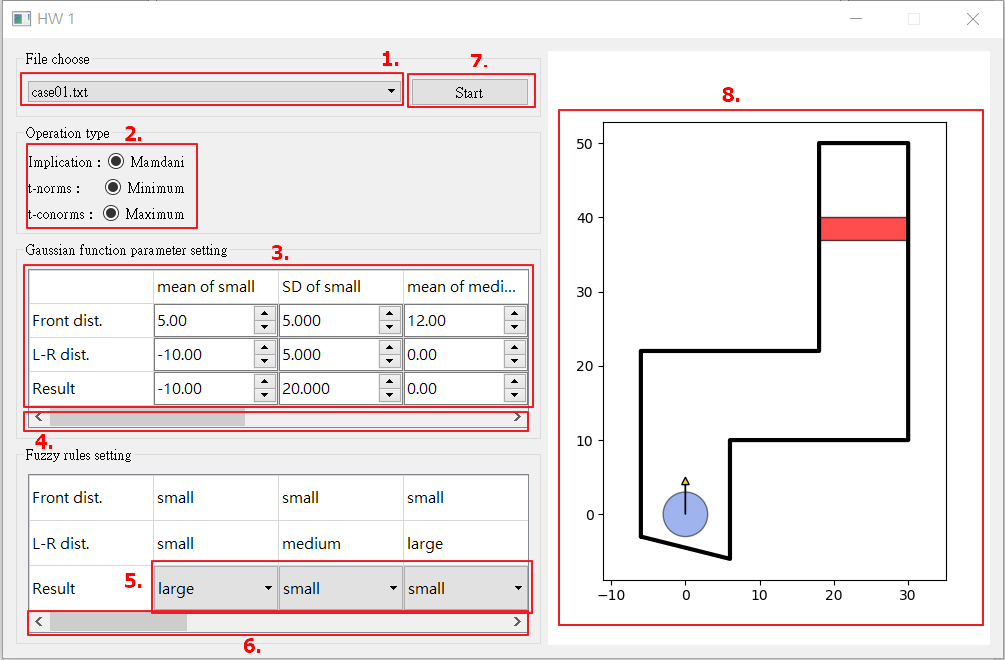
關鍵程式碼解釋

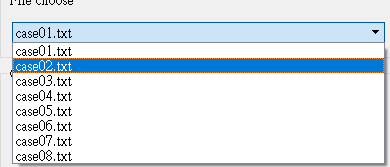
模糊函數的設計理由及分析  
實驗設計

模糊規則設計、實驗結果(包含移動軌跡截圖)與分析，實驗內容需能以繳交檔案重現，請於文件內說明如何操作。

1. 程式介面與操作說明



* 1. 選擇輸入軌道檔案，展開如下圖所示。讀取檔案為執行檔案的同個目錄下的「data」資料夾中所有txt格式檔。



* 1. 顯示模糊蘊含、模糊交集、模糊聯集的運算方式，若之後需要擴充也可以直接增加按鈕在此區域。
  2. 設定每個模糊變數的歸屬函數的參數(高斯函數的mean、standard deviation)，預設是實驗出可以到達終點的參數。另外此處措辭集中的small用的高斯函數是單調遞減函數，小於mean部分歸屬程度都為1；措辭集中的medium用的是正常高斯函數；措辭集中的large用的高斯函數是單調遞增函數，大於mean部分歸屬程度都為1。
  3. 橫移軸用來調整表格橫向位置
  4. 設定每個模糊規則的後鑑部，有large、medium、small可以選擇。
  5. 橫移軸用來調整表格橫向位置。
  6. 當參數、規則都設定好了之後，就可以按下此鍵，進行運算。
  7. 當start被按下並計算完，此處就會畫出車子移動過程與軌跡。若撞到牆車子會變成紅色並停止；若抵達終點，車子會變成綠色並停止，然後會在執行檔案的同個目錄下直接輸出紀錄檔，train4D.txt、train6D.txt。

1. 程式碼說明
   1. 本次作業總共分為四個部分，main.py、gui\_root.py、run.py、plot.py，分別用來讀取檔案、建立介面、執行計算(包含模糊系統相關計算、碰撞處理)、將計算結果、軌跡顯示在地圖中。
   2. library部分主要引入四個， pyqt5 用來構築GUI介面 ( 按鈕、數值輸入、選項 )，shapely、descartes用來處理碰撞、抵達終點問題，matplotlib 則是處理輸出影像 ( 右側顯示區域 )
   3. 首先介紹main.py，其中主要功能就是把讀入檔案傳給gui\_root.py。

def main():

...代表省略不重要程式碼

"""讀入檔案傳給gui\_root"""

gui\_root = GuiRoot(read\_file())

...

def read\_file():

"""讀取檔案function"""

road\_map = namedtuple('road\_map', ['start', 'x', 'y'])

datapath = join(os.path.realpath(os.path.join(os.getcwd(), os.path.dirname(\_\_file\_\_))), "data")

folderfiles = os.listdir(datapath)

dataset = {}

paths = (join(datapath, f) for f in folderfiles if isfile(join(datapath, f)))

for idx, content in enumerate(list(map(lambda path: open(path, 'r'), paths))):

i = 0

for line in content:

if i == 0:

dataset[folderfiles[idx]] = road\_map(list(map(float, line.split(','))), [], [])

else:

dataset[folderfiles[idx]].x.append(float(line.split(',')[0]))

dataset[folderfiles[idx]].y.append(float(line.split(',')[1]))

i += 1

return dataset