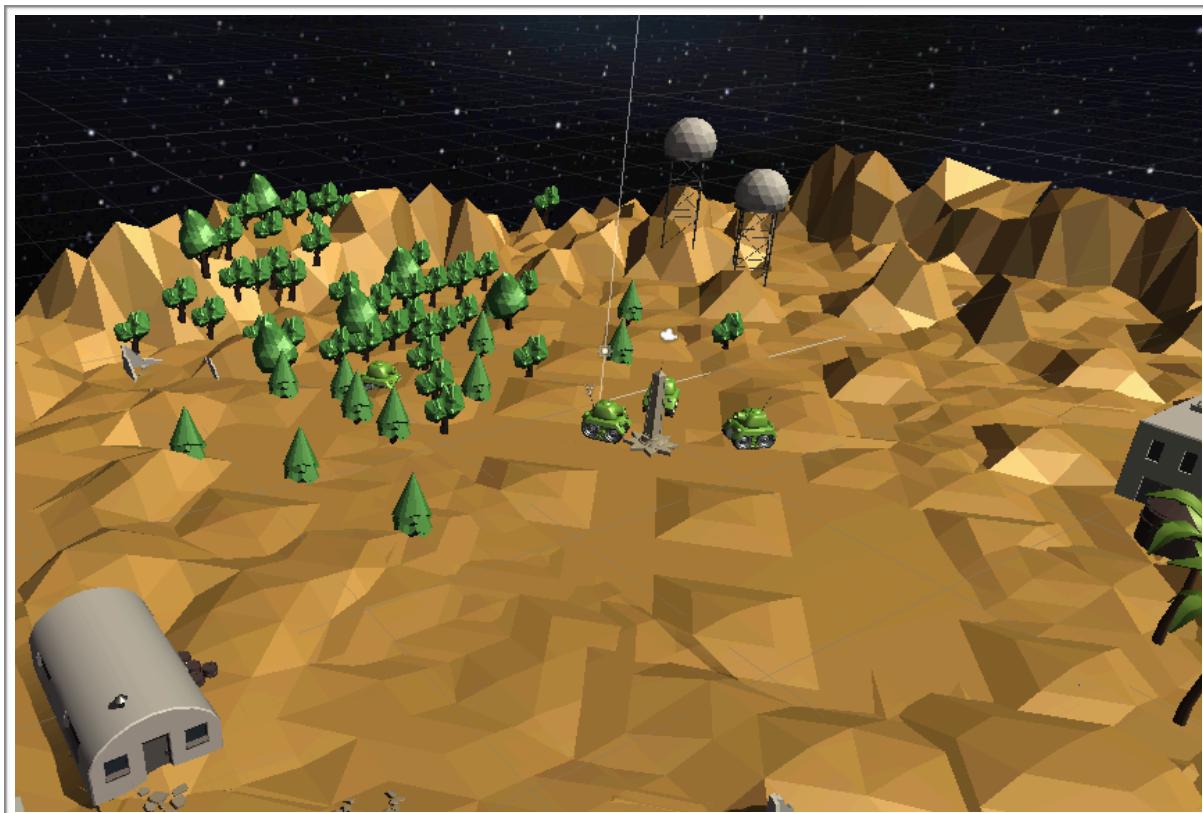


# 2018 虛擬實境 Final Project



星球塔防遊戲

B03201010 數學四

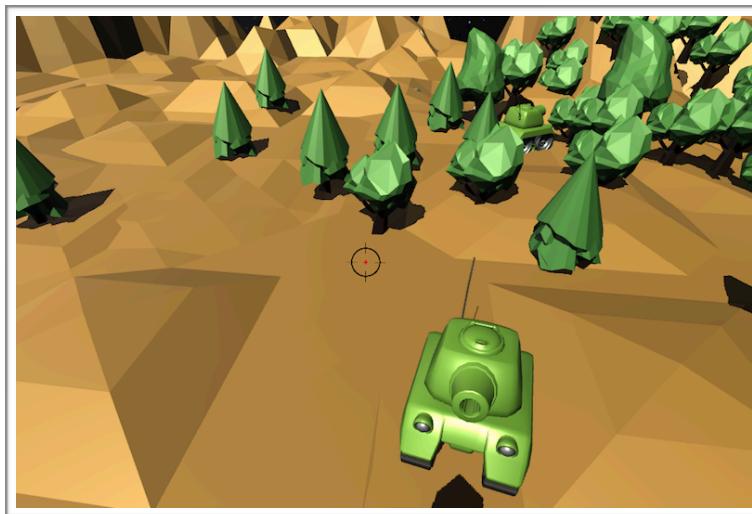
黃明翰

# 虛擬實境 Final Project

## 星球塔防遊戲

### 前言

第一次上課時教授就向同學們展示了以前學長做的VR遊戲，令我對於這堂課產生了濃厚的興趣，因此想要嘗試做出一款屬於自己的小遊戲。這是一款塔防遊戲，遊戲中坦克車會從四面八方攻過來，玩家要試著保衛自己的塔並獲取高分。因為能力的不足，這個遊戲裡的物件都是來自網路或是作業中的建築物，希望未來可以在美術及介面方面做更大的改進。



