大连理工大学计算机学院

程序设计训练实验报告

|  |  |
| --- | --- |
| 实验名称 ： | 一个完整的跑酷游戏 |
| 学生姓名 ： | 李锐 |
| 学生学号 ： | 201638023 |
| 联系方式 ： | 18342782034 |
|  |  |
|  |  |

一个完整的跑酷游戏

# 实验简介

**选题背景：**

本人对于游戏开发有着浓厚的兴趣，在之前也曾自己做过一些类似于贪吃蛇、马里奥之类的小游戏，但是一方面那些游戏相对简单，代码量不大，另一方面我对于那些游戏的制作都谈不上完整，比如没有开始界面，死亡动画，背景音乐或是互动菜单等等。

于是，借着这一次程序设计训练的机会，我决定制作一个真正意义上较为完整的游戏，并以此来学习更多的编程知识和提高自己的编程能力。

**实验整体内容：**

一个具有背景音乐、开始界面、选择菜单、死亡动画、图像画面的跑酷类游戏。

规则：w起跳，s滑铲（注：切记为小写）

游戏图片及音乐素材来自于网络

代码量：765行

# 实验目的

培养自己设计游戏的思维与制作游戏的能力；

掌握C++的图像动画处理；

掌握C++的音频播放处理；

掌握Windows的消息响应机制；

了解并部分掌握MFC和Windows API；

提升自己发现BUG并解决BUG的能力。

# 实验相关原理与技术

MFC编程框架

Windows API函数

C++图像及音频处理技术

