

高级语言程序设计

课设报告

学 号： 周子渔

姓 名： 20074214

指导教师： 周琨

提交日期：

成绩评价表

报告内容	报告结构	流程图	程序的结构	最终成绩
<input type="checkbox"/> 丰富正确 <input type="checkbox"/> 基本正确 <input type="checkbox"/> 设计部分偏少 <input type="checkbox"/> 过于简单	<input type="checkbox"/> 完全符合要求 <input type="checkbox"/> 基本符合要求 <input type="checkbox"/> 有欠缺	<input type="checkbox"/> 符合规范 <input type="checkbox"/> 基本符合规范 <input type="checkbox"/> 有一些错误	<input type="checkbox"/> 结构合理、清晰 <input type="checkbox"/> 函数划分基本合理 <input type="checkbox"/> 部分函数过大 <input type="checkbox"/> 结构较乱	
程序功能	程序执行情况	回答问题情况	其他	
<input type="checkbox"/> 完成基本功能 <input type="checkbox"/> 少量扩展功能 <input type="checkbox"/> 多个扩展功能 <input type="checkbox"/> 未完成基本功能	<input type="checkbox"/> 顺畅 <input type="checkbox"/> 界面美观 <input type="checkbox"/> 按钮效果好 <input type="checkbox"/> 界面有闪动 <input type="checkbox"/> 操作卡顿	<input type="checkbox"/> 清楚、正确 <input type="checkbox"/> 基本正确 <input type="checkbox"/> 回答有部分错误 <input type="checkbox"/> 不能回答问题		

教师签字：_____

目录

1 需求分析.....	3
1.1 功能.....	3
1.1.1 基本功能.....	3
1.1.2 扩展功能.....	4
1.2 数据.....	4
1.3 界面.....	4
1.3.1 欢迎界面.....	4
1.3.2 初始化界面.....	5
1.3.3 游戏界面.....	6
1.4 开发与运行环境.....	6
2. 概要设计.....	6
2.1 主要数据结构.....	7
2.2 程序总体结构.....	7
2.2.1 信息介绍及初始化.....	7
2.2.2 按钮控制.....	8
2.2.3 飞艇的管理.....	8
2.2.4 区分回答正误.....	9
2.2.5 鼠标操作.....	9
2.2.6 时间控制.....	9
2.2.7 文件管理.....	9
3 详细设计.....	10
3.1 主程序的流程图.....	10
3.2.1 信息介绍及初始化.....	10
3.2.2 按钮控制.....	11
3.2.3 算术题的管理.....	11
3.2.4 区分回答正误.....	11
3.2.5 鼠标操作.....	12
3.2.6 时间控制.....	12
2.2.7 文件管理.....	12
4 测试.....	13
4.1 鼠标控制点击按钮测试.....	13
4.2 信息显示测试.....	13
4.3 时间信息测试.....	14
4.4 算术题生成测试.....	14
4.5 按钮功能测试.....	14
5 开发中遇到的问题及解决办法.....	16
6 总结.....	17
6.1 课程设计总结.....	17
6.2 对本课程意见与建议.....	17
附件：程序源代码.....	18

1 需求分析

用 C 语言编写一个简单的“导弹算术题”游戏，答对一题得 1 分，没答对减 1 分，时间到游戏结束，可以显示目前答对的题、答错的题以及剩余的题。

1.1 功能

游戏区里算数题目从界面上方落下，题目内容通过链表储存，以随机数的形式生成。控制区有“开始”、“暂停”、“跳过”、“查看”、“退出”五项功能，表现为按钮，通过鼠标控制左键敲击实现各自功能。显示区显示有玩家的用户名、游戏剩余时间、给出的答案、当前得分，前三项内容由键盘输入，得分由系统计算显示。

1.1.1 基本功能

1、输入（用户名、游戏时长、题目数量）

程序运行时先输入用户名，再输入游戏时长，再输入试题的数量。将输入的三项信息存入结构体中。输入的试题数量作为参数传入创建题目的函数，以此决定显示在游戏区的题目的数量。输入的时间作为参数传入倒计时的函数，决定游戏的总时长。

2、显示（背景、按钮、题目）

a.游戏区：获取文件中的背景图，以固定大小将其显示在屏幕上；题目以“ $a \pm b$ ”的形式显示在背景之上，字体字号颜色固定、位置会随时间的变化而改变，但并不跳出游戏区的左右边界；图像“鱼”作为题目的标识，位于题目上方，和题目以同样的速度和同样的方向移动。

b.控制区：绘制图线以区分游戏区显示区和控制区；绘制按钮,当有鼠标敲击的时候，实现分别按钮所示的功能。

c.显示区：将游戏开始之前输入的用户名、游戏时间显示出来：其中用户名保持不变，时间部分以倒计时的形式显示，直到时间为 0 结束；显示两个文本框，将游戏过程中键盘输入的答案和系统判断的得分实时显示在文本框内，输入答案显示后自动清空。

3、按钮（开始、暂停、跳过、查看、退出）

开始：点“开始”按钮，则游戏开始，第一道数学题从顶部出现下落，斜线下落，触及游戏区左右边界时反弹，但总体趋势还是下落直到消失在底部。前一题开始下落后，后面的题也逐个出现，会同时出现几道题在一个界面内的情况。运动轨迹与前一题类似。点击后倒计时开始执行。

暂停：点“暂停”按钮，题目和“鱼”保持静止不动，倒计时暂停。

跳过：点“跳过”按钮，位于游戏区最下方的题目和“鱼”同时消失，随后会重新从顶部下落。

查看：点“查看”按钮，在屏幕中间显示一个新界面，显示出答对的题、答错的题和还没答的题，再次点击“查看”按钮则退出此页面，游戏继续。在查看过程中，题目保持不动，倒计时暂停。待游戏继续进行后再计时。

退出：点“退出”按钮，游戏结束，退出界面并储存信息。

4、计时

以输入的时间为起点进行倒计时，在游戏中断即没有继续运行（暂停、查看）的情况下

计时停止。

1.1.2 扩展功能

- 1、暂停不计时。
- 2、当按下“退出”按钮后，将当前游戏信息（用户名、时间、分数、做题情况）储存在文件中。
- 3、点击按钮的时候有变色特效。

1.2 数据

用户信息用结构体储存，包括用户名、得分、设定的做题时间和题目数量。

1.3 界面

设计 3 个界面，一个是欢迎界面，一个是初始化界面，一个是游戏主界面。

1.3.1 欢迎界面

欢迎界面会显示出游戏的名称“导弹算术题”，同时下方出现两个按钮，分别为“START”和“INTRODUCTION”，点击前者则进入后面初始化信息的界面，点击后者则弹出一个文本框，内容为此游戏的规则，玩家阅读规则完毕后可以继续点击右下方“continue”按钮进入后面初始化信息界面。欢迎界面如图 1.1 所示，其后续规则介绍界面如图 1.2 所示。



