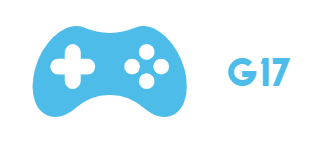
详细设计

打豆豆

——基于windows操作系统下的2d塔防游戏



项目名称： 打豆豆

专业班级： 软件工程2004

小组成员：李时博、范伟、秦思豪

指导教师： 杨枨

**版本历史**

版本记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 修订日期 | 版本/状态 | 修订人 | 备注 |
| 01 | 2022.11.16 | 0.1 | 李时博 | 无 |
| 02 | 2022.11.18 | 0.2 | 李时博/范伟 | 无 |

[1引言 4](#_Toc291534488)

[1.1编写目的 4](#_Toc320896926)

[1.2背景 4](#_Toc1765851930)

[1.3定义 4](#_Toc27078108)

[1.4参考资料 4](#_Toc558662696)

[2程序系统的结构 5](#_Toc1517160120)

[3游戏主程序设计说明 5](#_Toc962113949)

[3.1程序描述 5](#_Toc1827929880)

[3.2功能 5](#_Toc1225244061)

[3.3性能 5](#_Toc363897284)

[3.4输人项 6](#_Toc1087410321)

[3.5输出项 6](#_Toc368669562)

[3.6算法 6](#_Toc1017339525)

[3.7流程逻辑 6](#_Toc1920743881)

[3.8接口 6](#_Toc1772547747)

[3.9存储分配 6](#_Toc1804761141)

[3.10注释设计 6](#_Toc861210837)

[3.11限制条件 6](#_Toc205763265)

[3.12测试计划 7](#_Toc206429974)

[3.13尚未解决的问题 7](#_Toc1866055382)

[4 游戏控制模块设计说明 7](#_Toc1870121787)

[4.1程序描述 7](#_Toc99224998)

[4.2功能 7](#_Toc1759513610)

[4.3性能 7](#_Toc1828727701)

[4.4输人项 7](#_Toc2082978086)

[4.5输出项 7](#_Toc1598160718)

[4.6算法 8](#_Toc86046839)

[4.7流程逻辑 8](#_Toc1100815417)

[4.8接口 8](#_Toc102359857)

[4.9存储分配 8](#_Toc1933083571)

[4.10注释设计 8](#_Toc2002773755)

[4.11限制条件 8](#_Toc1018948760)

[4.12测试计划 8](#_Toc756479673)

[4.13尚未解决的问题 8](#_Toc270640893)

[5 怪物模块设计说明 9](#_Toc884592162)

[5.1程序描述 9](#_Toc1241367400)

[5.2功能 9](#_Toc1022987626)

[5.3性能 9](#_Toc1545749684)

[5.4输入项 9](#_Toc44947420)

[5.5输出项 9](#_Toc1982190021)

[5.6算法 10](#_Toc1848183481)

[5.7流程逻辑 10](#_Toc976746475)

[5.8接口 10](#_Toc935312459)

[5.9存储分配 10](#_Toc1142372002)

[5.10注释设计 10](#_Toc1716510315)

[5.11限制条件 10](#_Toc1216840333)

[5.12测试计划 11](#_Toc1728067907)

[5.13尚未解决的问题 11](#_Toc1572571051)

[6 自动计时模块设计说明 11](#_Toc392528316)

[6.1程序描述 11](#_Toc1682256310)

[6.2功能 11](#_Toc366552074)

[6.3性能 11](#_Toc889132188)

[6.4输人项 11](#_Toc322651860)

[6.5输出项 11](#_Toc208771310)

[6.6算法 11](#_Toc1675129046)

[6.7流程逻辑 11](#_Toc123228876)

[6.8接口 11](#_Toc705160009)

[6.9存储分配 11](#_Toc2014713634)

[6.10注释设计 11](#_Toc1403285011)

[6.11限制条件 11](#_Toc1133514044)

[6.12测试计划 11](#_Toc1302540048)

[6.13尚未解决的问题 11](#_Toc39994883)

[7排行榜模块设计说明 13](#_Toc2044059851)

[7.1程序描述 13](#_Toc1622262448)

[7.2功能 13](#_Toc2100413047)

[7.3性能 13](#_Toc1032242358)

[7.4输入项 13](#_Toc1160740966)

[7.5输出项 14](#_Toc1379896183)

[7.6算法 14](#_Toc1295193554)

[7.7流程逻辑 14](#_Toc925119826)

[7.8接口 15](#_Toc1940869167)

[7.9存储分配 15](#_Toc1340643149)

[7.10注释设计 15](#_Toc1023161932)

[7.11限制条件 15](#_Toc1174745257)

[7.12测试计划 15](#_Toc689122091)

[7.13尚未解决的问题 16](#_Toc1140775842)

**详细设计说明书**

# 1引言

## 1.1编写目的

编写这份详细设计说明书的目的是便于进行本项目的开发，保证项目团队按时保质地完成项目目标，预期的读者为小组成员与周围同学。

## 1.2背景

本文档用于说明对于《打豆豆》的总体设计。

本项目是通过windows环境下本地运行exe程序，登录账号密码后进入2d塔防游戏界面，玩法是基于经典游戏《植物大战僵尸》改编而来。

伴随着我们童年一起成长的有无数经典著作，比如红色警戒、消消乐等等游戏巨著，而在这么多游戏之中，有一款游戏令我至今都记忆犹新，它是我人生之中接触到的第一款塔防类游戏，可以说它甚至比当今时代那些游戏新生儿（比如说部落冲突、海岛奇兵）都要优秀得多，它恐怕也是这类游戏的开山鼻祖，他就是由宝开游戏公司开发的游戏——植物大战僵尸。

这款游戏已经经历了无数次的重大更新，从最开始的95版一直到现在才更新的平安时代，它的优化已经做得越来越好，原本呆头呆脑的豌豆射手，现在增加了往前吐的特效，还有无数诸如此类的细节改动，这些看似朴实无华的举动，事实上正好完美地朝显出游戏公司对于植物大战僵尸的在意。

我们小组一直都是一个植物迷，它对我有着非凡的意义。这款游戏在我印象中一直都是非常不错的。记得我有一个朋友，他也同样喜欢植物，他的植物大战僵尸已经解锁了很多的世界，就因为这一点，我对他总是高看一眼。有人问我：“你为什么喜欢玩植物而胜过其它游戏呢？”我的回答往往是：“因为它带给我了无数童年的欢笑”。

不过我们也不得不承认，这款游戏也许真的不适应市场发展趋势了。在我童年时期还和我一起玩植物的朋友如今早已将它抛弃了，也许是他们心智成熟了，又或许是它真的老了。没错，植物大战僵尸这款游戏身上也有很多的缺点，比如说它是单机游戏，在当今互联网飞速发展的今天，它不再顺应历史发展潮流；再比如它的玩法很少，仅管宝开公司已经可以说做到了极致，但现代游戏讲究的是“多元”二字，既然它违背了“生存的法则”那么它的淘汰又或许是意料之中的事。总的来说，这款游戏给我的童年带来了太多太多，无论当今时代玩它的人还有多少，但只要它还未下架，我将必定继续鼎力支持它的。