# 实验环境

**机器配置：**无硬性要求；

**编程环境及应用环境：**

Microsoft Visual Studio 2017

直接解压压缩包中的design文件夹，运行项目文件，然后编译运行其中的.cpp文件即可。

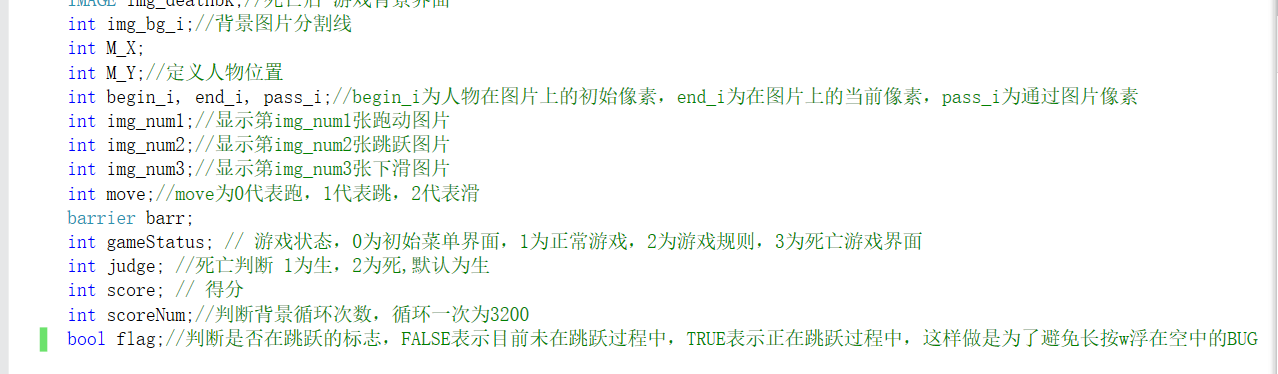
注：若无法找到graphics.h头文件，需安装EasyX图形库。

# 实验方案与过程

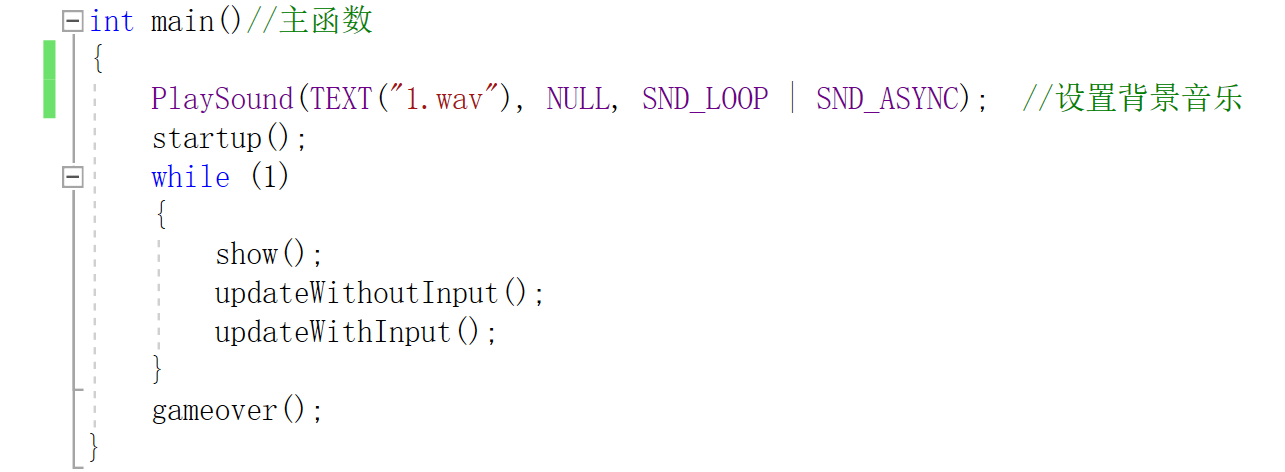
**函数步骤说明及相应代码：**

1. 全局变量及其意义注释如下：





1. 主函数如下：



1. 用PlaySound()设置背景音乐，代码如下：

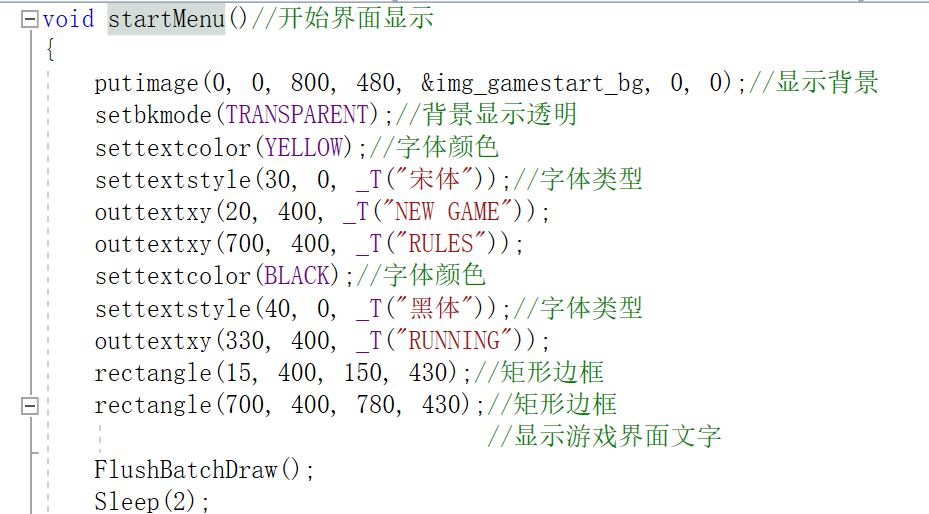


1. startup()用于对图片的加载和对变量的赋初值，代码如下：





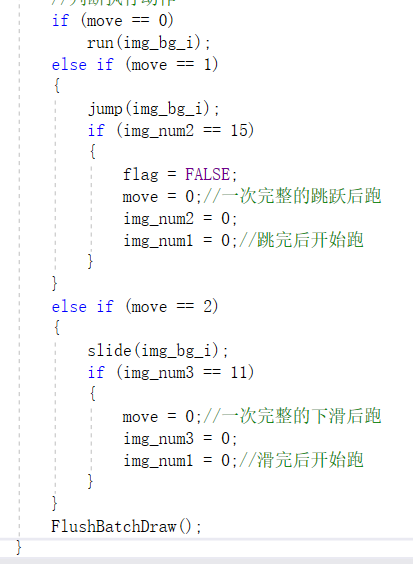
1. 接下来while(1)是一个无限循环，只有当鼠标单击死亡菜单中的EXIT时退出函数，然后执行gameover函数，关闭游戏；
2. show()函数是控制着整个游戏的显示及进程跳转：
3. 当游戏状态的标志变量gameStatus为0时，跳转到游戏开始菜单startMenu()函数，该函数在完成菜单图片显示的同时还接收来自鼠标的信息，根据鼠标左键单击不同的位置相应的改变gameStatus变量的值，其显示图片的代码如下：



