

课程设计报告

**2023—2024学 年 第 二 学 期**

**《软件工程课程设计》报告**

专业班级 计算2101

姓 名 许祖耀

学 号 2107010120

开课系室 软件工程

考核日期 2024年5月21日

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 报告内容 | 需求分析 | 概要设计 | 详细设计 | 测试 | 工具使用 | 总分 |
| 得 分 | 0 | 0 | 0 | 0 | 0 | 0 |
| 阅卷人 |  | | | | | |

|  |  |
| --- | --- |
| 需求分析评语 |  |
| 概要设计评语 |  |
| 详细设计评语 |  |
| 软件测试评语 |  |
| 工具使用评语 |  |

[《软件工程课程设计》报告 0](#_Toc14974)

[一、 前言 3](#_Toc12343)

[1.项目描述： 3](#_Toc13401)

[2.项目背景： 3](#_Toc3582)

[3.项目目的： 4](#_Toc19745)

[二、需求分析 6](#_Toc28913)

[1.功能需求 6](#_Toc20872)

[2.性能需求： 9](#_Toc15516)

[3.环境需求： 10](#_Toc16901)

[4.接口需求： 10](#_Toc22094)

[5.用户界面需求： 11](#_Toc658)

[6.数据需求： 12](#_Toc32348)

[二、 逻辑模型 15](#_Toc5485)

[1.基础应知： 15](#_Toc19545)

[2.数据流图的生成 15](#_Toc8423)

[3.数据字典 22](#_Toc11835)

[4.加工条目 27](#_Toc16503)

[三、 概要设计： 33](#_Toc2439)

[1.系统结构划分 33](#_Toc25527)

[2.系统初始结构图 34](#_Toc22265)

[四、 详细设计 39](#_Toc20978)

[1.模块流程图 39](#_Toc5182)

[2.模块N-S盒图 44](#_Toc24421)

[六、系统测试： 46](#_Toc1215)

[1. 白盒测试： 47](#_Toc2415)

[2. 黑盒测试： 54](#_Toc16787)



# 前言

## 1.项目描述：

Steam游戏平台是由Valve Corporation开发的，面向PC游戏玩家的数字发行平台和社交网络。它于2003年推出，最初是作为解决游戏更新和维护的平台，但随着时间的推移，它发展成为一个庞大的数字游戏分发平台和社交网络，并成为全球最大的综合性数字发行平台之一，全球玩家可以在该平台购买、下载、讨论、上传和分享游戏和软件。

## 2.项目背景：

2000年代是游戏产业的快速发展时期。在这个时期，随着计算机技术的进步和互联网的普及，游戏行业进入了一个新的黄金时代。传统的游戏发行模式逐渐失去效益，数字发行成为一种新兴的趋势。Valve公司在这一背景下意识到了数字发行的潜力，并决定创建一个专门的平台来满足玩家和开发者的需求，于是Steam应运而生，并迅速发展为全球最大的数字游戏分发平台，服务着千万级别的来自全球各地的游戏玩家。

## 3.项目目的：

Steam游戏平台目的是为玩家和游戏行业相关从业人员提供功能强大的一站式解决方案，向广大玩家和从业人员提供一个数字游戏分发平台，公司可以在其中发布宣传推广制作的游戏，而玩家可以在平台中购买游戏，并享受平台提供的技术支持与客户服务。

同时Steam游戏平台还为全球游戏玩家提供了一个大型游戏社交网络，全球玩家可以在Steam上游戏进行测评，发布与游戏相关的文章或动态，同时与其他玩家进行互动讨论等。

Steam的目标用户主要分为两类，一类是游戏玩家、社区成员、内容创作者等个人用户，Steam需要为个人用户提供优秀的游戏全流程体验，满足个人用户的娱乐需求；另一类则是游戏开发商、开发公司、独立开发者等企业用户，Steam需要满足企业用户的商业要求，为企业用户的商业活动、开发活动提供支持和保护，保障商业利益。

Steam为这两类不同用户群体需要解决不同的关键问题：

* 面向游戏玩家、社区成员、内容创作者：

1. **提供全面的游戏购买和下载功能**：Steam平台需要提供一个易于浏览和搜索的游戏商店界面，让玩家能够方便地找到和购买他们感兴趣的游戏，并提供稳定高效的下载服务，确保玩家能够快速下载所购游戏。
2. **提供个性化的游戏体验**：Steam需要根据玩家的游戏偏好和历史行为，提供个性化的游戏推荐和服务，帮助玩家发现新的游戏并提升他们的游戏体验。
3. **提供丰富的社交功能**：Steam需要提供强大的社交功能，如好友列表、聊天功能、游戏组队等，让玩家能够与朋友和其他玩家进行互动和交流，增强他们的社交体验。
4. **提供安全可靠的服务**：Steam需要提供安全可靠的购买、支付和账号管理服务，保护玩家的个人信息和账号安全，并采取有效的反作弊措施，保护游戏环境的公平性。
5. **提供优质的技术支持**：Steam需要提供及时有效的技术支持服务，帮助玩家解决在使用平台或游戏中遇到的问题和困难，确保他们能够顺利地享受游戏。

