关于碰撞检测

知识点

- 碰撞体：Collider
- 触发器：Is Trigger
- 检测方法：OnCollisionEnter ， onTriggerEnter
- 碰撞对象的结构：Collision，Collider，
- 运动学物体：Is Kinematic
- 碰撞体处理分类：静态碰撞体、刚体碰撞体、非刚体碰撞体
- 碰撞事件与碰撞体类型的关系

大数据与物联网学院 邵亮

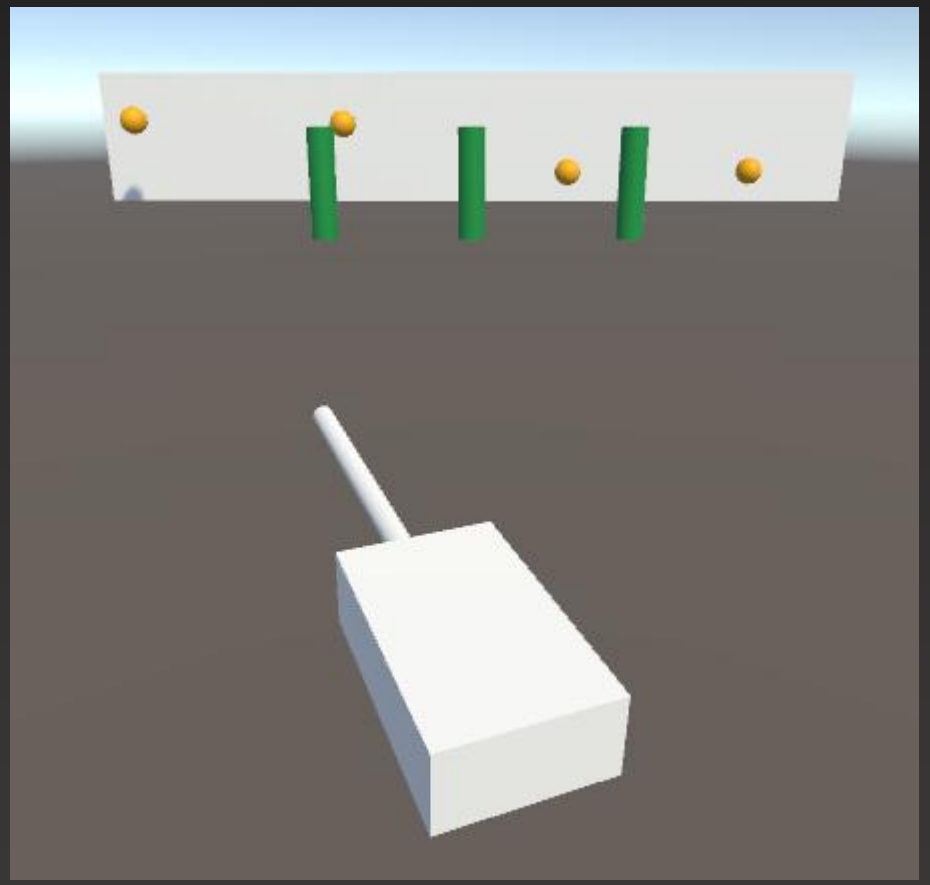
课堂案例：坦克炮弹击毁目标

功能要求：

1、按下空格，炮塔能够发射炮弹（上讲内容，已完成）；

2、炮弹击中目标（绿色圆柱体），目标和炮弹均击毁；

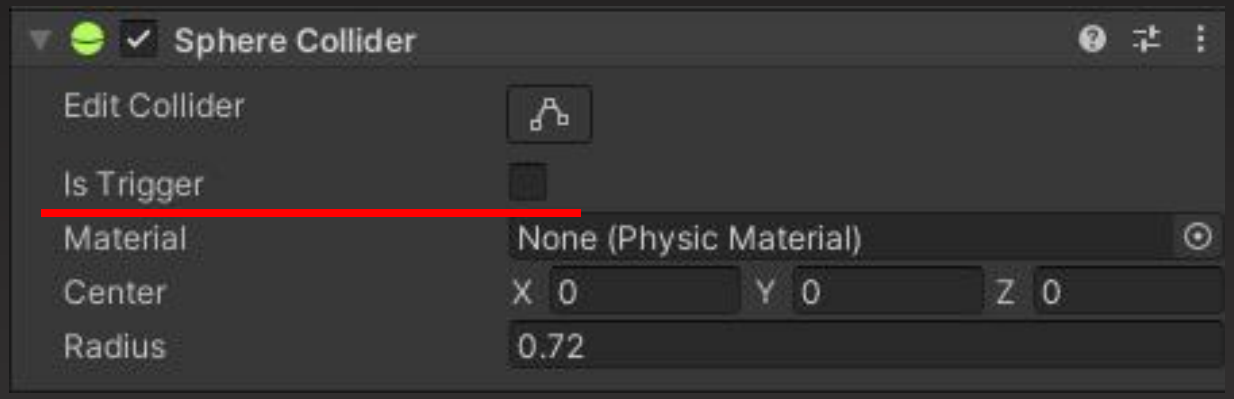
3、炮弹击中目标白墙（脱靶），在弹着点出现黄球。



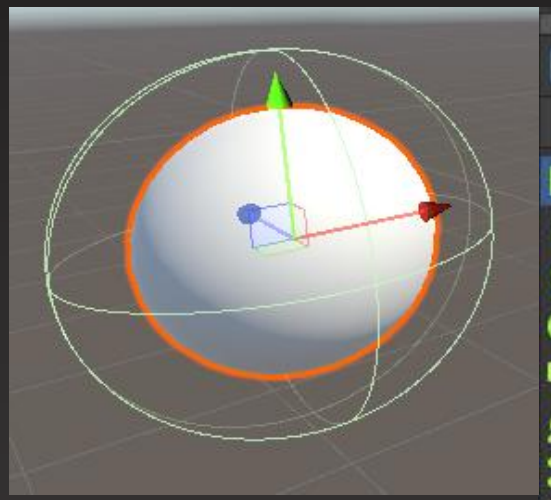
碰撞检测的基本原理

1、碰撞体：Collider

碰撞体属性



球型碰撞体（绿线）



各种碰撞体组件

2、触发器 Is Trigger

- 打开：可以穿过；（至少有一个刚体）
- 关闭：不能穿过。

3、碰撞检测方法

Is Trigger 打开（可穿过）	Is Trigger 关闭（不可穿过）	
Trigger	Collision	含义
OnTriggerEnter（）	OnCollisionEnter（）	碰撞开始
OnTriggerStay（）	OnCollisionStay（）	一直在撞
OnTriggerExit（）	OnCollisionExit（）	碰撞结束

检测方法应用

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class projectile : MonoBehaviour
{

