**程序设计开发实践报告**

**题 目：东方名校幻想乡（Eastern\_Famous\_School\_Fantasyland）**

**序 号： 72**

**学 号： 20232241110**

**姓 名： 刘晨旭**

**任课教师： 姜厚云**

**成 绩：**

大连理工大学软件学院

2024年7月

**目录**

[相关材料（电子报告及项目源码压缩包）的链接： 2](#_Toc172184451)

[1 技术调研报告 3](#_Toc172184452)

[1.1 第一周学习总结 3](#_Toc172184453)

[1.1.1 内容简介 3](#_Toc172184454)

[1.1.2 难点和解决办法 3](#_Toc172184455)

[1.1.3 学习案例 3](#_Toc172184456)

[1.2 第二周学习总结 8](#_Toc172184457)

[1.2.1 内容简介 8](#_Toc172184458)

[1.2.2 难点和解决办法 8](#_Toc172184459)

[1.2.3 学习案例 9](#_Toc172184460)

[2 项目开发报告 16](#_Toc172184461)

[2.1 项目简介 16](#_Toc172184462)

[2.2 需求分析 17](#_Toc172184463)

[2.2.1 功能模块 17](#_Toc172184464)

[2.2.2 需求说明 17](#_Toc172184465)

[2.3 系统设计 18](#_Toc172184466)

[2.3.1 界面设计（UI） 18](#_Toc172184467)

[2.3.2 流程设计 19](#_Toc172184468)

[2.4 系统实现 20](#_Toc172184469)

[2.4.1 实现效果 20](#_Toc172184470)

[2.4.2 核心代码 23](#_Toc172184471)

[2.5 系统测试 71](#_Toc172184472)

[2.6 项目总结 72](#_Toc172184473)

**评分细则及标准**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **考察项目** | **总分** | **评分细则（总分40分）** | **分数** | **得分** |
| **技术调研（第一周）** | 20分 | 调研报告 | 10分 |  |
| 平时表现 | 10分 |
| **技术调研（第二周）** | 20分 | 调研报告 | 10分 |  |
| 平时表现 | 10分 |
| **项目开发评分（总分60分）** | | | |  |
| 问题规模 | 10分 | 创新超额完成指定任务，工作量饱满 | 8-10分 |  |
| 基本完成指定任务，工作量一般 | 6-8分 |
| 指定任务未完成，工作量不足 | 0-5分 |
| 技术难度 | 10分 | 模型设计合理优化，采用合适的方法实现 | 8-10分 |  |
| 模型设计基本正确，采用较合适的方法实现 | 6-8分 |
| 模型设计存在问题，采用方法不合理 | 0-5分 |
| 实现程度 | 20分 | 系统实现完整，界面友好，测试全面无误 | 8-10分 |  |
| 系统实现完整，界面友好，存在少许错误 | 6-8分 |
| 系统实现不完整，界面不友好，存在错误 | 0-5分 |
| 报告质量 | 10分 | 报告完整、格式统一、结构清晰、图表正确 | 8-10分 |  |
| 报告较为规范、结构较为清晰、图表基本正确 | 6-8分 |
| 内容不完整不规范、结构不清晰、图表有错误 | 0-5分 |
| 项目汇报 | 10分 | 阐述清晰准确，回答问题准确到位 | 8-10分 |  |
| 阐述基本准确，回答问题基本准确 | 6-8分 |
| 阐述不够清晰完整，回答问题不准确 | 0-5分 |
| 最终得分 | 100分 |  | | |

相关材料（电子报告及项目源码压缩包）的链接：

源代码链接：[https://github.com/AcidBarium/Eastern\_Famous\_School\_Fantasyland](https://github.com/AcidBarium/Eastern_Famous_School_Fantasyland" \t "_new)

# 1 技术调研报告

## 第一周学习总结

在了解过有关easyx的基础概念和基本操作之后，通过编写棋盘绘制的代码和钟表绘制的代码加深了对于easyx的印象。

### 1.1.1 内容简介

学习了通过initgraph函数绘制绘图窗口；通过setorigin和setaspectratio函数设置坐标原点并改变坐标轴缩放比例以及方向；可以使用预定义常量、RGB 宏、16 进制数字以及 HSL/HSV 转 RGB 转换函数表示颜色。通过 setfillcolor、setlinecolor 等函数设置填充色和线条颜色；一些基本的图形绘制函数如circle、rectangle、line等，其中一些图像函数还具有相应的填充（fill）和无边框（solid）模式；使用 outtextxy 在指定位置输出字符串，使用 drawtext 在指定区域内按格式输出文本，还可以设置文字颜色和样式。

### 1.1.2 难点和解决办法

在绘制棋盘的过程中，需要显示数字。由于 outtextxy 函数的参数只能是字符数组，而数字是 int 类型，因此不能直接显示。原本打算使用 to\_string 函数将 int 转化为 string 再转化为字符数组，但编译器不支持 to\_string 函数。经过查阅资料，采用了 ostringstream 来接受 int 类型的数字，再转化为 string 类型，然后转化为 c\_str 类型，最终通过 outtextxy 函数输出。

