# 1 绪论

## 1.1电子游戏的历史与发展简述

如今电子游戏已经成为年轻一代人不可或缺的娱乐项目，乃至对世界的经济，生活等各个方面产生了十分广泛和深远的影响。

电子游戏在1970年代开始以一种商业娱乐媒体被引入，成为1970年代末日本、美国和欧洲一个重要娱乐工业的基础。在1983年美国游戏业萧条事件及继而重生后的两年，电子游戏工业经历了[超过](https://baike.baidu.com/item/%E8%B6%85%E8%BF%87/7836274)两个年代的增长，成为了达100亿美金的工业，并与电影业竞争成为世界上最获利的娱乐产业。电子游戏早期是以主机运算、图形性能、以及主要储存媒介为世代区分标准。平均大约一个世代历时五至六年。世代之间的游戏机性能差别很大。

现如今，只要能掌握一定的编程技术，每个人都能成为一个游戏制作者，这使游戏制作变得不那么遥远，不那么复杂，一个只有几人的工作室也能做出令人喜爱的游戏来。当然，游戏模式类型多种多样，ACT，STG，FTG，AVG，RPG等等。若是想让游戏顺利运行，夺得更多玩家的青睐，每个游戏都需要很强大的后运算和后台管理。游戏不仅在剧情和操作上很重要，后台的优化也是必不可少的。

## 1.2课题的提出

目前市面上优秀的游戏太多太多，作为一组两个人的课程作业，我们尝试从最初级，最小型的游戏后台做起，来体验一个游戏后台管理系统所需的方方面面。并且从中来体现数据库的建表，联系，触发器，存储过程的知识点。因此要开发基于五子棋游戏的后台管理系统。其具体使用的技术有：最基本的网页语言HTML，Java web应用，数据库MySQL的应用，SQL语句的应用，Photoshop软件的操作，以及用于页面前端的JavaScript和CSS。

**1.2.1 Java web概述**

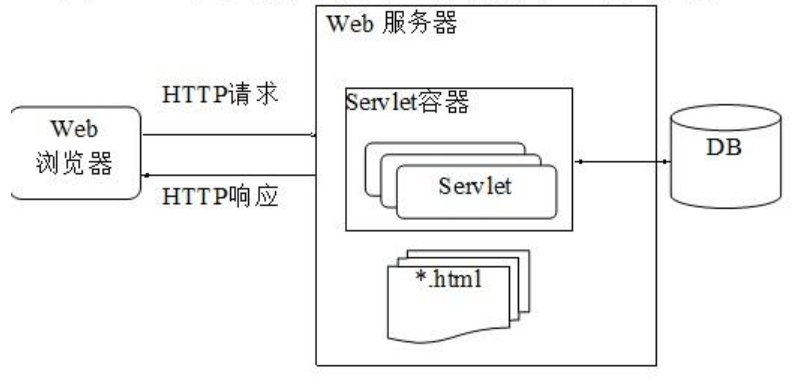
Java Web是用Java技术来解决互联网Web相关领域的技术总和，包括Web服务器和Web客户端两部分。Java在服务器端的应用非常丰富，比如Servlet、JSP和第三方框架等。Java技术对Web领域的发展注入了强大的动力。

**1.2.2 Servlet 概述**

Servlet可翻译成服务器端小程序，它是使用ServletAPI以及相关的类编写的Java程序，这种程序运行在Web容器中，主要用来扩展Web服务器的功能。Servlet自从1997年出现以后，由于所具有的平台无关性、可扩展性以及能够提供比CGI脚本程序更优越的性能等特征，使它的应用得到了快速的增长，并成为JavaEE平台的关键组件。Servlet技术实际上是CGI技术的一种替代。

**1.2.3 Web容器**

Web服务器使用一个单独的模块装载和运行Servlet与JSP页面，这个模块称为Servlet容器，或称Web容器。Tomcat就是一个Web容器，它在整个Web应用系统中处于中间层的地位，如图1-1所示



