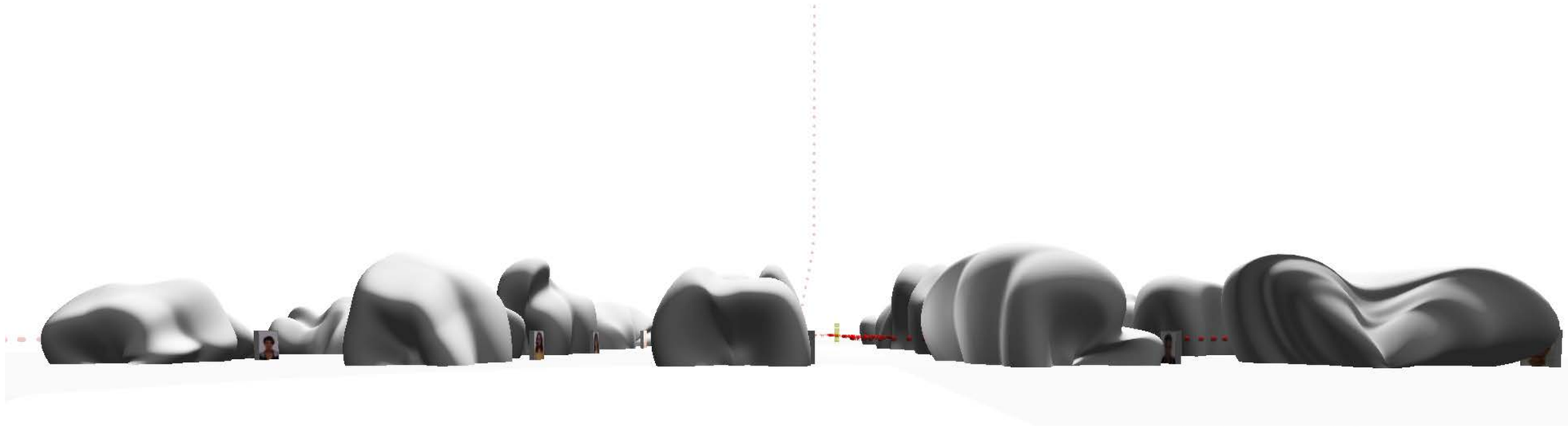


PARAMETRIC FORM .COM

Gameify! (Unity 3D)

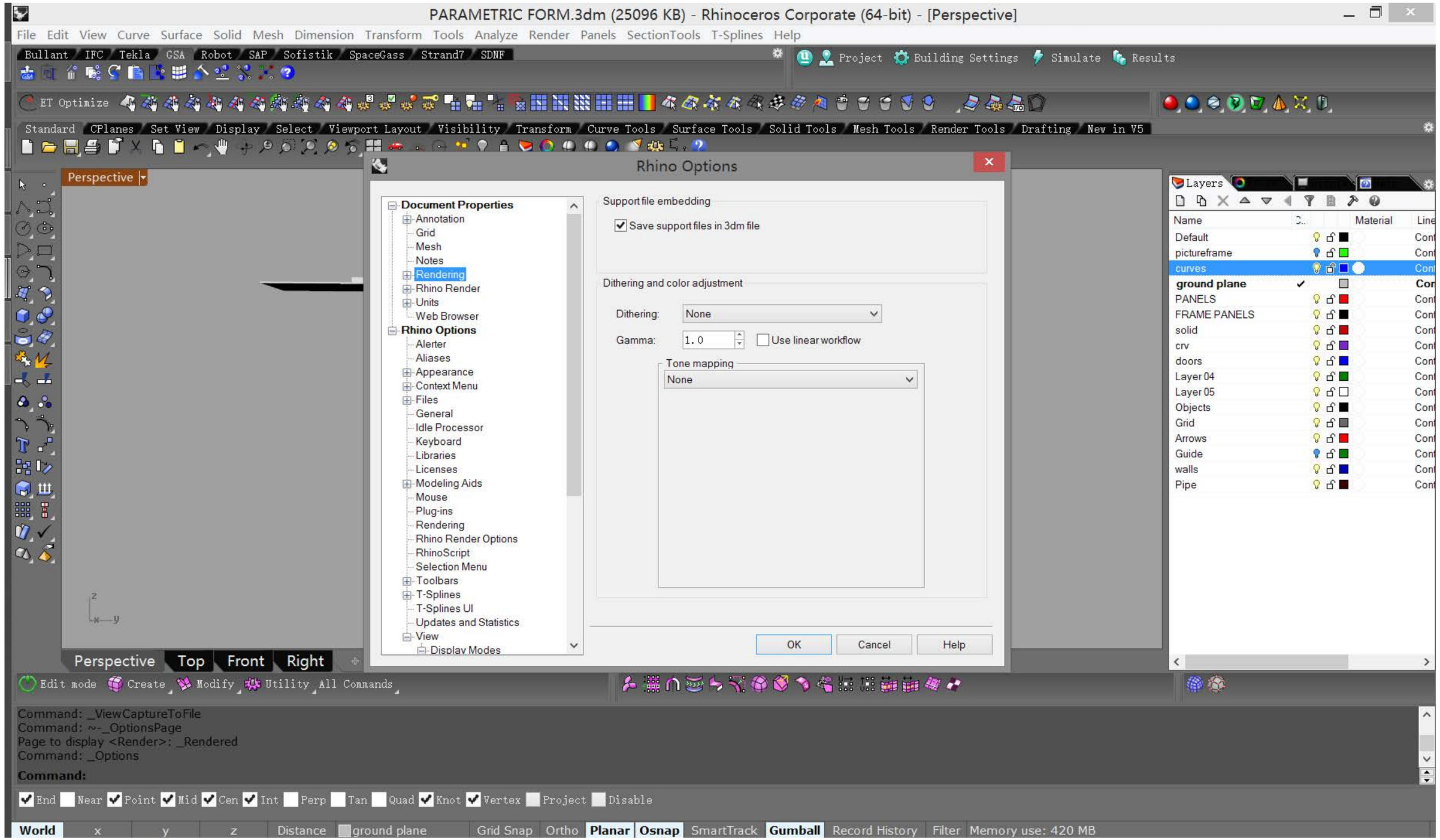
Digital Tools

Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

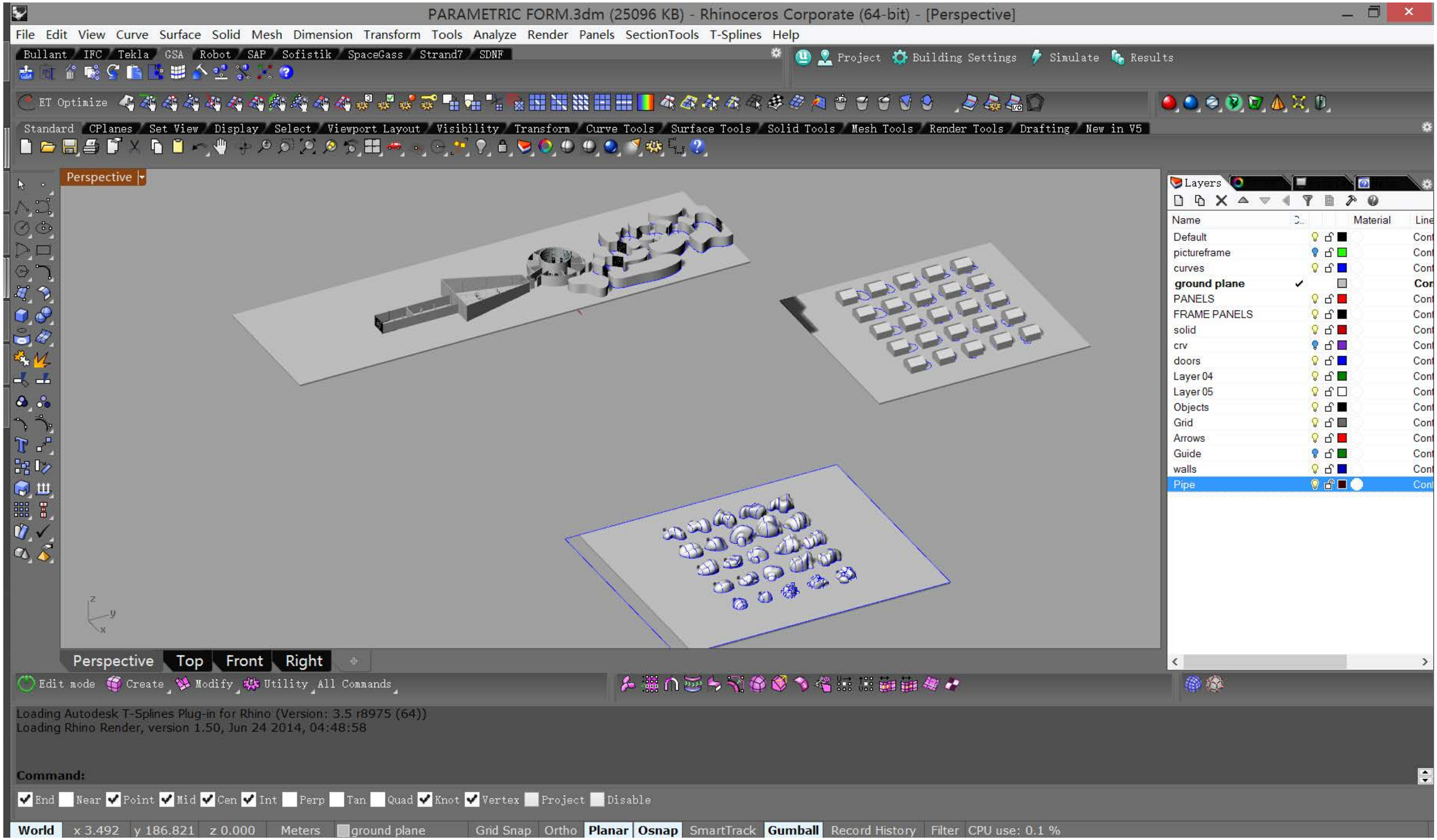
Iaac

Setting up Rhino

To be able to work in different files (with textures applied) and then join them without losing the textures we enable "save support files in 3dm file" in the options menu.



Modeling a virtual world in Rhino



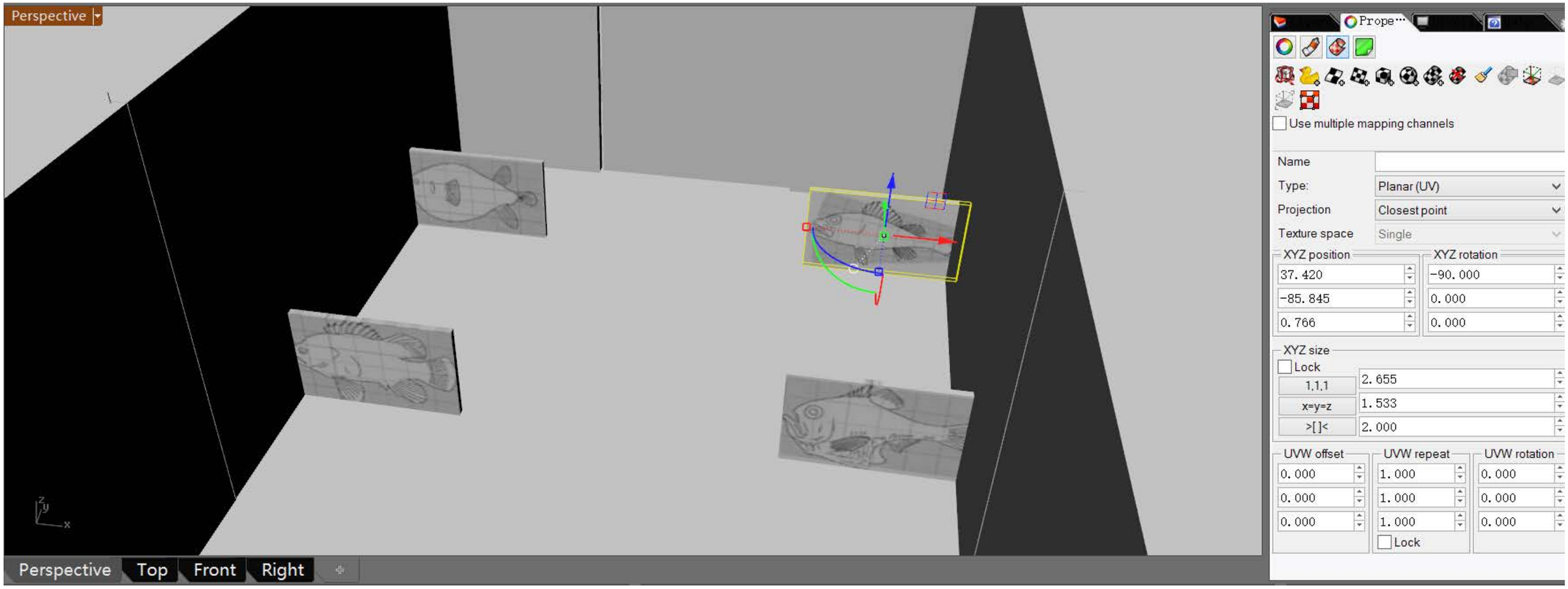
Institute for Advanced Architecture of Catalonia
2014/// Digital Tools
Iaac

Assignment 3 - Gameify! (Unity 3D)

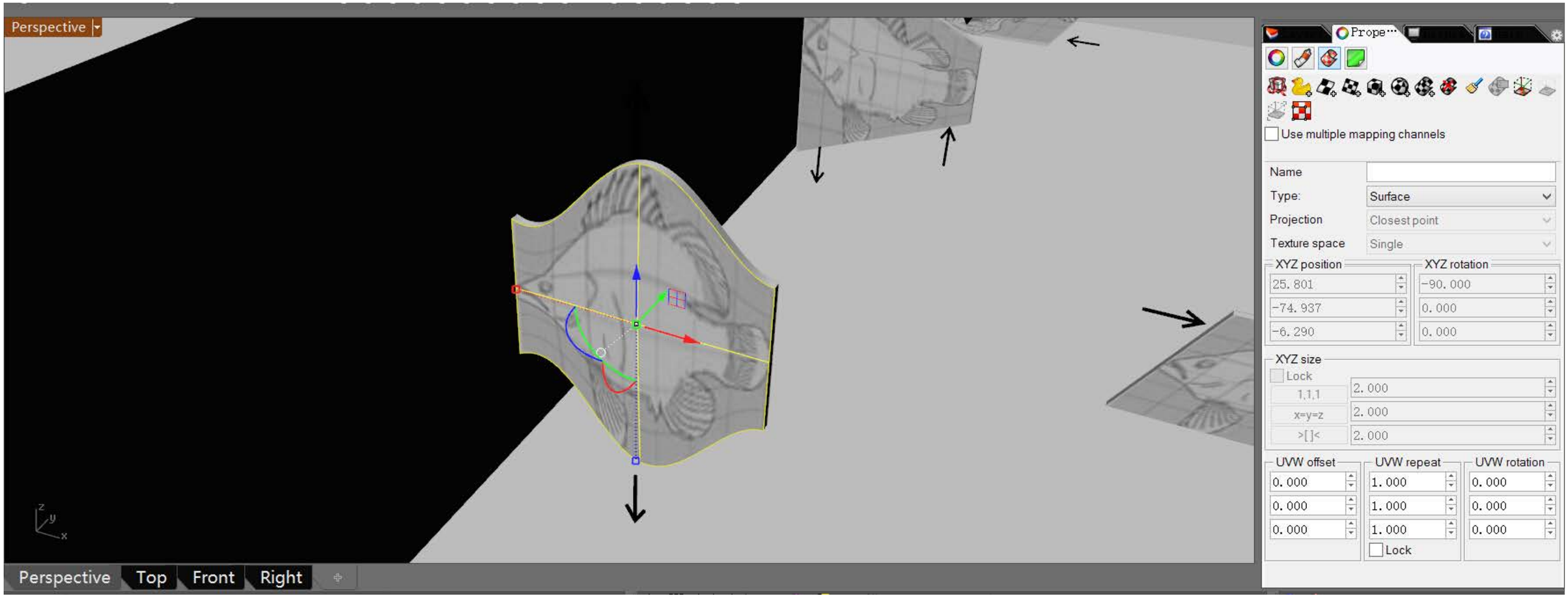
Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Applying textures to the objects

We used different types of UV mapping
(Box, Surface, Plane)



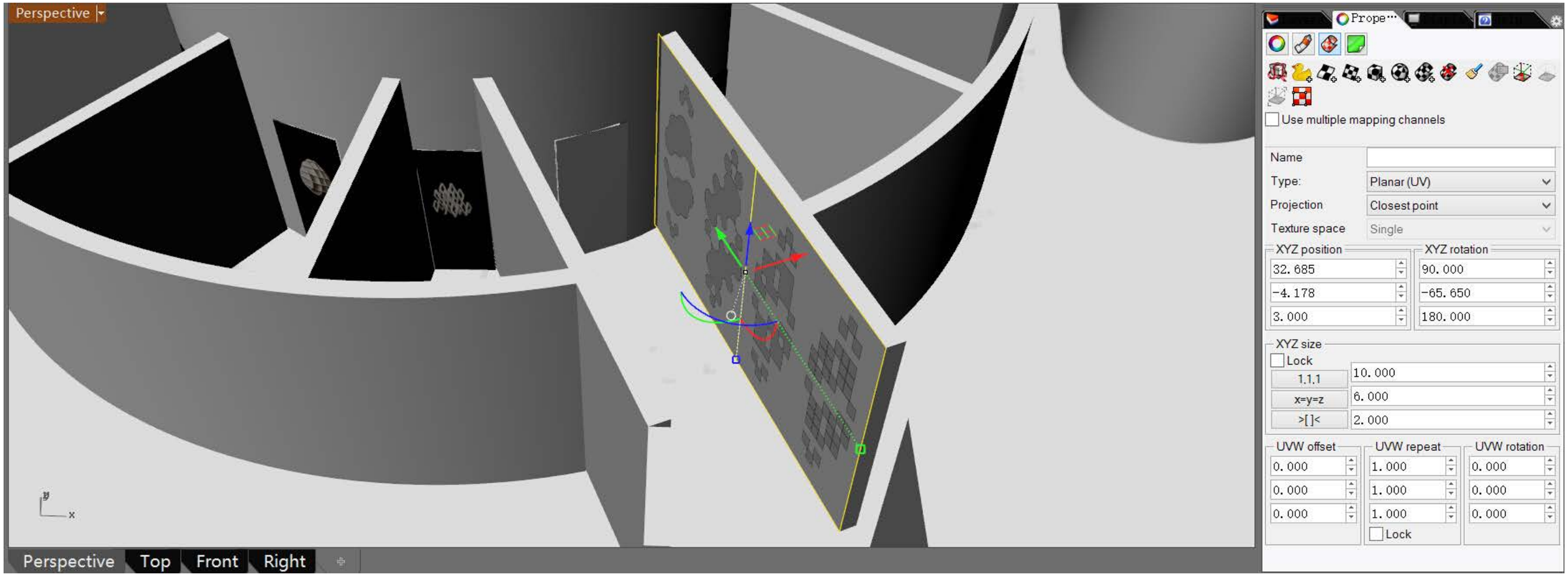
Applying textures to the objects



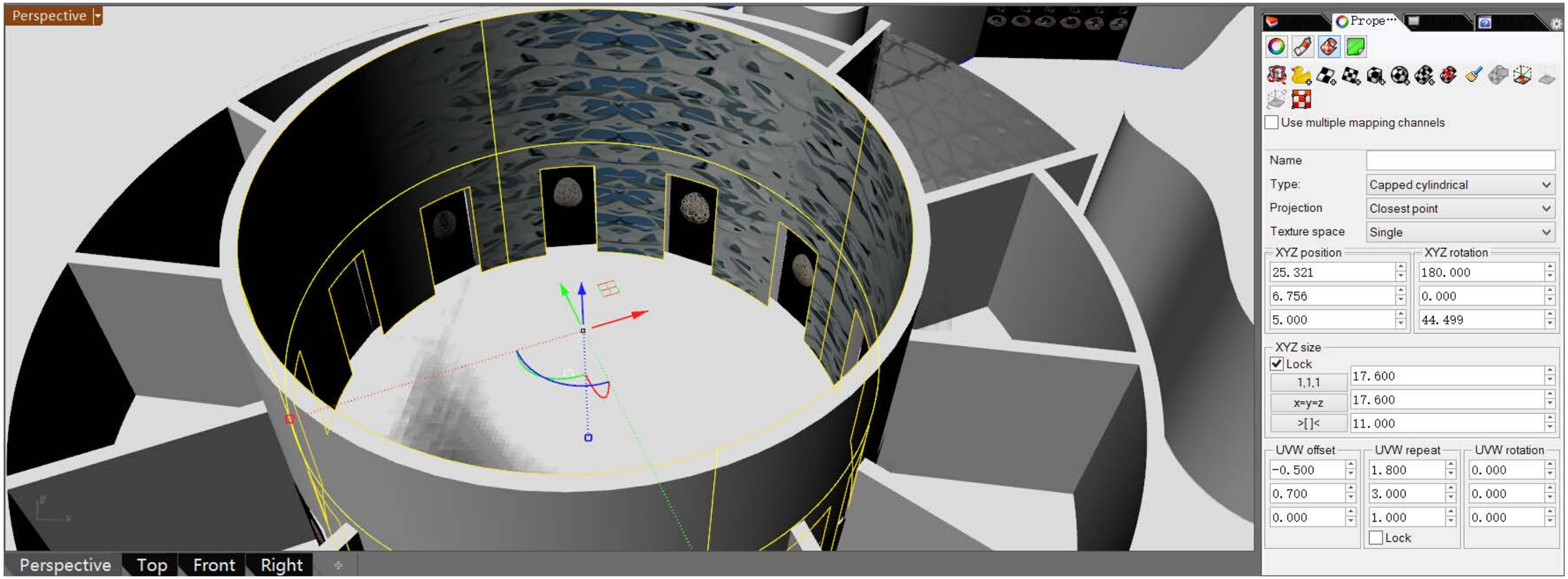
Assignment 3 - Gameify! (Unity 3D)

Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Applying textures to the objects



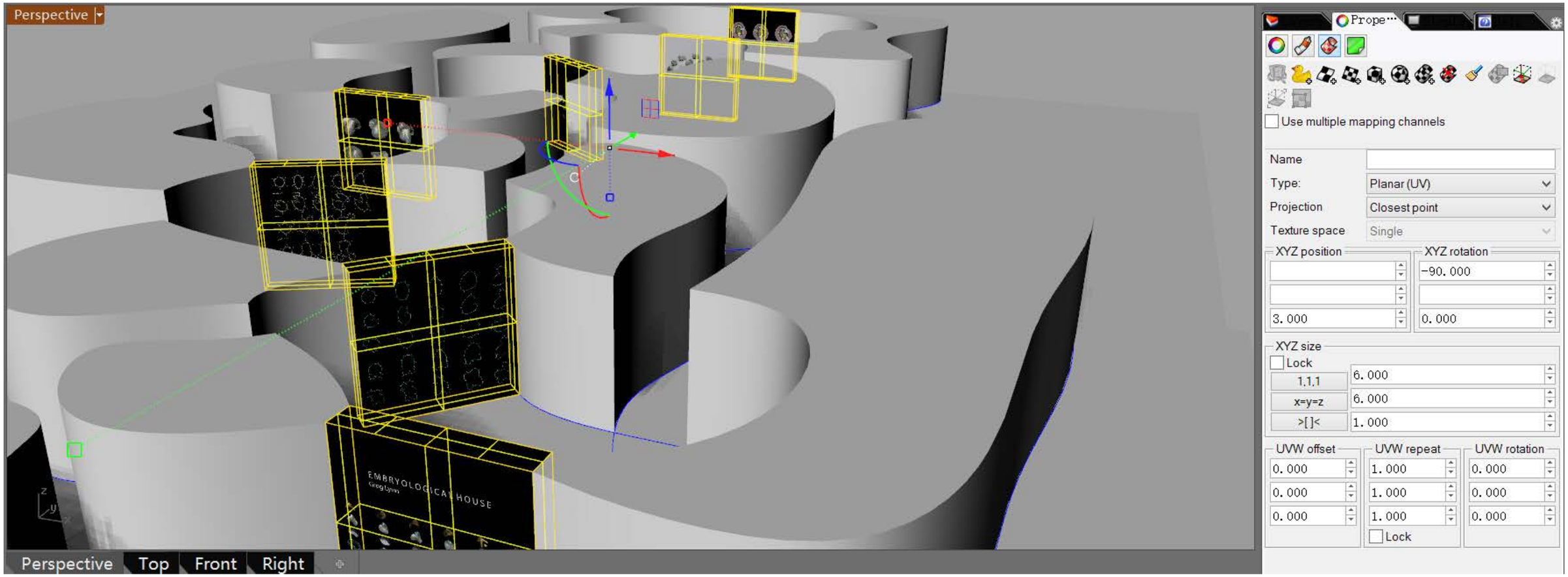
Applying textures to the objects



Assignment 3 - Gameify! (Unity 3D)

Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Applying textures to the objects

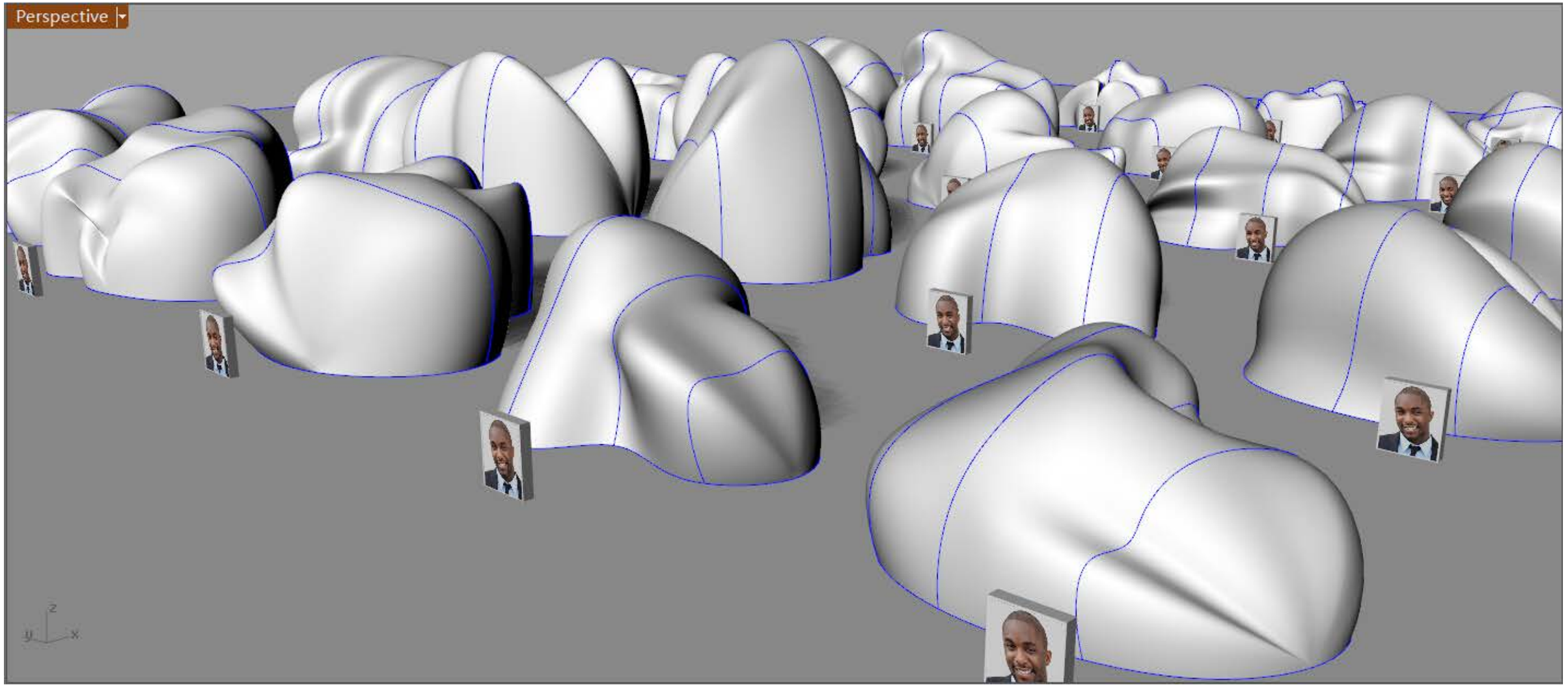


Assignment 3 - Gameify! (Unity 3D)

Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Applying textures to the objects

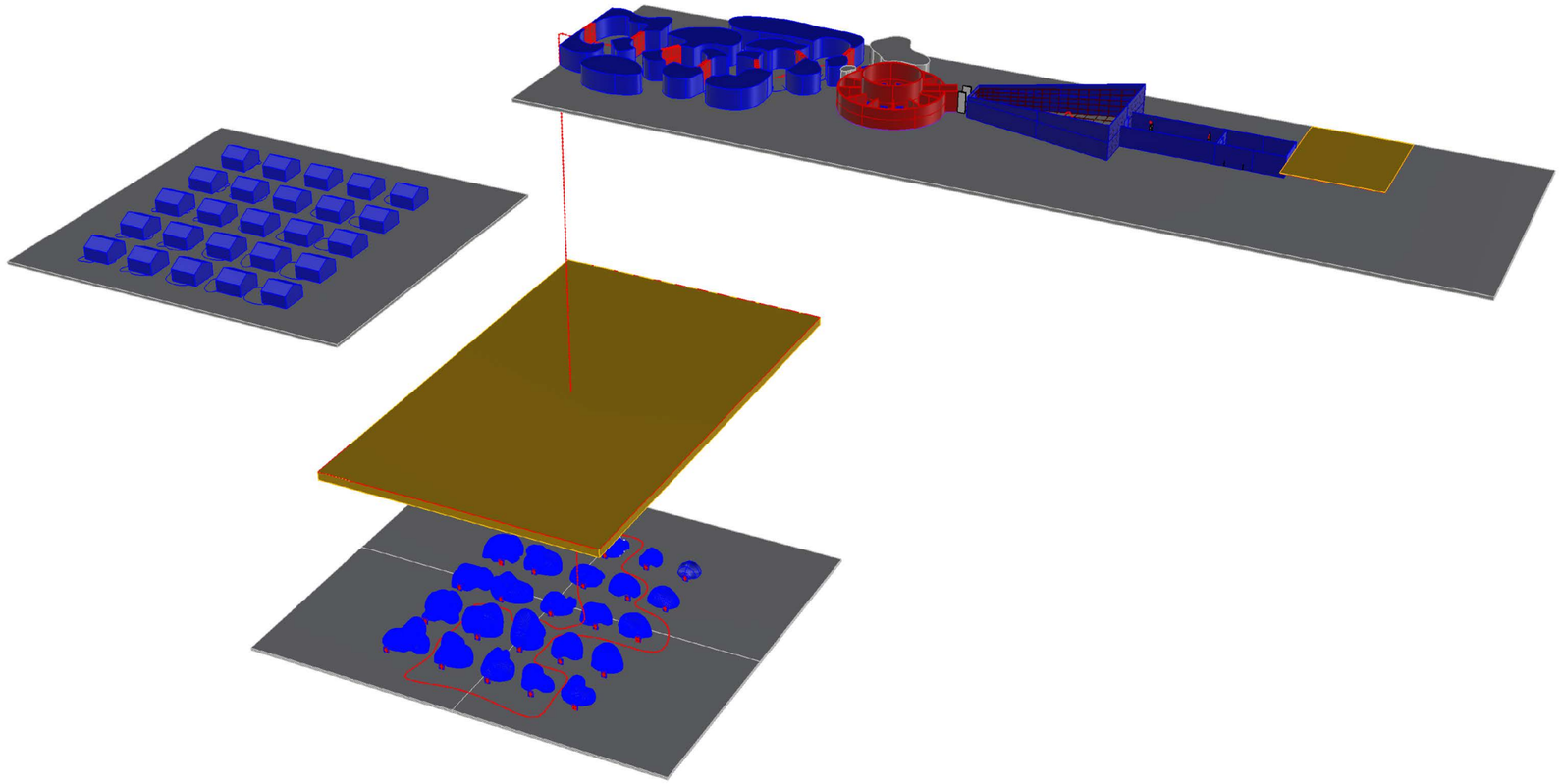
Since the textures will have to be applied again in Unity, we apply the same texture to all the objects with the same proportion just to create the UV projector.



