




# JudgeWindow Manual

## 1) Environment setup:

0. 刪除 **project** 中你原本的 **JudgeWindow**(或其他你自創的 **class** 名字)(請記得先存檔)
1. Add class **JudgeWindow** (設計師表單類別)



2. 將助教提供的檔案(.cpp & .h)中的內容複製並覆蓋掉你新建的.cpp & .h 的內容
3. 以 **notepad++**(建議) 或記事本打開 **judgewindow.ui**(因 QT 中無法編輯), 將助教提供的.ui 中的內容複製並覆蓋掉你新建的 **judgewindow.ui** 內容 (找到你新建的.ui, 應該在你的專案目錄下)

	judgewindow.h	2020/1/10 下午 12:26	C Header File	3 KB
	judgewindow.ui	2020/1/10 下午 12:26	Qt UI file	3 KB
	lcsbstring.cpp	2019/12/15 下午 05:38	C++ Source File	3 KB

4. 記得在 **Building** 中要宣告 **JudgeWindow** 的 **Object** 當 **data member**
5. \*如果你的 **building** 中有 **setSeed()**請刪除那行程式碼

如果以上步驟做完你的程式沒有跑出紅字錯誤的話, 代表你成功了恭喜~  
有任何問題請在討論區提出

## 2) Database setup:

### 1. 改成你自己的 DB

```
//改成你自己的UserName & Password  
db.setHostName("localhost");  
db.setUserName("root");  
db.setPassword("0000");  
db.setPort(3306);  
(db.open())?QDebug()<<"JUDGE DB OK":
```

### 2. Datapath 也改成你自己的 datapath

```
QString path = "D://ProgramData//MySQL//MySQL Server 8.0//Uploads//testdata_1225.csv";
```

現在可以試著跑看看, 如果你的 ui 會跑你已經成功一半了~  
有任何問題請在討論區提出

### 3) 使用方法:

#### 1. Handshaking

以下為 `judgewindow.h` 的部分程式碼(public 部分)

```
public:

    explicit JudgeWindow(QWidget *parent = nullptr);
    ~JudgeWindow();

    string getData(int floorindex, int b, int& times);
    //floorindex : 0~26, b=(0,1):(出電梯,進電梯)
    bool submitData(string);

    qint64 getSpendTime(){return costtime;}
    QSqlDatabase& getDatabase(){return db;}
    //get intial condition by db from this function
    int getConditionNum(){return conditionNum;}
    int getDistance(){return distance;}
    int getElevatorPeople(){return this->elevactorPeople;}

    void scheduleEnd();
    //necessary to call this function when simulation is over
```

助教的程式碼不會有 `building` 去 `set judge` 的情況發生, ex `setDistance(int)`之類的, 所有的處理都是在 `judge` 中處理再以 `call function` 時 `return`

最重要的部分是請在結束 `schedule` 後記得要 `call scheduleEnd()`, 否則會造成不可預料之錯誤~

### 4) 執行方法:

根據決賽規則, 第 0 局會先空跑得到跑分結果

```
#include "judgewindow.h"
#include "ui_judgewindow.h"
#include <QSqlError>
#include <fstream>

#define FIRSTRUN 1

JudgeWindow::JudgeWindow(QWidget *parent) :
    QWidget(parent),
```

請將 `FIRSTRUN` 設為 1 (預設為 1), 之後的每次重跑請將這個值設為 0, 你的結果便會與上次執行的結果比較並統計出你的分數

\*也可以看作設為 1 為初始化, 可當作一局的開始

假設你的程式當掉, 可以直接重新執行程式並 `simulation(FIRSTRUN 為 0 的前提下)`, 程式會自動將上次當掉的樓層打勾

假設你的程式 **idle** 太久, 請按

```
gridLayout->addWidget(showline[i][j],i+1,j
```



停止(小紅框)結束, 同上可直接重新執行程式, 程式會自動將上次 **idle** 過久的樓層打勾

## 5) 計分方式

初賽計分方式已經在之前公告過了，此為再次提醒

之前每筆測資都是跑 10 次，這支程式為跑 50 次(之後可能 100~1000)，如果你的程式會因此 idle 太久請試著去加速你的演算法並先將此樓層數記下來，決賽時再打勾

由於各個測資會比時間，所以出現以下狀況不用意外：

**$((\text{Correct} * 100) \neq \text{Score})$**

Correct	Score
10	800

超重部分：

會有相對應懲罰