**UpGame框架**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 创建日期 | 内容 | 修订人 |
| V0.0.1 | 2014-10-17 | 创建文档 | 王加静 |
|  |  |  |  |
|  |  |  |  |

文档目录

[1 介绍 2](#_Toc471726718)

[2 项目文件夹 2](#_Toc471726719)

[2.1 主文件夹 2](#_Toc471726720)

[2.2 每个版本文件夹 3](#_Toc471726721)

[2.3 公用接口文件 3](#_Toc471726722)

[2.4 Tools静态工具 3](#_Toc471726723)

[3 引擎与显示的关联 4](#_Toc471726724)

[3.1 ED对象的添加 4](#_Toc471726725)

[3.2 ED对象的移除 4](#_Toc471726726)

[3.3 EDBullet子弹 4](#_Toc471726727)

[3.4 EDRole角色 4](#_Toc471726728)

[3.5 EDRole角色血条 5](#_Toc471726729)

[3.5.1 类型 : 0 不显示血条 5](#_Toc471726730)

[3.5.2 类型 : 1 被攻击时显示 5](#_Toc471726731)

[3.5.3 类型 : 2 总是显示 5](#_Toc471726732)

[3.5.4 类型 : 3 固定显示在屏幕最顶端 5](#_Toc471726733)

[3.5.5 类型 : 4 (始终显示等级)受伤后永久显示血条,用做末日冲突的一般角色 5](#_Toc471726734)

[3.5.6 类型 : 5 (显示等级)激活后永久显示血条,总部老家,显示血量 6](#_Toc471726735)

[3.5.7 类型 : 6 (显示等级)总是显示血条,其他建筑 7](#_Toc471726736)

[3.5.8 类型 : 7 (不显示等级,但显示图标)总是显示,并显示血量,箭塔 7](#_Toc471726737)

[3.6 EDSkill技能 7](#_Toc471726738)

[4 层级管理方法 8](#_Toc471726739)

[5 显示对象的显示和隐藏 9](#_Toc471726740)

[5.1 设计要求 9](#_Toc471726741)

[5.2 设计方法 9](#_Toc471726742)

[5.3 数据设计 9](#_Toc471726743)

[5.4 影子显示 10](#_Toc471726744)

[5.5 分层特效设计 10](#_Toc471726745)

[5.5.1 技能特效存放目录和命名规范 11](#_Toc471726746)

[5.5.2 技能特效的制作方式 12](#_Toc471726747)

[5.6 震动事件桥 14](#_Toc471726748)

[6 角色AI控制 14](#_Toc471726749)

[6.1 用户介入操作 14](#_Toc471726750)

[6.1.1 控制人物走动 15](#_Toc471726751)

[6.1.2 控制人物攻击特定怪 15](#_Toc471726752)

[6.1.3 控制人物释放技能 15](#_Toc471726753)

[6.2 仇恨形基础AI 15](#_Toc471726754)

[6.3 顽固AI 15](#_Toc471726755)

[6.4 随机AI 15](#_Toc471726756)

[6.5 逃跑AI 16](#_Toc471726757)

[6.6 BOSS AI 16](#_Toc471726758)

[7 战斗的逻辑 16](#_Toc471726759)

[7.1 角色逻辑 16](#_Toc471726760)

[7.2 技能目标对象逻辑 21](#_Toc471726761)

[7.3 地面持续特效提供BUFF支持 22](#_Toc471726762)

[8 战斗数据的记录 22](#_Toc471726763)

[9 PK模式 22](#_Toc471726764)

[10 英雄系统 23](#_Toc471726765)

[10.1 启动技能 23](#_Toc471726766)

[10.2 升星 23](#_Toc471726767)

[10.3 升星经验 24](#_Toc471726768)

[10.4 特殊属性 24](#_Toc471726769)

[11 多人联网 25](#_Toc471726770)

[11.1 能量系统 25](#_Toc471726771)

[12 不同版本串联 25](#_Toc471726772)

[13 人工智能(机器人) 25](#_Toc471726773)

[14 版本V1 25](#_Toc471726774)

[14.1 V1版本 26](#_Toc471726775)

[14.2 V1 BUG收集 26](#_Toc471726776)

# 介绍

UpGame是一套向上运行的游戏架构

竖版的游戏提供框架.开发工具融入GaGaFrameForDevelopMax工具中.

战斗与数据一起处理,这样渲染可以使用各种类型的渲染模式,方便在CPU和GPU,以及网页等各个模式下的切换.

