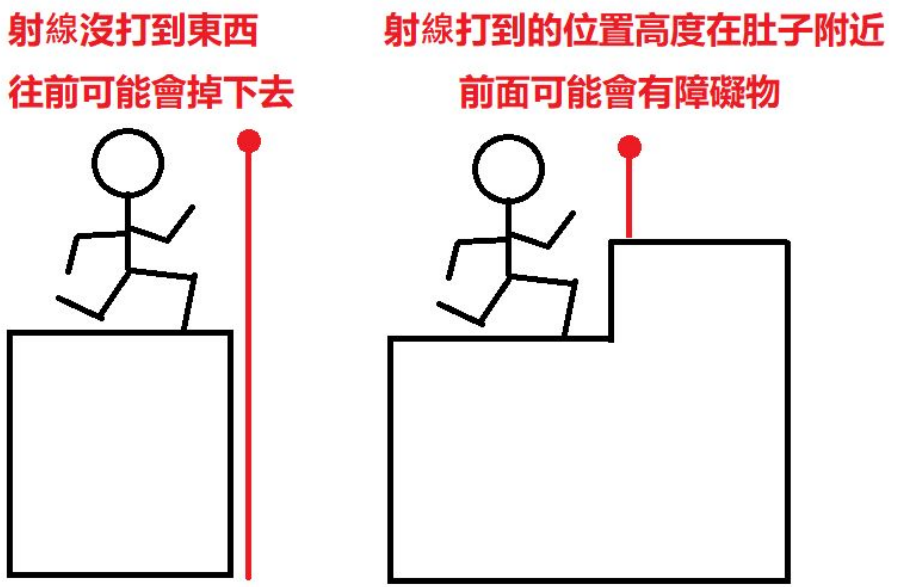
Unity LAB 6.5

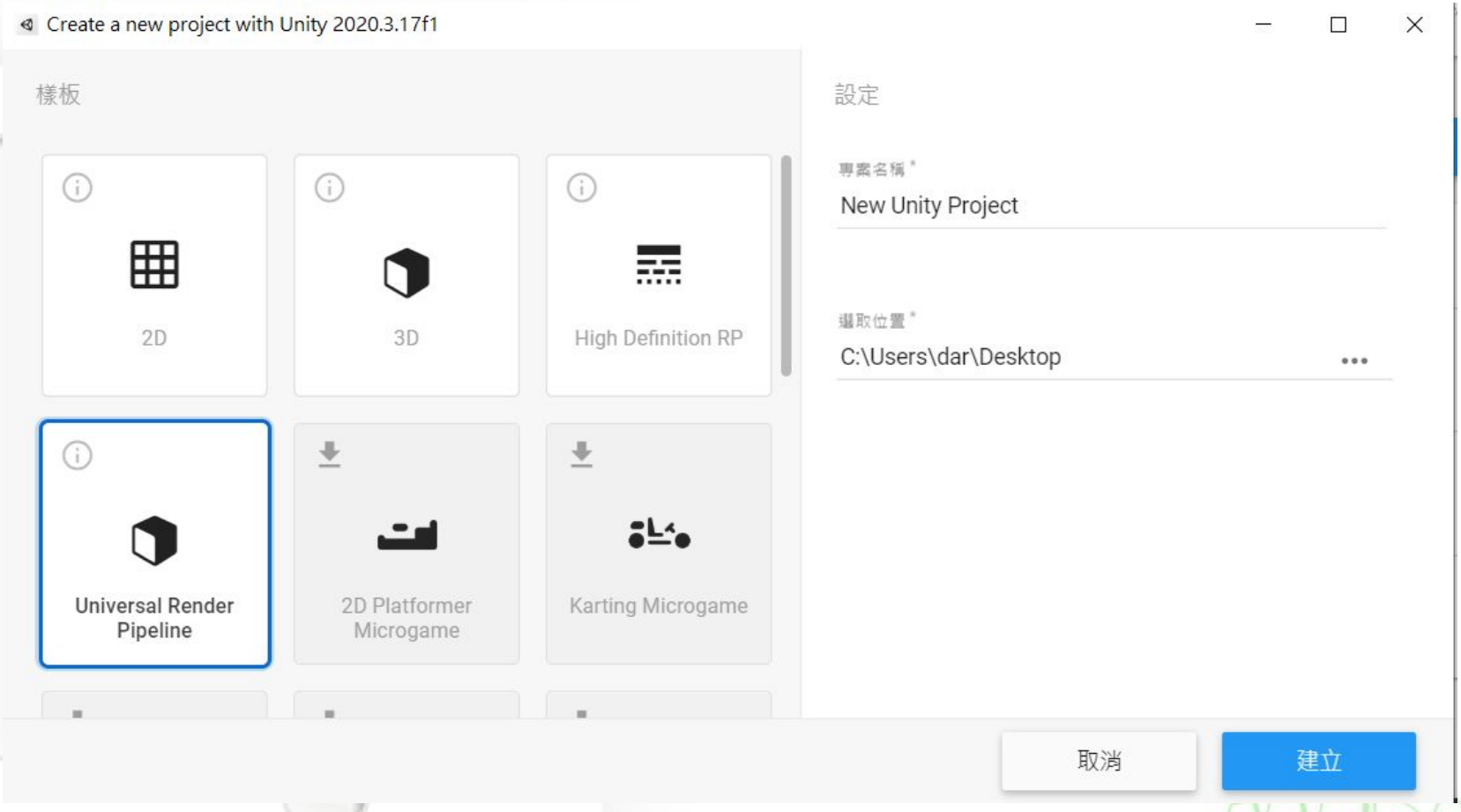
Raycast

Raycast

Raycast(射線檢測):射出一條Ray(射線)來檢測射到的物體，可以透過射線偵測到的物體資訊來做即時的判斷。例如:射擊游戲的瞄準、跑酷遊戲的周邊環境偵測



開啓URP project



匯入素材

- Asset Store裏搜尋Starter Asset First Person Character，並匯入Unity，匯入時如果有出現選項按yes。
- Asset Store裏找一個喜歡的槍的模型匯入Unity。
- 匯入本次會用到的素材：Raycast unitypackage
- 或者下載已經有以上素材的文件夾：FPS.zip

