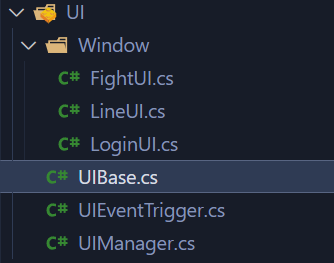
技术设计

**一、UI管理系统**

UI管理系统主要负责游戏中用户界面的实例化、功能实现以及更新，整个系统由若干类共同构成，主要分为UIManager类，UIBase类及其派生类以及UIEventTrigger类。



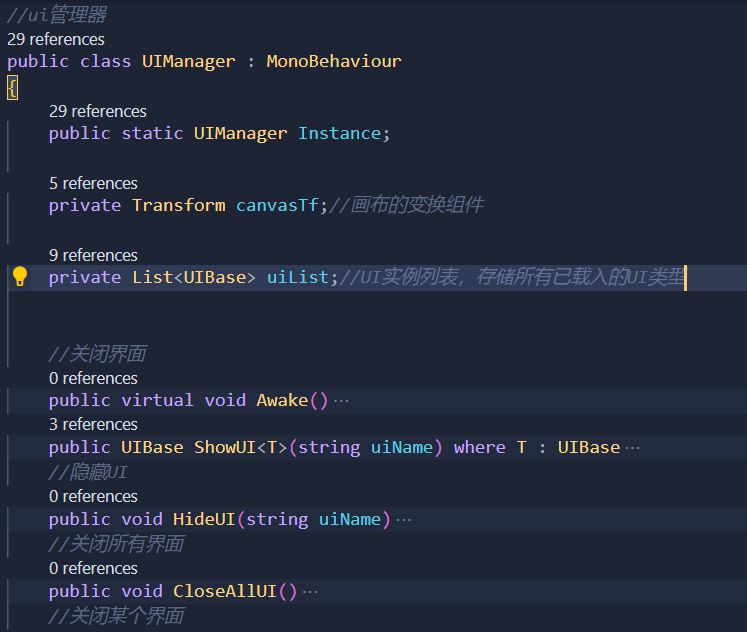
UI界面示例：





**1.UIManager类**

UIManager类负责管理所有UI从资源到游戏内存的实例化，管理UIBase类的列表，为UI的显示和功能提供了基本条件。其部分功能如下图所示：



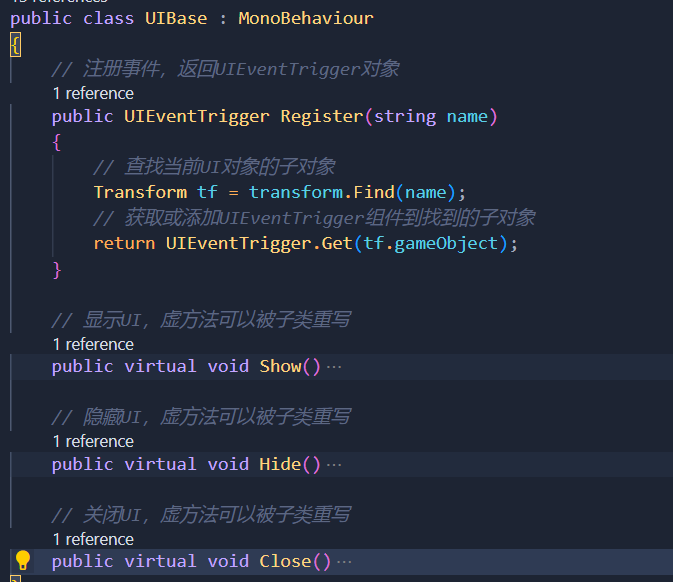
其中泛型方法ShowUI<T>(string uiName)为UI实例创建于管理的主要方法，它接受一个UI类型的名字如FightUI，返回并显示该类型UI。若该类型不存在于管理器的list中，则从Resources文件夹下加载实例化同名预制体。