图 1.1 欢迎界面

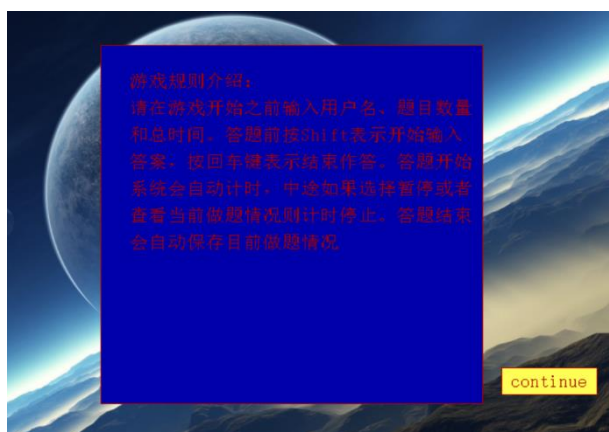


图 1.2 规则介绍界面

1.3.2 初始化界面

初始化界面包含三个文本框，分别需要从键盘读入信息。按照顺序为“用户名”（图 1.3）、“题目数量”（图 1.4）、“时间”（图 1.5）进行读入，输入完毕后需要点击“确认”按钮来完成下一次输入。

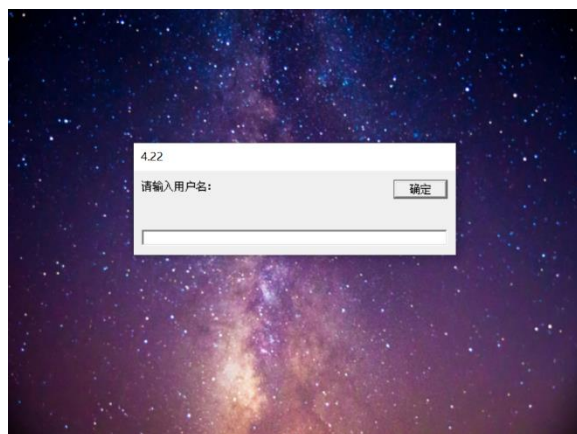


图 1.3 输入用户名界面

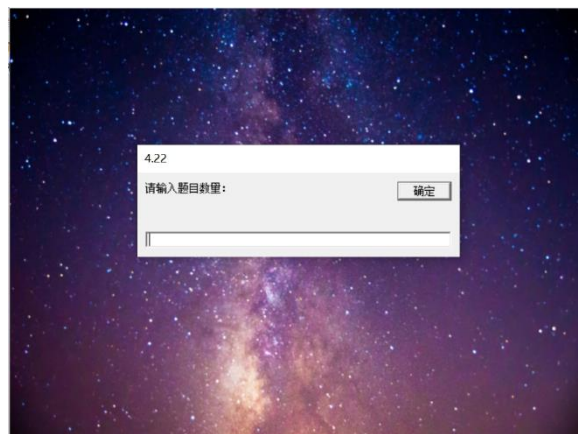


图 1.4 输入题目数量界面

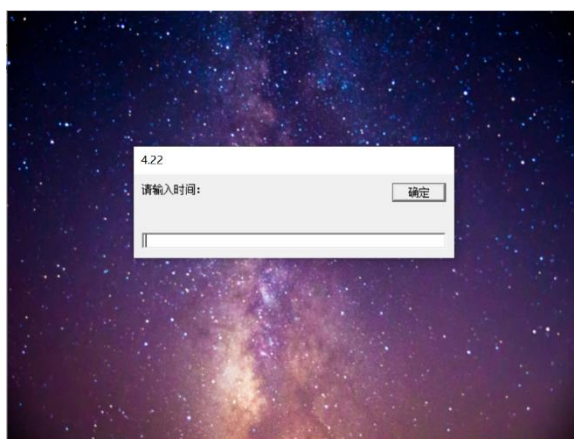


图 1.5 输入时间界面

1.3.3 游戏界面

界面共分为三个区域，分别为“游戏区”、“控制区”和“显示区”。游戏区显示算术题，控制区包括“START”、“PAUSE”、“SKIP”、“REVIEW”、“LOGOUT”五个按钮，鼠标点击后分别执行“开始”、“暂停”、“跳过”、“查看”、“退出”五项功能，显示区则显示当前玩家的用户名、游戏剩余时间、给出的答案和当前得分四项信息。游戏界面如图 1.6 所示。

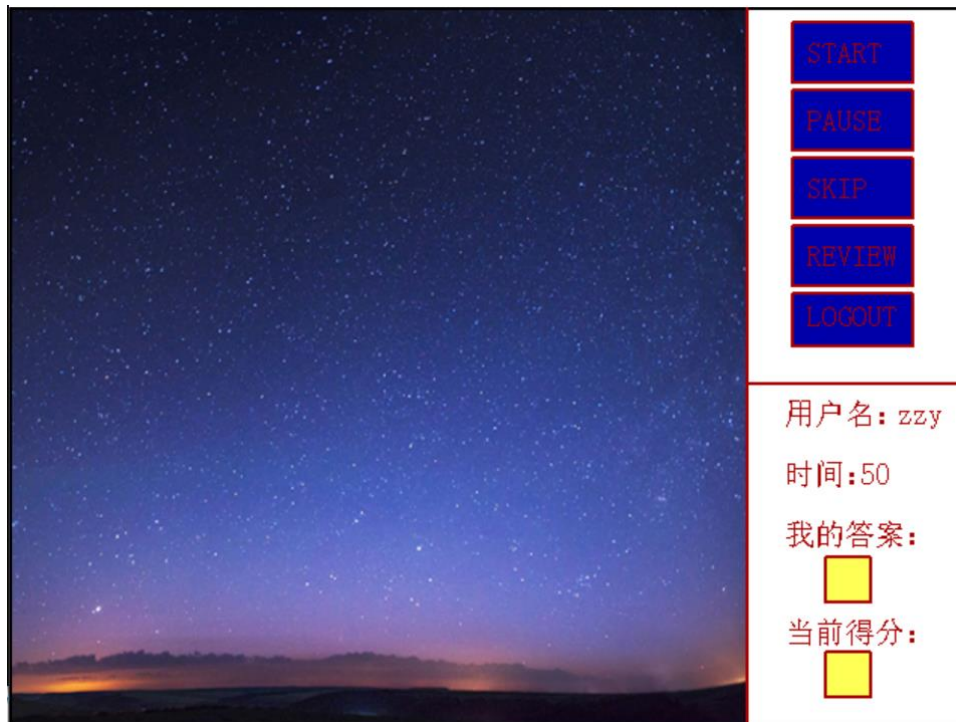


图 1.6 游戏界面

1.4 开发与运行环境

Visual studio 2019 和 EasyX2021 版。

2. 概要设计

根据分析，划分 8 个模块，有信息介绍、初始化、按钮控制、计算题管理、区分回答正误的题目、鼠标操作、时间控制、文件管理。

2.1 主要数据结构

1、用户结构，结构体储存用户的姓名、得分、时间、题目数量

```
typedef struct user {  
    char name[20];  
    int  score = 0;  
    char time[10];  
    char num[10];  
}USER;
```

