# 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;

## 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

### 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。

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。

## 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**

## 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

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。

## 专有名词

|  |  |
| --- | --- |
| Stage：盒子（为啥不是box啦） |  |
| Spawn：生成 |  |
|  |  |

# JumpTogether策划&测试手册

## 玩法技巧idea

## 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。

## 游戏bug单元测试