# JumpTogether程序员开发手册

## 变量命名规则

遵从原则：大体遵从驼峰式（除了第一个单词外的其他单词的开头字母大小，比如testCounter）

### 组件

private组件：m\_camel规则，如

private Rigidbody m\_rigidBody;

public组件：camel规则，如：

public Rigidbody rigidBody

注意：组件明明不允许中间有其他\_

×：private Rigidbody m\_rigidBody\_Do;

√：private Rigidbody m\_RigidBodyDo;

### 属性

private属性,应在前面加上\_，表示私有制：

private int \_num;

public 属性，首字母大写，表示光明正大：

public string Name;

static修饰：以s为开头，如：

static bool s\_flag;

如果定义较长，应该以\_为分隔符，如：

private float \_state\_time;

public float State\_time;

### 函数

普通函数和静态函数一律遵从和Unity类似的大写规则，如：

Update()

OnEnable()

OnInspectorGUI()

函数普通参数应全部小写，且变量名尽量保持一致，如：

OnJump(float elapse)

### 局部变量

如果是局部变量，全部小写字母，加\_，如：

var random\_scale;

但局部组件如下：

GameObject objectG;

### 特殊文件

如GameManager，AdaptorManager，带单例的等非常重要的文件，可属性也驼峰模式。

## 程序书写规范

遵从原则：简单清晰明了，利用region等指令，规范“代码块”

### region

将一些有关联的代码组织在一起，然后折叠起来，方便程序员浏览，熟悉。如：

#region component

private *Rigidbody* m\_rigidBody;

//当前盒子

private *GameObject* m\_currentStage;

public *GameObject* Stage;

#endregion

#region propertity

private float \_state\_time;

public float Factor = 1;

public float Max\_distance = 5;

#endregion

## 开发问题&解决思路

### 跳动效果

如何让Player有跳动的效果？

：利用物理引擎，给Player添加Rigidbody组件，Player受到向下的力，当给它一个向上+向前的力时候，Player就会做抛物线运动，也就是跳动的效果。

### 滑步问题

Player是一个sphere+cylinder，所以在施加力让他跳动的时候，有时会在落地出现滑步？

：cylinder的碰撞盒也是这个形状，重心不稳，当换成box collider，并代码控制降低Player重心时，就会避免滑步现象。

//修改物理组件的重心到body底部

m\_rigidBody.*centerOfMass* = *Vector3*.*zero*;

### 相机跟随

如何让相机跟随Player?

：用DotWeen插件，记载最开始摄像机和Player之间的差距，每跳一次，就移动一次摄像机。

Camera.DOMove(*transform*.position + Camera\_relative\_position, 1);

### position和scale区别

比如说一个Plane，reset得到position是(0,0,0),Scale是(1,1,1)，那么scale是一个unity单位？position的单位又是啥？

：position 的单位是m，在3ds max中建模一米的物体，放到场景中是1

Scale默认是1，就是个缩放比例的factor，最终的大小 = 原大小 \* scale

他们的1都表示1个unit。

### 预告Player运动轨迹

为方便手残程序员debug跳动，给出随着按键时间，运动轨迹的预告，类似愤怒小鸟？

：

### 分数显示

TextMeshPro下的text，矢量图形，非像素级。使用GameManager来管理分数显示。

public *TextMeshProUGUI* ScoreText;

温馨提示：在2D下对UI进行调整很舒服。

### 蓄力效果

如何在用户按下键的时候，出现Player的缩放，盒子的缩放，给出蓄力的感觉？

：scale缩放，利用插件进行缩放，恢复原状。

Body.*transform*.DOScale(0.1f, 1);

Head.*transform*.*DOLocalMoveY*(0.29f, 0.5f);

m\_currentStage.*transform*.*DOLocalMoveY*(0.25f, 0.2f);

m\_currentStage.*transform*.DOScale(new *Vector3*(1, 0.5f, 1), 0.2f);

### 参数暴露

如何把一些必要的参数统一暴露出来，方便测试师(策划)进行游戏数值调整，比如说蓄力时每帧缩放的参数，跳远的距离等参数?

：使用[SerializeField]

### 落地判定

现在是拿物理做的gameover判定，但是ground碰撞盒大小有限。

至少你现在这个10 10的scale是肯定不够用的，要么随着游戏进度生成ground，要么扩大scale参数，要么使用别的判定方式，如y轴位置小于0。

:ground可以拿掉，因为Camera自带solid color，判定写位置就行。

### 对象池回收和销毁

对象池里面预热1个Stage，场景中初始化一个Stage，并拿出一个Stage，这时场景里有两个Stage。

当Player跳到下一个Stage时候，回收上一个Stage，并生成下一个Stage(刚回收就拿走)。

但是对象池回收无法确定顺序,总是拿最新的Stage回收。

：把回收代码写在Active的Stage上，计算时间10s，到时间就自动回收(不是理想的方法)。

每次游戏重新LoadScene的时候，发生了什么？对象池是重新建立instance还是如何，上次建立的那些Stage又如何处理了？为何感觉不间断LoadScene重玩，会越来越慢？

