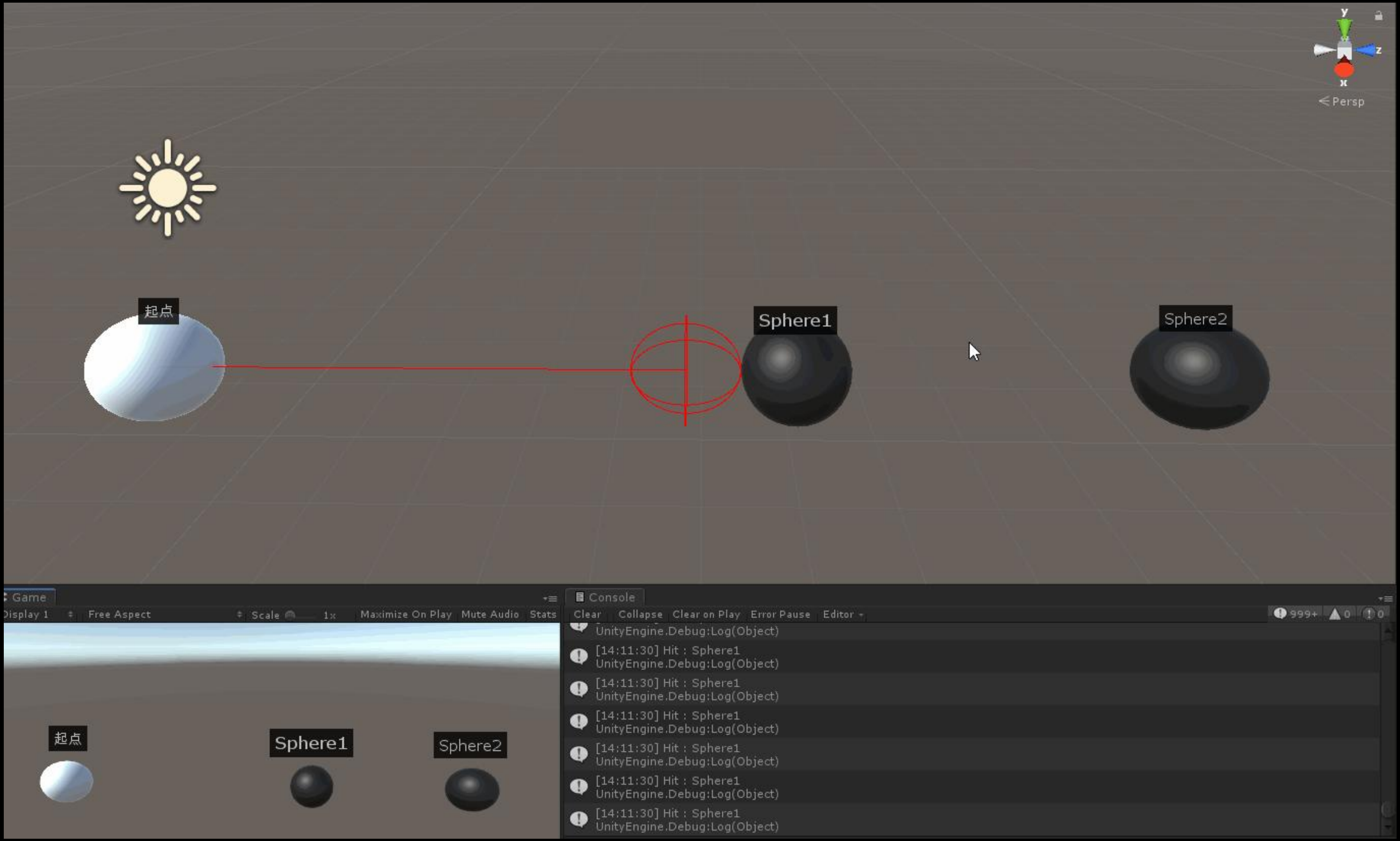
射线基础

大数据与物联网学院　邵亮

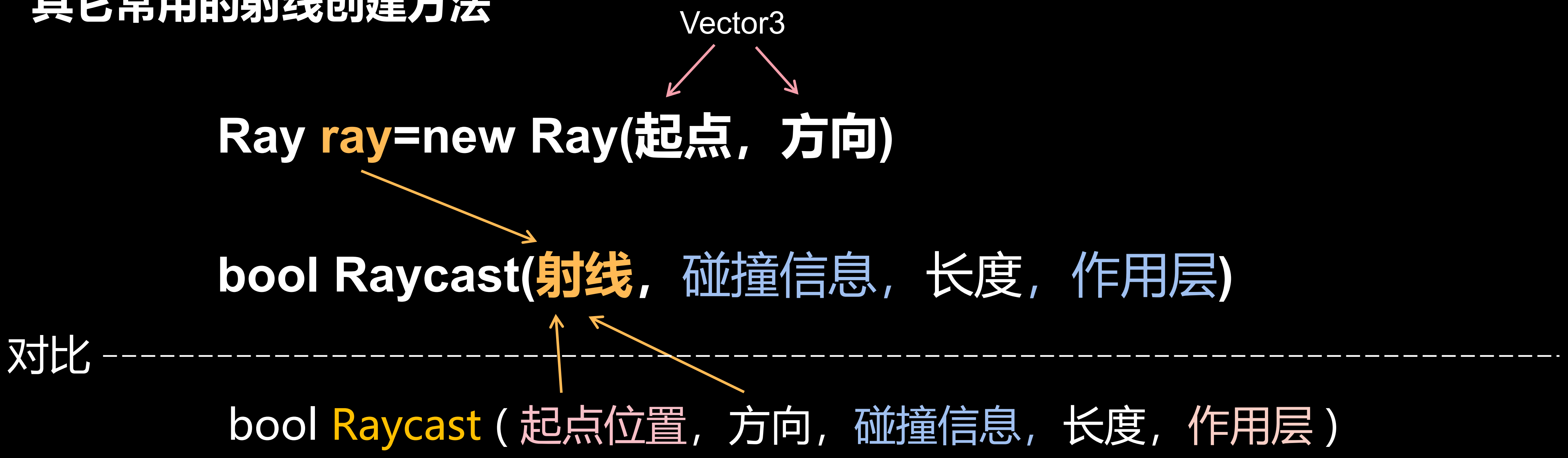
核心知识：射线



```
UnityEngine.Debug:Log(Object)
[14:11:30] Hit : Sphere1
UnityEngine.Debug:Log(Object)
[14:11:30] Hit : Sphere1
UnityEngine.Debug:Log(Object)
[14:11:30] Hit : Sphere1
UnityEngine.Debug:Log(Object)
[14:11:30] Hit : Sphere1
UnityEngine.Debug:Log(Object)
[14:11:30] Hit : Sphere1
UnityEngine.Debug:Log(Object)
[14:11:30] Hit : Sphere1
UnityEngine.Debug:Log(Object)
```

GIF动画

其它常用的射线创建方法



Vector3

Ray ray=new Ray(起点，方向)

bool Raycast(射线，碰撞信息，长度，作用层)

对比

bool Raycast（起点位置，方向，碰撞信息，长度，作用层）

核心知识：发射射线

Physics . Raycast()

物理学对象

RaycastHit hit;