    void OnTriggerEnter(Collider obj)
    {
        if(obj.name=="box")
        {
            Destroy(obj.gameObject);
        }
    }
}
```

Collider碰撞对象常用属性

属性	备注
name	撞击体的名字
transform	1、撞击体对象的变型子对象 2、position位置子对象常用
gameObject	撞击体对象自身

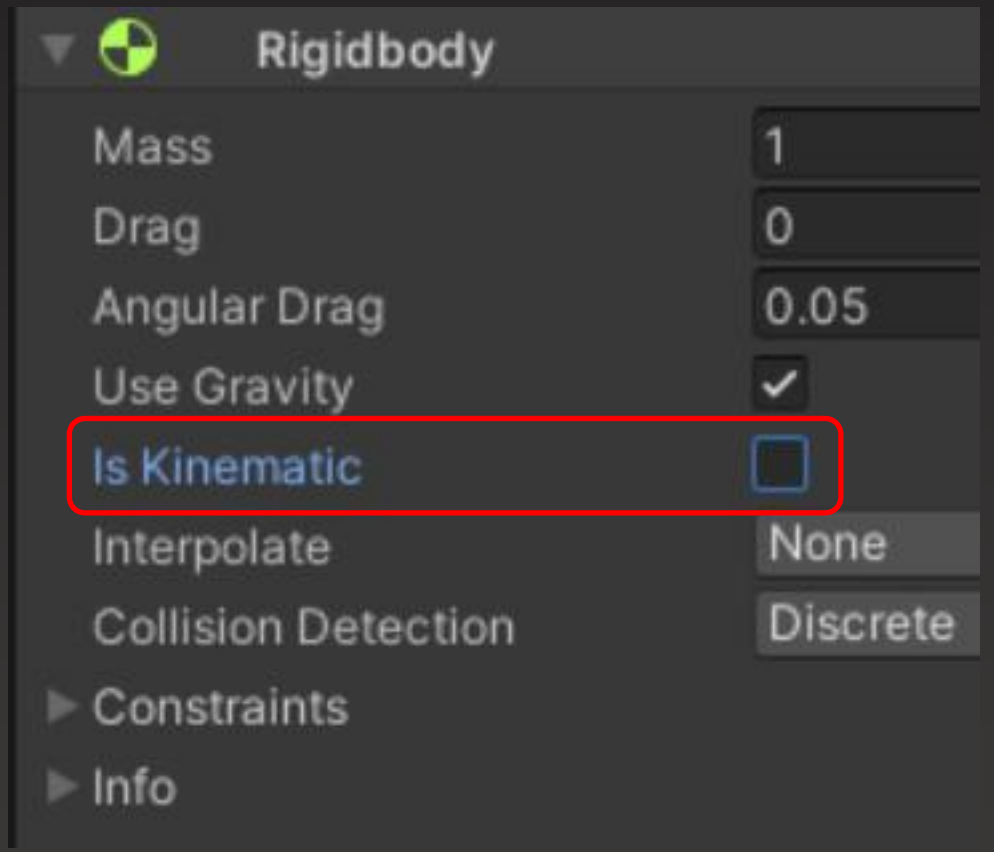
Collision对象相同

4、运动学物体 Kinematic

打开：

- 保持刚体类型，但不受物理控制
- 通过坐标系进行运动

刚体组件



5、碰撞体处理分类：

静态碰撞体
Static Collider

- 无刚体组件的碰撞体

不位移、无物理影响
如：地形、障碍物等

刚体碰撞体
Rigidbody Collider

- 有刚体组件的碰撞体

Rigidbody组件，
受物体力学和脚本力学影响

运动学碰撞体
Kinematic Collider

- 有刚体组件的碰撞体
- 打开了Kinematic

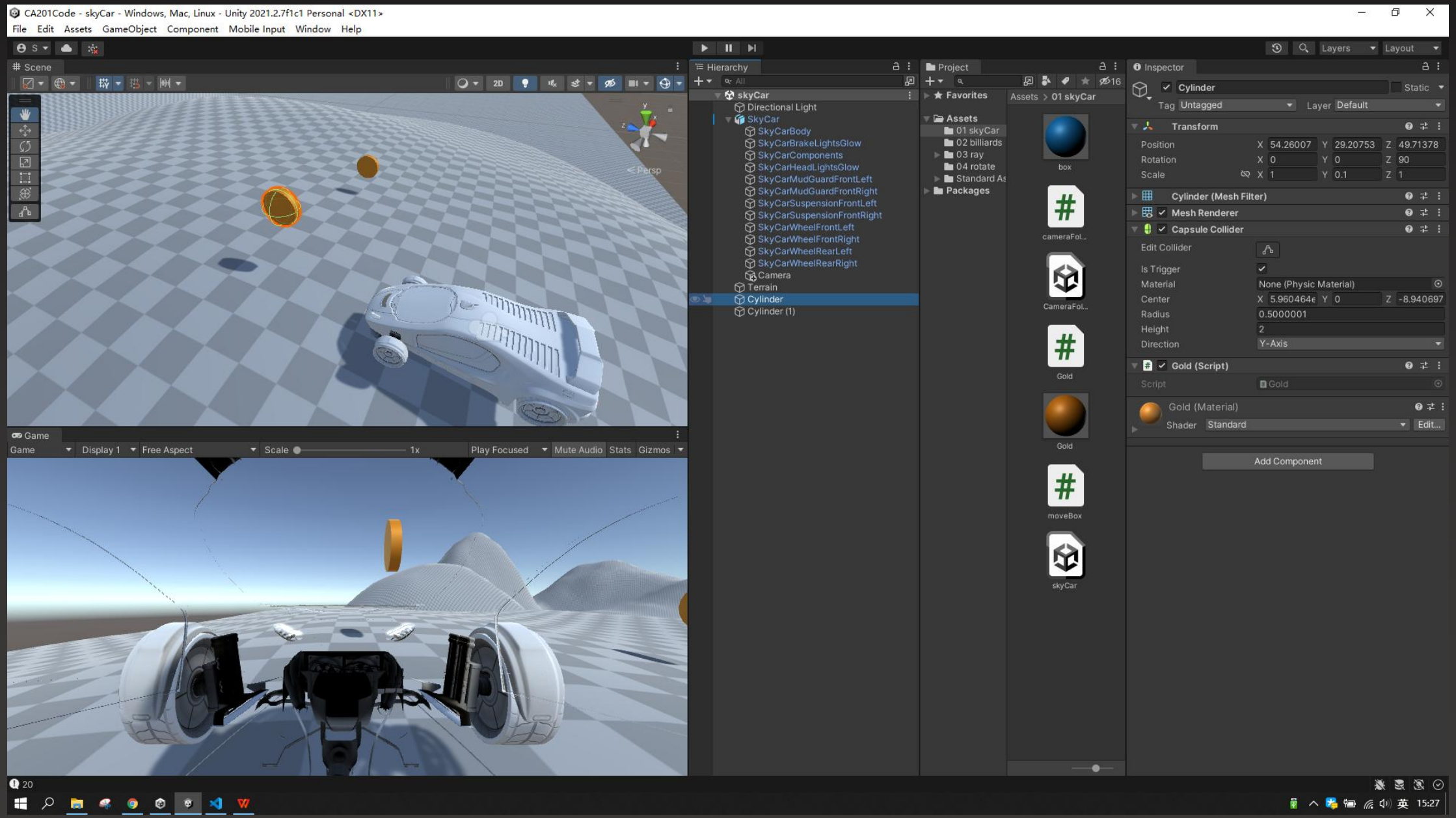
不受物理力学影响，
通过脚本Transform移动

6、碰撞事件与碰撞体类型的关系：

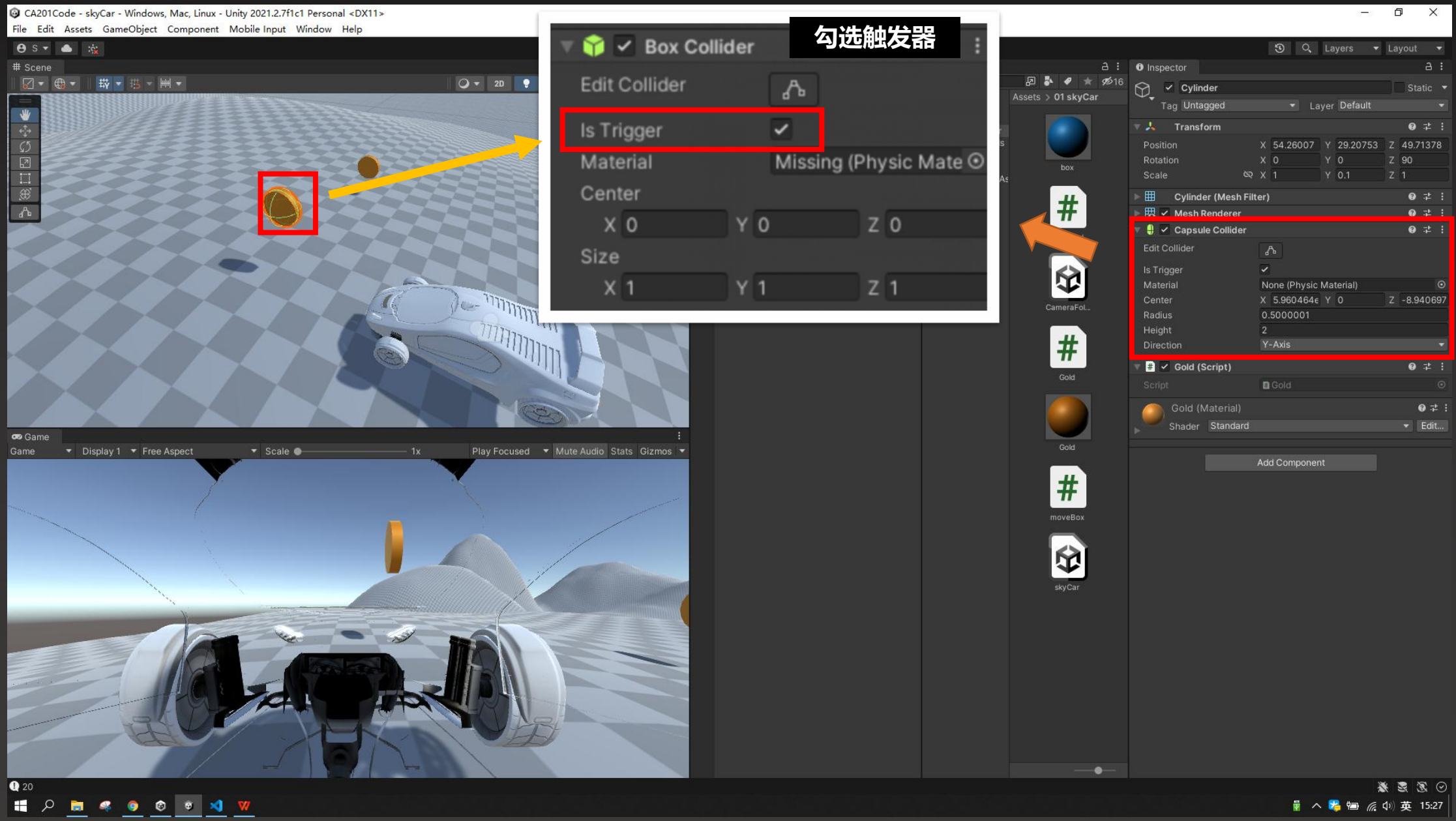
（1）碰撞器方法　OnCollisionEnter　　（2）触发器方法　OnTriggerEnter

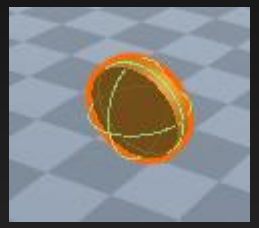
