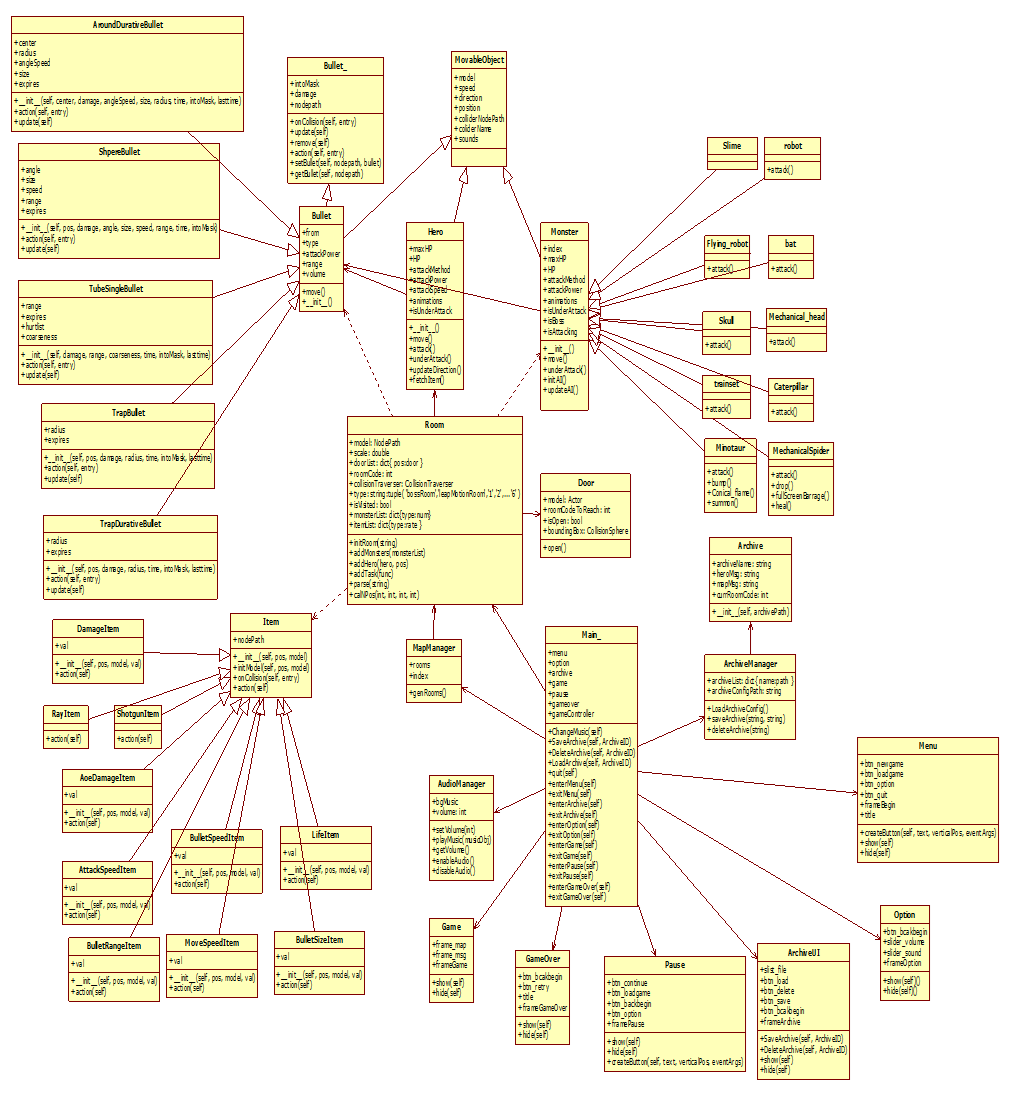
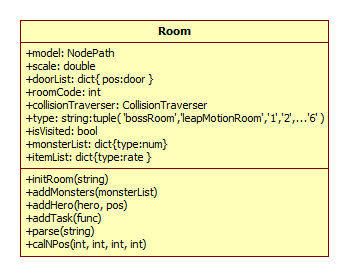
详细设计文档

# 整体类图：



# Room类：



## 数据成员：

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| model | NodelPath | 房间模型的句柄 | 链接到render节点，游戏其他模型（门、怪物、道具、主角）都链接到房间模型节点下 |
| scale | double | 房间模型缩放的比例 | 房间与门模型共用同一缩放比例 |
| doorList | Dict{ pos:door } | 房间中存在的门的集合，pos表示门的方位（上下左右），door代表门实例 |  |
| roomCode | int | 房间编号 | 一个编号对应唯一的一个房间 |
| collisionTraverser | CollisionTraverser | Panda3d碰撞检测事件管理器 | 房间中所有碰撞检查都要注册到这个管理器中，统一管理 |
| type | string | 标识房间的类型，房间类型属于( 'bossRoom','leapMotionRoom','1','2',...'6' )中的一种 | ‘bossRoom’是关卡主路径的最终房间，会出现boss级怪物  ‘leapMotion房间’是只出现在分支路径的房间，一个关卡地图只有一个，可获取强力道具  ‘1’---‘6’房间等级越高，怪物等级越高，道具加成越大 |
| IsVisited | bool | 标识该房间是否访问过 | 小地图要根据该变量决定是否显示该房间的位置 |
| monsterList | Dict{ type:num } | 房间中的怪物信息集合，type代表怪物类型，num代表该类型怪物的数量 | 当怪物死亡后，相应num要减小 |
| ItemList | Dict{ type:rate } | 该房间可获取的道具的集合，type代表道具类型，rate代表该类道具爆出的几率 | 道具是怪物死亡时根据房间类型、怪物等级、rate和一个随机数计算结果来判断是否爆出，这个变量只是道具列表信息 |

## 成员函数：

### initRoom(self, roomInfo)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| RoomInfo | String | json格式描述的房间具体信息(包括门、怪物、道具、房间编号) | |
| **返回值类型** | | | **返回值含义** |
| Bool | | | 初始化是否成功，如果发生异常、模型文件不存在、信息不完整则返回false |
| **函数说明** | | | 用于初始化房间，包括加载房间和门模型、解析传入的房间信息字符串，计算怪物位置，加载怪物模型 |
| **备注** | | | 由关卡管理器调用，关卡管理器只维护一个房间实例，创建实例时需要加载房间模型，进入新房间时只需要清空之前的信息，再根据传入的房间信息进行构造 |

