

## Bài thực hành số 3 – Tạo và cập nhật Heads-Up Display

### Mục tiêu

- Biết cách tạo và cập nhật Heads-Up Display

.....

.....

## Bài 1: Thiết lập HUD

### Hướng dẫn

1. Tạo mới một project Unity 3D với tên là **displayhud** và import: Controller, Physics Materials và Scripts assets.
2. Tạo một folder mới tên là **Textures** nằm trong asset của project displayhud.
3. Download file **textures.zip** từ LMS (tài nguyên cung cấp sẵn)
4. Trong unity ta import package battery: **Assets – Import New Asset...** và import tất cả các textures bạn đã download trong bước 3. Unity sẽ lưu trữ các textures trong thư mục Texture.
5. Download **battery package** được cung cấp sẵn trong tài nguyên
6. Trong Unity ta import package battery: **Assets – Import Package – Custom Package...** đây là một **Prefab** của đối tượng battery với textures đã tồn tại. Unity sẽ đặt trong thư mục **Prefab**.
7. Tạo một file mới JavaScript với tên là **BatteryCollect** và thêm đoạn code sau:

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

```
var charge1tex:Texture2D;
```

```
var charge2tex:Texture2D;
```

```
var charge3tex:Texture2D;
```

```
var charge4tex:Texture2D;
```

```
var charge0tex:Texture2D;
```

```
function Start(){
```

```
    guiTexture.enabled = false;
```

```
    charge = 0;
```

```
}
```

```
/*
```

update function checks status of charge variable

which is increased via an external script

each time the player collides (collects) with a battery

```
*/
```

```
function Update () {  
  
    /*  
  
        first battery collected  
        assign the first texture image to guiTexture  
        and enable (display) texture  
  
    */  
  
    if(charge == 1){  
  
        guiTexture.texture = charge1tex;  
        guiTexture.enabled = true;  
  
    }  
  
    // display subsequent textures to indicate power collected
```

---

```
else if(charge == 2){

    guiTexture.texture = charge2tex;

}

else if(charge == 3){

    guiTexture.texture = charge3tex;

}

else if(charge == 4){

    guiTexture.texture = charge4tex;

}

else if(charge == 0){

    guiTexture.texture = charge0tex;

}

}
```

8. Trong Unity từ thư mục **Textures** ta chọn texture **battery\_nocharge** sau đó từ **Main Menu** ta chọn **GameObject – Create Other – GUI Texture**.

Texture empty battery sẽ xuất hiện ở trung tâm của cửa sổ **Game**. Bạn sẽ cần điều chỉnh tọa độ **x, y** để di chuyển texture sang góc left bottom của cửa sổ Game. Trong cửa sổ Hierarchy ta đổi tên **battery\_nocharge** thành **HUD**.

9. Bây giờ bạn chọn script BatteryCollect, bạn sẽ thấy 5 biến khai báo publish trong cửa sổ **Inspector**.
10. Nếu bạn chọn folder **Textures** bạn sẽ thấy 5 battery charge textures. Kéo mỗi texture từ folder Texture vào các biến phù hợp trong script.

Điều này làm cho mối liên hệ giữa các biến trong script và texture liên kết với nhau trong script của HUD.

11. Tạo một Plane, add một chất liệu phù hợp. Tạo một nguồn sáng Direct và kéo thả vào Controllor trong scene.

12. Thêm một vị trí cho battery Prefabs trong scene.

13. Tạo một file mới JavaScript tên là PlayerCollision and và thêm đoạn code sau:

```
function Start () {  
  
  
}
```

```
function Update () {
```

```
}
```

```
function OnTriggerEnter(collisionInfo : Collider){
```

```
if(collisionInfo.gameObject.tag == "battery"){
```

```
BatteryCollect.charge++;
```

```
Destroy(collisionInfo.gameObject);
```

```
}
```

```
}
```

#### 14. Attach script **BatteryCollect** vào đối tượng **HUD**

Chạy scene. Bạn có thể điều hướng các va chạm tới object battery trong HUD và cập nhật cho phù hợp.

Bài tập:

1. Tạo một artwork thay thế trong Photoshop và replace HUD textures. Sử dụng format phù hợp là png 32.

2. Tạo một kịch bản với tính năng khối spheres có 5 màu để người chơi có thể tập hợp. Bộ sưu tập spheres sẽ hiển thị trong HUD với màu sắc liên quan.

