

OOP 期末專題報告

題目：拳皇再世

百川 0819814-潘丞詡(50%) 百川 0819823-陳子祈(50%)


(請填上相關資料並在名字後面填上分工比例)

A. 題目描述

(請說明專題內容並以條列方式說明完成的功能)

專題內容:重現拳皇遊戲中，雙方對打，增加自己藍色魔力以及造成對方扣血。

完成的功能:

- 1.遊戲一開始會是初始畫面，玩家隨意用滑鼠點擊畫面中任何一個地方即可開始遊戲(有一串 " Tap To Continue " 在螢幕下方閃動，呈現真實遊戲常有的畫面)，進入遊戲後會有背景音樂。
- 2.左玩家運用 WSAD 控制上(跳)下(防禦)左右移動，右玩家運用  控制上(跳)下(防禦)左右移動。
- 3.左玩家按 J 近攻、K 遠攻、U 必殺技；右玩家按數字鍵 1 近攻、2 遠攻、3 必殺技
- 4.遊戲中兩個角色的 "所有動作" 都有動畫以及音效，不管是向左走向右走、跳躍、近(遠)距離攻擊以及必殺技。所有畫面不管攻擊還是走路等等，都是用 Sketch Book 剪輯的，因此，動畫多元，完成視覺觀感極度饗宴。
- 5.遊戲中，只要成功攻擊對方，不管是何種攻擊方式，該被攻擊者會自動扣血，血條會減少，而攻擊方會獲得 BUFF 能量條增加，除非對方成功於被攻擊時按下了防禦功能，才能抵擋該攻擊與扣血。
- 5.一旦 BUFF 能量條滿了，就可以觸發必殺技的功能。
- 6.執行必殺技功能時，會出現必殺技前置畫面，加上音效。前置畫面大概 1 秒，1 秒後會回到遊戲畫面，執行必殺技一連串動畫。
- 7.如果某方血條歸零，則該玩家失敗，另一玩家獲勝。

B. 程式架構

(請以 Block Diagram 方式畫出 function 之間的關係和執行流程，另外以條列或表格方式說明 function 的功能)

由於我們的程式的架構主要由判斷式組成，為了方便前後文的判斷，我們的判斷式定義為 global variable，並且寫在 main 裡面，只有寫 funtion 在上下左右移動及影像、聲音產生功能我們 main 裡面判斷式的流程如下:

1.進入遊戲開始畫面

(a)遊戲開始畫面

```
2666 if (menu_initial)
```

```
{
```

```
    if (clickmenu...)
```

```

    {
        menu_initial = false;
        menu_finish = true;
    }
}

```

(b)檢查滑鼠狀態

```
1531 mouseHandleEvent(&e, &mouseState, &mouseX, &mouseY);
```

(c)如果有按任何鍵就進入遊戲

```
2644 case IN_LB_PR: // Inside, Left Button, Press
```

```
    clickmenu = true;
```

2.進入遊戲後，一開始先檢查鍵盤狀態，如果有按按鍵就會執行對應的動作
按下鍵盤會做三件事情

I. 判斷此動作有沒有正在執行動畫，沒有就可以進行此動作

II. 此動作的音效開啟

III. 此動作的 timer 啟動

以近攻為例:

```
1599 if (!f1front_hit_operating)
```

```

{
    f1front_punch_hit = true;
    f1lasttime = f1curtime;
}

```

除此之外，個別還會分別做以下事情:

(a)跳躍

i. y 方向取速度

```
1563 f1VeY = VELOCITY;
```

(b)走路

i. 紀錄有沒有一直接著左右鍵，供以後有無持續走路用

```
1574 f1pressed_walking = true;
```

ii. 左鍵 true，右鍵 false，反之亦然

```
1576 f1standing_left = true;
```

```
1577 f1standing_right = false;
```

(c)近攻

i. 紀錄目前集更新上次攻擊方法，供以後判斷破防效果用

```
1595 f1last_attack = f1this_attack;
```

```
1596 f1this_attack = 'j';
```

(d)遠攻

同近攻

```
1608 f1last_attack = f1this_attack;
```

```
1609 f1this_attack = 'u';
```

(e)大招

i. 將上次攻擊方法及這次攻擊方法歸'0'

```
1624 f1last_attack = f1this_attack = '0';
```

(f)防禦

放開鍵盤

I. 停止音效

II. 效果消失

(a)走路

(b)防禦

除了防禦動作消失效果也消失

```
1748 f1defend_effective = false;
```

```
1749 f1lasttime = f1curtime;
```

3.設定 timer

```
1796 f1curtime = SDL_GetTicks();
```

```
1797 f1timediv = (f1curtime - f1lasttime) % f1period;
```

4.生命值為 0 則結束遊戲、播放死亡音效(結束畫面)，非 0 則繼續遊戲

```
1823 if (f1HPIndex == F1HPFRAME - 1)
```

```
{
```

```
    Mix_PlayChannel(1, f1deadeffect, 0);
```

```
    Mix_PlayChannel(1, f2defeateffect, 0);
```

```
}
```

5.藍 buff 滿則大招可以啟動

```
1829 else if (f2HPIndex == F2HPFRAME - 1)
```

```
{
```

```
    Mix_PlayChannel(1, f2deadeffect, 0);
```

```
    Mix_PlayChannel(1, f1defeateffect, 0)
```

```
}
```

6.檢查攻擊手段有無改變

```
1860 if (f1last_attack != f1this_attack)
```

```
{
```

```
    f2being_hit_different_way = true;
```

```
    f1last_attack = f1this_attack;
```

```
}
```

7.檢查要不要被攻擊

```
1880 if ((f1front_punch_hit) && (abs(f1PosX - f2PosX + 250) < HITDISTANCE) && (abs(f1PosY - f2PosY + 250) < HITDISTANCE))
```

```
    f2being_hit = true;
```

8. 判斷 defend 有沒有效

```
1889 if (f1defend_)
```

```
{
```

```
    if (f1being_hit_different_way)
```

```
        f1defend_effective = false;
```

```
    else
```

```
        f1defend_effective = true;
```

```
}
```

9. 判斷近攻武器會不會造成傷害

```
1920 if (!f2defend_effective && f1front_punch_hit && ((abs(f1PosX - f2PosX) < HITDISTANCE) &&
(abs(f1PosY - f2PosY) < HITDISTANCE))))
    {...}
```

10. 判斷遠攻武器會不會造成傷害

```
if (!f2defend_effective && f1far_punch_hit_weapon_effective && ((abs(f1farpunchhitweaponPosX -
f2PosX) < HITDISTANCE) && (abs(f1farpunchhitweaponPosY - f2PosY) < HITDISTANCE))))
    {...}
```

