



# VR棒球遊戲結合腦波 觀測之應用

指導教授:謝明哲老師

10612112 張元豪

10612120 李維

10612124 洪久洲

10612126 溫舜傑

# 大綱

第一部分 背景與動機

第二部分 目的

第三部分 系統設計與架構

第四部分 成果展示

第五部分 評估

第六部分 心得



## 第一部分

背景與動機

# 1-1 背景



棒球的特點：  
接觸者眾多

趣味性高

易凝聚人的向心力



# 1-1-1 腦波介紹

腦波種類		頻率	特性
Alpha ( $\alpha$ )	慢速 $\alpha$ 波	8-9 HZ	臨睡前意識逐漸模糊
	中間 $\alpha$ 波	9-12 HZ	靈感、直覺或點子爆發狀態 身心輕鬆而注意力集中
	快速 $\alpha$ 波	12-14 HZ	高度警覺狀態
Beta ( $\beta$ )	Low Range	12.5-16 HZ	放鬆但精神集中
	Middle Range	16.5-20 HZ	思考、處理接收到外界資訊
	High Range	20.5-28 HZ	激動、焦慮
Gamma ( $\gamma$ )		25-100 HZ	提高意識、幸福感、減輕壓力



## 1-1-2 文獻探討

過去有許多研究指出運動後之alpha腦波活動明顯比運動前增加，並推論運動後腦波反映出放鬆或焦慮降低之現象。

流暢經驗可能出現在各種場合，包括工作上、學校裡、生活中和運動時。當運動員處於流暢狀態時，會徹底投入所從事的活動當中，完全將其他事物摒除在外，所以流暢經驗和專注有很大的關聯性。

運動表現與專注力在大腦活動的研究上，大多以大腦左右半球 $\alpha$ 波功率的差異來探討注意力的轉換。 $\alpha$ 波的產生是大腦皮質的錐狀細胞同步化的結果，其功率會因感官刺激、自主動作及認知作業而降低

腦波的實驗相關方法如下：通常是給一外在的刺激或撤銷刺激後，在大腦所引起的電位變化，再經由儀器紀錄中間的變化情形

# 1-2 確認問題與動機

人數不足

天氣因素

使用者無法客製化訓練

希望觀測受測者腦波數據以訓練  
選手





## 第二部分

目的

## 2-1 目的



利用虛擬實境結合腦波測量，以達到訓練選手的  
目的



## 2-2 目標



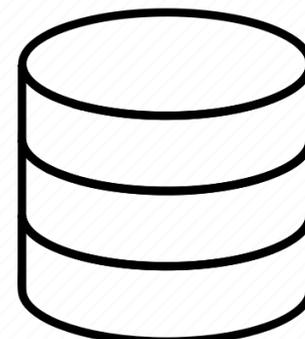
解決人數不足  
的問題



克服天候因素



客製化訓練



結合腦波儀以便  
觀測腦波狀態

## 2-3 解決方案

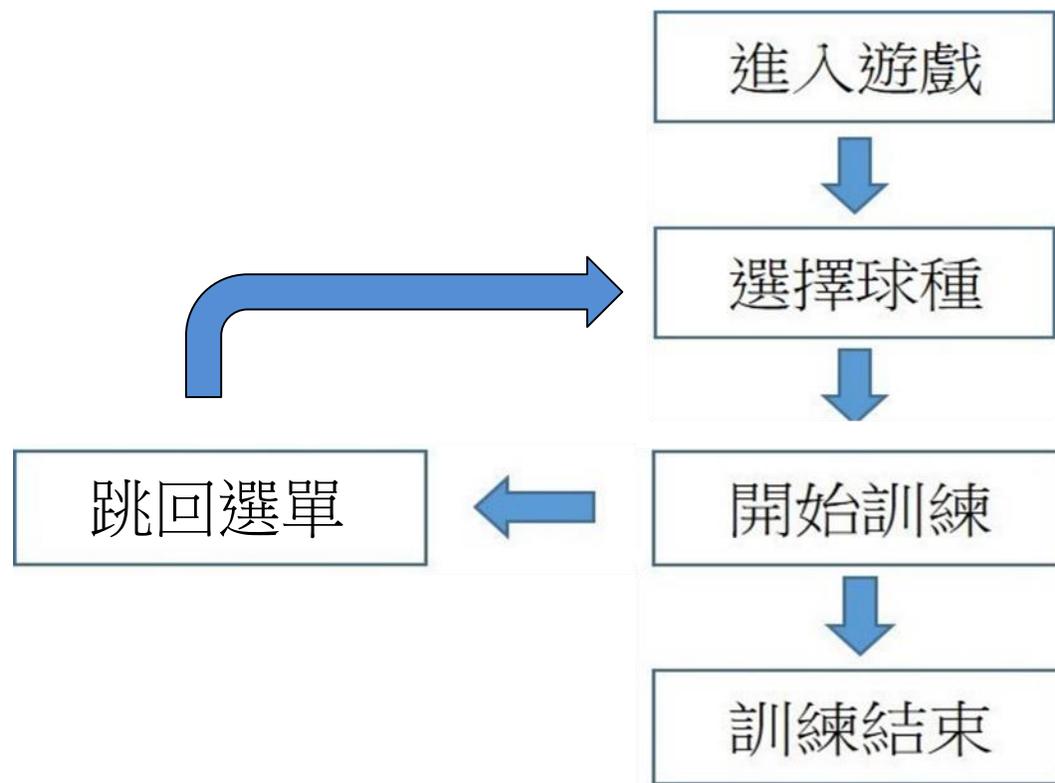
解決方案:透過unity以及VR技術製作虛擬棒球場，棒球，球種，球棒與腦波技術結合觀測受測者數據用以訓練。