因此，我们小组决定，在《植物大战僵尸》这款游戏的原有玩法上，加入自己的创作，和联网的内容，并且对一些好的方面进行改进和提高。

当然用户：杨枨老师

典型用户：李时博、范伟、秦思豪、鲍明、潘劲帆、张宁宁

## 1.3定义

## 1.4参考资料

本章应列出本文档引用的所有文档的编号、标题、修订版本和发行日期，也应标识不能通过正常的供货渠道获得的所有文档的来源。

1. 软件需求说明（GB8567-88）
2. SE2022-G17-系统设计说明.doc
3. MD5 加密算法详细介绍<https://blog.csdn.net/qq_37469055/article/details/87927087>

## 1.5界面设计

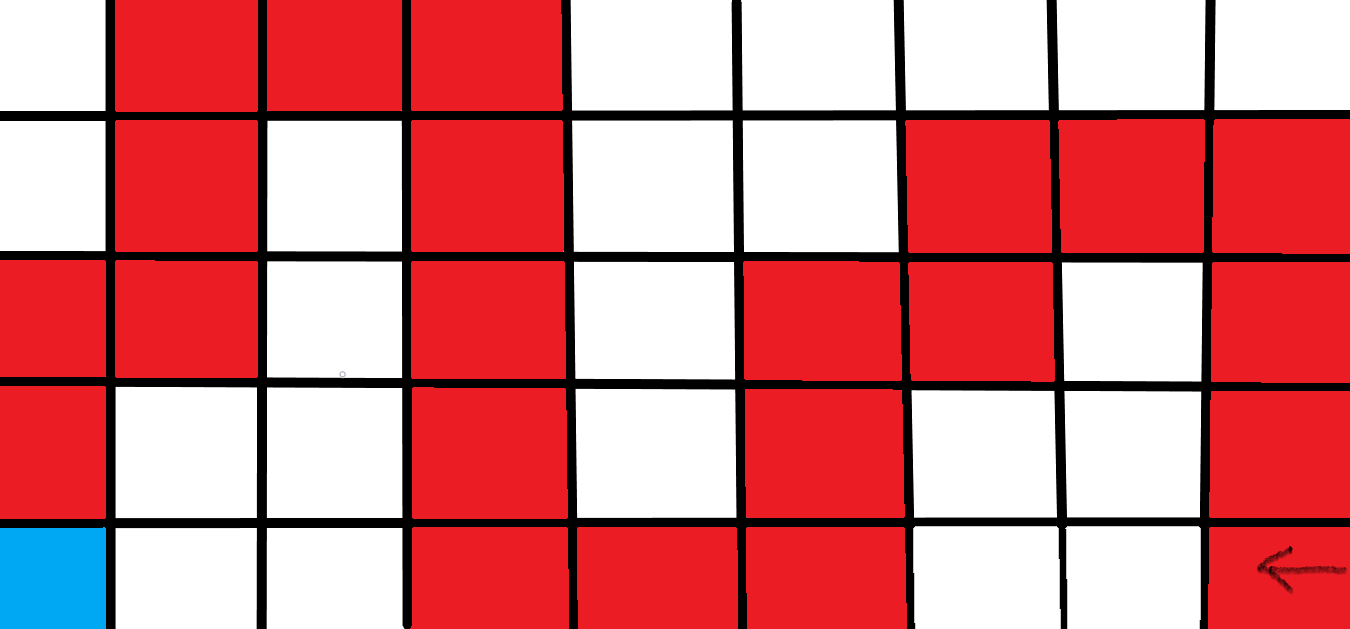
登录界面



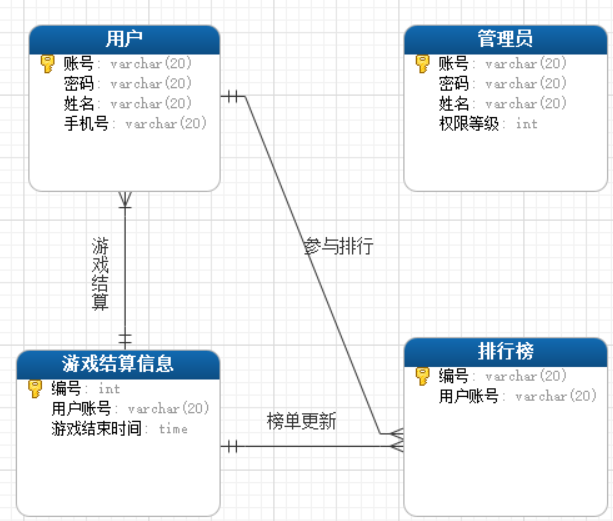
主界面



游戏界面



## 1.6数据库设计



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用户表设计（user） | | | | |
| 字段名 | 中文名称 | 数据类型 | 能否为空 | 说明 |
| **user**\_**ID** | **用户账号** | **Varchar** | **否** | **主键** |
| user\_Password | 用户密码 | Varchar | 否 |  |
| user\_Name | 用户名 | Varchar | 否 |  |
| user\_phone | 用户手机号 | Varchar | 否 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 管理员表设计（admininstrator） | | | | |
| 字段名 | 中文名称 | 数据类型 | 能否为空 | 说明 |
| admininstrator \_**ID** | **管理员账号** | **Varchar** | **否** | **主键** |
| admininstrator \_Password | 管理员密码 | Varchar | 否 |  |
| admininstrator \_Name | 管理员用户名 | Varchar | 否 |  |
| admininstrator \_Level | 管理员权限等级 | Int | 否 |  |

