

شرح مبسط للكود الخاص بالبرنامج التفاعلي:

أولاً: كود الخاص بتحريك الكامرة:

أوامر أولية خاصة باليونتي لا تغير أي شي هنا

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
```

أوامر تعريف العناصر.

الاسمي المحددة بالأحمر يمكن تغييرها ولاكن يجب تغييرها في كل مكان يتم ذكر الاسم داخل الكود.

الاسم المحدد بالأصفر يجب ان يكون نفس اسم ملف الكود المنشأ في اليونتي.

```
6 public class CamRot : MonoBehaviour
7 {
8     [SerializeField] private Camera cam;
9     [SerializeField] private Transform target;
10    public Transform RARound; // a gameobject
11    public float RotationSpeed = 200f;
12    public float scrollSpeed = 2f;
13    public float PanSensitivity = 5f;
14    Vector3 lastPanPosition;
15
```

الكامرة

العنصر الكامرة موجهها نحوه "المنتج"

العنصر الخاص بتدوير الكامرة

كود خاص بالسوتج او التوقل Toggle لا يوجد تغير هنا

```
16 // for toggling
17 public GameObject toggle_RCR_Cam;
18 public GameObject toggle_RCR_Product;
19 public GameObject toggle_RCR_Env;
20 public GameObject [] LightingRotating;
21
22
23 private void Start()
24 {
25
26    toggle_RCR_Cam.GetComponent<Toggle>().isOn = true;
27    toggle_RCR_Product.GetComponent<Toggle>().isOn = false;
28    toggle_RCR_Env.GetComponent<Toggle>().isOn = false;
29
30 }
31
```

السوتج او العاكس من واجهة المستخدم الخاص بالكامرة

السوتج او العاكس من واجهة المستخدم الخاص المنتج

السوتج او العاكس من واجهة المستخدم الخاص بالمحيط

مجموعة الخاصة بالإضاءات

الكود الخاص بزر الفأرة الأيسر لتدوير العناصر لا يوجد تغيير هنا

```

32 void Update()
33 {
34
35     if (Input.GetMouseButton(0))
36     {
37         if (toggle_RCR_Cam.GetComponent<Toggle>().isOn == true)
38         {
39             RAround.transform.RotateAround(target.transform.position, Vector3.up, ((Input.GetAxisRaw("Mouse X") * Time.deltaTime) * RotationSpeed));
40             RAround.transform.RotateAround(target.transform.position, transform.right, -((Input.GetAxisRaw("Mouse Y") * Time.deltaTime) * RotationSpeed));
41         }
42
43         if (toggle_RCR_Product.GetComponent<Toggle>().isOn == true)
44         {
45             GameObject Product = GameObject.Find("Product");
46             Product.transform.Rotate(new Vector3(0, Input.GetAxis("Mouse X") * -1*10, 0), Space.World);
47         }
48
49         if (toggle_RCR_Env.GetComponent<Toggle>().isOn == true)
50         {
51             foreach (GameObject x in LightingRotating)
52             {
53                 x.transform.Rotate(new Vector3(0, Input.GetAxis("Mouse X") * -1 * 10, 0), Space.World);
54             }
55         }
56     }
57 }
58

```

الكود الخاص بعجلة الفأرة للتكبير منظور الكامرة لا يوجد تغيير هنا

```

59
60 // for zooming
61 // see if mouse wheel is used
62 if (Input.GetAxis("Mouse ScrollWheel") != 0)
63 {
64     // you need add max and min
65     float ScrollWheelChange = Input.GetAxis("Mouse ScrollWheel");
66     cam.transform.position += cam.transform.forward * ScrollWheelChange * scrollSpeed;
67 }
68
69

```

الكود الخاص بزر الفأرة الأيمن لتحريك منظور الكامرة عامودياً وافقياً لا يوجد تغيير هنا

```

70
71 if (Input.GetMouseButtonDown(1))
72 {
73     lastPanPosition = Input.mousePosition;
74 }
75 else if (Input.GetMouseButton(1))
76 {
77     PanCamera(Input.mousePosition);
78 }
79
80 }
81
82
83 void PanCamera(Vector3 newPanPosition)
84 {
85     // Determine how much to move the camera
86     Vector3 offset = cam.ScreenToViewportPoint(lastPanPosition - newPanPosition);
87     Vector3 move = new Vector3(offset.x * PanSensitivity, offset.y * PanSensitivity, 0);
88
89     // Perform the movement
90     cam.transform.Translate(move, Space.Self);
91
92     // Cache the position
93     lastPanPosition = newPanPosition;
94 }
95
96
97 }
98

```

ثانياً: كود الخاص بالأزرار:

أوامر أولية خاصة باليونتي لا تغير أي شي هنا

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
```