point : 碰撞点（世界坐标）

collider：碰撞体 → gameObject：对象

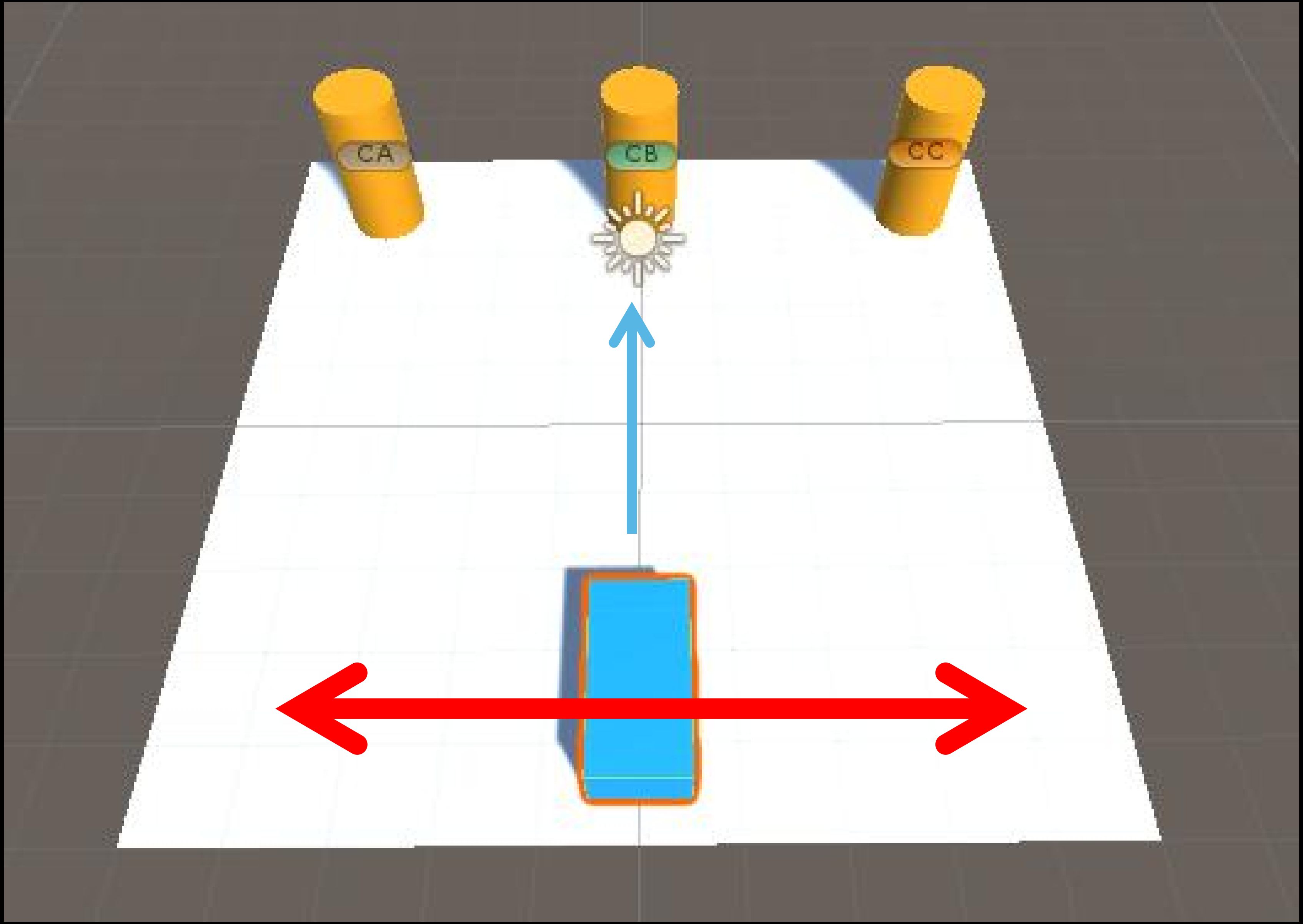
transform：变型组件

bool Raycast(起点位置，方向，碰撞信息，长度，作用层)

```
if (Physics.Raycast ( transform.position， Vector3.forward， out  hit，  1000 ))
{
      print(hit.collider.gameObject);
}
```

LayerMask.GetMask("car")