在完成绘制时钟的案例后，希望时钟能够实时更新以反映系统时间。为此，在相关函数的外部添加了 while(true) 循环，并在函数结尾处加入 Sleep(1000)，使钟表每秒钟跳动一次。虽然这样实现了根据系统时间跳动，但偶尔会出现闪烁现象。通过查阅相关资料，了解到可以通过双缓冲绘图减少屏幕的直接绘图操作，从而加快绘图速度并消除闪烁。最终，通过该方法成功解决了这一问题。。

### 1.1.3 学习案例

**案例一、绘制棋盘：**

相关核心代码与相关解释：

|  |
| --- |
| initgraph(800, 800, EX\_SHOWCONSOLE);  setbkcolor(RGB(224, 149, 79));  cleardevice();  setorigin(0, 800);     // 将绘图窗口的左下角作为坐标原点  setaspectratio(1, -1); // 缩放因子为负数，可以实现坐标轴的翻转，此行可使y轴向上为正  setlinecolor(BLACK);  setlinestyle(0, 5); |
| 绘制棋盘的背景，设置棋盘的大小，颜色，坐标轴等参数 |

|  |
| --- |
| line(50, 50, 50, 750);  line(50, 50, 750, 50);  line(750, 50, 750, 750);  line(50, 750, 750, 750);  setlinestyle(0, 2);  settextcolor(BLACK);  \_rep(i, 1, 13) line(50 + i \* 50, 50, 50 + i \* 50, 750), line(50, 50 + i \* 50, 750, 50 + i \* 50);  setfillcolor(BLACK);  solidcircle(200, 200, 10);  solidcircle(200, 600, 10);  solidcircle(600, 200, 10);  solidcircle(600, 600, 10);  solidcircle(400, 400, 10); |
| 绘制棋盘的所有线段和棋盘上的圆点  注：\_rep为for循环的宏定义，参考：  #define \_rep(i, a, b) for (int i = (a); i <= (b); ++i)  #define rep(i, a, b) for (int i = (a); i < (b); ++i) |

|  |
| --- |
| settextstyle(34, 17, \_T("Courier"));  setaspectratio(1, 1);  char c = 'A';  \_rep(i, 1, 15)  {      c = 'A' + i - 1;      outtextxy(-10 + i \* 50, -40, c);  }  int cNum = 0;  \_rep(i, 1, 15)  {      cNum = 0 + i;      ostringstream ss;      ss << cNum;      outtextxy(10, -20 - 50 \* i, ss.str().c\_str());  } |
| 输出棋盘上的文字 |

最终效果可见图Figure 1

|  |
| --- |
|  |
| **Figure 1 棋盘案例效果图** |

**案例一、绘制钟表：**

相关核心代码与相关解释：

|  |
| --- |
| initgraph(600, 600, EX\_SHOWCONSOLE);  setbkcolor(WHITE);  cleardevice();  BeginBatchDraw();  while (1)  {      DWORD beginTime = GetTickCount();      setlinestyle(0, 0);      setorigin(300, 300);      setcolor(BLACK);      drawThickCircle(0, 0, 280, 3);      setfillcolor(RGB(210, 210, 210));      fillcircle(0, 0, 275); // 内圆      // 显示里面的线      setlinestyle(0, 0);      \_rep(i, 1, 60) line(265 \* cos(PI / 30 \* i), 265 \* sin(PI / 30 \* i), 273 \* cos(PI / 30 \* i), 273 \* sin(PI / 30 \* i));      setlinestyle(0, 2);      \_rep(i, 1, 12) line(265 \* cos(PI / 6 \* i), 265 \* sin(PI / 6 \* i), 273 \* cos(PI / 6 \* i), 273 \* sin(PI / 6 \* i));      int charRadios = 240;      settextcolor(BLACK);      settextstyle(50, 25, \_T("Courier"));      setbkmode(TRANSPARENT);      int textR = 230;      \_rep(i, 1, 12)      {          ostringstream ss;          ss << i;          outtextxy((int)(textR \* cos(-PI / 2 + PI / 6 \* i)) - 20, (int)(textR \* sin(-PI / 2 + PI / 6 \* i)) - 20, ss.str().c\_str());      }      char rolexLogo[10] = "ROLEX";      int h = textheight(rolexLogo);      int w = textwidth(rolexLogo);      outtextxy(-w / 2, 50 - h / 2, rolexLogo);      setfillcolor(RGB(0, 0, 0));      solidcircle(0, 0, 12); // 黑色表芯      setfillcolor(RGB(210, 0, 0));      solidcircle(0, 0, 6); // 红色表芯      SYSTEMTIME time; // 定义系统时间变量      int second;      // 秒数      int minute;      // 分钟数      int hour;        // 小时数      int shizhenLength = 150;      int fenzhenLength = 250;      int miaozhenLength = 200;      GetLocalTime(&time);   // 获取本地系统时间      second = time.wSecond; // 设置秒数      minute = time.wMinute; // 设置分钟数      hour = time.wHour;     // 设置小时数      setlinestyle(0, 4);      line(0, 0, shizhenLength \* cos(-PI / 2.0 + (hour % 12 + minute / 60.0 + second / 3600.0) \* 2.0 \* PI / 12.0),           shizhenLength \* sin(-PI / 2.0 + (hour % 12 + minute / 60.0 + second / 3600.0) \* 2.0 \* PI / 12.0));                                                               // 画时针      line(0, 0, fenzhenLength \* cos(-PI / 2.0 + (minute + second / 60.0) \* 2.0 \* PI / 60.0), fenzhenLength \* sin(-PI / 2.0 + (minute + second / 60.0) \* 2.0 \* PI / 60.0)); // 画时针      setlinecolor(RED);      line(0, 0, miaozhenLength \* cos(-PI / 2.0 + 2.0 \* PI / 60.0 \* second), miaozhenLength \* sin(-PI / 2.0 + 2.0 \* PI / 60.0 \* second));      FlushBatchDraw();      DWORD endTime = GetTickCount();      DWORD elapsedTime = endTime - beginTime;      if (elapsedTime < 1000 / 1)          Sleep(1000 / 1 - elapsedTime);  } |
| 其中使用了FlushBatchDraw()来进行双缓冲绘图，减少绘图闪烁的情况，使用了drawThickCircle来绘制粗一点的线，但是后来才意识到可以直接通过设置line的style来绘制粗一点的线。最后使用了if (elapsedTime < 1000 / 1)Sleep(1000 / 1 - elapsedTime);来确保每一秒跳动一次 |