### AddMonster（self,monsterList）

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| monsterList | dict | 成员变量 | |
| **返回值类型** | | | **返回值含义** |
| Bool | | | 添加怪物模型成功返回true |
| **函数说明** | | | 根据怪物列表放置怪物模型 |
| **备注** | | | 需要计算出不相交的k个位置，再加载模型并放置。k是该房间怪物总数,这个K由关卡生成器生成房间的时候生成 |

### addHero(self,hero)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| hero | Hero | 自定义类型，维护英雄模型及相应操作 | |
| Pos | int | 英雄模型应该放置的位置，与之前进入的门的方位相对，或放置到中间（初始房间 | |
| **返回值类型** | | | **返回值含义** |
| void | | |  |
| **函数说明** | | | 把主角放到房间中 |
| **备注** | | | 主角由game类维护，全局只有一个实例 |

### AddTask(self,func)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| func | Function | 碰撞事件处理函数 | |
| **返回值类型** | | | **返回值含义** |
| void | | |  |
| **函数说明** | | | 往房间CollisionTraverser添加碰撞处理函数 |
| **备注** | | | 房间中的碰撞处理都交由CollisionTraverser处理，处理函数都要用这个函数注册 |

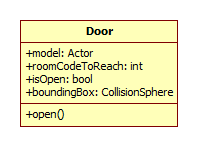
### Prase(self,roomInfo)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| roomInfo | string | Json格式的房间具体信息 | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 解析并设置房间信息 |
| **备注** | | |  |

### calNPos(self, num, width ,len ,sideLen)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Num | Int | 要计算的位置的数量 | |
| Width | Int | 房间宽度 | |
| Len | Int | 房间长度 | |
| SideLen | Int | 每个位置的边长 | |
| **返回值类型** | | | **返回值含义** |
| [ ( x,y) ] | | | N个位置左上角坐标 |
| **函数说明** | | | 在l\*w长方形中计算n个a\*a的不相交位置 |
| **备注** | | |  |

# Door类



## 数据成员

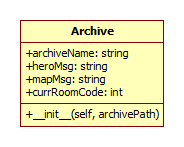
|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| model | Actor | 门模型的句柄 | 链接到房间模型节点，有开门动画 |
| RoomCodeToReach | int | 该门通向的房间的编号 |  |
| IsOpen | bool | 标识门是否打开 | 如果为true，要把门模型进行旋转后放置，做成打开的门 |
| BoundingBox | CollisionSphere | 球形包围盒 | 碰撞处理函数要注册到room对象中 |

## 成员函数

### Open(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
| void | | |  |
| **函数说明** | | | 在玩家杀完房间中的怪物并碰撞到一扇门后，门的碰撞处理函数调用该函数，播放门打开的动画，更改门的状态 |
| **备注** | | |  |

# Archieve类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| archiveName | string | 该存档的名称 | 用户命名 ，使用unicode编码以支持中文 |
| heroMsg | String | Json格式表示的英雄对象信息 | 可使用python库重建Hero对象 |
| MapMsg | string | Json格式表示的地图列表信息 | 可使用python库重建地图 |
| CurrRoomCode | int | 保存该存档时英雄所在房间的编码，如果该房间的怪物没有清空，则保存来到该房间之前的房间的编码 |  |

## 成员函数

### \_\_init\_\_(self,archievePath)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| archivePath | string | 该存档保存文件的绝对路径(建议相对路径) | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 加载传入的存档文件 |
| **备注** | | |  |

# ArchiveManager类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| archiveList | Dict{ name:path } | 所有存档的集合 name是存档的名称  Path是相应存档文件的路径 |  |
| archiveConfigPath | String | 存档配置文件的路径 | 存档配置文件中一行存储一份存档的名称；文件路径（csv格式） |

## 成员函数

### \_\_init(self, archiveConfigPath=’./archive’)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| archiveConfigPath | String | 存档配置文件的路径 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 根据传入的路径创建一个ArchiveManager对象 |
| **备注** | | | 路径有默认值 ‘./archive/’ |

### LoadArchiveConfig（self）

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 加载存档配置文件 |
| **备注** | | | 存档配置文件路径是ArchiveManager构造时传入的，有默认值 ‘./archive/’ |

### SaveArchive(self,archivePath,name)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| archivePath | string | 存档文件路径 | |
| Name | string | 存档名称，玩家命名，unicode格式 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 保存游戏到存档文件 |
| **备注** | | | 保存游戏地图信息、当前进度（第几关、所在房间编号）、英雄信息。通过调用相应类的序列化方法来获取相应json格式的字符串，然后将记录写入到存档配置文件中 |

### deleteArchive(self,name)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| name | string | 存档名称 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 删除存档名称对应的存档 |
| **备注** | | | 先删除存档文件，再删除archiveList中的记录，最后更改存档配置文件 |

# AudioManager类



|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| bgMusic |  | 背景音乐的句柄 | AudioManager构造时读入 |
| Volume | int | 音量大小 |  |

## 数据成员

## 成员函数

### SetVolume(volume)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| volume | int | 新的音量大小 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 设置音量大小 |
| **备注** | | |  |

### GetVolume(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
| int | | | 当前音量大小 |
| **函数说明** | | | 获取当前音量大小 |
| **备注** | | |  |

### playMusic(self, musicObj)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| musicObj |  | 要播放的音效的句柄 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 播放背景音、音效 |
| **备注** | | | 要先根据globalSoundEffect是否开启音效设置，为true才播放,可在  enableAudio（）和disableAudio()中设置 |

### enableAudio(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 开启游戏音效 |
| **备注** | | |  |

### disableAudio(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 关闭游戏音效 |
| **备注** | | |  |

# Item类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| nodePath | NodePath | 道具在场景树下的节点，包括模型节点与碰撞节点 | nodePath |

## 成员函数

### \_\_init\_\_(self,pos,model)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Pos | Vec3() | 模型初始位置 | |
| Model | String | 模型所在的文件路径 | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 1. 内部调用self.initModel()进行初始化， 2. 在当前房间对象注册该模型 |
| **备注** | | |  |

### initModel(self,pos,model)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Pos | Vec3 | 模型在场景中额初始位置 | |
| Model | Str | 道具模型所在文件路径 | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 在当前房间生成道具  1.判断当前房间是否为空，如果为空，返回  2.新建NodePath，命名为”item”，加载model，链接到当前房间节点下，设置位置为pos，将NodePath赋值给nodePath  3.新建CollisionNode，命名为”item”，在该节点添加一个球状碰撞体，链接到上面节点下  4.为上面碰撞节点设置碰撞掩码，添加到碰撞系统中 |
| **备注** | | |  |