**图1-1 Web组件示意图**

图中给出Web应用系统的各种不同的组件构成，其中HTML文件存储在文件系统中，Servlet和JSP运行在Web容器中，业务数据存储在数据库中。

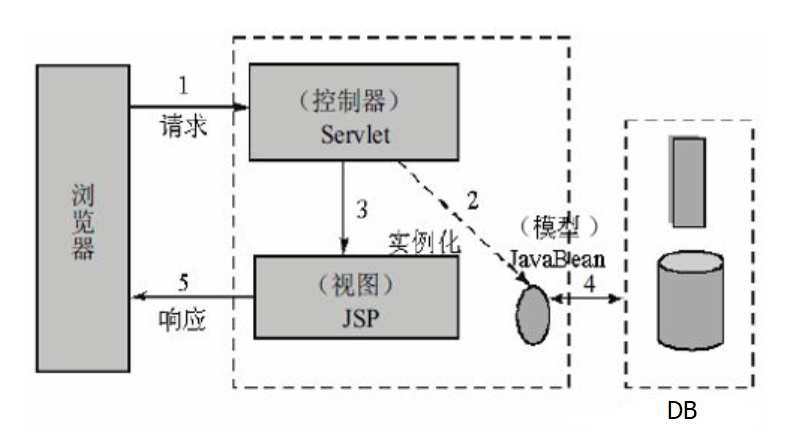
浏览器向Web服务器发送请求。如果请求的目标使HTML文件，Web服务器可以直接处理。如果请求的是Servlet或JSP页面，Web服务器将请求转发给Web容器，容器将查找并执行该Servlet或JSP页面，Servlet和JSP页面都可以产生动态输出。Tomcat作为一个Web容器可以有三种运行方式：独立运行的、进程内运行的和进程外运行的。独立运行方式是指Tomcat可以既作为Web服务器又作为Web容器运行。在这种情况下，整个Tomcat服务器包含两个模块：主Web服务器和Web容器。主Web服务器用来处理对静态内容的请求，Web容器用来处理对Servlet和JSP页面的请求。

**1.2.4 JSP页面**

JSP页面是在HTML页面中嵌入JSP元素的页面，这些元素成为JSP标签。JSP元素具有严格定义的语法并包含完成各种任务的语法元素，比如声明变量和方法、JSP表达式、指令和动作等。因此，JSP页面是一个由主动的JSP标签和被动的HTML标签混合而成的Web页面。在运行时，Web容器将JSP页面转换成Servlet类执行后将结果发送给客户。与其他的Web页面一样，JSP页面也有一个唯一的URL，客户可以通过它访问该页面。

**1.2.5 MVC设计模式**

在MVC设计模式的结构中，将Web组件分为模型、视图和控制器，每种组件完成各自的任务。在这种结构中所有请求的目标都是Servlet或Filter，它充当应用程序中的控制器。Servlet分析请求并将相应所需要的数据收集到JavaBeans对象或POJO对象中，该对象作为应用程序的模型。最后，Servlet控制器将请求转发到JSP页面。这些页面使用存储在JavaBeans中的数据产生相应。JSP页面构成了应用程序的视图。



**图1-2 MVC设计模式结构**

该模型的最大优点是将业务逻辑和数据访问从表示层里分离出来。控制器提供了应用程序的单一入口点，它提供了较清晰的实现安全性和状态管理的方法，并且这些组件可以根据需要实现重用。然后根据客户的请求，控制器将请求转发给合适的表示组件，由该组件来响应客户。这使得Web页面开发人员可一直关注数据的表示，因为JSP页面不需要任何复杂的业务逻辑。

## 1.3论文的组织结构

论文在第一章介绍了电子游戏的历史与发展，介绍了现今游戏必要的需求，介绍了建设后台管理所需要的基本技术，在第二章中描述了后台管理系统的整体结构，建立的数据库具体内容和约束，给出数据字典，第三章详细介绍每个模块实现所用到的技术，介绍了如何运用这些技术来实现各模块的功能。在后面的第四章中，介绍了本系统的配置和操作。在最后的第五章中，总结了这篇论文和不足之处。论文章节结构图如图1-3所示。

