

Лабораторна робота №5

Тема: Створення порталу в в AR

Мета роботи: набути практичні навички з використання технології доповненої реальності при створенні ігор

Теоретичні відомості: лек.8

Завдання на лабораторну роботу

1. Завдання 1. Створіть застосунок-портал в AR.
 - 1.1. Клонуйте [Unity+ARFoundation](#) для прикладу echoAR.
 - 1.2. Проведіть налаштування дотримуючись інструкцій на [сторінці документації](#), щоб [встановити ключ API](#).
 - 1.3. [Додайте об'єкти 3D](#) та їх медалані з папки [Models](#) на консоль.
 - 1.4. Перезапишіть наявний сценарій `Assets/echoAR/CustomBehaviour.cs` новим файлом [CustomBehaviour.cs](#).
 - 1.5. Скопіюйте всю папку [Resources](#) в `Assets/echoAR/`.
 - 1.6. [Створіть і запустіть додаток AR](#).

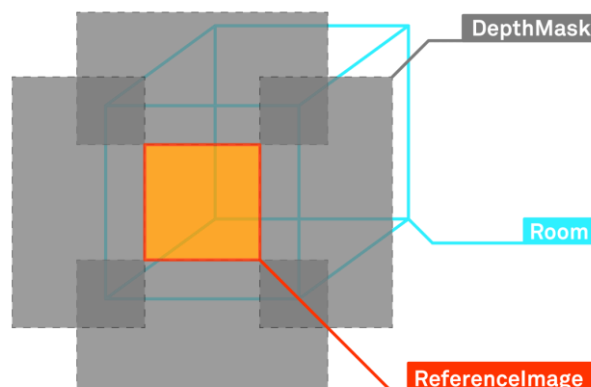
Скомпілювати два файли: під ОС Android 7.0 або вище та під ОС iOS 6.0 або вище.

2. Завдання 2. Додати до проекту ще одні двері та перехід в іншу реальність або виконати завдання 3.

3. Завдання 3 (додаткове).

Створіть другий AR-портал, що з'являється при дотику до екрану. Портал повинен мати правильну перспективу та ефект викривлення простору. Передбачити, що через портал користувач може увійти до уявного світу, а реальний світ уде ніби за межами порталу.

- 3.1. Включіть інсталяцію ARFoundation, ARKit (або ARCore) у менеджері пакетів, а також створіть камеру AR у сцені та переконайтеся, що ви знаєте, як використовувати функцію відстеження зображень.
- 3.2. Створіть ігровий об'єкт фізичного розміру «літак» (Поради: розмір літака за замовчуванням — 10 метрів) і назвіть його «ReferenceImage»;
- 3.3. Застосуйте матеріал із вашим опорним зображенням як текстуру для базового кольору на ReferenceImage;
- 3.4. Відповідно до ReferenceImage, створіть 4 ігрових об'єкта «площина» з назвою «DepthMask» і нанесіть на них матеріал, використовуючи «DepthMask.shader»;
- 3.5. Створіть свою кімнату та перемістіть її за DepthMask.



3.6. Шейдери

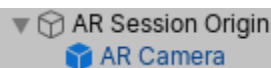
3.6.1. 1.DepthMask.shader

```

1 Shader "Custom/DepthMask" {
2
3     SubShader {
4         // Render the mask after regular geometry,
5         //but before masked geometry and
6         // transparent things.
7
8         Tags { "Queue" = "Geometry-10" }
9
10        // Don't draw in the RGBA channels; just the depth buffer
11
12        ColorMask 0
13        ZWrite On
14
15        // Do nothing specific in the pass:
16
17        Pass {}
18    }
19 }
```

3.7. Двері

3.7.1. AR камера



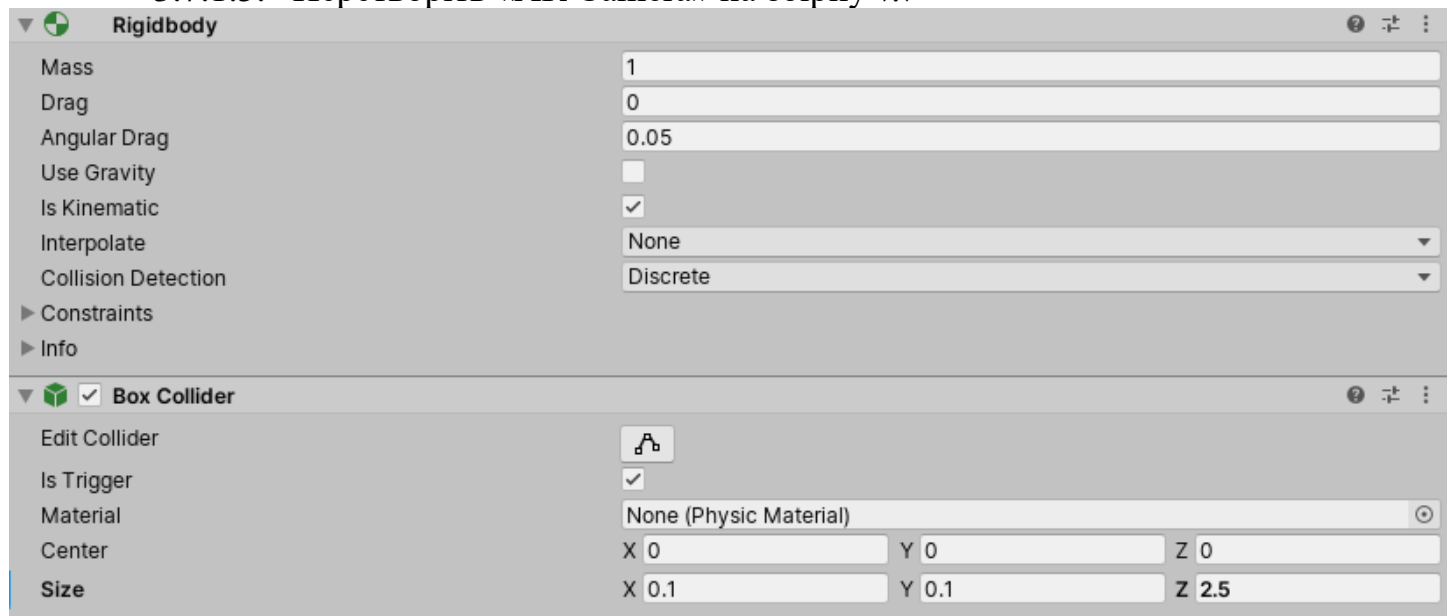
3.7.1.1. Додайте компоненти «Rigidbody» і «Box Collider» до камери AR.

