**北京理工大学珠海学院**

**Java高级程序设计说明书**

\_2019\_—\_2020\_学年第\_ 2 \_学期

**题目: 生命游戏**

**学 院： 计算机学院**

**专业班级：18软件工程3班、4班**

**学 号： 180021100969、180021102316**

**180021102400、180021105120**

**180410106818**

**学生姓名： 蔡梓沁、陈杰伟、罗坤**

**刘堉嘉、王若雨**

**指导教师： 魏志军**

**成 绩：**

**时 间： 2020.06.18**

小组分工

|  |  |  |
| --- | --- | --- |
| **姓名** | **学号** | **完成任务** |
| 刘堉嘉 | 180021105120 | 整合代码、客户端搭建与对话连接、生命模块设计、文档撰写 |
| 蔡梓沁 | 180021100969 | 数据库设计、服务端搭建、文档撰写及整合 |
| 陈杰伟 | 180021102316 | 用户模块设计、文档撰写 |
| 罗坤 | 180021102400 | 音乐模块设计、文档撰写 |
| 王若雨 | 180410106818 | 游戏界面设计、文档撰写 |

生命游戏

摘 要

生命游戏是英国数学家约翰·何顿·康威在1970年发明的细胞自动机。其游戏原理是在简单3条规则下给与一定的初始条件，并静候格子的演变，在演变过程中会得到很多奇妙的图案，从中可以感受生命的演化过程，让玩家能够拥有很好的游戏体验。实现此游戏开发，主要利用Java作为编程语言，运用Sock模拟简单HTTP连接进行完整的请求分发，数据封装完成C/S架构交互,采用SQL Server数据库的设计存储，通过Java Swing完成对生命游戏的用户和游戏的GUI界面设计，并将所有功能接口整合进UI，实现交互。具有较为合理性的客户端服务端数据库分层构建，最大限度完成游戏的可视化编程。

关键词：细胞自动机 生命演化 可视化

目 录

[小组分工 1](#_Toc43478108)

[生命游戏 2](#_Toc43478109)

[摘 要 2](#_Toc43478110)

[目 录 3](#_Toc43478111)

[1.生命游戏模板设计 5](#_Toc43478112)

[1.1 生命模块需求分析 5](#_Toc43478113)

[1.2生命周期设计 5](#_Toc43478114)

[1.3生命和矩阵存储 8](#_Toc43478115)

[1.4游戏算法优化 9](#_Toc43478116)

[2.游戏数据库设计 10](#_Toc43478117)

[2.1数据库的作用与使用工具 10](#_Toc43478118)

[2.2数据库存储的对象 11](#_Toc43478119)

[2.3数据库部分需要实现的功能 13](#_Toc43478120)

[2.4具体设计与使用 14](#_Toc43478121)

[3.服务端与客户端搭建 19](#_Toc43478122)

[3.1服务端与客户端搭建作用 19](#_Toc43478123)

[3.2 搭建运用的类 19](#_Toc43478124)

[3.3 服务端搭建 19](#_Toc43478125)

[3.4客户端搭建 20](#_Toc43478126)

[4.用户界面功能设计 22](#_Toc43478127)

[4.1登陆界面模块图 22](#_Toc43478128)

[4.2注册功能 23](#_Toc43478129)

[4.3登录功能 24](#_Toc43478130)

[4.4修改信息功能 25](#_Toc43478131)

[5.音乐播放器 26](#_Toc43478132)

[5.1背景音乐 26](#_Toc43478133)

[5.2音乐功能位置 26](#_Toc43478134)

[5.3音乐功能 27](#_Toc43478135)

[5.4功能实现 27](#_Toc43478136)

[6. 游戏画布 28](#_Toc43478137)

[6.1主界面交互特点 28](#_Toc43478138)

[6.2．具体实现 28](#_Toc43478139)

[参考文献 30](#_Toc43478140)

[心得体会 31](#_Toc43478141)

[教师评语 32](#_Toc43478142)

1.生命游戏模板设计

1.1 生命模块需求分析

设计清单：

1.基础功能设计

点击，拖动，图案去放置生命，画布循环，选择生命区域，放置图案，生命棋盘的放大放小。

2.生命演变及其视图绘制

对《生命游戏》基础三条规则，进行模拟推演，显示。以及对基础功能在画板上的绘制。

3.“生命”设计

对构建完成的一个有具体规律的生命图案进行对象生成，构建。

4.数据/矩阵存储

对生命和完整的整个画布进行矩阵转换成密文，并且保存到剪贴板，进行分享。

1.2生命周期设计

1.2.1画布类核心属性：

Runing：控制游戏开始停止状态

CELL\_Size：控制生命大小

Xsize,ysize：计算出的窗口细胞数量

SIZE ：画布最大大小

button[] ：记录按键状态

rangecolor等： 画布等颜色

table：现有生命状态

neighbors：下一次的生命情况

lasttable：画布上生命状态

mergetable： 图案矩阵

animator：绘制线程

isupdata：控制是否更新整个画布

movestatic:绘制画笔状态

1.2.2 生命周期演变

* 生命演变绘制

画布的绘制使用boolean二维数组存储绘制，采用传统的线程绘制的方式，设置了runing进行绘制线程控制开始停止，为了观察生命演变设置线程sleep进行线程绘制频率控制，通过动态调整睡眠时间进行控制。

* 模拟生命游戏

演变周期分为3个步骤，并且设置table数组线程同步防止绘制

1.从当前生命导出状态neighbors数组

2.计算neighbors数组每个位置是否存活，并且更新到table数组

3.进行重新绘制table

* 生命绘制详细

1.通过计算窗口大小和CELL的大小计算出窗口容纳生命数xsize,ysize。

2.计算在table数组哪一段，利用Graphics开始绘制方格需要注意线程同步，绘制过程table不能有所改变。

3.绘制

1.2.3画布监听控制及功能实现

* 按钮监听

通过监听使用按下和松开，控制boolean button[]数组，调整button数组对应按钮的状态。进行判断快捷建。采用监听内部类，使监听在外部操作可以操作不局限在画布窗口。

