Intro to Computer Graphics HW4

Group 1

《Champion Spotlight of Jinx》

111550040 資工三 曾紹幃
111550149 資工三 林悅揚
111550106 資工三 何義翔

I. Title

Story name:《Champion Spotlight of Jinx》
groupID: 1
members of team：曾紹幃、林悅揚、何義翔

II. Introduction

模仿英雄聯盟介紹角色的系列影片－Champion Spotlight，來介紹 Jinx 這位角色，我們的動畫會模擬 Jinx 在遊戲中的四個技能，分別是機關槍火箭炮普通攻擊，雷射槍，手榴彈，和火箭炮飛彈攻擊。利用 Geometry shader 來模擬爆炸的效果。

III. Implementation Details

1. Object Modeling

利用 sketchfab 尋找 Jinx 和其他素材的 model 和其對應的 texture 並利用 blender 將要動畫的部分進行切塊的處理，例如將 Jinx 的人物切成 body、righthand、lefthand。再將其以 .obj 檔導出，以便後續 GLSL 的處理。

2. Environment map

從該網站獲得遊戲地圖的 3D 模型，利用 Blender 渲染並拍攝六張照片構成 cubemap 當成背景。

3. 爆炸特效 & 顏色

使用 Geometry Shader，讓每個三角形隨著時間沿法線向量射出，由於每個物體使用相同的 Geometry Shader，在 Shader 中會利用 time 判斷是否需要爆炸。子彈的爆炸效果是利用其飛行距離，超過一定的值後便開始計時，將此時間傳進 Geometry Shader 便可達到子彈爆炸的效果。而手榴彈(E)的效果，則是當其抵達地面時開始計時，先在原地旋轉與跳躍，待一定時間後再開始爆炸。

爆炸後碎片的顏色會隨著時間而更改。將設定好的爆炸顏色傳入 Geometry shader，利用以下程式計算該時間的顏色並傳至 Fragment shader。

```
ExplosionColor = vec4(aExplosionColor, min(0.0, 1.0 - time * 2.0));
mixValue = min(1.0, time * 2.0);
```

在 Fragment shader 則利用以下程式將 Fragment 著色

```
FragColor = mix(texture(ourTexture, TexCoord), ExplosionColor, mixValue);
```

4. **Camera Movement**

為了讓錄製影片的時候，攝影機的視角能夠更加自由，我除了在 HW3 上面 camera 移動的基礎上，再加上向上下旋轉、攝影機左右平移。開始以為這部分會很簡單，但是實作起來比想像中困難很多。Camera 向前後旋轉的部分，不能夠直接對 X 軸進行旋轉，而是需要先利用 cameraUP、camera lookat 的外積，計算 Camera 的左邊，再讓 Camera 以 Camera 的左邊進行旋轉。

```
glm::vec3 left = glm::normalize(glm::cross(camera.up, currentLook - camera.position));
cameraModel = glm::rotate(cameraModel, glm::radians(camera.rotationX), left);
```

Camera 平移的部分則更加困難。與旋轉一樣，要先利用外積計算左邊，然後再讓 camera lookat 與 camera position 同時向左/右平移。困難的點是這邊的 camera position 與 camera model 的 position（實際位置）並不相同，因為會受到旋轉的影響。原本是想要找到 model position 對應的 camera position，但後面發現直接把 rotation 設置為 0 更加的快速，因此最後是把 camera position 改為平移後的 model position，同時把 rotation 重製為 0，達到平移的效果。

我還有加入按鍵 P，重製 camera 的 position 到初始位置。

5. **子彈時間（T）**

將所有有關於 count 的數值全部乘以一個 time coefficient，按下 T，Time Coefficient 會從 1 → 0.1，因此會有慢慢減速的效果；再次按下 T，Time Coefficient 會從 0.1 → 1。這部分實作十分簡單，但畫面效果很好，是我最喜歡的一個 part。

6. **切換技能**

在 load Jinx 的時候，會一次 load 3 個 Jinx model，分別拿著機關槍（Q 一般普攻）、小槍（W）、鯊魚槍（R、Q 火箭炮普攻）。按下不同技能的時候，會切換model，並同時讓手做旋轉，就會有切換槍的效果。