2、算术题结构， 用链表储存需要运算的数据、运算符、以及题目的位置坐标信息

```
typedef struct node {  
    int data1, data2, ans;  
    int locx;  
    int locy;  
    char sign;  
    struct node* next;  
}NODE;
```

2.2 程序总体结构

程序的总体结构包含以下几个方面，如图 2.1 所示：

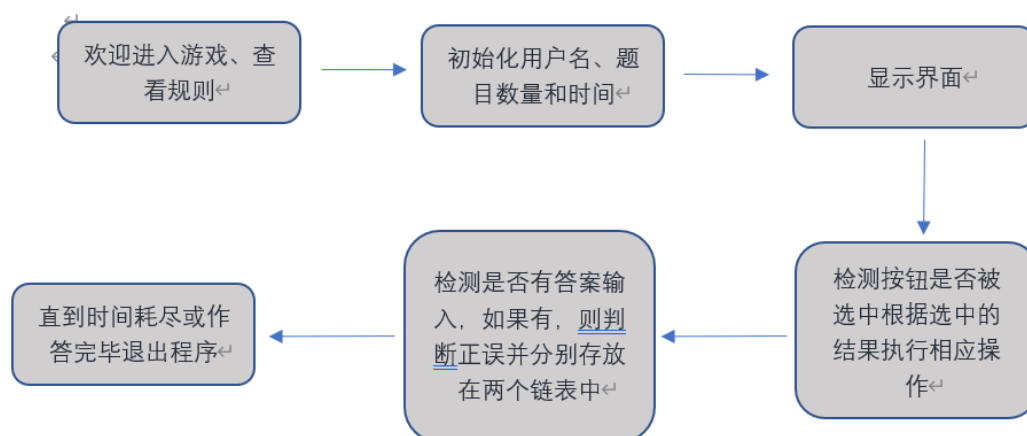


图 2.1 程序的总体结构

2.2.1 信息介绍及初始化

本模块用于显示初始化界面输入用户名、题目数量和游戏时间，随后显示游戏界面。

相关函数：

- ◆ void showrule()
功能：展示规则
参数：无
返回值：无
说明：无
- ◆ void initname (USER& user)
功能：输入用户名
参数：USER &user
返回值：无
说明：输入的用户名存到了 user.name 中
- ◆ void initnum (USER& user)
功能：输入题目数量
参数：USER &user
返回值：无
说明：输入的用户名存到了 user.num 中
- ◆ int inittime (USER& user)
功能：输入时间
参数：USER &user
返回值：Time
说明：输入的时间存到了 user.time 中，将初始化给定的时间传入主函数计时功能相应模块中。

2.2.2 按钮控制

- ◆ void displayButton()
功能：绘制按钮
参数：无
返回值：无
说明：将五个按钮显示在控制区
- ◆ int checkButton(int mouseX, int mouseY)
功能：检测鼠标点击位置
参数：int mouseX, int mouseY
返回值：具有符号作用的数字
说明：无

2.2.3 飞艇的管理

本模块用于和算术题相关的操作

- ◆ NODE *create(USER &user)
功能：创建储存算术题的链表
参数：USER &user
返回值：头指针
说明：将生成的算术题存入链表

2.2.4 区分回答正误

- ◆ void judgeT(NODE *p, NODE*hT)
功能：将回答正确的题目存入链表 hT 中
参数：NODE *p, NODE*hT
返回值：无
说明：无
- ◆ void judgeF(NODE *p, NODE*hF)
功能：将回答错误的题目存入链表 hF 中
参数：NODE *p, NODE*hF
返回值：无
说明：无

2.2.5 鼠标操作

- ◆ int checkmouse()
功能：检测是否有鼠标敲击
参数：无
返回值：checkButton 的返回值
说明：无

2.2.6 时间控制

- ◆ int stoptime(int &t)
功能：暂停时停止计时
参数：int &t
返回值：输出的时间
说明：无

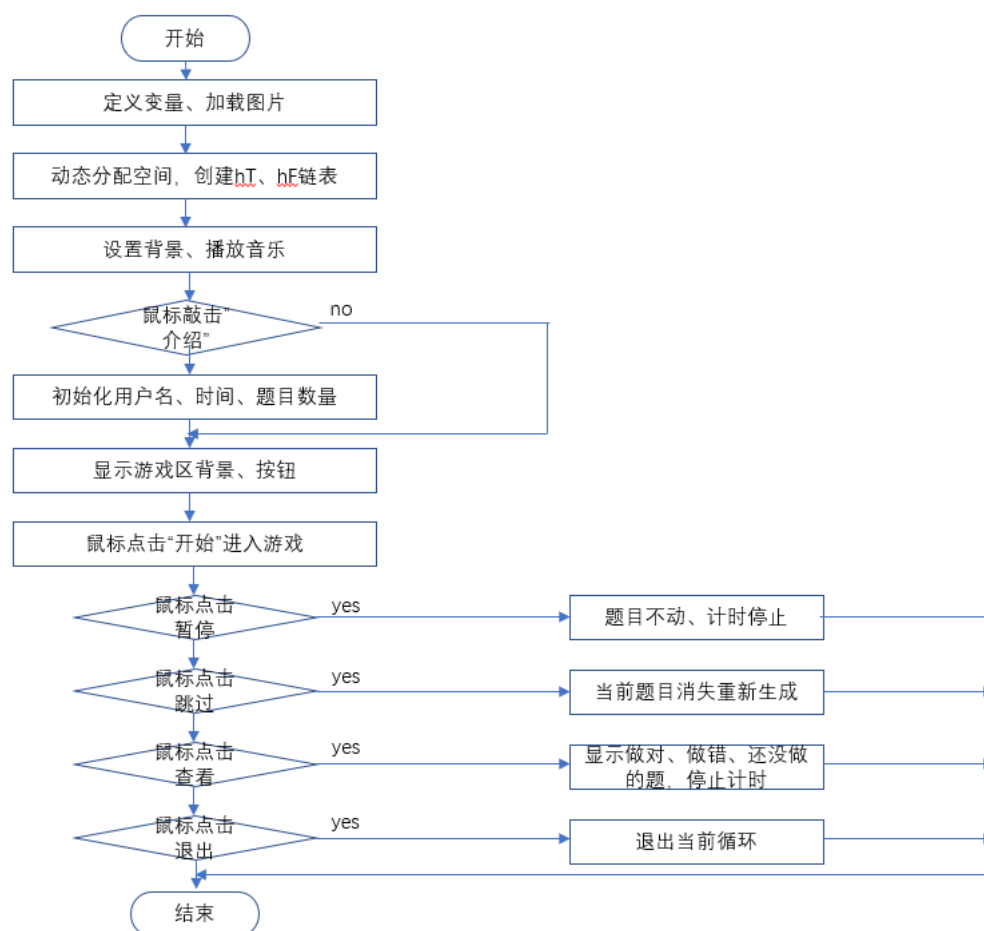
2.2.7 文件管理

- ◆ void save(USER& user, NODE* hT, NODE* orip, NODE* hF, int t1)
功能：退出时储存信息
参数：USER& user, NODE* hT, NODE* orip, NODE* hF, int t1
返回值：无
说明：无