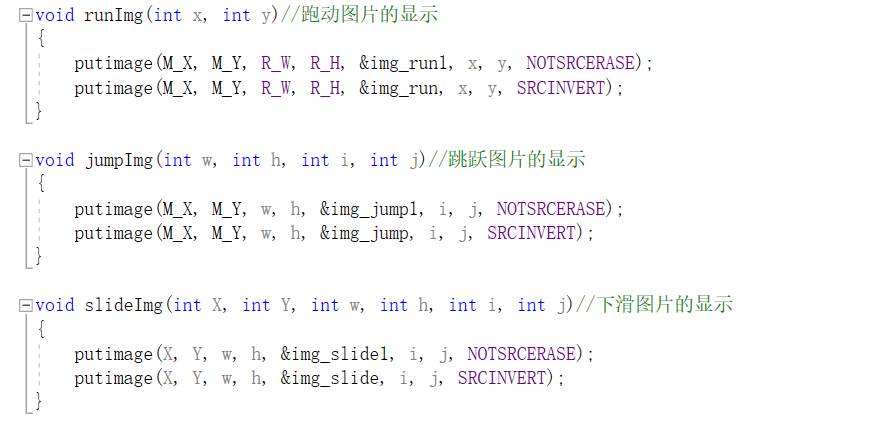
其获取鼠标信息并改变gameStatus变量的值，代码如下：



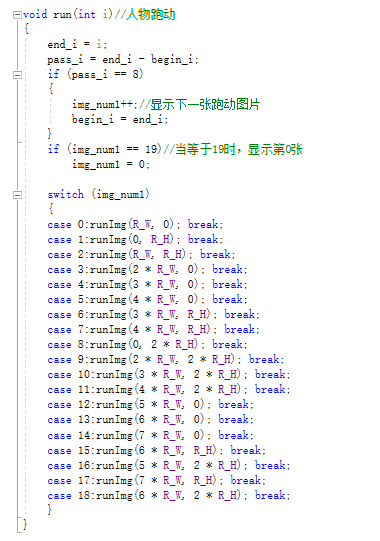
1. 当游戏状态的标志变量gameStatus为2时，跳转到游戏规则界面rules()函数，该函数功能与startMenu()函数几乎一样，只是鼠标点击位置发生了变化，故代码在此处不再给出；
2. 当游戏状态的标志变量gameStatus为3时，跳转到游戏死亡界面showdeathbk()函数，该函数功能与startMenu()函数也几乎一样，只是背景图片和鼠标点击位置发生了变化，并且多了一个数据重新初始化的函数，代码在此处也就不再给出；
3. 而当游戏状态的标志变量gameStatus为1时，则代表此时正在游戏过程中，从而要循环播放背景并且通过move变量的值来判断当前所要做的动作，由于篇幅问题，在此处仅给出判断执行动作的代码：

