**2018兩岸交流大作業設計報告**

我的設計最外層總共有4個包，分別為general.uidata 、maingame、server、client。以下分別講述各個包裡class的用途。

1. **general.uidata**

這個包裡放的是兩個class：DynamicUIData跟LocalUIData，顧名思義，DynamicUIData用來存放需要透過server傳遞的資料，而LocalUIData則是用來存放本地資料。

1. 在DynamicUIData裡，我把資料分成下面的類型：

|  |  |  |
| --- | --- | --- |
|  | 不更新(只從server傳到client一次) | 會更新 |
| 兩client相同 | (1)牆壁位置  (2)蛇洞位置 | (1)食物位置  (2)食物是否被吃掉  (3)蛇的速度 |
| 兩client不同 | (1)玩家序號(判斷你是哪一個玩家)  (2)是否開始(只從client傳到server一次) | (1)蛇的身體位置  (2)蛇移動方向  (3)分數  (4)生命  (5)是否死亡 |

1. 而在LocalUIData裡，則存放其他所有遊戲會用到的資料，包括：
2. 遊戲版面的寬高及一格的單位(寬高都為40格，一格單位為20，蛇每一動一次就走一格)
3. 牆壁寬度
4. 定義{上下左右}分別為{-1 1 2 -2}，用來判斷方向
5. 字體
6. 用來切換panel的cardLayout
7. 三個遊戲的panel(開始、遊戲、結束)
8. 用來管理三個遊戲panel的panel(控制cardLayout轉換)
9. 遊戲panel裡所有的東西(JLabel、JTestField、判斷蛇身狀態的boolean值、蛇身長度、)
10. 玩家序號、對手序號(接收server傳來的資料並更新本地資料時使用)
11. 判斷是否結束
12. ObjectOutputStream用來輸出資料給server
13. ObjectInputStream用來接收server傳出的資料

**二、maingame**

這個包裡放的是兩個class：GameClient跟GameServer，分別為client和server的main函式所在。

1. GameClient繼承JFrame，建構式裡執行下列事情：
2. 設定JFrame的名字為”Snakesss”
3. 創建cardLayout跟pane(控制panel切換)
4. 更改執行JFrame左上角的圖示

main函式裡建立以下物件：

1. DynamicUIData
2. LocalUIData
3. StartPanel(遊戲開始畫面)
4. Client(接收server傳來的資料並更新本地資料)
5. GameServer裡只有一個main函式，用來取得server所在的IP位址並使用JOptionPane來告訴用戶IP位址是多少，之後創建一個Server對象並讓它開始運作。

**三、server**

server的部分我是依照助教講解的方法來實踐，總共有四個類，Server、Initialization、Updater、Broadcaster。

1. Server

Server繼承JFrame，建構式裡設定JFrame的一些參數，並在JFrame中顯示”Server running. To shut down server, close this window”，告訴用戶server已經在執行了，如果要關掉server就要關掉這個視窗。

在執行的函式(run())中，會建立一個ServerSocket的對象，並且待命接收client的連接，如果接收到2個client就會將這兩個client生成一個group(ArrayList<Sockets>類)，指向分別與兩個client連接的socket，並開線程讓這兩個client執行遊戲。

1. Initialization

Initialization有一個內部類FoodThread用來判斷如果兩個蛋都被吃完了，隔兩秒鐘後就再次生成兩個蛋。剛開始我在寫的時候就一直讓它在迴圈裡運行，但後來發現這樣在資料還沒跟新之前它就會再判斷，導致蛋會在短時間內生成好幾次，JFrame會不段的閃爍蛋的位置。為了解決這個問題，我讓它生成完蛋之後sleep 200m，這樣就不會一直重複判斷了。

Initialization最主要的兩個函式為initializeUIData()與start()。

initializeUIData用來初始化遊戲介面：

1. 初始化牆壁位置，利用亂數初始化牆壁位置，共有兩道牆，一直一橫，每道牆是用point來儲存，長度都是15格(遊戲版面寬高都為40格)，根據點的位置來判斷牆的位置。
2. 初始化蛇洞的位置，利用亂數初始化蛇洞的位置，共有兩個蛇洞，蛇洞不會在牆上也不會重疊。
3. 初始化兩條蛇的蛇身位置，兩條蛇的位置分別位於兩個蛇洞，遊戲一開始蛇就從蛇洞出來。
4. 初始化蛇蛋的位置，利用亂數初始化蛇蛋的位置，並且判斷蛇蛋不在牆上、蛇蛋不和蛇頭重合、蛇蛋不和蛇洞重合、兩顆蛇蛋本身不重合。
5. 初始化其他DynamicUIData裡的值

start用來執行初始化的動作(呼叫initializeUIData)並啟動Updater、Broadcaster和FoodThread。start會先把一開始初始化的遊戲介面傳給兩個client， Updater線程會創立兩個，分別用來更新兩個client傳來的資料，創建的匙後會傳一個參數i給Updater讓它知道它接收的是哪個client傳來的資料。