## 第三部分

系統設計與架構

# 3-1 系統流程圖



## 3-2 系統架構製作

1. 場景製作

2. 球棒製作

3. 棒球製作

4. 人物動畫

5. 使用者介面

6. 腦波儀結合



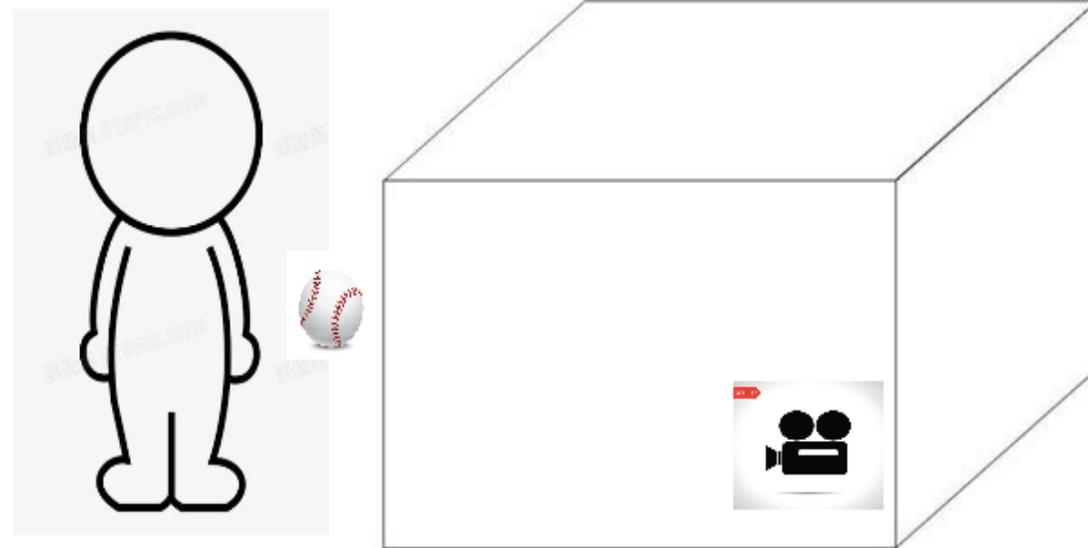
## 3-2-1 場景製作

方法一：在Google街景上先拍出全景照片，接著利用PTGUI將照片分割為六個部分，最後在Unity中製作一個六面體並將照片一一放入，即可形成一個虛擬場景



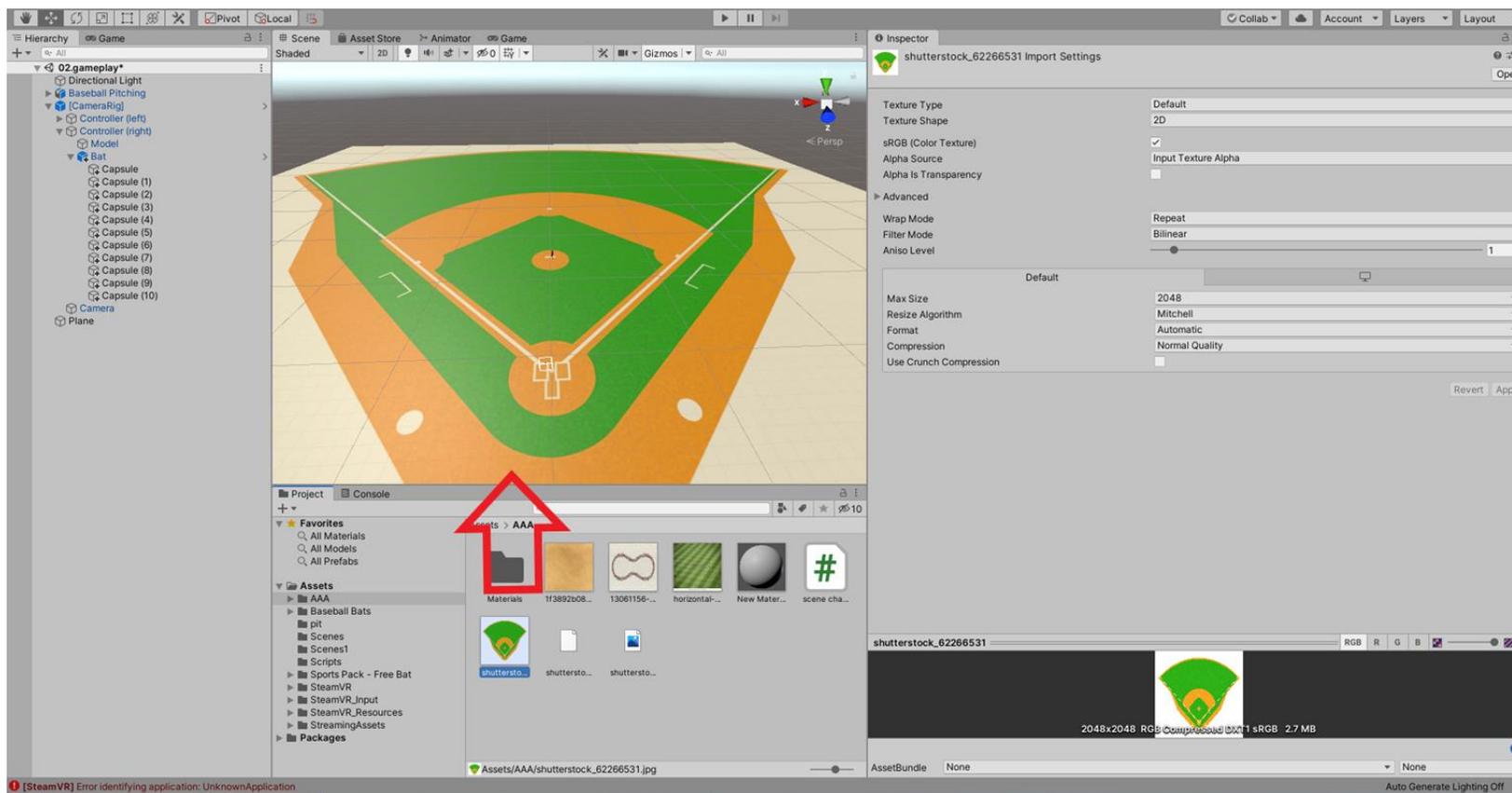
## 3-2-1 場景製作

方法一瓶頸：不過這個方法的問題是會使投手在照片後面，球要通過場景照片才能看的到球



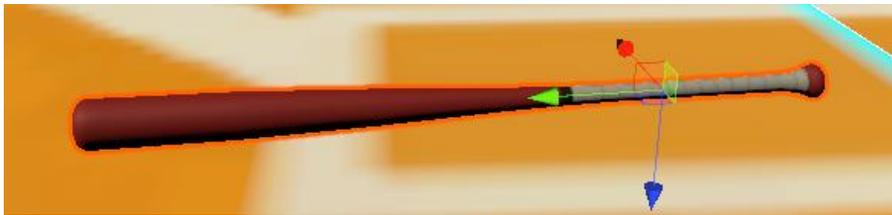
## 3-2-1 場景製作

方法二：建立一個方形平台，再將棒球場貼圖掛上

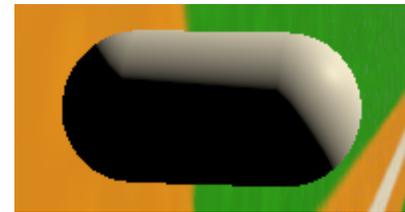


## 3-2-2 球棒製作

將棒球的貼圖貼上多個膠囊，使球棒變成實體，再加上Rigidbody等屬性，使球棒能碰撞



+



## 3-2-2 球棒製作

但因為重力的關係，膠囊會脫離球棒，所以我們加了Batcapsulefollower

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class BatCapsuleFollower : MonoBehaviour
{
    private BatCapsule _batFollower;
    private Rigidbody _rigidbody;
    private Vector3 _velocity;

    [SerializeField]
    private float _sensitivity = 100f;

    private void Awake()
    {
        _rigidbody = GetComponent<Rigidbody>();
    }

    private void FixedUpdate()
    {
        Vector3 destination = _batFollower.transform.position;
        _rigidbody.transform.rotation = transform.rotation;

        _velocity = (destination - _rigidbody.transform.position) * _sensitivity;