案例：坦克射击模拟



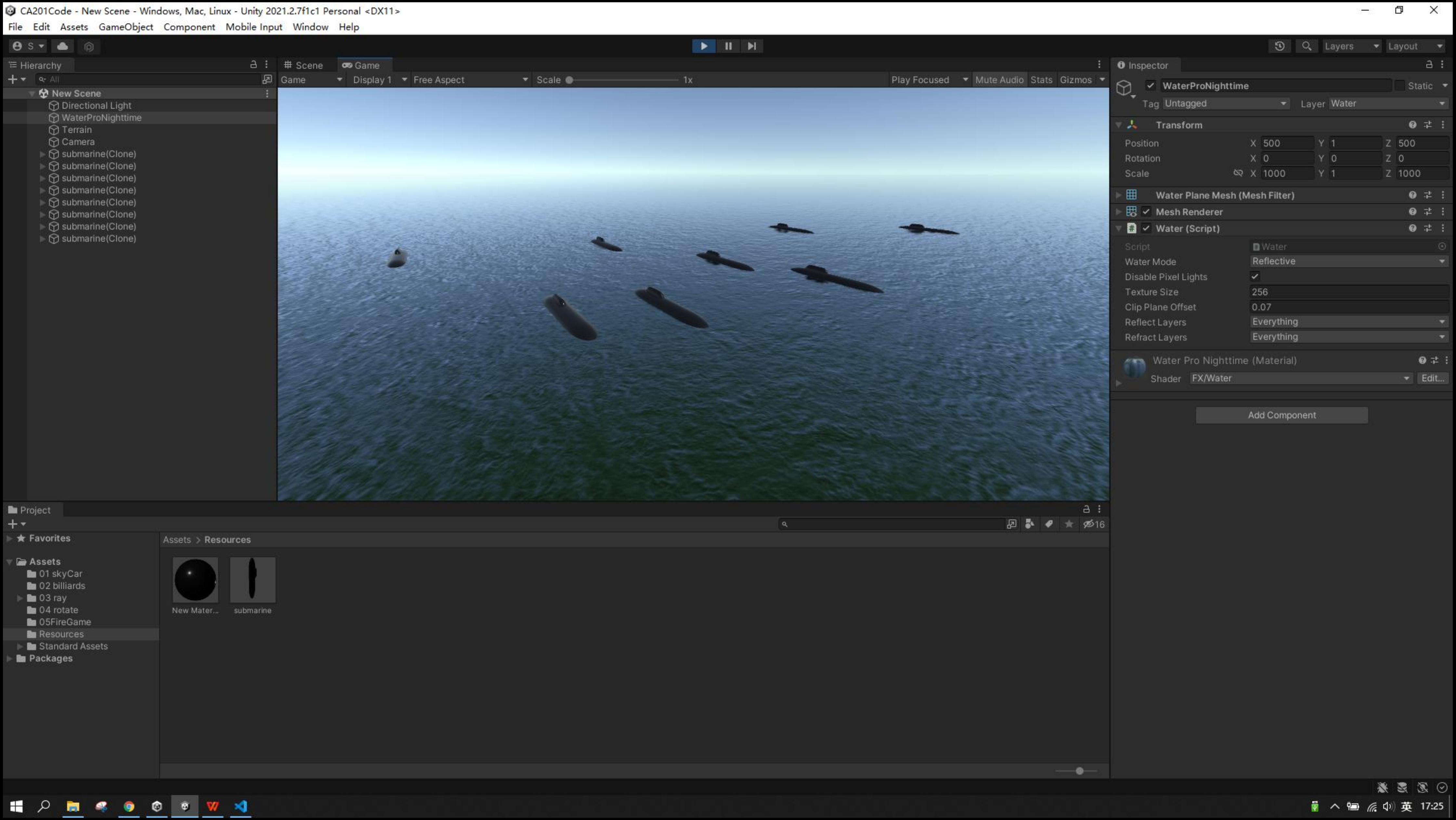
```
public class tank : MonoBehaviour
{
    // Start is called before the first frame update
    float offsetX;
    RaycastHit hit;
    Ray ray;
    void Start()
    {    }

    void Update()
    {
        offsetX=Input.GetAxis("H")*0.1f;
        transform.position+=new Vector3(offsetX,0f,0f);

        if(Input.GetKeyDown(KeyCode.Space))
          {
              ray=new Ray(transform.position,new Vector3(0f,0f,1000f));
              if(Physics.Raycast(ray,out hit,1000))
              {
                  Debug.DrawLine(transform.position,hit.point,Color.red);
                  print(hit.collider.gameObject.name);
                  Destroy(hit.collider.gameObject);
              }


          }
    }
}
```

案例：祖国的潜艇



CA201Code - New Scene - Windows, Mac, Linux - Unity 2021.2.7f1c1 Personal <DX11>

Hierarchy:

- New Scene
  - Directional Light
  - WaterProNighttime
  - Terrain
  - Camera
  - submarine(Clone)
  - submarine(Clone)
  - submarine(Clone)
  - submarine(Clone)
  - submarine(Clone)
  - submarine(Clone)
  - submarine(Clone)
  - submarine(Clone)

