

# 高级语言程序设计

## 课设报告

学 号： 20074221

姓 名： 游佳慧

指导教师： 周琨

提交日期： 2021. 5. 22

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.....	1
1.1 功能.....	1
1.1.1 基本功能.....	1
1.1.2 扩展功能.....	2
1.2 数据.....	3
1.3 界面.....	3
1.4 开发与运行环境.....	6
2 概要设计.....	7
2.1 主要数据结构.....	7
2.2 程序总体结构.....	8
2.2.1 初始化游戏菜单页.....	9
2.2.2 初始化地图.....	9
2.2.3 进行游戏.....	10
2.2.4 文件管理.....	11
2.2.5 结束游戏.....	11
3 详细设计.....	12
3.1 主程序的流程图.....	12
3.2 模块的详细设计.....	14
3.2.1 初始化游戏菜单页.....	14
3.2.2 初始化地图.....	15
3.2.3 进行游戏.....	18
3.2.4 文件管理.....	20
3.2.5 结束游戏.....	20
4 测试.....	21
4.1 游戏时间测试.....	21
4.2 用户信息测试.....	22
4.3 弹出菜单栏测试.....	23
4.4 游戏中途退出存档测试.....	24
4.5 游戏结束退出并存储用户信息测试.....	25
4.6 游戏读档测试.....	26
5 开发中遇到的问题及解决办法.....	27
5.1 弹出菜单栏后回到游戏页面.....	27
5.2 批量绘图实现人物的移动.....	27
5.3 计时的实现.....	27
5.4 用二进制方法存档和读档.....	28
6 总结.....	28
6.1 课程设计总结.....	28
6.2 对本课程意见与建议.....	28
附件：程序源代码.....	28

# 1 需求分析

用 C 语言编写一个“迷宫大逃离”的游戏，游戏共有三个地图，玩家可以选择开始新游戏或继续上次游戏，在逃离迷宫的过程中，通过上下左右键控制人的走动，采到花朵加分，采到炸弹减分，加或减的分数随机，并能动态显示玩家得分，当走到出口时，玩家胜利，可重新选择关卡开始游戏或退出游戏并储存用户名和得分。

## 1.1 功能

进入程序时开启背景音乐，在欢迎界面时可选择开始新游戏或读取上次游戏，若选择读取游戏，则将文件中存储的游戏状态恢复到屏幕上，可以继续上次的游戏；若选择开始新游戏，则输入用户名和宝箱的个数，（若没有输入个数，默认是 5 个），并进入游戏关卡选择界面，共有三个关卡，通过鼠标选择后进入游戏界面，输入的用户名显示在信息区。

在游戏界面时，游戏区中有迷宫，走迷宫的人和一些宝箱，宝箱的位置随机，宝箱下面可能是小花，也可能是炸弹。按上下左右键控制人的走动和改变方向，走到宝箱的位置时按下空格键则显示出宝箱下面是什么，是小花加分，是炸弹减分，加或减的分数随机，然后小花或炸弹消失；若走到宝箱的位置，没按空格键，则宝箱不变化；动态显示玩家得分。

在控制区有功能按钮，通过鼠标点击实现。点击“显示”按钮，则显示出宝箱的下面是小花还是炸弹，点击“隐藏”按钮，则显示宝箱；点击“增加”按钮，则增加一个宝箱（小花或炸弹）；点击“清空”按钮，则把所有的小花、炸弹清除，宝箱也不再显示；点击“设置”按钮，则弹出游戏设置菜单栏，再次点击“设置”按钮可关闭菜单栏回到游戏。在菜单栏中，点击“帮助”按钮则显示游戏规则说明，点击“关卡”按钮则返回游戏关卡选择界面，重新选择关卡开始游戏，点击“音乐”按钮则关闭或开启背景音乐，点击“退出”按钮，则将当前游戏状态（人、小花、炸弹、用户名、分数等所有信息）存入文件，退出程序。

当走到出口时，弹出对话框，若点击“继续游戏”按钮，则回到游戏关卡选择界面；若点击“退出游戏”按钮，则把当前用户名和得分追加到指定文件中（即文件中存储了所有玩过此游戏的用户名和分数），退出程序。

### 1.1.1 基本功能

#### 1、输入用户名和宝箱个数

程序运行时先输入用户名，再输入宝箱的个数（若没有输入，默认是 5 个），然后进入游戏关卡选择界面，选择后进入游戏界面，宝箱的位置随机，并把用户名显示在信息显示区中。

#### 2、控制区按钮

单击“显示/隐藏”按钮，按钮在“显示”和“隐藏”之间切换，单击“显示”，显示宝箱下面是什么，单击“隐藏”，隐藏小花炸弹，显示宝箱；

单击“增加”按钮，增加一个宝箱；

单击“清空”按钮，把所有的小花、炸弹，宝箱都清除，链表清空，释放空间；

单击“设置”按钮，显示游戏设置菜单栏，再次单击“设置”按钮关闭菜单栏。

在游戏设置菜单栏中，单击“退出”按钮，将当前游戏状态（人、小花、炸弹、用户名、分数等所有信息）存入文件，退出程序。

当走到出口时，弹出对话框，单击“退出游戏”按钮，将当前用户名和得分追加到指定文件中（即文件中存储了所有玩过此游戏的用户名和分数），退出程序。

### **3、游戏区**

显示迷宫，走迷宫的人和一些宝箱，走到宝箱的位置时按下空格键则显示出宝箱下面是什么（小花或炸弹），然后道具消失，若没按空格键，则宝箱不变化。

### **4、人的移动**

通过上下左右键控制人的走动。

### **5、小花、炸弹的位置**

位置随机产生，显示的是宝箱。

### **6、计分**

碰到小花得分，碰到炸弹减分，加或减的分数随机。

### **7、显示得分**

游戏开始后，在信息显示区显示玩家的姓名和得分，得分动态变化。

## **1.1.2 扩展功能**

### **1、实现多关卡选择**

在游戏关卡选择界面时，共有三个关卡可选择，每个关卡的地图和风格均不同。在游戏中可随时回到选择界面重新选择关卡。

### **2、增加背景音乐**

进入程序时开启背景音乐，可在游戏界面中点击“设置”按钮后，在弹出的游戏设置菜单栏中点击“音乐”按钮，关闭或开启音乐。

### **3、美化按钮**

在欢迎界面，鼠标单击按钮时，按钮有被按下去的效果，松开时，有弹回来的效果。

在选择游戏关卡界面，当鼠标移到按钮上时，按钮变色，移开时，按钮恢复原色。

### **4、增加界面切换动画**

从欢迎界面切换到游戏关卡选择界面时，增加画面由暗到亮的动画效果。

### **5、将游戏的当前状态存盘和导入**

存盘：将当前游戏的状态（人、小花、炸弹、用户名、分数等所有信息）存入文件。当点击游戏设置菜单栏中的“退出”按钮后，完成存档并退出游戏。

导入：将文件中存储的游戏状态恢复到屏幕上，可以继续上次的游戏。当点击欢迎界面的“读取游戏”按钮后，导入上次游戏。

### **6、所有小花，炸弹的分数随机**

玩家加或减的分数随机。

### **7、通过键盘控制人物移动**

通过上下左右键控制人的走动，可改变方向和速度。

### **8、增加计时功能**

从进入游戏时开始计时，并将已进行游戏时长实时显示在信息区。中途切换关卡则重新开始计时。

### **9、增加游戏设置菜单栏**

在游戏界面，点击“设置”按钮，则弹出游戏设置菜单栏，再次点击“设置”按钮可关闭菜单栏回到游戏。在游戏设置菜单栏中，单击“帮助”按钮，显示游戏规则说明；单击“关卡”按钮，返回游戏关卡选择界面；单击“音乐”按钮，关闭或开启背景音乐；单击“退出”按钮，将当前游戏状态（人、小花、炸弹、用户名、分数等所有信息）存入文件，退出程序。

#### 10、走到出口时，可选择继续游戏或退出游戏

当走到出口时，弹出对话框，若点击“继续游戏”按钮，则回到游戏关卡选择界面；若点击“退出游戏”按钮，则储存当前用户名和得分，退出程序。

## 1.2 数据

宝箱（包括坐标，是小花还是炸弹）用链表存储，消失时从链表中删除，清空时将链表清空，释放空间。

人的坐标，用户信息（包括用户名、宝箱数量，得分），用结构体存放。

三个迷宫用三维数组存储。

## 1.3 界面

设计 5 个界面，一个是欢迎界面，一个是游戏关卡选择界面，和三个不同关卡的游戏界面。

### 1.3.1 欢迎界面

欢迎界面如图 1.1 所示。



图 1.1 欢迎界面

### 1.3.2 游戏关卡选择界面

游戏关卡选择界面如图 1.2 所示



图 1.2 游戏界面

### 1.3.3 游戏界面

界面分为三个区域：游戏区、控制区、信息显示区。游戏区有迷宫，走迷宫的人和一些宝箱；控制区有“显示/隐藏，增加，清空，设置”4 个按钮；信息显示区有玩家姓名和得分。

游戏界面如图 1.3.1，图 1.3.2，图 1.3.3 所示

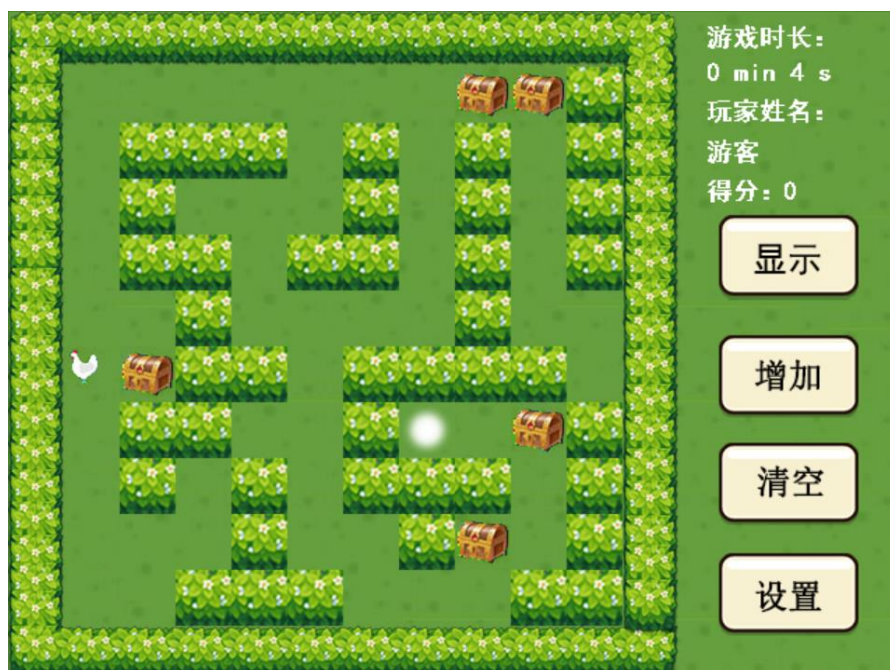


图 1.3.1 游戏界面



图 1.3.2 游戏界面



图 1.3.3 游戏界面

## 1.4 开发与运行环境

Visual studio 6.0 和 EasyX\_2013 冬至版。



## 2 概要设计

根据分析，共划分为 6 个模块，有初始化游戏菜单页，初始化地图，进行游戏，用户管理，文件管理，结束游戏。如图 2.1 所示。

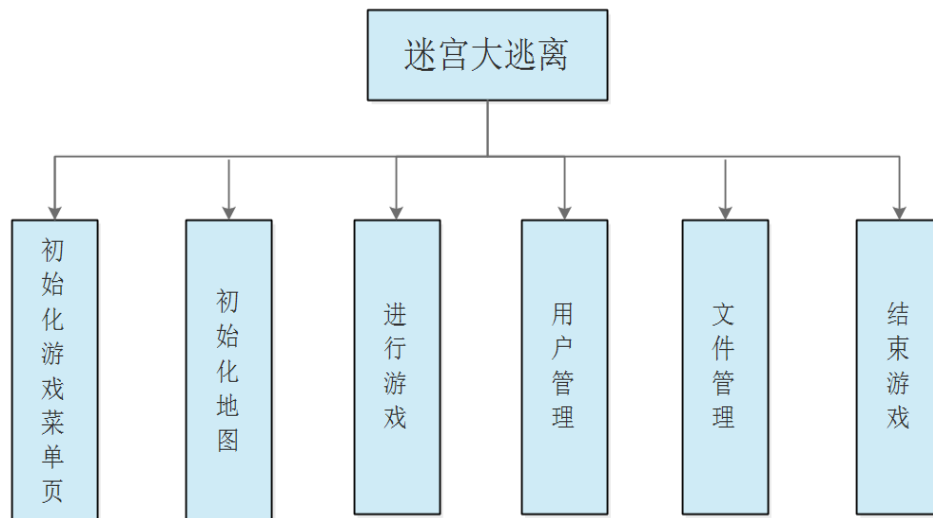


图 2.1 程序的主要模块

### 2.1 主要数据结构

1、用户结构， 用结构体储存用户名、宝箱数量，得分

```
typedef struct users {  
    char    name[20];           //用户名  
    int     num = 5;            //道具数量  
    int     score = 0;          //用户得分  
}USER;
```

