

目录

[一、 题目要求 3](#_Toc92757996)

[二、 游戏理念设计 3](#_Toc92757997)

[(一) 游戏开发思路 3](#_Toc92757998)

[(二) 游戏UI设计 5](#_Toc92757999)

[三、 代码构思 5](#_Toc92758000)

[(一) 运行环境说明 5](#_Toc92758001)

[(二) 函数库声明和预加载 5](#_Toc92758002)

[(三) 宏定义 6](#_Toc92758003)

[(四) 结构体创建 6](#_Toc92758004)

[(五) 函数设计 7](#_Toc92758005)

[四、 编写过程中部分解决的问题及方法 8](#_Toc92758006)

[五、 使用到的一些图形库的函数 8](#_Toc92758007)

[六、 待完善的功能 9](#_Toc92758008)

[七、 心得体会 9](#_Toc92758009)

[八、 对课设的建议 10](#_Toc92758010)

## 题目要求

编写一个贪吃蛇游戏，使用图形界面，通过键盘控制蛇的移动

游戏规则为：

1. 游戏开始时，在游戏区域固定位置出现贪吃蛇，随机的在区域内某个位置出现食物。
2. 贪吃蛇会自动沿着其头部的方向移动。
3. 玩家可以做的操作有：键盘控制蛇的上下左右移动。
4. 当蛇遇到食物时，会把当前食物吃掉，并成为玩家的得分，新的食物会随机出现在区域内的某个位置上。
5. 当蛇遇到墙壁时，游戏结束。

加分部分：

1. 可以考虑增加关卡，不同的关卡，难度不同，例如蛇的移动速度增加等等。
2. 可以在游戏区域内增加障碍物，蛇碰到障碍物后游戏结束。
3. 在游戏区域增加障碍物的基础上，可以考虑让蛇在某种条件下能够毁坏障碍物，例如蛇吃掉某种特殊的食物后，具有暂时毁坏障碍物的能力。

## 游戏理念设计

### 游戏开发思路

#### 整体框架

1. 菜单界面

我使用了 EasyX 图形库绘制一个640X400的游戏界面（加入头文件 #include <graphics.h>，而我自己是设计了UI, 然后将这张图片绘制成了界面的背景。在界面上增加了我想要点击的按钮，比如三种模式、开关音乐、退出等等。这里也只是将界面上绘制出我们想要拥有的按键，但是实现功能还在后面。

1. 功能的按键实现

针对界面上的每个按键，我使用了 EasyX 图形库中获取鼠标信息的功能，通过点击按钮，通过判断鼠标在哪一块范围，针对不同的信息然后将其功能实现。比如点击退出游戏，我们再绘制一个新的界面，背景就显示再见。点击音乐的图标, 就会自动播放音乐, 再次点击就会关闭。在小秘密中则包含了游戏提示。

1. 游戏模式的实现

在代码中, 我实现了三种游戏模式, 其实三种关卡模式的逻辑是差不多的，只是我在最基础的普通模式上面增加了一些限制或者解除了一些限制，所以才有了另外两种模式, 比如当鼠标移在普通模式按键的范围后，通过点击左键就开始实现我们的普通模式。

