.2.0-py3-none-any.whl (123 kB)
Collecting google-auth-oauthlib<0.5,>=0.4.1
  Using cached google_auth_oauthlib-0.4.6-py2.py3-none-any.whl (18 kB)
Collecting requests<3,>=2.21.0
  Using cached requests-2.28.1-py3-none-any.whl (62 kB)
```

```

Anaconda Prompt (miniconda3)
Using cached requests-2.28.1-py3-none-any.whl (62 kB)
Collecting google-auth<3,>=1.6.3
Using cached google_auth-2.9.1-py2.py3-none-any.whl (167 kB)
Collecting tensorboard-data-server<0.7.0,>=0.6.0
Using cached tensorboard_data_server-0.6.1-py3-none-any.whl (2.4 kB)
Requirement already satisfied: wheel>=0.26 in c:\users\dimitris\miniconda3\envs\ml-agents\lib\site-packages (from tensorboard>=1.15->mlagents==0.26.0) (0.37.1)
Collecting werkzeug>=1.0.1
Downloading Werkzeug-2.2.1-py3-none-any.whl (232 kB)
----- 232.4/232.4 kB 1.8 MB/s eta 0:00:00
Collecting zipp>=0.5
Using cached zipp-3.8.1-py3-none-any.whl (5.6 kB)
Requirement already satisfied: typing-extensions>=3.6.4 in c:\users\dimitris\miniconda3\envs\ml-agents\lib\site-packages (from importlib-metadata->mlagents==0.26.0) (4.3.0)
Collecting rsa<5,>=3.1.4
Using cached rsa-4.9-py3-none-any.whl (34 kB)
Collecting cachetools<6.0,>=2.0.0
Using cached cachetools-5.2.0-py3-none-any.whl (9.3 kB)
Collecting pyasn1-modules>=0.2.1
Using cached pyasn1_modules-0.2.8-py2.py3-none-any.whl (155 kB)
Collecting requests-oauthlib>=0.7.0
Using cached requests_oauthlib-1.3.1-py2.py3-none-any.whl (23 kB)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dimitris\miniconda3\envs\ml-agents\lib\site-packages (from requests<3,>=2.21.0->tensorboard>=1.15->mlagents==0.26.0) (2022.6.15)
Collecting charset-normalizer<3,>=2
Using cached charset_normalizer-2.1.0-py3-none-any.whl (39 kB)
Collecting urllib3<1.27,>=1.21.1
Using cached urllib3-1.26.11-py2.py3-none-any.whl (139 kB)
Collecting idna<4,>=2.5
Using cached idna-3.3-py3-none-any.whl (61 kB)
Collecting MarkupSafe>=2.1.1
Using cached MarkupSafe-2.1.1-cp37-cp37m-win_amd64.whl (17 kB)
Collecting pyasn1<0.5.0,>=0.4.6
Using cached pyasn1-0.4.8-py3-none-any.whl (77 kB)
Collecting oauthlib>=3.0.0
Using cached oauthlib-3.2.0-py3-none-any.whl (151 kB)
Installing collected packages: tensorboard-plugin-wit, pywin32, pyasn1, zipp, urllib3, tensorboard-data-server, six, rsa, pyyaml, pypiwin32, pyasn1-modules, protobuf, Pillow, oauthlib, MarkupSafe, idna, h5py, cloudpickle, charset-normalizer, cachetools, attrs, absl-py, werkzeug, requests, importlib-metadata, grpcio, google-auth, cattrs, requests-oauthlib, ml-agents-envs, markdown, google-auth-oauthlib, tensorboard, mlagents
Successfully installed MarkupSafe-2.1.1 Pillow-9.2.0 absl-py-1.2.0 attrs-22.1.0 cachetools-5.2.0 cattrs-1.0.0 charset-normalizer-2.1.0 cloudpickle-2.1.0 google-auth-2.9.1 google-auth-oauthlib-0.4.6 grpcio-1.47.0 h5py-3.7.0 idna-3.3 importlib-metadata-4.12.0 markdown-3.4.1 mlagents-0.26.0 mlagents-envs-0.26.0 oauthlib-3.2.0 protobuf-3.19.4 pyasn1-0.4.8 pyasn1-modules-0.2.8 pypiwin32-223 pywin32-304 pyyaml-6.0 requests-2.28.1 requests-oauthlib-1.3.1 rsa-4.9 six-1.16.0 tensorboard-2.9.1 tensorboard-data-server-0.6.1 tensorboard-plugin-wit-1.8.1 urllib3-1.26.11 werkzeug-2.2.1 zipp-3.8.1

```

6. Αν εγκαταστάθηκαν όλα σωστά θα πρέπει να εμφανίζεται το περιεχόμενο της παρακάτω εικόνας, με την εκτέλεση της εντολής

### mlagents-learn --help

```
Anaconda Prompt (miniconda3)

(ml-agents) C:\Users\Dimitris>mlagents-learn --help
C:\Users\Dimitris\miniconda3\envs\ml-agents\lib\site-packages\torch\cuda\_init_.py:52: UserWarning: CUDA initialization
n: Found no NVIDIA driver on your system. Please check that you have an NVIDIA GPU and installed a driver from http://ww
w.nvidia.com/Download/index.aspx (Triggered internally at  ..\c10\cuda\CUDAFuncions.cpp:100.)
  return torch._C._cuda_getDeviceCount() > 0
usage: mlagents-learn.exe [-h] [--env ENV_PATH] [--resume] [--force]
                        [--run-id RUN_ID] [--initialize-from RUN_ID]
                        [--seed SEED] [--inference] [--base-port BASE_PORT]
                        [--num-envs NUM_ENVS] [--debug] [--env-args ...]
                        [--torch] [--tensorflow] [--results-dir RESULTS_DIR]
                        [--width WIDTH] [--height HEIGHT]
                        [--quality-level QUALITY_LEVEL]
                        [--time-scale TIME_SCALE]
                        [--target-frame-rate TARGET_FRAME_RATE]
                        [--capture-frame-rate CAPTURE_FRAME_RATE]
                        [--no-graphics] [--torch-device DEVICE]
                        [trainer_config_path]