### 遊戲內容

在這專案中，除了基本的物件，我在遊戲裡也加入了許多輔助性的物件，或是增加遊戲性的關卡。以下分別介紹：

#### 1. 地圖

遊戲地圖為一個長方形的地圖，地圖中有森林、軍事基地以及沙漠，玩家會在正中間的塔防衛四個重生點(紅點)產生的坦克車進行的攻擊。

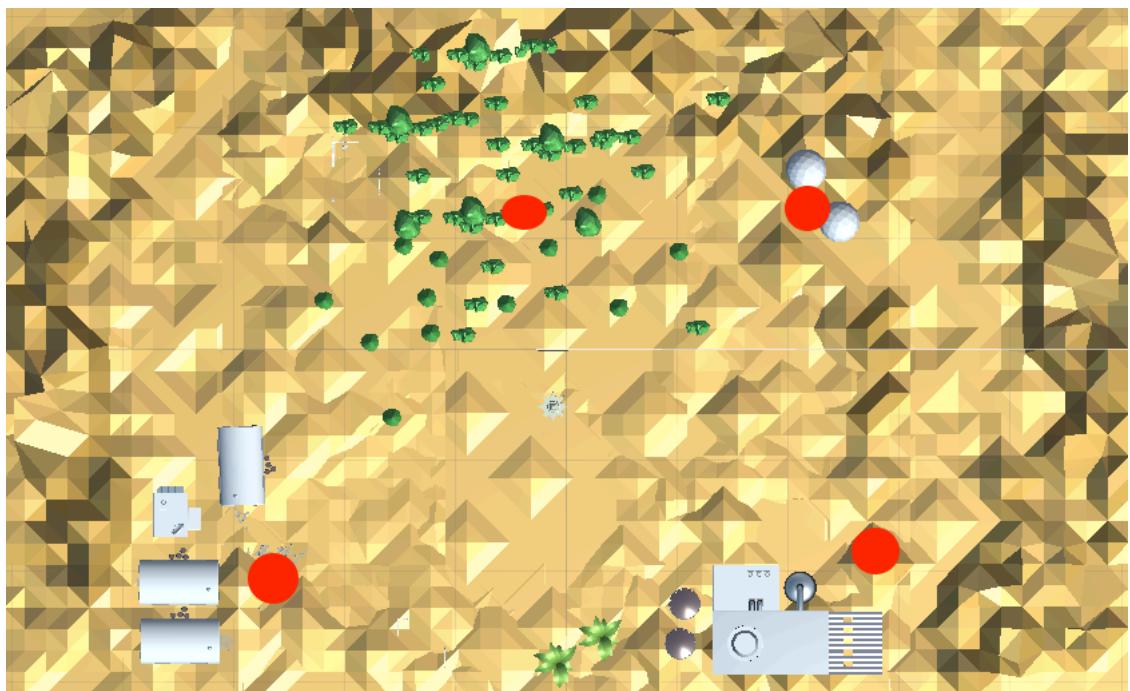


Fig.1 地圖

## 2. 地圖物件

### 2.1.塔

塔位於地圖正中間，是玩家主要要保衛的對象。

每次遊戲塔會有三條命，被坦克車撞到會少一條命，若塔的生命歸零了，遊戲就結束。



Fig.2 塔

### 2.2.坦克車&Boss

坦克車是這個遊戲的主要敵人，會從地圖中的四個重生點以不同的速率生成。每擊敗一輛坦克可以獲得一分，而擊敗大台的坦克(Boss)則可以獲得五分，是快速得到高分的途徑。



Fig.3 由左至右為一般敵人、Boss以及爆炸特效

### 3. 準星

準星位於畫面正中央，可以幫助玩家瞄準坦克。在實際遊戲後我覺得瞄準太困難了，因此加入這個輔助物件。



Fig.4 準星

### 4. 計分板

計分板位於畫面左上角，會實時顯示塔剩餘的血量以及目前的分數。

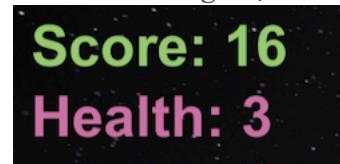


Fig.5 計分板

### 5. 雷達

雷達位於畫面左下角，會實時顯示目前敵人與塔的相對位置，讓讓玩家更好判斷目前敵人的數量以及敵人的位置。右圖中藍點是塔的位置，紅點是敵人坦克，大紅點是Boss坦克。

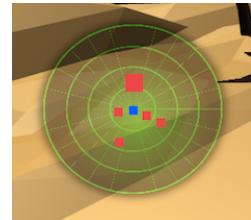


Fig.6 雷達

## 程式碼

我在這個遊戲中嘗試做了許多不同的物件來增加遊戲性，例如擊敗後會得到高分的大坦克、輔助性的雷達以及主菜單中的高分榜。接下來將對於較特殊的功能進行程式碼的介紹：

### 1. 主選單

#### 1.1. Play Botton

```
public void PlayGame ()  
{  
    SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);  
}
```

在這個使用到的是ScenceManager的功能。在Build時會有Build的編號，因此我們可以藉由+1or-1來在不同的Scence中切換。

<input checked="" type="checkbox"/> Scenes/menu2	0
<input checked="" type="checkbox"/> Scenes/SampleScene	1