### onCollision(self,entry)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Entry | []python列表 | 碰撞记录 | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 用于ShowBase注册碰撞事件的回调函数  1. 通过entry.getIntoNodePath()得到道具NodePath，通过NodePath得到道具对象  2.调用道具对象的action()  3.调用道具对象的nodePath. removeNode()  4.在当前房间注销该道具 |
| **备注** | | |  |

### Action(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 这里定义了一个接口，用于道具被销毁之后产生效果，实现为空，子类对这个函数进行实现产生自己的相关动作 |
| **备注** | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
|  |  |  |  |

# DamageItem类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| Val | Int | 道具生效时改变主角的攻击值 |  |

## 成员函数

### \_\_init\_\_(self,pos,model, valAttack)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Pos | Vec3 | 生成道具的位置 | |
| Model | String | 道具所在文件路径 | |
| ValAttack | Int | 道具能改变的攻击值 | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 道具类的默认构造函数  1.val= ValAttack  2.调用self.initModel (pos, model) 生成道具  3.在当前房间对象注册该道具 |
| **备注** | | | 其中的initModel是继承于父类的函数 |

### Action(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 道具被销毁之后产生效果  1.当前Hero攻击=攻击+val |
| **备注** | | |  |

# moveSpeedItem类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| Val | Int | 道具生效时改变主角的移动速度值 |  |

## 成员函数

### \_\_init\_\_(self,pos,model, valSpeed)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Pos | Vec3 | 生成道具的位置 | |
| Model | String | 道具所在文件路径 | |
| ValSpeed | Int | 道具能改变的移动速度值 | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 道具类的默认构造函数  1.val= ValSpeed  2.调用self.initModel (pos, model) 生成道具  3.在当前房间对象注册该道具 |
| **备注** | | | 其中的initModel是继承于父类的函数 |

### Action(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 道具被销毁之后产生效果  1.当前Hero移速=移速+val |
| **备注** | | |  |

# AttacSpeedItem类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| Val | Int | 道具生效时改变主角的攻击速度值 |  |

## 成员函数

### \_\_init\_\_(self,pos,model, valAttackSpeed)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Pos | Vec3 | 生成道具的位置 | |
| Model | String | 道具所在文件路径 | |
| ValAttackSpeed | Int | 道具能改变的攻击速度值 | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 道具类的默认构造函数  1.val= ValAttackSpeed  2.调用self.initModel (pos, model) 生成道具  3.在当前房间对象注册该道具 |
| **备注** | | | 其中的initModel是继承于父类的函数 |

### Action(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 道具被销毁之后产生效果  1.当前Hero攻速=攻速+val |
| **备注** | | |  |

# BulletSpeedItem类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| Val | Int | 道具生效时改变主角发射的子弹的移动速度速度值 |  |

## 成员函数

### \_\_init\_\_(self,pos,model, ValBulletSpeed)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Pos | Vec3 | 生成道具的位置 | |
| Model | String | 道具所在文件路径 | |
| ValBulletSpeed | Int | 道具能改变的子弹移动速度值 | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 道具类的默认构造函数  1.val= ValBulletSpeed  2.调用self.initModel (pos, model) 生成道具  3.在当前房间对象注册该道具 |
| **备注** | | | 其中的initModel是继承于父类的函数 |

### Action(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 道具被销毁之后产生效果  1.当前子弹速度=子弹速度速+val |
| **备注** | | |  |

# BulletRangeItem类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| Val | Int | 道具生效时改变主角发射的子弹的射程值 |  |

## 成员函数

### \_\_init\_\_(self,pos,model, ValBulletRange)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Pos | Vec3 | 生成道具的位置 | |
| Model | String | 道具所在文件路径 | |
| ValBulletRange | Int | 道具能改变的子弹射程值 | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 道具类的默认构造函数  1.val= ValBulletRange  2.调用self.initModel (pos, model) 生成道具  3.在当前房间对象注册该道具 |
| **备注** | | | 其中的initModel是继承于父类的函数 |

### Action(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 道具被销毁之后产生效果  1.当前Hero攻速=射程值+val |
| **备注** | | |  |

# BulletSizeItem类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| Val | Int | 道具生效时改变主角发射的子弹的大小值 |  |

## 成员函数

### \_\_init\_\_(self,pos,model, ValBulletSize)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Pos | Vec3 | 生成道具的位置 | |
| Model | String | 道具所在文件路径 | |
| ValBulletSize | Int | 道具能改变的子弹大小值 | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 道具类的默认构造函数  1.val= ValBulletSize  2.调用self.initModel (pos, model) 生成道具  3.在当前房间对象注册该道具 |
| **备注** | | | 其中的initModel是继承于父类的函数 |

### Action(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 道具被销毁之后产生效果  1.当前Hero的子弹大小=子弹大小+val |
| **备注** | | |  |

# LifeItem类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| Val | Int | 道具生效时改变主角发射的子弹的大小值 |  |

## 成员函数

### \_\_init\_\_(self,pos,model, ValLife)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Pos | Vec3 | 生成道具的位置 | |
| Model | String | 道具所在文件路径 | |
| ValLife | Int | 要改变的血量 | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 道具类的默认构造函数  1.val= ValLife  2.调用self.initModel (pos, model) 生成道具  3.在当前房间对象注册该道具 |
| **备注** | | | 其中的initModel是继承于父类的函数 |

### Action(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 道具被销毁之后产生效果  1.当前Hero的血量=血量+val |
| **备注** | | |  |

# AoeDamageItem类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| Val | Int | 道具生效时产生范围伤害的伤害值 |  |

## 成员函数

### \_\_init\_\_(self,pos,model, ValAoeDamage)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Pos | Vec3 | 生成道具的位置 | |
| Model | String | 道具所在文件路径 | |
| ValAoeDamage | Int | 产生范围伤害的伤害值 | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 道具类的默认构造函数  1.val= ValAoeDamage  2.调用self.initModel (pos, model) 生成道具  3.在当前房间对象注册该道具 |
| **备注** | | | 其中的initModel是继承于父类的函数 |

### Action(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 道具被销毁之后产生效果  1.生成一个覆盖整个地图的对敌子弹，伤害为val |
| **备注** | | |  |

# RayItem类



## 数据成员

## 成员函数

### Action(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 道具被销毁之后产生效果  1.改变主角攻击机制为发射激光 |
| **备注** | | | 这里仅仅改变主角发射子弹的类型，对其他属性不进行改变 |