* 面向游戏开发商、开发公司、独立开发者：
  1. **提供简单易用的游戏上传和发布流程**：Steam需要提供一个直观且易于理解的界面，使游戏制作者能够轻松上传和发布他们的游戏。
  2. **提供透明且公正的销售和分成政策**：Steam需要明确说明销售和分成政策，确保游戏制作者能够清晰地了解他们在平台上的收入情况，并获得公平的待遇。
  3. **提供专业的发行服务和支持：**Steam需要与游戏发行商合作，提供专业的发行服务和支持，帮助他们成功地将游戏发布到Steam平台上，并制定有效的营销策略。
  4. **提供丰富的销售数据和分析工具：**Steam需要提供详尽的销售数据和分析工具，让游戏发行商能够全面了解他们在平台上游戏的销售情况，并根据数据制定相应的销售策略。
  5. **提供定制化的合作方案和推广支持：**Steam需要与游戏发行商合作，提供定制化的合作方案和推广支持，帮助他们最大限度地提升游戏在平台上的曝光度和销售量。

# 二、需求分析

1.功能需求：

Steam作为一个数字发行平台和社区，旨在提供全面的游戏购买、下载、社交和体验服务，满足广大游戏玩家及开发者的需求；为实现这一目标，Steam需要具备以下七大功能需求：

1. 数字游戏分发平台： Steam游戏平台的首要目标是作为一个数字游戏分发平台，为开发者提供一个渠道，使其能够将游戏直接提供给全球玩家。通过Steam，开发者可以实现更广泛的游戏发行，不受地理位置和传统零售渠道的限制，而全球玩家则可以通过Steam浏览到来自世界各地的开发者所创造的电子游戏，并使用Steam简单便捷地购买、下载、游玩游戏。



图 1 Steam的商店页面

1. 游戏库存管理：Steam游戏平台需要记录管理玩家通过Steam平台所购买的所有游戏，提供直观易懂的菜单目录，分门别类列出玩家拥有的全部游戏，同时还需支持游戏的动态智能分类、玩家手动分类等，同时Steam还需将玩家在游戏时产生的如存档、游玩时长等的个人数据储存在云端服务器，实现玩家数据的多端同步。

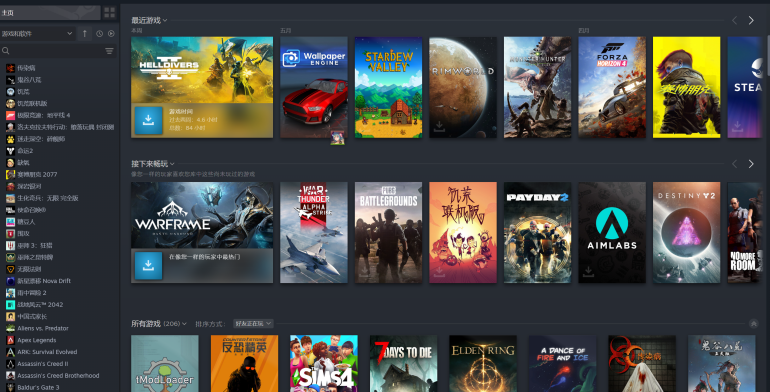


图 2 Steam的玩家库存页面

1. 大型游戏社交网络：除此之外，Steam游戏平台还需可以作为一个庞大的游戏社交网络，使玩家能够与好友交流、加入游戏群组、分享游戏内容和成就，增强游戏的社交体验。



图 3 Steam的社区活动页面

1. 提供游戏技术支持与服务：Steam平台需要为其所分发的游戏提供技术支持，帮助用户下载和更新游戏，并提供游戏社区、游戏工具、游戏媒体、等相关服务。

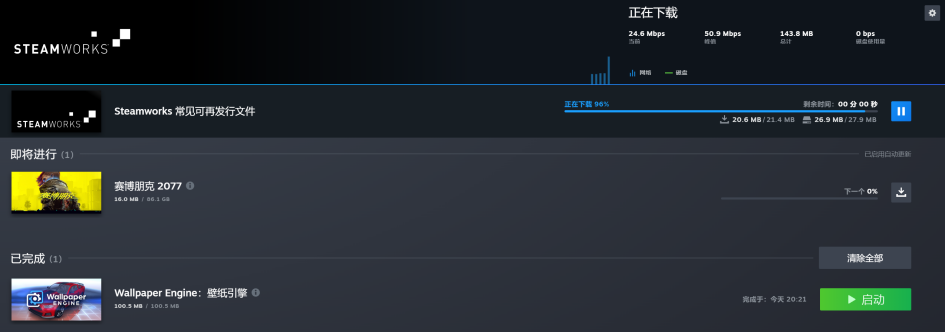


图 4 Steam的游戏下载页面

1. 游戏发行渠道：Steam同时需要为全球的广大开发商、开发公司、独立开发者提供一个安全可靠的数字游戏发行渠道。通过数字版权技术和反盗版措施，保障游戏内容的安全性，使开发者能够放心将游戏发布在平台上，并依法保护开发者的权益及利益。
2. 游戏模组分享：游戏模组是由独立的第三方开发者，一般是具有一定编程能力的游戏爱好者，所制作的修改或增强游戏内容的外部游戏文件。Steam通过其特色的“创意工坊”，允许用户上传自己独立制作的游戏模组，并分享给全球玩家进行下载，并通过评论评分功能获得来自其他玩家的反馈