地图文件的扩展名 .upgame

# 项目文件夹

不会有多个模块的战斗出现,所以加入公共类中

* 提供版本管理,不通的版本可以提供不同的功能,使用接口串接起来
* 外面使用接口来管理
* 关于不同接口直接的管理问题

## 主文件夹

|  |  |
| --- | --- |
| **名称** | **说明** |
| **common** | 接口文件夹 |
| **V1** | 第一个版本 |

## 每个版本文件夹

|  |  |
| --- | --- |
| **名称** | **说明** |
| **engine** | 模型数据处理 |
| **Info** | 数据模型层 |
| **render** | 渲染控制层 |
| **tools** | 静态工具层 |

## 公用接口文件

这里的接口是每个项目的接口

|  |  |
| --- | --- |
| **名称** | **说明** |
|  |  |
|  |  |
|  |  |
|  |  |

## Tools静态工具

常用的一些静态方法

|  |  |
| --- | --- |
| **名称** | **说明** |
|  |  |
|  |  |
|  |  |
|  |  |

# 引擎与显示的关联

按照种类来分类。

EDRole角色根据技能创建EDBullet或者是EDSkill对象

EDBullet子弹上附带信息可以创建EDSkill对象

最后都由EDSkill里有释放特效的内容

EDRole ->DisplayEDRole [1对1对象, 多面由一个对象完成装载]

EDBullet -> U2 / DisplayEDBitmap [1对多对象]

EDSkill -> DisplayEDSkill [1对多对象 , 非DisplayObject]

EDSkillEnd -> 后续的内容

显示对象的移除

## ED对象的添加

当显示列表找不到对应的ED对象的时候，自动添加对应的ED显示对象．

## ED对象的移除

在驱动层移除ED对象的时候，查看是否打开显示对象驱动，如果是打开，也移除显示对象驱动层内的内容．

还有一种对象是虽然移除ED,但是显示层还是要播放的

## EDBullet子弹

子弹的ED对象,显示驱动,根据ED对象内的信息来控制界面上的内容

伤害数字的控制也放在子弹上

显示对象是多个,每个子弹多个显示对象,对影子进行处理

## EDRole角色

六面和单面使用一个ED驱动完成

BUFF的控制

在EDRole中加入EDBuff队列，Buff的对象和显示不是完成挂钩的，２个中毒只显示一个效果，进行计数处理，跟随目标特效都会置入目标（分层后只有人物层会这样做）

人物U2路径,方向,镜像存储

添加行为开始时间

人物行走,记录开始时间,并且根据速度,对时间进行叠加,将时间推送进U2模型,来完成人物行走的加速行为.

技能播放,记录开始时间,当释放技能转向的时候不会重置时间,保持各个方向的释放技能动作连贯.

显示对象装载进容器, 角色也可以根据数据来完成对显示对象的控制,方向和坐标直接控制

血条对象,根据值完成.显示对象管理

## EDRole角色血条

人物的血条分很多种，这里对血条配置和样式进行整理。

血条由二个地方控制，模型表**（UpGameModelInfo）**和怪物表，人物先从模型表里取，如果怪物表中也出现，将取怪物表中覆盖模型表中。

### 类型 : 0 不显示血条

始终不显示血条

### 类型 : 1 被攻击时显示

默认不显示，被攻击的时候显示，未被攻击后切换为不显示

### 类型 : 2 总是显示

初始化完成后一直显示血条

### 类型 : 3 固定显示在屏幕最顶端

这个是BOSS血条

工具upGame.engine.methodTopBoolmd函数,如果有配置将EDRole传递到函数里进行血条显示,如果没有配置将使用总是显示

### 类型 : 4 (始终显示等级)受伤后永久显示血条,用做末日冲突的一般角色

使用**DisplayEDRoleBoolmd**类进行配置

type\_4567\_white : 闪光的白色条

当角色初始化后,将显示出角色的等级,当角色被攻击后将显示出角色的血条.而且是永久显示,当角色被攻击,将血条上覆盖白色条,渐变成无色后移除白色,是一种特效.

默认出来的角色,脑袋上先显示出等级

当失血后将持续显示出血条

### 类型 : 5 (显示等级)激活后永久显示血条,总部老家,显示血量

背景使用新的一套资源,条上的资源使用类型4的资源

这个是BOSS老家的血条,只有老家是这样显示的,出血条的时候是动画

默认情况下老家是显示出等级的,并没有播放出场动画

被攻击后,或激活后,将显示出血条,并且显示出血量,那个留白是没有的

### 类型 : 6 (显示等级)总是显示血条,其他建筑

和类型4的显示出全部的效果相同

一出来就显示出等级,并且显示出血条

### 类型 : 7 (不显示等级,但显示图标)总是显示,并显示血量,箭塔

一出来就显示成这个样子

## EDSkill技能

技能发射前的一个控制, 还需要一个技能发射后的状态(Buff, 技能后续特效,技能控制)等等控制

技能启动 -> 前摇 -> 释放 -> 结束 -> 改变状态为:待机 -> 有目标,继续对目标处理

技能释放过程中 -> 用户释放其他技能 -> 打断前一个技能

技能释放过程中 -> 用户行走行为 -> 打断当前技能

击中特效的控制

特效

释放 -> 释放特效释放者的行为动作(冲锋,位移形,跳) , U2释放特效的特效效果绑定U2

飞出 -> 有子弹就飞出

击中 -> 燃烧弹 -> 燃烧弹 击中 ?

