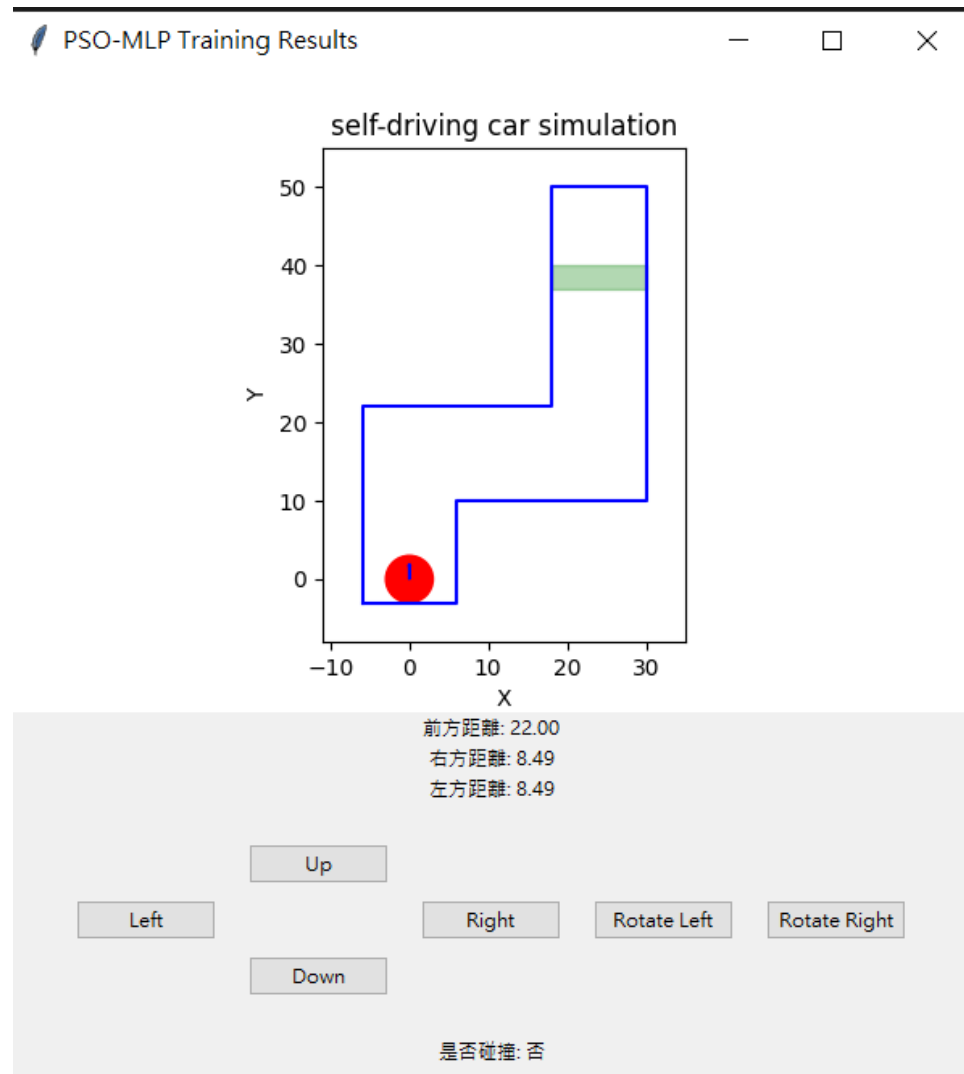


1. 程式介面說明

打開 exe 檔案 PSO 會先進行一段時間的訓練，可以在 CMD 看到訓練進度，訓練完後即會開始跑



下方按鈕為娛樂用，要按也可以按

2. 實驗結果與 PSO 實作說明

MLP 的參數如下，為沿用上學期類神經網路作業之設定

```
class MLPnetwork:
    def __init__(self, input_size=3, hidden_size=5, output_size=1):
```

PSO 參數如下

```
class PSO:
    def __init__(self, dim=26, particle_size=30, max_iter=100, lb=-1, ub=1):
```