3.7.1.2. Зніміть прапорець «Використовувати гравітацію» та поставте прапорець «Чи є кінематичним» у «Rigidbody».

3.7.1.3. Установіть прапорець «Чи є тригером» у «Box Collider».

3.7.1.4. Встановіть розмір «Box Collider» на x 0,1, y 0,1, z 2,5.

3.7.1.5. Перетворіть «AR Camera» на збірну ※



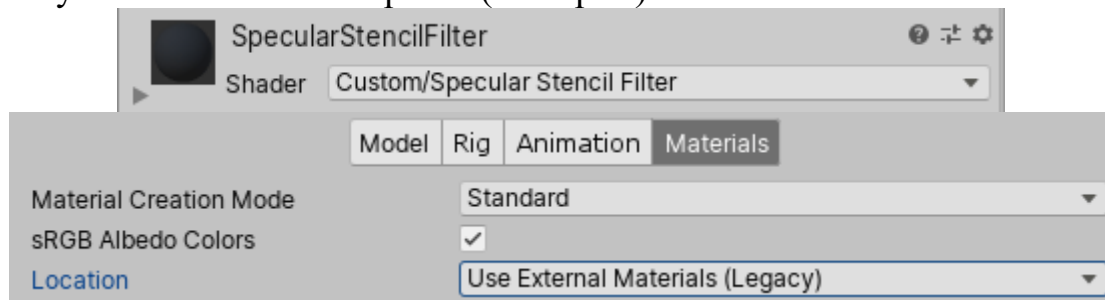
3.7.2. InnerWorld



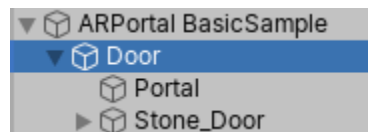
3.7.2.1. Переконайтеся, що матеріали, які використовуються в будь-якому ігровому об'єкті в InnerWorld, крім Sky або інших спеціальних об'єктів,

мають використовувати шейдер «SpecularStencilFilter.shader», який можна знайти нижче.

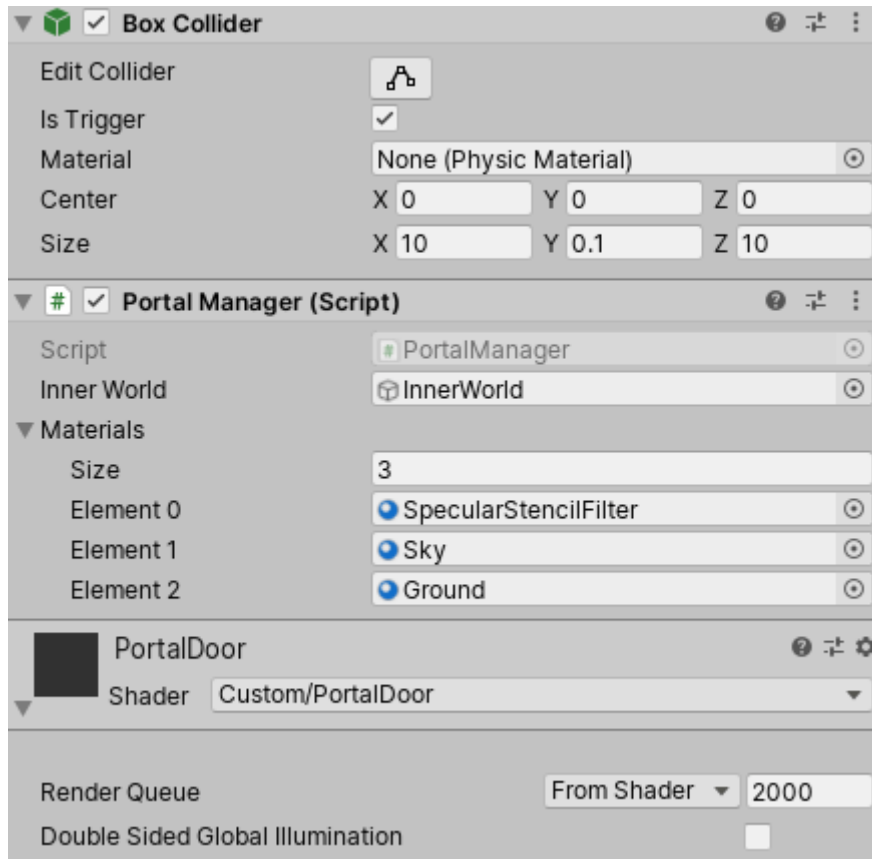
Примітка: Якщо вам потрібні окремі матеріали, просто перейдіть у налаштування імпортера об'єкта, перейдіть на вкладку матеріалів і скористайтеся розділом «Використовувати зовнішні матеріали (застарілі)».



3.7.3. Двері



- 3.7.3.1. Для каркаса дверей можна використовувати будь-який матеріал, крім матеріалів із використанням шейдерів «SpecularStencilFilter.shader» або «PortalDoor.shader», які ви можете знайти нижче.
- 3.7.3.2. Створіть ігровий об'єкт «Площина або квадроцикл» і перейменуйте його на «Портал».
- 3.7.3.3. Додайте компонент «Box Collider» на портал і відрегулюйте розмір.
- 3.7.3.4. Виберіть Portal>Box Collider>Is Trigger.
- 3.7.3.5. Додайте компонент «PortalManager», який можна знайти нижче.
- 3.7.3.6. Перетягніть «InnerWorld» у Portal>PortalManager>Inner World.
- 3.7.3.7. Налаштуйте Portal>PortalManager>Матеріали та перетягніть усі матеріали, які використовують у «InnerWorld».
- 3.7.3.8. Додайте матеріал за допомогою шейдера «PortalDoor.shader» на порталі.



3.7.4. Анімовані двері

У дверях за бажанням можна додати аніматора з анімацією: а кнопка на полотні використовується для керування дверима зі скриптом DoorController.cs для запуску анімації (не обов'язково).