普通 -> 对人物播放击中特效 (特效位置,固定,还是和首创者位置绑定)

持续产生效果的作用 -> 击中特效 , 击中后在固定地方释放持续效果 , 消失的时候如何消失. 持续的时间 , 我持续效果 被击中后什么情况.被击中后的行为,还有特效

产生状态的作用

特效击中行为, upgame\_skill\_action,中动作类型体现,冲锋的话进行约定

持续伤害特效的击中特效 (燃烧弹击中)

击中特效的播放位置 (固定, 还是跟谁),

修正比例 (击中特效根据人物比例缩放比例)

受创动作,受创行为

技能分为人物方和非人物方(燃烧弹),Buff特效

EDRole技能,需要控制人物播放帧,就是人物前摇,释放等动作.

释放出去的技能转换为特效释放来展示

技能的控制,技能里有前摇,这个的控制如何完成

那么技能释放点和释放时间可以在这个地方算出来.

可以存放持续的特效,瞬发特效,Buff特效

# 层级管理方法

先使用了简单的图层控制.

通过一个二个值做管理的层级管理方法,首先将对象置入对象里,对象就有一个index

当值发生变化,找出要去的位置,那么从旧位置到新位置之间的值都加1,或减1,

# 显示对象的显示和隐藏

## 设计要求

* 界面上内容要显示,没在界面上的内容不显示
* 不能对整个列表进行遍历,会发生性能的过度消耗
* 整个处理要小于1%的CPU消耗

## 设计方法

* 每次多计算30个像素,这样30像素范围内的移动就直接不处理.
* 要把多计算的部分外加上震屏的范围
* 由于震动是一个奇怪的参数,所以现在的震动使用偏移量来计算,这样可以保存震动的时候不会丢失原始X,Y坐标属性,所以震动只可以在图层和主容器上实现,U2里已有偏移坐标

## 数据设计

整个的数据设计在编辑器里就处理掉.

gameStage场景范围:现在的整个表现所在的范围

displayStage显示范围:现在这个显示范围内的内容都要显示出来,

震动范围,x,y2个坐标的震动范围.

叠加优化像素处理 x, y;

displayStage = gameStage.x + 震动范围X + 叠加优化像素处理X;

触发计算的 gameStage +叠加优化像素处理X

上次计算的场景X坐标 正负 叠加像素处理就不处理

记录现在的上下最大的100单位的位置.移动的时候先判断这个有没有发生变化

只考虑这100单位内的显示和隐藏

以100为单位 [然后全部的内容放到这个内容中]

5,6,7,8,9 查看这些范围内是否还在,只有2头发生变化的才处理,屏幕大范围移动的时候,老的范围内容肯定都没有了,直接就干了(如果全干属于优化算法)

a[50, 60] , b[42, 60], c[42, 60] 要考虑到高度

把受影响的范围内的都加入到这个内,这样有重复的不管

然后在专门对这个内的内容进行处理

## 影子显示

影子放在**assets\model\img\shadow\{ID}.png**

影子放在这个文件夹下,当ID为0的时候,不会显示影子.

影子的比例 : **UpGameModelInfo.shadowScaleX**, **UpGameModelInfo.shadowScaleY**

影子放置的图层 : **(floorEffect)地板特效图层,地板燃烧**

## 分层特效设计

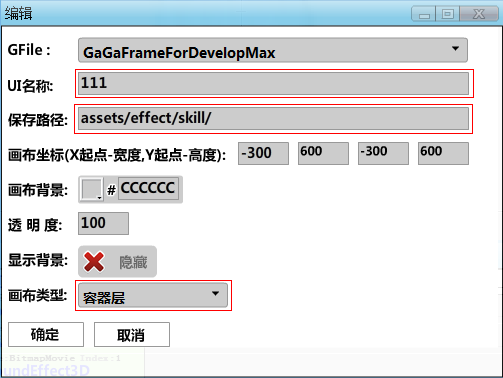
　　为达到战斗相关特效在不同图层播放，而不用程序和策划介入，所以设定了规则，首先把不同层级的特效使用Ｕ２制作出来，并使这些特效在展示的时间上保持合理的长度与开始时间，比如延后一些的特效前面可以加入空白帧，然后建立Ｕ２的容器层，将各个特效分别放置在特定的Ｕ２容器层，将这个层起名为下列名称。