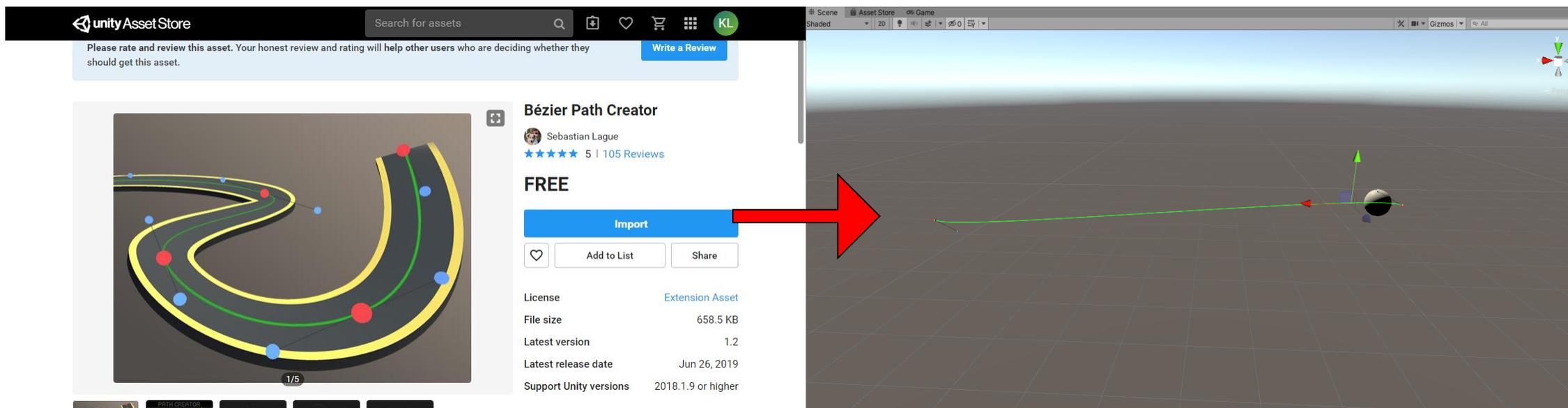
        _rigidbody.velocity = _velocity;
        transform.rotation = _batFollower.transform.rotation;
    }

    public void SetFollowTarget(BatCapsule batFollower)
    {
        _batFollower = batFollower;
    }
}
```



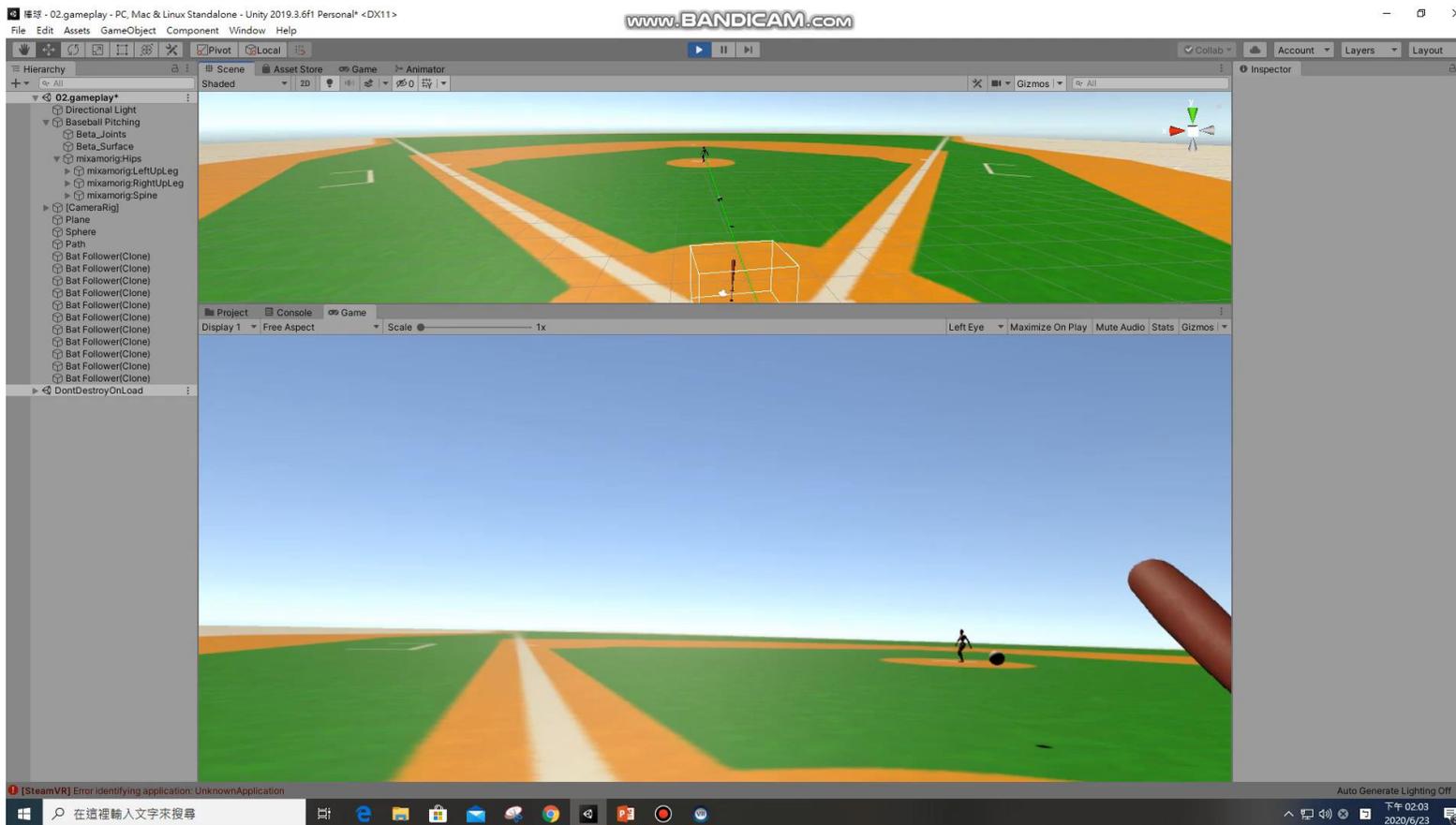
## 3-2-3 棒球製作

方法一：使用AssetStore中的Path程式畫出球路



# 3-2-3 棒球製作

方法一：使用AssetStore中的Path程式畫出球路



無法正常打擊



## 3-2-3 棒球製作

### 方法二： 撰寫棒球程式腳本

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Baseball : MonoBehaviour
{
    public AudioSource src;

    private Rigidbody rb;
    private float velocityMax = 20f;

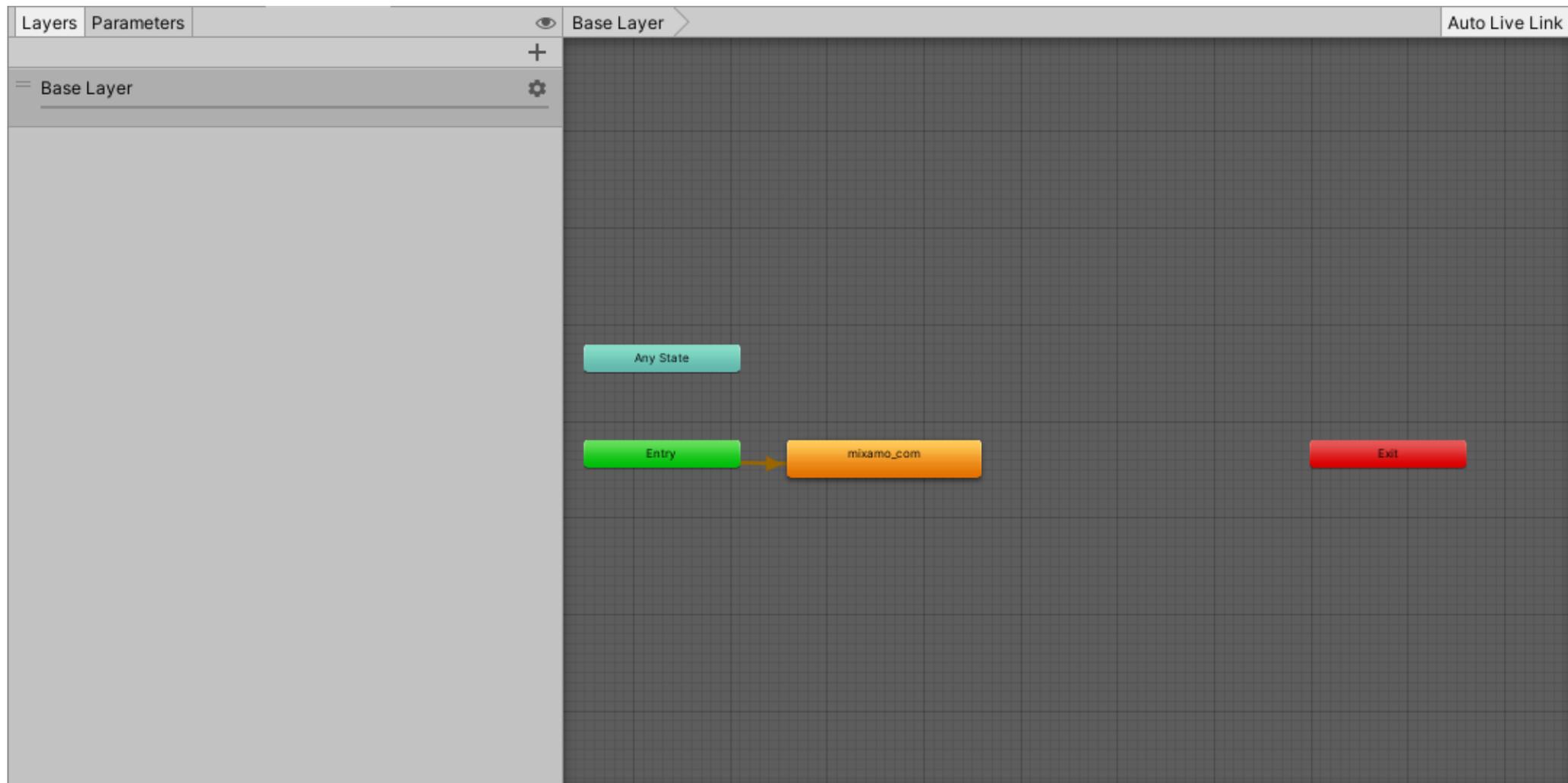
    void Awake()
    {
        rb = gameObject.GetComponent<Rigidbody>();
        rb.AddForce(transform.forward * velocityMax, ForceMode.Impulse);
    }
    private float GetBatForce(Rigidbody batRB)
    {
        return batRB.velocity.magnitude / velocityMax * 50f;
    }
    private void OnCollisionEnter(Collision collision)
    {
        if (collision.gameObject.name == "Bat")
        {
            rb.velocity = Vector3.zero;
            src.Play();

            float forceMultiplier = GetBatForce(collision.gameObject.GetComponent<Rigidbody>());
            Vector3 direction = (transform.position - collision.contacts[0].point).normalized;
            rb.AddForce(direction * forceMultiplier, ForceMode.Impulse);
            rb.useGravity = true;
            Destroy(gameObject, 2f);
        }
    }
}
```

