



作業五：

電影 ”返校” 劇情設計 (地圖化)

資工系 2B A7223598 蔡欣叡

- ▶ 授權文化大學資工系觀摩參考
- ▶ 課程：網路/視窗程式設計
- ▶ 授課教師：郭宗翰

HomeWork5

返校劇情 地圖化

初期:不知道劇情能寫成甚麼，就突然想可以做一個小地圖可對話的小程式

一開始設計地圖

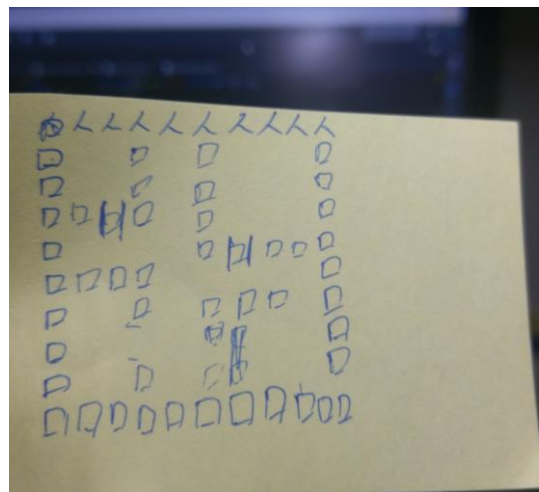
如果有想到需要加上的功能就先加到專案檔上記住
遇到出現的問題還沒解決也可以先記錄在上面

```
//BackToSchoolHomeWork    **testing playground port3 測試服
public class GameStart extends JFrame{
    //java如何偵測鍵入 按件    (完成)
    //做一個輸出圖作文地圖 操做一次進行更新    (完成)
    //設計按下一個按鍵進行對話 (必須要在腳色面前) (完成)
    //玩家如果想撞牆進行限制    (完成)
    //教官說完話之後就用罐頭話    (完成)

    //更改ctrltalk 修正沒偵測到時 從true改成false    (完成)
    //將版本改為 someone screen 版本 (完成)

    //修正玩家可以位移吃掉人物    (完成)

    static Boolean controlLeft=false;
    static Boolean controlRight=false;
    static Boolean controlUp=false;
    static Boolean controlDown=false;
    static Boolean controlED=false;
    static Boolean controlTalk=false;
    static Boolean TheKey=false;
    static Boolean Thedead=false;
```



A7223598_蔡欣叡_資工2B

耗時 19/11/1 0:00am~3am 3HR
11/1 10:00am~12:30pm 2.5hr
11/3 10:25am~1:25pm 3hr

```
for(int i=0;i<10;i++) {
    for(int j=0;j<10;j++) {
        if(i==0 || j==0 || i==9 || j==9)
            a[i][j]="□";
        else
            a[i][j]=" ";
    }
}
a[1][3]="□";
a[2][3]="□";
a[3][3]="□";
a[3][1]="□";
```

繪製圍牆
地圖細節

```
a[1][1]="人";

a[1][5]="□";
a[2][5]="□";
a[3][5]="□";
a[4][5]="□";
a[4][7]="□";
a[4][8]="□";
a[2][8]="人";

a[5][1]="□";
a[5][2]="□";
a[5][3]="□";
a[6][3]="□";
a[8][3]="□";
a[8][1]="人";

a[6][5]="□";
a[6][6]="□";
a[6][7]="□";
a[7][5]="□";
a[8][5]="□";
a[7][6]="人";
```