3 详细设计

…输入姓名、游戏时间，完成初始化后，程序进入无限循环，直到选择了退出按钮，程序结束。在这个循环中，检测鼠标的操作，根据操作完成相应的功能。

3.1 主程序的流程图



3.2 模块的详细设计

3.2.1 信息介绍及初始化

本模块用于显示初始化界面输入用户名、题目数量和游戏时间，随后显示游戏界面。

相关函数：

◆ void showrule()

功能：当用户在最开始的欢迎界面点击“INTRODUCTION”按钮后，系统会弹出一个文本框，显示游戏的规则。用户阅读完毕后可以点击右下“continue”按钮前往下一步。设置此功能旨在让用户尽快明白游戏规则和操作方法。

接下来是三项信息初始化的内容，每输入一个信息后需要点击界面上的“确认”按钮继续。

- ◆ void initname (USER& user)

功能：屏幕上会弹出一个文本框，用户可以在其中输入用户名，输入的用户名存入 user.name 中。

- ◆ void initnum (USER& user)

功能：屏幕上会弹出一个文本框，用户可以在其中输入用户名，输入的题目数量存入 user.num 中。

- ◆ int inittime (USER& user)

功能：屏幕上会弹出一个文本框，用户可以在其中输入用户名，输入的时间存入 user.time 中。

3.2.2 按钮控制

- ◆ void displayButton()

功能：将设置好位置、字体、文字颜色、填充颜色和填充大小的按钮显示在控制区，同时绘制区分三个区域的直线。

- ◆ int checkButton(int mouseX, int mouseY)

功能：根据传入的鼠标横纵坐标判断鼠标点击位置是否对应五个按钮之一，如果是的话返回一个相应的数字，在主程序中这个返回的数字将作为信号决定继续进入哪部分程序。

3.2.3 算术题的管理

本模块用于和算术题相关的操作

- ◆ NODE *create(USER &user)

功能：利用尾插法，创建一个带头结点的链表并返回头结点。根据输入的题目数量 user.num 决定循环的次数，每次循环生成两个随机数 p->data1 和 p->data2，同时利用随机数生成运算符。判断当运算符为减号时，如果被减数小于减数即运算结果为负数，则令二者交换顺序，因此能保证结果恒正。同时，在创建题目的同时设定其坐标，从第一道题到第 user.num 道题坐标以一定差值递减，效果即所有题都创建完毕并显示在一个较大的区域内，随着程序的进行题目会依次进入游戏区显示出来。

3.2.4 区分回答正误

- ◆ void judgeT(NODE *p, NODE* hT)

功能：在主程序中判断如果回答结果正确，则执行此函数，用链表 hT 存储刚刚回答的计算题。

◆ void judgeF(NODE *p, NODE*hF)

功能：在主程序中判断如果回答结果错误，则执行此函数，用链表 hF 存储刚刚回答的计算题。

3.2.5 鼠标操作

◆ int checkmouse()

功能：当有鼠标敲击的时候，用 mmsg.x 和 mmsg.y 接受鼠标左右键的敲击，再调用 checkButton 函数确定点击的具体位置，执行具体功能。

3.2.6 时间控制

◆ int stoptime(int &t)

功能：当点击特定功能的按钮即暂停和查看做题时，这段时间没有进行游戏所以不应该计时，因此 stoptime 这个函数可以在暂停的这个时间让程序进入一个死循环，同时进入的时候开始计时，退出循环时将开始计时的这段时间加到屏幕上显示的倒计时中，这样就实现了暂停不计时的功能。

2.2.7 文件管理

◆ void save(USER& user, NODE* hT, NODE* orip, NODE* hF, int t1)

