

???????



???????EggDetector?



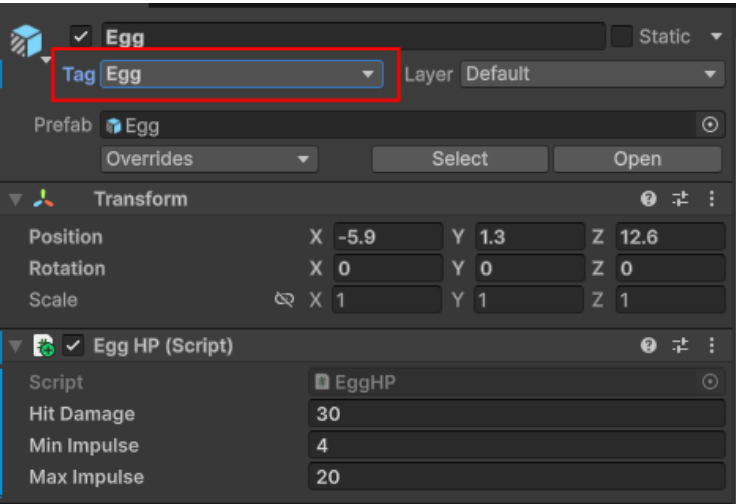
Egg

```
// 
public class EggDetector : MonoBehaviour
{
    // 
    public event System.Action<GameObject> Detected;

    private void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Egg"))
            Detected?.Invoke(other.gameObject);
    }
}
```



Egg





GameManager ???

?????

```
public class GameManager : MonoBehaviour
{
    // 卵のPrefab
    [SerializeField]
    private GameObject eggPrefab;

    // 卵のスタート位置
    [SerializeField]
    private Transform eggStart;

    // 卵の検出範囲
    [SerializeField]
    private EggDetector goal;

    // 卵を作成
    private void CreateEgg()
    {
        // 卵のPrefabをInstantiateして、eggStartの位置に配置
    }
}
```

```

        var egg = Instantiate(eggPrefab);
        egg.transform.position = eggStart.position;
    }

    // 鸡蛋碰撞检测
    private void OnEggDetected(GameObject egg)
    {
        // 鸡蛋碰撞检测
        Destroy(egg);
        CreateEgg();
    }

    private void Start()
    {
        // 鸡蛋碰撞检测
        CreateEgg();

        // 鸡蛋碰撞检测
        goal.Detected += OnEggDetected;
    }

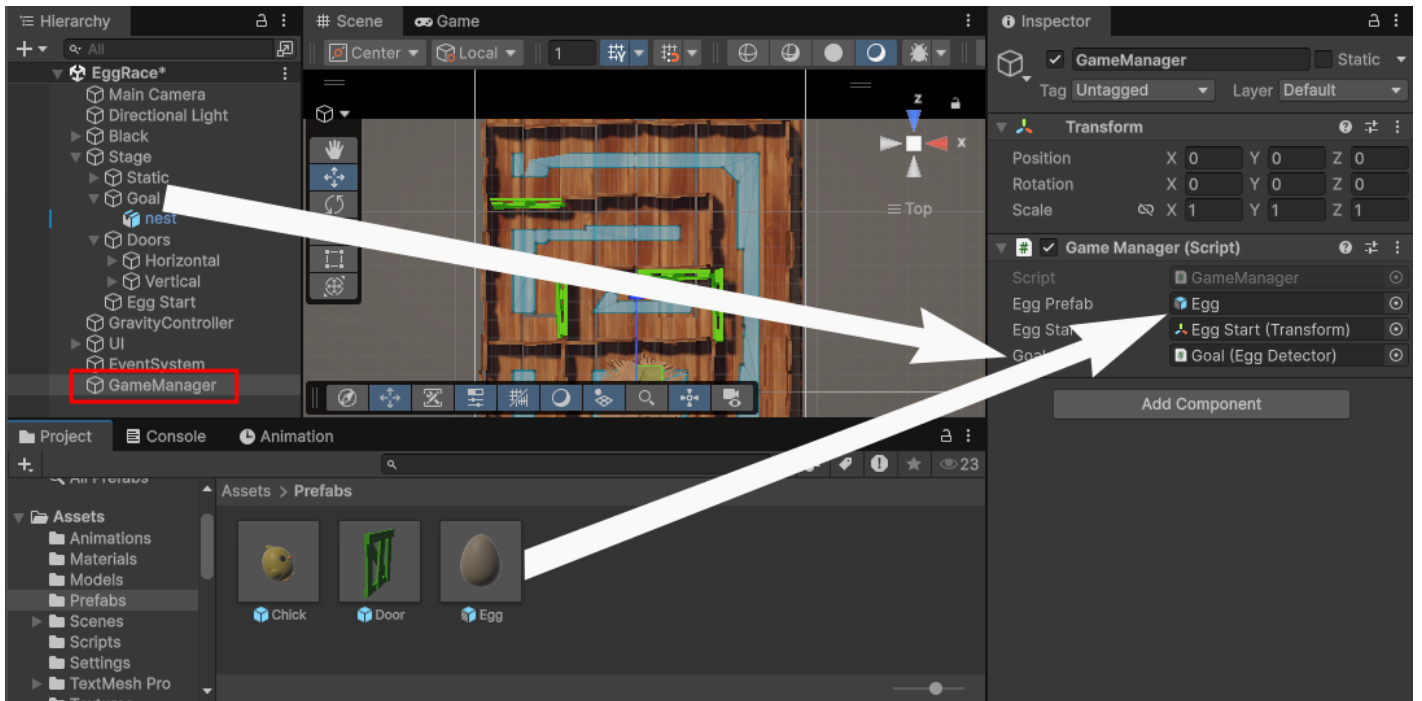
    private void OnDestroy()
    {
        // 鸡蛋碰撞检测
        goal.Detected -= OnEggDetected;
    }
}

```

Unity???

Diagram illustrating memory buffers:

- Top buffer: Labeled "GameManager", contains 16 slots, with the first 15 slots filled with blue blocks.
- Bottom buffer: Empty, contains 32 slots.



??????

EggHP

■ ■ ■

```

private void OnCollisionEnter(Collision other)
{
    // (空空)

    // 鸡蛋碰撞到障碍物时
    if (hp <= 0)
    {
        Destroy(gameObject);
        Destroyed?.Invoke(this);
    }
}
}

```

空空 GameManager 空空

```

// 鸡蛋
private void CreateEgg()
{
    // 鸡蛋
    var egg = Instantiate(eggPrefab);
    egg.transform.position = eggStart.position;

    // 鸡蛋
    EggHP eggHp = egg.GetComponent<EggHP>();
    eggHp.Destroyed += OnEggDestroyed;
}

// 鸡蛋
private void OnEggDestroyed(EggHP eggHp)
{
    // 鸡蛋
    eggHp.Destroyed -= OnEggDestroyed;
    CreateEgg();
}

// 鸡蛋
private void OnEggDetected(GameObject egg)
{
    // 鸡蛋
    EggHP eggHp = egg.GetComponent<EggHP>();
    eggHp.Destroyed -= OnEggDestroyed;
}

```

```
// 鸡蛋孵化时间
```

```
Destroy(egg);
```

```
CreateEgg();
```

```
}
```

Revision #4

Created 2026-07-23 07:59:56 UTC by Menendez Francisco

Updated 2026-07-24 04:35:29 UTC by Menendez Francisco