2、人物结构， 用结构体储存人的坐标

```
typedef struct {  
    int x;  
    int y;  
}People;
```

3、宝箱信息，用链表存储，结点的数据包括位置、是小花还是炸弹

```
typedef struct Tool {  
    int     x, y;                //坐标  
    int     flag;                //0是小花，1是炸弹  
    struct  Tool* next;  
}tool;
```

- 4、用一个 int 类型三维数组 `a[3][10][10]` 来保存迷宫的信息。
- 5、游戏胜利并退出游戏时存储用户信息（包括用户名和得分），追加到指定文件中 `fprintf(fp, "%s %d\n", user.name, user.score)`。
- 6、游戏中途退出时将当前游戏的状态（人的坐标、小花、炸弹、用户名、分数等所有信息）存入文件，将储存用户信息和人物坐标的结构体变量用二进制存储。

## 2.2 程序总体结构

程序的总体结构如图 2.2 所示：

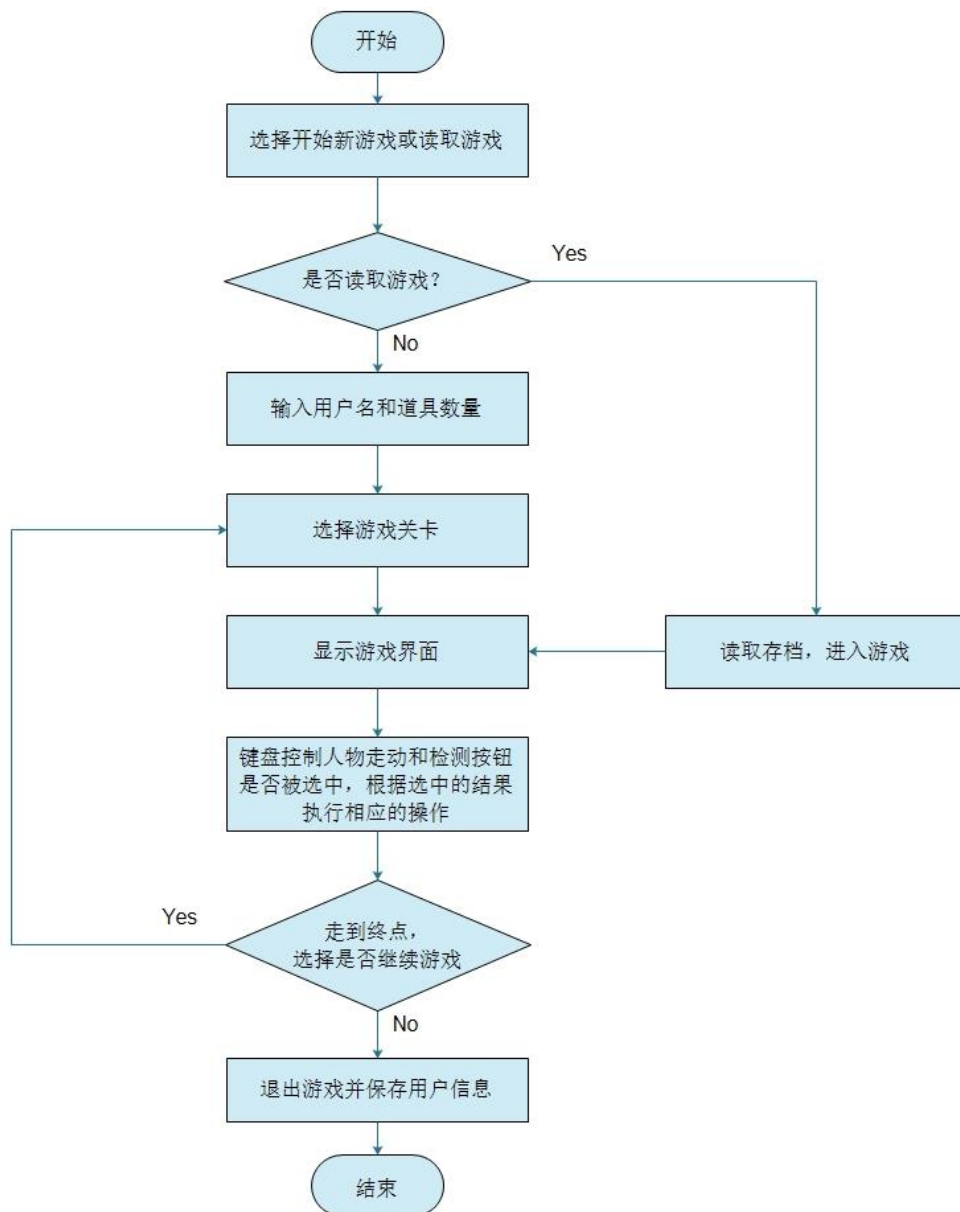


图 2.2 程序的总体结构

### 2.2.1 初始化游戏菜单页

本模块用于显示开始界面，选择开始新游戏或读取游戏，输入用户名和道具数量，显示游戏关卡选择界面并选择关卡。

相关函数：

◆ **void LoadImg();**

功能：加载所有图片全局变量

参数：无

返回值：无

◆ **void Music();**

功能：播放背景音乐

参数：无

返回值：无

◆ **void Startmenu(USER& user);**

功能：显示开始界面，选择开始新游戏或读取游戏，输入用户名和道具数量

参数：存放用户信息的结构体变量

返回值：无

说明：若选择开始新游戏，则调用函数 `Mainmenu(USER& user)`，将输入的用户名存到 `user.name` 中，输入的道具数量存到 `user.num` 中；若选择读取游戏，则调用函数 `Read(USER& user)`

◆ **void Mainmenu(USER& user);**

功能：显示游戏关卡选择界面,选择游戏关卡

参数：存用户信息的结构体变量

返回值：无

说明：调用了函数 `checkButton(int mouseX, int mouseY)`和函数 `Initgame(USER& user, int gamemode)`

◆ **int checkButton(int mouseX, int mouseY);**

功能：在游戏关卡选择界面，判断鼠标是否选中按钮

参数：鼠标位置的 `x` 坐标，鼠标位置的 `y` 坐标

返回值：若选中三个按钮中任意一个，分别返回 1，2，3；若未选中，返回-1

### 2.2.2 初始化地图

本模块用于初始化游戏地图，根据所选关卡绘制游戏界面。

相关函数：

◆ **void Initgame(USER& user, int gamemode);**

功能：初始化游戏地图，创建 `int` 类型的三维数组存放三个地图，创建链表存放道具

参数：存用户信息的结构体变量，选择的关卡

返回值：无

说明：根据 `gamemode` 的值选择调用函数 `Mode1(USER& user, int a[10][10], tool*& head, People people)`，`Mode2(USER& user, int a[10][10], tool*& head, People people)` 或 `Mode3(USER& user, int a[10][10], tool*& head, People people)`

- ◆ **void Mode1(USER& user, int a[10][10], tool\*& head, People people)**  
 功能: 绘制关卡“花园寻宝”的游戏地图  
 参数: 存放用户信息的结构体变量, 存放地图的二维数组, 存放道具的链表的头结点, 存放人物坐标的结构体变量  
 返回值: 无  
 说明: 调用函数 InGame(USER& user, int a[10][10], tool\*& head, int gamemode, People people)
- ◆ **void Mode2(USER& user, int a[10][10], tool\*& head, People people)**  
 功能: 绘制关卡“沙漠探险”的游戏地图  
 参数: 存放用户信息的结构体变量, 存放地图的二维数组, 存放道具的链表的头结点, 存放人物坐标的结构体变量  
 返回值: 无  
 说明: 调用函数 InGame(USER& user, int a[10][10], tool\*& head, int gamemode, People people)
- ◆ **void Mode3(USER& user, int a[10][10], tool\*& head, People people)**  
 功能: 绘制关卡“勇闯矿山”的游戏地图  
 参数: 存用户信息的结构体变量, 存放地图的二维数组, 存放道具的链表的头结点, 存放人物坐标的结构体变量  
 返回值: 无  
 说明: 调用函数 InGame(USER& user, int a[10][10], tool\*& head, int gamemode, People people)

### 2.2.3 进行游戏

本模块用于和进行游戏有关的函数, 包括与人物、道具、功能按钮、游戏时长相关的操作。

相关函数:

- ◆ **void InGame(USER& user, int a[10][10], tool\*& head, int gamemode, People people);**  
 功能: 实现与进行游戏相关的操作  
 参数: 存放用户信息的结构体变量, 存放地图的二维数组, 存放道具的链表的头结点, 选择的关卡, 存放人物坐标的结构体变量  
 返回值: 无  
 说明: 移动人物时调用了函数 Display(People people), 增加道具时调用了函数 addBox(USER& user, int a[10][10], tool\*& head, tool\*& p), 清空道具时调用了函数 delBox(USER& user, int a[10][10], tool\*& head, tool\*& p), 游戏结束时调用了函数 score(USER user) 和 Gameover(), 游戏中途退出时调用了函数 Remember(USER user, int a[10][10], tool\* head, int gamemode, People people) 和 Gameover()
- ◆ **void Display(People people);**  
 功能: 在新的位置显示人物  
 参数: 存放人物坐标的结构体变量  
 返回值: 无
- ◆ **void gametime(time\_t starttime)**  
 功能: 记录已进行游戏时长并显示在信息区  
 参数: 存放开始游戏时间的 time\_t 类型的变量

返回值：无

◆ **void addBox(USER& user, int a[10][10], tool\*& head, tool\*& p)**

功能：增加一个道具，创建一个结点并插入链表的头结点后面

参数：存放用户信息的结构体变量，存放地图的二维数组，存放道具的链表的头结点，空结点

返回值：无

◆ **void delBox(USER& user, int a[10][10], tool\*& head, tool\*& p)**

功能：删除所有道具，清空链表，释放空间

参数：存放用户信息的结构体变量，存放地图的二维数组，存放道具的链表的头结点，空结点

返回值：无

## 2.2.4 文件管理

本模块用于和文件相关的操作，包括存档，读取存档，游戏结束后记录用户名和得分。

相关函数：

◆ **void score(USER user);**

功能：游戏结束后将用户名和得分追加到指定文件。

参数：存用户信息的结构体变量

返回值：无

◆ **void Remember(USER user, int a[10][10], tool\* head, int gamemode, People people);**

