

## Bài thực hành số 7 – Sử dụng Colour Switcher; Quản lý Tag, collisions và triggers

### Mục tiêu

- Sử dụng Colour Switcher; Quản lý Tag, collisions và triggers

.....

.....

## Bài 1a: Chuyển đổi màu sắc của một đối tượng thông qua một phím bấm

### Hướng dẫn

1. Từ **File Menu** ta chọn **New Project**.
2. Trong **New Project dialogue box** ta chọn the các gói có thể import như: **Character Controller, Physics Materials, Scripts**. Lưu project là **Scripting Examples**.
3. Từ **GameObject Menu** ta chọn **Create Other – Cube**. Một cube sẽ xuất hiện trong scene.
4. Từ **Assets Menu** ta chọn **Create – Javascript**.
5. Một script có tên mặc định là **NewBehaviourScript** sẽ xuất hiện trong **Assets** của cửa sổ **Project**. Ta đổi tên script thành **ColorSwitcher**.
6. Double click vào script trong cửa sổ project để mở trình editor và thêm đoạn code sau trong hàm **Update**:

```
if(Input.GetKeyDown(KeyCode.R)){  
  
    gameObject.renderer.material.color = Color.red;  
  
}
```

7. Lưu script. Bây giờ kéo script từ cửa sổ **Project** và thả nó vào đối tượng **Cube** trong cửa sổ **Hierarchy**.

8. Lựa chọn play từ controller. Màu sắc của cube sẽ thay đổi sang red khi bạn nhấn phím R.

9. Thực hành:

Sử dụng đoạn mã trên như một ví dụ về các khối(cube) có màu green, blue, và white. Ta cần thay đổi các màu của cube khi nhấn các phím sau trên bàn phím: G, B, và W.

Lưu scene là ColourSwitcher khi bạn hoàn thành.

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

## **Bài 1b: Sử dụng Tag Manager**

### **Hướng dẫn**

Bài này hướng dẫn bạn cách làm thế nào để tạo ra 2 illuminated spheres (quả cầu chiếu sáng) trong World và Tag (name) chúng khi va chạm và kích hoạt sự kiện (trigger events) tương tác người điều khiển

1. Tạo mới một project và đặt tên là **Object Collisions**, sau đó import các assets sau:

### **Character Controller, Physics Materials, Scripts, Terrain.**

2. Tạo một **Terrain** và sau đó thiết lập độ phân giải là **500 x 500**
3. Tạo một nguồn sáng trực tiếp (**Direct Light**) trong scene.
4. Chọn terrain và từ thuộc tính của terrain trong cửa sổ **Inspector** ta chọn **Edit Textures** – và chọn texture **Grass (Hill)**.
5. Từ **Assets Menu** ta chọn **Create–Material** và chọn màu **Red** cho chất liệu.
6. Tạo một chất liệu thứ 2 với màu xanh **Blue**.
7. Từ **Game Object Menu** ta chọn **Create Other - Sphere**.
8. Từ **Game Object Menu** ta chọn **Create Other – Point Light**.
9. Trong cửa sổ **Inspector** ta gán tọa độ vị trí cho **Sphere** và **Point Light** đều là **0,0,0**.
10. Kéo **Point Light** dưới Sphere trong cửa sổ **Hierarchy** để Sphere thành cha của Point Light.
11. Tạo một **Prefab** từ các objects được kéo từ Sphere vào cửa sổ **Project–Assetts** và đổi tên thành **LightSphere**.
12. Đặt **LightSphere** ở đâu đó xung quanh khu vực trung tâm của terrain với **2** đơn vị **y** trên bề mặt.

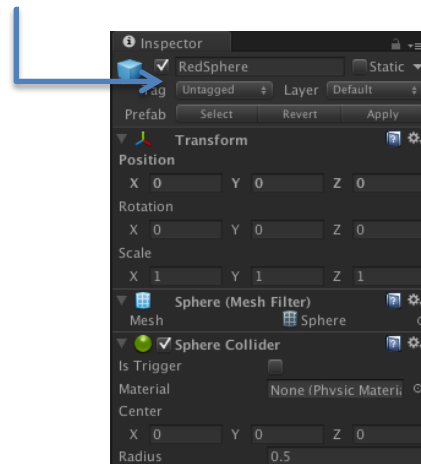
13. Kéo chất liệu **Red** vào **LightSphere** trong cửa sổ **Hierarchy** và đổi tên thành **RedSphere**.

Bước tiếp theo sẽ hướng dẫn bạn cách làm thế nào để quản lý **Tag** (**name**) cho một

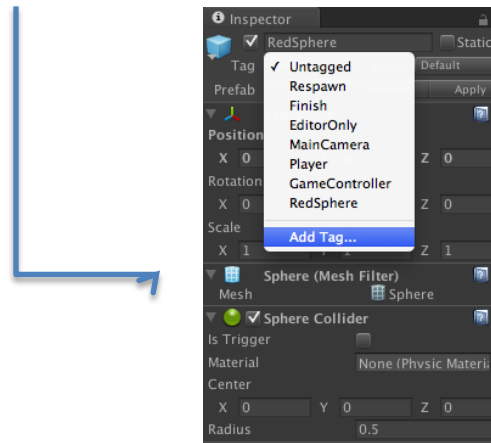
Object để có thể tham chiếu đến **script**.

14. Lựa chọn **RedSphere** trong cửa sổ **Hierarchy**.

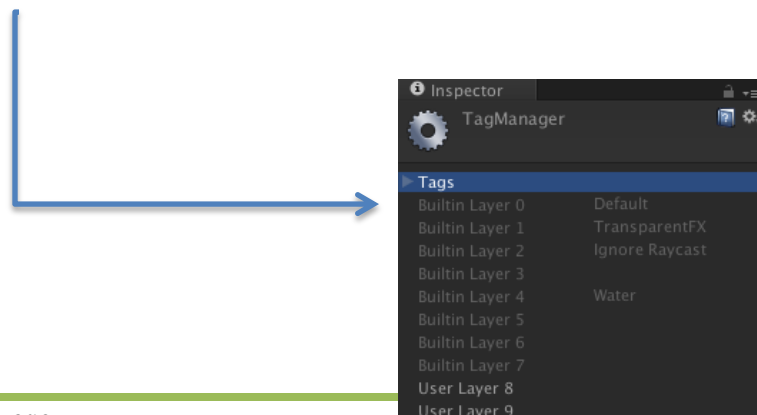
15. Trong cửa sổ **Inspector** ta chọn từ **Tag** xổ xuống



