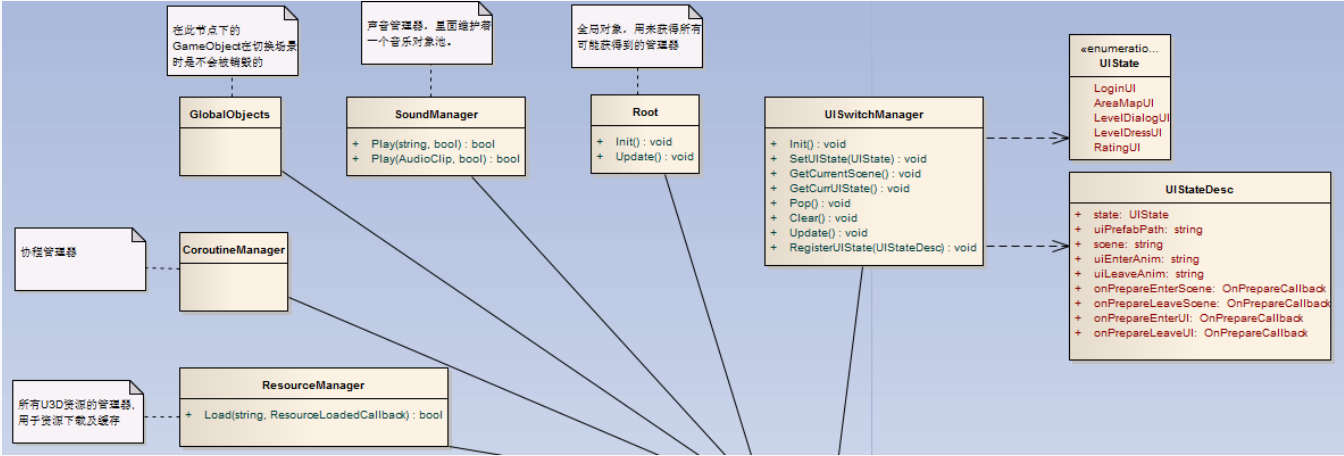


# 《暖暖》第二阶段Demo框架设计

## 交互模块：

### 通用全局对象：

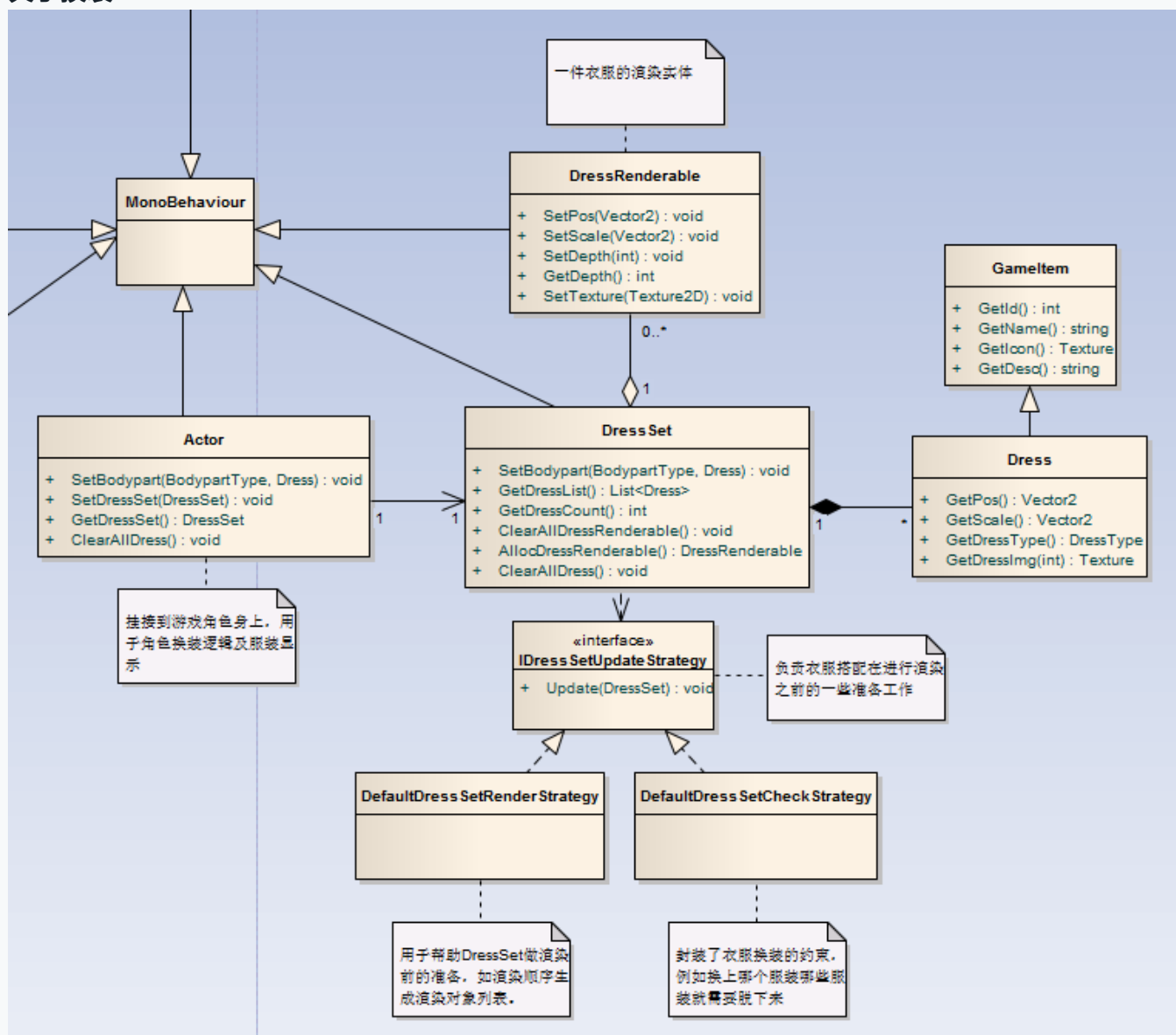


- **Root:**用于获取各种管理器。
- **CoroutineManager:**协程管理器，目前是将U3D协程申请都统一到此对象上来，便于后续扩展。
- **ResourceManager:**资源管理器，用于资源的下载及缓存。
- **UISwitchManager:**界面切换管理器。
- **SoundManager:**声音管理器，管理着一个声音对象池。

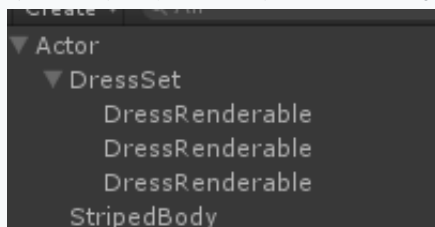
### 常驻UI页面

- **NoticeBox:**用于提示奖励信息，及与游戏内容相关的弹窗提示信息。
- **MessageBox:**用于提示与游戏内容无关的弹窗提示信息。
- **LoadingUI:**用于在转场时加载资源时弹出的等待界面。
- **BusyUI:**在一个场景内执行耗时操作的提示UI。
- **LockerUI:**用于将当前正在显示的页面锁定，禁止响应点击。（例如，人物在地图走动时界面上的所有元素均不可点击）

