unity

DisColl.enabled = false; enabled表示这个wuti是否被启用。还可以运用其他方面，比如：GetComponent<AudioSource>().enabled = false; 让当前的音乐禁用。

SceneManager.LoadScene(SceneManager.GetActiveScene().name); 加载场景。
SceneManager.GetActiveScene().name, 获得当前场景的名称
SceneManager.GetActiveScene().buildIndex，获得当前场景的编号

Time.timeScale = 0f; 更改游戏中的运行时间 ，正常的速度为1，停止不动为0。

player = Game0bj ect . FindGame0bj ec tWithTag("Player")查找标签为player 的物体

```
0 个引用
void Update () {
    float vertical = Input.GetAxis("Vertical");
    float horizontal = Input.GetAxis("Horizontal");
    Vector3 dir = new Vector3(horizontal, 0, vertical);
    if (dir != Vector3.zero)
    {
        transform.rotation = Quaternion.LookRotation(dir);
        transform.Translate(Vector3.forward * 2 * Time.deltaTime);
    }
```

3d移动

void Update () {
float vertical = Input . GetAxis("Vertical");
float horizontal = Input . GetAxis("Horizontal");
Vector3 dir = new Vector3 (horizontal, 0,vertical);
if (dir != Vector3. zero)
{
transform. rotation = Quaternion . LookRotation(dir);
transform . Translate(Vector3. forward * 2 * Time. deltaTime) ;}

audioMixer.SetFloat("BgmMusion", sliderVolum.value);//调节声音

[(方法/函数重载Method Overloading] (面试常问的小题)指的就是方法/函数名称相同，但是括号内的参数不同，

I 把方法当作参数来传递的话，就要用到委托

```
//弧度转角度
float rad = 1;
float anger = rad * MathF. Rad2Deg;
//角度转弧度
anger = 1;
rad = anger * Mathf . Deg2Rad;
```

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

public class CursorManager : MonoBehaviour
{
private float h;
private float v;
private bool canClick;
private void Update()
{
h = Input.mousePosition.x;
v = Input.mousePosition.y;
canClick = ObjectAtMousePosition();
  if (canClick && Input.GetMouseButton(0))
    {
        //检测鼠标互动情况。
        ClickAction(ObjectAtMousePosition().gameObject);
    }
}
private void ClickAction(GameObject clickObject)
{
    switch (clickObject.tag)
    {
        case "Teleport":
            var teleport = clickObject.GetComponent<Telepoit>();
            teleport?.TeleportToScene();
            break;
        case "Item":
            var item = clickObject.GetComponent<Item>();
            item?.ItemClick();

            break;
    }
}
```

```
36 private Vector3 mouseWorldPoint => Camera.main.ScreenToWorldPoint(new Vector3(h, v, 0));
37 private Collider2D ObjectAtMousePosition()
38 {
39     return Physics2D.OverlapPoint(mouseWorldPoint);//检查碰撞体是否与空间中的某个点重叠.
40 }
```

}

若当你使用Random。Range没有反应的时候，在前面加一个UnityEngine

eg：float randomX = UnityEngine.Random.Range(-patrolRange, patrolRange);

UnityEngine.Random.value会随机生成一个0--1之间的数值，可以用来判断是否暴击。

eg: characterState.isCritical = UnityEngine.Random.value > (characterState.attackDate.criticalChance);

characterState.attackDate.criticalChance):攻击数值中的暴击率

//射击瞄准点

```
Ray ray =Camera.main.ScreenPointToRay(Input.mousePosition);
RaycastHit hit;
if(Physics.Raycast(ray,out hit, Mathf.Infinity)&&b!=null)
{
b.transform.LookAt(hit.point);
}
```

2D中 RaycastHit2D hit = Physics2D.Raycast(pos, Vector2.right, 4f);

返回的hit包含所碰撞的信息。

3D中 RaycastHit hit = Physics.Raycast(pos, Vector2.right, 4f);

返回的是个bool值