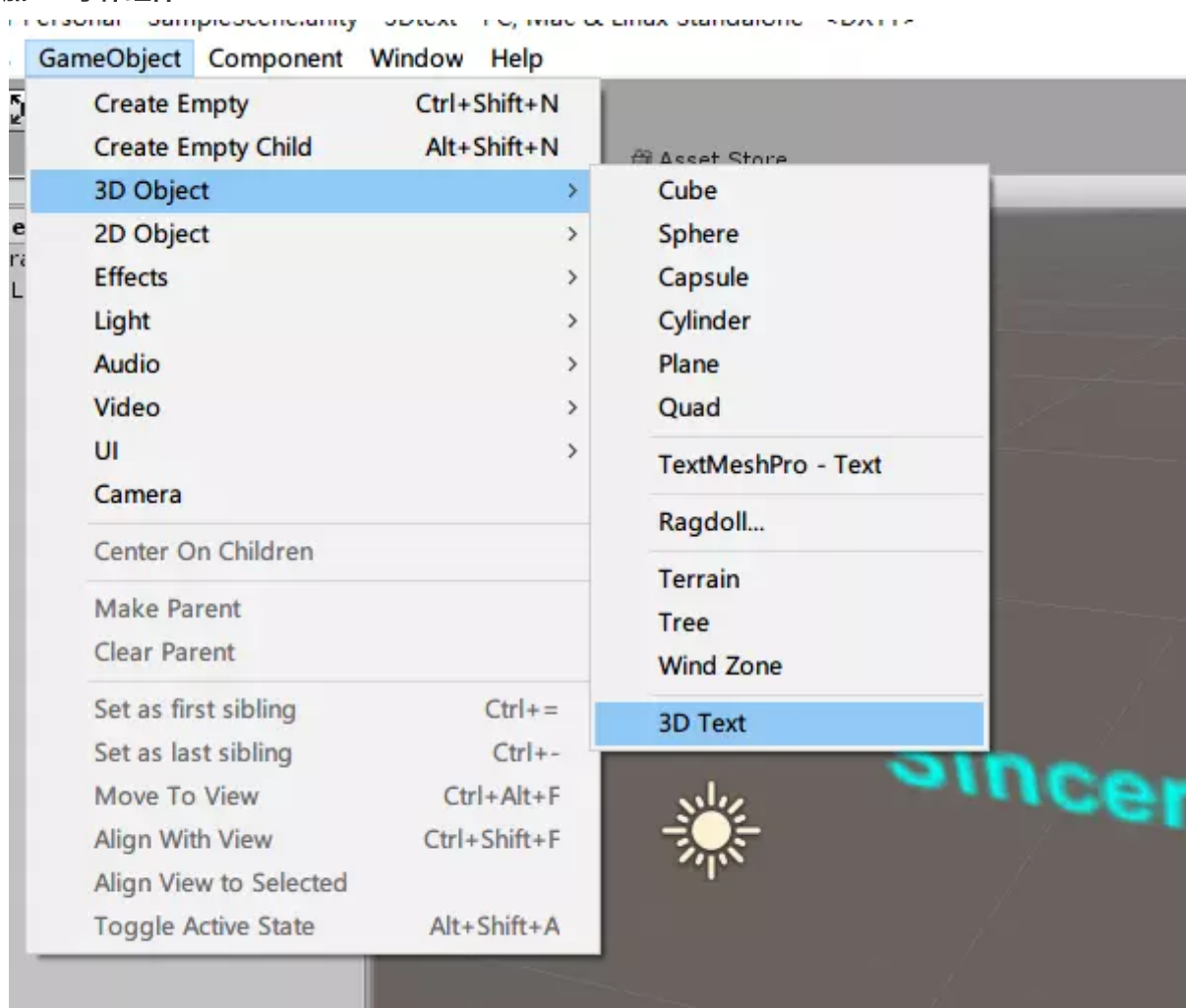


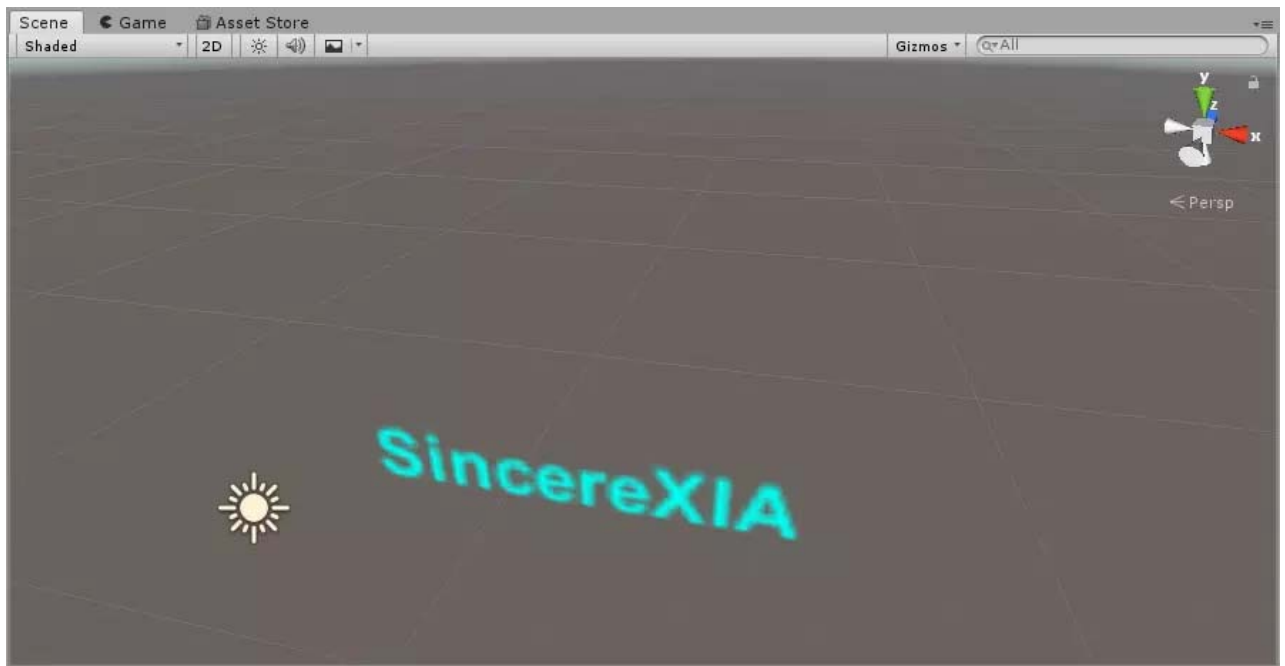
例子

3D 字体

根据鼠标移动距离来旋转 3D 字体

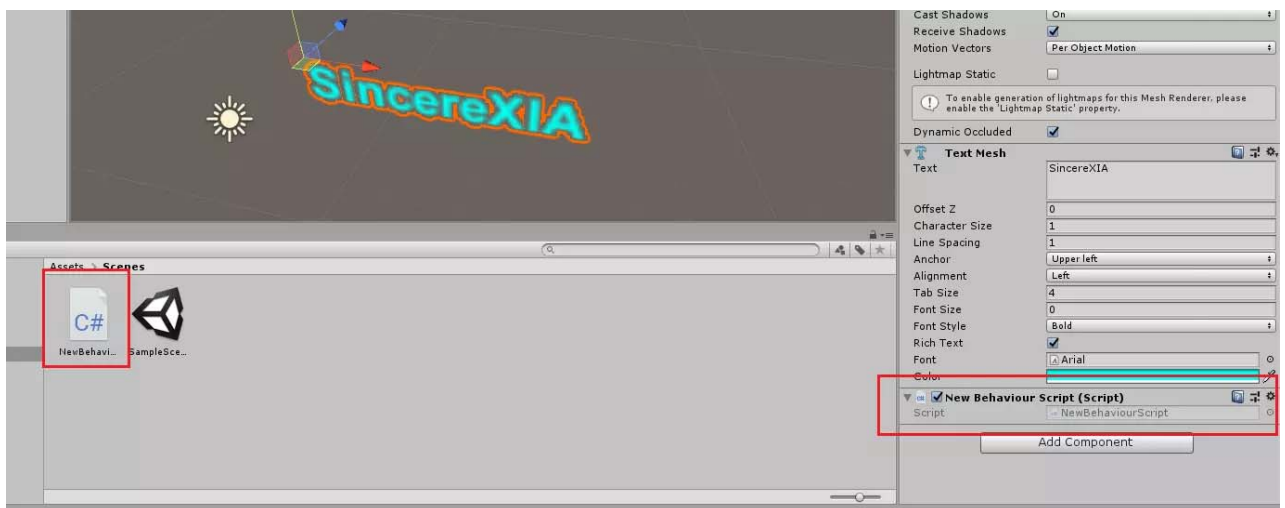
1. 添加 3D 字体组件



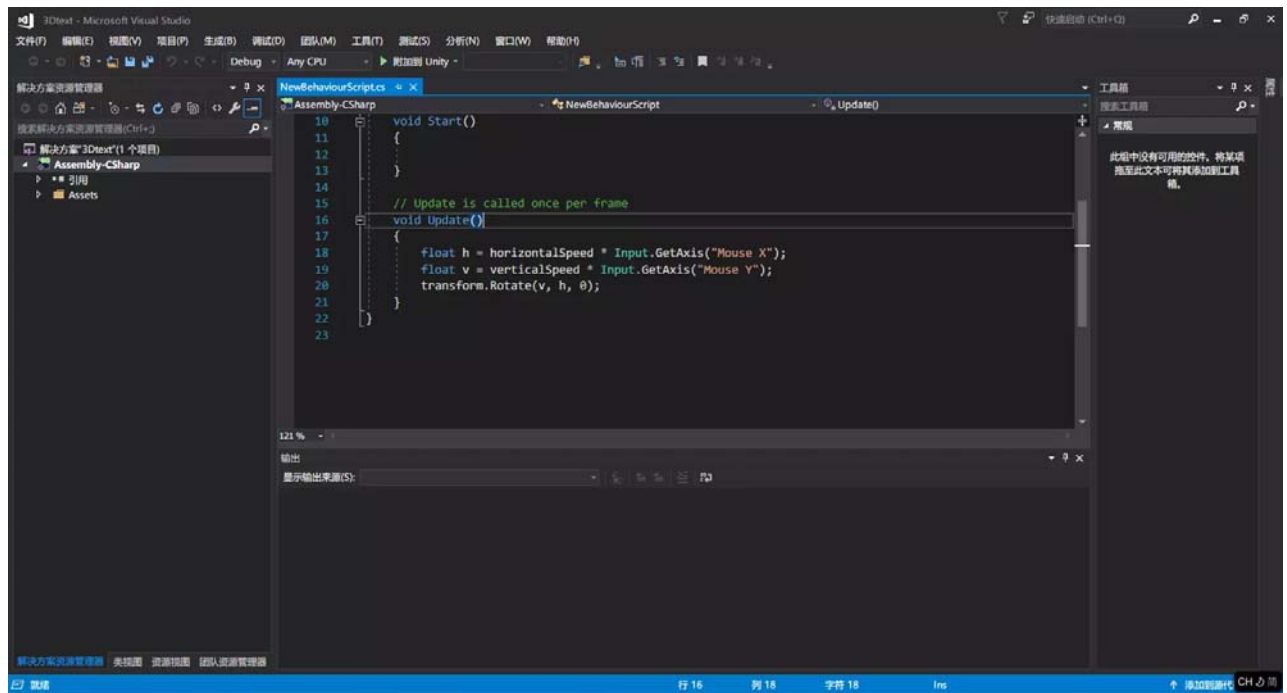


2. 新建脚本文件

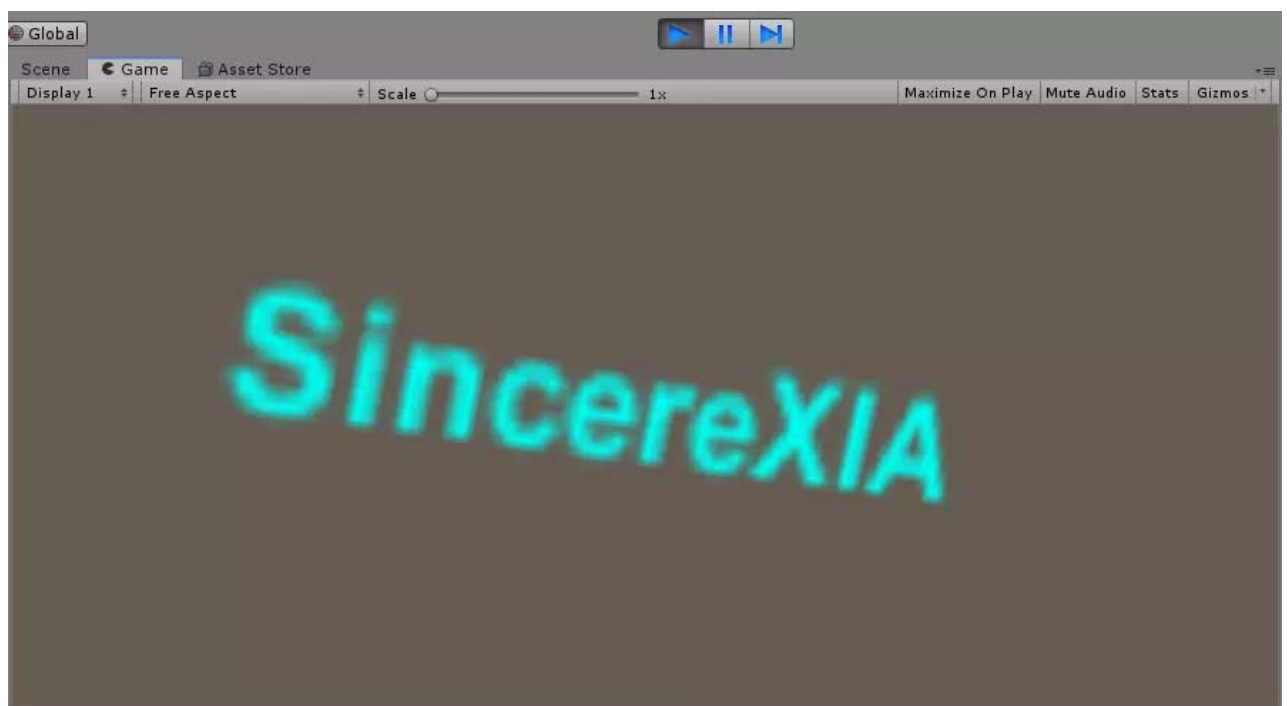
3. 将脚本添加到创建的游戏对象上



4. 编辑脚本



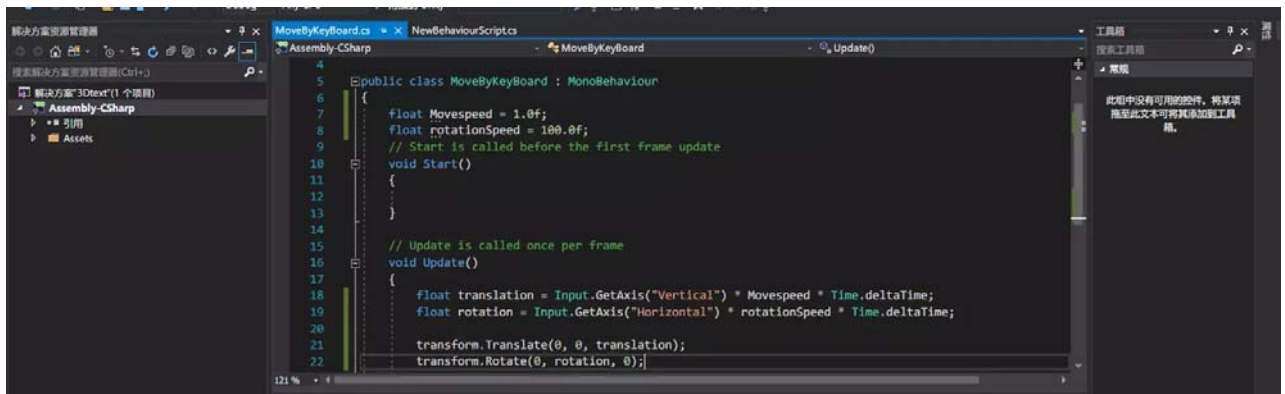
5. 启动游戏运行



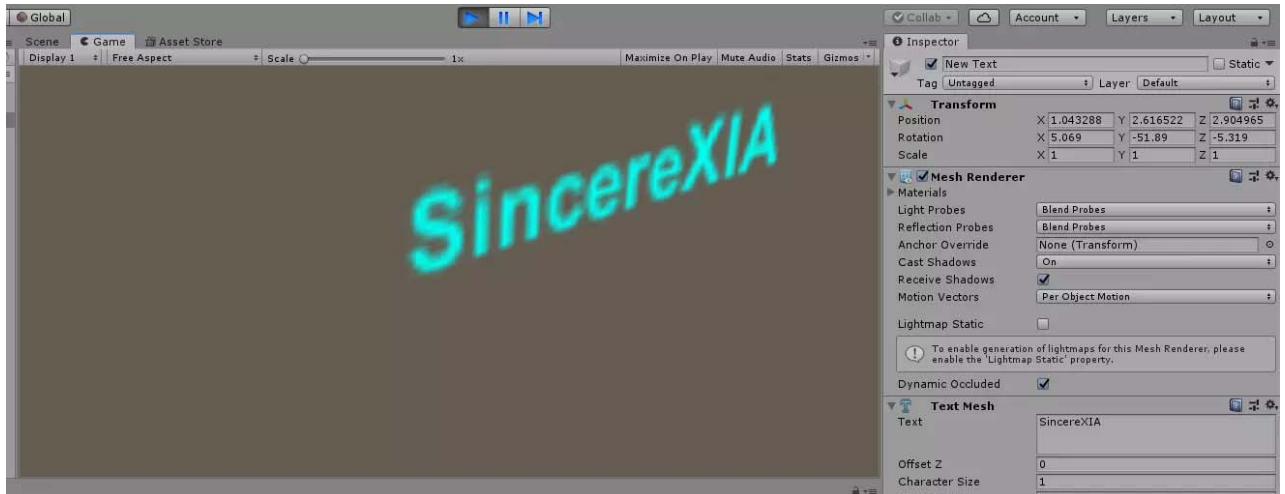
游戏内文字随着鼠标的移动而旋转