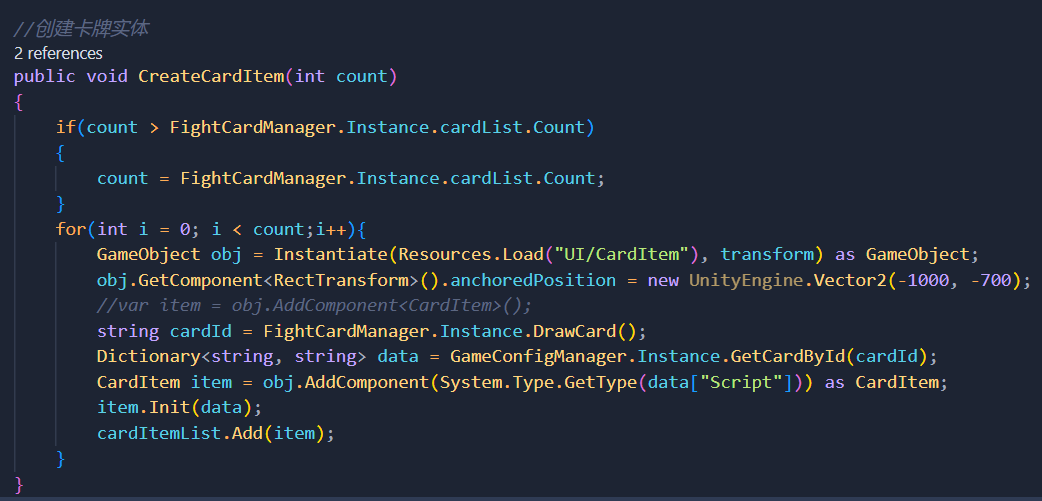
**2.UIBase类**

UIBase类为所有UI类的基类，提供了显示、注册事件、关闭等可被子类重写的虚方法，利用多态性实现了FightUI、LoginUI等不同的登录界面的不同功能。

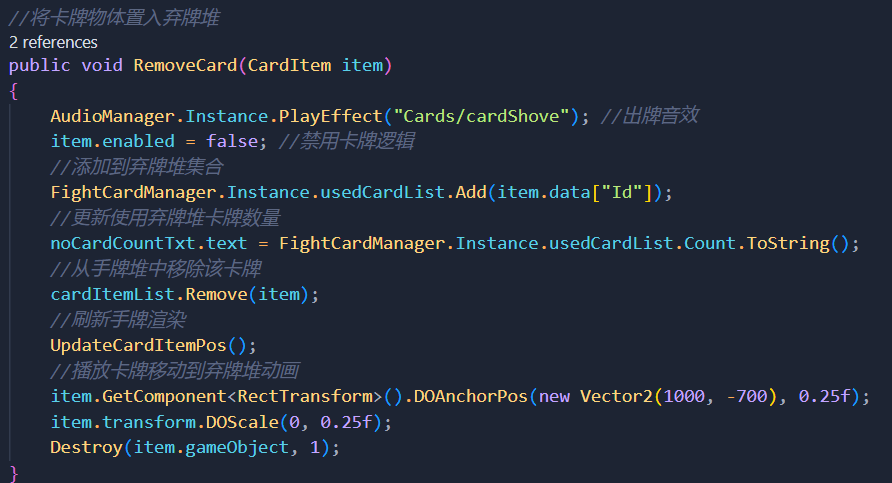
UIBase类实现：



FightUI示例：需要在FightUI中实现战斗界面玩家血量等UI显示更新、卡牌生成与显示、发牌弃牌动画等UI逻辑

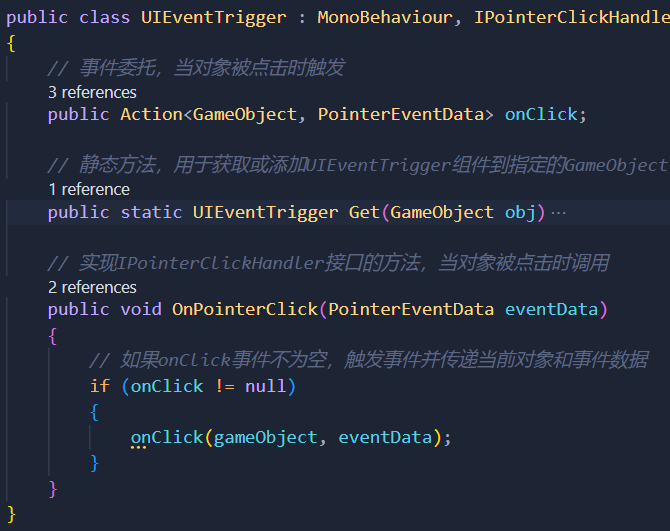






**3.UIEventTrigger类**

UIEventTrigger类用于监听鼠标点击事件，挂载于各个UI类的实例上，为UI点击事件提供了委托注册和处理逻辑。



**二、音效管理系统**

由音效管理器AudioManager单个类实现，该类为单例模式，在游戏主循环中start()位置完成初始化与实例化，可被其他脚本在需要播放音乐或音效时调用。