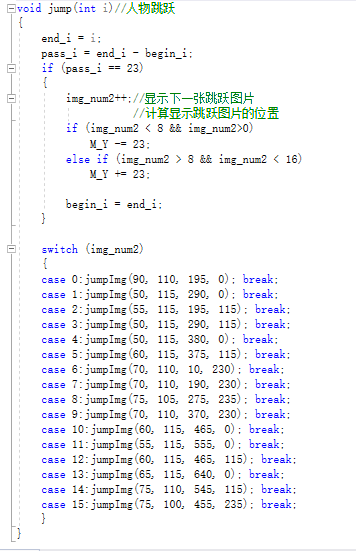


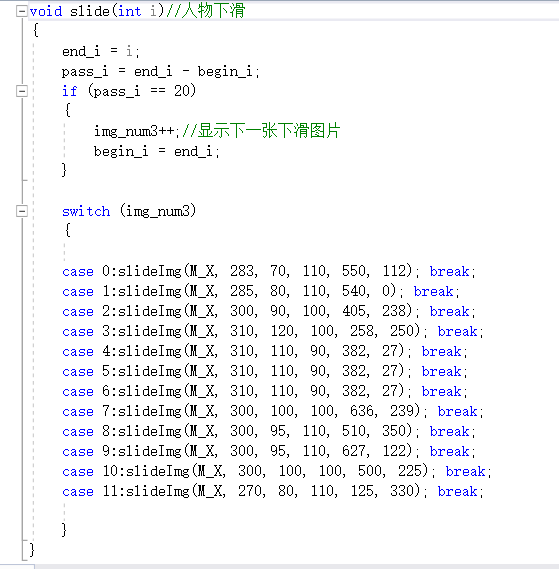
1. 跑、跳、滑铲都有相应的动画显示函数，代码如下：



1. 接下来给出run()、jump()、slide()的代码：





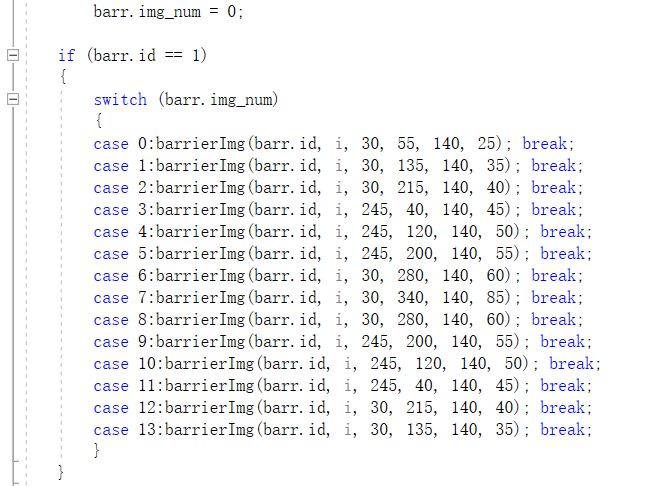


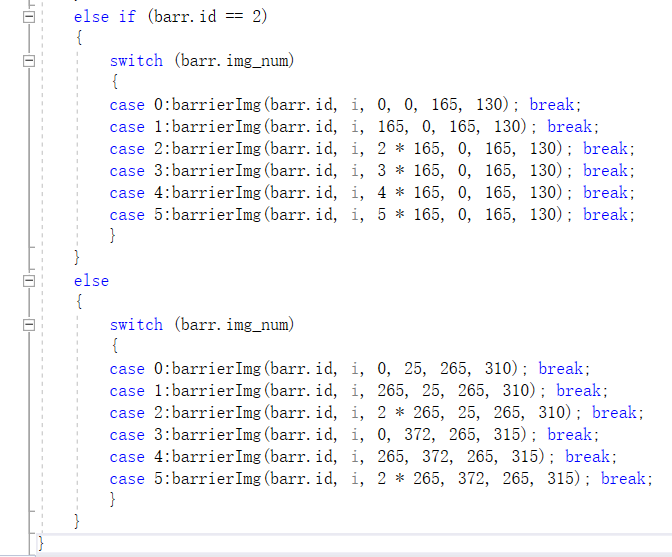
1. 障碍物的显示和碰到障碍物死亡动画函数代码如下：



1. 接下来给出随机产生障碍物类型函数isBarrier()的代码：

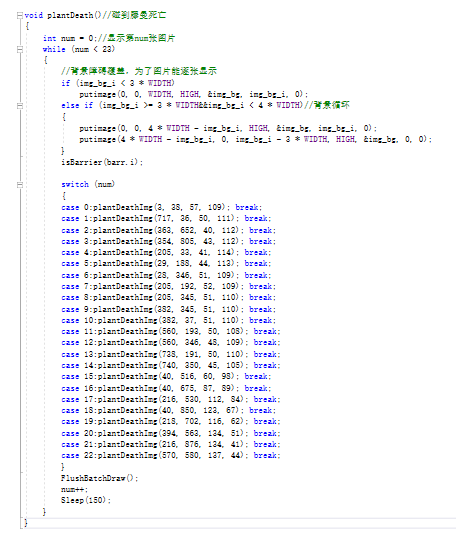




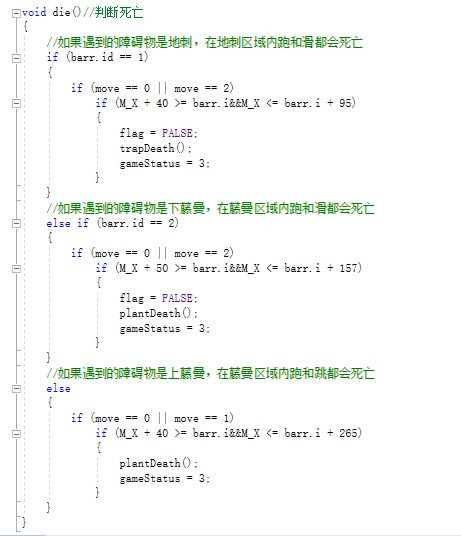


