

專案題目：Fat Guy Match

小組名稱：right?

小組成員：103062121 劉亮廷 | 103062307 江柏翰 | 103062238 林子淵
103062104 邱思燈 | 103062315 黃孟霖

●背景與目的

當大家已經著迷不管是一直以來的 LOL 或者最近剛出的 Overwatch，我們卻已經在想應該是時候做出一款自己爽玩的連線對戰遊戲，而從一剛開始只想著要跟 LOL 一樣的遊戲進行方式，到後來幾經波折的演變到現在我們遊戲進行的方式，簡單、刺激、技術運氣兼需、而且耗時不多卻能令你連玩 10 幾場都很爽的遊戲，就這樣誕生了。

沒錯，我們不管甚麼 gamification，我們只要玩得爽的遊戲，我們想要的是享受跟三五好友一起連線在一個戰場上的感覺，藉由許多的隨機元素，讓你在玩的過程中總是會突然有著各種驚奇白癡的狀況，大家一起認真在技術操作上，即使目的很單一，但卻因為各種的不確定因素，讓趣味性提升到一個很高的檔次。對我們來說，做出一個能令大家都很想玩的遊戲，就是我們唯一的訴求了。

●設計概念

在一剛開始大家進入遊戲後，會隨機分配到地圖的各個位置，而地圖中也會隨機分布著塔，而塔是會在玩家接近時攻擊玩家角色的，而將塔摧毀也會的到掉落的靈魂，靈魂吃下去後會有隨機的效果，而效果中也有當場立即死亡這種負面的效果，增加了遊戲的惡搞感，而這也是一個遊戲中不可或缺的元素，一切都看玩家的造化啦。

再來是勝利條件，把噩夢男爵打死的人獲得勝利，而王一直都會在地圖中間的部分，但你並不是直接去攻擊就會對他造成傷害的。你必須去不斷收集靈魂，才有機會得到「倍增形影」的效果，而你變巨大後的攻擊才會對噩夢男爵造成傷害，不過你可能會覺得那不就先變大的人先贏嗎？那你就錯了，在一樣的地圖大小下，你躲避塔及其他人攻擊的難度就變得極高，因此你還需要小心不讓其他玩家發覺你已經巨大化了。

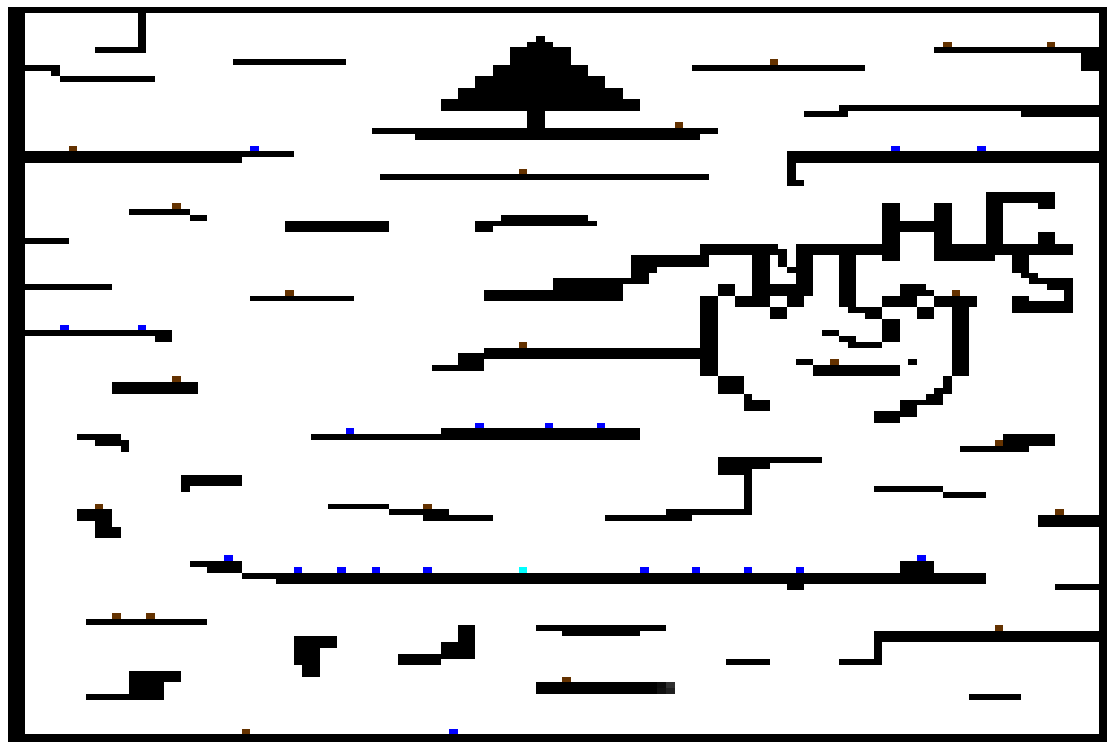
而最後我們需要一個讓玩家互相廝殺的動力，雖然單純廝殺已經很爽了，但我們設定了玩家死亡後也會留下靈魂，因此你也可以透過不斷獵殺他人而獲取靈魂，而從此中也演變出一個流派就是，守在噩夢男爵的附近專門狙殺巨大化並靠過來的玩家，並藉由得到他們死亡後的靈魂賭運氣看會不會巨大化並且能最接近的直接打王。

這遊戲在簡單的規則，簡單的技能設定下，還增添了一味就是隨機傳送門，走進門後按上即會被傳送到地圖的隨機位置。這增加了遊戲的刺激感，舉個例子你可能會在跟別人打一打的時候，突然有個人傳到你們中間然後被你們兩個的炮火一起打死。