图 5 Steam的创意工坊页面

1. 游戏物品交易：Steam也为虚拟的游戏物品提供了线上交易平台，玩家可以通过Steam所提供的“社区市场”，进行游戏中虚拟物品的交易、互换。同时Steam将通过图表形式显示交易物品的市场价格波动，并提供了一系列交易规则，保障玩家利益。

|  |  |  |
| --- | --- | --- |
|  |  |  |

图 6 Steam的社区市场页面（组图）

为了完成上述功能需求，Steam游戏平台主要划分为了商店模块、库存模块、社区模块、客服帮助模块、个人资料模块、设置模块等六大功能模块，商店功能模块向玩家以及游戏相关从业人员提供数字游戏分发功能，而社区功能模块则向全球玩家提供大型的游戏社交网络，供用户之间相互互动交流，建立联系。Steam游戏平台旨在为所有游戏爱好者提供一站式的解决方案。

## 2.性能需求：

（1）Steam游戏平台应能够支持百万款游戏数据的存储与分发，和千万计用户数据的存储，适应不同用户的游戏需求。

（2）严格限制：Steam游戏平台的并发用户数应不低于1000万，以确保平台的稳定性和可用性

（3）在Steam游戏平台中，搜索功能的响应时间不得超过2秒，以提供良好的用户体验。

（4）本系统的输入项至少包括：游戏上传和更新、用户评论和评分、游戏推荐算法等。

（5）本系统的输出项至少包括：游戏列表展示、用户评论展示、游戏推荐结果、二进制可执行游戏文件等。

（6）Steam游戏平台应支持多平台使用，包括但不限于Windows、macOS、Linux操作系统。

（7）平台的网络环境应具备高效稳定的连接，确保全球范围内的用户都能流畅地进行游戏下载、更新和在线互动。

（8）平台应提供数据备份和恢复机制，确保用户数据和游戏进度的安全。

## 3.环境需求：

Steam游戏平台的界面设计应支持多种语言，包括中文、英文等，以服务全球用户，平台界面也应该新潮科技，以提高游戏玩家的使用体验，同时需要具备高可用性,交互逻辑清晰明确，使不同年龄段的用户均可无障碍使用平台，满足不同用户的需求。

## 4.接口需求：

Steam游戏平台所需要的外部接口如下：

第三方游戏开发者接口：

提供API供游戏开发者上传和管理他们的游戏。

支持开发者获取游戏销售数据和用户反馈。

内容审核接口：

提供用户反馈和客服支持渠道。

实现工单系统和常见问题解答（FAQ）。

外部支付接口：

实现安全的支付流程，支持多种支付方式。

提供交易历史记录和发票生成功能。

## 5.用户界面需求：

Steam游戏平台的用户界面应设计得直观、易用，同时提供必要的功能和信息，以提升用户体验。界面应遵循平台的设计指南，同时考虑到不同设备和分辨率下的适配性。具体来说，用户界面应包含以下功能：

主页界面显示热门游戏、最新游戏、特别优惠和推荐游戏，并提供搜索栏；游戏详情界面显示游戏的描述、评分、评论、系统要求和截图，并提供购买、添加到愿望单或赠送游戏作为礼物的选项；

用户个人资料界面显示用户的游戏库、成就、好友列表和交易历史，允许用户编辑个人资料、上传头像和修改密码；

社交互动界面显示好友列表、当前在线好友和最近的游戏活动，并提供即时消息聊天窗口和游戏内邀请功能；

商店购物车界面显示用户已选游戏的列表、总价和购买选项，并提供优惠券应用和支付方式选择的界面；

客户支持界面提供常见问题解答（FAQ）、知识库和客服联系方式，显示用户工单状态和历史，以及提交新工单的表单；

设置和偏好界面允许用户配置游戏控制器、音频设置和显示选项，并提供语言选择和区域设置选项。

## 6.数据需求：

为了能够准确的获取Steam游戏平台的数据需求，下面将采用E-R图的形式对需求进行分析和描述。具体分析如下图所示：

|  |
| --- |
| Steam游戏平台的整体E-R图 |
| All-er |

Steam游戏平台的整体E-R图中包含大量实体和联系，需依据逻辑对ER图进行分解，绘制相应的局部E-R图，局部E-R图如下所示：

|  |
| --- |
| 以用户为中心的局部E-R图 |
| user-er |
| 以个人资料为中心的局部E-R图 |
| profile-er |
| 以游戏为中心的局部E-R图 |
| game-er |
| 以社区内容发布表为中心的局部E-R图 |
| community-er |