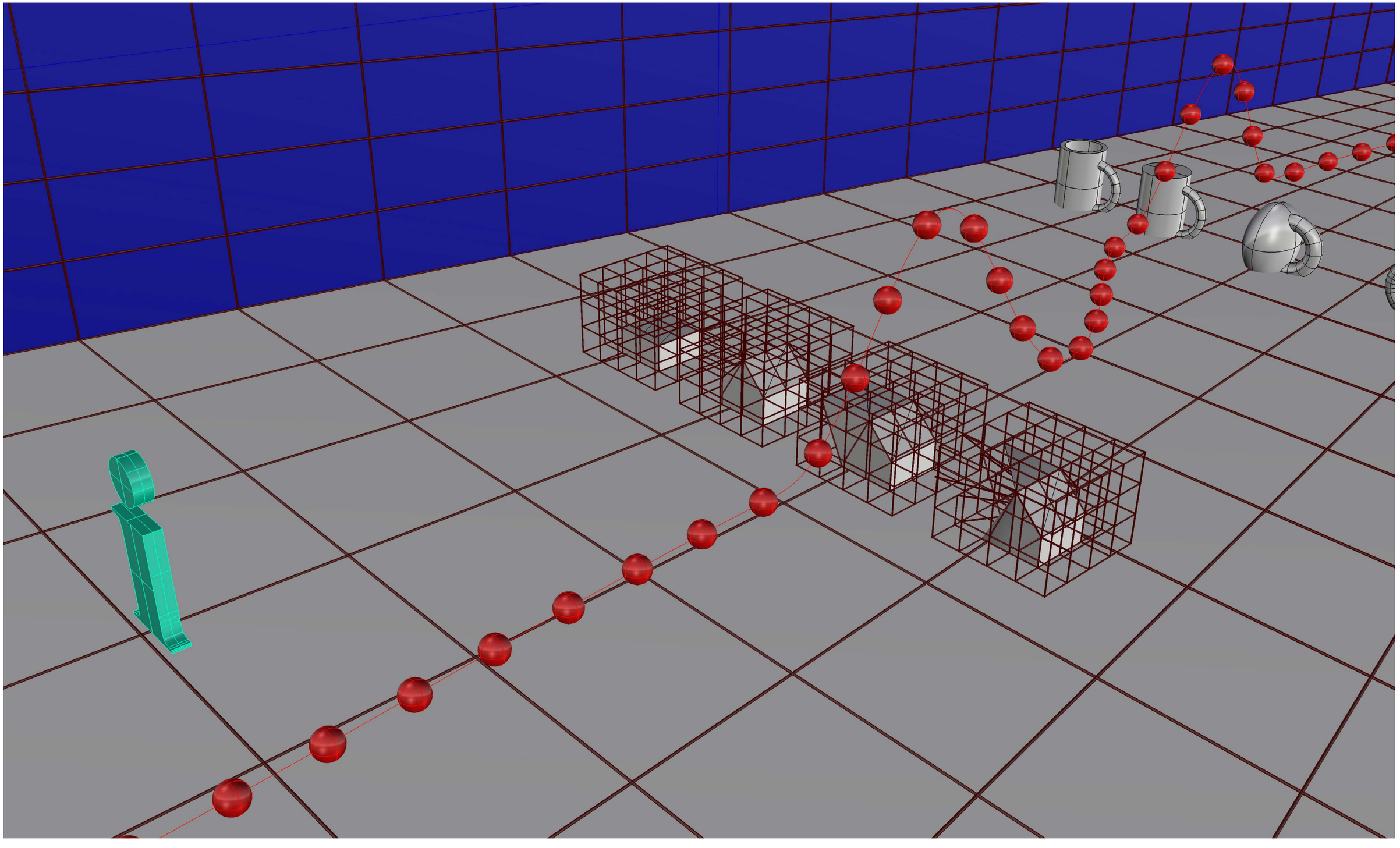
Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Modeling the “guiding” objects

To guide the player through the game we create a path of floating spheres, some information points with floating “i”, and some objects that will be transparent to hold a message that will be displayed when the player collides with them.



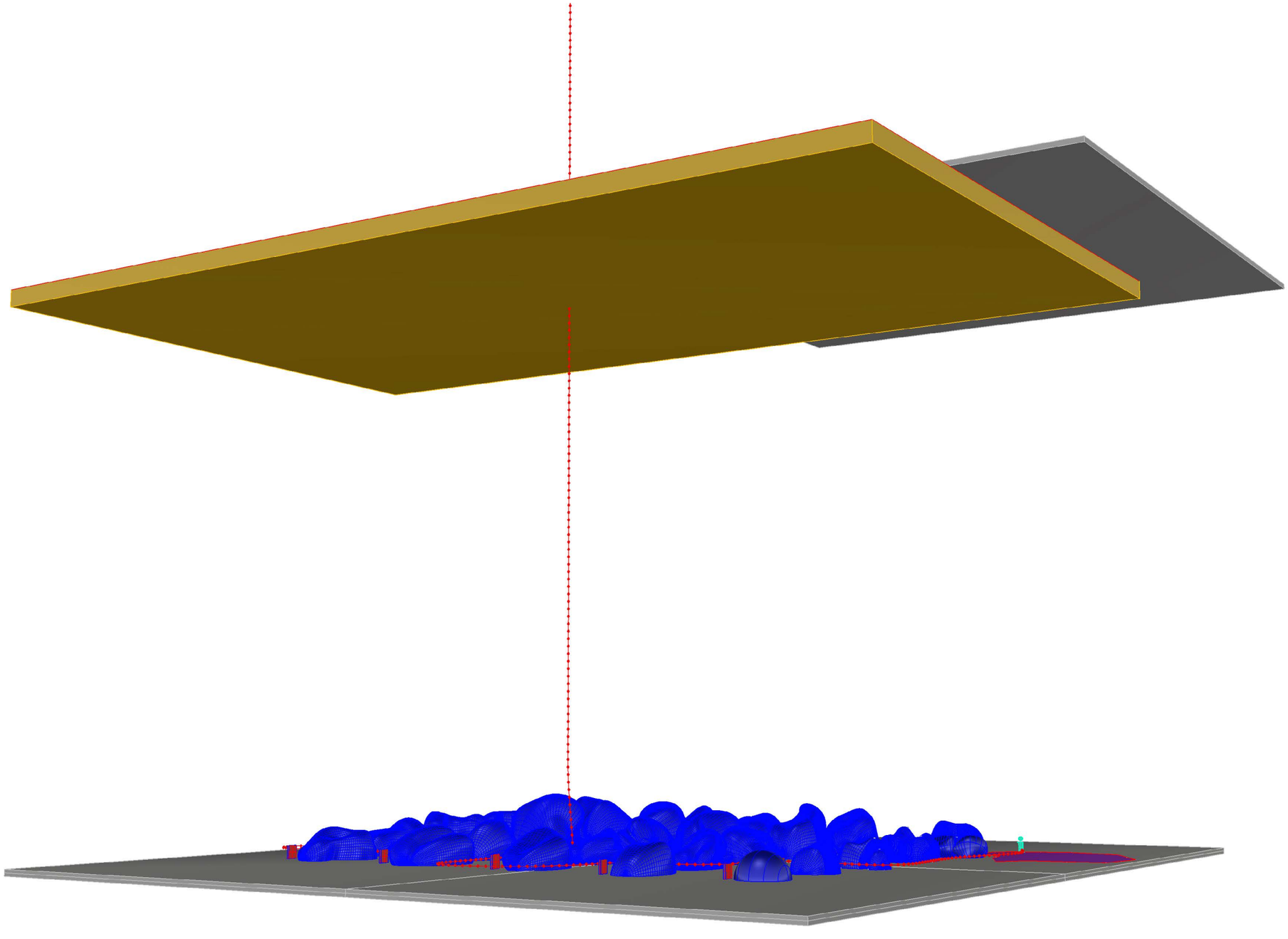
Modeling the “guiding” objects



Assignment 3 - Gameify! (Unity 3D)

Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Modeling the “guiding” objects



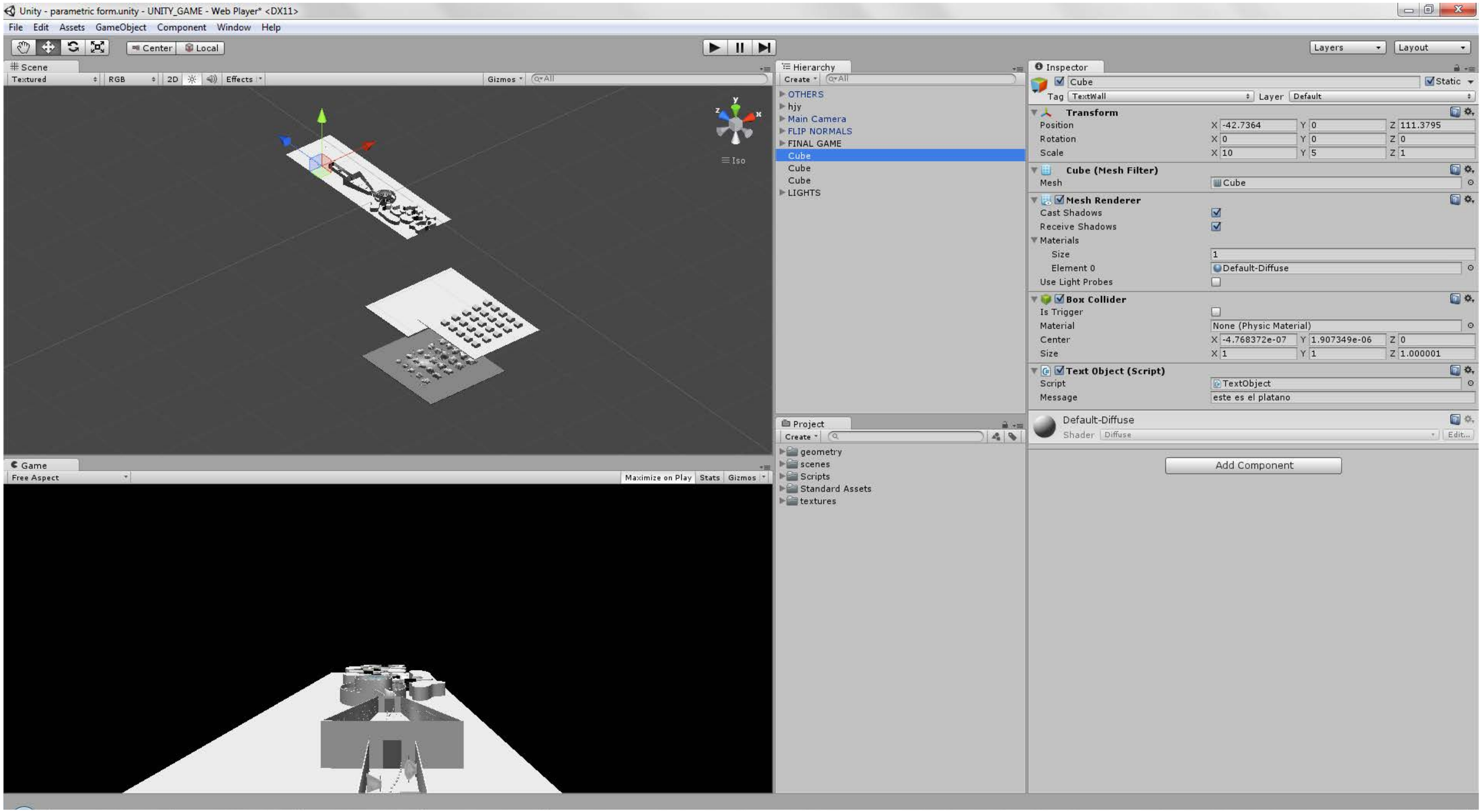
Institute for Advanced Architecture of Catalonia
2014/// Digital Tools

Assignment 3 - Gameify! (Unity 3D)

Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Importing the geometry into Unity

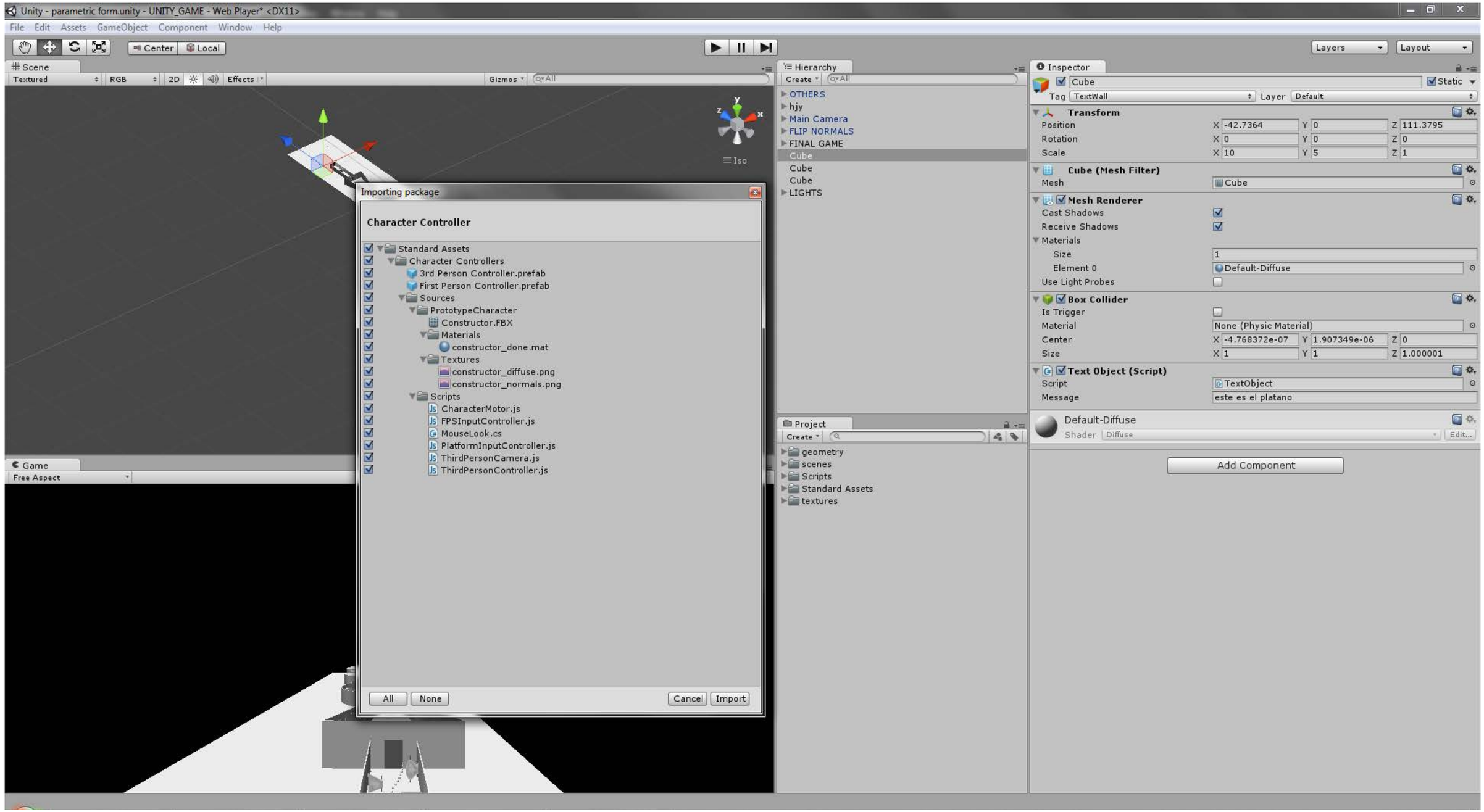
We export the geometry from Rhino in .OBJ format (that generates a folder with all the textures) and we import it as a new asset into Unity.
We generate colliders for all those objects that we don't want to be "phisically transparent" (especially ground planes).



Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Importing the Character Controller package

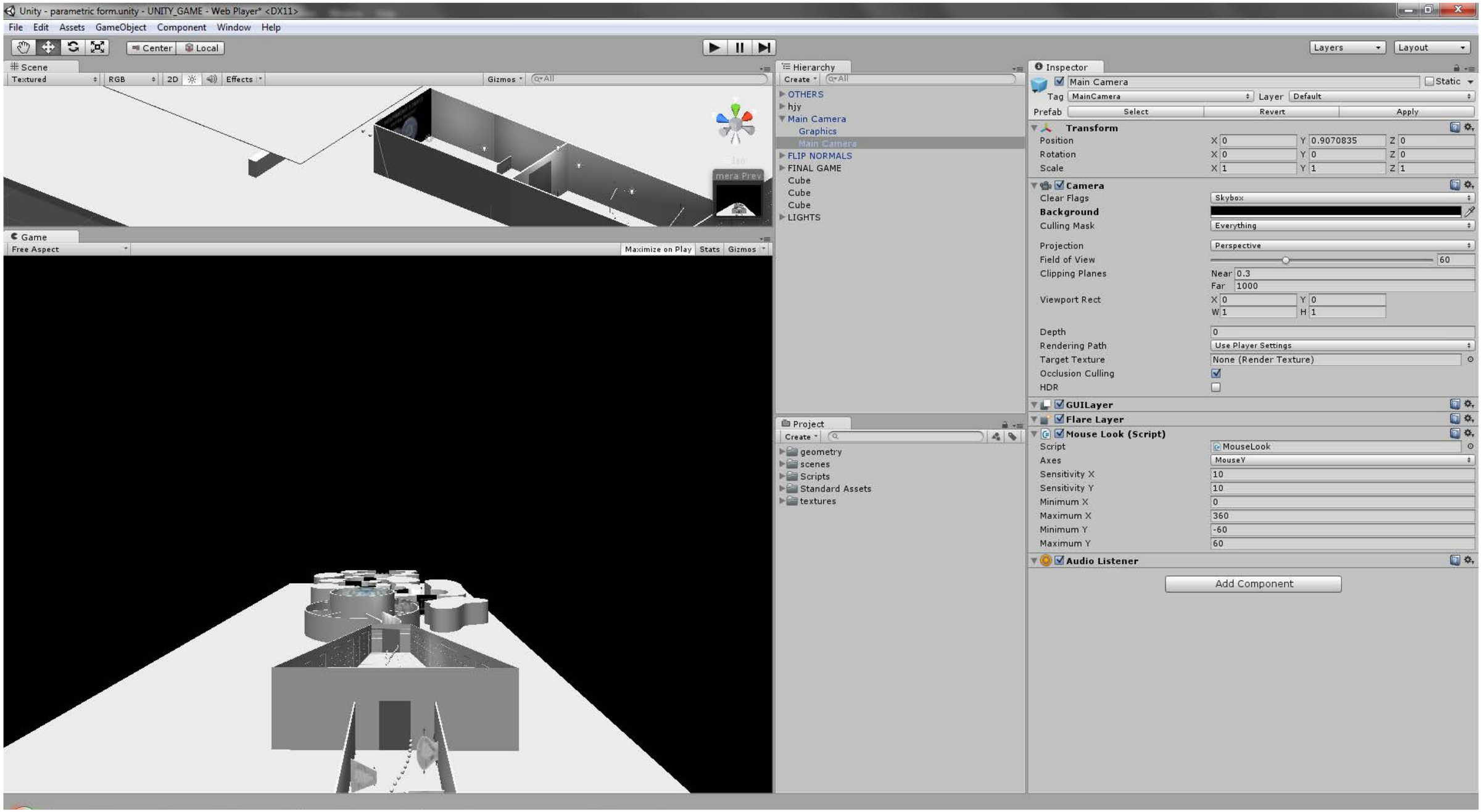
To be able to use the mouse and keyboard to move around the game we import the “Character Controller package” and we use the first person controller.



Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

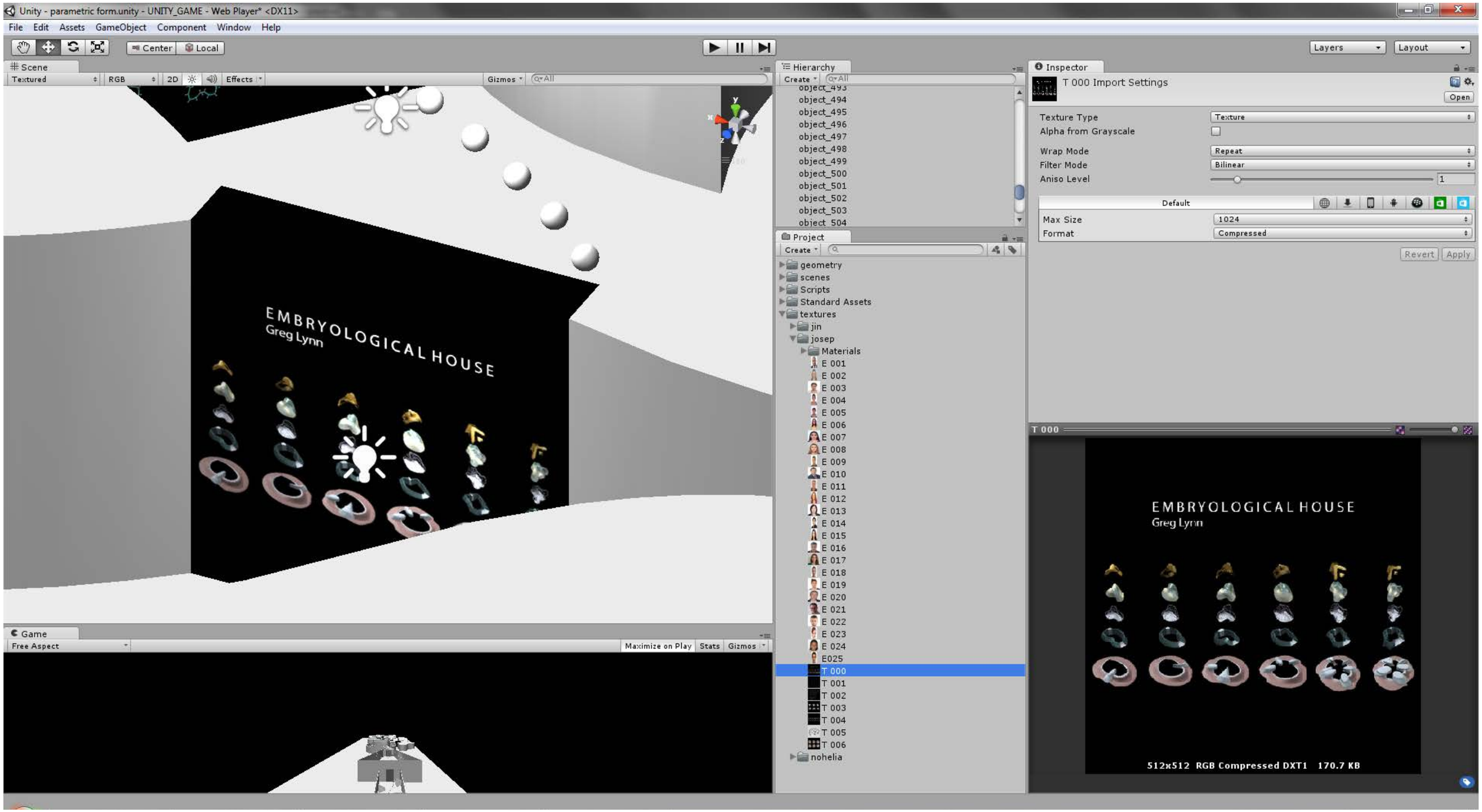
Changing the Background Color

To create a more abstract world we change the background color to a dark grey instead of blue.



