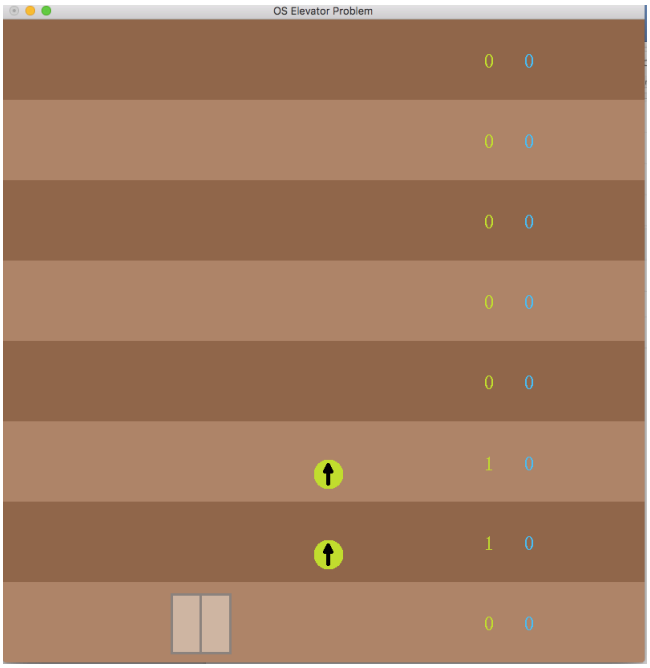


ELEVATOR CUSTOMER SCHEDULER

[文件副標題]

作業系統期末專案

第十二組

103703033 吳綠庭

103703035 賴宥安

103703047 黃彥魁

目錄

一、報告主題簡介

本組的實作題目是 Elevator Customer Schedule Problem。我們必須實作一個趨近無限快、每一層樓間只需花好 100 ticks 的電梯。當沒有人按電梯時它會處於休眠狀態，收到來自 Passenger 的通知後便會醒來並前往指定樓層載客。

而 Passenger 有兩個參數：startFloor 和 endFloor，分別代表按下電梯的樓層和欲前往的樓層。在隨機的時間點會產生隨機變數的 Passenger，電梯必須判斷現在應該前往哪一個樓層接獲送出乘客。

二、組員分工

- 103703033 吳綠庭： 主程式優化（電梯演算法優化、Poisson 隨機分佈時間生成乘客）、GUI 圖片製作
- 103703035 賴宥安 ： 主程式建構（電梯載客雛形、乘客生成雛形、Multi-Thread 管理）
- 103703047 黃彥魁 ： GUI 建構、主程式細節調整

三、執行緒(Thread)類型需求以及各自功能

依照題目要求，我們將要搭乘電梯的乘客與電梯分別做成執行緒

(1) 乘客的執行緒(void t_passenger(int startFloor,int endFloor))：

每個隨機產生出來的乘客，都各自擁有獨立的 thread，裡面做的事情依序為:按電梯(上樓或下樓)->等電梯->進電梯並按下欲前往之樓層->等待電梯到達目的地->出電梯，當此執行緒被判定為出電梯之後，就結束。

（此圖僅上樓程式碼，下樓與上樓雷同，為免冗長故不列出）

```
void t_passenger(int startFloor,int endFloor){/*{{{*/
    bool isGoingUp;
    int dataVectorPlace=0;
    if(startFloor<endFloor)
        isGoingUp=true;
    else
        isGoingUp=false;

    Data localData;
    localData.startFloor=startFloor;
    localData.endFloor=endFloor;
    localData.isGoingUp=isGoingUp;
    dataVectorPlace=dataVector.size();
    dataVector.push_back(localData);


    if(isGoingUp){
        mtx_havePeople.lock();
        havePeople_up[startFloor-1]=true;
        passengers[startFloor-1]->count++;
        std::cout<<"BORN : ["<<std::this_thread::get_id()<<"] at "<<startFloor<<" to "<<endFloor<<std::endl;  //when born
        mtx_havePeople.unlock();

        std::unique_lock <std::mutex> lck(mtx_floor);
        cvFloorIn_up[startFloor-1].wait(lck);

        mtx_destination.lock();
        destination_up[endFloor-1]=true;
        passengers[startFloor-1]->count--;
        std::cout<<"["<<std::this_thread::get_id()<<"] INNNNN!!!! from "<<startFloor<<" to "<<endFloor<<std::endl;  //when in
```