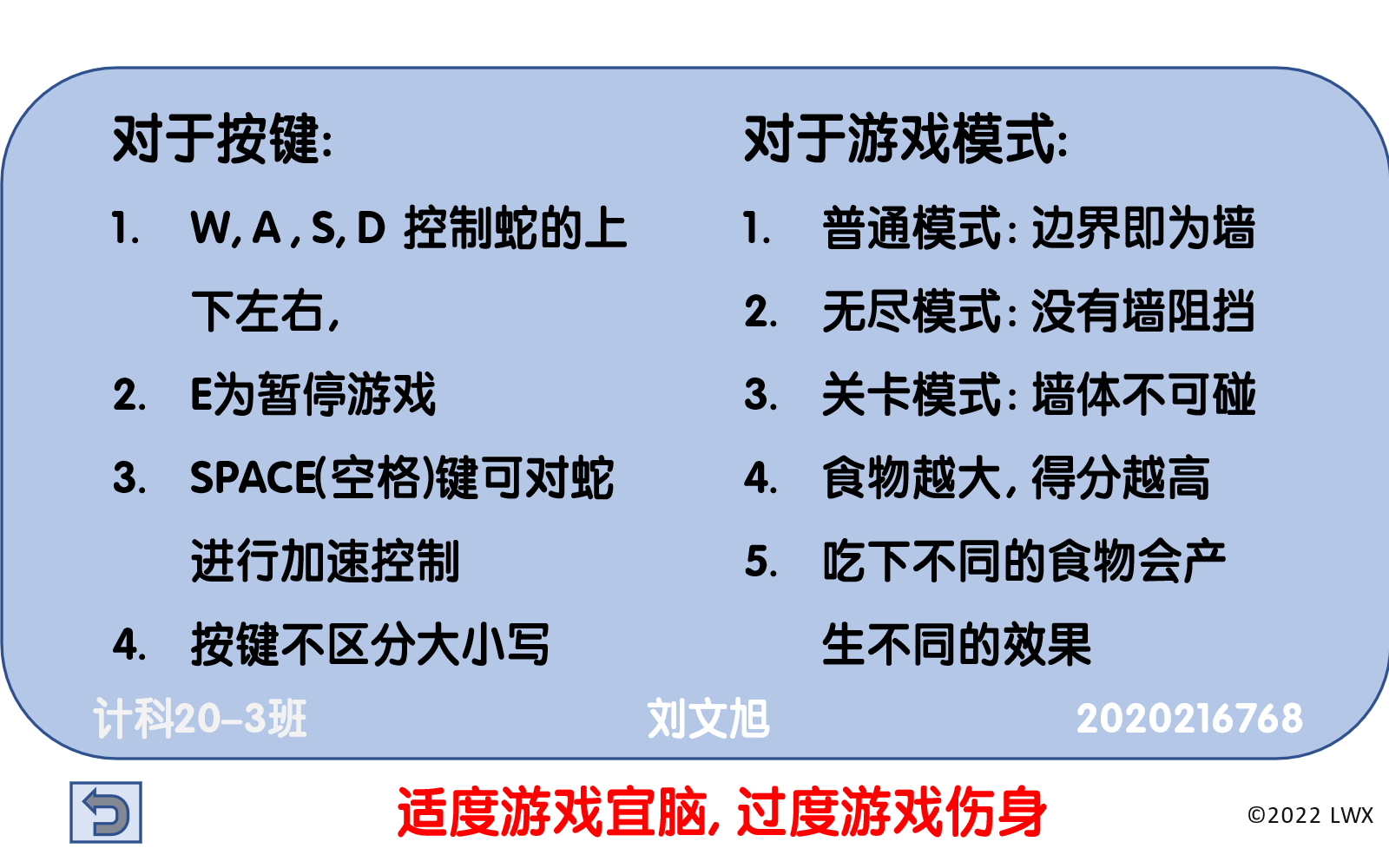
#### 具体措施

在普通模式和无尽模式中首先要重新绘制一个游戏的界面，在初始界面的形成时给出所有的参数，比如蛇的参数（如初始长度、初始坐标、每节的半径、初始方向、蛇的颜色）、食物的参数（如随机产生的食物的坐标、食物的半径、食物的颜色、食物有没有被吃掉的判断标志）。而在关卡模式中, 需要先把墙画出来, 然后和上面一样的步骤把蛇和食物初始化, 特别的是这个模式中蛇不能碰到墙体, 碰到墙体那么蛇就会死亡。

