# 程序源代码

#include <stdio.h>

#include <graphics.h>

#include <conio.h>

#include <time.h>

#include <stdlib.h>

#define WIDTH 740//长

#define HEIGHT 462//宽

#define GAME\_HEIGHT (HEIGHT\*5/6)//385 游戏界面宽

typedef struct

{

int x;

int y;

}Location;

typedef struct node // 流星链表

{

int x, y,s;

int data;

struct node\* next;

}METEOR;

typedef struct wenti //题目链表

{

int num1;//第一个数

int num2;//第二个数

char c;//加减乘除

int result;//结果

struct wenti\* next;

}TITLE;

typedef struct //答题者链表

{

TCHAR name[20];

int score;

}USER;

void Background() //背景

{

IMAGE back;

initgraph(740, 462); //设置图形显示方式

loadimage(&back, \_T("image\\background.jpg")); //设置背景图片

cleardevice();

putimage(0, 0, &back);

setcolor(RED);

setlinestyle(PS\_SOLID, 2);

rectangle(1, 1, 739, 461);

line(0, 385, 740, 385);

line(370, 385, 370, 462);

}

void drawAlpha(IMAGE\* picture, int picture\_x, int picture\_y) //x为载入图片的X坐标，y为Y坐标

{

// 变量初始化

DWORD\* dst = GetImageBuffer(); // GetImageBuffer()函数，用于获取绘图设备的显存指针，EASYX自带

DWORD\* draw = GetImageBuffer();

DWORD\* src = GetImageBuffer(picture); //获取picture的显存指针

int picture\_width = picture->getwidth(); //获取picture的宽度，EASYX自带

int picture\_height = picture->getheight(); //获取picture的高度，EASYX自带

int graphWidth = getwidth(); //获取绘图区的宽度，EASYX自带

int graphHeight = getheight(); //获取绘图区的高度，EASYX自带

int dstX = 0; //在显存里像素的角标

// 实现透明贴图 公式： Cp=αp\*FP+(1-αp)\*BP ， 贝叶斯定理来进行点颜色的概率计算

for (int iy = 0; iy < picture\_height; iy++)

{

for (int ix = 0; ix < picture\_width; ix++)

{

int srcX = ix + iy \* picture\_width; //在显存里像素的角标

int sa = ((src[srcX] & 0xff000000) >> 24); //0xAArrggbb;AA是透明度

int sr = ((src[srcX] & 0xff0000) >> 16); //获取RGB里的R

int sg = ((src[srcX] & 0xff00) >> 8); //G

int sb = src[srcX] & 0xff; //B

if (ix >= 0 && ix <= graphWidth && iy >= 0 && iy <= graphHeight && dstX <= graphWidth \* graphHeight)

{

dstX = (ix + picture\_x) + (iy + picture\_y) \* graphWidth; //在显存里像素的角标

int dr = ((dst[dstX] & 0xff0000) >> 16);

int dg = ((dst[dstX] & 0xff00) >> 8);

int db = dst[dstX] & 0xff;

draw[dstX] = ((sr \* sa / 255 + dr \* (255 - sa) / 255) << 16) //公式： Cp=αp\*FP+(1-αp)\*BP ； αp=sa/255 , FP=sr , BP=dr

| ((sg \* sa / 255 + dg \* (255 - sa) / 255) << 8) //αp=sa/255 , FP=sg , BP=dg

| (sb \* sa / 255 + db \* (255 - sa) / 255); //αp=sa/255 , FP=sb , BP=db

}

}

}

}

void displayButton() //按键

{

IMAGE k1, k2, k3, k4;

loadimage(&k1, \_T("image\\k1.png"));

loadimage(&k2, \_T("image\\k2.png"));

loadimage(&k3, \_T("image\\k3.png"));

loadimage(&k4, \_T("image\\k4.png"));

drawAlpha(&k1,34, GAME\_HEIGHT + 5);

drawAlpha(&k2,118, GAME\_HEIGHT + 5);

drawAlpha(&k3,202, GAME\_HEIGHT + 5);

drawAlpha(&k4,286, GAME\_HEIGHT + 5);

outtextxy(43, GAME\_HEIGHT +55, \_T("开始"));

outtextxy(108, GAME\_HEIGHT + 55, \_T("暂停/继续"));

outtextxy(212, GAME\_HEIGHT + 55, \_T("刷新"));

outtextxy(295, GAME\_HEIGHT + 55, \_T("退出"));