11. 遠攻武器要不要跑

```
if (f1far_punch_hit_weapon)
    {...}
```

12. 進攻、遠攻、走路、防禦等動畫

都會

I. 看左右邊

II. 檢查 timediv 有沒有超過 period，有就跳出來

```
if ((f2timediv >= f2period - 50) && (!f2pressed_walking))
    {...}
```

13. 大招

(a) 前置動畫

```
if (f1fucking_kill_on && f1fucking_kill_triller)
    {...}
```

(b) 人物動作

```
if (f1fucking_kill_ex)
    {...}
```

(c) 武器施放

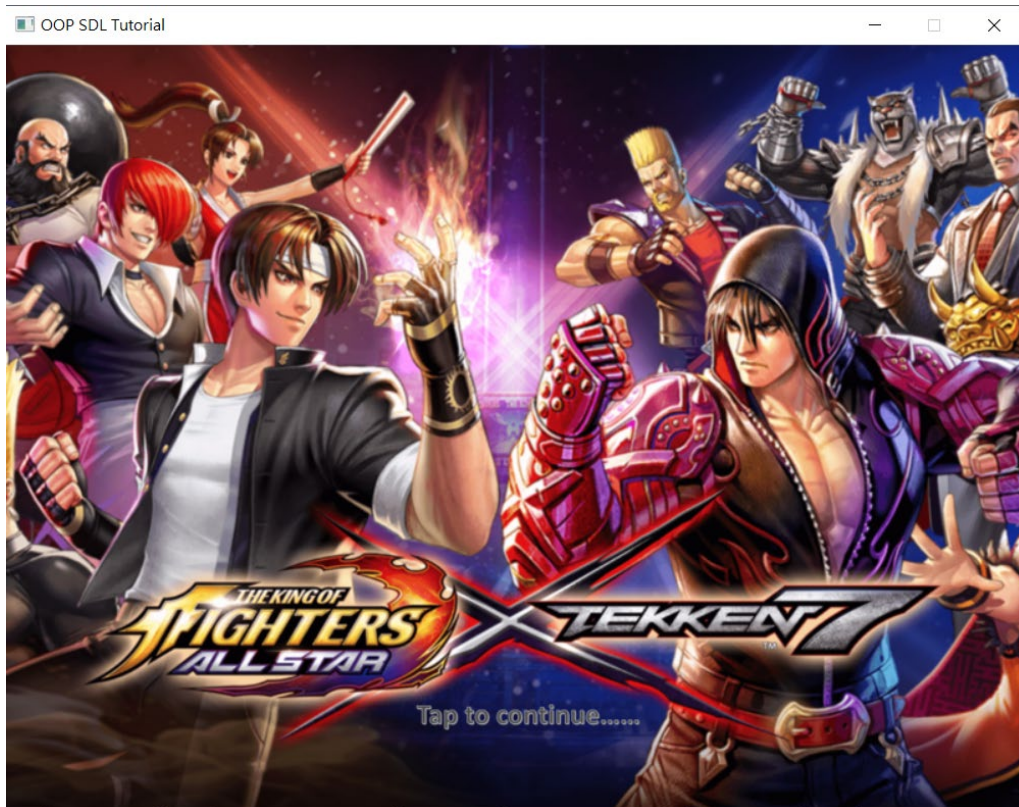
i. 施放位置 = 對手位置

```
2519 if (f1fucking_kill_weapon)
{
    f1fuckingkillweaponPosX = f2PosX - 105;
    f1fuckingkillweaponPosY = f2PosY - 30;
    if (f1fuckingkillweaponspeedtimediv >= f1fuckingkillweaponspeedperiod - 50)
    {
        f1fucking_kill_weapon = false;
        fucking_operating = false;
        f2HPIndex += 2;
        f1BUFFIndex = 0;
    }
    if (f1fuckingkillweaponIndex == 1)
        f1fuckingkillweaponPosY -= 100;
    f1fuckingkillweaponIndex = f1fuckingkillweaponspeedtimediv *
    F1FUCKINGKILLWEAPONFRAME / f1fuckingkillweaponspeedperiod;
    imgRender(renderer, f1fuckingkillweapon[f1fuckingkillweaponIndex],
```

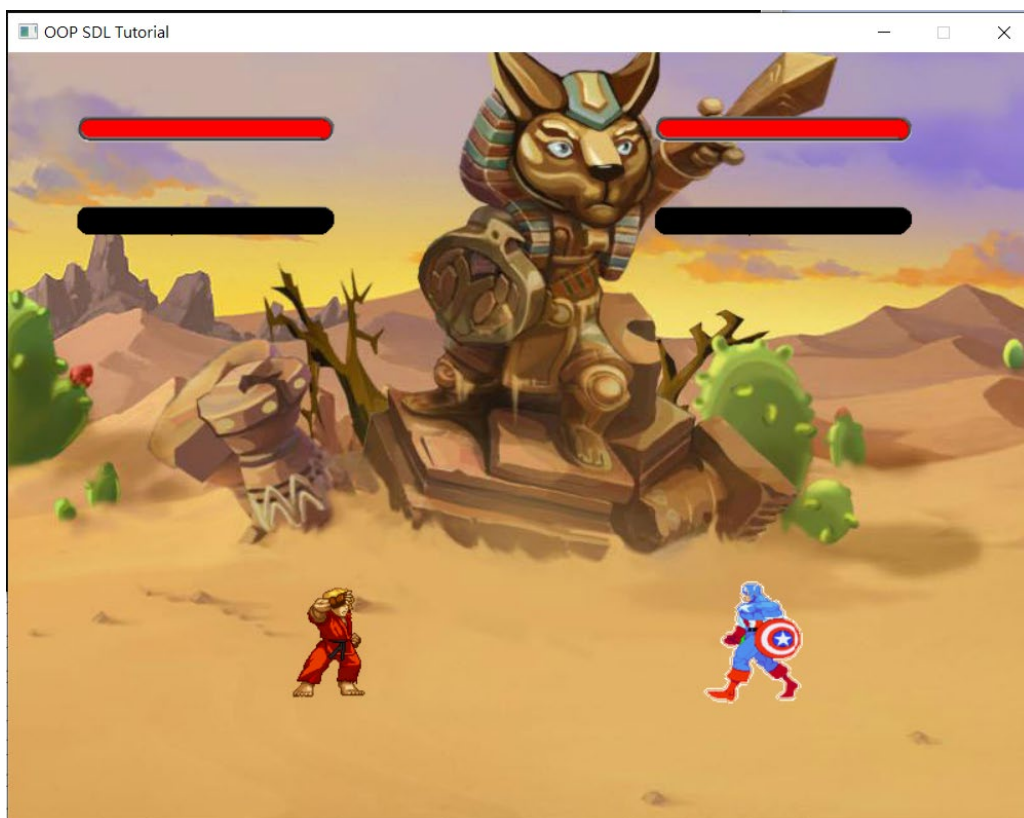
```
f1fuckingkillweaponPosX, f1fuckingkillweaponPosY);
```

C. 執行結果

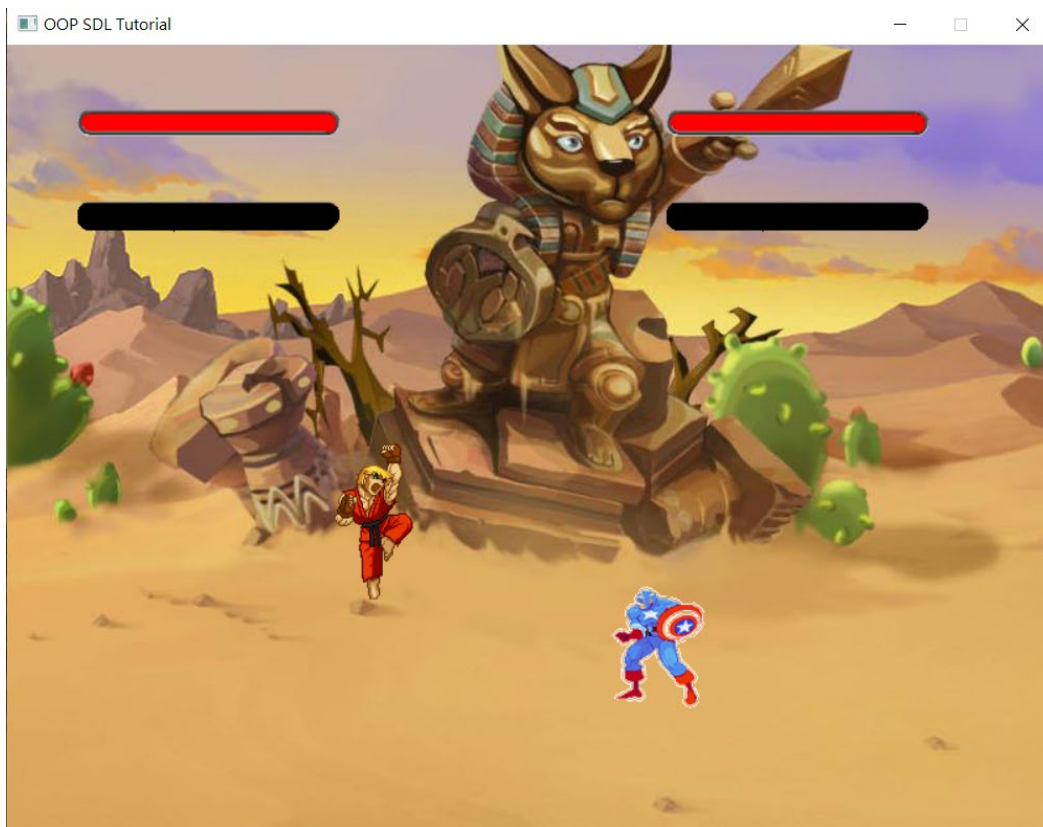
(請以截圖方式說明程式執行的流程與結果)