* 鼠标监听

检查鼠标滚动检测，数值变化，记录并且使用，一样的通过检查clicsta改变点击变化

* 图案放置/点击放置/拖动放置

设置mouthsta控制画布合并状态，通过监听调整状态进而改变点击时的效果，包括添加删除生命的点击，添加图案的点击，选择区域的点击。

a.常规点击

根据点击事件坐标轴和当前再屏幕的xy对细胞取余数，计算细胞位置，如果是点击和消除情况，改变mouthsta控制合并清除，然后调用mergetable进行判断当前所属合并类型进行table调整变化，同样的需要线程同步绘制。

b.区域选择

设置mouthsta 3，选择区域在mergetable的时候不进行操作只计算细胞位置，而在绘制的时候检查是否为区域选择，在之前记录的2点坐标绘制透明矩形，进行可视化。

c.图案放置

设置mouthsta 2，设置合并数组，在点击的时候，mergetable进行table合并。

* 画布放大放下/速度调整

通过滚轮监听，鼠标上下滑动的大小计算细胞在画布上的大小，然后进行重新计算绘制。

滚轮监听的时候检查按键监听的ctrl是否按住，进行判断是否根据滚动调整频率。

1.3生命和矩阵存储

### 1.3.1矩阵存储

1.设计思路

由于，生命的存储采用数组的形式，而且有存储需求，进行转换方便使用，我们采用矩阵转换成字符串存储。

2.算法概括

而矩阵的存储采用类图像压缩，记录连续的格子数去存储，也是我研究了专业软件Golly的压缩方式，发现其规律，然后这样更加方便搬运，和分享生态，贴吧有许多分享的有趣的生命图案可以直接使用。

3.具体实现（有经过随机函数进行效验正确性）

对矩阵进行转换字符串

对字符串进行矩阵转换

### 1.3.2生命类构建

1.生命类设计

生命类对图案矩阵存储，存储矩阵的x y 矩阵是实现 加密方面直接调用。然后采用jframe类，在范型的时候对生命于画布中展示预览。方便直接调用。

2.具体实现

a.生命类采用和画布类的模式，对矩阵直接绘制，因为无动态和观察需求，直接采用绘制对应生命块。

b.因为生命图案大小不一，通过计算得出最优位置放于画布中央，居中显示更加美观

c.对太大放不下的图案，采用默认图案“大生命”绘制原理是创建“大生命”对象，然后将不能够绘制对象的Graphics传递给大生命对象进行绘制，结构更加清晰

1.4游戏算法优化

1.4.1数据层优化

对生命计算的时候，原先采用每个点计算周围的生命数去计算其生存等，后面采用点“剥夺资源”模式计算，只有有生命的对周围分散点数，然后判断，这样在画布大小增大的时候，可以只计算生命的点，速度会快很多。采用画布循环的方式这样可以扩展画布的大小不局限在可见窗口上（定点放大功能没时间实现，现在只是0点放大）。

1.4.2 视觉层绘制优化

* 版本1：直接绘制整个table数组。

通过直接绘制每一个小格子绘制table数组。由于是重写即使没在窗口内部，仍然会进行绘制导致当画布在小的时候仍然需要大量的资源进行无所谓的绘制。

* 版本2：绘制窗口显示部分

通过计算窗口大小，绘制窗口显示大小部分。优化后绘制速度在小的时候可以快速实现绘制，可是当窗口大的时候，绘制仍然在400\*400以上的矩阵的时候，基本需要几秒钟的绘制时间。

* 版本3：记录窗口状态动画刷新

由于绘制每次动态改变的只有对应的点，所以用lasttable数组记录每次绘制在画布上的细胞，发生改变就会重写绘制。这样当调整窗口大小的时候，需全局重新绘制，导致全局绘制还是很慢。

* 版本 4：绘制方式改变

多次试验发现，绘制最主要还是绘制方块的数量，而核心的生命方块绘制其实很少，且切换颜色会导致效率大大减少。采用把白色方块和红色方块分开绘制，能提高速度；把白色方块的绘制采用背景颜色的方法，先绘制生命块再进行绘制线条，背景颜色代替白色方块，减小绘制空间。在局部绘制上虽快，但导致右边位置线条刷新绘制慢。结合前版本，全局刷新采用线条刷新，局部采用动态刷新。

* 版本5：采用双缓冲技术

后续进展的时候，发现一种可以解决刷新问题的方法就是，双缓存技术，将画布存在image里面，然后等绘制完成后，在把imge绘制出来，这样完美解决了线条刷新的弊端，屏幕刷新痕迹基本看不出来了。

2.游戏数据库设计

2.1数据库的作用与使用工具

2.1.1数据库作用

1. 存储数据

我们所设计的这款游戏，除了要考虑如何带给玩家好的画面感与简单方便的使用界面外，还需要考虑玩家用户信息的存储以及玩家所作的画布信息存储，我们采用将数据放入所建立的数据库方式，方便平台前后端数据的交互与数据的调用。

1. 用户管理

一个游戏往往不会只有一个用户，而好的游戏甚至在线玩家会破亿，而游戏的用户管理往往离不开数据库，用户的密码匹配以及用户信息资料的保存与用户信息的更改都需要数据库做桥梁进行。

1. 数据的备份与还原

一个游戏的数据是游戏里很重要的一部分，如果发生了不可抗的意外导致数据丢失那么对于一个上市游戏来说是影响非常严重的问题，所以数据的备份是不可少的，而数据的备份与还原也是利用数据库进行的。

### 2.1.2 数据库使用工具

Microft SQL Server

2.2数据库存储的对象

2.2.1 概述

数据库中在初步使用阶段建立了三张游戏所需的表分表为用户表（users）、画布表（cancas）、模板表（Template）,后续将根据实际情况向数据库UserDB添加游戏所需的表。

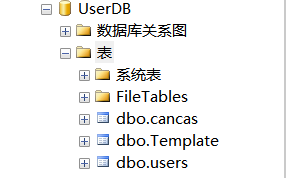


图2-1 数据库中的表

2.2.2 用户表（users）

用户表在用户登录游戏界面进行用户注册时会将用户的信息记录到此表中，在后续的更改密码中也基于此表进行。表中属性有三个分别为：ID、username、password，其中ID为自增型整数，为列表中添加的用户自动分配ID以方便后续根据ID进行用户画布的查找。

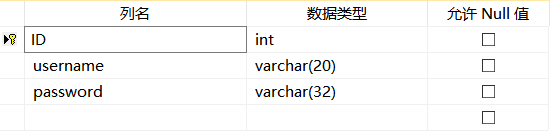


图2-2 用户表设计

2.2.3 画布表（cancas）

画布表所存放的数据为用户登录后运行游戏所设计的游戏画布，可以为用户提供查询自己所作画布的功能，画布表中包含七个属性分表为：cname（画布名称）、data（画布数据）、cycle（周期）、X（画布的长）、Y（画布的宽）、userid（用户ID）、ID（画布ID）。其中userid作为外键与users表中的ID对应，方便用户进行自己的画布查找。