### UI适配

如何适配不同设备的分辨率？Sprite的图片png应该有多大可以cover不同的设备，不至于模糊？

：

<https://blog.csdn.net/midashao/article/details/8232341>

<https://blog.csdn.net/n5/article/details/50083205>

<https://www.jianshu.com/p/9dfa070006d9>

### 数值资源利用

如何让一些数值，可以序列化显示在inspector面板上，又可以被全局使用呢？比如计算出来的宽高比之类的一些有关适配的值。

：如果是单纯的配置文件，常用xml或者ini，而asset文件实际上更加灵活一些。当然这时先不考虑性能优化问题,ScriptObject和Seriazlied以及自定义Editor结合使用。

<https://blog.csdn.net/LIQIANGEASTSUN/article/details/50510615>

但是有一个问题就是，我们都知道在unity在运行状态时，修改面板上的参数，可以在游戏中立即生效看到效果。那么asset文件上参数，怎么做到修改立即生效呢？(生效不是指面板上的改动，而是在游戏中对游戏性的改动)

：？

### Camera.backgroundColor无透明度

尝试让背景色透明一点，但是毫无效果？明明有alpha值的。

## 插件

### DotWeen免费版

一款简单动画插件，可很较容易的实现各种渐变效果。

### TextMesh Pro

一款文字插件，支持更加好看的文字效果，方便开发。

## 开发小知识

### Time.deltaTime解释

<https://blog.csdn.net/chinarcsdn/article/details/82914420>

**transform.Translate(0, 0, Time.deltaTime \* 10) 也就是1s物体在z轴方向运动10m**

### StartCoroutine解释

协程，相当于是一种多线程。

<https://blog.csdn.net/qq_15020543/article/details/82701551>

### 自定义Inspector界面

对象池脚本，挂载到PoolManager物体上，自定义对象池脚本的Editor。

在Scripts文件夹下新建Editor文件夹，创建PoolManagerEditor:Editor。

但是似乎单例模式有点不同。

### 单例模式

可以不用实例化类对象，直接调用instance.xxx即可。如下：

单例模式之前:

PoolManager poolManager = new PoolManager();

Debug.log(poolManager.name);

单例模式之后:

Debug.log(PoolManager.instance.name);

### 函数运行顺序

[RuntimeInitializeOnLoadMethod(*RuntimeInitializeLoadType*.*BeforeSceneLoad*)]->Awake->OnEnable->Start

### MeshFilter组件

可以用来生成mesh，代码自由度较高。

### position和localposition的区别

position是世界坐标系中的坐标，localposition是相对于父的坐标，在inspector面板中的transform下，显示的是相对坐标。

当然一级物体，position和localposition是一样的。

### 读取asset file的方法

1. 通过AssetDatabase.LoadAssetAtPath加载
2. 通过Resources.Load加载

### [ExecuteInEditMode]

[ExecuteInEditMode]意思是，在EditMode下也执行脚本。unity默认是在运行时才执行，加上该属性，不运行也能执行脚本。

Update : 只有当场景中的某个物体发生变化时，才调用。

OnGUI : 当GameView接收到一个Event时才调用。

OnRenderObject 和其他的渲染回调函数 : SceneVidw或者GameView重绘时，调用。

Awake与Start调用规则：

Awake：加载时调用。

Start：第一次激活时调用。

### 脚本机制内部原理

U3D会用Mono将所有的脚本编译为.Net dll文件，然后在运行时对这些dll文件进行JIT编译(ios等特殊平台除外).JIT是Mono内部的编译器，看起来游戏程序员能了解到.Net dll文件这块就已经游刃有余了。

### png透明

## 性能优化

如何让游戏性能最优，如何设计游戏开发模式？

### 盒子销毁(对象池)

在合适的时间销毁跳过的盒子，防止占据内存？

这个肯定是要销毁的，你可以用一个FIFO队列来维护

但是！频繁的销毁生成对性能有很大的影响，最好还是维护一个对象池。

<https://zhuanlan.zhihu.com/p/72770529>

### 更换摄像机背景色造成卡机

想得分超过一定数值，就更新一下背景，但是直接更新camera.backgroundColor似乎会造成大卡机.

:因为把每次判断都放在Update里面了，造成一定的性能开销。如果放在每次socre++的后面就会好很多。

### 3D cube做盒子vs Sprite

感觉用3d cube做盒子，似乎总是很慢，没有微信小程序那种程序很轻盈的感觉，是否可以全部改成2D，用Sprite做盒子，上面加载碰撞盒？

## 专有名词

|  |  |
| --- | --- |
| Stage：盒子（为啥不是box啦） | GameObject go：常用于函数参数 |
| Spawn：生成 |  |
|  |  |

# JumpTogether策划&测试手册

## Game idea

### 场景

想要盒子悬浮在空中的效果，有些盒子还会来回漂浮，有的上下漂浮，有些还有震动效果。

平衡板：需要两个小人同时跳到平衡板的两端，单独跳就会失去平衡掉下去。

## 游戏参数平衡

### 摄像机位置

摄像机的位置应该摆在哪里，才能像微信游戏那样，视角体验那么好？好难调整。--cover，理解了观看的角度，就不是很难调。重点是动脑子，不要用脚思考。

## 游戏bug测试

### 小人和盒子缩放变形

### 边缘跳跃下次会跳空