●技術細節

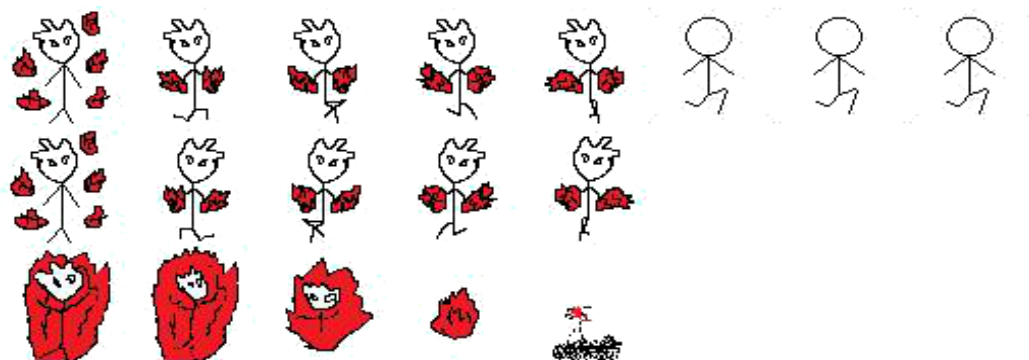
一、讀地圖辦法

因為我們不想把地圖寫死。首先我們畫一張圖片，並且讀每一格的 pixel，藉由顏色的區分，會把特定物件安插在特定顏色的位置上，像是藍色 pixel 代表塔的放置位置。當然其他使用者可以自己更改顏色的放置位置，來達成編輯地圖的效果。



二、角色動畫

我們把同一隻角色的動畫分鏡圖，放在同一張圖片，如下



把他放在(自己新增的 class) picture Sheet 裡，然後再藉由一個 picture class 來讀取，而讀取的方法以 pixel 為單位，用迴圈每次讀一固定單位的 pixel 已拿取角色動畫的一格分鏡，並將每次讀到的分鏡做為一個圖片存到陣列裡，而此陣列就是用來顯示角色動畫的圖片陣列。

三、遊戲藝術

在此遊戲中所運用到的所有不管是角色圖、技能圖、塔的圖，全部都是我們想好以後用小畫家手繪出來的，因此所有圖片皆為原創圖，尤其是角色分鏡圖，僅僅五個也需要仔細的慢慢分排動作，因此藝術方面也算是本遊戲的一個特點。

四、鎖定玩家視角

新增了一個 class 叫 camera 來實現這個效果，他會以玩家所在的地圖座標為中心點，向左減視窗寬度的一半、向上減視窗高度的一半。以此來實現角色在不斷移動的過程中，能始終保持在視窗的中間。

五、Server 和 Client 的架構

採用 model-view-controller pattern，把底層遊戲運行交給 server 來處理，並統一由 server broadcast 資訊給 client，client 負責回饋畫面給玩家以及讀入玩家操作給 server。Server 及 Client 都各有一份遊戲所有的物件，透過 socket 的通訊，使兩邊的所有 data 同步。

Server 擁有一個 threadPool 裡面裝有接收各個 Client 訊息的 thread，並且每個 thread 都有一個 id 來辨識是哪一個 client 傳來的，依此進行判斷要更新哪一個角色的資料。

六、角色資料計算(Server)

移動：

velX, velY 為角色 X 座標的速度, Y 座標的速度, 當為靜止時 velX = velY = 0，每次更新時，將角色 X 座標加上 velX, Y 座標加上 velY。

跳躍和掉落：

設有 jumping falling 兩個 boolean 值和 gravity, 若 jumping 為 true，則遞減 gravity 加到角色的 velY，呈現一個嚴格遞減變加速度的運動，一開始會跳很快，越往上面會漸漸停住，當 gravity 小於某個值時，代表已經跳到頂端了，此時將 jumping 設為 false，falling 設為 true。

當 falling 為 true，遞增 gravity 的值，但 velY 是減去 gravity 的值，呈現越掉越快的狀態。

碰撞偵測：

所有物體的 Super Class Entity, Tile, Skill 有 getBounds() 可以回傳他們物體所在地點相同大小的 Rectangle，利用 Rectangle.intersects() 來判斷兩個物體是否有碰撞。遊戲運行中，各個角色會不斷檢查自己和所有物體是否有碰撞。

如果撞到技能，查看技能的 ID 是誰的，-1(Tower), -2(Boss), 1~n(玩家)，並依據技能類型來扣血，最重要的是紀錄 lastAttackID，紀錄是誰最後攻擊這個角色，這樣在角色本身死亡後才能知道是由誰擊殺的，如此才能算分數，而技能物件本身會進行 die() 的呼叫，讓自己消失。

如果撞到牆壁，強制將 velX, velY 設為 0，讓角色靜止不動，除此之外，如果是角色頂部碰到則將 jumping 設為 false，falling 設為 true，如果底部碰到，則將 falling 設為 false，還有一個細節是，如果底部沒碰到牆壁就要將 falling 設為 true(不在平台上)。

如果撞到靈魂，隨機效果，有可能死掉, 加血, 獲得閃·飛雷神效果或是倍化形影效果，當靈魂數吃到 5 個以上時，就有機會能夠將 canAttackBoss 設成 true，但就不能再補血了(遊戲平衡機制，吃那麼多靈魂就是欠殺)。

技能發射：

其實並不是寫在角色的 Class 裡面，在 Server 中接收 Client 訊息時，若發現玩家發射技能，尋找技能中沒有被使用的，將它的 ID 設為此玩家 ID 並發射出去。

閃・飛雷神(移動速度加快且有殘影)：

另外一個 Class Trail，能夠顯示角色圖片，並依據時間漸漸變透明，當完全透明時，就會消滅。在角色當前位置不斷建立 Trail 物件(其實一開始就建好了只是拉來用)，如此就會產生殘影效果，並將角色的速度都提升 1.5 倍。

倍化形影(巨大化)：

單純的就是將角色寬高都設成兩倍，其他判定因為使用 Rectangle 判定，所以不會產生問題。

角色死亡：

進行死亡動畫，如果動畫結束，建立一個靈魂(拉來用)在角色死亡位置，重設角色所有狀態(如去除閃・飛雷神)並隨機重生在地圖上。

角色動畫補充：

設有 frame 整數，來選擇要顯示哪一張圖片，當移動時(move 為 true)，會變更 frame 的值，如此一來會顯示不同圖片。

七、技能更新(Server)

