

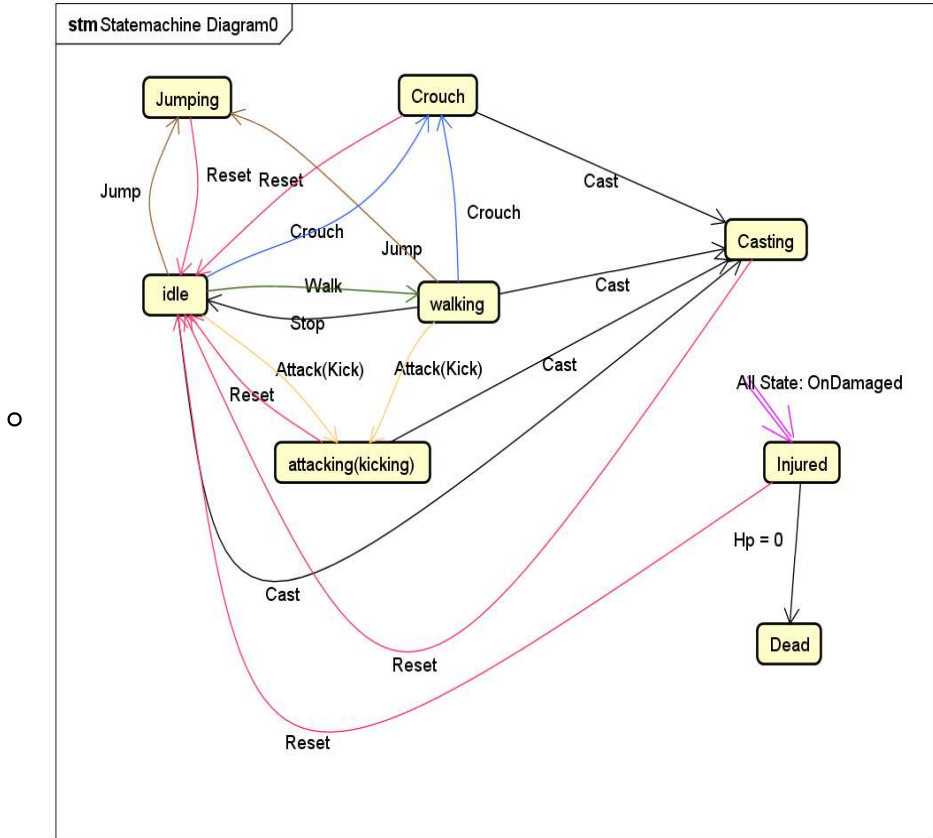
FOOP Final Project Report 蛋黃酥聯盟

組員

姓名	學號	參與事項	貢獻度
張翔文	B07902109	程式撰寫、動畫編排、UI撰寫	40%
李智源	B07902089	程式撰寫、UI撰寫	30%
周俊廷	B07902091	程式撰寫、圖像及音訊處理	20%
陳君翰	B07902059	UI撰寫	10%

Classes之間的關係

- 下列關係圖原圖都有放在Repo
- State Machine



The diagram illustrates the relationships between various classes in a game engine. The classes are organized into several groups:

- ImageRenderers:** KnightImageRenderer, SkillImageRenderer, LongSkillImageRenderer, FullScreenSkillImageRenderer, ImageRenderer (base), and ImageRenderer (note).
- Flying Objects:** IceWallFlying, LightningFlying, IceWall, Lightning, FireRing, and ... (note).
- Sequence:** Casting, Flying, and Triggered.
- Sprites:** Sprite, HealthPointSprite, MagicPointSprite, and UltimatePointSprite.
- FireBall:** FireBall (note).
- Character/Action:** knight, Gray, GrayAttacking, GrayKicking, Alita, Alita..., Characters..., States..., and Casting.

Relationships are shown as follows:

- Uses:** KnightImageRenderer uses KnightImageRenderer, SkillImageRenderer, LongSkillImageRenderer, FullScreenSkillImageRenderer, and Casting. SkillImageRenderer uses IceWallFlying, LightningFlying, and IceWall. LongSkillImageRenderer uses LightningFlying. FullScreenSkillImageRenderer uses ... (note). IceWall uses IceWallFlying. Lightning uses LightningFlying. FireRing uses LightningFlying. Casting uses IceWall, Lightning, FireRing, and FireBall. Casting also triggers Flying via the done() method.
- extends:** HealthPointSprite extends Sprite. MagicPointSprite extends HealthPointSprite. UltimatePointSprite extends MagicPointSprite. FireBall extends Sprite.
- Other Relationships:** knight uses Casting, Attacking, States..., Gray, Alita, and Characters... Gray uses GrayAttacking and GrayKicking. Alita uses Alita... Characters... uses ... (note).