图2-3 画布表设计

2.2.4 模板表（Template）

模板表所存放的是指定用户（管理员）设计的游戏模板，即用户可以选择此模板进行画布的设计，该表中共五个属性分别为：data（模板画布数据）、cycle（周期）、X（画布的长）、Y（画布的宽）、ID（模板ID）。

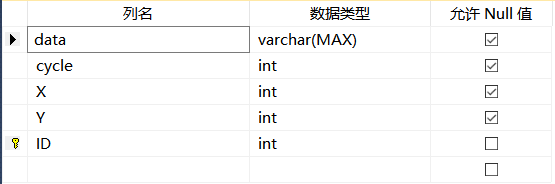


图2-4 模板表设计

2.3数据库部分需要实现的功能

2.3.1 基础功能

(1)数据库的连接

利用sqljdbc4实现JAVA与SQL数据库的连接

1. 向数据库中插入信息

通过传入sql语句向数据库指定表中插入数据

1. 修改数据库信息

通过传入sql语句修改数据库表中的指定数据

1. 查询数据库信息

通过传入sql语句查询数据库指定表中的数据

2.3.2 针对不同对象设计的功能

(1)用户部分所要实现的功能

a.用户注册完成将用户信息插入用户表

b.用户成功登陆后可进行密码修改

c.用户可查询自己的画布信息

1. 画布部分所要实现的功能

用户在游戏结束后系统将用户在游戏中所作的画布信息存入画布表中

(3)模板部分所要实现的功能

核对用户是否为管理员，是则可进行对模板表的信息存储，否则不可进行此操作

(4)管理员可实现的功能选项

1. 管理员可查看所有用户信息

b.管理员可查看所有的画布信息

c.数据库的备份与还原

2.4具体设计与使用

2.4.1 所创建的类

(1)数据存储、修改、查询部分主要是基于Cancas类与User类进行，其中User类对应users表，Cancas则对应模板表与用户画布表，通过不同的构造函数方法进行用户表与模板表的区分。

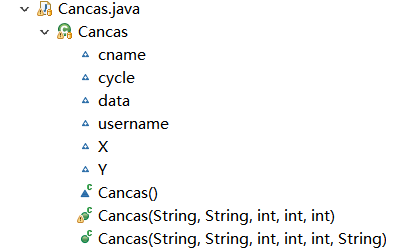


图2-5 Cancas类

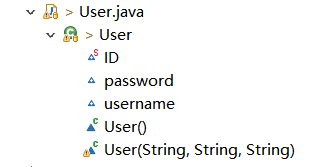


图2-6 User类

1. 数据的备份与还原是基于data类进行，其中属性name为备份数据时的文件名，path为备份文件的存储路径

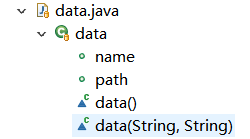


图2-7 data类

2.4.2 数据库连接及创建

（1）数据库连接流程

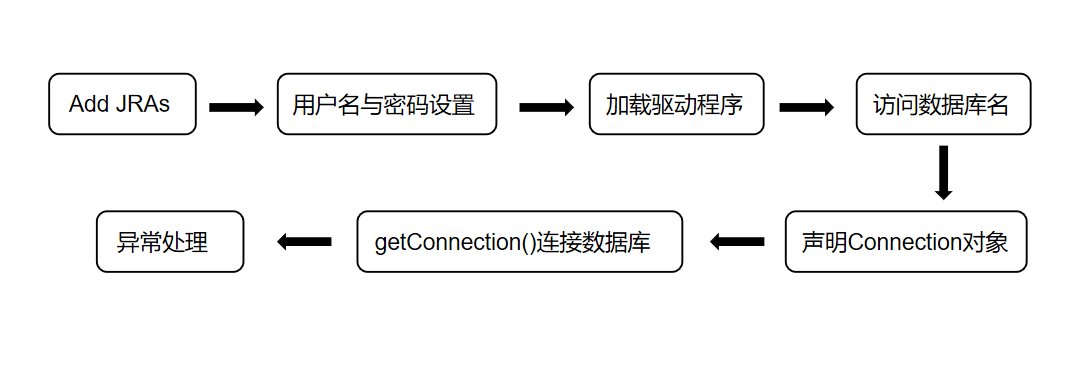


图2-8 数据库连接流程

1. 数据库创建流程

数据库创建部分是在JAVA客户端工具里利用addBatch进行数据库与游戏所需表的创建，这样避免了SQL客户端工具与JAVA客户端工具的来回切换，创建流程如下：

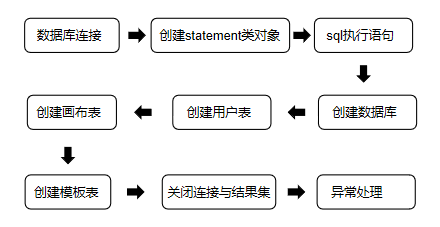


图2-9 数据库创建流程

注：由于画布表中的外键userid与用户表中的ID相关联，所以在创建表时需先创建用户表在创建画布表

2.4.3 数据库数据插入

数据库的数据插入分为三部分，分别为用户数据插入、用户画布插入、模板画布插入，分别编写了三个类似函数进行三种数据的插入，其插入流程都如下：

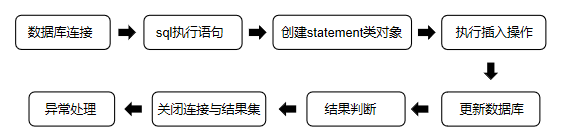


图2-10 数据库数据插入流程

注：1.用户数据插入通过用户注册时输入的用户名字与密码调用函数用户数据的插入函数。

1. 用户画布数据插入通过输入用户在游戏中为画布所起的名字及所作画布的数据、周期、画布的长、画布的宽及用户名调用画布数据插入的函数，其中通过用户名寻找其ID，ID存在时才可进行数据的插入。
2. 模板画布数据插入与用户画布数据插入相似，不同的是增加了判断用户是否为游戏管理员，只有身份为游戏管理员的用户才可进行模板数据的插入。