键盘操作

1. 编辑新的脚本



2. 绑定脚本并进行测试

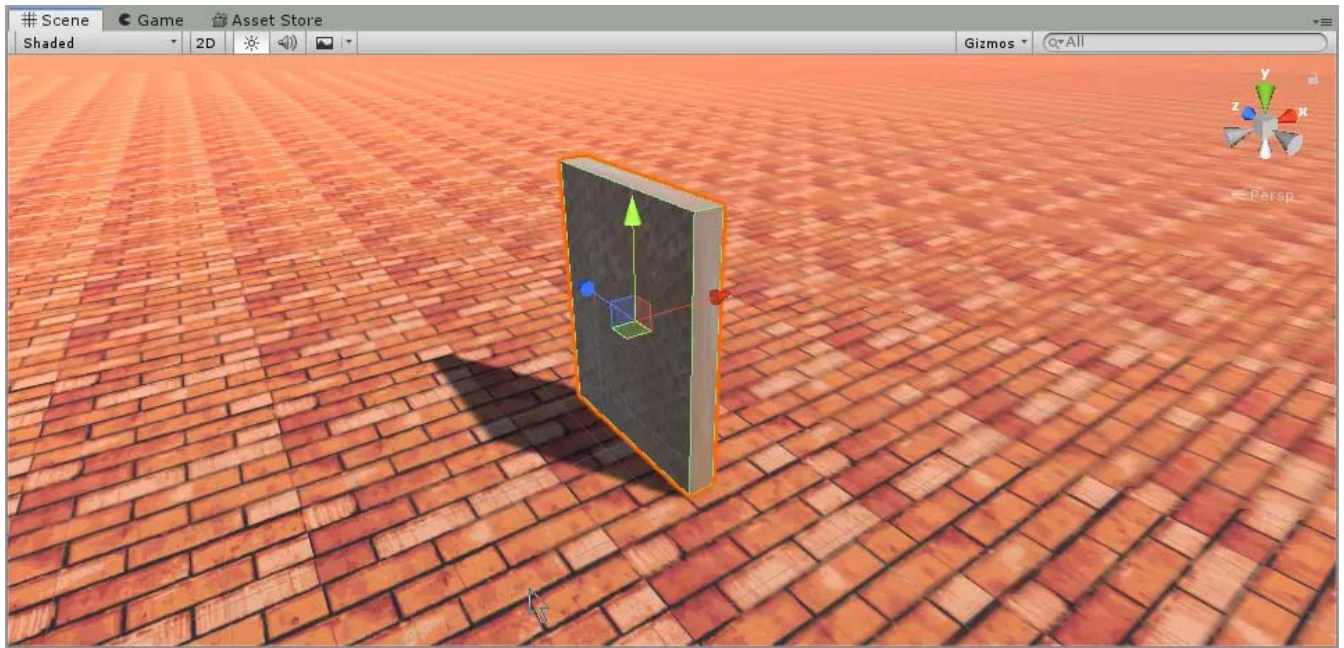


使用方向键可以控制模型在 x-z 平面上的移动

练习

游戏对象跟随鼠标移动、旋转

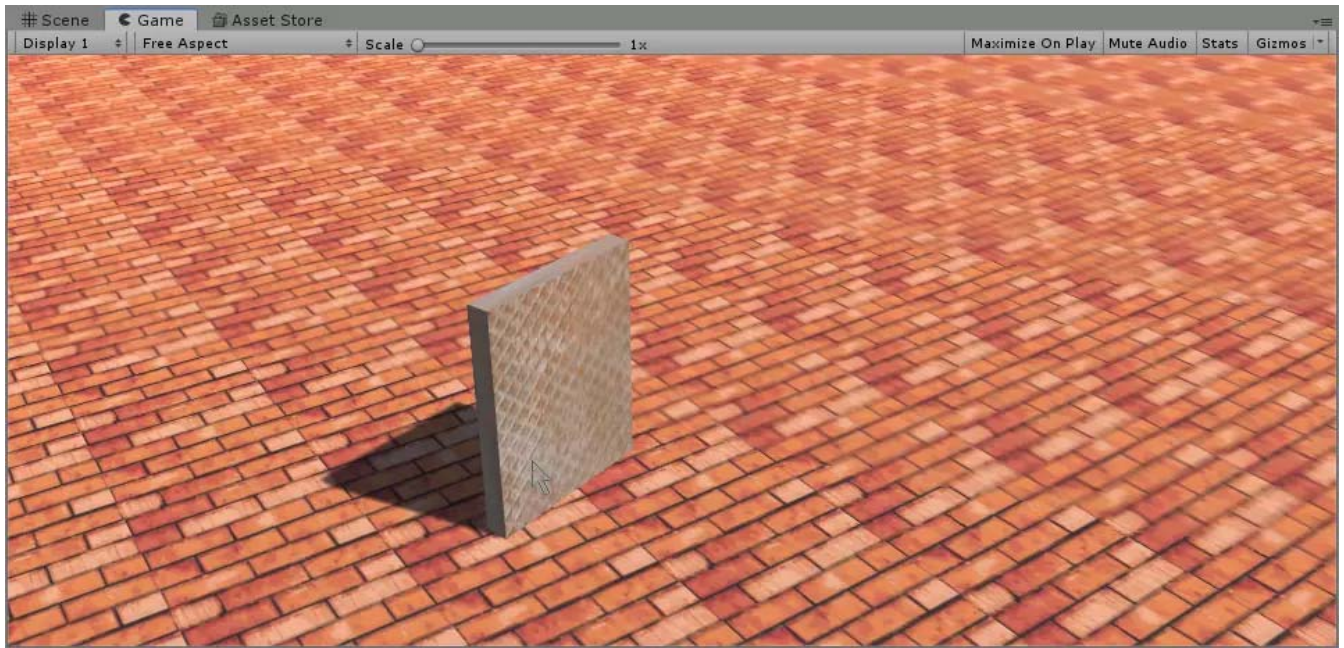
建立 Cube 游戏对象，并添加纹理贴图



编写跟随鼠标移动脚本

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Follow : MonoBehaviour
6  {
7      Vector3 world; // 物体要移动到的位置 (世界坐标系)
8      bool isfixed = false;
9      void Update()
10     {
11         Vector3 targetposition =