Reassigning Textures

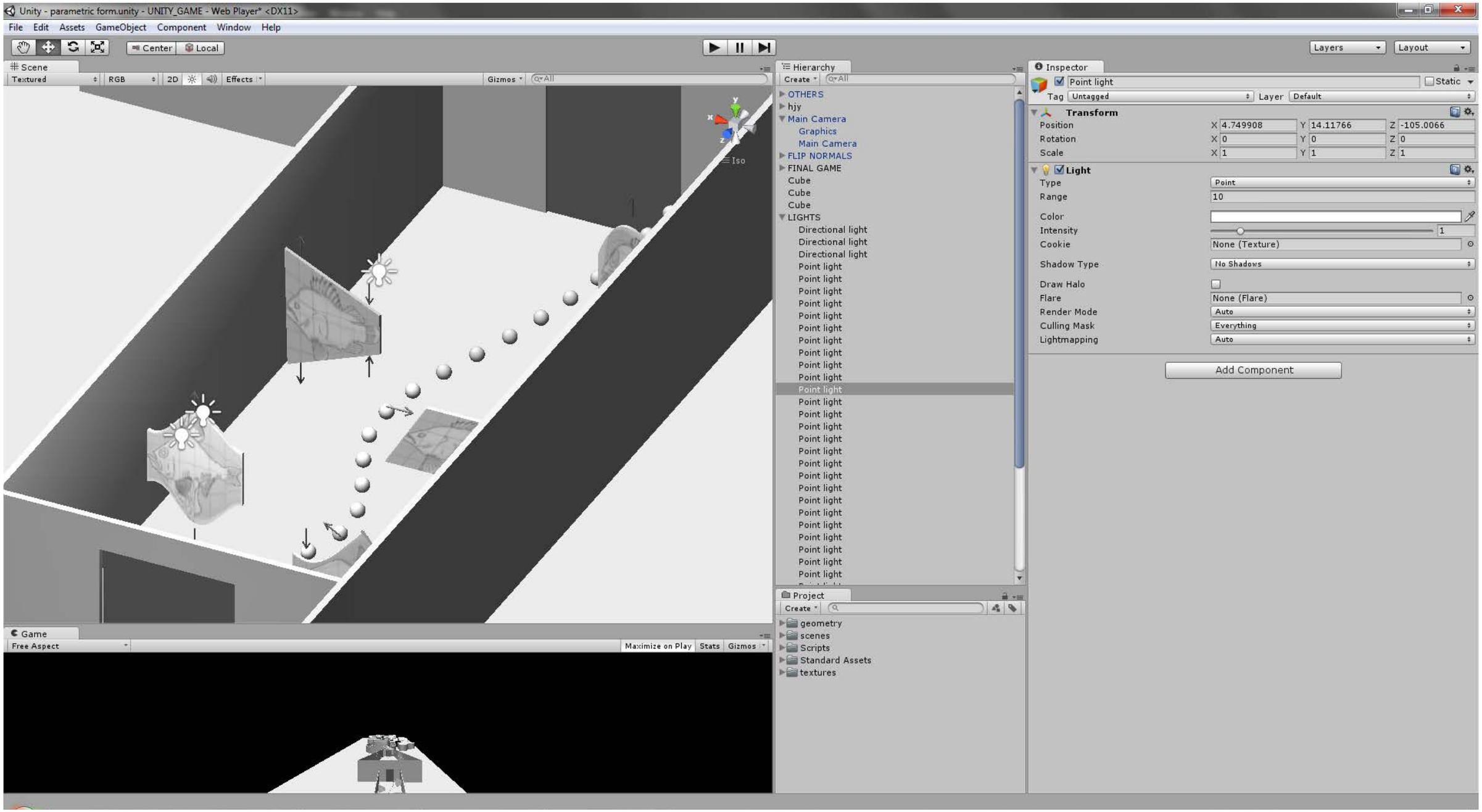
Each texture has to be reassigned manually to its object.



Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Creating Lights

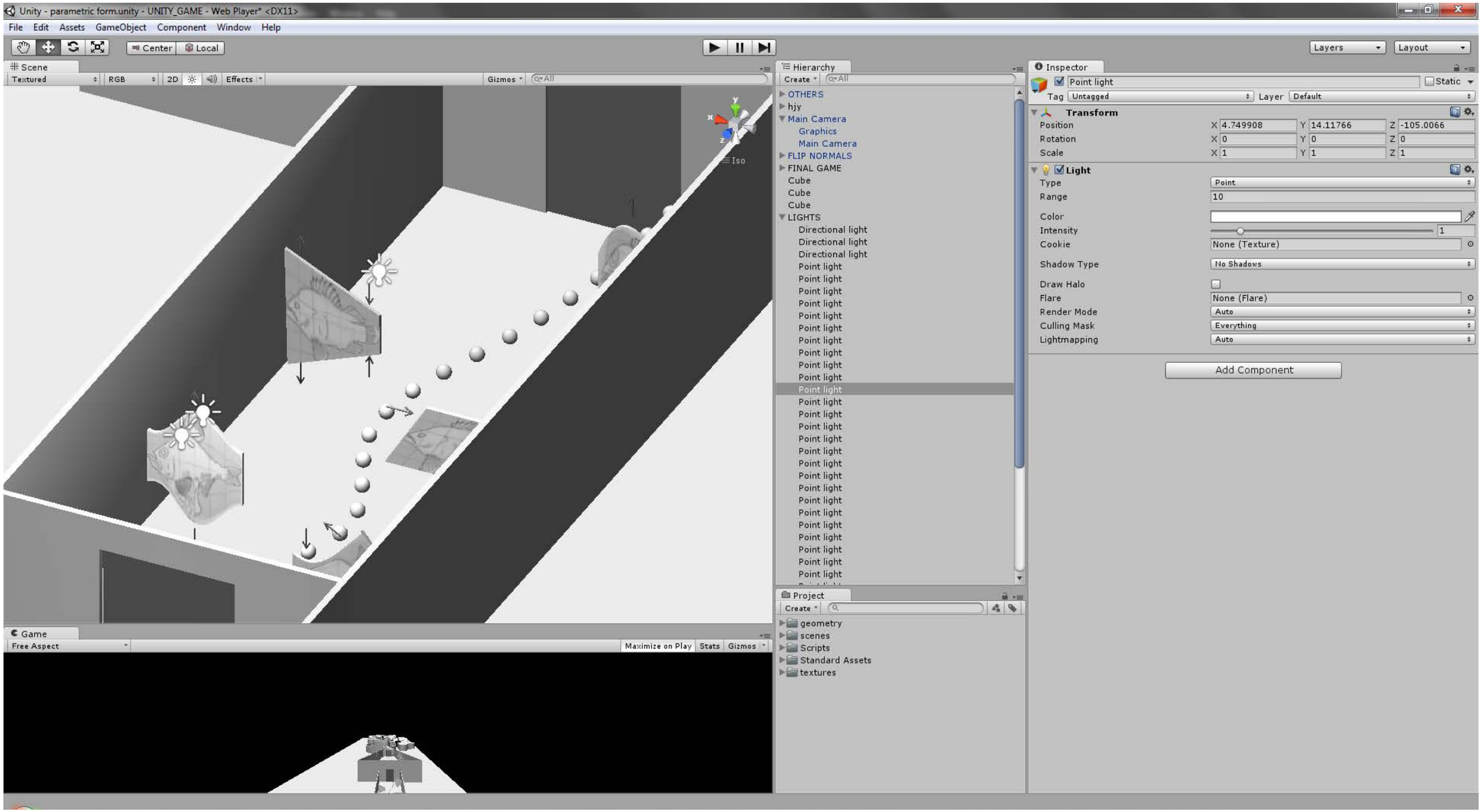
To light the scene we use some directional and spot lights.



Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Assigning materials to multiple objects

Unity doesn't have an option to assign one material to multiple objects at the same time. However we need to assign the same material to all the spheres that create the path to guide the player. The solution is creating a script and save it in a folder called "Editor" inside the "Assets" Folder. By doing so we will be able to access the command from the top toolbar.



Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Assigning materials to multiple objects

AssignMaterial Script

Assembly-CSharp-Editor - Editor\AssignMaterial.cs - MonoDevelop-Unity

File Edit View Search Project Build Run Version Control Tools Window Help

Debug | Default | MonoDevelop-Unity | Press 'Control+,' to search

Solution

- UNITY_GAME
  - Assembly-CSharp
  - Assembly-CSharp-Editor
    - References
    - Editor
      - AssignMaterial.cs
  - Assembly-CSharp-firstpass
  - Assembly-UnityScript-firstpass

AssignMaterial.cs

No selection

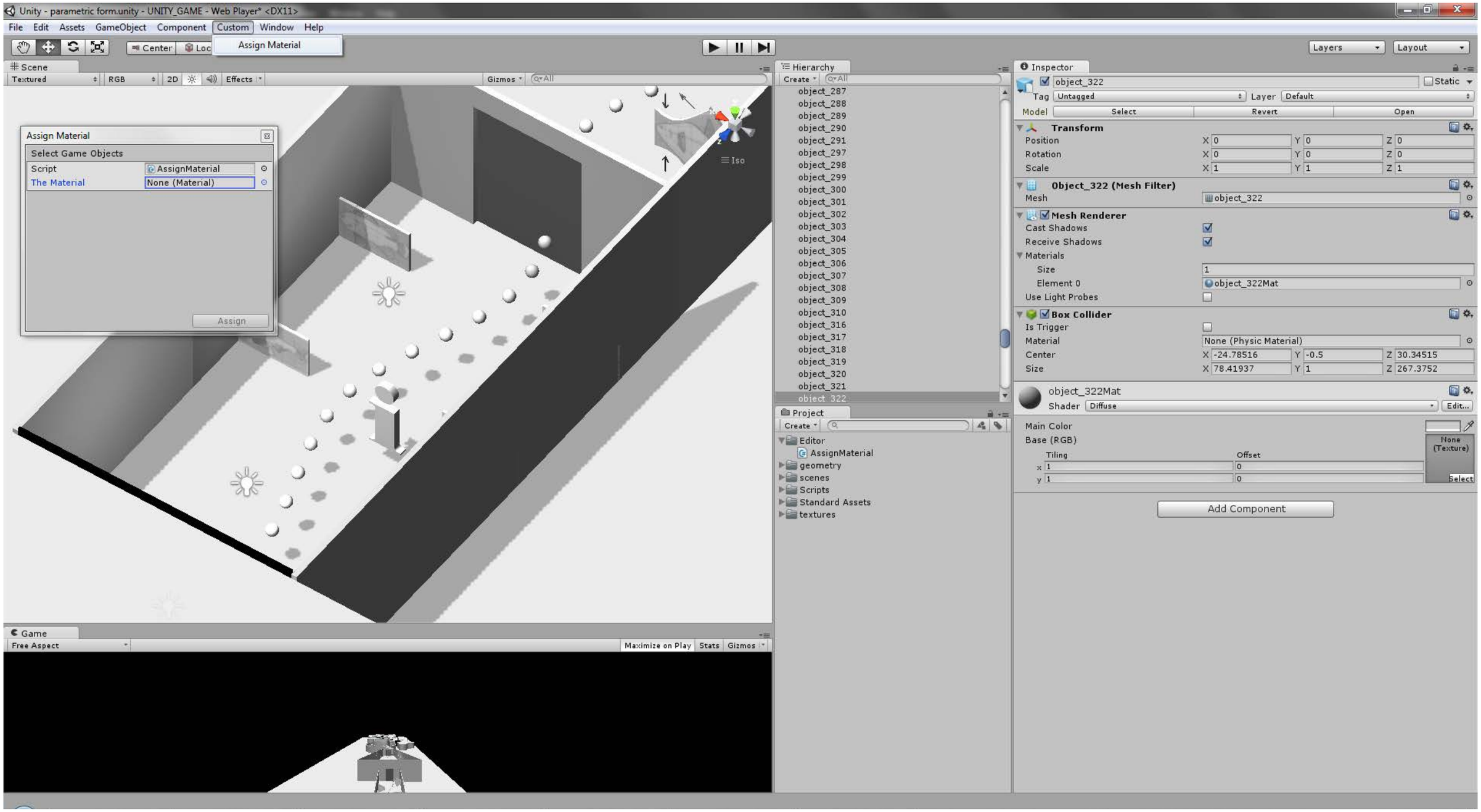
```
1  using UnityEngine;
2  using System.Collections;
3  using System;
4  using UnityEditor;
5
6  public class AssignMaterial : ScriptableWizard
7  {
8
9      public Material theMaterial;
10     String strHelp = "Select Game Objects";
11     GameObject[] gos;
12
13     void OnWizardUpdate ()
14     {
15         helpString = strHelp;
16         isValid = (theMaterial != null);
17     }
18
19     void OnWizardCreate ()
20     {
21         gos = Selection.gameObjects;
22         foreach (GameObject go in gos)
23         {
24             go.renderer.material = theMaterial;
25         }
26     }
27
28     [MenuItem ("Custom/Assign Material", false, 4)]
29     static void assignMaterial()
30     {
31         ScriptableWizard.DisplayWizard ("Assign Material", typeof(AssignMaterial), "Assign");
32     }
33 }
```

Toolbox | Properties | Document Outline | Unit Tests

Errors | Tasks

Assigning materials to multiple objects

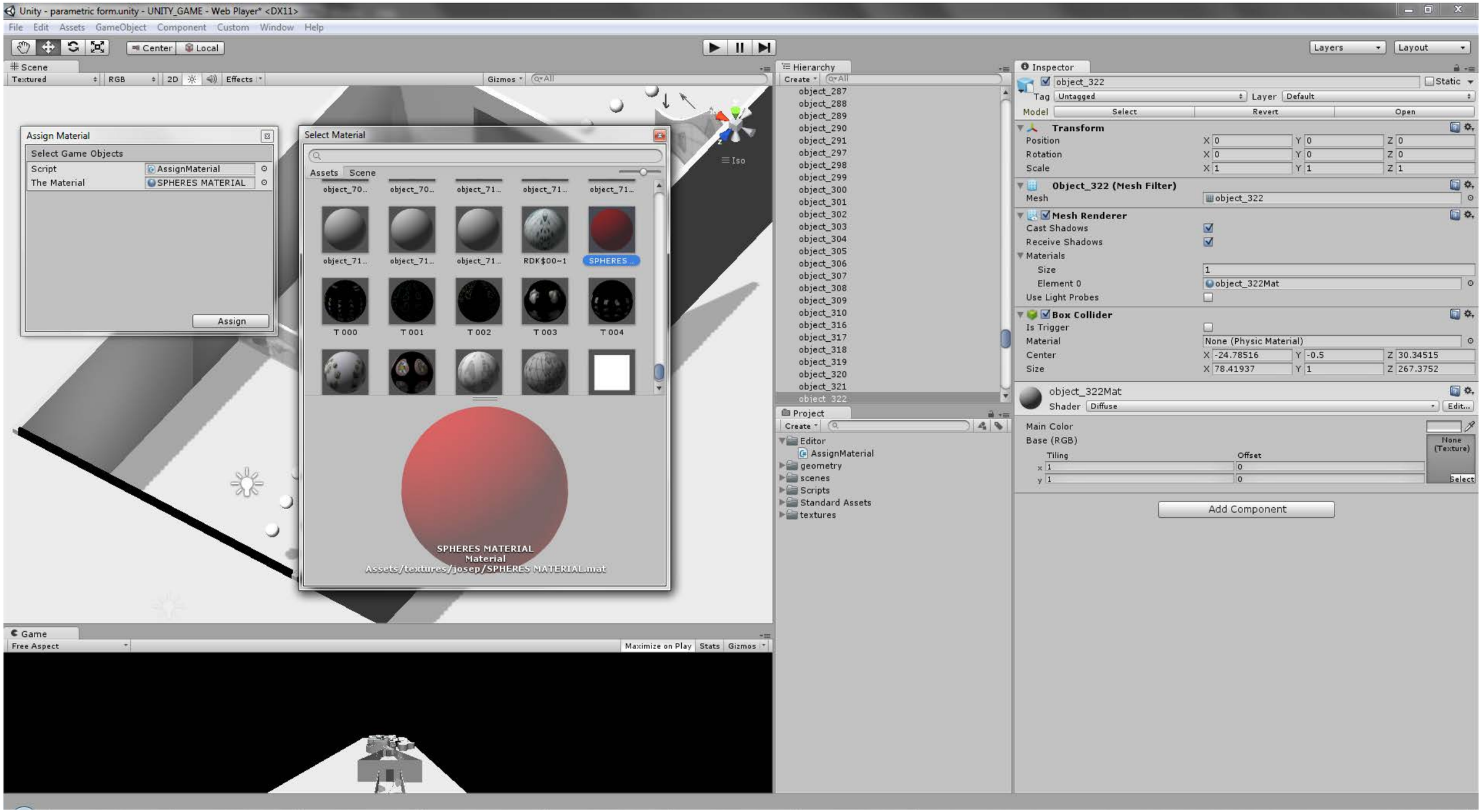
Running the Assign Material script from the top toolbar.



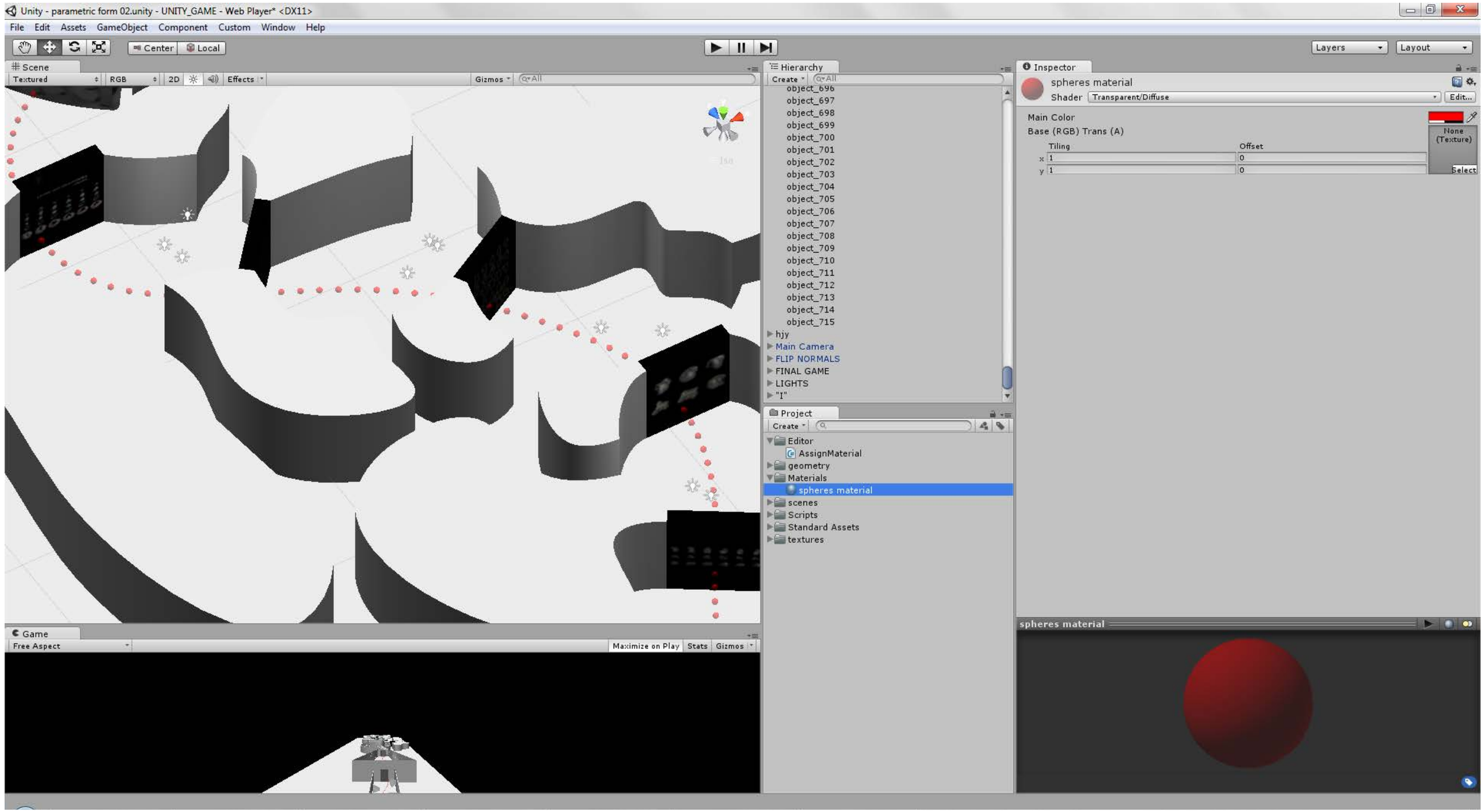
Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Assigning materials to multiple objects

We select the material that we previously created (transparent diffuse red material) for the spheres



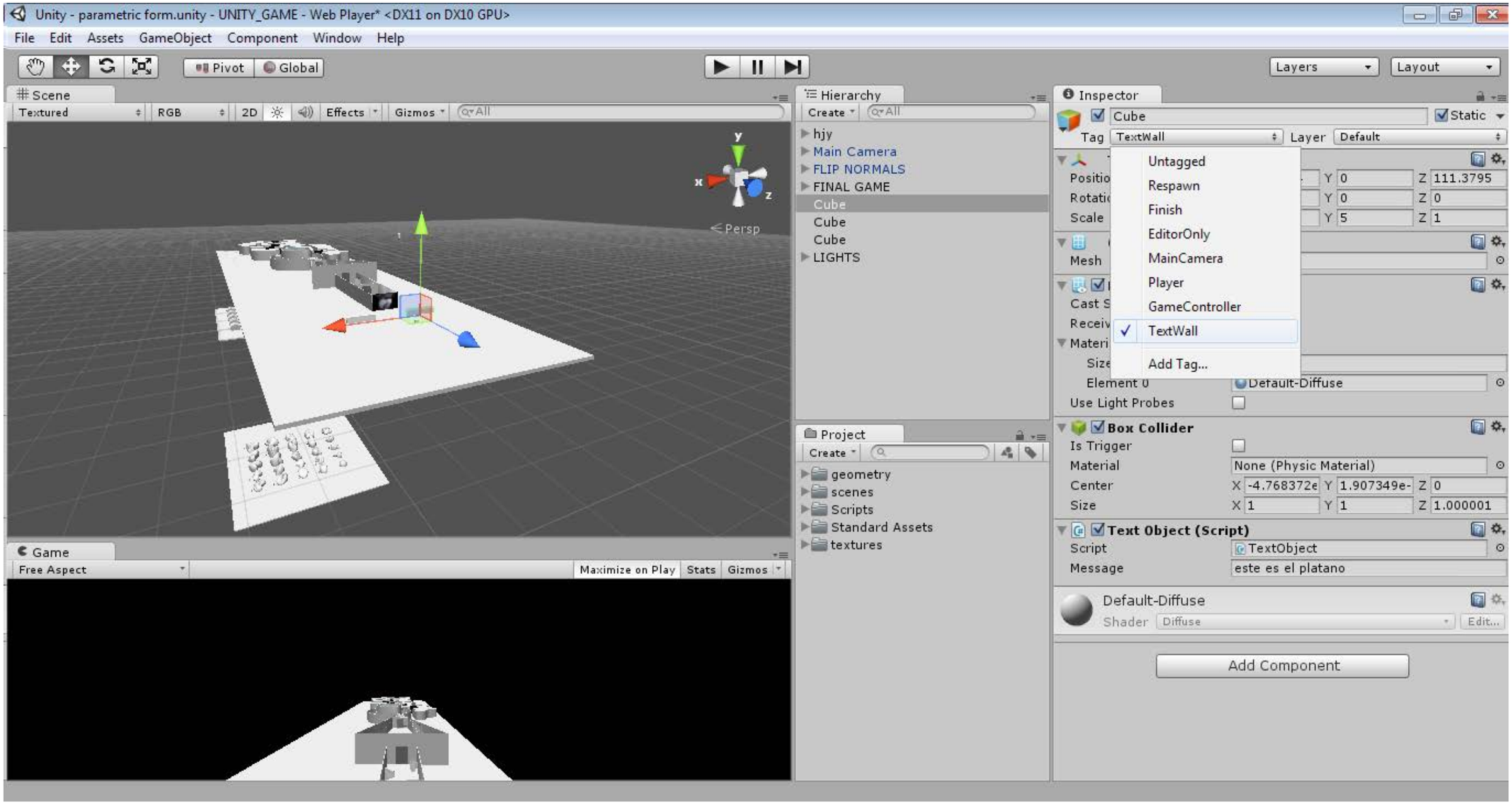
Assigning materials to multiple objects



Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

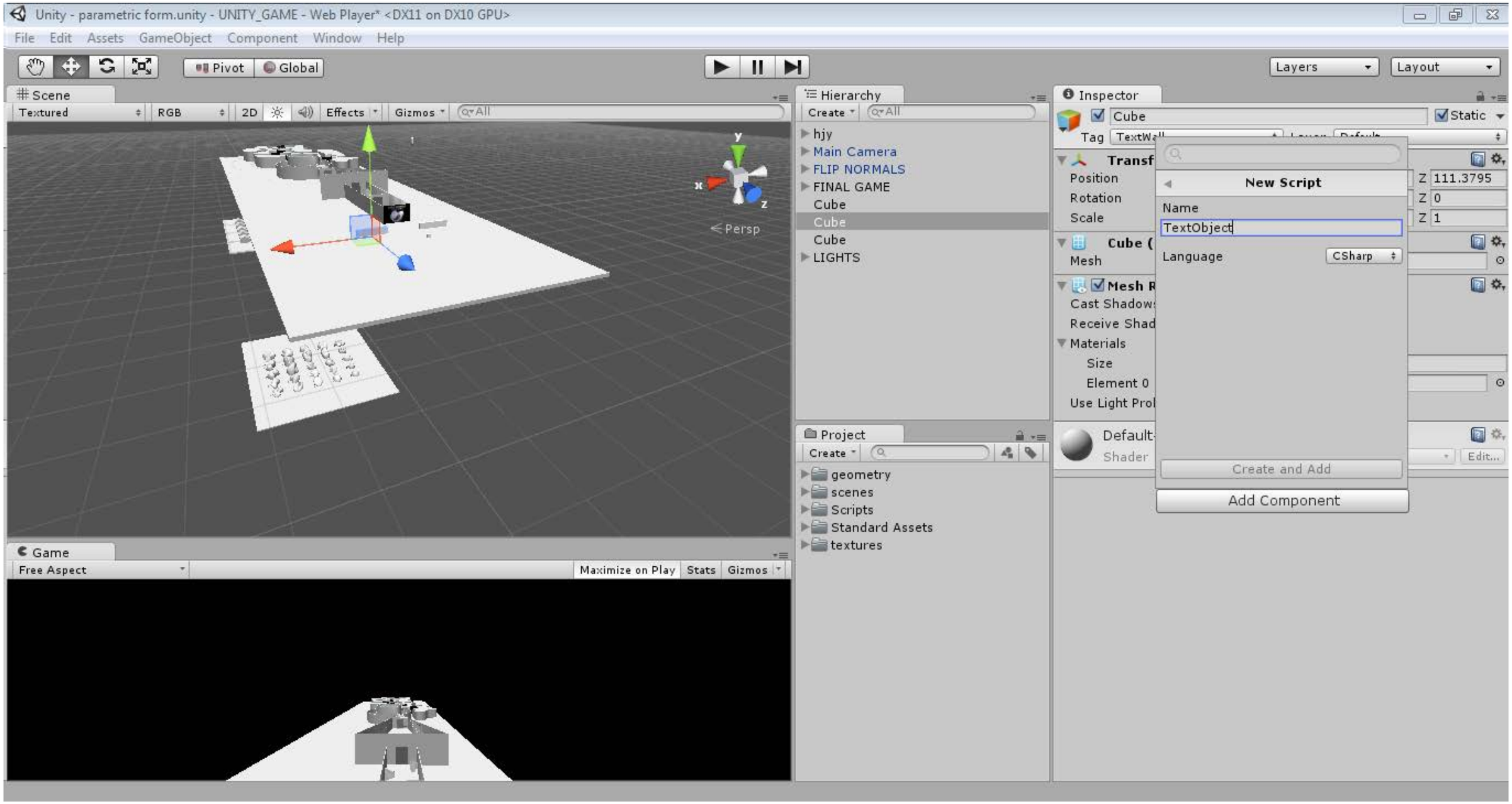
Creating objects that display a message when the player collides with them

We create the tag "TextWall" to differentiate objects
that contain information from those that don't.