Inspector:

WaterProNighttime Static

Tag Untagged Layer Water

Transform

Position	X 500	Y 1	Z 500
Rotation	X 0	Y 0	Z 0
Scale	X 1000	Y 1	Z 1000

Water Plane Mesh (Mesh Filter)

Mesh Renderer

Water (Script)

Script	Water
Water Mode	Reflective
Disable Pixel Lights	✓
Texture Size	256
Clip Plane Offset	0.07
Reflect Layers	Everything
Refract Layers	Everything

Water Pro Nighttime (Material)

Shader FX/Water

Project:

Assets > Resources

- Assets
  - 01 skyCar
  - 02 billiards
  - 03 ray
  - 04 rotate
  - 05FireGame
  - Resources
  - Standard Assets
- Packages

New Mater... submarine

17:25

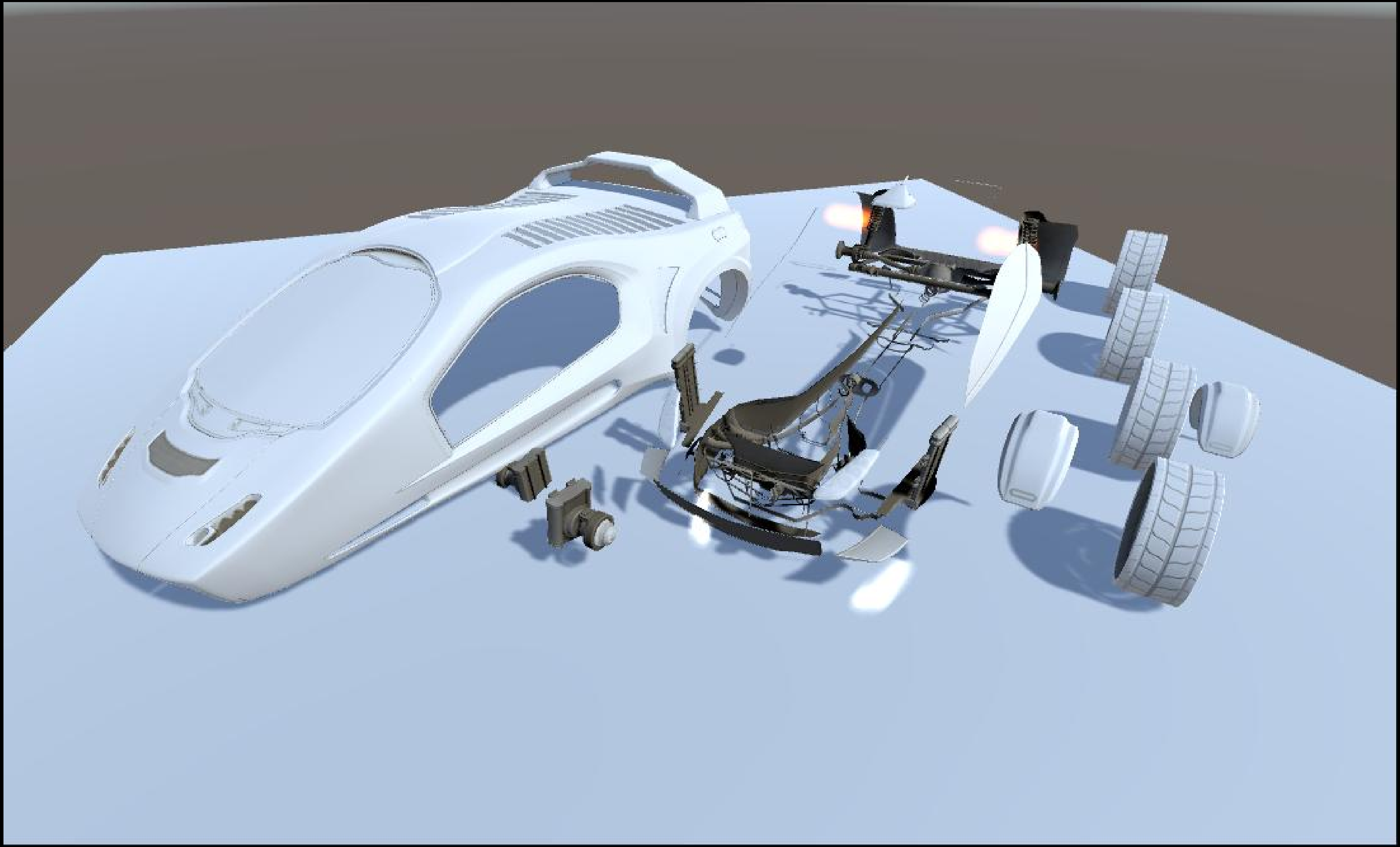
代码

```
public class sm : MonoBehaviour
{
   public GameObject sm;  // 挂载潜艇预制体;
   GameObject smCopy;
   RaycastHit hit;
   void Start()
   {
         }
   void Update()
   {
       if (Input.GetMouseButtonDown(0))
      {
        Ray r = Camera.main.ScreenPointToRay(Input.mousePosition);
        if (Physics.Raycast(r, out hit))
        {
           smCopy = GameObject.Instantiate(sm);
           smCopy.transform.position = hit.point;

        }
      }
   }
}
```