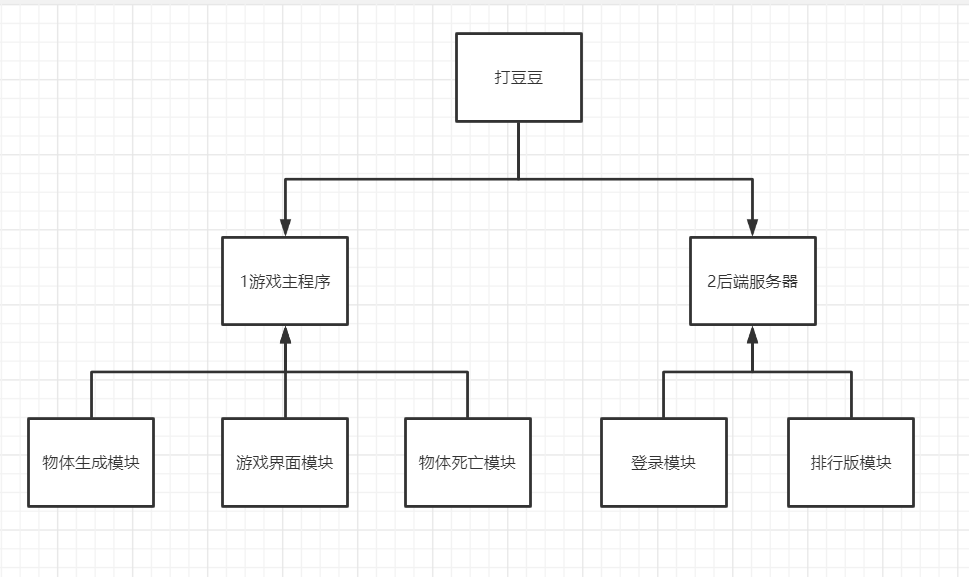
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 游戏结算信息（information） | | | | |
| 字段名 | 中文名称 | 数据类型 | 能否为空 | 说明 |
| **information**\_**ID** | **信息编号** | **Varchar** | **否** | **主键** |
| **user**\_**ID** | 用户账号 | Varchar | 否 | 外键 |
| Finish\_time | 结束时间 | Time | 否 |  |
| score | 游戏得分 | Int | 否 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 排行榜（RankList） | | | | |
| 字段名 | 中文名称 | 数据类型 | 能否为空 | 说明 |
| **RankList\_ID** | **榜单编号** | **Varchar** | **否** | **主键** |
| Update\_time | 更新时间 | Time | 否 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 防御塔信息（Tower\_information） | | | | |
| 字段名 | 中文名称 | 数据类型 | 能否为空 | 说明 |
| **Tower\_ID** | **防御塔编号** | **Int** | **否** | **主键** |
| **Tower\_Damage** | **伤害** | **Int** | **否** |  |
| **Tower\_AttackRate** | **攻击时间** | **Time** | **否** |  |
| **Tower\_Range** | **攻击范围** | **Int** | **否** |  |
| **Tower\_CRIT** | **暴击率** | **Int** | **否** |  |
| **Tower\_Cost** | **费用** | **Int** | **否** |  |
| **Tower\_English** | **英文名** | **Varchar** | **否** |  |
| **Tower\_DamageF** | **伤害类型** | **Varchar** | **否** |  |
| **Tower\_Chinese** | **中文名** | **Varchar** | **否** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 怪物信息（Monster\_information） | | | | |
| 字段名 | 中文名称 | 数据类型 | 能否为空 | 说明 |
| **Monster\_ID** | **怪物编号** | **Int** | **否** | **主键** |
| **Monster\_Damage** | **伤害** | **Int** | **否** |  |
| **Monster\_AttackRate** | **攻击时间** | **Int** | **否** |  |
| **Monster\_Range** | **攻击范围** | **Int** | **否** |  |
| **Monster\_Cost** | **伤害类型** | **Varchar** | **否** |  |
| **Monster\_English** | **英文名** | **Varchar** | **否** |  |
| **Monster\_Chinese** | **中文名** | **Varchar** | **否** |  |
| **Monster\_DamageF** | **移动速度** | **Int** | **否** |  |

# 2程序系统的结构

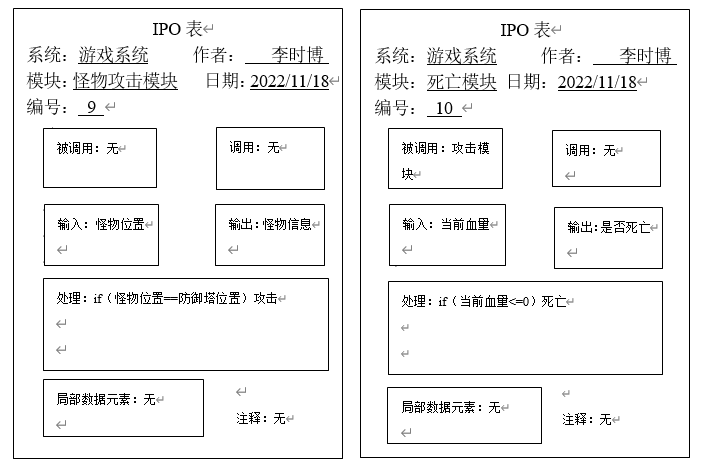
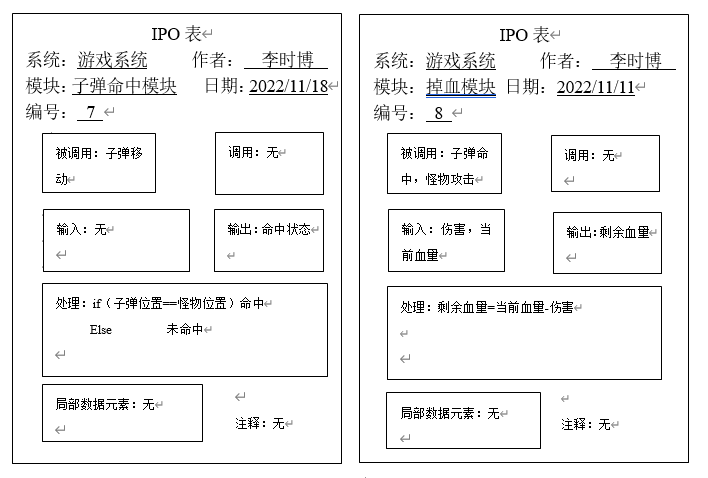
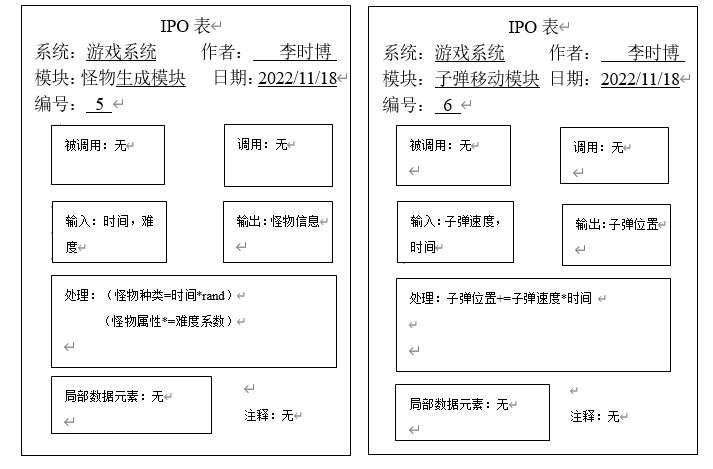
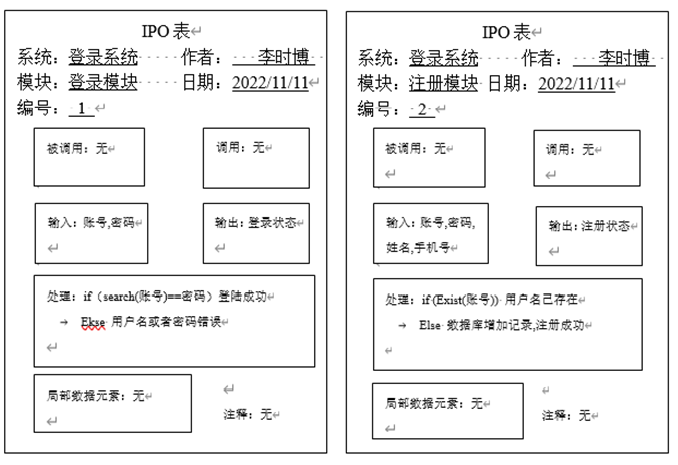