当将这些参数初始化好了之后，就通过 EasyX 图形库的众多功能将这些参数给绘制出来。接下来就是让小蛇动起来。之后为了看到一条在图形内移动的小蛇，需要通过按键控制小蛇的移动使它接近食物。接着要判断有没有吃到食物，如果吃到了，食物的判断标志将变成亦吃掉，并将继续生成新的食物。然后程序将往复的进行循环（绘制此时蛇的参数—>蛇按蛇头的方向移动—>控制蛇改变移动方向—>判断是否吃到食物），直到蛇头碰到蛇身就死亡。蛇死亡后绘制一个新的界面，最终再回到开始菜单。

### 游戏UI设计

#### 图片预设



## 代码构思

### 运行环境说明

/\* Environment : Visual Studio 2019 (Community) \*/

/\* Notice : EasyX [Version: 2021-11-09] \*/

### 函数库声明和预加载

#include <ctime> // 操作时间

#include <conio.h> // 按键交互

#include <stdlib.h> // Sleep等函数

#include <iostream> // 输入输出

#include <graphics.h> // 图形界面

#include <mmstream.h> // 播放音乐

#pragma comment(lib, "winmm.lib") // 多媒体设备接口

using namespace std;

### 宏定义

// 解决VS中对函数内存做边界,内存越界等问题的检查

#define \_CRT\_SECURE\_NO\_WARNINGS

#define WIDTH 640 // 界面宽度

#define HIGHT 400 // 界面高度

#define SNAKE\_MAX 400 // 蛇最大节数

#define WALL\_MAX 10 // 墙最大数量

#define color\_SnakeHead RGB(254, 67, 101) // 蛇头颜色

#define color\_SnakeBody RGB(252, 157, 154) // 蛇身颜色

### 结构体创建

|  |  |  |  |
| --- | --- | --- | --- |
| 用途声明 | 结构体名称 | 成员 | 成员备注 |
| 蛇 | Snake | POINT size[SNAKE\_MAX] | 蛇的最大长度 |
| int num; | 节数 |
| int score | 分数 |
| int dir | 移动朝向 |
| int speed | 速度 |
| 食物 | Food | unsigned long color | 颜色 |
| int x, y, r | 坐标和半径 |
| bool flag | 存在与否的标志 |
| 墙 | Wall | int x1, y1, x2, y2 | 坐标 |
| unsigned long color | 颜色 |

### 函数设计

void menuInit(); // 菜单界面

void secret(); // 小秘密界面

void gameExit(); // 退出游戏

void gameOver(); // 游戏结束

bool isShowGrid(); //游戏界面是否显示网格

void showLenth(int len); // 显示长度

void showScore(int score); // 显示分数

void Eat(); // 吃食物, 得分(取决于半径大小1, 2, 3, 4), 重新随机生成食

void keyControl(); // 按键交互

bool isAcceleration(); // 判断是否加速

bool snakeMove1(); // 普通模式判断是否能移动蛇

bool snakeMove2(); // 无尽模式和关卡模式判断是否能移动蛇

void gameInit1(); // 普通模式和无尽模式初始化

void gameInit2(); // 关卡初始化

void wallDraw(); // 画墙

void lineDraw(); // 画网格线

void gameDraw1(); // 普通模式

void gameDraw2(); // 无尽模式

void gameDraw3(); // 关卡模式

int getScore(); // 获取分数

int getLen(); // 获取长度

## 编写过程中部分解决的问题及方法

1. // 解决VS中对函数内存做边界,内存越界等问题的检查

#define \_CRT\_SECURE\_NO\_WARNINGS

1. 鼠标结构体MOUSEMSG在图中像素位置的确定:

使用画图工具, QQ的截图功能

1. UI的设计

直接使用PPT把UI设计出来, easyx的文字不好控制, 图形比较丑

1. 关卡的设定

直接在画图工具上画出想要的关卡, 但苦于关卡跳转出了BUG, 没能实现这一功能

1. 解决outtextxy文字报错问题

Visual studio需要在项目的高级功能处将字符集改为 ”使用多字节字符集”