- 優點
 - 新增角色、技能非常方便，沒有破壞助教原本寫的扣的OCP
 - 關係圖清楚，方便別人閱讀code
 - 圖片、音訊和Source Code分離，方便區隔占空間的資料夾，push上Github時使用LFS也較為方便
 - 使用CardLayout切換畫面(JPanel)，新增畫面或刪除畫面簡單
- 缺點
 - 由於使用Swing來設計UI，調整版面相比Web困難許多

- 只有使用awt, util, io等原生的package

遊戲畫面

- JFrame: GameView.java使用CardLayout

1. 主頁選單畫面(Intro.java): 玩家可按空白鍵開始遊戲或退出遊戲

- **START**: 按下空白鍵後開始遊戲且進入角色選擇畫面
- **QUIT** button: 退出遊戲並且關閉視窗

2. 角色選擇畫面(CharacterMenu.java): 兩位玩家可以各自選擇兩個角色

- 玩家可點擊附有角色icon的button來選擇角色
- **FIGHT** button: 當兩個玩家各自選擇兩個角色時, 此button會被enabled, 點擊之後可進入場景選擇畫面, 否則是disabled狀態

3. 場景選擇畫面(StageMenu.java): 玩家可選擇打鬥場景

- 玩家可點擊附有場景icon的button來選擇場景
- **START** button: 當玩家選擇場景之後可以點擊此button開始遊戲, 如果沒選擇場景, 點此button後會選擇default場景

4. 遊戲戰鬥畫面(Canvas.java): 玩家戰鬥時的畫面

- 此畫面會顯示兩個玩家的當前各自派出的角色並且戰鬥
- **Box** button: 玩家可透過此按鈕來決定是否開啟戰鬥時的Bounding Box(偵測角色邊框以及技能邊框)
- 每個人角色會有三個Bar分別為:
 1. Health Point Bar: 顯示該角色當前的Health Point
 - 綠色範圍: 殘餘Health Point
 - 紅色範圍: Max health Point - 殘餘Health Point
 2. Magic Point Bar: 顯示該角色當前的Magic Point
 - 深藍色範圍: 殘餘Magic Point
 - 灰色範圍: Max Magic Point - 殘餘Magic Point
 3. Ultimate Point Bar: 顯示該角色當前的Ultimate Point
 - 淺藍色範圍: 殘餘Ultimate Point
 - 灰色範圍: Max Ultimate Point - 殘餘Ultimate Point

