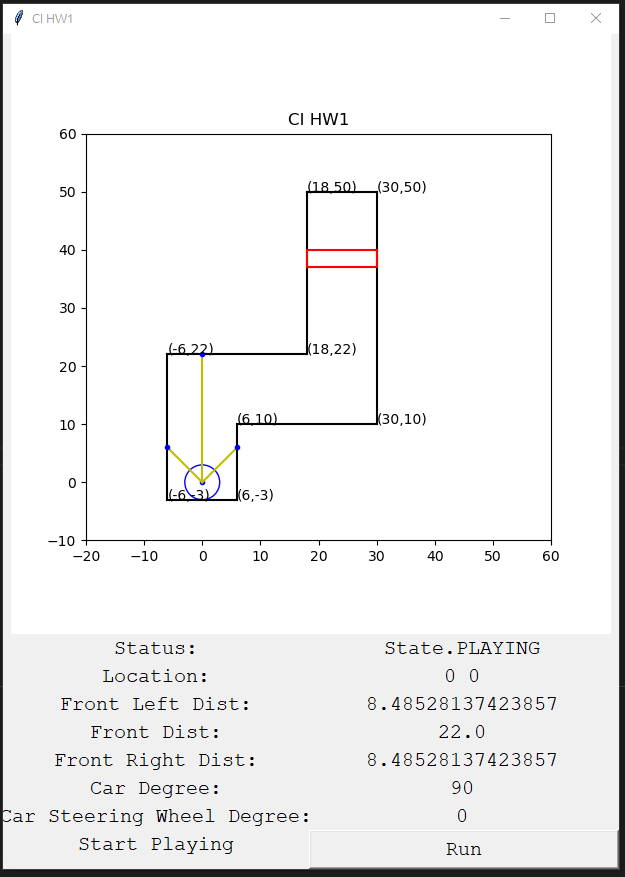
**作業一 電腦模擬車的實作** 資工碩一109522027 陳逸星

1. **程式介面說明**



**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

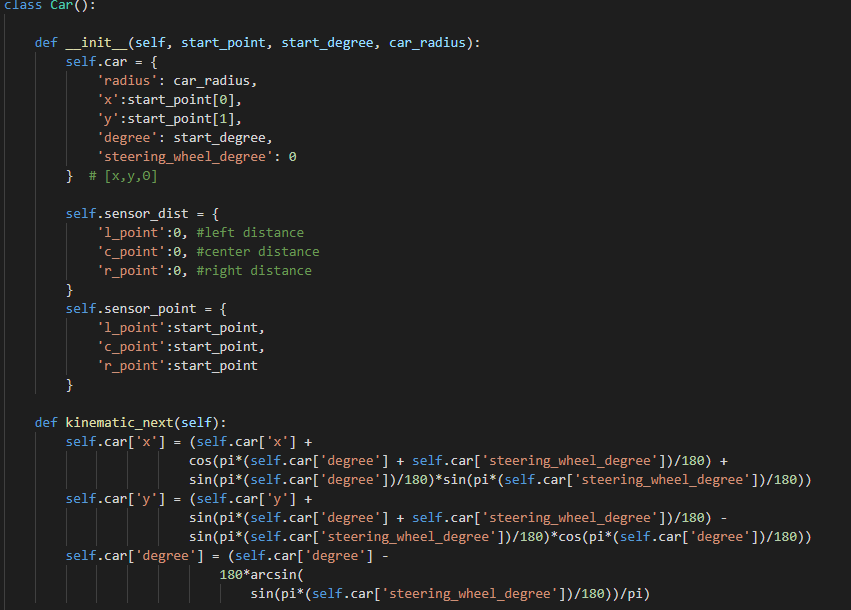
Car Degree為車子目前角度

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

**二、程式碼說明**

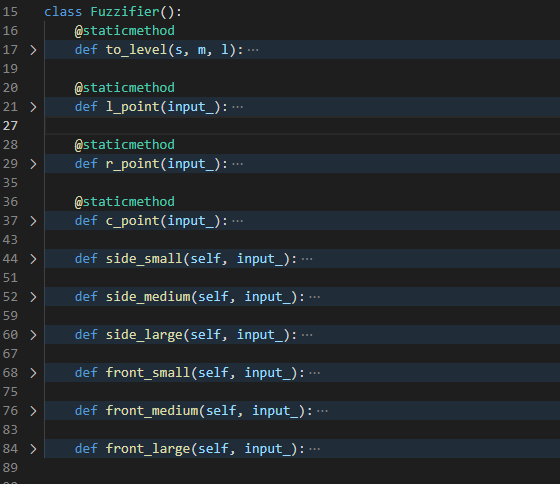