# 逻辑模型

## 1.基础应知：

：箭头，表示数据流；

：圆或椭圆，表示变换数据的处理；

：方框，表示数据的三原点或者终点；

：双杠或者单杠，表示数据存储文件。

源点和终点是系统之外的实体，表明数据的来源和去向；

加工是对数据处理的单元，通常为动词短语；

数据流是数据在系统中运动的方向，通常为名词或者名词短语；

数据存储是暂时存储数据的。

## 2.数据流图的生成

（1）确定系统实体及加工：

实体：游戏玩家，内容创作者，社区成员，游戏发行商；

加工主要分为：游戏分发，游戏下载，游戏发布，社区内容发布，社区反馈，模组上传，模组社区反馈，游戏推广

（2）数据流图：

#### 第0层

|  |
| --- |
|  |
| 图 2.1 Steam游戏平台的顶层数据流图 |

#### 第1层

如下图所示：

对于Steam游戏平台进行进一步划分：

|  |
| --- |
|  |
| 图 2.2 Steam游戏平台1层数据流程图 |

#### 第2层

|  |
| --- |
| 加工1子图 |
|  |
| 加工2子图 |
|  |
| 加工3子图 |
|  |
| 加工4子图 |
|  |
| 加工5子图 |
|  |
| 加工6子图 |
|  |
| 加工7子图 |
|  |
| 加工8子图 |
|  |
| 加工9子图 |
|  |
| 加工10子图 |
|  |
| 加工11子图 |
|  |
| 加工12子图 |
|  |
| 加工13子图 |
|  |
| 加工14子图 |
|  |
| 图 2.3 Steam游戏平台2层数据流程组图 |