2.4.4 数据库信息修改

数据库的信息修改主要是针对于用户的密码修改，在用户成功登陆后才可进行其用户对应的密码修改，流程如下：

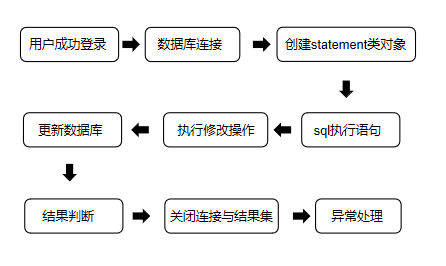


图2-11 数据库信息修改流程

2.4.5 数据库数据查询

数据库数据查询分为三部分，分别为用户数据查询、用户画布查询、模板画布查询，分别编写了三个类似函数通过传入的查询语句进行不同的查询，其查询流程如下：

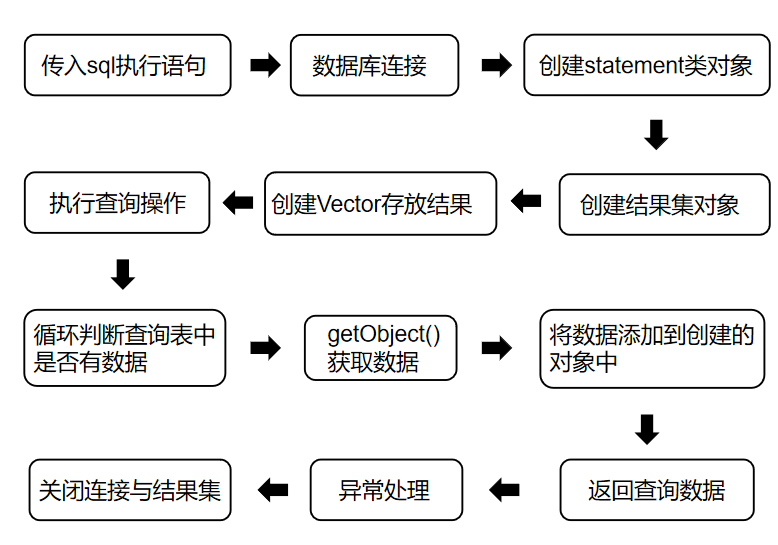


图2-12 数据库数据查询流程

注：1.用户数据查询分为查询是否有重复用户名、查询所有用户、用户密码匹配、查询用户ID，其中查询所有用户只有管理人员可以进行，是否有重复用户名用于用户注册时判断用户名是否已存在，用户密码匹配用于用户登录时核对用户名与用户密码是否匹配，查询用户ID用于根据ID寻找用户创建的画布。

1. 画布查询分为查询所有用户创建的画布、查询某用户创建画布，其中查询所有用户创建的画布只有管理人员可以进行，查询某用户创建画布用于用户查询自己创建的画布。
2. 模板查询只有管理人员可以进行相关操作。

2.4.6 数据库备份与还原

(1)数据备份设计

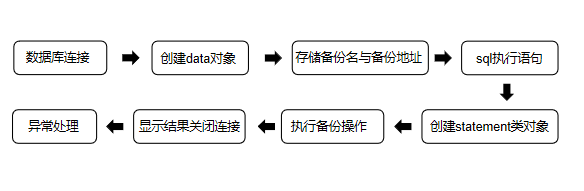


图2-13 数据库备份设计思路

(2)数据还原设计

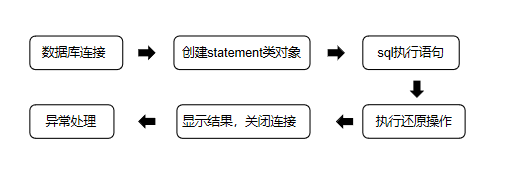
****

图2-14 数据库还原设计思路

3.服务端与客户端搭建

3.1服务端与客户端搭建作用

服务端与客户端的搭建利用TCP协议面向连接的、可靠地、有序的，以字节的方式发送数据，通过两端的连接建立进行通信，其中Server Socket用于服务器端，Socket是建立网络连接时使用的。在连接成功时，应用程序两端都会产生一个Socket实例，操作这个实例，完成所需的会话。

3.2 搭建运用的类

在服务端与客户端的数据传输部分除了运用到前文所提的JAVA Socket，还创建了DataJson类作为传输媒介更方便地进行服务端与客户端的数据传输，其中用到Hash Map通过不同的关键字（键值）实现不同的功能选择。

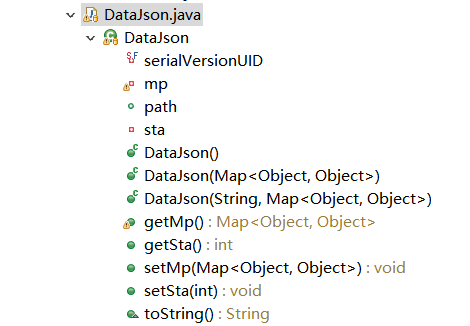


图3-1 DataJson类

3.3 服务端搭建

服务端的工作主要是在接收到客户端发来的请求数据后根据其请求对应的功能选项例：插入用户画布信息，进行实现操作并将结果数据封装到DataJson类中返回客户端，具体设计流程如下：

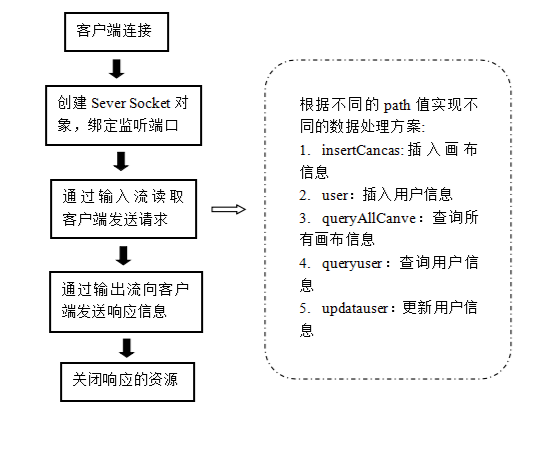


图3-2 服务端设计流程

3.4客户端搭建

### 3.4.1客户端设计模拟Http设计思路

客户端采用传统的模拟Http请求的设计，并且对传输的过程进行封装。

1.多线程请求

客户端每次对服务器进行请求一次Sock，会创建一个独立的线程进行，数据交互，让网络请求合理，快速化。

2.回调请求

由于网络请求是异步操作，所以在客户端，需要实现当在请求线程，获得响应得到数据之后进行回调。

3.数据封装

在数据传输上，采用DataJson类作为传输对象，对数据传输和前端代码的分离解耦，而将数据封装分离开来，多了一层进行API封装。

3.4.2 客户端具体实现



图3-3 实现流程

* 由于需要请求回调，所以创建了回调接口CallBack，要求前端代码在发起请求的时候，需要接口实现回调函数进行异步操作。

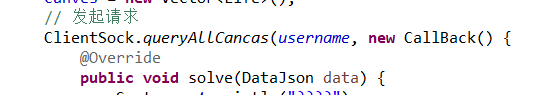
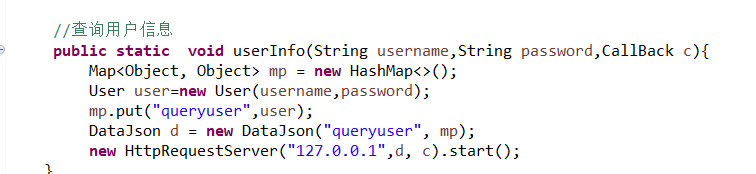