參考資料

1. 如何使用java delay 在網路上實現的發法很多

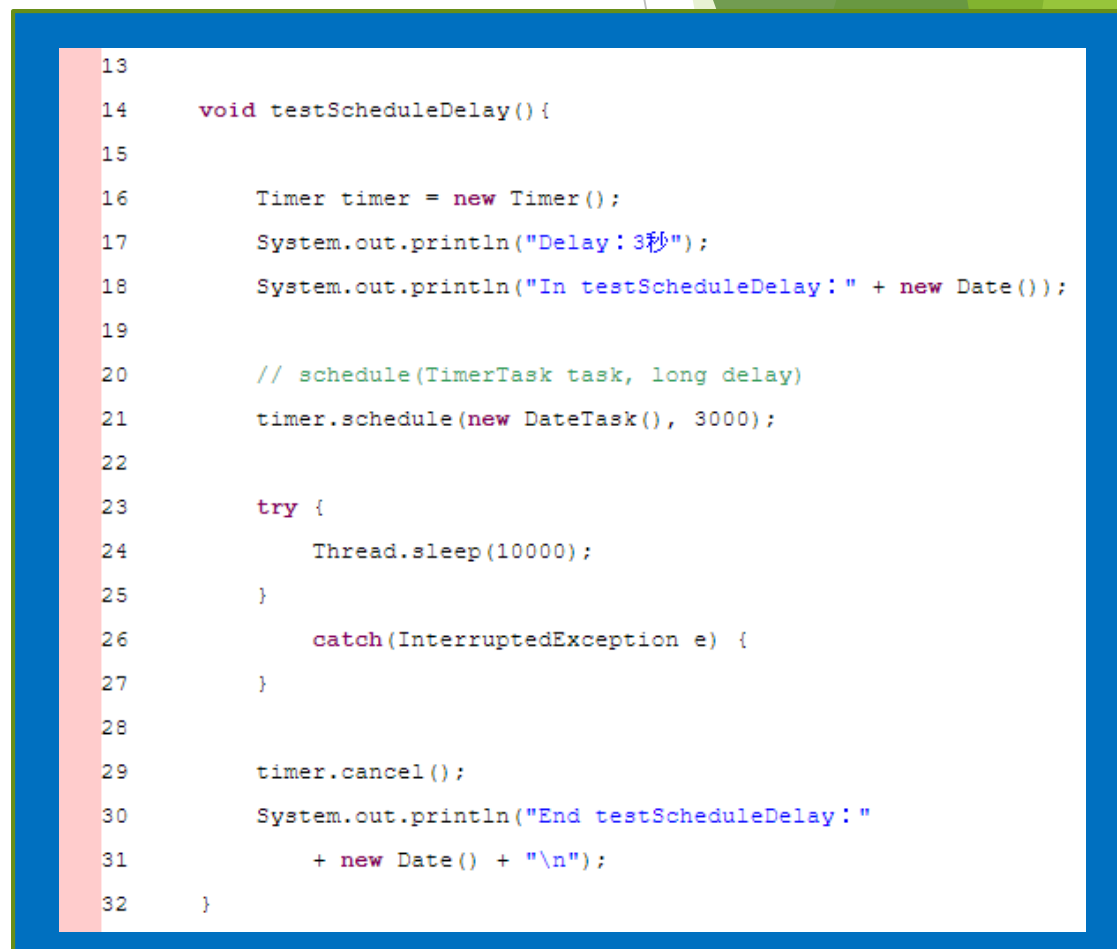
