

4398小游戏代码设计文档

目录

4398小游戏代码设计文档

目录

1 引言

1.1 编写目的

1.2 背景

2 总体设计

2.1 需求规定

2.2 运行环境

2.3 总体结构设计

2.3 基本设计概念

2.4 结构

主页流程图

Flappy Bird游戏流程图

Space Ship游戏流程图

exit操作流程图

2.5 人工处理过程

3 接口设计

3.1 用户接口

载入游戏界面：

游戏主界面：

3.2 软件接口

3.3 内部接口

4 运行设计

4.1 运行组合

4.2 运行控制

5 系统数据结构设计

5.1 逻辑结构设计要点

5.2 物理结构设计

6 出错处理设计

6.1 出错输出信息

7 系统维护设计

1 引言

在现今这个压力大的社会中，游戏成了人们首选的消遣。游戏的选择很多，导致游戏市场参差不齐，有些游戏不仅让人们花费更多时间和精力，而且体验感极差。一款好的游戏应具备让人们放松，同时也不占人们时间的条件。4398小游戏就是为了这个目的而创建出来的。4398小游戏可以大大降低人们的压力程度，提高各自工作的效率。

1.1 编写目的

4398小游戏以“画面简单易玩”为其中一个目标。目前4398小游戏已经兼容了 flappy bird 和 space ship 两款单机游戏。用户可以自主选择游戏。这两款游戏中途都可以随时退出，方便用户忙碌自己的事情。

1.2 背景

本项目的任务提出者及开发者是北京航空航天大学软件学院天天向上小组：

系统设计分析编码员：朱章涌，时冉，王佳媛

软件测试员：朱章涌，时冉，王佳媛

用户为各界人士，该软件在WINDOW 10系统和MACOS系统下，在INTELLIJ IDEA JAVAFX环境下完成开发，请提前配置好环境。

2 总体设计

2.1 需求规定

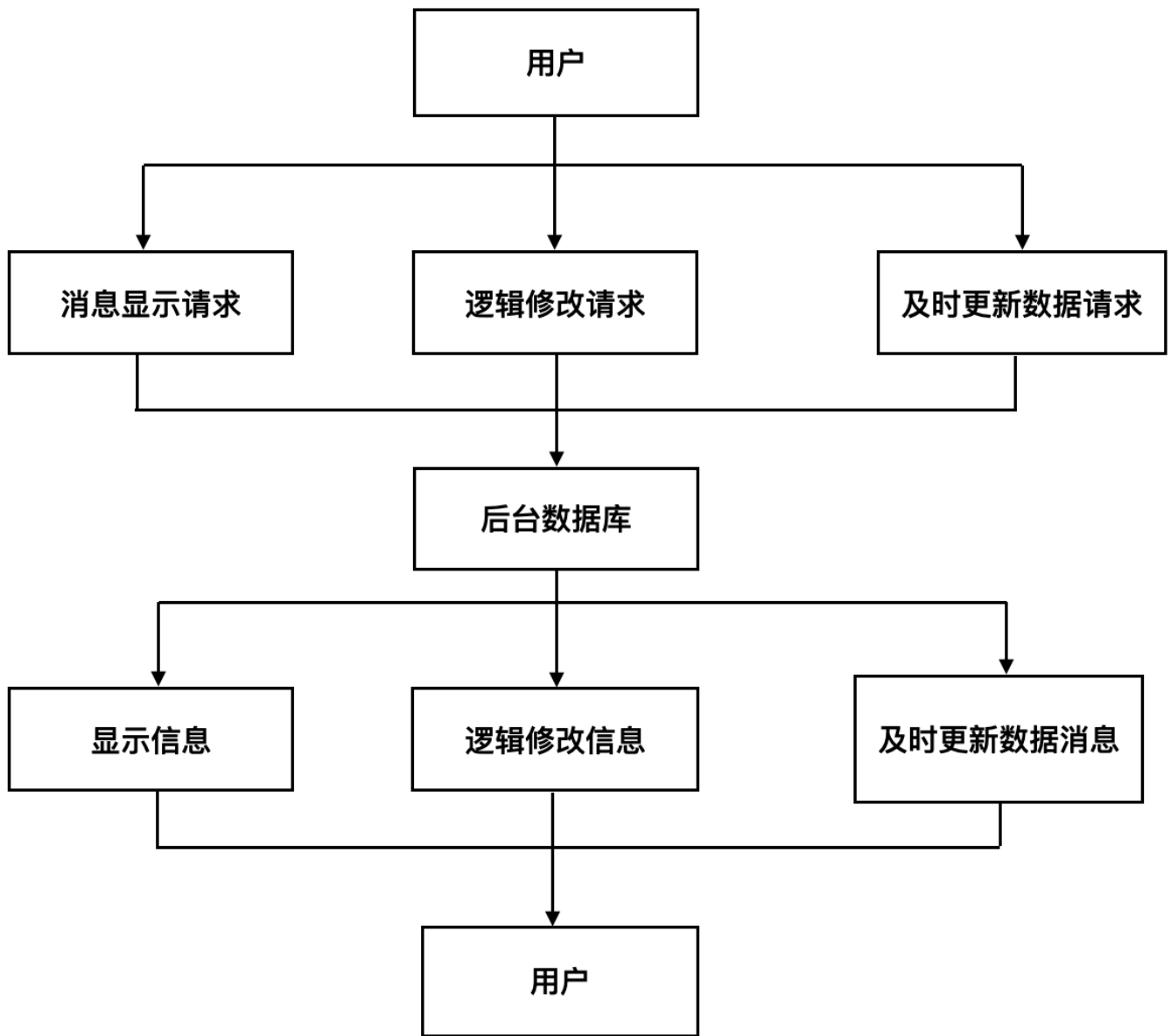
本软件系统的各种用户是唯一的参与者，参与者通过使用事件与系统进行交互。本系统通过用户操作键盘及屏显示交互设定相关系统、传感器参数和系统运行状态。