## 关于换装：

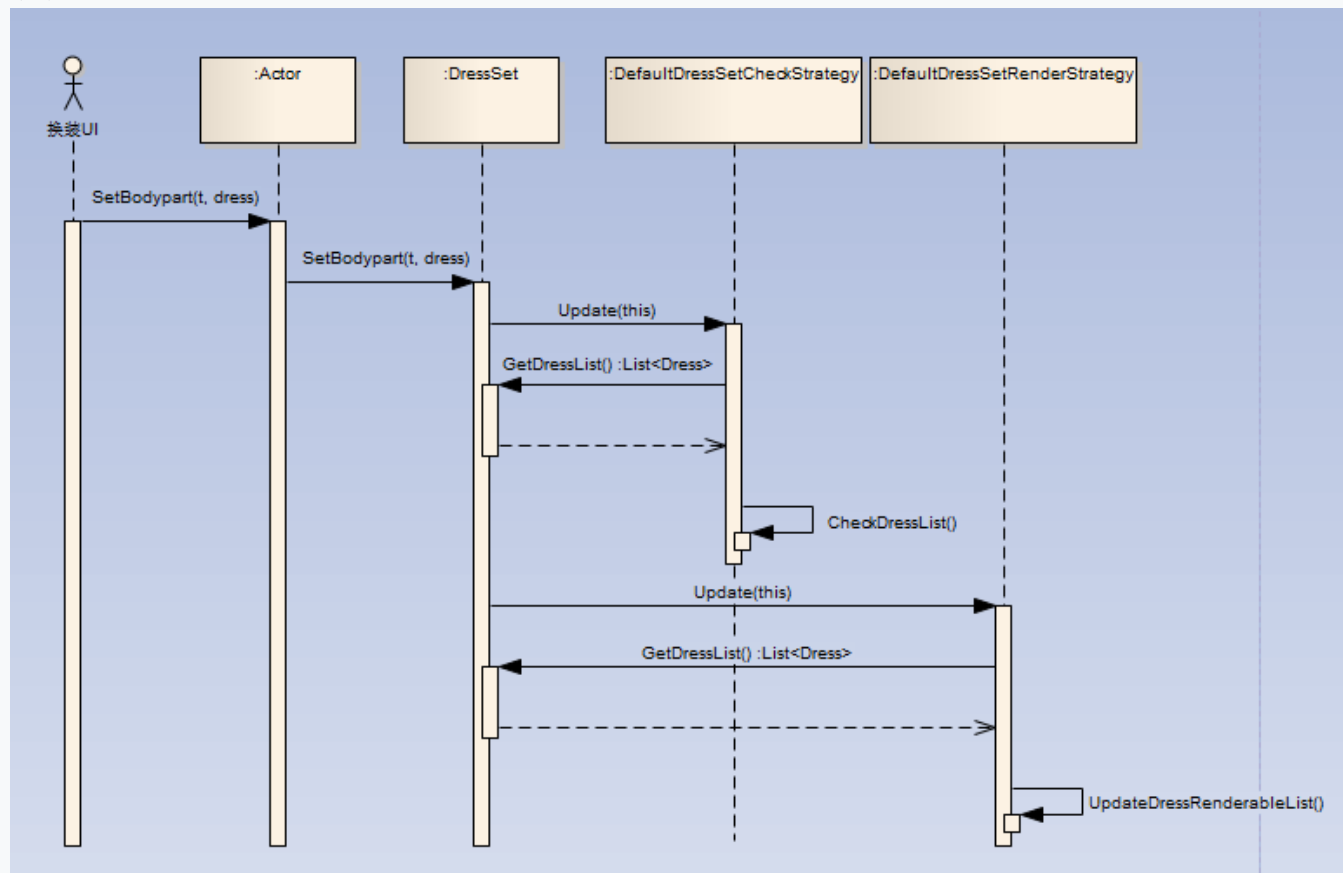


- **Actor**: 此脚本挂接在人物对象根节点，外部直接操作此对象换装。换装外界直接调用 `Actor.SetBodypart`。
- **DressSet**: 表示一个换装搭配方案，`DressSet` 管理当前衣服着装列表。并管理着一个 `DressRenderable` 池，每个 `DressRenderable` 是挂接在一个 `GameObject` 上用于实际显示服装，`DressRenderable` 可根据需要动态生成。`DressRenderable` 的渲染顺序由 `DefaultDressSetRenderStrategy` 决定，通过设置 `DressRenderable` 的 `Depth` 来达成。而衣服之间的互斥关系由 `DefaultDressSetCheckStrategy` 来修正。以下是 `Actor` 及 `DressSet` 中场景节点示意图。

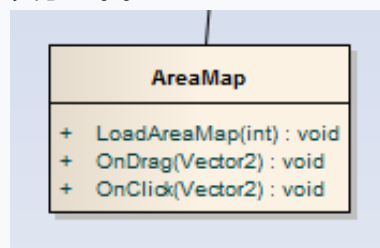


- **DressRenderable**: 衣服渲染对象，此脚本是一个 `Texture` 控件。
- **GameItem**: 游戏物品
- **Dress**: 游戏中的衣服物品

## 换装逻辑:



## 关于地图：



- **AreaMap**:此类管理地图显示及操作响应。

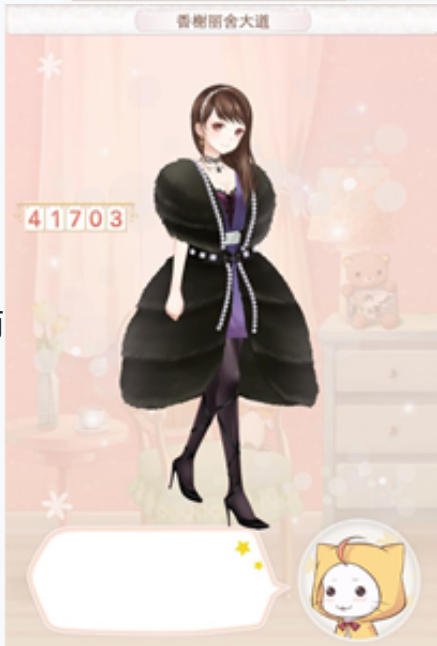
**UI界面定义：**在此游戏中独占手机屏幕的所有UI元素集合叫一个“UI界面”。一个场景可以有多个“UI界面”。在一个场景中某一时刻只有一个“UI界面”处于显示状态。