如下图所示：

## 3.数据字典

（1）数据流条目

账号信息 = 用户名 + 密码

支付凭证 = 收据单号 + 支付时间 + 凭证验证信息

下载请求 = 用户名 + 游戏编号 + 请求时间

购买请求 = 用户编号 + 游戏编号 + 请求时间

聊天消息 = 用户编号 + 目标用户编号 + 消息内容 + 发送时间

上传社区内容 = 用户编号 + 社区内容编号 + 社区内容文本信息 + 社区内容文件 + 上传时间

游戏文件 = 二进制可执行文件 + 静态资源文件 + 模组文件 + 玩家数据文件

商店销售信息 = 游戏名称 + 游戏版本售价 + 游戏简介 + 游戏评价 + 游戏类型 + 游戏开发商名称 + 游戏发行商名称 + 游戏介绍 + 游戏发布日期

聊天记录信息 = 用户编号 + 消息内容 + 发送时间

审核信息 = 审核是否通过 + 审核意见 + 审核时间

待审核内容 = 待审核文本内容 + 待审核数据文件 + 待审核内容标识字段 + 请求审核时间

销售数据信息 = 社区总体评价信息 + 销售量 + 销售金额 + 销售图表 + 销售时长

游戏作品 = 游戏名称 + 二进制可执行文件 + 静态资源文件 + 游戏简介 + 游戏介绍 + 游戏类型 + 游戏文档

发布方案 = 游戏名称 + 游戏销售策略 + 游戏社区初始内容 + 发布时间 + 预售时间 + 发售时间

公告内容 = 发行商名称 + 公告编号 + 公告标题 + 公告文本内容 + 发布时间

促销策略 = 促销活动名 + 游戏名称 + 折扣力度 + 折扣起始时间 + 折扣结束时间

社区评价 = 用户名 + 游戏名称 + 是否推荐该游戏 + 评价文本内容 + 评价时的游戏时长 + 评价日期

总体评价 = 推荐评价总数 + 不推荐评价总数 + 近期推荐评价数 + 近期不推荐评价数

有效购买记录 = 用户名 + 游戏名称 + 购买请求 + 支付凭证

游戏入库信息 = 用户名 + 游戏名称 + 入库时间

模组内容 = 社区内容编号 + 模组简介 + 模组介绍 + 模组文件 + 模组上传日期

富文本内容 = 文本内容 + 图片 + 视频 + 链接

待审核社区内容 = 社区内容 + 审核标识号

库存信息 = 游戏名称 + 入库时间

玩家数据 = 用户名 + 游戏名称 + 存档文件 + 游玩时长 + 成就完成情况 + 订阅的社区内容

游戏数据 = 切分、压缩成组的游戏文件

模组文件 = 切分、压缩成组的模组文件

好友列表 = 用户名 + 用户昵称 + 用户个人资料 + 成为好友时间

个人资料信息 = 用户名称 + 用户地区 + 用户简介 + 用户介绍 + 用户头像 + 用户社区情况

待审核的社区内容 = 社区内容 + 审核标识号

审核通过的社区内容 = 社区内容 + 审核标识号 + 审核意见 + 审核通过与否

社区内容 = 用户编号 + 测评文本内容 + 发表时间

版本文件 = 游戏名称 + 二进制可执行程序 + 静态资源文件 + 版本标识号

产品信息 = 产品名称 + 产品售价 + 产品简介 + 产品评价 + 类型 + 开发商名称 + 发行商名称 + 产品介绍 + 产品发布日期

销售方式 = 游戏名称 + 游戏销售策略 + 发布时间 + 预售时间 + 发售时间

待审核方案 = 发布方案 + 审核标识号

（2）数据存储条目

文件名：玩家库存

组成：用户名 + 游戏名称 + 入库时间

组织方式：索引文件，以用户名和游戏名称为关键码

文件名：社区内容

组成：用户名 + 社区内容编号 + 社区内容文本信息 + 社区内容文件 + 上传时间

组织方式：索引文件，以社区内容为关键

文件名：游戏数据

组成：游戏名 + 游戏编号 + 分组压缩的游戏数据文件

组织方式：索引文件，以游戏编号为关键

文件名：游戏模组

组成：模组名 + 社区内容编号 + 游戏名 + 游戏编号 + 分组压缩的模组数 据文件

组织方式：索引文件，以社区内容编号为关键

文件名：社区内容

组成：内容标题 + 社区内容编号 + 游戏名 + 游戏编号 + 文本内容 + 二 进制文件内容 + 其他用户反馈文本 + 其他用户反馈数据

组织方式：索引文件，以社区内容编号为关键

（3）数据项

1. 游戏名

- 别名：无

- 类型：字符串

- 长度：不定长

- 描述：用于标识特定游戏的名称

2. 游戏编号

- 别名：无

- 类型：长整型

- 长度：根据实际编号系统确定（例如64位）

- 描述：唯一标识游戏的编号

3. 社区内容编号

- 别名：无

- 类型：长整型

- 长度：64位

- 描述：唯一标识社区内容的编号

4. 模组名

- 别名：无

- 类型：字符串

- 长度：不定长，根据实际模组名称长度确定

- 描述：用于标识特定游戏模组的名称

5. 内容标题

- 别名：无

- 类型：字符串

- 长度：不定长，根据实际标题长度确定

- 描述：社区内容的标题，用于概括内容主题

6. 文本内容

- 别名：无

- 类型：字符串

- 长度：不定长，根据实际文本长度确定

- 描述：社区内容的详细文本描述

7. 二进制文件内容

- 别名：无

- 类型：二进制文件

- 长度：不定长，根据文件大小确定

- 描述：存储的二进制文件，如图片、视频等多媒体内容

8.其他用户反馈文本

- 别名：无

- 类型：字符串

- 长度：不定长，根据反馈文本长度确定

- 描述：其他用户对社区内容的文本反馈

9. 其他用户反馈数据

- 别名：无

- 类型：长整型或字符串（根据反馈数据类型确定）

- 长度：根据数据类型确定（例如64位）

- 描述：其他用户对社区内容的反馈数据，可能包括评分、标签等

10. 用户编号

- 别名：无

- 类型：长整型

- 长度：根据实际编号系统确定（例如64位）

- 描述：唯一标识用户的编号

## 4.加工条目

加工名：Steam游戏平台

* + - 编号：0
    - 输入：游戏购买记录，独立游戏或游戏的扩展魔族，游戏评测、评论，游戏推广方案、大型游戏发布
    - 输出：可执行游戏文件，下载数据及社区反馈，社区用户反馈，销售数据

加工名：游戏分发

* + - 编号：1
    - 输入：游戏购买记录
    - 输出：游戏下载许可，游戏入库许可
    - 加工逻辑：

if 用户拥有对游戏的所有权 and 游戏购买记录正确，

then 生成游戏下载许可、游戏入库许可；

else 拒绝用户请求并发出提示；

endif

加工名：核验购买记录

* + - 编号：1.1
    - 输入：游戏购买记录
    - 输出：游戏下载请求、游戏购买请求
    - 加工逻辑：

if 购买记录完整

then 游戏下载请求、游戏购买请求；

else 终止操作并提示；

endif

加工名：确认购买记录

* + - 编号：1.2
    - 输入：游戏购买请求
    - 输出：游戏入库许可
    - 加工逻辑：根据游戏购买请求生成相应的游戏入库许可。

加工名：确认用户订阅的社区内容

* + - 编号：1.3
    - 输入：用户订阅信息
    - 输出：用户订阅的社区内容信息
    - 加工逻辑：根据用户的订阅信息检索社区内容并发送。

加工名：生成验证信息

* + - 编号：1.4
    - 输入：游戏下载请求、用户订阅的社区内容信息
    - 输出：游戏下载许可信息
    - 加工逻辑：根据游戏下载请求和用户订阅的社区内容信息生成对应的游戏下载许可信息。

加工名：游戏下载

* + - 编号：2
    - 输入：游戏下载许可信息
    - 输出：可执行游戏文件
    - 加工逻辑：根据游戏下载许可信息和用户数据确定最合适的传输服务器并传输分组压缩的游戏数据和社区内容数据，并在用户主机上实时合并解压生成可执行游戏文件。

加工名：分配下载服务器

* + - 编号：2.1
    - 输入：游戏下载许可
    - 输出：服务器信息
    - 加工逻辑：根据用户的账户信息，分配最适宜的下载传输服务器。

加工名：检索整理游戏数据

* + - 编号：2.2
    - 输入：服务器信息
    - 输出：游戏相关数据
    - 加工逻辑：根据服务器信息，从游戏数据服务器中检索游戏模组内容、游戏数据内容，并整理打包输出游戏相关数据。

加工名：传输数据

* + - 编号：2.3
    - 输入：游戏相关数据
    - 输出：可执行二进制游戏文件
    - 加工逻辑：从游戏数据服务器中将游戏相关数据传输至用户主机，并在传输过程中保持高速率，高稳定。