positional arguments:
  trainer_config_path

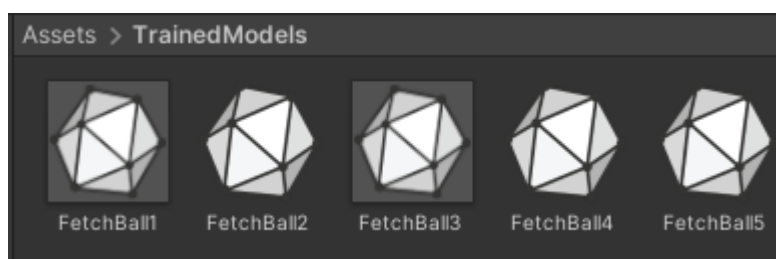
optional arguments:
  -h, --help            show this help message and exit
  --env ENV_PATH        Path to the Unity executable to train (default: None)
  --resume             Whether to resume training from a checkpoint. Specify
                        a --run-id to use this option. If set, the training
                        code loads an already trained model to initialize the
                        neural network before resuming training. This option
                        is only valid when the models exist, and have the same
                        behavior names as the current agents in your scene.
                        (default: False)
  --force              Whether to force-overwrite this run-id's existing
                        summary and model data. (Without this flag, attempting
                        to train a model with a run-id that has been used
                        before will throw an error. (default: False)
  --run-id RUN_ID      The identifier for the training run. This identifier
                        is used to name the subdirectories in which the
                        trained model and summary statistics are saved as well
                        as the saved model itself. If you use TensorBoard to
                        view the training statistics, always set a unique run-
                        id for each training run. (The statistics for all runs
                        with the same id are combined as if they were produced