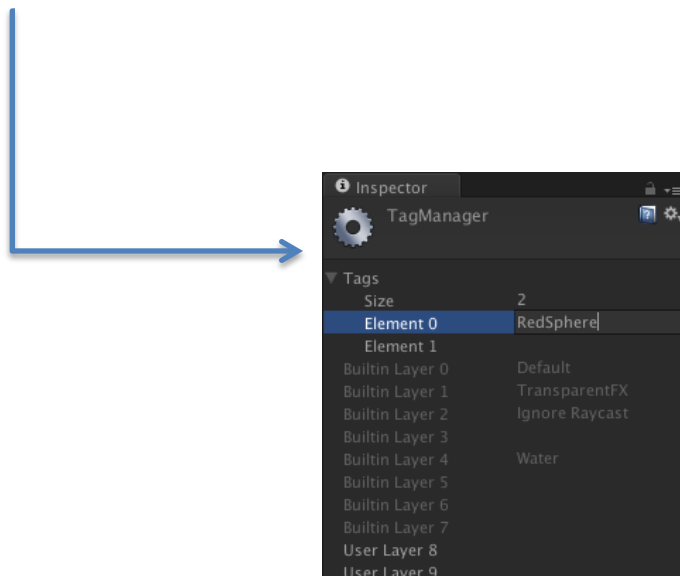
16. Từ **Drop List** ta chọn **Add Tag...**



17. Lựa chọn **Tags Arrow** để thấy **Tags Element Slots**



18. Lựa chọn vùng phải của những tag **available Tags Element Slot** và gõ **RedSphere**.



19. Bây giờ có thể tạo **LightSphere** khác từ **Prefab** mà bạn tạo ở bước **11**.
20. Lặp lại các bước từ **12** đến **18** sử dụng chất liệu với màu **Blue** và đổi tên từ **LightSphere** thành **BlueSphere** và thêm một **Tag** bằng cách sử dụng **Tag Manager**.
21. Khi hoàn thành bạn có **Red Sphere** và **Blue Sphere** trong scene và có prefab **LightSphere**, có tên tương ứng là **RedSphere** và **BlueSphere**. Mỗi đối tượng sẽ có **Tag Name** tương ứng được liệt kê trong **Tag Manager**.
22. Lưu scene là **LightSpheres**.

