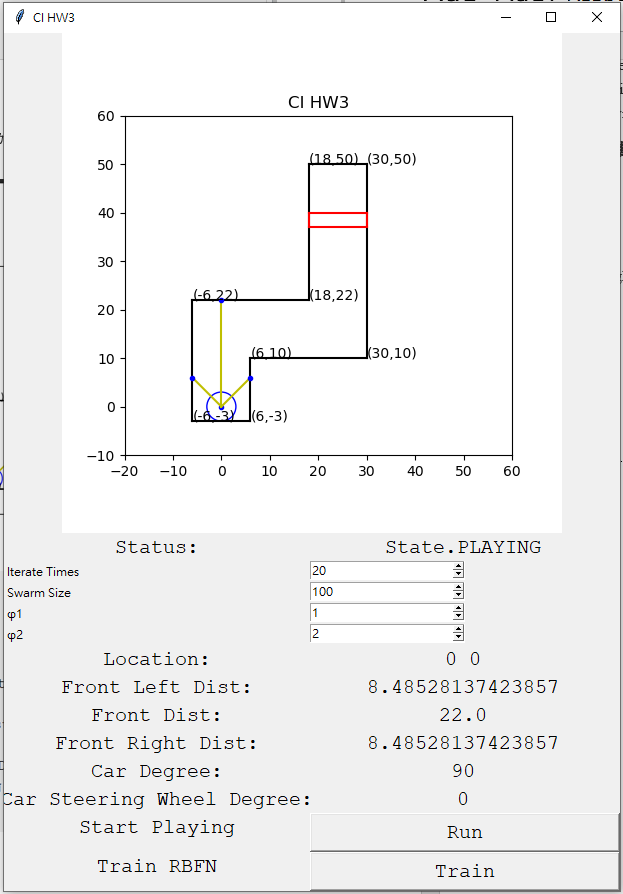
**作業三 - 電腦模擬車的實作 (基因演算法)**資工碩一109522027 陳逸星

**程式介面說明**

程式介面如下圖所示。啟動程式後按下Run即可進行模擬。



Status會告知使用者目前程式是否已完成模擬

Location為目前車車中心點座標

Front Left Dist為左側感測器至牆壁距離

Front Dist為前方感測器至牆壁距離

Front Right Dist為右側感測器至牆壁距離

Car Degree為車子目前角度

Car Steering Whell Degree為車子方向盤目前角度

Start Playing旁的按鍵Run則告知程式執行模擬

**相關參數設定及使用**

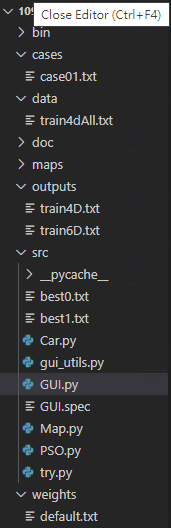
Iterate Times, Swarm Size φ1, φ2為PSO演算法所使用的參數。

改良完參數可以按下Train進行訓練。訓練完成的weights會放在weights資料夾，軌跡會存放於outputs資料夾中。

模型預設有已經跑好的參數在裡面。

**程式碼說明**

程式碼架構如下圖所示。



說明src資料夾內各模組功能：Car.py負責自走車行走、感測器偵測、及方向盤角度調整以及記錄自走車行徑、狀態之模組，包含將最終結果寫入檔案的部分；Map.py會載入case01.txt當作自走車的地圖。

本次作業主要使用到PSO.py，裡面分別有class Particle、class RBFN與class PSO分別定義了pso每個particle的元素以及rbf network還有整個POS演算法所需的各式功能。

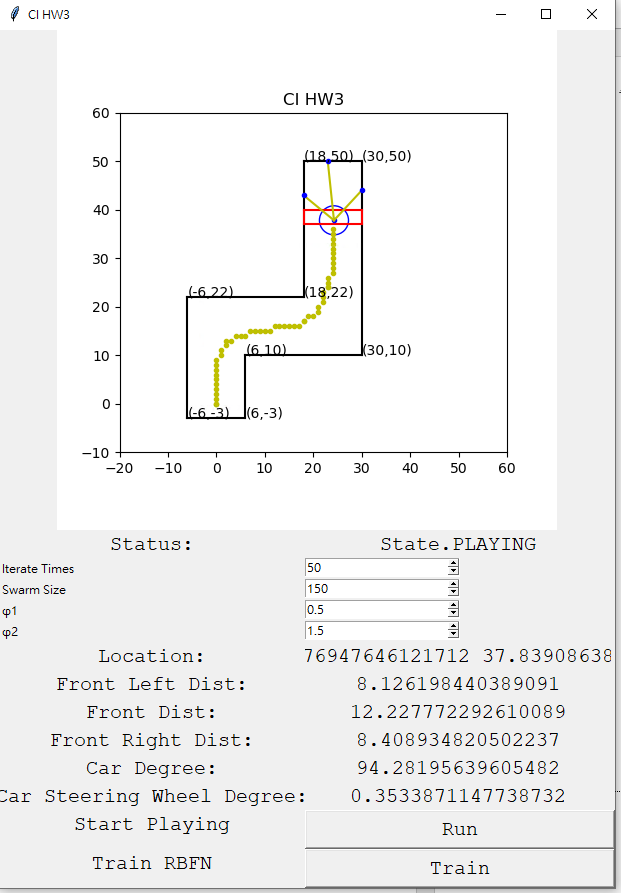
在PSO演算法的方面實驗定義了bestneighbor的particle與全域fitness最佳的pocket\_particle，過程中不斷的運用exp\_weight與neighbor\_weight來更新每個particle的各項參數，最後在經過一定的iteration後運用pocke\_particle來運行RBFN。

gui.py為圖形化介面模組，負責統整所有元件，為主程式進入點；gui\_utils.py實做一些增加圖形化介面元件的函式；

**實驗結果**

實驗結果與行徑軌跡如下圖。

**4D**



**實驗分析**

在PSO演算法的方面實驗定義了bestneighbor的particle與全域fitness最佳的pocket\_particle，過程中不斷的運用exp\_weight與neighbor\_weight來更新每個particle的各項參數，最後在經過一定的iteration後運用pocke\_particle來運行RBFN。

而訓練成功的資料4D使用 4個隱藏神經元，並且Swarm\_size為150，總共迭代了50次，φ1與φ2分別為0.5與1.5，程式在執行基因演算法的過程因為要不斷去計算rbfn的predict因此會花費較多時間。

在做rbfn時，有可能因為基因演算法隨機生成的參數不好，使用train4dall，有可能會讓模型在原地打圈，因此可能需要多嘗試幾次，以找出真的能夠在這張地圖跑出來的訓練資料。