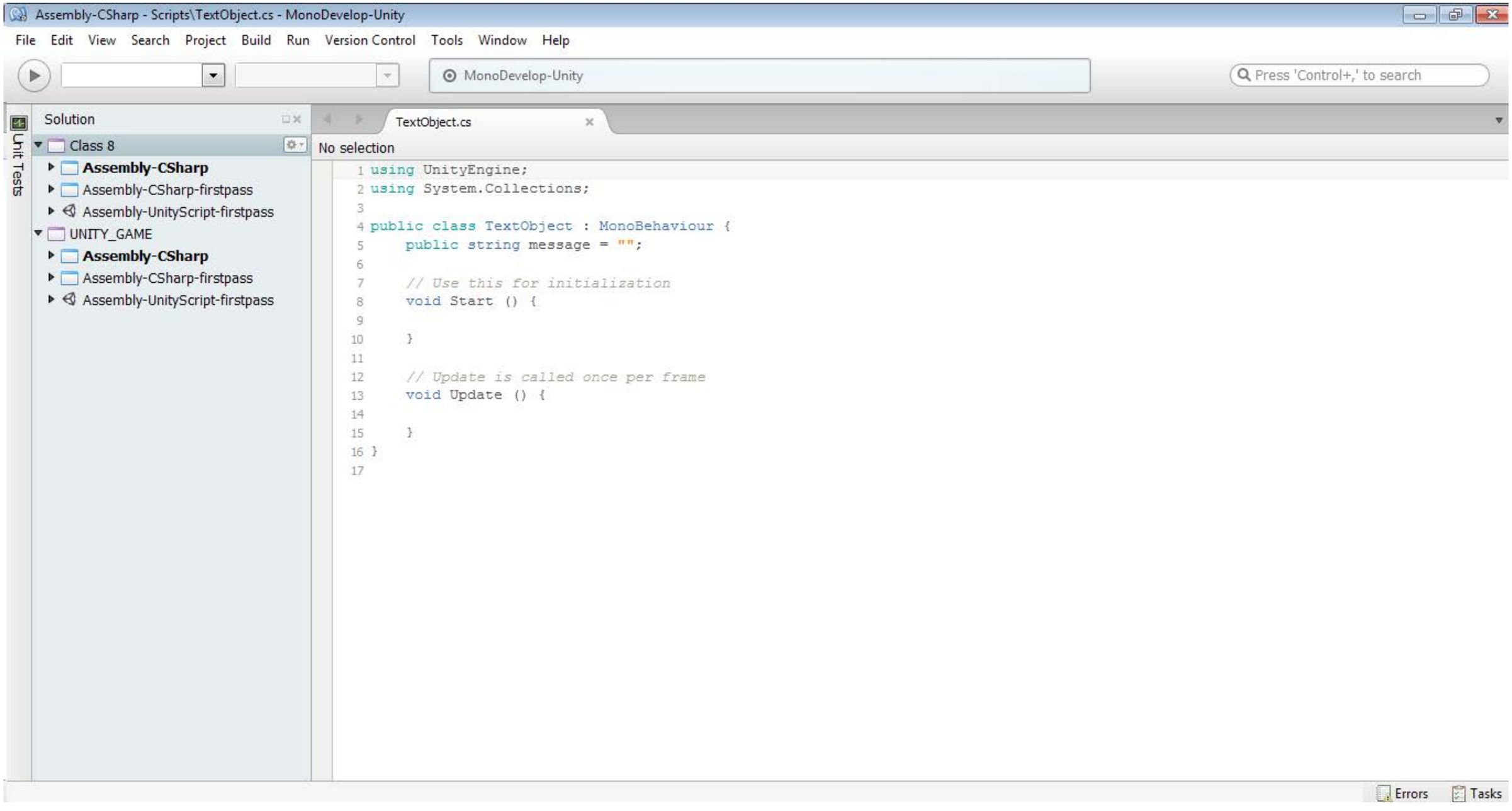
Creating objects that display a message when the player collides with them

We create the script "TextObject" that will store the message to be displayed.



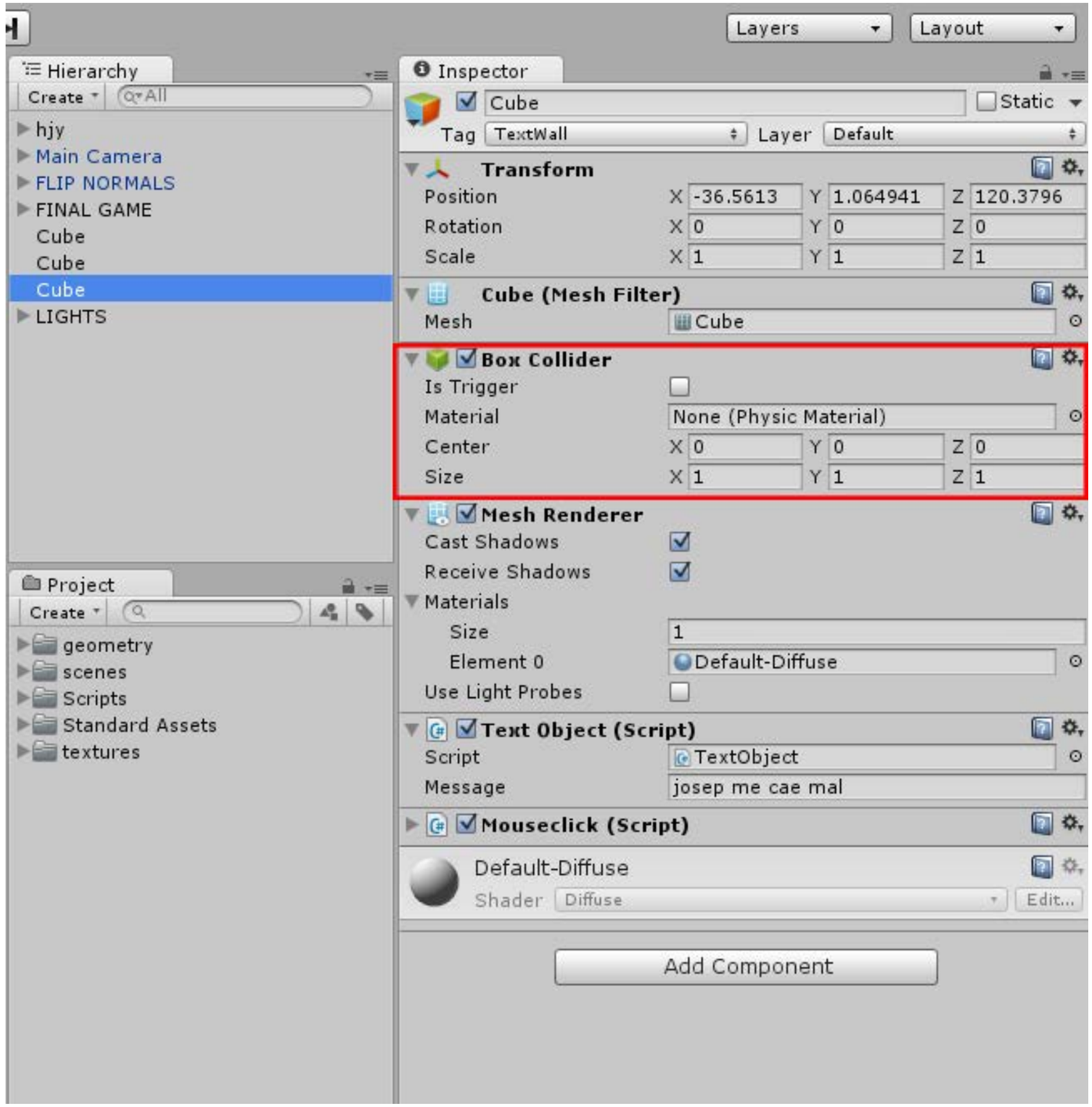
Creating objects that display a message when the player collides with them

The script "TextObject" only contains a public message that can be accessed from the player controller. Every object has its own message.



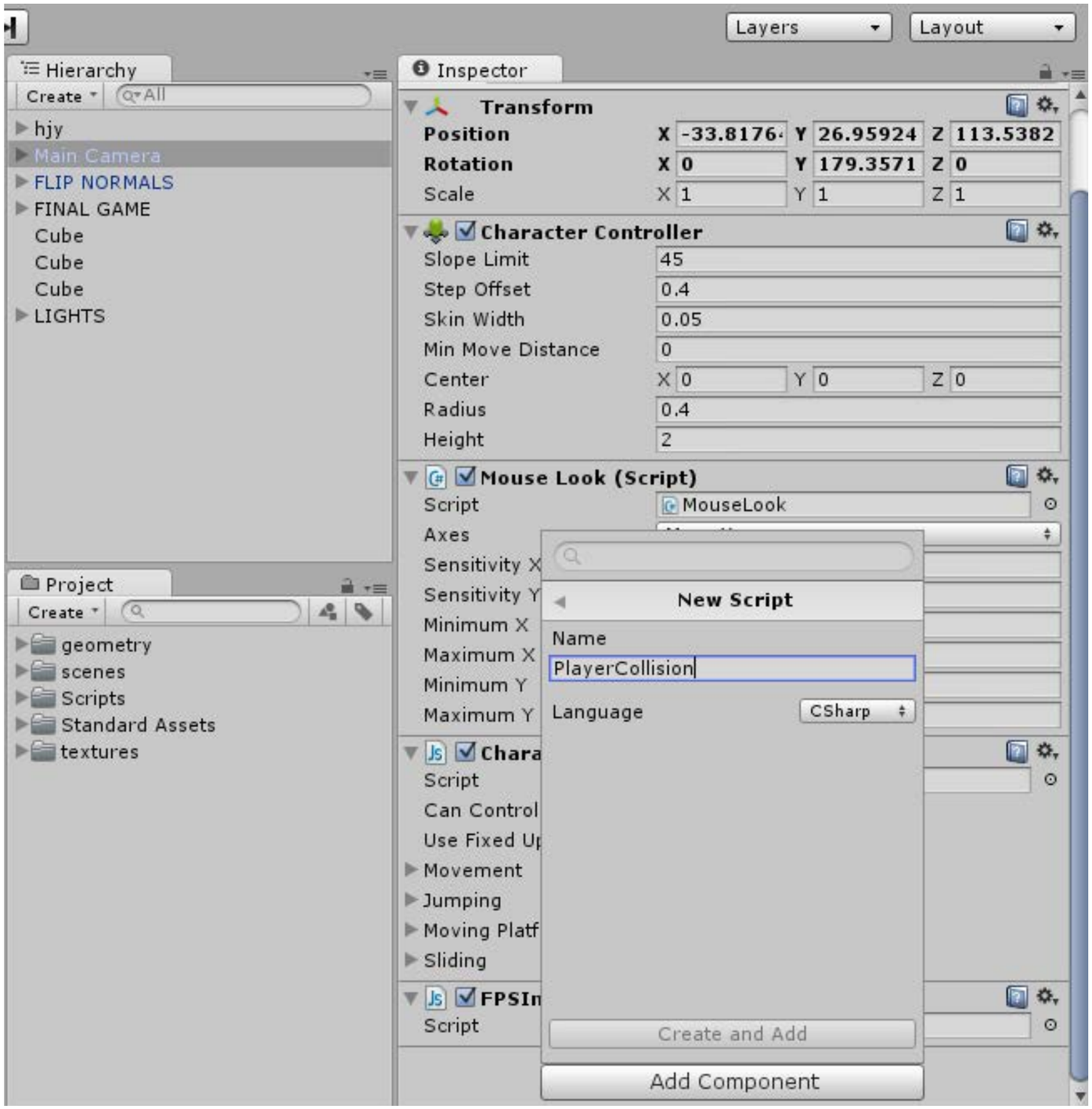
Creating objects that display a message when the player collides with them

Every object from this class must have a proper collider. The best collider is the simplest one that fits the object. For example mesh collider, which is heavier, is useful for highly irregular objects, for simple geometric forms a geometric collider can be used.



Creating objects that display a message when the player collides with them

We create the script "PlayerCollision" that will handle the collision with the objects that contain information.



Iaac | Institute for Advanced Architecture of Catalonia
2014/// Digital Tools

Creating objects that display a message when the player collides with them

"PlayerCollision" script.

Assembly-CSharp - Scripts\PlayerCollision.cs - MonoDevelop-Unity

File Edit View Search Project Build Run Version Control Tools Window Help

MonoDevelop-Unity

Press 'Control+,' to search

Solution

- Class 8
  - **Assembly-CSharp**
  - Assembly-CSharp-firstpass
  - Assembly-UnityScript-firstpass
- UNITY_GAME
  - **Assembly-CSharp**
    - References
    - Scripts
      - color changing.cs
      - PlayerCollision.cs
      - TextObject.cs
  - Assembly-CSharp-firstpass
  - Assembly-UnityScript-firstpass

PlayerCollision.cs

No selection

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class PlayerCollision : MonoBehaviour {
5      private string msg;              // The message that is being displayed
6      private float timeCol;           // Time since the last collision
7      private GameObject collidedObj; // Object that collided
8
9      // Use this for initialization
10     void Start () {
11         msg = "";            // No message to show at the beginning
12         timeCol = 0.0f;      // No previous collision
13
14         GUI.color = new Color(1.0f, 1.0f, 1.0f, 1.0f);
15     }
16
17     // Update is called once per frame
18     void Update () {
19         /* If there is a message and more than 5 seconds have passed
20            since the last collision, clear the previous message
21            and return the collision capability to the wall
22          */
23         if (msg.Length != 0 && Time.time - timeCol >= 5.0){
24             msg = "";
25             collidedObj.collider.enabled = true;
26         }
27     }
28
29     void OnControllerColliderHit(ControllerColliderHit col){
30         // If the object collides with a TextWall (wall with information)
31         if (col.gameObject.tag == "TextWall") {
32
33             // Return the capability to collide to the previous object
```

Toolbox Properties Document Outline Unit Tests

Tasks

Creating objects that display a message when the player collides with them

Three variables are needed: the message, the time since previous collision (to make the message disappear after a established period of time) and the object that collides with the player.

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class PlayerCollision : MonoBehaviour {
5      private string msg;              // The message that is being displayed
6      private float timeCol;           // Time since the last collision
7      private GameObject collidedObj; // Object that collided
8
9      // Use this for initialization
10     void Start () {
11         msg = "";            // No message to show at the beginning
12         timeCol = 0.0f;      // No previous collision
13
14         GUI.color = new Color(1.0f, 1.0f, 1.0f, 1.0f);
15     }
```

At the beginning there is nothing to show, so the message should be empty and the time 0.
GUI.color changes the default color of the font to be fully opaque

```
5      private string msg;              // The message that is being displayed
6      private float timeCol;           // Time since the last collision
7      private GameObject collidedObj; // Object that collided
8
9      // Use this for initialization
10     void Start () {
11         msg = "";            // No message to show at the beginning
12         timeCol = 0.0f;      // No previous collision
13
14         GUI.color = new Color(1.0f, 1.0f, 1.0f, 1.0f);
15     }
16
17     // Update is called once per frame
18     void Update () {
19         /* If there is a message and more than 5 seconds have passed
20            since the last collision. clear the previous message
```

Creating objects that display a message when the player collides with them

If we are displaying a message and more than 5 seconds have passed since the last collision, the message is cleared and the other object's collider are activated again.
The collider is deactivated when the player hits the object in order to let the player pass through the object.

```
14          GUI.color = new Color(1.0f, 1.0f, 1.0f, 1.0f);
15      }
16
17      // Update is called once per frame
18      void Update () {
19          /* If there is a message and more than 5 seconds have passed
20             since the last collision, clear the previous message
21             and return the collision capability to the wall
22           */
23          if (msg.Length != 0 && Time.time - timeCol >= 5.0){
24              msg = "";
25              collidedObj.collider.enabled = true;
26          }
27      }
28
29      void OnControllerColliderHit(ControllerColliderHit col){
```

Creating objects that display a message when the player collides with them

This is the function that controls the collision between the player and other objects. It reacts when the other object has the tag "TexWall". This "if" condition ensures that the following code is applied only to objects that store a message.

```
27      }
28
29      void OnControllerColliderHit(ControllerColliderHit col){
30          // If the object collides with a TextWall (wall with information)
31          if (col.gameObject.tag == "TextWall") {
32
33              // Return the capability to collide to the previous object
34              if ( msg.Length != 0 ){
35                  collidedObj.collider.enabled = true;
36              }
37
38              // Update the information
39              var box = col.gameObject;
40              msg = box.GetComponent<TextObject> ().message;  // Retrieve the message
41              collidedObj = col.gameObject;                   // Update the colliding object
42              collidedObj.collider.enabled = false;           // Disable the other object's collider
43              timeCol = Time.time;                            // Update the time
44
45          }
46      }
47
48      void OnGUI() {
```

Creating objects that display a message when the player collides with them

The message contained in the object is retrieved and stored, the object's collider is disabled and the time since the last collision is updated.

```
34              if ( msg.Length != 0 ){
35                  collidedObj.collider.enabled = true;
36              }
37
38              // Update the information
39              var box = col.gameObject;
40              msg = box.GetComponent<TextObject> ().message;  // Retrieve the message
41              collidedObj = col.gameObject;                   // Update the colliding object
42              collidedObj.collider.enabled = false;           // Disable the other object's collider
43              timeCol = Time.time;                            // Update the time
44
45          }
46      }
47
48      void OnGUI() {
```

If the message of another object is being displayed, the collider is enabled again to allow future collisions.

```
27      }
28
29      void OnControllerColliderHit(ControllerColliderHit col){
30          // If the object collides with a TextWall (wall with information)
31          if (col.gameObject.tag == "TextWall") {
32
33              // Return the capability to collide to the previous object
34              if ( msg.Length != 0 ){
35                  collidedObj.collider.enabled = true;
36              }
37
38              // Update the information
39              var box = col.gameObject;
40              msg = box.GetComponent<TextObject> ().message;  // Retrieve the message
41              collidedObj = col.gameObject;                   // Update the colliding object
```

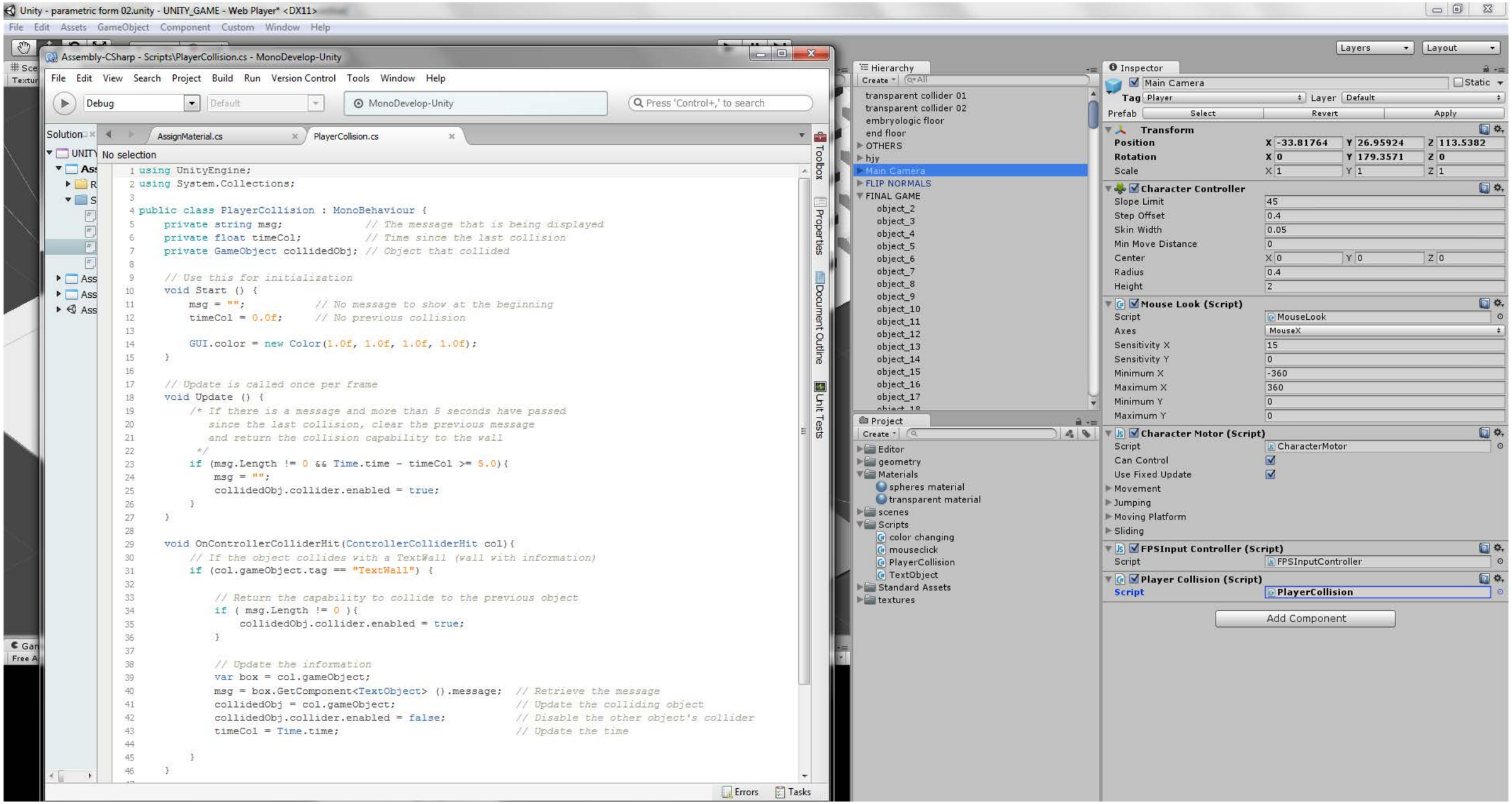
Creating objects that display a message when the player collides with them

The message is displayed is displayed in the upper left of the screen (starts in 20,20 and ends in 1/4 of the screen width horizontally and 1/2 of the screen height vertically.

```
            var box = col.gameObject;
            msg = box.GetComponent<TextObject> ().message;   // Retrieve the message
            collidedObj = col.gameObject;                    // Update the colliding object
            collidedObj.collider.enabled = false;            // Disable the other object's collider
            timeCol = Time.time;                             // Update the time

        }
    }

    void OnGUI() {
        if (msg.Length != 0) {
            GUI.Box (new Rect (20, 20, Screen.width / 4, Screen.height / 2), msg);
        }
    }
}
```

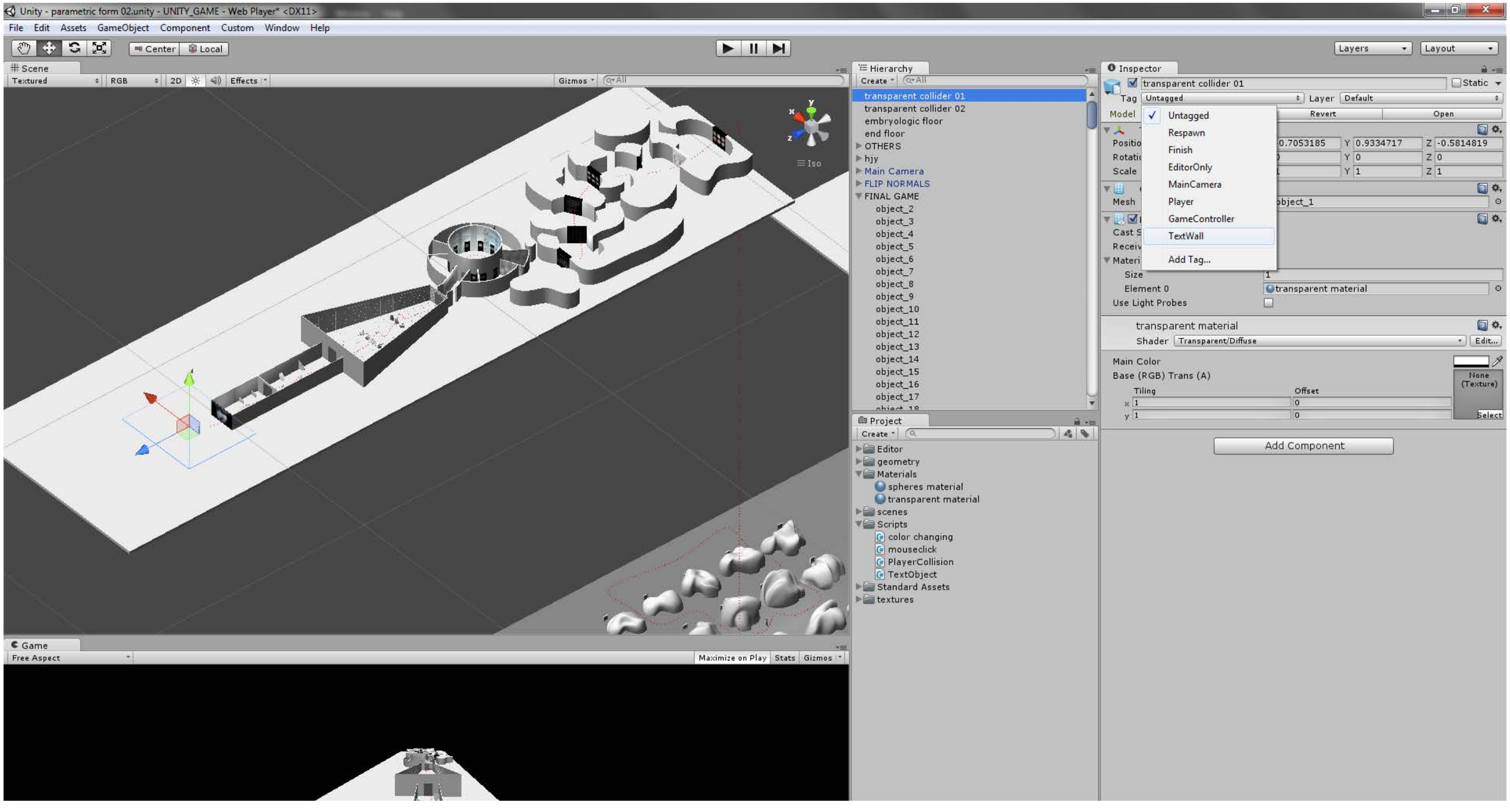
Creating objects that display a message when the player collides with them

We apply the PlayerCollision script to the camera.



Creating objects that display a message when the player collides with them

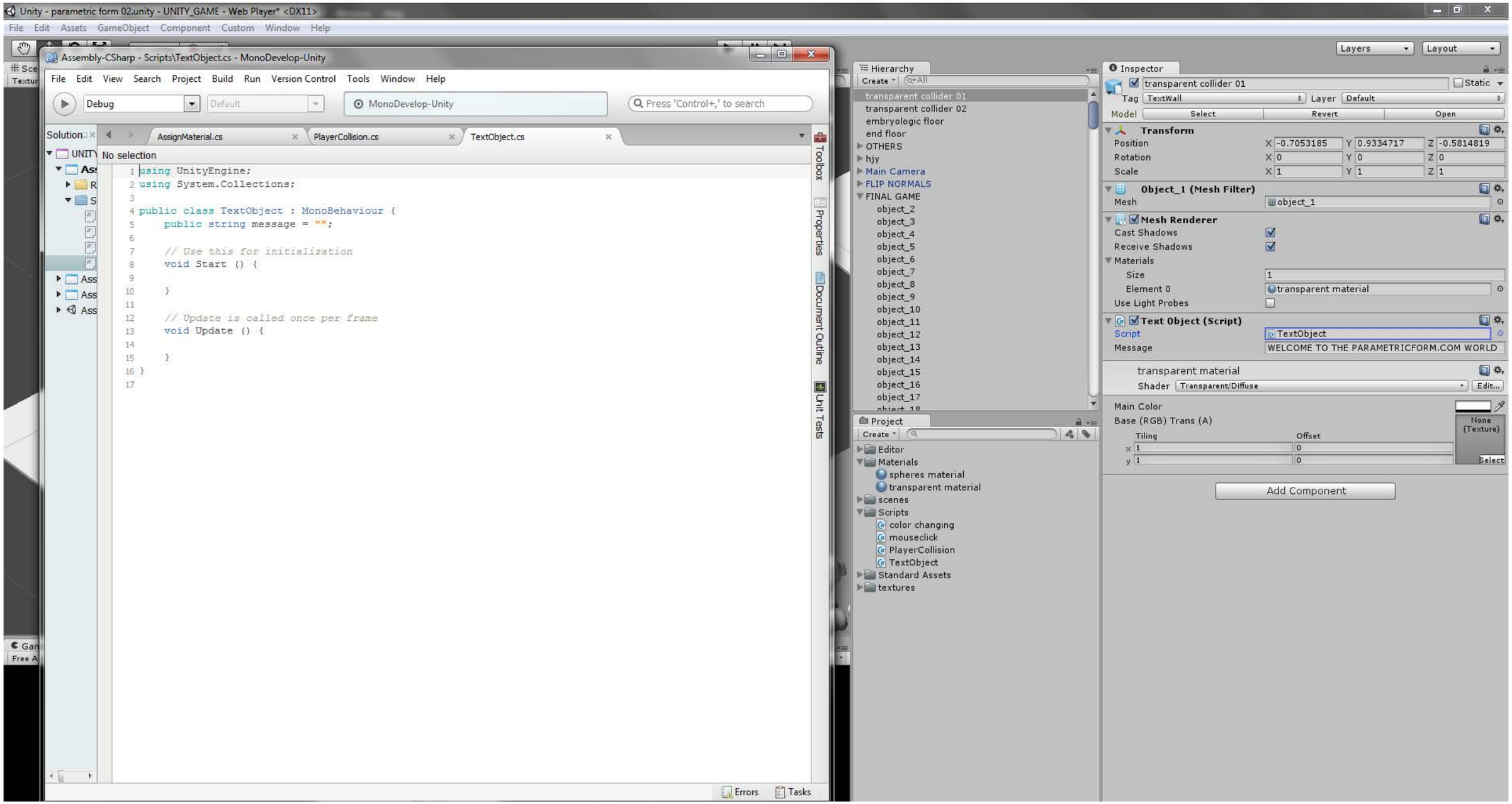
We tag the object that we want to contain a message with the ¨Text Wall¨ tag.



Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Creating objects that display a message when the player collides with them

We apply the ¨TextObject¨ script to the chosen objects and write the message in the Inspector.
We don´t need to open the script because the message is a public variable.



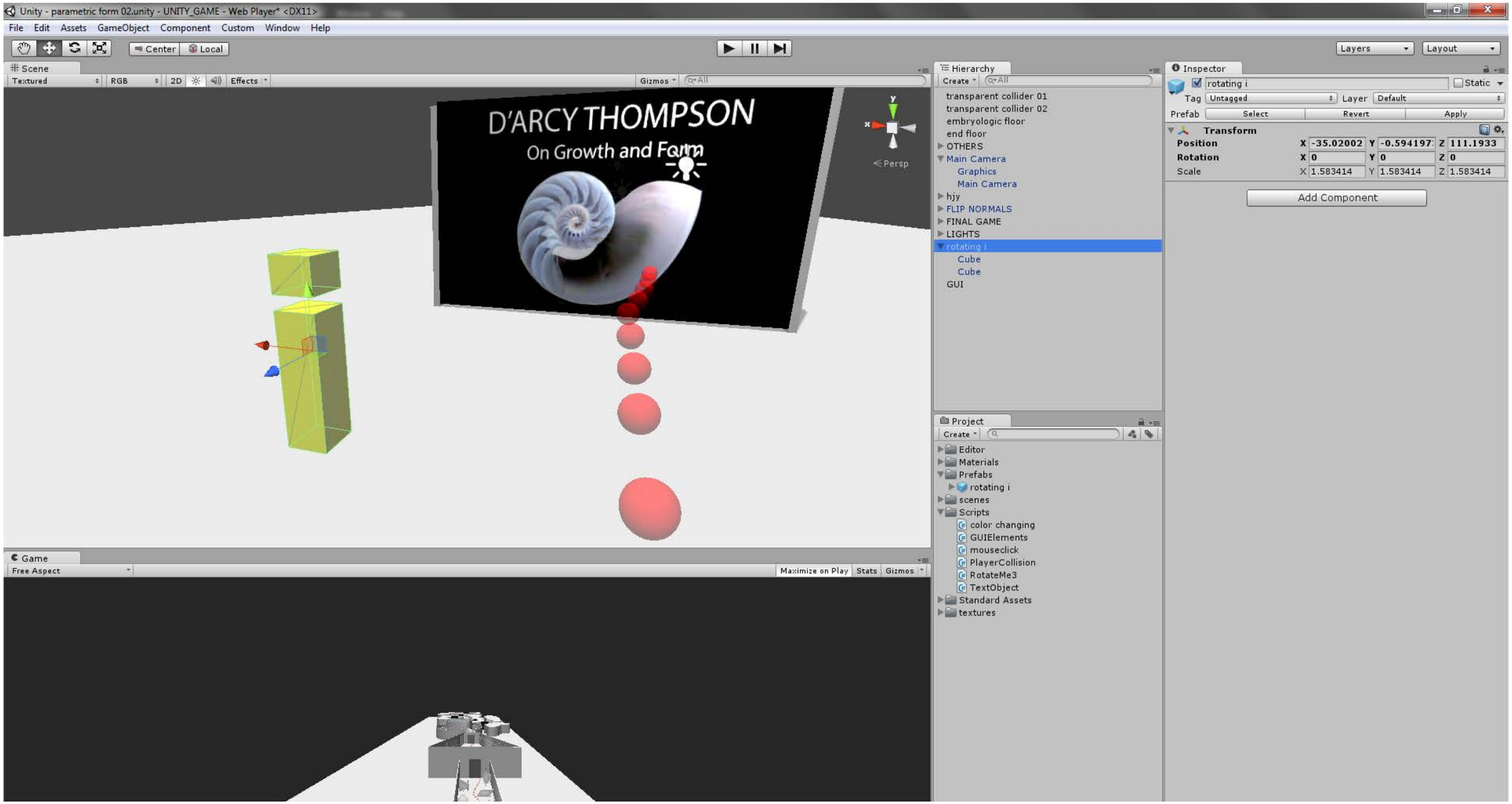
Iaac | Institute for Advanced Architecture of Catalonia
2014/// Digital Tools

Assignment 3 - Gameify! (Unity 3D)

Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Creating the information points

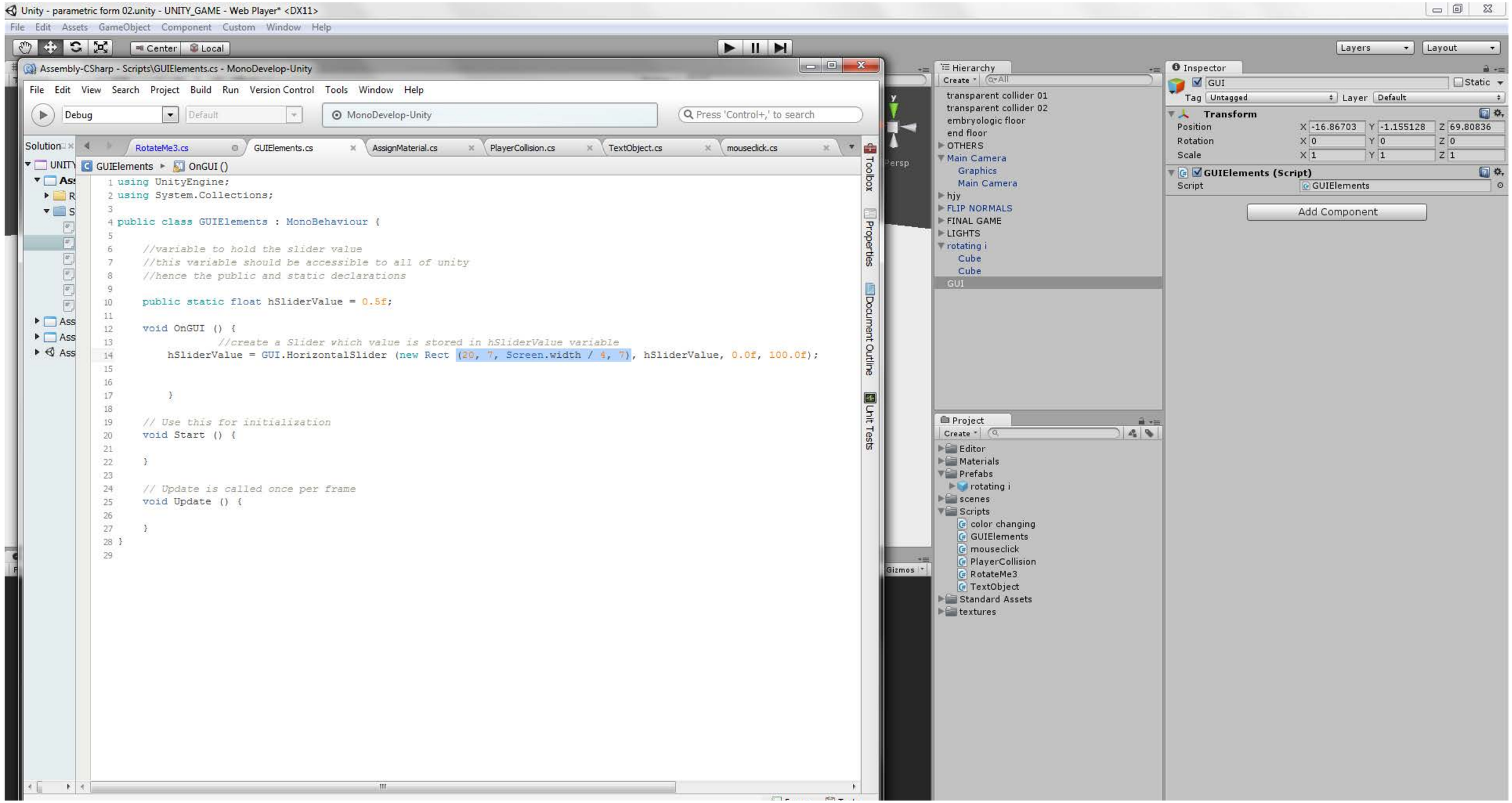
The information points are 3D floating rotating ¨i¨ with 3 features.
1. They rotate around and the rotation speed can be controlled with the slider. 2. They display a message when the player collides with them. 3. Their color changes when we hover the mouse over them.



Creating the information points

ROTATION

We create the GUIElements script and apply it to an empty Game Object.
The script creates a slider and stores the value of the slider in a public static variable.
We match the size of the slider with the size of the text box that displays the messages.

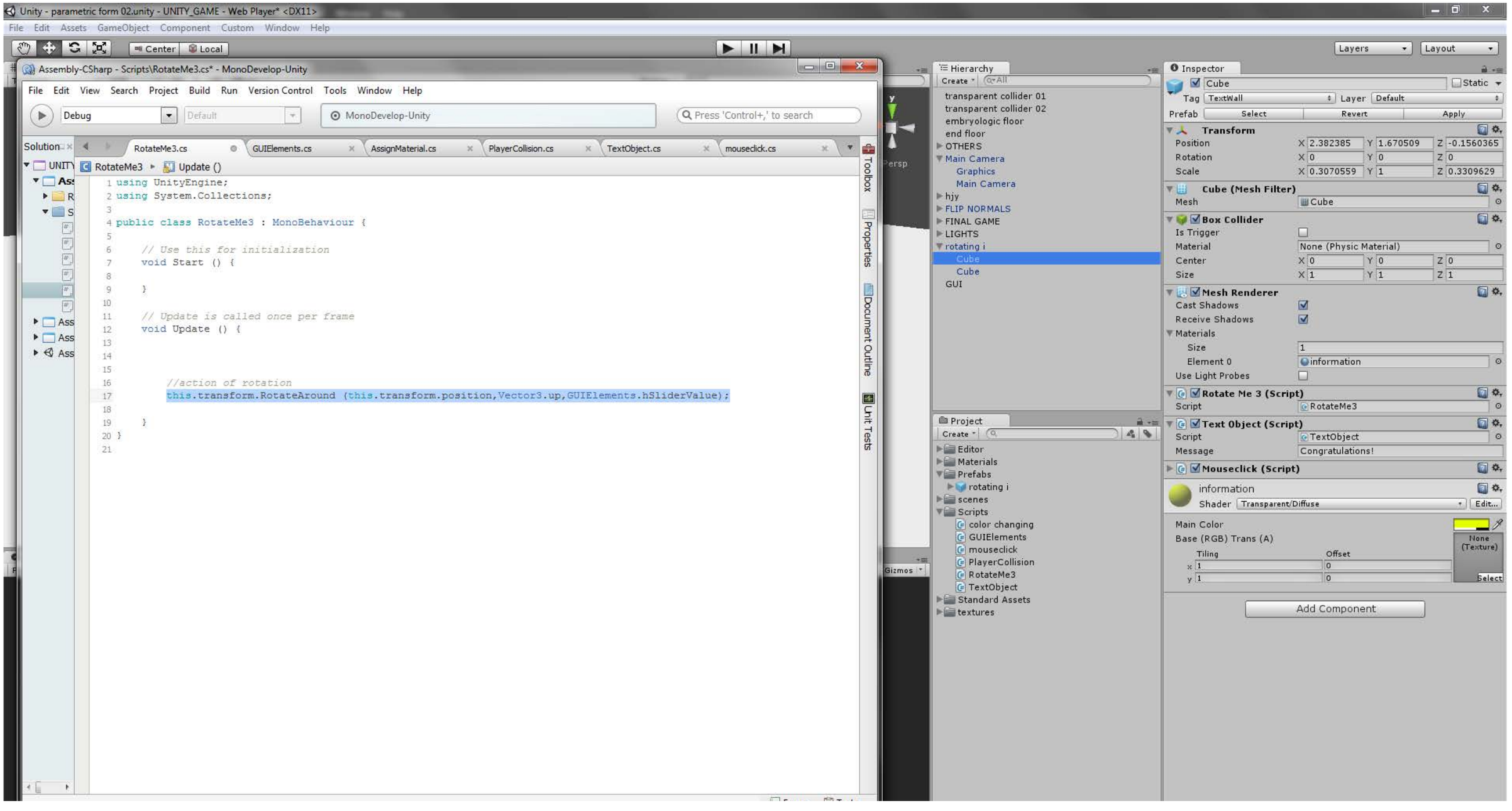


Creating the information points

ROTATION

We create the RotateMe script and apply it to the ï¨ object.

The rotation speed is controlled by the value of the slider.

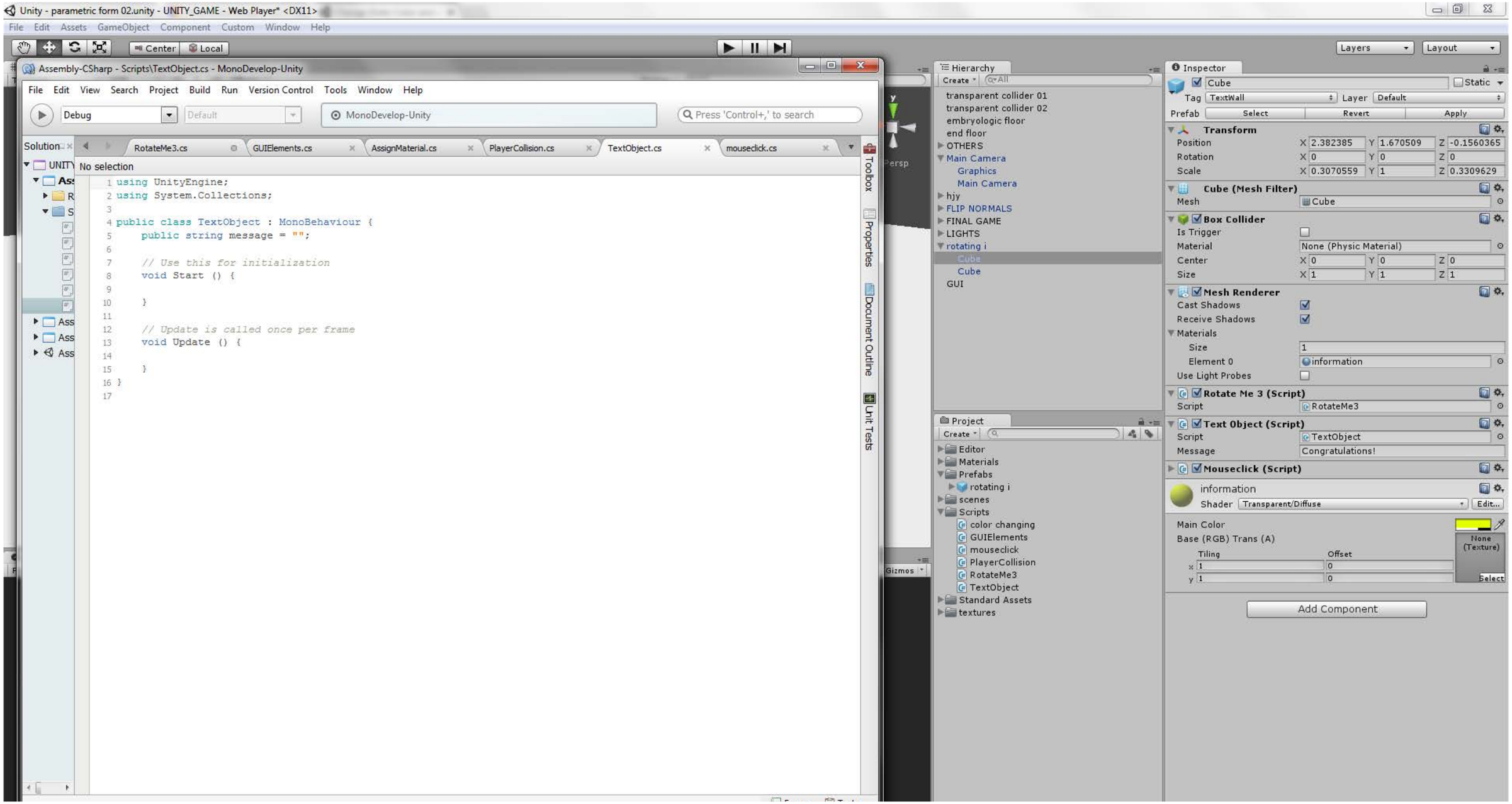


Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Creating the information points

DISPLAY MESSAGE

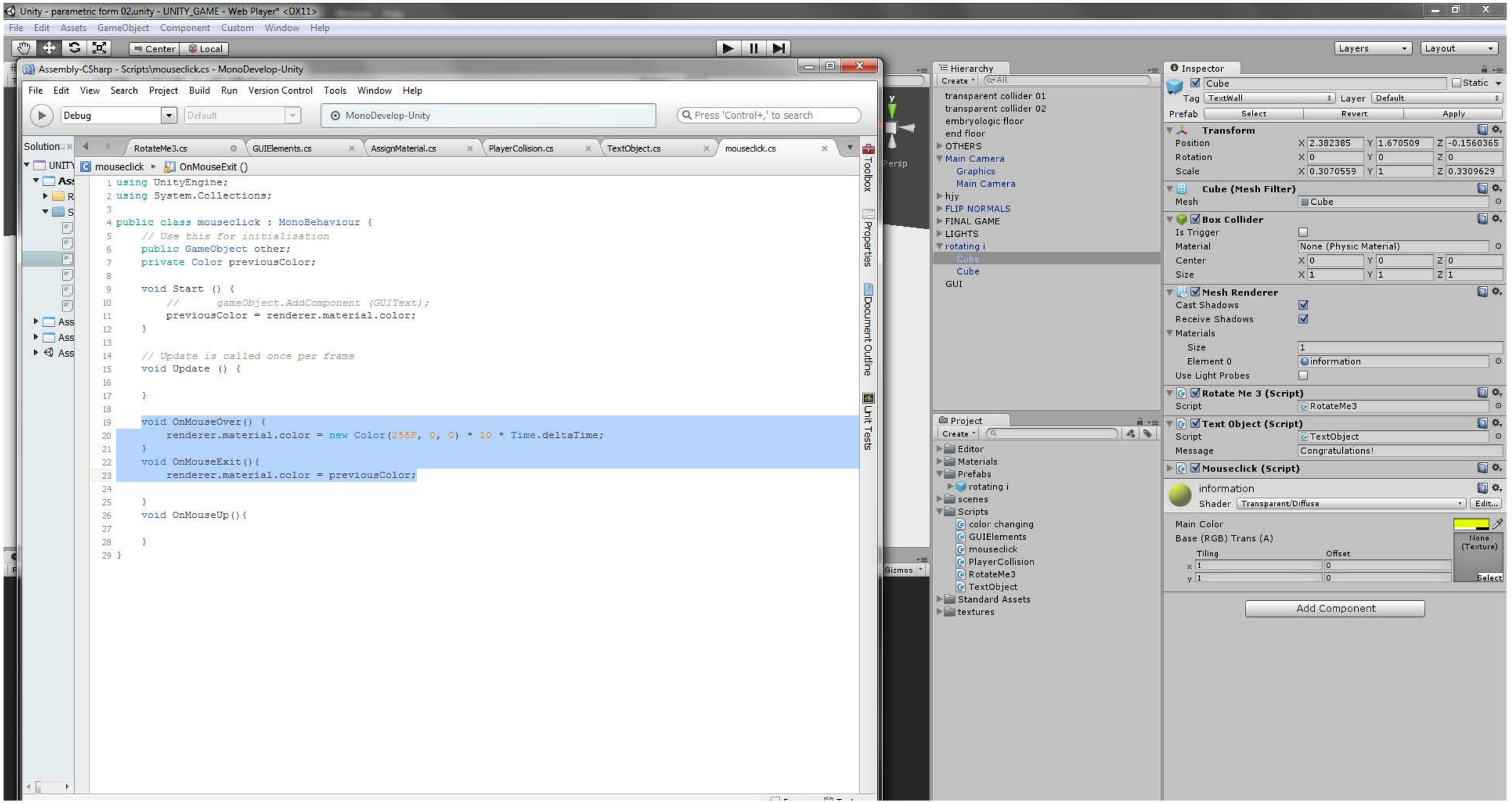
We apply the TextObject script to the ¨i¨.



Creating the information points

CHANGING COLOR

We apply the Mouseclick script to the ¨i¨object.



Creating the information points

CHANGING COLOR

Two actions are considered in this script: mouse over and mouse exit.
When the mouse is on an object, it changes its color.

```
18      void OnMouseOver() {
19          renderer.material.color -= new Color(0.1F, 0, 0) * 10 * Time.deltaTime;

21      void OnMouseExit(){
22          renderer.material.color = previousColor;
```

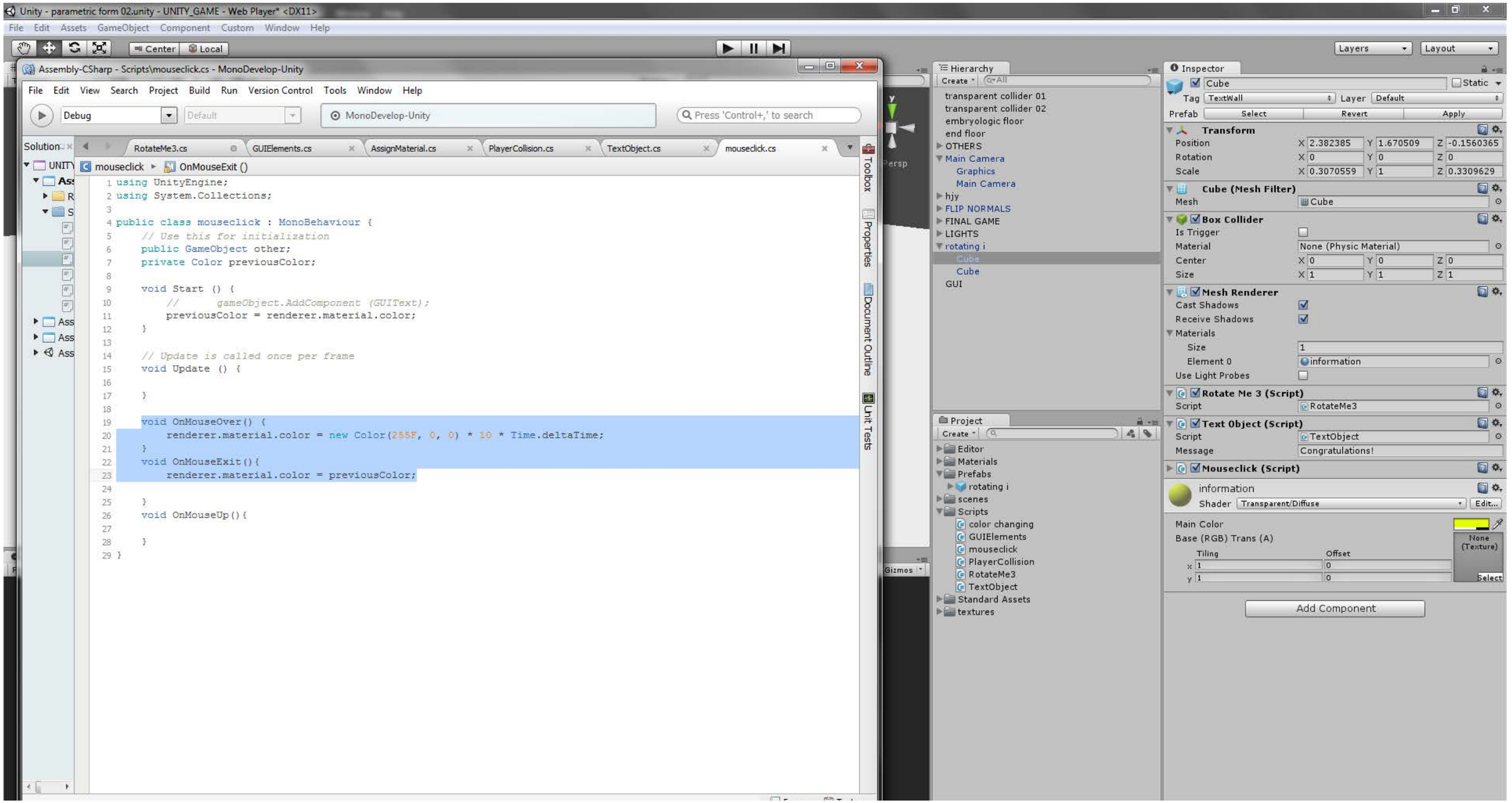
When the mouse leaves the object, the object changes to its original color.
To achieve this we saved its previous color before.

```
7       private Color previousColor;
8
9       void Start () {
10
11          previousColor = renderer.material.color;
12      }
```

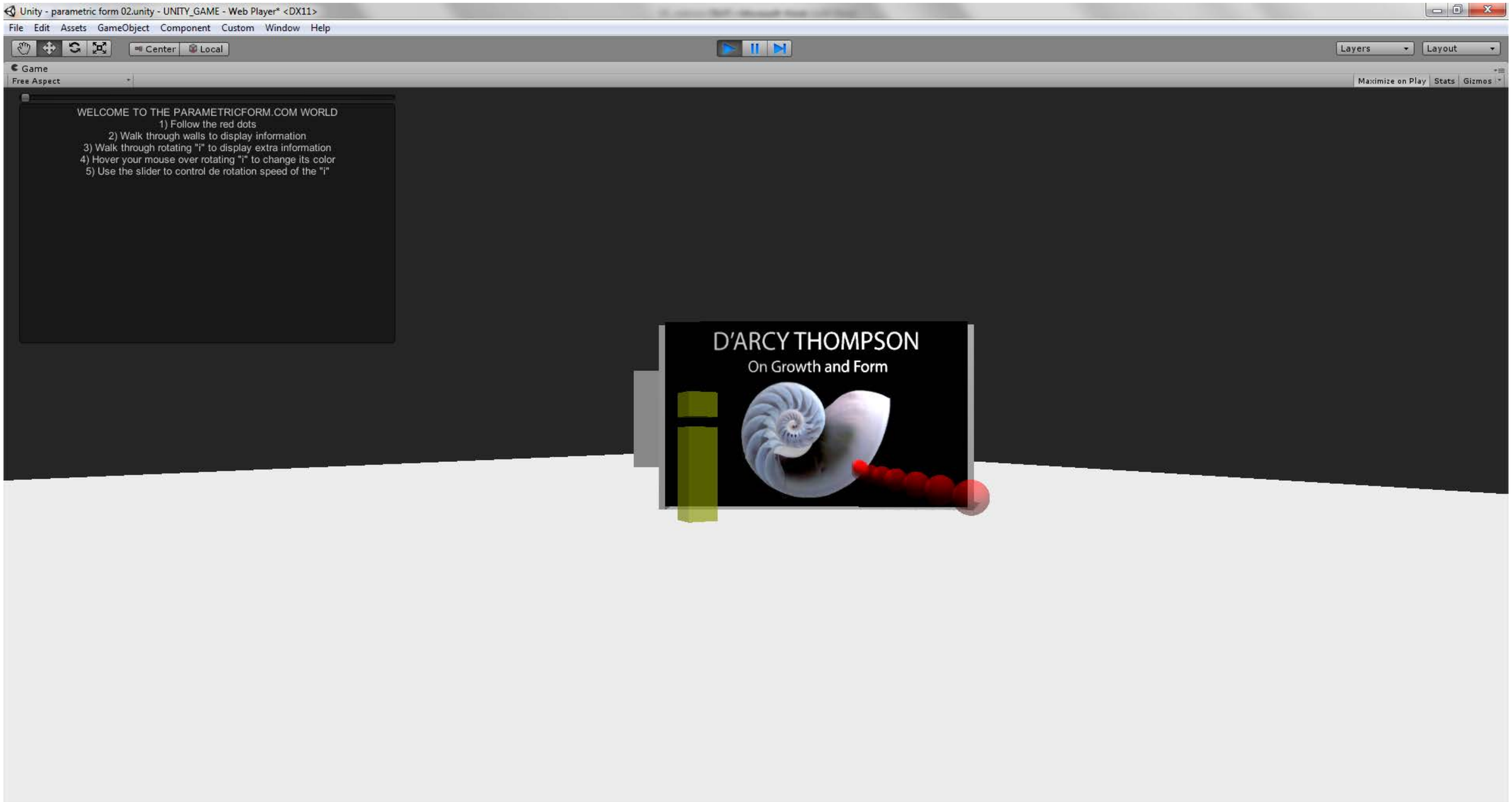
Creating the information points

CHANGING COLOR

We apply the Mouseclick script to the ¨i¨object.