                        by a the same session.) (default: ppo)
  --initialize-from RUN_ID
                        Specify a previously saved run ID from which to
                        initialize the model from. This can be used, for
                        instance, to fine-tune an existing model on a new
                        environment. Note that the previously saved models
```

## Οδηγίες εκτέλεσης εφαρμογής

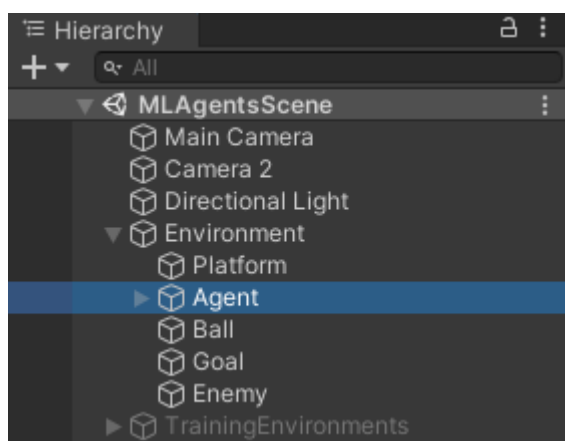
Στον φάκελο **TrainedModels** υπάρχουν ήδη 5 εκπαιδευμένα μοντέλα. Αυτό με την καλύτερη επίδοση, είναι το μοντέλο με όνομα **FetchBall3**. Η εφαρμογή λειτουργεί κατευθείαν, αφού έχει ανατεθεί ήδη στον πράκτορα αυτό το μοντέλο.

Ακολουθούν τα βήματα για την αλλαγή του μοντέλου που έχει ανατεθεί στον πράκτορα:

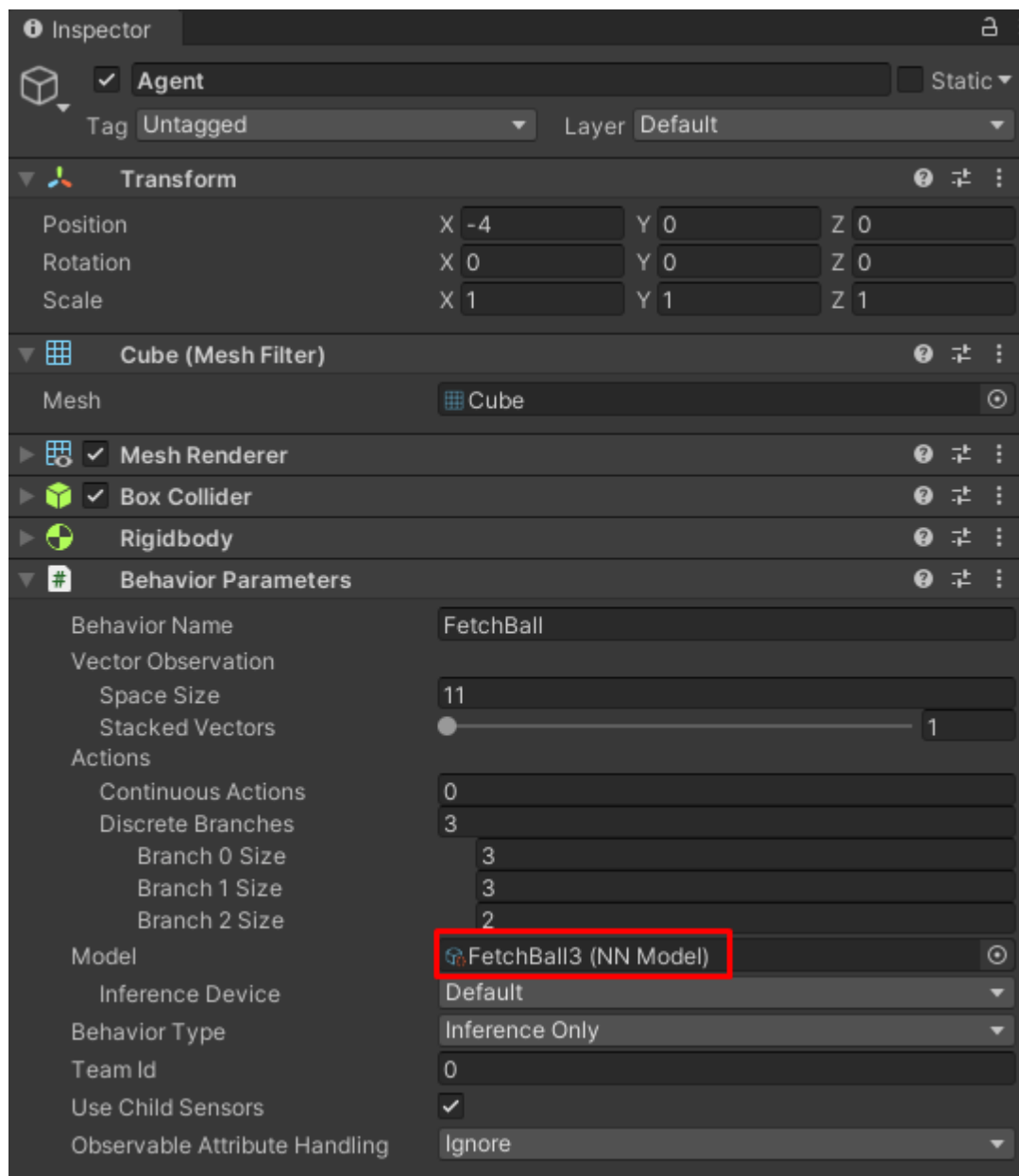
Μετάβαση στον φάκελο **TrainedModels**.



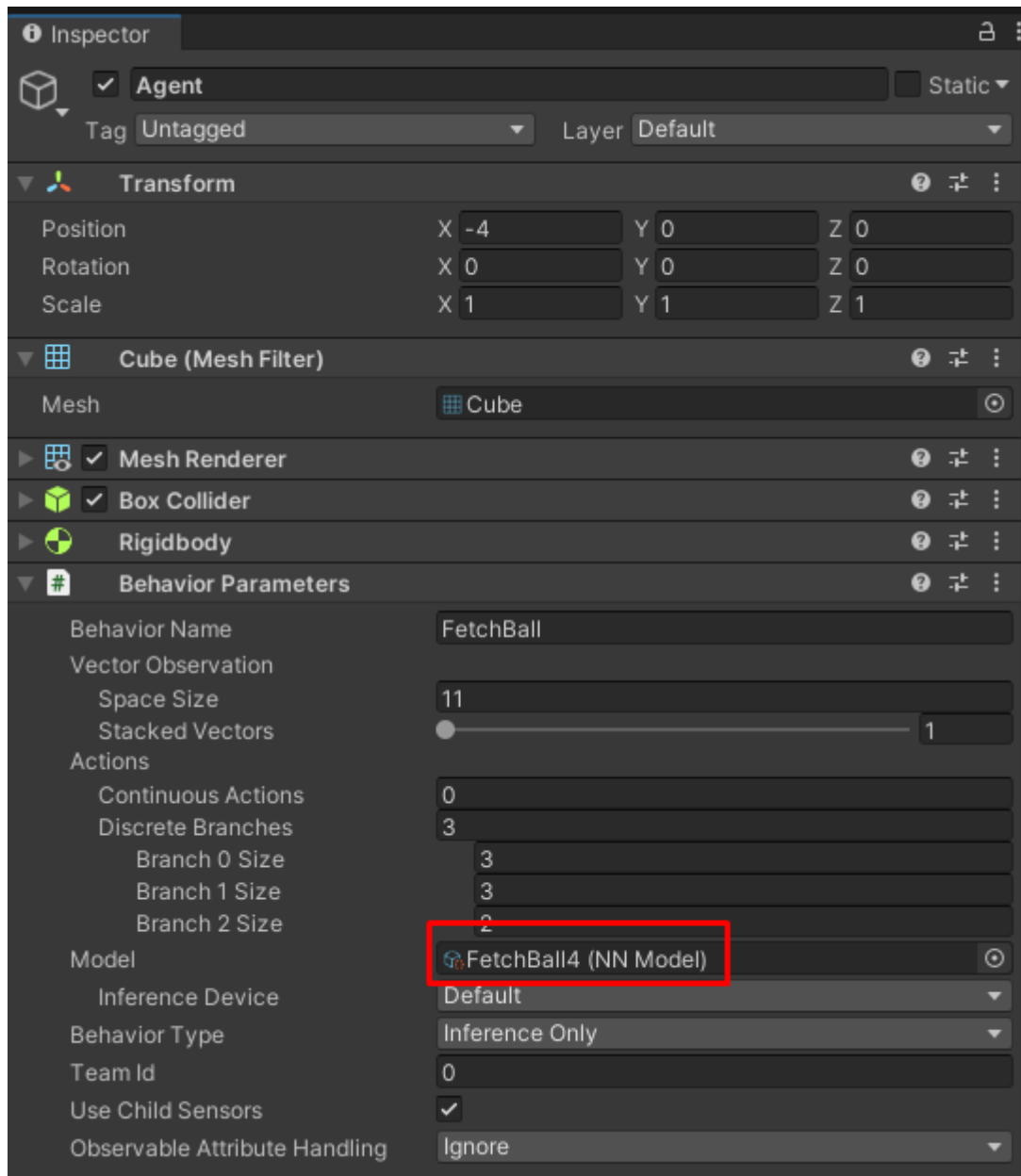
Στο Hierarchy του project, επιλέγεται **Environment > Agent**.



Στον Inspector, μετάβαση στο **Behavior Parameters** -> **Model**.



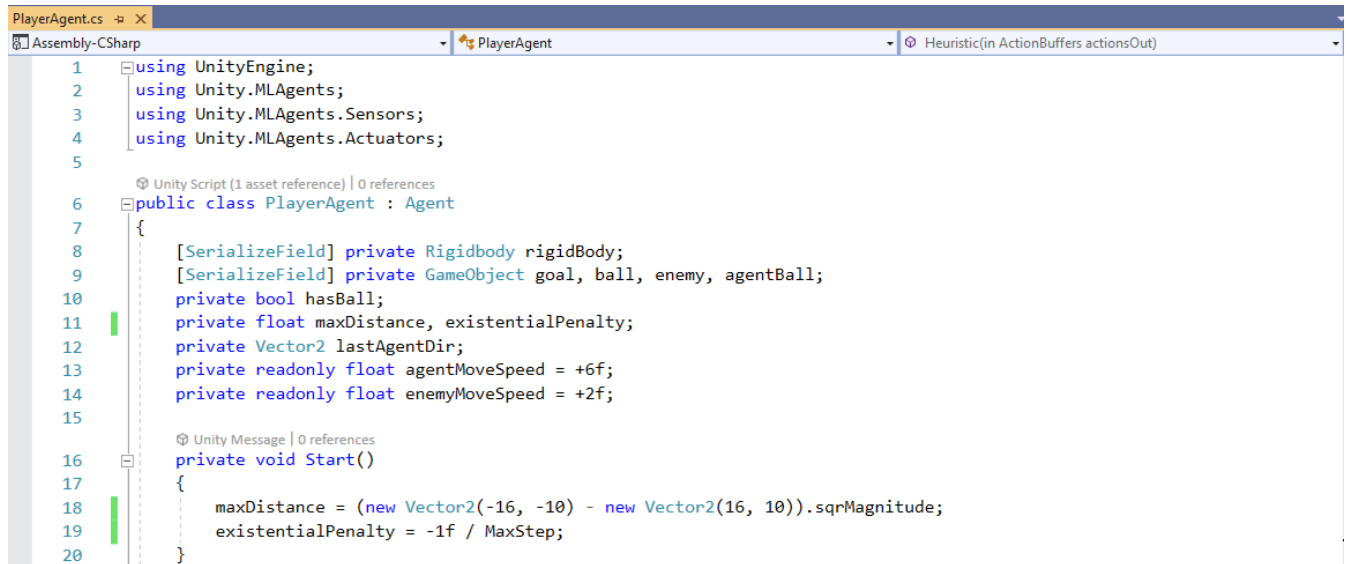
Αλλαγή με Drag and drop ενός εκπαιδευμένου μοντέλου του φακέλου **TrainedModels** στο πεδίο **Model** του component **Behavior Parameters**. Εδώ ανατέθηκε το μοντέλο **FetchBall4**.



## Τεκμηρίωση κώδικα

Ο κώδικας αποτελείται από ένα κεντρικό script που ονομάζεται **PlayerAgent.cs**. Το script βρίσκεται στον φάκελο **Scripts**.

Στην αρχή υπάρχουν δηλώσεις μεταβλητών, ενώ στη συνάρτηση **Start()** αρχικοποιούνται δύο μεταβλητές.