鼠标抓取物体拖动

案例：汽车结构认知虚拟仿真



核心知识：鼠标指向的射线

如何判断鼠标到的场景对象?

```
Ray ray = Camera.main.ScreenPointToRay ( Input.mousePosition );
```

射线类型

从摄影机位置发射射线到
指定点位置

鼠标的屏幕坐标

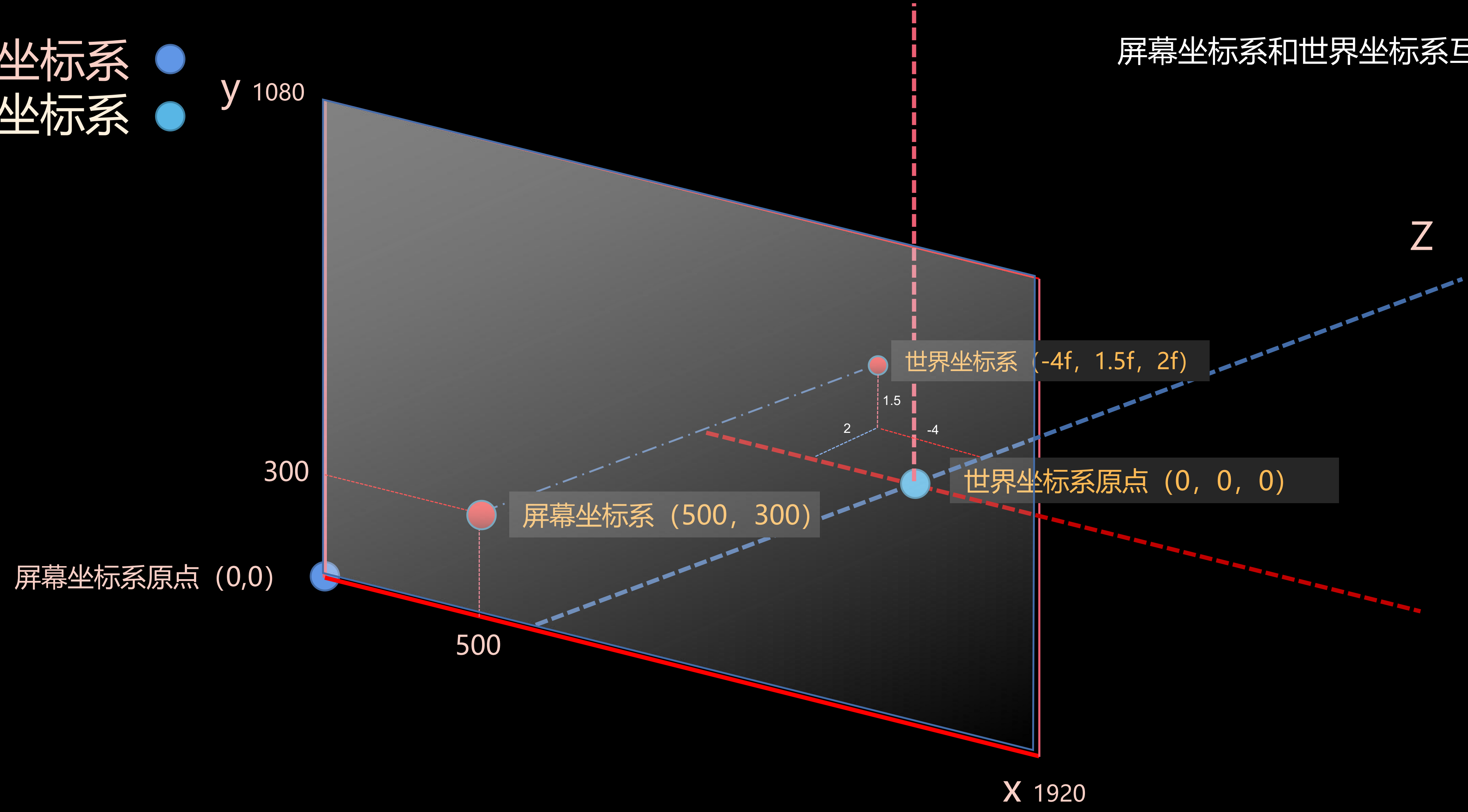
```
if (Physics.Raycast( ray , out hitObj)
 {
      print(hitObj.transform.gameObject);
 }
```