為了便利 Server 和 Client 之間的通訊，一開始會建立好一定數量的技能，並將它放在地圖外，要來用的時候將 used 設為 true，而 used 為 true 的技能才會被更新和顯示。更新很簡單，只是不斷的更新 X, Y 座標。比較特別的是追蹤彈，會擁有當前目標的資料，並不斷向目標前進。

八、Tower 及 Boss

Boss: 不斷的檢查所有 Character 是否在攻擊範圍內(使用 `Reactangel.intersections()`)，如果是的話就發射技能。也會檢查自己有沒有被角色技能打到，如果被打到了就會扣血，並記錄是誰最後打到他的，這樣在死亡後就能夠紀錄擊殺分數，並且 boss 有兩種攻擊技能，一種是快且比較不痛的，

另一種是慢但必死的，第二種施放前 boss 會有攻擊動畫(smile)，當他笑完就會發動攻擊，這部分就是另外設個變數 spell，當動畫結束後，spell 設為 1，否則都為 0，然後第二種技能在 spell=1 的時候才會施放即可。

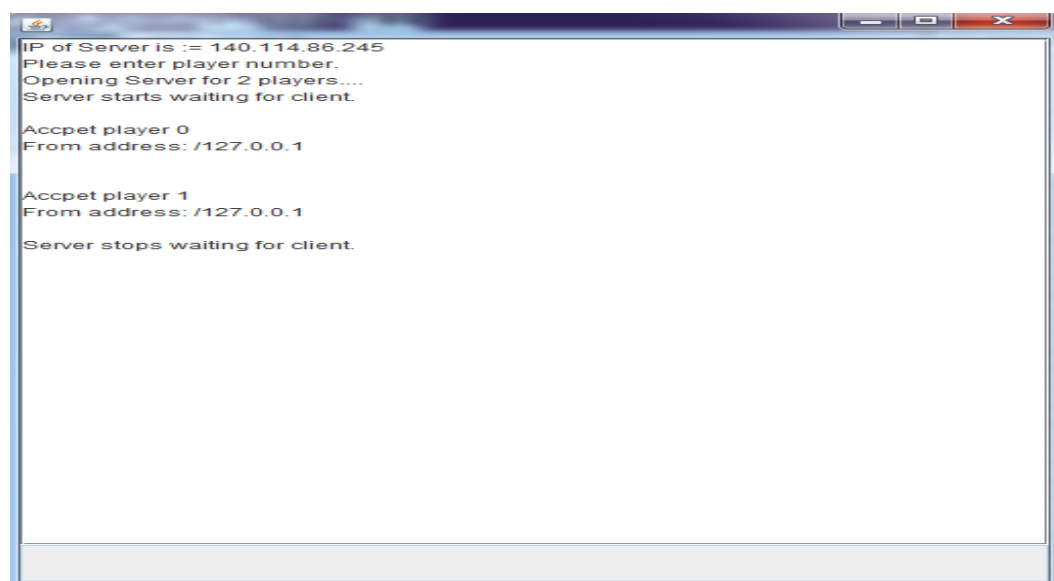
Tower:塔的部分一樣是用 Rectangel.intersections()偵測角色，在範圍就攻擊，不同的是其只有一種攻擊方式，並且掉落的物品每個角色都可以拿。

九、傳送門

傳送門會判斷角色是否有在門的範圍裡(150,150)，並且角色必須在門的範圍裡按下 W，然後就會被全地圖隨機傳送，門也跟塔一樣是隨機生成，不同的是因為門不會消失，所以當前這次的門是固定的，下次才會在隨機生成。

●實作成果

一、Server



Server 介面

Server 會顯示出自己的 IP 位址，並在輸入完玩家數量後，開始等待 Client 端的連結，而玩家連上了以後也會顯示玩家 IP，遊戲開始或結束也會顯示訊息。