* 只有下列名称的图层会被播放
* 图层的层级先后循序将和下列图层排列循序相关
* 下列图层靠下的图层排序低，靠上的图层排序高
* 所有的图层动画将只播放一次就停止播放，并会被移除，所以每个图层的Ｕ２不能为纯图层，需要为序列帧内容

|  |  |
| --- | --- |
| **名称** | **说明** |
| **flyHarm** | 空中部队伤害数字图层 |
| **flyEffectTop** | 空中部队头顶特效图层 |
| **fly** | 空中部队 |
| **flyEffectEnd** | 空中部队脚下特效图层 |
| **groundHarm** | 地面部队伤害数字图层 |
| **groundEffectTop** | 地面部队头顶特效图层 |
| **ground** | 地面部队 |
| **groundEffect3D** | 地面部队的有立体感的特效层级 |
| **ground3DItem** | 地面部队掉落装饰物 |
| **groundEffectEnd** | 地面部队脚下特效 |
| **floorEffect** | 地板特效图层,地板燃烧 |

### 技能特效存放目录和命名规范

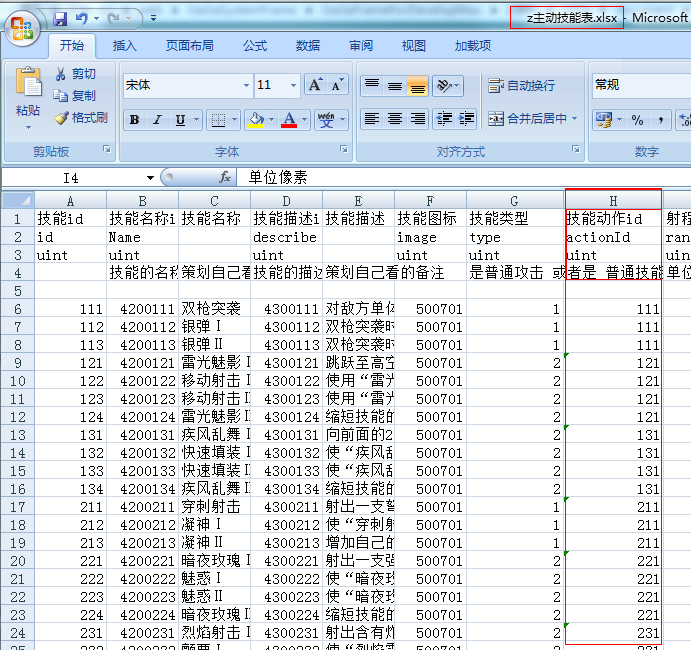
技能特效的存放目录是GFile目录下的：assets/effect/skill/



文件名使用策划的文档

E:\web\phoenix\zombie\cehua\13战斗系统（战斗规则，主动技能释放）\ z主动技能表.xlsx

其中有技能动作ID，文件名和这个技能动作ID挂钩，保存就OK了。



### 技能特效的制作方式

图解:



　　可以将模型导入进行模拟特效运行，由于“模型”名称图层未在特定名图层，所以正常游戏中是看不到“模型”图层。

　　特效的透明度,角度,缩放比例,和位置偏移是有效果的。

击中特效与子弹落地特效为单次播放特效，单次播放特效，需要将选中的U2文件的FPS配合出正确的时间周期.

　　BUFF的特效击中特效的坐标点在脚底.

　　击中特效，击中效果在人物的碰撞点,就是人物.

## 震动事件桥

事件桥:战斗.场景震动

(uint毫秒) 震动时间

(uint) 震动类型, map整张地图震动, child地图内图层分别震动(ShockType)

(uint) 延迟时间

(Number) 震动的比例

(Number) 震动的比例

震动类型

0 : 全地图震动

1 : 全地图震动

2 : 地图内的层

3 : 层和地图一起震

震动的比例(参考值)