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

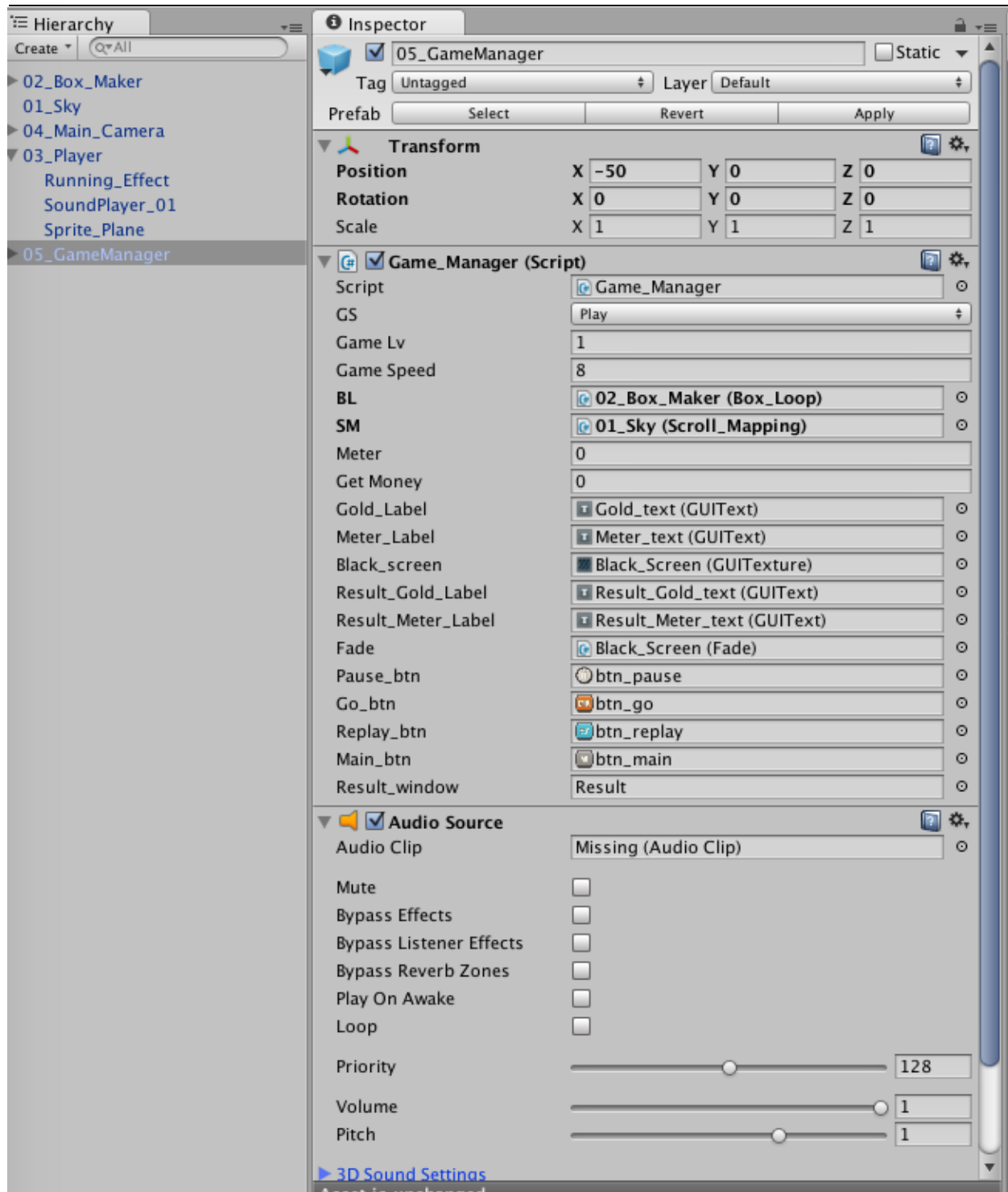


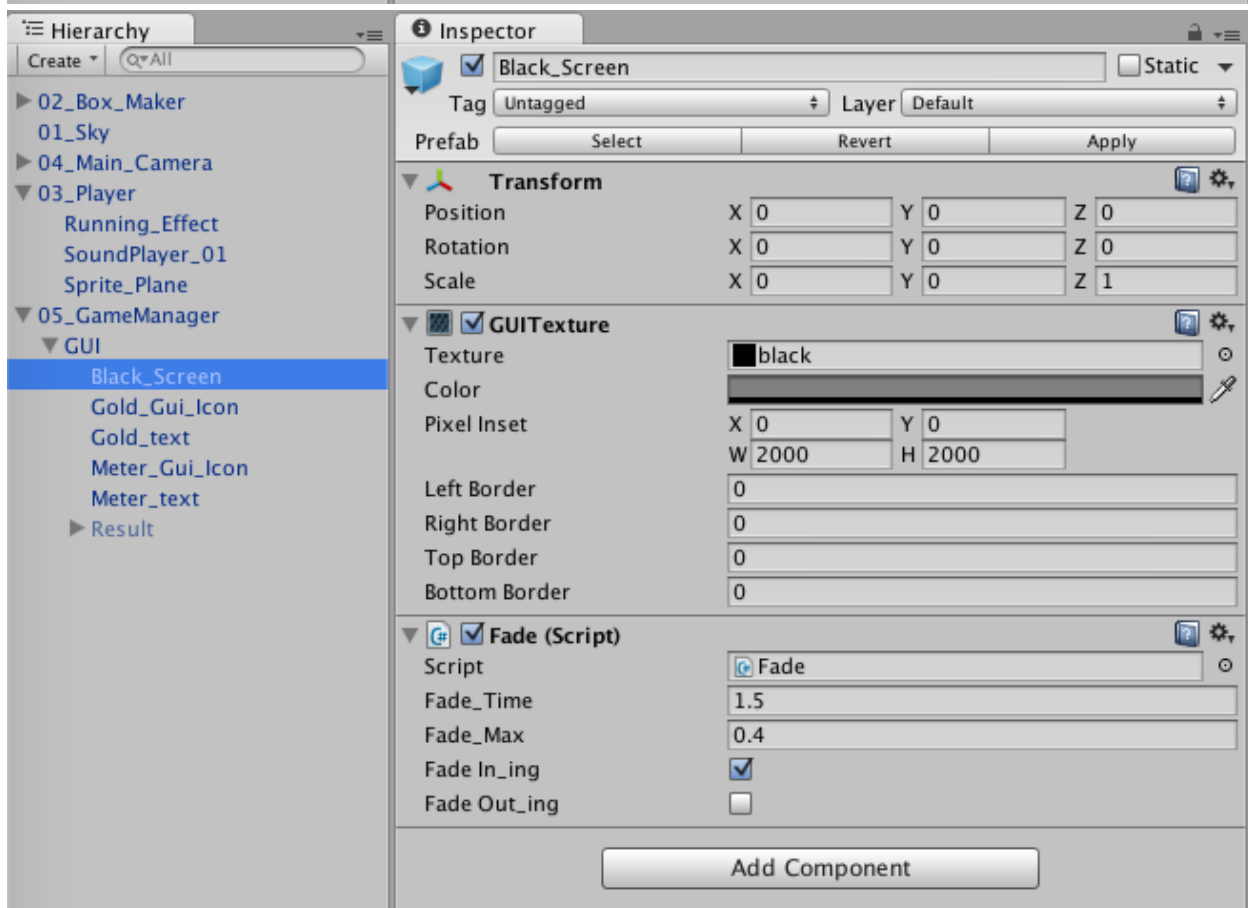
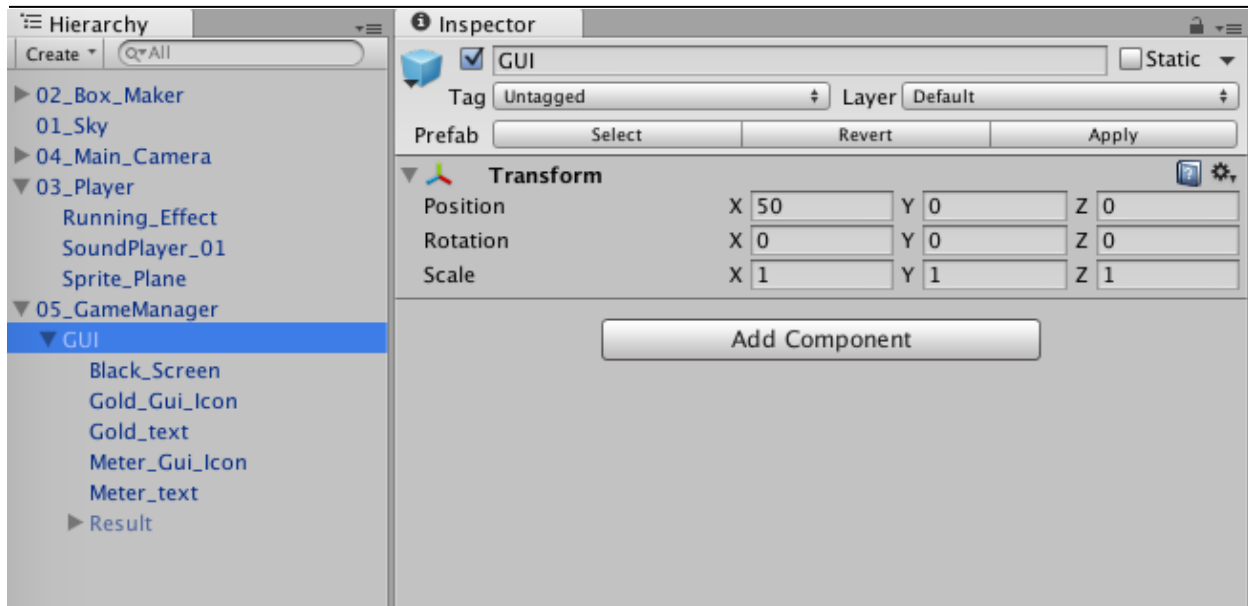