二、Client

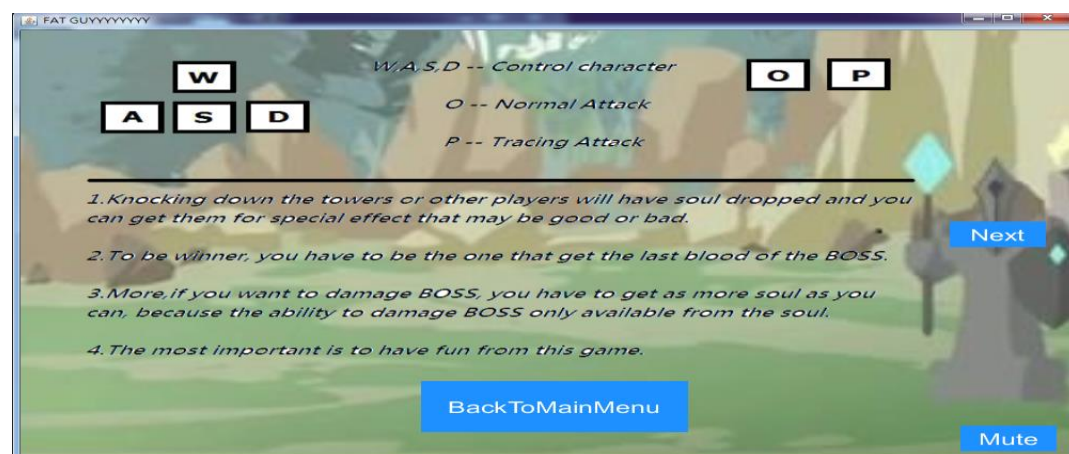
1. MainMenu



初始畫面

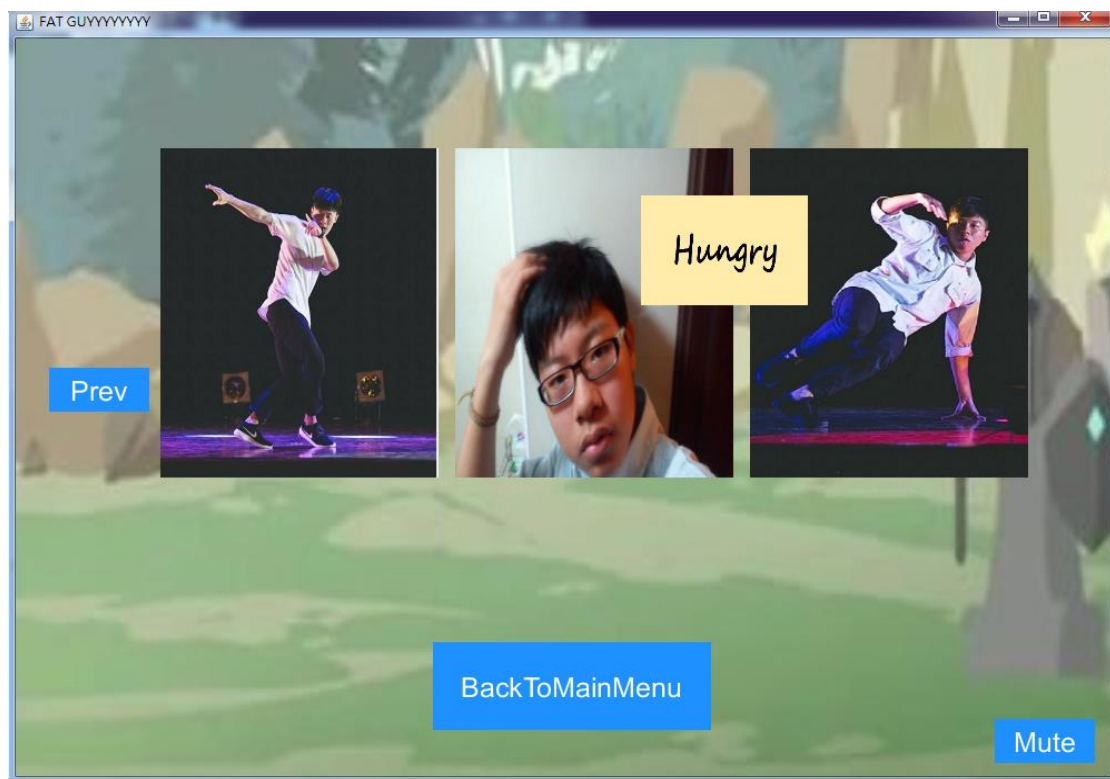
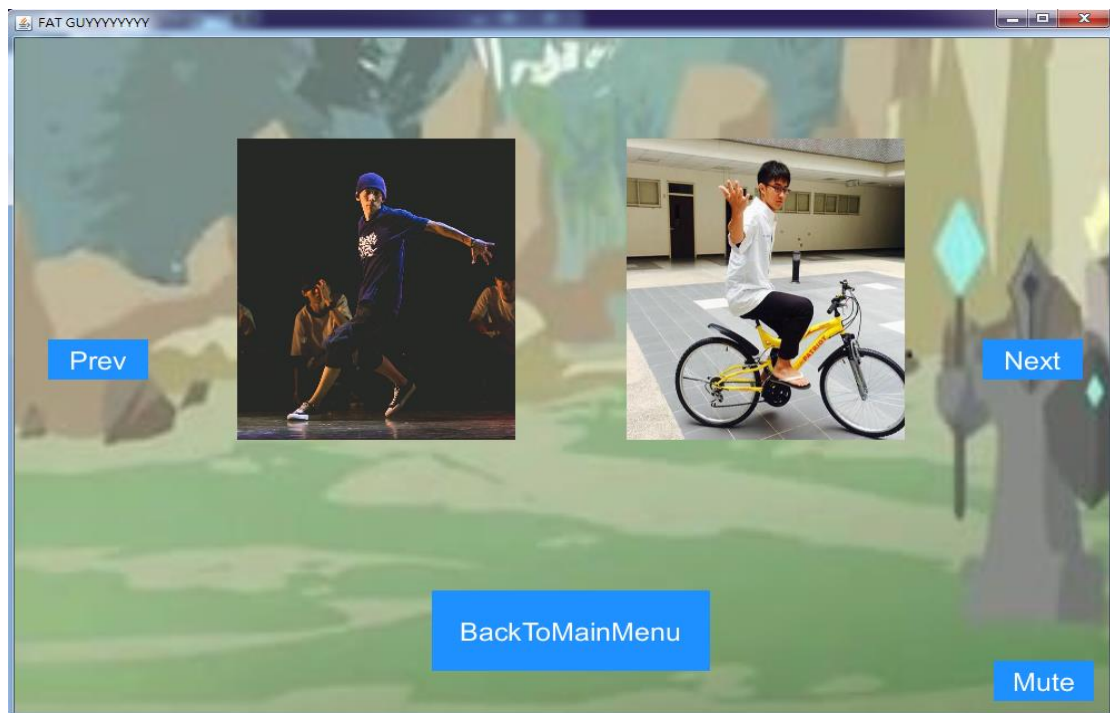
玩家須輸入 Server IP 後，方可連結至 Server；或者，按下 Information 按鈕後，即可詳閱遊戲說明；另外，遊戲有背景音樂，可透過 Mute 鈕選擇是否要靜音。

2. Information 畫面



遊戲規則

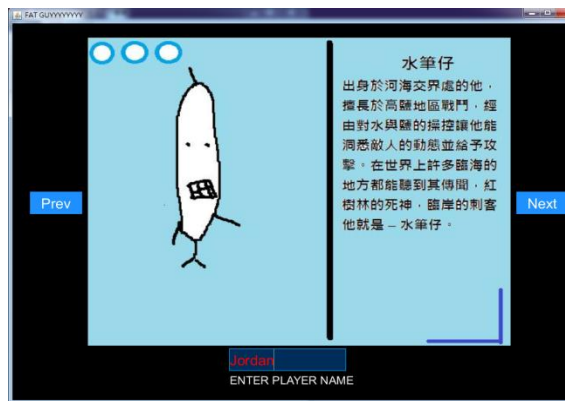
透過鍵盤 WASD 鍵可上下左右移動角色，且 OP 鍵可發射武器攻擊，其中，O 鍵為普通彈、P 鍵為追蹤彈。