功能：将当前游戏的状态（人、小花、炸弹、用户名、分数等所有信息）存入文件。

参数：存用户信息的结构体变量，存放当前关卡地图的二维数组，存放道具的链表的头结点，存放人物坐标的结构体变量

返回值：无

说明：考虑到结构体所占字符数不规则，故用二进制方式向文件写入数据

◆ **void Read(USER& user);**

功能：读取存档，将文件中存储的游戏状态恢复到屏幕上。

参数：存用户信息的结构体变量

返回值：无

说明：用二进制方式从文件读入数据

## 2.2.5 结束游戏

本模块用于退出游戏。

相关函数：

◆ **void Gameover();**

功能：退出游戏。

参数：无

返回值：无

## 3 详细设计

在开始界面时选择开始新游戏或读取存档，若选择读取存档，则从文件中读取数据并将上次游戏状态恢复到屏幕上；若选择开始新游戏，则进入游戏关卡选择界面，完成关卡选择并初始化地图和道具后，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，检测鼠标和键盘的操作，根据操作完成相应的功能。

### 3.1 主程序的流程图

主程序的流程图如图 3.1 所示。

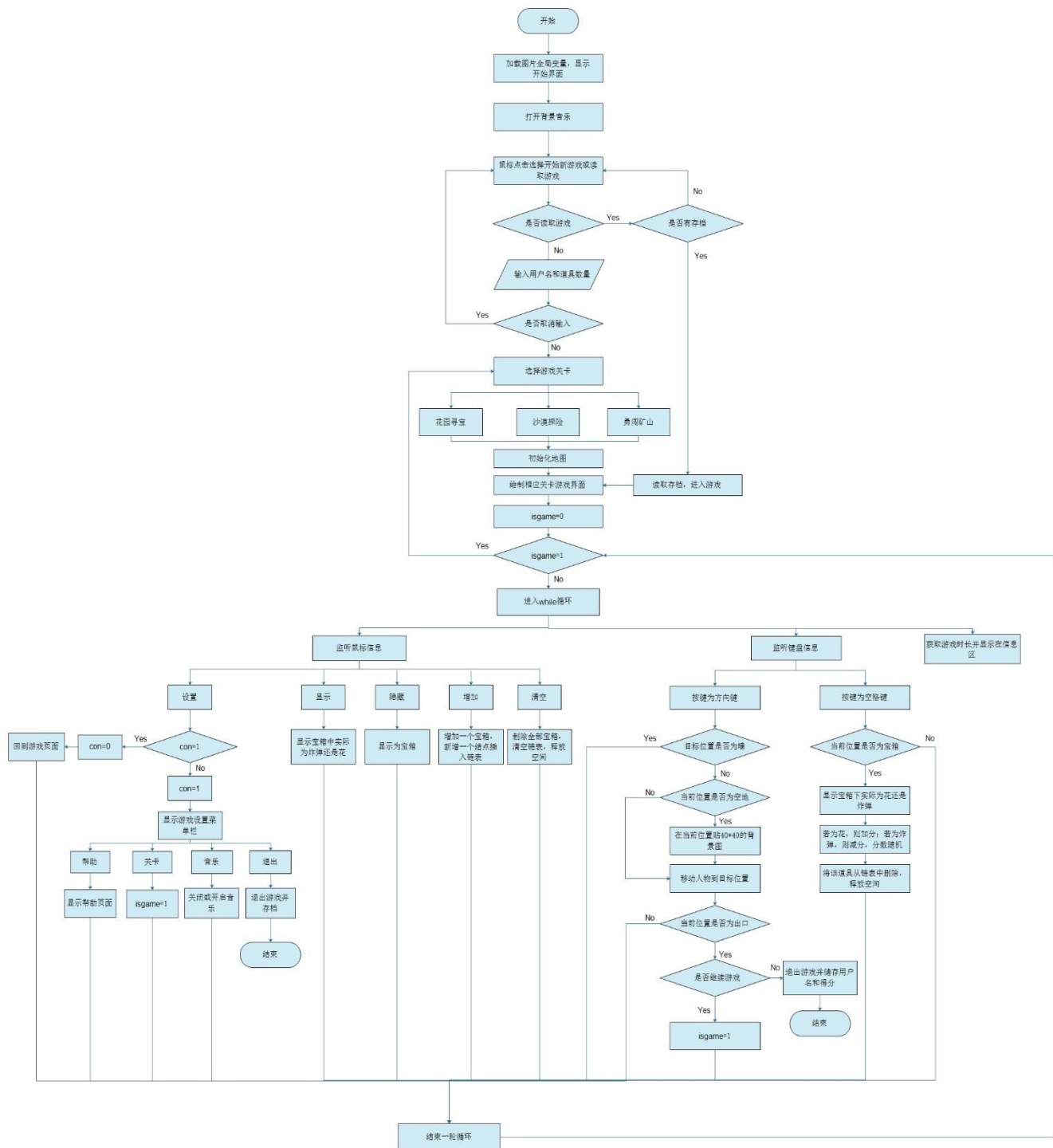


图 3.1 主程序的流程图

## 3.2 模块的详细设计

### 3.2.1 初始化游戏菜单页

本模块用于显示开始界面，选择开始新游戏或读取游戏，输入用户名和道具数量，显示游戏关卡选择界面并选择关卡。

本模块的流程图如图 3.2.1 所示。

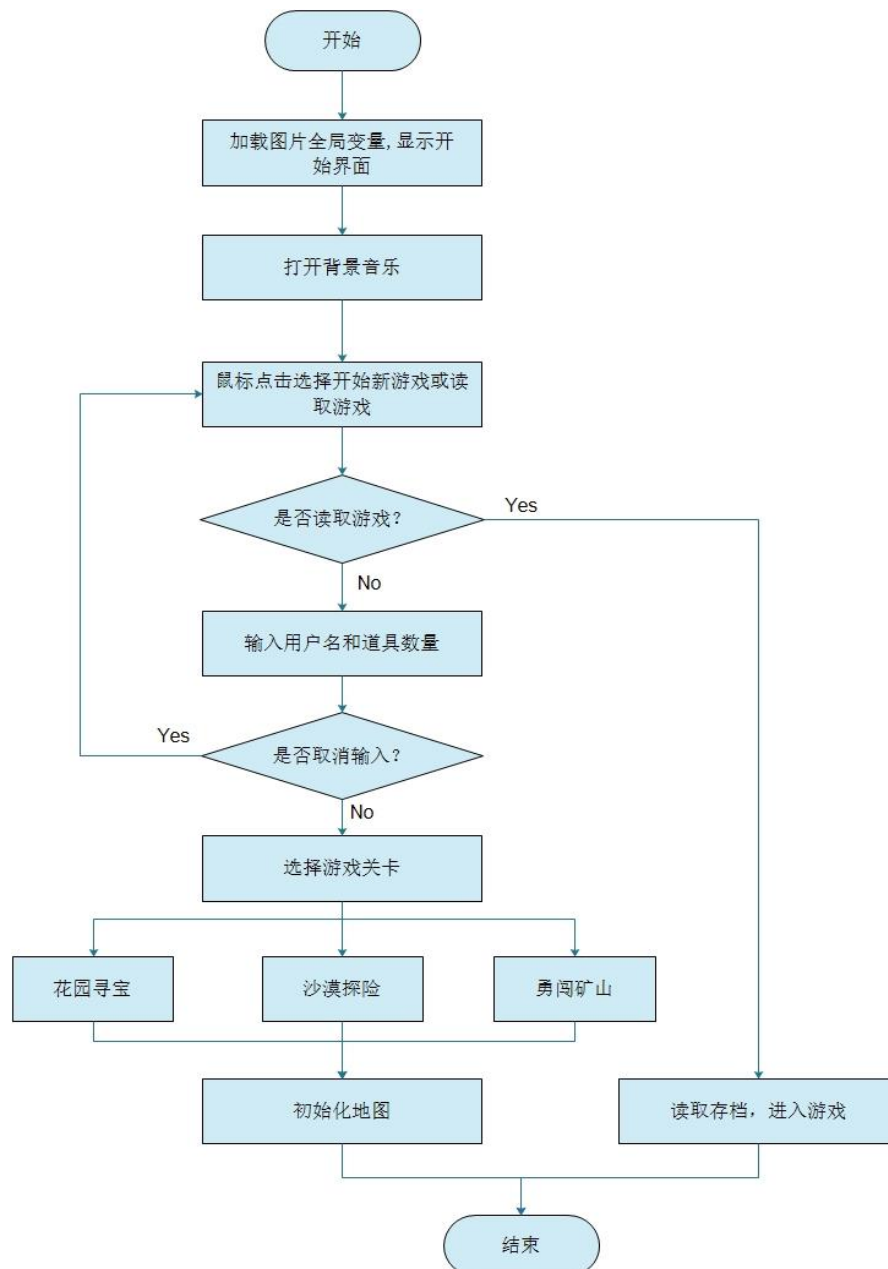


图 3.2.1 初始化游戏菜单页流程图

相关函数：

◆ void LoadImg();



功能: 函数用于加载存放所有图片全局变量

◆ **void Music();**

功能: 打开背景音乐

◆ **void Startmenu(USER& user);**

功能: 显示开始界面, 有两个按钮供用户选择, 函数传入参数为存用户信息的结构体变量的引用, 用于存放用户信息。函数中定义了局部变量 **MOUSEMSG mmsg**, 用于接受鼠标信息。用户可选择开始新游戏或读取游戏, 若选择读取游戏, 则调用函数 **Read(USER& user)**, 将文件中存储的游戏状态恢复到屏幕上; 若选择开始新游戏, 则输入用户名和宝箱的个数, (若没有输入个数, 默认是 5 个), 存入 **user**, 并进入游戏关卡选择界面。

◆ **void Mainmenu(USER& user);**

功能: 显示游戏关卡选择界面, 有三个关卡可供用户选择, 函数中设置了页面切换动画和按钮响应, 定义了局部变量 **int mouseX, mouseY**, 用于存放鼠标坐标, 定义了局部变量 **int gamemode**, 用于存放所选关卡。调用了函数 **checkButton(int mouseX, int mouseY)**, 用于检测鼠标是否位于按钮位置, 最后调用函数 **Initgame(USER& user, int gamemode)**, 传入 **user** 和 **gamemode**, 进入地图初始化。