Raycast in Unity

```
bool Physics.Raycast(Vector3 origin, Vector3 direction,
out RaycastHit hitInfo, float maxDistance, int layerMask);
```

origin:射線的發射點

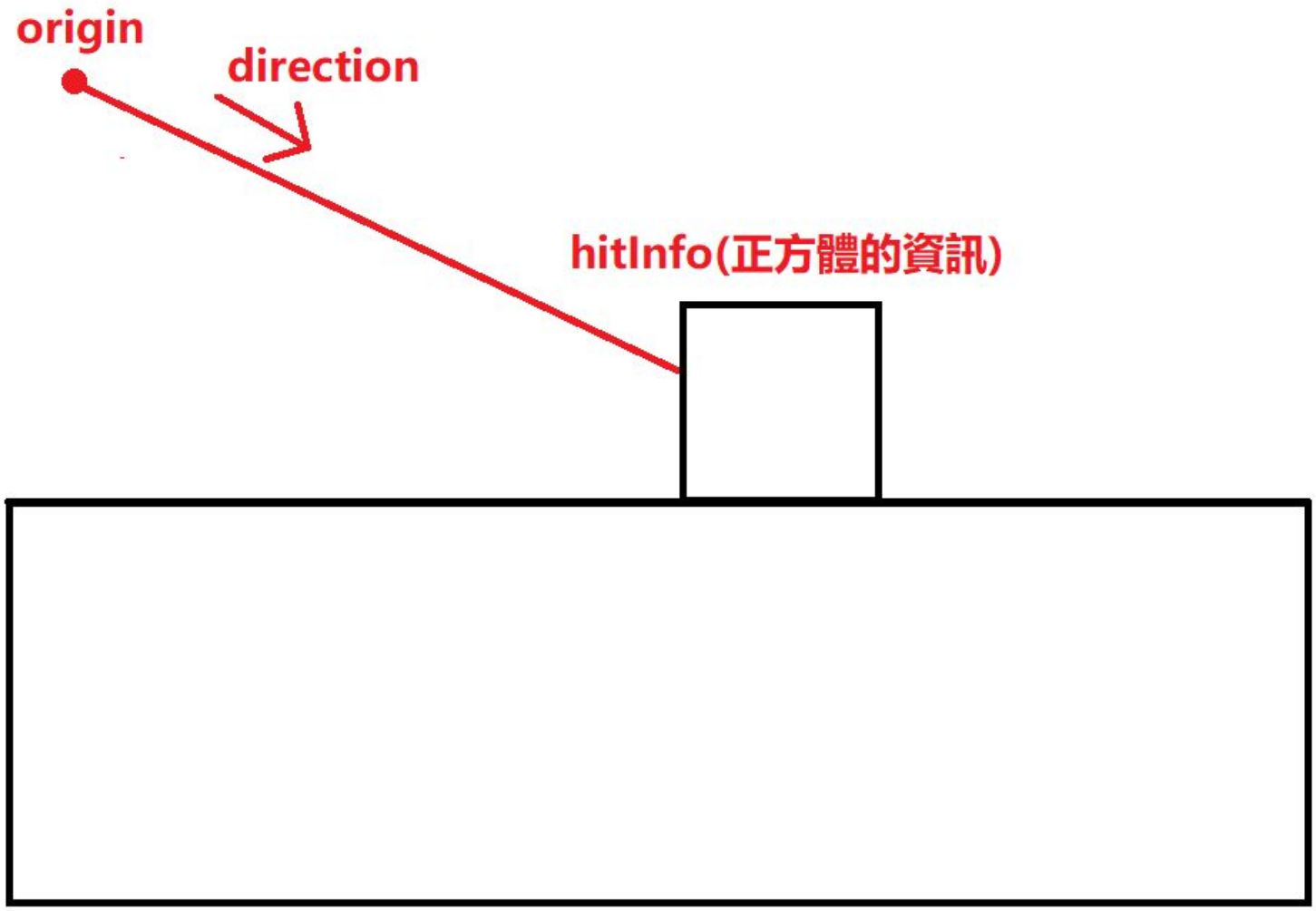
direction：射線的發射方向

hitInfo：射線偵測到的物體資訊

maxDistance：射線偵測的最大距離

layerMask：射線會偵測到的Layer

Raycast



RaycastHit

HitInfo記錄了射線偵測到的相關資訊

HitInfo.collider：被偵測物體的collider

Hitinfo.rigibody：被偵測物體的rigidbody

HitInfo.transform.position：被偵測物體的位置

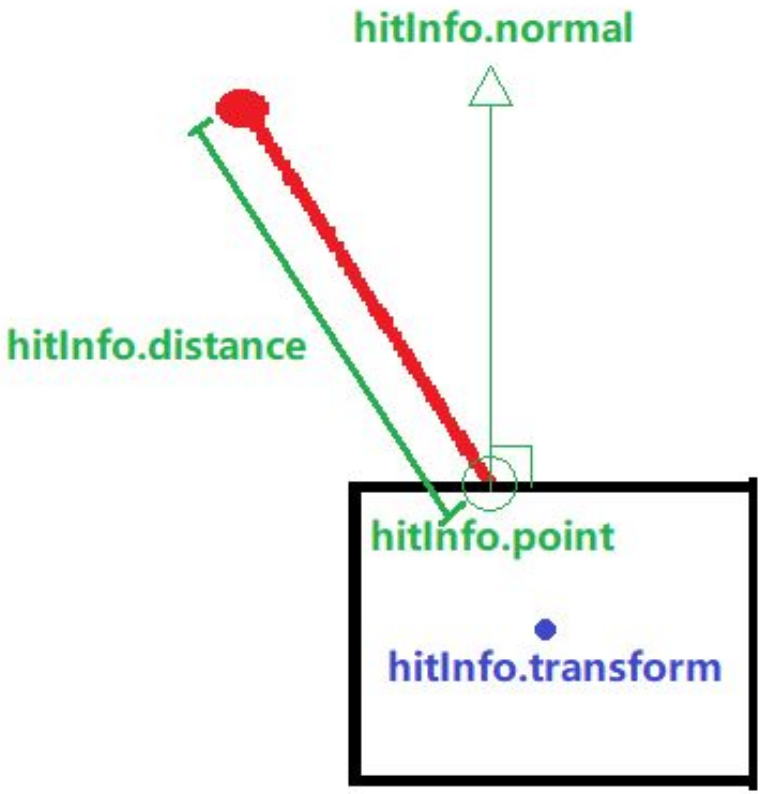
HitInfo.collider.gameObject.name：被偵測物體的名字