作者介紹

將滑鼠移至圖片上時，會顯示關於作者的介紹。

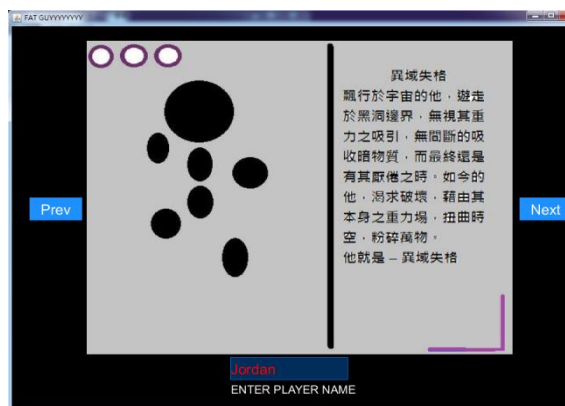
2. 選角畫面



角色 1：水筆仔



角色 2：紅蓮羅生



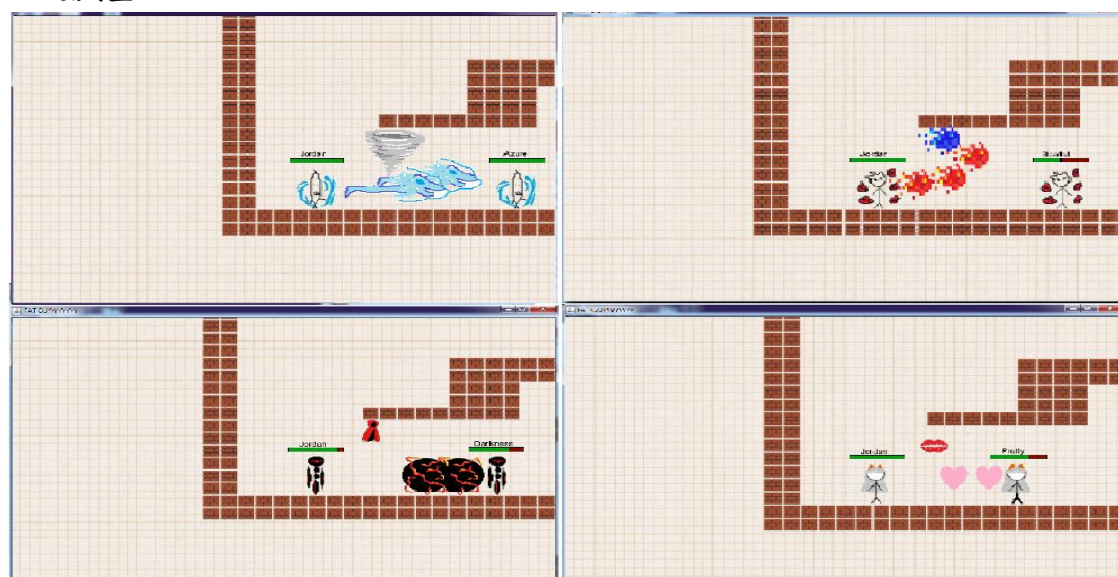
角色 3：異域失格



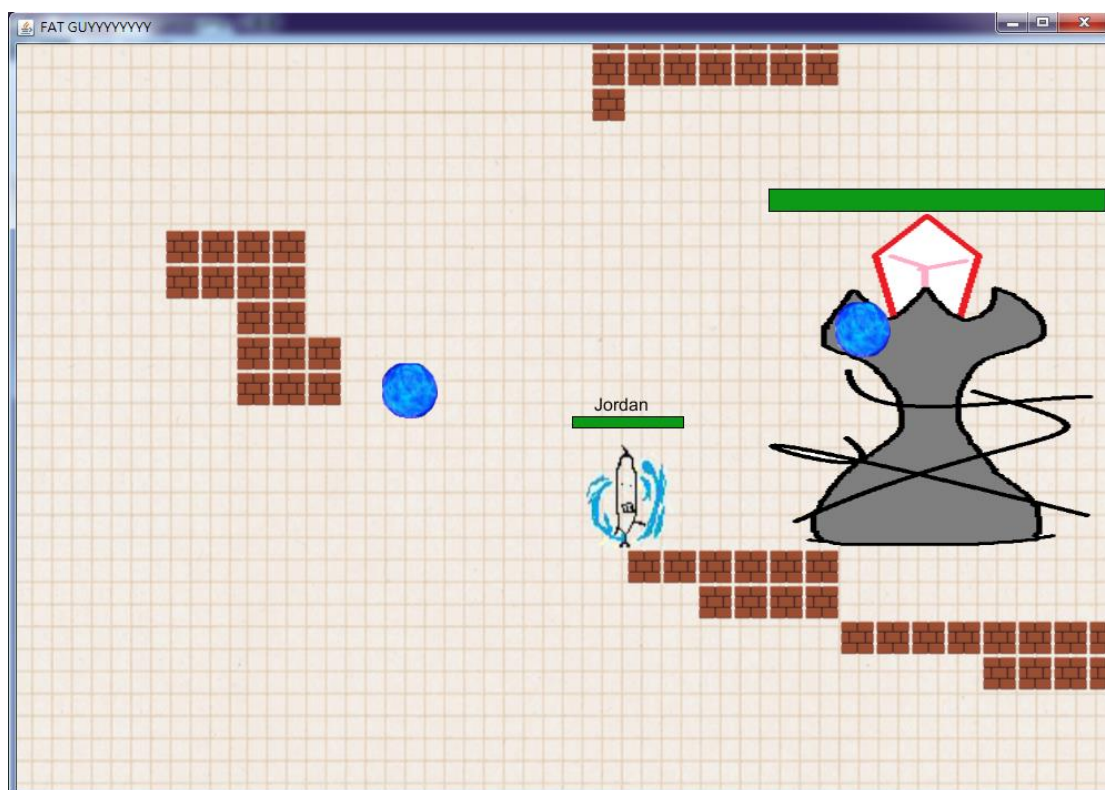
角色 4：銀夢颯想

玩家可在詳細閱讀完每隻角色的故事背景後，從四種各具不同特色的角色中選擇一隻進行遊戲體驗；並且，可在輸入一個角色 ID 後按下 Enter 進入遊戲。

3. 遊戲畫面

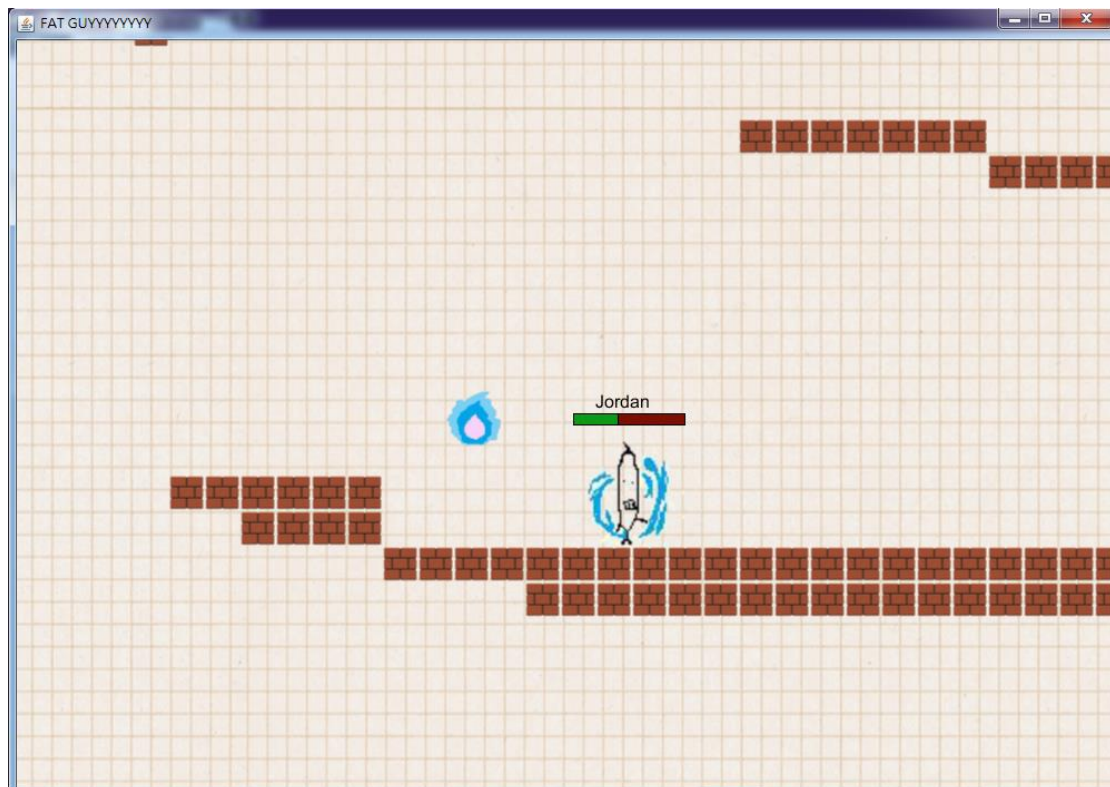


