

Proly

本次沒有解題思路

參考rule(抄了0分)_([https://github.com/PAIA-Playful-AI-](https://github.com/PAIA-Playful-AI-Arena/MLGame3D/blob/develop/examples/proly_mlplay.py)

[Arena/MLGame3D/blob/develop/examples/proly_mlplay.py](https://github.com/PAIA-Playful-AI-Arena/MLGame3D/blob/develop/examples/proly_mlplay.py)).

PPO參考模板(會的話可以抄)_(<https://gist.github.com/seanpai96/8f2ccad60a7e7a8ad87e8949f522c6d1>).

README(proly)_(<https://github.com/PAIA-Playful-AI-Arena/Proly/blob/main/README.md>).

下載遊戲(proly)_(<https://github.com/PAIA-Playful-AI-Arena/Proly/releases/tag/0.4.0>).

README(mlgame3d)_(<https://github.com/PAIA-Playful-AI-Arena/MLGame3D/blob/develop/README.md>).

繳交期限: 2025/06/01 中午12:00:00

第一次比賽時間: 2025/06/05 上課時間

助教信箱: NE6134045@gs.ncku.edu.tw

如果有問題，請先問問ChatGPT，如果還是看不懂再聯繫助教。如果確定教材或公告有錯誤，請直接聯繫助教。

評分標準&賽制

這次的作業使用比賽名次進行評分。

分數分配如下:

名次	分數
第一名	100
第二名	96
4強	92
16強	88
32強	84
未進入前32強	80
未交作業/作業未按規定繳交/程式跑不起來	0

助教會視情況加扣分。

賽制如下:

1. 一開始會四人一組，預計分為15組進行五場比賽，各組取前二名。

若沒有碰到第一個checkpoint，即使在前兩名還是會進入敗部。

(有摸到第一個checkpoint的才會算是前二名)

2. 剩下敗部組會3~4人一組進行單次比賽的敗部復活，根據各自抵達的checkpoint數量及完成時間進行排名，取足夠多名次(checkpoint>0, 以checkpoint->時間排列 取滿32人)進入32強賽。

若不足32強將有隨機的組別以三位選手進行比賽。

3. 32人會進入4人一組進行五場比賽，取出前兩名，進入16強。

4. 16強會進入4人一組進行五場比賽，取出第一名，取出前4強。

5. 4強為冠軍戰，得到1, 2名，3, 4名以四強選手計分。

若有不足的規則以助教為準。

比賽將隨機生成8格目標點(checkpoint)。

第一場比賽將不會有任何道具，次週的第二場比賽會啟用部分道具，屆時會再公布。

本次不接受補交。

每位選手都必須使用機器學習來訓練自己的proly，並使用自己的模型出賽。

模型並不限制監督式或RL。

然後不要讓你比賽時的程式進行與比賽無關的動作（如搜集資料,訓練等）。遊戲會跑很久。

若會嚴重影響比賽時長，助教會修改你的程式碼，停用不相關的功能。

環境設置

請在Anaconda Prompt中運行

請確定終端機前面有 (base) 或 (paia) 的提示

```
#建立新環境
conda create -n proly python=3.11
conda activate proly

#安裝套件
pip install mlgame3d scikit-learn
pip install numpy==1.26.4
```

本次的作業將使用新的環境。

由於會使用助教的電腦進行比賽，所以若同學使用不在本環境之中的套件，請來信給助教。若出現套件版本衝突，助教將寄信通知受到影響的同學進行溝通，請再留意信箱。

助教的電腦並沒有CUDA(是mac)，請確保你的模型跟程式在cpu下可以正常運作。

載入遊戲指令

請在Anaconda Prompt中運行

請確定終端機前面有 (base) 或 (...) 的提示

```
#進入paia的anaconda環境
conda activate proly
#進入上次作業的MLGame資料夾（請記得這個資料夾，你的程式碼都會擺在這裡）
#cd any/place/you/want

cd MLGame

###
# 解壓縮moodle上的附件到你的MLGame資料夾
# 從Github下載屬於你系統的Proly，擺進去proly/資料夾
###

#進入資料夾並以VSCode開啟
cd proly

code .
```

透過上面的指令，我們會將這次的作業clone至電腦中，並且以VSCode開啟。

遊戲指令

請在上面教學裡clone下來的 `proly` 資料夾中執行指令碼，否則遊戲將不會啟動


要在(proly)的環境中才可以執行喔~

另外，參數請以上面的README為主~

- 遊戲指令


- windows

```
python -m mlgame3d -f 60 -i ./ml/ml_play_template.py -i ./ml/ml_play_template.py ./
```



- mac

```
python -m mlgame3d -f 60 -i ./ml/ml_play_template.py -i ./ml/ml_play_template.py ./
```