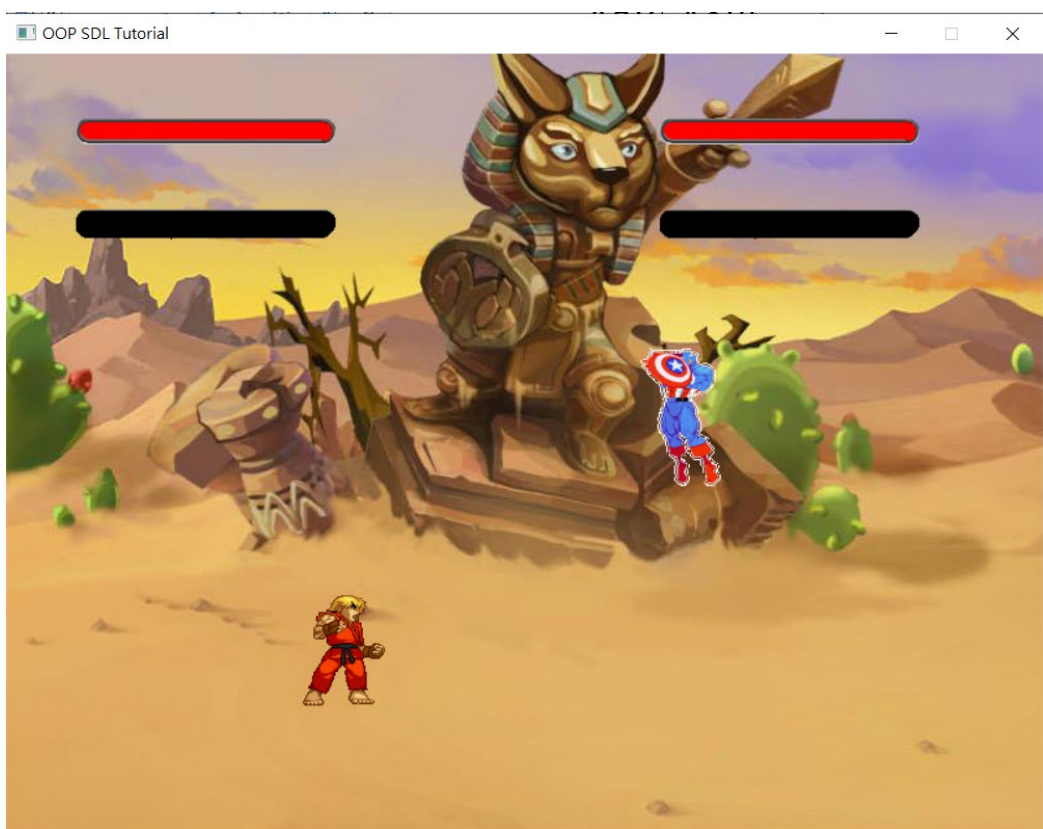
上圖為進入遊戲前的初始畫面(下方"Tap To Continue" 是會閃爍的)



