# 高级语言程序设计2-2 大作业

南开大学 计算机大类

姓名：陈宇昕

学号：2310675

班级：计卓

2024年5月12日

高级语言程序设计

实验报告

目录

[高级语言程序设计大作业实验报告 1](#_Toc21286)

[一. 作业题目 1](#_Toc29969)

[二. 开发软件 1](#_Toc18364)

[三. 课题要求 1](#_Toc742)

[四. 主要流程 1](#_Toc25240)

[五. 单元测试 3](#_Toc28265)

[六. 收获 4](#_Toc23116)

高级语言程序设计大作业实验报告

1. **作业题目**

基于easyx实现的音游小游戏

1. **开发软件**

Visual Studio 2022

1. **课题要求**
2. 面向对象。
3. 单元测试。
4. 模型部分
5. 验证
6. **主要流程**
   1. **整体流程**

实现思路：

1.定义方块类node，成员方法为

class Node

{

public:

Node() :Node(0, 0, 0xFFFFFF) {};

Node(int x, int y, COLORREF color) :m\_x(x), m\_y(y), m\_color(color) {};

void draw()//画方块

{

setlinecolor(acolor);

setfillcolor(m\_color);

fillrectangle(m\_x, m\_y, m\_x + 20, m\_y + 10);

fillrectangle(m\_x, m\_y, m\_x + 10, m\_y + 10);

fillrectangle(m\_x, m\_y, m\_x - 10, m\_y + 10);

}

void clear()

{

clearrectangle(m\_x, m\_y, m\_x + 20, m\_y + 10);

clearrectangle(m\_x, m\_y, m\_x + 10, m\_y + 10);

clearrectangle(m\_x, m\_y, m\_x - 10, m\_y + 10);

}

//方块位移

void move(int dx, int dy)

{

m\_x += dx;

m\_y += dy;

if (m\_x == -10)m\_x = 490;

if (m\_x == 500)m\_x = 0;

if (m\_y == -10)m\_y = 490;

if (m\_y == 500)

return;

}

int m\_x;

int m\_y;

COLORREF m\_color;

};

2.定义按钮函数button和检测鼠标是否点击的函数inarea

bool inarea(int x, int y, int w, int h, int mx, int my)

{

if (mx > x && mx<x + w && my>y && my < y + h)

{

return true;

}

return false;

}

bool button(int m,int x, int y, int w, int h, const char\* text)

{

setfillcolor(m);

fillroundrect(x, y, x + w, y + h, 5, 5);

int rw = (w - textwidth(text)) / 2;

int rh = (h - textheight(text)) / 2;

outtextxy(x + rw, y + rh, text);

if (inarea(x, y, w, h, msg.x, msg.y) && msg.message == WM\_LBUTTONDOWN)

{

return true;

}

return false;

}

3.实现游戏初始化界面函数gamestart

int gamestart()

{

mciSendString(TEXT("open NULL type waveaudio alias audio"), NULL, 0, NULL);

initgraph(640, 480);

setbkcolor(RGB(45, 204, 51));

setbkmode(TRANSPARENT);

settextcolor(BLACK);

IMAGE img; //准备图片

loadimage(&img, "哆啦A梦.jpg"); //加载图片 L"背景.jpg"是图片的相对路径,也可以使用绝对路径

putimage(0, 0, &img);

while (true)

{

peekmessage(&msg, EM\_MOUSE);

BeginBatchDraw();

if (button(WHITE,350, 120, 150, 35, "game start"))

{

cout << "gamestart";

cleardevice();

EndBatchDraw();

msg = { 0 };

return 1;

}

if (button(WHITE,350, 60, 150, 35, "introduction"))

{

cleardevice();

EndBatchDraw();

msg = { 0 };

return 2;

}

EndBatchDraw();

msg = { 0 };

}

getchar();

}

4.初始化关卡选择界面函数choose（）

int choose()

{

initgraph(640, 480);

setbkcolor(RGB(45, 204, 51));

setbkmode(TRANSPARENT);

settextcolor(BLACK);

IMAGE img; //准备图片

loadimage(&img, "pic.jpg"); //加载图片 L"背景.jpg"是图片的相对路径,也可以使用绝对路径

putimage(0, 0, &img);

while (true)

{

peekmessage(&msg, EM\_MOUSE);

BeginBatchDraw();

if (button(0x55FFFF,150, 180, 150, 35, "two"))

{

cout << "gamestart";

cleardevice();

EndBatchDraw();

msg = { 0 };

return 2;

}

if (button(0x0000AA,150, 60, 150, 35, "one"))

{

cleardevice();

EndBatchDraw();

msg = { 0 };

return 1;

}

if (button(0xFFD700,150, 300, 150, 35, "three"))

{

cleardevice();

EndBatchDraw();

msg = { 0 };

return 3;

}

EndBatchDraw();

msg = { 0 };