## 3-2-4 人物動畫



## 3-2-4 動畫控制



## 3-2-4 Lanchbox

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Lanchbox : MonoBehaviour
{
    public float timeBetweenPitches;
    public GameObject ball;
    public float launchAngle;

    void Start()
    {
        StartCoroutine(Pitch());
    }

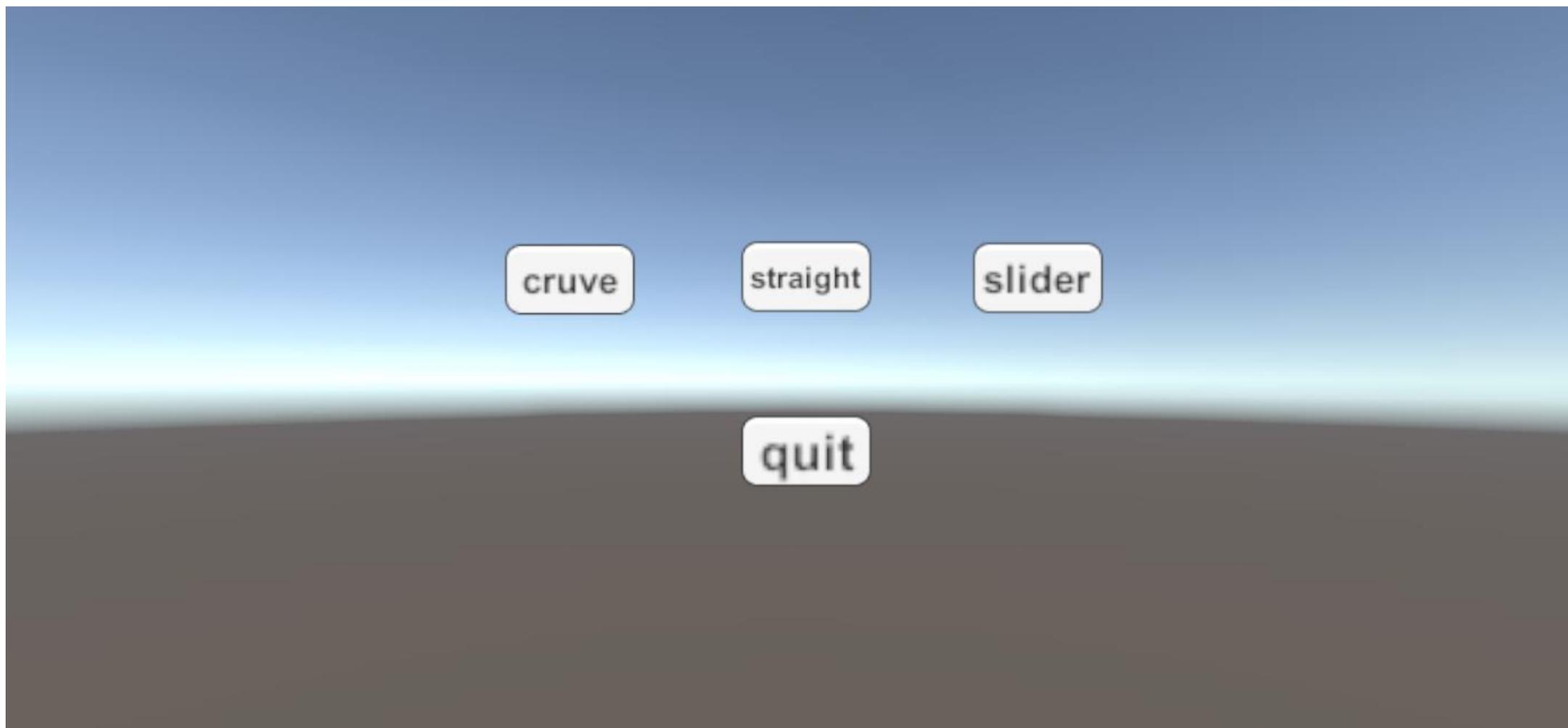
    private IEnumerator Pitch()
    {
        while (true)
        {
            yield return new WaitForSeconds(timeBetweenPitches);
            Vector3 launchDirection = GetLaunchDirection();
            Quaternion q = Quaternion.Euler(launchDirection);