**Car.py**



所有跟車子有關的事，包含車子初始化作標，感測器位置，感測器偵測的距離等，皆在Car.py實現

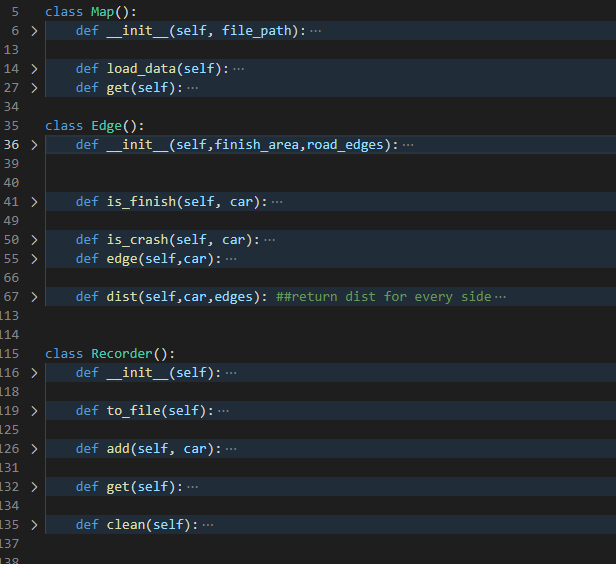
其中較為關鍵的有kinematic\_next為車車的運動模型，update\_sensor在車子移動後不斷去更新車子三個感測器到牆壁的距離。

**fuzz.py**



Class fuzzifier則定義了所有模糊規則詳細規則會在下點詳述

**Map.py**



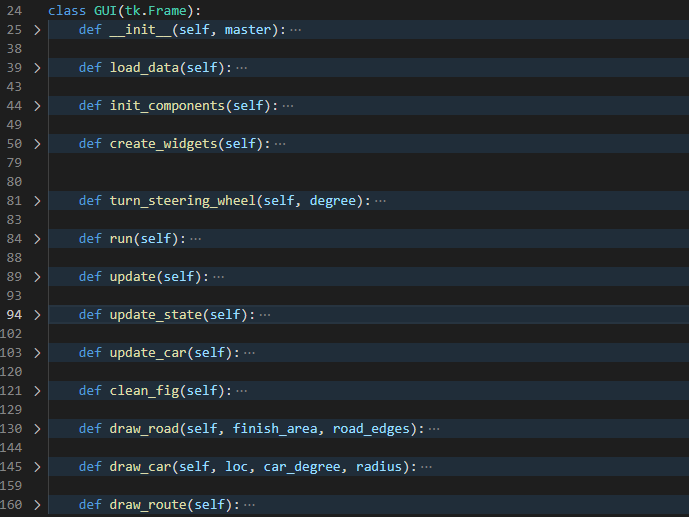
所有跟地圖有關，包含讀取檔案(class Map)，定義地圖邊緣與判斷是否撞車(class edge)，以及紀錄路徑(class Recorder)皆在此檔案