◆ **int checkButton(int mouseX, int mouseY);**

功能: 在游戏关卡选择界面, 判断鼠标是否选中按钮, 根据所选中按钮分别返回 1, 2, 3, 若未选中按钮, 则返回-1。

### 3.2.2 初始化地图

本模块用于初始化游戏地图, 根据所选关卡绘制游戏界面。

本模块的流程图如图 3.2.2 所示。

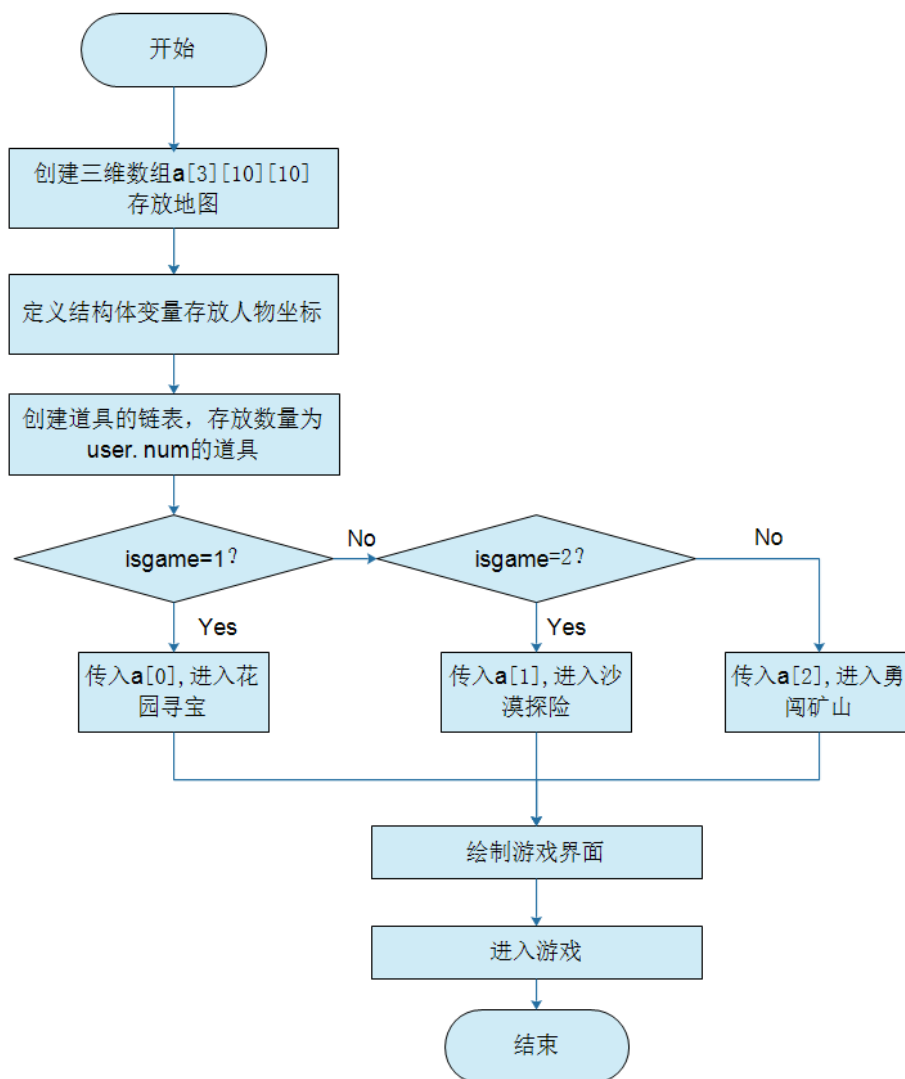


图 3.2.2 初始化地图流程图

相关函数：

◆ **void Initgame(USER& user, int gamemode);**

功能：初始化游戏地图，创建链表存放道具，根据 gamemode 的值进入对应的游戏关卡。函数中定义了局部变量 `int a[3][10][10]` 存放三个地图，其中 0 代表空地，1 代表墙，3 代表出口；定义了局部变量 `People people` 存放人物坐标；定义了局部变量 `tool* head,* p`，其中 `head` 为头结点，创建道具的链表，其中 `x,y,flag` 的值均为随机产生。根据 gamemode 的值，分别调用函数 `Mode1`，`Mode2`，`Mode3`，并分别传入 `a[0]`，`a[1]`，`a[2]`。

产生道具的流程图如图 3.2.3 所示。

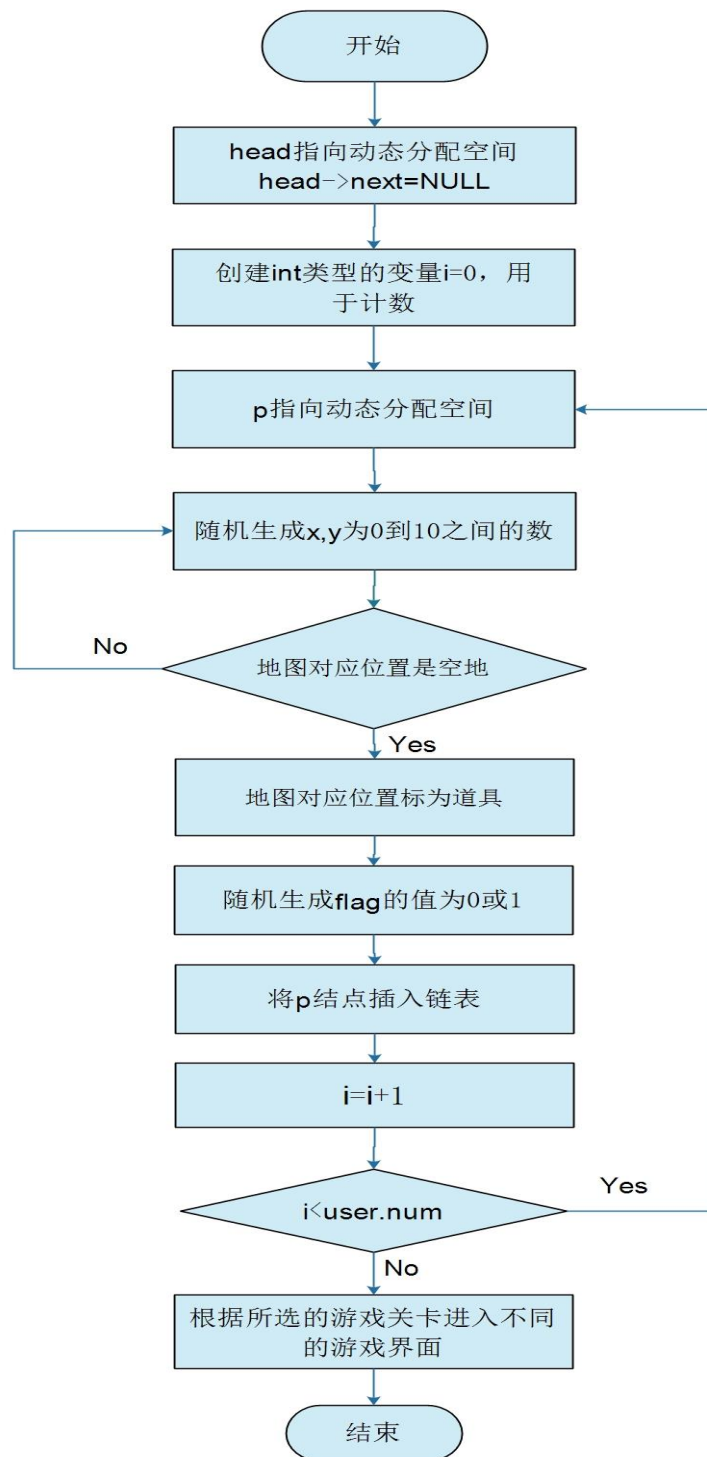


图 3.2.3 产生道具的流程图

◆ **void Mode1(USER& user, int a[10][10], tool\*& head, People people)**

功能: 绘制关卡“花园寻宝”的游戏地图, 用 BeginBatchDraw()和 EndBatchDraw()进行批量绘图, 最后调用函数 InGame(USER& user, int a[10][10], tool\*& head, int gamemode, People people)。

◆ **void Mode2(USER& user, int a[10][10], tool\*& head, People people)**

功能: 绘制关卡“沙漠探险”的游戏地图, 用 BeginBatchDraw()和 EndBatchDraw()进

行批量绘图，最后调用函数 `InGame(USER& user, int a[10][10], tool*& head, int gamemode, People people)`。

◆ **void Mode3(USER& user, int a[10][10], tool\*& head, People people)**

功能：绘制关卡“勇闯矿山”的游戏地图，用 `BeginBatchDraw()` 和 `EndBatchDraw()` 进行批量绘图，最后调用函数 `InGame(USER& user, int a[10][10], tool*& head, int gamemode, People people)`。

### 3.2.3 进行游戏

本模块用于和进行游戏有关的函数，包括与人物移动和显示、道具的增加、清除、显示、隐藏、功能按钮的响应与执行、实时显示游戏时长相关的操作。

相关函数：

◆ **void InGame(USER& user, int a[10][10], tool\*& head, int gamemode, People people);**

功能：监听鼠标信息和键盘信息，实现对按钮的响应，并执行相关功能。

定义了局部变量 `MOUSEMSG mmsg` 用于接受鼠标信息；定义了局部变量 `int flag=0`，值为 0 和 1 分别表示当前处于道具的隐藏状态和显示状态；定义了局部变量 `int con=0`，值为 0 和 1 分别表示当前处于设置菜单栏的关闭状态和显示状态；定义了局部变量 `int isBgm=0`，值为 0 和 1 分别表示当前处于音乐的播放状态和暂停状态；定义了局部变量 `int isgame=0`，值为 0、1 或 2 时分别表示游戏中、回到选择关卡、走到出口状态。

在 `while` 循环中不断监听鼠标和键盘信息，每次循环，若处于游戏中，则调用 `Display(people)` 和 `gametime(starttime)`。

处理鼠标信息：点击“显示”按钮时，遍历存放道具的链表，根据 `p->flag` 的值在相应位置贴花或炸弹的图，按钮变为“隐藏”；点击“增加”按钮时，调用函数 `addBox(USER& user, int a[10][10], tool*& head, tool*& p)`；点击“清空”按钮时，调用函数 `delBox(USER& user, int a[10][10], tool*& head, tool*& p)`；点击“设置”按钮时，在指定位置贴上设置菜单栏的图，再次点击设置返回游戏。在设置页面时，点击菜单栏中“帮助”按钮，贴上相应的图，点击“关卡”按钮，则退出循环，调用函数 `Mainmenu(USER& user)` 回到关卡选择页面，点击“音乐”按钮时，暂停音乐，点击“退出”按钮时，调用函数 `Remember(USER user, int a[10][10], tool* head, int gamemode, People people)` 和 `GameOver()`。

处理键盘信息：若按键为上下左右，则判断人物移动的下一个位置是否为墙，为墙则不移动，不为墙则人物坐标做相应加减变化；若按键为空格，则判断人物当前位置是否为宝箱，若为宝箱，则显示当前宝箱实际为花还是炸弹，并随机加减得分，存到 `user.score` 中。

◆ **void Display(People people);**

功能：在新的位置显示人物。用 `getimage` 预先保存绘图区，在新的位置绘制人物，等待，再用预先保存的图像覆盖绘制的图像。调用了 `BeginBatchDraw()` 和 `FlushBatchDraw()` 实现批量绘图。

◆ **void gametime(time\_t starttime);**

功能：记录已进行游戏时长并显示在信息区。定义了局部变量 `time_t endtime`，`thistime`，每次进入函数时调用函数 `time(NULL)`，获取当前时间，并令 `endtime` 等于返回值，`thistime=endtime-starttime`，`thistime` 的值即为游戏已进行时间秒数。定义了局部变量 `struct tm* Tm`，`Tm = localtime(&thistime)`，利用 `localtime` 函数计算秒数。将时间打

印在信息区。

◆ **void addBox(USER& user, int a[10][10], tool\*& head, tool\*& p);**

功能：增加一个道具，创建一个结点并插入链表的头结点后面。定义局部变量 `int s=0`，用来计数。先给 `p` 动态分配空间，而后进入循环，不断随机产生 `p->x` 和 `p->y` 的值，每次循环 `s++`，直到满足 `a[p->x][p->y] == 0 && s >= 1000` 时跳出循环。定义变量 `s` 用来计数是为了防止当地图上所有空地均已产生道具时，进入死循环。跳出循环后进行判断，若满足 `a[p->x][p->y] == 0`，则令数组该位置的值变为 2，并随即产生 `flag` 的值，再将结点 `p` 用头插法插入链表的头结点之后，令 `user.num` 的值加 1，在地图相应位置贴上宝箱的图。

◆ **void delBox(USER& user, int a[10][10], tool\*& head, tool\*& p);**

功能：删除所有道具，清空链表，释放空间。先遍历存放地图的数组 `a[10][10]`，若存放值为 2，则在对应地方贴上大小为 `40*40` 的背景图，覆盖住宝箱。然后判断头结点 `head` 是否为空，若不为空，再进行下列操作：令 `head=head->next`，而后遍历链表，将所有结点删除，并释放空间，再遍历存放地图的数组 `a[10][10]`，将所有原本为 2 的值修改为 0，并令 `user.num=0`。最后，判断 `head` 是否为空，若为空，则给 `head` 分配空间，并令 `head->next=NULL`。清空链表的流程图如图 3.2.4 所示。