5. 遊戲結束畫面: 遊戲結束後會彈到此畫面詢問玩家是否繼續遊玩

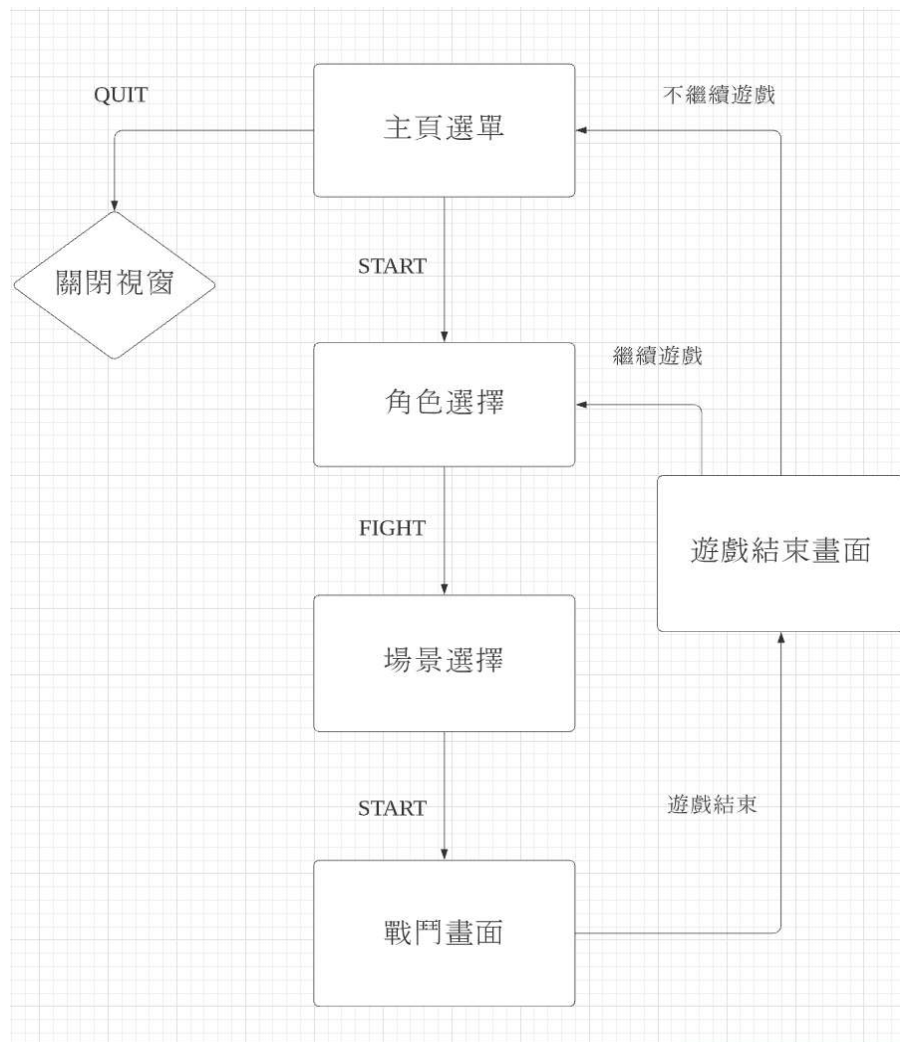
- **Yes** button: 跳到角色選擇畫面

- **No** button: 跳到主頁選單畫面

遊戲流程

1. 在主頁選單畫面選擇開始遊戲然後進入角色選擇畫面或者退出遊戲並且關閉視窗
2. 在角色選擇畫面兩個玩家可用滑鼠點擊角色的icon來選取角色，當兩位玩家都各自選擇了兩個角色便可進入場景選擇畫面
3. 在場景選擇畫面玩家可選擇任一場景進行戰鬥，選擇之後可進入遊戲戰鬥畫面
4. 在遊戲戰鬥畫面中，玩家可根據不同的角色使出不同技能並且對敵方照成傷害，直到某方的兩個角色都死亡遊戲便結束，進入遊戲結束畫面
5. 在遊戲結束畫面，玩家可選擇是否繼續遊玩，是的話進行角色選擇畫面，否則就進入主頁選單畫面

遊戲畫面流程圖



遊戲規則

1. 這遊戲**必須**有兩個玩家才可進行
2. 每個玩家必須各自選擇兩個角色，否則**FIGHT** button會是**disabled**的狀態
3. 場景選擇頁面如果玩家沒選擇場景，會使用**預設場景**
4. 一個玩家每次只能派出一個**角色**進行戰鬥
5. 當Health Point ≤ 0 時，該角色判定為死亡，玩家會自動派出另外一個還未死亡的角色，如果兩個角色都死亡則遊戲結束，另一個玩家勝利
6. 遊戲每秒會使當前戰鬥中角色的Magic Point自動增加5
7. 每當角色使用punch/kick攻擊敵對並且成功對敵對造成傷害，該角色的Ultimate Point 會增加50
8. 當角色當前的Magic Point \geq first skill 的required Magic Point，才可以使用first skill
9. 當角色當前的Magic Point \geq second skill 的required Magic Point，才可以使用second skill
10. 當角色當前的Magic Point \geq ultimate skill 的required Magic Point且Ultimate Point \geq Max Ultimate Point，才可以使用 ultimate skill
11. 當玩家只剩下一個角色，另一個角色死亡時，玩家無法使用change character這個功能

Character的attribute

1. 每個角色出拳和踢腳的Damage Box都不相同，各有特色
2. First skill: 每個character有專屬的first skill,每個first skill required的Magic Point都不同，對敵人造成的傷害也不同
3. Second skill: 每個character有專屬的second skill,每個second skill required的Magic Point都不同，對敵人造成的傷害也不同
4. Ultimate skill: 每個character有專屬的ultimate skill, 每個ultimate skill required的Magic Point都不同，對敵人造成的傷害也不同, 但都需要該角色Ultimate Point達到最大值才可以使用
5. Health Point: 預設值為500，最大值為500
6. Magic Point: 預設值為200，最大值為200
7. Ultimate Point: 預設值為0, 最大值為200