1. 接下来给出碰到地刺死亡函数trapDeath()和碰到藤曼死亡函数plantDeath()的代码：

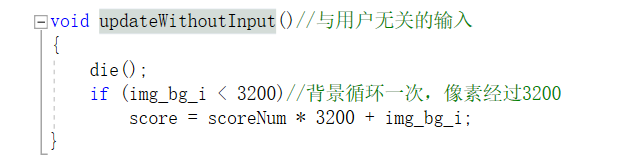




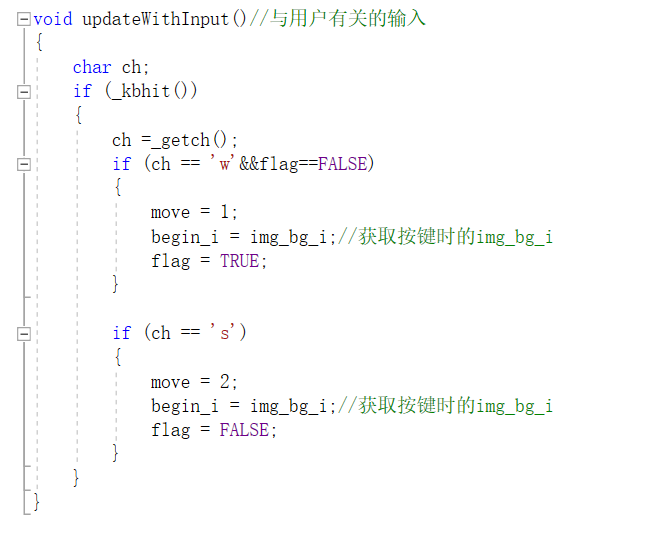
1. 接下来给出判断人物碰到地刺或藤曼时死亡的函数die()的代码：



1. 与show()函数相关的大部分函数已经介绍完毕，接着便是与用户无关的输入函数，其中包括die()函数以及分数变量score的赋值，代码如下：

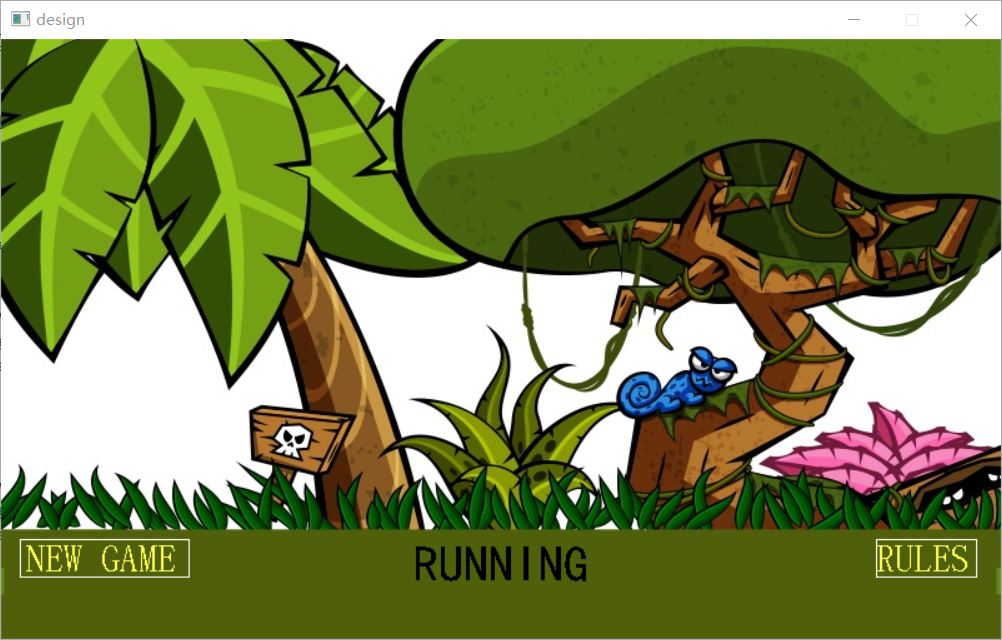


1. 接着是与用户有关的输入函数，即处理用户的键盘输入，利用\_kbhit()函数来判断是否有按键，然后用一个字符变量来得到这个按键值，接着对move变量进行相应的更改，从而达到控制跳跃和滑铲的目的，代码如下：