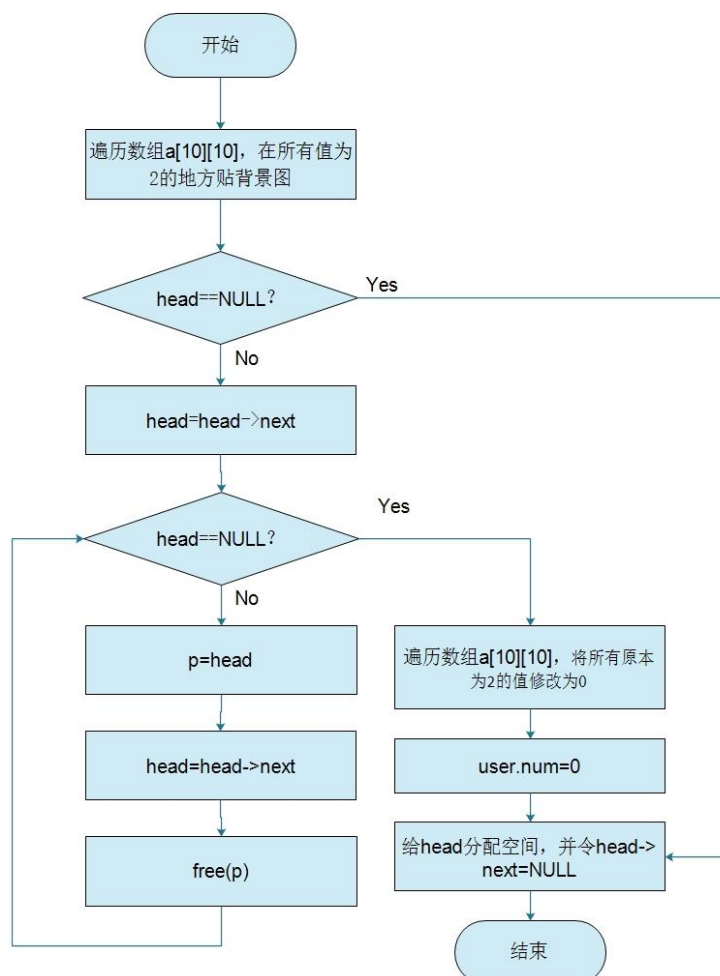


图 3.2.4 清空道具的流程图

### 3.2.4 文件管理

本模块用于和文件相关的操作，包括存档，读取存档，游戏结束后记录用户名和得分。

相关函数：

◆ **void score(USER user);**

功能：游戏结束后将用户名和得分追加到指定文件。定义局部变量 FILE\* fp，如果成功打开文件，则将 user.name 和 user.score 以追加到末尾的方式存入文件。

◆ **void Remember(USER user, int a[10][10], tool\* head, int gamemode, People people);**

功能：将当前游戏的状态（人、小花、炸弹、用户名、分数等所有信息）存入文件。定义了局部变量 FILE\* fp, tool\* p, 如果成功打开文件，则以二进制方式将人物当前坐标、宝箱信息（坐标、是花还是炸弹）、用户信息（用户名、用户得分、用户设定的道具数量）、迷宫地图存入文件。

◆ **void Read(USER& user);**

功能：读取存档，将文件中存储的游戏状态恢复到屏幕上。定义了局部变量 FILE\* fp1; tool\* p, \* head; People people; int gamemode; int a[10][10]; 如果成功打开文件，则以二进制方式依次读取人物当前坐标、宝箱信息（坐标、是花还是炸弹）、用户信息（用户名、用户得分、用户设定的道具数量）、迷宫地图存入相应变量。

