-國立中央大學

計算型智慧-HW1

自走車 Q-Learning

學　號：113323098

學　生：葉律旻

授課教授：蘇木春 教授

中華民國 113 年 4 月 10 日

第一章 程式介面說明

- **系統構成**

本專案包含三個主要檔案：

- **playground.py**
  主程式，包含 Q-Learning 訓練、模擬環境的定義（Playground）以及車體（Car）之運動模型。
- **simple_geometry.py**
  幾何運算輔助模組（例如 Point2D、Line2D 類別），用以計算物件位置、距離和角度。
- **軌道座標點.txt**
  定義跑道邊界與終點區域的資料檔，供模擬環境讀取並繪製軌道圖形。

- **程式功能**

1. **車體運動模擬**
   根據簡化運動模型更新車體位置和角度。
2. **Q-Learning 算法**
   以 Q-Table 的方式學習車體如何根據三個感測器（前、右、左）的距離獲取合適的方向盤角度，從而讓車體在跑道上安全運行並最終抵達終點。
3. **視覺化 UI**
   利用 matplotlib 製作動畫，繪製跑道、車體、方向箭頭、感測器射線及車體行走軌跡。

- **使用方法**
  - 執行方法：
    - 執行打包後的 exe 檔 (**playground.exe**)，程式會先進行 Q-Learning 訓練，再以 GUI 模擬方式展示車體運動與學習策略。
  - 可執行檔：本次作業提供的 exe 檔包含完整 UI 介面，可顯示模擬結果，不依賴其他 AI 框架。

第二章 實驗結果

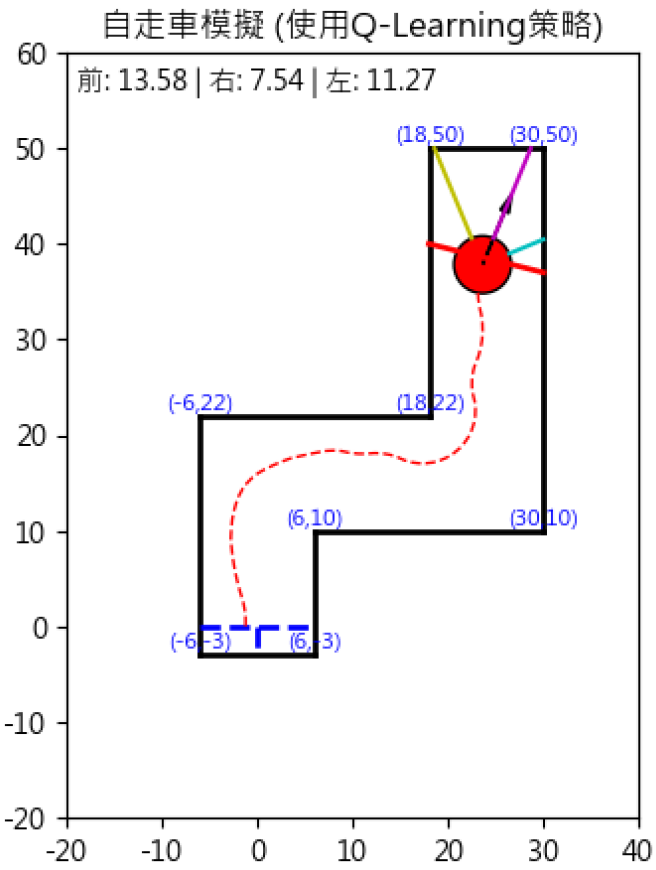

圖 1 最終模擬畫面截圖

最終模擬畫面截圖如圖 1 所示，其中包含：

- 跑道邊界（用黑色實線表示）
- 終點區域（用紅色線條標示）
- 車體位置與方向箭頭
- 感測器射線（用不同顏色顯示前、右、左）
- 車體運動軌跡（以紅色虛線記錄移動路徑）

部分訓練回合結果如下：

```
到達終點區域
Episode 5000 finished in 66 steps, total reward: 136.24161169989895
Training complete.
Highest reward was 138.0112459280673 in episode 3060
```

圖 2 訓練回合結果

第三章 感測器距離與 Reward 之關係

- **感測器說明**

本系統使用三個感測器，分別偵測車體正前方、右側與左側與障礙（跑道邊界）之距離。

- **Reward 規則**

  - 當車體進入終點區域時，Reward 為 +100；
  - 當車體發生碰撞時（車體與牆線或障礙物過近）Reward 為 -100；
  - 若既未碰撞、也未到達終點，則 reward 根據車體中心與終點線的距離給出，公式為：

$$Reward = 0.02 \cdot (50 - distance\_to\_destination)$$

其中 *distance_to_destination* 為車體中心到終點線的距離。

- 當距離 *distance_to_destination* 越小（即離終點越近），reward 越高；
- 當 *distance_to_destination* 大於 50 時，reward 為負，表示車體與終點過遠。

第四章 分析與心得

- **Q-Learning 設計問題與思考：**

1. **狀態空間的離散化**
   - 由於感測器數值為連續值，本作業使用離散化策略（將 0～50 分成 10 個區間），如果區間太大，可能導致狀態表示不足；反之，區間過細則會導致 Q 表維度過高。
   - 試驗並調整離散區間（本程式範例中參數可在 main 部分設定 NUM_BINS、SENSOR_MIN、SENSOR_MAX）來達到最佳平衡。
2. **Reward 設計的難點**
   - 獎勵函數必須在鼓勵車體迅速到達終點與懲罰碰撞間取得平衡。
   - 我們在基本獎勵上加入根據與終點距離決定的獎勵，使得代理學會朝正確方向移動，但獎勵值必須小心調整，否則會出

現策略偏差（例如代理可能只為了減少步數而採取風險行為）。

3. **策略收斂**
    - 使用 Q 表對狀態與動作進行「窮舉」，雖然概念上直觀，但當狀態空間較大時可能收斂較慢。
    - 此外，初始狀態的隨機性以及 epsilon 衰減策略也對模型收斂有很大影響。

- **心得**

在設計 Q-Learning 系統時，我們遇到了以下挑戰：

- 如何合理離散化連續狀態。
- 如何設計獎勵函數，使得車體不僅追求減少步數，還能穩健地避開碰撞並正確朝向終點移動。

這次作業讓我更深入理解強化學習的核心概念，特別是 ε -greedy 策略在探索與利用之間的平衡應用。透過實作 Q-Learning，我學會如何將感測器輸入離散化為狀態、設定獎勵機制，以及建立 Q-Table 並持續更新，最終讓自走車能夠學會避開障礙並成功到達終點。過程中我也發現動態規劃的概念實際上就是 Q-Learning 背後的理論基礎之一。雖然在設計 reward 時遇到困難，但最終成功讓模型學會策略，帶給我很大的成就感，也讓我對強化學習的實際應用產生濃厚興趣，獲益良多。