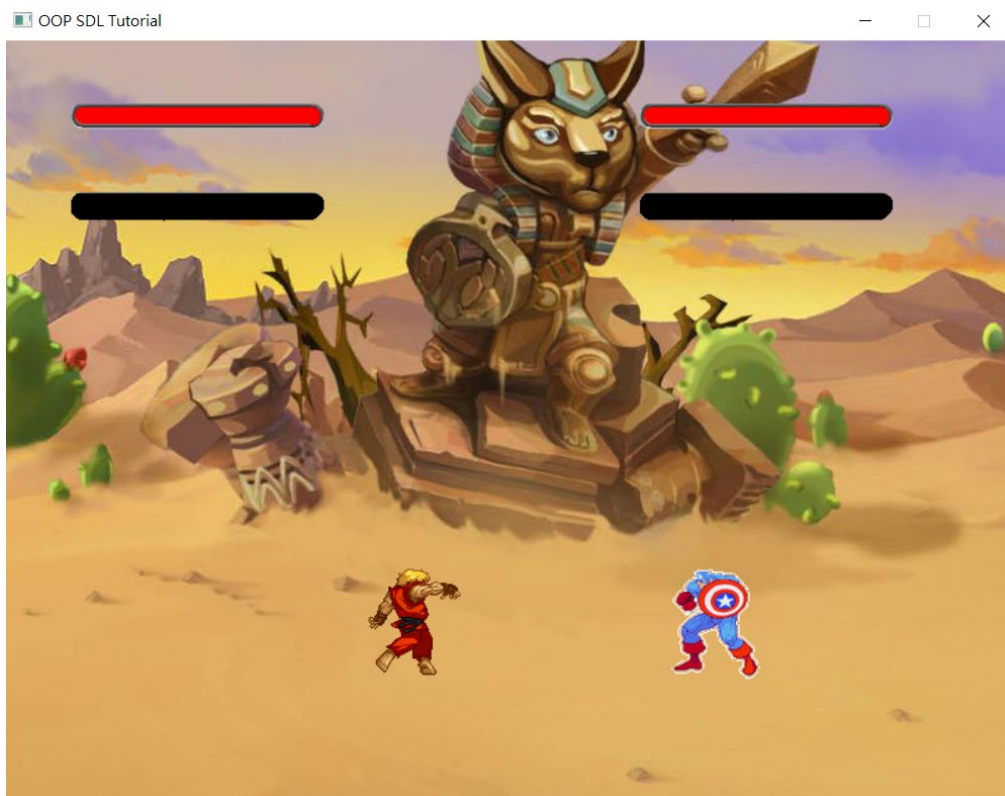
上圖為左右角色走動時的畫面



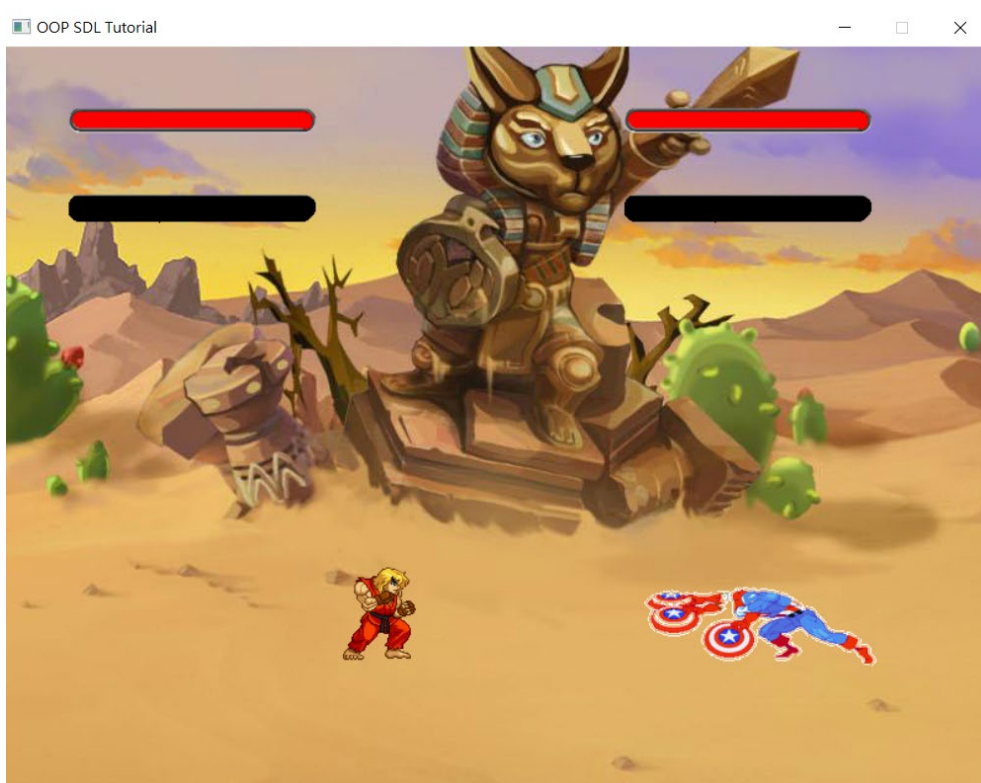
上圖為左玩家跳躍時畫面



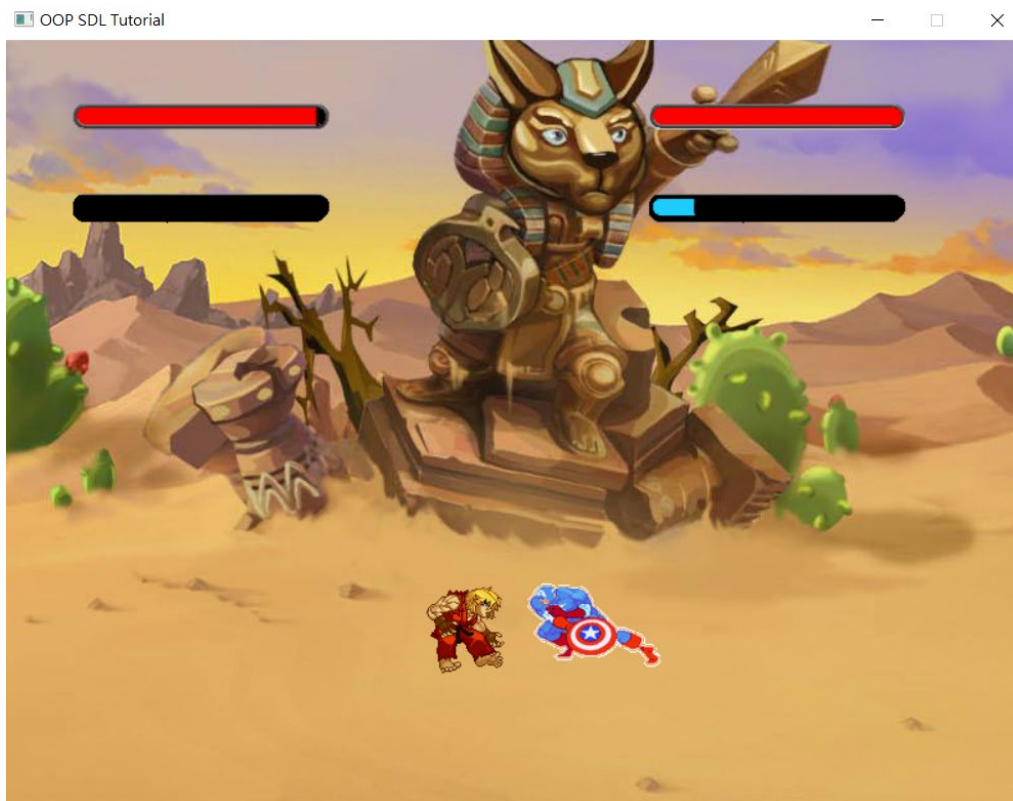
上圖為右玩家跳躍時的畫面



上圖為左玩家近攻時畫面

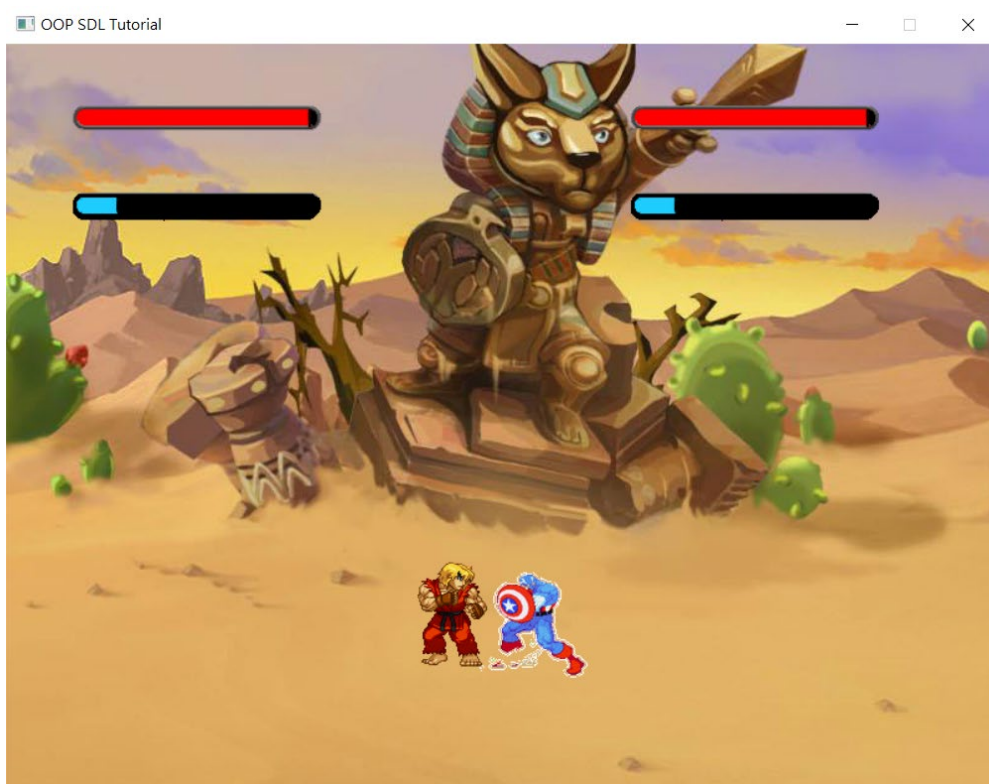


上圖為右玩家近攻畫面



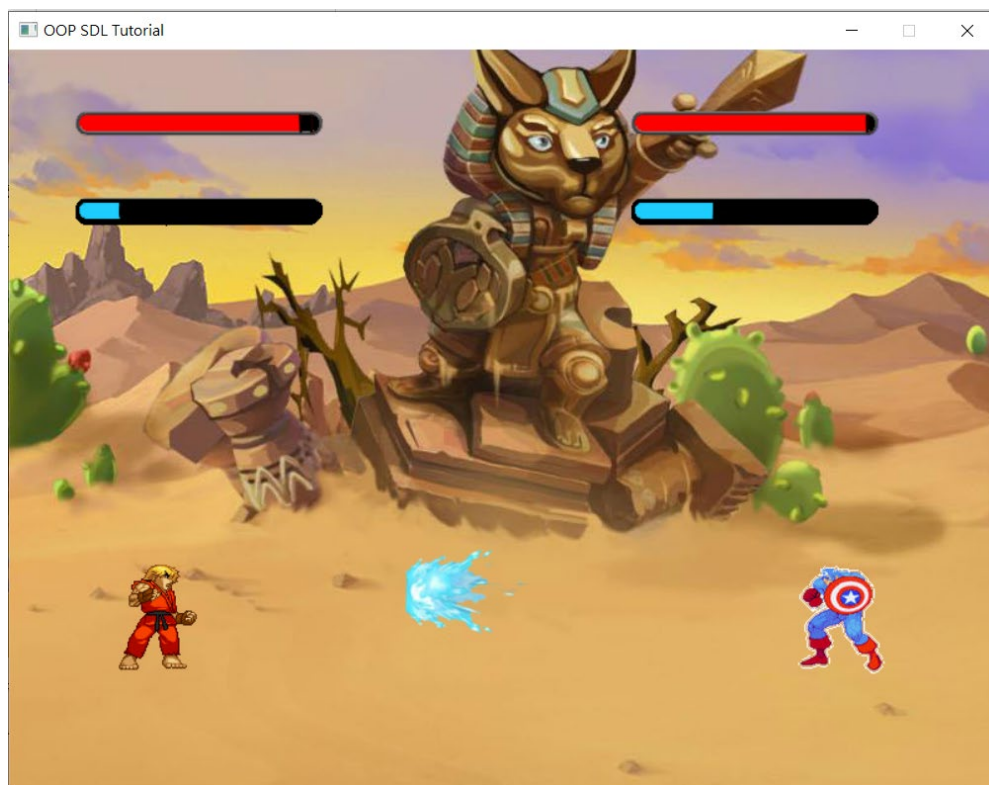
上圖為右玩家近公攻擊左玩家以及左玩家被攻擊時的畫面

此時右玩家能量 **BUFF** 條增加，左玩家血量條減少

