Predator-prey model in Unity using ml-agents toolkit

This repository contains the agent-based model outlined in the "Investigating the emergence of complex behaviours in an agent-based model using reinforcement learning" paper.

Requirements

- Python programming language: https://www.python.org/downloads/
- Unity version **2018.4.15** can be downloaded via the Unity Hub from here: https://unity3d.com/get-unity/download
- Install **ml-agents** in python: https://github.com/Unity-Technologies/ml-agents

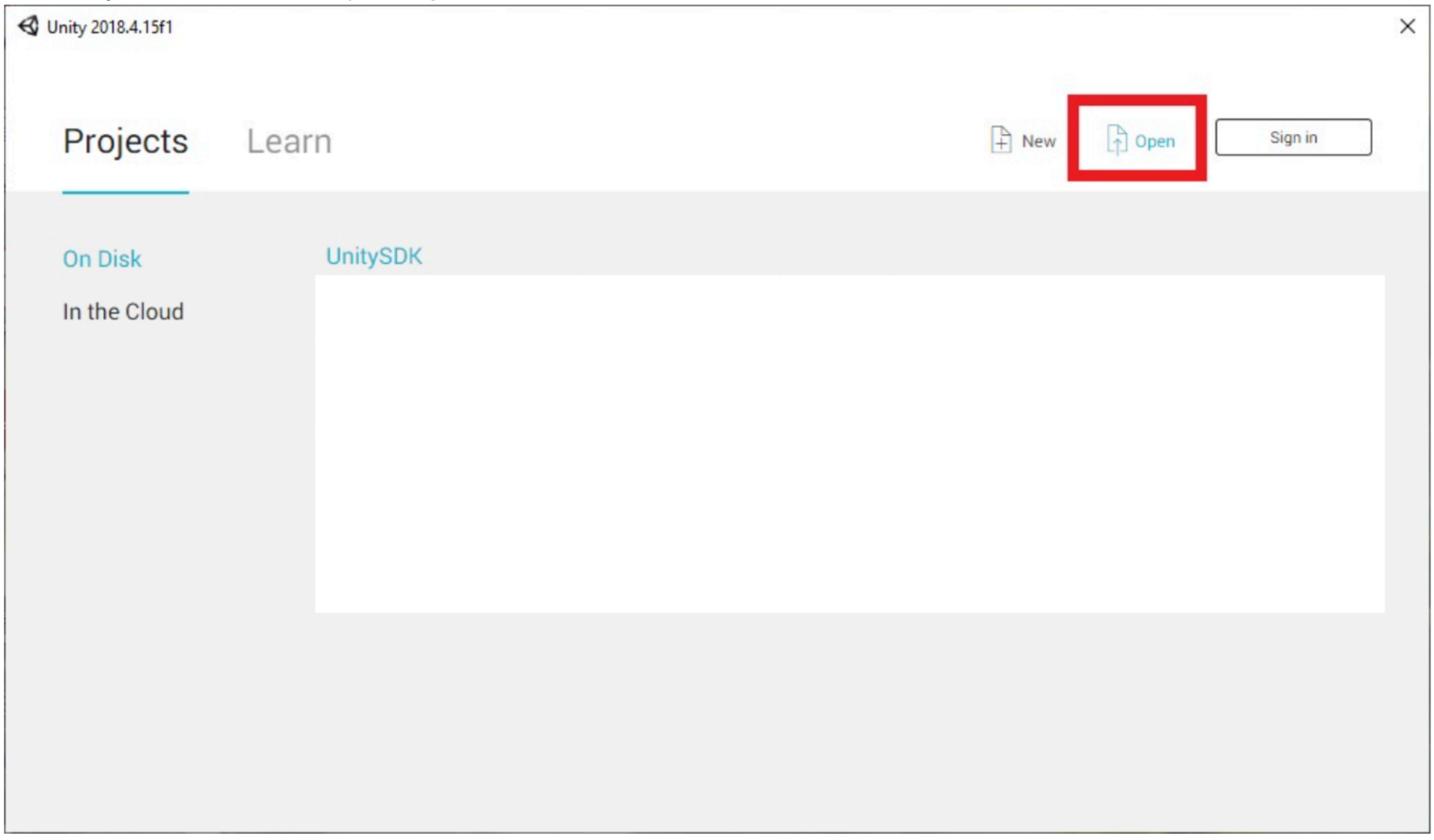
How to access the predator-prey model

ml-agents-master-2

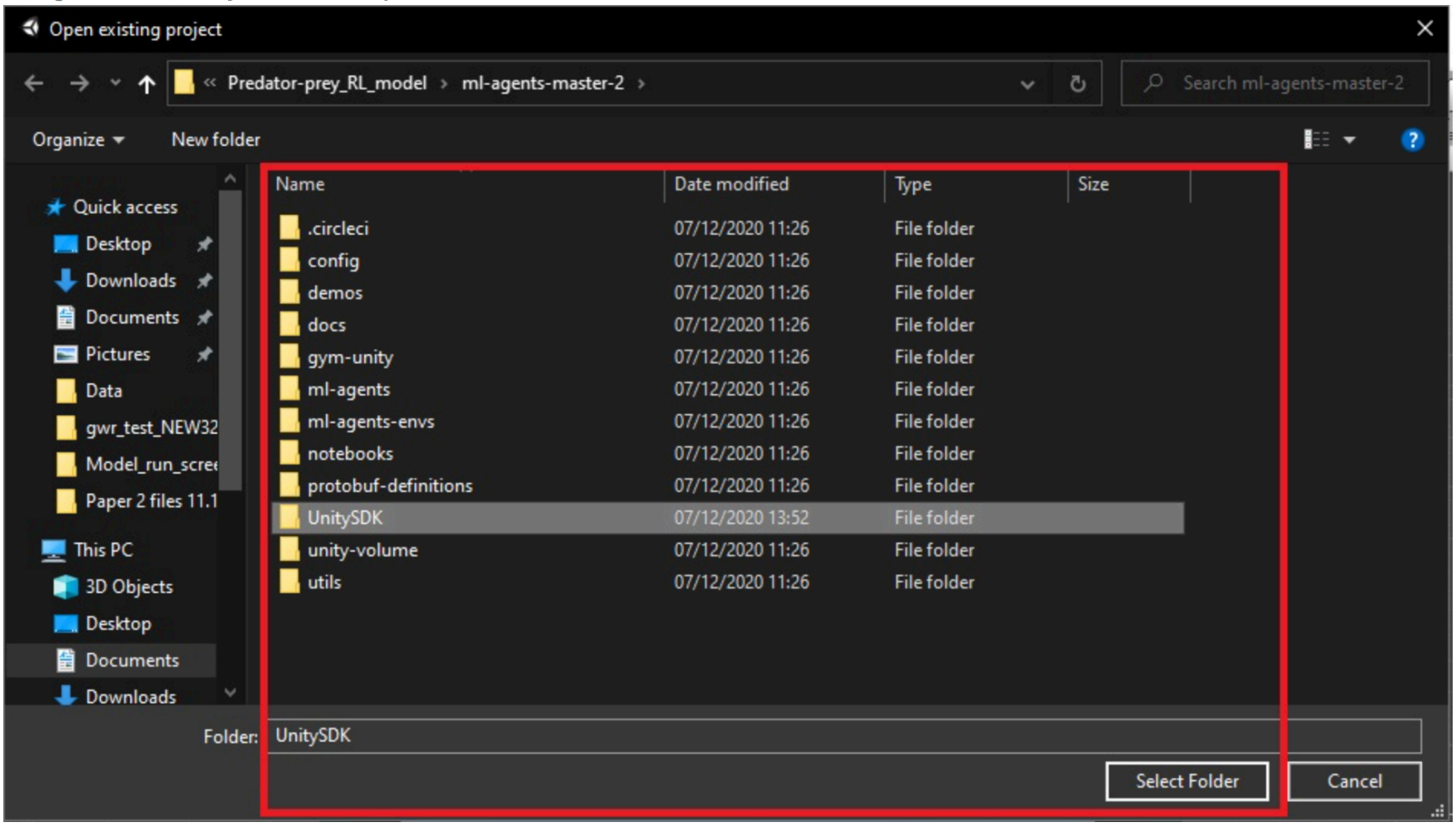
- UnitySDK/
  - Assets/
    - predator-prey/
      - Brains/
        - neural_network_1.nn (This is the brain that is referred to as Scenario_1 in the literature)
        - neural_network_2.nn (This is the brain that is referred to as Scenario_2 in the literature)
        - neural_network_3.nn (This is the brain that is referred to as Scenario_3 in the literature)
      - Editor/
      - Materials/
      - Prefabs/
        - PreyAgent.prefab
        - Environment.prefab
        - Predator.prefab
        - badPoint.prefab
        - goodPoint.prefab
      - Scenes/
        - predator_prey_scene.unity **(Double click this file to launch the initial configuration of the model).**
      - Scripts/
        - AIPredator.cs
        - Prey.cs
        - PPEnvironment.cs
        - PointLogic.cs
        - PredatorPreyAcademy.cs