## **Bài 2: Dự án game Assignment: Xây dựng scene GameManager (phần 1)**

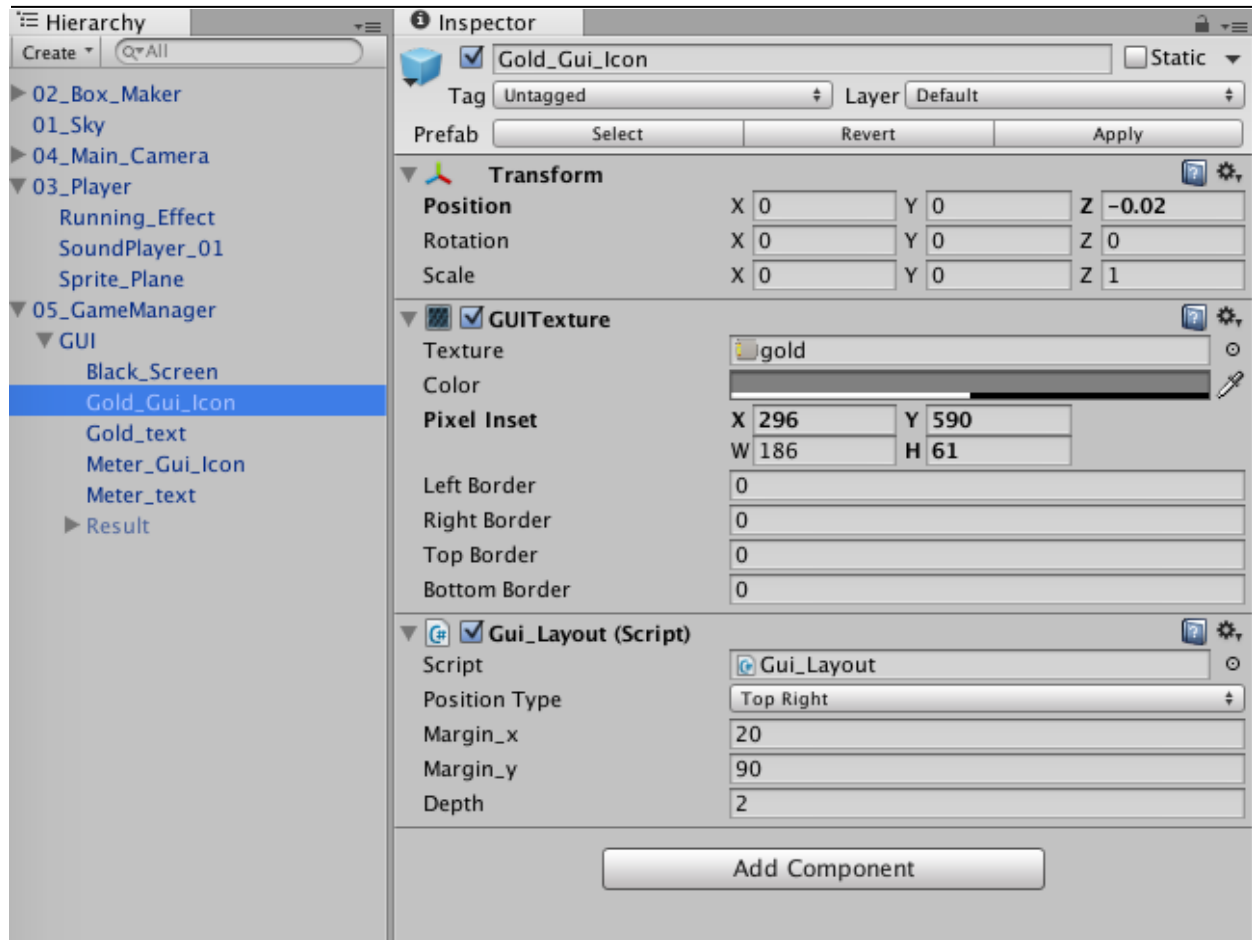
**Xây dựng game hoàn chỉnh: nhân vật chạy, ăn coin, có điều khiển bằng bàn phím**