核心知识：认识两个世界坐标系

（动画版）

1.屏幕坐标系 ●
2.世界坐标系 ●

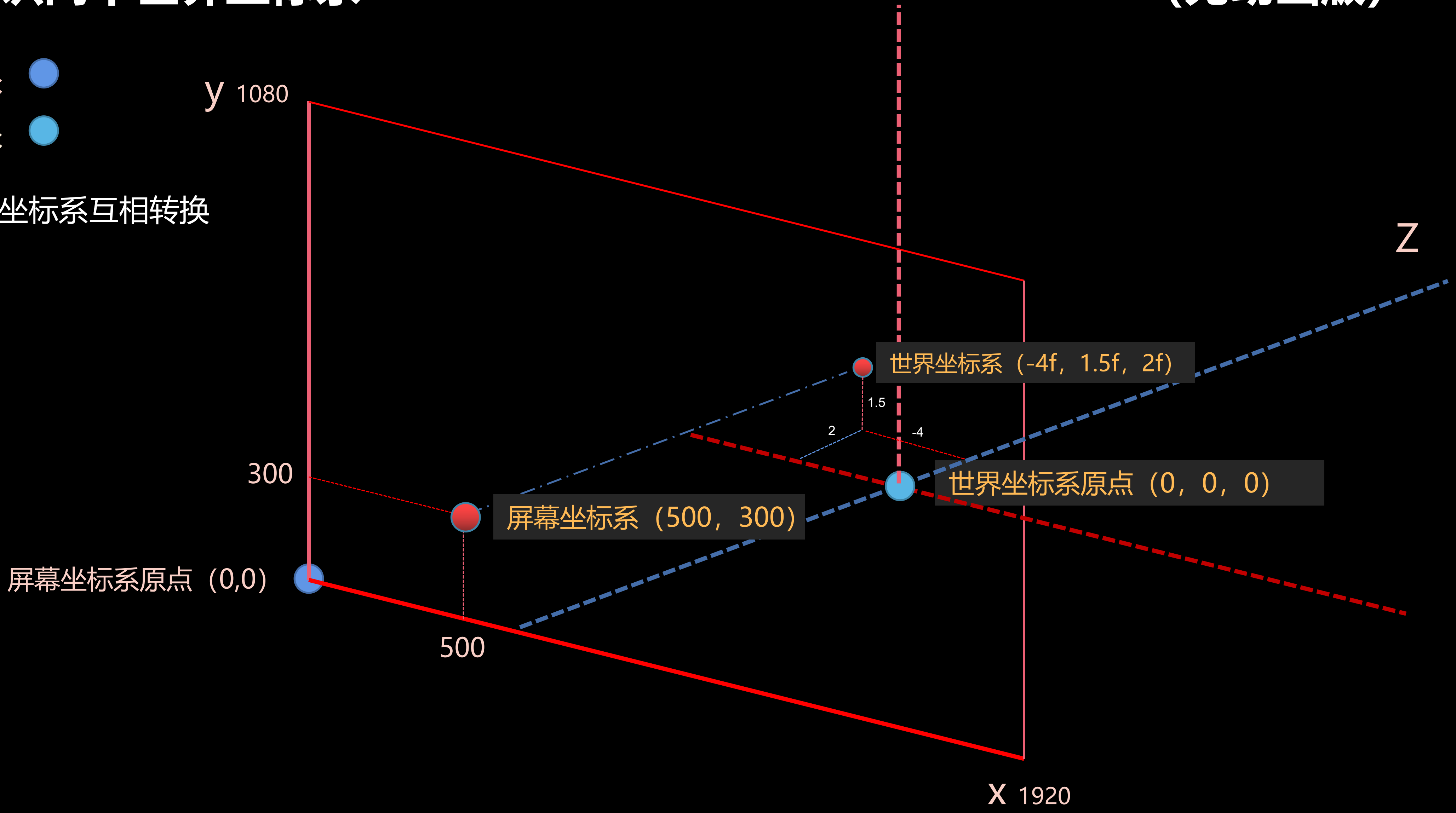
屏幕坐标系和世界坐标系互相转换



核心知识：认识两个世界坐标系 （无动画版）

1.屏幕坐标系 ●
2.世界坐标系 ●

屏幕坐标系和世界坐标系互相转换



核心知识：认识两个世界坐标系

屏幕坐标系和世界坐标系互相转换

1.世界坐标系 > 屏幕坐标系

Camera.main.**World**ToScreenPoint (Vector3)

2.屏幕坐标系 > 世界坐标系

Camera.main.**Screen**ToWorldPoint (Vector3)

核心代码

```
 if (Draping)  //当鼠标拖动时
{
          // 获得目标对象的屏幕坐标;
           Vector3 targetScreenPos = Camera.main.WorldToScreenPoint(target.transform.position);
         // 生成鼠标的三维屏幕坐标;
           Vector3 mousePos = new Vector3(Input.mousePosition.x, Input.mousePosition.y, targetScreenPos.z);
         // 生成鼠标的三维世界坐标;
           pos = Camera.main.ScreenToWorldPoint(mousePos);
         // 设置目标对象位置为鼠标世界坐标加目标与鼠标的距离
           target.transform.position = pos ;
}
```

案例实施

（WPS播放+放大）

```
public class CameraMove : MonoBehaviour
{

    Vector3 pos;
    GameObject target;
    RaycastHit hitObj;
    bool Draping=false;
    void Start()
    {
        target = GameObject.Find("objTemp");

    }
    void Update()
    {

        if (Input.GetMouseButtonDown(0))
        {
            Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
             if (Physics.Raycast(ray, out hitObj, 1000f,LayerMask.GetMask("car")))
            {
                Draping=true;