Character

- Emily



- 特性: 穿著治癒術士的制服降低敵人的警戒，再給對方突如其來的一擊！
- 技能: LightningBolt(100mp,50damage), Lightning(150mp,100damage), **IceWall**(ultimate skill,200damage)
- 故事: 小時候夢想成為受人歡迎的治癒術士，平常最大的愛好就是收集治癒術士的制服，沒想到治癒術士公會不接受男性加入，於是從此踏上了反抗世界的征途

- Gray



- 特性: 蛤？Nazi算哪根蔥？我才是真的火龍王！
- 技能: FireBall(125mp,100damage), Fire(150mp,150damage), **FireRing**(ultimate skill,200damage)
- 故事: 曾經是納茲最忠實的粉絲，一心只鑽研究極的火焰魔法，在一場比賽中發現Nazi使用紫電後大受打擊，誓言要打到Nazi這個偽君子

- Alita



- 特性: 機械族的少女，能夠高速且悄無聲息的移動，可靠又可怖的殺手
- 技能: LightningBolt(50mp 50damage), FireBall(100mp 100damage), **AtomicBomb(LightningBolt + FireBall)**(ultimate skill,150damage)
- 故事: 甦醒於失落國度的古遺跡，除了知道自己的名字之外已喪失所有記憶，僅有額頭上的寶珠作為線索，發誓要找到王國隕落的真相

- Nazi



- 特性: 如果有一顆火球解決不了的事情, 那...可以考慮紫電 (?)
- 技能: LightningBolt(50mp,50damage), FireBall(100mp,100damage), **SuperFireBall**(ultimate skill,100 damage)
- 故事: 原本是最強的烈焰魔導士, 但在某次眾神的聚宴中惹惱了天空之神因而受到懲罰, 失去大部分操控火焰的能力, 但也意外的獲得紫電的技能?

Skills

- 圖片以Casting->Flying->Triggered的順序來呈現
- FireBall



- Fire(無trigger動畫)



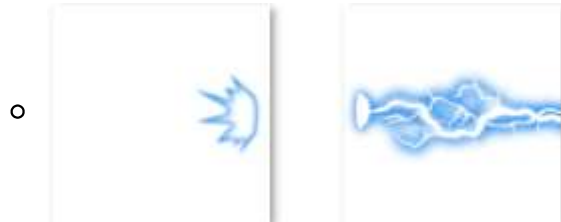
- FireRing



- IceWall(實際遊玩時冰柱會變成5個連在一起)



- Lightning(無trigger動畫)



- LightningBolt



Key-binding

Player 1:

- Control
 - **W, A, S, D**: jump, left, crouch, right
 - **F**: punch, **G**: kick, **E**: change character
- Combo:
 - **S+S+F**: first skill of character
 - **S+S+G**: second skill of character
 - **S+A/D+F+G**: ultimate skill of character

Player 2:

- Control
 - **ArrowKey_UP, LEFT, DOWN, RIGHT**: jump, left, crouch, right
 - **K**: punch, **L**: kick, **P**: change character
- Combo:
 - **DOWN+DOWN+K**: first skill of character
 - **DOWN+DOWN+L**: second skill of character
 - **DOWN+LEFT/RIGHT+K+L**: ultimate skill of character

組員心得

張翔文

這次做java期末project學到很多有用的東西，首先看懂助教的範例程式了解整個flow，知道簡單的一個sprite可以做那麼多事，了解oop的一個優點在於可重複性，之後使用OCP新增技能、角色時就發現助教的code真的寫得太OoP了，減少非常多重複的code。

Frame跟圖片動畫化也很有趣，如何調整WaitPerFrame和圖片間的關係，進而讓角色動的順，或是放上酷炫的技能動畫(一樣是用圖片做)都非常的有趣。

GameView和GameLoop分Thread跑也是非常有趣的作法，上網以一查才發現似乎這就是正常的方式，深入了解後發現SP上課學到的Thread終於應用到了好爽。

遊戲做好後，編寫介面時學習Swing也覺得很有趣，因為以前很常碰web的前端，碰到以前沒碰過的swing竟發現他們有一定程度的類似，但swing完全是縮減版，code容易寫得很髒，且設定也很少，但畢竟是不同年代的語言。