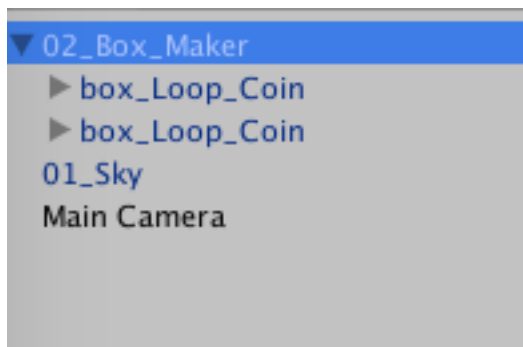
**Sản phẩm phải nộp:** Nộp toàn bộ dự án game vừa tạo

## Bài 2 – Dự án Game Assignment: Xây dựng scene Box\_Maker:

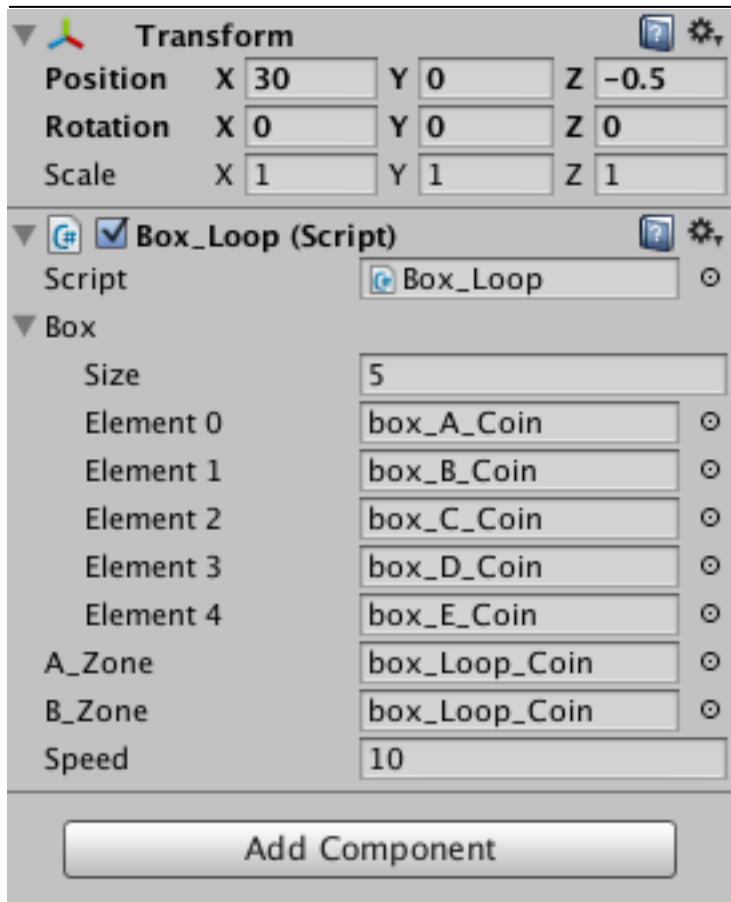
### Background và chương ngại vật chuyển động không cùng tốc độ

#### Hướng dẫn

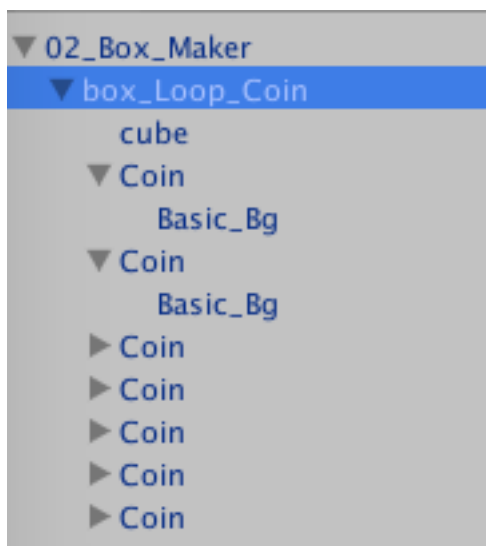
- Download bài 2 của Lab2 và làm tiếp
- Tạo đối tượng cha 02\_Box\_Maker