# 3游戏主程序设计说明

## 3.1程序描述

本程序是供玩家游玩的主程序，设计目的是给玩家提供塔防游戏服务，根据玩家的需要购买和放置防御塔来抵御怪物额达入侵。

## 3.2功能



## 3.3性能

游玩人员可以自由放置防御塔和移除防御塔，体现了灵活性，对于子弹速度和怪物位置的实时更新体现了游戏精度，实时更新也是本游戏的时间特性

## 3.4输人项

手动输入

|  |  |  |
| --- | --- | --- |
| **user**\_**ID** | **用户账号** | **Varchar** |
| user\_Password | 用户密码 | Varchar |
| user\_Name | 用户名 | Varchar |
| user\_phone | 用户手机号 | Varchar |
| admininstrator \_ID | 管理员账号 | Varchar |
| admininstrator \_Password | 管理员密码 | Varchar |
| admininstrator \_Name | 管理员用户名 | Varchar |
| admininstrator \_Level | 管理员权限等级 | Int |

## 3.5输出项

排行榜

|  |  |  |
| --- | --- | --- |
| **RankList\_ID** | **榜单编号** | **Varchar** |
| Update\_time | 更新时间 | Time |

游戏结束后自动输出

|  |  |  |
| --- | --- | --- |
| **information**\_**ID** | **信息编号** | **Varchar** |
| **user**\_**ID** | 用户账号 | Varchar |
| Finish\_time | 结束时间 | Time |
| score | 游戏得分 | Int |

## 3.6算法

## MD5的作用

①一致性检验，最上面那个例子

②数字签名，还是最上面那个例子。只是把md5看出了一个指纹，按了个手印说明[独一无二](https://www.baidu.com/s?wd=%E7%8B%AC%E4%B8%80%E6%97%A0%E4%BA%8C&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd)了。

③安全访问认证，这个就是平时系统设计的问题了。

在用户注册时，会将密码进行md5加密，存到数据库中。这样可以防止那些可以看到数据库数据的人，恶意操作了。

## MD5是什么

message-digest algorithm 5（信息-摘要算法）。经常说的“MD5[加密](https://so.csdn.net/so/search?q=%E5%8A%A0%E5%AF%86&spm=1001.2101.3001.7020)”，就是它→信息-摘要算法。

在下载一下东西时，经常在一些压缩包属性里，看到md5值。而且这个下载页面，很可能会在某一个地方，写了一句，此文件的MD5值为XXXXXXXXX。这有什么作用呢？

md5，其实就是一中算法。可以将一个[字符串](https://so.csdn.net/so/search?q=%E5%AD%97%E7%AC%A6%E4%B8%B2&spm=1001.2101.3001.7020)，或文件，或压缩包，执行md5后，就可以生成一个固定长度为128bit的串。这个串，基本上是唯一的。

所以，有人修过压缩包后，就会生成新的串，这时就可以拿网站提供的串和新生成的串对比，如果不同，那就是被人修过过了。

怪物血量=怪物原本血量-防御塔攻击次数\*伤害

防御塔血量=防御塔原本血量-怪兽攻击次数\*怪兽伤害

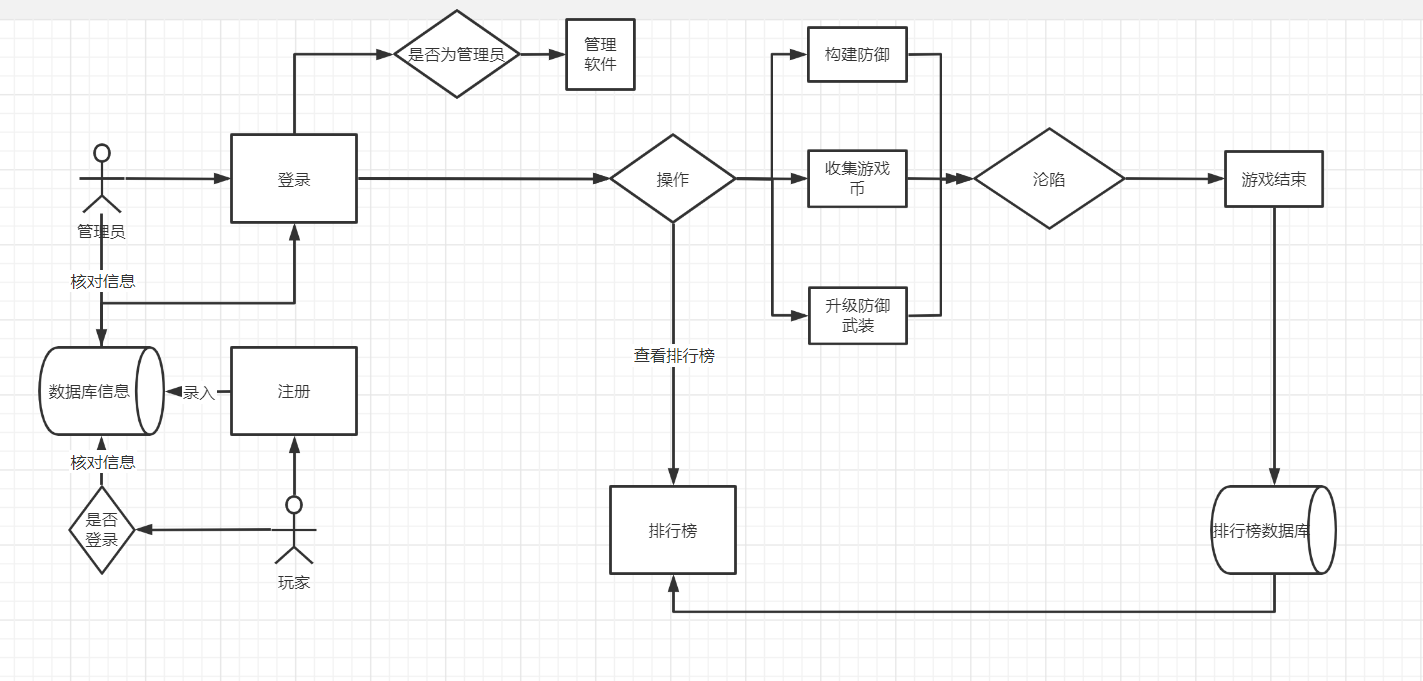
怪物距离=时间\*速度

子弹飞行时间=（防御塔位置—怪兽位置）/子弹速度

Md5加密算法

## 3.7流程逻辑

## 3.8接口



|  |  |  |
| --- | --- | --- |
| **接口** | **模块** | **描述** |
| UserRegister() | 用户注册 | 用户注册 |
| AdministratorRegister() | 管理员注册 | 管理员注册 |
| UserSignIn() | 用户登录 | 用户登录 |
| AdministratorSignIn() | 管理员登录 | 管理员登录 |
| UpdateInfo() | 更改用户信息 | 更改用户信息 |
| DeleteInfo() | 删除用户信息 | 删除用户信息 |
| UpdateRank() | 更新榜单 | 更新榜单 |