Running the model

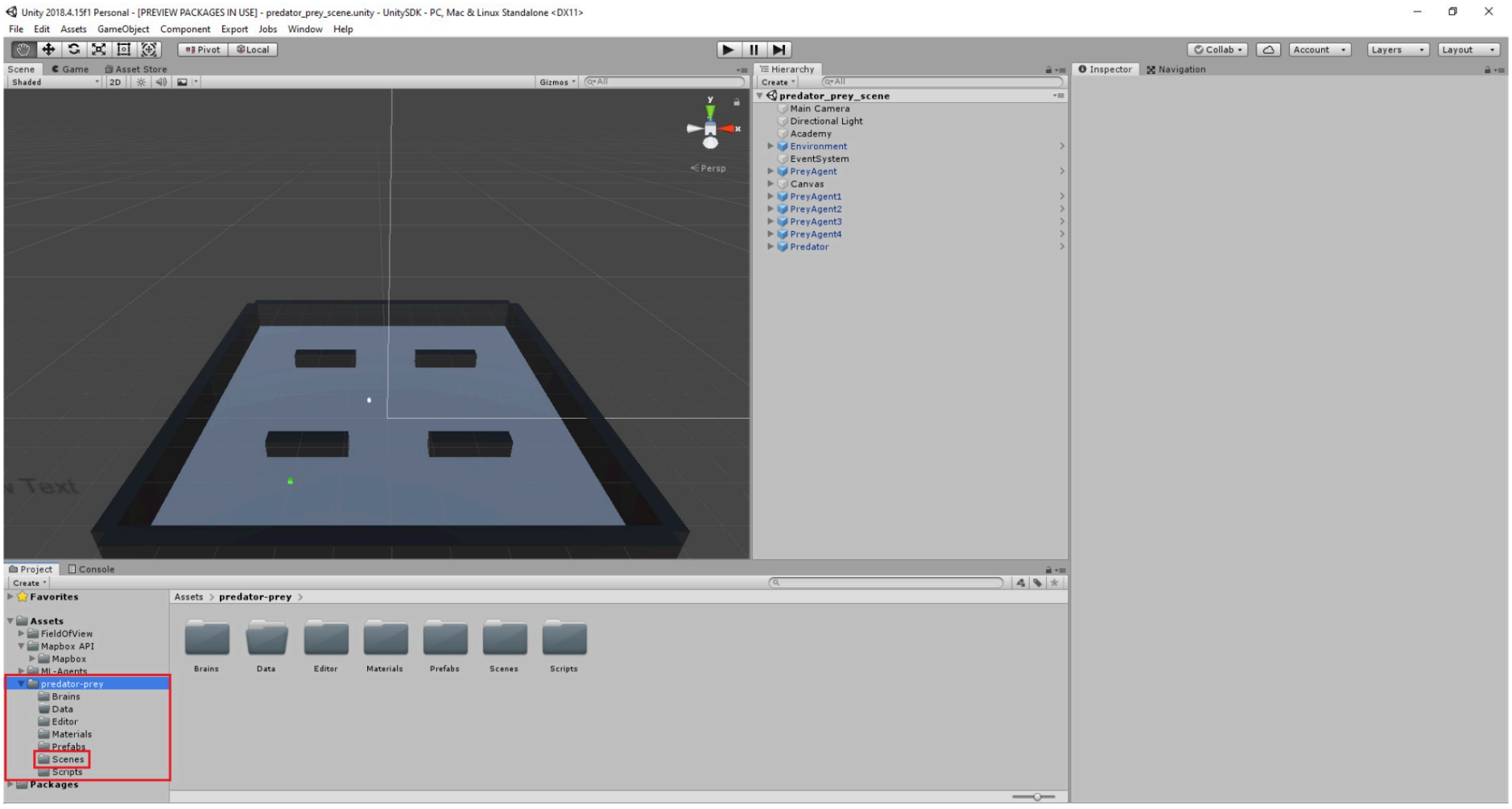
Once Unity has been downloaded, click **Open**