            Instantiate(ball, transform.position, q);
        }
    }

    private Vector3 GetLaunchDirection()
    {
        return new Vector3(
            launchAngle, 0, Random.Range(0f, 180f));
    }
}
```



## 3-2-5 使用者介面



## 3-2-6 腦波儀結合



## 3-2-6 腦波儀結合

設定腦波專注，放鬆  
數值與打擊力度的關係

```
private float GetBatForce(Rigidbody _rigidbody)
{
    return _rigidbody.velocity.magnitude / velocityMax * 50f;
}
private float GetBatForce2(Rigidbody _rigidbody)
{
    return _rigidbody.velocity.magnitude / velocityMax * -20f;
}

void OnCollisionEnter(Collision collision)
{
    if (collision.gameObject.name == "Capsule" && script.attention1 >= 80 && script.meditation1 <= 60)
    {
        rb.velocity = Vector3.zero;
        audioSource.clip = src;
        audioSource.Play();
        Physics.IgnoreCollision(collision.gameObject.GetComponent<CapsuleCollider>(), gameObject.GetComponent<SphereCollider>());

        float forceMultiplier = GetBatForce(collision.gameObject.GetComponent<Rigidbody>());
        Vector3 direction = (transform.position - collision.contacts[0].point).normalized;
        rb.AddForce(direction * forceMultiplier, ForceMode.Impulse);
        rb.useGravity = true;
        Destroy(gameObject, 2f);
    }

    if (collision.gameObject.name == "Capsule" && script.attention1 <= 30 && script.meditation1 >= 60)
    {
        rb.velocity = Vector3.zero;
        audioSource.clip = src;
        audioSource.Play();
        Physics.IgnoreCollision(collision.gameObject.GetComponent<CapsuleCollider>(), gameObject.GetComponent<SphereCollider>());