## 3.9存储分配

根据需要，说明本程序的存储分配。

## 3.10注释设计

说明准备在本程序中安排的注释，如：

1. 加在模块首部的注释；
2. 加在各分枝点处的注释；
3. 对各变量的功能、范围、缺省条件等所加的注释；
4. 对使用的逻辑所加的注释等等。

## 3.11限制条件

系统：Windows7以上系统

软件：装有vb6.0以及SQL SERVER数据库

## 3.12测试计划

## 1、测试方法：让几位同学来游玩

## 2、测试用例：不同的电脑

## 3、用一半用户反复登录，多次点击测试，验证系统的牢固性

## 3.13尚未解决的问题

软件工程项目课程

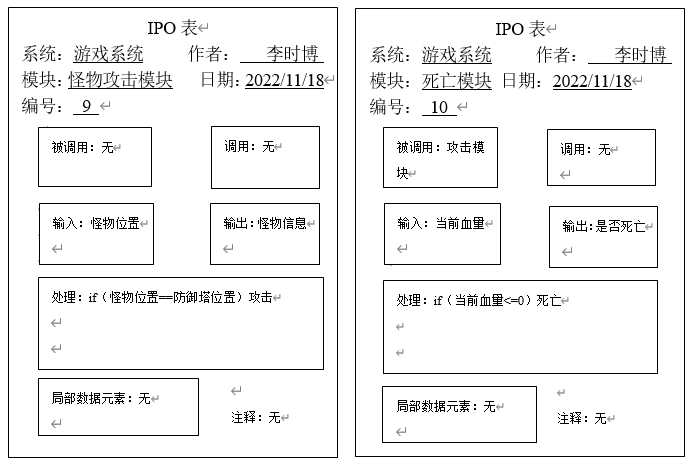
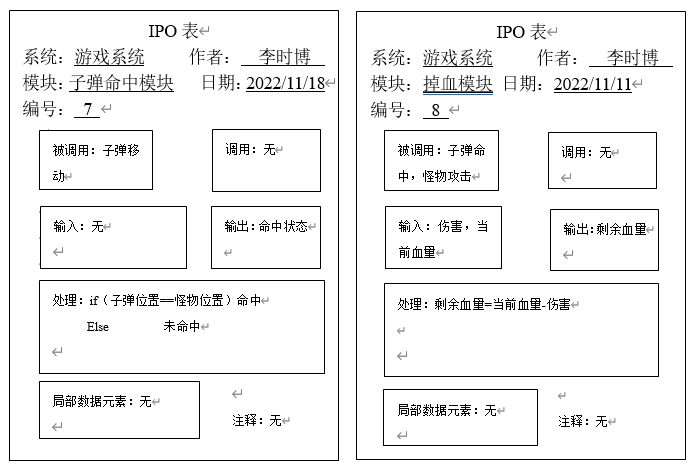
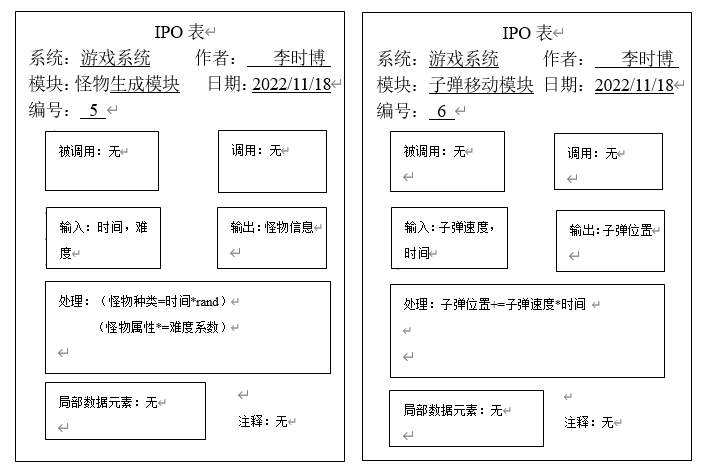
# 4 防御塔模块设计说明

## 4.1程序描述

防御塔模块是用来描述和计算防御塔在游戏系统中的布置与交互

## 4.2功能

## 



## 4.3性能

说明对该程序的全部性能要求，包括对精度、灵活性和时间特性的要求。

为了玩家更好的体验，防御塔模块需要在0.1s内对玩家的操作作出响应。

数据精度为双精度浮点小数。

需要保证一定的可复用性与鲁棒性。

## 4.4输入项

给出对每一个输入项的特性，包括名称、标识、数据的类型和格式、数据值的有效范围、输入的方式。数量和频度、输入媒体、输入数据的来源和安全保密条件等等。

对于防御塔模块来说，输入项主要来自于玩家的操作如用足够的金币来购置相对应的防御塔，通过玩家点击各个功能，收集记录信息，玩家每次的操作都会给防御塔模块输入这些输入项。