1. 解决蛇加速和暂停问题

设置蛇加速的函数, 使用异步键盘的函数, (GetAsyncKeyState(VK\_SPACE) & 0x8000)解决这一问题, 并通过Sleep()函数的不同刷新值来实现这一效果, 而暂停功能就是按键交互实现的

1. 解决无尽模式中蛇穿墙功能

让蛇头瞬间移动到边界的另外一处, 直接坐标交换。就和蛇自身更新一样, 然后就完成了。

1. 解决返回的问题

还是同样用MOUSEMSG, 只不过是在那张图片中再加一个死循环, 直到我点到了那个位置才让它有下一步的操作。

## 使用到的一些图形库的函数

mciSendString(TEXT("open GameMusic.mp3"), NULL, 0, NULL); //播放音乐

mciSendString("play GameMusic.mp3", NULL, 0, NULL); //关闭音乐

GetMouseMsg(); //获取鼠标信息

BeginBatchDraw(); //开始批量绘制

EndBatchDraw(); //结束批量绘制

initgraph(WIDTH, HIGHT); //打开界面

loadimage(&image\_begin, "开始界面.png", WIDTH, HIGHT); //加载图片

putimage(0, 0, &image\_begin); //导入图片

setbkmode(TRANSPARENT); //设置背景模式

setlinestyle(PS\_SOLID, 1, NULL, 0); //设置线的风格

setlinecolor(BLACK); //设置线的颜色

settextstyle(26, 0, \_T("Consolas")); //设置字体的宽+字体的高+字体的风格

settextcolor(RGB(66, 76, 78)); //设置文本颜色

sprintf(s, "%d", score); //输出文本

outtextxy(115, 32, s); // 输出文本

\_kbhit() //判断有无输入

GetAsyncKeyState(VK\_SPACE) // 获取是否有其他按键

setfillcolor(RGB(95, 158, 160)); //设置填充颜色

solidrectangle() //实心矩形

solidcircle() //实心圆

## 待完善的功能

对不同半径的食物做出的反应, 游戏界面中对游戏的设置未完成, 关卡的跳转, 图形界面的优化, 代码书写和注释的规范。吃食物时会闪现一个点、边界问题没有做好、不能存档读档, 没能利用学到的文件的知识设置一个排行榜等等

## 心得体会

虽然贪吃蛇是个简单的小游戏，但是当我第一次写出自己敲出的代码，还是成就感满满。对编程语言知识的理解其实也是一个很好的巩固。第一次自己去主动学习这么多的新语法、完成了第一个工程文件的基本构建，还成功结合了自己会的其他软件（UI设计方面）。总体来说，这次课程设计带给我的提升，不是一言两语能说清的。因为做这个游戏的时候发生了许多困难，所以做的过程中多少也有所感动，猜测了很多未来会发生的事情。希望自己的进步不止于此，而是不断地去发掘更多的潜力，让未来的自己打开这个稚嫩的程序时，真的会露出宠溺的微笑吧。

在整个创作过程中，从零到有的感觉，是由一个个错误修复而来的。在一个灵感突然迸发、并且真的解决了一个错误的时候，那种幸福感确实是难以言喻的。整个课设的基础部分完成之后，其实还是有很多不足要去修复。但是，为了更好的学习，我会选择假期去完善。虽然时间短暂、技术限制，它还有很多不足，但我相信未来的我，会做出一件又一件更好的作品。而这些作品，会让我忘记这些遗憾和不足。因为，我们总是在“无限进步”！

## 对课设的建议

由于时间安排不合理, 没有把这次课程设计做好, 希望老师能平时在群里面对一些简单的东西指导一下, 让我们能够立马有思路和动力去做, 同时也希望在下一届的课程设计中能够有更加有趣的游戏出现。同时我也认为完全可以加大力度，增大难度, 不要只把难度加在加分部分环节，把难度加在作品整体上, 而是让课设的游戏选择部分更让人耳目一新。