3.7.5. SkySphere

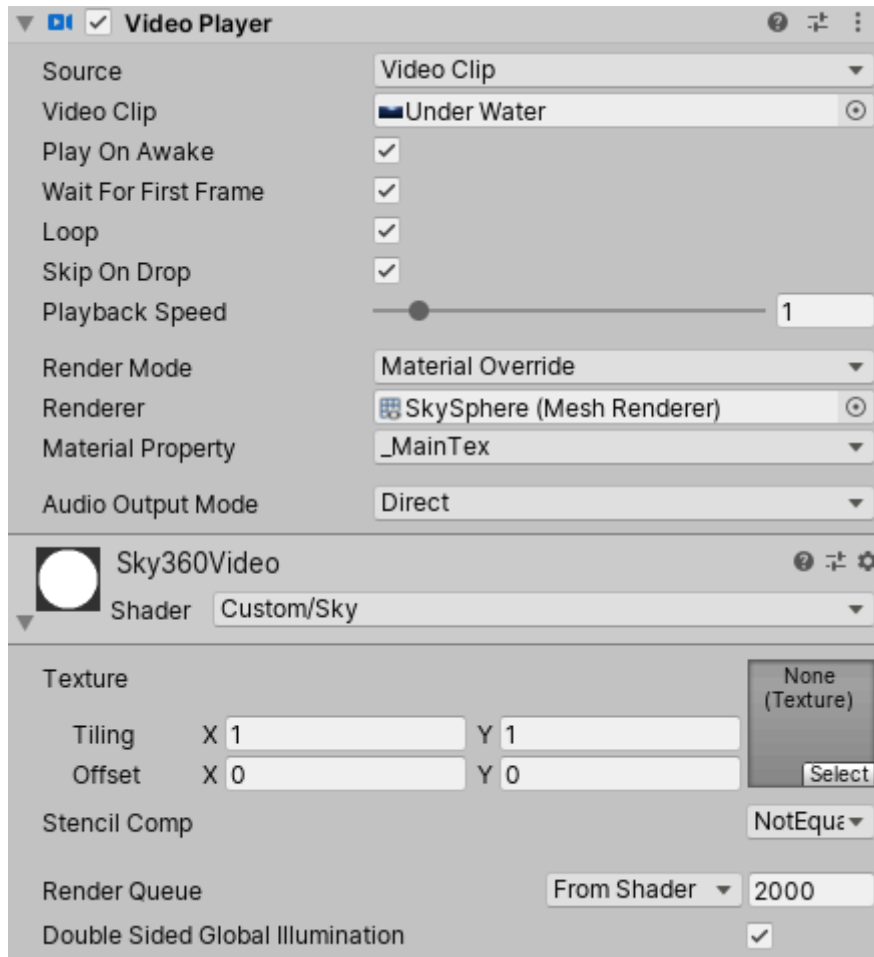
3.7.5.1. Створіть ігровий об'єкт «сфера» та перейменуйте «SkySphere»

3.7.5.2. Додайте компонент «Відеоплеєр» на SkySphere і перетягніть своє 360 відео (яке можна легко знайти у Vimeo).

3.7.5.3. SkySphere>Відеоплеєр>Відеокліп.

3.7.5.4. Додайте матеріал за допомогою шейдера «Custom/Sky» на SkySphere (Шейдер можна знайти нижче).

3.7.5.5. Переконайтеся, що трафаретний комп'ютер матеріалу вибрано «NotEqual»



3.8. Шейдери

3.8.1. StencilMask.shader

```

1 // (c) 2020 Tongzhou Yu
2
3 Shader "Custom/StencilMask"
4 {
5     SubShader
6     {
7         Tags { "RenderType" = "Opaque" }
8         LOD 100
9         Zwrite Off
10        ColorMask 0
11        Cull off
12
13        Pass
14        {
15            Stencil{
16                Ref 1
17                Comp always
18                Pass replace
19            }
20
21            // To further understand Stencil, check the link (EN):
22            // https://docs.unity3d.com/Manual/SL-Stencil.html
23            // or (CN)
24            // https://blog.csdn.net/linjf520/article/details/94867519
25
26            CGPROGRAM
27            #pragma vertex vert
28            #pragma fragment frag
29
30            #include "UnityCG.cginc"
31
32            struct appdata

```

```

33     {
34         float4 vertex : POSITION;
35     };
36
37     struct v2f
38     {
39         float4 vertex : SV_POSITION;
40     };
41
42     v2f vert(appdata v)
43     {
44         v2f o;
45         o.vertex = UnityObjectToClipPos(v.vertex);
46         return o;
47     }
48
49     fixed4 frag(v2f i) : SV_Target
50     {
51         return fixed4(0.0, 0.0, 0.0, 0.0);
52     }
53     ENDCG
54 }
55 }
56 }

```

3.8.2. StencilSpecular.shader, модифікація на основі вбудованих шейдерів Unity

```

1 // (c) 2020 Tongzhou Yu
2
3 Shader "Custom/StencilSpecular"
4 {
5     Properties
6     {
7         [Enum(CompareFunction)] _StencilComp("Stencil Comp", Int) = 3
8         _Color("Color", Color) = (1,1,1,1)
9         _MainTex("Albedo", 2D) = "white" {}
10
11         _Cutoff("Alpha Cutoff", Range(0.0, 1.0)) = 0.5
12
13         _Glossiness("Smoothness", Range(0.0, 1.0)) = 0.5
14         _GlossMapScale("Smoothness Factor", Range(0.0, 1.0)) = 1.0
15         [Enum(Specular Alpha,0,Albedo Alpha,1)] _SmoothnessTextureChannel("Smoothness texture channel", Float) = 0
16
17         _SpecColor("Specular", Color) = (0.2,0.2,0.2)
18         _SpecGlossMap("Specular", 2D) = "white" {}
19         [ToggleOff] _SpecularHighlights("Specular Highlights", Float) = 1.0
20         [ToggleOff] _GlossyReflections("Glossy Reflections", Float) = 1.0
21
22         _BumpScale("Scale", Float) = 1.0
23         [Normal] _BumpMap("Normal Map", 2D) = "bump" {}
24
25         _Parallax("Height Scale", Range(0.005, 0.08)) = 0.02
26         _ParallaxMap("Height Map", 2D) = "black" {}
27
28         _OcclusionStrength("Strength", Range(0.0, 1.0)) = 1.0
29         _OcclusionMap("Occlusion", 2D) = "white" {}
30
31         _EmissionColor("Color", Color) = (0,0,0)
32         _EmissionMap("Emission", 2D) = "white" {}
33
34         _DetailMask("Detail Mask", 2D) = "white" {}
35
36         _DetailAlbedoMap("Detail Albedo x2", 2D) = "grey" {}
37         _DetailNormalMapScale("Scale", Float) = 1.0
38         [Normal] _DetailNormalMap("Normal Map", 2D) = "bump" {}
39