(2) 電梯的執行緒(void t_elevator()):

此執行緒是用來模擬電梯的移動透過 Shared Variables 來取得資訊並移動，而電梯會有上樓、下樓，或者是沒人按時停止不動，此執行緒並不會因為沒有乘客而結束，而是繼續等待下一個隨機產生出來的乘客。

（此圖僅上樓程式碼，下樓與上樓雷同，為免冗長故不列出）

```
void t_elevator(){/*{{{*/
    while(!elevatorIsAwake){
        //usleep(1*Time_TIMES_NUM);
        std::this_thread::sleep_for(std::chrono::microseconds(Time_TIMES_NUM));
        std::cout<<"Elevator Sleeping"<<std::endl;
    }
    std::cout<<"Elevator Awaken"<<std::endl;
    int i;
    while(peopleNum!=0){
        int targetFloor;
        targetFloor=stairSelect();
        std::cout<<"target = "<<targetFloor<<std::endl;
        if(targetFloor>nowFloor){
            upOrDown=true;
        }
        else if(targetFloor<nowFloor){
            upOrDown=false;
        }
        else if(targetFloor==nowFloor){
            cvFloorOut_up[nowFloor-1].notify_all();
            cvFloorIn_up[nowFloor-1].notify_all();
            havePeople_up[nowFloor-1]=false;
            destination_up[nowFloor-1]=false;
            std::cout<<"open door!"<<std::endl;
        }
        while(!dataVector[nowDataPlace].isDone){
            int initialFloor=nowFloor;
            if(upOrDown==true){
                for(i=0;i<targetFloor-initialFloor+1;i++){
                    mainElevator->getPos(nowFloor,upOrDown);
```

四、共有資源與共享變數(Shared Variables)需要進行 Multi-Thread (or -Process)之間的同步與合作

(1) Shared Variables:

bool havePeople_up[NumberOfFloors];

有要上樓的乘客的樓層

bool destination_up[NumberOfFloors];

上樓乘客要去的樓層

bool havePeople_down[NumberOfFloors];

有要下樓的乘客的樓層

bool destination_down[NumberOfFloors];

下樓乘客要去的樓層

以上的 Shared Variables 是為了要讓電梯與乘客去做溝通，例如:乘客在五樓想要去七樓，那麼 havePeople_up[4]就會被改成 true，進電梯後，因為已經載到人了，電梯執行緒又會把它改成 false，然後乘客執行緒就會將 destination_up[6]改成 true，電梯將人載到目的地後，再改成 false，以此類推。

(2) Mutex:

std::mutex mtx_havePeople;

為了不要讓 havePeople_up[NumberOfFloors] 和 havePeople_down[NumberOfFloors]被乘客與電梯同時修改。

std::mutex mtx_destination;

為了不要讓 destination_up[NumberOfFloors] 和 destination_down[NumberOfFloors]被乘客與電梯同時修改。

std::mutex mtx_floor;

與 condition variable 配合使用

(3) Condition Variable:

std::condition_variable cvFloorIn_up[NumberOfFloors];

notify_all()在 N 樓等上樓電梯的乘客進電梯

std::condition_variable cvFloorOut_up[NumberOfFloors];

notify_all()要上到 N 樓的乘客下電梯

std::condition_variable cvFloorIn_down[NumberOfFloors];

notify_all()在 N 樓等下樓電梯的乘客進電梯

std::condition_variable cvFloorOut_down[NumberOfFloors];

notify_all()要下到 N 樓的乘客下電梯

(4) Multi-Thread:

電梯先等待乘客，一有乘客之後，就取得 data 並且前往載客，並用 Condition Variable 來告訴乘客何時該進出電梯。

五、關鍵區域程式（使用 C++）

t_Passenger 底下，每個時間點只能有一個 passenger 按電梯

```
if(isGoingUp){
    mtx_havePeople.lock();
    havePeople_up[startFloor-1]=true;
    passengers[startFloor-1][0]->setIO(0,isGoingUp);
    std::cout<<"["<<std::this_thread::get_id()<<"] (from "<<startFloor<<" to "<<endFloor<<") \033[30;42mpressed the elevator.\033[m"<<std::endl;  //wh
orn
    mtx_havePeople.unlock();

    std::unique_lock <std::mutex> lck(mtx_floor);
    cvFloorIn_up[startFloor-1].wait(lck);

    mtx_destination.lock();
    destination_up[endFloor-1]=true;
    passengers[startFloor-1][0]->setIO(1,isGoingUp);
    std::cout<<"["<<std::this_thread::get_id()<<"] (from "<<startFloor<<" to "<<endFloor<<") \033[30;43mget in .\033[m"<<std::endl;  //when in
    dataVector[dataVectorPlace].isIn=true;
    mtx_destination.unlock();
```

t_Elevator 底下，passenger 出電梯後電梯要改變樓層狀態，所以同

一時間點不能有 passenger 按電梯

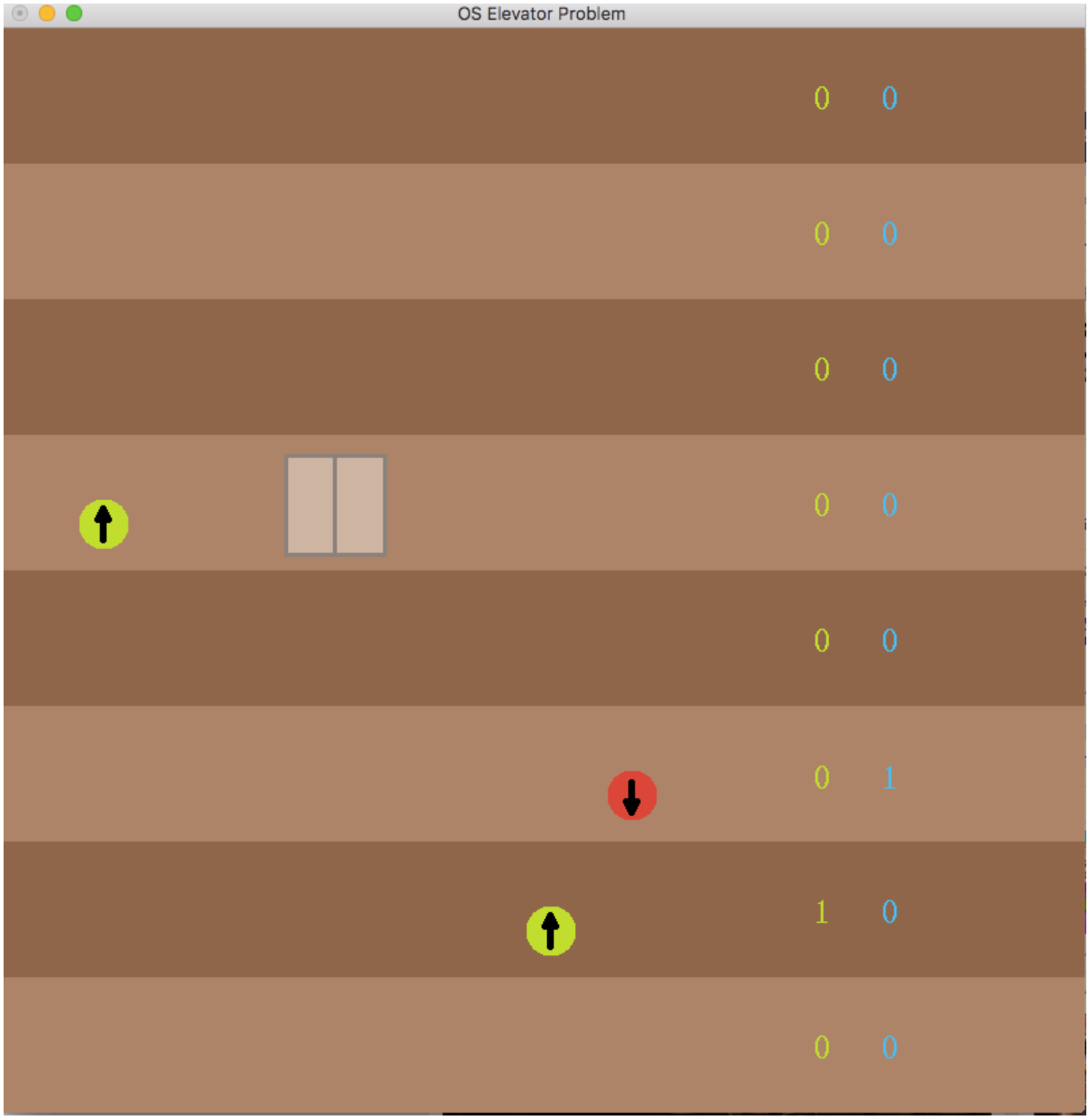
```
327         while(!dataVector[nowDataPlace].isDone){
328             int initialFloor=nowFloor;
329             if(upOrDown==true){
330                 for(i=0;i<targetFloor-initialFloor+1;i++){
331                     mainElevator->getPos(nowFloor,upOrDown);
332                     std::cout << "\033[1;31melevator arrive at "<<nowFloor<<"\033[m"<<std::endl;
333                     mtx_havePeople.lock();mtx_destination.lock();
334                     if(havePeople_up[nowFloor-1]==true || destination_up[nowFloor-1]==true){
335                         cvFloorOut_up[nowFloor-1].notify_all();
336                         cvFloorIn_up[nowFloor-1].notify_all();
337                         havePeople_up[nowFloor-1]=false;
338                         destination_up[nowFloor-1]=false;
339                         //std::cout<<"open door!"<<std::endl;
340                     }
341                     mtx_havePeople.unlock();mtx_destination.unlock();
```