例如：



是CoreGameScene的一个“UI界面”，

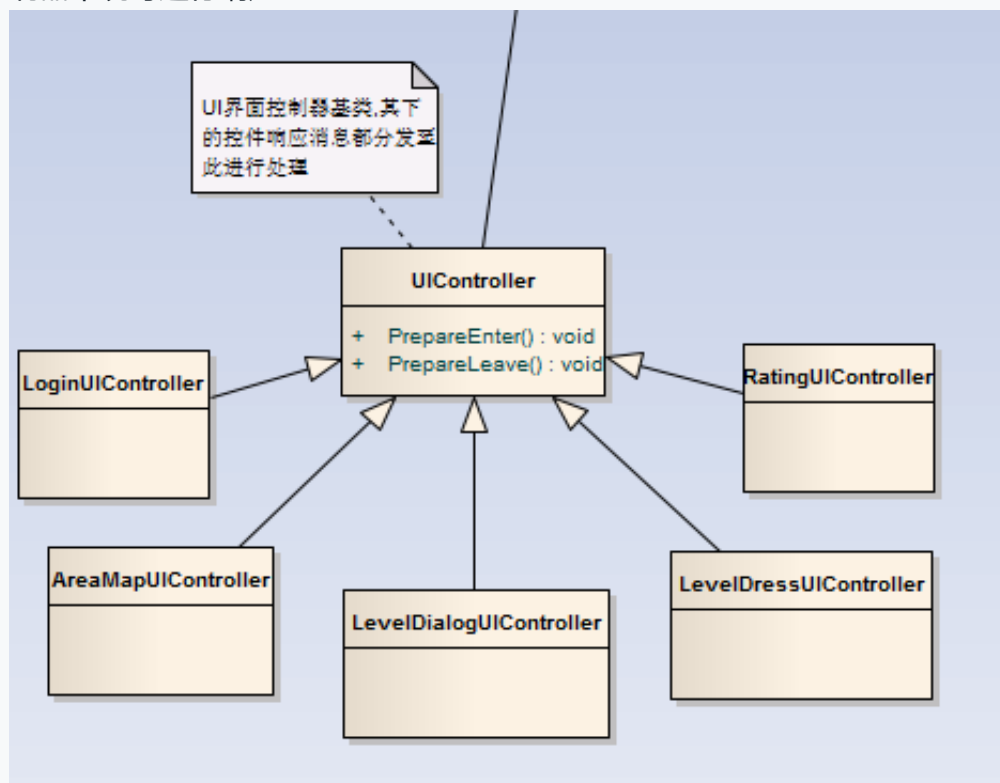
而



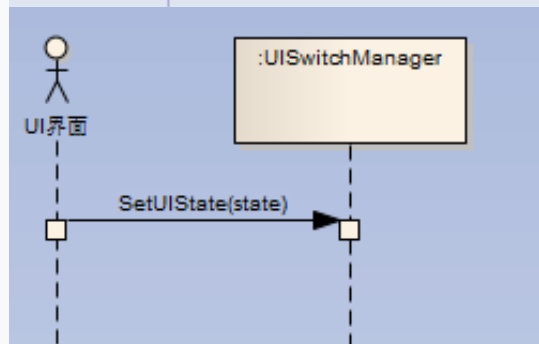
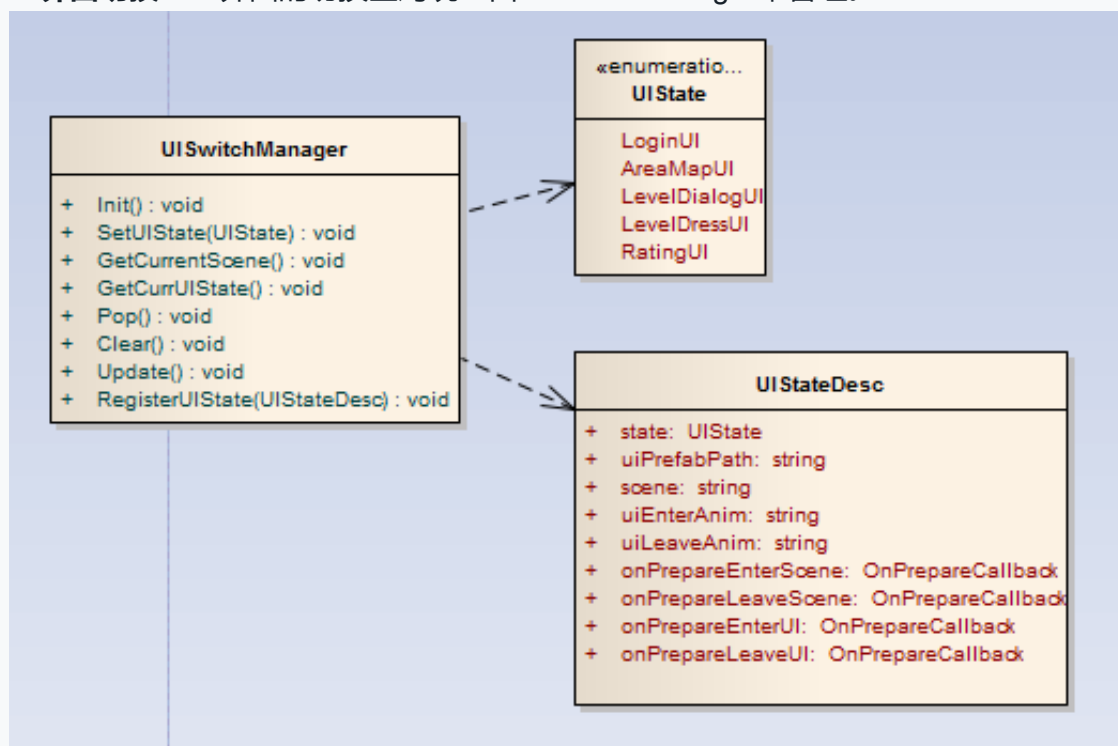
是CoreGame Scene的另一个“UI界面”。