```

```

40 [Enum(UV0,0,UV1,1)] _UVSec("UV Set for secondary textures", Float) = 0
41
42 // Blending state
43 [HideInInspector] _Mode("__mode", Float) = 0.0
44 [HideInInspector] _SrcBlend("__src", Float) = 1.0
45 [HideInInspector] _DstBlend("__dst", Float) = 0.0
46 [HideInInspector] _ZWrite("__zw", Float) = 1.0
47 }
48
49 CGINCLUDE
50 #define UNITY_SETUP_BRDF_INPUT SpecularSetup
51     ENDCG
52
53     SubShader
54     {
55         Tags { "RenderType" = "Opaque" "PerformanceChecks" = "False" }
56         LOD 300
57
58         Stencil{
59             Ref 1
60             Comp[_StencilComp]
61         }
62
63         // -----
64         // Base forward pass (directional light, emission, lightmaps, ...)
65         Pass
66         {
67             Name "FORWARD"
68             Tags { "LightMode" = "ForwardBase" }
69
70             Blend[_SrcBlend][_DstBlend]
71             ZWrite[_ZWrite]
72
73             CGPROGRAM
74             #pragma target 3.0
75
76             // -----
77
78             #pragma shader_feature _NORMALMAP
79             #pragma shader_feature_local _ _ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
80             #pragma shader_feature _EMISSION
81             #pragma shader_feature_local _SPECGLOSSMAP
82             #pragma shader_feature_local _DETAIL_MULX2
83             #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
84             #pragma shader_feature_local _SPECULARHIGHLIGHTS_OFF
85             #pragma shader_feature_local _GLOSSYREFLECTIONS_OFF
86             #pragma shader_feature_local _PARALLAXMAP
87
88             #pragma multi_compile_fwdbase
89             #pragma multi_compile_fog
90             #pragma multi_compile_instancing
91             // Uncomment the following line to enable dithering LOD crossfade. Note: there are more in the file to uncomment for other
92             // #pragma multi_compile _ LOD_FADE_CROSSFADE
93
94             #pragma vertex vertBase
95             #pragma fragment fragBase
96             #include "UnityStandardCoreForward.cginc"
97
98         ENDCG
99     }
100     // -----
101     // Additive forward pass (one light per pass)
102     Pass
103     {
104         Name "FORWARD_DELTA"
105         Tags { "LightMode" = "ForwardAdd" }

```

```

106     Blend[_SrcBlend] One
107     Fog { Color(0,0,0,0) } // in additive pass fog should be black
108     ZWrite Off
109     ZTest LEqual
110
111     CGPROGRAM
112     #pragma target 3.0
113
114     // -----
115
116     #pragma shader_feature _NORMALMAP
117     #pragma shader_feature_local _ _ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
118     #pragma shader_feature_local _SPECGLOSSMAP
119     #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
120     #pragma shader_feature_local _SPECULARHIGHLIGHTS_OFF
121     #pragma shader_feature_local _DETAIL_MULX2
122     #pragma shader_feature_local _PARALLAXMAP
123
124     #pragma multi_compile_fwdadd_fullshadows
125     #pragma multi_compile_fog
126     // Uncomment the following line to enable dithering LOD crossfade. Note: there are more in the file to uncomment for other
127     // #pragma multi_compile _ LOD_FADE_CROSSFADE
128
129     #pragma vertex vertAdd
130     #pragma fragment fragAdd
131     #include "UnityStandardCoreForward.cginc"
132
133     ENDCG
134 }
135 // -----
136 // Shadow rendering pass
137 Pass {
138     Name "ShadowCaster"
139     Tags { "LightMode" = "ShadowCaster" }
140
141     ZWrite On ZTest LEqual
142
143     CGPROGRAM
144     #pragma target 3.0
145
146     // -----
147
148
149     #pragma shader_feature_local _ _ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
150     #pragma shader_feature_local _SPECGLOSSMAP
151     #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
152     #pragma shader_feature_local _PARALLAXMAP
153     #pragma multi_compile_shadowcaster
154     #pragma multi_compile_instancing
155     // Uncomment the following line to enable dithering LOD crossfade. Note: there are more in the file to uncomment for other
156     // #pragma multi_compile _ LOD_FADE_CROSSFADE
157
158     #pragma vertex vertShadowCaster
159     #pragma fragment fragShadowCaster
160
161     #include "UnityStandardShadow.cginc"
162
163     ENDCG
164 }
165 // -----
166 // Deferred pass
167 Pass
168 {
169     Name "DEFERRED"
170     Tags { "LightMode" = "Deferred" }
171

```



```

172     CGPROGRAM
173     #pragma target 3.0
174     #pragma exclude_renderers nomrt
175
176
177     // -----
178
179     #pragma shader_feature _NORMALMAP
180     #pragma shader_feature_local _ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
181     #pragma shader_feature _EMISSION
182     #pragma shader_feature_local _SPECGLOSSMAP
183     #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
184     #pragma shader_feature_local _SPECULARHIGHLIGHTS_OFF
185     #pragma shader_feature_local _DETAIL_MULX2
186     #pragma shader_feature_local _PARALLAXMAP
187
188     #pragma multi_compile_prepassfinal
189     #pragma multi_compile_instancing
190     // Uncomment the following line to enable dithering LOD crossfade. Note: there are more in the file to uncomment for other
191     // #pragma multi_compile _ LOD_FADE_CROSSFADE
192
193     #pragma vertex vertDeferred
194     #pragma fragment fragDeferred
195
196     #include "UnityStandardCore.cginc"
197
198     ENDCG
199 }
200
201     // -----
202     // Extracts information for lightmapping, GI (emission, albedo, ...)
203     // This pass it not used during regular rendering.
204     Pass
205     {
206         Name "META"
207         Tags { "LightMode" = "Meta" }
208
209         Cull Off
210
211         CGPROGRAM
212         #pragma vertex vert_meta
213         #pragma fragment frag_meta
214
215         #pragma shader_feature _EMISSION
216         #pragma shader_feature_local _SPECGLOSSMAP
217         #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
218         #pragma shader_feature_local _DETAIL_MULX2
219         #pragma shader_feature EDITOR_VISUALIZATION
220
221         #include "UnityStandardMeta.cginc"
222         ENDCG
223     }
224 }
225
226     SubShader
227     {
228         Tags { "RenderType" = "Opaque" "PerformanceChecks" = "False" }
229         LOD 150
230
231         Stencil{
232             Ref 1
233             Comp[_StencilComp]
234         }
235     // -----
236     // Base forward pass (directional light, emission, lightmaps, ...)
237     Pass