小震动 0.5

中震动 0.8

大震动 1

# 角色AI控制

仇恨互助 : 不同种类的怪物仇恨无法复制,

没有怪物的行动方向，一直向前走，地图可以向回走

先杀掉后面的怪物

过关条件，怪物全死掉，主怪挂掉，其他挂掉

屏幕地图上有固定范围内

A\* 是有A\*的范围,在范围内在制作A\*移动

## 用户介入操作

　　这里是说用户关闭了自动战斗，使用自己控制角色。助战好友是不能控制的。

* 介入后，关闭角色自动行走，角色不被控制时一直站原地。
* 介入后，关闭角色大技能自动释放功能。
* 介入后，角色对进入视野范围内目标还会继续功能

### 控制人物走动

人物走动中，不会产生任何的技能释放行为，仇恨应该是需要建立的。

人物到达目标后，进行搜索杀怪。

### 控制人物攻击特定怪

选中怪后判断是否进入攻击范围，进入攻击范围进行杀怪。

当怪物死后，进入自动搜索杀怪，但是不会主动移动。

### 控制人物释放技能

选中技能后，当拖到适当位置，或者释放，角色会打断当前行为，进行技能的释放，当技能释放完毕后会进入自动搜索杀怪。

## 仇恨形基础AI

最普通的怪物,或则人物一般是用这个仇恨系统

技能释放：满足技能释放条件（cd达到）后立刻释放技能，当有技能释放中就等待

【仇 恨 值】：进入视野，被攻击，都会增加一定的仇恨值；

【仇恨列表】：按照仇恨值的大小进行排序的列表，在没有干预的情况下，ai会自动攻击仇恨列表中最靠前的敌人；

【仇恨上限】：在列表中的仇恨值都会有一个上限；

【仇恨降低】：仇恨值会随着时间自动降低

## 顽固AI

通常用于攻击后排的怪物

基础目标选择：初始化目标时，随机选择一个单位，只会选择一次，目标确定后将不会更换；

有普通的仇恨列表，按顺序排列的仇恨列表

技能释放：满足技能释放条件（cd达到）后立刻释放技能，当有技能释放中就等待

## 随机AI

用于超远程的怪物

基础目标选择：每隔一段时间（10秒）段就会重新选择完全随机选择一次目标；

技能释放：满足技能释放条件（cd达到）后立刻释放技能，当有技能释放中就等待

## 逃跑AI

血量低于20%时会逃走一定距离,生命值回复80%后，将再次锁定仇恨列表顶端角色攻击

## BOSS AI

BOSS AI 会根据血量的变化释放特殊技能的ai

技能释放：满足技能释放条件（cd达到且需要释放，主要是躲避的技能，会在将要被一个延迟性的攻击技能击中前释放，而后进行躲避）后立刻释放技能；但是同时会遵循下面的血量条件进行技能的释放

技能目标的选择：根据治疗技能的目标范围进行目标选择，（例如随机范围的技能，就会随机释放到以几个点为中心的范围上）；

血量>50%时会释放技能1和2

血量<50%时会释放技能1,2,和3（如果可以释放）

血量<10%时候会释放技能1,2,3,4（如果可以释放）

# 战斗的逻辑

## 角色逻辑

角色技能逻辑

主动技能挂射程,需要保证这个主动技能外的动作中的作用效果都用这个射程

1. **准备和释放技能  
   RoleCardModel(角色卡牌表) ------(Cnskill/CHskill/CHskill2 == id)-----> SkillActiveModel(主动技能表) ----------(actionId == id)---------> UpGameSkillAction(主动技能动作表) ----–(time1skill/time2skill/time3skill/time4skill/time5skill == id 多对多)----> UpGameSkill(技能效果表)**
2. **发出技能行为  
   UpGameSkill(技能效果表)[id相同的条数为多条]  
     
   当bulletId 为0  
   相同bulletId效果effectId收集在一起,用bulletId == UpGameBulletInfo.id释放子弹,并挂入收集的effectId集合给对应子弹  
     
   当bulletId 不为0  
   立即释放effectId的效果**
3. **效果的释放effectId  
   UpGameSkillEffect(技能作用效果表)--------------(effectId == id)-------------->技能效果释放中**
4. **技能效果的种类  
   (这个有问题在说)**