**UI界面元素定义：**处于“UI界面”中的所有控件叫作“UI界面元素”。“UI界面元素”由“UI界面”控制显隐及行为。

**UI界面控制器**：用于负责一个界面的控件交互逻辑，在此界面下的所有控件的交互消息都转发至此控制器中统筹进行响应。



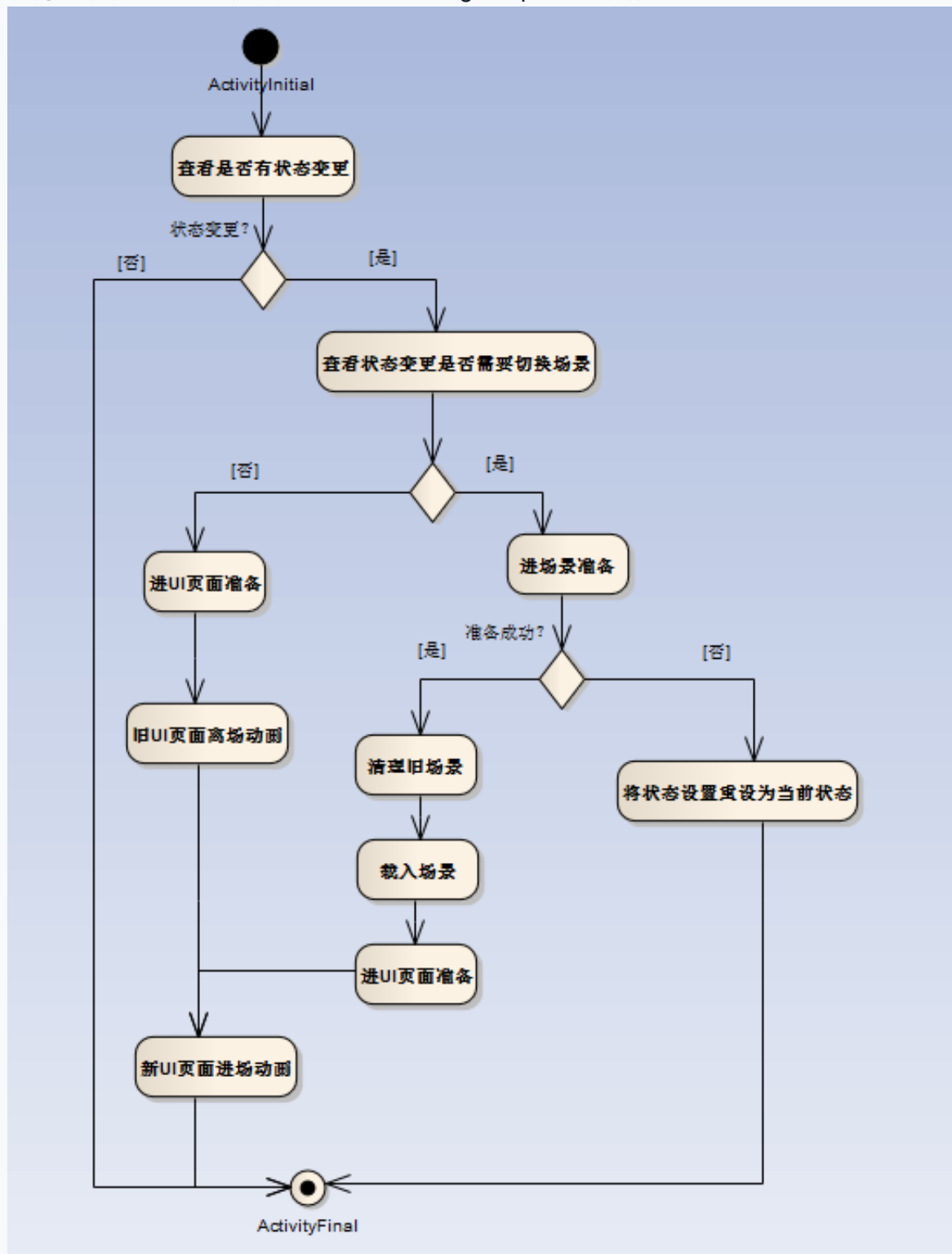
