

## Bài thực hành số 6 – Leader board, archivement trong google play service

### Mục tiêu

- Cách sử dụng google play service để tạo leader board, archivement trong game

.....

.....

## Bài 1: Kết hợp Component và Scripts

### Hướng dẫn

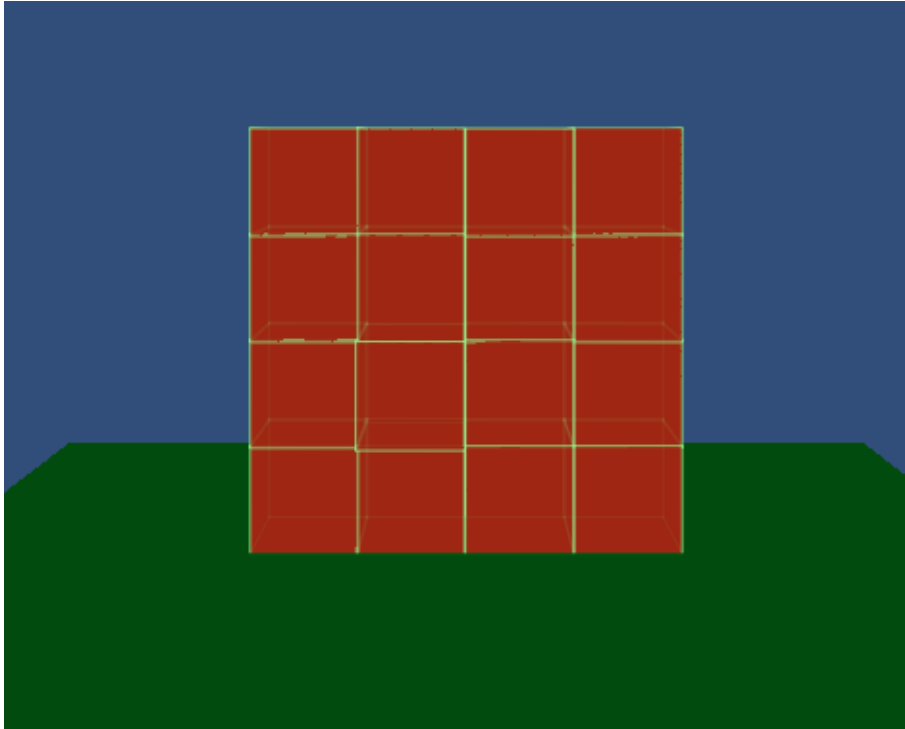
1. Tạo đối tượng mới Plane, đặt tên là Ground và vị trí của đối tượng có tọa độ (0,0,0)
2. Tạo một cube object, tên là **Physics Cube** và attach màu đỏ (red)
3. Lựa chọn PhysicsCube trong cửa sổ Hierarchy và từ cửa sổ Inspector, ta chọn Add Component, Physics, Rigidbody,

Trong cửa sổ Inspector ta thiết lập thuộc tính **Rigidbody** với **Mass** là **1**,

Vị trí của Cube có tọa độ (0,.5,0). Việc này cần thiết lập chính xác vị trí trên bề mặt ground plane.

Tạo một **Prefab** từ PhysicsCube mà bạn đã tạo trong bước 2 và bước 3

4. Tạo một wall từ 16 PhysicsCube Prefabs giống như hình dưới



Sử dụng lệnh **Duplicate** và **coordinate positions** cho **x** và **y** trong cửa sổ **Inspector** để enable bạn giong chính xác các hình cubes như hình.