# ShotGunItem类



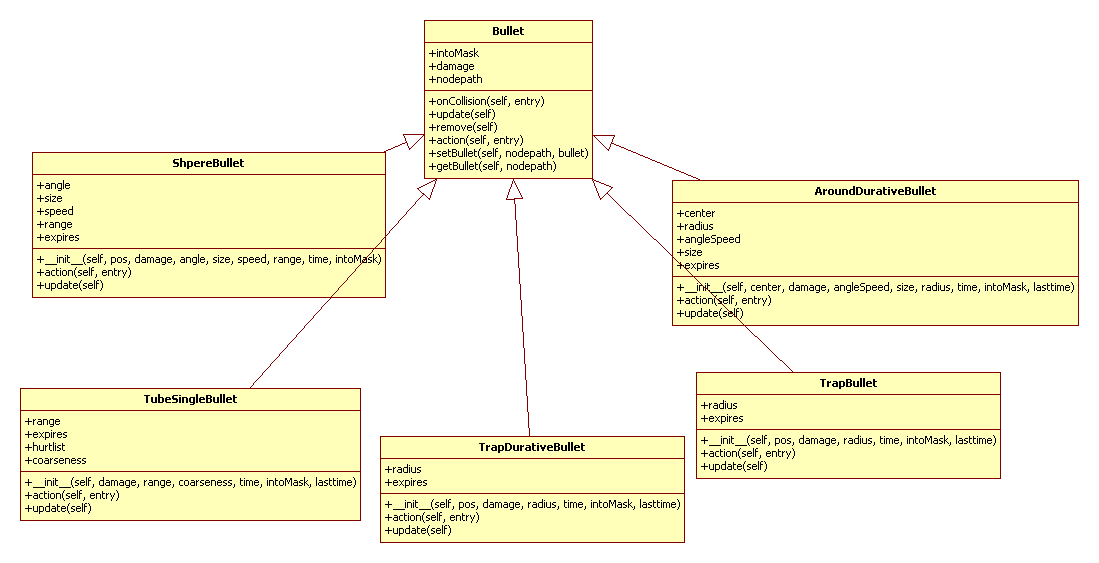
## 数据成员

## 成员函数

### Action(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 道具被销毁之后产生效果  1.改变主角攻击机制为发射霰弹 |
| **备注** | | | 这里仅仅改变主角发射子弹的类型，对其他属性不进行改变 |

# Bullet类



## 数据成员：

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| intoMask | BitMask32 | 碰撞掩码，用于区分伤害对象 |  |
| damage | Int | 伤害值 |  |
| nodepath | NodePath | 子弹的场景节点 | n |

## 成员函数：

### OnCollision(self,entry)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| entry | []pythonlist | 碰撞记录 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 生成房间时，碰撞处理器向所有from物体注册的对Bullet及其子类的碰撞回调函数：  1. 通过entry.getIntoNodePath()得到子弹NodePath，通过NodePath得到子弹  2.调用子弹对象的action(entry)  3.调用子弹对象的remove(self) |
| **备注** | | |  |

### Update(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 每一帧调用的更新子弹当前状态的函数，由子类实现 |
| **备注** | | | 接口函数 |

### Remove（self）

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 子弹达成移除条件时调用的函数  1. 调用nodePath. removeNode() |
| **备注** | | | 接口函数 |

### Action(self,entry)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 子弹生效时调用的函数，由子类实现 |
| **备注** | | | 接口函数 |

### SetBullet(self,nodepath,bullet)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Nodepath | NodePath | 要设置子弹对象的节点 | |
| Bullet | Bullet | 要绑定的子弹对象 | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 利用NodePath的setPyhonTag将nodepath与bullet绑定 |
| **备注** | | |  |

# ShereBullet类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| Angle | 依据具体实现 | 子弹当前飞行角度 |  |
| size | double | 子弹大小倍率 | 子弹由模型缩放而来 |
| speed | double | 子弹飞行速度 |  |
| range | double | 子弹的射程 |  |
| expires | Int | 子弹消失时间 | 子弹将会消失的时刻 |

## 成员函数

### \_\_init\_\_(self,pos,damage,angle,size,speed,range,time,intoMask)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Pos | Vec3 | 子弹的初始位置 | |
| Damage | Int | 子弹可以造成的伤害值 | |
| Angle | 待定 | 子弹的飞行角度 | |
| Size | Double | 子弹相对模型的缩放比例 | |
| Speed | Int | 子弹的飞行速度 | |
| Range | Int | 子弹的射程 | |
| Time | 根据具体实现 | 当前时刻（不一定是系统时刻） | |
| IntoMask | 依据具体实现 | 碰撞掩码，用于检测是否需要发生碰撞事件（不会对队友造成伤害） | |
| **返回值类型** | | | **返回值含义** |
| Void | | |  |
| **函数说明** | | | 构造函数   1. 对angle，damage ，size，speed，range，intoMask赋值，expires= time+( range/ speed) 2. 根据angle，size，pos和加载默认模型构造nodepath，通过setBullet将nodepath与子弹对象绑定 |
| **备注** | | |  |

### Update（self）

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 每一帧调用的更新子弹当前状态的函数  1.判断是否expires>task.time，若否，调用remove()  2.根据angle与speed更新子弹的位置与旋转角度 |
| **备注** | | |  |

### Action(self,entry)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Entry | []python list | 碰撞记录列表 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 子弹生效时调用的函数  1. 在entry中获取From节点，再获取From对象  2. 对象的血量=血量-damage  3. 调用remove() |
| **备注** | | |  |

# TubeSingleBullet类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| range | double | 子弹的射程 |  |
| expires | Int | 子弹消失时间点 |  |
| hurtlist | List<Ogre > | 已造成伤害对象列表 |  |
| coarseness | double | 子弹的界面的半径 |  |

## 成员函数

### \_\_init\_\_(self, damage, range, coarseness, time, intoMask,lasttime)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Damage | Int | 子弹可以造成的伤害值 | |
| Range | Int | 子弹的射程 | |
| Coarseness | Double | 子弹截面的半径 | |
| Time | 依据具体实现 | 当前时刻 | |
| Lasttime | Int | 子弹生命周期 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 构造函数   1. 对damage，range， coarseness，intoMask赋值，expires= time+ lasttime 2. 根据pos，range ，coarseness和加载默认模型构造nodepath，通过setBullet将nodepath与子弹对象绑定 |
| **备注** | | |  |