```
1 using UnityEngine;
2 using Unity.MLAgents;
3 using Unity.MLAgents.Sensors;
4 using Unity.MLAgents.Actuators;
5
6 public class PlayerAgent : Agent
7 {
8     [SerializeField] private Rigidbody rigidBody;
9     [SerializeField] private GameObject goal, ball, enemy, agentBall;
10    private bool hasBall;
11    private float maxDistance, existentialPenalty;
12    private Vector2 lastAgentDir;
13    private readonly float agentMoveSpeed = +6f;
14    private readonly float enemyMoveSpeed = +2f;
15
16    private void Start()
17    {
18        maxDistance = (new Vector2(-16, -10) - new Vector2(16, 10)).sqrMagnitude;
19        existentialPenalty = -1f / MaxStep;
20    }
```

Στη συνάρτηση **OnEpisodeBegin()** γίνεται η επανεκκίνηση της σκηνής. Κάθε φορά που γίνεται επανεκκίνηση της σκηνής, η μπάλα και ο στόχος έχουν τις ίδιες συντεταγμένες x, αλλά τυχαίες συντεταγμένες z εντός ενός διαστήματος. Ο πράκτορας αλλά και ο εχθρός έχουν τυχαίες συντεταγμένες x και z εντός ενός διαστήματος. Τα διαστήματα αυτά φαίνονται στον κώδικα. Η συνάρτηση αυτή καλείται όταν ο πράκτορας πετυχαίνει τον στόχο του (δηλαδή φτάνει στο σημείο-στόχο) ή όταν χάνει (υπέρβαση των 1000 βημάτων ή επαφή με τον εχθρό).

```
PlayerAgent.cs
Assembly-CSharp PlayerAgent Heuristic(in ActionBuffers actionsOut)

20 }
21
22 0 references
23 public override void OnEpisodeBegin()
24 {
25     // Reset the scene
26     transform.localPosition = new Vector3(Random.Range(-5f, 5f), 0f, Random.Range(-4.5f, 4.5f));
27     transform.localRotation = Quaternion.identity;
28     Vector2 agentPos = new Vector2(transform.localPosition.x, transform.localPosition.z);
29     Vector2 ballPos = new Vector2(ball.transform.localPosition.x, ball.transform.localPosition.z);
30     lastAgentDir = (ballPos - agentPos).normalized;
31     goal.transform.localPosition = new Vector3(-7.5f, 0f, Random.Range(-3f, 3f));
32     ball.transform.localPosition = new Vector3(+7.5f, 0f, Random.Range(-3f, 3f));
33     enemy.transform.localPosition = new Vector3(Random.Range(0f, 2f), 0f, Random.Range(-4f, 4f));
34     hasBall = false;
35     goal.SetActive(false);
36     ball.SetActive(true);
37     agentBall.SetActive(false);
38 }
```



Στη συνάρτηση **CollectObservations()** συλλέγονται κάθε φορά 11 παρατηρήσεις από το περιβάλλον του πράκτορα, προκειμένου να παρθεί μια απόφαση από το μοντέλο που εκπαιδεύεται ή που έχει εκπαιδευτεί. Για τη μπάλα συλλέγονται 4 παρατηρήσεις, για το σημείο-στόχο συλλέγονται 3 παρατηρήσεις και για τον εχθρό συλλέγονται 4 παρατηρήσεις.

```
PlayerAgent.cs  Assembly-CSharp  PlayerAgent  Heuristic(in ActionBuffers actionsOut)
0 references
39 public override void CollectObservations(VectorSensor sensor)
40 {
41     Vector2 agentPos = new Vector2(transform.localPosition.x, transform.localPosition.z);
42
43     // Observations for the ball
44     int hasBallInt = hasBall ? 1 : 0;
45     Vector2 ballPos = new Vector2(ball.transform.localPosition.x, ball.transform.localPosition.z);
46     Vector2 dirToBall = ballPos - agentPos;
47     float distanceToBall = dirToBall.sqrMagnitude / maxDistance;
48     Vector2 dirToBallNormalized = dirToBall.normalized;
49     sensor.AddObservation(hasBallInt);
50     sensor.AddObservation(distanceToBall);
51     sensor.AddObservation(dirToBallNormalized.x);
52     sensor.AddObservation(dirToBallNormalized.y);
53
54     // Observations for the goal
55     Vector2 goalPos = new Vector2(goal.transform.localPosition.x, goal.transform.localPosition.z);
56     Vector2 dirToGoal = goalPos - agentPos;
57     float distanceToGoal = dirToGoal.sqrMagnitude / maxDistance;
58     Vector2 dirToGoalNormalized = dirToGoal.normalized;
59     sensor.AddObservation(distanceToGoal);
60     sensor.AddObservation(dirToGoalNormalized.x);
61     sensor.AddObservation(dirToGoalNormalized.y);
62
63     // Observations for the enemy
64     Vector2 enemyPos = new Vector2(enemy.transform.localPosition.x, enemy.transform.localPosition.z);
65     Vector2 dirToEnemy = enemyPos - agentPos;
66     float distanceToEnemy = (enemyPos - agentPos).sqrMagnitude / maxDistance;
67     Vector2 dirToEnemyNormalized = dirToEnemy.normalized;
68     float agentDirAngleToEnemy = Vector2.Angle(lastAgentDir, dirToEnemyNormalized) / 180f;
69     sensor.AddObservation(distanceToEnemy);
70     sensor.AddObservation(dirToEnemyNormalized.x);
71     sensor.AddObservation(dirToEnemyNormalized.y);
72     sensor.AddObservation(agentDirAngleToEnemy);
73 }
74
```

Στη συνάρτηση **OnActionReceived()** επιστρέφονται οι αποφάσεις του μοντέλου. Οι αποφάσεις αυτές είναι 2 διακριτές τιμές, η κάθε μια από τις οποίες μπορεί να είναι 0, 1 ή 2. Οι τιμές αυτές μεταφράζονται σε κίνηση ή μη κίνηση κατά τον άξονα x και κατά τον άξονα z αντίστοιχα. Συγκεκριμένα:

- 1<sup>η</sup> τιμή:
  - 0: κίνηση αριστερά
  - 1: καθόλου κίνηση
  - 2: κίνηση δεξιά
- 2<sup>η</sup> τιμή:
  - 0: κίνηση πίσω
  - 1: καθόλου κίνηση
  - 2: κίνηση μπροστά

Ακόμα, στη συνάρτηση αυτή γίνεται και η κίνηση του εχθρού, ο οποίος κατευθύνεται συνεχώς προς τον πράκτορα. Τέλος, σε κάθε βήμα που εκτελείται, αφαιρείται ένα μικρό ποσό ( $1/1000 = 0.001$ ) από τη βαθμολογία του πράκτορα, έτσι ώστε να εκπαιδεύεται το μοντέλο στο να πετυχαίνει ο πράκτορας τον στόχο του σε ένα εύλογο διάστημα (το πολύ σε 1000 βήματα). Αν ο πράκτορας υπερβεί τα 1000 βήματα, εφόσον σε κάθε βήμα χάνει 0.001 βαθμούς, λαμβάνει τελική βαθμολογία  $-(1000 * 0.001) = -1$  σε αυτή την προσπάθεια.