**gui\_utils.py**



簡單定義一些函數用來畫GUI

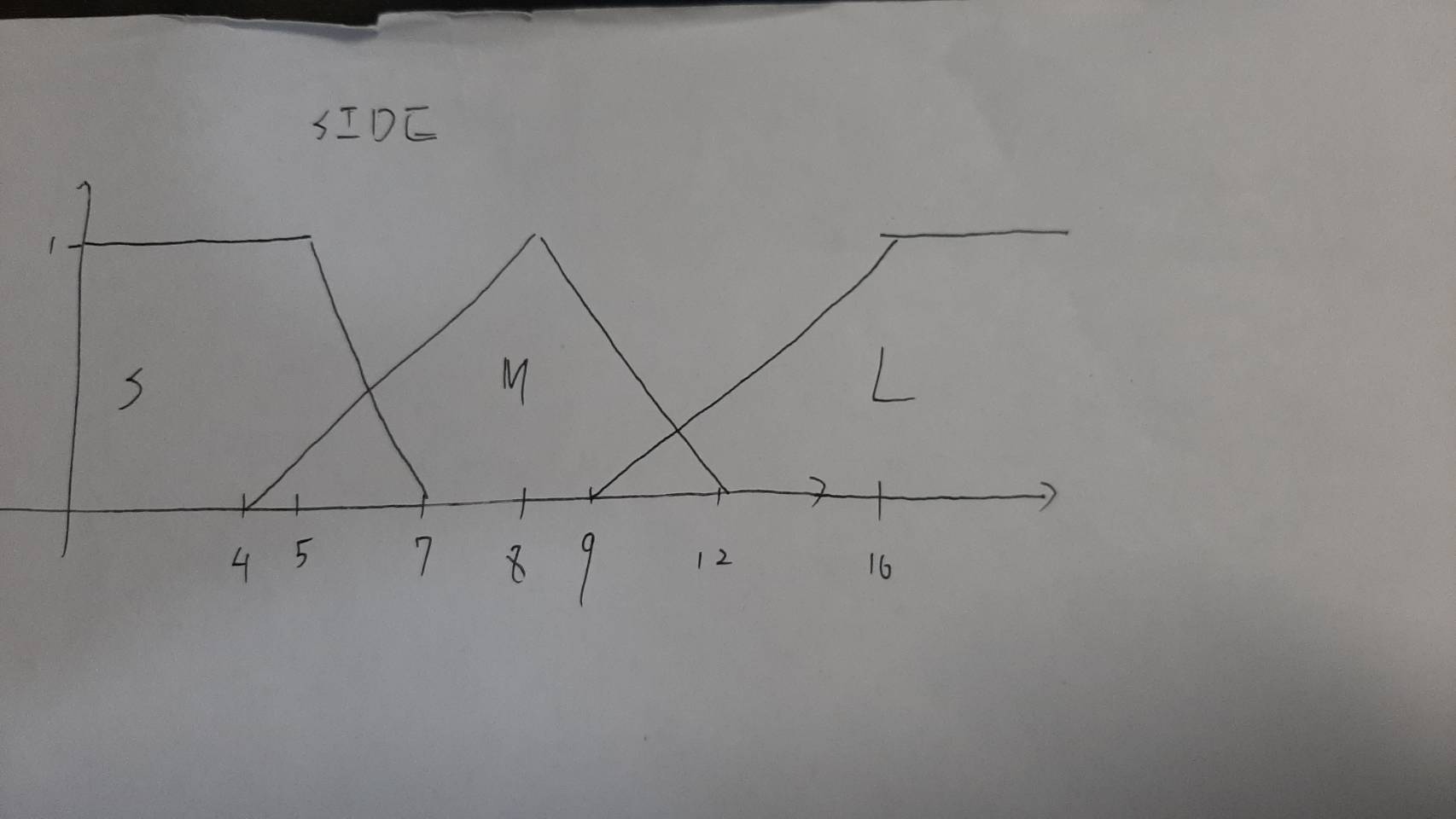
**GUI.py**

為所有程式的主程式，藉由GUI互動去call不同class裡面的function，根據模糊規則去不斷更新車車的位置，感測器的各項參數。

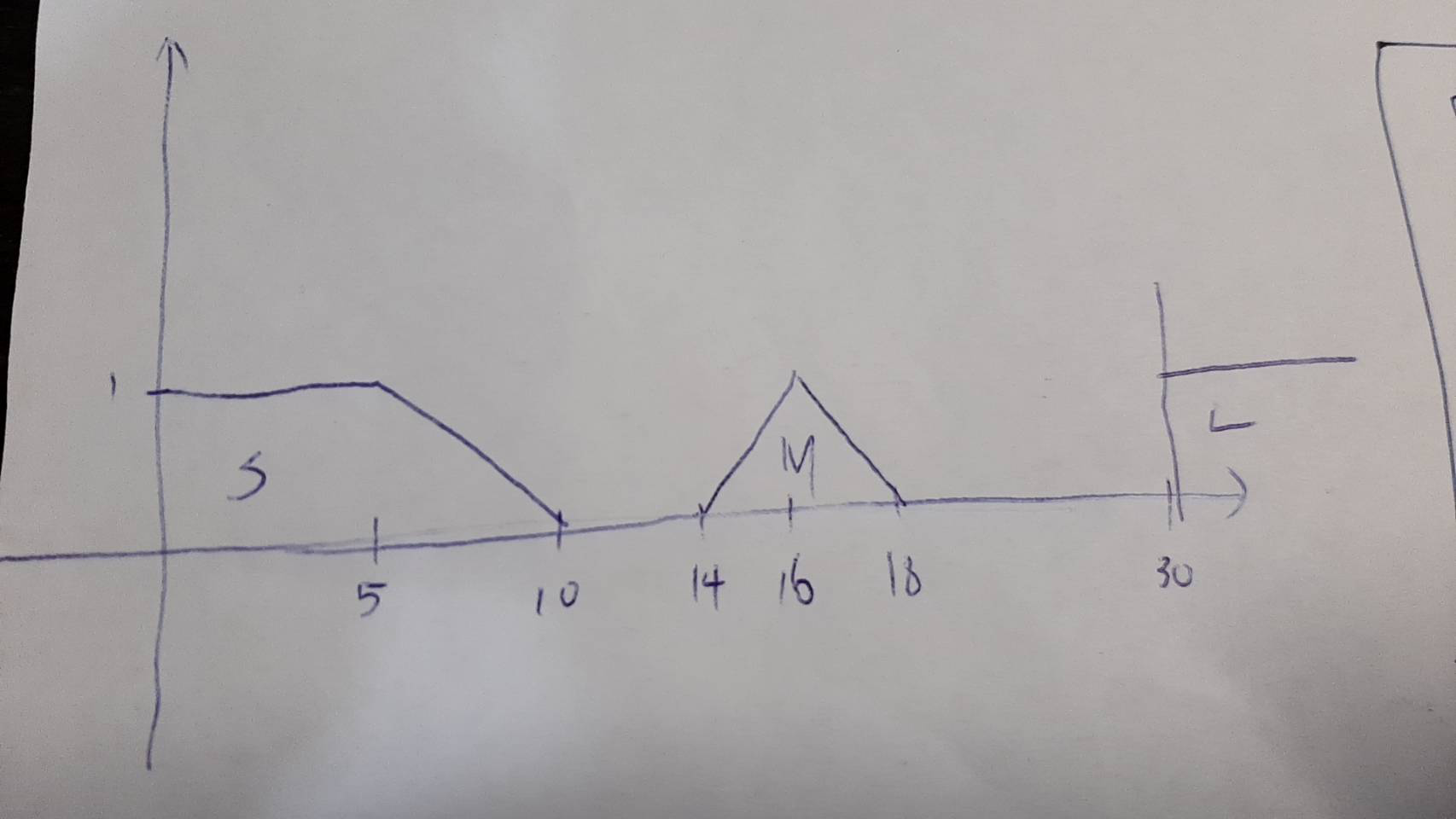
**三、模糊規則設計**

程式模糊規則的設計上，前件部根據要求取三個模糊變數分別為，前方距離、左45度距離、右45度距離。後件部的模糊變數則是設定為唯一輸出，方向盤角度。

而變數歸屬函數的部分，因感測器有左、前、右，左邊與右邊可以類比，因此歸納出兩個模糊函數

