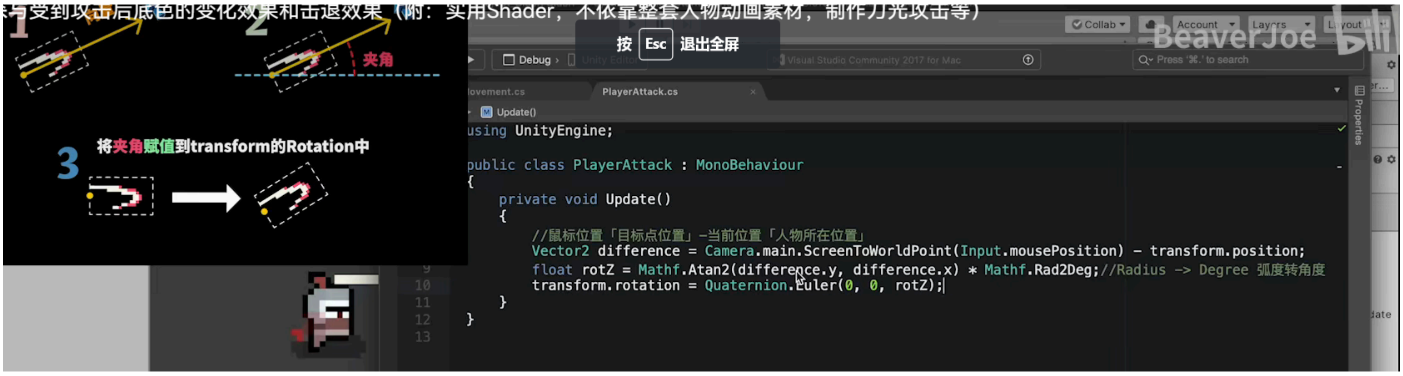
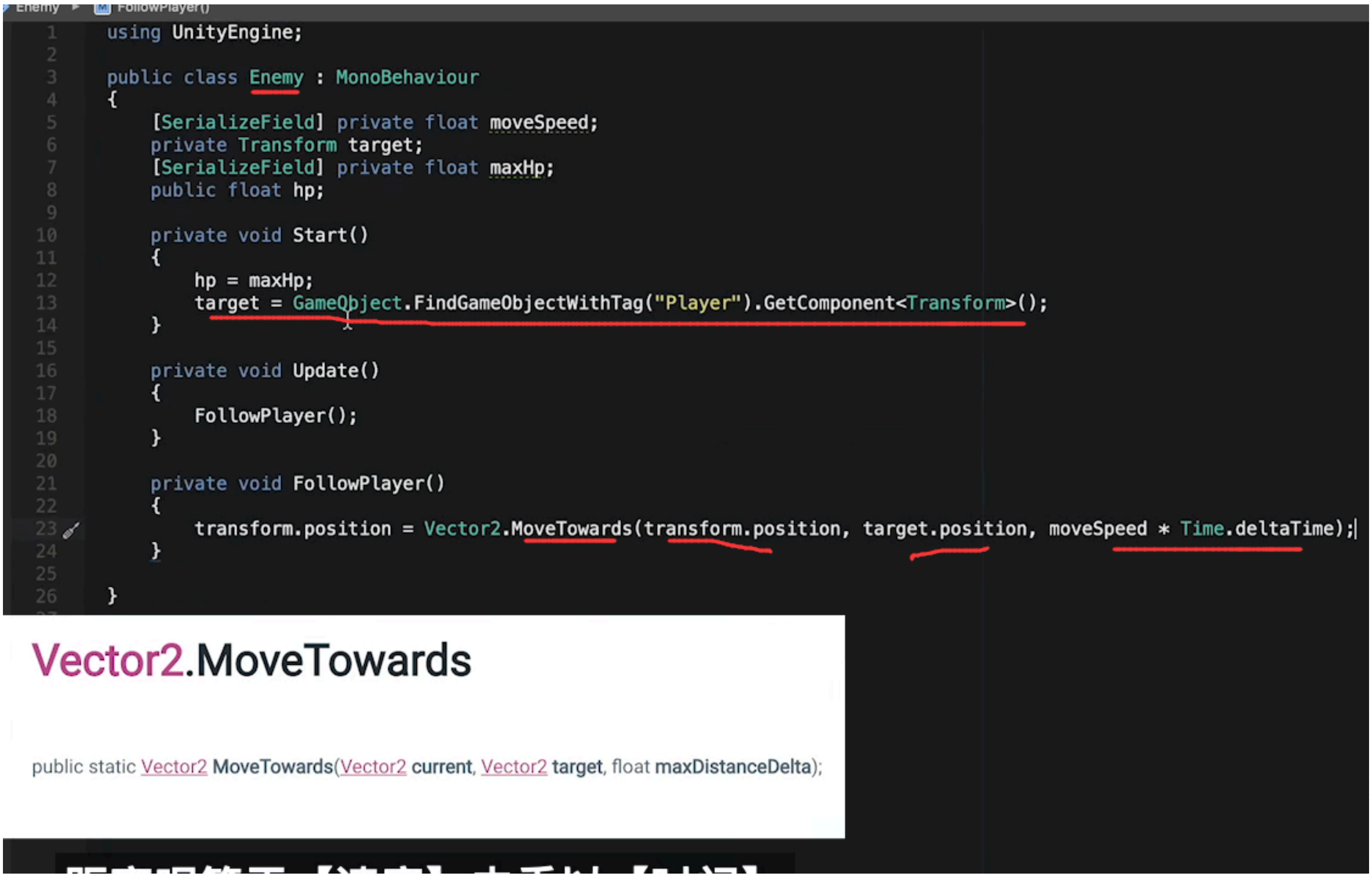
蓄力跳跃

```
public float jumpTime;      //跳跃的最大蓄力时间
float timeJump;             //跳跃当前的蓄力时间
public void Jump()
{
if (Input.GetKeyDown(KeyCode.Space))
{
jumpState = true;  //进入跳跃状态
moveSpeed.y += jumpPower;//初始添加向上的力
timeJump = 0;//蓄力时间清零
}
else if (Input.GetKey(KeyCode.Space) && jumpCount <= 2 && jumpState)
{
timeJump += Time.deltaTime;//蓄力时间增加
if (timeJump < jumpTime)
{
moveSpeed.y += jumpPower;//蓄力
}
}
else if (Input.GetKeyUp(KeyCode.Space))
{
jumpState = false;//退出跳跃状态
timeJump = 0;//蓄力时间清零
}
}
```