2.2 运行环境

本软件属于游戏运行系统。WINDOW 10系统和MACOS系统下的INTELLIJ IDEA JAVAFX环境皆可运行。

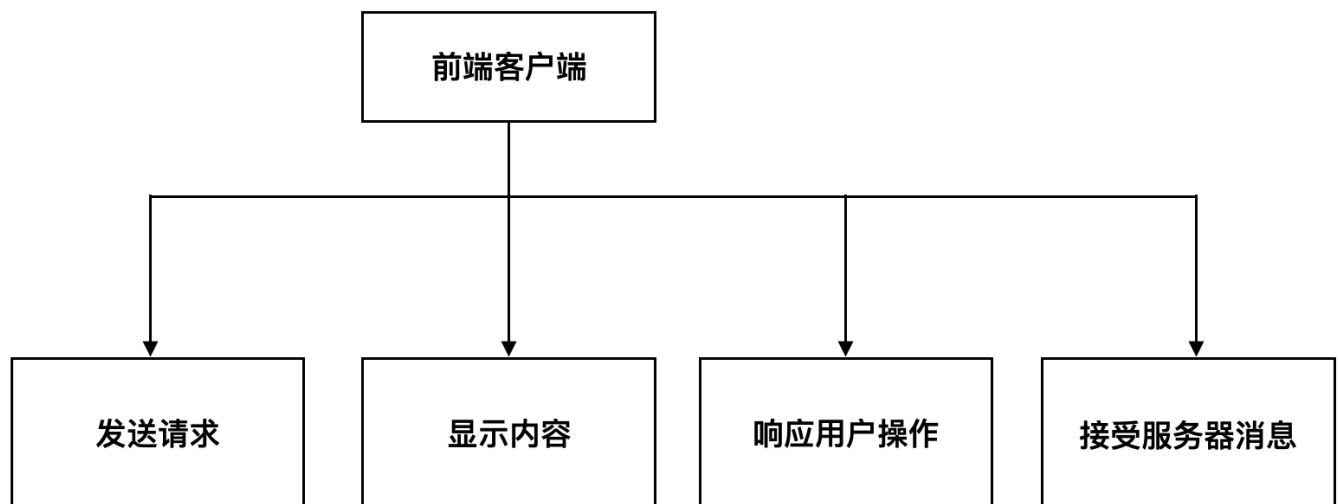
2.3 总体结构设计

用户的操作并不会改变游戏中的实际逻辑，而是以向后台数据发送数据更改的申请。后台数据库还需要对修改信息的申请进行判断，如果操作真实有效，不属于恶意请求则修改后台数据逻辑，并向用户界面发送修改后的结果。