而 E 只是把手旋轉的方向相反，就會像是把手榴彈丟出去的效果。

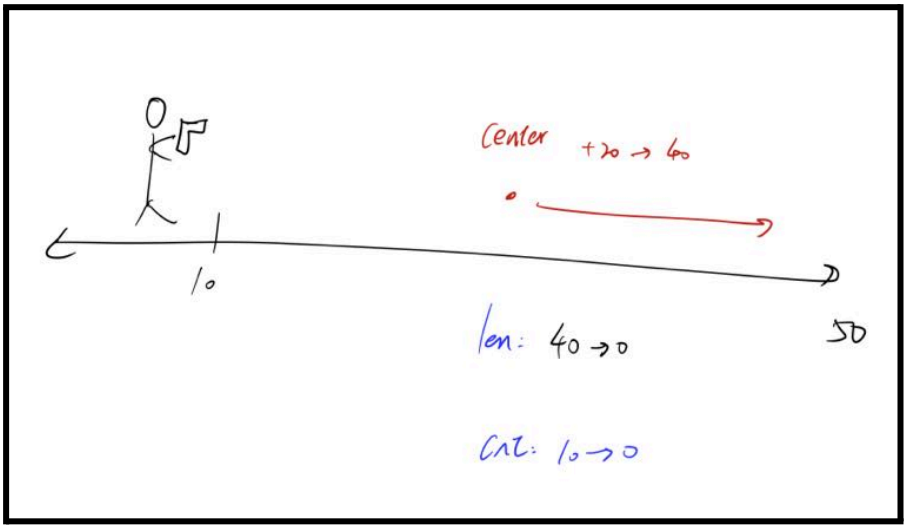
而每次切換技能的時候，手的旋轉都會重製，這樣就會有類似於槍戰遊戲切槍的效果。

7. 子彈(2 種 Q 普攻、R)

按下 A 之後，會把子彈的位置資訊加入一個 vector，3 種子彈會加入不同的 vector，而在 render 的時候，會遞迴那個 vector，把所有bullet 都 render 到畫面上。在 update 的時候，每次都會遞迴 vector，把位置加上一個 value 做移動。而當 bullet 超出一個範圍的時候，就會執行上面的爆炸效果。

8. W 技能

W 技能使用 cube 去渲染。一開始會先產生長方形，後來長方形慢慢地往W的最遠方縮短，這是依靠一個 counter 去執行，每次 update 會更新這個 counter。counter 從 10 → 0，center point 20 → 40，長方形的長度 40 → 0，如圖片所示。



9. E 技能

E 技能設定三個不同初始的速度的手榴彈，根據加速度 (0, -50, 0)，來調整手榴彈的速度以及位置，使手榴彈呈現拋物線的運動軌跡，直到 `position.y <= 0`，就停住接上彈跳的動畫和爆炸的特效。

```
std::vector<glm::vec3> velocitiesE = {
    glm::vec3(0.0f, velY, velZ),
    glm::vec3(velX, velY, velZ),
    glm::vec3(-velX, velY, velZ)
};
```

```
glm::vec3 acclerateE = glm::vec3(0.0f, -50.0f, 0.0f);
```

```
// Bomb
for (int i = 0; i < bombs.size(); i++) {
    bombs[i].velocity += bombs[i].acceleration * deltaTime * (timeCoef * !stop);
    bombs[i].position += bombs[i].velocity * deltaTime * (timeCoef * !stop);
```

IV. Discussion

Proposer(何義翔):提出可以做奧術的小動漫，一開始 Jinx 用很多槍跟 Vi 打架，利用 Geometry Shader 讓角色互打爆炸，流血等特效，人物互毆臉會瘀青或凹陷等等。
Critic(曾紹幃):太短的的動畫很難表達所有劇情，人的 model 很複雜，很難做出很浮誇的特效，加上大家還有很多其他期末作業這樣可能會來不及做完，而且我還沒看奧術，如果不小心做出劇情，會劇透到我。
Negotiator(林悅揚):不然我們做 Jinx 英雄聚光燈，就是 lol 官方介紹英雄技能的影片，並且只做 QWER 四個技能就好，簡化我們要做的事，我們只要做好切槍，丟手榴彈，爆炸等動畫就可以表現好了，其他內容可以留在解說，這樣能保持 lol 的元素，又可以確保實行的可能性。

V. Work assignment:

- 何義翔(HackMD 大師):爆炸特效製作、生成背景
- 曾紹幃(OpenGL 大師):各種技能製作、Camera 移動製作
- 林悅揚(大導演):E 技能製作、object / texture 處理、影片導演、剪片負責人

VI. References:

- https://p3dm.ru/files/others/others/15244-summonersrift-map.html#google_vignette
- https://skfb.ly/oJvYT
- https://skfb.ly/6QSxx
- https://skfb.ly/nk4j32fd0
- https://skfb.ly/oQZqX
- https://skfb.ly/oC78N
- https://www.youtube.com/watch?v=FKXRiAiQFiY

VII. Results:

- https://youtu.be/SIGnUQT0NhY
- https://github.com/Sean20405/NYCU-ICG-Final