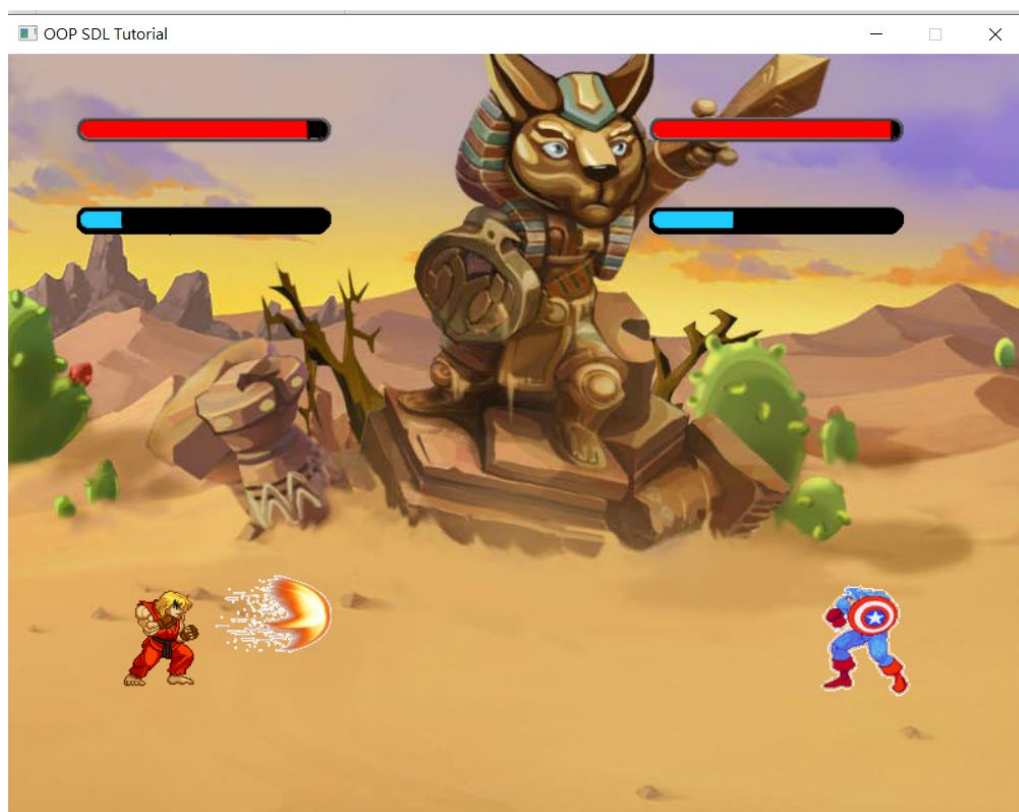


上圖為左玩家攻擊右玩家以及右玩家被攻擊時的畫面

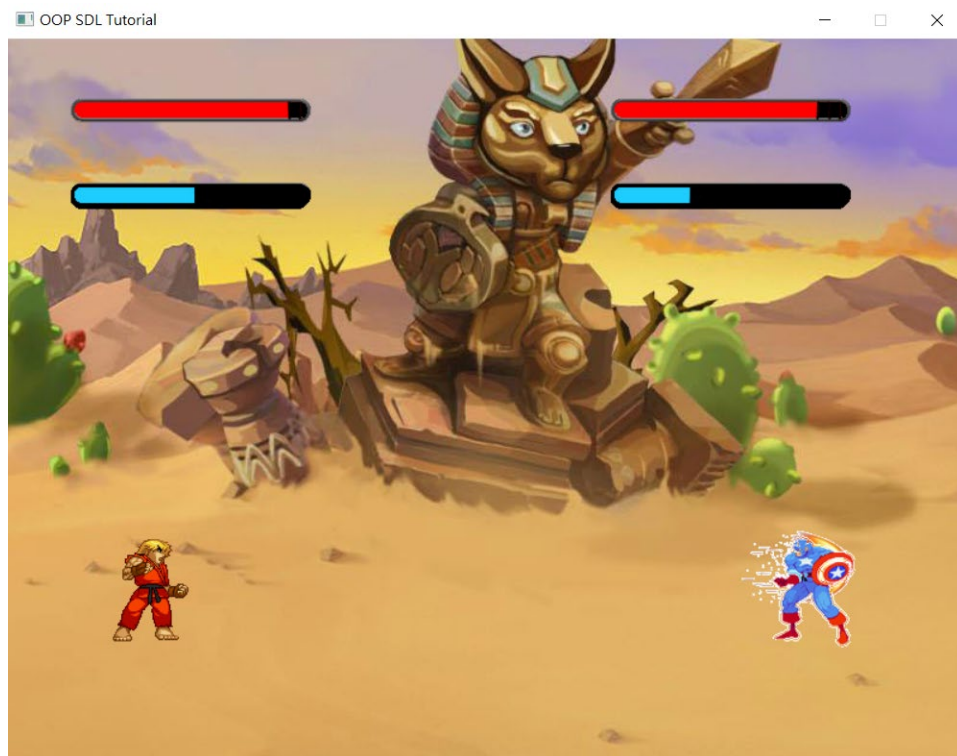
此時左玩家能量 **BUFF** 條增加，右玩家血量條減少



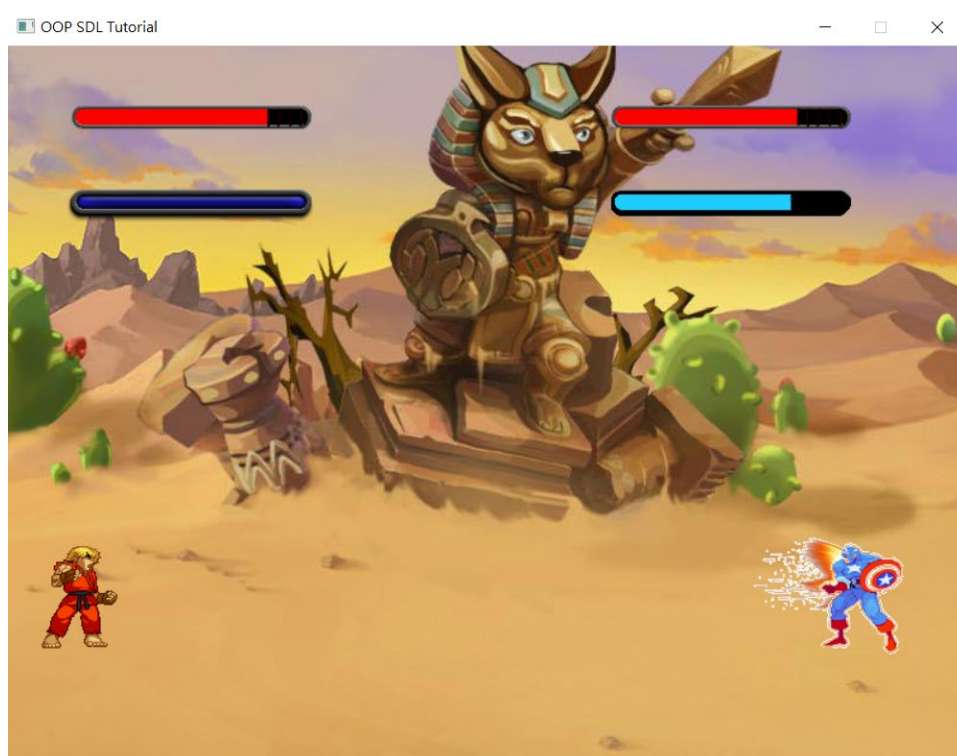
上圖為右玩家遠攻時畫面(BUG:此時左玩家先扣血，右玩家 **BUFF** 先增加)



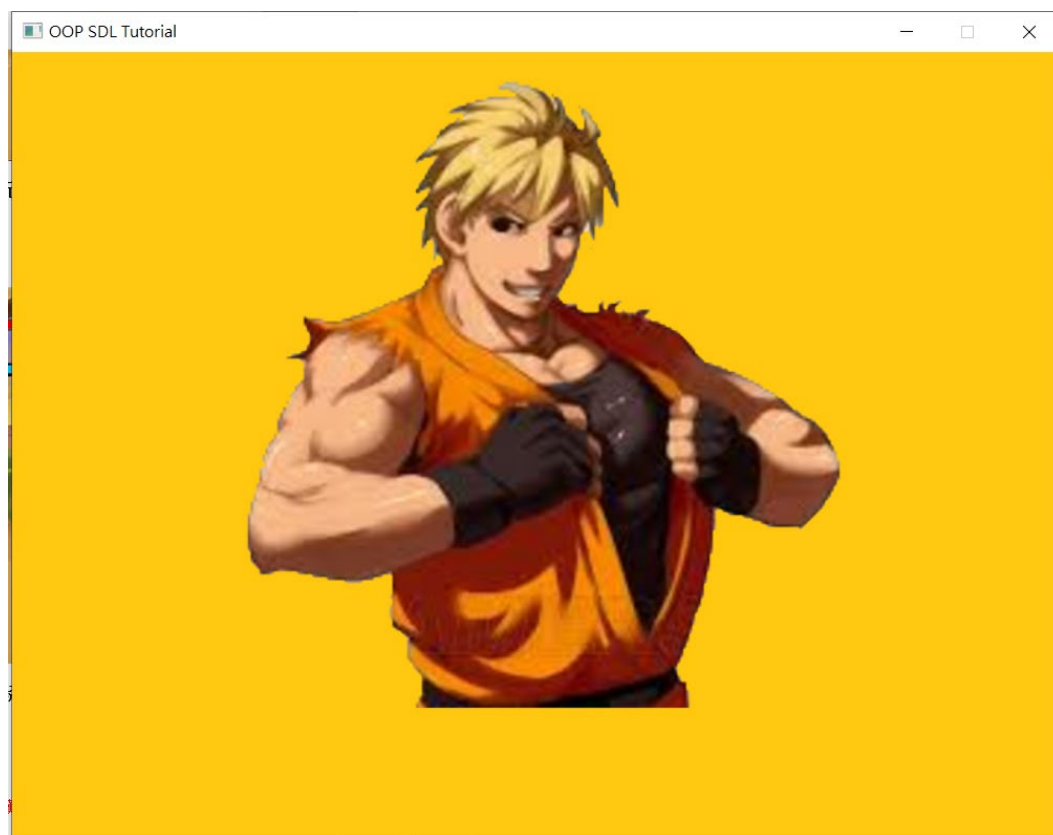
上圖為左玩家遠攻時畫面(此時右玩家血量以及左玩家 **BUFF** 條正常)