```

```

238 {
239     Name "FORWARD"
240     Tags { "LightMode" = "ForwardBase" }
241
242     Blend[_SrcBlend][_DstBlend]
243     ZWrite[_ZWrite]
244
245     CGPROGRAM
246     #pragma target 2.0
247
248     #pragma shader_feature _NORMALMAP
249     #pragma shader_feature_local _ _ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
250     #pragma shader_feature _EMISSION
251     #pragma shader_feature_local _SPECGLOSSMAP
252     #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
253     #pragma shader_feature_local _SPECULARHIGHLIGHTS_OFF
254     #pragma shader_feature_local _GLOSSYREFLECTIONS_OFF
255     #pragma shader_feature_local _DETAIL_MULX2
256     // SM2.0: NOT SUPPORTED shader_feature_local _PARALLAXMAP
257
258     #pragma skip_variants SHADOWS_SOFT DYNAMICLIGHTMAP_ON DIRLIGHTMAP_COMBINED
259
260     #pragma multi_compile_fwdbase
261     #pragma multi_compile_fog
262
263     #pragma vertex vertBase
264     #pragma fragment fragBase
265     #include "UnityStandardCoreForward.cginc"
266
267     ENDCG
268 }
269 // -----
270 // Additive forward pass (one light per pass)
271 Pass
272 {
273     Name "FORWARD_DELTA"
274     Tags { "LightMode" = "ForwardAdd" }
275     Blend[_SrcBlend] One
276     Fog { Color(0,0,0,0) } // in additive pass fog should be black
277     ZWrite Off
278     ZTest LEqual
279
280     CGPROGRAM
281     #pragma target 2.0
282
283     #pragma shader_feature _NORMALMAP
284     #pragma shader_feature_local _ _ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
285     #pragma shader_feature_local _SPECGLOSSMAP
286     #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
287     #pragma shader_feature_local _SPECULARHIGHLIGHTS_OFF
288     #pragma shader_feature_local _DETAIL_MULX2
289     // SM2.0: NOT SUPPORTED shader_feature_local _PARALLAXMAP
290     #pragma skip_variants SHADOWS_SOFT
291
292     #pragma multi_compile_fwdadd_fullshadows
293     #pragma multi_compile_fog
294
295     #pragma vertex vertAdd
296     #pragma fragment fragAdd
297     #include "UnityStandardCoreForward.cginc"
298
299     ENDCG
300 }
301 // -----
302 // Shadow rendering pass
303 Pass {

```

```

304     Name "ShadowCaster"
305     Tags { "LightMode" = "ShadowCaster" }
306
307     ZWrite On ZTest LEqual
308
309     CGPROGRAM
310     #pragma target 2.0
311
312     #pragma shader_feature_local _ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
313     #pragma shader_feature_local _SPECGLOSSMAP
314     #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
315     #pragma skip_variants SHADOWS_SOFT
316     #pragma multi_compile_shadowcaster
317
318     #pragma vertex vertShadowCaster
319     #pragma fragment fragShadowCaster
320
321     #include "UnityStandardShadow.cginc"
322
323     ENDCG
324 }
325 // -----
326 // Extracts information for lightmapping, GI (emission, albedo, ...)
327 // This pass it not used during regular rendering.
328 Pass
329 {
330     Name "META"
331     Tags { "LightMode" = "Meta" }
332
333     Cull Off
334
335     CGPROGRAM
336     #pragma vertex vert_meta
337     #pragma fragment frag_meta
338
339     #pragma shader_feature _EMISSION
340     #pragma shader_feature_local _SPECGLOSSMAP
341     #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
342     #pragma shader_feature_local _DETAIL_MULX2
343     #pragma shader_feature EDITOR_VISUALIZATION
344
345     #include "UnityStandardMeta.cginc"
346     ENDCG
347 }
348 }
349
350 // FallBack "VertexLit"
351 //CustomEditor "StandardShaderGUI"
352 }

```

3.8.3. StencilMetallic.shader, модифікація на основі вбудованих шейдерів

Unity

```

1 // (c) 2020 Tongzhou Yu
2
3 Shader "Custom/StencilMetallic"
4 {
5     Properties
6     {
7         [Enum(CompareFunction)] _StencilComp("Stencil Comp", Int) = 3
8
9
10    _Color("Color", Color) = (1,1,1,1)
11    _MainTex("Albedo", 2D) = "white" {}
12
13    _Cutoff("Alpha Cutoff", Range(0.0, 1.0)) = 0.5
14

```

```

15     _Glossiness("Smoothness", Range(0.0, 1.0)) = 0.5
16     _GlossMapScale("Smoothness Factor", Range(0.0, 1.0)) = 1.0
17     [Enum(Metallic Alpha,0,Albedo Alpha,1)] _SmoothnessTextureChannel("Smoothness texture channel", Float) = 0
18
19     [Gamma] _Metallic("Metallic", Range(0.0, 1.0)) = 0.0
20     _MetallicGlossMap("Metallic", 2D) = "white" {}
21
22     [ToggleOff] _SpecularHighlights("Specular Highlights", Float) = 1.0
23     [ToggleOff] _GlossyReflections("Glossy Reflections", Float) = 1.0
24
25     _BumpScale("Scale", Float) = 1.0
26     [Normal] _BumpMap("Normal Map", 2D) = "bump" {}
27
28     _Parallax("Height Scale", Range(0.005, 0.08)) = 0.02
29     _ParallaxMap("Height Map", 2D) = "black" {}
30
31     _OcclusionStrength("Strength", Range(0.0, 1.0)) = 1.0
32     _OcclusionMap("Occlusion", 2D) = "white" {}
33
34     _EmissionColor("Color", Color) = (0,0,0)
35     _EmissionMap("Emission", 2D) = "white" {}
36
37     _DetailMask("Detail Mask", 2D) = "white" {}
38
39     _DetailAlbedoMap("Detail Albedo x2", 2D) = "grey" {}
40     _DetailNormalMapScale("Scale", Float) = 1.0
41     [Normal] _DetailNormalMap("Normal Map", 2D) = "bump" {}
42
43     [Enum(UV0,0,UV1,1)] _UVSec("UV Set for secondary textures", Float) = 0
44
45     // Blending state
46     [HideInInspector] _Mode("__mode", Float) = 0.0
47     [HideInInspector] _SrcBlend("__src", Float) = 1.0
48     [HideInInspector] _DstBlend("__dst", Float) = 0.0
49     [HideInInspector] _ZWrite("__zw", Float) = 1.0
50 }
51
52 CGINCLUDE
53 #define UNITY_SETUP_BRDF_INPUT MetallicSetup
54     ENDCG
55
56     SubShader
57     {
58         Tags { "RenderType" = "Opaque" "PerformanceChecks" = "False" }
59         LOD 300
60
61         Stencil{
62             Ref 1
63             Comp[_StencilComp]
64         }
65
66         // -----
67         // Base forward pass (directional light, emission, lightmaps, ...)
68         Pass
69         {
70             Name "FORWARD"
71             Tags { "LightMode" = "ForwardBase" }
72
73             Blend[_SrcBlend][_DstBlend]
74             ZWrite[_ZWrite]
75
76             CGPROGRAM
77             #pragma target 3.0
78
79             // -----
80