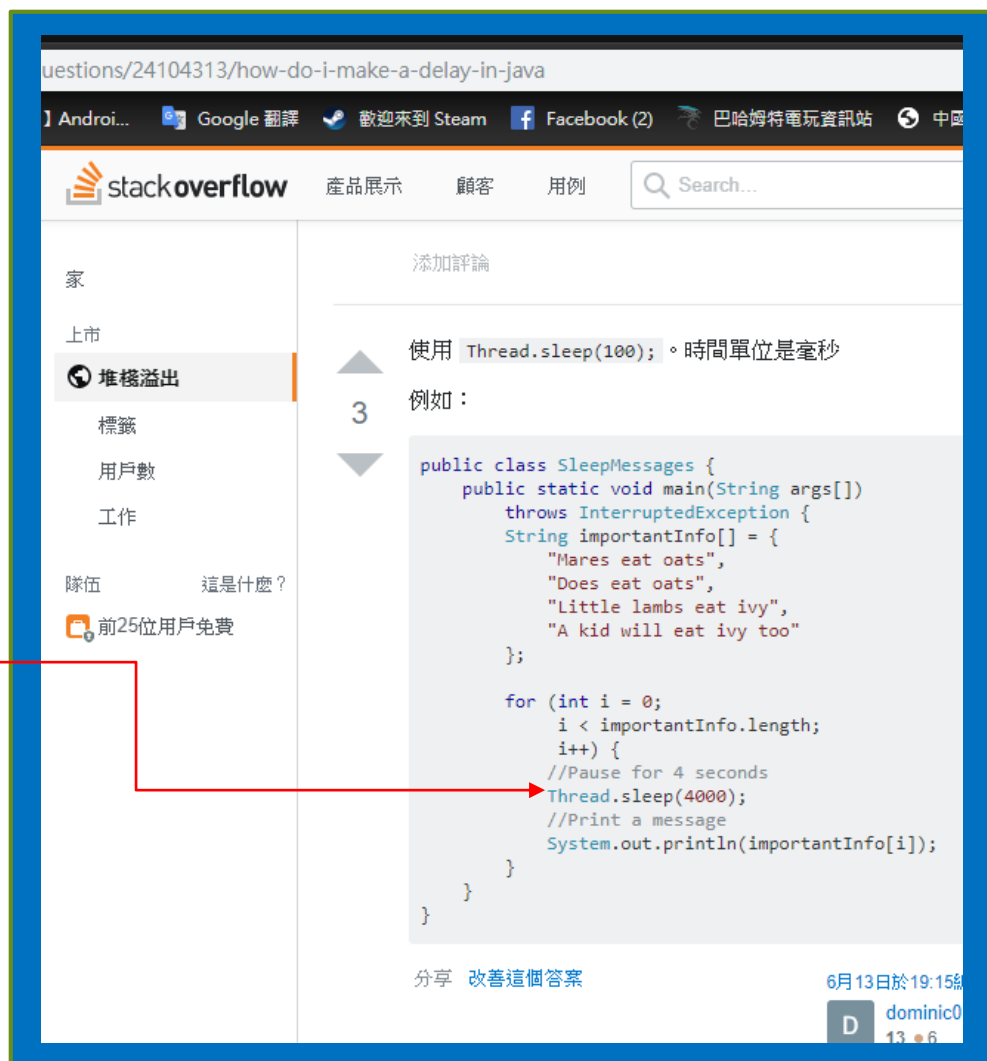
2. Java 控制視窗/監視視窗