**UI界面切换**：UI界面的切换全局统一由UISwitchManager来管理。



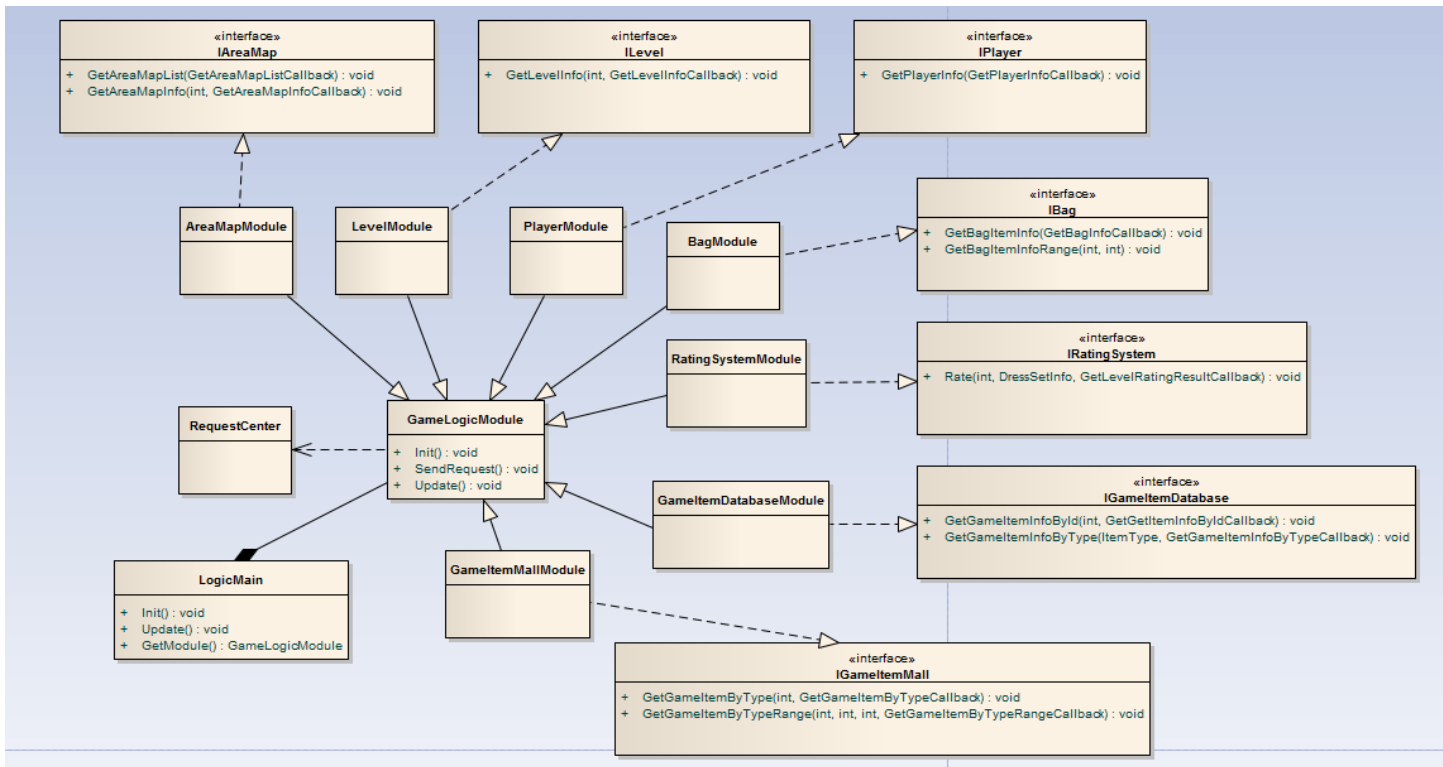
使用者只需要简单SetUIState就可以跳转界面，而不用管此界面属于哪个Scene。