图3-4 前端实现solve回调函数

* 创建ClientSock类对前端的数据封装成DataJson对象，并且设置请求服务器对应的路径，进行封装。

 图3-5 ClientSock数据封装

* 创建Http类，对sock传输进行控制，由内部线程类reQuestServer对DataJson对象Sock传输到服务器，。

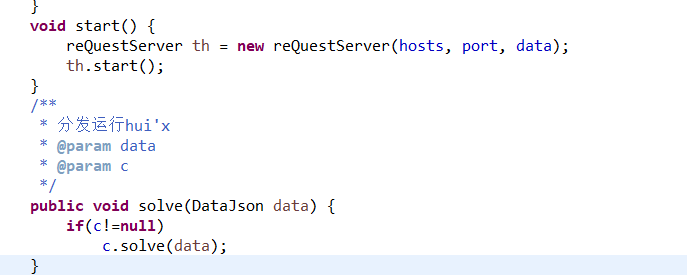


图3-6 Http类对发起Sock线程

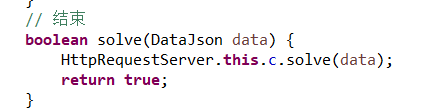


图3-7 ReQusertServer类请求完毕调用回调函数

4.用户界面功能设计

4.1登陆界面模块图

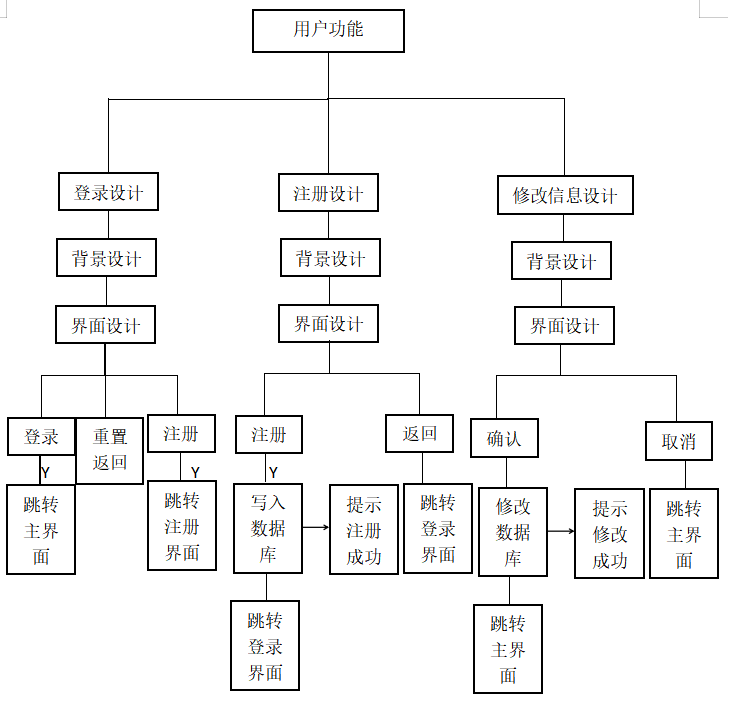


图4-1：用户功能图

4.2注册功能

### 4.2.1 注册功能实现

定义背景实现，是以panel面板，在面板内通过方法paintComponent在panel上画图片，然后把界面的控件加入panel面板里面，让面板隐藏。实现注册按钮的监听器是把账号和密码在客户端发送到服务端，然后服务端通过比较账号是否重复做出判断，如果重复就在服务端中通过键值返回-1到客户端，然后输出错误提示“用户名重复！”；没有重复就调用数据库的插入函数，把用户名和用MD5加密的密码写入数据库，然后在服务端以键值的形式返回0到客户端，显示注册成功，确认后关闭注册界面。返回按钮则是关闭注册界面。

### 4.2.2 注册注意事项

1.注册需填信息包括用户名和用户密码

2.用户名，密码不允许为空

3.判断用户名是否重复问题

4.用户名的长度不能大于20个字符，密码需要在6-20个字符

5.需输入两次密码，并且两次密码要相同

6.注册时写入数据库，密码要进行MD5加密

7.注册时写入数据库，需要通过服务端提供数据库的插入服务

4.3登录功能

### 4.3.1 登录功能实现

登录界面中的界面背景是建立一个Mypanel类，和注册界面的界面背景一样放置一个paintComponent方法，然后对控件添加到panel面板，将面板隐藏。建立JPasswordFieldHinListener类和JTextFieldHintListener类对账号和密码文本中添加提示语，当点击文本时，提示语消失，点击外部时，提示语会出现。

登录按钮的监听器是把账号和密码通过客户端发送到服务端，然后在服务端中调用数据库的用户密码判断语句，如果判断正确，服务端就会通过键值返回值，如果是0，则为没有此用户信息，则加错误提示语“没有这个用户！”；如果为-1，则为密码错误，输出对应的错误提示框；如果为1，则登录成功，登录界面关闭并且打开游戏界面。注册按钮则是打开注册界面，不关闭登录界面。

### 4.3.2 登录的注意事项

1.登录需填信息包括用户名和用户密码

2.用户名，密码不允许为空

3.用户名的长度不能大于20个字符，密码需要在6-12个字符

4.写入用户名输入框和密码输入框的提示语

5.登录时的密码判断，需要通过服务端提供数据库的查询服务

4.4修改信息功能

### 4.4.1修改信息功能的实现

先是定义背景实现，是以panel面板，在面板内通过方法paintComponent在panel上画图片，然后把界面的控件加入panel面板里面，让面板隐藏。实现确认按钮的监听器是先判断用户的旧密码是否相同，如果不相同就返回，相同则把账号和密码在客户端发送到服务端，然后服务端调用数据库的密码修改函数，通过键值返回值，如果值为1则修改成功，值为0则修改失败。