<https://stackoverflow.com/questions/24104313/how-do-i-make-a-delay-in-java>

https://blog.csdn.net/qq_36761831/article/details/81545050

網路上可以收尋到不同的delay用法(左圖)與(右圖)，這次我使用左圖方法

因為不確定確切的語法，所以上網查並抓取需要的程式碼



1. 如何使用java delay

所有自己不確定不會使用的程式碼最好開一個獨立的java檔進行測試，確定效果可行在調用到我們正在做的專案上，以免錯誤要在主專案中挑骨頭

```
3
4
5 public class Main {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         String[][] a=new String[10][10];
10
11
12         a[0][0]="人";
13         if(a[0][0]=="人") {
14             System.out.println("is same");
15         }
16         else {
17             System.out.println("is not same ");
18         }
19
20         while(true) {
21             System.out.println("yes");
22             Thread.sleep(100);
23         }
24     }
25 }
```

```
4
5 public class Main {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         String[][] a=new String[10][10];
10
11
12         a[0][0]="人";
13         if(a[0][0]=="人") {
14             System.out.println("is same");
15         }
16         else {
17             System.out.println("is not same ");
18         }
19
20         while(true) {
21             System.out.println("yes");
22             Thread.sleep(100);
23         }
24     }
25 }
```

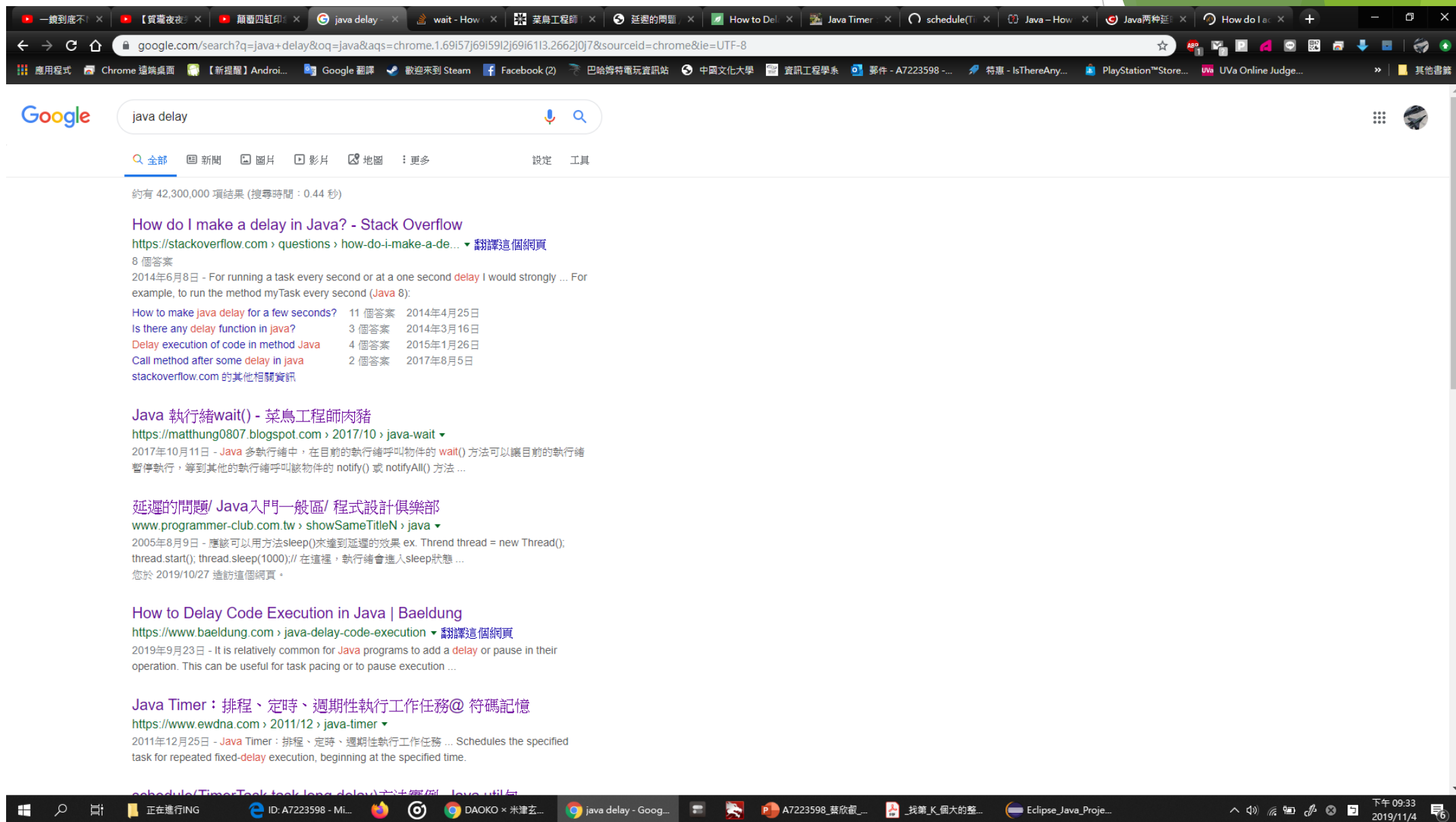
Unhandled exception type InterruptedException
2 quick fixes available:
Add throws declaration
Surround with try/catch

```
4
5 public class Main {
6
7     public static void main(String[] args) throws InterruptedException {
8         // TODO Auto-generated method stub
9         String[][] a=new String[10][10];
10
11
12         a[0][0]="人";
13         if(a[0][0]=="人") {
14             System.out.println("is same");
15         }
16         else {
17             System.out.println("is not same ");
18         }
19
20         while(true) {
21             System.out.println("yes");
22             Thread.sleep(100);
23         }
24     }
25 }
```

```
1 package com.pccu.tsai.yoo.testing;
2
3
4
5 public class Main {
6
7     public static void main(String[] args) throws InterruptedException {
8         // TODO Auto-generated method stub
9         String[][] a=new String[10][10];
10
11
12         a[0][0]="人";
13         if(a[0][0]=="人") {
14             System.out.println("is same");
15         }
16         else {
17             System.out.println("is not same ");
18         }
19
20         while(true) {
21             System.out.println("yes");
22             Thread.sleep(100);
23         }
24     }
25 }
```