}

TITLE \*CreatTitle()//创建题目

{

int x , c, i;

TITLE\* p, \* head = NULL, \* r = NULL, \* k;

p = (TITLE\*)malloc(sizeof(TITLE));

srand((unsigned)time(NULL));

for (x = 0; x < 5; x++)

{

p = (TITLE\*)malloc(sizeof(TITLE));

p->num1 = rand() % 10;

p->num2 = rand() % 9+1;

if ((p->num1) % (p->num2) != 0)

c = rand() % 3;

else

c = rand() % 4;

if (c == 0)

{

p->c = '+';

p->result = p->num1 + p->num2;

}

else if (c == 1)

{

p->c = '-';

p->result = p->num1 - p->num2;

}

else if (c == 2)

{

p->c = '\*';

p->result = p->num1 \* p->num2;

}

else if (c == 3)

{

p->c = '/';

p->result = p->num1 / p->num2;

}

p->next = NULL;

if (head == NULL)

{

head = p;

r = p;

}

else

{

k = head;

for (i = 0; i < x; i++) //防止答案重复

{

if (k->result == p->result)

break;

else

k = k->next;

}

if (i == x)

{

r->next = p;

r = p;

}

else

x--;

}

}

return head;

}

METEOR \*Creatmeteor(TITLE \*T) //创建流星

{

int x=0,b,c,i,j,k;

int a[10],s[8];

srand((unsigned)time(NULL));

for (j = 0; j < 8; j++)

{

s[j] = rand() % 8;

for (k = 0; k < j; k++)

{

if (s[j] == s[k])

j--;

}

}

METEOR \* p, \* head = NULL, \* r = NULL;

while (x < 8)

{

if (T != NULL) //将答案输入流星链表

{

p = (METEOR\*)malloc(sizeof(METEOR));

p->data = T->result;

p->x = (rand() % 100) - 300;

p->y = 45 + 40 \* s[x];

p->s = (rand() % 2) + 1;

p->next = NULL;

a[x] = p->data;

if (head == NULL)

{

head = p;

r = p;

}

else

{

r->next = p;

r = p;

}

x++;

T = T->next;

}

else //随机数输入流星链表

{

b = rand() % 82;

for (i = 0; i < x; i++)

{

if (a[i] == b) //防止重复

break;

}

if (i == x)

{

p = (METEOR\*)malloc(sizeof(METEOR));

p->data = b;

p->x = (rand() % 100) - 300;

p->y = 45 + 40 \* s[x];

p->s = (rand() % 2) + 1;

p->next = NULL;

r->next = p;

r = p;

a[x] = b;

x++;

}

}

}

return head;

}

int checkButton(int mouseX, int mouseY)//检查按钮

{

if (mouseX >= 34 && mouseX <= 84 && mouseY >= 390 && mouseY <= 440)

return 1;

else if (mouseX >= 118 && mouseX <= 168 && mouseY >= 390 && mouseY <= 440)

return 2;

else if (mouseX >= 202 && mouseX <= 252 && mouseY >= 390 && mouseY <= 440)

return 3;

else if (mouseX >= 286 && mouseX <= 336 && mouseY >= 390 && mouseY <= 440)

return 4;

else

return -1; //没选中

}

void MIANbody(int i,IMAGE \*p,IMAGE \*temp,METEOR \*h,TITLE \*T,USER user)//主体函数

{

int j, a, w,jud,which;

int mouseX; //鼠标位置坐标X

int mouseY;

int flag = 0;

METEOR \*k;

k = h;

MOUSEMSG mmsg; //鼠标消息变量

LOGFONT f;

TCHAR s[3],s1[3],data[3],sco[3];

LOGFONT g;

for (j = i; ; j++)

{

if (T == NULL)

{

while (1)

{

BeginBatchDraw();

getimage(temp, 0, 0, 740, 462);

gettextstyle(&g);

g.lfHeight = 50;