تعريف العناصر

الاسم المحدد بالأصفر يجب ان يكون نفس اسم ملف الكود المنشأ في اليونتي.

```
7 public class ButtonCode : MonoBehaviour
8 {
9
10
11     [SerializeField] private Camera cam;
12     public Transform RAround;
13     public Transform target;
14
15     // For all intial values
16     private Vector3 initPosition;
17     private Quaternion initRotationCAM;
18     private Quaternion initRotationProduct;
19     private Quaternion initRotationLighting;
20
21     // DToggle
22     private GameObject[] DT;
23     private GameObject[] toggle;
24
25     // List of object share similar material
26     public GameObject[] Gm1;
27     public GameObject[] Gm2;
28
29     // Materials to be used for changing
30     public Material M1;
31     public Material M2;
32     public Material M3;
33     public Material M4;
34
35     // For object hiding showing
36     public GameObject obj1;
37     public GameObject obj2;
38
39     // For environment changing
40     public GameObject[] EnvSets;
41     public GameObject[] lights;
42     public Color CamBG01Color;
43     public Color CamBG02Color;
```

الكamera

العنصر الخاص بتدوير الكamera

العنصر الكamera موجهها نحوه "المنتج"

عناصر خاصة بحفظ قيم أولية

مجموعة لحفظ عناصر ذات رمز DToggle

مجموعة لحفظ عناصر ذات رمز Toggle

مجموعة لوضع العناصر التي تشارك الخامة الأولى

مجموعة لوضع العناصر التي تشارك الخامة الثانية

مجموعة لوضع الخامة الأولى

مجموعة لوضع الخامة الثانية

مجموعة لوضع الخامة الثالثة

مجموعة لوضع الخامة الرابعة

العنصر الأول لإظهاره

العنصر الثاني لإظهاره

مجموعة لوضع عناصر المحيط

مجموعة لوضع عناصر الإضاءة

لاختار اللون الأول لخلفية الكamera

لاختار اللون الثاني لخلفية الكamera

```

44
45 // For animation play stop
46 public Animator AnimationCont;
47
48 public GameObject ExpToggle;
49 public GameObject AnimToggle;
50
51 public Color imageColor;
52 public GameObject animPanal;
53 public GameObject animPlay;
54 public GameObject expPanal;
55 public GameObject expPlay;
56
57

```

وضع العنصر الخاص بمتحكم الانيميشن
 وضع العنصر الذي يحتوي على السوج الخاص بالحركة المفككة
 وضع العنصر الذي يحتوي على السوج الخاص بحركة الأجزاء
 اختيار لون لتغطية شكل زر التحكم بالانيميشن
 شكل خلفية واجهة المستخدم لزر الانيميشن لتحريك الأجزاء
 شكل واجهة المستخدم لزر الانيميشن لتحريك الاجزاء
 شكل خلفية واجهة المستخدم لزر الانيميشن لتحريك المفكك
 شكل واجهة المستخدم لزر الانيميشن لتحريك المفكك

الأوامر الخاصة عند بداية المشروع

```

58 // Start is called before the first frame update
59 void Start()
60 {
61     // showing the object to start with
62     obj1.SetActive(true);
63     obj2.SetActive(false);
64
65     // showing the environment to start with
66     EnvSets[0].SetActive(true);
67     lights[0].SetActive(true);
68
69     // get the intail rotation and position of cam position and RARound rotation
70     initPosition = cam.transform.position;
71     initRotationCAM = RARound.transform.rotation;
72     initRotationProduct = target.transform.rotation;
73     initRotationLighting = lights[0].transform.rotation;
74
75
76 }
77

```

اظهر العنصر الأول
 أخفى العنصر الثاني
 اظهار المحيط الأول
 اظهار الإضاءة الأولى
 الأوامر الخاصة بحفظ الأوضاع الأولية للكامرة والمنتج والإضاءة

```

78 // Update is called once per frame
79 void Update()
80 {
81
82 // For Animation play stop
83 if (AnimToggle.GetComponent<Toggle>().isOn == true)
84 {
85     AnimationCont.SetBool("AnimPart", true);
86     ExpToggle.GetComponent<Toggle>().interactable = false;
87     expPanal.GetComponent<Image>().color = imageColor;
88     expPlay.GetComponent<Image>().color = imageColor;
89 }
90 else
91 {
92     AnimationCont.SetBool("AnimPart", false);
93     ExpToggle.GetComponent<Toggle>().interactable = true;
94     expPanal.GetComponent<Image>().color = Color.white;
95     expPlay.GetComponent<Image>().color = Color.white;
96 }
97 if (ExpToggle.GetComponent<Toggle>().isOn == true)
98 {
99     AnimationCont.SetBool("ExpAnim", true);
100     AnimToggle.GetComponent<Toggle>().interactable = false;
101     animPanal.GetComponent<Image>().color = imageColor;
102     animPlay.GetComponent<Image>().color = imageColor;
103 }
104 else
105 {
106     AnimationCont.SetBool("ExpAnim", false);
107     AnimToggle.GetComponent<Toggle>().interactable = true;
108     animPanal.GetComponent<Image>().color = Color.white;
109     animPlay.GetComponent<Image>().color = Color.white;
110 }
111
112 // to get ESC key and exit the application
113 if (Input.GetKey("escape"))
114 {
115     Application.Quit();
116 }
117 }