HitInfo.point：射線與被偵測物體的接觸點

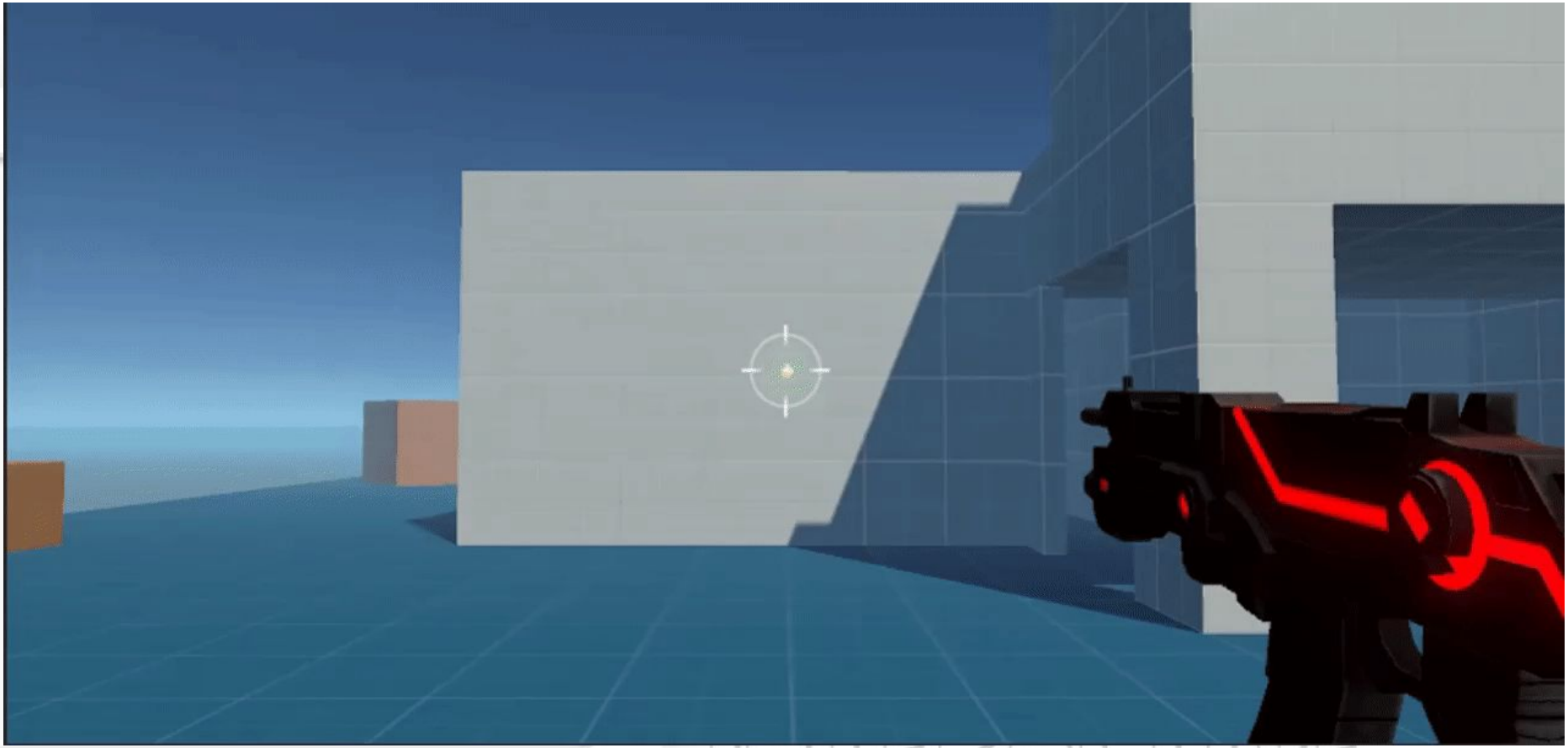
HitInfo.distance：發射點與接觸點的距離

HitInfo.normal：接觸點所在平面的法向量

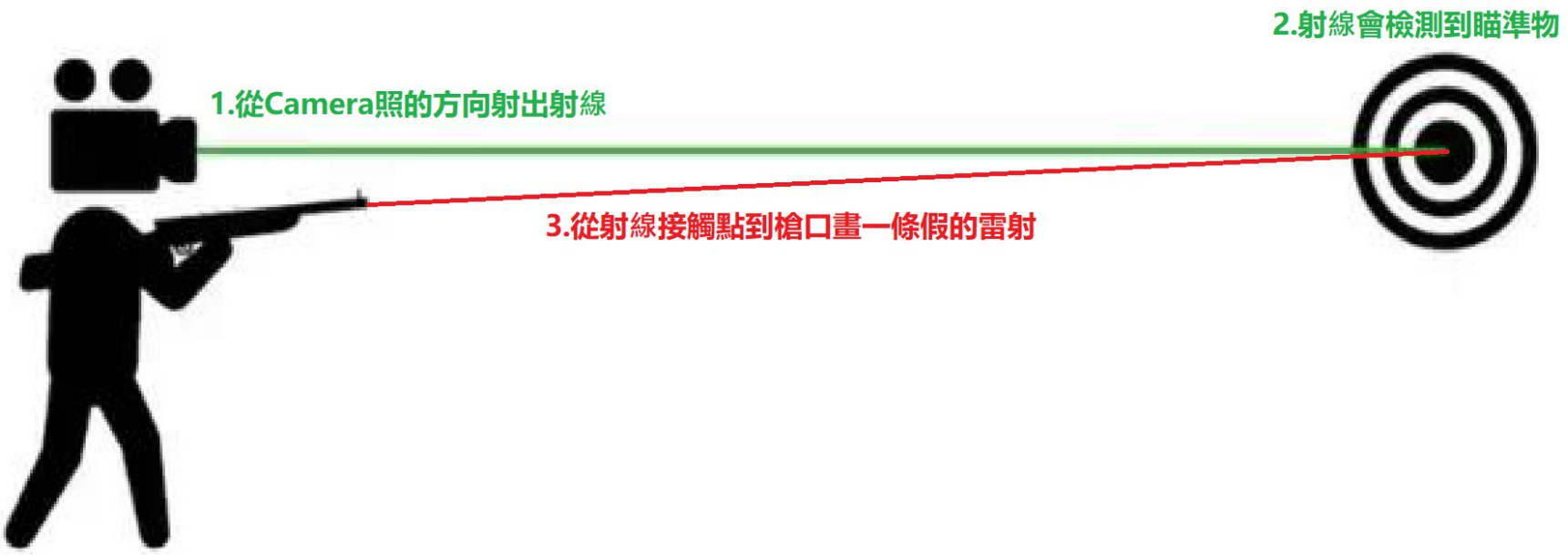
RaycastHit



範例: fps製作