|  |  |
| --- | --- |
| **名称** | **说明** |
| **RoleCardModel jjuesekapai** | 卡牌角色策划原始数据  cardRole中获取 |
| **UpGameModelInfo upgame\_model\_info** | 模型数据表  upgameModelInfo中获取，通过RoleCardModel.Cmodel获取  攻击发射点数据 |
| **SkillActiveModel zhudongjineng** | 主动技能数据表  skillActive中获取，通过cardRole. Cnskill/CHskill/CHskill2获取   * 寻找目标 : 射程,目标类型,阵营作用效果 * 目标数量 : 单个目标, 多目标,比如3目标治疗 * 释放时机 : 技能类型,走CD或攻速,CD,CD初始值 * 释放长度 : 循环次数,循环总时间 * 狂点释放 : 是否是狂点技,点击开始,点击总时,点击最大次数   多目标 -> 单弹头 -> 变为多弹头  多目标 -> 多弹头 -> 变为超多弹头  预警提示 |
| **UpGameSkillAction upgame\_skill\_action** | 主动技能动作表  upgameSkillAction中获取，通过SkillActiveModel.actionId获取   * 模型展示 : 攻击者形象，角色ID,动作类型,动作类型有跳等，程序处理 * 循环播放 : 循环播放区间开始和结束时间 * 组合技能 : 按循序播放1,2,3,4,5时间点和技能 * 单次技能 : 直接使用1的技能   释放技能的坐标点,震屏，形象在U2中, 额外动作通过模型展示  冲锋使用走路加速 |
| **UpGameSkill upgame\_skill** | 主动技能信息连接表  upgameSkill中获取,通过SkillActiveModel.time1bullet获取   * 是否释放子弹 : 使用子弹载体运送技能效果   子弹淡出,增加显示出来的透明度,起始透明度,完全显示移动距离  SkillActiveModel.time1bullet 会释放一裤衩的内容,需要做整合 |
| **UpGameBulletInfo upgame\_bullet\_info** | 子弹数据表   * 子弹数量 : 子弹类型,散射角度,发射数量 * 子弹外观 : 特效类型,特效是否需要旋转 * 子弹速度 : 子弹最大速度,启动速度,加速度 * 子弹弹道 : 子弹弹道   upgameBulletInfo中获取，通过UpGameSkill.bulletId获取  标识子弹的弹道，外观，速度，是否散弹，散弹信息 |
| **UpGameSkillEffect upgame\_skill\_effect** | 技能作用效果表  upgameSkillEffect中获取，通过UpGameSkill.effectId获取  可以通过2个地方带入公式,完成优化，只有里面的合成值会发生变化，而且会有Copy份，当引用全部用掉，Copy也不会在存在   * 范围判定 : 范围类型，范围半径 * 持续时间 : 作用类型，作用次数，作用时间 * 普通特效: 击中特效，特效是否跟随目标(欠缺)，特效根据目标横竖缩放比例(欠缺) 持续特效 : 持续特效击中特效(欠缺)，持续特效击中特效是否跟随目标(欠缺)，持续特效根据目标模型横竖缩放比例(欠缺) **特效播放效果** **Buff效果** : Buff效果都为置入对象(多个Buff显示一个特效)，显示周期为播放完毕 **持续效果** : 持续效果播放的时间也持续时间挂钩 * 控制Buff : 删除目标Buff,删除全部BUff (控制Buff叠加,替换) * 伤害计算 : 计算命中 -> 命中后计算是否爆击 -> 计算爆击伤害 -> 合成伤害 * 命中计算 : 通过命中公式，命中初始值，命中增量 (命中传递量) * 爆击公式 : 通过爆击公式, 爆击一系列值进行计算   对自己(所有的效果对自己),对其他  有效阵营,敌方,我方,全体  有子弹(敌方,碰敌人子弹生效,我方,碰我方子弹生效,全体,随便碰个就生效)  标识释放某一种特定的效果，伤害值，命中值，爆击值  标识出这个效果的作用范围（地图距离上的）  传递公式作用，将作用效果的公式直接带入首创者对象上，结合首创者属性进行计算。  通过作用类型，来完成下面３中类型的效果  即刻效果：（立即完成计算，然后可以丢掉）  持续效果：（推进作用对象列表，持续作用）   * 作用间隔 * 生效次数   Buff效果：（推进作用对象列表，持续作用）   * 作用间隔   在持续效果和Buff效果里，还有中是在对象以外发生持续作用  人物光环，人本身就是一个作用发射器，提供范围内持续作用，需要推入对象，管理范围内对象有无一直在范围内 |

|  |  |
| --- | --- |
| **名称** | **说明** |
| **初始化ED对象** | 使用卡牌的ID号来初始化出一个ED对象的模型   * 模型初始化 * 仇恨列表初始化 * ＡＩ初始化 * 技能初始化 |
| **创建技能发生器** | 由这个来控制发射子弹等动作 |
| **技能创建器** | 无载体：分为普通技能发生器，瞬发技能发生器  有载体：子弹技能发生器 |
| **AI查找目标** | 使用自己的一套AI系统来找到目标 |
| **AI攻击** | 使用技能机器人发子弹或者技能 |
| **子弹攻击** | 子弹绑定效果对象，飞出 |
| **子弹击中** | 子弹销毁，特效继续播放不受控制  效果对象启动   * 即刻效果，立即释放效果，销毁自己，释放界面特效。 UI：效果对象将播放次数推出就可以了 * 即刻效果穿透，立即释放效果，特效，并保存已经被攻击对象，继续飞行直到射程结束，摧毁自己 * 持续效果，将在阵营中建立独立的效果对象列表，这个将在运行扣血动作后执行。 UI：效果对象将循环多少时间推出就可以了 * Buff特效，效果公式创建一个AIBuff的对象，这个对象自动启动开始作用 UI：AIBuff对象控制UI的播放   释放并控制主效果特效  将效果对象绑入目标对象内，或者这个效果对象直接丢入显示界面上。 |
| **效果完毕** | 效果的特效自动结束移除。  但是效果作用还未完成，继续完成效果特效，发出要清理效果界面内容的内容列表。  将自己清理 |
| **UpGameBulletInfo** | 子弹数据表  upgameBulletInfo中获取，通过UpGameSkill.bulletId获取  标识子弹的弹道，外观，速度，是否散弹，散弹信息 |
| **UpGameSkillEffect** | 技能作用效果表  upgameSkillEffect中获取，通过UpGameSkill.effectId获取  标识释放某一种特定的效果，伤害值，命中值，爆击值  标识出这个效果的作用范围（地图距离上的）  传递公式作用，将作用效果的公式直接带入首创者对象上，结合首创者属性进行计算。  通过作用类型，来完成下面３中类型的效果  即刻效果：（立即完成计算，然后可以丢掉）  持续效果：（推进作用对象列表，持续作用）   * 作用间隔 * 生效次数   Buff效果：（推进作用对象列表，持续作用）   * 作用间隔   在持续效果和Buff效果里，还有中是在对象以外发生持续作用  人物光环，人本身就是一个作用发射器，提供范围内持续作用，需要推入对象，管理范围内对象有无一直在范围内 |

如果射程,阵营,目标类型,CD合适,就释放主动技能(寻找目标)

在主动技能表中找到 - > 主动技能动作表

设置模型,和模型的动作类型(比如技能1,跳什么的)

如果可以找到模型,设置模型

循环技能 -> 刀塔(凤凰),持续释放某一个特定技能

组合动作 -> 刀塔(熊猫),一段时间内释放组合技能

等待顺序的子弹发出时间点.