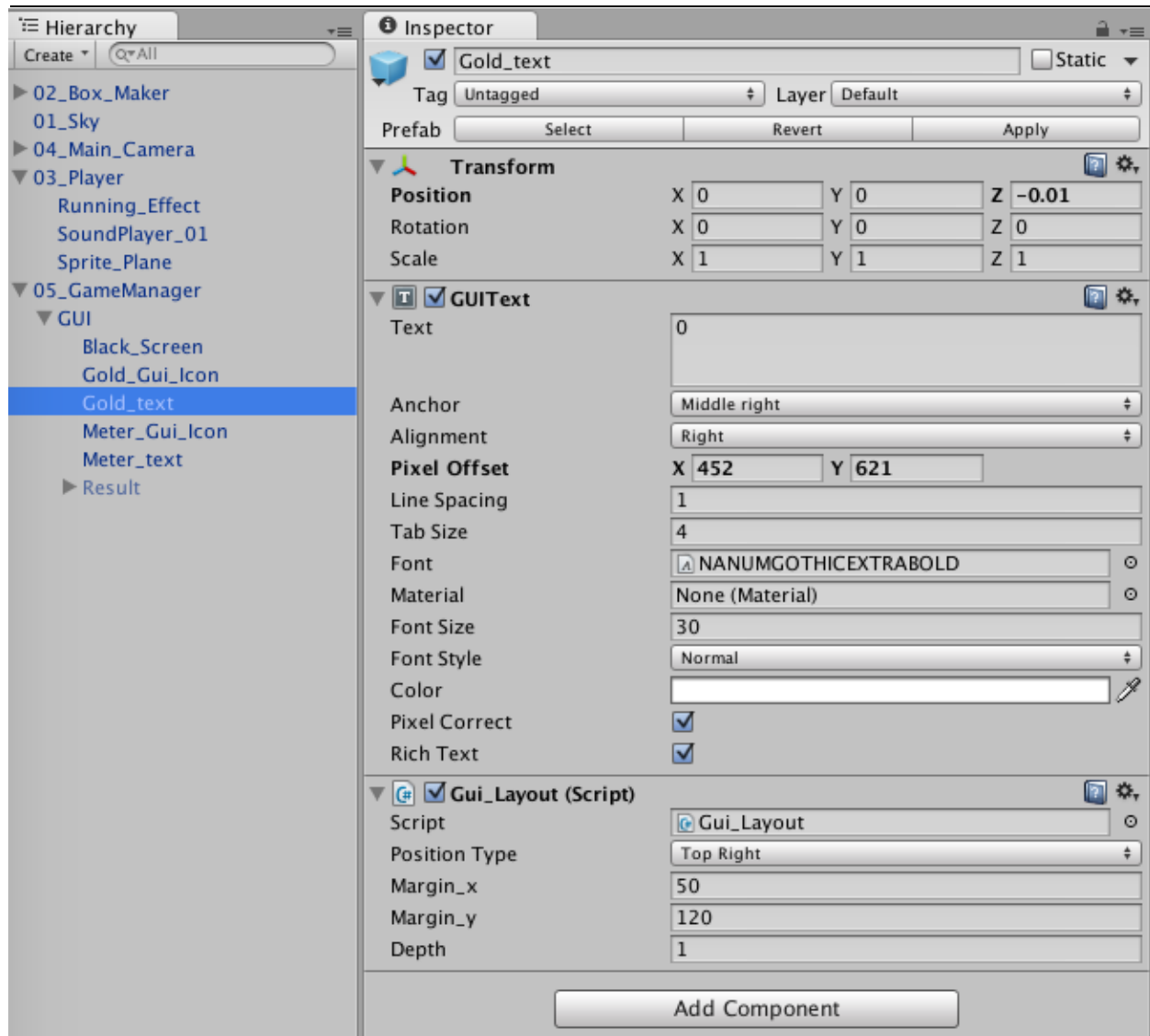
### **Hướng dẫn**

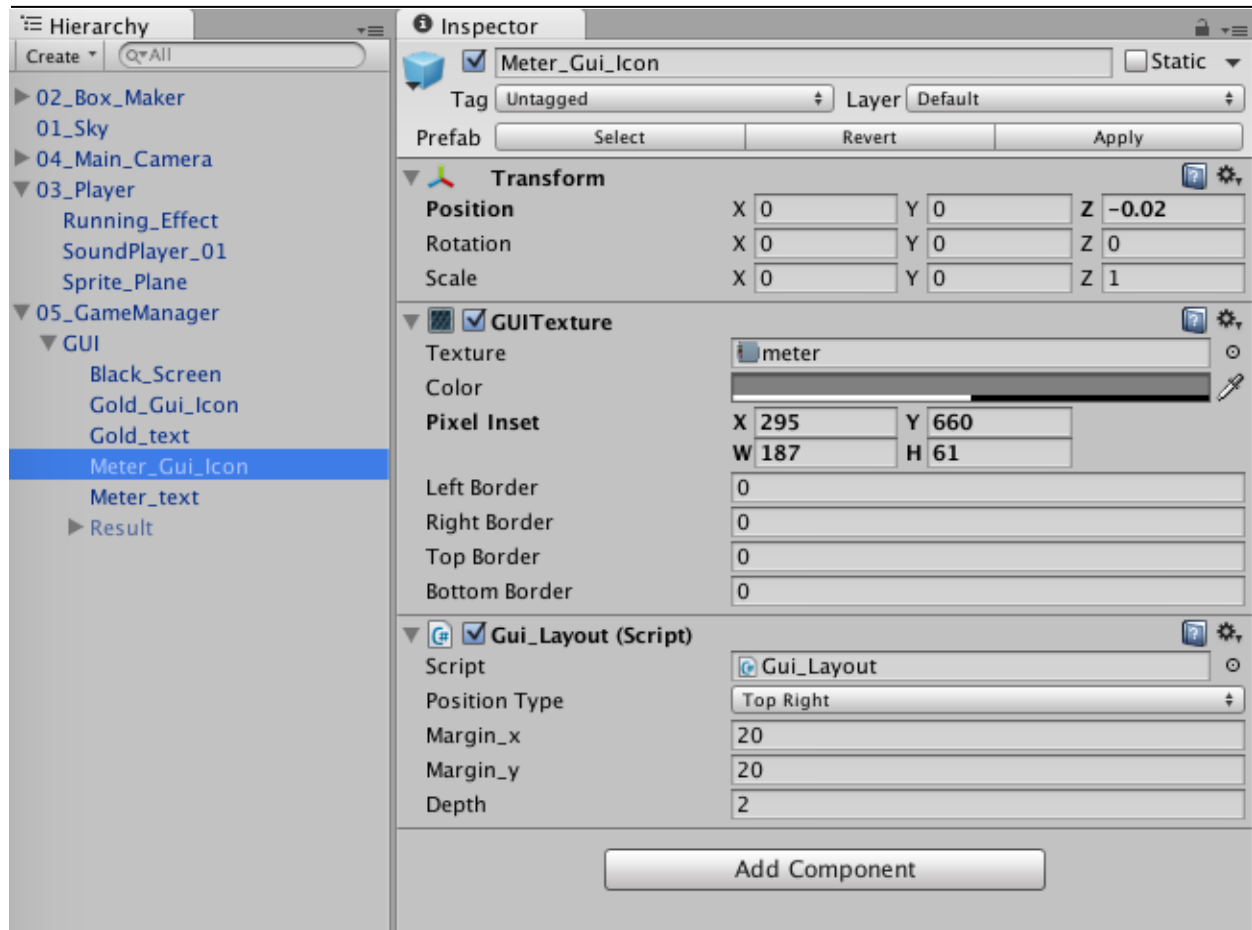
- Download bài 2 của Lab6 (Project Assignment) và làm tiếp
- Xây dựng scene 05\_GameManager