功能：点击退出后将当前做对的题、做错的题和没做的题记录到文件中。同时记录的还有用户名、游戏结束的时间。在文件夹“game”中可以查看相关信息。

4 测试

对程序从下面几部分进行了测试。

4.1 鼠标控制点击按钮测试

测试用例：在欢迎界面点击“INTRODUCTION”按钮，如图 4.1 所示。

测试结果：会调用 checkmouse 函数进入介绍规则的界面，如图 4.2 所示。



图 4.1

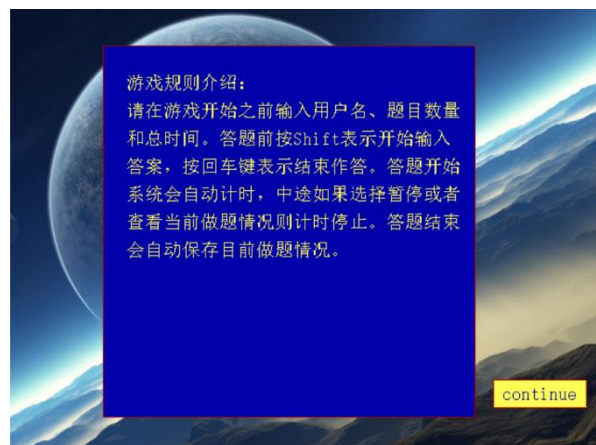


图 4.2

4.2 信息显示测试

测试用例：在弹出框中输入用户名，如图 4.3 所示。

测试结果：完成信息初始化后游戏界面中显示出用户名，如图 4.4 所示。

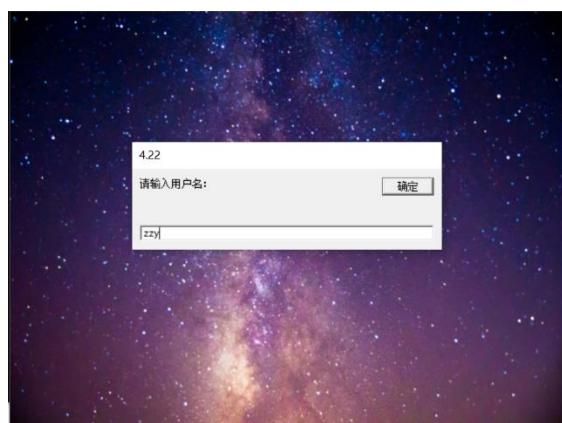


图 4.3



图 4.4

4.3 时间信息测试

测试用例：在弹出框内输入游戏需要的时间，如图 4.5 所示。

测试结果：游戏区会显示时间，并且暂停不计时，如图 4.6 所示。

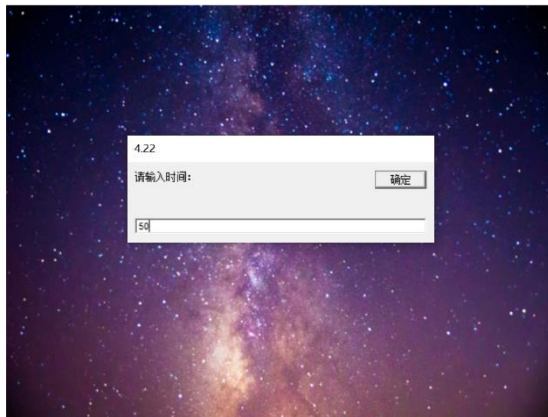


图 4.5



图 4.6

4.4 算术题生成测试

测试用例：在弹出框内输入算术题的数量，进入游戏界面后，点击“开始”。

测试结果：随机生成的算术题从界面顶部斜线落下，如图 4.7 所示。

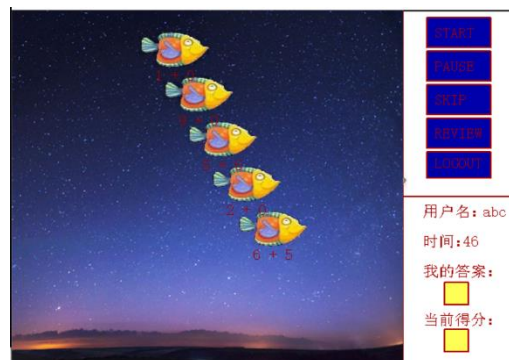


图 4.7

4.5 按钮功能测试

测试用例：进入游戏界面后，点击“开始”游戏开始运行。分别点击“暂停”、“跳过”、“查看”、“退出”。

测试结果：点击“暂停”后，算术题保持不动，计时停止如图 4.8 所示；点击“跳过”后，位于最下方的算术题从屏幕上消失，随后再次从界面顶部出现如图 4.9 所示；点击“查看”后，显示出一个文本框，内含目前的回答情况；点击“退出”后退出游戏，如图 4.10 所示。



图 4.8 题目保持不动，计时停时

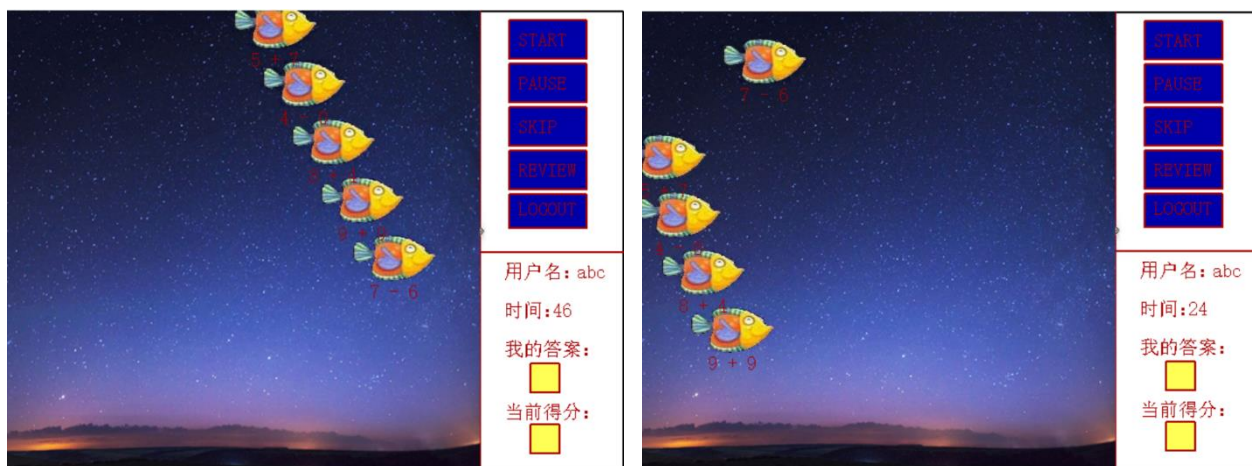


图 4.9 跳过的题目立刻从界面中消失 随后从顶部重新显示



图 4.10 显示当前做题情况

5 开发中遇到的问题及解决办法

开发中遇到了一些问题，并分别通过不同的途径进行解决。

1. 问题描述：在初始绘制界面的过程中，经常出现背景加载时频闪的现象，文本框从上方落下会有卡顿。

解决方法：出现频闪可能是清屏和绘图的之间存在时间间隔，导致屏幕上存在短暂留白。针对这个问题，我采取的方法是新建一个空项目，探索 `BeginBatchDraw()`、`FlushBatchDraw()`、`putimage()`、`Sleep()` 这四个函数的调用规律，最终找出了函数之间最佳排列顺序及其与主循环 `while(1)` 之间的关系，从原理上解决了绘图频闪的问题。

2. 问题描述：在创建链表的过程中，针对带头结点的链表原理理解不清，在编写 `create()` 函数时耗费较长时间。

解决方法：重新复习上个学期所学创建链表的尾插法，结合开学初老师所讲带头结点链表的创建相关内容，检查自己代码的问题。反复多次调试创建链表的三个函数，深入理解分离链表并插进新链表的方法原理，最终实现理想效果。

3. 问题描述：在完成计时功能的时候，对计时原理理解不清，对显示时间的原理理解不清，为实现该功能花费较长时间。

解决方法：在反复调试无果的情况下，我选择向学长求助。他为我提示了完成计时功能的思路，经过后来测试发现使用 `difftime` 函数与否都可以完成该功能。同时也发现时间相关变量最好统一用 `clock_t` 函数定义，如果同时用 `time` 和 `clock` 函数有时会出现问题。

4. 问题描述：在完成从键盘读入答案功能时，没有找到最合适最快速的方法，导致显示在屏幕上的输入值经常出现乱码。

解决方法：前往 CSDN 上查找关于 `getch()` 函数的相关信息，发现其接受返回值应为一字符数组，新定义一个 `char a[20]` 来接受输入的值后才可以使使用。同时，关于这个函数需要人工唤醒其读入键盘信息的功能，即通过按 `shift` 键后进行输入操作，答案才显示在屏幕上。

6 总结

6.1 课程设计总结

满打满算两个半月的课程设计即将结束，回望过去的这段日子可以算是收获颇丰了。在整个程序编写的过程中，随着研究的逐渐深入和遇见的 bug 逐渐复杂，我越来越理解老师在指导书中所写“本课程设计的主要目的是通过一个趣味盎然的游戏程序的设计与实现使学生亲身经历一个对初学者而言较为复杂的程序的设计与开发过程。”这句话的含义，我也在不断地发现问题和解决问题的同时提高了自己对程序设计的认识，提高了自己对输入输出函数、链表、文件等重要 c 语言知识的掌握程度，为以后更复杂、难度更大的程序开发和科研工作奠定了良好基础。