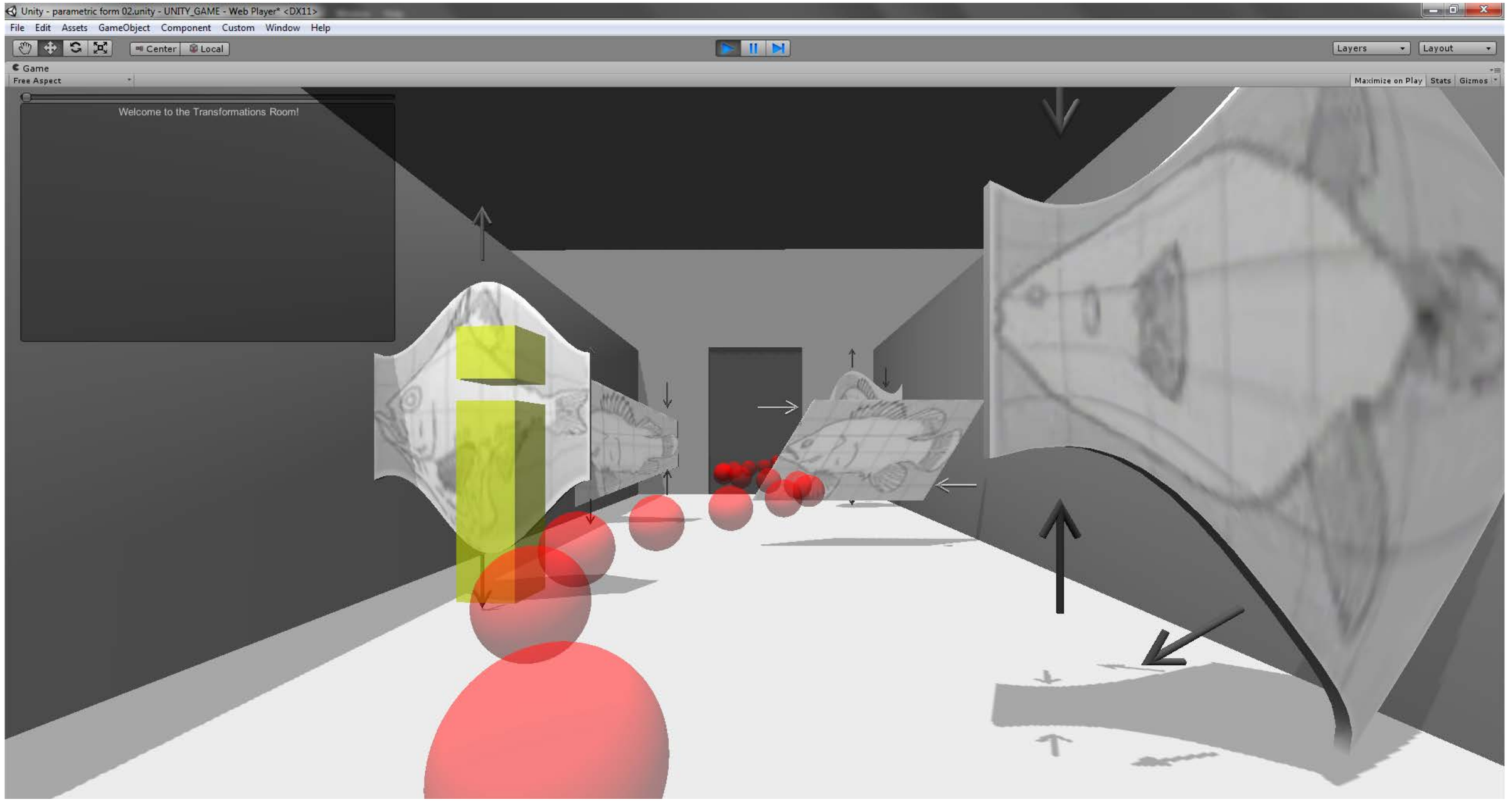


Playing the game in Unity



Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

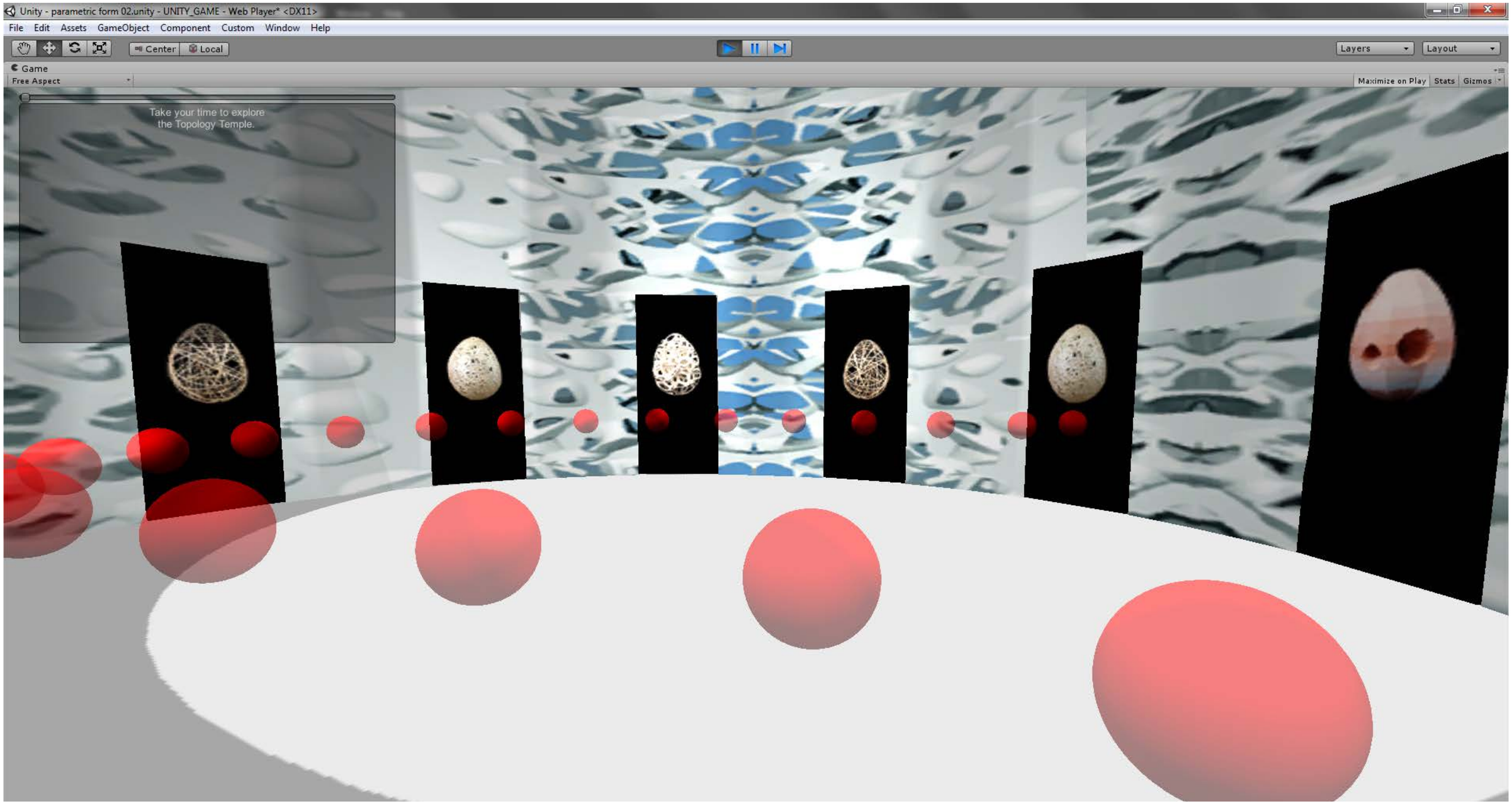
Playing the game in Unity



Assignment 3 - Gameify! (Unity 3D)

Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Playing the game in Unity

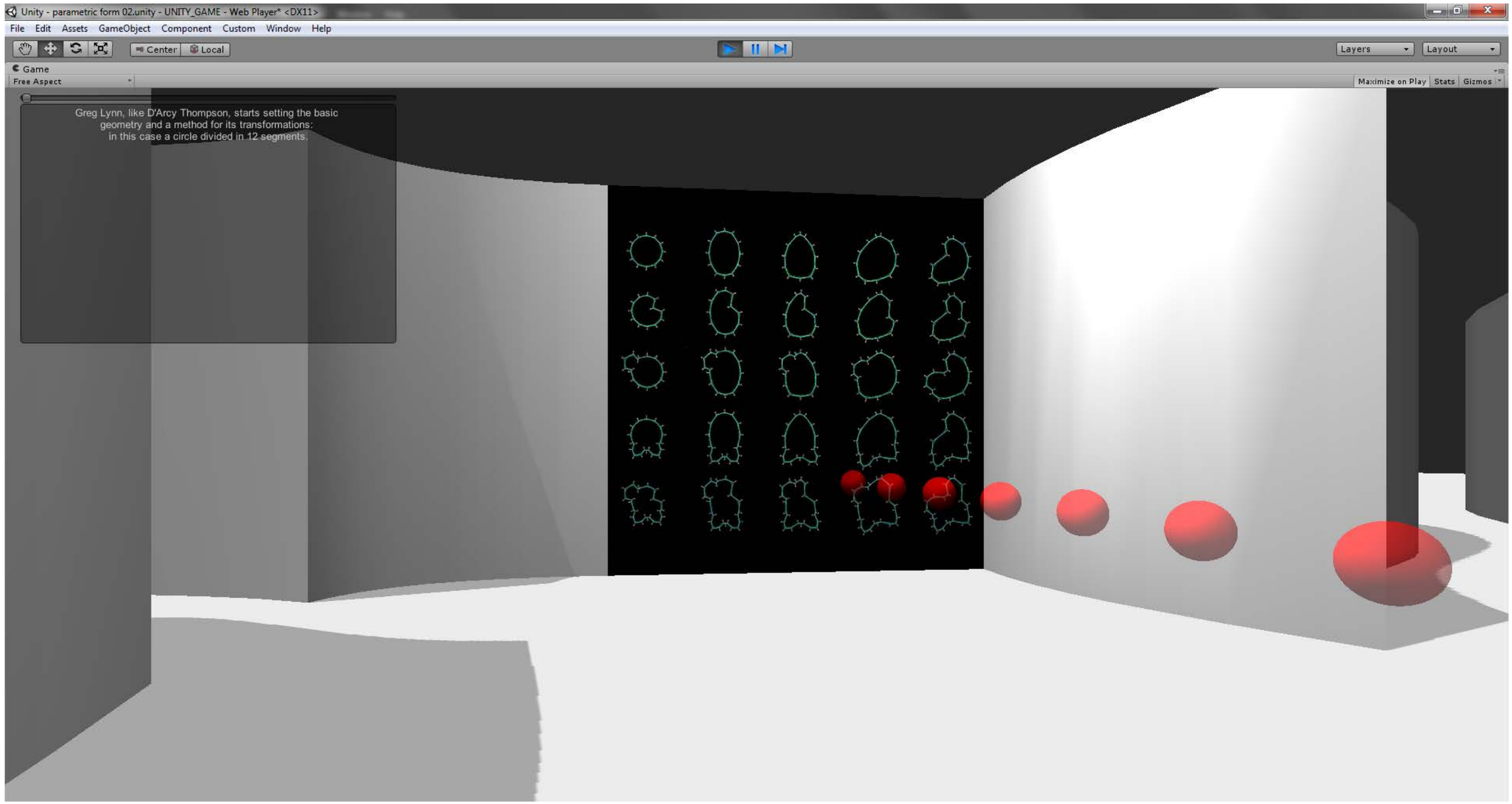


Institute for Advanced Architecture of Catalonia
2014/// Digital Tools

Assignment 3 - Gameify! (Unity 3D)

Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

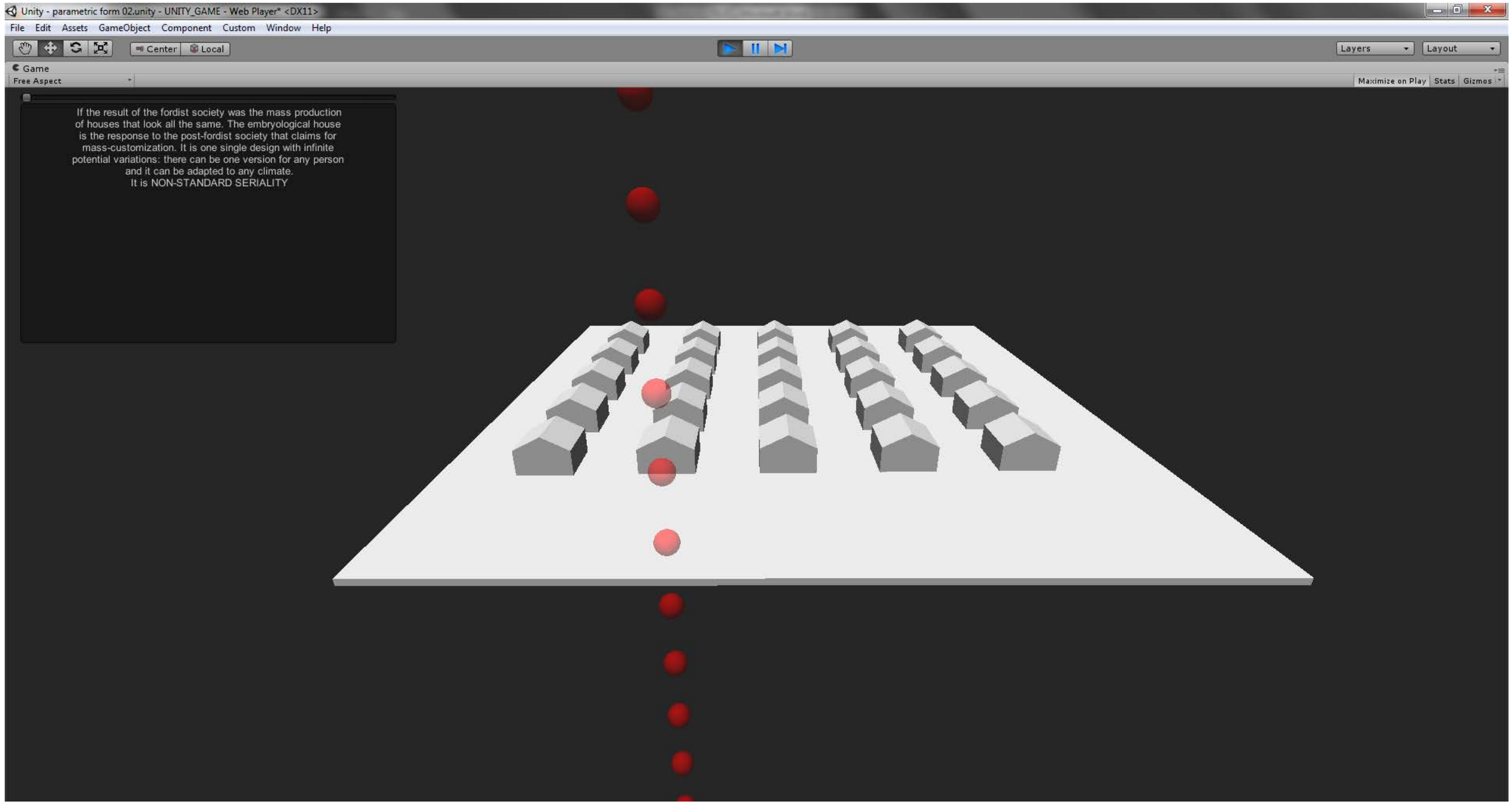
Playing the game in Unity



Assignment 3 - Gameify! (Unity 3D)

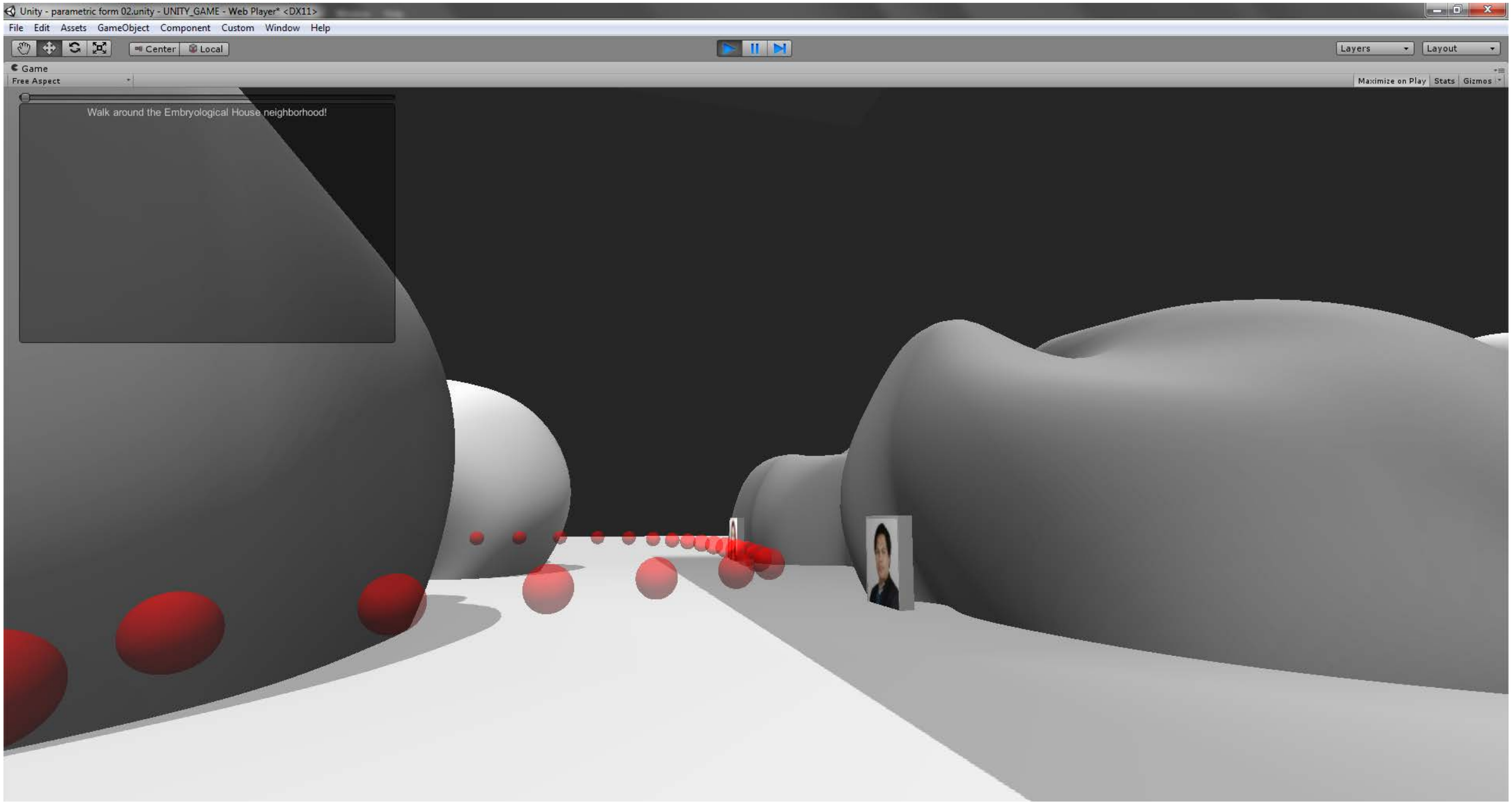
Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Playing the game in Unity



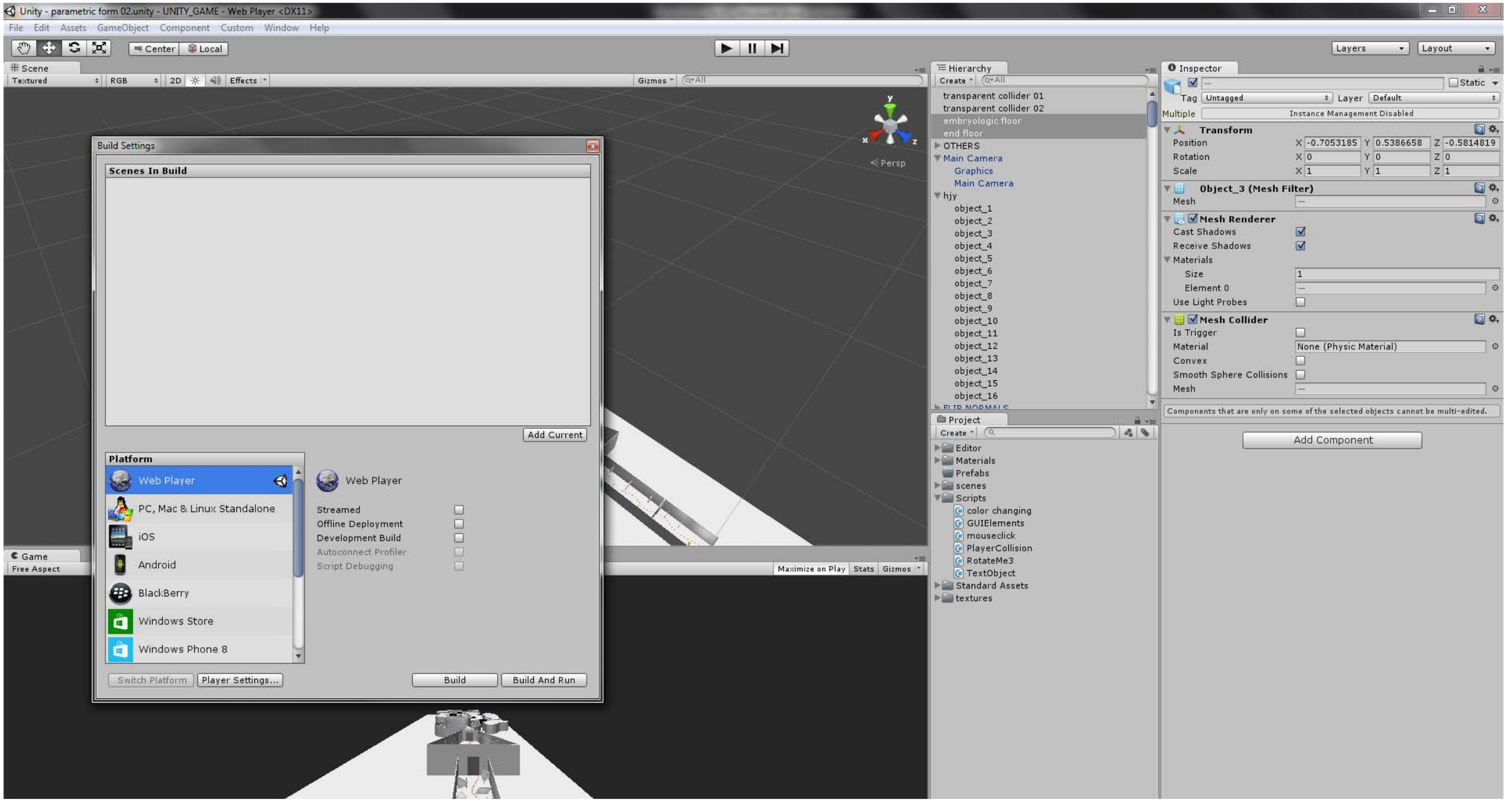
Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Playing the game in Unity



Exporting the game for the web

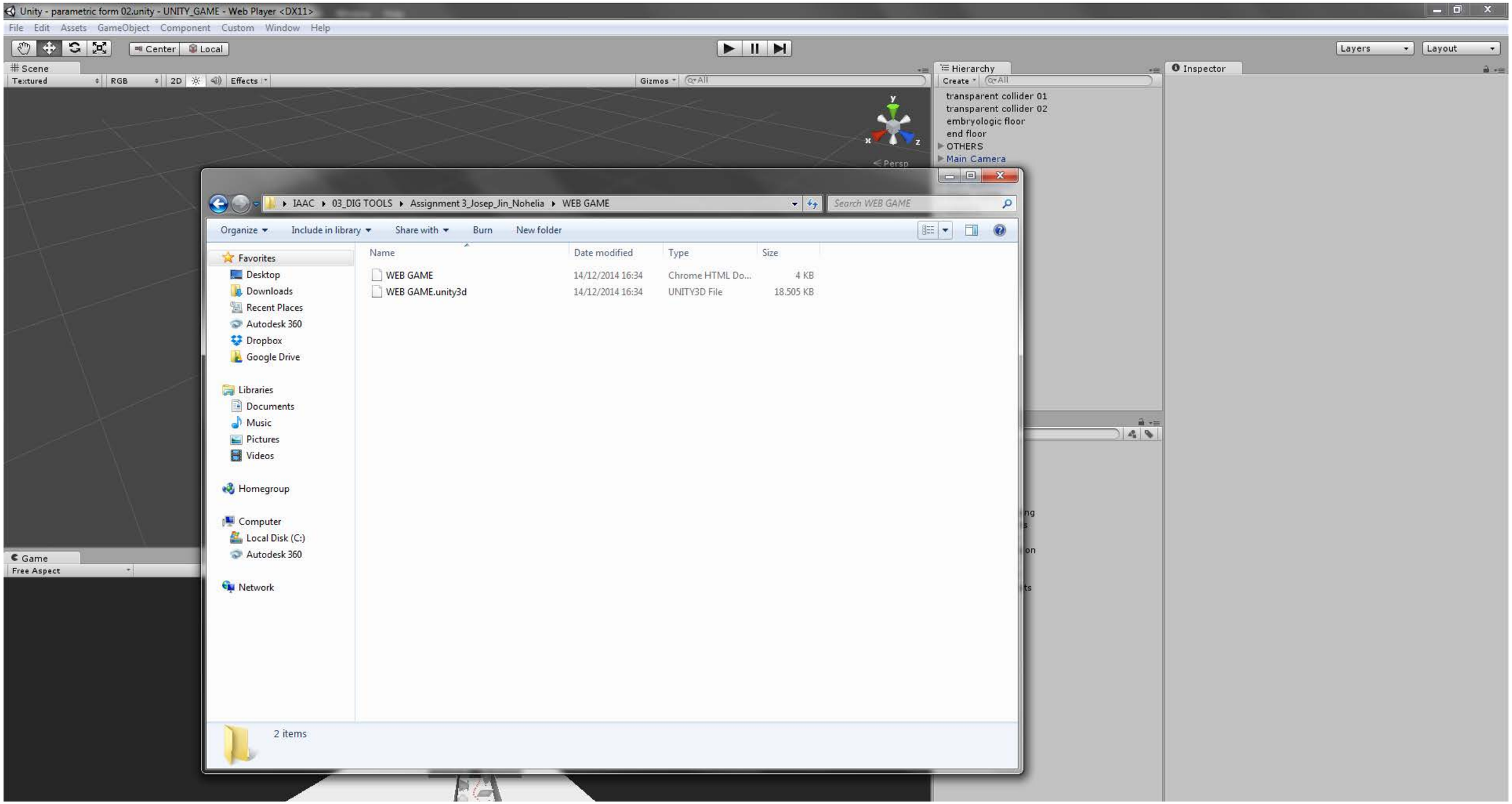
We build the game using Web Player format.



Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

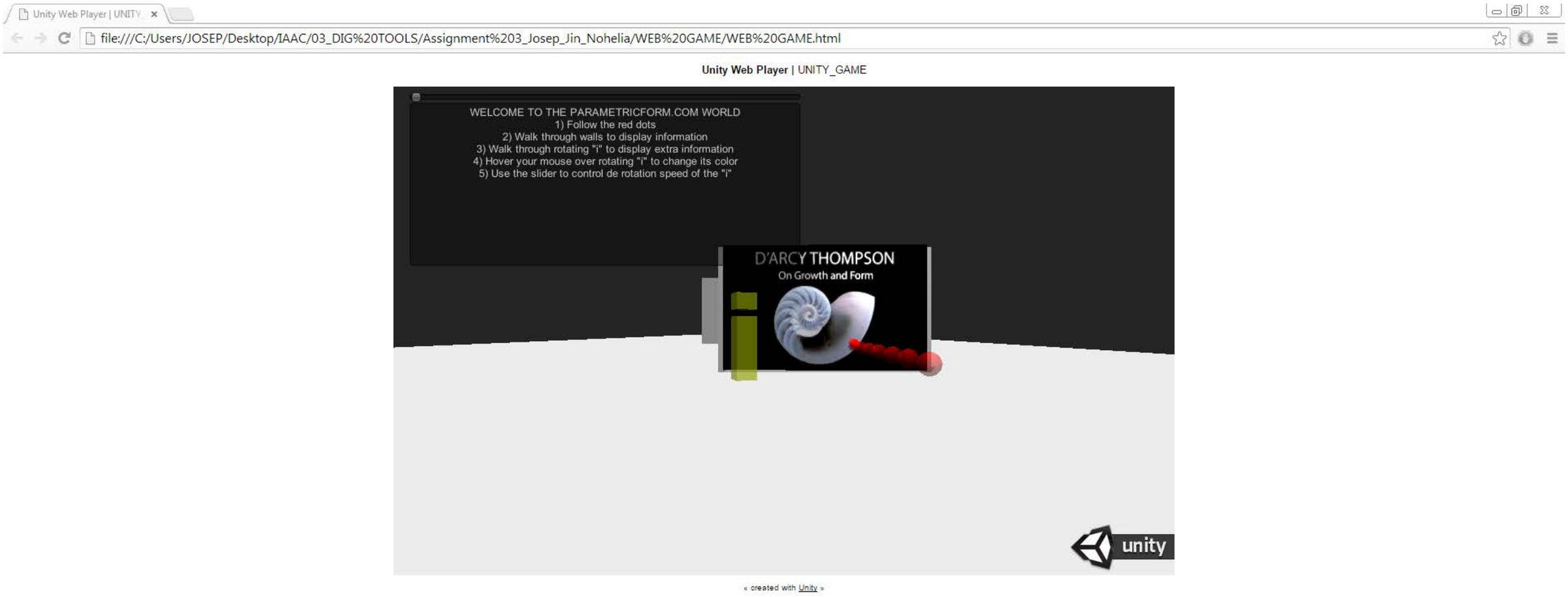
Exporting the game for the web

Two files are created.
One .html file and one .unity3d file.
They have to be in the same location in order to be able to run the game.



Playing the game from the computer with a web browser

The .html file can be opened with a web browser that support WebGL.
Despite running it with a web browser the game is not online.



Changing the background color of the .html file

We open the .html file of the game with a code editor and change the background color from white to black.
Therefore the black texts won´t be displayed and the page will look cleaner.

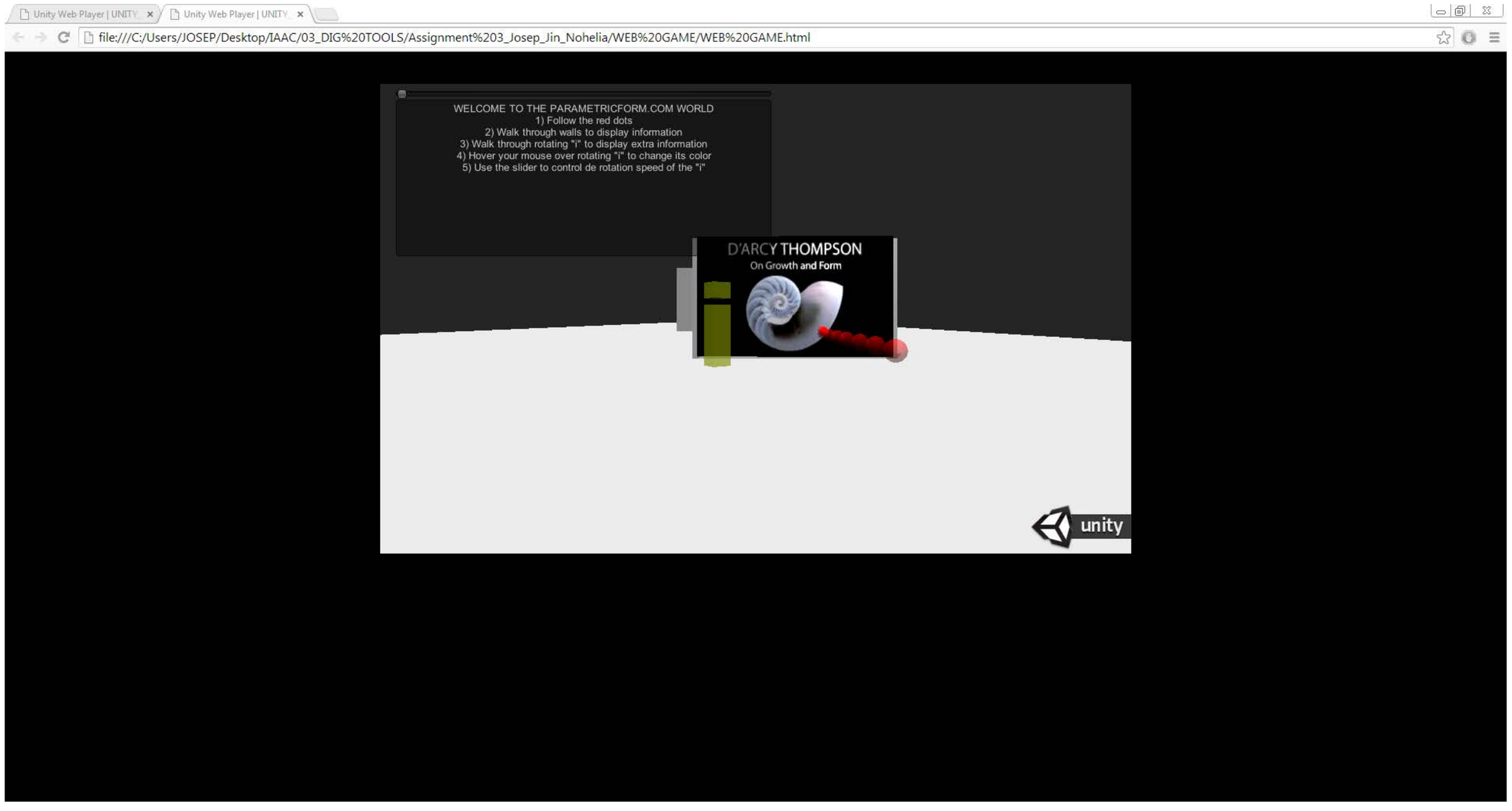
*C:\Users\JOSEP\Desktop\IAAC\03_DIG TOOLS\Assignment 3_Josep_Jin_Nohelia\WEB GAME\WEB GAME.html - Notepad++ [Administrator]

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

index.html | work.html | mobile_home.html | mobile_work.html | mobile_spherical.html | mobile_pannellum.html | cerabox.css | parametricform.html | parametricinstructions.html | parametricgame.html | WEB GAME.html

```
                        case "missing":
                            $missingScreen.find("a").click(function (e) {
                                e.stopPropagation();
                                e.preventDefault();
                                u.installPlugin();
                                return false;
                            });
                            $missingScreen.show();
                        break;
                        case "installed":
                            $missingScreen.remove();
                        break;
                        case "first":
                        break;
                    }
                });
                u.initPlugin(jQuery("#unityPlayer")[0], "WEB GAME.unity3d");
            });
        -->
        </script>
        <style type="text/css">
        <!--
        body {
            font-family: Helvetica, Verdana, Arial, sans-serif;
            background-color: black;
            color: black;
            text-align: center;
        }
        a:link, a:visited {
            color: #000;
        }
        a:active, a:hover {
            color: #666;
        }
        p.header {
            font-size: small;
        }
        p.header span {
            font-weight: bold;
        }
        p.footer {
            font-size: x-small;
        }
        div.content {
            margin: auto;
            width: 960px;
        }
        div.broken,
        div.missing {
            margin: auto;
            position: relative;
```

Changing the background color of the .html file



Creating other .html pages for the website

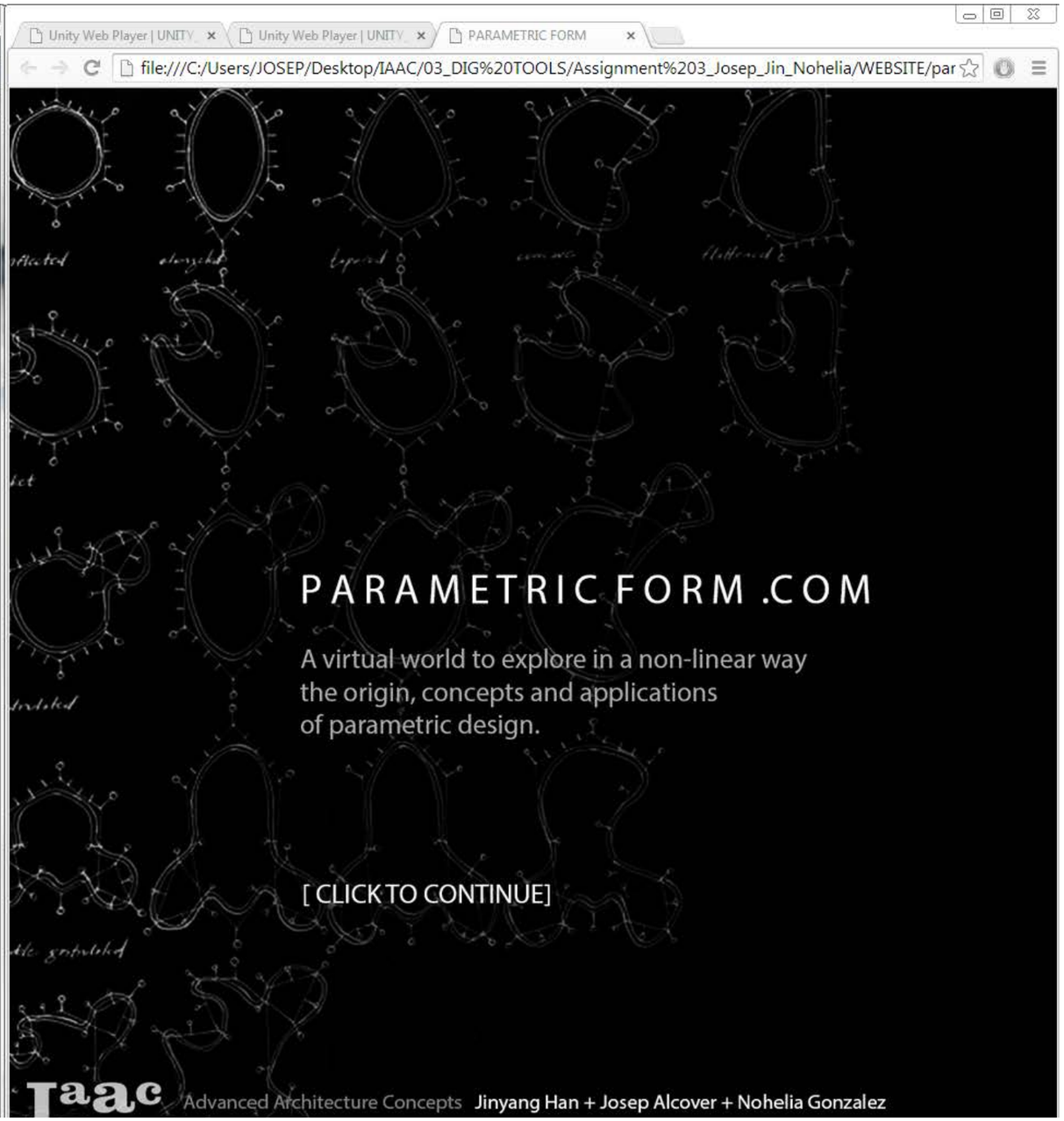
We want to upload the game to the internet but we don´t want it to be directly accessible.
We create a home page to receive the visitors and a page with the basic instructions of the game.
The structure ot the site will be HOME PAGE click INSTRUCTIONS PAGE click GAME.

C:\Users\JOSEP\Desktop\IAAC\03_DIG TOOLS\Assignment 3_Josep_Jin_Nohelia\WEBSITE\parametricform.html - Notepad++ [Administrator]

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

index.html | work.html | mobile_home.html | mobile_work.html | mobile_sphencsl.html | mobile_pannellum.html | cerabox.css | parametricform.ht

```
<!DOCTYPE html>
<html>
  <head>
    <title>PARAMETRIC FORM</title>

            <style type="text/css">

#fullscreen {
        z-index: -2;
        height: 100%;
        width: auto;
        position: fixed;
        top: 0;
        left:0px;
        margin-left: -2px;

}
    </style>

  </head>

    <body>


    <a href="parametricinstructions.html">  <img src="parametrichome.jpg" alt="" id="fullscreen"/>  </a>

  </body>

</html>
```



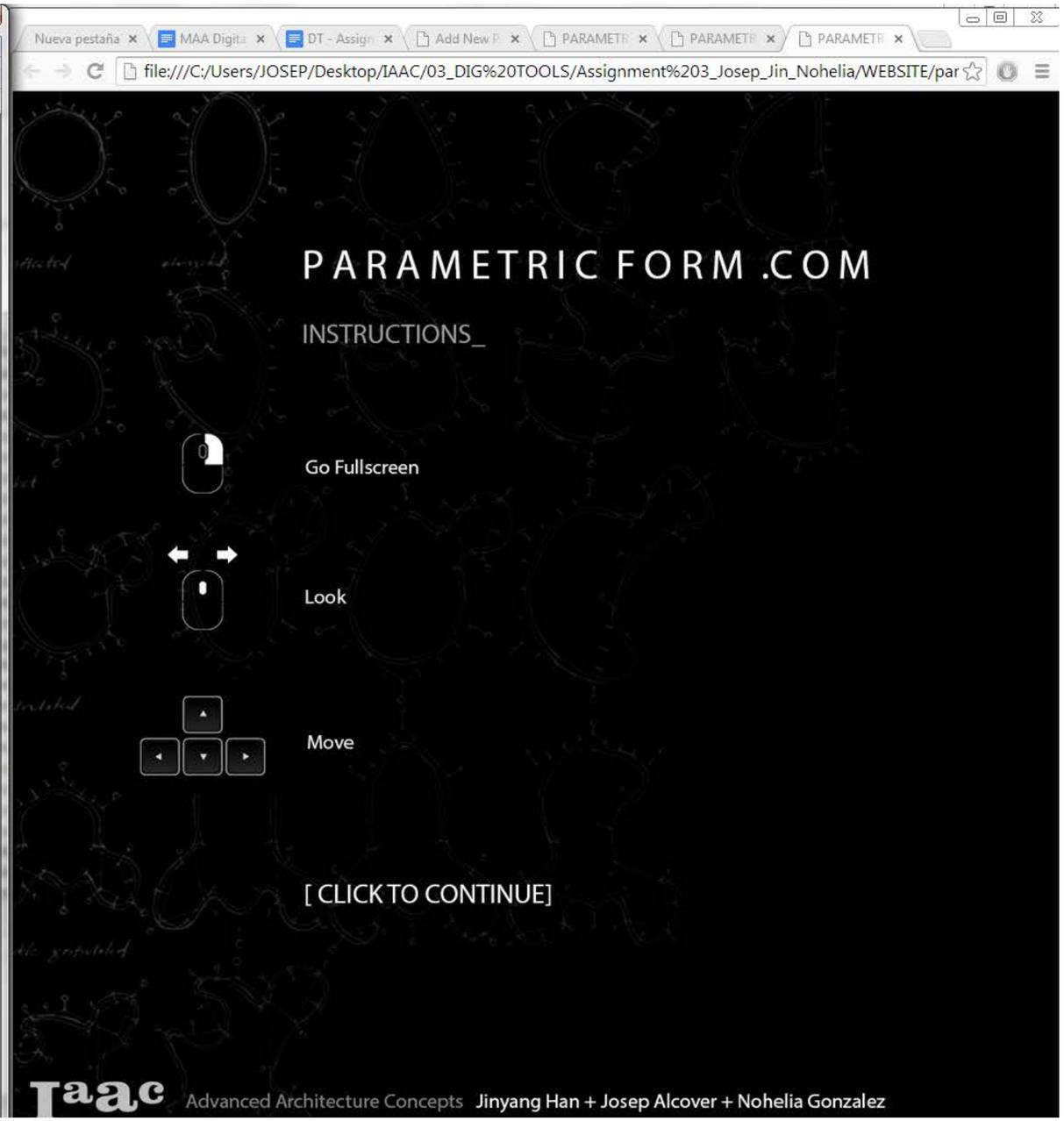
Creating other .html pages for the website

C:\Users\JOSEP\Desktop\IAAC\03_DIG TOOLS\Assignment 3_Josep_Jin_Nohelia\WEBSITE\parametricinstructions.html - Notepad++ [Administrator]

File Edit Search View Encoding Language Settings Macro Run Plugins Window ?

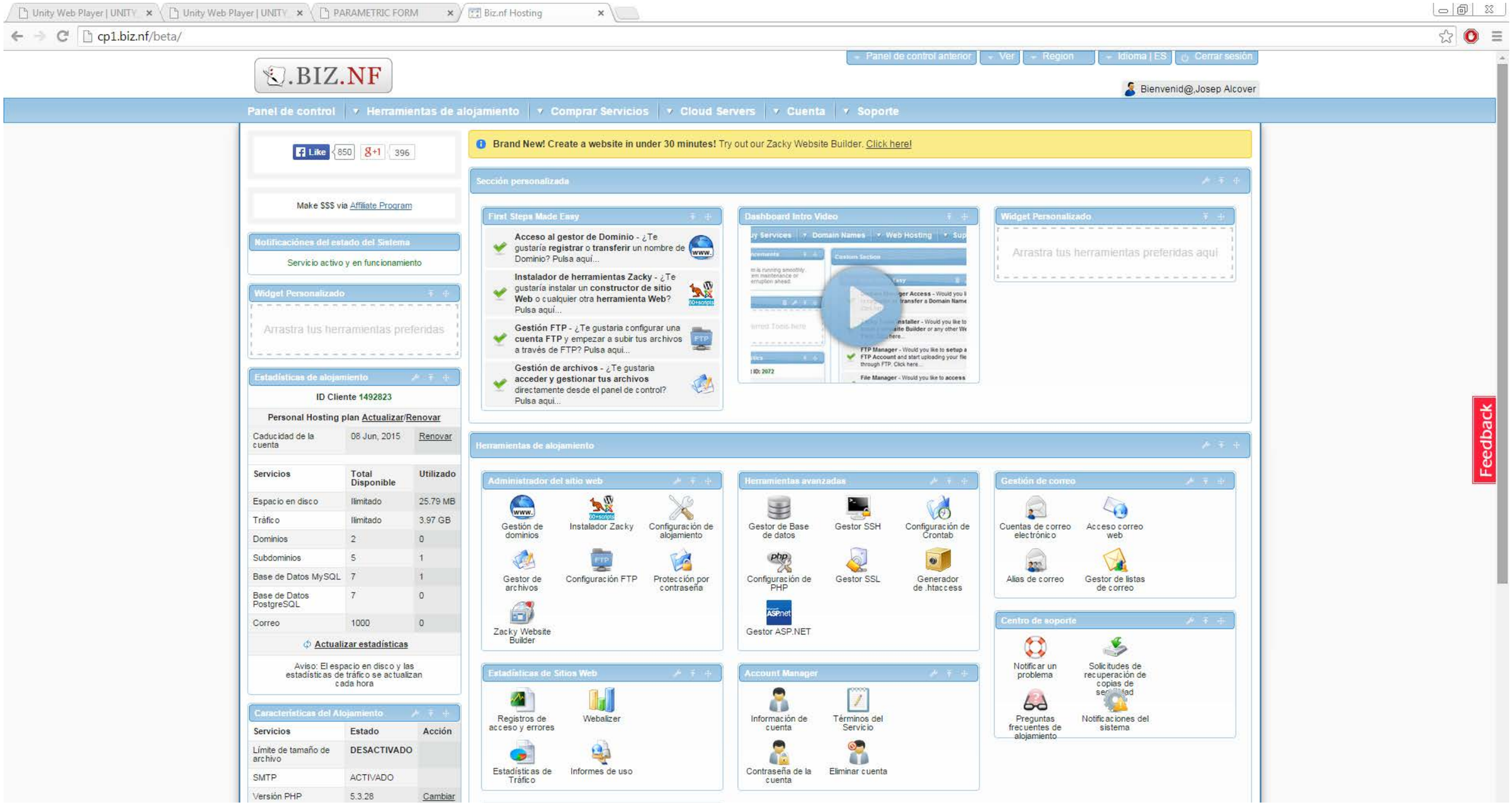
index.html | work.html | mobile_home.html | mobile_work.html | mobile_spherical.html | mobile_pannelum.html | cerabox.css | parametricform.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>PARAMETRIC FORM</title>

        <style type="text/css">



#fullscreen {
        z-index: -2;
        height: 100%;
        width: auto;
        position: fixed;
        top: 0;
        left:0px;
        margin-left: -2px;


}

    </style>

  </head>


  <body>


    <a href="WEB GAME.html">  <img src="parametricinstructions.jpg" alt="" id="fullscreen"/>  </a>