## 4.5输出项

给出对每一个输出项的特性，包括名称、标识、数据的类型和格式，数据值的有效范围，输出的形式、数量和频度，输出媒体、对输出图形及符号的说明、安全保密条件等等。

对于防御塔模块来说，输出项主要有防御塔位置信息和防御塔的攻击信息。

## 4.6算法

怪物血量=怪物原本血量-防御塔攻击次数\*伤害

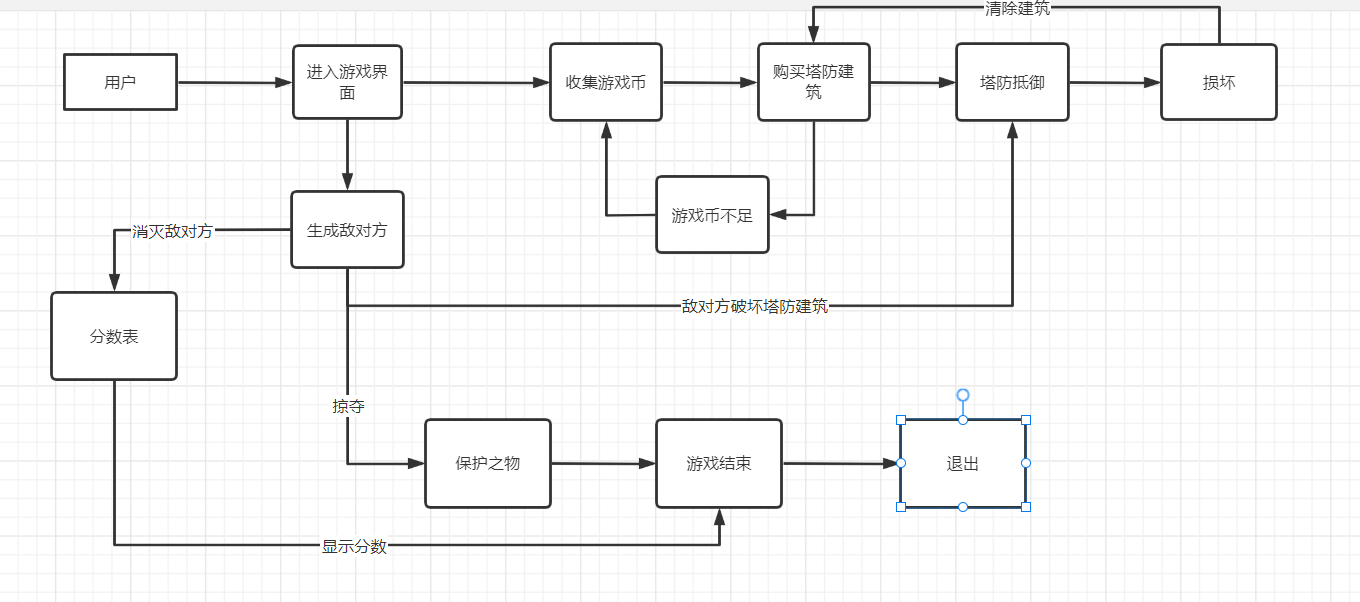
防御塔血量=防御塔原本血量-怪兽攻击次数\*怪兽伤害

怪物距离=时间\*速度

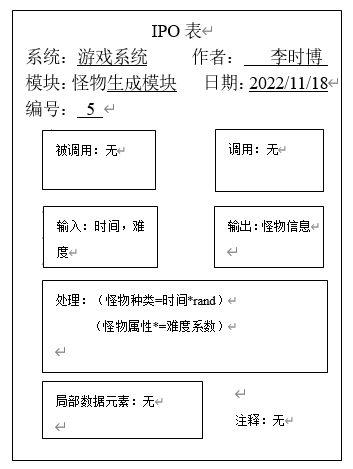
子弹飞行时间=（防御塔位置—怪兽位置）/子弹速度

## 4.7流程逻辑

## 4.8接口



用图的形式说明本程序所隶属的上一层模块及隶属于本程序的下一层模块、子程序，说明参数赋值和调用方式，说明与本程序相直接关联的数据结构（数据库、数据文卷）。



## 4.9存储分配

根据需要，说明本程序的存储分配。

## 4.10注释设计

说明准备在本程序中安排的注释，如：

1. 加在模块首部的注释；
2. 加在各分枝点处的注释；
3. 对各变量的功能、范围、缺省条件等所加的注释；
4. 对使用的逻辑所加的注释等等。

## 4.11限制条件

服务器运算性能，用户计算机硬件性能

## 4.12测试计划

## 1、测试方法：让几位同学来游玩

## 2、测试用例：不同的电脑

## 3、用一半用户反复登录，多次点击测试，验证系统的牢固性

## 4.13尚未解决的问题

无

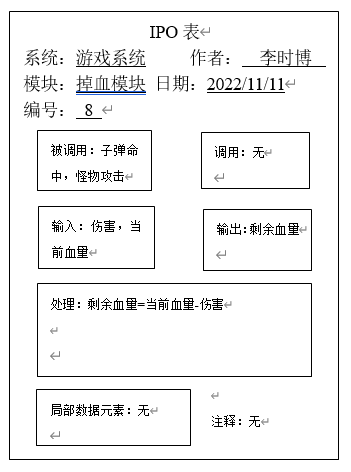
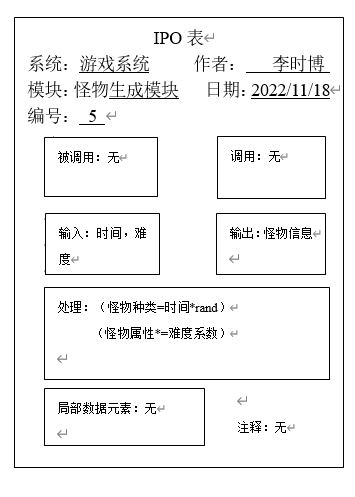
# 5 怪物模块设计说明