                target = hitObj.collider.gameObject;
            }

        }
        if (Input.GetMouseButtonUp(0))
        {
            Draping=false;
            target = GameObject.Find("objTemp");
        }

        if (Draping)
        {
            Vector3 targetScreenPos = Camera.main.WorldToScreenPoint(target.transform.position);
            Vector3 mousePos = new Vector3(Input.mousePosition.x, Input.mousePosition.y, targetScreenPos.z);
            pos = Camera.main.ScreenToWorldPoint(mousePos);

            target.transform.position = pos;
        }

         zoomCamera();  //自定义方法，拖进摄影机，方便观察抓取到的物体
    }
}
```

```
void zoomCamera()
    {
        if (Draping)
        {
            float v = Input.GetAxis("V") * Time.deltaTime * 10;
            offsetV=(transform.position-target.transform.position)*v;
            transform.position += offsetV*-1f ;

            float h = Input.GetAxis("H") * Time.deltaTime * 100*-1f;
            target.transform.Rotate(0f, h, 0f);

        }
        else
        {
            float v = Input.GetAxis("V") * Time.deltaTime * 10;
            float h = Input.GetAxis("H") * Time.deltaTime * 10;
            float w = Input.GetAxis("Mouse ScrollWheel") * Time.deltaTime * 100;

            transform.Translate(h, v, w);

        }
```