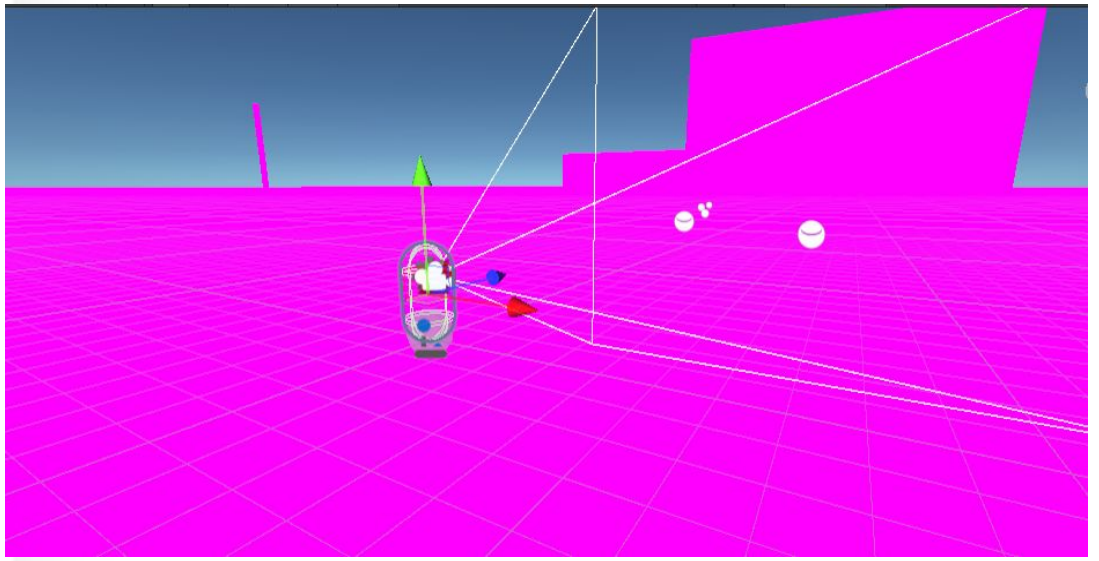


fps的射線檢測原理



場景設定

打開starter asset的PlayGround, 此時應該畫面會是一片粉紅。

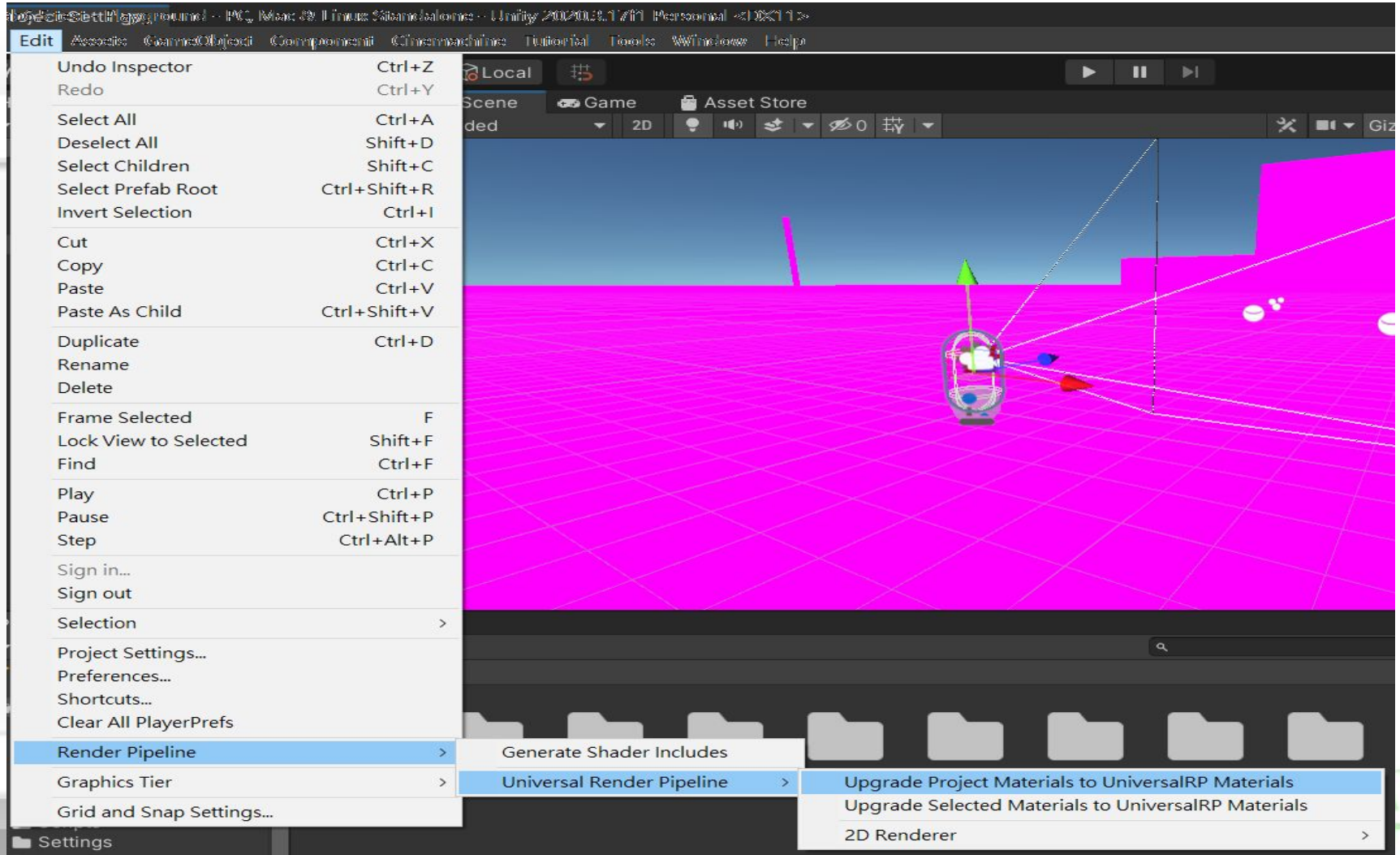


原因是場景物體的材質使用的是一般的材質, 不是URP的材質

Universal Render Pipeline (URP)