這次final project是目前我寫過最有趣的project，以前對java寫app和web programming之間的不同沒有概念，做了許多web project都不了解java在幹嘛，這次做了一份和web完全不一樣的app才了解兩者的差別，用java寫好像比較有趣，但web比較方便快捷，故以後有時間會嘗試多碰java增加這次project內容。

李智源

這次的final project讓我了解到OCP的重要性，如果不是OCP我們也沒辦法以助教的原始碼當基礎進而拓展成我們的遊戲。之前覺得寫noodle code沒什麼，只要弄得出來就好，但是由於這次是團隊合作，所以OCP的精神讓我們可以更好的互相配合，也讓自己學習如何寫出具有高度OCP的code。有次的印象特別深刻就是實作玩家MP bar的時候，寫了某個class不符合OCP，結果我連續改了好幾個被影響到的class的code，後來發覺直接用extend某個class來處理就可以輕而易舉解決那個問題了。這次也要謝謝隊員翔文，每次有問題他都會很積極地回答並且幫我解惑，讓我在實作上更有效率以及進步了許多。除此之外，這次由於人數較多，所以github的版本控制對我們的開發來說相當重要，也大幅提高了我們的效率，尤其是對我這種每次不小心把code寫得很髒的，不經過嚴格審核就merge真的會很慘。

周俊廷

這一次的期末Project我們小組是以助教所提供的範本中提取出想做類似於街霸的雙人格鬥遊戲。我所負責的是一開始對於人物、技能的圖片、音效搜尋及對應的前置處理，實際應用技能的程式的初步撰寫，後期則是由翔文進行整合以及對整體遊戲進行封裝方面，Makefile的事宜。在這一次Project中，我發現想要製作一款遊戲，並將遊戲所需的各種不同功能彼此結合、應用，需要將每一步都細細推敲。在這方面，在最一開始以助教的範例程式開始進行Project時，便對助教撰的寫各個部件之間的溝通與相互間的關係搞暈了方向，讓我明白為何助教會要求我們附上各個java模組間的關係圖。幸好最終有翔文居中統籌，將我撰寫零散的人物技能的程式整合到整體遊戲裡面。期間我

也協助了一些關於GUI的背景處理，在這方面我學到了關於java中對canvas, swing的實際應用，了解各個界面是怎麼一步一步堆疊所需的功能，最終形成我們所想要的界面。通過使用JPanel, JFrame之類的套件，對於構建java圖形界面有了基礎的了解。最後遊戲整體部分我們曾想要將其弄成.jar檔但最有考慮到使用直接將遊戲跑起來更省事，因此最後沒將這部分實做。總的來說，在這一次的Project中我更多的是在各個小部分上提供一些幫助，整體還是要歸功於翔文居中將我零散的程式整合到遊戲裡，更是因為不怎麼熟悉github而時常詢問一些基礎的github的操作，還好智源在這方面有經驗能提供幫助。

陳君翰

在這次的project中，我原先是負責GUI的部分，雖然對於awt, swing都已經有一定的認識，也閱讀了許多資料，在單純撰寫頁面上都沒有太大的問題，然而，對於助教所提供的code的運作方式在理解上有所困難，因此遲遲未能做出進展，後來經過翔文的幫助和解釋才將死結打開，而在GUI中許多的細解與實作最後都是由翔文修正並完成，非常感謝他的幫助。對於這次project而言，我認為我在理解、活用既有程式碼的能力上有所缺失，尤其都已經是提供OCP的環境中卻仍無法好好做事，我認為我還有很大的改進空間，也同時非常感謝其他組員的協助，這次的project主要都是依靠他們才能完成，未來將會持續學習OCP, 程式改寫活用, git版本控制等知識，以成為有用的組員！

詳細資料

- Github連結:

<https://github.com/Fundamental-OOP/final-project-b07902091> (<https://github.com/Fundamental-OOP/final-project-b07902091>).

- Music Reference

Cherry Metal by Arthur Vyncke | <https://soundcloud.com/arthurvost>
(<https://soundcloud.com/arthurvost>).

Creative Commons Attribution-ShareAlike 3.0 Unported
https://creativecommons.org/licenses/by-sa/3.0/deed.en_US
(https://creativecommons.org/licenses/by-sa/3.0/deed.en_US).

Music promoted by <https://www.chosic.com/> (<https://www.chosic.com/>).