### 3.2.5 结束游戏

本模块用于退出游戏。

相关函数：

◆ **void Gameover();**

功能：调用 exit(0)退出游戏。

## 4 测试

对程序从下面几部分进行了测试。

### 4.1 游戏时间测试

测试用例

进入游戏界面时开始计时，并实时显示游戏已进行时长。如图 4.1 所示。

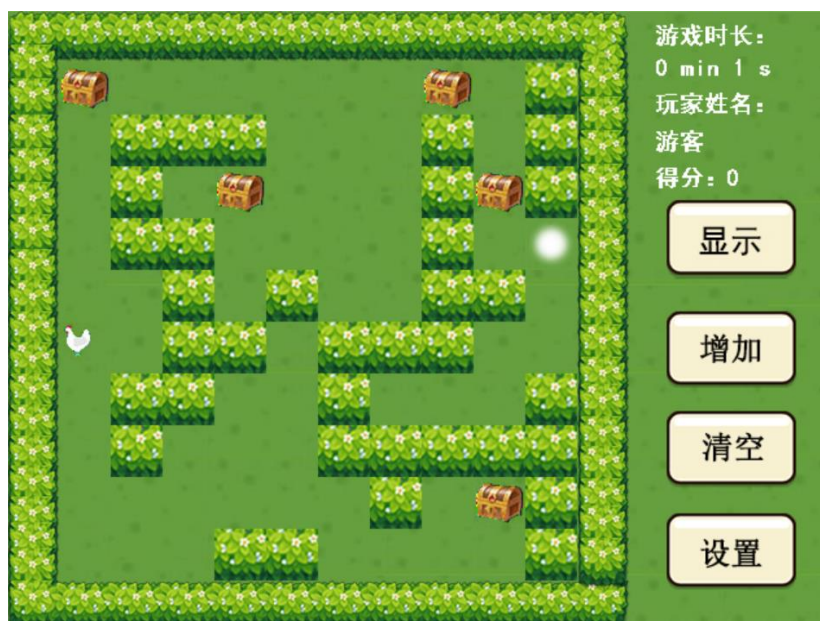


图 4.1 进入游戏开始计时

测试结果

在游戏界面显示的时长能实时变化，如图 4.2 所示。

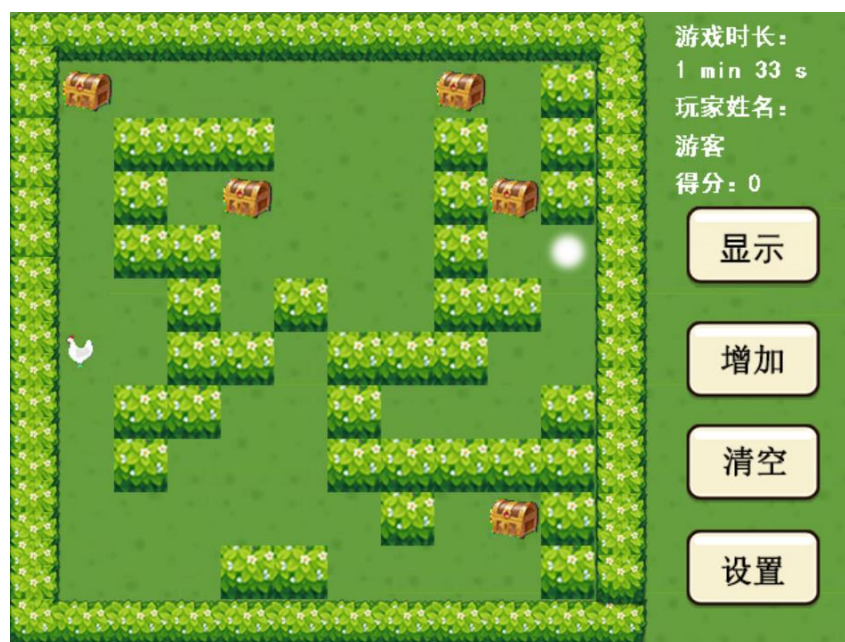


图 4.2 实时显示时长的界面

## 4.2 用户信息测试

测试用例：在开始菜单界面时若选择开始游戏，则弹出输入框，输入用户名（若不输入，则默认为游客）和道具数量（若不输入，则默认为5）。如图 4.3 和 4.4 所示。



图 4.3 输入用户名的界面



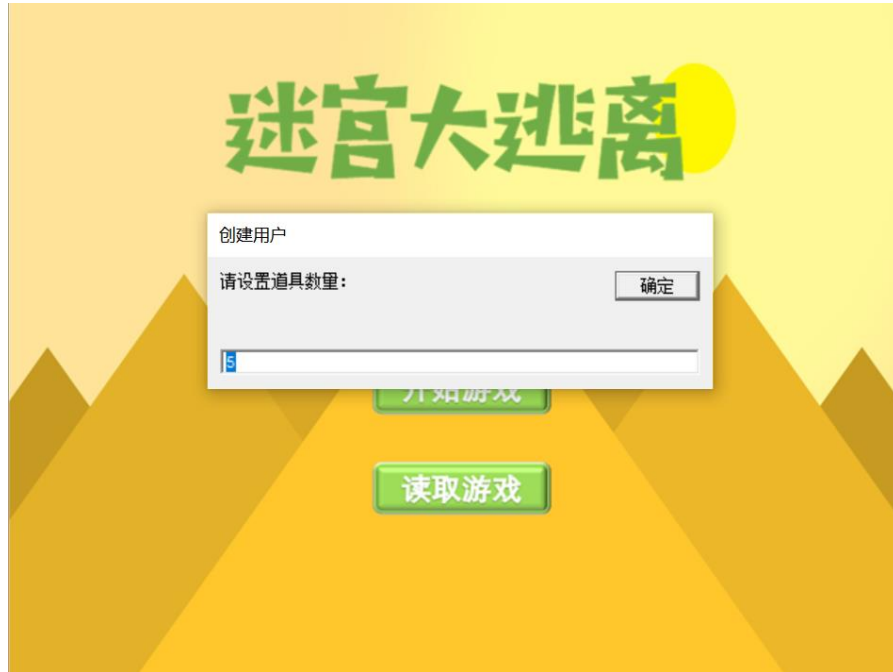


图 4.4 输入道具数量的界面

测试结果：能弹出输入框，输入用户信息并显示在游戏界面信息区。如图 4.5 所示。

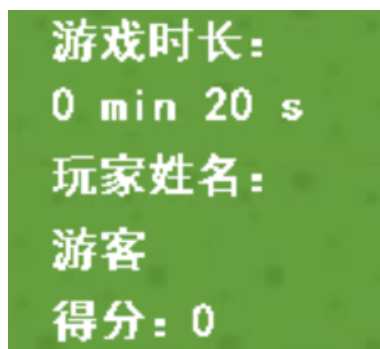


图 4.5 显示用户名和得分的信息区

### 4.3 弹出菜单栏测试

测试用例：游戏中单击“设置”按钮时弹出游戏设置菜单栏，点击“帮助”按钮时弹出帮助页面，再次点击“设置”按钮返回游戏。

测试结果：能成功弹出页面和返回游戏。如图 4.6 和 4.7 所示。



图 4.6 游戏设置菜单栏

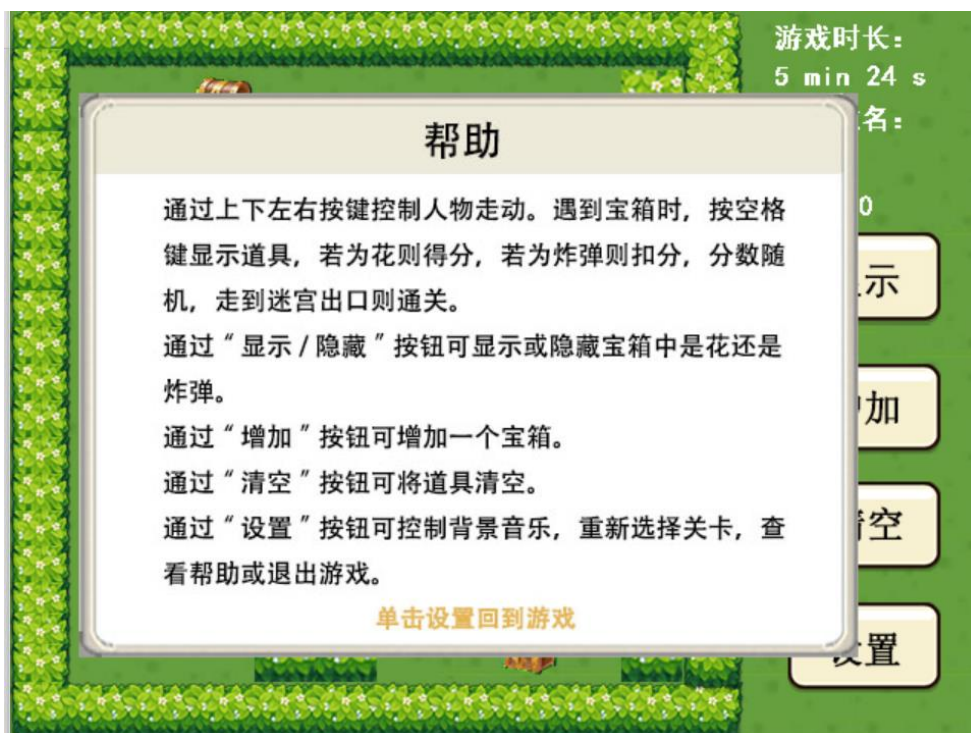


图 4.7 帮助页面

## 4.4 游戏中途退出存档测试

测试用例：游戏中单击“设置”按钮时弹出游戏设置菜单栏，点击“退出”按钮时消息弹窗提示，退出游戏并存档。

测试结果：成功弹出页面提示并用二进制方式完成存档。如图 4.8 和 4.9 所示。



图 4.8 退出游戏弹窗

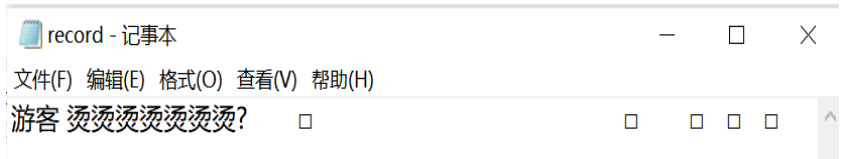


图 4.9 存档文件

## 4.5 游戏结束退出并存储用户信息测试

测试用例：走到出口时，弹出消息框，点击“结束”按钮后结束程序，并将用户名和得分追加到指定文件中。

测试结果：成功退出游戏并存储用户信息，如图 4.10 和 4.11 所示。



图 4.10 游戏胜利弹窗

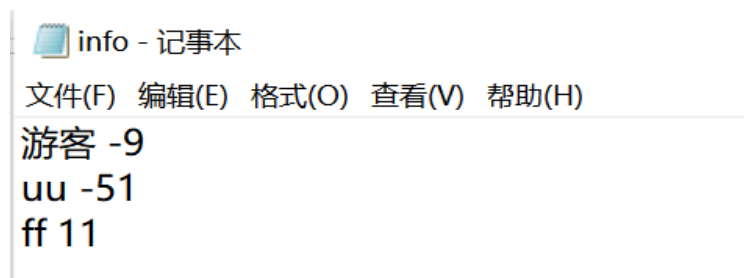


图 4.11 存储用户信息文件

## 4.6 游戏读档测试

测试用例：在开始菜单界面时若点击“读取游戏”按钮，则从指定文件中读取存档，恢复上次游戏状态，若无存档，则弹出消息提示。

测试结果：有存档时能恢复游戏状态，无存档时能弹出提示。如图 4.12 所示。

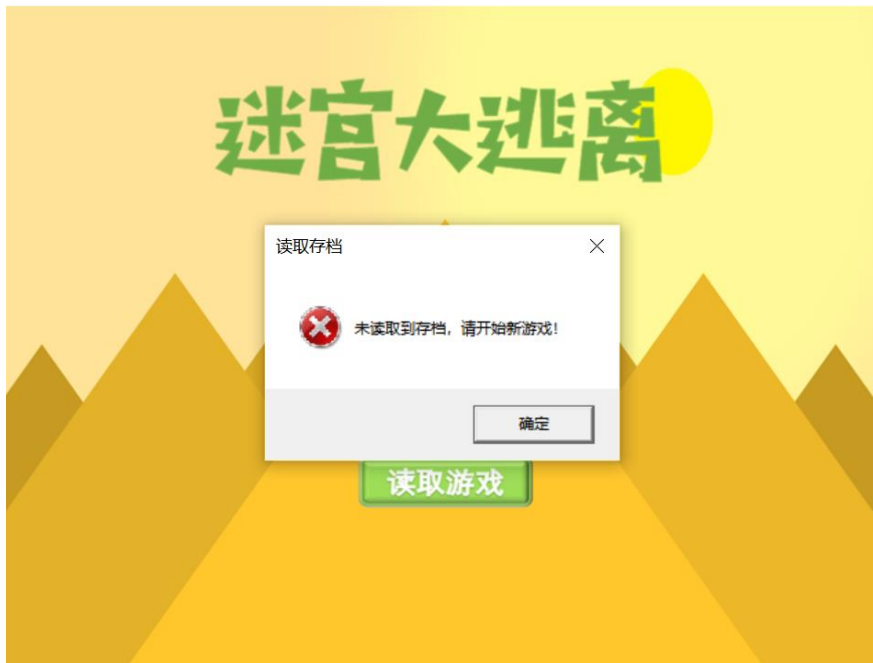


图 4.12 未读取到存档消息提示

## 5 开发中遇到的问题及解决办法

### 5.1 弹出菜单栏后回到游戏页面

游戏界面中单击“设置”按钮时弹出游戏设置菜单栏，再次点击设置时回到游戏页面，为了避免每次都重新绘制整个界面，先用 `getimage` 函数保存即将画图的那片区域，再贴上游戏设置菜单栏的图片，恢复时用 `putimage` 函数把保存的图像归位。

### 5.2 批量绘图实现人物的移动

游戏中需要用键盘控制人物在图像上移动，为了避免每次移动时都重新绘制地图，采用 `getimage` 预先保存绘图区，在新的位置绘制人物，等待，再用预先保存的图像覆盖绘制的图像。在实现人物移动的过程中发现存在闪屏问题，故用 `BeginBatchDraw()` 和 `FlushBatchDraw()` 批量绘图，再 `Sleep()` 适当时间，解决了闪屏问题。

### 5.3 计时的实现

在游戏开始时调用 `time` 函数获取当前系统时间，每次进入循环时调用 `time` 函数，每次返回值与初始时调用返回值的差即为游戏已进行时长（秒数），再利用 `localtime` 函数换算单位。

## 5.4 用二进制方法存档和读档

存档时需要存入人物当前坐标、宝箱信息（坐标、是花还是炸弹）、用户信息（用户名、用户得分、用户设定的道具数量）、迷宫地图，其中人物坐标和用户信息是以结构体变量储存，考虑到结构体所占字符数不规则，故用二进制方式向文件写入数据和从文件读取数据，调用了 `fread` 和 `fwrite` 函数。

# 6 总结

## 6.1 课程设计总结

本学期的《高级语言程序设计课设》课程，让我们运用上学期所学的 C 语言知识，同时结合了 EasyX 库图形界面，编写迷宫大逃离游戏。经过两个多月的编写和测试，我对 C 语言知识的掌握程度、对模块化程序设计的理解和运用、对代码的编写能力均有大幅提升。不仅如此，我还收获了自主学习、自主纠错的能力，在代码编写期间，我也遇到过各种各样的 bug，学会了通过查阅资料、阅读文献来解决问题，并且明白了代码的规范度、框架的整洁度、逻辑的合理度的重要性。总的来说，我对自己这次课设的完成程度较为满意。

## 6.2 对本课程意见与建议

《高级语言程序设计课程设计》这门课程的开设是为了培养我们的程序分析和设计能力，掌握结构化程序设计的思想和方法。希望今后在对于游戏的选择上可以更好地兼顾趣味性和难度，既能提高同学对课程的兴趣，也能更好地锻炼同学的能力。

## 附件：程序源代码

```
//系统函数库
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include <time.h>
#include <windows.h>
#include<mmsystem.h>
```

```

#pragma comment(lib,"winmm.lib")
////////////////////以下是 宏定义参数 //////////////////////

#define      WIDE      640      //屏幕宽度
#define      HIGH      480      //屏幕高度

#define  UP      72
#define  DOWN    80
#define  LEFT    75
#define  RIGHT   77
#define  SPACE   32

////////////////////以下是 全局变量 //////////////////////

IMAGE      menu1, menu2, menu3, set, help, end, door1, door2, door3;
//菜单背景
IMAGE      back1, back2, back3, img1, img2, img3, img4, man1, man2, wall1, wall2, wall3, box1,
box2, flower1, flower2, boom1, boom2, xianshi1, yincang1, xianshi2, yincang2, xianshi3, yincang3;
//游戏场景背景

////////////////////以下是数据结构定义////////////////////

//用户信息
typedef struct users {
    char      name[20];          //用户名
    int       num = 5;           //道具数量
    int       score = 0;         //用户得分
    //struct   users* next;
}USER;

//道具信息
typedef struct Tool {
    int       x, y;              //坐标
    int       flag;              //花 or 炸弹 0 花 1 炸弹
    struct    Tool* next;
}tool;

typedef struct {
    int x;
    int y;
}People;
////////////////////以下是 函数声明 //////////////////////

void                                                LoadImg();

```

```

//加载图片
void Music();
//播放背景音乐
void Startmenu(USER& user);
//加载开始菜单界面
void Mainmenu(USER& user);
//加载游戏菜单界面
int checkButton(int mouseX, int mouseY);
//鼠标是否在按钮内
void Initgame(USER& user, int gamemode);
//初始化迷宫
void InGame(USER& user, int a[10][10], tool*& head, int gamemode, People people);
//进入游戏
void Display(People people);
//绘制人物
void gametime(time_t starttime);
//游戏时长
void Gameover();
//退出游戏
void Remember(USER user, int a[10][10], tool* head, int gamemode, People people);
//存档
void score(USER user);
//存用户信息
void Read(USER& user);
//读档

void Mode1(USER& user, int a[10][10], tool*& head, People people);
//关卡 1 绘图
void Mode2(USER& user, int a[10][10], tool*& head, People people);
//关卡 2 绘图
void Mode3(USER& user, int a[10][10], tool*& head, People people);
//关卡 3 绘图
void addBox(USER& user, int a[10][10], tool*& head, tool*& p);
//增加道具
void delBox(USER& user, int a[10][10], tool*& head, tool*& p);
//清空道具

int main()
{
    USER user; //创建用户
    initgraph(WIDE, HIGH); //初始化窗口
    SetWindowText(GetHWnd(), "迷宫大逃离:-"); //设置窗口标题
    srand((unsigned)time(NULL));

```



```

LoadImg();                                //加载图片
Music();                                  //开启背景音乐
putimage(0, 0, &menu1);
Startmenu(user);
closegraph();
return 0;

}

void LoadImg()
{
    loadimage(&menu1, "image\\1.png");
    loadimage(&menu2, "image\\2.png");
    loadimage(&menu3, "image\\3.png");
    loadimage(&back1, "image\\4.png");
    loadimage(&wall1, "image\\5.jpg");
    loadimage(&man1, "image\\man1.jpg");
    loadimage(&man2, "image\\man2.jpg");
    loadimage(&box1, "image\\6.jpg");
    loadimage(&box2, "image\\7.jpg");
    loadimage(&boom1, "image\\8.jpg");
    loadimage(&boom2, "image\\9.jpg");
    loadimage(&flower1, "image\\10.jpg");
    loadimage(&flower2, "image\\11.jpg");
    loadimage(&set, "image\\12.jpg", 460, 200);
    loadimage(&help, "image\\13.jpg");
    loadimage(&end, "image\\14.jpg");
    loadimage(&back2, "image\\16.png");
    loadimage(&back3, "image\\17.png");
    loadimage(&wall2, "image\\18.png");
    loadimage(&wall3, "image\\19.png");
    loadimage(&door1, "image\\door1.jpg");
    loadimage(&door2, "image\\door2.jpg");
    loadimage(&door3, "image\\door3.jpg");
    loadimage(&xianshi1, "image\\xianshi1.jpg");
    loadimage(&xianshi2, "image\\xianshi2.jpg");
    loadimage(&xianshi3, "image\\xianshi3.jpg");
    loadimage(&yincang1, "image\\yincang1.jpg");
    loadimage(&yincang2, "image\\yincang2.jpg");
    loadimage(&yincang3, "image\\yincang3.jpg");
}

void Music() {
    mciSendString("open image\\BGM.mp3 alias bgm", 0, 0, 0);
}