```

```

81 #pragma shader_feature_local _NORMALMAP
82 #pragma shader_feature_local _ _ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
83 #pragma shader_feature _EMISSION
84 #pragma shader_feature_local _METALLICGLOSSMAP
85 #pragma shader_feature_local _DETAIL_MULX2
86 #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
87 #pragma shader_feature_local _SPECULARHIGHLIGHTS_OFF
88 #pragma shader_feature_local _GLOSSYREFLECTIONS_OFF
89 #pragma shader_feature_local _PARALLAXMAP
90
91 #pragma multi_compile_fwdbase
92 #pragma multi_compile_fog
93 #pragma multi_compile_instancing
94 // Uncomment the following line to enable dithering LOD crossfade. Note: there are more in the file to uncomment for other
95 // #pragma multi_compile _ LOD_FADE_CROSSFADE
96
97 #pragma vertex vertBase
98 #pragma fragment fragBase
99 #include "UnityStandardCoreForward.cginc"
100
101 ENDCG
102 }
103 // -----
104 // Additive forward pass (one light per pass)
105 Pass
106 {
107     Name "FORWARD_DELTA"
108     Tags { "LightMode" = "ForwardAdd" }
109     Blend[_SrcBlend] One
110     Fog { Color(0,0,0,0) } // in additive pass fog should be black
111     ZWrite Off
112     ZTest LEqual
113
114     CGPROGRAM
115     #pragma target 3.0
116
117 // -----
118
119 #pragma shader_feature_local _NORMALMAP
120 #pragma shader_feature_local _ _ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
121 #pragma shader_feature_local _METALLICGLOSSMAP
122 #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
123 #pragma shader_feature_local _SPECULARHIGHLIGHTS_OFF
124 #pragma shader_feature_local _DETAIL_MULX2
125 #pragma shader_feature_local _PARALLAXMAP
126
127 #pragma multi_compile_fwdadd_fullshadows
128 #pragma multi_compile_fog
129 // Uncomment the following line to enable dithering LOD crossfade. Note: there are more in the file to uncomment for other
130 // #pragma multi_compile _ LOD_FADE_CROSSFADE
131
132 #pragma vertex vertAdd
133 #pragma fragment fragAdd
134 #include "UnityStandardCoreForward.cginc"
135
136 ENDCG
137 }
138 // -----
139 // Shadow rendering pass
140 Pass {
141     Name "ShadowCaster"
142     Tags { "LightMode" = "ShadowCaster" }
143
144     ZWrite On ZTest LEqual
145
146     CGPROGRAM

```

```

147     #pragma target 3.0
148
149     // -----
150
151
152     #pragma shader_feature_local _ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
153     #pragma shader_feature_local _METALLICGLOSSMAP
154     #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
155     #pragma shader_feature_local _PARALLAXMAP
156     #pragma multi_compile_shadowcaster
157     #pragma multi_compile_instancing
158     // Uncomment the following line to enable dithering LOD crossfade. Note: there are more in the file to uncomment for other
159     // #pragma multi_compile _LOD_FADE_CROSSFADE
160
161     #pragma vertex vertShadowCaster
162     #pragma fragment fragShadowCaster
163
164     #include "UnityStandardShadow.cginc"
165
166     ENDCG
167 }
168 // -----
169 // Deferred pass
170 Pass
171 {
172     Name "DEFERRED"
173     Tags { "LightMode" = "Deferred" }
174
175     CGPROGRAM
176     #pragma target 3.0
177     #pragma exclude_renderers nomrt
178
179
180     // -----
181
182     #pragma shader_feature_local _NORMALMAP
183     #pragma shader_feature_local _ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
184     #pragma shader_feature _EMISSION
185     #pragma shader_feature_local _METALLICGLOSSMAP
186     #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
187     #pragma shader_feature_local _SPECULARHIGHLIGHTS_OFF
188     #pragma shader_feature_local _DETAIL_MULX2
189     #pragma shader_feature_local _PARALLAXMAP
190
191     #pragma multi_compile_prepassfinal
192     #pragma multi_compile_instancing
193     // Uncomment the following line to enable dithering LOD crossfade. Note: there are more in the file to uncomment for other
194     // #pragma multi_compile _LOD_FADE_CROSSFADE
195
196     #pragma vertex vertDeferred
197     #pragma fragment fragDeferred
198
199     #include "UnityStandardCore.cginc"
200
201     ENDCG
202 }
203
204 // -----
205 // Extracts information for lightmapping, GI (emission, albedo, ...)
206 // This pass it not used during regular rendering.
207 Pass
208 {
209     Name "META"
210     Tags { "LightMode" = "Meta" }
211
212     Cull Off