六、文字顯示狀態與結果

```
Elevator is waiting...
Elevator is waiting...
Elevator is waiting...
[0x70000e83b000] (from 3 to 4) pressed the elevator.
Elevator is waiting...
elevator arrive at 1
elevator arrive at 2
[0x70000e62f000] (from 1 to 5) pressed the elevator.
elevator arrive at 3
[0x70000e83b000] (from 3 to 4) get in .
elevator arrive at 3
[0x70000e8be000] (from 3 to 4) pressed the elevator.
elevator arrive at 4
[0x70000e83b000] (from 3 to 4) come out.
elevator arrive at 4
[0x70000e941000] (from 5 to 2) pressed the elevator.
elevator arrive at 3
elevator arrive at 2
[0x70000e9c4000] (from 4 to 2) pressed the elevator.
elevator arrive at 1
elevator arrive at 1
[0x70000e62f000] (from 1 to 5) get in .
elevator arrive at 2
[0x70000ea47000] (from 2 to 6) pressed the elevator.
elevator arrive at 3
[0x70000e8be000] (from 3 to 4) get in .
elevator arrive at 4
[0x70000e8be000] (from 3 to 4) come out.
elevator arrive at 5
[0x70000e62f000] (from 1 to 5) come out.
elevator arrive at 5
[0x70000eb4d000] (from 2 to 1) pressed the elevator.
elevator arrive at 5
[0x70000e941000] (from 5 to 2) get in .
elevator arrive at 4
[0x70000e9c4000] (from 4 to 2) get in .
[0x70000ebd0000] (from 4 to 5) pressed the elevator.
elevator arrive at 3
```

七、GUI 呈現

ＧＵＩ一開始試圖使用 Qt 實作，但在主程式出來後發現不大能把 std::thread 接到 Qt 上，也許是對 Qt 比較不熟的緣故，在試了幾次後放棄改使用 OpenGL



