

計算型智慧

作業一

Q_learning

學號：109401553

系級：資管三 B

學生：楊雲杰

日期：2024/04/30

一、程式介面說明

我的執行檔路徑為 exe_file/simple_playground/simple_playground

 _internal	2024/4/28 下午 09:46	檔案資料夾
 simple_playground	2024/4/28 下午 09:46	應用程式

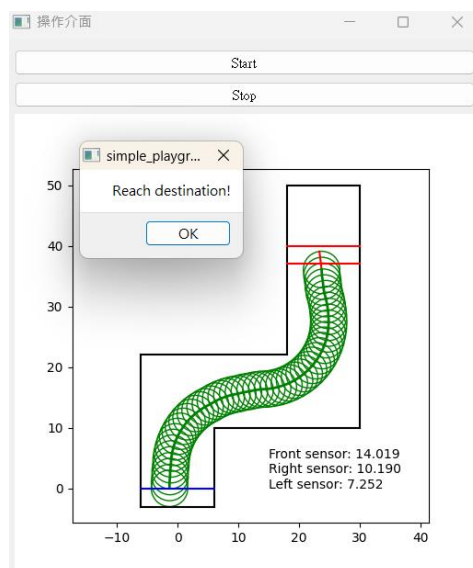
圖一、執行檔路徑

執行檔案後會先進行 2000 次訓練，約需等待 30 秒，待訓練完成後可點擊 Start 讓車子走至終點。若於訓練時點擊畫面會沒有反應。

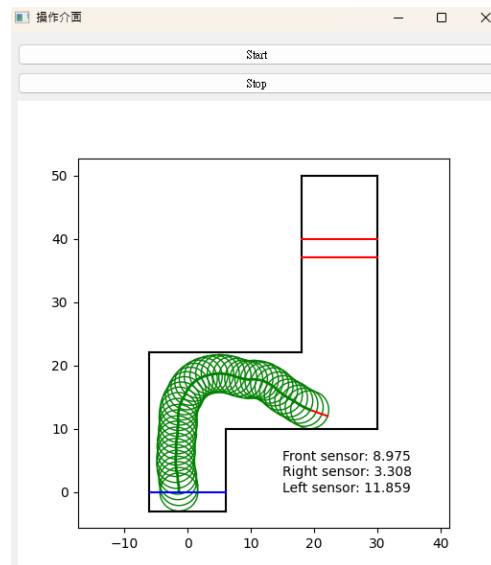


圖二、等待訓練畫面

起點線為藍色線，終點為紅色長方形，車子為綠色空心圓圈(為了方便看出軌跡)。介面上方有按鈕 Start，點擊即可讓車子進行一次模擬。Stop 則是在模擬進行中可以點擊暫停觀察狀況。介面右下角另有三行文字，分別為正前方、右 45 度角、左 45 度角的感測器所測量出的距離。



圖三、成功畫面



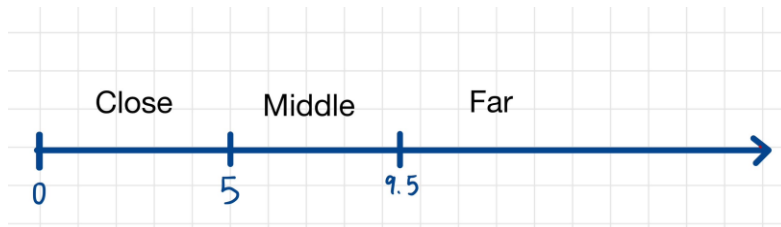
圖四、失敗畫面(撞牆會停止)

二、歸屬函數說明

State

1. 正前方感測器接收距離 $dist$ 之歸屬函數

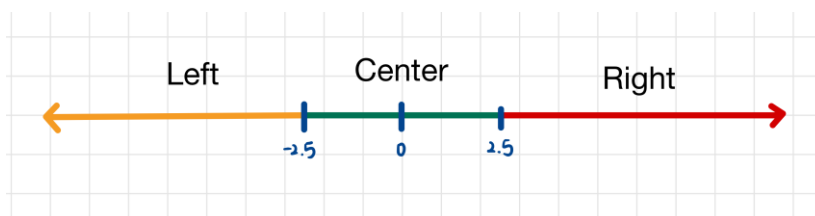
$9.5 < dist$, 距離是 far
 $5 \leq dist \leq 9.5$, 距離是 middle
 $dist < 5$, 距離是 close