2.Java 控制視窗/監視視窗

查詢自己需要的資訊，查詢結果通常很多網頁中只有一個是你要多開尋找，英文版面也沒差chrome會翻譯，英文通常也只是解釋直接看程式碼即可



2.Java 控制視窗/監視視窗

https://blog.csdn.net/qq_36761831/article/details/81545050

網路上的程式碼會像他網頁上執行的樣子一樣，按下什麼按鍵就sysout什麼，依造我的需求只要有監視的這個視窗，也不要他sysout東西會破壞我地圖的流暢性，可不希望玩到一半中間一直穿插”方向鍵右以按下”，所以測試完網路上的程式碼後開始修改成我要的功能

```
public class t3 extends JFrame{

    public t3() {
        setTitle("Hern");
        setBounds(400, 400, 400, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        final JLabel label = new JLabel();
        label.setText("請輸入");
        add(label, BorderLayout.WEST);

        final JScrollPane scrollPane = new JScrollPane();
        add(scrollPane, BorderLayout.CENTER);
        JTextArea textArea = new JTextArea();
        textArea.addKeyListener(new KeyListener() {

            public void keyPressed(KeyEvent e) { // 按鍵被按下時觸發

                String keyText = KeyEvent.getKeyText(e.getKeyCode()); // 獲得描述keyCode的標籤
                if (e.isActionKey()) { // 判斷按下的是否為動作鍵

                    System.out.println("您按下的是動作鍵" + keyText + "");

                } else {

                    System.out.print("您按下的是非動作鍵" + keyText + "");
                    int keyCode = e.getKeyCode(); // 獲得與此事件中的鍵相關聯的字符

                    switch (keyCode) {
                        case KeyEvent.VK_CONTROL: // 判斷按下的是否為Ctrl 鍵
                            System.out.print(", Ctrl鍵被按下");
                            break;
                        case KeyEvent.VK_ALT: // 判斷按下的是否為alt 鍵
                            System.out.print(", Alt鍵被按下");
                            break;
                        case KeyEvent.VK_SHIFT: // 判斷按下的是否為Shift 鍵
                            System.out.print(", Shift鍵被按下");
                            break;
                    }
                    System.out.println();
                }
            }
        })
    }
}
```

```
public GameStart() { //控制視窗 {網路上抓取進行 控制執行修改}
    setTitle("Hern");
    setBounds(400, 400, 400, 400);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    final JLabel label = new JLabel();
    label.setText("請輸入");
    add(label, BorderLayout.WEST);

    final JScrollPane scrollPane = new JScrollPane();
    add(scrollPane, BorderLayout.CENTER);
    JTextArea textArea = new JTextArea();
    textArea.addKeyListener(new KeyListener() {

        public void keyPressed(KeyEvent e) { //按鍵被按下時觸發

            String keyText = KeyEvent.getKeyText(e.getKeyCode()); //獲得描述keyCode的標籤
            if (e.isActionKey()) { //判斷按下的是否為動作鍵

                System.out.println("你按下的是動作鍵" + keyText + "");
                int keyCode = e.getKeyCode();
                switch (keyCode) { //修改部分*****
                    case KeyEvent.VK_LEFT:
                        controlLeft=true;
                        break;
                    case KeyEvent.VK_RIGHT:
                        controlRight=true;
                        break;
                    case KeyEvent.VK_UP:
                        controlUp=true;
                        break;
                    case KeyEvent.VK_DOWN:
                        controlDown=true;
                        break;
                    //*****
                }
            }
        }
    })
}
```

Java 控制視窗/監視視窗 與 主程式互動

Gamestart.java

```
static Boolean controlLeft=false;
static Boolean controlRight=false;
static Boolean controlUp=false;
static Boolean controlDown=false;
static Boolean Thedead=false;
```

主偵測迴圈

玩家死前都不會跳出

專案程式碼(玩家向左)

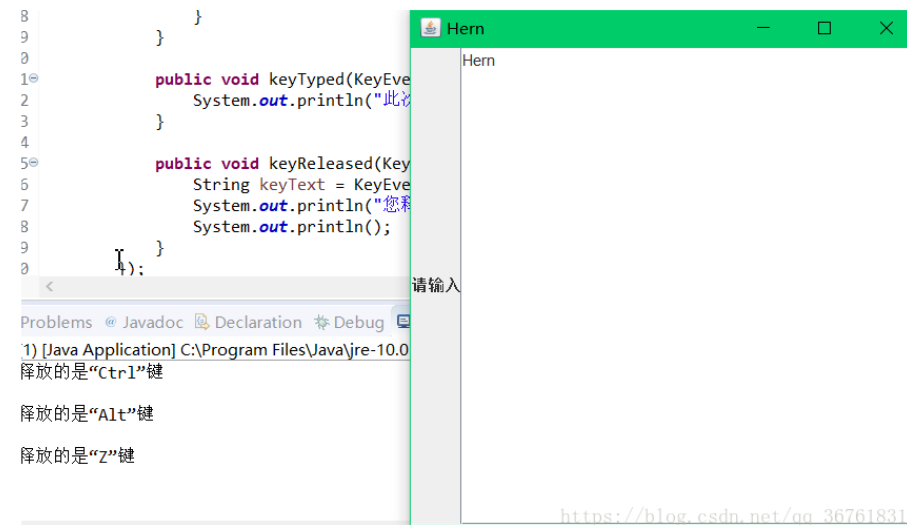
```
While(Thedead!=true){
    if(controlLeft==true)
        玩家向左;
        controlLeft=false;
    else if(controlRight==true)
        玩家向右;
        controlRight=false;
    else if(controlUp==true)
        玩家向上;
        controlUp=false;
    else if(controlDown==true)
        玩家向下;
        controlDown=false;
}
```