根据子弹发出的循序找到技能效果表ID

特定时间,发出特定的效果表ID

效果载体

子弹(根据击中来完成效果释放)

瞬发,马上进入效果释放

效果类型

技能中心点效果, 对自己生效

技能作用效果id

//-------------------------------------------------------------------------------------------------------------------------

技能的表现

技能的周期控制

技能的效果控制

效果的抽象：

效果的发出者，挂入效果抽象对象中，而不是子弹和武器上。

效果的生命周期，和效果自身有关系，而和子弹的生命周期无关。

几种常见武器的解决方案

子弹：子弹发出，带出效果，攻击到目标，子弹播放爆炸特效，效果挂入对象。

燃烧弹：丢出，带出效果，攻击到目标，燃烧弹爆炸特效，效果要控制燃烧弹的效果？

效果：要控制效果的特效。因为效果特效的存在是因为效果的存在。效果如何挂住一裤衩的显示对象。

Dict[效果对象] = List<显示对象> / 显示对象

效果　－》　效果对象　－》　显示驱动层　－》　一裤衩的显示对象列表

效果　－》　效果对象　－》　对显示对象透明度，坐标控制代码　－》　整体修正一裤衩的显示对象内容

效果消失　－》　效果对象　－》　销毁显示对象　－》　得到命令播放特效自动销毁特效（完毕后自动移除）并清理掉

Ｕ２特效的ＦＰＳ和游戏的ＦＰＳ挂钩

Ｕ２的ＦＰＳ时间控制抽象出来，将由一个时间控制器整体管理。

抽象出来时间控制器，可以在各个地方应用

游戏结束的时候游戏的时间控制器停止，Ｕ２时间控制器自动启用。

时间控制器可以整体提供变速，停止的功能。

时间控制器提供移动动画的支持。将几个都互相串联起来

U2的Stage3D驱动

Buff命中的同时生效,即使生效

近战,特效技能

## 技能目标对象逻辑

本属性只针对非普通技能,普通技能获取目标用专属方式。

特殊技能不使用本逻辑，在**SkillActiveModel(zhudongjineng.txt).specialType**为0的时候走本套逻辑。

**SkillActiveModel(zhudongjineng.txt).goalUseType**

1. 通过主动技能表获取技能目标类型
2. 通过**UpGameSkillAction(upgame\_skill\_action)**主动技能动作表获取目标类型  
   当**UpGameSkillAction(upgame\_skill\_action)**主动技能动作表的第一个动作的**goalType**查到目标即可以释放本技能，后续每个动作都会调取本参数重新获取目标对象进行释放。当无法获取目标对象不会终止技能释放，而是继续运行到全部动作结束。  
     
   注：第一个动作无将直接无目标

作用：

* 本功能可以实现一个技能中5个动作，获取５个随机位置目标攻击点。

## 地面持续特效提供BUFF支持

特殊配置方法

* 按照地面持续特效去配置，效果的type类型配置为3，即地面持续特效命中后提供效果Buff
* 地面的持续样式按照地面持续特效配置，type类型配置为５，作用１次，持续时间写总事件，然后将样式写入
* 实际的Buff效果在其他条目写入，目标对己方写２，对敌人写３，以此类推，实际的地面持续特效找目标会按照５即坐标点去选择。
* Buff的作用次数填写固定的，比如20次，实际每次强制命中后添加1次Buff，如果还站在上面继续命中。作用时间写每一小节的作用时间。

## 召唤技能

如果使用了145,146这2种技能召唤,会将召唤者的Buff属性先取消掉,然后召唤的时候使用本体属性进行合算,召唤完毕会将Buff叠加回去,这样如果先前有一个增加hp上限的buff,然后血已经加满,经过处理后会残血.

### 146变身技能

首先按照召唤者自身属性映射到召唤出的怪物身上,然后将自己杀掉

将开始中的BUFF也覆盖到召唤出来的角色上去

## 复活技能

清理掉仇恨列表

清理掉身上的Buff

复活技能的时间配置

复活技能的特效展示

# 战斗数据的记录

战斗数据需要在本地进行记录, FPS数据转换回来 -> 到游戏中来

一场战斗 10 分钟 = 600 秒

60FPS 数据节奏 = 36000 添加单纯Object 9 毫秒

30FPS 数据节奏 = 18000 添加单纯Object 4毫秒

每秒产生的变化对象数量

只有怪物和人物有变化在场人数设定 : 40 429毫秒

每个人物的变化数据

坐标 X, Y , 释放技能, 播放特效,人物血量变化

记录人为操作的内容

这样其他自动的部分不能有随机变化,或者随机变化是根据参数计算出来的,是可以控制的随机变化值.