1. Updater

Updater繼承線程。一次遊戲會有兩個Updater，分別用來更新兩個client傳來的資料。updater會一直讀取client傳的資料，並用此資料更新server端的dynamicUIData。當所連接的client斷線時，updater會用JOptionPane告訴用戶哪一個client已經斷線了。

1. BroadcastTimer

BroadcastTimer裡除了建構式之外，只有一個start()函式，會由Initialization呼叫。start裡會開始一個Timer，這個Timer是用來將server的資料傳給client，每隔20ms會傳一次，要兩個玩家都開始後才會開始傳資料。

**四、client**

client包裡有一個controller包跟一個Client類，還有三個遊戲介面：StartPanel、PlayPanel、EndPanel類。

1. controller包

controller包裡有兩個類：controllerP1、controllerP2，兩者都繼承KeyAdapter類，分別用來添加到第一個客戶端跟第二個客戶端。第一個客戶端用上下左右來控制方向，並用enter鍵來加速、shift鍵減速，第二個客戶端用WASD來控制方向，並用G來加速、F減速。

1. Client與StartPanel

遊戲一開始執行時就顯示StartPanel，有兩個JTextField分別要輸入IP跟你的名字，有兩個按鈕connect跟start。要先連接上server才可以開始遊戲，如果還沒連接就按start，會有訊息顯示提醒你要先連接。如果你輸入錯誤的IP，也會有訊息告訴你IP錯了。

連接上以後會顯示connected，接著輸入你的名字按start。server會等兩個client都開始之後才會開始傳資料，蛇也會等兩個client都按開始後隔兩秒鐘才會開始動。

為了讓開始介面可以適應window大小的變化，我用GridBagLayout來排StartPanel的介面，就算把視窗放大到全螢幕也是可以用。

順道一提，還有一個類是JTextFieldLimit，他繼承PlainDocument，用來限制名字輸入的長度，為了讓名字在遊戲介面上可以晚整顯示。

Client裡主要的函式為connect()，用來創建客戶端的socket，若已經成功連接上server就移除connect按鈕的listener。server會等2個client都連接上了才會開始傳資料，在那之前client會一直讀，直到讀到為止。一開始會先讀一次server傳來的資料初始化client本地的資料(dynamicUIData)，Client接收server傳來的資料一次以後，PlayPanel才會按照server初始化的遊戲地圖建立。之後就開一個線程不斷讀取server傳來的資料並更新本地資料。

1. PlayPanel

PlayPanel本身繼承JPanel又實踐Runnable接口。這個線程是讓蛇動起來，蛇如果沒有吃到身體、撞牆或吃到別人才移動，移動的方式為去掉一個尾巴，用一個Point oldTail儲存，接著依照現在的移動方向增加一個頭。每移動一次結束就傳資料給server，讓server更新蛇的身體位置，接著判斷移動後蛇的狀態，並重新在panel上畫一次。

函式initialSnake()會重新初始化蛇的身體，間隔2秒後隨機從蛇洞出來。

函式judgeSnakeMoved()判斷每一次蛇移動後的狀態，如果蛇吃到食物，他的身體就會加長(多一個point oldTail)，並且分數加1；如果蛇撞到牆壁、吃到自己、吃到另一條蛇，分數就會減半，然後在生命不等於零的條件下呼叫initialSnake()重新初始化蛇的身體；如果蛇進到蛇洞，就會先儲存目前蛇身長度，建立一個Timer按照蛇移動的速度把身體一一移除，直到只剩一個頭，隔兩秒鐘後，創建另一個Timer按照蛇移動的速度把蛇的身體一一加回來，蛇會隨機從一個蛇洞出來，並隨機指定一個方向。

1. EndPanel

當任何一方生命等於零時就會切換到EndPanel，EndPanel告訴你輸贏並顯示雙方的分數。

**問題解決：**

在寫大作業時我遇到了很多問題，以下舉3個問題以及解決辦法：

1. reset ObjectOutputStream

原本server一直無法傳遞修改後的資料，在server端更新資料後傳到client還是原本沒更新的資料。這個問體我花了整整兩天的時間尋找答案，式了千百種方法還是無法解決。後來助教告訴我在穿資料之前可以先reset ObjectOutputStream，就可以成功傳遞資料了。網路上說如果沒有先reset，有可能會有剩餘的暫存，導致資料傳遞沒辦法實踐。

1. JPanel set focusable、request focus

在添加KeyAdapter到JPanel後必須將此panel set focusable(true)，在request focus，鍵盤事件才可以正常執行。

1. client用線程更新

原本client用Timer來更新，但發現不同步，會有延遲的現象。後來改用線程就解決這個問題了。