### Update(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 每一帧调用的更新子弹当前状态的函数  1.判断是否expires>task.time，若否，调用remove() |
| **备注** | | |  |

### Action(self,entry)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Entry | Python列表 | 碰撞记录 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 子弹生效时调用的函数  1.在entry中获取From节点，再获取Ogre对象，判断hurtlist里是否有该对象，若有，则返回，若无，则在hurtlist添加该对象  2. Ogre对象的血量=血量-damage |
| **备注** | | |  |

# TrapBullet类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| radius | double | 子弹的半径 |  |
| expires | Int | 子弹消失时间 |  |

## 成员函数

### \_\_init\_\_(self, pos,damage, radius, time, intoMask,lasttime)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Pos | Vec3 | 子弹初始位置 | |
| Damage | Int | 子弹可以造成的伤害值 | |
| Radius | Double | 子弹造成伤害的半径 | |
| Time | 依据具体实现 | 当前时刻 | |
| Lasttime | Int | 子弹生命周期 | |
| IntoMask | 依据具体实现 | 碰撞掩码 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 构造函数  1.对damage，radius，intoMask赋值，expires= time+ lasttime  2.根据pos，radius和加载默认模型构造nodepath，通过setBullet将nodepath与子弹对象绑定 |
| **备注** | | |  |

### Update(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 每一帧调用的更新子弹当前状态的函数  1.判断是否expires>task.time，若否，调用remove() |
| **备注** | | |  |

### Action(self,entry)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Entry | Python列表 | 碰撞记录 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 子弹生效时调用的函数  1.在entry中获取From节点，再获取From对象，  2.对象的血量=血量-damage  3. 调用remove() |
| **备注** | | |  |

# TrapDurativeBullet类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| radius | double | 子弹造成伤害的半径 |  |
| expires | Int | 子弹消失时间 |  |

## 成员函数

### \_\_init\_\_(self, pos,damage, radius, time, intoMask,lasttime)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Pos | Vec3 | 子弹初始位置 | |
| Damage | Int | 子弹可以造成的伤害值 | |
| Radius | Double | 子弹造成伤害的半径 | |
| Time | 依据具体实现 | 当前时刻 | |
| Lasttime | Int | 子弹生命周期 | |
| IntoMask | 依据具体实现 | 碰撞掩码 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 构造函数  1.对damage，radius，intoMask赋值，expires= time+ lasttime  2.根据pos，radius和加载默认模型构造nodepath，通过setBullet将nodepath与子弹对象绑定 |
| **备注** | | |  |

### Update(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 每一帧调用的更新子弹当前状态的函数  1.判断是否expires>task.time，若否，调用remove() |
| **备注** | | |  |

### Action(self,entry)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Entry | Python列表 | 碰撞记录 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 子弹生效时调用的函数  1.在entry中获取From节点，再获取From对象，  2.对象的血量=血量-damage  3. 调用remove() |
| **备注** | | |  |

# AroundDurativeBullet类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| radius | double | 子弹截面的半径 |  |
| expires | Int | 子弹消失时间 |  |
| center | LPoint3 | 子弹旋转圆心 |  |
| angleSpeed | double | 子弹旋转角速度 |  |
| size | double | 子弹相对模型缩放的比例 |  |
| angle | double | 子弹当前旋转角度 |  |

## 成员函数

### \_\_init\_\_(self, center,damage, angleSpeed , size ,radius, time, intoMask,lasttime)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Center | 依据具体实现 | 子弹旋转的圆心所在的轴 | |
| Damage | Int | 子弹可以造成的伤害值 | |
| Radius | Double | 子弹的截面的半径 | |
| AngleSpeed | Int | 子弹旋转的角速度 | |
| Size | Double | 子弹相对模型的缩放比例 | |
| Time | 依据具体实现 | 当前时刻 | |
| Lasttime | Int | 子弹生命周期（时间长度） | |
| IntoMask | 依据具体实现 | 碰撞掩码 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 构造函数，  center ,damage,angleSpeed,radius,size,intoMask赋值，angle=0,expires= time+ lasttime  2.根据center，radius, size, angle和加载默认模型构造nodepath，通过setBullet将nodepath与子弹对象绑定 |
| **备注** | | |  |

### Update(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 每一帧调用的更新子弹当前状态的函数  1.判断是否expires>task.time，若否，调用remove() |
| **备注** | | |  |

### Action(self,entry)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| Entry | Python列表 | 碰撞记录 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 子弹生效时调用的函数  1.在entry中获取From节点，再获取From对象，  2.对象的血量=血量-damage  3. 调用remove() |
| **备注** | | |  |

# MoveBle类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| model | Actor | 模型 |  |
| nodepath | NodePath | 模型的场景节点 |  |
| speed | int | 每秒可以移动的距离 |  |
| direction | int | 面向的方向，相对于相机的y轴正方向的逆时针角度 |  |
| position |  | 模型在场景3维坐标系下的位置 |  |
| colliderNodePath | NodePath | 碰撞体的节点路径 |  |
| colliderName | string | 碰撞体的名称 |  |
| sounds | map<string,sound> | 音效映射：move、attack、underAttack |  |

## 成员函数

# Hero类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| maxHP | int | 主角最大生命值 |  |
| HP | int | 主角当前生命值，当其为0时主角死亡，可通过道具进行回复 |  |
| attackMethod | int | 主角的攻击方式，用于在主角攻击时，设定主角的攻击特性 |  |
| attackPower | int | 主角的攻击力，在受攻击者的防御为0时，受到的伤害值 |  |
| animations | map<string,string> | 主角动画映射：move、attack、underAttack |  |
| isUnderAttack | int | 指示主角是否处于受攻击状态 |  |

## 成员函数

### \_\_init\_\_(self,modelFile,speed,direction,position,maxHP,HP,attackMethod,attackPower)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| 详见数据成员部分 |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 1. 根据模型文件名称载入模型、并载入相应的动画（命名规则“模型文件名-动画名“）、音效（命名规则”模型文件名-音效名） 2. 设定主角的移动速度、朝向、位置、最大生命值、当前生命值、攻击方式、攻击力等属性 3. 设置主角接收控制事件 4. 设置主角碰撞体及其名称，将场景节点与当前对象绑定 |
| **备注** | | |  |

### move(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 处理玩家发送移动指令事件   1. 判断 “主角是否处于受攻击状态” 是否为真，如果为真，直接返回； 2. 根据主角的朝向、位置、速度来更新主角的位置； 3. 同时播放”主角移动“动画；   向音效管理器发送“主角移动”事件； |
| **备注** | | |  |