加工名：游戏发布

* + - 编号：3
    - 输入：游戏文件
    - 输出：发布确认
    - 加工逻辑：

if 游戏文件和元数据完整且符合标准，

then 将游戏上架并通知用户；

else 反馈错误信息并要求补充或更正；

endif

加工名：验证游戏合规性

* + - 编号：3.1
    - 输入：游戏相关数据
    - 输出：合规游戏文件、不合规游戏文件
    - 加工逻辑：

if 游戏文件符合标准规范

then 输出合规游戏文件

else 输出不合规游戏文件并提示

endif

加工名：保存游戏数据

* + - 编号：3.2
    - 输入：合规游戏文件
    - 输出：分组压缩的游戏数据
    - 加工逻辑：对合规游戏文件进行分组压缩并保存至相应游戏数据服务器中。

加工名：社区内容发布

* + - 编号：4
    - 输入：用户生成内容，用户信息
    - 输出：内容发布状态，用户反馈
    - 加工逻辑：

if 内容符合社区准则，

then 发布内容并通知用户；

else 拒绝发布并说明理由；

endif

加工名：审核评测评论

* + - 编号：4.1
    - 输入：游戏评测评论
    - 输出：通过审核的评测评论、未通过审核的评测评论
    - 加工逻辑：

if 用户的评测评论不违反规定 and 用户信誉良好，

then 输出通过审核的评测评论；

else 输出未通过审核的评测评论；

endif

加工名：上传至社区

* + - 编号：4.2
    - 输入：游戏评测、评论
    - 输出：规范化的社区内容
    - 加工逻辑：将通过审核的游戏评测、评论规范化，并上传至社区内容数据库中。

加工名：社区反馈

* + - 编号：5
    - 输入：用户反馈，相关社区内容
    - 输出：反馈处理结果，社区内容更新
    - 加工逻辑：

if 反馈合理且有建设性，

then 更新社区内容并感谢用户；

else 记录反馈并进行评估；

endif

加工名：确定社区内容

* + - 编号：5.1
    - 输入：社区内容编号
    - 输出：内容信息
    - 加工逻辑：根据输入的社区内容编号检索社区内容数据库，输出相对应的社区内容信息。

加工名：检索社区反馈

* + - 编号：5.2
    - 输入：内容信息
    - 输出：社区反馈信息
    - 加工逻辑：根据相应的社区内容信息检索社区内容中其他用户的评价与反馈，组成社区反馈信息并输出。

加工名：显示反馈内容

* + - 编号：5.3
    - 输入：社区反馈信息
    - 输出：社区内容反馈
    - 加工逻辑：加社区内容信息和社区反馈信息综合处理，输出社区内容反馈。

加工名：模组上传

* + - 编号：6
    - 输入：模组文件
    - 输出上传确认
    - 加工逻辑：

if 模组文件符合上传标准 and 上传者具有相应权限，

then 上传模组至游戏模组数据；

else 反馈错误信息并要求更正；

endif

加工名：验证模组合规性

* + - 编号：6.1
    - 输入：模组文件
    - 输出：合规游戏模组、不合规模组
    - 加工逻辑：

if 游戏模组符合标准规范

then 输出合规游戏模组

else 输出不合规模组并提示

endif

加工名：保存游戏模组

* + - 编号：6.2
    - 输入：合规游戏模组
    - 输出：分组压缩后的模组数据文件
    - 加工逻辑：将输入的合规游戏模组切分压缩，组织成规范化的数据文件并保存至游戏模组数据中。

加工名：模组社区反馈

* + - 编号：7
    - 输入：用户反馈，模组信息
    - 输出：反馈处理结果，模组更新状态
    - 加工逻辑：

if 用户反馈有效 and 模组存在可改进之处，

then 将反馈提交给模组开发者并记录；

else 记录无效反馈并进行后续评估；

endif

加工名：确定模组内容

* + - 编号：7.1
    - 输入：模组编号
    - 输出：模组信息
    - 加工逻辑：根据输入的模组编号检索游戏模组数据库，输出相对应的模组信息。

加工名：检索社区反馈

* + - 编号：7.2
    - 输入：模组信息
    - 输出：社区反馈信息
    - 加工逻辑：根据相应的模组信息检索社区内容中其他用户的评价与反馈，以及下载数据信息组成社区反馈信息并输出。

加工名：游戏推广

* + - 编号：8
    - 输入：游戏推广方案
    - 输出：合规推广方案、不合规推广方案
    - 加工逻辑：

if 推广计划合理

then 输出合规推广方案并执行；

else 输出不合规推广方案并提示

endif

加工名：推广方案审核

* + - 编号：8.1
    - 输入：游戏推广方案
    - 输出：合规推广方案、不合规推广方案
    - 加工逻辑：

if 游戏模组符合标准规范

then 输出合规游戏模组

else 输出不合规模组并提示

endif

加工名：修改社区内容

* + - 编号：8.2
    - 输入：合规推广方案、不合规推广方案
    - 输出：社区内容变动信息
    - 加工逻辑：根据输入的合规推广方案进行执行，修改对应的社区内容。