```

```

213
214 CGPROGRAM
215 #pragma vertex vert_meta
216 #pragma fragment frag_meta
217
218 #pragma shader_feature _EMISSION
219 #pragma shader_feature_local _METALLICGLOSSMAP
220 #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
221 #pragma shader_feature_local _DETAIL_MULX2
222 #pragma shader_feature EDITOR_VISUALIZATION
223
224 #include "UnityStandardMeta.cginc"
225 ENDCG
226 }
227 }
228
229 SubShader
230 {
231     Tags { "RenderType" = "Opaque" "PerformanceChecks" = "False" }
232     LOD 150
233
234     Stencil{
235         Ref 1
236         Comp[_StencilComp]
237     }
238     // -----
239     // Base forward pass (directional light, emission, lightmaps, ...)
240     Pass
241     {
242         Name "FORWARD"
243         Tags { "LightMode" = "ForwardBase" }
244
245         Blend[_SrcBlend][_DstBlend]
246         ZWrite[_ZWrite]
247
248         CGPROGRAM
249         #pragma target 2.0
250
251         #pragma shader_feature_local _NORMALMAP
252         #pragma shader_feature_local _ _ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
253         #pragma shader_feature _EMISSION
254         #pragma shader_feature_local _METALLICGLOSSMAP
255         #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
256         #pragma shader_feature_local _SPECULARHIGHLIGHTS_OFF
257         #pragma shader_feature_local _GLOSSYREFLECTIONS_OFF
258         // SM2.0: NOT SUPPORTED shader_feature_local _DETAIL_MULX2
259         // SM2.0: NOT SUPPORTED shader_feature_local _PARALLAXMAP
260
261         #pragma skip_variants SHADOWS_SOFT DYNAMICLIGHTMAP_ON DIRLIGHTMAP_COMBINED
262
263         #pragma multi_compile_fwdbase
264         #pragma multi_compile_fog
265
266         #pragma vertex vertBase
267         #pragma fragment fragBase
268         #include "UnityStandardCoreForward.cginc"
269
270     ENDCG
271 }
272 // -----
273 // Additive forward pass (one light per pass)
274 Pass
275 {
276     Name "FORWARD_DELTA"
277     Tags { "LightMode" = "ForwardAdd" }
278     Blend[_SrcBlend] One

```

```

279     Fog { Color(0,0,0,0) } // in additive pass fog should be black
280     ZWrite Off
281     ZTest LEqual
282
283     CGPROGRAM
284     #pragma target 2.0
285
286     #pragma shader_feature_local _NORMALMAP
287     #pragma shader_feature_local __ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
288     #pragma shader_feature_local _METALLICGLOSSMAP
289     #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
290     #pragma shader_feature_local _SPECULARHIGHLIGHTS_OFF
291     #pragma shader_feature_local _DETAIL_MULX2
292     // SM2.0: NOT SUPPORTED shader_feature_local _PARALLAXMAP
293     #pragma skip_variants SHADOWS_SOFT
294
295     #pragma multi_compile_fwdadd_fullshadows
296     #pragma multi_compile_fog
297
298     #pragma vertex vertAdd
299     #pragma fragment fragAdd
300     #include "UnityStandardCoreForward.cginc"
301
302     ENDCG
303 }
304 // -----
305 // Shadow rendering pass
306 Pass {
307     Name "ShadowCaster"
308     Tags { "LightMode" = "ShadowCaster" }
309
310     ZWrite On ZTest LEqual
311
312     CGPROGRAM
313     #pragma target 2.0
314
315     #pragma shader_feature_local __ALPHATEST_ON _ALPHABLEND_ON _ALPHAPREMULTIPLY_ON
316     #pragma shader_feature_local _METALLICGLOSSMAP
317     #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A
318     #pragma skip_variants SHADOWS_SOFT
319     #pragma multi_compile_shadowcaster
320
321     #pragma vertex vertShadowCaster
322     #pragma fragment fragShadowCaster
323
324     #include "UnityStandardShadow.cginc"
325
326     ENDCG
327 }
328 // -----
329 // Extracts information for lightmapping, GI (emission, albedo, ...)
330 // This pass it not used during regular rendering.
331 Pass
332 {
333     Name "META"
334     Tags { "LightMode" = "Meta" }
335
336     Cull Off
337
338     CGPROGRAM
339     #pragma vertex vert_meta
340     #pragma fragment frag_meta
341
342     #pragma shader_feature _EMISSION
343     #pragma shader_feature_local _METALLICGLOSSMAP
344     #pragma shader_feature_local _SMOOTHNESS_TEXTURE_ALBEDO_CHANNEL_A

```



```

345     #pragma shader_feature_local _DETAIL_MULX2
346     #pragma shader_feature EDITOR_VISUALIZATION
347
348     #include "UnityStandardMeta.cginc"
349     ENDCG
350 }
351 }
352
353 }
354
355

```

3.8.4. Sky.shader, модифікація, заснована на вбудованих шейдерах Unity , повинна застосовуватися до ігрового об'єкта Sphere, який використовується як скайбокс , ось документація від Unity про те, як відбирати передню або задню частину (як правило, за замовчуванням вибирається задня частина , тому ви не можете побачити нічого всередині тривимірного об'єкта).

```

1 Shader "Custom/Sky"
2 {
3     Properties
4     {
5         _MainTex("Texture", 2D) = "white" {}
6         [Enum(CompareFunction)] _StencilComp("Stencil Comp",Int) = 3
7
8     SubShader
9     {
10        Tags { "RenderType"="Opaque" }
11        LOD 100
12        Cull Front
13        Pass
14        {
15            Stencil{
16                Ref 1
17                Comp[_StencilComp]
18            }
19            CGPROGRAM
20            #pragma vertex vert
21            #pragma fragment frag
22            // make fog work
23            #pragma multi_compile_fog
24            #include "UnityCG.cginc"
25            struct appdata
26            {
27                float4 vertex : POSITION;
28                float2 uv : TEXCOORD0;
29            };
30            struct v2f
31            {
32                float2 uv : TEXCOORD0;
33                UNITY_FOG_COORDS(1)
34                float4 vertex : SV_POSITION;
35            };
36            sampler2D _MainTex;
37            float4 _MainTex_ST;
38            v2f vert (appdata v)
39            {
40                v2f o;
41                o.vertex = UnityObjectToClipPos(v.vertex);
42                o.uv = TRANSFORM_TEX(v.uv, _MainTex);
43                UNITY_TRANSFER_FOG(o,o.vertex);
44                return o;
45            }
46            fixed4 frag (v2f i) : SV_Target
47            {

```

```

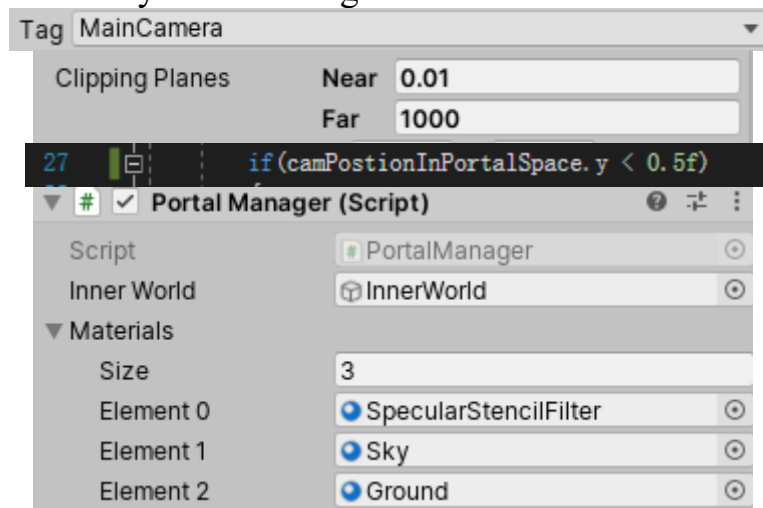
48     // sample the texture
49     fixed4 col = tex2D(_MainTex, i.uv);
50     // apply fog
51     UNITY_APPLY_FOG(i.fogCoord, col);
52     return col;
53 }
54 ENDCG
55 }
56 }
57 }
58