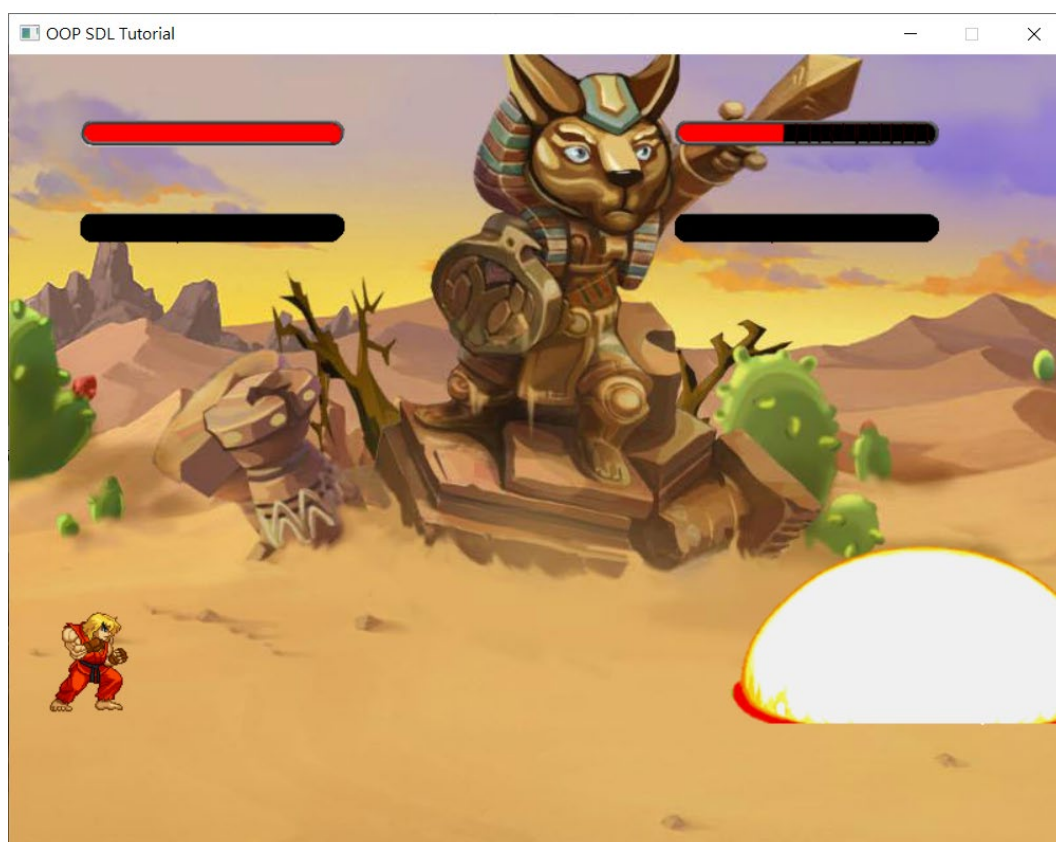
上圖為右玩家被左玩家遠攻，當光波擊到右玩家身體，右玩家血量減少，左玩家 **BUFF** 增加。



