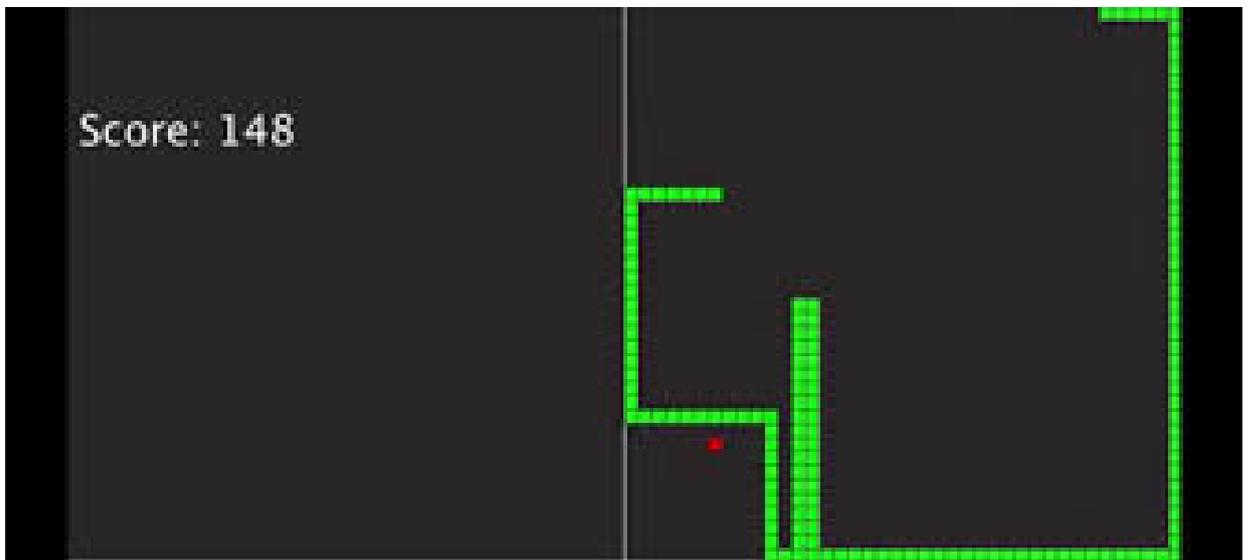


Project 3

108034542 溫致淵

使用 Q-Learning 建構貪吃蛇的遊戲環境

有鑑於機器學習與深度學習在各個領域分別有突破性的建樹，本次 IIE 期末 project 預計使用網路上的 Q-Learning code 模板來建構一個貪吃蛇的生存環境。Q-Learning 是一種增強式學習的 Deep Learning 方式，不需要進行模型的預先訓練。



但 Q-Learning 最困難的地方即為環境的建置與參數調整，正因為 Q-Learning 無須訓練資料，所以非常依靠環境&參數協助電腦訓練模型。本次期末 Project 期望使用網路上建置好的環境參數進行訓練；未來展往是調整建置好的環境參數使其成為一個可使程式成功運行的環境。

貪吃蛇環境建立:

貪吃蛇因為環境建立簡單，參數設定很少的緣故；故訂定為本次 Q-Learning 的建構模板。原先希望建構一個 100 * 100 的貪吃蛇活動環境，但在建立後發覺環境過大導致訓練時間過長；後來就改成了 50 * 50 的貪吃蛇活動環境。

Q-Learning Model 建立:

最初使用網路上的開源代碼直接嘗試應用於貪吃蛇遊戲中；但發現效果不佳，最後有參考 GitHub 上已經有成果的數個貪吃蛇遊戲，嘗試理解他們的 Q-Learning code 後來修改並整合進自己的貪吃蛇遊戲。

```
def get_state(self, game, player, food):
    state = 1
    (player.x_change == 30 and player.y_change == 0 and ((list(map(add, player.position[-1], [20, 0])) in player.position) or
    player.position[-1][0] + 20 >= (game.game_width - 20)) or (player.x_change == -20 and player.y_change == 0 and ((list(map(add, player.position[-1], [-20, 0])) in player.position) or
    player.position[-1][0] - 20 < 20) or (player.x_change == 0 and player.y_change == -20 and ((list(map(add, player.position[-1], [0, -20])) in player.position) or
    player.position[-1][1] - 20 < 20) or (player.x_change == 0 and player.y_change == 20 and ((list(map(add, player.position[-1], [0, 20])) in player.position) or
    player.position[-1][1] + 20 >= (game.game_height - 20))), # danger straight

    (player.x_change == 0 and player.y_change == -20 and ((list(map(add, player.position[-1], [20, 0])) in player.position) or
    player.position[-1][0] + 20 > (game.game_width - 20)) or (player.x_change == 0 and player.y_change == 20 and ((list(map(add, player.position[-1],
    [-20, 0])) in player.position) or player.position[-1][0] - 20 < 20) or (player.x_change == -20 and player.y_change == 0 and ((list(map(
    add, player.position[-1], [0, -20])) in player.position) or player.position[-1][1] - 20 < 20) or (player.x_change == 20 and player.y_change == 0 and (
    list(map(add, player.position[-1], [0, 20])) in player.position) or player.position[-1][1] + 20 < 20) or (player.x_change == 20 and player.y_change == 0 and (
    -1] + 20 >= (game.game_height - 20))), # danger right

    (player.x_change == 0 and player.y_change == 20 and ((list(map(add, player.position[-1], [20, 0])) in player.position) or
    player.position[-1][0] + 20 > (game.game_width - 20)) or (player.x_change == 0 and player.y_change == -20 and ((list(map(
    add, player.position[-1], [-20, 0])) in player.position) or player.position[-1][0] - 20 < 20) or (player.x_change == 20 and player.y_change == 0 and (
    list(map(add, player.position[-1], [0, -20])) in player.position) or player.position[-1][1] - 20 < 20) or (
    player.x_change == -20 and player.y_change == 0 and ((list(map(add, player.position[-1], [0, 20])) in player.position) or
    player.position[-1][1] + 20 >= (game.game_height - 20))), # danger left
```

訓練模型並收集數據:

建構好整個 project 後便進行貪吃蛇遊戲的遊戲結果收集；並藉由調整 Q-Learning 的參數來獲得最佳的遊戲結果。



參數調整(1/6)



參數調整(2/6)

- 由先前調整確認了最佳學習率介於 0.0005~0.0001 間。後續在 LR=0.0005 條件下調整獎勵與懲罰參數

分數/學習率	0.01	0.001	0.0005	0.0001
最低分數	0	0	0	0
最高分數	2	49	54 	52

分數/獎勵權重	+30/0	+20/0	+10/-10	0/-20	0/-30
最低分數	0	0	0	0	0
最高分數	8	10	56 	2	0

由上圖可知除了 Q-Table 的 Gamma 參數外學習率的調整也可以大幅度的影響測試結果。

後續可改進之處:

這次的 project 我覺得還有許多可以改進的地方，如:

1. 調整環境大小
2. 利用實驗設計的方式找出最佳的參數與參數設定
3. 持續優化貪吃蛇環境讓介面更優美