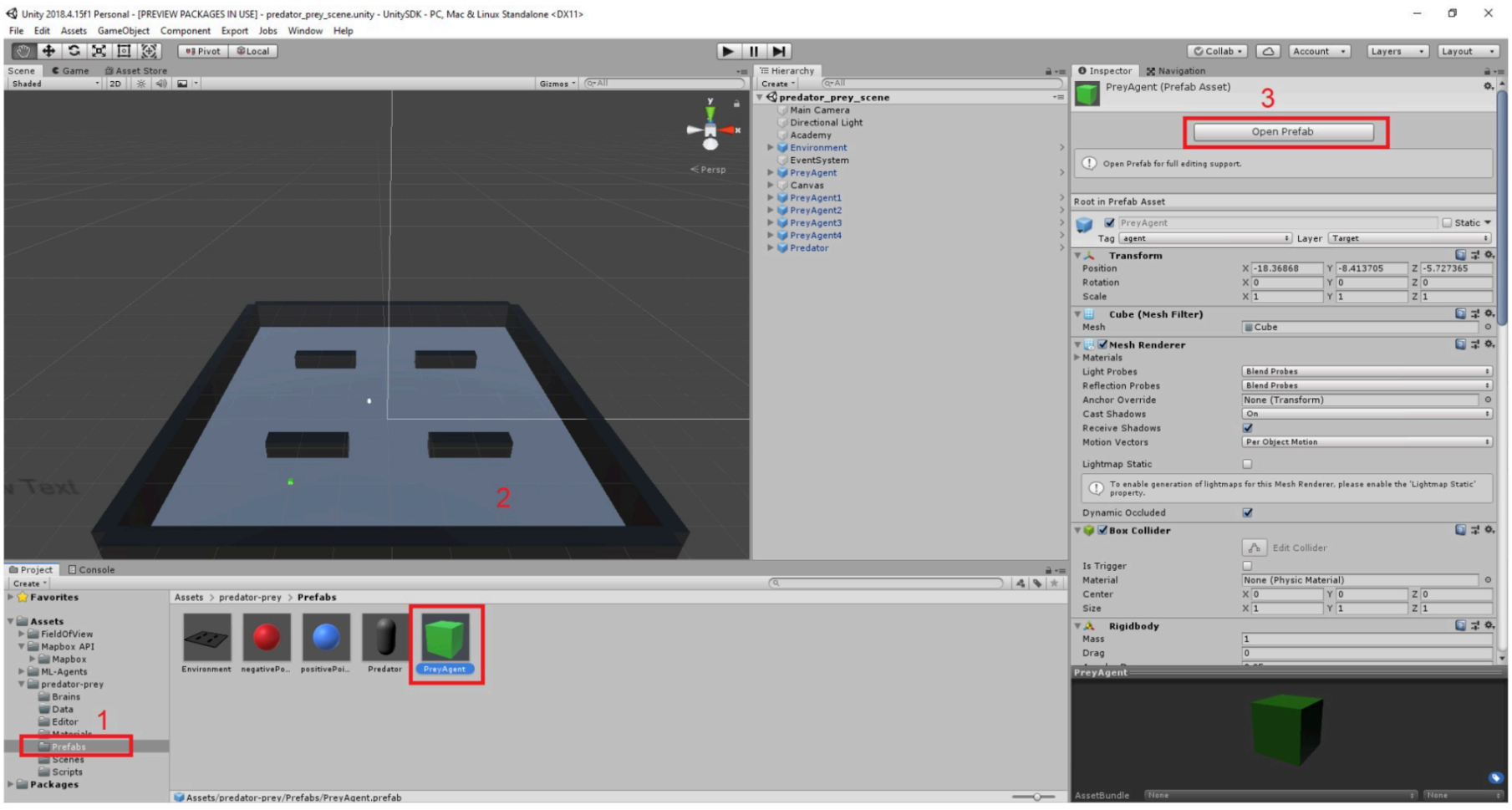
Navigate to the **UnitySDK** directory and click **Select folder**