反推的话可以完成

# PK模式

PK模式为2人互殴模式,开启PK模式代码如下:

UpGame.modeAttack = true;

打开PK模式后,有主场和客场之分：

主场：玩家处于Camp1队列下。UpGame.modeTurn = false

客场：玩家处于Camp2队列下。UpGame.modeTurn = true

当玩家处于客场的情况下，由于玩家还需要显示在下方，所以界面需要颠倒模式，所以要开启颠倒模式，颠倒模式下坐标ＸＹ都会进行对折。

Camp1总是位于下面, Camp2总是位于上面

在对折模式下EDCamp的左右塔是在对折前的情况，而towerEnd也是在对折前的情况进行处理。

在处理结果的时候，翻转模式实际上是拿到Camp2的胜利情况

为胜利条件EDCamp中增加了主塔,左右塔的变量来记录情况

翻转后血条和模型也要进行翻转

# 英雄系统

英雄从卡牌属性表中获取,通过英雄属性来判断是否是影响.

## 启动技能

为英雄提供一种启动技能，当英雄出来的时候，自动释放出这个技能

主动技能表，增加一种自动释放的字段，当有的时候初始化人物的时候，自动释放这个技能，这个技能不会释放任何表现，效果会永久增加在人物身上。掉血的时候不会变红。

## 升星

英雄一共分为五星,

《英雄星级表》



控制英雄升级经验,增加HP,增加攻击,解锁技能(卡牌属性表ID)

|  |  |  |
| --- | --- | --- |
| **名称** | **能力变化** | **备注** |
| **0** | 普攻、专属技能2 | （普攻，主动技能回城） |
| **1** | 解锁：专属技能3 | （自动施放技能） |
| **2** | 解锁：专属技能4 | （自动施放技能BUFF） |
| **3** | 解锁：专属技能5 | （主动技能大招） |
| **4** | 额外增加属性 |  |
| **5** | 额外增加属性 |  |

## 升星经验

死亡：不获取星级经验

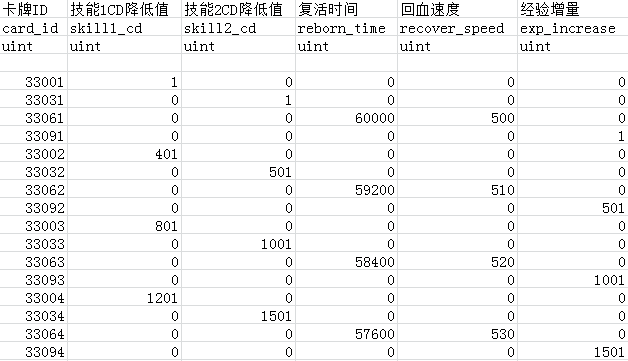
待机：获取50%的经验

出战：获取100%的经验

我方的全部攻击造成敌人死亡后,在死亡怪物身上有死亡经验.增加到英雄里

## 特殊属性

《通用技能表》(影响英雄的技能CD,复活等)



技能1CD降低值，技能2CD降低值，复活时间，回血速度，经验增量

# 多人联网

提供多个人一起娱乐

## 能量系统

能量最大值：**UpGame.engine.time.energyMax** 游戏能量的最大值

能量增长速度：**UpGame.engine.time.timeScale** 真实时间和游戏时间的比例

能量增长比例：**UpGame.engine.time.timeEnergyScale** 游戏能量条的增长速度

能量现在值：**UpGame.engine.camp1.energyValue** 已经取整了

能量现在值：**UpGame.engine.camp1.energyTemp** 未取整的值

# 不同版本串联

战斗中会碰到不同版本的对战等问题.

也会碰到不同版本在线上运营.如何实现不同版本的数据校验

这里我们使用,所有版本都使用接口串联起来

# 人工智能(机器人)

机器人做为人机对战自动战斗的必要元素

机器人分智力等级

进攻,防御,中庸,策略,推算

单线进攻,多线进攻,兵种配合,节奏控制,摆放控制

# 版本V1

这里介绍V1版本的一些内容.

本版本为第一个版本,项目依托与GaGaFrameForClient框架,编辑器使用GaGaFrameForDevelopMax工具来编写,显示对象使用FLASH原生的显示对象,主要以BITMAP来处理,游戏可以在渲染中,加速,减速,回退,快进,跳转等操作.

在编辑器中基本可以完成对战等内容.

游戏使用的显示对象是以U2格式为显示对象.

## V1版本

## V1 BUG收集

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **严重程度** | **说明** | **解决方案** |
|  |  |  |  |