每隻角色在移動與死亡時都各會有由五張圖片組成的各具特色動畫。而在武器方面，每隻角色都有不同形態的兩種武器，包含普通彈和追蹤彈。普通彈會根據目前角色的面對方向直線前進，且速度較快；追蹤彈會在附近有其他角色時才能施放，且持續追著該角色直到一段時間後消失，而速度會因為距離該角色越遠而越快。



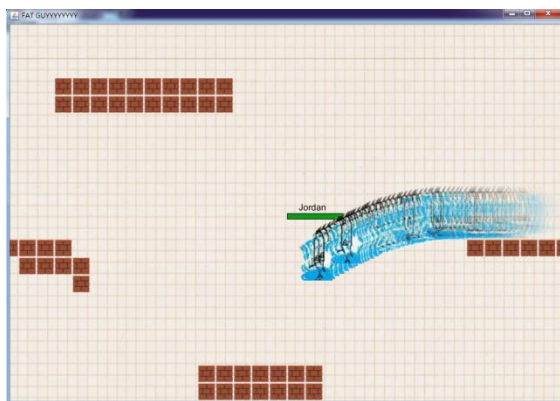
遊戲進行中 - 塔

進入遊戲後會在地圖上隨機固定位置中取一定數量生成，EX：從 25 個固定位置中隨機取 8 個生成塔；塔死亡後會一樣會在地圖上剩餘的固定位置中隨機取一個來重生，維持遊戲中的塔的總數保持一樣；而在遊戲進行中，塔會自動偵測其附近的角色並攻擊他。