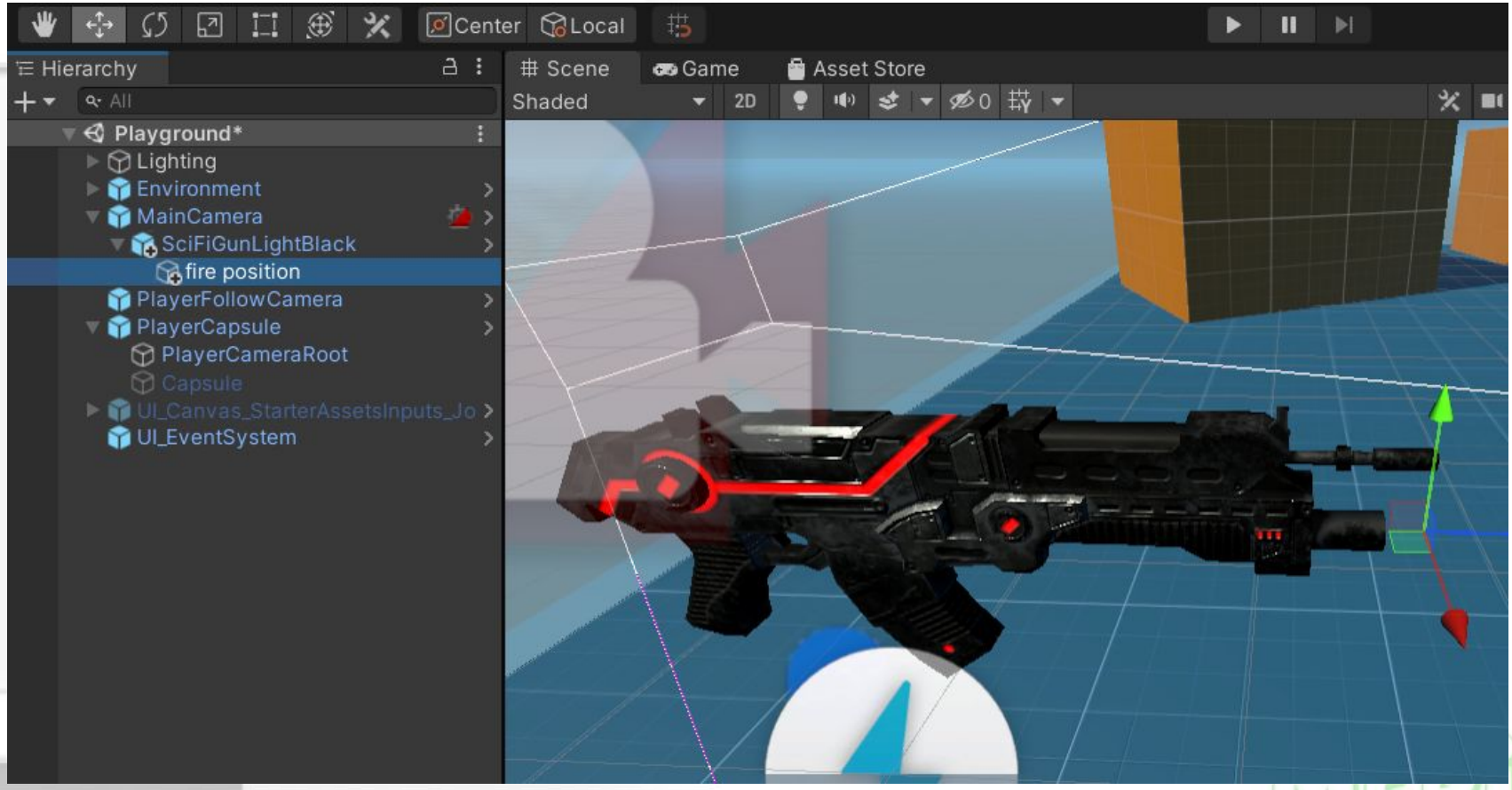
場景設定

點Edit>Render Pipeline>Upgrade Project Materials To URP Material,
做完後應該就能正常顯示



場景設定

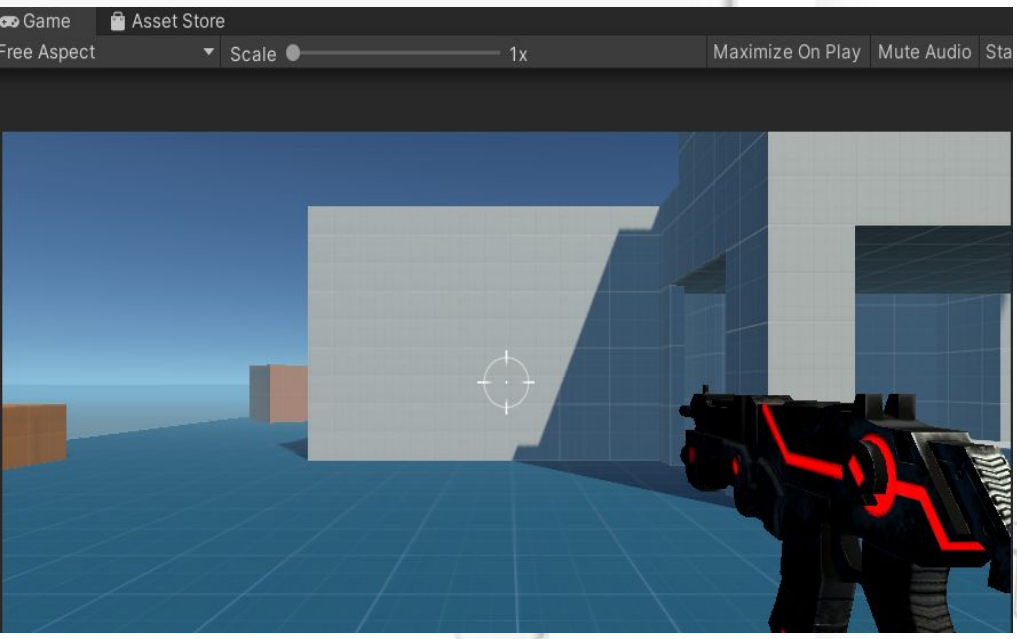
把Capsule隱藏或刪掉。把槍的模型放到合適的位置，並拉到Main Camera下面(子物件)。點槍的模型按Create Empty，並移動到槍口位置，並命名為Fire Position