## 5.1程序描述

本模块包括怪物生成和怪物血量模块

## 5.2功能

本模块用于管理本游戏主程序中所有的怪物的移动。在游戏开始后怪物会在游戏场景中自动生成， 且往基地移动，对防御塔展开破坏。



## 5.3性能

怪物随着时间有着不同的难度系数，会随着游戏时间的边长而变困难

说明对该程序的全部性能要求，包括对精度、灵活性和时间特性的要求。

为了玩家更好的体验，怪物模块需要在0.1s内对玩家的操作作出响应。

数据精度为双精度浮点小数。

需要保证一定的可复用性与鲁棒性。

## 5.4输入项

对于怪物模块来说，重要的输入项是怪物的移动速度、怪物攻击力，怪物血量以及怪物难度系数等。这些数据都是通过游戏控制模块获取的。

## 5.5输出项

输出怪物的位置以及血量，通过怪物生成初始时间和移动时间来得出怪物具体位置，怪物会随机出现在5行不同的路线。

## 5.6算法

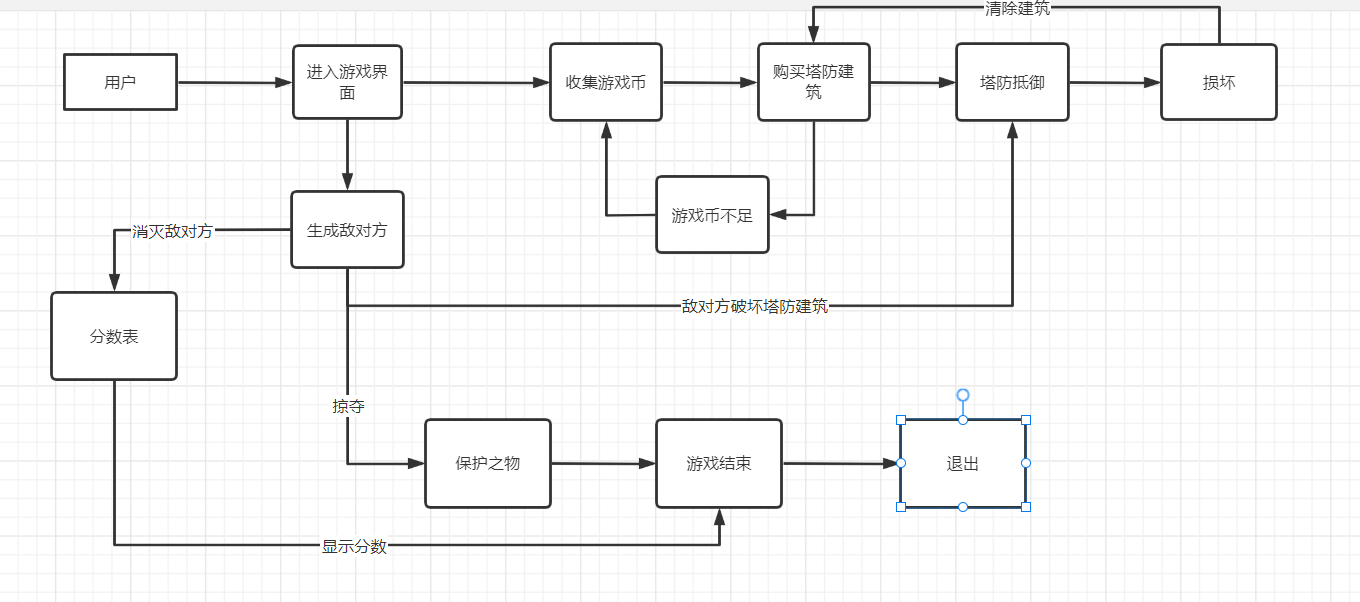
怪物血量=怪物原本血量-防御塔攻击次数\*伤害

怪物距离=时间\*速度

子弹飞行时间=（防御塔位置—怪兽位置）/子弹速度

## 5.7流程逻辑

## 5.8接口



怪物的展现交由游戏控制模块。

用图的形式说明本程序所隶属的上一层模块及隶属于本程序的下一层模块、子程序，说明参数赋值和调用方式，说明与本程序相直接关联的数据结构（数据库、数据文卷）。

## 5.9存储分配

本模块没有需要持久化的数据，所有数据在内存中完成处理。

## 5.10注释设计

说明准备在本程序中安排的注释，如：

1. 加在模块首部的注释；
2. 加在各分枝点处的注释；
3. 对各变量的功能、范围、缺省条件等所加的注释；
4. 对使用的逻辑所加的注释等等。

## 5.11限制条件

由于玩家设备的处理器性能有限，当需要被控制的敌人过多时，本程序的响应可能会过慢；由于玩家的反应速度有限，本程序不应无限生成怪物以及无线加大难度。

## 5.12测试计划

## 1、测试方法：让几位同学来游玩

## 2、测试用例：不同的电脑

## 3、用一半用户反复登录，多次点击测试，验证系统的牢固性

## 5.13尚未解决的问题

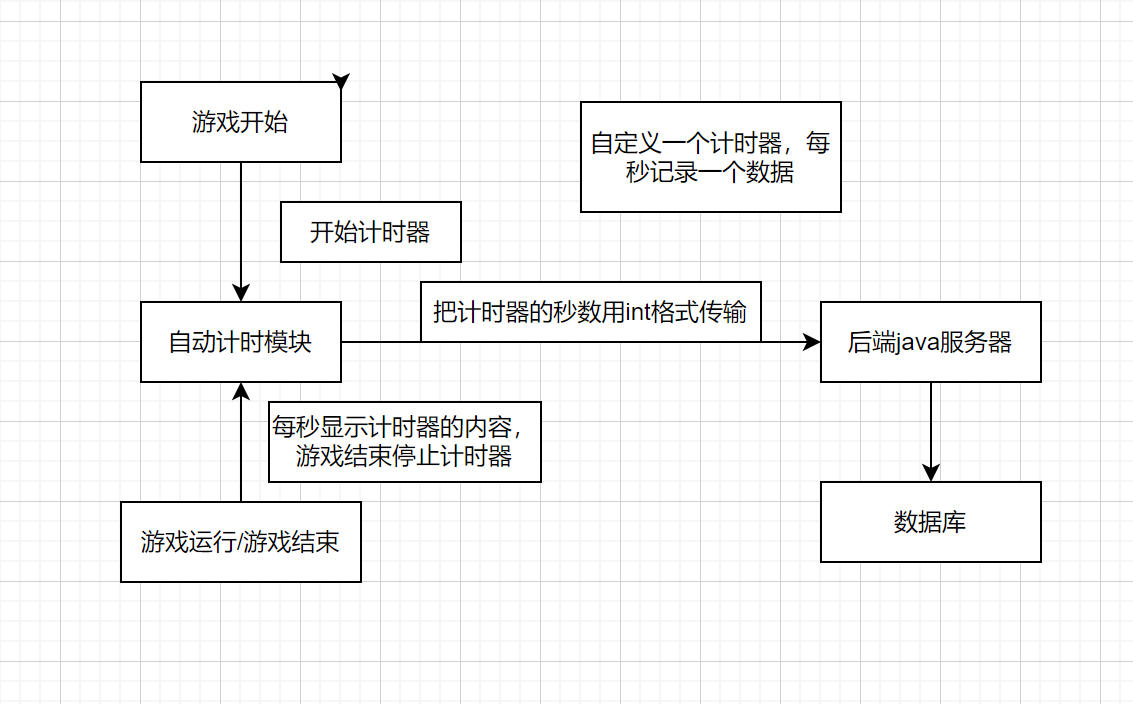
暂无。

# 6 自动计时模块设计说明

## 6.1程序描述

该自动计时模块的设计主要是为了向排行榜模块设计服务，用于统计用户游玩的时间，是非常驻内存，只在游戏运行的时候运行这个模块，玩家游戏游玩结束，该模块就不运行了，是属于游戏主程序的子程序，没有覆盖需求，属于并发处理，当用户一边玩游戏的时候，一边运行这个模块，等到游戏游玩结束，这个模块会把数据记录下来，然后然后显示在屏幕上，最后录入排行榜当中。