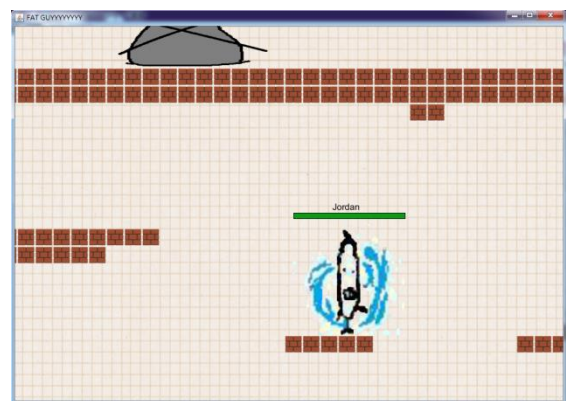


遊戲進行中 - 靈魂

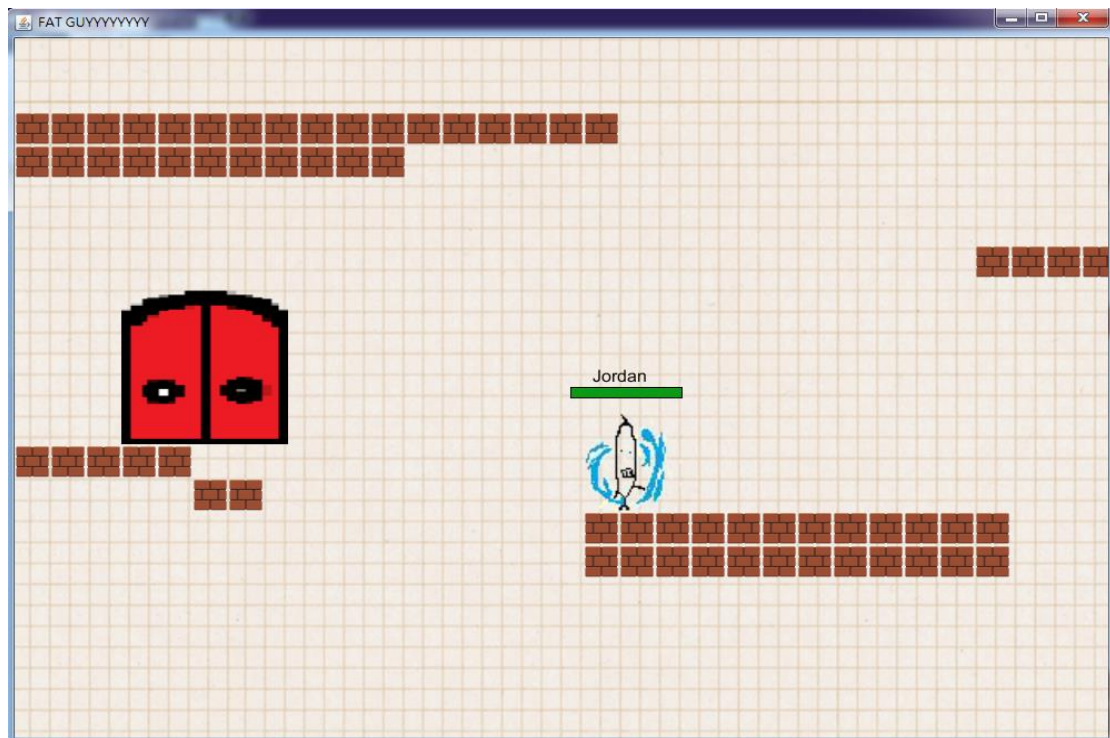
每當角色或塔死亡後，會掉落靈魂，當玩家吃掉靈魂後會獲得不同效果。效果共有四種，分別是閃·飛雷神(移動速度加快且有殘影)、虛無恐懼(立即死亡)、星體加護(回復 HP)、倍化形影(巨大化)。