### 4.4.2修改信息的注意事项

1.修改信息需要填写的是用户新的密码

2.密码长度需要在6-12个字符

3.新密码需要进行2次填写

4.新密码写入数据库要进行MD5加密

5.修改信息时，需要通过服务端提供数据库的修改服务

5.音乐播放器

5.1背景音乐

设计思路：选定一首音乐，当你进入游戏时就会播放，无需自己按播放按钮。

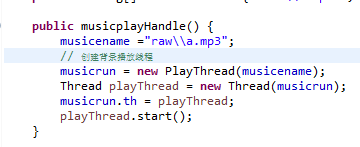
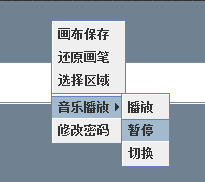


图5-1 播放音乐函数

5.2音乐功能位置

位于游戏主面板下方点击右键即可看到音乐播放，下列有播放，暂停，切换（下一首）。



5-2 界面功能选项

5.3音乐功能

### 5.3.1播放

播放按钮的作用是为了重启暂停的音乐，因为当你进入游戏时，就会有音乐。播放音乐功能的实现，是提前把歌曲的路径用字节数组存起来，当需要播放时，路径会直接载入播放线程里，用文件输入流和缓冲输入流把音乐载入，通过JL包相应代码播放。暂停后，继续播放实现，用暂停监听器保存的帧数，用相应代码即可播放从保存帧数到末尾的音乐。

### 5.3.2暂停

暂停按钮可使歌曲中途停下。暂停音乐功能的实现，当音乐播放player里播放音乐时，按下暂停，监听器监听此时帧数，为了后面继续播放保存帧数。判断是否为此状态，把播放状态调为0，通过JL包相应代码和AdvancedPlayer的作用，暂停音乐，关闭相关的所有输入流，清空player。

### 5.3.3切换

手动切换歌曲，把歌曲调至下一首。切换音乐功能实现，通过把载入播放线程里的路径改掉，以达成播放其他音乐的功能，停止当前播放的线程，把帧数调到0，开始播放。

5.4功能实现

JL包和AdcancedPlayer.JLayer是一个Java类库用来解码，转换，播放MP3文件。AdvancedPlayer是对player的一种拓展应用。

6. 游戏画布

6.1主界面交互特点

1.可进行画布时间轴的控制（暂停、开始、快速、慢速）

2.可以通过点击列表的方式来选择模板并预览

3.可以通过右键菜单进行更多交互

4.包括快捷键和按键多种交互选择

5.通过按键颜色来对功能进行具象化（列如不可撤销的按键会用高饱和度的颜色进行标注）

6.通过对话框对用户进行交互引导（用户可以选择打断或者暂停正在进行重要操作）

6.2．具体实现

### 6.2.1概述

将所有功能接口整合进UI，实现交互。主界面UI通过SWING进行编写，GamePage.java中UI代码占多数，核心的部分主要是选择列表和文本框下的Listener，这两项负担着用户对后台数据库的操作与反馈。

### 6.2.2列表

作用：接收数据库中的该用户GOL模板数据

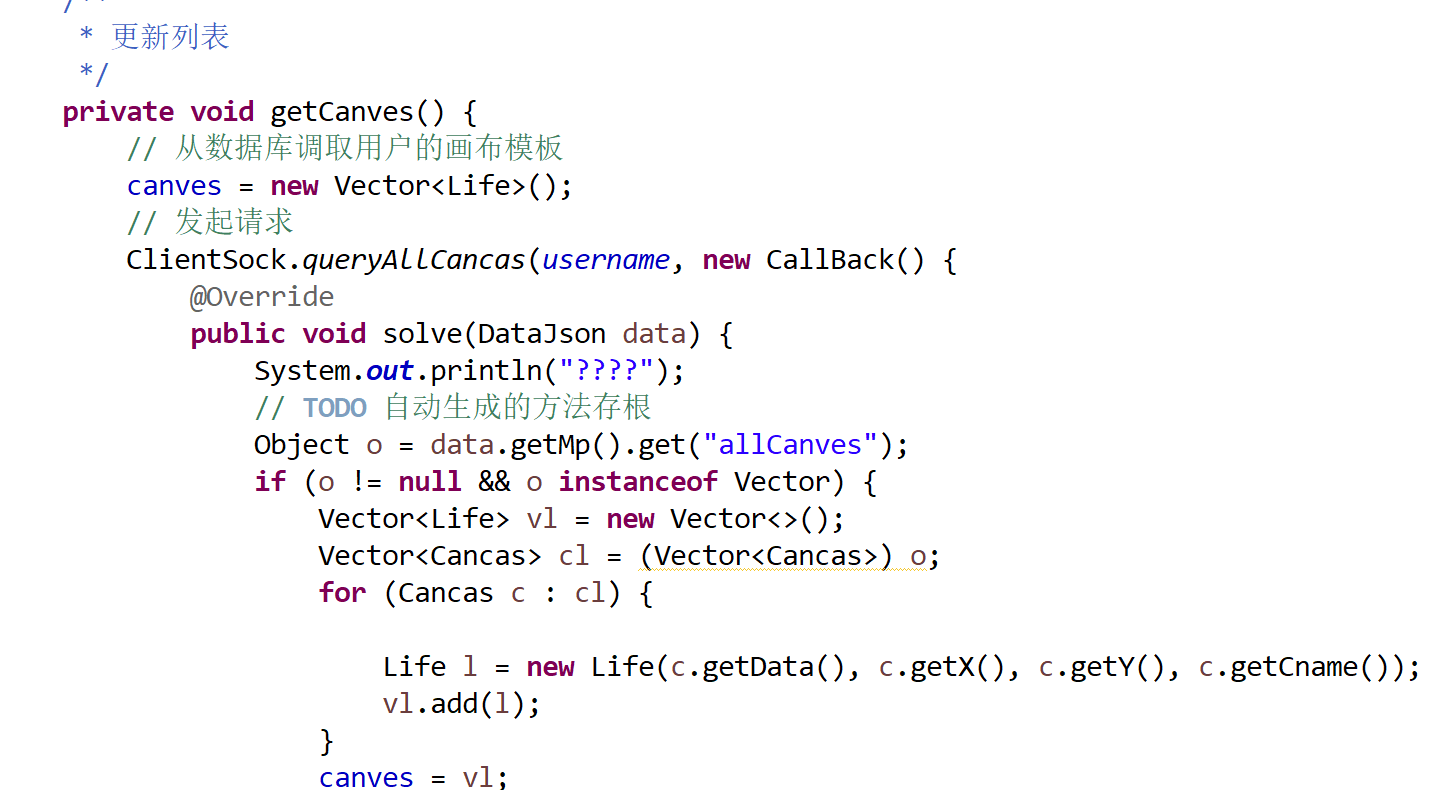


图6-1 列表函数

例如：每次更新列表都会调用getCanves()方法通ClientSock.queryAllCancas()对数据库访问，返回最新的画布数据。

### 6.2.2文本框

作用：对数据库中的GOL模板进行更改，在上传操作完成后，会更新列表数据，让用户能立刻使用刚才上传的模板。



图6-2 文本框函数

列如：

当点击提交按钮后，按钮监听器会首先通过调用对话框来确认用户是否进行或打断操作

当用户选择继续操作时，对话框会获得用户设定的模板名称并与文本框中的文本通过ClientSock.insertCanvas()打包后发送至服务器，在这之前会检查数据是否能够提交，如果数据在打包过程中出现错误会返回“失败”提醒