```
PlayerAgent.cs + x
Assembly-CSharp PlayerAgent Heuristic(in ActionBuffers actionsOut)
0 references
75 public override void OnActionReceived(ActionBuffers actions)
76 {
77     // Move the agent
78     int moveX = actions.DiscreteActions[0] - 1; // -1 -> Left, 0 -> No Move, +1 -> Right
79     int moveZ = actions.DiscreteActions[1] - 1; // -1 -> Back, 0 -> No Move, +1 -> Forward
80     Vector3 force = new Vector3(moveX, 0f, moveZ);
81     Vector2 lastAgentPos = transform.localPosition;
82     transform.localPosition += agentMoveSpeed * Time.deltaTime * force;
83
84     // Out of bounds constraints
85     float newX = transform.localPosition.x;
86     float newZ = transform.localPosition.z;
87     if (newX >= +7.5f) newX = +7.5f;
88     if (newX <= -7.5f) newX = -7.5f;
89     if (newZ >= +4.5f) newZ = +4.5f;
90     if (newZ <= -4.5f) newZ = -4.5f;
91     transform.localPosition = new Vector3(newX, 0f, newZ);
92
93     // Calculate the direction of the agent
94     lastAgentDir = (new Vector2(transform.localPosition.x, transform.localPosition.z) - lastAgentPos).normalized;
95
96     // Move the enemy
97     Vector2 agentPos = new Vector2(transform.localPosition.x, transform.localPosition.z);
98     Vector2 enemyPos = new Vector2(enemy.transform.localPosition.x, enemy.transform.localPosition.z);
99     Vector2 enemyDirToPlayer = (agentPos - enemyPos).normalized;
100    Vector3 enemyForce = new Vector3(enemyDirToPlayer.x, 0f, enemyDirToPlayer.y);
101    enemy.transform.localPosition += enemyMoveSpeed * Time.deltaTime * enemyForce;
102
103    // Add an existential penalty
104    AddReward(existentialPenalty);
105 }
```

Η συνάρτηση **Heuristic()** είναι βοηθητική συνάρτηση και χρησιμοποιείται μόνο αν δεν έχει ανατεθεί κάποιο εκπαιδευμένο μοντέλο στον πράκτορα. Σε αυτή την περίπτωση, οι αποφάσεις που επιστρέφονται στην **OnActionReceived()** δεν προέρχονται από κάποιο εκπαιδευμένο μοντέλο, αλλά από τον ίδιο τον χρήστη με τα πλήκτρα W (μπροστά), S (πίσω), A (αριστερά) και D (δεξιά).

```
PlayerAgent.cs Assembly-CSharp PlayerAgent OnTriggerEnter(Collider other)
106
107 // If the agent has not been assigned a trained model,
108 // the player controls the agent's movement.
109 public override void Heuristic(in ActionBuffers actionsOut)
110 {
111     ActionSegment<int> discreteActions = actionsOut.DiscreteActions;
112     discreteActions[0] = Mathf.RoundToInt(Input.GetAxisRaw("Horizontal")) + 1;
113     discreteActions[1] = Mathf.RoundToInt(Input.GetAxisRaw("Vertical")) + 1;
114 }
115
```

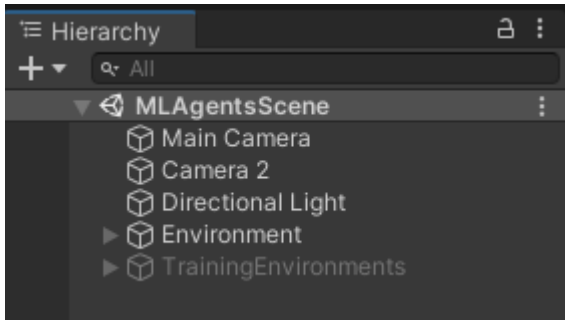
Στη συνάρτηση **OnTriggerEnter()** γίνεται ο έλεγχος για το αν ο πράκτορας ακούμπησε την μπάλα, το σημείο-στόχο ή τον εχθρό. Αν ο πράκτορας φτάσει στην μπάλα, κερδίζει 0.5 βαθμούς. Αν φτάσει στο σημείο-στόχο, κερδίζει άλλους 0.5 βαθμούς και διαμορφώνεται η τελική του βαθμολογία για αυτή την προσπάθεια ως η βαθμολογία που έχει εκείνη τη στιγμή. Η τιμή αυτή εξαρτάται από το πόσα βήματα έχουν εκτελεστεί, αφού χάνει 0.001 βαθμούς ανά βήμα. Αν ο πράκτορας φτάσει στον εχθρό, τότε η βαθμολογία του γίνεται -1 σε αυτή την προσπάθεια.