- **UISwitchManager**:UI界面切换管理器，每个UI界面需要先在此管理器中注册界面描述结构体（UIStateDesc）。UISwitchManager负责界面切换状态记录，以便界面返回后切换至之前状态。
- **UIState**:UI状态枚举，每个UI界面一个状态。
- **UIStateDesc**:UI界面描述信息，里面记录了实例化此UI界面的Prefab地址、所属场景、进出场动画、进出场景准备动作回调、进出UI界面准备动作回调。

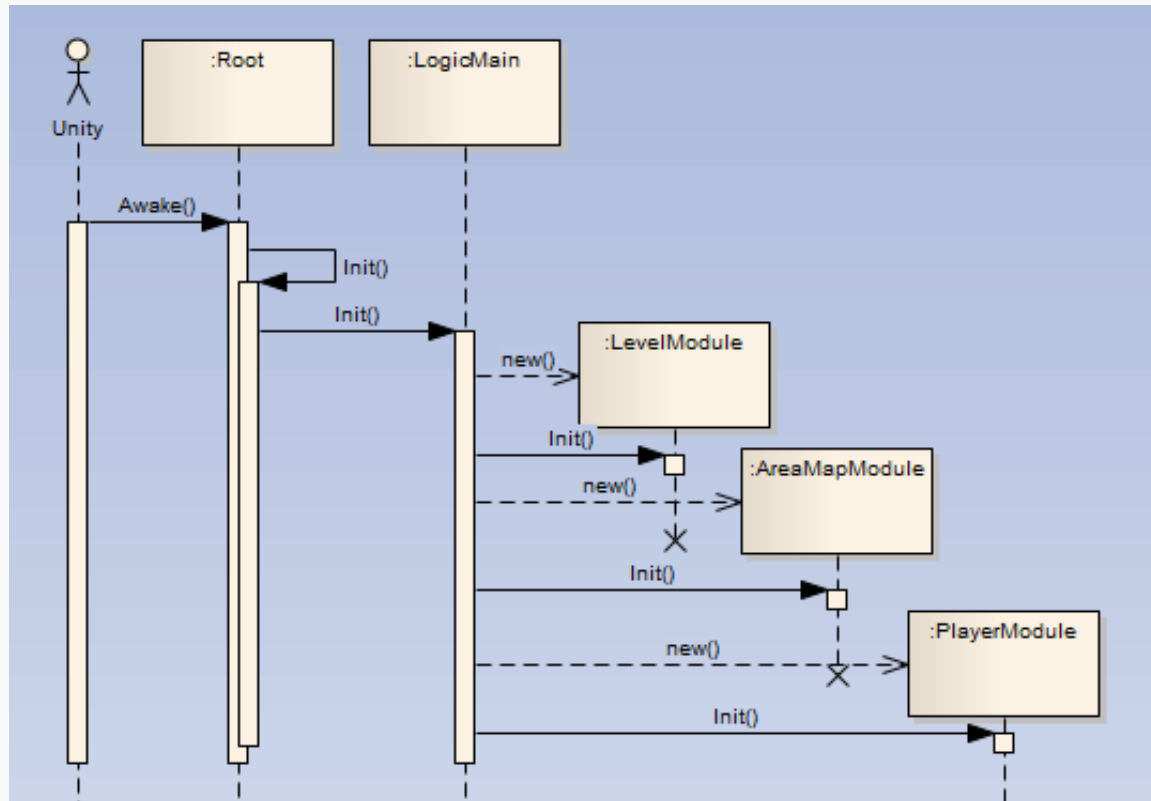
**UI界面切换逻辑**：每次调用UISwitchManager.Update时触发