# 概要设计：

用结构化设计方法从需求分析得到的数据流图导出系统初始结构图。

## 1.系统结构划分

Steam游戏平台具体实现可分为八大部分：

* + - 1. 游戏分发部分：

分为读取、核验、检索社区、检索库存、确认购买、确认订阅、生成、输出八个模块；

* + - 1. 游戏下载部分

分为分配、检索、传输三个模块；

* + - 1. 游戏发布部分

分为输入、验证、保存、输出四个模块；

* + - 1. 社区内容发布部分

分为输入、审核、上传、输出四个模块；

* + - 1. 社区反馈部分

分为确认、检索、显示三个模块；

* + - 1. 模组上传部分

分为输入、验证、保存、输出四个模块；

* + - 1. 模组社区反馈部分

分为确认、检索、反馈三个模块；

* + - 1. 游戏推广部分

分为读取、审核、执行、输出四个模块。

## 2.系统初始结构图

|  |
| --- |
| 第一部分数据流图： |
|  |
| 第一部分结构图： |
|  |
| 第二部分数据流图： |
|  |
| 第二部分结构图： |
|  |
| 第三部分数据流图： |
|  |
| 第三部分结构图： |
|  |
| 第四部分数据流图： |
|  |
| 第四部分结构图： |
|  |
| 第五部分数据流图： |
|  |
| 第五部分结构图： |
|  |
| 第六部分数据流图： |
|  |
| 第六部分结构图： |
|  |
| 第七部分数据流图： |
|  |
| 第七部分结构图： |
|  |
| 第八部分数据流图： |
|  |
| 第八部分结构图： |
|  |

# 详细设计

## 1.模块流程图

使用流程图对于模块进行详细设计，模块的流程图如下：

|  |
| --- |
| 1.游戏分发模块程序流程图： |
|  |
| 2.游戏下载模块程序流程图： |
|  |
| 3.游戏发布模块程序流程图： |
|  |
| 4.社区内容发布模块程序流程图： |
|  |
| 5.社区反馈模块程序流程图： |
|  |

## 2.模块N-S盒图

使用N-S盒图对于模块进行详细设计，模块的N-S盒图如下：

|  |
| --- |
| 6.模组上传模块N-S盒图： |
|  |
| 7.模组反馈模块N-S盒图： |
|  |
| 8.游戏推广模块N-S盒图 |
|  |

# 六、系统测试：

测试用例设计的基本目的是确定一组最有可能发现某个错误或某类错误的测试数据。好的测试用例可以在测试过程中重复使用，但不可能测试程序的每一条路径，也不可能把所有的输入数据都试一次。因此，测试用例的设计人员必须努力以最少量的测试用例来发现最大量的可能错误。

## 白盒测试：

### 对游戏分发模块进行白盒测试。

|  |
| --- |
| 1.游戏分发模块程序流程图： |
|  |

部分测试代码如下（Cpp）

|  |
| --- |
| 游戏分发模块测试代码 |
| 1. void test(){ 2. int op; 3. do{ 4. printf("请输入用户请求"); 5. scanf("%s", &req); 6. if(is\_purchase(req)){   *// 检测请求是否为购买请求* 7. printf("请输入外部支付记录"); 8. scanf("%s", &pay); 9. if(is\_valid(pay)) *// 检测外部支付记录是否有效* 10. inventory(gget\_req(req));   *// 游戏入库* 11. else report(pay);   *// 输出错误信息* 12. } 13. else {  *// 说明是游戏下载请求* 14. Game game = gget\_req(req);  *// 获取当前游戏* 15. if(check\_inv(game)){    *// 检索玩家库存是否拥有该游戏* 16. Subs subs = check\_subs(game);   *// 检索玩家的社区订阅内容* 17. gen\_permission(game, subs);     *// 生成下载许可信息* 18. } 19. else 20. wrong\_info(game);    *// 输入游戏为拥有提示* 21. } 22. printf("输入0终止测试\n"); 23. scanf("%d", &op); 24. } while (op) 25. } |

#### 对游戏分发模块进行语句覆盖

I (req为购买请求) and (pay无效) and (玩家库存未拥有此游戏)

II (req为购买请求) and (pay有效) and (玩家库存未拥有此游戏)

II （req为下载请求）and (pay有效) and (玩家库存拥有此游戏)

III （req为下载请求）and (pay有效) and (玩家库存未拥有此游戏)

使用以上四条测试用例可以实现对游戏分发模块的语句覆盖

#### 对游戏分发模块进行判定覆盖

使用以上I、II、III、IIII四条测试用例可以实现对游戏分发模块的判定覆盖

#### 对游戏分发模块进行条件覆盖

使用以上I、II、III、IIII四条测试用例可以实现对游戏分发模块的判定覆盖

### 对游戏下载模块进行白盒测试

|  |
| --- |
| 2.游戏下载模块程序流程图： |
|  |

部分测试代码如下（Cpp）:

|  |
| --- |
| 游戏下载模块测试代码 |
| 1. void test2(){ 2. int op; 3. do{ 4. Download\_Permisson per = get\_permission();  *// 获得游戏下载许可* 5. confirm\_down(per);   *// 确认下载许可* 7. Server\_List lis = get\_sList();  *// 获取服务器列表* 8. Server ser = select\_server(lis); 9. Game game = per.gget(); *// 获取当前要下载的游戏* 10. GameData gd = ser.search(game); *// 检索整理游戏数据* 11. ser.download(gd);   *// 流式传输游戏数据* 13. printf("输入0终止测试\n"); 14. scanf("%d", &op); 15. } while (op) 16. } |

