【目录】

【责任声明】

【摘要】

【背景调查】

【项目运行代码截图】

【游戏运行流程】

【项目代码解析】

【毕业设计目的及出发点】

设计的定位：本毕业设计是一个Demo（样例）级别的RPG（角色扮演游戏）

开发目的及出发点：

1，作者团队对游戏开发有浓厚的兴趣，以此作为毕业设计题材有助于增强积极性和主动性。

1，作者团队曾从事过游戏开发相关的工作，对游戏开发有较为全面的知识储备。且作者团队年轻的时间受著名网络游戏《冒险岛》的影响较深，本次毕业也是向《冒险岛》致敬的作品

1，希望通过本次毕业设计进一步了解游戏，理解中大型游戏的设计需求及可能遇到的问题和难点。寻找相应的解决办法，增强作者团队的编程能力。

1，作者团队希望通过把本毕业设计的代码开源，能帮助到同样对游戏开发有兴趣的程序员，作为开发样例参考。

1，由于时间精力有限，本设计只实现游戏的大体框架，只填充一些参考数值。 不实现具体的数值计算。

【毕业设计分工】

20122100031毛静文负责：

【使用的技术及框架】

【游戏名词词典】

1，任务：

1，成就：

1，装备：

1，装备等级：

1，副本：

1，物品：

1，碎片：

1，药剂：

1，金币：

1，钻石：

1，技能：

1，属性:

1，职业：

1，等级：

1，生命值（血量）：生命值用于维持生命，角色受到敌人的攻击时会扣损生命值。当角色生命值为零时，视为本次闯关失败。

1，法力值（法量）：

1，物理攻击（物攻）：

1，法术攻击（法功）：

1，物理抗性（物抗）：

1，法术抗性（法抗）：

1，攻击速度（攻速）：

1，移动速度（移速）：

1，单体攻击（单攻）：

1，群体攻击（群攻）：

【主要游戏逻辑介绍】

本作品结合了：

1,著名网游《冒险岛》的动画风格:即本作品主要的图片及动画素材是来自于《冒险岛》游戏的素材。

2,当下流行的手机卡牌游戏的角色养成系统:游戏的养成系统（游戏角色能力的成长系统）是参考当下流行的手机卡牌类游戏。

3,以及惊险刺激的打怪升级游戏模式:主要的游戏逻辑则是采用打怪升级的游戏逻辑，通过玩家操作角色击败敌人，获得奖励，进行闯关。

【主要框架功能介绍】

主要功能模块简要介绍：

欢迎界面：

1，每日任务：玩家通过执行任务获得游戏奖励，增强游戏角色的能力，从而探索游戏中更多的内容。而每日任务通过每天特定任务给玩家设置每天游戏的目标，对玩家每天的游戏设置了目标指导，增强玩家的积极性和习惯性。

2，成就荣誉：与每日任务类似，给出一系列的任务目标，对游戏起到指导性作用。但与每日任务不同的是。成就荣誉给出的任务在整个游戏中只能被完成一次，而且相对于每日任务，完成成就荣誉的难度要更大。但响应的获得的奖励也更丰富。成就荣誉的设计是当给玩家一个长期奋斗的目标，一个长期的游戏指引。当玩家完成每日任务时，仍有继续进行游戏的目的

3，装备系统：本游戏设置有装备系统（只指提升游戏角色能力的一种辅助系统，通常以衣服着装的形式出现。通过穿着装备，以及穿着不同的装备，提升游戏角色的能力）。本游戏设置还有武器商店，可以进行装备的买卖操作。通过设置装备系统，提升游戏的趣味性并丰富游戏的可玩性。

4，药品系统：游戏设置有药品系统，玩家在游戏中通过补给相应的药剂恢复生命值或法力值，以增强游戏的策略性和可玩性。本游戏还设置有药品商品，玩家可以在药品商店中购买自己需要的药品。但每次闯关只能携带特定一种生命值药剂和法力值药剂。

