

Cha4-Unity3d 和 C#的双剑合璧 13

在上一课的内容中，我们的精灵公主通过”开天眼“的仙术提前侦查到了敌人。

而在这一课的内容中，我们将完成一个重要任务，那就是彻底消灭大魔王。

是的，这里说的不是意念攻击，或是在 Console 视图里面纸上谈兵，而是真正的从屏幕上将大魔王销毁~

好了，让我们开始这一课的学习吧~

个人微信号：iseedo

微信公众号：vrlife

13Unity 和 C#的双剑合璧-彻底消灭大魔王

在这一课的内容中，我们将使用 instantiation (初始化) 和 destruction (销毁) 方法来生成和摧毁大魔王~

什么是 instantiation 方法和 destruction 方法呢？

举个简单的例子，在 FPS 游戏中，我们希望从热兵器发射子弹。在开火之前，这些子弹是没办法手动添加到场景中的。随着玩家触发了开火的事件，比如通过交互事件扣动扳机后（按下键盘上的某个键，点击鼠标，触碰屏幕，或是使用游戏手柄上的操作按钮），代码将在程序中自动生成所需要的子弹。

当然，当子弹击中目标物体后，或者飞出天际之后，如果我们不销毁子弹游戏对象，那么在场景中就会出现大量的冗余无用的垃圾。这些垃圾游戏对象会极大影响游戏的运行性能，而且会有一些潜在的不良后果，无法预测~

instantiation 的另一个例子是，在有些网络游戏中，在特定的区域内会随着时间的推移“刷怪”。这些 NPC 怪物的数量、等级和位置都是相对比较随机的，游戏开发者更不可能提前就把这些怪物放置在游戏场景之中。这个时候，我们就可以用到 instantiation 方法在场景的特定位置随机“刷怪”了~

好了，现在大家已经知道为什么我们要用 instantiation 和 destruction 方法了。

接下来一起看看，在代码中究竟应该怎么用这两种方法来实现游戏开发的目标~

首先我们将使用 instantiation 方法在场景中生成宝箱。

在 Unity 中打开项目，在 Project 视图中的 _Scripts 文件夹中找到 TreasureChest.cs，双击在 Visual Studio 中将其打开。

为了给游戏增加点乐趣，我们可以在宝箱的位置同时生成金币（可惜不是 BTC~）。为此，需要创建一个变量来保存金币的预设体，同时还需要创建一个变量保存金币的生成位置。

为此，在 Start()方法之前添加以下变量定义的代码：

```
//5.创建变量保存金币预设体
```

```
public Rigidbody coinPrefab;
```

```
//6.创建变量保存金币所生成的位置
```

```
public Transform spawner;
```

在上面的代码中，我们首先创建了一个 Rigidbody 类型的变量，用来保存金币的预设体。此外，我们还创建了一个 Transform 类型的变量，用来保存金币所生成的位置。

接下来需要更改 Update 方法中的代码：

```
// Update is called once per frame
```

```
void Update()
```

```
{
```

```
    //3.当 interactable 是 true,而且玩家按下了空格键时
```

```
    if(interactable && Input.GetKeyDown(KeyCode.Space))
```

```
    {
```

```
        //4.将动画控制器中的 openChest 参数设置为 true
```

```

anim.SetBool("openChest",true);

//5.定义一个 Rigidbody 类型的临时变量，用来保存使用 Instantiate 方法生成的金币
Rigidbody coinInstance;
//6.使用 Instantiate 方法生成金币，
coinInstance = Instantiate(coinPrefab, spawner.position, spawner.rotation) as Rigidbody;
//7.给新生成的金币添加一个向上的力
coinInstance.AddForce(spawner.up * 100);
}

}

```

在以上代码中，新增加的是注释编号为 5，6，7 的代码，这里简单解释一下。

5..定义一个 Rigidbody 类型的临时变量，用来保存使用 Instantiate 方法生成的金币。之所以定义为 Rigidbody 类型，是因为我们希望所生成的金币对象一开始就有刚体组件，不然就会直接掉到宝箱下面不见了~

6.这里使用了 Instantiate 方法来生成金币游戏对象。

关于 Instantiate 方法的详细知识，可以参考这里：

<https://docs.unity3d.com/ScriptReference/Object.Instantiate.html>

<https://docs.unity3d.com/Manual/InstantiatingPrefabs.html>

可以看到，Instantiate 方法的基本用法如下：

```
public static Object Instantiate(Object original, Vector3 position, Quaternion rotation, Transform parent);
```

首先，这是一个 static 静态方法，所以不需要某个具体的游戏对象来调用。

其次，它一共可以设置 4 个参数，分别是游戏对象的预设体，游戏对象的生成位置，游戏对象的旋转，以及父对象的 Transform 信息。

这里我们只使用了其中的前三个参数。

最后，我们使用 as rigidbody 来结束了这行代码，这个看起来似乎有点奇怪~

对于新手来说，可能有点不好理解，我们需要的是金币的整个游戏对象，为什么这里只获取它的 rigidbody 组件呢？

其实不然，在使用 Instantiate 方法时，不管是否添加最后的 as rigidbody，我们都获取了整个的游戏对象，而不仅仅是它的 rigidbody 组件。