**图1-3 论文章节结构图**

# 2 后台管理体系结构及数据库

## 2.1功能分析

游戏后台管理系统主要提供如下功能（如图）。

**图2-1主要功能**

1. 玩家信息管理。提供玩家信息基本录入，维护与查询功能，包括

* 玩家录入，更新自身基本信息；
* 管理员查询，删除玩家基本信息

1. 规则管理。提供结算分数方法的制定，玩家分数的计算功能，包括

* 管理员制定，更新结算分数的规则；

1. 战绩表管理。提供查询

* 管理员删除玩家战绩表
* 所有用户可查询结算的积分，段位，排行

1. 留言管理。提供留言查询，回复留言，删除留言功能，包括：

* 管理员删除玩家留言
* 玩家回复玩家留言
* 所有用户可查询玩家留言；

1. 玩家权限管理。提供玩家管理，登录及权限验证等功能，包括：

* 管理员删除用户
* 玩家登录及权限验证
* 管理员更改用户密码

## 2.2数据需求及业务规则分析

数据库的数据需求可以根据与用户的交流和设计者自己对企业或组织的业务分析得到。我们设计的以五子棋为例的游戏后台的系统数据需求及业务规则分析如下：

(1)一个游戏可以由很多玩家参与，**玩家**由玩家编号唯一标识，玩家在首次进入游戏时需要注册，在数据库中存储玩家编号、用户名、密码、注册日期、邮箱等信息。

(2)**成绩**由成绩编号唯一标识，需记录棋局编号、玩家编号、对局时间、持续时间、胜负、步数等信息。一个玩家可以有多个成绩，一条成绩只能属于一个玩家。

(3)**积分**由积分编号唯一标识，需记录棋局编号、玩家编号、对局时间、胜负、分数等信息。一条积分只能对应一条成绩，一条成绩只能生成一条积分。

(4)**战绩**由玩家编号唯一标识，需记录玩家编号、段位、总分、胜利数、失败数等信息。其中，段位、积分、胜利数、失败数为派生属性，它的值可通过统计积分中的分数与胜负而得到。段位可通过段位规则而决定。

(5)**留言**由留言编号唯一标识，需记录留言编号、用户名、评论标题、评论内容、评论时间等信息。一个玩家可以发表多条留言，一条留言只能属于一个玩家。

(6)**回复**由回复编号唯一标识，需记录回复编号、留言编号、用户名、回复内容、回复时间等信息。一个玩家可以发表多个回复，一个回复只能属于一个玩家。一条评论可以对应多条回复，一条回复只能对应一条评论。

(7)**管理员**由管理员编号唯一标识，需记录管理员编号、管理员密码、邮箱等信息。

(8)**段位规则**由段位唯一标识，需记录段位、起始积分、结束积分等信息。

(9)**积分规则**由积分唯一标识，需记录积分、胜负、起始步数、结束步数等信息。

(10)若评论与回复中有不当言论，管理员可进行封号操作。

(11)管理员信息手动录入

## 2.3数据约束分析

(1)一次积分增减不超过15分。

(2)积分下限为-500分。

(3)胜负属性为W或L

(4)玩家编号为自增长属性

(5)用户名为唯一属性

(6)棋局编号为自增长属性

(7)留言序号为自增长属性

(8)回复序号为自增长属性

## 2.4数据库ER图

## 2.5数据字典

实体集：

1. 玩家（user）实体集。其属性有玩家序号（UserID）、昵称（Nickname）、密码（Password）、注册日期（RegisterDate）、邮箱（Email）。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 实例 |
| ***UserID*** | 玩家序号 | 主码 | int（11），代表玩家，序号自增且唯一。不允许空值 | 1 |
| Nickname | 玩家昵称 |  | varchar（12），名字唯一。不允许空值 | Rival |
| Password | 玩家密码 |  | varchar（16），不允许空值 | 123456 |
| RegisterDate | 注册日期 |  | datetime | 2018-11-20 20:32:46 |
| Email | 注册邮箱 |  | varchar（30）要包含“@”，“.com”字符 | aa@163.com |

1. 成绩（score）实体集。其属性有棋局号（SID）、下棋时间（Time）、输赢结果（WorL）、步数（Step），下棋日期（Date）、单局积分（Score）、玩家序号（UserID）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 实例 |
| **SID** | 棋局序号 | 主码 | int（11），代表棋局，序号自增且唯一。不允许空值 | 2 |
| Time | 下棋时间 |  | double（6），不允许空值 | 42.65 |
| WorL | 输赢结果 |  | char（1），取值范围{‘L’,‘W’}。不允许空值 | W |
| Step | 步数 |  | int（3）。不允许空值 | 20 |
| Date | 下棋日期 |  | datetime | 2018-11-20 20:48:26 |
| Score | 单局积分 |  | int（2），取值范围[-30,30]。不允许空值 | 15 |
| *UserID* | 玩家序号 | 外码 | int（11），代表玩家。不允许空值 | 1 |

1. 战绩（all1）实体集。其属性有玩家序号（UserID）、排名（Rankname）、总积分（SC），赢局次数（COUNTW）、输局次数（COUNTL）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 实例 |
| ***UserID*** | 玩家序号 | 主码，外码 | int（11），代表玩家，序号唯一。不允许空值 | 1 |
| *Rankname* | 段位 | 外码 | varchar（255） | 小有成就 |
| SC | 总积分 | 派生 | int（11），取值范围[-500,+ ∞），从成绩实体集统计得到。不允许空值 | 65 |
| COUNTW | 赢局次数 | 派生 | int（11），从成绩实体集统计得到。不允许空值 | 10 |
| COUNTL | 输局次数 | 派生 | int（11），从成绩实体集统计得到。不允许空值 | 2 |