  </body>

</html>
```



Web Hosting

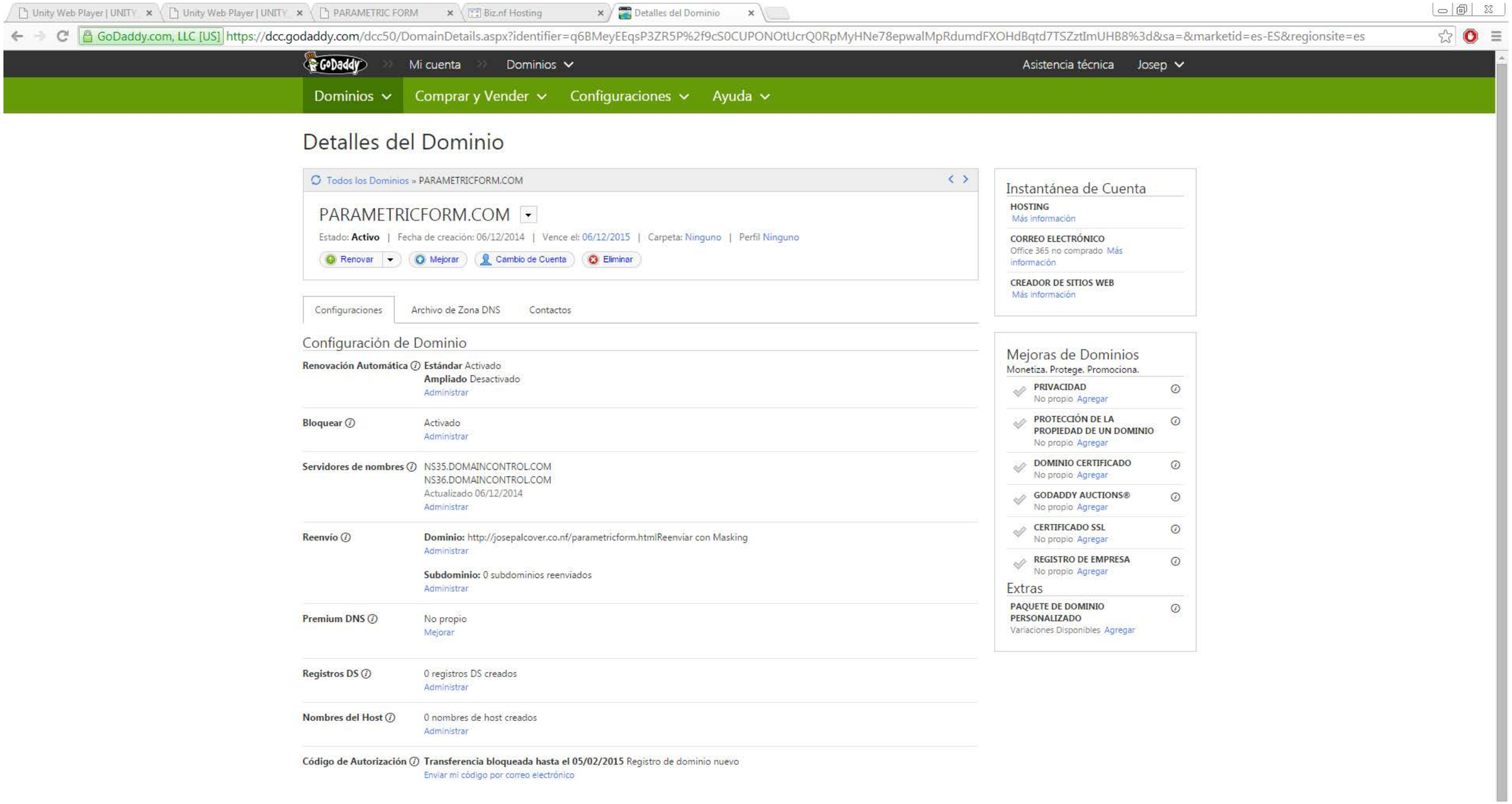
We sign up for a free web hosting service in www.biz.nf.
This gives us space in the internet.
We get a free .co.nf domain.



.com Domain

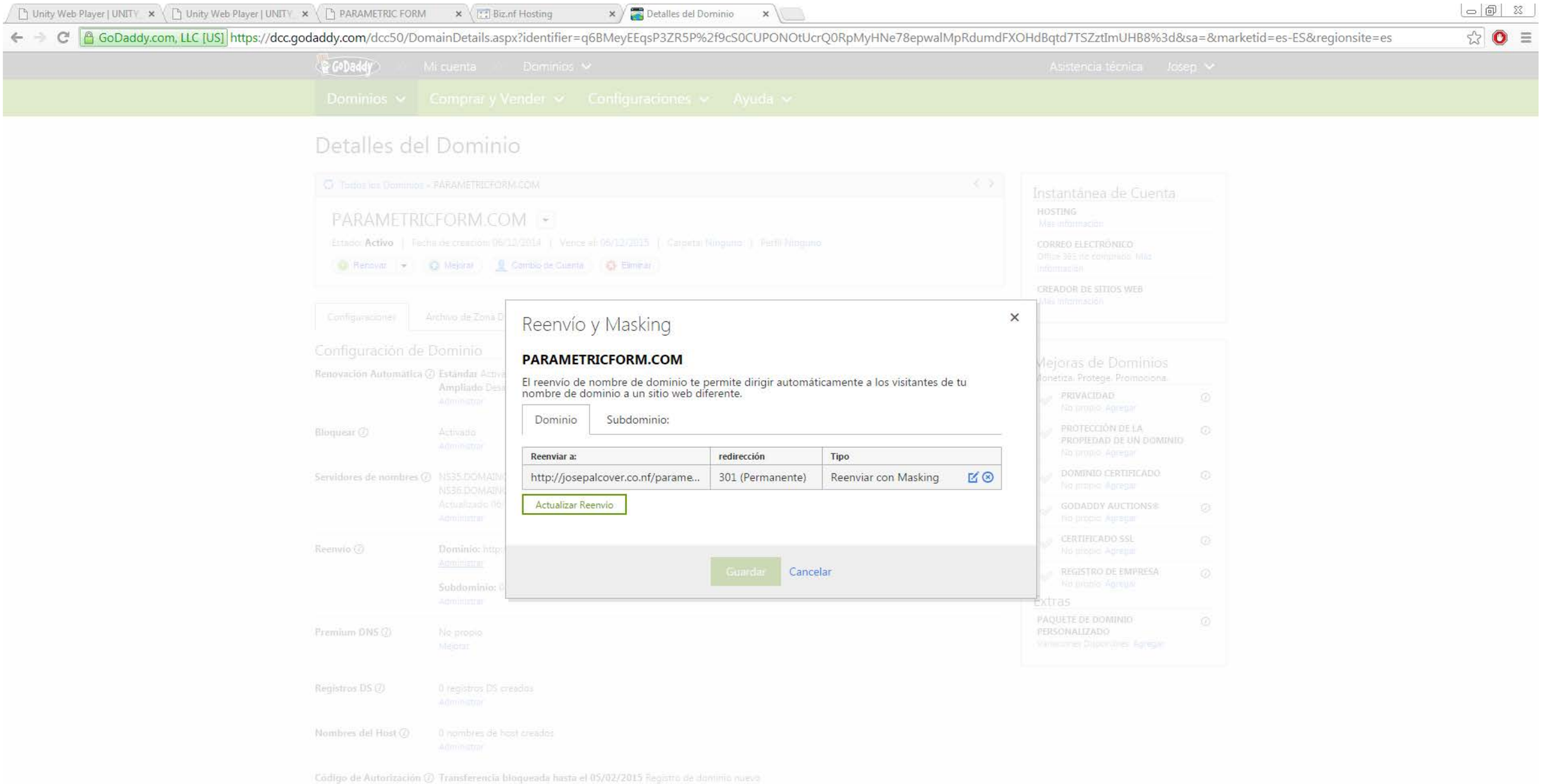
We buy a .com domain name from www.godaddy.com.

it is PARAMETRICFORM.COM



Forwarding parametricform.com to the .co.nf domain

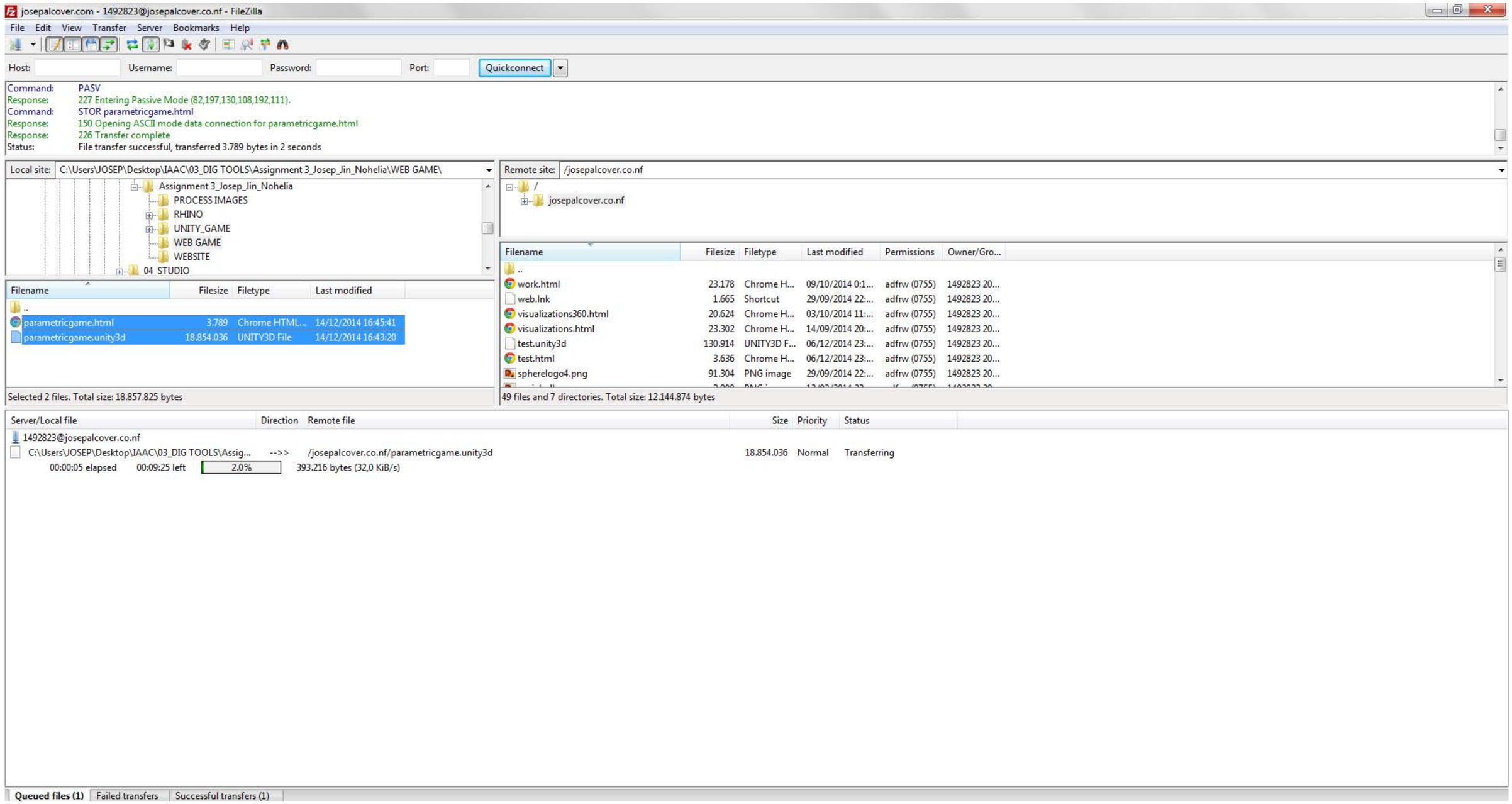
We forward parametricform.com to the .co.nf domain that is linked to the web host server where we will upload the game and other .html files of the website.
We activate the masking. This means that in the address bar of the browser we will always see parametricform.com.



Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

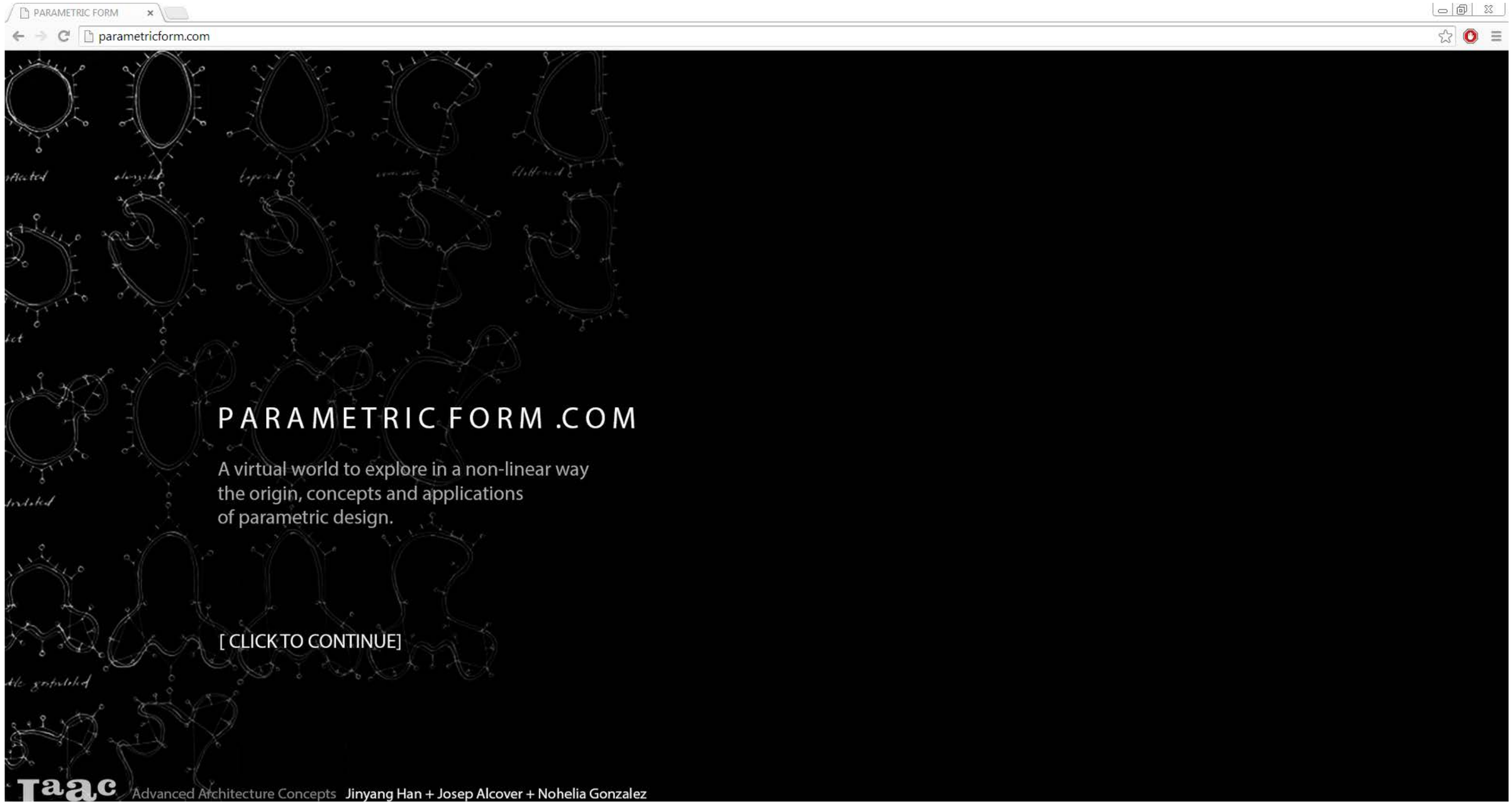
Uploading the files with FileZilla

We use FileZilla to upload the files to our web host server.
We upload 3 .html files (the home page, the instrucions page and the game page)
and 1 .unity3d file that stores information of the game.

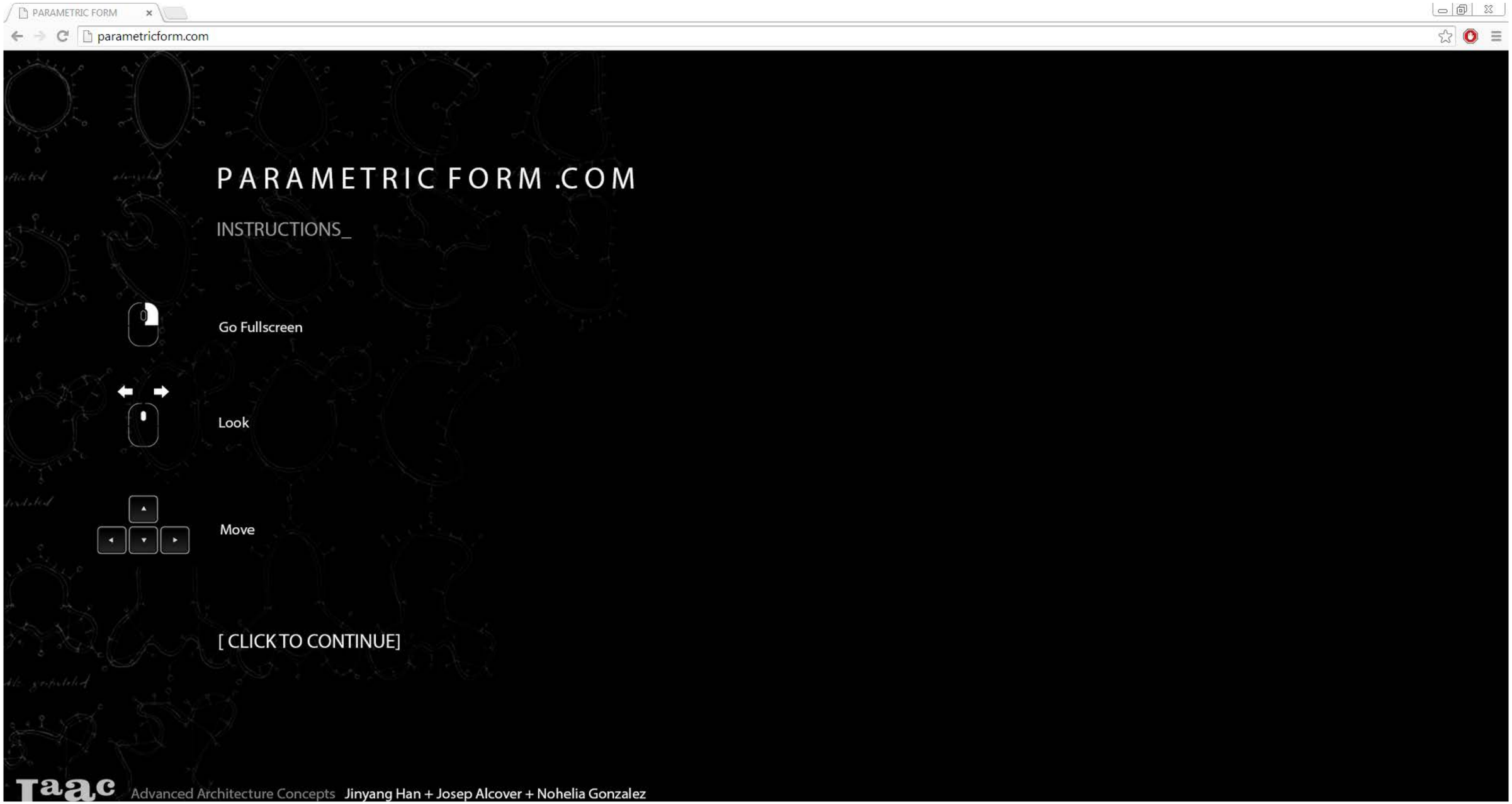


Playing the game online

Now the game is in PARAMETRICFORM.COM
and everyone can play it.

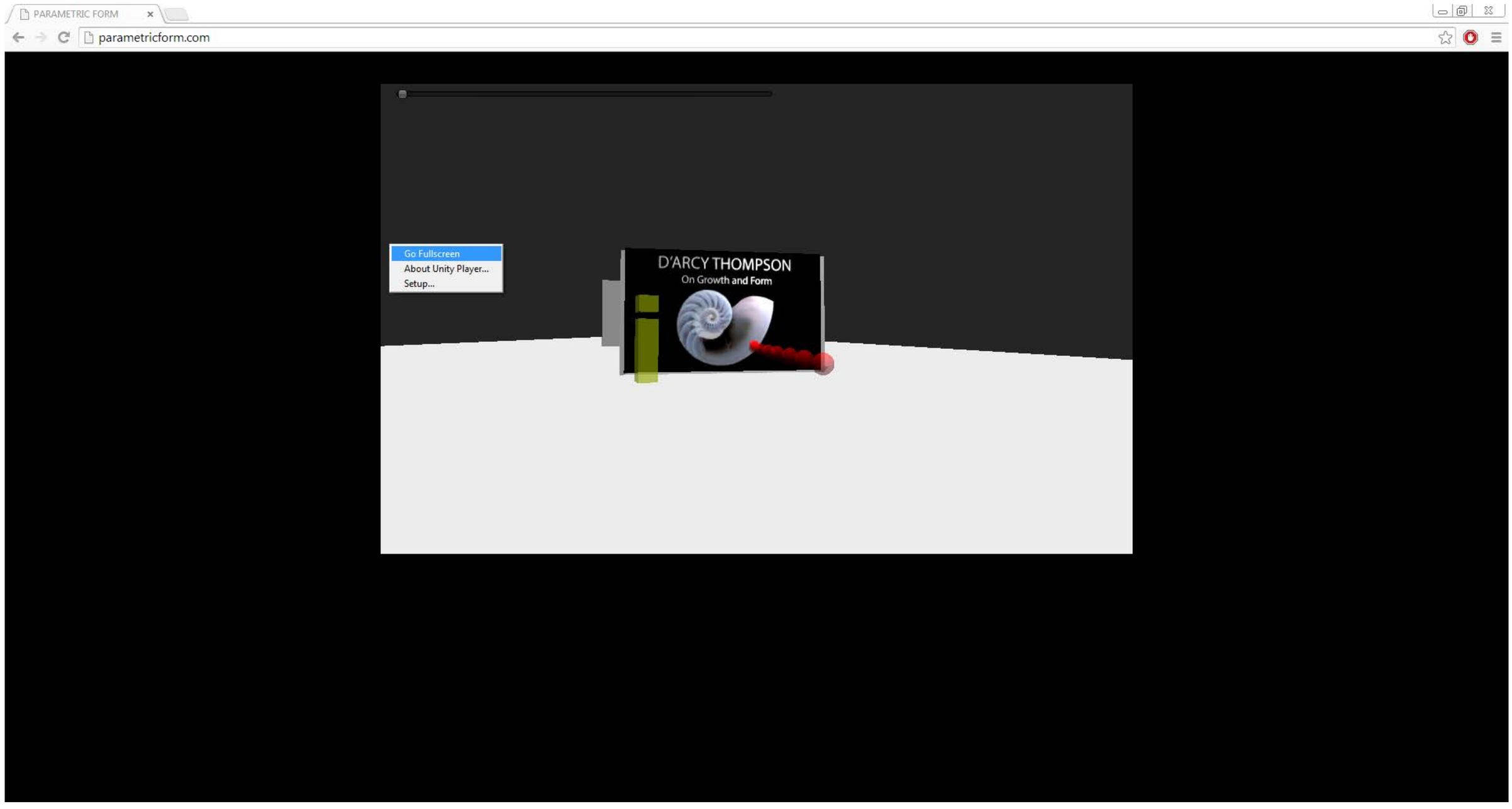


Playing the game online



Playing the game online

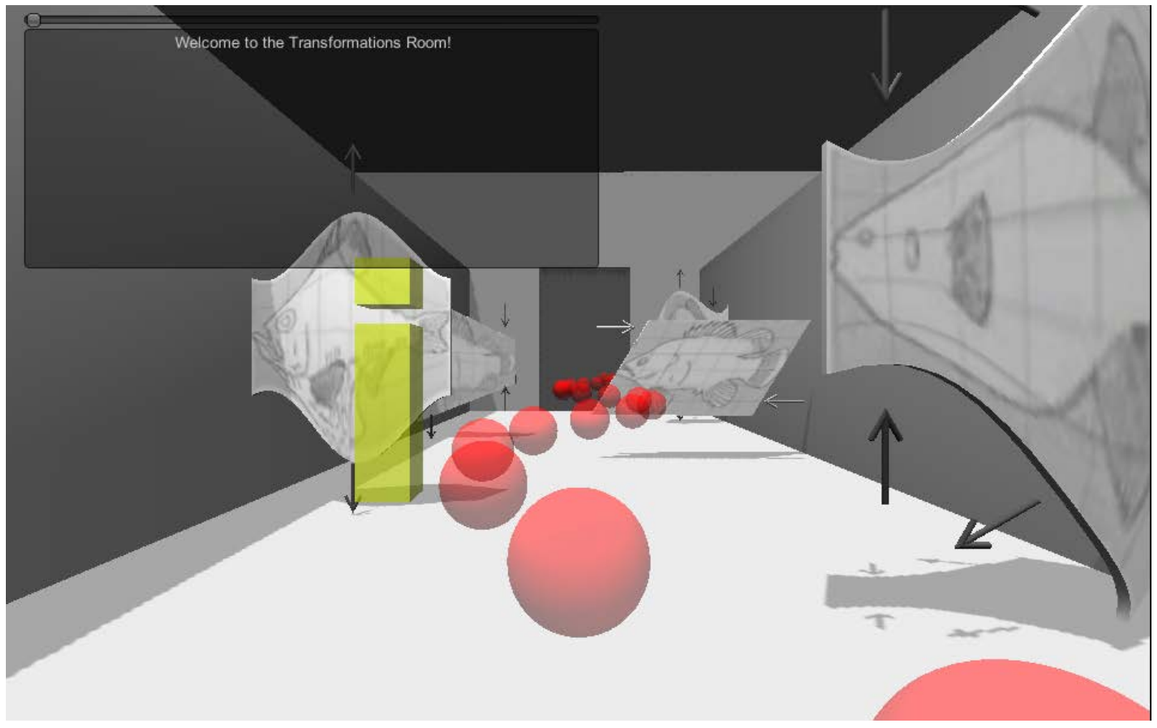
Playing the game online

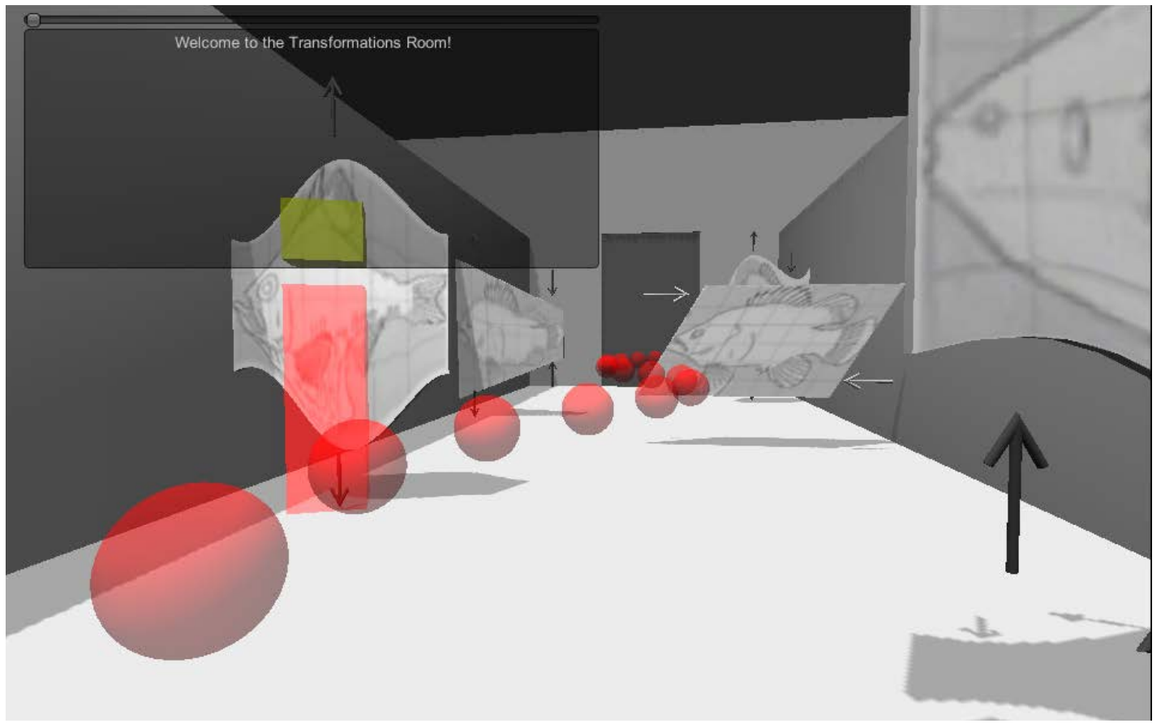


Assignment 3 - Gameify! (Unity 3D)

Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

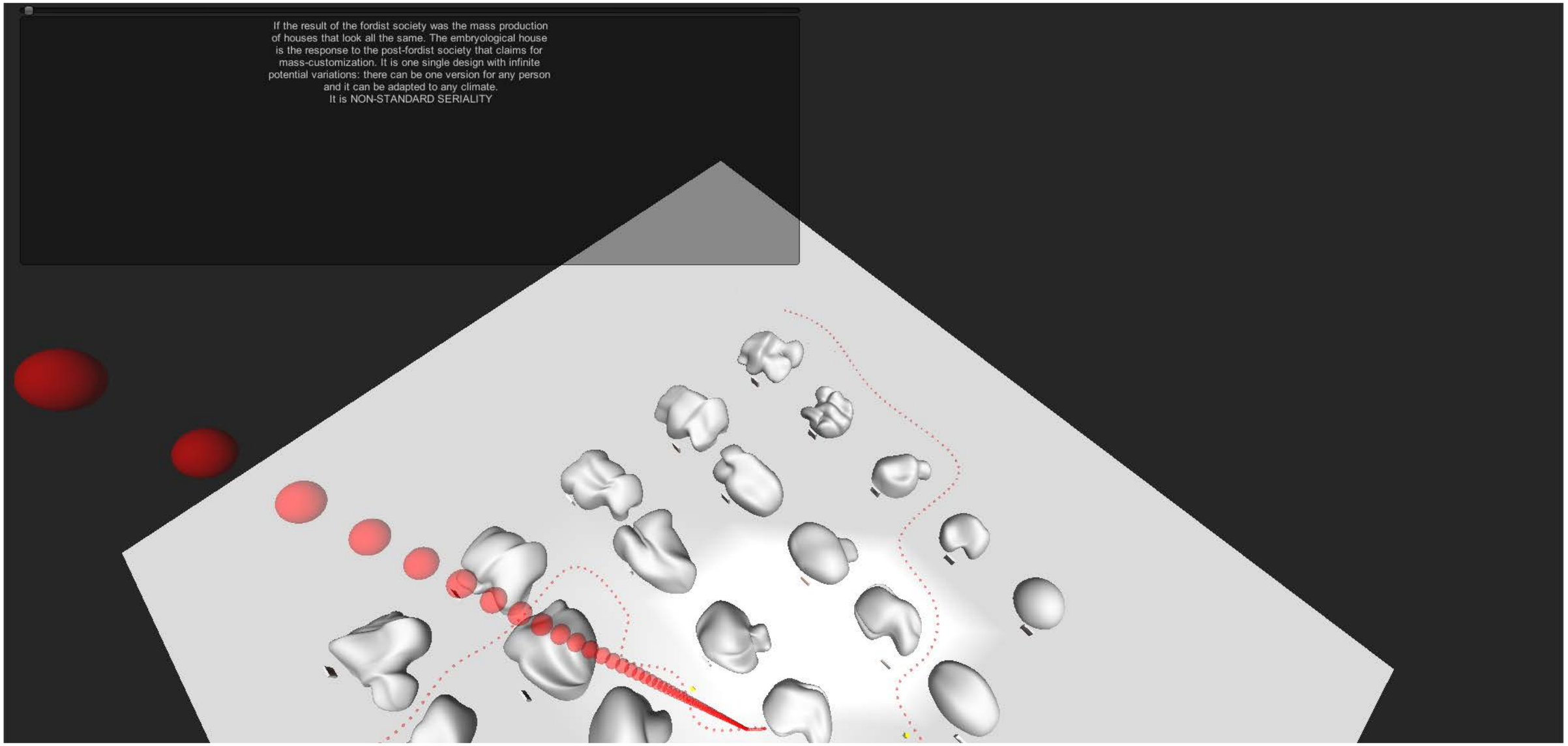
Playing the game online





Institute for Advanced Architecture of Catalonia
2014/// Digital Tools

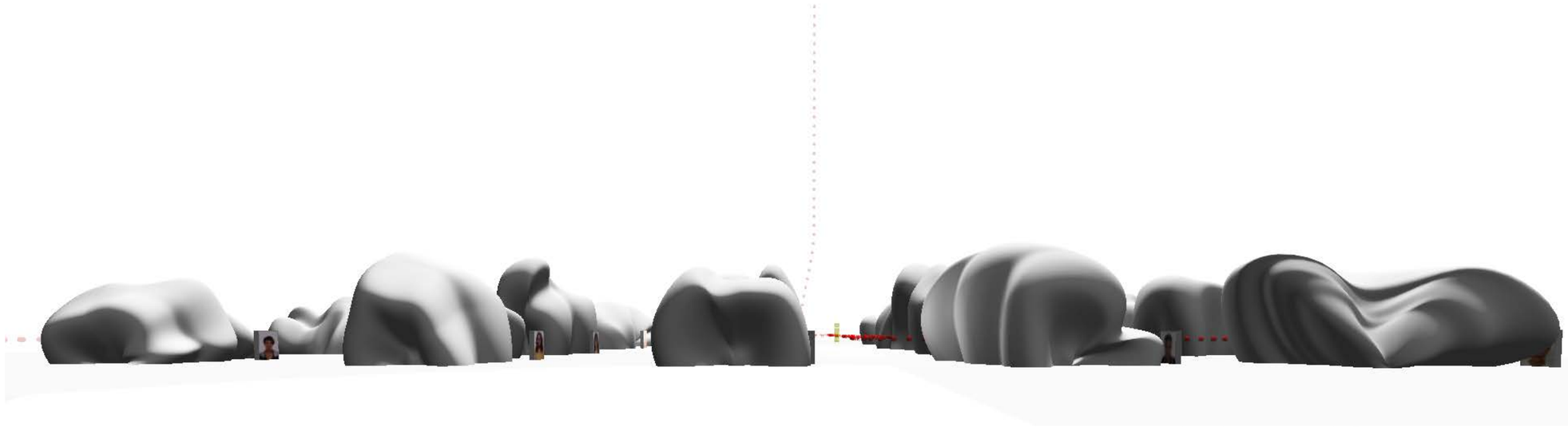
Playing the game online



Iaac | Institute for Advanced Architecture of Catalonia
2014/// Digital Tools

Assignment 3 - Gameify! (Unity 3D)

Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han



PARAMETRIC FORM .COM

Gameify! (Unity 3D)

Digital Tools

Josep Alcover Llubiá + Nohelia Gonzalez + Jinyang Han

Iaac