```
PlayerAgent.cs Assembly-CSharp PlayerAgent Start()
116 private void OnTriggerEnter(Collider other)
117 {
118     if (other.CompareTag("Ball")) // The agent touched the ball
119     {
120         hasBall = true;
121         ball.SetActive(false);
122         goal.SetActive(true);
123         agentBall.SetActive(true);
124         AddReward(+0.5f);
125     }
126     else if (other.CompareTag("Goal")) // The agent touched the goal
127     {
128         AddReward(+0.5f);
129         EndEpisode();
130     }
131     else if (other.CompareTag("Enemy")) // The enemy touched the agent
132     {
133         SetReward(-1f);
134         EndEpisode();
135     }
136 }
137
138
```

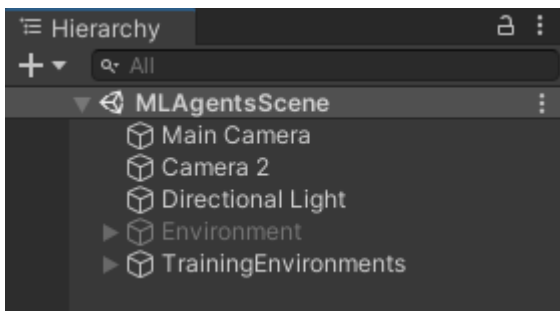
## Εκπαίδευση μοντέλου

Για να είναι δυνατή η εκπαίδευση ενός μοντέλου, πρέπει πρώτα να έχει γίνει η εγκατάσταση του πακέτου **mlagents** της Python, όπως περιγράφεται στις οδηγίες εγκατάστασης. Εφόσον έχει ολοκληρωθεί αυτή η διαδικασία, τα βήματα για την εκπαίδευση ενός μοντέλου από την αρχή, είναι τα εξής:

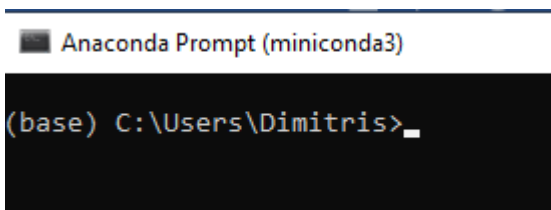
Μετάβαση στο Hierarchy του project.



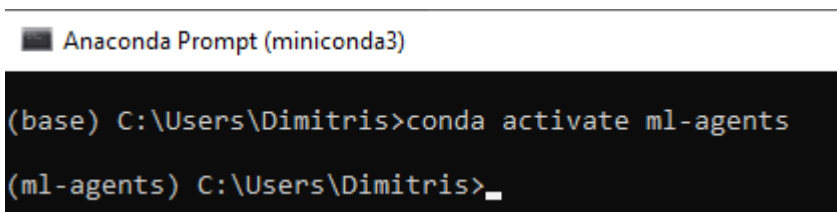
Απενεργοποίηση του **Environment** και ενεργοποίηση του **TrainingEnvironments**.



Άνοιγμα του **Anaconda Prompt**.



Ενεργοποίηση του virtual environment που δημιουργήθηκε από τις οδηγίες εγκατάστασης. Αυτό έχει όνομα **ml-agents**.



Μετάβαση στον φάκελο του project.

Anaconda Prompt (miniconda3)

```
(base) C:\Users\Dimitris>conda activate ml-agents
(ml-agents) C:\Users\Dimitris>cd Desktop
(ml-agents) C:\Users\Dimitris\Desktop>cd "Intelligent Agents"
(ml-agents) C:\Users\Dimitris\Desktop\Intelligent Agents>_
```

Στον φάκελο **config** υπάρχει το αρχείο **FetchBall.yaml**. Σε αυτό υπάρχουν κάποιοι παράμετροι για την εκπαίδευση του μοντέλου. Οι πληροφορίες για τις παραμέτρους εκπαίδευσης βρίσκονται στον σύνδεσμο [https://github.com/Unity-Technologies/ml-agents/blob/release\\_17\\_docs/docs/Training-Configuration-File.md](https://github.com/Unity-Technologies/ml-agents/blob/release_17_docs/docs/Training-Configuration-File.md). Η τροποποίηση του αρχείου είναι προαιρετική. Στην παρακάτω εικόνα φαίνονται τα περιεχόμενα του αρχείου, από το τελευταίο μοντέλο που εκπαιδεύτηκε (**FetchBall5**).

```
behaviors:
  FetchBall:
    trainer_type: ppo
    hyperparameters:
      batch_size: 256
      buffer_size: 2048
      learning_rate: 3.0e-4
      beta: 5.0e-4
      epsilon: 0.2
      lambda: 0.99
      num_epoch: 3
      learning_rate_schedule: linear
    network_settings:
      normalize: false
      hidden_units: 128
      num_layers: 4
    reward_signals:
      extrinsic:
        gamma: 0.99
        strength: 1.0
      gail:
        strength: 0.01
        demo_path: Demos/FetchBallDemo.demo
    behavioral_cloning:
      strength: 0.25
      demo_path: Demos/FetchBallDemo.demo
    max_steps: 500000
    time_horizon: 64
    summary_freq: 10000
```

Εκκίνηση εκπαίδευσης, εκτελώντας την εντολή

**mlagents-learn config\FetchBall.yaml --run-id=<id>**

Στη θέση του <id> μπαίνει το επιθυμητό id του μοντέλου. Στο συγκεκριμένο παράδειγμα το <id> είναι το **FetchBallDemoModel**.

```
Anaconda Prompt (miniconda3)