鍵盤 監視視窗

```
int keyCode = e.getKeyCode();
switch (keyCode) {
    case KeyEvent.VK_LEFT:
        controlLeft=true;
        break;
    case KeyEvent.VK_RIGHT:
        controlRight=true;
        break;
    case KeyEvent.VK_UP:
        controlUp=true;
        break;
    case KeyEvent.VK_DOWN:
        controlDown=true;
        break;
}
```

移動程式執行完進行Boolean復位

鍵盤 監視視窗



```
String keyText = KeyEvent.getKeyText(e.getKeyCode()); //獲得描述keyCode的標籤
if (e.isActionKey()) { //判斷按下的是否為動作鍵

    System.out.println("你按下的是動作鍵" + keyText + "");
    int keyCode = e.getKeyCode();
    switch (keyCode) { //修改部分*****
        case KeyEvent.VK_LEFT:
            controlLeft=true;
            break;
        case KeyEvent.VK_RIGHT:
            controlRight=true;
            break;
        case KeyEvent.VK_UP:
            controlUp=true;
            break;
        case KeyEvent.VK_DOWN:
            controlDown=true;
            break;
    }
}
```

人物說話腳本

If(controlTalk==true)
執行右圖

```
SomeOne StudentName =new SomeOne();  
StudentName.name=" 學生";  
Screen Student = new Screen() { //人物說話腳本
```

```
int studentTalk=0;  
@Override  
public void Character(SomeOne so) {  
    // TODO Auto-generated method stub  
    System.out.println(so.name);  
}
```

```
@Override  
public void PassTalk() {  
    // TODO Auto-generated method stub
```

```
if(studentTalk==0) {  
    System.out.println("我這有一把鑰匙給你");  
    System.out.println("++獲得一把鑰匙++");  
    TheKey=true;  
}  
else if(studentTalk==1){  
    System.out.println("還有不要告訴教官我在這喔");  
}  
else {  
    System.out.println("萬事拜託你了");  
}  
studentTalk++;  
}
```

```
};
```

```
}else if(x-1==1 && y==1) { //unknown位置  
    unknow.Character(UnknowName);  
    unknow.PassTalk();  
}  
else if(x-1==0 && y==4) {  
    keyControl.PassTalk();  
}  
controlTalk=false;  
Thread.sleep(1500);  
}  
else if(a[x+1][y]=="人"){ //downdowndown  
    if(x+1==8 && y==1) { //學生位置  
        Student.Character(StudentName);  
        Student.PassTalk();  
    }  
    else if(x+1==7 && y==6) { //教官位置  
        Instructor.Character(InstructorName);  
        Instructor.PassTalk();  
    }  
    else if(x+1==2 && y==8) { //老師位置  
        Teacher.Character(TeacherName);  
        Teacher.PassTalk();  
    }  
    else if(x+1==1 && y==1) { //unknown位置  
        unknow.Character(UnknowName);  
        unknow.PassTalk();  
    }  
    controlTalk=false;  
    Thread.sleep(1500);  
}  
else {  
    controlTalk=false;  
}
```

//studentTalk標記是否有跟玩家交談過，說過的話不再重複

int studentTalk=0;