}

getchar();

}

5.定义下落按键函数tab（int t,int n）其中参数t代表按键出现的时间参数n代表按键出现的位置

int tab(int t, int n)

{

int p = 0;

char x = 0;

switch (n)

{

case 1:

x = 'A';

p = 10;

break;

case 2:

x = 'D';

p = 213;

break;

case 3:

x = 'J';

p = 426;

break;

case 4:

x = 'L';

p = 620;

break;

}

Node n1(p, 0, acolor);

Node n2(p, -10, WHITE);

int i = 49;

Sleep(t);

while (i--)

{

BeginBatchDraw();

n1.draw();

n1.move(0, 10);

n1.clear();

if (i < 5 && (GetAsyncKeyState(x) & 0x8000))

{

co++;

n1.clear();

return 0;

}

EndBatchDraw();

Sleep(20);

}

return 0;

}

6.分别实现三个关卡game1，game2，game3（由于这段代码比较长，请参照github）这里仅做简单介绍：

通过thread关键字实现了同一时间出现多个键位

通过mciSendString(\_T("open music3.mp3 alias bkmusic"), NULL, 100, NULL);函数导入音乐

通过loadimage(&img, "pic.jpg"); //加载图片 L"背景.jpg"是图片的相对路径,也可以使用绝对路径

putimage(0, 0, &img);导入背景图片

7.游戏结算环节输出得分并关闭游戏

void end()

{

initgraph(500, 500);

IMAGE img; //准备图片

loadimage(&img, "pic.jpg");

putimage(0, 0, &img);

TCHAR s[40];

\_stprintf(s, \_T("你的分数:%d"), co);

outtextxy(50, 50, s);

getchar();

// 关闭窗口

closegraph();

}

8．单独在gamestart界面定义了instruction按钮导向instruction界面介绍游戏的玩法相关

void instruction() {

cleardevice(); // 清除图形

// 初始化说明页面的图形环境

initgraph(500, 500);

IMAGE img;

loadimage(&img, "pic.jpg");

putimage(0, 0, &img);

TCHAR s[40];

settextstyle(24, 0, \_T("宋体"));

\_stprintf(s, \_T("游戏说明：从左到右四个键位为A D J L"));

outtextxy(50, 50, s);

\_stprintf(s, \_T("在滑块落在红线和底部之间点击即可得分"));

outtextxy(50, 90, s);

\_stprintf(s, \_T("一共只做了三首歌"));

outtextxy(50, 130, s);

\_stprintf(s, \_T("因为作者太菜了（）"));

outtextxy(50, 170, s);

\_stprintf(s, \_T("还是希望这个游戏给你带来一点点的快乐"));

outtextxy(50, 210, s);

\_stprintf(s, \_T("剩下的介绍放在readme里面了"));

outtextxy(50, 250, s);

\_stprintf(s, \_T("按任意键返回开始界面"));

outtextxy(50, 290, s);

// 等待用户按下任意键

while (!\_kbhit()) {

Sleep(100);

}

\_getch(); // 清除输入缓冲区中的按键

// 清除图形

cleardevice();

}

9.通过main函数运行游戏

int main() {

initgraph(640, 480);

int a;

bool initialized = false; // 用于标记图形环境是否已经初始化

while (true) {

if (!initialized) {

initgraph(640, 480); // 在第一次循环前初始化图形环境

initialized = true;

}

a = gamestart();

switch (a) {

case 1:

int b;

b = choose();

switch (b) {

case 1:

acolor = 0x0000AA;

game1();

end();

break;

case 2:

acolor = 0x0055AA;

game2();

end();

break;

case 3:

acolor = 0xFFD700;

game3();

end();

break;

}

break;

case 2:

instruction();

break;

}

// 不再重新初始化图形环境

}

}

* 1. **算法或公式**
     1. set和clear函数：easyx自带，用于实现方块的绘制和清除
     2. outtextxy函数：实现在指定地点显示文字或图像
     3. mciSendString函数：用于生成背景音乐
     4. loadimage和putimage函数：用于生成背景图
     5. BeginBatchDraw()和EndBatchDraw()：防止图像卡顿

1. **单元测试**

进行了多次测试，在生成的tab键较密集时出现卡顿现象

除此之外未发现明显问题，需要在后续开发过程中继续调整完善

1. **收获**

**1.类的继承与派生**

**通过node类简化了代码量**

**2．Msg.message指针**

msg.message指针可以检测鼠标的位置和点击，是游戏进行的重要部分

还有键盘指针GetAsyncKeyState用于检测按键，是游戏得分的关键判定

**3．除此之外，通过这次游戏的开发制作过程，我掌握了大量easyx中的绘图和指令相关函数加深了自己对于图形化编程的理解，但囿于水平有限，游戏整体还存在很多缺点，流程简单，玩法单一，画面简陋，我会在后续的开发过程中争取进一步完善**