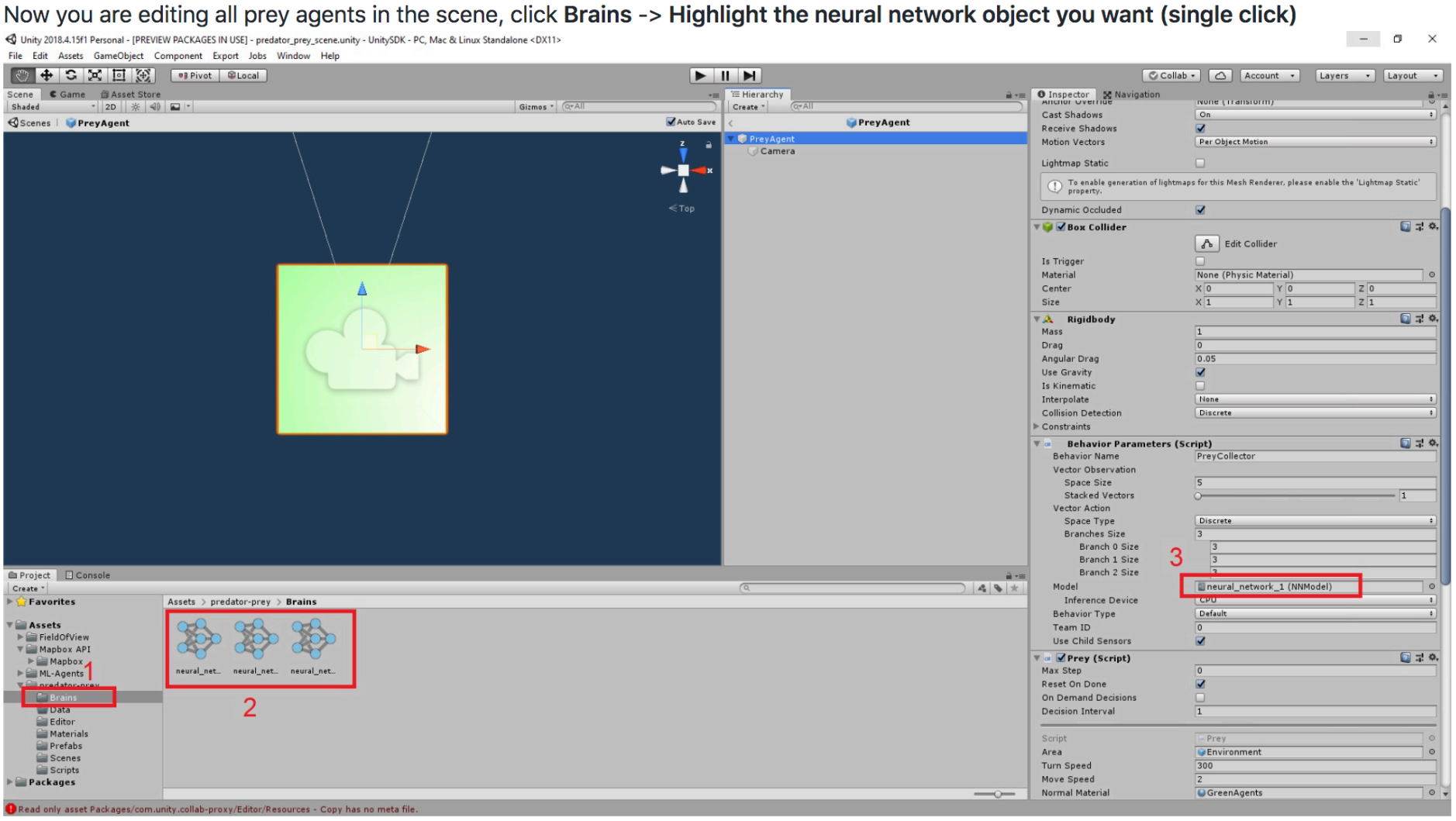
If the **predator_prey_scene** is not automatically opened, click **predatory-prey** -> **Scenes** -> **predator_prey_scene** to open it.