- 參數說明

- `-f` 為指定遊戲幀數，請注意不要快到電腦跟不上 (optional, default=60)
 - `-i` 指定MLGame class所在的檔案，遊戲將基於這個檔案運行。

第一個帶入的檔案為玩家1，第二個帶入的檔案為玩家2，以此類推，最多四個玩家，若未帶入則該玩家會站在原地

- `-ng` 會不渲染畫面。
 - `-e` 決定比賽的場數。
 - `-gp` 是要帶給遊戲內部的參數，本遊戲的部分參數如下：
 - `checkpoint` 決定checkpoint的數量
 - `items` 決定產生幾種道具，以itemID,itemID2為填入方式。填入0則不會有道具。

請把Unity遊戲執行檔放於最後一個參數，更多參數請參考頂部的兩份README。

遊戲操作說明

這個遊戲會出現很多旗子(也就是Checkpoint)，照著順序走完成路線吧！

當出現第一位達到終點的玩家，遊戲就會開始倒數，時間到了之後遊戲就會結算，比出誰比較厲害！

要注意地板上的泥巴，會讓你走路變慢；要注意島嶼的邊界，掉出去會回到上一個checkpoint！

第二次比賽開始會出現不同的道具，透過道具來為自己製造優勢，贏得遊戲吧！

在 `update` 函式中，我們需要回傳兩個陣列製成的tuple，分別為一個連續數值陣列，跟一個離散數值陣列(`continuous_actions`, `discrete_actions`)。

資料型態是 (`List[float]`, `List[int]`)。

- 連續動作是代表他要移動方向的速度(`ax`, `az`)，會被標準化到-1~1之間。
- 離散動作有兩個值：
 - `discrete_actions[0]`：若為1，則切換下一個道具
 - `discrete_actions[1]`：若為1，則使用道具

observation 有什麼資料？

請以遊戲實際輸出為準

```
{
  "target_position": [
    -3.4675967693328857,
    1.3070000410079956,
    -4.0775465965271
  ],
  "agent_position": [
    -1.0,
    4.316281795501709,
    1.0
  ],
  "agent_forward_direction": [
    0.0,
    -1.0
  ],
  "agent_velocity": [
    0.0,
    0.0
  ],
  "agent_health": 5.0,
  "agent_health_normalized": 1.0,
  "last_checkpoint_index": -1.0,
  "current_time": 0.3733333349227905,
  "current_time_normalized": 0.002074074000120163,
  "inventory_item_count": 0.0,
  "inventory_items": [
    {
      "item_id": 0.0
    },
    {
      "item_id": 0.0
    },
    {
      "item_id": 0.0
    }
  ],
  "selected_item_index": -1.0,
  "nearby_items": [
    {
      "relative_position": [
        0.0,
        0.0
      ],
      "item_id": 0.0
    },
    {
      "relative_position": [
        0.0,
        0.0
      ],
      "item_id": 0.0
    },
    {
      "relative_position": [
        0.0,
        0.0
      ]
    }
  ]
}
```

```
    ],
    "item_id": 0.0
  },
  {
    "relative_position": [
      0.0,
      0.0
    ],
    "item_id": 0.0
  },
  {
    "relative_position": [
      0.0,
      0.0
    ],
    "item_id": 0.0
  }
],
"nearby_map_objects": [
  {
    "relative_position": [
      2.9600000381469727,
      -1.5399999618530273
    ],
    "object_type": 1.0
  },
  {
    "relative_position": [
      3.5802001953125,
      -1.1259499788284302
    ],
    "object_type": 1.0
  },
  {
    "relative_position": [
      0.0,
      0.0
    ],
    "object_type": 0.0
  },
  {
    "relative_position": [
      0.0,
      0.0
    ],
    "object_type": 0.0
  },
  {
    "relative_position": [
      0.0,
      0.0
    ],
    "object_type": 0.0
  }
],
"other_players": [
  {
```

```

    "relative_position": [
      1.25,
      0.0,
      0.0
    ],
    "relative_velocity": [
      0.0,
      0.0,
      0.0
    ]
  },
  {
    "relative_position": [
      1.25,
      0.0,
      -0.9000000357627869
    ],
    "relative_velocity": [
      0.0,
      0.0,
      0.0
    ]
  },
  {
    "relative_position": [
      -0.10000002384185791,
      0.0,
      -0.9000000357627869
    ],
    "relative_velocity": [
      0.0,
      0.0,
      0.0
    ]
  }
],
"terrain_grid": [
  [
    {
      "relative_position": [
        -2.0,
        -2.0
      ],
      "terrain_type": 0.0
    },
    {
      "relative_position": [
        -2.0,
        -1.0
      ],
      "terrain_type": 0.0
    },
    {
      "relative_position": [
        -2.0,
        0.0
      ],