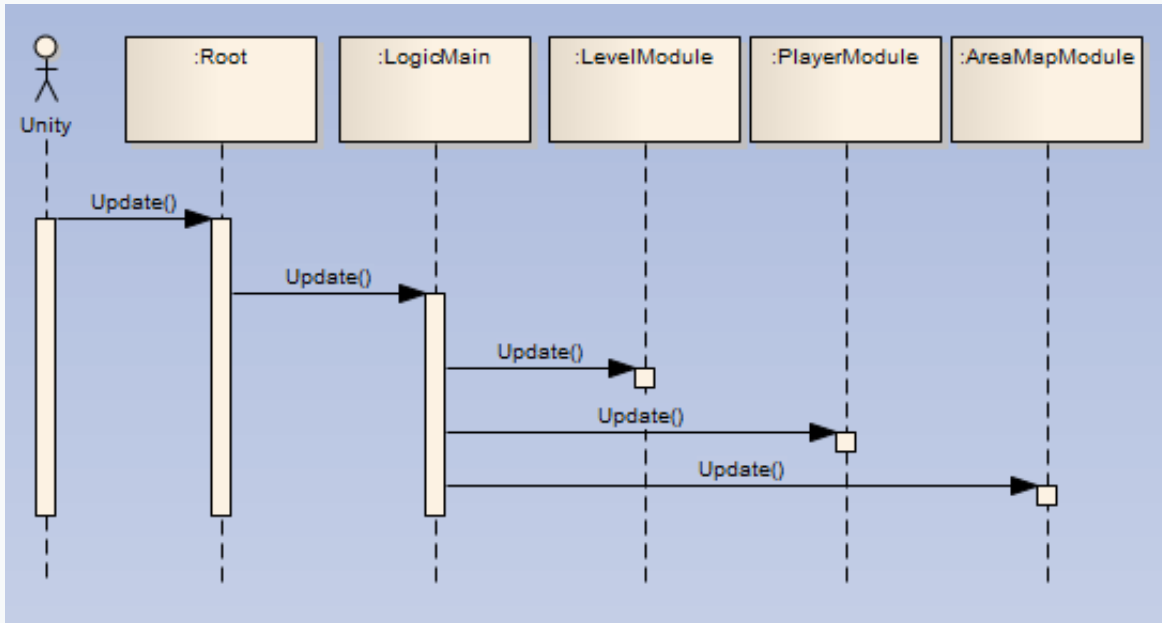
**游戏逻辑模块：**



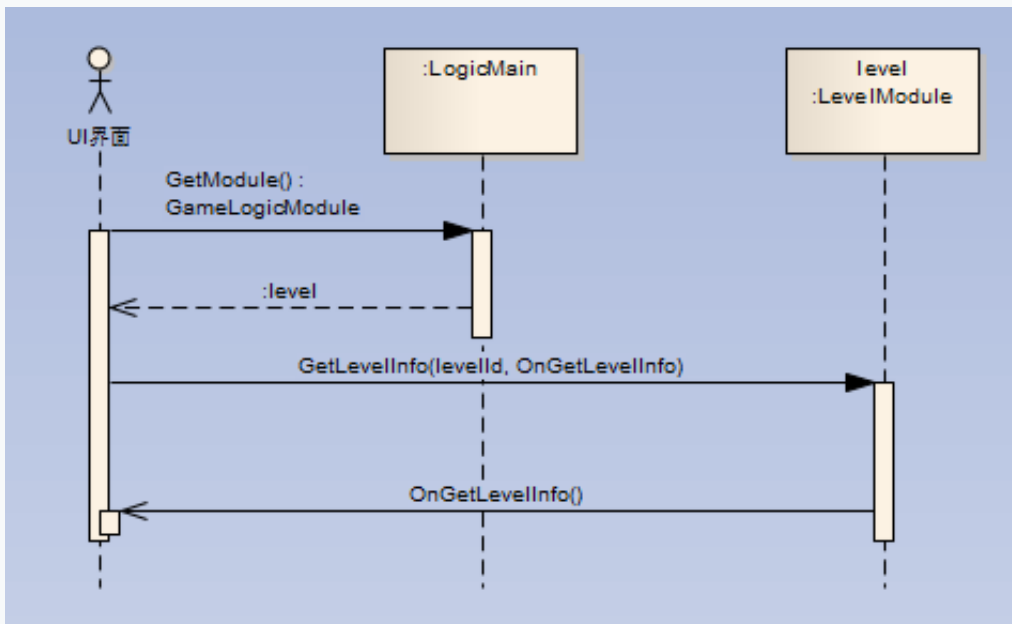
游戏逻辑模块初始化：在Root的GameObject创建时，进行初始化。



**游戏逻辑模块更新：**逻辑模块会在Root的带动下每帧进行更新。每帧的更新操作主要是用来处理从逻辑模块从网络得来的消息数据，并解析数据最终派发给目标回调函数。



**对上层UI接口：**对于上层的UI界面来说，只关心数据的显示，上层UI界面会通过以下接口获取自身感兴趣的数据。每个接口都可以获取游戏中的一类信息。UI界面与游戏逻辑模块交互是异步的，调用函数逻辑接口后并不立即返回结果，而是由UI层传递给游戏逻辑层相应回调函数，当游戏逻辑模块得出结果会调用之前UI层传递的回调来通知UI层。