Fig.7 Build Setting

而要實現按鈕的感覺，也就是當你指向按鈕時按鈕會顯示不同顏色表示你正在按他，則可以利用Unity中Button的功能讓他的透明度在高低不同間轉換。



Fig.8 按下的  
按鈕特效及  
設定。

## 1.2. High Score

在主畫面會有個高分榜，在每次遊戲時紀錄玩家遊玩時獲得的最高分數，讓玩家可以不斷的挑戰自我。

```
if (ScoreScript.scoreValue > HighScore.HighscoreValue)
    HighScore.HighscoreValue = ScoreScript.scoreValue;
```

要實現這個功能其實不難，我寫了一個Script叫HighScore並將他用在主畫面中顯示高分榜，並在獲取分數時將新的分數與高分榜的分數比較，若較高則更新高分榜的分數。

## 2. 生成坦克

會在設定好的重生點不斷產生以塔為目標的坦克車。

坦克車的生成頻率可以做調整，並會在遊戲中越生越快。

```
void Spawn()
{
    //在重生點重生敵人
    GameObject enemyObj = Instantiate(enemy, transform.position, transform.rotation) as GameObject;
    enemyObj.transform.parent = transform;
    //設定目標
    enemyObj.GetComponent<EnemyNavigation> ().target = target;
    //繼續重生循環
    Invoke("Spawn", spawnInterval + Random.Range(-spawnVariance, spawnVariance));
}

void Update()
{
    //最小重生時間為一秒
    if (spawnInterval > 1f)
    {
        //讓重生越來越快(每50秒降低1秒)
        float timeReduction = Time.deltaTime / 50;

        //確保亂數時間不會過低
        spawnInterval = Mathf.Max(1f, spawnInterval - timeReduction);
        spawnVariance = spawnInterval * .5f;
    }
}
```