```

```

        mciSendString("play bgm repeat", 0, 0, 0);
    }

void Startmenu(USER& user)
{
    MOUSEMSG mmsg;
    char numb[15];
    int flag = 0;
    int i = 1;
    while (i) //
    在开始菜单界面时
    {
        if (MouseHit())
        {
            mmsg = GetMouseMsg();
            if (mmsg.x > 264 && mmsg.x < 376 && mmsg.y > 253 && mmsg.y < 284) //开
            始游戏按钮
            {
                if (WM_LBUTTONDOWN == mmsg.uMsg)
                {
                    putimage(0, 0, &menu2);
                    flag = 1;
                }
                else if (WM_LBUTTONUP == mmsg.uMsg && flag == 1)
                {
                    flag = 2;
                    i = 0;
                }
            }
            if (mmsg.x > 264 && mmsg.x < 376 && mmsg.y > 319 && mmsg.y < 349) //
            读取游戏按钮
            {
                if (WM_LBUTTONDOWN == mmsg.uMsg)
                {
                    putimage(0, 0, &menu3);
                    flag = 3;
                }
                else if (WM_LBUTTONUP == mmsg.uMsg && flag == 3)
                {
                    flag = 4;
                    i = 0;
                }
            }
            FlushMouseMsgBuffer();
        }
    }
}

```

```

    }
}
if (flag == 2)
{
    putimage(0, 0, &menu1);
    Sleep(15);
    i = InputBox(user.name, 20, "请输入玩家昵称: ", "创建用户", NULL, 0, 0, false);
//创建用户信息
    if (i == false)
    {
        MessageBox(GetHWnd(), "您已取消!", "创建用户", MB_OK | MB_ICONHAND);
        Startmenu(user);
    }
    else
    {
        if (strlen(user.name) == 0)
            strcpy(user.name, "游客");
        InputBox(numb, 10, "请设置道具数量: ", "创建用户", "5", 0, 0);
        user.num = atoi(numb);
        Mainmenu(user);
//进入游戏菜单
    }
}
else if (flag == 4)
{
    putimage(0, 0, &menu1);
    Read(user);
//读取上次存档
}
}

void Mainmenu(USER& user)
{
    int mouseX = 0, mouseY = 0;
    int isMouseDown = 0;
    int gamemode = 0;
    MOUSEMSG mmsg;

    for (int i = 0; i < 256; i += 5)
    {
        setbkcolor(RGB(i, i, i));           //设置背景色，原来默认黑色
        cleardevice();                       //清屏
        Sleep(15);                           //延时 15ms
    }
}

```

```

setbkmode(TRANSPARENT);
//标题
settextcolor(RGB(241, 149, 60));
settextstyle(58, 30, "黑体", 0, 0, 600, false, false, false);
outtextxy(152, 92, "选择游戏关卡");
settextcolor(RGB(255, 227, 153));
outtextxy(150, 90, "选择游戏关卡");

FlushMouseMsgBuffer();
BeginBatchDraw();
while (gamemode == 0)
{
    //游戏模式选择按钮
    setlinecolor(RGB(255, 199, 44));
    setlinestyle(PS_SOLID, 1);
    settextstyle(26, 10, _T("宋体"));
    setfillcolor(RGB(255, 199, 44));
    if (MouseHit()) //读取鼠标信息
    {
        mmsg = GetMouseMsg();
        switch (mmsg.uMsg)
        {
            case WM_MOUSEMOVE:      mouseX = mmsg.x; mouseY = mmsg.y; break;
            case WM_LBUTTONDOWN:    isMouseDown = 1; break;
            case WM_LBUTTONUP:      isMouseDown = 0; break;
        }
    }
    //模式 1 按钮
    if (checkButton(mouseX, mouseY) == 1)
    {
        settextcolor(RGB(255, 199, 44));
        if (isMouseDown) //按键响应
            gamemode = 1;
    }
    else
        settextcolor(RGB(255, 255, 255));
    fillrectangle(250, 220, 375, 255);
    outtextxy(275, 225, _T("花园寻宝"));

    //模式 2 按钮
    if (checkButton(mouseX, mouseY) == 2)
    {
        settextcolor(RGB(255, 199, 44));
    }
}

```

```

        if (isMouseDown)
            gamemode = 2;
    }
    else
        settextcolor(RGB(255, 255, 255));
    fillrectangle(250, 280, 375, 315);
    outtextxy(275, 285, _T("沙漠探险"));

    //模式 3 按钮
    if (checkButton(mouseX, mouseY) == 3)
    {
        settextcolor(RGB(255, 199, 44));
        if (isMouseDown)
            gamemode = 3;
    }
    else
        settextcolor(RGB(255, 255, 255));
    fillrectangle(250, 340, 375, 375);
    outtextxy(275, 345, _T("勇闯矿山"));
    FlushBatchDraw();
}
EndBatchDraw();
Initgame(user, gamemode);           //初始化迷宫
}

int checkButton(int mouseX, int mouseY)
{
    if (mouseX >= 250 && mouseX <= 375 && mouseY >= 220 && mouseY <= 255)
        return 1;
    else if (mouseX >= 250 && mouseX <= 375 && mouseY >= 280 && mouseY <= 315)
        return 2;
    else if (mouseX >= 250 && mouseX <= 375 && mouseY >= 340 && mouseY <= 375)
        return 3;
    else
        return -1;           //没选中
}

void Initgame(USER& user, int gamemode)           //1 墙 2
道具 3 出口
{
    int a[3][10][10] =
    {
        {
            {0,0,0,0,0,0,0,0,0,1},

```

```

        {0,1,1,1,0,1,0,1,0,1},
        {0,1,0,0,0,1,0,1,0,1},
        {0,1,1,0,1,1,0,1,0,1},
        {0,0,1,0,0,0,0,1,0,0},
        {0,0,1,1,0,1,1,1,1,0},
        {0,1,1,0,0,1,3,0,0,1},
        {0,1,0,1,0,1,1,1,0,1},
        {0,0,0,1,0,0,1,0,0,1},
        {0,0,1,1,1,0,0,0,1,1}
    },
    {
        {0,0,0,0,0,1,0,0,0,0},
        {0,1,1,1,0,0,0,1,0,1},
        {0,1,0,0,0,1,1,1,0,0},
        {0,1,1,0,0,0,0,1,1,0},
        {0,0,1,1,1,0,0,1,1,0},
        {0,0,1,0,3,1,1,1,0,0},
        {0,1,1,0,0,0,0,0,1,0},
        {0,1,0,1,1,1,0,1,1,0},
        {0,0,0,0,1,0,0,1,0,0},
        {0,0,1,1,1,1,0,0,0,1}
    },
    {
        {0,0,0,0,0,0,0,0,0,1},
        {0,1,1,1,0,0,0,1,0,0},
        {0,1,0,0,0,1,1,1,1,0},
        {0,1,1,0,0,0,0,0,1,0},
        {0,0,1,0,1,1,1,0,0,1},
        {0,0,1,1,0,1,0,1,0,0},
        {0,1,1,0,0,1,0,0,0,0},
        {0,1,0,0,1,1,1,1,1,0},
        {0,0,0,0,0,1,1,0,0,0},
        {0,0,0,1,1,3,0,0,0,1}
    }
};

tool* head, * p;
People people = { 5,0 };
head = (tool*)malloc(sizeof(tool));
head->next = NULL;
for (int i = 0; i < user.num; i++)
{
    p = (tool*)malloc(sizeof(tool));
    do
    {

```

```

        p->x = rand() % 10;
        p->y = rand() % 10;
    } while (a[gamemode - 1][p->x][p->y] != 0);
    a[gamemode - 1][p->x][p->y] = 2;
    p->flag = rand() % 2;
    p->next = head->next;
    head->next = p;
}
switch (gamemode)
{
case 1:    Mode1(user, a[0], head, people); break;
case 2:    Mode2(user, a[1], head, people); break;
case 3:    Mode3(user, a[2], head, people); break;
}
}

void    Mode1(USER& user, int a[10][10], tool*& head, People people)
{
    putimage(0, 0, &back1);
    putimage(509, 146, &xianshi1);
    getimage(&img2, 40, 40, 40, 40);
    getimage(&img3, 40, 40, 110, 20);
    BeginBatchDraw();
    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            if (a[i][j] == 1)
                putimage((j + 1) * 40, (i + 1) * 40, &wall1);
            if (a[i][j] == 2)
            {
                putimage((j + 1) * 40, (i + 1) * 40, &box1, SRCAND);
                putimage((j + 1) * 40, (i + 1) * 40, &box2, SRCINVERT);
            }
            if (a[i][j] == 3)
                putimage((j + 1) * 40, (i + 1) * 40, &door1);
        }
    }
    EndBatchDraw();
    InGame(user, a, head, 1, people);
}

void    Mode2(USER& user, int a[10][10], tool*& head, People people)

```

```

{
    putimage(0, 0, &back2);
    putimage(509, 146, &xianshi2);
    getimage(&img2, 40, 40, 40, 40);
    getimage(&img3, 40, 40, 110, 20);
    BeginBatchDraw();
    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            if (a[i][j] == 1)
                putimage((j + 1) * 40, (i + 1) * 40, &wall2);
            if (a[i][j] == 2)
            {
                putimage((j + 1) * 40, (i + 1) * 40, &box1, SRCAND);
                putimage((j + 1) * 40, (i + 1) * 40, &box2, SRCINVERT);
            }
            if (a[i][j] == 3)
                putimage((j + 1) * 40, (i + 1) * 40, &door2);
        }
    }
    EndBatchDraw();
    InGame(user, a, head, 2, people);
}

```

```

void Mode3(USER& user, int a[10][10], tool*& head, People people)
{
    putimage(0, 0, &back3);
    putimage(506, 145, &xianshi3);
    getimage(&img2, 40, 40, 40, 40);
    getimage(&img3, 40, 40, 110, 20);
    BeginBatchDraw();
    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            if (a[i][j] == 1)
                putimage((j + 1) * 40, (i + 1) * 40, &wall3);
            if (a[i][j] == 2)
            {
                putimage((j + 1) * 40, (i + 1) * 40, &box1, SRCAND);
                putimage((j + 1) * 40, (i + 1) * 40, &box2, SRCINVERT);
            }
        }
    }
}

```



```

        if (a[i][j] == 3)
            putimage((j + 1) * 40, (i + 1) * 40, &door3);
    }
}
EndBatchDraw();
InGame(user, a, head, 3, people);
}

void InGame(USER& user, int a[10][10], tool*& head, int gamemode, People people)
{
    tool* p, * r;
    MOUSEMSG mmsg;
    time_t starttime; //开始时间
    starttime = time(NULL);
    char s[10]; //存得分
    int flag = 0; //0 道具隐藏状态 1 道具显示状态
    int con = 0; //0 设置关闭状态 1 设置显示状
    态 2 返回游戏
    int isgame = 0; // 0 游戏中 1 回到选择关卡 2
    游戏结束
    int isBgm = 0; //0 音乐播放状态 1 音乐暂停
    状态
    setbkmode(TRANSPARENT);
    sprintf(s, "%d", user.score);
    settextcolor(RED);
    settextstyle(18, 0, "黑体", 0, 0, 600, false, false, false);
    outtextxy(500, 10, "游戏时长: ");
    outtextxy(500, 63, "玩家姓名: ");
    outtextxy(500, 92, user.name);
    outtextxy(500, 120, "得分: ");
    outtextxy(555, 120, s);
    BeginBatchDraw();
    while (isgame != 1)
    {
        if (MouseHit())
        {
            mmsg = GetMouseMsg();
            if (WM_LBUTTONDOWN == mmsg.uMsg)
            {
                if (mmsg.x > 510 && mmsg.x < 610 && mmsg.y > 148 && mmsg.y < 200)
                //按钮“显示/隐藏”响应
                {
                    //此处加上道具显示代码
                    if (flag == 0)