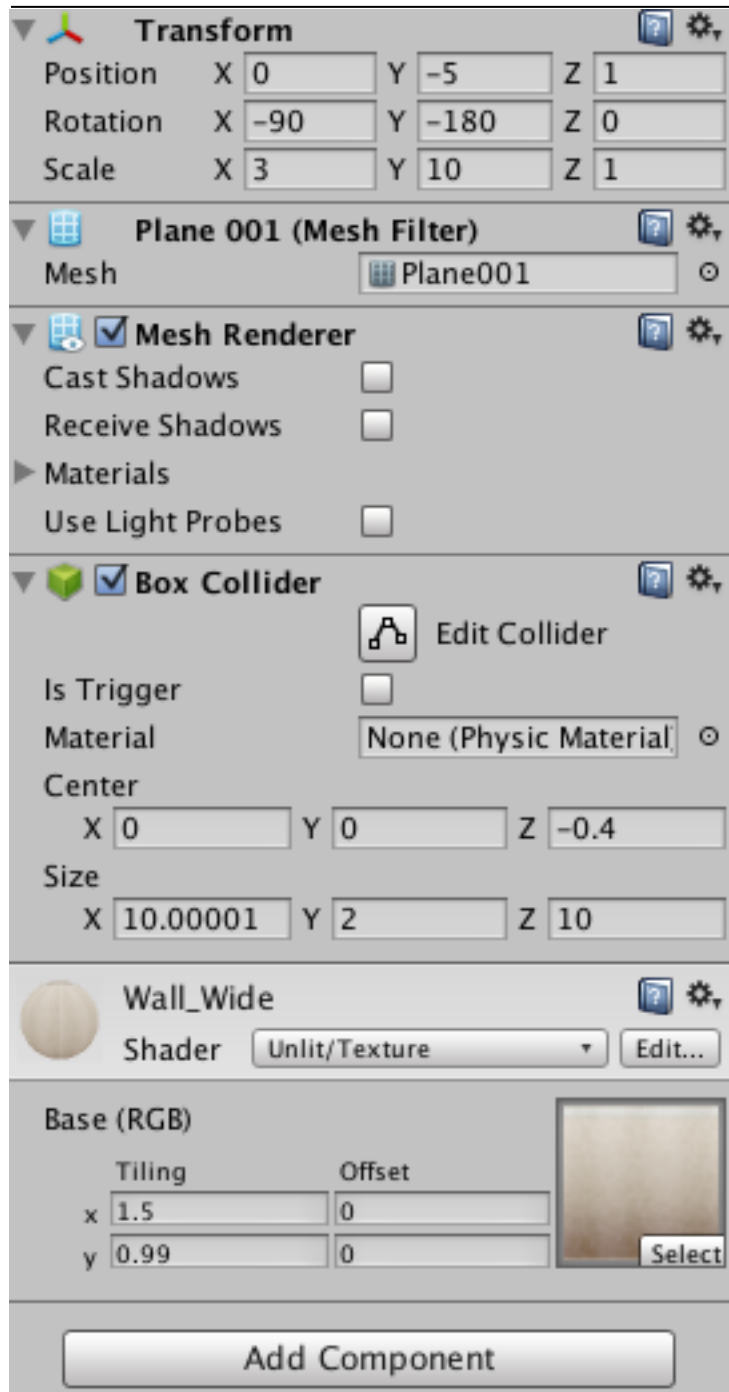




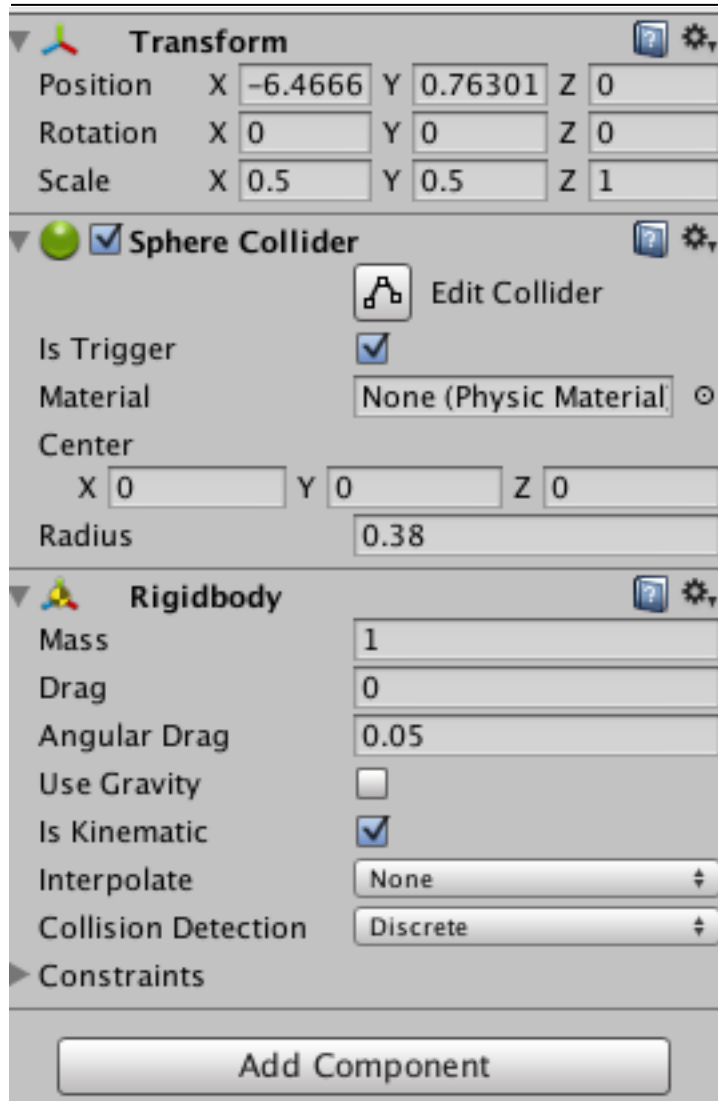
- Tạo đối tượng cha box\_Loop\_Coin



- Chướng ngại vật cube



-Cài đặt coin:



-Cài đặt Box\_Loop.cs

```

Box_Loop.cs
1  using UnityEngine;
2  using System.Collections;
3
4  public class Box_Loop : MonoBehaviour {
5
6      /**
7       *
8       * @author hungnq
9       */
10     //khai bao mang Box
11     public GameObject[] Box;
12     //khai bao gameobject kieu A
13     public GameObject A_Zone;
14     //khai bao gameobject kieu B
15     public GameObject B_Zone;
16
17     //khai bao toc do di chuyen chuong ngai vat
18     public float Speed = 5f;
19
20     void Update () {
21
22         MOVE();
23     }
24
25
26     //Ham tao chuong ngai vat
27     public void MAKE(){
28         //gan vi tri chuong ngai vat B sang A
29         B_Zone=A_Zone;
30         //lay ngau nhien mot vi tri
31         int a = Random.Range(0,5);
32         //khởi tạo chuong ngai vat A
33         A_Zone = Instantiate(Box[a], new Vector3(30,0,0), transform.rotation) as GameObject;
34     }
35
36
37
38     //Ham di chuyen chuong ngai vat
39     public void MOVE(){
40         //di chuyen vi tri chuong ngai vat A
41         A_Zone.transform.Translate(Vector3.left * Speed *Time.deltaTime, Space.World);
42         //di chuyen chuong ngai vat B
43         B_Zone.transform.Translate(Vector3.left * Speed *Time.deltaTime, Space.World);
44         //Neu trung vi tri thi huy chuong ngai vat
45         if(A_Zone.transform.position.x<=0){
46             DEATH();
47         }
48     }
49
50
51     //ham huy chuong ngai vat
52     public void DEATH(){
53         //huy chuong ngai vat B
54         Destroy(B_Zone);
55         //goi ham tao chuong ngai vat
56         MAKE();
57     }
58 }
59
--

```

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

### **Bài 3 – Dự án Game Lab: Sử dụng Enemy di chuyển theo quỹ đạo định sẵn**

#### **Hướng dẫn**

- Download bài 2 của Lab2 và làm tiếp, sau đó nén lại thành project và nộp lên LMS
- Sinh viên viết đoạn code sau vào file Bullet.cs

```

Bullet.cs
1  using UnityEngine;
2  using System.Collections;
3
4  public class Bullet : MonoBehaviour {
5      /**
6       *
7       * @author hungnq
8       */
9      private bool IsFiring = false;
10     private float direction = -1;
11     public int Damage = 1;
12     float startX = 0;
13     // Use this for initialization
14     void Start () {
15
16     }
17     void OnCollisionEnter2D(Collision2D collision)
18     {
19         if (collision.gameObject.tag == "Enemy")
20         {
21             MovingEnemy enemy = collision.gameObject.GetComponent<MovingEnemy>();
22             if (enemy != null)
23             {
24                 if (enemy.Blood > 0)
25                 {
26                     enemy.Blood -= Damage;
27                 }
28                 if (enemy.Blood <= 0)
29                 {
30                     PlayerController.Instance.KillEnemy(enemy);
31                     Destroy(gameObject);
32                 }
33             }
34         }
35         else if (collision.gameObject.tag == "Ground")
36         {
37
38             Vector2 v=rigidbody2D.velocity;
39             v.x -= 0.5f*Mathf.Sign(rigidbody2D.velocity.x);
40             rigidbody2D.velocity = v;
41             if (Mathf.Abs(rigidbody2D.velocity.x) < 1f)
42                 Destroy(gameObject);
43         }
44     }
45 }

```

```

46     public void Fire(float velocity)
47     {
48         Vector2 force = new Vector2();
49         force.x = velocity *1000;
50         force.y = 50;
51         direction = velocity;
52         rigidbody2D.AddForce(force);
53         IsFiring = true;
54     }
55     // Update is called once per frame
56     void Update () {
57         if (IsFiring && Mathf.Sign(direction)!=Mathf.Sign(rigidbody2D.velocity.x))
58         {
59             Destroy(gameObject);
60         }
61         else
62         {
63             rigidbody2D.rotation += 1;
64             if (rigidbody2D.rotation >= 180)
65                 rigidbody2D.rotation = 0;
66         }
67     }
68 }

```

**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>\_Lab3, 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
	-	
	-	
	-	