#include <graphics.h>

#include <conio.h>

#include <stdio.h>

#include <time.h>

#include <mmsystem.h>

#include <math.h>

#pragma comment(lib, "winmm.lib") // 用于播放音效

// 游戏状态

enum GameState {

START\_SCREEN,

PLAYING,

GAME\_OVER

};

// 游戏全局变量

int score = 0;

GameState gameState = START\_SCREEN;

int targetDuration = 3000; // 初始目标持续时间(ms)

clock\_t targetSpawnTime = 0;

POINT targetPos = { -100, -100 }; // 初始目标位置在屏幕外

bool targetActive = false;

// 加载资源

IMAGE bgImage; // 背景图片

IMAGE targetImage; // 目标白点图片

// 初始化游戏

void InitGame() {

// 加载背景图片

loadimage(&bgImage, \_T("background.jpg")); // 请确保有名为background.jpg的图片

// 创建目标白点图片(白色小圆点)

SetWorkingImage(&targetImage);

int size = 40;

targetImage.Resize(size, size);

setfillcolor(WHITE);

solidcircle(size / 2, size / 2, size / 2 - 2);

SetWorkingImage(); // 恢复默认绘图目标

// 初始化随机种子

srand((unsigned)time(NULL));

}

// 生成新目标

void SpawnTarget() {

int width = getwidth();

int height = getheight();

// 随机位置，确保不会太靠近边缘

targetPos.x = rand() % (width - 60) + 30;

targetPos.y = rand() % (height - 100) + 50; // 顶部留出空间给计分板

targetSpawnTime = clock();

targetActive = true;

}

// 播放MP3音效

void PlayMP3(const TCHAR\* filename) {

TCHAR cmd[256];

\_stprintf\_s(cmd, \_T("open \"%s\" type mpegvideo alias mp3"), filename);

mciSendString(\_T("close mp3"), NULL, 0, NULL);

mciSendString(cmd, NULL, 0, NULL);

mciSendString(\_T("play mp3"), NULL, 0, NULL);

}

// 绘制游戏

void DrawGame() {

// 双缓冲绘图

BeginBatchDraw();

// 清屏

cleardevice();

// 绘制背景

putimage(0, 0, &bgImage);

if (gameState == START\_SCREEN) {

// 开始界面

setbkmode(TRANSPARENT);

settextcolor(WHITE);

settextstyle(48, 0, \_T("宋体"));

outtextxy(getwidth() / 2 - 100, getheight() / 2 - 50, \_T("射击游戏"));

settextstyle(24, 0, \_T("宋体"));

outtextxy(getwidth() / 2 - 80, getheight() / 2 + 30, \_T("点击开始游戏"));

}

else if (gameState == PLAYING) {

// 绘制计分板

setbkmode(TRANSPARENT);

settextcolor(WHITE);

settextstyle(24, 0, \_T("宋体"));

TCHAR scoreStr[32];

\_stprintf\_s(scoreStr, \_T("得分: %d"), score);

outtextxy(20, 20, scoreStr);

// 绘制目标

if (targetActive) {

putimage(targetPos.x - targetImage.getwidth() / 2,

targetPos.y - targetImage.getheight() / 2,

&targetImage);

// 绘制剩余时间条

double elapsed = (double)(clock() - targetSpawnTime) / targetDuration;

if (elapsed > 1.0) elapsed = 1.0;

setfillcolor(GREEN);

fillrectangle(20, 60, 20 + (int)((getwidth() - 40) \* (1.0 - elapsed)), 70);

}

}

else if (gameState == GAME\_OVER) {

// 游戏结束界面

setbkmode(TRANSPARENT);

settextcolor(WHITE);

settextstyle(48, 0, \_T("宋体"));

outtextxy(getwidth() / 2 - 100, getheight() / 2 - 50, \_T("游戏结束"));

settextstyle(24, 0, \_T("宋体"));

TCHAR scoreStr[32];

\_stprintf\_s(scoreStr, \_T("最终得分: %d"), score);

outtextxy(getwidth() / 2 - 80, getheight() / 2 + 30, scoreStr);

outtextxy(getwidth() / 2 - 80, getheight() / 2 + 70, \_T("点击重新开始"));

}

// 结束双缓冲绘图

EndBatchDraw();

}

// 处理鼠标点击

void HandleMouseClick(int x, int y) {

// 每次点击都播放点击音效

PlayMP3(\_T("click.mp3")); // 需要有click.mp3文件

if (gameState == START\_SCREEN) {

// 开始游戏

gameState = PLAYING;

score = 0;

targetDuration = 3000; // 重置目标持续时间

SpawnTarget();

}

else if (gameState == PLAYING && targetActive) {

// 检查是否点击了目标

int dx = x - targetPos.x;

int dy = y - targetPos.y;

int distance = (int)sqrt(dx \* dx + dy \* dy);

if (distance <= targetImage.getwidth() / 2) {

// 击中目标

PlayMP3(\_T("hit.mp3")); // 播放MP3音效(需要有hit.mp3文件)

score++;

// 随着分数增加，目标持续时间减少(最低500ms)

targetDuration = max(500, 3000 - score \* 100);

// 生成新目标

SpawnTarget();

}

}

else if (gameState == GAME\_OVER) {

// 重新开始游戏

gameState = START\_SCREEN;

}

}

// 更新游戏状态

void UpdateGame() {

if (gameState == PLAYING && targetActive) {

// 检查目标是否超时

if (clock() - targetSpawnTime > targetDuration) {

targetActive = false;

gameState = GAME\_OVER;

}

}

}

int main() {

// 初始化图形窗口

initgraph(1280, 720);

// 初始化游戏

InitGame();

// 消息循环

ExMessage msg;

while (true) {

// 处理消息

while (peekmessage(&msg, EX\_MOUSE)) {

if (msg.message == WM\_LBUTTONDOWN) {

HandleMouseClick(msg.x, msg.y);

}

}

// 更新游戏状态

UpdateGame();

// 绘制游戏

DrawGame();

// 延迟一下，减少CPU占用

Sleep(10);

}

// 关闭图形窗口

closegraph();

return 0;

}