其核心方法为PlayBGM与PlayEffect，两者分别接受需播放音频的名字string作为参数，从Resources中播放对应音效资源。



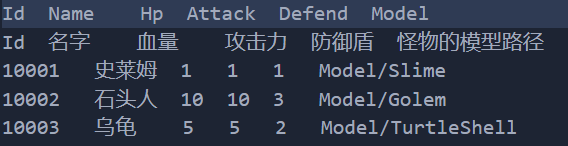
**三、信息配置系统**

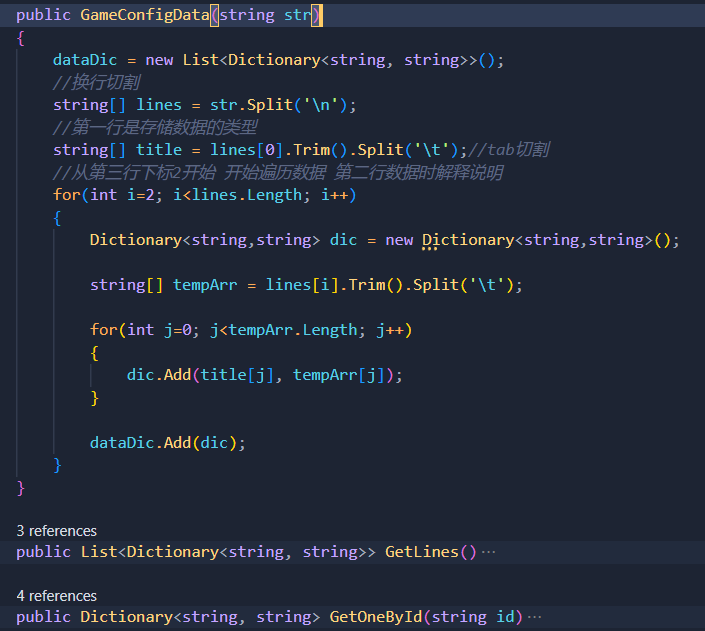
信息配置系统负责从配置文件（由各个excel配置表转换而来的txt文件）中读取卡牌、敌人、关卡的配置信息，将各个属性的键值对存储在相应字典中，并为其他类提供了查询方法。信息配置系统主要由GameConfigData类与GameConfigManager类构成。

1. **GameConfigData类**

GameConfigData类为游戏配置表类，卡牌、卡牌类型、敌人、关卡等每一个类型的配置表对应一个配置表类，该类实现了从txt中读取配置表信息，并将每一行数据存储到一个字典中，构成一个字典列表，且为其他类提供了根据Id查询信息的方法。