標記是否有跟玩家交談過
先前說過的話不再重複

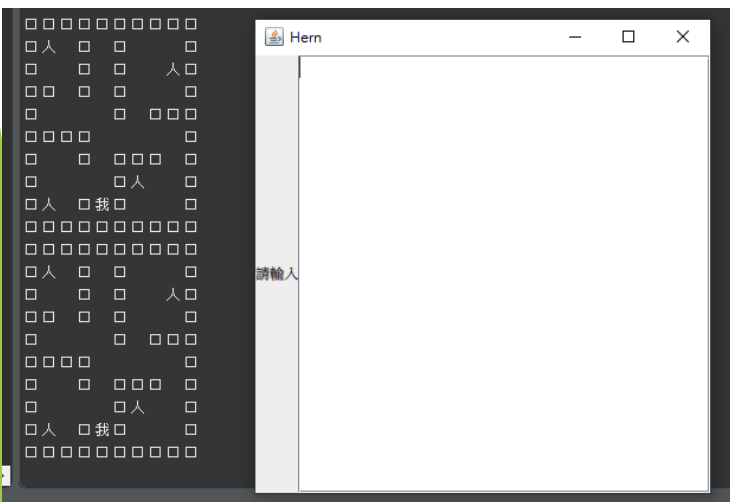
對角色進行二次對話(甚至是做到第三、四、五次對話)時就會顯示不同台詞，腳色劇情說完就可做成最後重複說得罐頭話

地圖以矩陣畫出
接收到controlTalk後if進行上、下、左、右的"人"進行檢查確定有人後，才再下去if找出這個人是誰"教官"or"老師"or"學生"，確定是學生在玩家面前後執行學生說話腳本

控制視窗

監視鍵盤輸入如果按下
方向鍵右，將boolean
controlRight改成true
Ctrl則觸發對話(如果旁邊有
人)

後面的主要的while迴圈就會
收到controlRight=true的訊
息執行特定的程式碼



```
42
43 public GameStart() { //控制視窗 {網路上抓取進行 控制執行修改}
44     setTitle("Hern");
45     setBounds(400, 400, 400, 400);
46     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
47
48     final JLabel label = new JLabel();
49     label.setText("請輸入");
50     add(label, BorderLayout.WEST);
51
52     final JScrollPane scrollPane = new JScrollPane();
53     add(scrollPane, BorderLayout.CENTER);
54     JTextArea textArea = new JTextArea();
55     textArea.addKeyListener(new KeyListener() {
56
57         public void keyPressed(KeyEvent e) { //按鍵被按下時觸發
58
59             String keyText = KeyEvent.getKeyText(e.getKeyCode()); //獲得描述keyCode的標籤
60             if (e.isActionKey()) { //判斷按下的是否為動作鍵
61
62                 // System.out.println("你按下的是動作鍵" + keyText + "");
63                 int keyCode = e.getKeyCode();
64                 switch (keyCode) { //修改部分*****
65                     case KeyEvent.VK_LEFT:
66                         controlLeft=true;
67                         break;
68                     case KeyEvent.VK_RIGHT:
69                         controlRight=true;
70                         break;
71                     case KeyEvent.VK_UP:
72                         controlUp=true;
73                         break;
74                     case KeyEvent.VK_DOWN:
75                         controlDown=true;
76                         break;
77                     //*****
78                 }
79             }
80             else {
```

int x=8;

int y=4;

設定玩家開始位置在外部，
讓不同需要玩家位置的程式
碼都可以取得目前玩家的位
置

1. 移動
2. 偵測

307行的while迴圈為主要的
迴圈，只重複進行偵測是否
有更新成為true的指令
與更新地圖

```
304 int x=8; //玩家開始位置
305 int y=4; //位置設定在外部使 移動 偵測都可以取得玩家位置進行解算
306
307 while(TheDead!=true) { //a[8][4]="我"; //while迴圈 如果接收到死亡就跳出while
308     if(controlED==true && controlLeft==true && a[x][y-1]!="□" && a[x][y-1]!="人") { //寫
309
310         a[x][y-1]="我";
311         a[x][y]=" ";
312         y--;
313         controlLeft=false;
314         mapPrint(a);
315     }
316
317     else if(controlRight==true && a[x][y+1]!="□" && a[x][y+1]!="人") {
318         controlED=true;
319         a[x][y+1]="我";
320         a[x][y]=" ";
321         y++;
322         controlRight=false;
323         mapPrint(a);
324     }
325     else if(controlUp==true && a[x-1][y]!="□" && a[x-1][y]!="人") {
326         controlED=true;
327         a[x-1][y]="我";
328         a[x][y]=" ";
329         x--;
330         controlUp=false;
331         mapPrint(a);
332     }
333     else if(controlDown==true && a[x+1][y]!="□" && a[x+1][y]!="人") {
334         controlED=true;
335         a[x+1][y]="我";
336         a[x][y]=" ";
337         x++;
338         controlDown=false;
339         mapPrint(a);
340     }
}
```