\_tcscpy\_s(g.lfFaceName, \_T("黑体"));

g.lfQuality = ANTIALIASED\_QUALITY;

settextstyle(&g);

setcolor(YELLOW);

outtextxy(120, 180, \_T("答题结束，请刷新或退出"));

EndBatchDraw();

Sleep(10);//暂停毫秒

putimage(0, 0, temp);

if (MouseHit())

{

mmsg = GetMouseMsg();

switch (mmsg.uMsg)

{

case WM\_LBUTTONDOWN:

mouseX = mmsg.x;

mouseY = mmsg.y; //取出鼠标x,y值

//鼠标在按钮区

if (mouseX > 0 && mouseX <= WIDTH / 2 && mouseY > GAME\_HEIGHT&& mouseY <= HEIGHT)

{

flag = checkButton(mouseX, mouseY);

}

if (flag == 3)

{

T = CreatTitle();

h = Creatmeteor(T);

MIANbody(0, p, temp, h, T,user);

}

else if (flag == 4) //退出并存档

{

FILE\* fp; //写入文件

errno\_t err;

err = \_wfopen\_s(&fp, TEXT("user.txt"), TEXT("a"));

fwprintf(fp, TEXT("\n%20s%10d"), user.name, user.score);

fclose(fp);

closegraph();

exit(1);

}

}

}

}

}

if (j > 740)

j = 0;

\_stprintf\_s(s,\_T("%d"), T->num1); //使int型可以正确显示，否则乱码

\_stprintf\_s(s1, \_T("%d"), T->num2);

\_stprintf\_s(sco, \_T("%d"),user.score);

gettextstyle(&f);

f.lfHeight = 20;

\_tcscpy\_s(f.lfFaceName, \_T("宋体"));

f.lfQuality = ANTIALIASED\_QUALITY;

settextstyle(&f);

setcolor(YELLOW);

BeginBatchDraw();

getimage(temp, 0, 0, 740, 462);

for (a = 0; k != NULL; a++)

{

\_stprintf\_s(data, \_T("%d"), k->data);

if(k->x + j \* k->s>=-200&& k->x + j \* k->s<=600)

drawAlpha(p, k->x + j \* k->s, k->y);

outtextxy(k->x + 165 + j \* k->s, k->y + 16, data);

k = k->next;

}

gettextstyle(&g);

g.lfHeight = 30;

\_tcscpy\_s(g.lfFaceName, \_T("黑体"));

g.lfQuality = ANTIALIASED\_QUALITY;

settextstyle(&g);

setcolor(YELLOW);

outtextxy(340, 20, s);

outtextxy(383, 20, T->c);

outtextxy(420, 20, s1);

outtextxy(WIDTH / 2 + 260, GAME\_HEIGHT + 23, sco);

EndBatchDraw();

Sleep(10);//暂停毫秒

putimage(0, 0, temp);

k = h;

if (MouseHit())

{

mmsg = GetMouseMsg();

switch (mmsg.uMsg)

{

case WM\_LBUTTONDOWN:

mouseX = mmsg.x;

mouseY = mmsg.y; //取出鼠标x,y值

//鼠标在按钮区

if (mouseX > 0 && mouseX <= WIDTH / 2 && mouseY > GAME\_HEIGHT&& mouseY <= HEIGHT)

{

flag = checkButton(mouseX, mouseY);

}

//鼠标点击流星

else

{

while(k!=NULL)

{

if (mouseX >= (k->x + j \* k->s) && mouseX <= (k->x + 220 + j \* k->s) && mouseY >= (k->y) && mouseY <= (k->y + 50))

{

if (k->data == T->result)

{

T = T->next;

h = h->next;

user.score += 1;

}

else

user.score -= 1;

break;

}

else

k = k->next;

}

k = h;

}

if (flag == 2) //暂停

{

while (1)

{

gettextstyle(&f);

f.lfHeight = 20;

\_tcscpy\_s(f.lfFaceName, \_T("宋体"));

f.lfQuality = ANTIALIASED\_QUALITY;

f.lfQuality = ANTIALIASED\_QUALITY;

settextstyle(&f);

setcolor(YELLOW);