(base) C:\Users\Dimitris>conda activate ml-agents
(ml-agents) C:\Users\Dimitris>cd Desktop
(ml-agents) C:\Users\Dimitris\Desktop>cd "Intelligent Agents"
(ml-agents) C:\Users\Dimitris\Desktop\Intelligent Agents>mlagents-learn config\FetchBall.yaml --run-id=FetchBallDemoModel
```

Όταν εκτελεστεί η εντολή, εμφανίζεται το εξής.

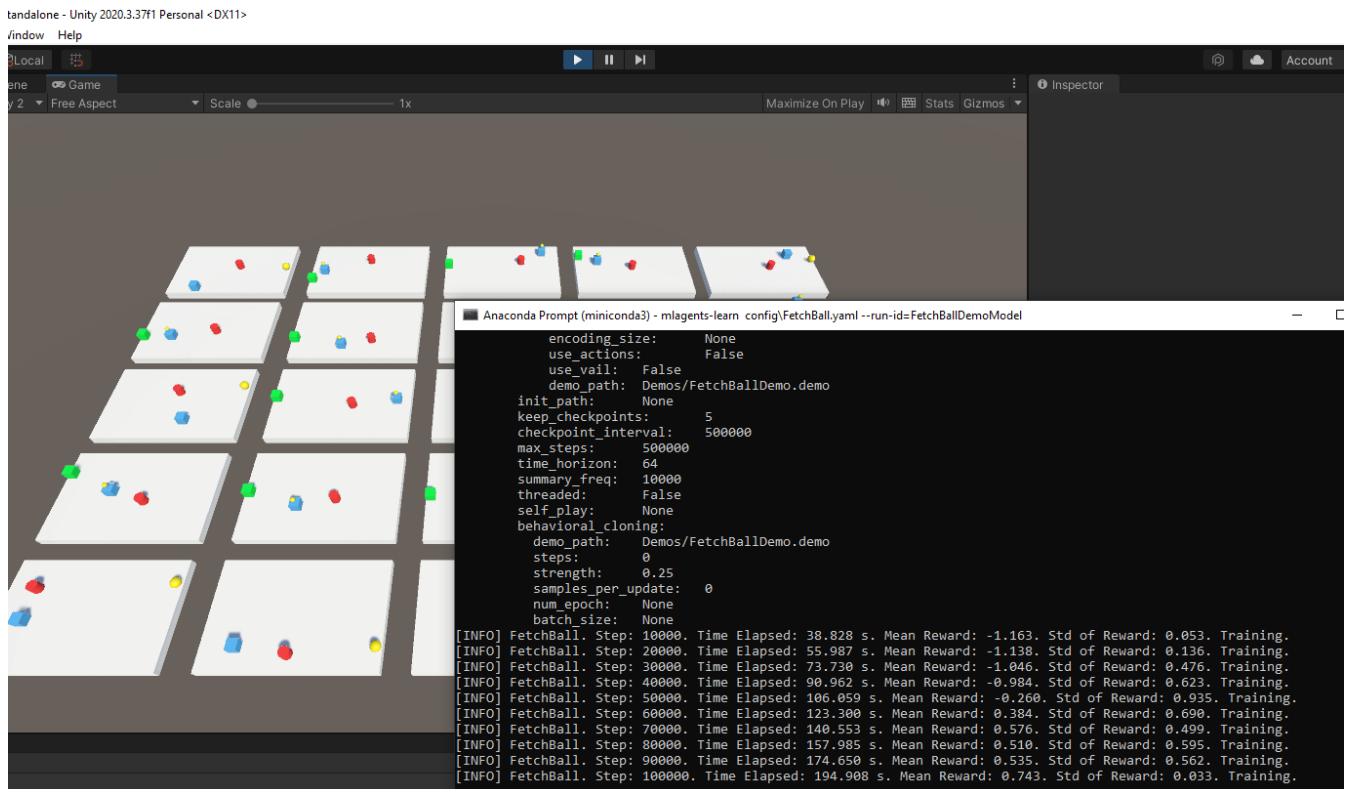
```
Anaconda Prompt (miniconda3) - mlagents-learn config\FetchBall.yaml --run-id=FetchBallDemoModel

n: Found no NVIDIA driver on your system. Please check that you have an NVIDIA GPU and installed a driver from http://www.nvidia.com/Download/index.aspx (Triggered internally at ..\c10\cuda\CUDAFunctions.cpp:100.)
return torch._C._cuda_getDeviceCount() > 0

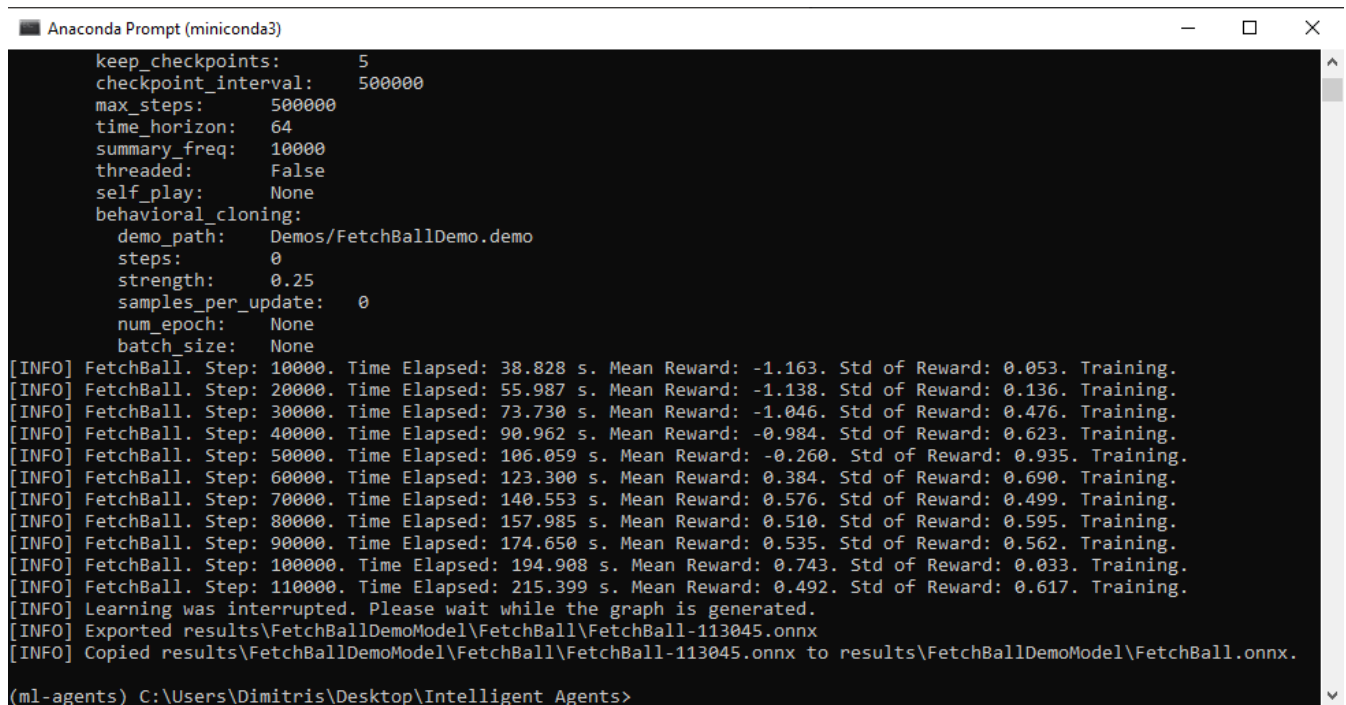
<img alt="Unity logo" data-bbox="135 425 630 545"/>