最後加入else進行所有control bool=false，原因在於可能按下了方向鍵右，ControlRight變更為true但是在玩家右邊有"口"作為牆壁而指令被if擋下，但true仍然還在會造成下次移動時，如果玩家往上，右邊又沒有牆時("口")，會一次移動兩次，向上又向右

```
else if(controlRight==true && a[x][y+1]!="口" && a[x][y+1]!="人") {
    controlED=true;
    a[x][y+1]="我";
    a[x][y]=" ";
    y++;
    controlRight=false;
    mapPrint(a);
}
```

按下ctrl對話鍵監視視窗將controlTalk設為true，這裡將會檢測為true後並偵測玩家附近是否有npc有的話開啟對應的對話

```
344 Thread.sleep(1000);
345 controlED=false;
346 }else { //如果都沒執行到if 進行else 所有命令歸位false 未執行造成順移
347     controlLeft=false;
348     controlRight=false;
349     controlUp=false;
350     controlDown=false;
351 }
352 for(int i=0;i<10;i++) { //重複進行地圖更新 讓畫面接近及時制
353     for(int j=0;j<10;j++) {
354         System.out.print(a[i][j]);
355     }
356     System.out.println("");
357 }
358 if(controlTalk==true) { //so1.name.equals("rg") //按下ctrl進行對話 如果附近是對的人就找對的人的腳本進行!
359     if(a[x][y-1]=="人"){ //辨識語法問題 而外開一個進行測試 {成功} leftleftleftleftleftlef
360         if(x==8 && y-1==1) { //學生位置
361             Student.Character(StudentName);
362             Student.PassTalk();
363         }else if(x==7 && y-1==6) { //教官位置
364             Instructor.Character(InstructorName);
365             Instructor.PassTalk();
366         }else if(x==2 && y-1==8) { //老師位置
367             Teacher.Character(TeacherName);
368             Teacher.PassTalk();
369         }else if(x==1 && y-1==1) { //unknow位置
370             unknow.Character(UnknowName);
371             unknow.PassTalk();
372         }
373         controlTalk=false;
374         Thread.sleep(1500);
375     } //做CTRL 離開對話
376     else if(a[x][y+1]=="人"){ //rightrightrightrightrightrightrightrightrightright
377         if(x==8 && y+1==1) { //學生位置
378             Student.Character(StudentName);
379             Student.PassTalk();
380         }else if(x==7 && y+1==6) { //教官位置
381             Instructor.Character(InstructorName);
382             Instructor.PassTalk();
```

對話延遲讓玩家有時間閱讀

指令完成讓Boolean值復位

While迴圈如果接收到
deadBool跳出迴圈執行後
續程式

```
402         }else if(x-1==1 && y==1) { //unknow位置
403             unknow.Character(UnknowName);
404             unknow.PassTalk();
405         }else if(x-1==0 && y==4) {
406             keyControl.PassTalk();
407         }
408         controlTalk=false;
409         Thread.sleep(1500);
410     }else if(a[x+1][y]==("人")){ //downdowndowndowndown
411         if(x+1==8 && y==1) { //學生位置
412             Student.Character(StudentName);
413             Student.PassTalk();
414         }else if(x+1==7 && y==6) { //教官位置
415             Instructor.Character(InstructorName);
416             Instructor.PassTalk();
417         }else if(x+1==2 && y==8) { //老師位置
418             Teacher.Character(TeacherName);
419             Teacher.PassTalk();
420         }else if(x+1==1 && y==1) { //unknow位置
421             unknow.Character(UnknowName);
422             unknow.PassTalk();
423         }
424         controlTalk=false;
425         Thread.sleep(1500);
426     }else {
427         controlTalk=false;
428     }
429 }
430 Thread.sleep(100); //延遲降低系統負荷 移除測試（未進行）
431 // ??? 老師
432 // 學生甲 教官
433 //學生甲 給你一把
434 }
435 System.out.println("YOU DIED");
436 System.out.println("Game Over");
437 }
438 }
439
440
```


Screen 把function 改成比較好順口的名字， SomeOne沒進行修改

```
GameStart.java | Screen.java x | SomeOne.java
1 package com.pccu.tsai.backtoschool;
2
3 public interface Screen {
4
5     void Character(SomeOne so);
6
7     void PassTalk();
8 }
9
```

```
GameStart.java | Screen.java | SomeOne.java x
1 package com.pccu.tsai.backtoschool;
2
3 public class SomeOne {
4     String name="";
5 }
6
```

影片範例

<https://youtu.be/OppDMxneRDc>