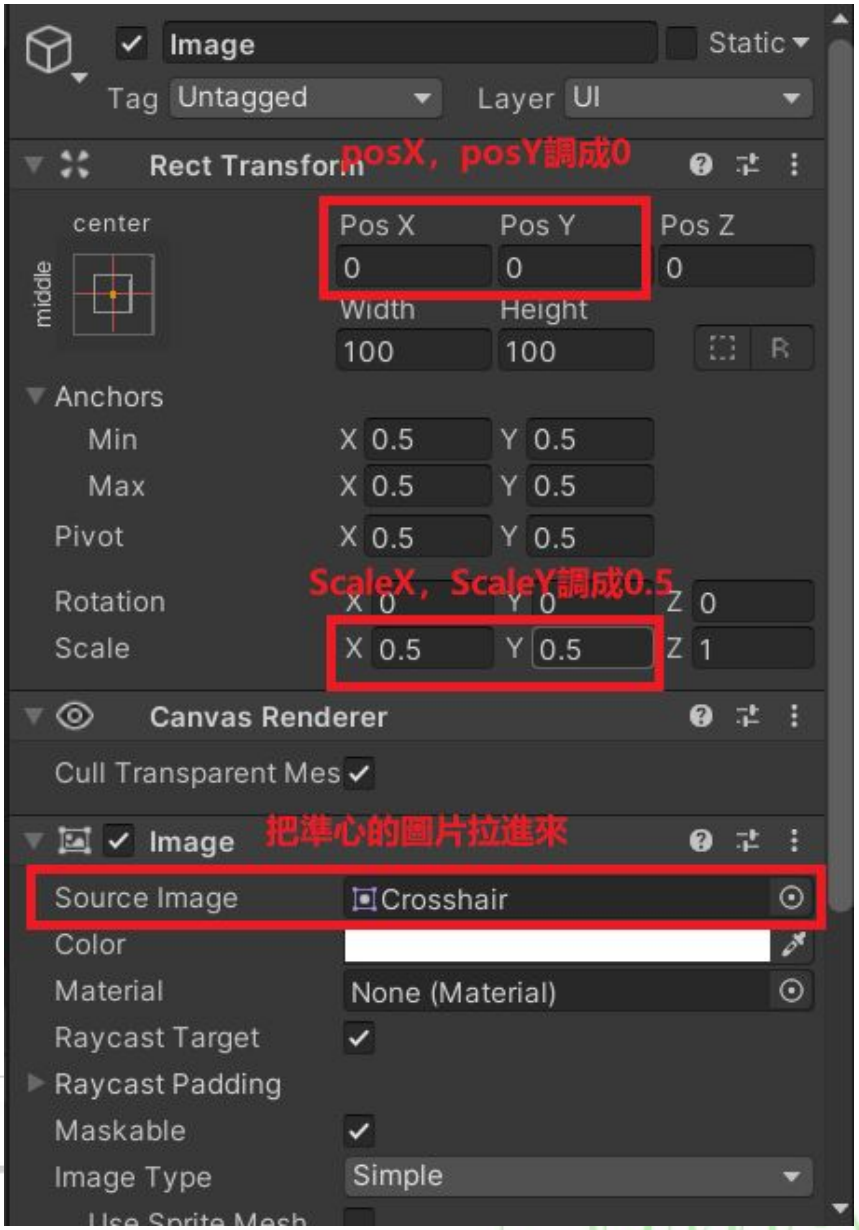


場景設定

新增一個Image的UI，並調整參數。

做完畫面如下





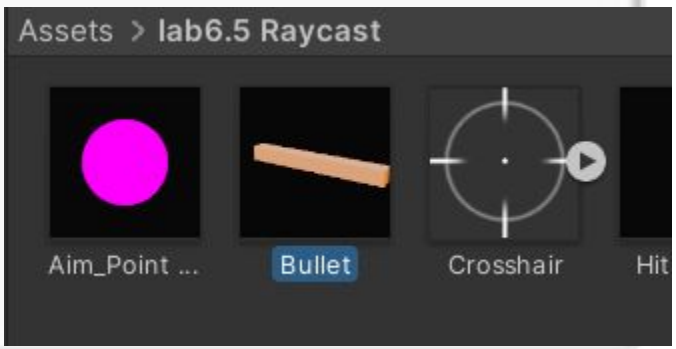
FPS Shooter script

在Main Camera新增一個script

```
public GameObject Aim_pointer;
public GameObject bullet;
public Camera Camera;
public Transform fire_pos;
RaycastHit hitInfo;
Vector3 Ray_pos;
Unity Message|0 個參考
void Start()
{
    Aim_pointer = Instantiate(Aim_pointer,Vector3.zero, Quaternion.identity);
}

Unity Message|0 個參考
void Update()
{
    //射線檢測
    Ray_pos = Camera.ViewportToWorldPoint(new Vector3(0.5f, 0.5f, 1));
    if (Physics.Raycast(Ray_pos, Camera.transform.forward,out hitInfo,99))
    {
        Aim_pointer.transform.position = hitInfo.point;
        //print(hitInfo);
    }
    else
    {
        Aim_pointer.transform.position = Ray_pos + 99*Camera.transform.forward;
    }

    //子彈發射
    if (Input.GetMouseButtonDown(0))
    {
        GameObject Fire_bullet= Instantiate(bullet, fire_pos.position,Quaternion.identity);
        Fire_bullet.transform.LookAt(hitInfo.point);
    }

}
```