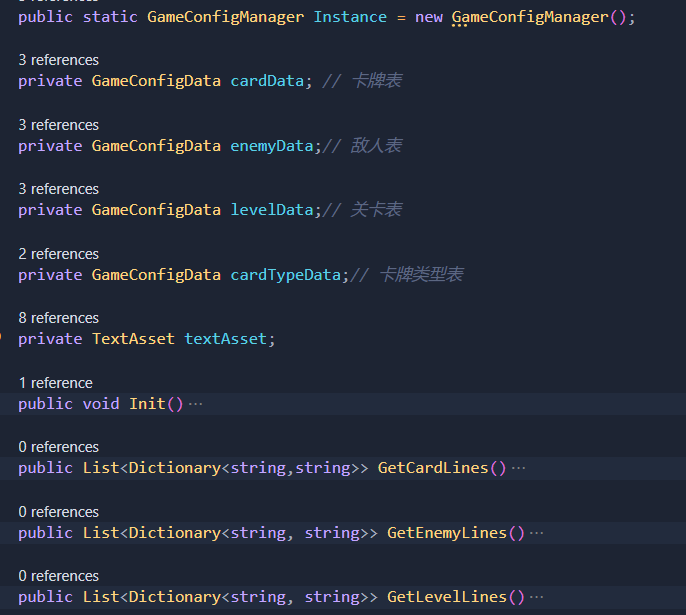
配置表示例：





1. **GameConfigManager类**

该类负责管理所有类型的配置表并为各个类型的配置表提供查询接口。该类为单例模式，便于其他脚本查询游戏配置信息。



**四、战斗逻辑系统**

战斗逻辑系统负责回合制战斗的逻辑处理，由RoleManager，EnemyManager，FightCardManager以及FightManager四个管理器以及与战斗逻辑有关的Card、Fight等相关脚本共同构成。它们共同实现了主角、敌人的实例化与初始化，以及双方战斗回合的行动逻辑与功能。

1. **RoleManager**

功能较为简单，负责管理主角持有的卡牌，维护一个主角目前的牌库，可以进行初始化，添加于删除。



1. **EnemyManager及相关脚本**

主要由Enemy与EnemyManager构成。

其中Enemy为敌方个体的管理类，存储相应模型预制体与怪物生命、攻击力等数据，实现了随机动作选择、显示行动意图、攻击受击动画播放与逻辑处理、血量槽与护盾槽更新等功能。

攻击动画播放与逻辑处理部分代码示例：



EnemyManager为敌方全体的管理类，实现了敌方单位实例化与模型加载、敌方回合行动顺序执行、敌方死亡单位模型移除等管理功能



1. **FightCardManager**

FightCardManager为卡牌管理器类，它与卡牌基类CardItem及其AttackCard，DefendCard等派生类共同管理游戏中的卡牌效果，如战斗开始卡牌初始化、抽牌弃牌逻辑、卡牌功能逻辑以及卡牌选中、拖拽、使用的动画效果以及功能，完善了游戏的卡牌逻辑。

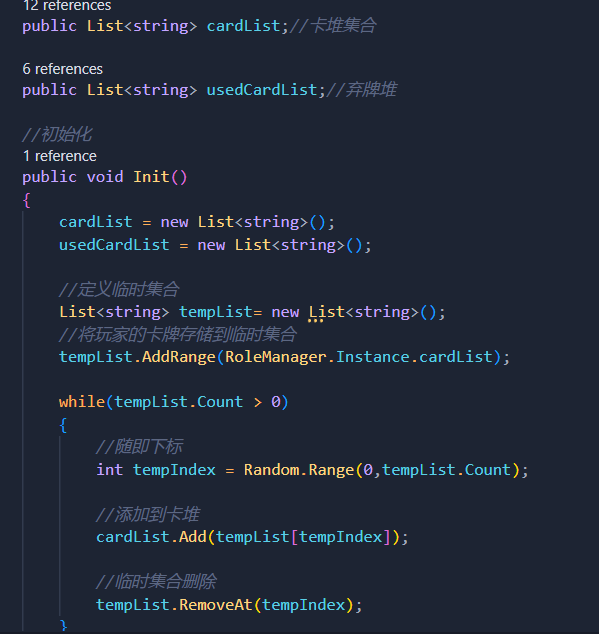
CardItem类：

提供了所有卡牌类型的通用方法，如鼠标悬停在卡牌上时放大卡牌并高亮边框、拖拽效果等

、

FightCardManager类：

从RoleManager中读入玩家牌库实现战斗中抽牌堆初始化、管理抽牌堆与弃牌堆，处理抽牌弃牌逻辑



1. **FightManager**

FightManager为战斗管理器类，它维护了玩家血量、能量、护盾等的当前值于最大值，与战斗回合FightUnit及其派生类Fight\_EnemyTurn、Fight\_PlayerTurn等共同管理战斗中的回合更替，以及敌我双方回合行动顺序与行动逻辑。