对于结果，我只能给自己评价 75 分左右，因为最终呈现的效果并不甚尽如人意。首先，计算题之间并没有错落有致的感觉，而是统一轨迹和速度落下，影响了游戏的趣味性。其次，界面设计有些简略，观赏性方面做的并不到位。再其次，读档功能并没有实现，只是实现了存档功能，整个游戏最终给人一种不是很完整的游戏体验，功能不够齐全。

当然，在整个设计的过程中也存在很多我自己较为满意的地方。比如无论是 CSDN 还是 github，我都能在遇到问题和 bug 的时候知道应该去什么地方求助，也逐渐学会了突出特点地描述问题向他人求助，这在我整个推进进度，一个又一个实现功能的过程中起到了很重要的作用。再比如我锻炼了自己模块化程序设计的思路，上个学期是单纯熟悉 c 语言知识点，而如今的课设就需要将某些知识点融合在一起，将解决游戏设计的问题拆分成编写若干具有特定功能函数的问题。我也在其中逐渐意识到全篇布局、统筹规划在程序设计中的重要性，这种思维模式将帮助我在之后的程序设计学习工作中走得更远。

6.2 对本课程意见与建议

建议老师在今后该课程的开展过程中多安排经验交流的展示，比如可以每周针对同学们编程中常见问题进行讲解，这样可以大大提高同学们的完成效率。建议老师在课设结束后发布一则调查问卷，收集今年同学们在课设过程中容易出现的 bug 或者错误，以及 vs2019、EasyX 使用上的一些方法，这样也可以为下一届同学解决不少客观阻碍。

附件：程序源代码

```
#include<graphics.h>
#include <conio.h>
#include<stdio.h>
#include<stdlib.h>
#include <time.h>
#include<mmsystem.h>
#pragma comment(lib,"winmm.lib")
#define WIDTH 640
#define HEIGHT 480
#define n 10

IMAGE welcome;
IMAGE back;
IMAGE p1;
IMAGE p2;
IMAGE p3;
IMAGE p4;
IMAGE login;

typedef struct user {
    char name[20];
    int  score = 0;
    char time[10];
    char num[10];
}USER;

typedef struct node {
    int data1, data2, ans;
    int locx;
    int locy;
    char sign;
    struct node* next;
}NODE;

void background();
void displayButton();
int checkButton(int mouseX, int mouseY);
int drawtime(int sec, int time, int start, int t1, int t2);
void funtime(int c, int flag, char* str);
```

```

void initname(USER& user);
int inittime(USER& user);
void initnum(USER& user);
void output(NODE*& hT, NODE*& p, NODE*& hF);
void stoptime(int &t);
void skip(NODE*& p);
void drawscore(int score);
int checkmouse();
void showrule();
void save(USER& user, NODE* hT, NODE* orip, NODE* hF, int t1);

```

```

NODE* create(USER& user)
{
    int x, y, flag;
    int i = 30;
    int j = 10;
    int t = 0;
    int num = atoi(user.num);
    NODE* p, * h;
    h = (NODE*)malloc(sizeof(NODE));
    h->data1 = 0;
    h->data2 = 0;
    h->ans = 0;
    h->next = NULL;
    srand((unsigned)time(NULL));
    for (int k = 0; k < num; k++) {
        x = rand() % 9+1;
        y = rand() % 9+1;
        flag = rand() % 2;
        p = (NODE*)malloc(sizeof(NODE));
        p->data1 = x;
        p->data2 = y;
        p->locx = i;
        p->locy = num*(-60)+j;
        i += 30;
        j += 60;

        if (flag == 1) {
            p->sign = '+';
            p->ans = p->data1 + p->data2;
        }
        else {
            p->sign = '-';
            p->ans = p->data1 - p->data2;
        }
    }
}

```

```

    }
    if (p->sign == '-') {
        if (p->data1 < p->data2) {
            t = p->data1;
            p->data1 = p->data2;
            p->data2 = t;
            p->ans *= -1;
        }
    }
    p->next = h->next;
    h->next = p;
}
return h;
}

```

```

void judgeT(NODE* p, NODE* hT)
{
    NODE* f;
    NODE* x;
    x = p;
    f = x;
    x = x->next;
    while (x != NULL) {
        f->next = x->next;
        x->next = hT->next;
        hT->next = x;
        x = f->next;
        break;
    }
}

```

```

void judgeF(NODE* p, NODE* hF)
{
    NODE* f;
    NODE* x;
    x = p;
    f = x;
    x = x->next;
    while (x != NULL) {
        f->next = x->next;
        x->next = hF->next;
        hF->next = x;
        x = f->next;
    }
}

```

```

        break;
    }
}

```

```

int main()
{

```

```

    USER user;
    int Time;
    NODE node;
    NODE* b;
    NODE* hT;
    NODE* hF;
    int mouseX;
    int mouseY;
    int flag = 0;
    int i = rand() % 400;
    int iori = i;
    int j = 10;
    int i1 = rand() % 400;
    int j1 = 10;
    int num = 10;
    int k = 1;
    int x = 0;
    int time1;
    int time2=0;
    time1 = atoi(user.time);
    int sign1 = 1;
    int sign2 = 1;
    int sign3 = 0;
    loadimage(&welcome, "image\\welcome.jpg", WIDTH, HEIGHT);
    loadimage(&back, "image\\background.jpg", WIDTH, HEIGHT);
    loadimage(&p1, "image\\fish02.jpg");
    loadimage(&p2, "image\\fish01.jpg");
    loadimage(&p3, "image\\fish04.jpg");
    loadimage(&p4, "image\\fish03.jpg");
    loadimage(&login, "image\\login.jpg", WIDTH, HEIGHT);

    MOUSEMSG mmsg;
    NODE* p;
    hT = (NODE*)malloc(sizeof(NODE));
    hT->next = NULL;
    hF = (NODE*)malloc(sizeof(NODE));

```

```

hF->next = NULL;
initgraph(WIDTH, HEIGHT, SHOWCONSOLE);
mciSendString("open D:\\Desktop\\4.22\\Debug\\music.mp3 alias bkmusic", 0, 0, 0,
0);

mciSendString("play bkmusic", 0, 0, 0);
BeginBatchDraw();
putimage(0, 0, &welcome);
while (1) {
    FlushBatchDraw(0, 0, 640, 480);
    Sleep(10);
    setfillcolor(BLUE);
    settextstyle(20, 0, "圆体");
    setlinecolor(RED);
    settextrcolor(RED);
    setbkmode(TRANSPARENT);
    fillrectangle(200, 50, 440, 100);
    outtextxy(270, 60, "导弹算术题");
    fillrectangle(200, 170, 440, 230);
    outtextxy(295, 185, "START");
    fillrectangle(200, 320, 440, 380);
    outtextxy(270, 330, "INTRODUCTION");
    x = checkmouse();
    if (x == 5) {
        break;
    }
    else if (x == 7) {
        showrule();
        break;
    }
    EndBatchDraw();
}
cleardevice();

putimage(0, 0, &login);
setbkcolor(WHITE);
setlinestyle(PS_SOLID, 2);
srand((unsigned)time(NULL));
initname(user);
initnum(user);
Time = inittime(user);
background();
displayButton();
outtextxy(590, 260, user.name);
outtextxy(565, 300, user.time);