#### 对游戏下载模块进行语句覆盖

I 输入下载许可、服务器列表。

游戏下载模块程序流程不存在分支，使用单条测试用例验证模块代码正确性即可。

### 对游戏发布模块进行白盒测试

|  |
| --- |
| 3.游戏发布模块程序流程图： |
|  |

部分测试代码如下（Cpp）:

|  |
| --- |
| 游戏发布模块测试代码 |
| 1. void test3(){ 2. int op; 3. do{ 4. GameFile gf = get\_gFile(); 5. if(gf.valid\_check())   *// 验证游戏合规性* 6. gf.save();  *// 保存游戏数据* 7. else    report(gf); *//  输出不合规提示* 8. printf("输入0终止测试\n"); 9. scanf("%d", &op); 10. } while (op) 11. } |

#### 对游戏发布模块进行语句覆盖

I (gf 不合规)

II (gf合规)

使用以上两条测试用例可以实现对游戏分发模块的语句覆盖

#### 对游戏分发模块进行判定覆盖

使用以上I、II两条测试用例可以实现对游戏分发模块的判定覆盖

#### 对游戏分发模块进行条件覆盖

使用以上I、II两条测试用例可以实现对游戏分发模块的判定覆盖

### 对社区内容发布模块进行白盒测试

|  |
| --- |
| 4.社区内容发布模块程序流程图： |
|  |

部分测试代码如下（Cpp）:

|  |
| --- |
| 社区内容发布模块测试代码 |
| 1. void test4(){ 2. int op; 3. do{ 4. CommnunityContent cc = get\_cFile(); *// 输入测评* 5. if(cc.judge)   *// 审核测评* 6. cc.upload();  *// 上传至社区* 7. else    info(cc); *//  输出审核未通过* 8. printf("输入0终止测试\n"); 9. scanf("%d", &op); 10. } while (op) 11. } |

#### 对游戏发布模块进行语句覆盖

I (cc 审核通过)

II (cc审核未通过)

使用以上两条测试用例可以实现对游戏分发模块的语句覆盖

#### 对游戏分发模块进行判定覆盖

使用以上I、II两条测试用例可以实现对游戏分发模块的判定覆盖

#### 对游戏分发模块进行条件覆盖

使用以上I、II两条测试用例可以实现对游戏分发模块的判定覆盖

## 黑盒测试：

### 等价类划分

|  |  |  |
| --- | --- | --- |
| 输入条件 | 有效等价类 | 无效等价类 |
| 游戏购买记录 | 编码有效的购买记录（1） | 无效编码的购买记录（2） |
| 用户请求 | 购买游戏（3）下载游戏（4） | 不在{购买游戏；下载游戏}中的其他集合（5） |
| 游戏文件 | 符合相关法律法规的游戏文件（6） | 不合规的游戏文件（7） |
| 测评、评论 | 符合法律法规的游戏测评及评论（8） | 不合规的游戏测评（9） |
| 游戏模组 | 符合法律法规的游戏模组（10） | 不合规的游戏模组（11） |
| 游戏推广方案 | 符合审核标准的推广方案（12） | 不符合审核标准的推广方案（13） |

### 为有效等价类设计测试用例

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖用例 |
| 编码有效的游戏购买记录 | 输入有效 | （1） |
| 购买游戏 | 输入有效 | （3） |
| 下载游戏 | 输入有效 | （4） |
| 合规的游戏文件 | 输入有效 | （6） |
| 合规的测评评论 | 输入有效 | （8） |
| 合规的游戏模组 | 输入有效 | （10） |
| 符合审核标准的推广方案 | 输入有效 | （12） |

### 为无效等价类设计测试用例

|  |  |  |
| --- | --- | --- |
| 测试数据 | 期望结果 | 覆盖用例 |
| 编码无效的游戏购买记录 | 输入无效 | （2） |
| 修改游戏 | 输入无效 | （5） |
| 不合规的游戏文件 | 输入无效 | （7） |
| 不合规的测评评论 | 输入无效 | （9） |
| 不合规的游戏模组 | 输入无效 | （11） |
| 不符合标准的推广方案 | 输入无效 | （13） |

### 边界值分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入等价类 | 测试用例说明 | 测试数据 | 期望结果 | 选取理由 |
| 游戏购买记录 | 空值 | null | 报告错误 | 输入为空置 |
| 用户请求 | 不存在的用户请求类型 | “alt” | 报告错误 | 没有该类型的用户请求 |
| 游戏文件 | 错误的游戏文件类型 | Game.docx | 报告错误 | 不是可以运行的游戏文件 |
| 测评、评论 | 长度为65535的文本内容 | 65535个重复的”\*“ | 报告错误 | 过长的测评评论 |
| 游戏模组 | 文本文件 | Mod.txt | 输入有效 | 游戏模组不限制文件类型 |
| 游戏推广方案 | 0.1折促销价 | 促销-99% | 输入有效 | 边界值 |