閃·飛雷神

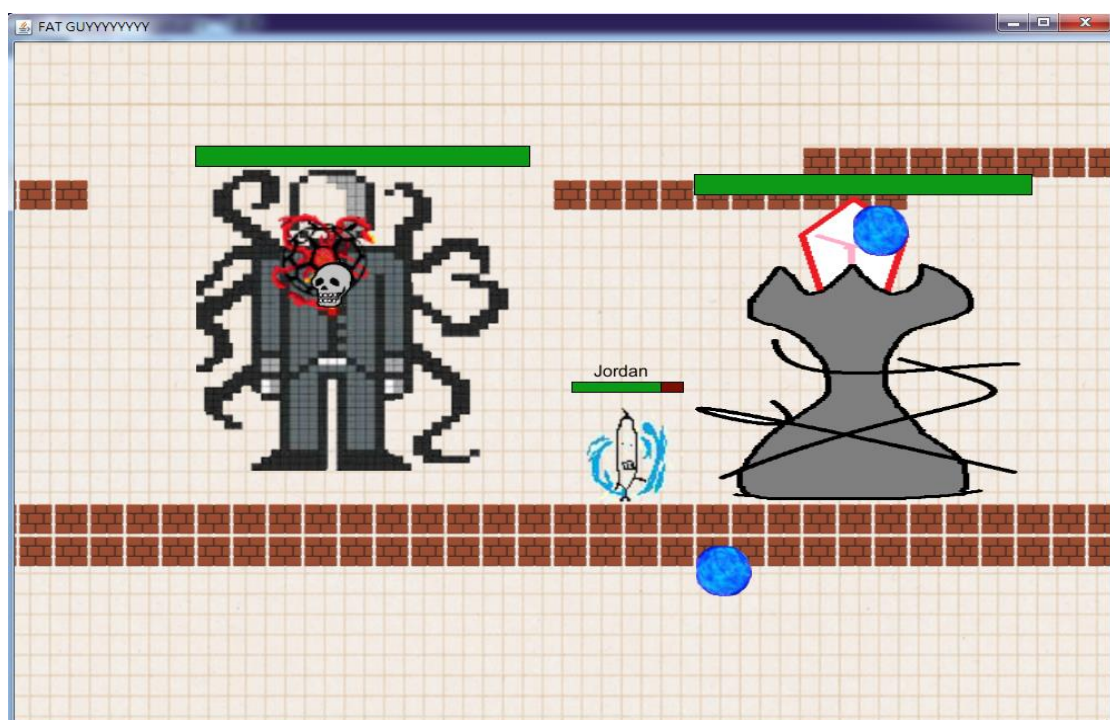


倍化形影



遊戲進行中 - 傳送門

進入遊戲後會與塔一樣，在地圖上隨機固定位置中取一定數量生成，玩家在門後按 W 鍵即可隨機傳送至地圖任一位置。



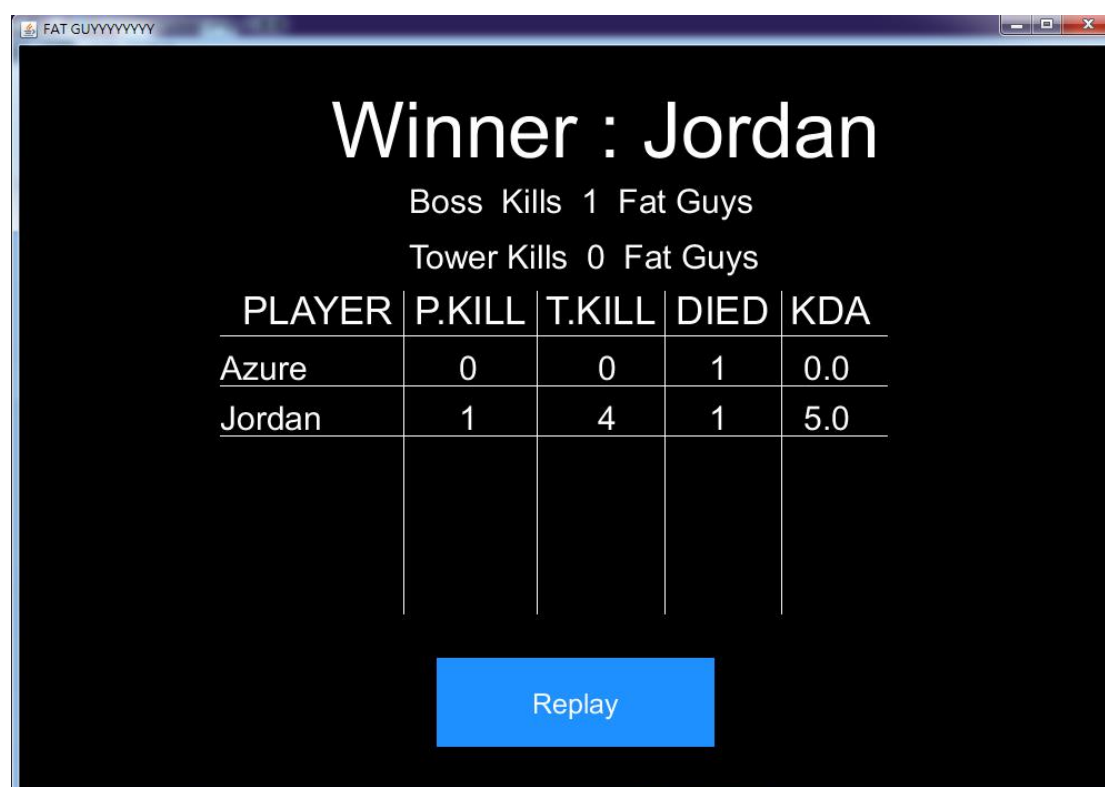
遊戲進行中 - 魔王噩夢男爵

魔王具有兩種攻擊方式，分別是魂奪駭骨與暗炎重力砲，前者為普通攻擊、後者為毀滅性攻擊，通常被打中一次就會死亡。

每個玩家在遊戲中都必須先收集超過五個靈魂才會開啟巨大化的功能，並且要在巨大化後對魔王的攻擊才會有效，否則永遠都會無效。

任一玩家只要擊殺了魔王後即為獲勝，並進入結算畫面。

4. 結算畫面



結算畫面

結算畫面包含整場遊戲的各種資訊，如獲勝者、魔王擊殺的人數、塔擊殺的人數以及每個玩家的擊殺人數、擊殺塔數、死亡數與 KDA 值；最後，按下 Replay 鈕即可回到 MainMenu，以重新開始遊戲。

●遇到的困難

1. GitHub 的 push & pull 的衝突

在開始做我們的專案之前，已經有先細想過我們的架構圖了；但儘管如此，在使用 SourceTree 來 pull 跟 push 時，我們仍是遇到很多預料之外的衝突，光是為了解決那些衝突讓我們花費了絕大部分的時間，投資報酬率極低。

最後，我們為了更節省時間，才決定將自己修改過的程式碼透過免空統一傳給組長來上傳至 GitHub，以提高效率。

2. Socket 連線造成 Thread 出錯

在接受 client 連線要求時，是採用 while(true)的迴圈，但如果用這個迴圈的話會造成 GUI 介面無法更新，因此後來使用 EDT 來解決。

●分工計畫

劉亮廷(20%)	Server 連線架構設計 遊戲元素設計、架構建置 內部測試 Proposal、Report 撰寫
江柏翰(20%)	遊戲元素設計、架構建置 美工設計 內部測試 Demo 上台報告 Proposal、Report 撰寫
林子淵(20%)	GUI 架構設計 遊戲元素設計、架構建置 美工設計 內部測試 Proposal、Report 撰寫
邱思燈(20%)	GUI 架構設計 遊戲元素設計、架構建置 美工設計 內部測試 Proposal、Report 撰寫
黃孟霖(20%)	GUI 架構設計 遊戲元素設計、架構建置 美工設計 內部測試 Proposal、Report 撰寫

●遊戲 Demo 影片連結

https://youtu.be/Mxowzbg3m_w

●GitHub 連結

https://github.com/tim70036/FatGuyMatch/tree/Right_2

最終完整檔案都放在 Right_2 的分支裡。

Branch: Right_2 ▾ New pull request

Create new file Upload files Find file Clone or download ▾

This branch is 114 commits ahead, 5 commits behind master. Pull request Compare

tim70036 Version 6 Latest commit ce3c441 6 days ago

settings	Init	25 days ago
bin	Version 6	6 days ago
lib	Init	25 days ago
src	Version 6	6 days ago
classpath	Eat 10 shifts to attack boss	10 days ago
gitignore	Add match png. Change fat to match	19 days ago
project	Init	25 days ago
87.png	No sleep , song La	19 days ago
Client.jar	Version 6	6 days ago
Server.jar	Version 6	6 days ago
author0.png	mainmenu ver4.0	14 days ago
author1.png	mainmenu ver4.0	14 days ago
author2.png	mainmenu ver4.0	14 days ago
author3.png	mainmenu ver4.0	14 days ago

●參考資料

<https://processing.org/reference/>

<https://forum.processing.org/one/topic/control-p5-changing-mouseover-color-font-size-font.html>

<http://stackoverflow.com/questions/23643600/java-online-game-should-the-game-happen-in-the-server-or-the-client>