        float forceMultiplier = GetBatForce2(collision.gameObject.GetComponent<Rigidbody>());
        Vector3 direction = (transform.position - collision.contacts[0].point).normalized;
        rb.AddForce(direction * forceMultiplier, ForceMode.Impulse);
        rb.useGravity = true;
        Destroy(gameObject, 2f);
    }
}
```

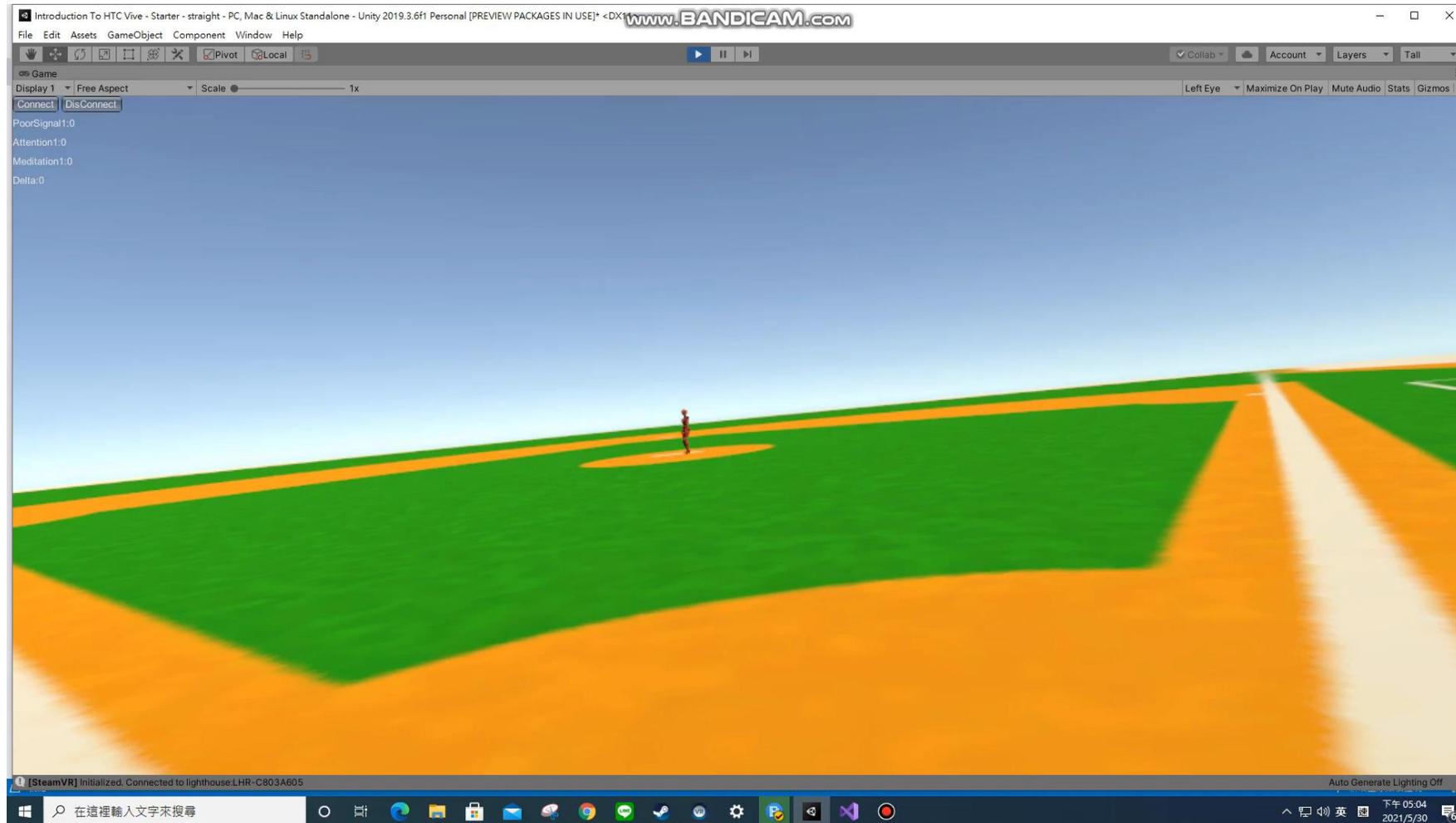