	Static Collider	Rigidbody Collider	Kinematic Collider	Static Trigger	Rigidbody Trigger	Kinematic Trigger
Static Collider		Y			Y	Y
Rigidbody Collider	Y	Y	Y	Y	Y	Y
Kinematic Collider		Y		Y	Y	Y
Static Trigger		Y	Y		Y	Y
Rigidbody Trigger	Y	Y	Y	Y	Y	Y
Kinematic Trigger	Y	Y	Y	Y	Y	Y

拓展自研：吃金币


CA201Code - skyCar - Windows, Mac, Linux - Unity 2021.2.7f1c1 Personal <DX11>
File Edit Assets GameObject Component Mobile Input Window Help
Scene
Persp
Game
Game Display 1 Free Aspect Scale 1x Play Focused Mute Audio Stats Gizmos
Hierarchy
skyCar
Directional Light
SkyCar
SkyCarBody
SkyCarBrakeLightsGlow
SkyCarComponents
SkyCarHeadLightsGlow
SkyCarMudGuardFrontLeft
SkyCarMudGuardFrontRight
SkyCarSuspensionFrontLeft
SkyCarSuspensionFrontRight
SkyCarWheelFrontLeft
SkyCarWheelFrontRight
SkyCarWheelRearLeft
SkyCarWheelRearRight
Camera
Terrain
Cylinder
Cylinder (1)
Project
Favorites
Assets
01 skyCar
02 billiards
03 ray
04 rotate
Standard As
Packages
Assets > 01 skyCar
box
cameraFol...
CameraFol...
Gold
Gold
moveBox
skyCar
Inspector
Cylinder
Static
Tag Untagged
Layer Default
Transform
Position X 54.26007 Y 29.20753 Z 49.71378
Rotation X 0 Y 0 Z 90
Scale X 1 Y 0.1 Z 1
Cylinder (Mesh Filter)
Mesh Renderer
Capsule Collider
Edit Collider
Is Trigger
Material None (Physic Material)
Center X 5.960464e Y 0 Z -8.940697
Radius 0.5000001
Height 2
Direction Y-Axis
Gold (Script)
Script Gold
Gold (Material)
Shader Standard Edit...
Add Component
15:27

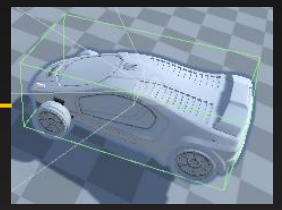
吃金币游戏


勾选触发器
Box Collider
Edit Collider
Is Trigger
Material
Missing (Physic Mate
Center
X 0 Y 0 Z 0
Size
X 1 Y 1 Z 1
CA201Code - skyCar - Windows, Mac, Linux - Unity 2021.2.7f1c1 Personal <DX11>
File Edit Assets GameObject Component Mobile Input Window Help
Scene
Game
Display 1
Free Aspect
Scale
1x
Play Focused
Mute Audio
Stats
Gizmos
Inspector
Cylinder
Static
Tag Untagged
Layer Default
Transform
Position X 54.26007 Y 29.20753 Z 49.71378
Rotation X 0 Y 0 Z 90
Scale X 1 Y 0.1 Z 1
Cylinder (Mesh Filter)
Mesh Renderer
Capsule Collider
Edit Collider
Is Trigger
Material None (Physic Material)
Center X 5.960464e Y 0 Z -8.940697
Radius 0.5000001
Height 2
Direction Y-Axis
Gold (Script)
Script Gold
Gold (Material)
Shader Standard
Edit...
Add Component
Assets > 01 skyCar
box
CameraFol...
Gold
Gold
moveBox
skyCar
Layers
Layout
英 15:27



金币代码