**效果截图：**

1. 开始界面



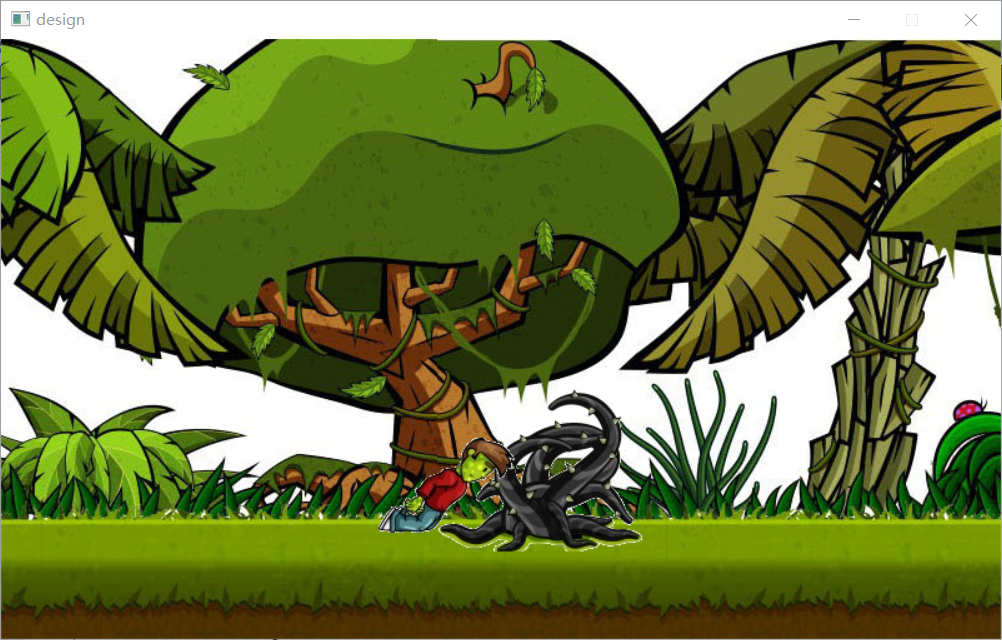
1. 规则界面



1. 游戏界面

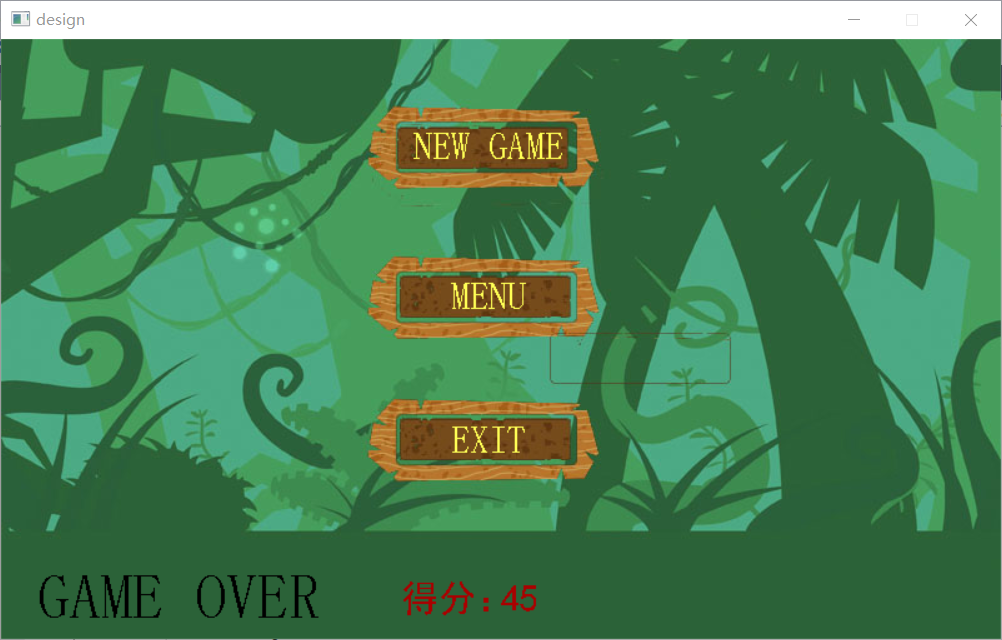


1. 死亡画面





1. 死亡菜单



# 实验总结

1. 本次程序设计内容为一个跑酷游戏，带有背景音乐、背景图片、选择菜单以及死亡动画等诸多游戏要素；
2. 这一次的程序设计进一步培养了我的游戏编程思维，提升了我的编程能力；
3. 在制作这个游戏的过程中，遇到了茫茫多的BUG，比如死亡后的菜单界面鼠标点击无效，按住w键会持续浮在空中等等等等，而在一次次思考与网络查询来解决这些BUG的过程中，我更是收获良多，学习到了很多解决BUG的思路与方法。