当数据成功打包提交后会调用getCanves()刷新列表，这样用户就可以马上使用刚刚所提交的模板了。

参考文献

[1]赵卓君等：《Java程序设计高级教材》 [B5] 清华大学出版社 2011

[2]AdvancedPlayer用法-EB/OL

http://www.javazoom.net/javalayer/docs/docs1.0/javazoom/jl/player/advanced/AdvancedPlayer.html

[3]音乐基础播放和JLayer包下载-EB/OL

<https://blog.csdn.net/xietansheng/article/details/85056358>

[4]JAVA GUI 开发---界面加载背景图片-EB/OL

<https://blog.csdn.net/weixin_34419326/article/details/91541909?utm_medium=distribute.pc_relevant.none-task-blog-baidujs-6>

[5]JAVA之设置背景图片的几种方法-EB/OL

<https://blog.csdn.net/huai814586181/article/details/45788263?utm_medium=distribute.pc_relevant_right.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase&depth_1-utm_source=distribute.pc_relevant_right.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase>

[6]波神 :Java socket之传输实体类对象-EB/OL

https://www.cnblogs.com/boshen-hzb/p/5891498.html

[7]张江:释放比特自由——Wolfram的“一种新科学”介绍-EB/OL

<http://swarmagents.cn.13442.m8849.cn/vm/articles/freebits.htm>

[8]混乱博物馆:生命游戏-EB/OL

<https://www.bilibili.com/video/BV1zx41187v3?from=search&seid=6256729695215427715>

[9]穆梓先生：java 双缓冲技术解决屏幕闪烁问题-EB/OL

<https://blog.csdn.net/l2533636371/article/details/80702062>

心得体会

完成这一个游戏最终的设计是通过小组五个人的配合与不断地实践才最终能展现一幅幅生命演化的画面。在小组进行分工后，每个人开始了自己部分的工作，利用git工具实现代码整合，对于不同的部分每个人都有不一样的心得体会。

罗坤：完成音乐设计这一部分的设计需要用到JLayer包，但是网上关于JLayer包的完整播放器模板近乎没有，一些相关的代码的用法也很难查到，所以在后期完善功能时，阻碍很大，只能借鉴网上其他拓展包和媒体文件的用法，自己尝试完善功能，其中得到了同小组的成员的想法和帮助，达到基本的目标。虽然在这次小组合作中，负责的份额很少，但是依然是个艰巨的任务，从零学起如何运用Java播放音乐的知识。

蔡梓沁：这一次实验主要完成的是数据库部分，在最初分配任务时由于对数据库比较感兴趣且选修了高级数据库编程，也希望能够学习更多关于数据库与各平台之间的相关操作所以选择了这一部分。这一次的游戏设计也是对上学期Java数据库部分知识点进行复习同时考虑如何在这个基础上去拓展更多的功能。在完成数据库部分后，也帮忙写了服务端的一个搭建部分，利用到的知识点和我们这学期最后一章的内容相关，也在实践中积累了经验，知道这个操作用什么样的方法去实现会更方便更好。团队间的配合也很好，有问题大家也会一起讨论如何去解决，一起想办法把我们这个游戏做的更好。

陈杰伟：本次的大实验中，我是扮演了一个GUI界面设计者以及客户端与服务端信息传递的一个java编写者，在设计界面背景时遇到了知识盲区，因此在设计界面上参考了很多博客的界面设计，工程量对我来说应该是有点大的，但毕竟大部分不是知识盲区，所以设计起来还算是能理解的。在设计界面的过程中，我学到了不少关于GUI设计的知识，包括界面背景，界面的数据库数据保存，socket的引用方法等。在gitee的运用下，我们小组成员之间的配合更加默契，同时每次在“工作”时，小组成员都能用腾讯会议进行问题讨论和提出设计思路，使我切实感受到开发团队的氛围，我很享受，也很热爱这种氛围。

王若雨：这次Java的项目合作让我比以往更加熟悉了SWING和刚刚学习的GIT的使用与操作。第一次以多人合作的方式去完成一个项目的开发和测试让我受益匪浅。完全通过Java去尝试制作UI也是一次苦涩的经历，让我充分认识到了swing在UI上的限制。通过整合其他组员的功能接口并调试组合让我对项目的分层结构有了自己的一定认知，平时老师所讲到的一些无法客观理解的道理也或多或少的有了些许解答。

刘堉嘉：这一次大实验小项目，让我受益匪浅，在团队协作上，每次上课开会进行阶段分析协作，讨论项目各个方面怎么去实现，分工，让我们的团队协作能力有了很大的锻炼。特别是采用Git进行项目管理，对团队项目编码规范等，有了进一步的理解，也认识到了git协作整合的好处。在项目模块的开发中，对代码的规范化，对面向对象的思想有了进一步的提升，特别是在设计构建的过程中，对解耦、内部类、多线程、异步、等知识点理解更加深刻。

在未来，如果有机会我们希望我们真的能够去参与一款游戏的设计，也希望学习更多关于游戏设计的知识，这一次期末项目对我们来说是一个非常好的实践，也收获了不少课本外的知识。