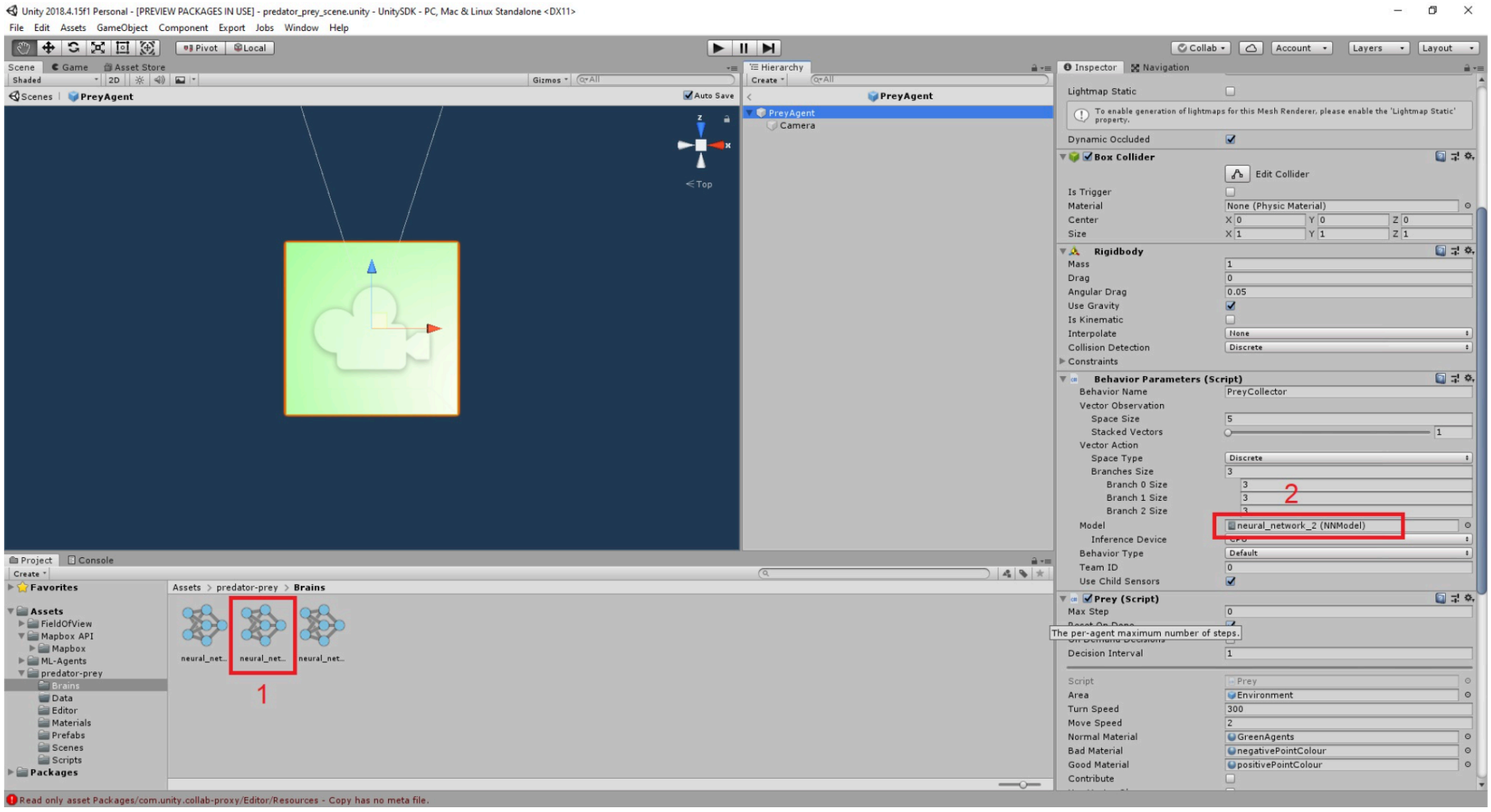
Initially neural network/brain one is attached to prey agents, however, if you wish to swap the brain object. Click **Prefab** -> **PreyAgent** -> **Open Prefab**