### Attack(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 处理玩家发送攻击指令事件   1. 判断 “主角是否处于受攻击状态” 是否为真，如果为真，直接返回； 2. 根据主角的朝向、位置、攻击力，初始化子弹的攻击力、初始化位置、朝向； 3. 根据主角攻击方式，设置子弹的来源、数量、体积倍数、朝向、速度； 4. 向房间发送”子弹生成“事件（负责将子弹加入碰撞检测事件处理） |
| **备注** | | |  |

### underAttack（self，collisionEntry）

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| CollisionEntry |  | 攻击者 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 处理主角受到攻击（近战或远程）事件   1. 判断 “主角是否处于受攻击状态” 是否为真，如果为真，直接返回； 2. 根据attackPower决定主角受到的伤害值，相应地减少主角的当前生命值； 3. 设置 “主角是否处于受攻击状态” 为真； 4. 如果是远程攻击，，并且子弹为”不穿透“类型，向房间发送”子弹攻击“事件（以销毁子弹）   如果主角当前生命值小于等于0，播放”主角死亡“动画，完毕后，向关卡管理器发送”主角死亡“事件 |
| **备注** | | |  |

### FetchItem(self,collisionEntry)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| CollisionEntry |  | 道具 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 处理主角碰撞道具事件  根据道具功效，提升主角属性 |
| **备注** | | |  |

### updateDirection(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 主循环任务  在每一帧中，根据鼠标位置，更新主角朝向 |
| **备注** | | |  |

# Monster类



## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| index | int | 怪物在房间中的编号 |  |
| maxHP | int | 怪物最大生命值 |  |
| HP | int | 怪物当前生命值，当其为0时主角死亡，可通过道具进行回复 |  |
| attackMethod | int | 怪物的攻击方式，用于在怪物攻击时，设定怪物的攻击特性 |  |
| attackPower | int | 怪物的攻击力，在受攻击者的防御为0时，受到的伤害值 |  |
| animations | map<string,string> | 怪物动画映射：move、attack、underAttack |  |
| isUnderAttack | int | 指示怪物是否处于受攻击状态 |  |
| isBoss | int | 指示怪物是否为boss |  |
| isAttacking | int | 指示当前怪物是否处于攻击状态 |  |

## 成员函数

### \_\_init\_\_(self,modelFile,speed,direction,position,maxHP,HP,attackMethod,attackPower)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| 详见数据成员 |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 1. 根据模型文件名称载入模型、并载入相应的动画（命名规则“模型文件名-动画名“）、音效（命名规则”模型文件名-音效名） 2. 设定怪物的移动速度、朝向、位置、最大生命值、当前生命值、攻击方式、攻击力等属性 3. 设置怪物AI（集群）、攻击事件   设置怪物碰撞体及其名称，将场景节点与当前对象绑定 |
| **备注** | | |  |

### Move(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 主循环任务   1. 判断 “是否处于受攻击状态” 是否为真，如果为真，直接返回； 2. 判断“是否为boss”为真且 “是否处于攻击状态” 是否为真，如果为真，直接返回； 3. 更新AI 4. 同时播放”移动“怪物动画； 5. 向音效管理器发送“怪物移动”事件； |
| **备注** | | |  |

### 史莱姆(Slime)和小机器人-大史莱姆和双生机器人-攻击

Attack（self,collisionEntry）

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| collisionEntry |  | 主角 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 处理与主角的碰撞事件   1. 判断 “是否处于受攻击状态” 是否为真，如果为真，直接返回； 2. 近战攻击   近战攻击：  （主角处理碰撞检测事件，怪物不操作） |
| **备注** | | |  |

### 蝙蝠（bat）和飞行机器人（Flying\_robot）-攻击

Attack（self,collisionEntry）

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| collisionEntry |  | 主角 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 处理与主角的碰撞事件   1. 判断 “是否处于受攻击状态” 是否为真，如果为真，直接返回； 2. 范围攻击   范围攻击：   1. 延迟一秒后，播放”自爆动画“ 2. 向房间发送”怪物死亡“事件 |
| **备注** | | |  |

### 机器头（Mechanical\_head）和骷髅头（Skull）-攻击

attack(self，collisionEntry)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| collisionEntry： |  | 主角 | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 处理与主角的碰撞事件   1. 判断 “是否处于受攻击状态” 是否为真，如果为真，直接返回； 2. 远程攻击   远程攻击：   1. 根据怪物的朝向、位置、攻击力，初始化子弹的攻击力、初始化位置、朝向； 2. 根据怪物攻击方式，设置子弹的来源、数量、体积倍数、朝向、速度；   将子弹列表发送到当前房间对象（负责将子弹加入碰撞检测事件处理） |
| **备注** | | |  |

### 魔法柱和电磁塔-攻击

public attack(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 主循环任务  第一次初始化：   1. 初始化子弹列表，设定子弹为”穿透”类型 2. 根据怪物攻击方式，设置子弹的来源、数量、体积倍数、朝向、速度、初始位置； 3. 将子弹列表发送到当前房间对象（负责将子弹加入碰撞检测事件处理）   攻击：  1、不断更新子弹的朝向 |
| **备注** | | |  |

### 毛毛虫（Caterpillar）和小火车（trainset）-攻击

public attack(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 主循环任务  第一次初始化：   1. 初始化子弹列表，设定子弹为”穿透”类型 2. 根据怪物攻击方式，设置子弹的来源、数量、体积倍数、朝向、速度、初始位置，设定速度为0 3. 将子弹列表发送到当前房间对象（负责将子弹加入碰撞检测事件处理）   攻击：  1、不断更新子弹range，当range为0时，向房间发送“子弹销毁”事件 |
| **备注** | | |  |

### 牛头幽灵（Minotaur）和机械蜘蛛（MechanicalSpider）-攻击

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | | | 主循环事件   1. 每五秒随机触发一种攻击 2. 设置‘是否处于攻击状态’为真 3. 具体攻击方式 |
| **备注** | | |  |

### 牛头幽灵技能

#### 技能1：冲撞

|  |  |
| --- | --- |
| 函数签名 | public bump(self) |
| 函数功能 | 1. 设置‘是否处于攻击状态’为真 2. 存储主角的位置pos 3. 延迟一秒后，向pos方向，加速移动，维持0.3秒后，恢复速度 4. 设置‘是否处于攻击状态’为假 |
| 输入参数 | self:类对象引用； |
| 返回结果 | 无 |