5，时空仓库：相当于角色的家，玩家可以在该模块查看到玩家拥有的一切资源，如装备，物品，碎片，药品，拥有的金币数，钻石数等等。也可以在该页面实现装备的穿着，物品的贩卖，药品的携带等操作。

6，玩家信息模块：在该模块可以查看玩家角色的相关信息，包括角色当前穿着的装备，角色的血量，法量，物攻，法攻等等。还可以进行装备的卸载和穿着操作。

7，潜能激发：该模块显示了玩家角色的技能信息以及属性信息，可以进行技能升级，属性升级操作。还可以进行特定技能的携带操作。

副本选择界面：

1，世界地图：显示本游戏拥有的战斗地图。实现地图选择的操作。点击地图下方的卷轴图标，查看该地图的具体信息

2，地图详细信息面板：用于查看选定地图的详细信息，如地图名，副本产出，推荐闯关等级，已经地图等级等信息。点击具体的地图难度，进入特定地图的战斗场景。

战斗闯关界面：

主要分成上中下三部分，分别为

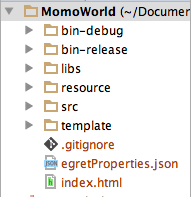
信息栏（顶端部分）：显示角色信息，如：角色等级，角色当前血量，角色当前法量，角色经验值。以及地图信息，如：地图名

战斗场景（中部栏）：是战斗主要场景，用于显示角色和怪物的战斗动画

操作栏（底端部分）：是游戏主要操作层，实现对游戏角色的控制，如左右移动，药剂的使用，技能的使用，物理攻击，已经角色的跳跃功能。

【各个功能框架及对应源码详细介绍】

1，**项目整体文件夹预览**



MomoWorld：项目源码文件夹，项目的全部资源将存放在这里

Bin-dubug:为存放Egret引擎编译TS文件生成的对应JS文件的文件夹

Bin-release:为项目Egret发布项目的文件夹，项目开发完成后将编译，压缩生成对应的项目发布版项目文件

Libs：存放项目用到的库代码文件

Resource：存放开发期间用到的媒体文件，如图片资源，音频资源等

Src：存放源代码文件TS文件的文件夹，是程序开发主要用到的文件夹。

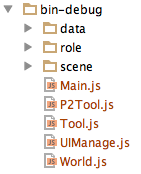
Template:存放项目发布时用到的配置文件的文件夹

egretProperties.json：文件为游戏引擎Egret的项目配置文件

index.html：是游戏入口，开发过程中，通过访问该文件进入游戏逻辑

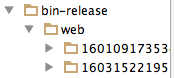
子文件夹介绍

2，bin-debug文件夹



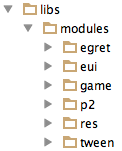
bin-debug文件夹是存放编译后对应JS文件的文件夹。该文件夹中的内容由游戏引擎Egret根据src源文件编译生成，一般情况下，不需要进行设置

3，bin-release



是发布项目发布包的文件夹，里面存放了发布到不同运行环境中的发布包。本项目是开发HTML5上的游戏，故本文件夹内只有web文件夹，用于保存浏览器环境的发布包。（Egret引擎还可以发布Android环境的发布包app，以及IOS运行环境上的app）