```

مجموعة الأوامر الخاصة بتشغيل الانميشن الحركة

الامر الخاص بالضغط على زر الخروج ESC على لوحة المفاتيح

```

120 public void Material_1()
121 {
122     foreach (GameObject x in Gm1)
123     {
124         x.GetComponent<Renderer>().material = M1;
125     }
126 }
127
128 public void Material_2()
129 {
130     foreach (GameObject x in Gm1)
131     {
132         x.GetComponent<Renderer>().material = M2;
133     }
134 }
135
136 public void Color_01()
137 {
138     foreach (GameObject x in Gm2)
139     {
140         x.GetComponent<Renderer>().material = M3;
141     }
142 }
143
144 public void Color_02()
145 {
146     foreach (GameObject x in Gm2)
147     {
148         x.GetComponent<Renderer>().material = M4;
149     }
150 }

```

الامر الخاص بزر تغيير الخامة الأولى

الامر الخاص بزر تغيير الخامة الثانية

الامر الخاص بزر تغيير الخامة الثالثة

الامر الخاص بزر تغيير الخامة الرابعة

```

152 public void Object01()
153 {
154     obj1.SetActive(true);
155     obj2.SetActive(false);
156 }
157
158 public void Object02()
159 {
160     obj1.SetActive(false);
161     obj2.SetActive(true);
162 }

```

الامر الخاص بزر اظهار العنصر الأول

اظهر العنصر الأول

أخفى العنصر الثاني

الامر الخاص بزر اظهار العنصر الثاني

أخفى العنصر الأول

اظهر العنصر الثاني

```

164 public void EnvChange01()
165 {
166     EnvSets[0].SetActive(true);
167     EnvSets[1].SetActive(false);
168     lights[0].SetActive(true);
169     lights[1].SetActive(false);
170     cam.backgroundColor = CamBG01Color;
171     DynamicGI.UpdateEnvironment();
172 }
173
174 public void EnvChange02()
175 {
176     EnvSets[0].SetActive(false);
177     EnvSets[1].SetActive(true);
178     lights[0].SetActive(false);
179     lights[1].SetActive(true);
180     cam.backgroundColor = CamBG02Color;
181     DynamicGI.UpdateEnvironment();
182 }
183

```

الامر الخاص بإظهار المحيط الأول

اظهر المحيط الأول

أخفى المحيط الثاني

اظهر الاضاءة الأولى

أخفى الاضاءة الثانية

اجعل خلفية الكامرة باللون الأول

الامر الخاص بزر اظهار العنصر الثاني

أخفى المحيط الأول

اظهر المحيط الثاني

أخفى الاضاءة الأولى

اظهر الاضاءة الثانية

اجعل خلفية الكامرة باللون الثاني

```

185 public void ResetCAM()
186 {
187     // reset to initial rotation and position
188     RAround.transform.rotation = initRotationCAM;
189     cam.transform.position = initPosition;
190 }
191
192 public void ResetAll()
193 {
194     // reset cam
195     RAround.transform.rotation = initRotationCAM;
196     cam.transform.position = initPosition;
197
198     // reset product
199     target.transform.rotation = initRotationProduct;
200
201     // reset lighting
202     foreach (GameObject light in lights)
203     {
204         light.transform.rotation = initRotationLighting;
205     }
206
207     // reset all toggle
208     DT = GameObject.FindGameObjectsWithTag("DToggle");
209     foreach (GameObject obj in DT)
210     {
211         obj.GetComponent<Toggle>().isOn = true;
212     }
213
214     toggle = GameObject.FindGameObjectsWithTag("toggle");
215     foreach (GameObject obj in toggle)
216     {
217         obj.GetComponent<Toggle>().isOn = false;
218     }
219 }
220

```

الامر الخاص بإعادة وضع الكامرة

الامر الخاص بإعادة وضع الجميع

```

222 public void OpenURL()
223 {
224     Application.OpenURL("https://www.google.com/search?q=murphy+bed");
225 }
226
227 public void exit()
228 {
229     Application.Quit();
230 }
231
232

```

الامر الخاص بزر ارفاق وصلة لموقع

موقع المراد الذهاب اليه

الامر الخاص بزر الخروج