#### 技能2：锥形火焰

|  |  |
| --- | --- |
| 函数签名 | public Conical\_flame(self) |
| 函数功能 | 1. 设置‘是否处于攻击状态’为真 2. 存储主角的位置pos 3. 播放boss”Conical\_flame”动画 4. 延迟一秒后， 5. 初始化火焰模型，创建火焰碰撞体，设置火焰攻击力、位置、方向（boss到pos的向量），向房间发送“火焰生成”事件（以进行碰撞检测），存留2s后销毁，向房间发送“火焰销毁“事件 6. 设置‘是否处于攻击状态’为假 |
| 输入参数 | self:类对象引用； |
| 返回结果 | 无 |

#### 技能3：召唤

|  |  |
| --- | --- |
| 函数签名 | public summon(self) |
| 函数功能 | 1. 设置‘是否处于攻击状态’为真 2. 播放boss”summon”动画 3. 延迟0.5s后， 4. 初始化3个史莱姆模型，设置其初始位置，并向房间发送”怪物生成“事件 5. 设置‘是否处于攻击状态’为假 |
| 输入参数 | self:类对象引用； |
| 返回结果 | 无 |

### 机械蜘蛛技能：

#### 技能1：坠地攻击

|  |  |
| --- | --- |
| 函数签名 | public drop(self) |
| 函数功能 | 1. 设置‘是否处于攻击状态’为真 2. 存储主角的位置pos 3. 播放”升天“动画（0.5s） 4. 更新位置为pos 5. 播放”坠地动画“（0.5s） 6. 设置‘是否处于攻击状态’为假 |
| 输入参数 | self:类对象引用； |
| 返回结果 | 无 |

#### 技能2：全屏弹幕

|  |  |
| --- | --- |
| 函数签名 | public fullScreenBarrage(self) |
| 函数功能 | 1. 设置‘是否处于攻击状态’为真 2. 延迟1s 3. 生成全屏随机子弹（一圈又一圈），每隔1s，增加一次，向房间发送“子弹生成”事件，持续5s 4. 设置‘是否处于攻击状态’为假 |
| 输入参数 | self:类对象引用； |
| 返回结果 | 无 |

#### 技能3：治疗

|  |  |
| --- | --- |
| 函数签名 | public v(self) |
| 函数功能 | 1. 设置‘是否处于攻击状态’为真 2. 延迟1s后 3. 播放”恢复“动画（5s） 4. 每秒恢复怪物一定（10）生命值 5. 持续五秒 6. 设置‘是否处于攻击状态’为假 |
| 输入参数 | self:类对象引用； |
| 返回结果 | 无 |

### underAttack(self,collisionEntry)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
| CollisionEntry |  | 主角（攻击者） | |
| **返回值类型** | | | **返回值含义** |
|  | | |  |
| **函数说明** | 处理怪物受到攻击（近战或远程）事件   1. 判断 “是否处于受攻击状态” 是否为真，如果为真，直接返回； 2. 根据攻击者的attackPower决定怪物受到的伤害值，相应地减少怪物的当前生命值； 3. 设置 “怪物是否处于受攻击状态” 为真； 4. 向房间发送”子弹攻击“事件（以销毁子弹）   如果普通怪物当前生命值小于等于0，播放”怪物死亡“动画，完毕后，向房间发送”怪物死亡“事件；而特定怪物当前生命值小于等于0，可以生成两个相同类型的普通怪物，并向房间发送”普通怪物生成“事件 | | |
| **备注** | | |  |

# MapManager类

## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| index | int | 当前的关卡编号：1-n |  |
| rooms | list<room> | 某一关卡的房间列表 |  |

## 成员函数

### genRoom(self)

|  |  |  |  |
| --- | --- | --- | --- |
| **参数名称** | **参数类型** | **参数含义** | |
|  |  |  | |
| **返回值类型** | | | **返回值含义** |
| List<Room> | | | 房间列表 |
| **函数说明** | | | 根据关卡编号，生成房间布局，并确定房间属性（不载入模型）、怪物分布、道具分布、主角属性等信息 |
| **备注** | | |  |

# Menu类

## 数据成员

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **类型** | **说明** | **备注** |
| btn\_newgame | DirectButton | 新游戏按钮 |  |
| btn\_loadgame | DirectButton | 加载游戏按钮 |  |
| btn\_option | DirectButton | 设置属性按钮 |  |
| btn\_quit | DirectButton | 退出按钮 |  |
| frameBegin | DirectFrame | 主界面frame |  |
| title | DirectLabel | 主界面标题label |  |

## 成员函数

### CreateButton()

|  |  |
| --- | --- |
| 函数签名 | createButton(self,text, verticalPos, eventArgs) |
| 函数功能 | 创建不同的按钮，并添加到frameBegin上 |
| 输入参数 | text:按钮文字,verticalPos:按钮部分坐标,eventArgs:时间传递参数 |
| 返回结果 | 无 |

### Show(self)

|  |  |
| --- | --- |
| 函数签名 | show(self) |
| 函数功能 | 展示整个界面 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### Hide(self)

|  |  |
| --- | --- |
| 函数签名 | hide(self) |
| 函数功能 | 隐藏整个界面 |
| 输入参数 | 无 |
| 返回结果 | 无 |

# Archive类

## 数据成员

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| slist\_file | DirectScrolledList | 存档列表 |
| btn\_load | DirectButton | 加载当前的存档 |
| btn\_delete | DirectButton | 删除当前的存档 |
| btn\_save | DirectButton | 保存存档 |
| btn\_bcakbegin | DirectButton | 返回开始界面按钮 |
| frameArchive | DirectFrame | 存档界面frame |

## 成员函数

### SaveArchive(self,archiveID)

|  |  |
| --- | --- |
| 函数签名 | SaveArchive(self,ArchiveID) |
| 函数功能 | 1. 修改存档界面上的存档信息 2. 将存档修改的信息发送给Main，让对应的函数处理存档文件的修改 |
| 输入参数 | ArchiveID:用户选择的存档id |
| 返回结果 | 无 |

### deleteArchieve(self,archieveID)

|  |  |
| --- | --- |
| 函数签名 | DeleteArchive(self,ArchiveID) |
| 函数功能 | 1. 删除存档界面上对应的存档信息   2、将存档删除的信息发送给Main，让对应的函数删除对应id的存档文件内容 |
| 输入参数 | ArchiveID:用户选择的存档id |
| 返回结果 | 无 |