1. 留言（messege）实体集。其属性有留言序号（QID）、留言人（Nickname），问题（Question）、具体内容（Text）、留言时间（Date）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 实例 |
| **QID** | 留言序号 | 主码 | int（11），代表问题，序号自增且唯一。不允许空值 | 15 |
| Nickname | 留言人 |  | varchar（12）。不允许空值 | Rival |
| Question | 问题 |  | varchar（30）。不允许空值 | 玩家登陆问题 |
| Text | 具体内容 |  | varchar（200）。 | 玩家登陆需要注册吗？ |
| Date | 留言时间 |  | datetime | 2018-11-20 20:48:26 |

1. 管理员（admin1）实体集。其属性有管理员序号（ADID）、昵称（ADname）、密码（ADpassword）、邮箱（ADemail）。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 实例 |
| **ADID** | 管理员序号 | 主码 | int（11），代表管理员，序号自增且唯一。不允许空值 | 3 |
| ADname | 管理员名字 |  | varchar（12），名字唯一。不允许空值 | ad1 |
| ADpassword | 管理员密码 |  | varchar（16）。不允许空值 | 123456 |
| ADemail | 管理员邮箱 |  | varchar（30）要包含“@”，“.com”字符 | aa@163.com |

1. 段位（rank1）实体集。其属性有起始积分（Startpoint）、终止积分（Endpoint）、段位（Rankname）。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 实例 |
| ***Rankname*** | 段位 | 主码，外码 | varchar（255），段位唯一。不允许空值 | 小有成就 |
| Startpoint | 起始积分 |  | int（11）。不允许空值 | 50 |
| Endpoint | 终止积分 |  | int（11）。不允许空值 | 100 |

1. 步数规则（rulest）实体集。其属性有步数规则序号（PID）、起始步数（StartStep）、终止步数（Endstep）、加分（eachS）。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 实例 |
| eachS | 步数加分 |  | int（2）取值范围[-15, 15]。不允许空值 | 15 |
| StartStep | 起始步数 |  | int（3）。不允许空值 | 0 |
| Endstep | 终止步数 |  | int（3）。不允许空值 | 15 |
| **PID** | 步数规则序号 | 主码 | int（2），代表步数规则，序号自增且唯一。不允许空值 | 1 |

1. 输赢规则（rules）实体集。其属性有输赢结果（WorL）、加分（each）。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 实例 |
| each1 | 输赢加分 |  | int（2）取值范围[-15,15]。不允许空值 | 15 |
| **WorL** | 输赢结果 | 主码 | char（1），取值范围{‘L’,‘W’}。不允许空值 | W |

联系集：

1. 发布（release）联系集。它是玩家实体集和留言实体集之间的一对多联系集，没有联系属性。
2. 处理（process）联系集。它是管理员实体集和玩家实体集之间的一对多联系集，没有联系属性。
3. 管理（manage）联系集。它是管理员实体集和留言实体集之间的一对多联系集，没有联系属性。
4. 拥有（own）联系集。它是玩家实体集和战绩实体集之间的一对一联系集，没有联系属性。
5. 下棋获得（get）联系集。它是玩家实体集和成绩实体集之间的一对一联系集，没有联系属性。
6. 改变（change）联系集。它是成绩实体集和战绩实体集之间的多对一联系集，没有联系属性。
7. 划分（identify）联系集。它是段位实体集和战绩实体集之间的一对一联系集，没有联系属性。
8. 回复（respond）联系集。它是玩家实体集和留言实体集之间的一对多联系集，其属性有回复序号（RID）、留言序号（QID）、回复人（Nickname）、回复内容（Answer）、回复时间（RDate）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 含义 | 类别 | 域及约束 | 实例 |
| **RID** | 回复序号 | 主码 | int（11），代表回复，序号自增且唯一。不允许空值 | 12 |
| Nickname | 回复人 |  | varchar（11） | ZX |
| Answer | 回复内容 |  | varchar（200） | 是的，需要先注册。 |
| RDate | 回复时间 |  | datetime | 2018-11-21 20:48:26 |
| *QID* | 留言序号 | 外码 | int（11），代表问题。不允许空值 | 15 |

## 2.5