Bullet Script

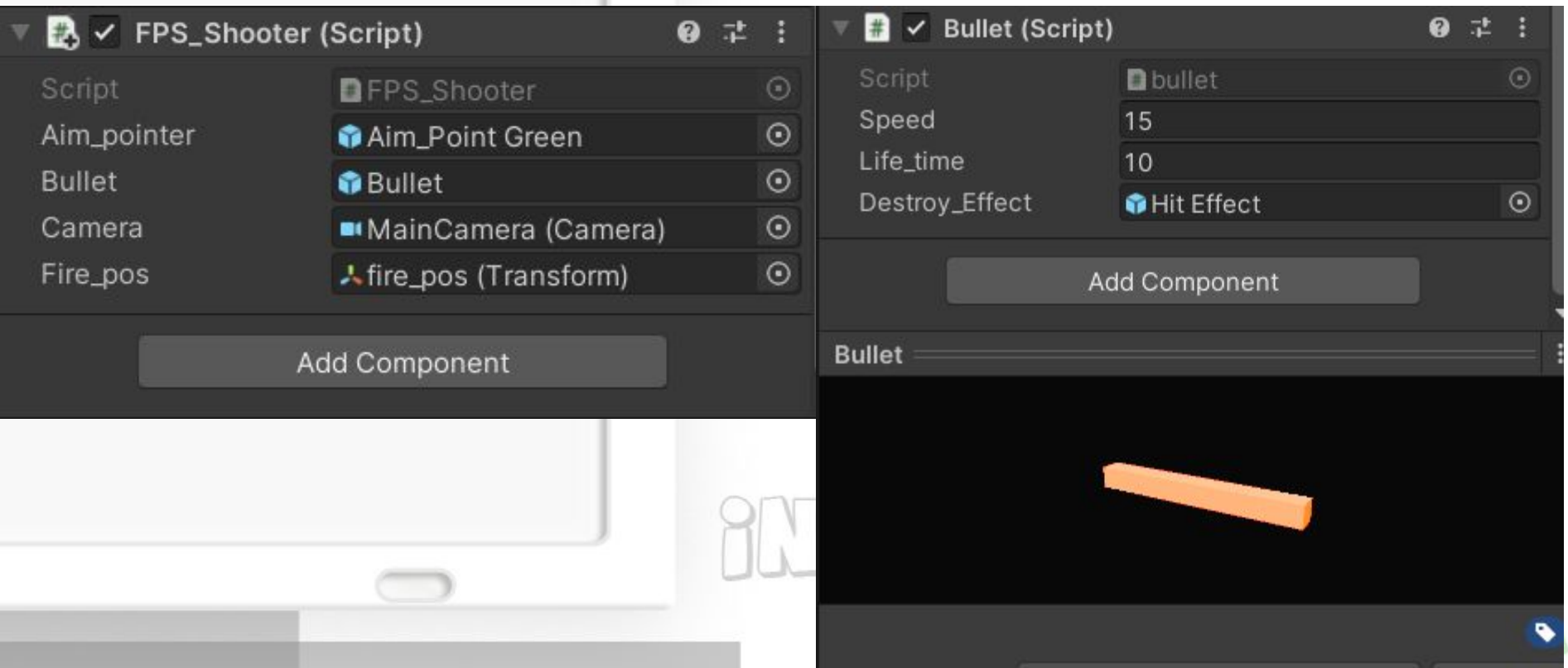
在Bullet增加一個子彈的script



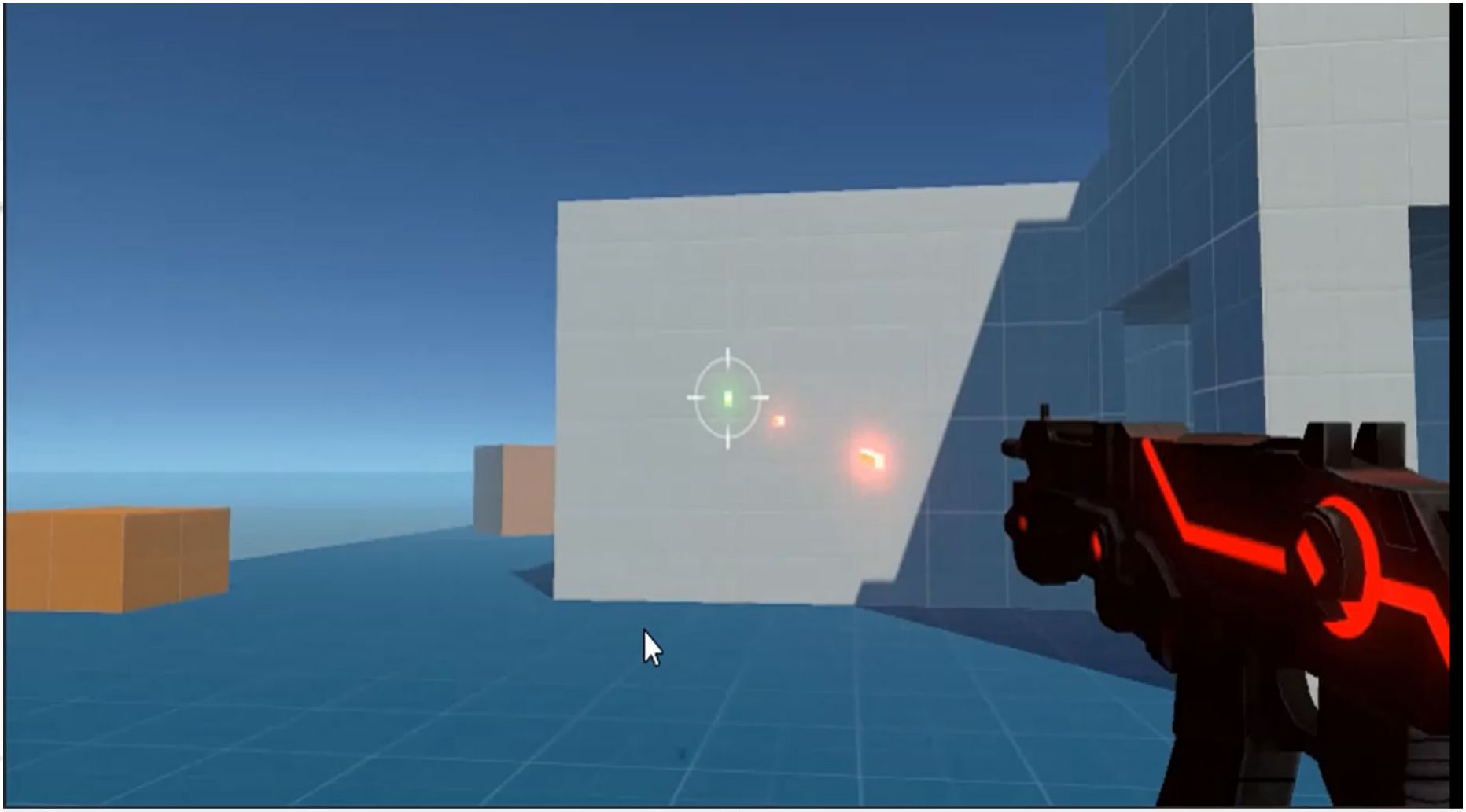

```
Unity 指令碼|0 個參考
public class bullet : MonoBehaviour
{
    public float speed=0;
    public float life_time = 10;
    public GameObject Destroy_Effect;

    float timenow=0;
    Rigidbody Rigidbody;
    Unity Message|0 個參考
    void Start()
    {
        Rigidbody = this.GetComponent<Rigidbody>();
    }

    Unity Message|0 個參考
    void Update()
    {
        Destroy_CountDown();
        Rigidbody.velocity = transform.forward * speed;
    }
    1 個參考
    private void Destroy_CountDown()
    {
        timenow += Time.deltaTime;
        if (timenow > life_time)
        {
            Destroy(this.gameObject);
        }
    }
    Unity Message|0 個參考
    private void OnCollisionEnter(Collision collision)
    {
        Instantiate(Destroy_Effect, transform.position, Quaternion.identity);
        Destroy(this.gameObject);
    }
}
```

場景設定

把對應的選項拉好(東西在Raycast文件夾)，speed為子彈速度，life_time是子彈自動摧毀時間



測試



補充(TPS)

第三人稱射擊遊戲教學影片