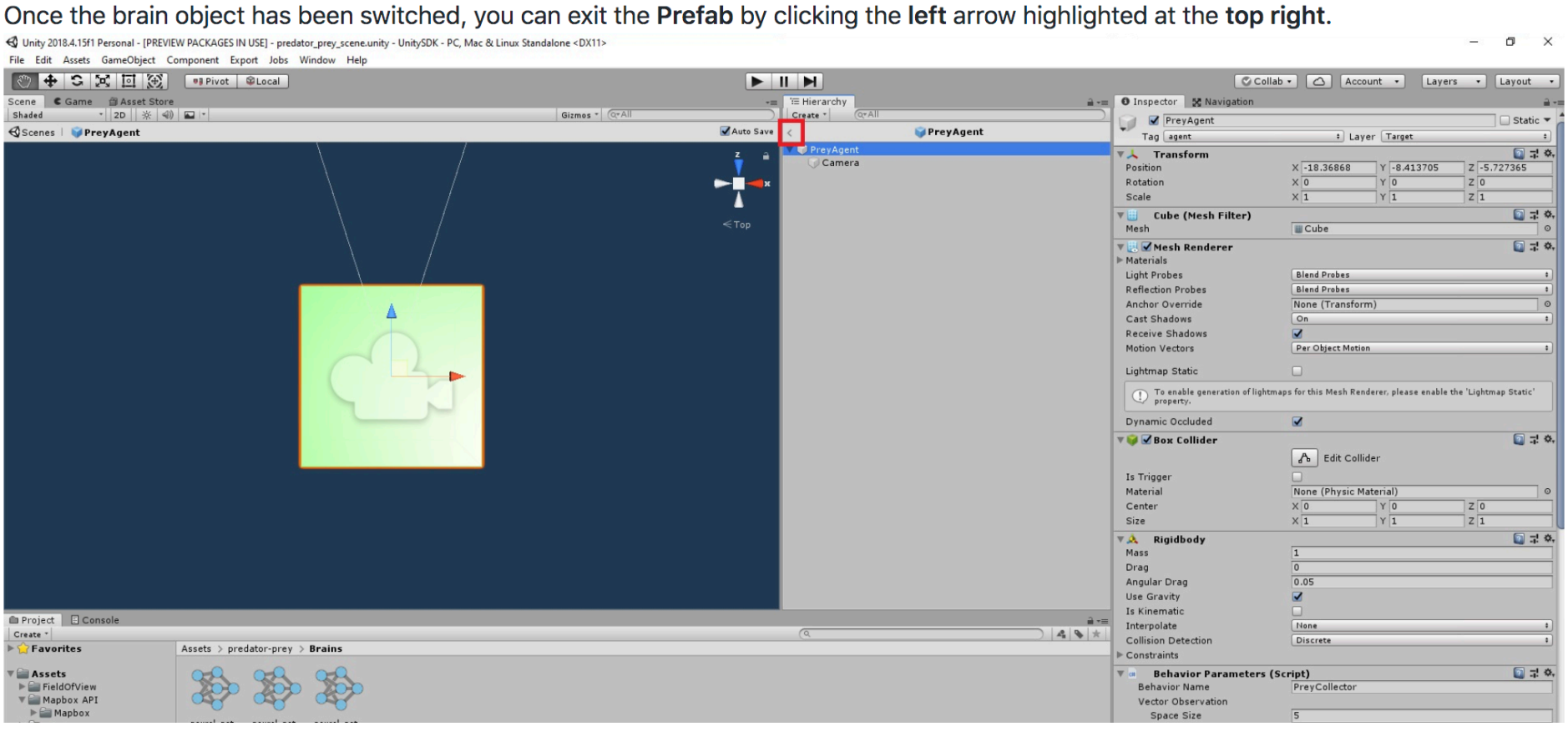
Now you are editing all prey agents in the scene, click **Brains -> Highlight the neural network object you want (single click)**