圖五、正前方感測器

2. 右側感測器距離($dist_r$)減左側感測器距離值($dist_l$)之歸屬函數

$2.5 < dist_r - dist_l$, 向右轉因為右側有空間(right)
 $-2.5 \leq dist_r - dist_l \leq 2.5$, 保持直行因為在路中央(center)
 $dist_r - dist_l < -2.5$, 向左轉因為左側有空間(left)



圖六、右側減左側感測器距離

Action

我將方向盤角度分為 7 種角度

[-30, -15, -10, 0, 10, 15, 30]

Reward

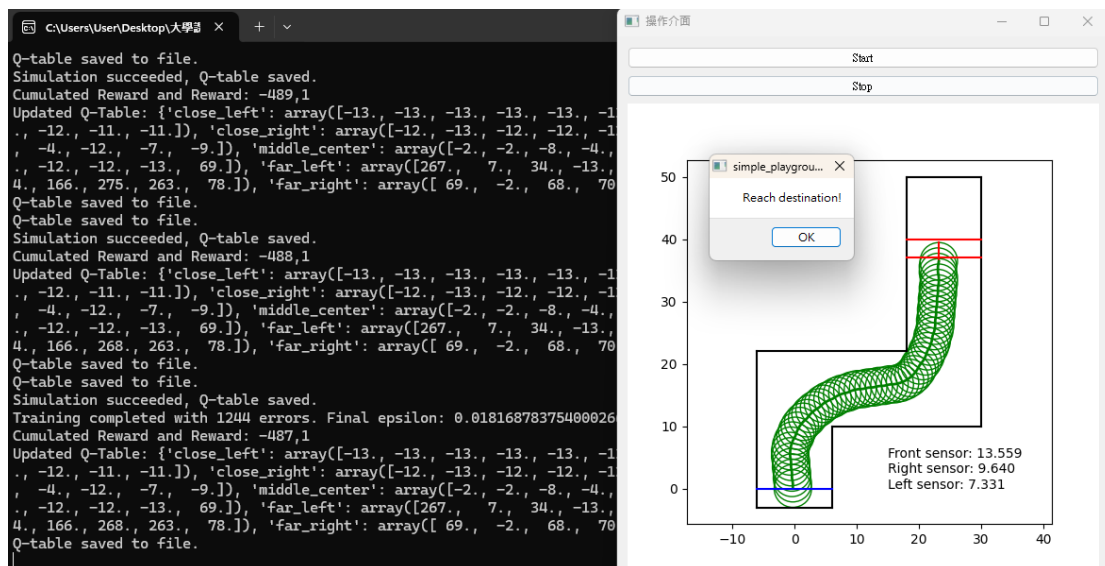
我針對抵達終點和撞牆有設置獎勵，另外不同距離也有不同的獎勵，因為我希望車子盡量保持在路中央。

表一、獎勵函數

	回饋
抵達終點	+1
撞牆	-1
前方感測器(< 4.5)	-0.08
右側感測器(< 5)	-0.08
左側感測器(< 5)	-0.08

三、實驗結果

我設置了 2000 次的 episode，讓我的車子學習如何走到終點。至於 Q_table 的部分則是狀態(3*3) * 行動(7)，是一個 9*7 的 table。另外，我的程式執行時，terminal 會印出 Q_table(詳見下圖七)



圖七、測試一次抵達終點

四、分析

在寫這次作業的過程中，獎勵函數的設置、狀態的設置是我認為非常有挑戰性的部分。

關於獎勵函數的設計，一開始，我的想法非常單純，我讓車子在每次轉動方

向盤時都給予獎勵，然而，這個方法效果不彰，大約十次會有八次撞牆。後來，我決定採用不同的獎勵方法，利用計算步數的策略，設定當步數走越多給予越多的獎勵，鼓勵車子學習走越遠越好的方法。使用這個方法後，我的車子有辦法抵達終點，但是抵達終點仍然不高。最後，我採用的方法是保有前面測試時的優點，並客製化一個獎勵函數，針對右方距離、左方距離小於 5 時給予負獎勵，前方距離小於 4.5 時也給予負獎勵，讓車子得以保持在道路的中央。這個獎勵函數也讓我的自走車能夠在起點是隨機的情況下了解如何避免障礙物，並抵達終點。

狀態(State)的部分，我則是採用距離牆壁遠(Far)以及中等距離(Middle)還有近(Close)三個類別，車子的偏離方向左(Left)、中(Center)、右(Right)三種類別。(這裡所謂的左是指左方有空間，右則是右方有空間。)

總而言之，這份作業花了我非常多的時間及心血，最終實現自走車可以走到終點的任務。也希望未來我能利用本次作業所學，將 Q learning 應用在各種不同的地方。