# 设置界面 Option

## 数据成员

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| btn\_bcakbegin | DirectButton | 返回开始界面按钮 |
| slider\_volume | DirectSlider | 声音音量调节滑条 |
| slider\_sound | DirectSlider | 音效音量调节滑条 |
| frameOption | DirectFrame | 设置界面frame |

**游**戏结束界面GameOver

## 数据成员

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| btn\_bcakbegin | DirectButton | 返回开始界面按钮 |
| btn\_retry | DirectButton | 重新开始按钮 |
| title | DirectLabel | 游戏胜利或者失败 |
| frameGameOver | DirectFrame | 游戏结束界面frame |

# 游戏主界面 Game

## 数据成员

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| frame\_map | DirectFrame | 小地图的frame |
| frame\_msg | DirectFrame | 血条以及个人信息的frame |
| frameGame | DirectFrame | 游戏主要界面frame |

**暂停界面：Pause**

## 数据成员

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| btn\_continue | DirectButton | 继续游戏按钮 |
| btn\_loadgame | DirectButton | 加载游戏按钮 |
| btn\_backbegin | DirectButton | 返回主菜单 |
| btn\_option | DirectButton | 设置属性按钮 |
| framePause | DirectFrame | 暂停界面Frame |

## 数据成员

### createButton(self,text, verticalPos, eventArgs)

|  |  |
| --- | --- |
| 函数签名 | createButton(self,text, verticalPos, eventArgs) |
| 函数功能 | 创建不同的按钮，并添加到frameBegin上 |
| 输入参数 | text:按钮文字,verticalPos:按钮部分坐标,eventArgs:时间传递参数 |
| 返回结果 | 无 |

### Show(self)

|  |  |
| --- | --- |
| 函数签名 | show(self) |
| 函数功能 | 展示整个界面 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### Hide(self)

|  |  |
| --- | --- |
| 函数签名 | hide(self) |
| 函数功能 | 隐藏整个界面 |
| 输入参数 | 无 |
| 返回结果 | 无 |

# Main

## 数据成员

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| menu | Menu | 主菜单对象 |
| archive | Archive | 存档界面对象 |
| option | Option | 设置界面对象 |
| pause | Pause | 暂停界面对象 |
| gameover | GameOver | 游戏结束界面对象 |

**Menu Option Archive Game Pause GameOver ChangeMusic SaveArchive DeleteArchive LoadArchive quit**

## 函数：

### ChangeMusic (self)

|  |  |
| --- | --- |
| 函数签名 | ChangeMusic (self) |
| 函数功能 | 音量设置函数的回掉函数，根据音量的设置改变背景音乐或者音效的音量 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### SaveArchive (self,ArchiveID)

|  |  |
| --- | --- |
| 函数签名 | SaveArchive (self,ArchiveID) |
| 函数功能 | 保存存档按钮的回掉函数。保存当前存档到对应的存档文件中。 |
| 输入参数 | ArchiveID：存档文件id |
| 返回结果 | 无 |

### DeleteArchive (self,ArchiveID)

|  |  |
| --- | --- |
| 函数签名 | DeleteArchive (self,ArchiveID) |
| 函数功能 | 删除存档按钮的回掉函数。删除对应存档id的存档。 |
| 输入参数 | ArchiveID：存档文件id |
| 返回结果 | 无 |

### LoadArchive (self,ArchiveID)

|  |  |
| --- | --- |
| 函数签名 | LoadArchive (self,ArchiveID) |
| 函数功能 | 加载存档按钮的回掉函数。加载对应存档id的存档文件。 |
| 输入参数 | ArchiveID：存档文件id |
| 返回结果 | 无 |

### quit(self)

|  |  |
| --- | --- |
| 函数签名 | quit(self) |
| 函数功能 | 退出整个游戏 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### enterMenu(self)

|  |  |
| --- | --- |
| 函数签名 | enterMenu(self) |
| 函数功能 | 进入游戏目录界面函数。监听目录中各种按钮的点击事件，做出相应的反应，显示目录 |
| 输入参数 | 无 |
| 返回结果 | 无 |

|  |  |
| --- | --- |
| 函数签名 | exitMenu (self) |
| 函数功能 | 退出游戏目录界面函数。撤销目录中按钮的监听事件，隐藏目录 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### enterArchive(self)

|  |  |
| --- | --- |
| 函数签名 | enterArchive(self) |
| 函数功能 | 进入保存存档界面函数。监听按钮的点击，从而匹配正确的回调函数。显示存档界面 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### exitArchive(self)

|  |  |
| --- | --- |
| 函数签名 | exitArchive(self) |
| 函数功能 | 退出保存存档界面函数。撤销按钮的监听事件。隐藏存档界面。 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### enterOption(self)

|  |  |
| --- | --- |
| 函数签名 | enterOption(self) |
| 函数功能 | 进入设置界面函数。监听返回按钮，以及两个音量滑条的改变。显示设置界面。 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### exitOption(self)

|  |  |
| --- | --- |
| 函数签名 | exitOption(self) |
| 函数功能 | 退出设置界面函数。撤销小部件的监听事件。隐藏界面。 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### enterGame(self)

|  |  |
| --- | --- |
| 函数签名 | enterGame(self) |
| 函数功能 | 进入游戏主界面函数。通过gameController初始化游戏的各种信息。显示游戏主界面 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### exitGame(self)

|  |  |
| --- | --- |
| 函数签名 | exitGame(self) |
| 函数功能 | 退出游戏主界面函数。隐藏游戏主界面。 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### enterPause(self)

|  |  |
| --- | --- |
| 函数签名 | enterPause(self) |
| 函数功能 | 进入游戏暂停界面。监听按钮的点击事件，对应到相应的处理函数。显示游戏暂停界面。 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### exitPause(self)

|  |  |
| --- | --- |
| 函数签名 | exitPause(self) |
| 函数功能 | 退出游戏暂停界面。撤销监听按钮的点击事件。隐藏游戏暂停界面。 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### enterGameOver(self)

|  |  |
| --- | --- |
| 函数签名 | enterGameOver(self) |
| 函数功能 | 进入游戏结束界面。监听两个按钮的点击事件。显示游戏成功或者失败状态。显示游戏结束界面。 |
| 输入参数 | 无 |
| 返回结果 | 无 |

### exitGameOver(self)

|  |  |
| --- | --- |
| 函数签名 | exitGameOver(self) |
| 函数功能 | 退出游戏结束界面。撤销按钮的点击事件。隐藏游戏结束界面。 |
| 输入参数 | 无 |
| 返回结果 | 无 |