Unity2D Learning

# 目录管理

project的assets下面通过目录来管理。Animator Materials Prefabs Scenes Scripts Sounds Textures。 其中 如果已经设计好的组件 放到Prefabs中。

# GameObject管理

GameObject中是实现了一个非常重要的组织形式就是 组合。这些是真正在游戏中动作的元素。所以 这里的层级关系就是组合模式。

GameObject如果有sprite，那么就是用来显示的。而如果GameObject是空的，就可以用作组合形式的parent.

调整z坐标来改变上下级的关系。

# 添加Sprite

Sprite是用来显示的。所以直接创建了一个sprite之后，直接将纹理拖入sprite的显示即可。

# 拆分一张图片上的多个Sprite

拆分一张图片上的多个精灵 使用到了TextureEditor.

# 添加碰撞盒

添加这个用来碰撞检测。

# 刚体

刚体使得碰撞发生刚体效果。这个如果在飞机游戏中可能不需要。主要设定重力加速度。

# *Sync MonoDevelop Project*

同步工程保持脚本和工程一致性。

# Script

## 生命周期

Awake: 仅仅被调用一次当组件被创建的时候。可以看作是对构造函数的替换。

Start:在Awake之后执行。不同之处在于如果脚本没有被enable，那么将不会调用。但是Awake会调用。

Update:每一帧都会被执行。

FixedUpdate:是一个固定的帧率执行。你应该使用这个函数来处理一些刚体的物理效果，而不是在update中处理。

Destroy:当对象被销毁。可以清理相应的资源。可以使用Destroy(gameobject, time)来防止子弹等对象的无限制产生。不过如果设置成离开视线才Destroy不是更好？？？

## 碰撞

OnCollisionEnter2D(CollisionInfo2D info)：当另外一个碰撞者与这个碰撞者相接触

OnCollisionExit2D(CollisionInfo2D info)： 当碰撞退出

OnTriggerEnter2D(Collider2D otherCollider)：当另外一个标记了”Trigger”的碰撞者接触了这个碰撞体

OnTriggerEnter2D(Collider2D otherCollider)：当另外一个标记了”Trigger”的碰撞者离开了这个碰撞体。

注意2D这个后缀不要用。

如果将 isTrigger 这个属性勾选，那么就不会产生物理上的碰撞模拟，而只是产生碰撞的事件。所以在飞行射击游戏中是非常适合将这个勾选的。

## 移动

PlayerScript 使用了rigidbody2D.velocity = movement; 移动player 应该还有其他的方案来移动一个非rigdbody2D物体。

移动是在 FixedUpdate中运行的。

void FixedUpdate()

{

// 5 - Move the game object

rigidbody2D.velocity = movement;

}

执行输入是 在update中运行的 使用的是Input.GetAxis.如何支持鼠标呢？

void Update()

{

// 3 - Retrieve axis information

float inputX = Input.GetAxis("Horizontal");

float inputY = Input.GetAxis("Vertical");

// 4 - Movement per direction

movement = new Vector2(

speed.x \* inputX,

speed.y \* inputY);

}

## 创建新对象

var projectile = Instantiate(projectilePrefab) as Transform; 使用这个语句通过设置的对象来产生新对象。

projectile.position = transform.position; 这条语句设置了炮弹的位置是从飞行器的位置发射的。问题是：如何设置从头部发出，尾部发出，从两侧发出呢？

# 视差滚屏移动

有两种方案，

方案一：Player和照相机移动，其他的静止

方案二：Player和照相机静止，其他的运动。

要实现一个好的视差需要方案一和二的混合解决方案。那么，我们可以将unity编辑器当作关卡编辑器。这是一个好的方案，除非你有时间和金钱开发一个更好的关卡编辑器。

关于敌人的生成，需要一种触发方案，在触发他们之前，他们是静态的，死的。

所以创建一个Level用来管理关卡。里面包含background以及敌人等。

照相机的滚动只需要调整他的transform即可。

那么背景的循环呢？

## 循环背景

具体算法是这样的，至少有两张背景（每一张都是一个独立的背景，两张每一张都能独立的覆盖整个照相机区域）。算法步骤如下：



# 照相机的控制区域

照相机的控制区域可以知道哪些物体是显示还是没有显示。这非常重要的函数。

下面的一段脚本代码就是来做这个处理的。是对render的一个扩展。是C#的一种语法直接扩展现有的类库。

using UnityEngine;

public static class RendererExtensions

{

public static bool IsVisibleFrom(this Renderer renderer, Camera camera)

{

Plane[] planes = GeometryUtility.CalculateFrustumPlanes(camera);

return GeometryUtility.TestPlanesAABB(planes, renderer.bounds);

}

}

# 疑问

## 如何设置子弹在离开视线返回后destroy这个子弹

在教程中是用Destroy(gameobject, time) 来消除子弹的。

## 如何设置子弹发射出去的位置从头部发出，尾部发出，从两侧发出呢

方案一：在飞机上增加发射点。那么如何计算子弹的区域或者飞机的区域呢 lastChild.renderer.bounds.max

lastChild.renderer.bounds.min

用这样的函数就获得了真正的区域。