4，libs文件夹



libs文件夹存放游戏用的的库文件。不同的库文件负责不同功能的实现。通过配置特定的库文件，缩小项目的体积，以及优化项目的性能。

Egret：是游戏引擎Egret的核心文件，用于基本的发布文件，已经游戏引擎的配置

Eui：是提供UI实现的库文件

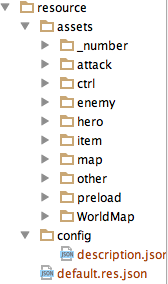
Game：是提供游戏基本元素实现的库文件，如显示对象，显示对象容器，精灵等

P2：是Egret游戏引擎对应物理引擎P2的库文件，主要提供物理计算的实现

Res:是资源管理文件，提供的资源管理的相关实现

Tween:是动画管理文件，用于提供帧间动画的实现

5，resource



resource是存放项目媒体资源的文件夹，项目用到的所有图片资源都存放在这里

assets：是存放图片的文件夹，在本文件夹内，笔者使用不同的文件夹细分不同的图片文件

\_number:存放数字相关的资源，如攻击特效数值，受伤特效数值等

attack:存放角色的攻击特效相关资源图片

ctrl：存放控制面板相应UI图片资源

enemy:存放敌人动作的相关动画资源

item:是存放各种图标资源，如装备图标，药品图标，物品图标等

map：用于存放地图资源

other:存放剩余琐碎的图片资源

preload:存放加载页面的图片资源

WorldMap:存放地图选择页面的UI资源

config：是资源管理的配置文件，用于设置资源的解析方式

description.json：即相应的资源解析配置文件

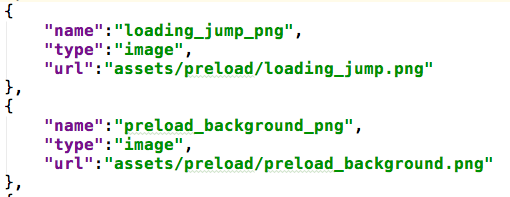
default.res.json：是存放图片索引配置的文件，

如下图，配置了

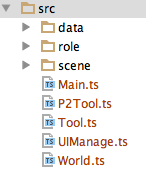
该图片的名字，引用资源时，使用的就是该图片名，

该资源的媒体类型，

以及该图片对应的实际图片路径，即该图片资源存放在哪个文件夹



6，src



src是逻辑代码文件夹，里面保存了项目主要运行逻辑的文件夹，是整个项目主要编写的文件夹，也是本文后续详细介绍的部分

data：data文件夹保存了游戏对象模型的相关信息。在这里起到数据库的作用。由于本项目是demo级别的，所以没有假设数据库，只通过本文件夹中的对象配置文件来模拟数据库数据

role：是项目涉及到的对象的解析文件，用于解析data文件夹中的数据。模拟人物动作，及相关逻辑的实现

scene：是保存游戏相关场景实现文件的文件夹

Main是项目的逻辑入口，项目媒体资源的加载，已经初始化配置就在本文件实现

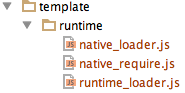
P2Tool：是作者写的P2物理引擎的工具类文件，用于加快物理引擎的相关配置

Tool:是作者写的Egret引擎的工具类文件，用于加快开发速度

UIManage:是场景控制器文件，用于切换不同的场景页面

World：是游戏世界类文件，用于协调各个对象的刷新，已经对应的键盘响应

7，template

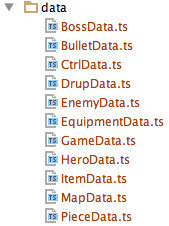


template:是设置发布包参数的文件，可以设置如何生成发布包等信息。

【src文件夹基本介绍】

data：data文件夹保存了游戏对象模型的相关信息。在这里起到数据库的作用。由于本项目是demo级别的，所以没有假设数据库，只通过本文件夹中的对象配置文件来模拟数据库数据

是MVC架构中的Model层，用于保存对象数据



BossData：用于保存Boss怪物的数据的文件，

BulletData：子弹信息文件，用于存放有“子弹”运动 的对象信息的类，这里存放的是弓箭的相关信息

CtrlData： 操作面板的静态数据信息文件，如控制面板中的装备商店，药品商店，每日任务等相关操作面板静态信息的数据信息文件

DrupData： 药品数据信息文件，存放药品相关信息。添加药品种类等需要在本文件夹中配置

EnemyData：敌人信息文件，设置敌人的相关属性，如各个动画的偏移值，怪物血量等

EquipmentData：装备信息文件，存放游戏角色相应装备的信息，如价钱，图标资源等

GameData：全局数据文件，游戏后需要保存的数据，游戏前加载的进度数据等相关信息保存在本文件内

HeroData：游戏角色信息文件，用于保存游戏角色的各方面参数，如血量，动作动画等

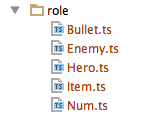
ItemData：物品信息文件，保存打击怪物后掉落物品的相关信息，如物品名，出售价格等