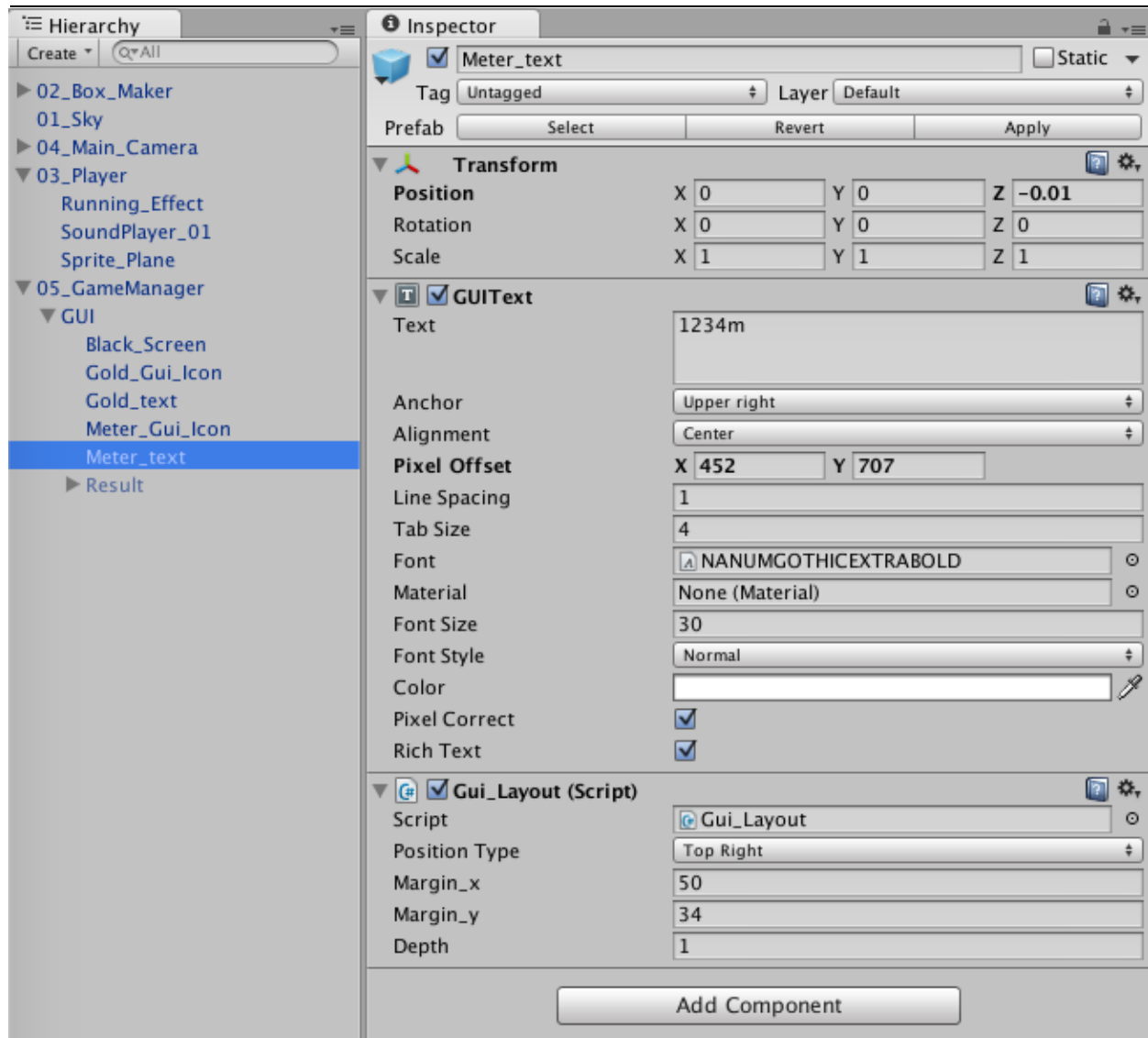


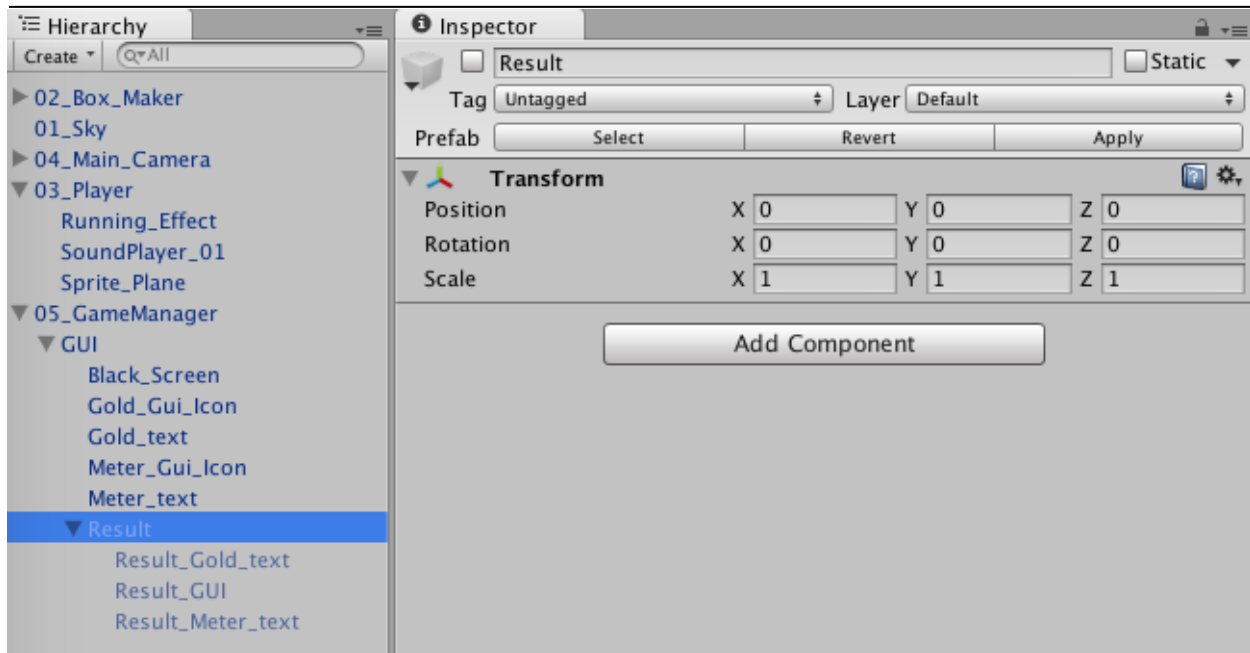




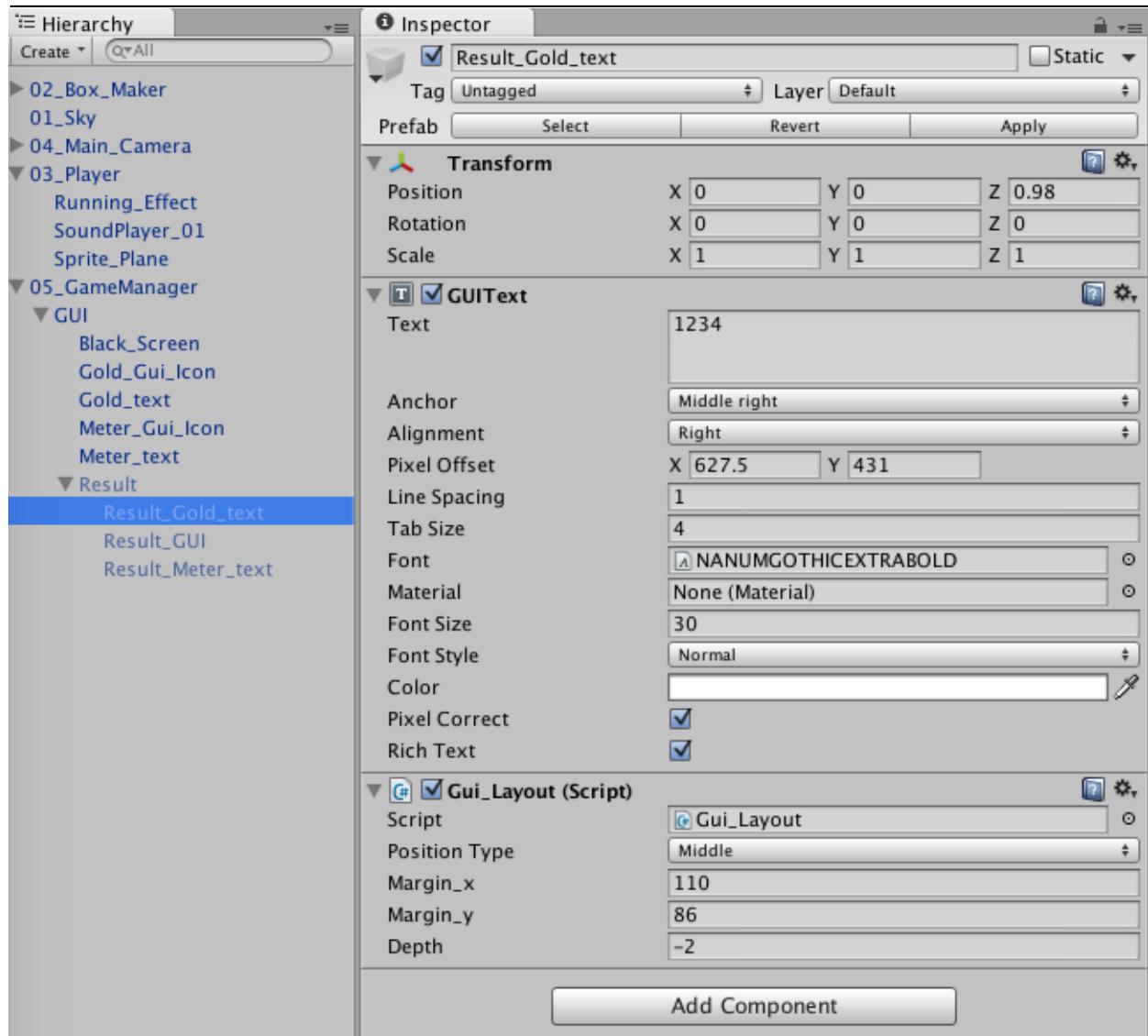


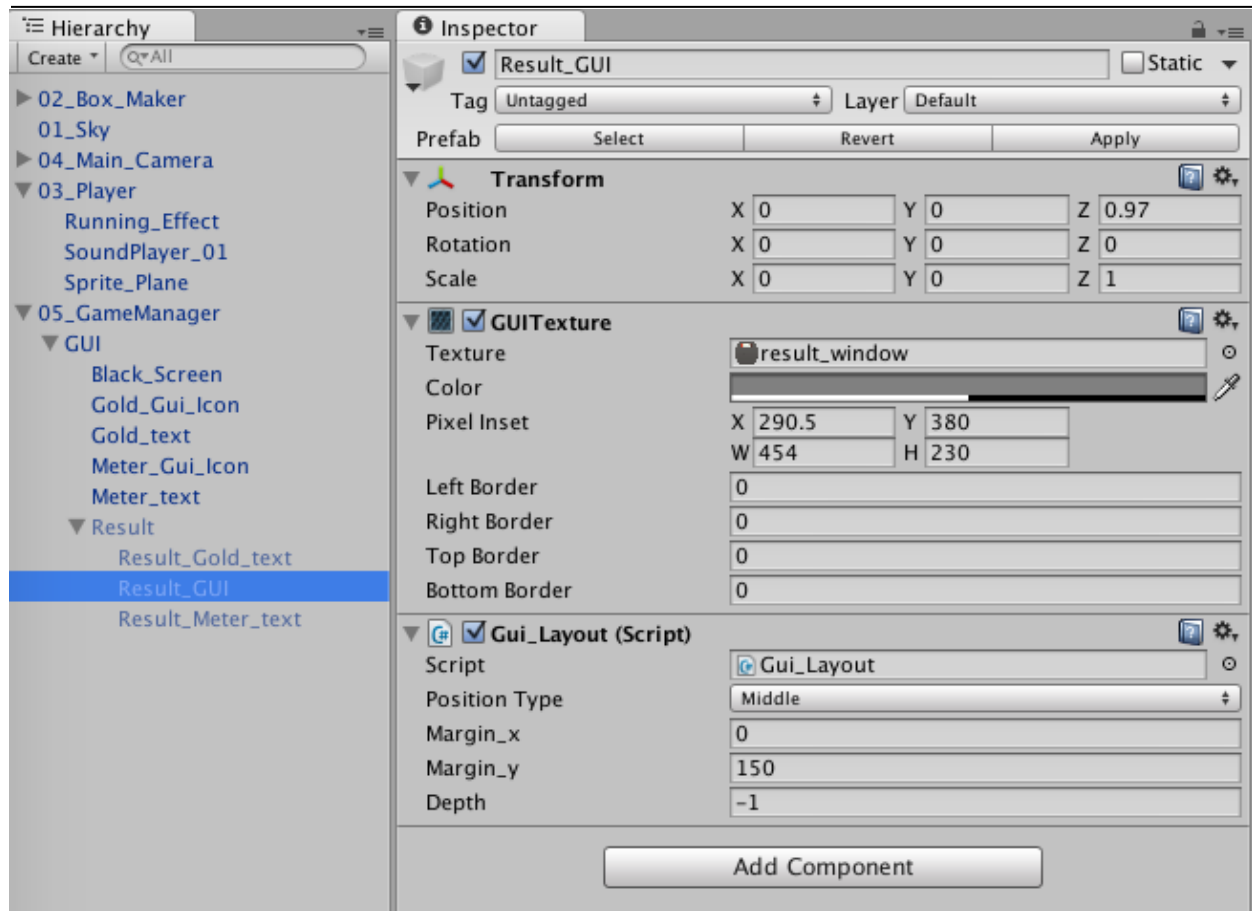


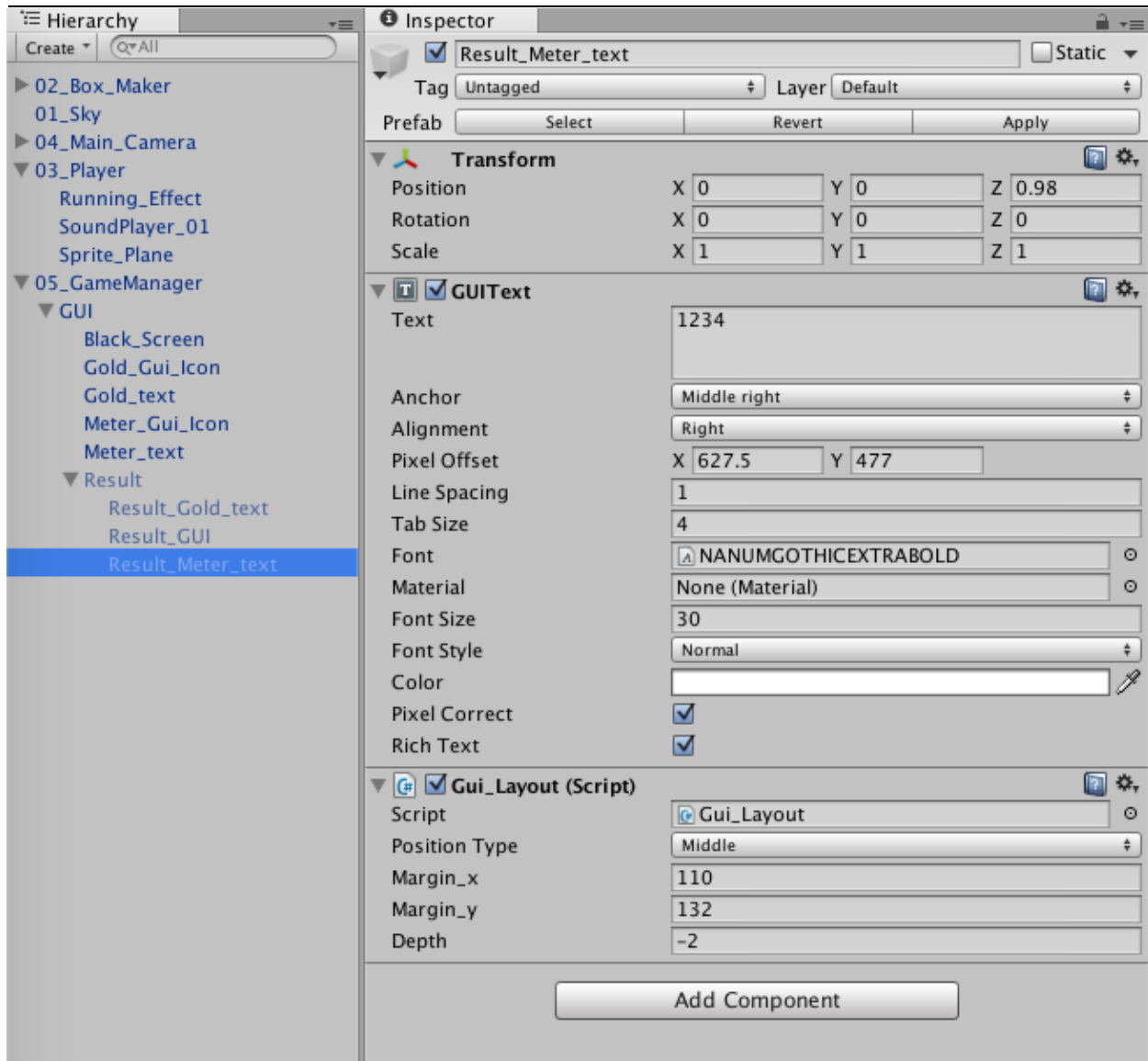












-Viết code cho Game\_Manager.cs

```

1  using UnityEngine;
2  using System.Collections;
3
4
5  //Khai báo trạng thái Game
6  public enum GameState
7  {
8      Play,
9      Pause,
10     End,
11 }
12
13 public class Game_Manager : MonoBehaviour
14 {
15     /**
16     *
17     * @author hungnq
18     */
19     //Khai báo các biến cần sử dụng
20     public GameState GS;
21     public int GameLv;
22     public float GameSpeed;
23     public Box_Loop _BL;
24     public Scroll_Mapping _SM;
25
26
27
28     public float Meter;
29     public int GetMoney = 0;
30
31
32
33
34     public GUIText Gold_Label;
35     public GUIText Meter_Label;
36     public GUITexture Black_screen;
37     public GUIText result_Gold_Label;
38     public GUIText result_Meter_Label;
39     GUIStyle guiRectStyle;
40     public Fade _fade;
41     public Texture Pause_btn;
42     public Texture Go_btn;
43     public Texture Replay_btn;
44     public Texture Main_btn;
45     public GameObject result_window;
46     float screenX;
47     float screenY;