BeginBatchDraw();

getimage(temp, 0, 0, 740, 462);

for (a = 0; k != NULL; a++)

{

\_stprintf\_s(data, \_T("%d"), k->data);

if (k->x + j \* k->s >= -200 && k->x + j \* k->s <= 600)

drawAlpha(p, k->x + j \* k->s, k->y);

outtextxy(k->x + 165 + j \* k->s, k->y + 16, data);

k = k->next;

}

gettextstyle(&g);

g.lfHeight = 30;

\_tcscpy\_s(g.lfFaceName, \_T("黑体"));

g.lfQuality = ANTIALIASED\_QUALITY;

settextstyle(&g);

setcolor(YELLOW);

outtextxy(340, 20, s);

outtextxy(383, 20, T->c);

outtextxy(420, 20, s1);

outtextxy(WIDTH / 2 + 260, GAME\_HEIGHT + 23, sco);

EndBatchDraw();

Sleep(10);//暂停毫秒

putimage(0, 0, temp);

k = h;

if (MouseHit())

{

mmsg = GetMouseMsg();

switch (mmsg.uMsg)

{

case WM\_LBUTTONDOWN:

mouseX = mmsg.x;

mouseY = mmsg.y; //取出鼠标x,y值

//鼠标在按钮区

if (mouseX > 0 && mouseX <= WIDTH / 2 && mouseY > GAME\_HEIGHT&& mouseY <= HEIGHT)

{

flag = checkButton(mouseX, mouseY);

}

if (flag == 2)

MIANbody(j, p, temp, h, T, user);

else if (flag == 4)

{

FILE\* fp; //写入文件

errno\_t err;

err = \_wfopen\_s(&fp, TEXT("user.txt"), TEXT("a"));

fwprintf(fp, TEXT("\n%20s%10d"), user.name, user.score);

fclose(fp);

closegraph();

exit(1);

}

}

}

}

}

else if (flag == 3) //增加流星，由于界面原因，在本作中其实是刷新题目

{

T = CreatTitle();

h = Creatmeteor(T);

MIANbody(0, p, temp, h, T,user);

}

else if (flag == 4) //退出并存档

{

FILE\* fp; //写入文件

errno\_t err;

err = \_wfopen\_s(&fp, TEXT("user.txt"), TEXT("a"));

fwprintf(fp, TEXT("\n%20s%10d"), user.name, user.score);

fclose(fp);

closegraph();

exit(1);

}

}

}

}

}

void start(USER user)//开始函数

{

IMAGE p,temp;

int x = 0;

METEOR \*h;

TITLE \*T;

loadimage(&p, \_T("image\\meteor.png"));

T=CreatTitle();

h = Creatmeteor(T);

MIANbody(0, &p, &temp, h,T,user);

}

void Mouse(USER user)//鼠标操作

{

int mouseX; //鼠标位置坐标X

int mouseY;

int flag=0;

Location ml = { 0,0 };

MOUSEMSG mmsg; //鼠标消息变量

while (1)

{

//处理鼠标消息

if (MouseHit())

{

mmsg = GetMouseMsg();

switch (mmsg.uMsg)

{

case WM\_LBUTTONDOWN:

mouseX = mmsg.x;

mouseY = mmsg.y; //取出鼠标x,y值

//鼠标在按钮区

if (mouseX > 0 && mouseX <= WIDTH / 2 && mouseY > GAME\_HEIGHT&& mouseY <= HEIGHT)

{

flag = checkButton(mouseX, mouseY);

}

if (flag == 1)

start(user);

else if (flag == 4)

{

closegraph();

exit(1);

}

}

}

}

}

void main() //主函数

{

Background();

displayButton();

USER user;

InputBox(user.name, 20, \_T("请输入用户名："), \_T("捞鱼能手"));

outtextxy(WIDTH / 2 + 30, GAME\_HEIGHT + 30, \_T("Name: "));

outtextxy(WIDTH / 2 + 80, GAME\_HEIGHT + 30, user.name);

outtextxy(WIDTH / 2 + 200, GAME\_HEIGHT + 30, \_T("Score: "));

user.score = 0;

Mouse(user);

}