```

```

while (1) {
    while (flag != -1) {
        clock_t start;
        start = clock();
        flag = checkmouse();

        if (flag == 1) {
            flag = 0;
            p = create(user);
            b = p->next;

            while (x != -1) {

                x = checkmouse();
                BeginBatchDraw();
                putimage(-150, 0, &back);
                while (b != NULL) {
                    putimage(b->locx, b->locy, &p1, SRCAND);
                    putimage(b->locx, b->locy, &p2, SRCINVERT);
                    char r[20], s[20], t[20];
                    sprintf_s(r, "%d", b->data1);
                    outtextxy(b->locx + 20, b->locy + 50, r);
                    sprintf_s(s, "%c", b->sign);
                    outtextxy(b->locx + 40, b->locy + 50, s);
                    sprintf_s(t, "%d", b->data2);
                    outtextxy(b->locx + 60, b->locy + 50, t);
                    b = b->next;
                }

                b = p->next;
                while(b != NULL) {
                    b->locx += 1.1*sign1;
                    b->locy += 1;
                    if ( b->locx > 405 ) {
                        sign1 *=-1;
                    }
                    b = b->next;
                }
                b = p->next;

                clock_t finish;
                finish = clock();
                Sleep(12);
                if (_kbhit() != 0)

```

```

{
    int l = 0;
    char g[20], m[20];
    while (1) {
        m[l] = _getch();
        outtextxy(550 + 10 * l, 375, m[l]);
        FlushBatchDraw();
        if (m[l] == 13) {
            break;
        }
        l++;
    }

    int j = atoi(m);
    if (j == b->ans) {
        user.score += 1;
        clearrectangle(541, 427, 573, 460);
        setfillcolor(YELLOW);
        fillrectangle(542, 428, 572, 458);
        drawscore(user.score);
        judgeT(p, hT);
    }
    else if (j != b->ans || b->locy > 480) {
        user.score -= 1;
        clearrectangle(541, 427, 573, 460);
        setfillcolor(YELLOW);
        fillrectangle(542, 428, 572, 458);
        drawscore(user.score);
        judgeF(p, hF);
    }
    clearrectangle(541, 364, 573, 396);
    setfillcolor(YELLOW);
    fillrectangle(542, 365, 572, 395);
}

if (x == 2) {
    FlushBatchDraw(0, 0, 640, 480);
    stoptime(time2);

    x = 0;
}

if (x == 3) {
    b = b->next;
    skip(p);
}

```



```

        x = 0;
    }
    if (x == 4) {
        output(hT, p, hF);
        FlushBatchDraw(0, 0, 640, 480);
        stoptime(time2);
        x = 0;
    }
    if (x == -1) {
        save(user, hT, p, hF, time2);
        break;
    }

    clearrectangle(564, 300, 585, 320);
    time1=drawtime(finish,Time,start,time2,time1);
    FlushBatchDraw();
    EndBatchDraw();
}
}
outtextxy(515, 260, "用户名: ");
outtextxy(515, 300, "时间: ");
outtextxy(515, 340, "我的答案: ");
outtextxy(515, 405, "当前得分: ");
}
break;
}
closegraph();
return 0;
}

```

```

void skip(NODE*& p)
{
    NODE* t, * b;
    b = p->next;
    t = p;
    while (t->next != NULL) {
        t = t->next;
    }
    b->locy = t->locy - 100;
    p->next = b->next;
    b->next = NULL;
    t->next = b;
}

```

```

void showrule()
{
    int x = 0;
    setfillcolor(BLUE);
    fillrectangle(100, 40, 500, 440);
    settextstyle(20, 0, "圆体");
    settextcolor(YELLOW);
    outtextxy(125, 70, "游戏规则介绍: ");
    outtextxy(125, 100, "请在游戏开始之前输入用户名、题目数量");
    outtextxy(125, 130, "和总时间。答题前按 Shift 表示开始输入");
    outtextxy(125, 160, "答案, 按回车键表示结束作答。答题开始");
    outtextxy(125, 190, "系统会自动计时, 中途如果选择暂停或者");
    outtextxy(125, 220, "查看当前做题情况则计时停止。答题结束");
    outtextxy(125, 250, "会自动保存目前做题情况。");

    setfillcolor(YELLOW);
    fillrectangle(520, 400, 620, 430);
    settextcolor(BLUE);
    outtextxy(530, 405, "continue");
    while (x!=8) {
        x = checkmouse();
    }
}

```

```

void output(NODE* &hT, NODE* &p, NODE* &hF)
{
    NODE* f;
    int i = 80;
    int j = 0;
    char s[20];
    setfillcolor(BLUE);
    fillrectangle(50, 50, 400, 400);
    outtextxy(100, 70, "作对的题: ");
    outtextxy(100, 200, "作错的题: ");
    outtextxy(100, 330, "剩余题目: ");
    f = hT->next;
    while (f != NULL) {
        sprintf_s(s, "%d", f->data1);
        outtextxy(i+20, j+90, s);
        sprintf_s(s, "%c", f->sign);
        outtextxy(i+30, j+90, s);
        sprintf_s(s, "%d", f->data2);
        outtextxy(i+40, j+90, s);
    }
}

```

```

        i += 40;
        f = f->next;
        if (i > 400) {
            j += 20;
            i = 0;
        }
    }
    f = hF->next;
    i = 80;
    j = 0;
    while (f != NULL) {
        sprintf_s(s, "%d", f->data1);
        outtextxy(i + 20, j + 220, s);
        sprintf_s(s, "%c", f->sign);
        outtextxy(i + 30, j + 220, s);
        sprintf_s(s, "%d", f->data2);
        outtextxy(i + 40, j + 220, s);
        i += 40;
        f = f->next;
        if (i > 400) {
            j += 20;
            i = 0;
        }
    }

    f = p->next;
    i = 80;
    j = 0;
    while (f != NULL) {
        sprintf_s(s, "%d", f->data1);
        outtextxy(i + 20, j + 360, s);
        sprintf_s(s, "%c", f->sign);
        outtextxy(i + 30, j + 360, s);
        sprintf_s(s, "%d", f->data2);
        outtextxy(i + 40, j + 360, s);
        i += 40;
        f = f->next;
        if (i > 400) {
            j += 20;
            i = 0;
        }
    }
}

