**計算型智慧 作業一**

**110403518資工4B林晉宇**

|  |
| --- |
| **一、程式介面說明** |

本次作業以Python的PyQT5開發互動式的GUI介面（圖一），整體介面分為左右兩區，左側為控制與顯示資訊的部分，可以調整超參、顯示車輛位置、探測器狀態；右側為顯示軌道邊界、自走車、與車子軌跡。

左側功能區：

1. **匯入軌道**：點選「Import Track File」可讀入軌道.txt檔案，並顯於右方顯示軌道
2. **超參設定**：包含學習率（Learning Rate）、ε（探索率）、ε 衰減、折扣因子（Discount Factor）與訓練次數（Episode）。
3. **訓練速度**：最低可設為 1ms 以達到快速訓練效果。
4. **控制按鈕**：提供開始、暫停訓練（Stop Training）與重置車輛（Reset Car）。
5. **Decision Log**：記錄訓練過程中的決策狀態。
6. **車輛資訊**：即時顯示車輛的座標 (x, y) 及角度θ。
7. **感測器資訊**：顯示三個方向（左前右）對邊界的偵測距離。

右側畫布區：

1. 顯示匯入的軌道圖形，包含白色邊界線、紅色起點、綠色終點區域。
2. 車輛以圓形表示，朝向以藍色線段表示。
3. 訓練過程中，車輛行經路徑會留下白色軌跡。

**一張含有 文字, 螢幕擷取畫面, 多媒體軟體, 軟體 的圖片

AI 產生的內容可能不正確。**

圖一、GUI介面展示

|  |
| --- |
| **二、實驗結果** |

經過大量實驗後，最終發現趨於穩定的設定如下：

1. Learning Rate (): 0.31
2. Epsilon (): 1
3. Epsilon Decay: 0.995
4. Discount Factor: 0.95
5. Episode: 3000

|  |
| --- |
| **三、判斷規則** |

1. **Discount factor:** γ 的數學作用是**讓 Bellman equation 在無限時間問題中收斂**（Bertsekas, 1995），避免回報發散。γ 控制 agent 對**未來獎勵的重視程度**，也就是 **“越小代表越急性子”**，越想要馬上拿到獎勵。

一張含有 文字, 名片, 螢幕擷取畫面, 字型 的圖片

AI 產生的內容可能不正確。

|  |
| --- |
| **四、分析與探討** |

**一開始state action切很細，但後來發現粗糙就好。**

**State 跟 action設計 一開始3000笨->聰明->笨**

**一開始state input 三個數字**

**State改成0~4 到最後 起點會一直右轉撞牆**

**我認為隨機性質還是站最大比重**

|  |
| --- |
| **五、總結** |

1. GUI介面