FightManager类中的回合更替：



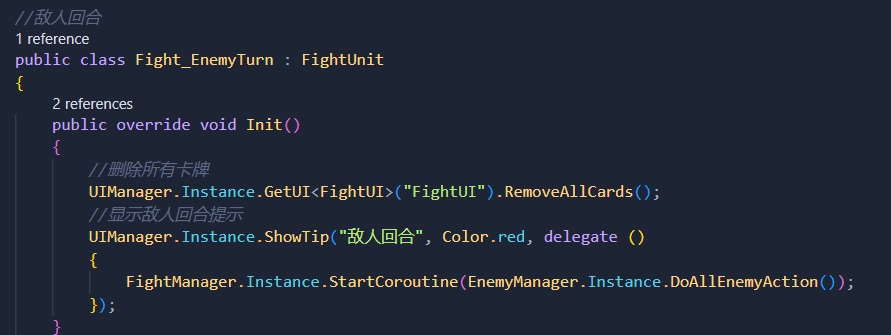
Fight\_PlayerTurn玩家回合：

弹出提示，回复能量并调用抽卡逻辑、更新UI



Fight\_EnemyTurn敌方回合：

弹出提示并通过协程启动EnemyManager中的敌方行动方法，实现多帧的动画播放

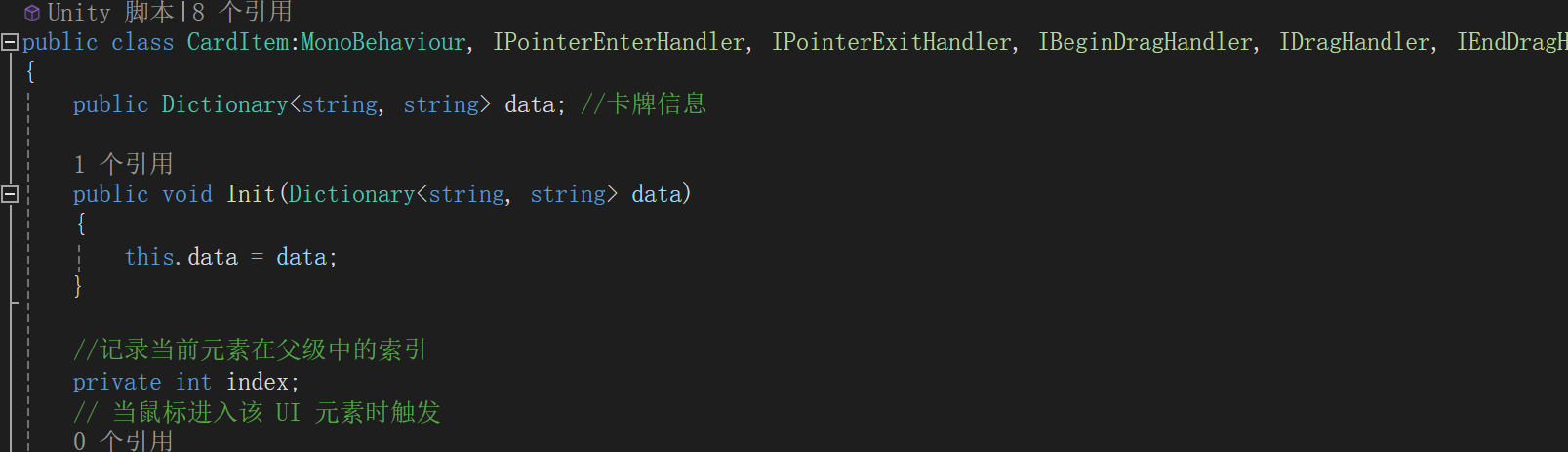


**五、卡牌管理系统**

卡牌管理系统主要负责具体卡牌的逻辑处理，由CardItem、AttackCardItem、DefendCard、AddCard等管理器及卡牌分支脚本构成，具体数量及内容与设计的具体卡牌有关，它们实现了战斗中卡牌相关的动画效果及打出之后的效果处理，除CardItem外其余卡牌分支类均继承自CardItem类。