```

```

{
    switch (gamemode)
    {
        case 1: putimage(509, 146, &yincang1); break;
        case 2: putimage(509, 146, &yincang2); break;
        case 3: putimage(506, 145, &yincang3); break;
    }
    p = head;
    while (p != NULL)
    {
        if (p->flag == 0)
        {
            putimage((p->y + 1) * 40, (p->x + 1) * 40, &img2);
            putimage((p->y + 1) * 40, (p->x + 1) * 40, &flower1,
SRCAND);
            putimage((p->y + 1) * 40, (p->x + 1) * 40, &flower2,
SRCINVERT);
        }
        if (p->flag == 1)
        {
            putimage((p->y + 1) * 40, (p->x + 1) * 40, &img2);
            putimage((p->y + 1) * 40, (p->x + 1) * 40, &boom1,
SRCAND);
            putimage((p->y + 1) * 40, (p->x + 1) * 40, &boom2,
SRCINVERT);
        }
        p = p->next;
    }
    flag = 1;
}
else if (flag == 1)
{
    switch (gamemode)
    {
        case 1: putimage(509, 146, &xianshi1); break;
        case 2: putimage(509, 146, &xianshi2); break;
        case 3: putimage(506, 145, &xianshi3); break;
    }
    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            if (a[i][j] == 2)
            {

```

```

        putimage((j + 1) * 40, (i + 1) * 40, &img2);
        putimage((j + 1) * 40, (i + 1) * 40, &box1, SRCAND);
        putimage((j + 1) * 40, (i + 1) * 40, &box2, SRCINVERT);
    }
}
}
flag = 0;
}
}
if (mmsg.x > 510 && mmsg.x < 610 && mmsg.y > 235 && mmsg.y < 287)
//按钮“增加”响应
{
    //此处加上道具增加代码
    addBox(user, a, head, p);
}
if (mmsg.x > 510 && mmsg.x < 610 && mmsg.y > 310 && mmsg.y < 362)
//按钮“清空”响应
{
    //此处加上道具清空代码
    delBox(user, a, head, p);
}
if (mmsg.x > 510 && mmsg.x < 610 && mmsg.y > 390 && mmsg.y < 442)
//按钮“设置”响应
{
    //此处加上设置代码
    if (con == 0)
    {
        getimage(&img4, 45, 55, 510, 370);
        putimage(90, 140, &set);
        con = 1;
    }
    else if (con == 1) //重新绘制界面
    {
        putimage(45, 55, &img4);
        con = 0;
    }
}
if (con == 1)
//在设置界面
{
    if (mmsg.x > 110 && mmsg.x < 190 && mmsg.y > 230 && mmsg.y < 263)
//按钮“帮助”响应
{
    putimage(45, 55, &help);
}
}

```

```

    }
    if (mmsg.x > 220 && mmsg.x < 300 && mmsg.y>230 && mmsg.y < 263)
//按钮 “关卡"响应
    {
        user.score = 0;
        isgame = 1;
    }
    if (mmsg.x > 330 && mmsg.x < 410 && mmsg.y>230 && mmsg.y < 263)
//按钮 “音乐"响应
    {
        if (isBgm == 0) {
            mciSendString("resume bgm", 0, 0, 0);
            isBgm = 1;
        }
        else if (isBgm == 1) {
            mciSendString("pause bgm", 0, 0, 0);
            isBgm = 0;
        }
    }
    if (mmsg.x > 440 && mmsg.x < 520 && mmsg.y>230 && mmsg.y < 263)
//按钮 “退出"响应
    {
        Remember(user, a, head, gamemode, people);
        MessageBox(GetHWnd(), "成功退出游戏并存档！ ", "退出游戏",
MB_OK);

        Gameover();
    }
}
if (isgame == 2)
{
    if (mmsg.x > 185 && mmsg.x < 270 && mmsg.y>272 && mmsg.y < 310)
    {
        isgame = 1;
        break;
    }
    if (mmsg.x > 350 && mmsg.x < 430 && mmsg.y>272 && mmsg.y < 310)
    {
        score(user);
        Gameover();
    }
}
FlushBatchDraw();
}
FlushMouseMsgBuffer();

```

```

    }
    if ((GetAsyncKeyState(VK_UP) & 1) == 1)
    {
        if (people.x > 0 && a[people.x - 1][people.y] != 1)           // 目标位
置在玩家位置的上边
        {
            if (a[people.x][people.y] == 0)
                putimage((people.y + 1) * 40, (people.x + 1) * 40, &img2);
            people.x--;
        }
    }
    if ((GetAsyncKeyState(VK_DOWN) & 1) == 1)
    {
        if (people.x < 9 && a[people.x + 1][people.y] != 1)           // 目标位置在玩
家位置的下边
        {
            if (a[people.x][people.y] == 0)
                putimage((people.y + 1) * 40, (people.x + 1) * 40, &img2);
            people.x++;
        }
    }
    if ((GetAsyncKeyState(VK_LEFT) & 1) == 1)
    {
        if (people.y > 0 && a[people.x][people.y - 1] != 1)           // 目标位置在玩
家位置的左边
        {
            if (a[people.x][people.y] == 0)
                putimage((people.y + 1) * 40, (people.x + 1) * 40, &img2);
            people.y--;
        }
    }
    if ((GetAsyncKeyState(VK_RIGHT) & 1) == 1)
    {
        if (people.y < 9 && a[people.x][people.y + 1] != 1)           // 目标位置在玩
家位置的右边
        {
            if (a[people.x][people.y] == 0)
                putimage((people.y + 1) * 40, (people.x + 1) * 40, &img2);
            people.y++;
        }
    }
    if ((GetAsyncKeyState(VK_SPACE) & 1) == 1)
    {

```

```

r = head;
p = head->next;
while (p != NULL)
{
    if (p->x == (people.x) && p->y == (people.y))
    {
        if (p->flag == 0)
        {
            putimage((p->y + 1) * 40, (p->x + 1) * 40, &img2);
            putimage((p->y + 1) * 40, (p->x + 1) * 40, &flower1, SRCAND);
            putimage((p->y + 1) * 40, (p->x + 1) * 40, &flower2, SRCINVERT);
            user.score += rand() % 10 + 1;          //随机加 1-10

            putimage(555, 120, &img3);

        }
        else if (p->flag == 1)
        {
            putimage((p->y + 1) * 40, (p->x + 1) * 40, &img2);
            putimage((p->y + 1) * 40, (p->x + 1) * 40, &boom1, SRCAND);
            putimage((p->y + 1) * 40, (p->x + 1) * 40, &boom2, SRCINVERT);
            user.score -= rand() % 10 + 1;          //随机加 1-10

            putimage(555, 120, &img3);

        }
        sprintf(s, "%d", user.score);          //动态显示得分
        outtextxy(555, 120, s);
        user.num--;
        a[p->x][p->y] = 0;
        r->next = p->next;
        free(p);
        p = r->next;
    }
    else
    {
        r = p;
        p = p->next;
    }
}
if (isgame != 2)
    gametime(starttime);
if (con == 1)
    FlushBatchDraw();

```

分

分

```

        if (con==0&&isgame == 0)
            Display(people);
        if (a[people.x][people.y] == 3)
        {
            isgame = 2;
            putimage(90, 140, &end);
            FlushBatchDraw();
        }
    }
    EndBatchDraw();

    if (isgame == 1)
        Mainmenu(user);
}

void    Display(People people)
{
    getimage(&img1, (people.y + 1) * 40, (people.x + 1) * 40, 40, 40);
    BeginBatchDraw();
    putimage((people.y + 1) * 40, (people.x + 1) * 40, &man1, SRCAND);
    putimage((people.y + 1) * 40, (people.x + 1) * 40, &man2, SRCINVERT);
    FlushBatchDraw();
    putimage((people.y + 1) * 40, (people.x + 1) * 40, &img1);
}

void    gametime(time_t starttime) {

    struct tm* Tm;
    time_t endtime, thistime;
    char t[20];                                     //存游戏时长
    endtime = time(NULL);
    thistime = endtime - starttime;
    Tm = localtime(&thistime);
    putimage(500, 35, &img3);
    sprintf(t, "%d min %d s", Tm->tm_min, Tm->tm_sec);           //动态显示得
分
    outtextxy(500, 35, t);
}

void    addBox(USER& user, int a[10][10], tool*& head, tool*& p) {

    int s = 0;

```

```

p = (tool*)malloc(sizeof(tool));
do
{
    p->x = rand() % 10;
    p->y = rand() % 10;
    s++;
} while (a[p->x][p->y] != 0 && s < 1000);
if (a[p->x][p->y] == 0) {

    a[p->x][p->y] = 2;
    p->flag = rand() % 2;
    p->next = head->next;
    head->next = p;
    user.num++;
    putimage((p->y + 1) * 40, (p->x + 1) * 40, &box1, SRCAND);
    putimage((p->y + 1) * 40, (p->x + 1) * 40, &box2, SRCINVERT);
}
}

```

```

void delBox(USER& user, int a[10][10], tool*& head, tool*& p) {

```

```

    for (int i = 0; i < 10; i++)
    {
        for (int j = 0; j < 10; j++)
        {
            if (a[i][j] == 2)
                putimage((j + 1) * 40, (i + 1) * 40, &img2);
        }
    }
    if (head != NULL) {
        head = head->next;
        while (head != NULL)
        {
            p = head;
            head = head->next;
            free(p);
        }
        for (int i = 0; i < 10; i++)
        {
            for (int j = 0; j < 10; j++)
            {
                if (a[i][j] == 2)
                    a[i][j] = 0;
            }
        }
    }
}

```



```

        }
        user.num = 0;
    }
    if (head == NULL) {
        head = (tool*)malloc(sizeof(tool));
        head->next = NULL;
    }
}

void Read(USER& user) {

    FILE* fp1;
    tool* p, * head;
    People people;
    int gamemode, a[10][10];
    head = (tool*)malloc(sizeof(tool));
    head->next = NULL;
    if ((fp1 = fopen("record.txt", "r")) == NULL) {
        MessageBox(GetHwnd(), "未读取到存档，请开始新游戏！", "读取存档", MB_OK |
MB_ICONHAND);
        Startmenu(user);
    }
    else if ((fp1 = fopen("record.txt", "r")) != NULL) {
        fread(&user, sizeof(USER), 1, fp1);
        fread(&people, sizeof(People), 1, fp1);
        for (int i = 0; i < 10; i++)
            for (int j = 0; j < 10; j++)
                fread(&a[i][j], sizeof(int), 1, fp1);
        fread(&gamemode, sizeof(int), 1, fp1);
        for (int i = 0; i < user.num; i++) {

            p = (tool*)malloc(sizeof(tool));
            fread(&p->x, sizeof(int), 1, fp1);
            fread(&p->y, sizeof(int), 1, fp1);
            fread(&p->flag, sizeof(int), 1, fp1);
            p->next = head->next;
            head->next = p;
        }
        fclose(fp1);
        switch (gamemode)
        {
            case 1: Mode1(user, a, head, people); break;
            case 2: Mode2(user, a, head, people); break;
            case 3: Mode3(user, a, head, people); break;
        }
    }
}

```

```

    }
}

void    score(USER user) {
    FILE* fp;
    if ((fp = fopen("info.txt", "a")) != NULL) {
        fprintf(fp, "%s %d\n", user.name, user.score);
        fclose(fp);
    }
}

void    Remember(USER user, int a[10][10], tool* head, int gamemode, People people) {
    FILE* fp;
    tool* p;
    if ((fp = fopen("record.txt", "w")) != NULL) {
        fwrite(&user, sizeof(USER), 1, fp);
        fwrite(&people, sizeof(People), 1, fp);
        for (int i = 0; i < 10; i++)
            for (int j = 0; j < 10; j++)
                fwrite(&a[i][j], sizeof(int), 1, fp);
        fwrite(&gamemode, sizeof(int), 1, fp);
        p = head->next;
        while (p != NULL) {
            fwrite(&p->x, sizeof(int), 1, fp);
            fwrite(&p->y, sizeof(int), 1, fp);
            fwrite(&p->flag, sizeof(int), 1, fp);
            p = p->next;
        }
        fclose(fp);
    }
}

void    Gameover() {
    exit(0);
}

```