```
public class Gold : MonoBehaviour
{
    GameObject car;

    void Start()
    {   }

    void Update()
    {       transform.Rotate(1f,1f,1f);    }

    void OnTriggerEnter(Collider other)
    {
        car=GameObject.Find(other.name);
        moveBox script=car.GetComponent<moveBox>();
        script.addScore();

        Destroy(gameObject);
    }


}
```



飞车代码完善

```
using UnityEngine;

public class moveBox : MonoBehaviour
{
    float offsetX, offsetZ, wheelSpeed;
    int moveSpeed = 10;
    int rotateSpeed = 80;
    int score=0;
    GameObject leftWheel, rightWheel;

    public void addScore()
    {
        score+=10;
        print(score);
    }

    void Start()
    {
        leftWheel = GameObject.Find("SkyCarWheelFrontLeft");
        rightWheel = GameObject.Find("SkyCarWheelFrontRight");
    }

    void Update()
    {
        offsetX = Input.GetAxis("H") * Time.deltaTime * rotateSpeed;
        offsetZ = Input.GetAxis("V") * Time.deltaTime * moveSpeed;
        wheelSpeed = Input.GetAxis("V") * 5;
        transform.Translate(0f, 0f, offsetZ);
        if (offsetZ != 0)
        {
            transform.Rotate(0f, offsetX, 0f);
        }

        leftWheel.transform.Rotate(wheelSpeed, 0f, 0f);
        rightWheel.transform.Rotate(wheelSpeed, 0f, 0f);
    }

}
```