```

3.9. Скрипти

3.9.1. PortalManger.cs, слід застосовувати до компонента PortalDoor.

- 3.9.1.1. Упевніться, що тег камери, яка використовується в сцені, встановлено на MainCamera
- 3.9.1.2. В інспекторі камери «поблизу» площин відсікання має бути встановлено те саме значення, що й у PortalManger.cs (if(camPostionInPortalSpace.y < 0.01f))
- 3.9.1.3. Перетягніть створений вами внутрішній світ у PortalManager>InnerWorld
- 3.9.1.4. Встановіть кількість матеріалів, що використовуються у Innerworld, і перетягніть їх у PortalManager>Material



```

1
2 using System.Collections;
3 using System.Collections.Generic;
4 using UnityEngine;
5 using UnityEngine.Rendering;
6 public class PortalManager : MonoBehaviour
7 {
8     public GameObject InnerWorld;
9     //This materials matter needs to be optimized!
10    public Material[] materials;
11    private Vector3 camPostionInPortalSpace;
12    bool wasInFront;
13    bool inOtherWorld;
14    bool hasCollided;
15    // Start is called before the first frame update
16    void Start()
17    {
18        SetMaterials(false);
19    }
20    void SetMaterials(bool fullRender)
21    {
22        var stencilTest = fullRender ? CompareFunction.NotEqual : CompareFunction.Equal;
23        foreach (var mat in materials)

```

```

24     {
25         mat.SetInt("_StencilComp", (int)stencilTest);
26     }
27 }
28 //Set bidirectional function
29 bool GetIsInFront()
30 {
31     GameObject MainCamera = GameObject.FindGameObjectWithTag("MainCamera");
32     Vector3 worldPos = MainCamera.transform.position + MainCamera.transform.forward * Camera.main.nearClipPlane;
33     camPostionInPortalSpace = transform.InverseTransformPoint(worldPos);
34     return camPostionInPortalSpace.y >= 0 ? true : false;
35 }
36 private void OnTriggerEnter(Collider collider)
37 {
38     GameObject MainCamera = GameObject.FindGameObjectWithTag("MainCamera");
39     if (collider.transform != MainCamera.transform)
40         return;
41     wasInFront = GetIsInFront();
42     hasCollided = true;
43 }
44 // Update is called once per frame
45 void OnTriggerExit(Collider collider)
46 {
47     GameObject MainCamera = GameObject.FindGameObjectWithTag("MainCamera");
48     if (collider.transform != MainCamera.transform)
49         return;
50     hasCollided = false;
51 }
52 void whileCameraColliding()
53 {
54     if (!hasCollided)
55         return;
56     bool isInFront = GetIsInFront();
57     if ((isInFront && !wasInFront) || (wasInFront && !isInFront))
58     {
59         inOtherWorld = !inOtherWorld;
60         SetMaterials(inOtherWorld);
61     }
62     wasInFront = isInFront;
63 }
64 private void OnDestroy()
65 {
66     SetMaterials(true);
67 }
68 private void Update()
69 {
70     whileCameraColliding();
71 }
72 }

```

3.9.2. PlaceOnPlaneOnce.cs, рекомендований для роботи з порталом.

```

1 using System.Collections.Generic;
2 using UnityEngine;
3 using UnityEngine.XR.ARFoundation;
4 using UnityEngine.XR.ARSubsystems;
5 /// <summary>
6 /// Listens for touch events and performs an AR raycast from the screen touch point.
7 /// AR raycasts will only hit detected trackables like feature points and planes.
8 ///
9 /// If a raycast hits a trackable, the <see cref="placedPrefab"/> is instantiated
10 /// and moved to the hit position.
11 /// </summary>
12 [RequireComponent(typeof(ARRaycastManager))]
13 public class PlaceOnPlaneOnce : MonoBehaviour
14 {
15     [SerializeField]
16     [Tooltip("Instantiates this prefab on a plane at the touch location.")]

```

```

17     GameObject m_PlacedPrefab;
18     /// <summary>
19     /// The prefab to instantiate on touch.
20     /// </summary>
21     public GameObject placedPrefab
22     {
23         get { return m_PlacedPrefab; }
24         set { m_PlacedPrefab = value; }
25     }
26     /// <summary>
27     /// The object instantiated as a result of a successful raycast intersection with a plane.
28     /// </summary>
29     public GameObject spawnedObject { get; private set; }
30     void Awake()
31     {
32         m_RaycastManager = GetComponent<ARRaycastManager>();
33     }
34     bool TryGetTouchPosition(out Vector2 touchPosition)
35     {
36     #if UNITY_EDITOR
37         if (Input.GetMouseButton(0))
38         {
39             var mousePosition = Input.mousePosition;
40             touchPosition = new Vector2(mousePosition.x, mousePosition.y);
41             return true;
42         }
43     #else
44         if (Input.touchCount > 0)
45         {
46             touchPosition = Input.GetTouch(0).position;
47             return true;
48         }
49     #endif
50         touchPosition = default;
51         return false;
52     }
53     void Update()
54     {
55         if (!TryGetTouchPosition(out Vector2 touchPosition))
56             return;
57         if (m_RaycastManager.Raycast(touchPosition, s_Hits, TrackableType.PlaneWithinPolygon))
58         {
59             // Raycast hits are sorted by distance, so the first one
60             // will be the closest hit.
61             var hitPose = s_Hits[0].pose;
62             if ((spawnedObject == null))
63             {
64                 spawnedObject = Instantiate(m_PlacedPrefab, hitPose.position, hitPose.rotation);
65             }
66         }
67     }
68     static List<ARRaycastHit> s_Hits = new List<ARRaycastHit>();
69     ARRaycastManager m_RaycastManager;
70 }

```

4. Закомітити файли та проект на гіт. Надайти доступ своєму викладачу.