一、UI

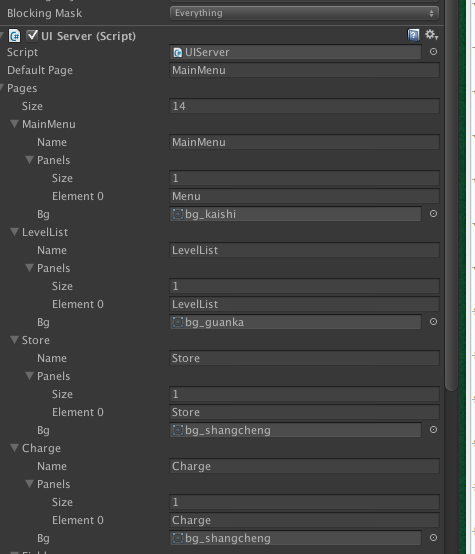
大多数控件直接制作放于场景上，除了常刷出的物体（位于Prefabs文件夹下，有商城充值交易物品，6中物体，各色炸弹，爆炸效果，槽位，关卡按钮），这些预制物体通过ContentAssistant刷出来。

重要脚本：

1、UIServer

该脚本挂在UI物体上，控制场景内各个模块现实和隐藏，回退。

添加新模块，先在场景中UI根目录下制作一个UI模块，然后在UI游戏物体上的UIServer的Pages添加一项Page（name，Panels，bg）name是你为这个模块取得名字，Panels是这个模块在场景中的name，bg是这个模块的背景。需要显示是直接调用showPage(name)即可。

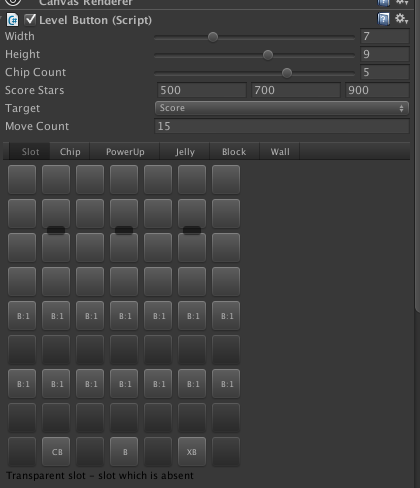


2、CPanel

各个模块父级物体上所需绑定脚本（即UI下的物体，如，Main,LevelList,Field），其控制各个模块展示和隐藏的动画。

3、LevelButton,LevelButtonLocker

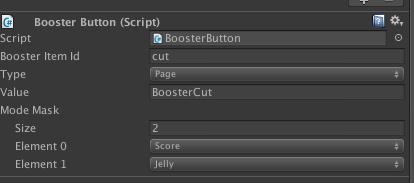
LevelButton是关卡按钮的脚本，决定了该关卡游戏的玩法（直接在inspector面板中设置）,具体可设置长宽（height，width）（即槽位长宽数量），游戏内宠物的种类（chip，count），三个目标得分（score star），游戏模式（Tagrget）,移动限制（move count），时间限制（time durcation），这两个限制跟游戏模式相关。最后是游戏地图的编辑（可分别设置槽位，宠物，炸弹，果冻，砖块，墙）。这个可通过修改LevelButtonEditor来重新编辑。



LevelButtonLocker是用于控制关卡解锁，每大关第一小关默认开启，每通过一关，后面一关则开启。用PlayerPrefs来存储关卡（Complete\_关卡名）是否解锁（1为解锁，0为未解锁），并用(Best\_Score\_关卡名)记录通关得分。

4、BooserList，BoosterButton

BoosterButton游戏内使用的道具，itemId：道具ID，mode mask 决定 的关口类型（score，timer，jelly），这些id一定要喝Core物体上的BerryStore所填写的物品ID一一对应。