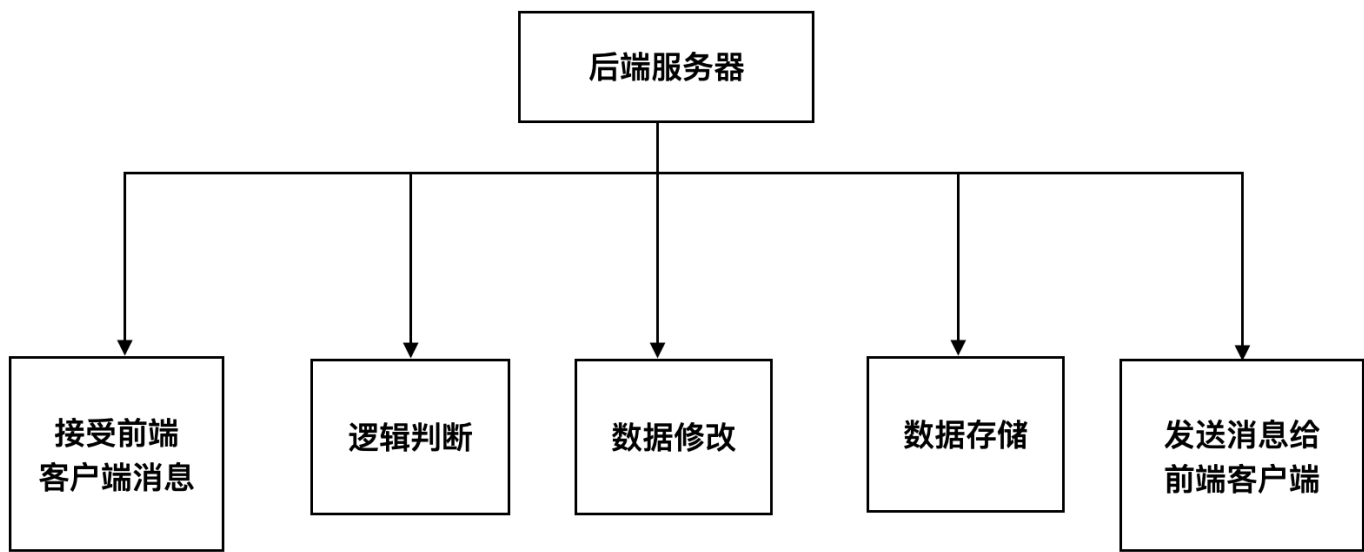
2.3 基本设计概念

用户操作的游戏为前端客户端，所需要的功能如下：



用户自行选择游戏后，发送请求，后台接受服务器消息并响应用户操作，再把内容页面发送至客户端页面。

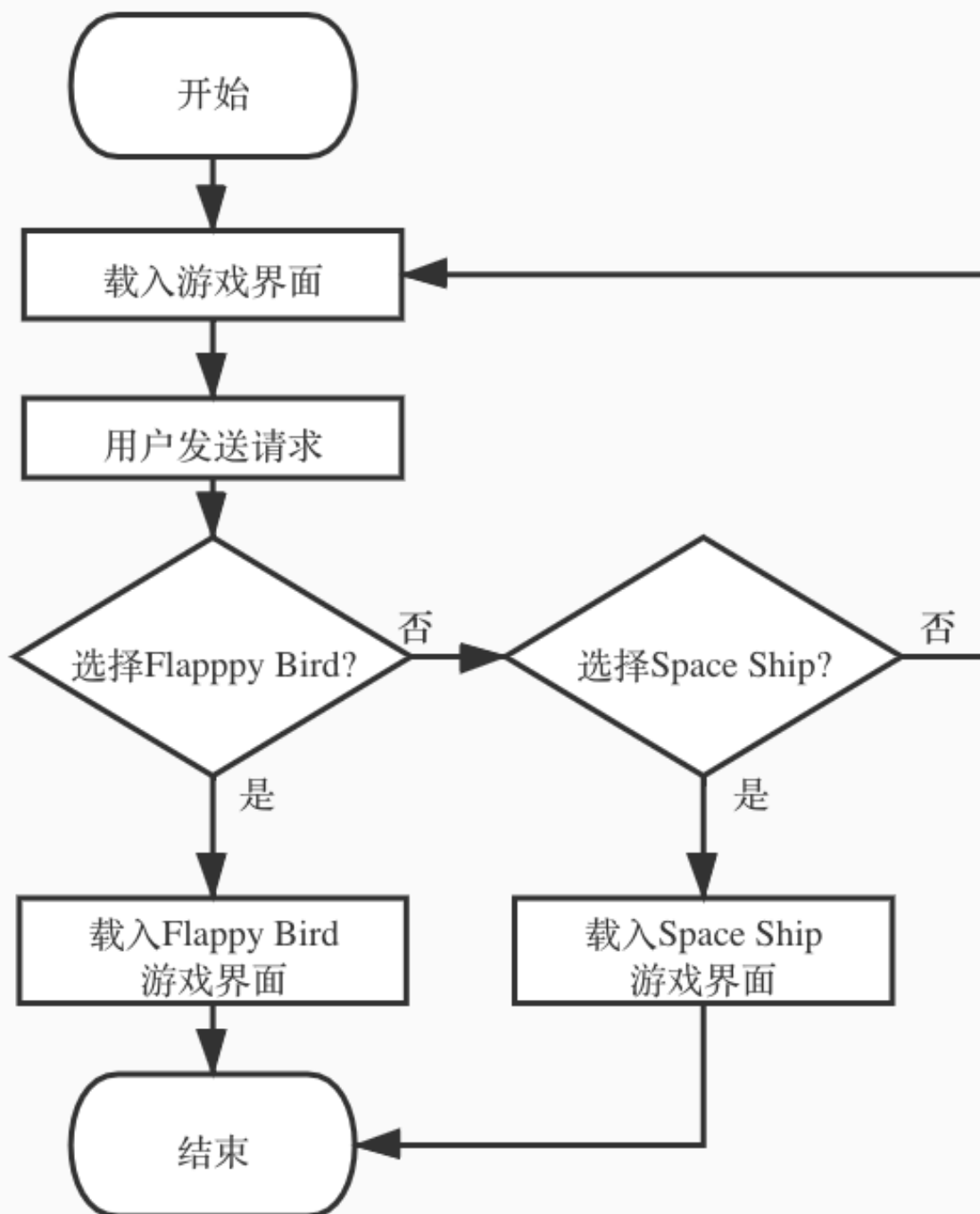
运营的服务器为后台服务器，所需功能如下：



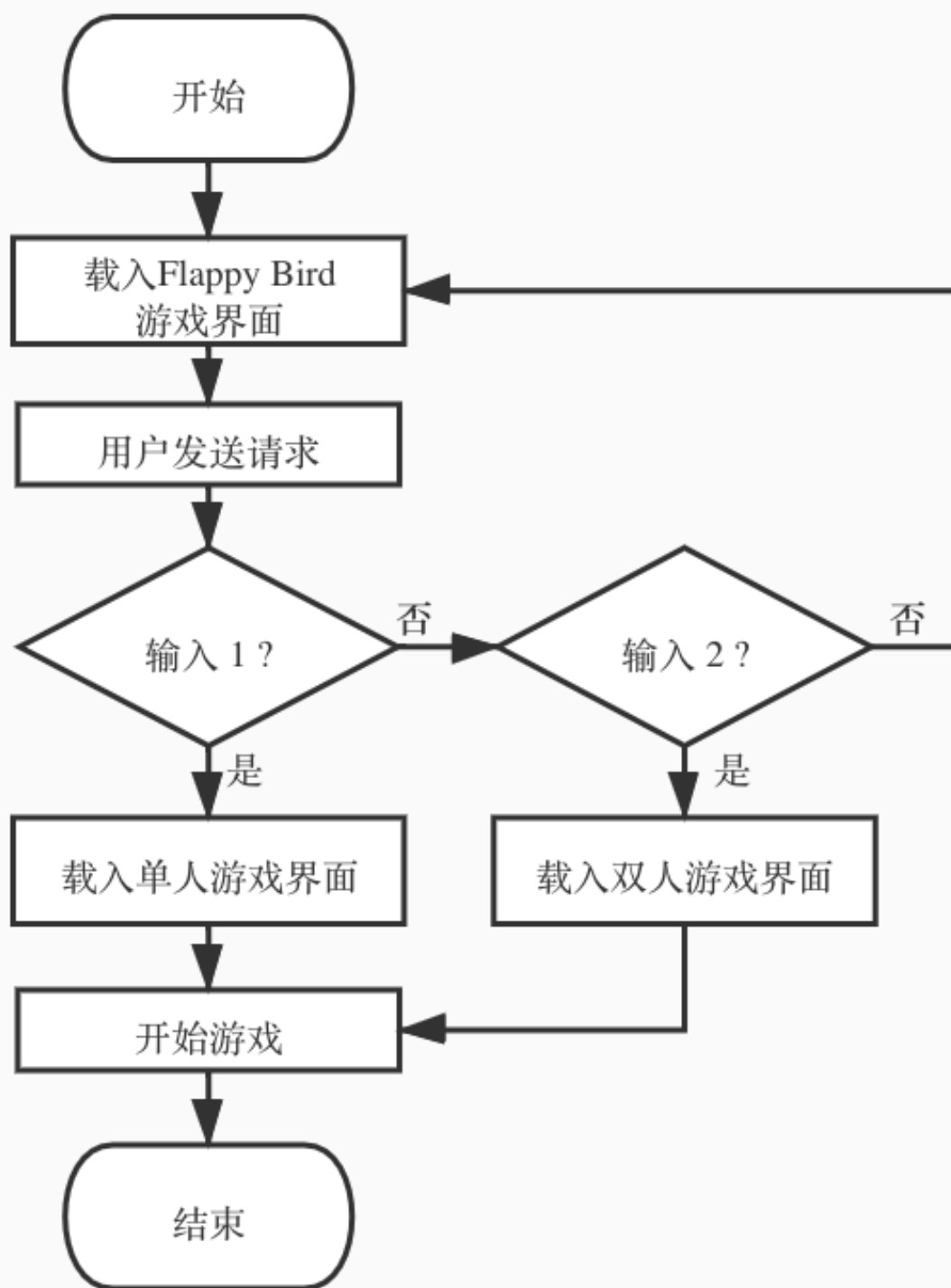
后台接受客户端请求后，判断逻辑并修改存储数据，再把内容页面发送至客户端页面。

2.4 结构

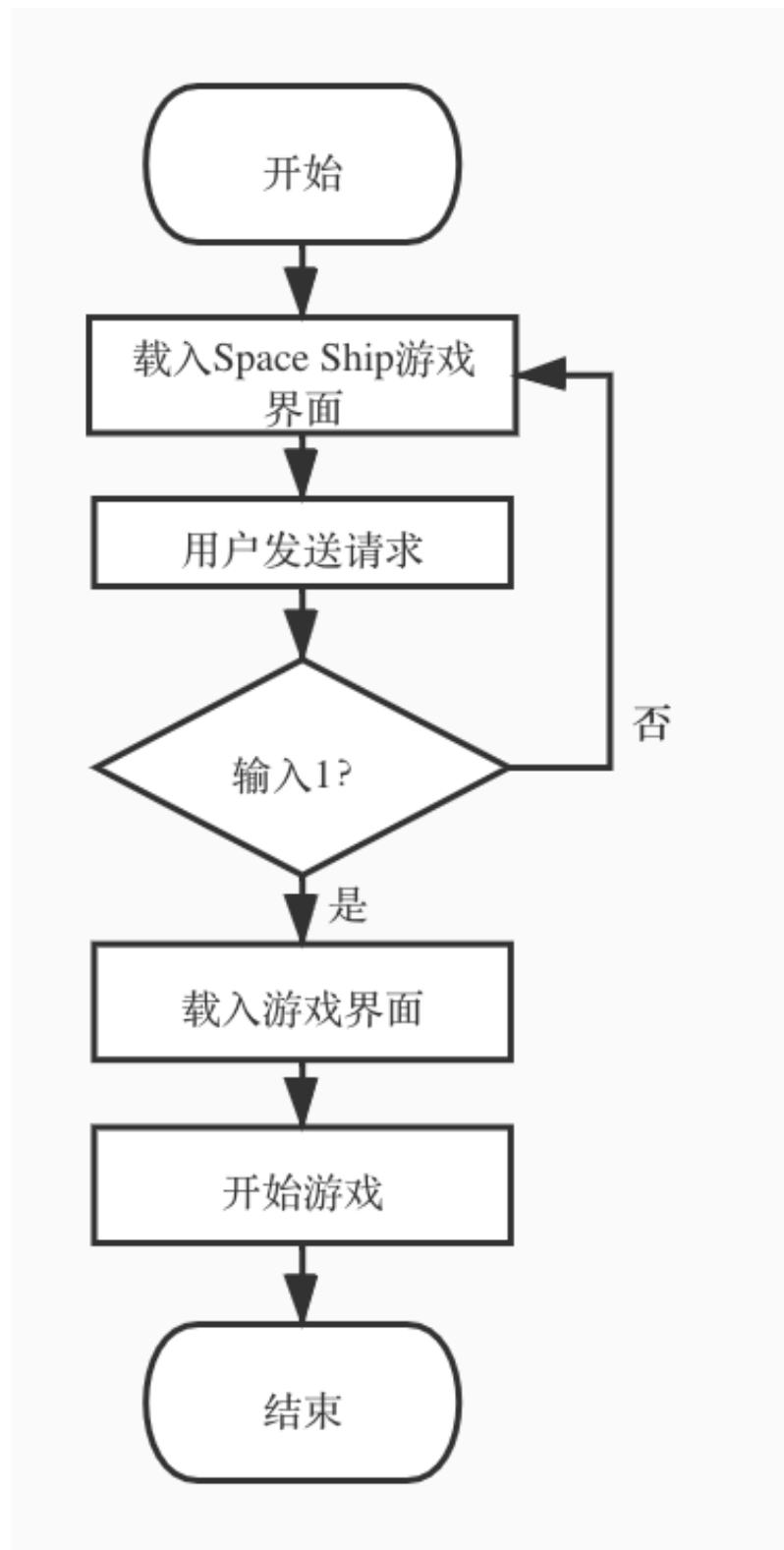