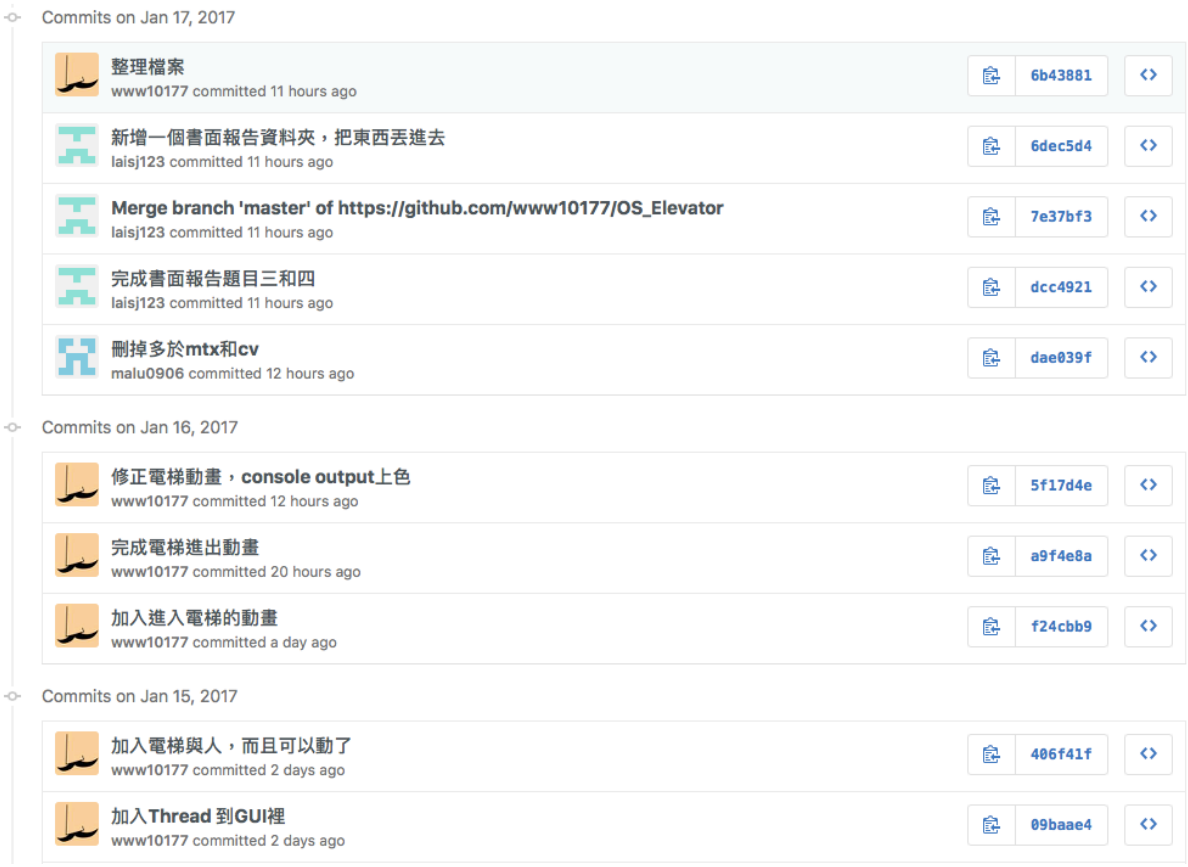
八、參考資料

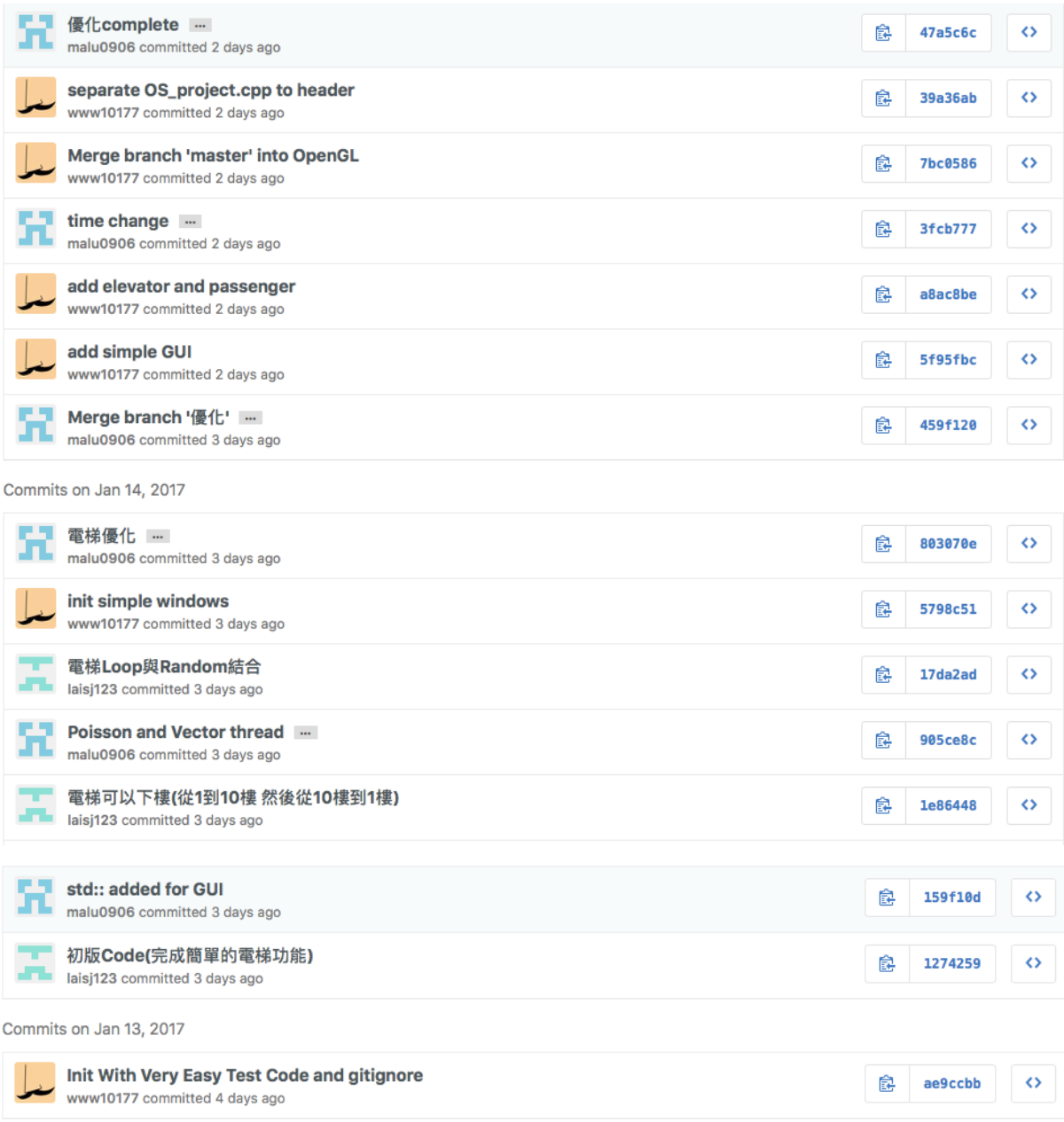
Poisson Distribution

C++ 的多執行序程式開發 Thread：基本使用

C++11 并发指南一(C++11 多线程初探)

九、Github




優化complete
malu0906 committed 2 days ago
47a5c6c
separate OS_project.cpp to header
www10177 committed 2 days ago
39a36ab
Merge branch 'master' into OpenGL
www10177 committed 2 days ago
7bc0586
time change
malu0906 committed 2 days ago
3fcb777
add elevator and passenger
www10177 committed 2 days ago
a8ac8be
add simple GUI
www10177 committed 2 days ago
5f95fbc
Merge branch '優化'
malu0906 committed 3 days ago
459f120
Commits on Jan 14, 2017
電梯優化
malu0906 committed 3 days ago
803070e
init simple windows
www10177 committed 3 days ago
5798c51
電梯Loop與Random結合
laisj123 committed 3 days ago
17da2ad
Poisson and Vector thread
malu0906 committed 3 days ago
905ce8c
電梯可以下樓(從1到10樓 然後從10樓到1樓)
laisj123 committed 3 days ago
1e86448
std:: added for GUI
malu0906 committed 3 days ago
159f10d
初版Code(完成簡單的電梯功能)
laisj123 committed 3 days ago
1274259
Commits on Jan 13, 2017
Init With Very Easy Test Code and gitignore
www10177 committed 4 days ago
ae9ccbb