Camera.main.WorldToScreenPoint(this.transform.position); // 将物体的世界坐标转换为屏幕坐标
12
13         Vector3 mouse_position = Input.mousePosition; // 鼠标在屏幕上的位置坐标
14         mouse_position.z = targetposition.z;
15
16         // world = Camera.main.ScreenToWorldPoint(mouse_position); // 这种情况下 会有穿透现象
17         world.x = Camera.main.ScreenToWorldPoint(mouse_position).x;
18         world.z = Camera.main.ScreenToWorldPoint(mouse_position).z;
19         world.y = this.transform.position.y;
20         if (this.transform.position != world)
21         {
22             this.transform.position = world;
23         }
24     }
25 }
26
```

将脚本添与 Cube 对象绑定，即可实现通过鼠标移动，控制物体在 x, z 平面上移动



跟随鼠标旋转

补充之前的脚本，增加按右键固定游戏对象位置，并跟随鼠标旋转的功能

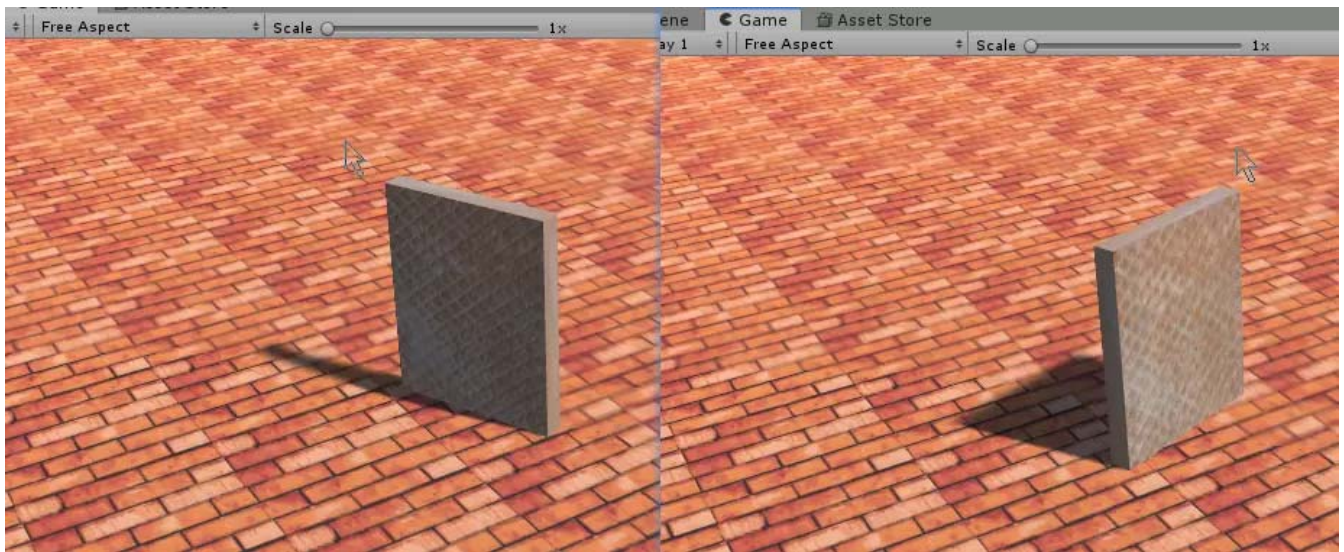
```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Follow : MonoBehaviour
6  {
7      // Start is called before the first frame update
8      void Start()
9      {
10
11     }
12
13     Vector3 world; // 物体要移动到的位置 (世界坐标系)
14     bool isfixed = false;
15
16     void Update()
17     {
18         Vector3 targetposition =
19         Camera.main.WorldToScreenPoint(this.transform.position); // 将物体的世界坐标转换为屏幕坐标
20
21         Vector3 mouse_position = Input.mousePosition; // 鼠标在屏幕上的位置坐标
22         mouse_position.z = targetposition.z;
23
24         // world = Camera.main.ScreenToWorldPoint(mouse_position); // 这种情况下 会有穿透现象
25
26         world.x = Camera.main.ScreenToWorldPoint(mouse_position).x;
27         world.z = Camera.main.ScreenToWorldPoint(mouse_position).z;
28         world.y = this.transform.position.y;
29
30         if (!isfixed)
31         {
```