5. Tạo mới một đối tượng Cube với tên là **Launcher**.
6. Vị trí của Cube trong World ở phía sau Main Camera trong khung nhìn theo trục X của Ground và ở phía trên bề mặt Ground.
7. Tạo một script để gọi **launchMissile**.
8. Tạo một biến **aMissile** có kiểu **GameObject** ở phần trên cùng của script.
9. Thêm một hàm có chức năng Update như sau:
 

```
function fireMissile(){

    Instantiate(aMissile, transform.position, transform.rotation);

}
```
10. Attach script vào Launcher
11. Tạo một sphere để gọi Missile và thêm chất liệu mà bạn chọn.
12. Thêm một đối tượng vật lý Rigid Body và một thành phần Constant Force với giá trị trục z nằm trong khoảng 200.
13. Tạo một Prefab từ sphere sử dụng cho việc khởi tạo.
14. Tạo một script với tên là **GUIControl** và viết code như sau.

```
#pragma strict
```

```
var cannon:GameObject;
```

```
static var numberOfMissiles:int = 0;
```

```
var missileCount: String;
```

```
function OnGUI () {
```

```
cannon = GameObject.Find("Launcher");
```

```
GUI.TextField (Rect (400, 30, 50, 20), missileCount, 25);
```

```
if (GUI.Button (Rect (10,10,50,40), "Fire!")) {
```

```
    cannon.GetComponent(launchMissile).fireMissile();
```

```
    setNumberOfTries();
```

```
}
```

```
}
```

```
function Start () {
```

```
}
```

```
function Update () {
```

```
missileCount = numberOfMissiles.ToString();
```

```
}
```

```
function setNumberOfTries(){
```

```
numberOfMissiles = numberOfMissiles + 1;
```

```
}
```

15. Attach script vào đối tượng **Launcher**.

16. Kéo và thả prefab Missile vào biến aMissile của script Launcher trong cửa sổ Inspector.
17. Chạy scene. Tên lửa (missile) nên cháy khi bạn ấn nút lửa và đập vào tường. Trường text nên update mỗi lần khi ấn nút lửa (button fire)
18. Attach script vào prefab của Missile và remove đối tượng từ World sau một khoảng thời gian thích hợp.

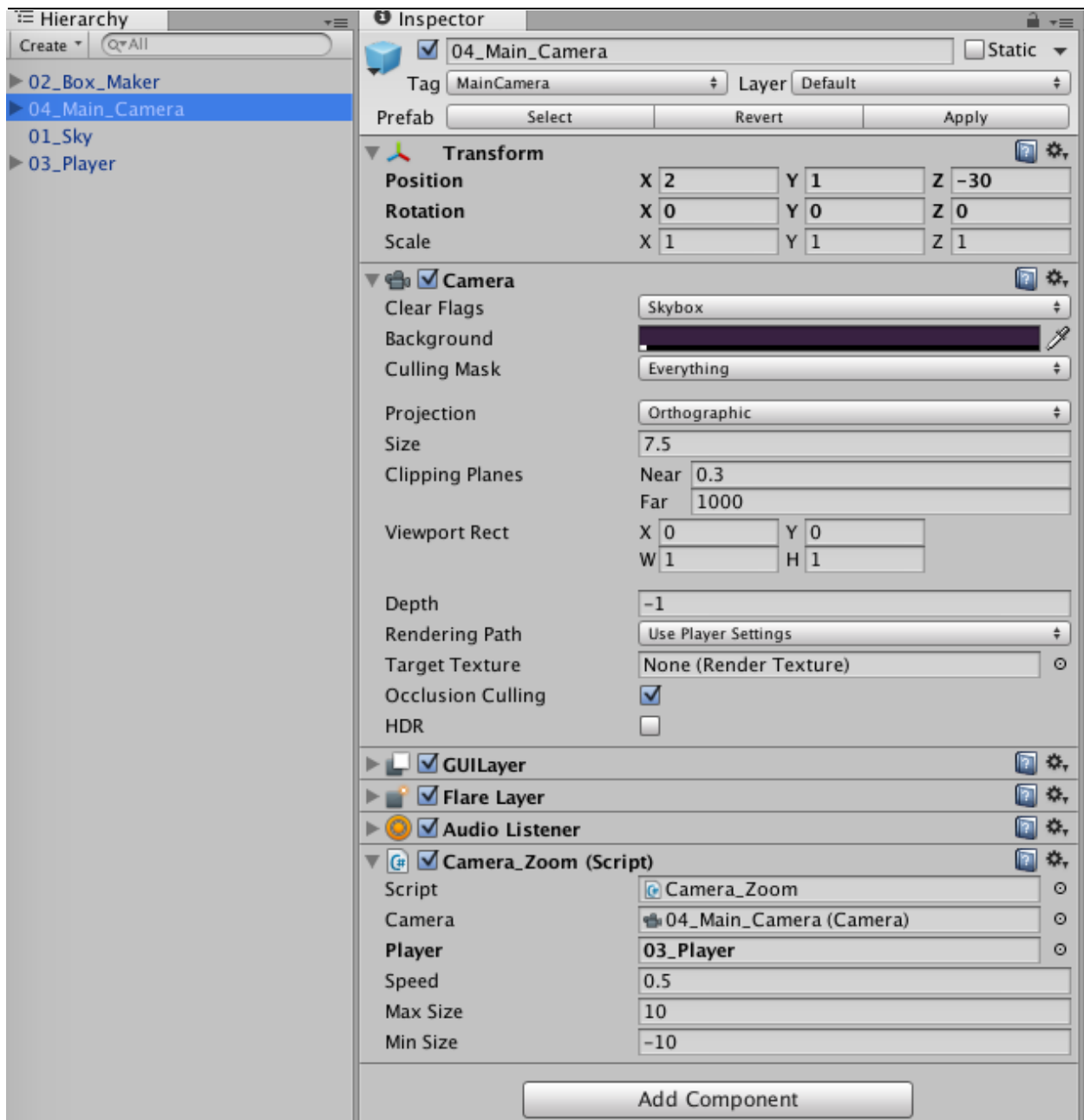
**Sản phẩm phải nộp:** Nộp toàn bộ dự án game vừa tạo