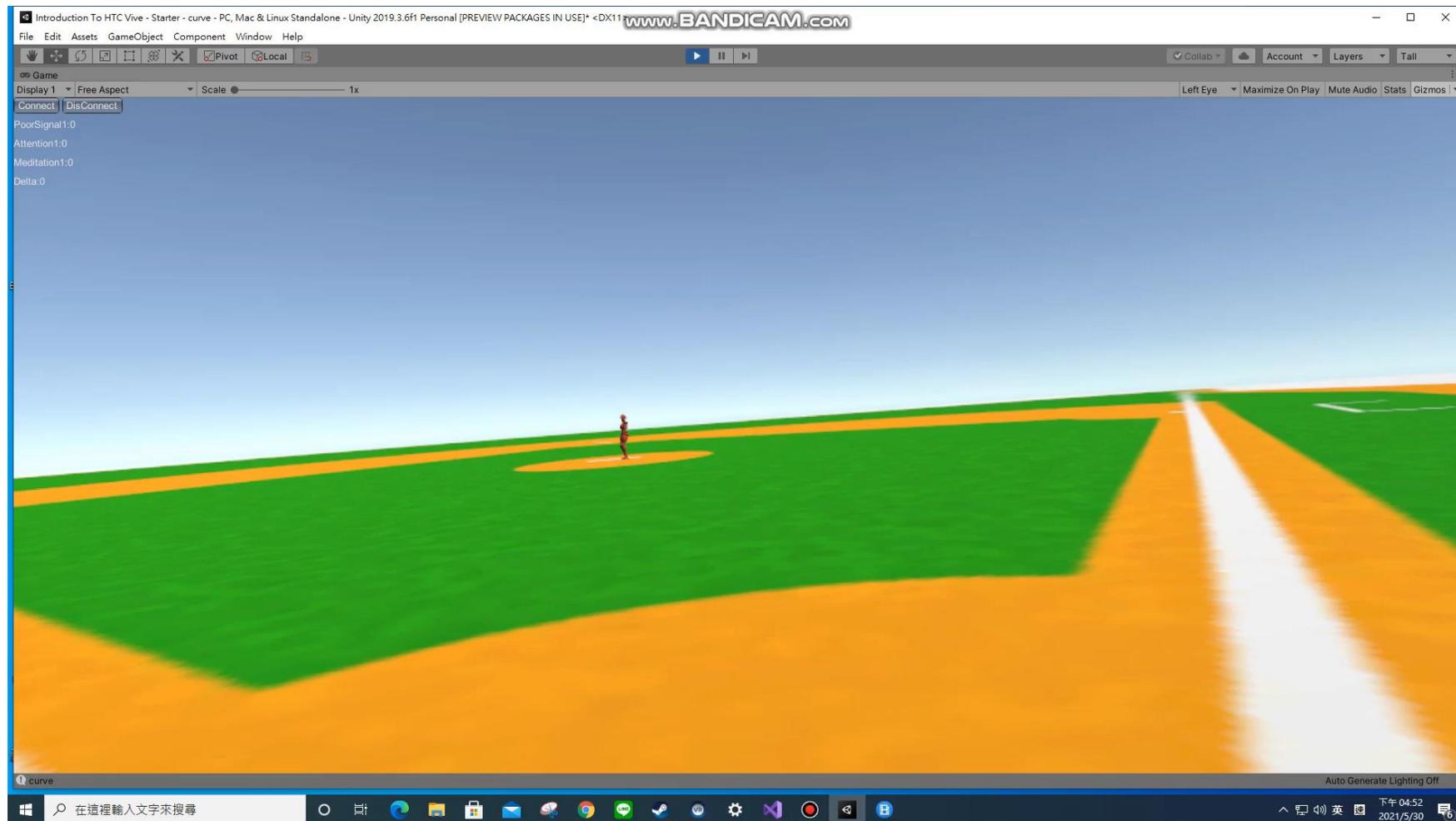
## 第四部分

成果展示

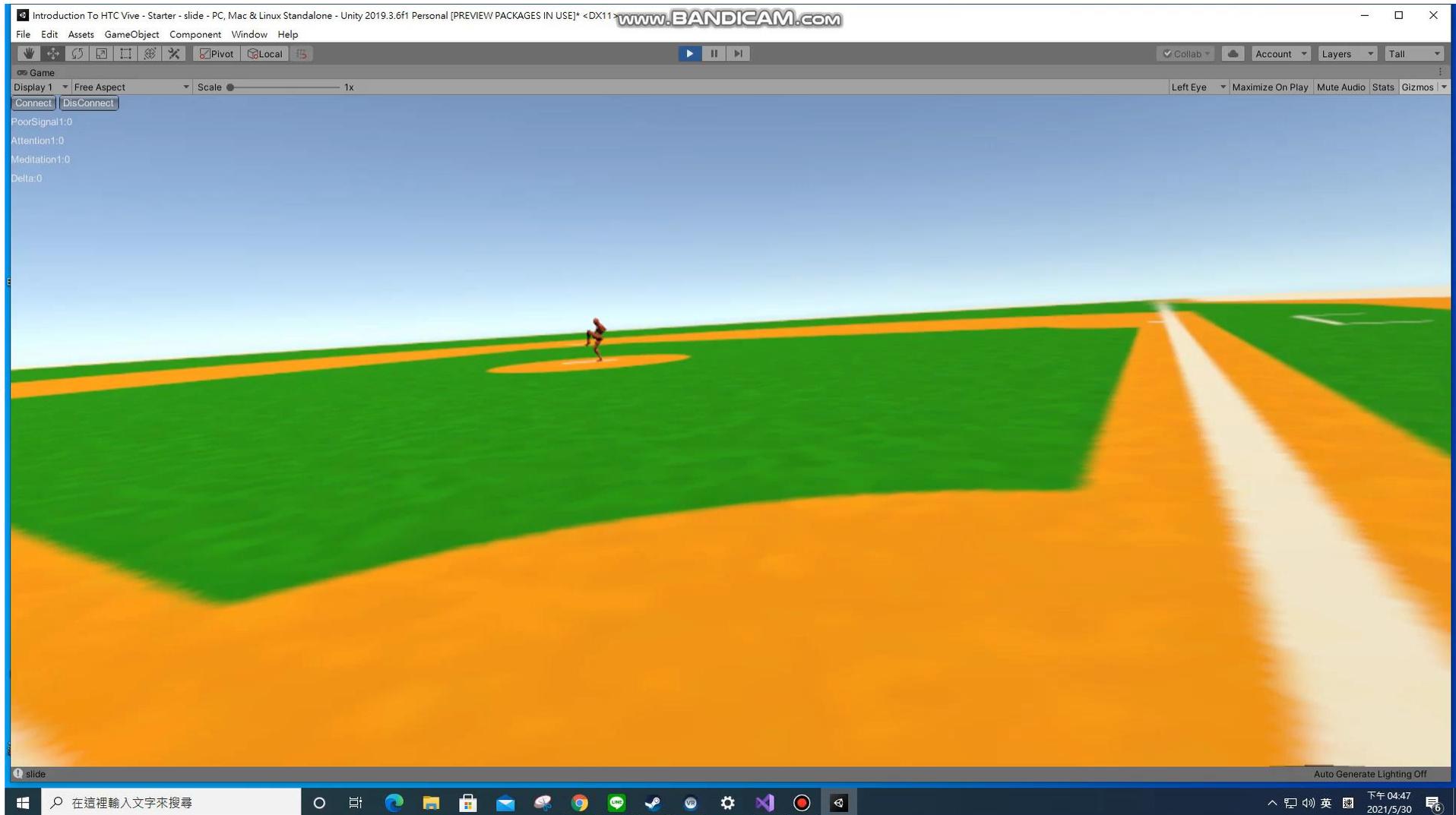
# 4-1 直球



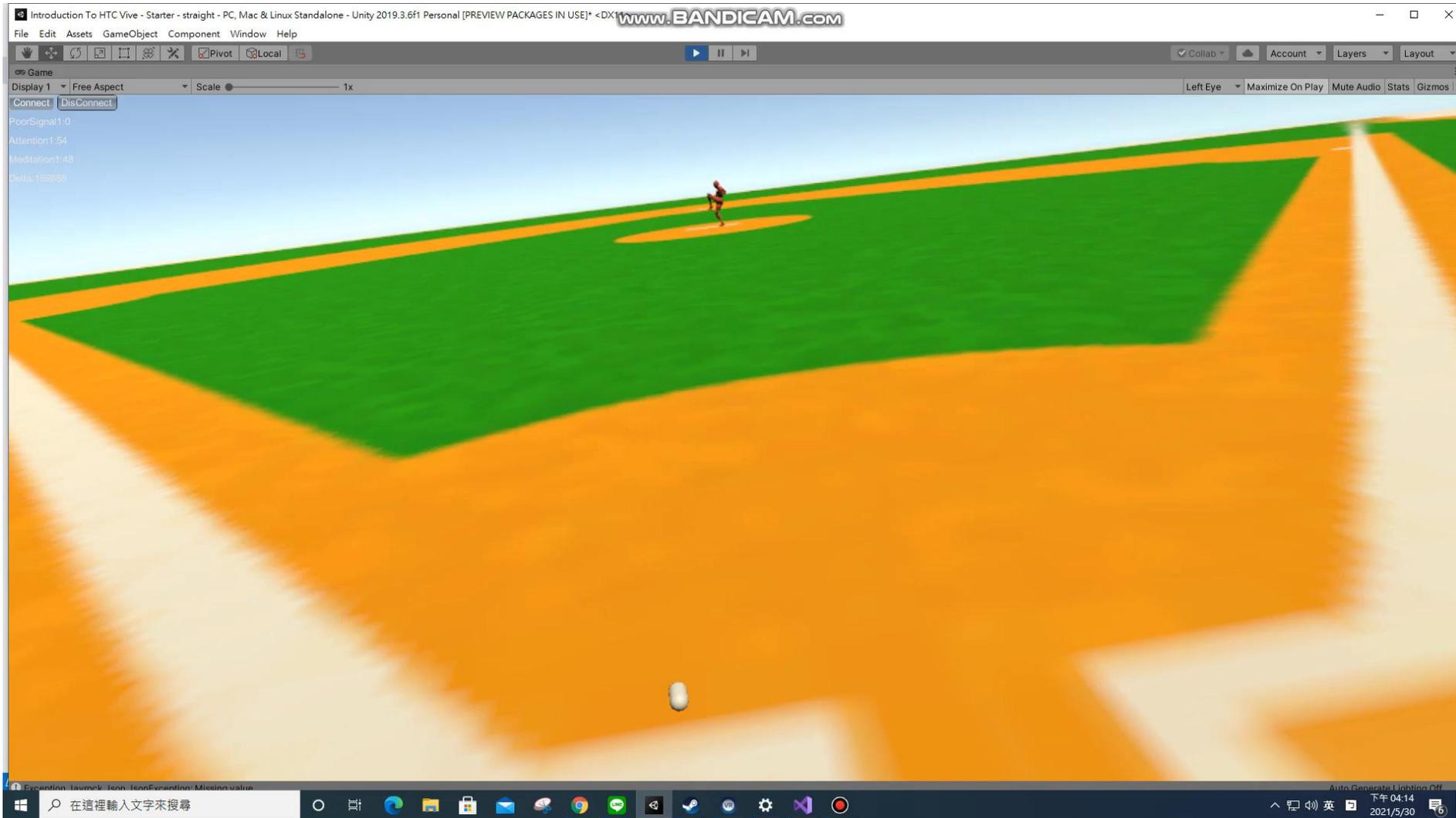
# 4-2 曲球



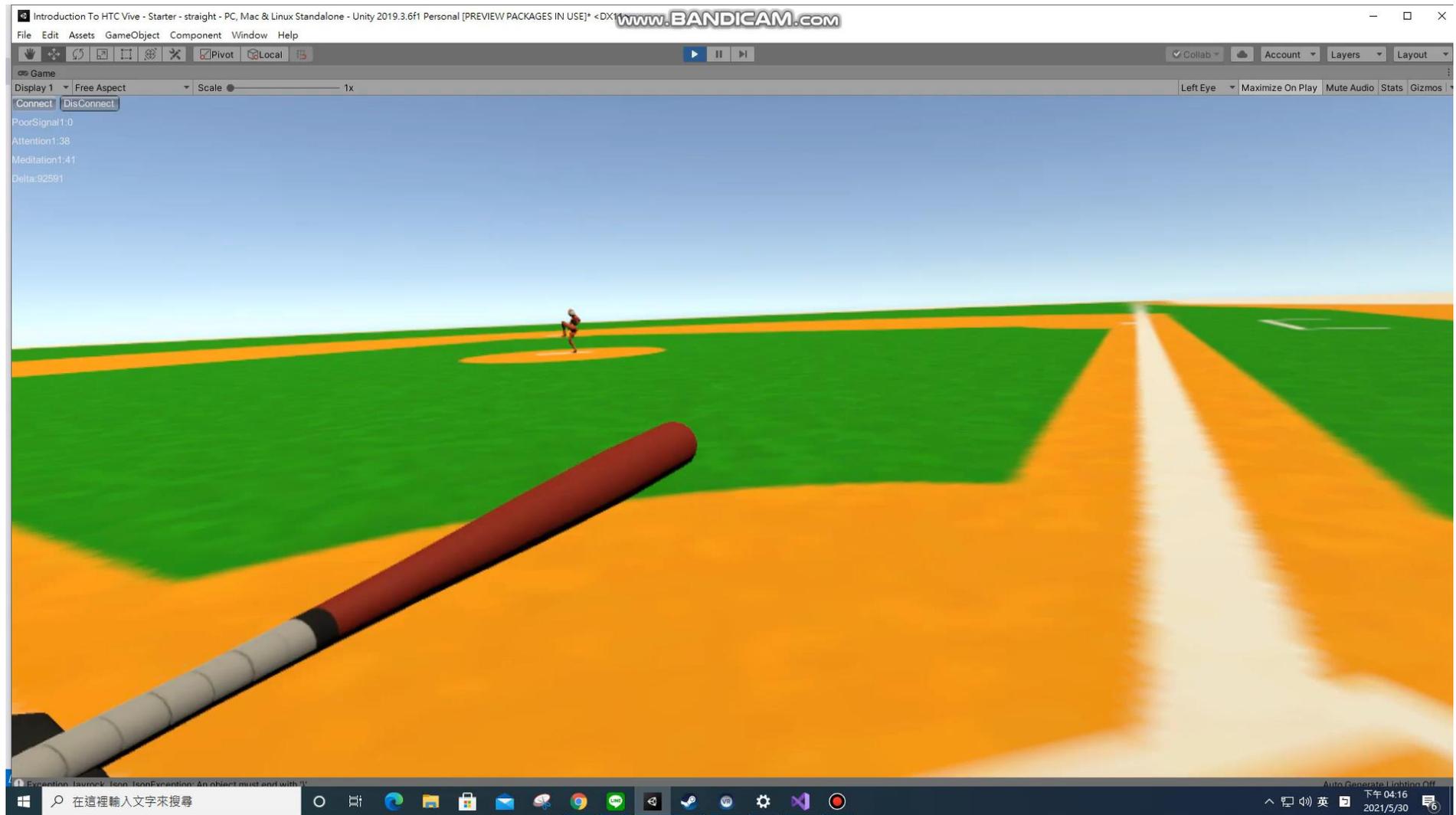
# 4-3 滑球



# 4-4 打擊力道小



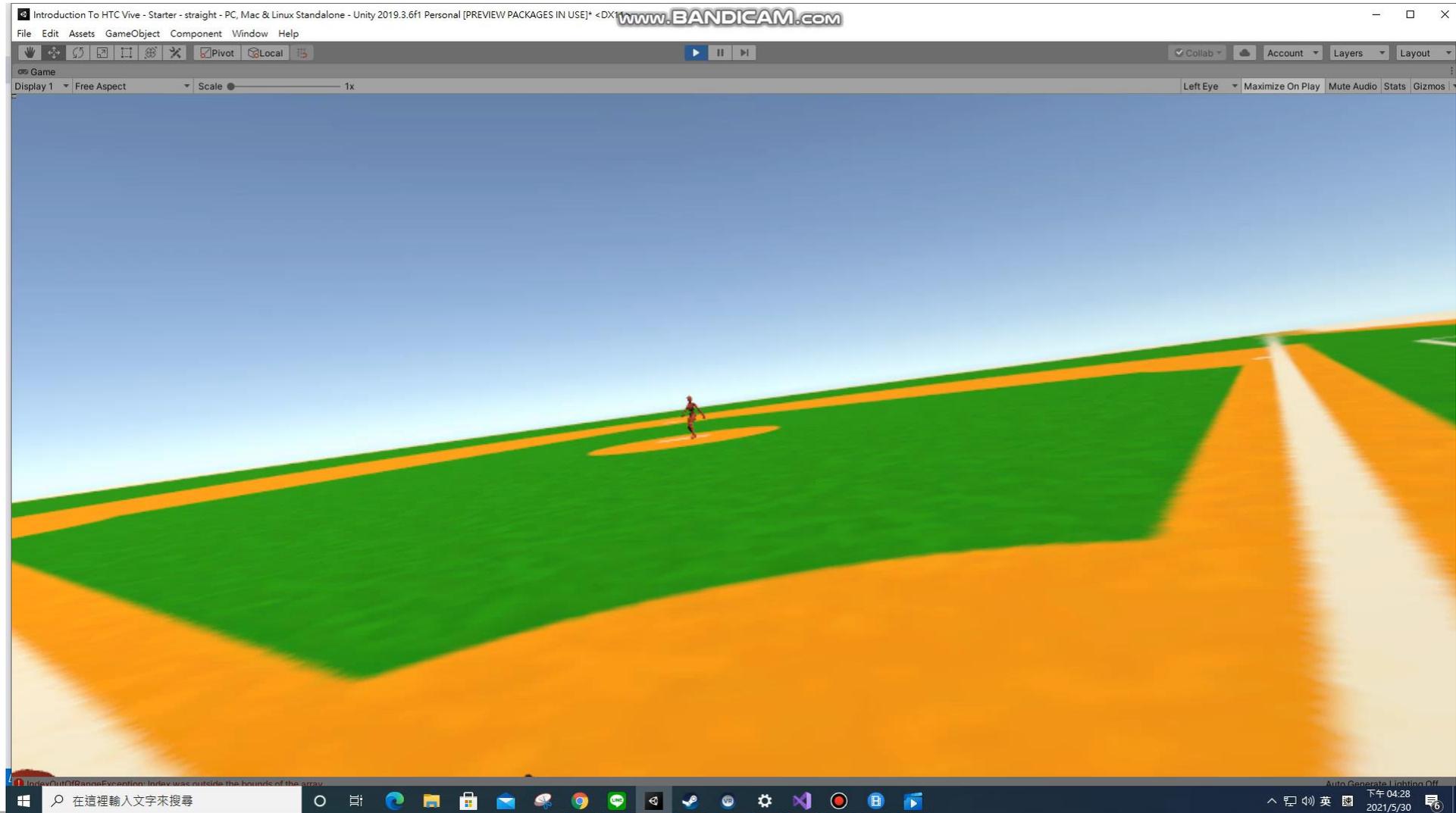
# 4-5 打擊力道正常



4-5 打擊力道正常



# 4-6 打擊力道大



4-6打擊力道大





## 第五部分

評估

# 5-1 評估

功能：

新增3種球種

從使用者介面跳轉到遊戲畫面

能順利觀察腦波的變化

結合腦波數值與打擊力度大小





## 第六部分

心得與未來展望

## 6-1 心得與未來展望

參加了這次的校園示範點專題創作後，我們了解到虛擬實境可以廣泛運用在許多領域，棒球只是一小部分的應用而已。在製作過程中我們也遇到了不少棘手的問題，尤其我們這組都是從零開始接觸UNITY，但很幸運地大部分都克服了。在未來我們希望可以讓遊戲的畫面變得更加精細，讓使用者更能有身歷其境的感覺。

## 6-2 參考資料

- 1.腦波與運動流暢經驗之可能關係([doi:10.6223/qcpe.2203.200809.1701](https://doi.org/10.6223/qcpe.2203.200809.1701))
- 2.大學生於中強度運動前、運動期間及運動後之腦波變化
- 3.專注與放鬆腦波辨識之研究



謝謝大家