```

31
32         if (this.transform.position != world)
33         {
34             this.transform.position = world;
35         }
36
37     }
38     if (Input.GetMouseButton(0))
39     {
40         isfixed = true;
41     }
42     if (Input.GetMouseDown(1))
43     {
44         isfixed = true;
45     }
46     this.transform.LookAt(world);
47     if (Input.GetMouseButtonUp(1))
48     {
49         isfixed = false;
50     }
51 }
52 }
53

```

进入游戏测试，可见游戏对象可以完成跟随鼠标右键旋转



键盘随输入某个键创建/实例化游戏对象

编写游戏脚本 `AddDomino.cs`

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class AddDomino : MonoBehaviour
6  {
7      // Start is called before the first frame update

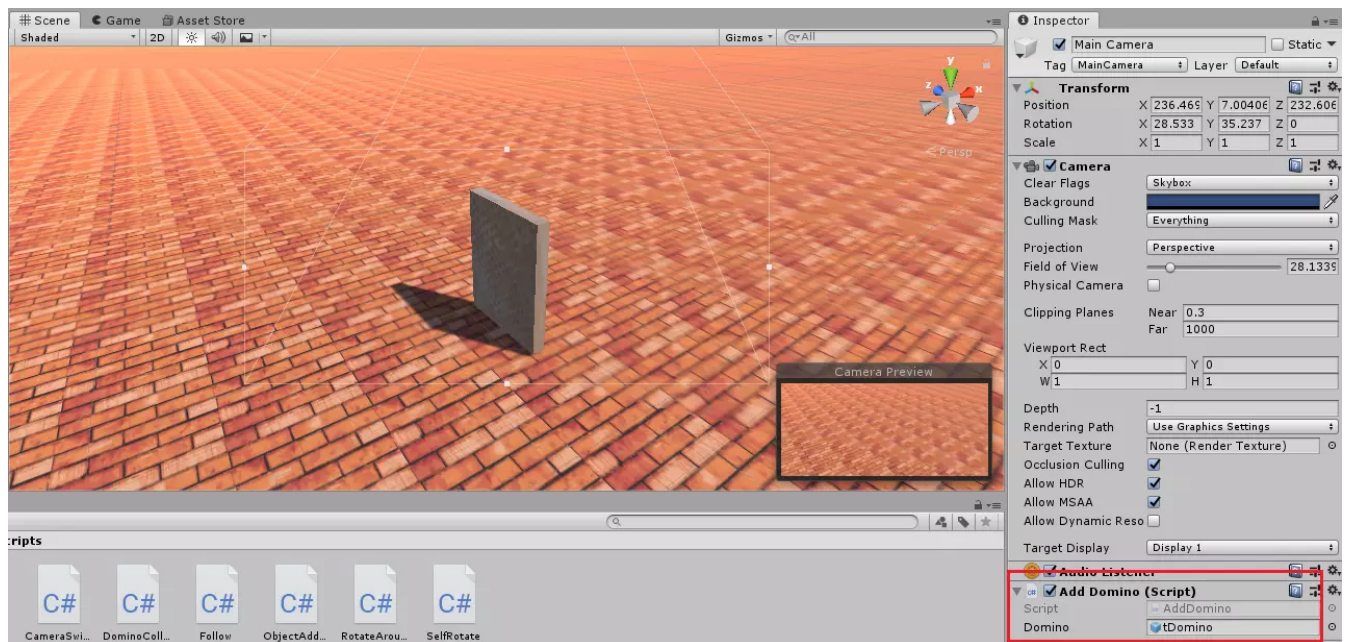
```

```

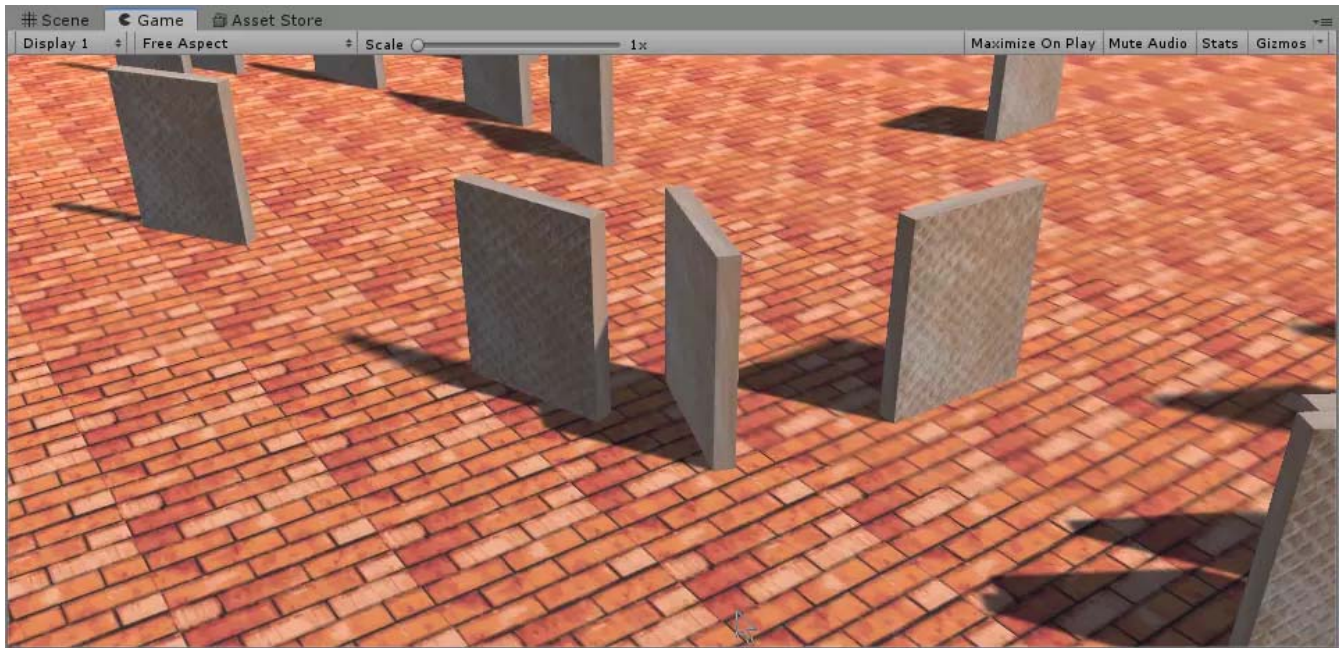
8      public GameObject Domino;
9      void Start()
10     {
11
12     }
13
14     // Update is called once per frame
15     void Update()
16     {
17         if (Input.GetKeyDown(KeyCode.Space))
18         {
19             GameObject domino = Instantiate(Domino, new Vector3(230f, 1.1f, 40f),
Quaternion.identity);
20         }
21     }
22 }
23