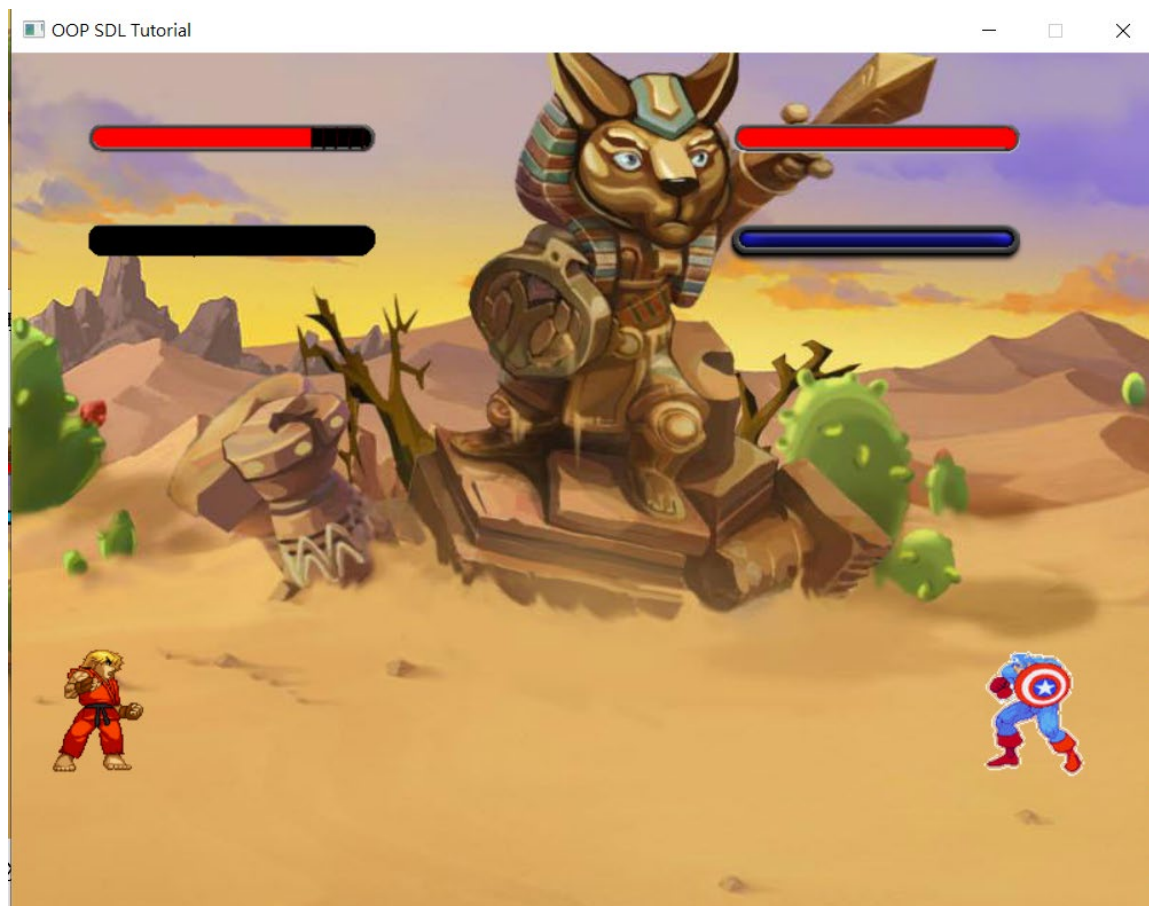
上圖為右玩家 **BUFF** 條集滿，因此可以觸發必殺技



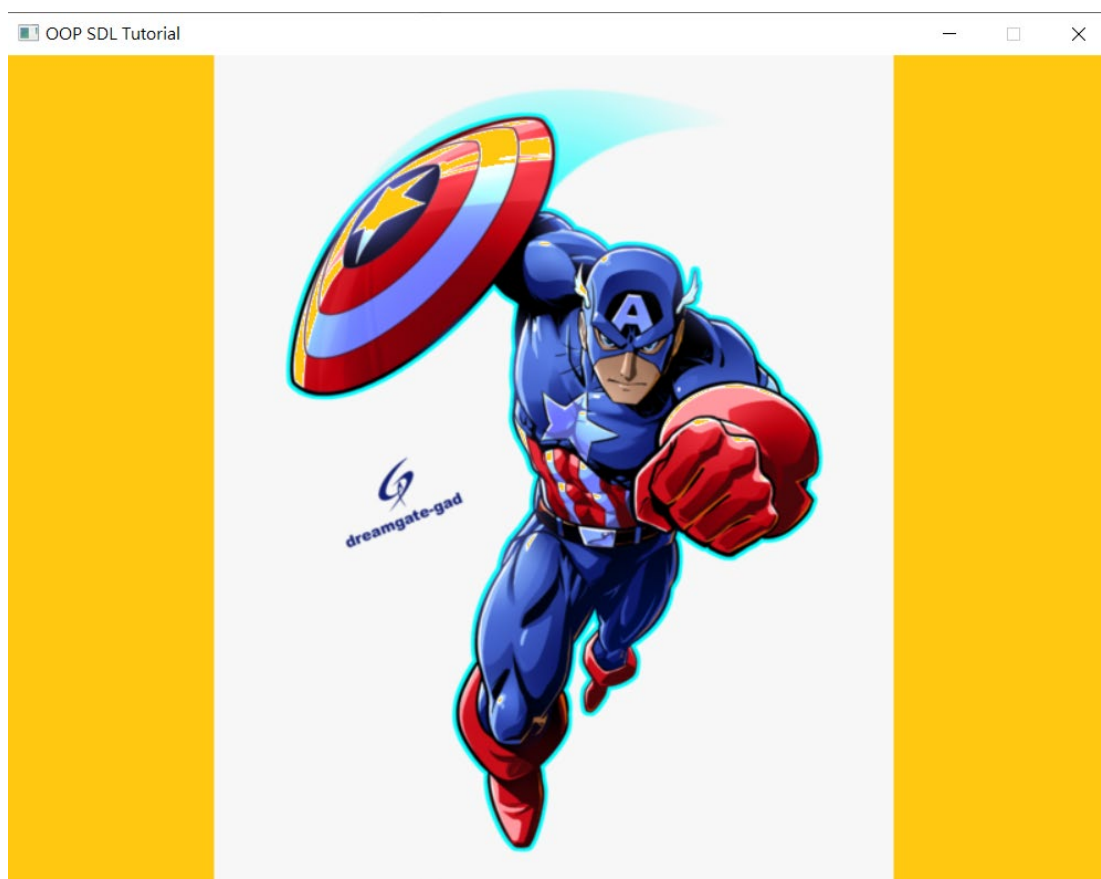
上圖為左玩家必殺技之前置畫面



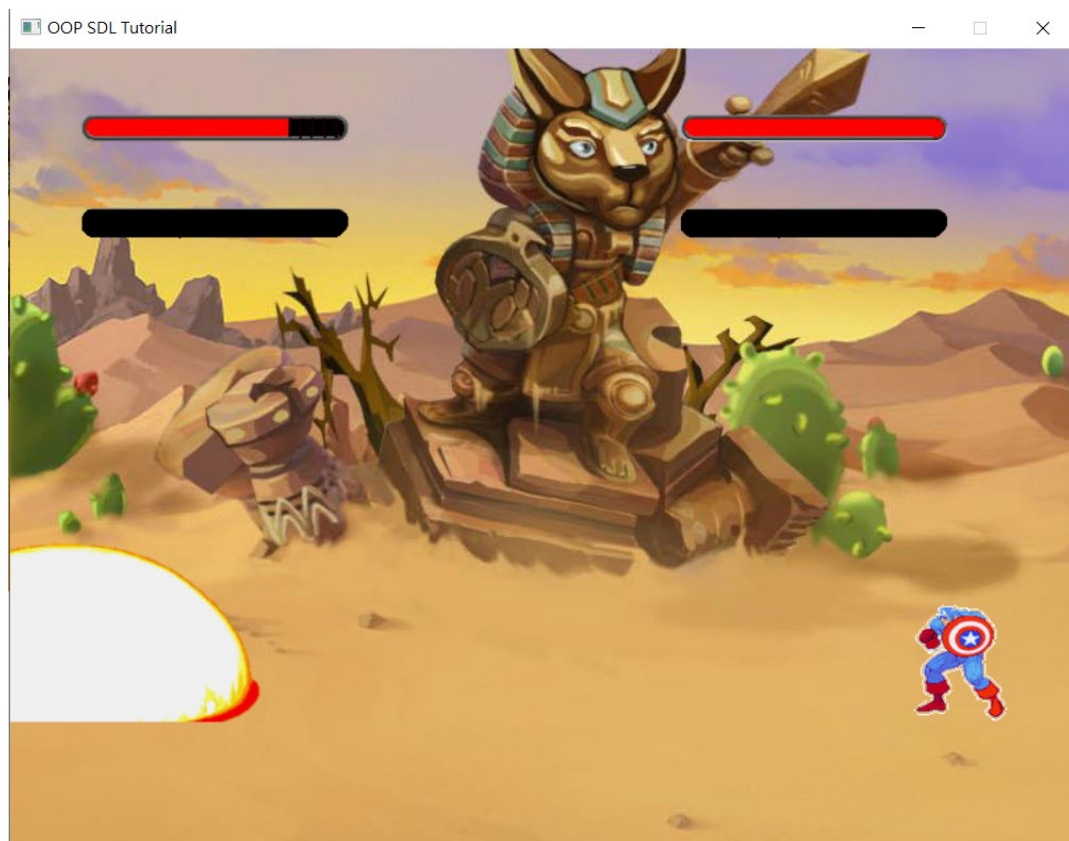
上圖為左玩家必殺技畫面



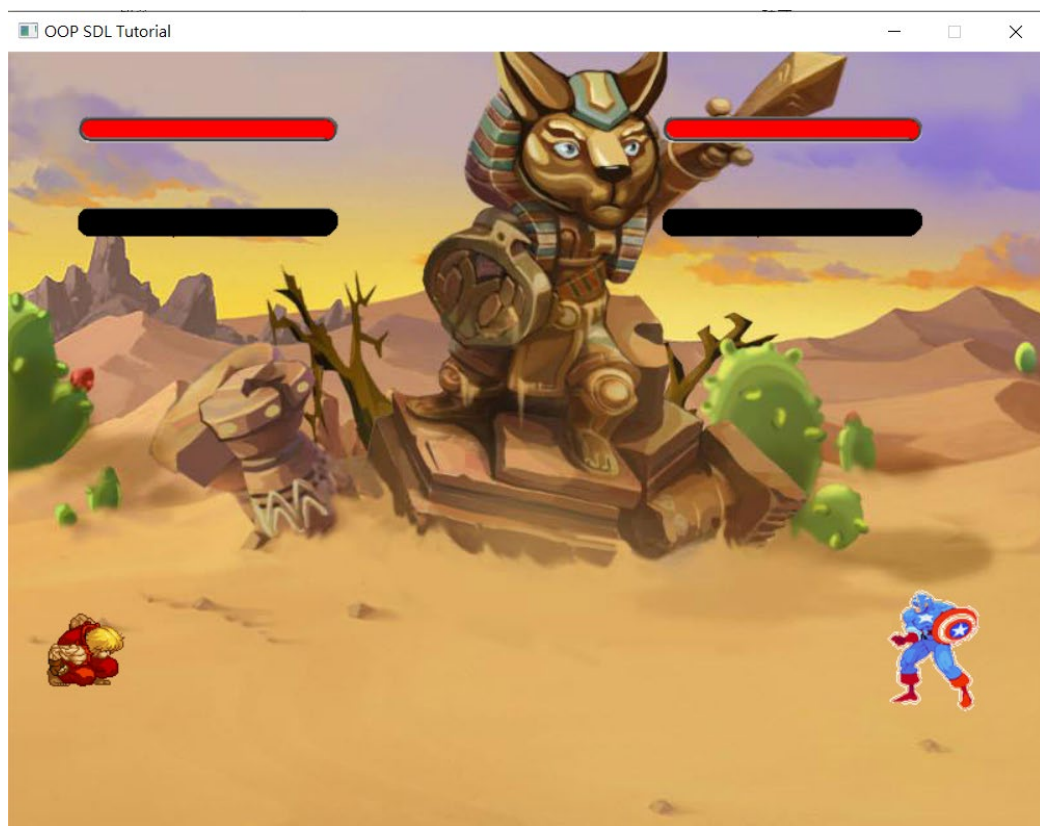
上圖為右玩家 **BUFF** 條集滿，可以觸發必殺技功能



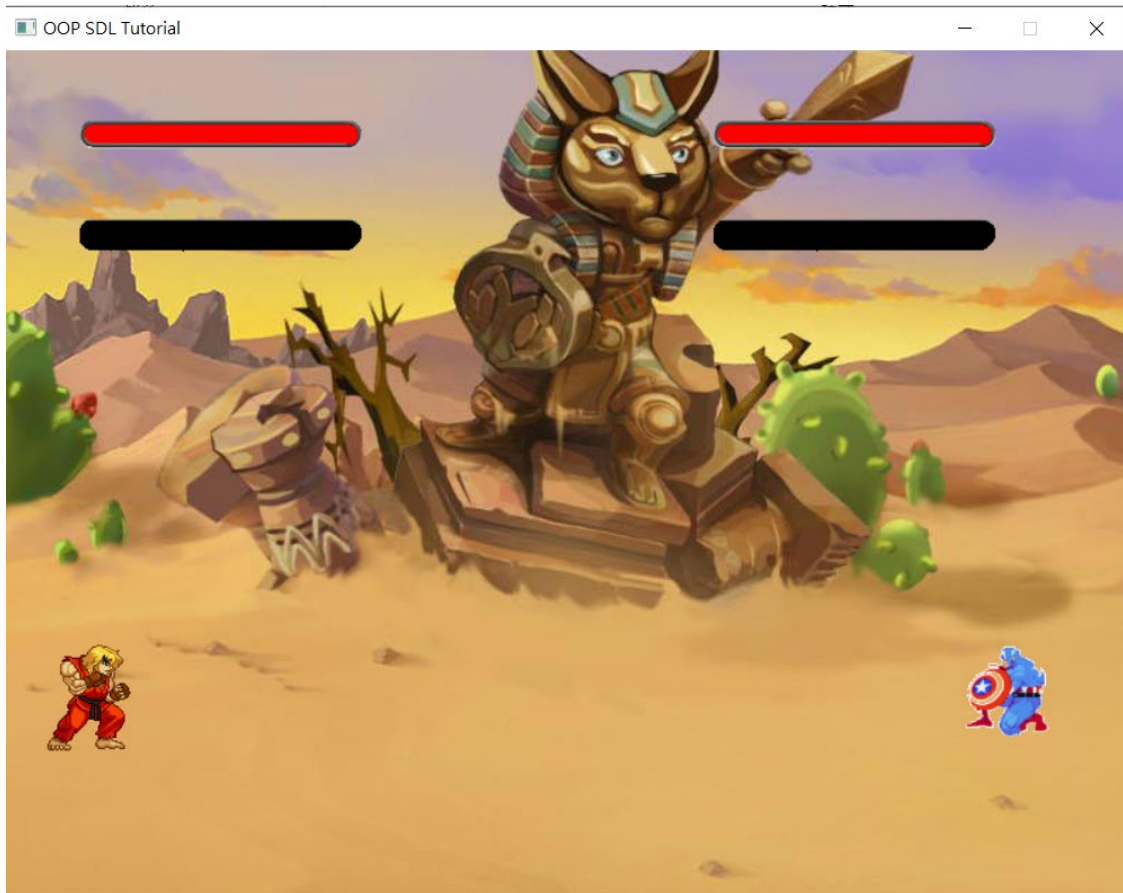
上圖為右玩家必殺技前置畫面



上圖為右玩家必殺技畫面



上圖為左玩家防禦畫面，此時右玩家如何攻擊都不會造成作玩家扣血或是右玩家
BUFF 條增加



上圖為右玩家防禦畫面，此時左玩家如何攻擊都不會造成左玩家扣血或是左玩家
BUFF 條增加

D. 專題關鍵

(請以條列方式說明程式中的關鍵點，將專題中有難度與技巧的問題提出分析說明並敘述如何解決，這部分是專題評分的重點)

此作品最重要的部分還是完整呈現了拳皇此遊戲的本質:攻擊，所有攻擊以及被攻擊、光波、必殺技等等都直接給玩家一個拳皇該有的樣子，攻擊其他玩家會造成對方扣血也能使自己 **BUFF** 能量條增加。

我們花了很多時間在剪輯圖片，裡面所有的圖像都是用了蠻多時間去剪輯的，但是最主要的還是研究攻擊以及被攻擊時的判斷式這部分，尤其是 **timer** 的地方，也許是不夠熟悉，但是最後還是有寫出來，像是剛開始的時候要做走路的樣子，卡在 **timer** 很久。

以下為走路的判斷式。

```
case SDLK_LEFT:
    Mix_PlayChannel(1, f2walkingeffect, -1);
    f2VelX -= VELOCITY;
    //walking動畫沒有進行時
    if (!f2walking_operating)
```

```

{
    f2walking = true;
    f2lasttime = f2curtime;
    f2pressed_walking = true;
}
f2standing_left = true;
f2standing_right = false;
break;

```

.....

以下為攻擊的判斷

case SDLK_j:

```

    Mix_PlayChannel(1, flfrontpunchhiteffect, 0);
    fllast_attack = flthis_attack;
    flthis_attack = 'j';
    flfront_punch_hit_count = true;
    if (!flfront_hit_operating)
    {
        flfront_punch_hit = true;
        fllasttime = flcurtime;
    }
    break;

```

.....

```

    flfrontpunchhitspeedtimediv = (flcurtime - fllasttime) % flfrontpunchhitspeedperiod;

```

.....

//f1

```

else if (flfront_punch_hit)

```

```

{
    if (flfrontpunchhitspeedtimediv >= flfrontpunchhitspeedperiod - 50)
    {
        flfront_punch_hit = false;
        flfront_hit_operating = false;
        printf("fltimediv = %d", fltimediv);
    }
    else
        flfront_hit_operating = true;
    if (flstanding_right)
    {

```

```

        flfrontpunchhitrightIndex = flfrontpunchhitspeedtimediv * F1FRONTPUNCHHITRIGHTFRAME /
flfrontpunchhitspeedperiod;
        imgRender(renderer, flfrontpunchhitright[flfrontpunchhitrightIndex], flPosX, flPosY);
    }
    if (flstanding_left)

```

```

        {
            flfrontpunchhitleftIndex = flfrontpunchhitspeedtimediv * FIFRONTPUNCHHITLEFTFRAME /
flfrontpunchhitspeedperiod;
            imgRender(renderer, flfrontpunchhitleft[flfrontpunchhitleftIndex], flPosX, flPosY);
        }
    }
}

```

以下為必殺技判斷

```

flfuckingkilltriller = loadTexture(flfuckingkilltrillerPath, true, 0xFF, 0xFF, 0xFF);