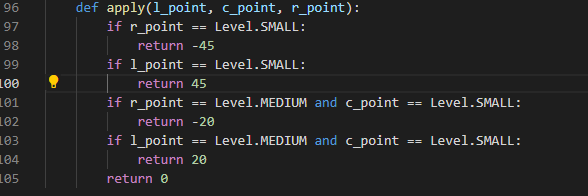


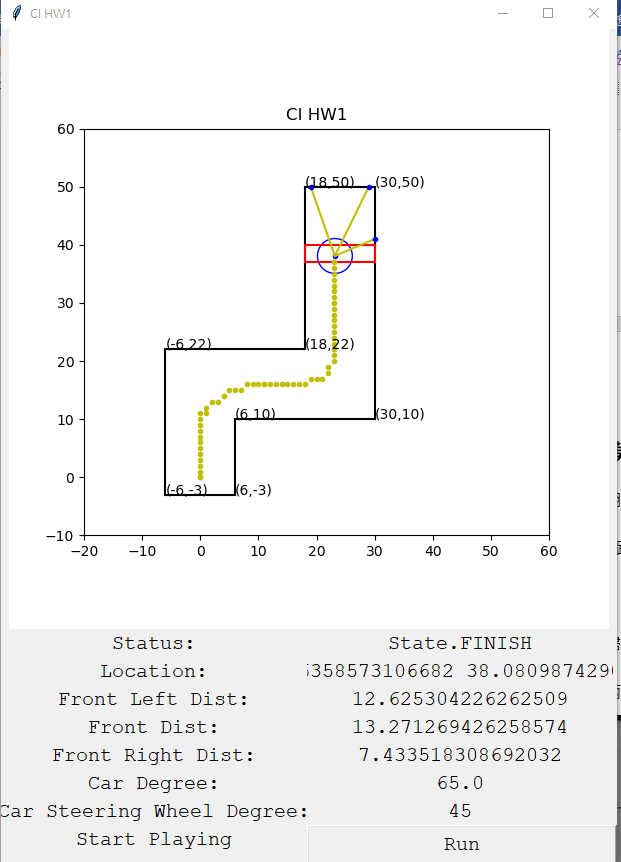
圖為左側右側之歸屬函數(SIDE)。



上圖為前側之歸屬函數(Front)

模糊聯集選擇最大值，模糊蘊含則如下圖所示



**四、實驗結果(包含移動軌跡截圖)**

**五、分析**

因作業時間及自己在GUI上一直卡關的關係，原本想試試看使用較為複雜的模糊規則，但只好先用簡單直接的模糊規則，因此模糊規則很有可能也是只限定在這種簡單的地圖上能有效使用，具體規則要怎麼定，其實還是有機會找出最佳化的方法。