教师评语

# 答辩表汇总

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 专业学院 | | | | 计算机学院 | | | | 专业 | | | 软件工程 | | | 姓名 | | | | 刘堉嘉 | | 学号 | | 180021105120 | | | |
| 课程报告  题目 | | | | 生命游戏 | | | | | | | | | | | | | | | | | | | | | |
| 答辩日期 | | | | 2020.6.23 | | | | | | | | | | | | 答辩时间 | | | | 21:00 | | | | | |
| 答  辩  提  问  及  其  回  答  记  录 | | 1. 问：图案是怎么加密的？   答：图案加密是采用类似图像压缩算法，记录连续相同的格子的数量，同时兼任专业软件Golly的加密密文,采用$代表回车b代表死亡，o代表生命，然后在对应字符前面写其长度去存储，矩阵变化。   1. 问：怎么实现客户端异步连接？   答： 实现回调接口，客户端编写代码实现回调接口 ，通过clientSock 进行API http封装， 开启线程请求后获取数据后， 使用接口函数实现函数回调 。   1. 问：主要实现了哪些功能？   游戏画布部分的游戏功能。 | | | | | | | | | | | | | | | | | | | | | | | |
| 成绩评定 | | **成绩评定权重** | | | | | | | | | | | | | | | | | | **总分** | | | | **总成绩**  **（五分制）** | |
| **考勤成绩**  **5** | | | | **课内实验作业**  **25** | | | | | | **期末成绩**  **（含作品20、报告20、答辩30）**  **70** | | | | | | | |
|  | | | |  | | | | | |  | | | | | | | |  | | | |  | |
| 评语 | | 教师签名：    日期： | | | | | | | | | | | | | | | | | | | | | | | |
| 专业学院 | | | | 计算机学院 | | | | 专业 | | 软件工程 | | | | 姓名 | | | | 罗坤 | | 学号 | | 180021102400 | | | |
| 课程报告  题目 | | | | 生命游戏 | | | | | | | | | | | | | | | | | | | | | |
| 答辩日期 | | | | 2020.6.23 | | | | | | | | | | | | 答辩时间 | | | | 21:00 | | | | | |
| 答  辩  提  问  及  其  回  答  记  录 | | 1. 问：音乐是怎么播放的？   答：通过JLayer包音乐播放代码即可播放。   1. 问：音乐是怎么切换的？   答：可以用路径直接带入播放线程，然后播放器就会重启播放线程，切换到下一个路径的音乐播放。   1. 问：音乐播放有用到线程吗？   答：有，音乐播放的时候就是用到的线程。   1. 问：可以直接在本地获取音乐吗？   答：起初的版本是这样的，但是音乐多起来之后实现较为繁琐，便改成定义一个string保存对应路径。 | | | | | | | | | | | | | | | | | | | | | | | |
| 成绩评定 | | **成绩评定权重** | | | | | | | | | | | | | | | | | | **总分** | | | | **总成绩**  **（五分制）** | |
| **考勤成绩**  **5** | | | | **课内实验作业**  **25** | | | | | **期末成绩**  **（含作品20、报告20、答辩30）**  **70** | | | | | | | | |
|  | | | |  | | | | |  | | | | | | | | |  | | | |  | |
| 评语 | | 教师签名：    日期： | | | | | | | | | | | | | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 专业学院 | | 计算机学院 | | 专业 | 软件工程 | | 姓名 | | 陈杰伟 | 学号 | 180021102316 | |
| 课程报告  题目 | | 生命游戏 | | | | | | | | | | |
| 答辩日期 | | 2020.6.23 | | | | | | 答辩时间 | | 21：00 | | |
| 答  辩  提  问  及  其  回  答  记  录 | 1. 用户加入数据库时，密码是通过明文加入还是加密加入？   答：用户加入数据库时，采用的是MD5加密手段。  2、登录功能是怎么实现的？  答：首先在用户界面，如果用户没有账号就先注册，然后在登录界面登录。登录时，客户端会将账号和密码发送到服务端，然后服务端通过调用数据库的用户查询块，通过用户名找相应的密码与发送的密码进行匹配，如果正确，服务端就返回1，用户登录成功；如果返回-1，则密码错误；返回0，则没有这个用户。 | | | | | | | | | | | |
| 成绩评定 | **成绩评定权重** | | | | | | | | | **总分** | | **总成绩**  **（五分制）** |
| **考勤成绩**  **5** | | **课内实验作业**  **25** | | | **期末成绩**  **（含作品20、报告20、答辩30）**  **70** | | | |
|  | |  | | |  | | | |  | |  |
| 评语 | 教师签名：    日期： | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 专业学院 | | 计算机学院 | | 专业 | 软件工程 | | 姓名 | | 王若雨 | 学号 | 180410106818 | |
| 课程报告  题目 | | 《生命游戏》 | | | | | | | | | | |
| 答辩日期 | | 2020.6.23 | | | | | | 答辩时间 | | 9.00 PM | | |
| 答  辩  提  问  及  其  回  答  记  录 | **问：举一个你所负责的交互逻辑的例子？**  答：当用户焦点选中文本框并将内容输入进去的时候，点击上传按钮后会显示提示给用户打断操作的机会，如果不打断操作继续进行则需要用户必须填写作品名称才能提交到数据库，这有效的将数据的提交和数据格式的规范集合在了一个操作动作上，节约了主页面的使用空间并简化了编写难度，所有的用户操作设计都遵从着这套逻辑。 | | | | | | | | | | | |
| 成绩评定 | **成绩评定权重** | | | | | | | | | **总分** | | **总成绩**  **（五分制）** |
| **考勤成绩**  **5** | | **课内实验作业**  **25** | | | **期末成绩**  **（含作品20、报告20、答辩30）**  **70** | | | |
|  | |  | | |  | | | |  | |  |
| 评语 | 教师签名：    日期： | | | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 专业学院 | | 计算机学院 | | 专业 | 软件工程 | | 姓名 | | 蔡梓沁 | 学号 | 180021100969 | |
| 课程报告  题目 | | 《生命游戏》 | | | | | | | | | | |
| 答辩日期 | | 2020.06.23 | | | | | | 答辩时间 | | 21:00 | | |
| 答  辩  提  问  及  其  回  答  记  录 | 1. 数据库怎么存储画布信息？   答：通过调用数据库部分的画布信息插入函数（insertCanvas()）将用户的信息插入到用户数据库中的Canvas表中。   1. 画布信息的存储数据格式是怎样的？   答：画布设计如下图所示：  2  存储的数据格式如下图所示（模板表）：  1   1. 使用的是什么数据库？   答：SQL Server。 | | | | | | | | | | | |
| 成绩评定 | **成绩评定权重** | | | | | | | | | **总分** | | **总成绩**  **（五分制）** |
| **考勤成绩**  **5** | | **课内实验作业**  **25** | | | **期末成绩**  **（含作品20、报告20、答辩30）**  **70** | | | |
|  | |  | | |  | | | |  | |  |
| 评语 | 教师签名：    日期： | | | | | | | | | | | |