5、GamePauseObject

放在模块物体下，其显示隐藏可控制游戏内的暂停

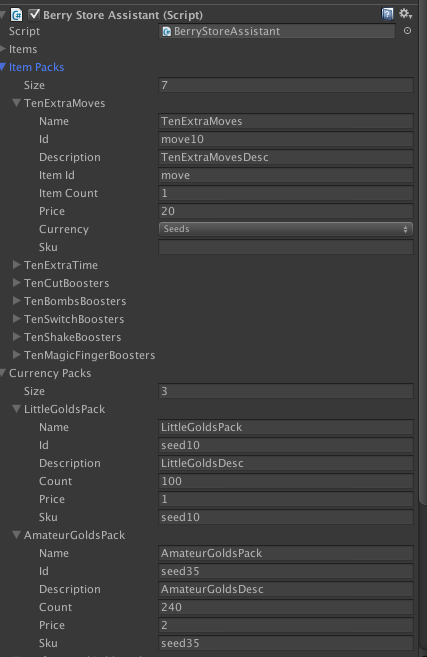
6、FieldCamera

通过摄像机来实现游戏内关卡切换的插值动画，可以通过调整defaultFOV变量，来改变效果

二、商店

1、BerryStoreAssistant

可在面板上直接编辑各个充值项，道具项，等等。尤其注意ID，是在游戏内识别的重要标志。并定义了充值购买过程中各个回调方法（，购买，游戏币数量变化，道具数量不够，道具数量变化，支付成功回调，购买道具成功回调）。



2、StoreInventory, StoreInfo

充值购买流程的核心类，其分别在Android和IOS两平台中分别重写了一次，待察。

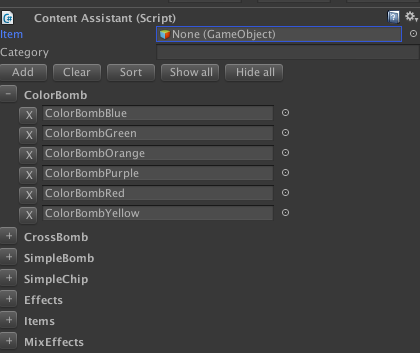
3、KeyValueStorage

游戏的存储逻辑，以键值对方式存储，提供了类似IDictionary的方法，主要是GetValue（）和SetValue（）方法，并分别实现Android和IOS平台上的接口

三、游戏内

1、ContentAssistant

预制物体生成器,但是预设得先在inspacter面板上注册，通过GetItem（name）获得预设实例对象。



2、FieldAssistant

游戏场景中的物体生成器，按照LevelButton所配置的样子来生成游戏内各个物体的位置（槽位，宠物，炸弹，砖块，果冻，隔墙），并提供获取页面上各位置物体的方法，增加炸弹，消除果冻和砖块的方法。

3、SessionAssistant

游戏内各个事物事件管理器（核心）。处理游戏的进程，提示，失败，成功等各个游戏事件。部分重要的注释在代码上，其他部分尚未具体分析。

StartSession():用于开始监听三种模式下（得分，时间，破冰）游戏的状态，通关与否以此为准。具体三个方法为ScoreSession(),TimerSession(),JellySession

FindingSolutionsRoutine(): 每隔一段时间提示可消除的物体，若没有，则刷新部分页面物体。

BurnLastMovesToPowerups()：将剩余步数转换为炸弹。

4、ControlAssistant

游戏内滑动算法控制器具体分为桌面和移动设备

三、总结

项目中所有的类都使用单例来访问对象。没有一个主轴，各个模块杜克随时切入。凡是游戏重要的类度放在了Core物体上。该游戏的UI部分全部已做好，然后再加入UIService的Pages初始化，通过UIService来展示和隐藏，而页面的刷新工作直接交给OnEnable，OnDisable方法。其他游戏内常用的一些物体则通过ContentAssistant来实例化。

除部分通关详情直接存储在PlayerPrefs中，其余道具游戏币的核心数据则存储于KeyValueStorage中，而在android和ios中则分别重写了这些方法，通过Android和IOS本地化实现存储。在支付流程中，作者写的较为复杂，我并没有细细去研究，不过已经接入了Google Play和苹果支付。而这两个支付也都是通过Android和IOS在本平台实现的，并且数据也在各自平台更新。所以在原版本上修改比较麻烦。不过可以添加商品，直接在Core物体上的BerryStoryAssistant各个变量中添加，（设置ID，价格，数量，简介等等）尤其注意ID，这个Sore和Charge模块下各个item获取信息的依据。

关于游戏玩法，游戏的编辑可在LevelButton的Inspector面板中设置（包括胜利条件，形状，物体位置），其中槽位及槽位中的物体可通过单击小面片设置，若需额外修改可以在SessionAssistant中修改代码，目前尚不清楚。