与受到攻击后底色的变化效果和击退效果（附：实用Shader，不依靠整套人物动画素材，制作刀光攻击等）
1
2
夹角
3
将夹角赋值到transform的Rotation中
按 Esc 退出全屏
Collab
Account
Layers
Layout
BeaverJoe
Debug
Visual Studio Community 2017 for Mac
Press '⌘.' to search
lovement.cs
PlayerAttack.cs
Update()
Properties

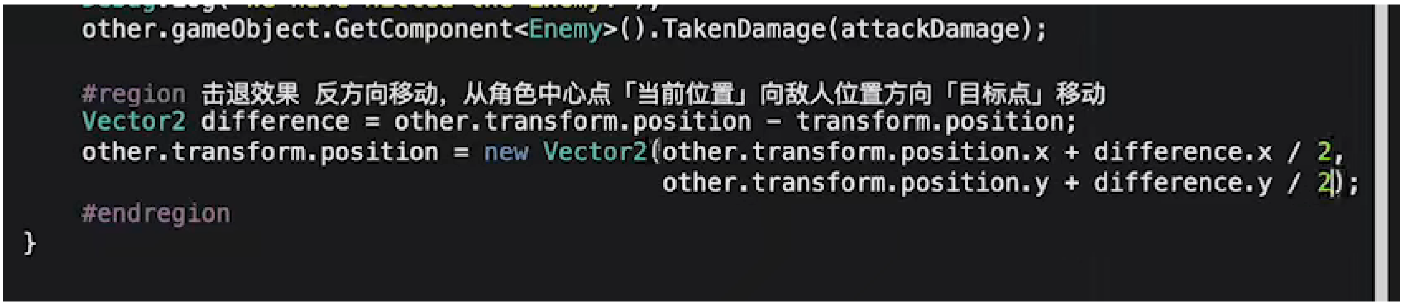
```
using UnityEngine;

public class PlayerAttack : MonoBehaviour
{
    private void Update()
    {
        //鼠标位置「目标点位置」-当前位置「人物所在位置」
        Vector2 difference = Camera.main.ScreenToWorldPoint(Input.mousePosition) - transform.position;
        float rotZ = Mathf.Atan2(difference.y, difference.x) * Mathf.Rad2Deg;//Radius -> Degree 弧度转角度
        transform.rotation = Quaternion.Euler(0, 0, rotZ);
    }
}
```