```



```

    "terrain_type": 0.0
  },
  {
    "relative_position": [
      -2.0,
      1.0
    ],
    "terrain_type": 0.0
  },
  {
    "relative_position": [
      -2.0,
      2.0
    ],
    "terrain_type": 0.0
  }
],
[
  {
    "relative_position": [
      -1.0,
      -2.0
    ],
    "terrain_type": 0.0
  },
  {
    "relative_position": [
      -1.0,
      -1.0
    ],
    "terrain_type": 0.0
  },
  {
    "relative_position": [
      -1.0,
      0.0
    ],
    "terrain_type": 0.0
  },
  {
    "relative_position": [
      -1.0,
      1.0
    ],
    "terrain_type": 0.0
  },
  {
    "relative_position": [
      -1.0,
      2.0
    ],
    "terrain_type": 0.0
  }
],
[
  {
    "relative_position": [

```

```

        0.0,
        -2.0
    ],
    "terrain_type": 0.0
},
{
    "relative_position": [
        0.0,
        -1.0
    ],
    "terrain_type": 0.0
},
{
    "relative_position": [
        0.0,
        0.0
    ],
    "terrain_type": 0.0
},
{
    "relative_position": [
        0.0,
        1.0
    ],
    "terrain_type": 0.0
},
{
    "relative_position": [
        0.0,
        2.0
    ],
    "terrain_type": 0.0
}
],
[
    {
        "relative_position": [
            1.0,
            -2.0
        ],
        "terrain_type": 0.0
    },
    {
        "relative_position": [
            1.0,
            -1.0
        ],
        "terrain_type": 0.0
    },
    {
        "relative_position": [
            1.0,
            0.0
        ],
        "terrain_type": 0.0
    }
]

```

```

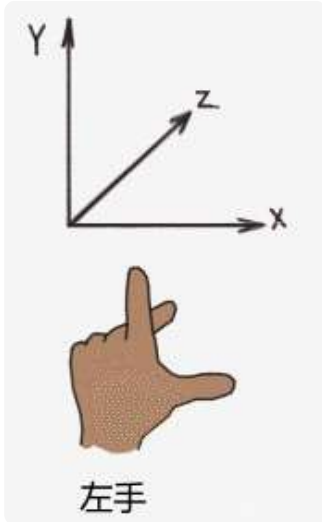
        "relative_position": [
            1.0,
            1.0
        ],
        "terrain_type": 0.0
    },
    {
        "relative_position": [
            1.0,
            2.0
        ],
        "terrain_type": 0.0
    }
],
[
    {
        "relative_position": [
            2.0,
            -2.0
        ],
        "terrain_type": 0.0
    },
    {
        "relative_position": [
            2.0,
            -1.0
        ],
        "terrain_type": 0.0
    },
    {
        "relative_position": [
            2.0,
            0.0
        ],
        "terrain_type": 0.0
    },
    {
        "relative_position": [
            2.0,
            1.0
        ],
        "terrain_type": 0.0
    },
    {
        "relative_position": [
            2.0,
            2.0
        ],
        "terrain_type": 0.0
    }
]
]
}

```

各欄位敘述請看Proly的README。

更多資訊(object ID之類的)都在最上面的README裡面，可以自己去看。

在本次的上帝視角中，向前為z+，向右為x+，向上(飛起來)為y+，如附圖(圖源: CSDN)。



模型讀取方法

請各位把模型與ml_play.py檔案放在同一個資料夾(ml/)，並且透過以下方法進行讀取：

```
import os

BASE_DIR = os.path.dirname(__file__) # ml_play.py所在的資料夾
model_path = os.path.join(BASE_DIR, "model.pickle")
#透過model_path去做該做的事
```

因為助教要批次跑程式的關係，所以會把你在ml的檔案搬到其他地方，因此你的檔案並不會在你的資料夾裡面。

沒改的話助教這次不會幫你改，上次已經提醒過了。用以下方式

```
path = 'ml/the_model_that_will_never_be_loaded.npz'
path = './the_model_that_will_never_be_loaded.pkl'
path = 'the_model_that_will_never_be_loaded.zip'
torch.load("l/the_model_that_will_never_be_loaded.pth")
```

會跟這個檔案名稱一樣，永遠不會被讀取，如因為找不到檔案跑不起來，後果自負。

作業繳交方法

請將壓縮檔命名為 學號_名字_HW4.zip (ex. F74106666_胡萬_HW4.zip)。

會使用同一個python檔案執行，模型個數不限。

請務必把要參戰的python檔案取名為 ml_play_battle.py 。

資料夾結構如下(不用附proly遊戲主程式):

```
F74106666_胡萬_HW4/  
├─proly/  
│   └─ml/  
│       ├──ml_play_battle.py  
│       └─model.pickle  
└─理論上你不用放其他東西在這裡/
```

如果你的程式在助教的電腦跑不起來，會當作未按規定繳交。