```

```

void stoptime(int &t)
{
    int x = 0;
    clock_t t1, t2;
    int t3;
    t1 = clock();
    while (1) {
        Sleep(20);
        t2 = clock();
        t3 = (t2 - t1) / 1000;
        x = checkmouse();
        if ( x==1 || x == 4) {
            break;
        }
    }
    t+= t3;
}

```

```

void save(USER& user, NODE *hT, NODE *orip, NODE *hF, int t1) {
    FILE* fp;
    NODE* p;
    errno_t err = fopen_s(&fp, "D:\\Desktop\\4.22\\game.txt", "w");
    if (fp != NULL) {
        fprintf(fp, "name:%s score:%d time:%d\\n", user.name, user.score, t1);
        fprintf(fp, "正确的:");
    }
    p = hT->next;
    while (p != NULL) {
        if (p->next != NULL) {
            fprintf(fp, " %d%c%d= ", p->data1, p->sign, p->data2);
            p = p->next;
        }
        else {
            fprintf(fp, " %d%c%d= ", p->data1, p->sign, p->data2);
            p = p->next;
        }
    }

    fprintf(fp, "\\n");
    fprintf(fp, "还没做的:");
    p = orip->next;
    while (p != NULL) {
        if (p->next != NULL) {

```

```

        fprintf(fp, " %d%c%d= ", p->data1, p->sign, p->data2);
        p = p->next;
    }
    else {
        fprintf(fp, " %d%c%d= ", p->data1, p->sign, p->data2);
        p = p->next;
    }
}

fprintf(fp, "\n");
fprintf(fp, "做错的:");
p = hF->next;
while (p != NULL) {
    if (p->next != NULL) {
        fprintf(fp, " %d%c%d= ", p->data1, p->sign, p->data2);
        p = p->next;
    }
    else {
        fprintf(fp, " %d%c%d= ", p->data1, p->sign, p->data2);
        p = p->next;
    }
}
fprintf(fp, "\n");
fclose(fp);
}

int checkmouse() {
    int x = 0;
    MOUSEMSG mmsg;
    int mouseX;
    int mouseY;
    if (MouseHit()) {
        mmsg = GetMouseMsg();
        switch (mmsg.uMsg) {
            case WM_LBUTTONDOWN:
                mouseX = mmsg.x;
                mouseY = mmsg.y;
                x = checkButton(mouseX, mouseY);
            }
        }
    return x;
}

```

```

void displayButton()
{
    setfillcolor(BLUE);
    settextrstyle(20, 0, "圆体");
    setlinecolor(RED);
    settextrcolor(RED);
    setbkmode(TRANSPARENT);
    fillrectangle(520, 10, 600, 50);
    outtextxy(530, 20, "START");
    setfillcolor(BLUE);
    fillrectangle(520, 55, 600, 95);
    outtextxy(530, 65, "PAUSE");
    fillrectangle(520, 100, 600, 140);
    outtextxy(530, 110, "SKIP");
    fillrectangle(520, 145, 600, 185);
    outtextxy(530, 155, "REVIEW");
    fillrectangle(520, 190, 600, 225);
    outtextxy(530, 195, "LOGOUT");
    setfillcolor(YELLOW);
    fillrectangle(542, 365, 572, 395);
    fillrectangle(542, 428, 572, 458);
    setcolor(RED);
    setlinestyle(PS_SOLID, 2);
    line(490, 0, 490, 480);
    line(490, 250, 640, 250);
}

int inittime(USER& user)
{
    int Time = 0;
    InputBox(user.time, 50, "请输入时间: ");
    Time = atoi(user.time);
    return Time;
}

void initname(USER& user)
{
    InputBox(user.name, 50, "请输入用户名: ");
}

void initnum(USER& user)
{
    InputBox(user.num, 50, "请输入题目数量: ");
}

```

```

void background()
{
    loadimage(&back, "image\\background.jpg", WIDTH, HEIGHT);
    cleardevice();
    setcolor(BLACK);
    setlinestyle(PS_SOLID, 2);
    putimage(-150, 0, &back);
    setlinestyle(PS_SOLID, 2);
    rectangle(1, 1, WIDTH - 2, HEIGHT - 2);
    line(490, 0, 490, 480);
    line(490, 250, 640, 250);
}

void drawscore(int score)
{
    char s[100];
    sprintf_s(s, "%d", score);
    outtextxy(550, 435, s);
}

int drawtime(int sec, int time, int start, int t1, int t2)
{
    char buf[10];
    t1 = time + (start - sec)/1000 + t2/1000;
    sprintf_s(buf, "%d", t1);
    outtextxy(565, 300, buf);
    return t1;
}

int checkButton(int mouseX, int mouseY)
{
    if (mouseX >= 520 && mouseX <= 600 && mouseY >= 10 && mouseY <= 50) {
        clearrectangle(519, 9, 601, 51);
        outtextxy(530, 20, "START");
        Sleep(20);
        setfillcolor(BLUE);
        fillrectangle(520, 10, 600, 50);
        outtextxy(530, 20, "START");
        return 1;
    }
}

```

```

    }

    else if (mouseX >= 520 && mouseX <= 600 && mouseY >= 55 && mouseY <= 95)
{
    clearrectangle(520, 55, 599, 94);
    outtextxy(530, 65, "PAUSE");
    Sleep(20);
    setfillcolor(BLUE);
    fillrectangle(520, 55, 600, 95);
    outtextxy(530, 65, "PAUSE");
    return 2;
}

    else if (mouseX >= 520 && mouseX <= 600 && mouseY >= 100 && mouseY <=
140) {
    clearrectangle(519, 99, 601, 141);
    outtextxy(530, 110, "SKIP");
    Sleep(20);
    setfillcolor(BLUE);
    fillrectangle(520, 100, 600, 140);
    outtextxy(530, 110, "SKIP");
    return 3;
}

    else if (mouseX >= 520 && mouseX <= 600 && mouseY >= 145 && mouseY <=
185) {
    clearrectangle(520, 145, 599, 184);
    outtextxy(530, 155, "REVIEW");
    Sleep(20);
    setfillcolor(BLUE);
    fillrectangle(520, 145, 600, 185);
    outtextxy(530, 155, "REVIEW");
    return 4;
}

    else if (mouseX >= 200 && mouseX <= 440 && mouseY >= 170 && mouseY <=
230) {
    return 5;
}

    else if (mouseX >= 200 && mouseX <= 440 && mouseY >= 320 && mouseY <=
380) {
    return 7;
}

    else if (mouseX >= 520 && mouseX <= 620 && mouseY >= 400 && mouseY <=480)
{

```



```

        return 8;
    }

    else if (mouseX >= 520 && mouseX <= 600 && mouseY >= 190 && mouseY <=
225) {
        clearrectangle(520, 190, 600, 225);
        outtextxy(530, 195, "LOGOUT");
        Sleep(20);
        setfillcolor(BLUE);
        fillrectangle(520, 190, 600, 225);
        outtextxy(530, 195, "LOGOUT");
        return -1;
    }
}

```