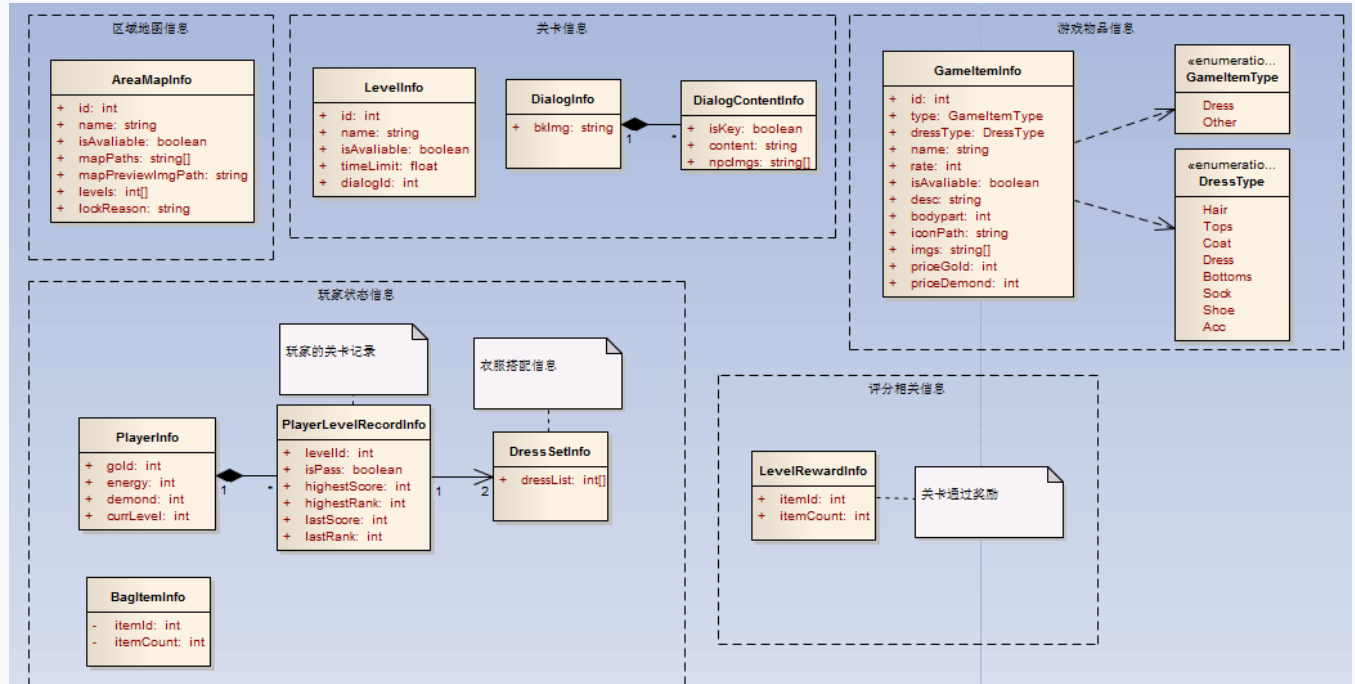
- **LogicMain:**对游戏逻辑模块外用来获得各逻辑模块的“外观”。
- **IAreaMap:**获取区域地图相关显示数据，包括游戏中所有区域地图列表及根据id提取某一区域地图详细信息。
- **ILevel:**获取关卡信息。
- **IPlayer:**获取玩家信息。
- **IBag:**获取玩家背包信息。
- **IRatingSystem:**打分系统，用于向逻辑模块传递衣服搭配信息与当前所在关卡，获取关卡评级



及奖励。

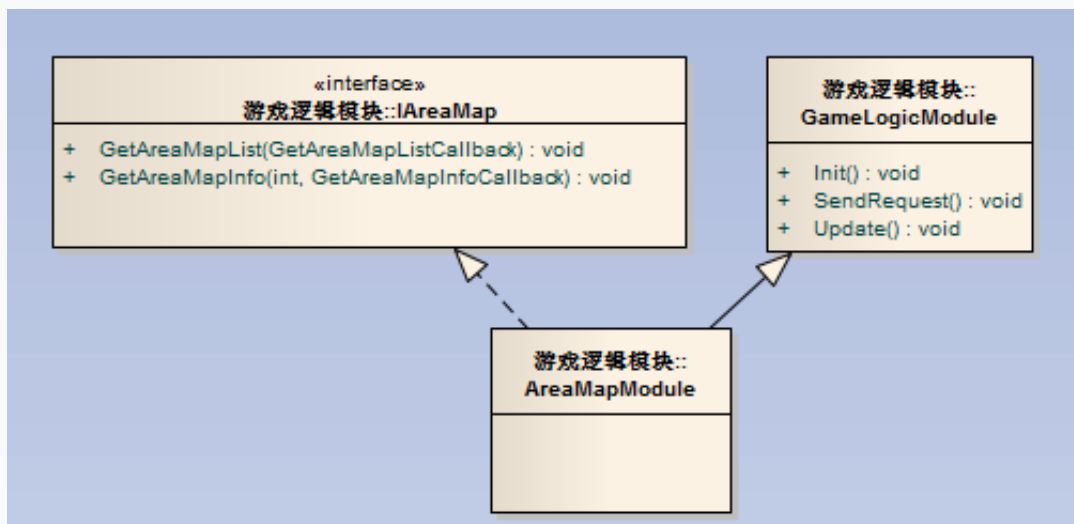
- **IGameItemDatabase**:游戏物品数据库接口，用于根据物品Id获取物品详细信息。
- **IGameItemMall**:游戏商城接口，目前不实现。

**与UI层交互的数据信息**：这些信息数据结构中的内容只是UI界面关心的。



**游戏逻辑模块**:游戏逻辑模块的职责是将UI层转发给服务器，并将服务器传回的内容解析并通知给UI

- **GameLogicModule**:每个游戏逻辑模块基类，每个游戏模块都从此继承并实现一个对UI层接口。



- **AreaMapModule**:区域地图模块，用来负责区域地图有关的信息呈递。
- **LevelModule**:关卡模块，用来负责关卡有关信息的呈递。
- **PlayerModule**:玩家模块，用来负责玩家状态信息的呈递。
- **BagModule**:背包模块，用来负责玩家背包信息的呈递。

- **RatingSystemModule:**评分系统模块，用来负责玩家关卡打分信息的呈递。
  - **GamelItemDatabaseModule:**游戏物品数据库模块，用来负责游戏中物品信息的查询。
  - **GamelItemMallModule:**游戏商城模块，用来负责游戏商城商品信息的呈递。
-