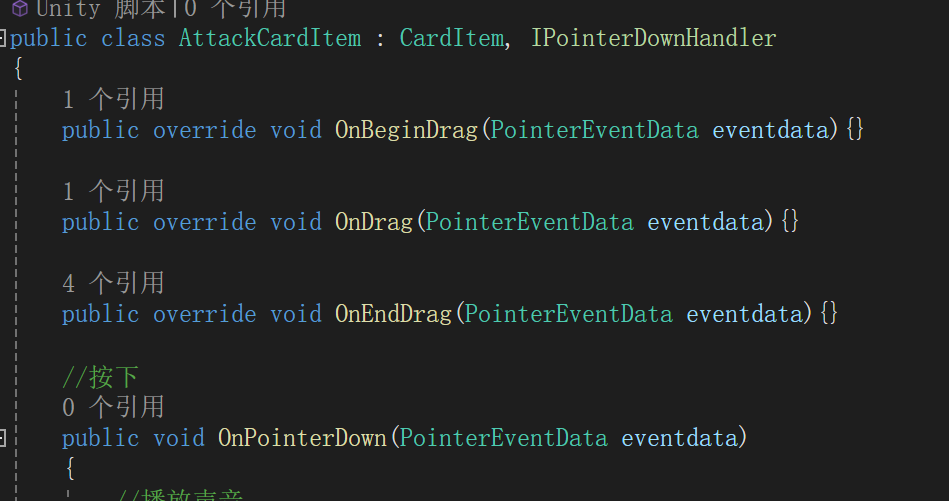
1. CardItem

主要负责管理卡牌相关的玩家交互效果处理，如光标悬浮时卡牌放大、离开后缩小至原本大小、拖拽卡牌前后的动画效果。此外它还负责处理玩家尝试打出卡牌时的逻辑，如判断费用是否足够并提示，以及在成功打出卡牌后的特效创建，相关内容均封装于CardItem类中。



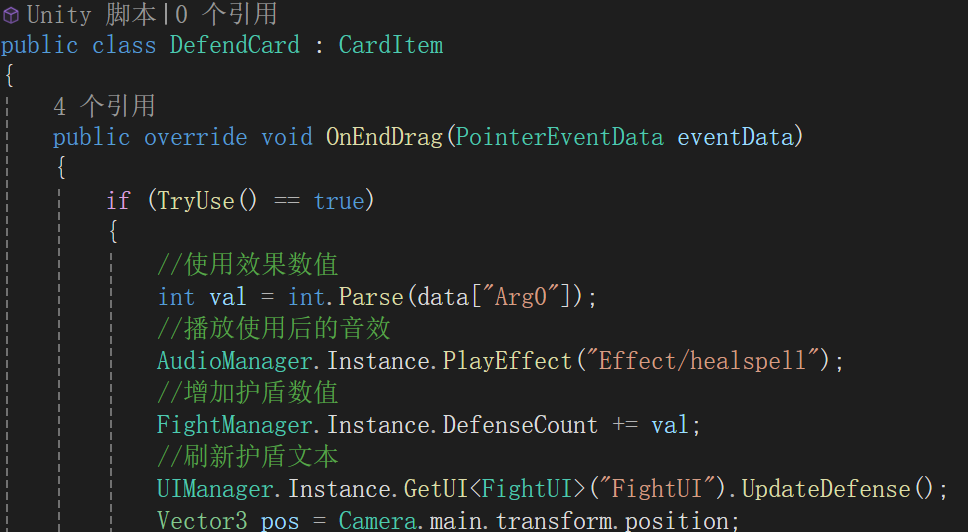
1. AttackCardItem

负责处理攻击牌的效果，包括显示攻击指向目标的贝塞尔曲线及其箭头、展示攻击特效，根据攻击结果对应降低敌人生命值等，相关接口均封装在AttackCardItem类中。



1. DefendCard

负责处理防御牌的效果，内容较为简单，播放对应特效并对应增加护盾值即可，内容均封装于DefendCard类中。



1. AddCard

负责处理抽卡牌的效果，首先判断牌库中是否有卡可以抽取，若有则播放将其抽入手牌的特效，相关内容均封装于AddCard类中。