Version information:
ml-agents: 0.26.0,
ml-agents-envs: 0.26.0,
Communicator API: 1.5.0,
PyTorch: 1.7.1+cu110
C:\Users\Dimitris\miniconda3\envs\ml-agents\lib\site-packages\torch\cuda\__init__.py:52: UserWarning: CUDA initialization
n: Found no NVIDIA driver on your system. Please check that you have an NVIDIA GPU and installed a driver from http://www.nvidia.com/Download/index.aspx (Triggered internally at ..\c10\cuda\CUDAFunctions.cpp:100.)
return torch._C._cuda_getDeviceCount() > 0
[INFO] Listening on port 5004. Start training by pressing the Play button in the Unity Editor.
```

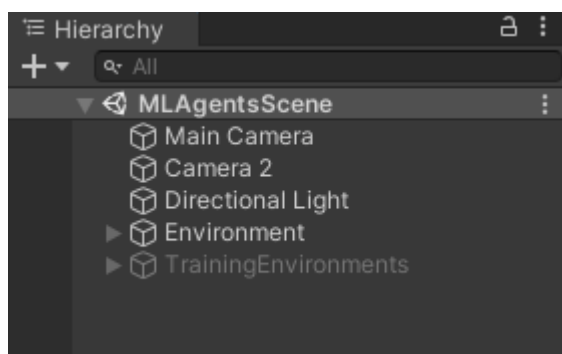
Πάτημα του κουμπιού **Play** στον editor της Unity. Εδώ φαίνεται ο editor με τα περιβάλλοντα εκπαίδευσης και το command line με κάποια στατιστικά, ενώ γίνεται η εκπαίδευση.



Πατώντας ξανά το κουμπί **Play**, σταματάει η εκπαίδευση και αποθηκεύεται το αρχείο του μοντέλου που εκπαιδεύτηκε.



Ενεργοποίηση του **Environment** και απενεργοποίηση του **TrainingEnvironments**.



Στον φάκελο του project, στη διαδρομή **results\<id>**, αντιγραφή του αρχείου **FetchBall.onnx**.

Αυτός ο υπολογιστής > Επιφάνεια εργασίας > Intelligent Agents > results > FetchBallDemoModel >

Όνομα	Ημερομηνία τροποποι...	Τύπος	Μέγεθος
FetchBall	2/8/2022 9:18 μμ	Φάκελος αρχείων	
run_logs	2/8/2022 9:18 μμ	Φάκελος αρχείων	
configuration.yaml	2/8/2022 9:18 μμ	Yaml Source File	3 KB
FetchBall.onnx	2/8/2022 9:18 μμ	Αρχείο ONNX	208 KB

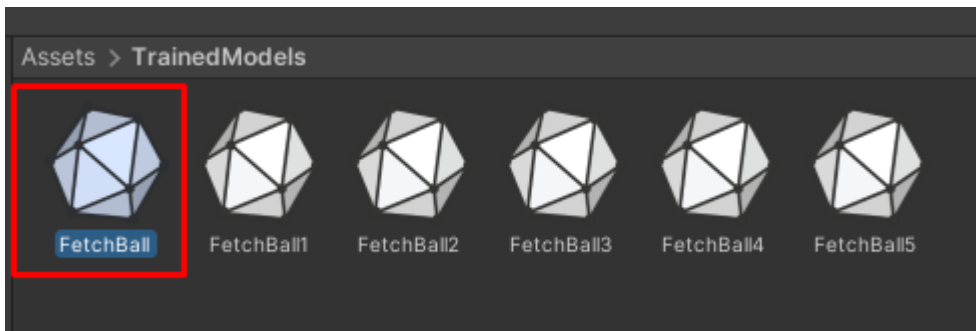
Επικόλληση του αρχείου στον φάκελο **TrainedModels**.

ός ο υπολογιστής > Επιφάνεια εργασίας > Intelligent Agents > Assets > TrainedModels

Όνομα	Ημερομηνία τροποποι...	Τύπος	Μέγεθος
FetchBall.onnx	2/8/2022 9:18 μμ	Αρχείο ONNX	208 KB
FetchBall1.onnx	1/8/2022 5:09 μμ	Αρχείο ONNX	143 KB
FetchBall1.onnx.meta	1/8/2022 5:09 μμ	Αρχείο META	1 KB
FetchBall2.onnx	1/8/2022 5:13 μμ	Αρχείο ONNX	143 KB



Πλέον υπάρχει και το καινούργιο μοντέλο που εκπαιδεύτηκε μαζί με τα άλλα. Μπορεί να χρησιμοποιηθεί, αναθέτοντάς το στον πράκτορα, όπως περιγράφεται στις οδηγίες εκτέλεσης εφαρμογής.



## Συμπεράσματα και αποτελέσματα

Το πακέτο ML-Agents της Unity είναι ένα πολύ ισχυρό εργαλείο με πολλές δυνατότητες, με το οποίο μπορούν να εκπαιδευτούν πολλά είδη πρακτόρων. Προσφέρει ένα φιλικό προς τον χρήστη interface, ενώ παράλληλα έχει πάρα πολλές παραμετροποιήσεις. Μόλις μάθει κάποιος πως λειτουργούν τα βασικά στοιχεία του πακέτου, η επέκταση σε εκπαίδευση πιο περίπλοκων μοντέλων είναι αρκετά απλή.

Από τα μοντέλα που εκπαιδεύτηκαν, την καλύτερη επίδοση έχει το μοντέλο **FetchBall3**. Δεν είναι τέλειο, εφόσον κάποιες φορές ο πράκτορας χάνει, όμως δεδομένου ότι δεν έγινε πολύ μεγάλος πειραματισμός των παραμέτρων του μοντέλου, το αποτέλεσμα είναι αρκετά ικανοποιητικό. Αυτό το project θα μπορούσε να βελτιωθεί με πειραματισμό παραμέτρων του μοντέλου, οι οποίες είναι πάρα πολλές. Ένας άλλος τρόπος βελτίωσης μπορεί να ήταν το να γίνει εκπαίδευση με ακόμα περισσότερο τυχαίες θέσεις των οντοτήτων του περιβάλλοντος, καθώς και η εύρεση καλύτερων χαρακτηριστικών εκπαίδευσης, στα οποία θεωρούμε όμως πως ήδη δόθηκε βάση.