**足球遊戲**

**API文件**

# 修改紀錄

|  |  |
| --- | --- |
| **修改項目** | **修改日期** |
| 查詢Q表、輸出Q表、產生模型檔、設定Q表、更新Q表  ML agent、初始unity、執行action | 2022/12/01~ 2022/12/07 |
| choose\_action、q\_table\_save、q\_table\_read、learn  UnityToGymWrapper、gym.reset、gym.step | 2023/01/05~  2023/01/11 |

紅字:修改

藍字:新增

綠字:刪除

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 查詢Q表 | | 修改為🡺 | choose\_action | |
| **輸入** | state([integer,integer])：當前狀態 | **輸入** | state([float,float])：當前狀態 |
| **輸出** | action(integer)：前、後、左、右擇一輸出 | **輸出** | action(integer)：前、後、左、右擇一輸出 |
| **參數** | 無 | **參數** | epsilon:貪婪度,選擇非最佳選的機率 |
| **方法** | 輸入當前狀態後，查詢Q表便會根據Q表輸出對應動作 | **方法** | 輸入當前狀態後，查詢Q表便會根據Q表輸出對應動作 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 產生模型檔 | | 修改為🡺 | q\_table\_save | |
| **輸入** | Q\_table([state,action])：輸入Q表陣列 | **輸入** | Q\_table([state,action])：輸入Q表陣列 |
| **輸出** | model(.csv)：將**Q表**產生成.csv模型檔 | **輸出** | model(.csv)：將**Q表**產生成.csv模型檔 |
| **參數** | path(string)：輸入模型檔的路徑 | **參數** | path(string)：輸入模型檔的路徑 |
| **方法** | 無 | **方法** | 無 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 設定Q表 | | 修改為🡺 | q\_table\_read | |
| **輸入** | state([integer,integer])：當前狀態 | **輸入** | state([integer,integer])：當前狀態 |
| **輸出** | 無 | **輸出** | 無 |
| **參數** | path(string)：輸入模型檔的路徑 | **參數** | path(string)：輸入模型檔的路徑 |
| **方法** | 先將模型檔內的所有資料加入Q表中，接著判斷state存不存在於Q表，若無，則在Q表新增一個空間 | **方法** | 先將模型檔內的所有資料加入Q表中 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 更新Q表 | | 修改為🡺 | learn | |
| **輸入** | state([integer,integer])：當前狀態  state’([integer,integer])：下個狀態  action(integer)：前、後、左、右  reward(integer)：根據我們定義的獎勵機制來給分 | **輸入** | state([float, float])：當前狀態  state’([float, float])：下個狀態  action(string)：前、後、左、右  reward(integer)：根據我們定義的獎勵機制來給分 |
| **輸出** | q\_update(float)：根據輸入計算出Q值 | **輸出** | q\_update(float)：根據輸入計算出Q值 |
| **參數** | learning\_rate (float) :學習速率  discount\_factor (float) : 折扣因子  e\_greedy (float): 貪婪度 | **參數** | learning\_rate (float) :學習速率  discount\_factor (float) : 折扣因子 |
| **方法** | 根據我們的動作所得到的reward算出一個Q值來更新Q表 | **方法** | 根據我們的動作所得到的reward算出一個Q值來更新Q表 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ML agent | | 修改為🡺 | UnityToGymWrapper | |
| **輸入** | unity\_env：ML\_agent產生之unity環境 | **輸入** | unity\_env：ML\_agent產生之unity環境 |
| **輸出** | gym\_env：將unity環境轉換為gym環境 | **輸出** | gym\_env：將unity環境轉換為gym環境 |
| **參數** | 無 | **參數** | 無 |
| **方法** | 將unity環境改為gym環境來溝通 | **方法** | 將unity環境改為gym環境來溝通 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 初始unity | | 修改為🡺 | gym.reset | |
| **輸入** | 無 | **輸入** | 無 |
| **輸出** | state([integer,integer])：當前狀態 | **輸出** | state([float, float])：當前狀態 |
| **參數** | 無 | **參數** | 無 |
| **方法** | Unity遊戲初始化 | **方法** | Unity遊戲初始化 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 執行action | | 修改為🡺 | gym.step | |
| **輸入** | action(,integer)：前、後、左、右 | **輸入** | action([integer,integer,integer])：  前後、左右、旋轉 |
| **輸出** | state([integer,integer])：當前狀態  done(bool)：遊戲是否已結束  reward(integer)：根據我們定義的獎勵機制來給分 | **輸出** | state([float, float])：當前狀態  done(bool)：遊戲是否已結束  reward(integer)：根據我們定義的獎勵機制來給分 |
| **參數** | 無 | **參數** | 無 |
| **方法** | 輸入動作並從遊戲中得到結果以及回饋分數，並將輸入動作呈現於遊戲中 | **方法** | 輸入動作並從遊戲中得到結果以及回饋分數，並將輸入動作呈現於遊戲中 |

* 專案 : q\_learning.py

|  |  |
| --- | --- |
| choose\_action | |
| **輸入** | state([float,float])：當前狀態 |
| **輸出** | action(integer)：前、後、左、右擇一輸出 |
| **參數** | epsilon:貪婪度,選擇非最佳選的機率 |
| **方法** | 輸入當前狀態後，**查詢Q表**便會根據Q表輸出對應動作 |

|  |  |
| --- | --- |
| q\_table\_save | |
| **輸入** | Q\_table([state,action])：輸入Q表陣列 |
| **輸出** | model(.csv)：將**Q表**產生成.csv模型檔 |
| **參數** | path(string)：輸入模型檔的路徑 |
| **方法** | 無 |

|  |  |
| --- | --- |
| q\_table\_read | |
| **輸入** | state([integer,integer])：當前狀態 |
| **輸出** | 無 |
| **參數** | path(string)：輸入模型檔的路徑 |
| **方法** | 先將模型檔內的所有資料加入Q表中 |

|  |  |
| --- | --- |
| learn | |
| **輸入** | state([float, float])：當前狀態  state’([float, float])：下個狀態  action(string)：前、後、左、右  reward(integer)：根據我們定義的獎勵機制來給分 |
| **輸出** | q\_update(float)：根據輸入計算出Q值 |
| **參數** | learning\_rate (float) :學習速率  discount\_factor (float) : 折扣因子 |
| **方法** | 根據我們的動作所得到的reward算出一個Q值來更新Q表 |

* 專案 : main.py

|  |  |
| --- | --- |
| UnityToGymWrapper | |
| **輸入** | unity\_env：ML\_agent產生之unity環境 |
| **輸出** | gym\_env：將unity環境轉換為gym環境 |
| **參數** | 無 |
| **方法** | 將unity環境改為gym環境來溝通 |

|  |  |
| --- | --- |
| gym.reset | |
| **輸入** | 無 |
| **輸出** | state([float, float])：當前狀態 |
| **參數** | 無 |
| **方法** | Unity遊戲初始化 |

|  |  |
| --- | --- |
| gym.step | |
| **輸入** | action([integer,integer,integer])：  前後、左右、旋轉 |
| **輸出** | state([float, float])：當前狀態  done(bool)：遊戲是否已結束  reward(integer)：根據我們定義的獎勵機制來給分 |
| **參數** | 無 |
| **方法** | 輸入動作並從遊戲中得到結果以及回饋分數，並將輸入動作呈現於遊戲中 |