```


将脚本与 Main Camera 绑定，并设定 Domino 游戏对象、



进入游戏进行测试，按 Space 键即可完成游戏对象的创建



自学 unity 用户手册

 **DOCUMENTATION**

ManualScripting API

Search manual...

unity3d.com

Language: English

Version: 2019.1 (switch to 2018.3 or 2017.4)

- Unity User Manual (2019.1)
- Packages documentation
- Working in Unity
- Importing
- 2D
- Graphics
- Physics
- Scripting**
 - Scripting Overview
 - Scripting Tools
 - Event System
 - C# Job System
- Multiplayer and Networking
- Audio
- Animation
- Timeline
- UI
- Navigation and Pathfinding
- Unity Services
- XR
- Open-source repositories
- Asset Store Publishing
- Platform-specific
- Experimental
- Legacy Topics

Unity User Manual (2019.1) / Scripting

←

→

Scripting

- Character
- AutoBackToTitle.cs
- ClickToStart.cs
- Explosion.cs
- Explosive.cs
- Fire.cs
- FloorSection.cs
- GameControl.cs
- GameGUI.cs
- Hose.cs
- MapIcons.cs
- MessageGUI.cs
- MoveBetweenPoints
- Player.cs
- Priority Particle Add.
- PriorityAlphaParticle
- SceneChanger.cs
- SmokeParticles.cs
- StopParticles

```
50 vignette.blur = (1-health) * 2 + smokeEffect * 10 + health * 10;
51 vignette.blurDistance = (1-health) * 2 + smokeEffect * 10;
52 vignette.chromaticAberration = healthEffect * 10;
53 }
54
55
56 void OnTriggerStay(Collider c)
57 {
58     var fire = c.GetComponent<Fire>();
59     if (fire && fire.alive)
60     {
61         float dist = 1-((transform.position - fire.transform.position).sqrMagnitude);
62         NearHeat(dist);
63     }
64
65     var smoke = c.GetComponent<SmokeParticle>();
66     if (smoke && smoke.GetComponent<Particle>().isActiveAndNotStopped)
67     {
68         float dist = 1-((transform.position - smoke.transform.position).sqrMagnitude);
69         NearSmoke(dist);
70     }
71 }
72
73
74 void OnCollisionEnter(Collision c)
```

[Leave feedback](#) [Other Versions](#)