## **Bài 2 – Dự án game Assignment: Xây dựng scene Main\_Camera**

### **Xây dựng Camera chạy theo Character**

#### **Hướng dẫn**

- Download bài 2 của Lab5 và làm tiếp
- Tạo scene 04\_Main\_Camera như hình sau:



-Viết code cho Camera\_Zoom.cs



```

1  using UnityEngine;
2  using System.Collections;
3
4  public class Camera_Zoom : MonoBehaviour
5  {
6      /**
7       *
8       * @author hungnq
9       */
10     //Khai báo các biến cần sử dụng
11     public Camera _camera;
12     public GameObject _player;
13     public float speed;
14     public float MaxSize = 7.5f;
15     public float MinSize = 5.5f;
16     float CameraSize = 6f;
17
18     void Start(){
19         //Lấy đối tượng được tạo trong cửa sổ Hierarchy
20         _player = GameObject.Find("03_Player");
21     }
22
23
24     void Update ()
25     {
26         //Thay đổi camera theo vị trí của chrracter
27
28         if (_player != null)
29             CameraSize = 5f + _player.transform.position.y;
30
31         if (CameraSize >= MaxSize) {
32             CameraSize = MaxSize;
33         }
34
35         if (CameraSize <= MinSize) {
36             CameraSize = MinSize;
37         }
38
39         //Cập nhật vị trí của Camera
40         _camera.orthographicSize = Mathf.Lerp (_camera.orthographicSize, CameraSize, Time.deltaTime / speed);
41
42     }
43 }
44

```

**Sản phẩm phải nộp:** Nộp toàn bộ dự án game vừa tạo

### Bài 3 – Dự án Game Lab: Các Block được đưa vào môi trường (Environment) hoạt động của nhân vật một cách ngẫu nhiên

#### Hướng dẫn

- Download bài 3 của Lab5 và làm tiếp (Project Lab), sau đó nén lại thành project và nộp lên LMS
- Sinh viên viết đoạn code sau vào file ScrollEnvironment.cs

```

ScrollEnvironment.cs
1  using UnityEngine;
2  using System.Collections;
3
4  public class ScrollEnvironment : MonoBehaviour {
5      /**
6       *
7       * @author hungnq
8       */
9      public float Velocity;
10     public GameObject Player;
11     private float offset = 0;
12     // Update is called once per frame
13     void Update () {
14         if (Player == null) return;
15         offset += Velocity * Player.rigidbody2D.velocity.x;
16         if (offset > 1)
17             offset = 1 - offset;
18         if (offset < -1)
19             offset = -1 - offset;
20         GetComponent<MeshRenderer>().material.mainTextureOffset = new Vector2(offset,0);
21     }
22 }
23

```

**Sản phẩm phải nộp:** Nộp toàn bộ dự án game vừa tạo

## Yêu cầu nộp bài

Cuối giờ thực hành, sinh viên tạo thư mục theo tên <Tên đăng nhập SV>\_Lab6, chứa tất cả sản phẩm của những bài lab trên, nén lại thành file zip và upload lên mục nộp bài tương ứng trên LMS.

## Đánh giá bài lab

STT	Bài số	Điểm
1	Bài 1	4
2	Bài 2	4
3	Bài 3	2
	-	

	-	
	-	