主页流程图



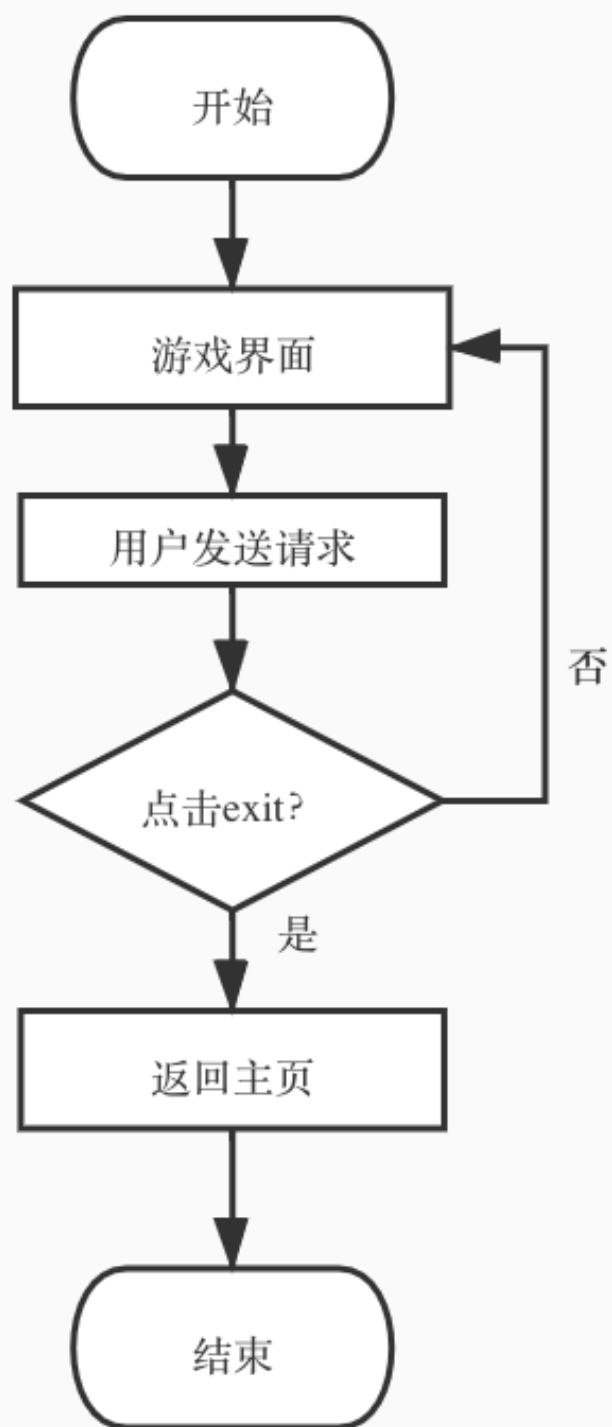
Flappy Bird游戏流程图



Space Ship游戏流程图



exit操作流程图



2.5 人工处理过程

用户通过键盘及显示屏，依靠系统显示页面提示对相关内容进行设置，以达到按照具体请求运行程序获得预期效果的结果。

3 接口设计

3.1 用户接口

载入游戏界面：

主要是为用户载入游戏的美术资源、代码。

游戏主界面：

背景：显示游戏中移动物体的摆放逻辑界面

移动物品图片：比如flappy bird里的柱子和鸟

按钮栏：点击exit可返回主页

积分栏：显示游戏中获得的积分

