屏幕大小

private Vector2 leftPos = Camera.main.ViewportToWorldPoint(new Vector2(0, 0));//左下

private Vector2 rightPos = Camera.main.ViewportToWorldPoint(new Vector2(1, 1));//右上

每几秒生成

private void TimeCoroutine()

{

time -= Time.deltaTime;

float temp = time;

if (temp <= 0)

{

time = 15f;

}

}

//四元数转化成欧拉角 eulerAngles为Quaternion类的一个Get方法 均可直接调用

Vector3 p = transform.rotation.eulerAngles;

//欧拉角转换成四元数

Quaternion rotation = Quaternion.Euler(p)

时间

Void Update(){

DateTime dt = DateTime.Now;

DateTimeOffset offset = new DateTimeOffset(dt);

nowTime = offset.ToUnixTimeSeconds();

temp = nowTime - anxiaTime;

dojishi = 10 - temp;

if (dojishi<0)

{

text.gameObject.SetActive(false);

isCilck = false;

isReCilck = true;

moveSpeed = 1f;

}

ShowTime();

}

private void ShowTime()

{

float Min = (int)dojishi / 60;

float Second = (int)dojishi % 60;

float Hour = Min / 60;

if (Second > 59.0)

{

Second = 10 - (Min \* 60);

}

if (Min > 59.0)

{

Min = Min - (Hour \* 60);

}

if (Hour >= 24.0)

{

Hour = 0;

}

//text.text = string.Format("{0:00}:{1:00}", Min, Second);

text.text ="(" +Second.ToString() + "s)";

}

当前ui的左坐标

GetComponent<RectTransform>().rect.xMin;

GetComponent<RectTransform>().rect.x;

当前ui的右坐标

GetComponent<RectTransform>().rect.xMax;

当前ui的高度

GetComponent<RectTransform>().rect.height;

当前ui底部相对于顶部的相对长度，负数为向下延展，同理则反

GetComponent<RectTransform>().rect.y;

当前ui的宽高

GetComponent<RectTransform>().sizeDelta.x;

GetComponent<RectTransform>().sizeDelta.y;

滑动后移动的位置（横向竖向 ）

transform.Find("Panel/Scroll View").GetComponent<ScrollRect>().verticalNormalizedPosition = 1;

添加里力/扭矩力

transform.GetComponent<Rigidbody2D>().AddForce(pos[a] \* 30, ForceMode2D.Impulse);

transform.GetComponent<Rigidbody2D>().AddTorque(angle[a], ForceMode2D.Impulse);

本地内存

levelNum = PlayerPrefs.GetInt("SharpLevel", 1);

PlayerPrefs.SetInt("SharpLevel", 1);

////详细分解

DOTween.To(

() =>

maskImage.color//我们想改变的对象值

, toColor//每次doTween经过计算得到的结果（当前值到目标值的插值）

=> maskImage.color = toColor, //将计算结果赋值给我们想要改变的对象值

new Color(0, 0, 0, 0), 2f);//目标值，完成动画的时间

transform.DOMove(targetTrans.position, 0.2f)

.OnComplete

(

() =>

{}

);