MapData：地图信息文件，保存战斗地图的相关信息，如地图名，地图存在的怪兽，地图背景等

PieceData：碎片信息文件，用于保存合成装备的装备碎片的 相关信息

Role文件夹：是项目涉及到的对象的解析文件，用于解析data文件夹中的数据。模拟人物动作，及相关逻辑的实现

对应MVC架构中事物对象的Controller操作层及对应的View视图层



Bullet：子弹操作类，实现子弹对象的各个运动及响应逻辑，如子射击，子弹的命中等

Enemy：敌人操作类，实现怪物对象的各个运动及响应逻辑，如怪物的攻击，行走等

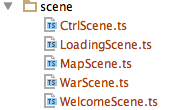
Hero：主角操作类，实现游戏角色对象的各个运动及响应逻辑，如主角的攻击，行走等

Item：物品操作类，实现图标物品对象的各个运动及响应逻辑，如物品的掉落，物品的捡拾

Num：数值操作类，实现数值对象的各个运动及响应逻辑，如数值的显示和消失

scene文件夹：是保存游戏相关场景实现文件的文件夹

对应MVC架构中游戏场景的Controller操作层及对应的View视图层



CtrlScene：操作层页面，用于显示触碰操作的控制区

LoadingScene：加载页面，游戏加载时显示的页面

MapScene：地图选择页面，进行游戏战斗地图时显示的地图选择页面

WarScene：具体战斗场景类，进行角色具体战斗时显示的页面

WelcomeScene：角色基地页面，也是欢迎页面（游戏进入后的第一个页面）

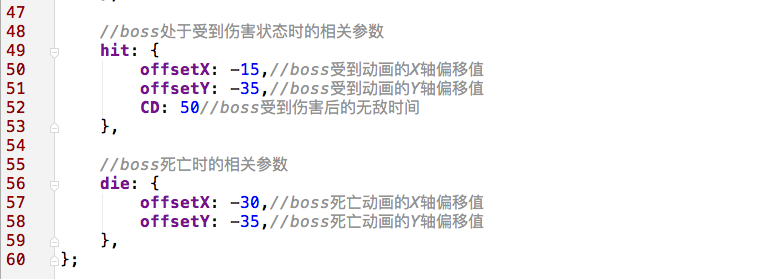
【源代码文件代码级介绍】

本部分主要介绍每个逻辑代码的主要内容，在此先有个大体的认识，后面的部分将会根据项目运行的逻辑顺序做个连贯的思路介绍。

本部分的介绍主要通过代码备注以及补充的备注内容进行介绍。但主要以代码备注为主

BossData:Boss信息配置文件





补充说明：

1，getBoss函数接受一个参数：即该boss 的索引下标。

函数通过JS的”[]”语法，组建程序需要的特定boss数据返回。

2，动画偏移值说明。在游戏逻辑中，每个对象都设置有一个中心点。通过该中心点进行位置的判断。而每一个动画相对于该中心点都不是全对称的。所以需要根据中心点配置该动画的偏移值

3，怪物的宽度设置，由于游戏主角碰到怪物时会受到伤害。所以要为每一个怪物设置其宽度及高度，用于结合对象的中心点判断游戏主角碰到怪物，受到伤害

BulletData：子弹信息配置文件



补充说明：

1，由于解析器的特殊设置，不同的子弹可能会有不同的参数。

2，其中bullet1是角色普通攻击是的相关参数

3，bullet2是角色使用单攻技能一（加速箭）时使用的参数

【待完善的地方】

【附录】

【参考文献】

【致谢】