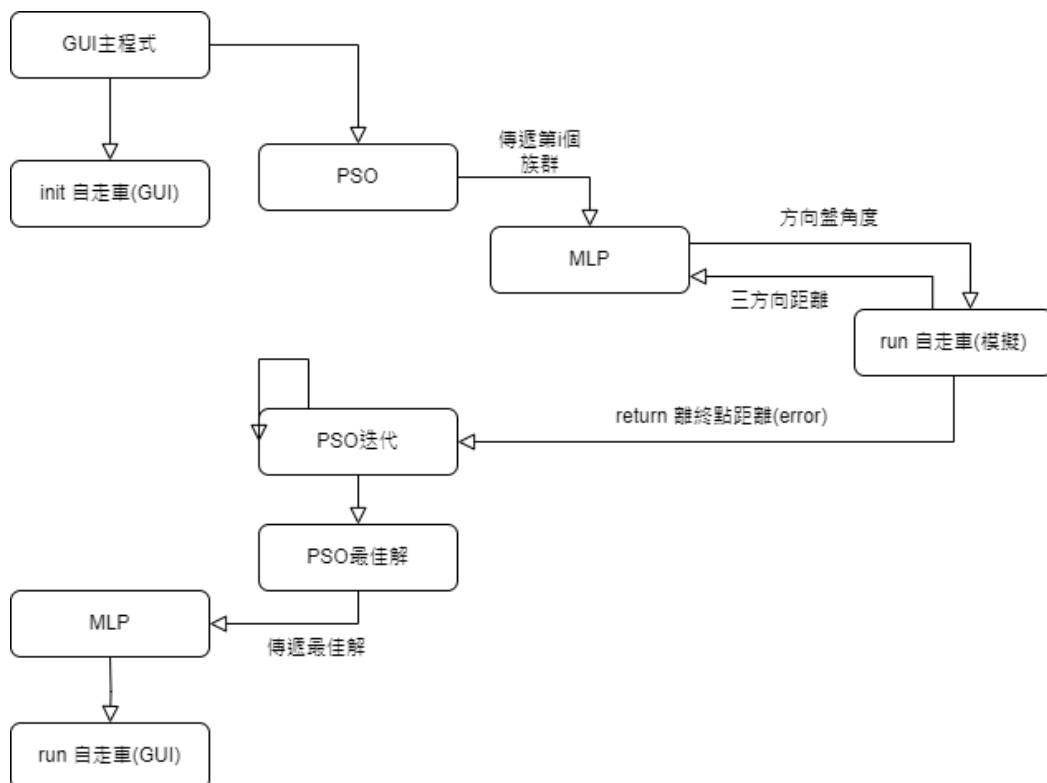
Dim= hidden * input + hidden + output * hidden + output

= $5*3 + 5 + 1*5 + 1 = 26$

PSO 的適應函數值採用離終點的距離值，並在取最佳解時取適應函式值最小的

```
def cal_score(self,x):
    car=SelfDrivingCar()
    self.mlp.set_params(x)
    while not(car.check_collision() or car.reach_goal()):
        #print(self.mlp.get_next_Th(car.get_distances()))
        car.update_state(self.mlp.get_next_Th(car.get_distances()))
    if car.reach_goal():
        return 0
    return MathTool.point_to_polygon_distance(car.x,car.y,car.end_area)
```

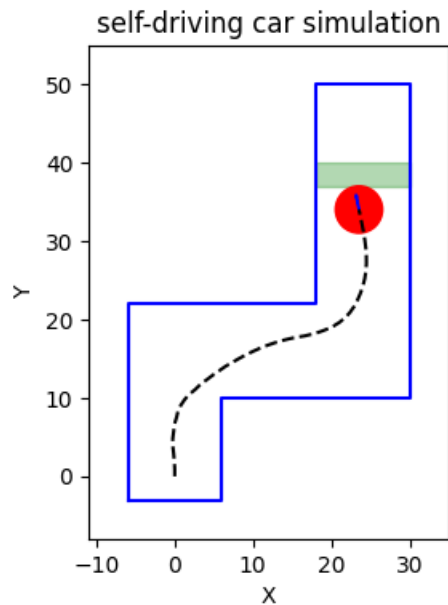
PSO 詳細流程圖



運行結果:

雖然 10 epochs 內就能收斂，經過反覆測試，100 epochs 的效果較穩定

```
PS C:\Users\James\Desktop\Self-Driving-Car> & D:/anaconda/envs/yoga/python.exe c:/Users/James/Desktop/Self-Driving-Car/PSO/gui.py
Iteration 0/100, Error(this epoch): 9.231516985685744
Iteration 1/100, Error(this epoch): 2.2826616307407526
Iteration 2/100, Error(this epoch): 0.45802080756493524
Iteration 3/100, Error(this epoch): 2.1138523746856968
Iteration 4/100, Error(this epoch): 0.36883477152442384
Iteration 5/100, Error(this epoch): 0.6160505138588329
Iteration 6/100, Error(this epoch): 0.8572965231138218
Iteration 7/100, Error(this epoch): 0.16216331103707185
Iteration 8/100, Error(this epoch): 0
Iteration 9/100, Error(this epoch): 0
```



前方距離: 16.27
右方距離: 11.70
左方距離: 6.65

Left

Up

Right

Rotate Left

Rotate Right

Down

3. 心得

這次只要確認好輸入輸出，以及公式沒有打錯，剩下的就可以交給魔法(PSO)了

如果還有問題，就把 `iteration` 調高一點