之所以这样来写，是因为我们立马就要使用 rigidbody 的方法来完成一个任务，那就是让金币向天空飞舞~

关于这个问题的详细解释可以参考这里：

<https://answers.unity.com/questions/1482646/instantiating-a-rigidbody-instead-of-a-gameobject.html>

好了，接下来让我们返回 Unity 编辑器。

在 Hierarchy 视图选择 Chest 游戏对象，可以看到在 Treasure Chest 组件中多了两个属性设置，分别是 Coin Prefab 和 Spawner。

在 Project 视图找到 Assets/Resources/Prefabs 目录，然后找到 Coin 预设体，把它拖动到 Treasure Chest 组件的 Coin Prefabs 属性处。

之所以可以直接这样设置，是因为 Coin 预设体中已经添加了 rigidbody 组件~

接下来要设置的是 Spawner 属性。

在 Hierarchy 视图选择 Chest 游戏对象，然后点击 Create-Create Empty Child，从而创建一个子对象，将其更名为 Spawner，然后使用移动工具把它的位置拖动到宝箱的顶部。

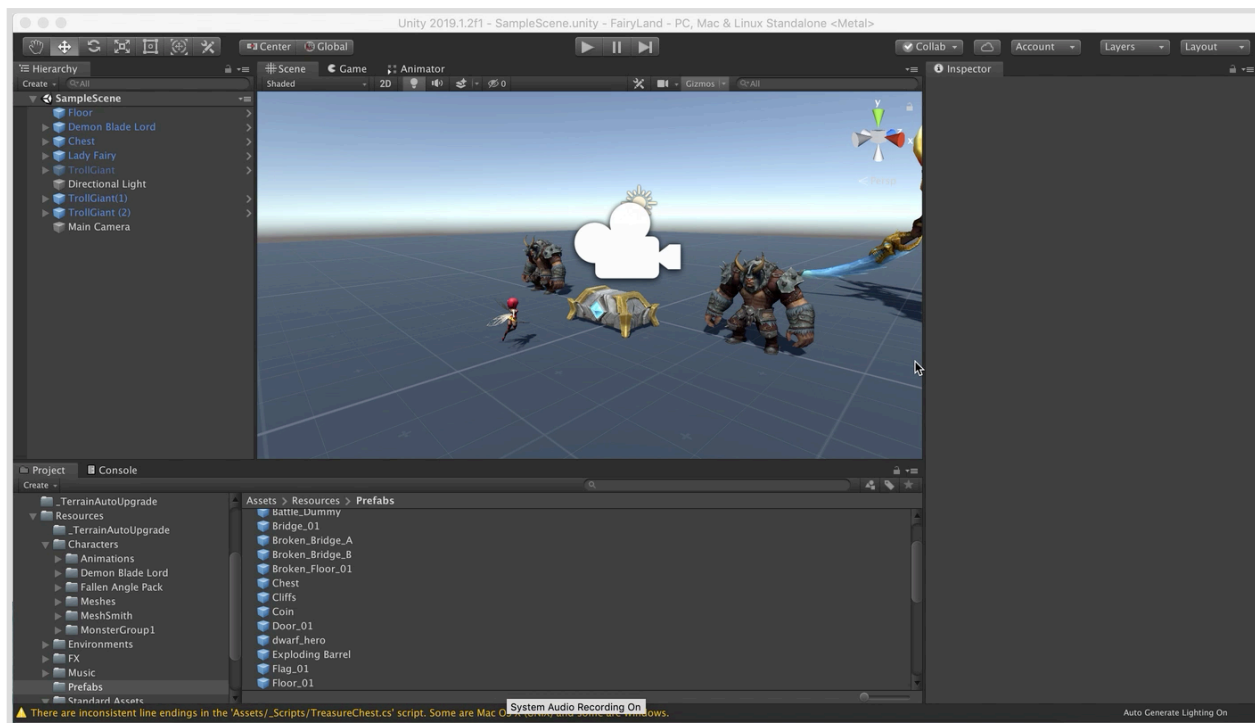
这样，所生成的金币就不至于被镇压五百年出不来了~

当我们按下空格键时，金币飞出一时爽，一直飞出一时爽啊~

然鹅，这些金币看起来可以一直堆到月球了。有没有什么办法让它们自动消失呢？答案是肯定的。

接下来我们需要创建一个新的脚本来完成这项任务。

在 project 视图中的 _Scripts 目录下新建一个脚本，命名为 Coin，双击在 Visual Studio 中打开。



在 Start()方法中添加一行代码：

//1.在两秒钟后自动销毁

```
Destroy(gameObject, 2.0f);
```

这里用到了 Destroy 这个系统方法，它的详细使用可以参考：

<https://docs.unity3d.com/ScriptReference/Object.Destroy.html>

Destroy 方法的基本使用如下：

```
public static void Destroy(Object obj, float t = 0.0f);
```

它是一个静态方法，因此无需使用实例化的对象来调用。

另外，它可以传递两个参数，一个是要销毁的游戏对象，而另一个则是等待时间。

所以，刚才这行代码的作用就一目了然了，在两秒钟之后将自己销毁~

返回 Unity 编辑器，在 Project 视图找到 Assets/Resources/Prefabs 目录，然后找到 Coin 预设体。

在 Inspector 视图中点击 Open Prefab，然后将 Coin.cs 这个脚本文件添加为 Coin 预设体的组件。

好了，一切就绪，再尝试一次吧~