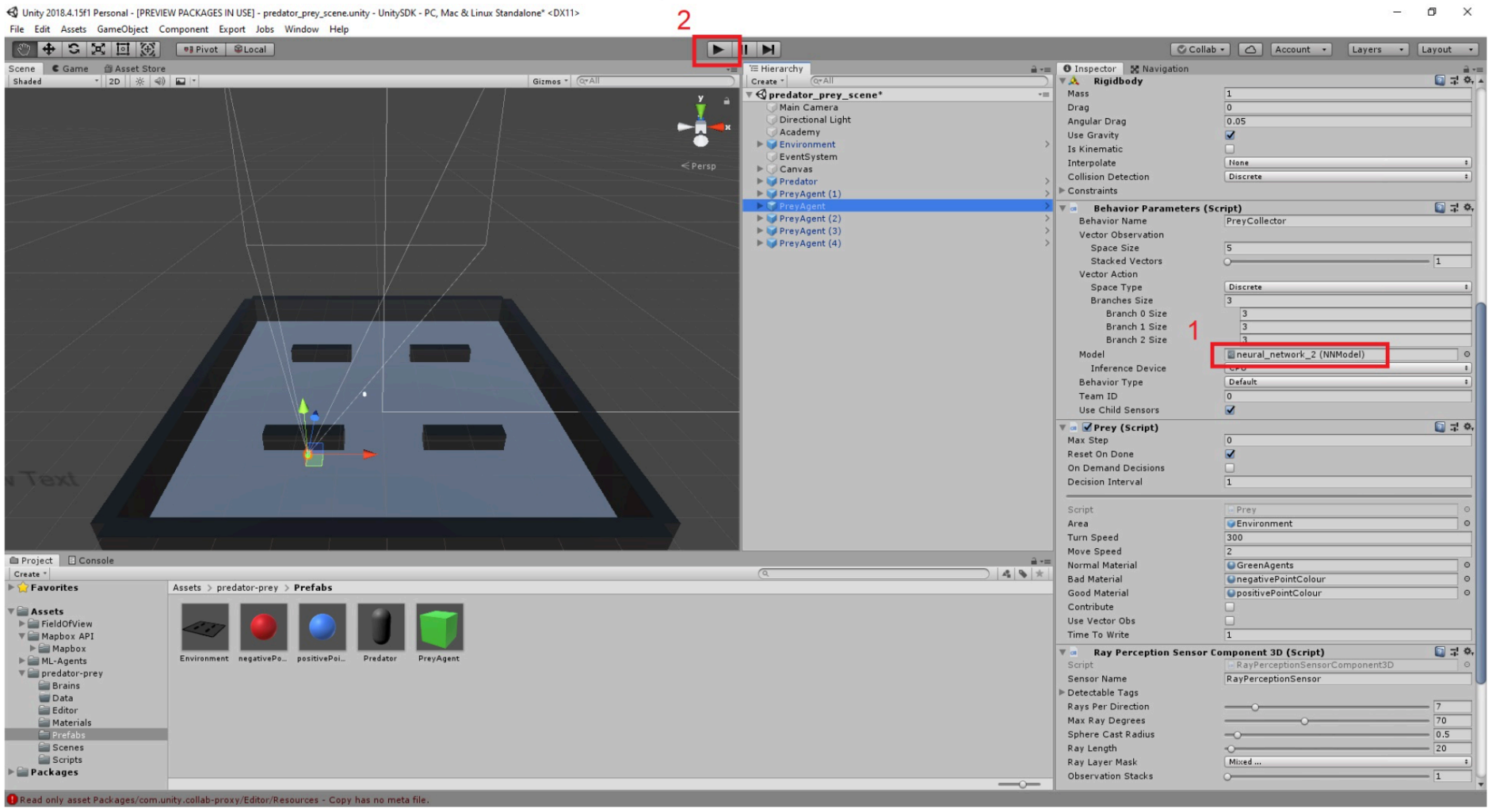
For this example, we highligh the second **neural_network 2** then drag and drop it into the **Behaviour Parameters script, model field.**



Once the brain object has been switched, you can exit the **Prefab** by clicking the **left** arrow highlighted at the **top right.**



As you return to the scene view, if you click on a **PreyAgent** object in the scene window, you can see that it's model parameter has changed to the neural network you attached. Now you can click the **Play** button to run the simulation.



Access dummy data produced by model run

Dummy data is produced by the model each time you **Play** it, this contains raw model data that can be used for analysis. A Data folder was created and is located at: **..\UnitySDK\Assets\predator-prey\Data**

- To make sure the data is exported to this or any other directory, you must edit the **Prey.cs** and **AIPredator.cs** script.
- These script files can be found in **..\UnitySDK\Assets\predator-prey\Scripts\AIPredator.cs** and **..\UnitySDK\Assets\predator-prey\Scripts\Prey.cs**.
- You can open these script files in any IDE.
- For the **Prey.cs** change the directory **../Data/prey_data.csv** at **Line 114** to a location of your choosing.

```
osition.x, this.transform.position.z, this.seenByPredator, this.wallTouch, myAcademy.totalScore, "../Data/prey_data.csv");
```

- For the **AIPredator.cs** change the directory **../Data/predator_data.csv** at **Line 121** to a location of your choosing.

```
this.transform.position.x, this.transform.position.z, dstToTarget, viewCastAngle, wallTouch, "../Data/predator_data.csv");
```

Data and experiment results

Access raw data analysed in the paper for experiments one and two in the **Synthetic-data-from-ABM** folder:

Synthetic-data-from-ABM/

- Experiment_1_model_condition_1-results
- Experiment_1&2_model_condition_2-results
- Experiment_2_model_condition_1-results
- Experiment_2_model_condition_3-results
- Statistical_significance_test_all_experiments.ipynb (Jupyter Notebook for analysis of results.)
- Stats Test Outputs_Filled.xlsx (results tabulated in an excel spreadsheet.)

Notes

- The Agent ID is consistent throughout each model-run. Every agent that is initialised will have a unique ID that is assigned to the object and this will remain the same for each agent throughout the experiments.