## 6.2功能



## 6.3性能

记录的时间是要精确到秒的，包括开始的时间和结束的时间各个怪物的生成时间，然后进行计算并给返回值。

记录时间的秒数的误差不能大于1s，因为这里面有排行榜，如果误差过大会影响用户数据的比较。

## 6.4输入项

输入项一共有两个：

开始游戏

当玩家点击开始游戏时，开始事件触发，计时器开始启动

游戏结束

当玩家阵亡，结束事件触发，计时器关闭

怪物生成

怪物生成计算怪物位置

## 6.5输出项

游戏耗时

把计时器记录的时间转化成int格式，然后传递给数据库

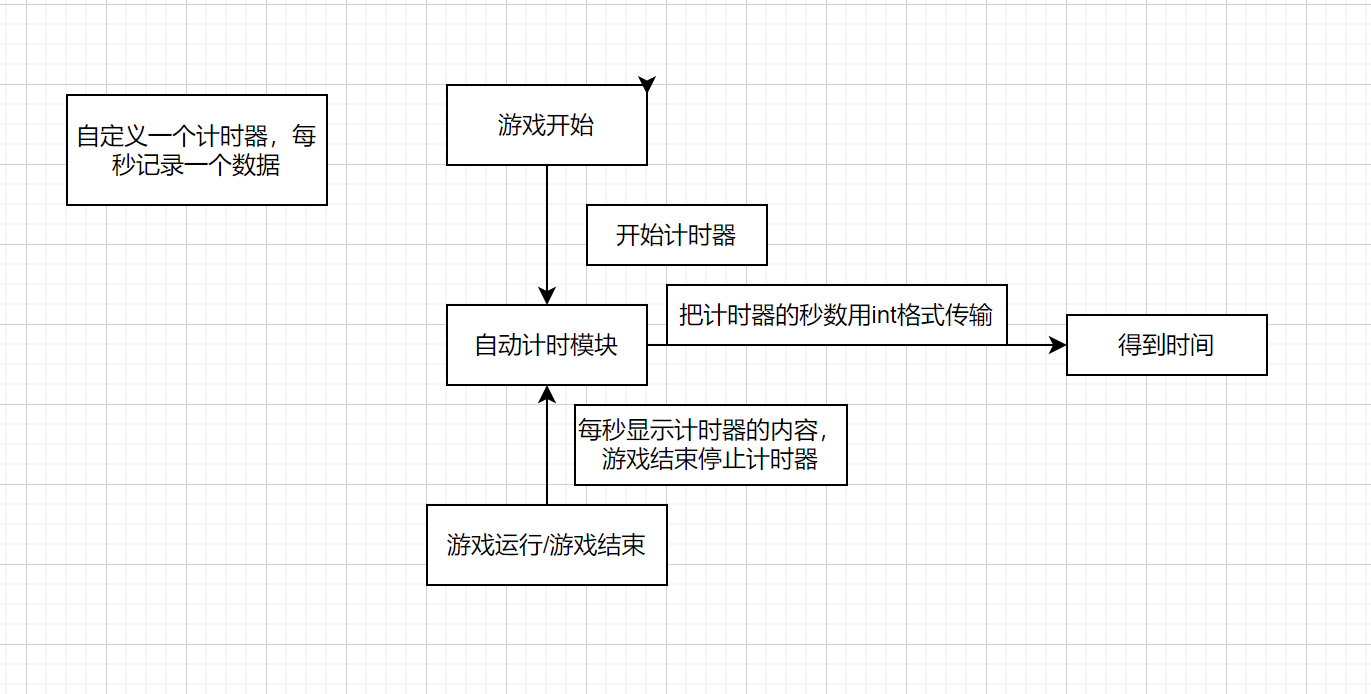
怪物位置

又时间来与怪物设定速度得出返回值

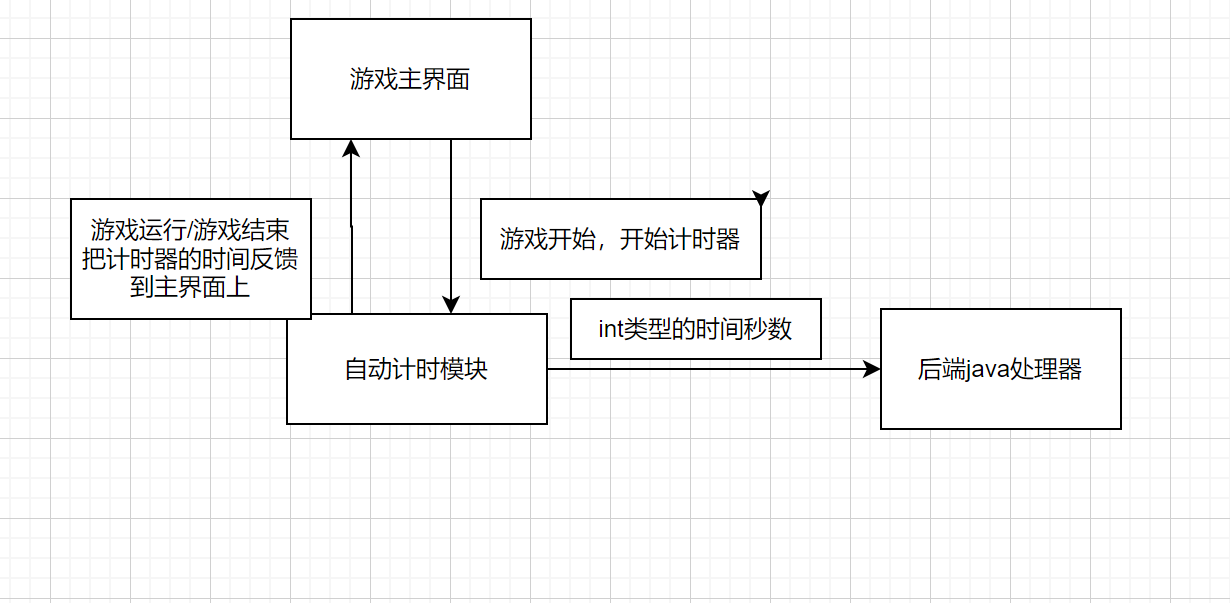
## 6.6算法

Java.util.time类

## 6.7流程逻辑



## 6.8接口



## 6.9存储分配

因为是自动计时模块，所要存储的数据全都是在服务器上部署运行的。

5.10注释设计

自动计时模块的注释

## 6.11限制条件

Int类型的范围

## 6.12测试计划

主要测试的方向有以下：

1：正常进行游戏，看计时器是否能正常跳转，看看有没有误差，误差是否在1s以内。

2：测试极端情况，测试上限是多少秒，如果一直运行准确度会不会变化，测试下限是多少，会不会出现负数的情况。

说明对本程序进行单体测试的计划，包括对测试的技术要求

## 6.13尚未解决的问题

..无

# 7排行榜设计说明

## 7.1程序描述

排行榜记录玩家的游戏得分并降序排列给出排名

## 7.2功能

排行榜记录玩家的游戏得分并降序排列给出排名

## 7.3性能

排行榜记录游戏的时间和开始的具体时间以及当局游戏的得分

排行榜记录游戏得分及其账号

## 7.4输人项

游戏数据

## 7.5输出项

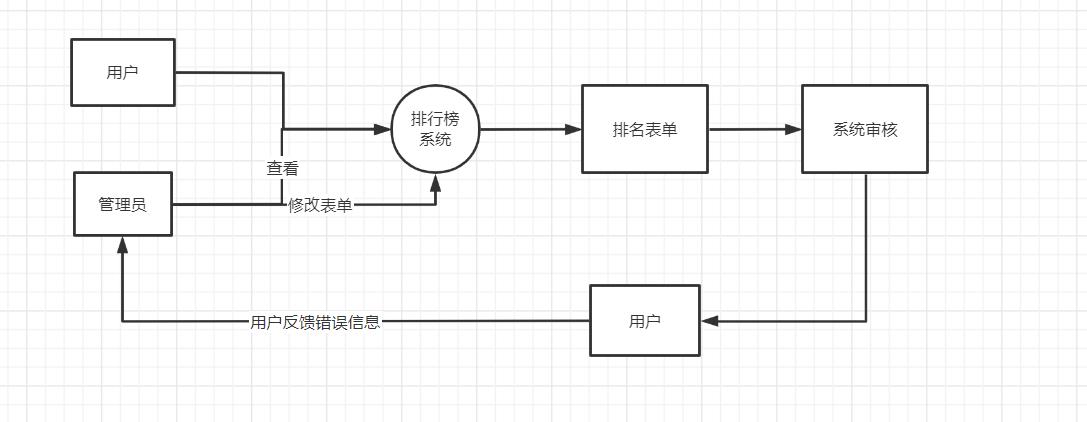
榜单excel文件

## 7.6算法

Sql查找数据并排序

## 7.7流程逻辑

## 7.8接口



Sql排序查询接口

## 7.9存储分配

无

## 7.10注释设计

无

## 7.11限制条件

数据库内存大小

## 7.12测试计划

1.进行多次游戏并查看排行榜

2.多个电脑查看排行榜

3.输入多组游戏数据并更新排行榜

## 7.13尚未解决的问题

无