```

```

48
49
50
51 void Start ()
52 {
53     GameObject a = GameObject.Find("02_Box_Maker");
54     if(a!=null)_BL = a.GetComponent<Box_Loop>();
55     GameObject b = GameObject.Find("01_Sky");
56     if(b!=null)_SM = b.GetComponent<Scroll_Mapping>();
57
58     GameSpeed = _BL.Speed;
59     SCREENSETTING ();
60
61
62 }
63
64 void Update ()
65 {
66     if (GS == GameState.Play) {
67         METERUPDATE ();
68     }
69
70 }
71 //Định nghĩa hàm thiết lập kích thước màn hình
72 void SCREENSETTING ()
73 {
74     screenX = Screen.width;
75     screenY = Screen.height;
76     guiRectStyle = new GUIStyle ();
77     guiRectStyle.border = new RectOffset (0, 0, 0, 0);
78     _fade.FadeIn ();
79 }
80 //Định nghĩa hàm cho người dùng tương tác qua giao diện
81 void OnGUI ()
82 {
83
84
85     if (GS == GameState.Play) {
86
87         if (GUI.Button (new Rect (20, 20, Pause_btn.width, Pause_btn.height), Pause_btn, guiRectStyle)) {
88             Black_screen.color = new Color (0, 0, 0, 0.4f);
89             GS = GameState.Pause;
90             Time.timeScale = 0;
91         }
92     }
93
94

```

```

95
96
97     if (GS == GameState.Pause) {
98
99
100         if (GUI.Button (new Rect (screenX * 0.5f - Go_btn.width * 0.5f, screenY * 0.5f
101             + Replay_btn.height * 0.5f + 10f, Go_btn.width, Go_btn.height), Go_btn, guiRectStyle)) {
102             Black_screen.color = new Color (0, 0, 0, 0);
103             Time.timeScale = 1;
104             GS = GameState.Play;
105
106         }
107
108         if (GUI.Button (new Rect (screenX * 0.5f - Replay_btn.width * 0.5f, screenY * 0.5f
109             - Replay_btn.height * 0.5f, Replay_btn.width, Replay_btn.height), Replay_btn, guiRectStyle)) {
110             Time.timeScale = 1;
111             Application.LoadLevel ("[PlayScene]");
112         }
113
114         if (GUI.Button (new Rect (screenX * 0.5f - Main_btn.width * 0.5f, screenY * 0.5f
115             - Replay_btn.height * 0.5f - Main_btn.height - 10f, Main_btn.width, Main_btn.height),
116             Main_btn, guiRectStyle)) {
117             Time.timeScale = 1;
118             Application.LoadLevel ("[IntroScene]");
119         }
120     }
121
122
123
124     if (GS == GameState.End) {
125
126
127
128
129         if (GUI.Button (new Rect (screenX * 0.5f - Replay_btn.width * 0.5f, screenY * 0.5f + 10f,
130             Replay_btn.width, Replay_btn.height), Replay_btn, guiRectStyle)) {
131             Application.LoadLevel ("[PlayScene]");
132         }
133
134         if (GUI.Button (new Rect (screenX * 0.5f - Main_btn.width * 0.5f, screenY * 0.5f
135             + Replay_btn.height + 20f, Main_btn.width, Main_btn.height), Main_btn, guiRectStyle)) {
136             //Application.LoadLevel ("[IntroScene]");
137         }
138     }
139 }
  
```

```

140 //Định nghĩa hàm GameOver
141 public void GAMEOVER ()
142 {
143
144
145     GS = GameState.End;
146     _fade.FadeOut ();
147     result_window.gameObject.SetActive (true);
148     result_Gold_Label.text = string.Format ("{0:N0}", GetMoney);
149     result_Meter_Label.text = string.Format ("{0:N0}", Meter);
150 }
151 //Định nghĩa hàm ăn Coin
152 public void GETCOIN ()
153 {
154     GetMoney += 1;
155     Gold_Label.text = string.Format ("{0:N0}", GetMoney);
156 }
157 //Hàm quản lý level
158 public void METERUPDATE ()
159 {
160     Meter += Time.deltaTime * GameSpeed;
161     Meter_Label.text = string.Format ("{0:N0}<color=#ff3366> m</color>", Meter);
162
163
164
165     if (Meter >= 50 && GameLv == 1) {
166         GameLevelUp ();
167     }
168
169     if (Meter >= 100 && GameLv == 2) {
170         GameLevelUp ();
171     }
172
173     if (Meter >= 150 && GameLv == 3) {
174         GameLevelUp ();
175     }
176
177     if (Meter >= 200 && GameLv == 4) {
178         GameLevelUp ();
179     }
180
181     if (Meter >= 250 && GameLv == 5) {
182         GameLevelUp ();
183     }
184
185     if (Meter >= 300 && GameLv == 6) {
186         GameLevelUp ();
187     }
188 }
189 //Định nghĩa hàm cập nhật level
190 public void GameLevelUp ()
191 {
192     GameLv += 1;
193     GameSpeed += 3;
194     _SM.ScrollSpeed += 0.1f;
195     _BL.Speed = GameSpeed;
196 }
197 }

```

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

## Bài 3 – Dự án Game Lab: Môi trường hoạt động của nhân vật (Environment) có thể di chuyển

### Hướng dẫn

- Download bài 3 của Lab6 và làm tiếp (Project Lab), sau đó nén lại thành project và nộp lên LMS
- Sinh viên viết đoạn code sau vào file EnvironmentMoving.cs

```

1  using UnityEngine;
2  using System.Collections;
3
4  public class EnviromentMoving : MonoBehaviour {
5      /**
6       *
7       * @author hungnq
8       */
9      public GameObject Player;
10     float distance = 0;
11     // Use this for initialization
12     void Start () {
13         if (Player != null)
14         {
15             distance = Player.transform.position.x - this.transform.position.x;
16         }
17     }
18
19     // Update is called once per frame
20     void Update () {
21         if (Player != null)
22         {
23             Vector3 pos = this.transform.position;
24             pos.x = Player.transform.position.x - distance;
25             this.transform.position = pos;
26         }
27     }
28 }
29

```



**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>\_Lab7, 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
	-	
	-	
	-	