背景音乐：播放背景音乐

3.2 软件接口

与数据库相连，向用户游戏界面的积分栏进行传递。

3.3 内部接口

用户事件：初始化，获取用户请求

主页事件：flappy bird请求，space ship请求

flappy bird事件：单人模式请求，双人模式请求，移动跳跃，触碰物品

space ship事件：左右移动，触碰物品

exit事件：返回主页

4 运行设计

4.1 运行组合

客户端依靠用户所使用的电脑，进入javafx环境进入游戏界面。

主页模块 —> flappy bird模块 —> 单人模式模块—> 系统参数模块

—>exit模块—>主页模块

主页模块 —> flappy bird模块 —> 双人模式模块—> 系统参数模块

—>exit模块—>主页模块

主页模块 —>space ship模块 —> 系统参数模块—>exit模块
—>主页模块

4.2 运行控制

由用户通过键盘和按键选项进行控制。

5 系统数据结构设计

5.1 逻辑结构设计要点

本系统各功能紧密结合，为尽量避免相互影响出现错误，系统严格按时间顺序进行，保证数据绝对准确，各端口数据独立接受，统一处理，保证数据处理不会混杂的前提下，保证更高的处理效率。

客户端主要分为两部分：

- 1. 游戏界面显示功能
- 2. 逻辑运算引擎

5.2 物理结构设计

游戏前端数据、信息、显示在用户的家用电脑内进行操作、计算，游戏后台逻辑数据、规则在游戏运营商的服务器上进行操作。

6 出错处理设计

6.1 出错输出信息

当软件进行硬件运行检查，发生错误会重复启动多次避免偶然情况导致硬件运行不正常，在多次检验无法通过时，会向客户端显示出错。

7 系统维护设计

正确性维护：在运行的过程中发现错误，根据错误的功能项找到相应模块，对出错模块单独测试和修改。

适应性维护：软件的运行平台上，限定住可能出现的问题范围，方便排查。

完善性维护：为了应对用户新提出的要求或改善性能时，增加新的功能时，由于系统模块间的独立性，新功能通常可以单独形成新的模块，经测试后拼加到系统中，而对其他模块影响不大；改善某模块的性能时，只需对相应模块进行改进，然后还原到系统中。