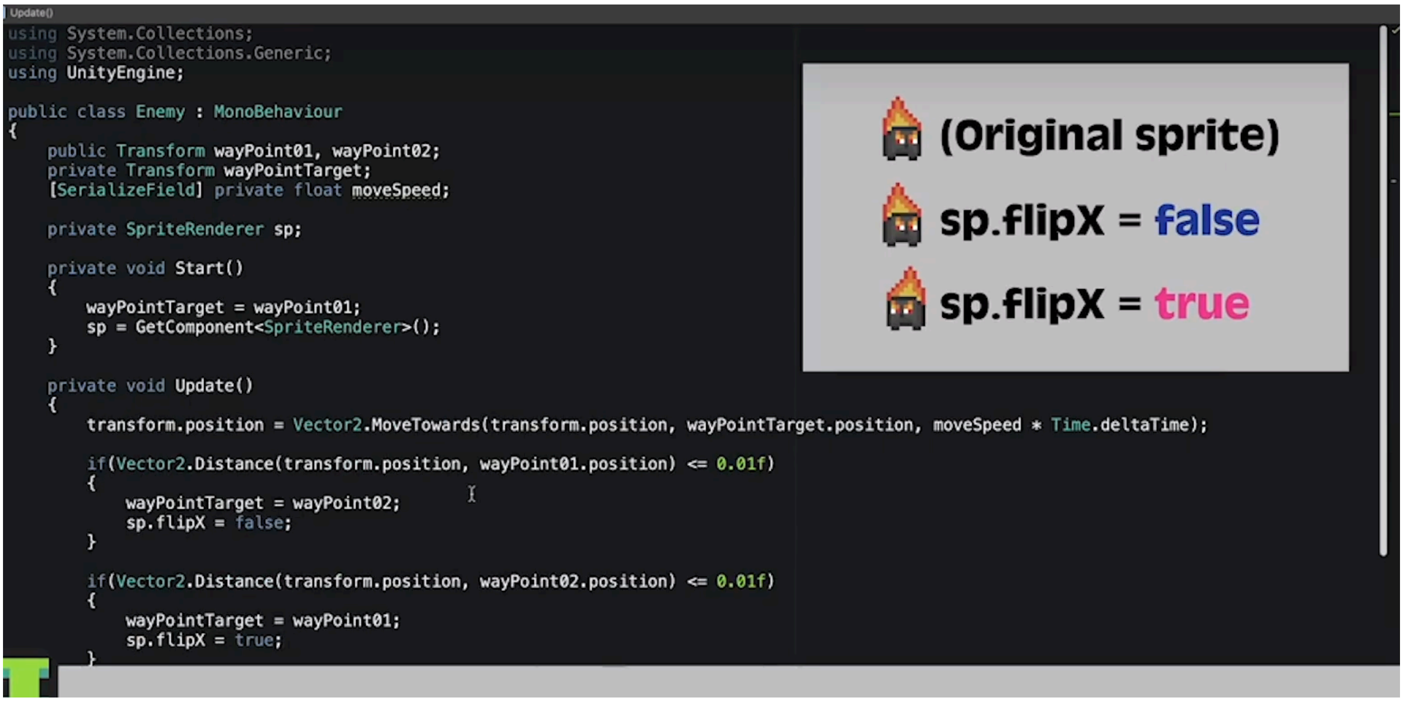
```
using UnityEngine;

public class Enemy : MonoBehaviour
{
    [SerializeField] private float moveSpeed;
    private Transform target;
    [SerializeField] private float maxHp;
    public float hp;

    private void Start()
    {
        hp = maxHp;
        target = GameObject.FindGameObjectWithTag("Player").GetComponent<Transform>();
    }

    private void Update()
    {
        FollowPlayer();
    }

    private void FollowPlayer()
    {
        transform.position = Vector2.MoveTowards(transform.position, target.position, moveSpeed * Time.deltaTime);
    }

}
```

Vector2.MoveTowards

public static Vector2 MoveTowards(Vector2 current, Vector2 target, float maxDistanceDelta);



```
    other.gameObject.GetComponent<Enemy>().TakenDamage(attackDamage);

    #region 击退效果 反方向移动，从角色中心点「当前位置」向敌人位置方向「目标点」移动
    Vector2 difference = other.transform.position - transform.position;
    other.transform.position = new Vector2(other.transform.position.x + difference.x / 2,
                                           other.transform.position.y + difference.y / 2);
    #endregion
}
```



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

public class Enemy : MonoBehaviour
{
    public Transform wayPoint01, wayPoint02;
    private Transform wayPointTarget;
    [SerializeField] private float moveSpeed;

    private SpriteRenderer sp;

    private void Start()
    {
        wayPointTarget = wayPoint01;
        sp = GetComponent<SpriteRenderer>();
    }

    private void Update()
    {
        transform.position = Vector2.MoveTowards(transform.position, wayPointTarget.position, moveSpeed * Time.deltaTime);

        if(Vector2.Distance(transform.position, wayPoint01.position) <= 0.01f)
        {
            wayPointTarget = wayPoint02;
            sp.flipX = false;
        }

        if(Vector2.Distance(transform.position, wayPoint02.position) <= 0.01f)
        {
            wayPointTarget = wayPoint01;
            sp.flipX = true;
        }
```

(Original sprite)

sp.flipX = false

sp.flipX = true