printf("fl fuckingkilltriller loadTexture\n");
for (int i = 0; i < FIFUCKINGKILLEXRIGHTFRAME; i++)
{
    char st[100];
    sprintf_s(st, 100, "%s13-%d.png", flfuckingkillexrightPath, i + 1);
    flfuckingkillexright[i] = loadTexture(st, true, 0xFF, 0xFF, 0xFF);
}
printf("fl fuckingkillexright loadTexture\n");

for (int i = 0; i < FIFUCKINGKILLEXLEFTFRAME; i++)
{
    char st[100];
    sprintf_s(st, 100, "%s13-%d.png", flfuckingkillexleftPath, i + 1);
    flfuckingkillexleft[i] = loadTexture(st, true, 0xFF, 0xFF, 0xFF);
}
printf("fl fuckingkillexleft loadTexture\n");

for (int i = 0; i < FIFUCKINGKILLWEAPONFRAME; i++)
{
    char st[100];
    sprintf_s(st, 100, "%sboom-%d.png", flfuckingkillweaponPath, i + 1);
    flfuckingkillweapon[i] = loadTexture(st, true, 0xFF, 0xFF, 0xFF);
}
printf("fl fuckingkillweaponleft loadTexture\n");

.....
int flfuckingkilltrillerspeed = 2, flfuckingkilltrillerspeedtimediv = 0;
int flfuckingkilltrillerperiod = 500;
int flfuckingkilltrillerspeedperiod = flfuckingkilltrillerperiod * flfuckingkilltrillerspeed;

int flfuckingkillexspeed = 2 * 2, flfuckingkillexspeedtimediv = 0;
int flfuckingkillexperiod = 500;
int flfuckingkillexspeedperiod = flfuckingkillexperiod * flfuckingkillexspeed;

```



```
int flfuckingkillweaponspeed = 2 * 2, flfuckingkillweaponspeedtimediv = 0;
int flfuckingkillweaponperiod = 500;
int flfuckingkillweaponspeedperiod = flfuckingkillweaponperiod * flfuckingkillweaponspeed;
```

.....

```
case SDLK_u:
    if (flfucking_kill_on && !flfucking_kill_triller_operating)
    {
        Mix_PlayChannel(1, flfuckingkilltrillereffect, 0);
        fllast_attack = flthis_attack = '0';
        flfucking_kill_triller = true;
        fllasttime = flcurtime;
    }
    break;
```

E. 心得感想

(就是心得與感想)

潘丞詡:

我真的覺得自己在程式語言這一塊需要很多時間去加強，從大一到現在都是這麼覺得。當然物件導向這一門課程真的讓我學到很多，雖然絕大多數時間我是落後的，但是我都很認真在跟著。為了這個專題，我和隊友花了很多時間，一大堆的夜晚。很多時候隊友也花很多時間在教我如何打出相關的判斷，我很感激他，他真的很 Carry!

不管是判斷還是函式，我清楚知道自己哪裡落後，也知道自己要加強哪些部分需要加強，當然還是希望已練到可以不看參考資料向式講義或是書本，當然這是理想狀態，我離那兒還有很長一段距離，我真的很希望自己可以再聰明一些，或是哪裡卡住解不開。我相信自己是學得來的，只要多一點時間我覺得是有機會的。

謝謝蔡媽持續教導著程式語言，我對於他是不會放棄的，我一定會加強打 Code，因為打 Code 真的在未來是很重要的一項技能，我知道我跟其他同學相差甚遠，但是我真的會努力!!暑假期間我會好好加強程式語言!希望兩個月時間可以拉近一點距離。

陳子祈:

這次程式碼主要由我來設計與規劃，在這次的程式設計專題中我才知道看起來很簡單的遊戲，背後有非常多判斷式，很考驗遊戲設計者的邏輯思維與耐心。還好我組員跟我配合得很好，我們兩人才能齊心協力完成這一次遊的遊戲設計，我們的工作分配主要是動畫用到的圖片交給組員，我主要設計整個程式的流程，我們都從中了解檢及圖片的技巧及應用在動畫 SDL2 上的技巧，我覺得我收穫很多，感謝蔡老師給我們這個機會好好學習實作技巧!

F. 參考資料

(請條列出所參考過的所有資料)

<https://www.sprisers-resource.com/>