### 3. 射擊

```
if (Input.GetButtonDown("Fire1")) //自動射擊 = X
//void AutoShooting() //自動射擊 = 0
{
    //...播放槍聲...
    gunFireAudio.Stop();
    gunFireAudio.Play();

    //...射出射線
    if (Physics.Raycast(transform.position, transform.forward, out rayHit, 100f))
    {
        //播放撞擊特效
        impactEffect.transform.position = rayHit.point;
        impactEffect.transform.rotation = Quaternion.Euler(270, 0, 0);
        impactEffect.Stop();
        impactEffect.Play();

        //如果是敵人就消滅物件
        if (rayHit.transform.tag == "Enemy")
        {
            explodeEffect.transform.position = rayHit.point;
            explodeEffect.transform.rotation = Quaternion.Euler(270, 0, 0);
            explodeEffect.Stop();
            explodeEffect.Play();
            ScoreScript.scoreValue += 1;
            Destroy(rayHit.transform.gameObject);
        }
    }
}
```

第一行設定接受到射擊的指令(滑鼠左鍵)後會進行射擊，之後會用impactEffect在子彈接觸到的物件上播放撞擊的特效，最後再用一個if函式判斷是否有攻擊到敵人(將坦克物件的tag設定為Enemy)，若是則播放爆炸的效果、加分並摧毀坦克物件。

### 4. 計分板

```
// Update is called once per frame
void Update () {
    score.text = "Score: " + scoreValue;
}
```

此函式會不斷的更新目前的分數，塔的血量的計分方式亦相同。

## 5. 雷達

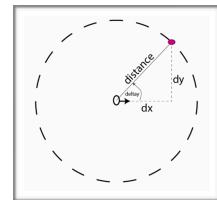
```
void DrawRadarDots()
{
    foreach (RadarObject ro in radObjects)
    {
        Vector3 radarPos = (ro.owner.transform.position - playerPos.position);
        float distToObject = Vector3.Distance(playerPos.position, ro.owner.transform.position) * mapScale;
        float deltay = Mathf.Atan2(radarPos.x, radarPos.z) * Mathf.Rad2Deg - 270 - playerPos.eulerAngles.y;
        radarPos.x = distToObject * Mathf.Cos(deltay * Mathf.Deg2Rad) * -1;
        radarPos.z = distToObject * Mathf.Sin(deltay * Mathf.Deg2Rad);

        ro.icon.transform.SetParent(this.transform);
        ro.icon.transform.position = new Vector3(radarPos.x, radarPos.z, 0) + this.transform.position;
    }
}

// Update is called once per frame
void Update ()
{
    DrawRadarDots();
}
```

因為我們要實時更新物件在地圖上的位置，並根據視角的轉動一起轉動物件在雷達上的位置，這邊主要是利用三角函數來進行位置的計算。

我將這邊的playPos設定為camara的位置，因此他就會不斷讀取我的視角方向，並以此來決定物件在雷達上會顯示的位置。



## 結論

第一次做遊戲後才發現從零到有做出一個遊戲會面臨到多少的困難以及Bug，難怪遊戲公司都要分好多的部門才能完成，更不用說第一次做就是做3D的遊戲，許多物件在空間中的位置都會因為Camera的位置而要調整，我覺得這一點是最困難的。在這次Project中學到非常多，尤其是C#語法的運用，C#對我來說在這個專案開始前完全是一個陌生的語言，而為了加入各種不同的功能，我在網路上參考各式各樣不同的教學才慢慢將這個遊戲的雛形做出來，也因此對於C#有了更進一步的認識。

未來有機會的話，對於這個遊戲我想在美術方面更加著墨，除了將介面變得更好看之外，也將坦克或地圖換成更有個人特色的物件，增加遊戲個人化的程度。