**运行效果**见图2、3、4、5

|  |  |
| --- | --- |
|  |  |
| **Figure 2 钟表运行图** | **Figure 3钟表运行图** |
|  |  |
| **Figure 4钟表运行图** | **Figure 5钟表运行图** |

## 第二周学习总结

进一步学习了easyx的相关操作、主要包括图像处理、双缓冲绘图、消息处理等。同时也了解了如何修改窗口的名称和设置弹窗消息、播放音频。最后通过编写画板和小球碰撞的两个小程序加深了学习。

### 1.2.1 内容简介

loadimage从文件中读取图像，可选择拉伸图像到指定宽度和高度；putimage在指定位置显示图像，可以选择显示整个图像或图像的一部分；通过putimage与掩码图和前景图进行透明贴图，TransparentBlt 和 AlphaBlend 函数实现 PNG 图片的透明贴图；使用BeginBatchDraw和EndBatchDraw进行批量绘图，减少闪烁并提高绘图速度；bool peekmessage(ExMessage \*msg, BYTE filter = -1, bool removemsg = true)：从消息缓冲区获取一个消息，并立即返回；通过mciSendString向媒体控制接口（Media Control Interface，MCI）设备发送命令字符串来控制多媒体设备，如音频和视频播放。

### 1.2.2 难点和解决办法

当需要只显示图片主体部分而不需要显示图片背景的时候，使用putimage无法完成这一操作，自定义函数putimage\_alpha便可以完成这一操作。进行消息的接受和后续处理的时候，会出现一些意想不到的状况，根据老师指导，先处理消息，再在后面进行计算，再进行显示，分为三个部分分开进行操作可以使得这一过程更加严谨可靠。

### 1.2.3 学习案例

**案例一、绘图程序：**

|  |
| --- |
| bool running = true;  // 主循环控制参数  ExMessage msg;        // 消息对象  int x = 400;          // 圆心X坐标  int y = 300;          // 圆心Y坐标  int r = 10;           // 圆半径  int edge = 10;        // 正方形边长  bool isyuan = false;  // 是否画圆形  bool issqure = false; // 是否画方形  bool isbiger = false;  bool isred = false;  bool isgreen = false;  bool isblue = false;  bool isClear = false;  initgraph(800, 600); // 初始化绘图窗口  BeginBatchDraw();    // 开启批量绘图 |
| 首先初始化所需的变量，设置窗口的基本参数 |

|  |
| --- |
| // 消息处理  DWORD beginTime = GetTickCount(); // 记录循环开始时间  while (peekmessage(&msg))  {      if (msg.message == WM\_MOUSEMOVE)      { // 圆的位置随鼠标位置变化          x = msg.x;          y = msg.y;      }      if (msg.message == WM\_LBUTTONDOWN)      {          isyuan = true;      }      if (msg.message == WM\_LBUTTONUP)      {          isyuan = false;      }      if (msg.message == WM\_RBUTTONDOWN)      {          issqure = true;      }      if (msg.message == WM\_RBUTTONUP)      {          issqure = false;      }      if (msg.message == WM\_KEYDOWN) // 按下按键处理      {          switch (msg.vkcode)          {          case 'r':          case 'R':              isred = true;              isgreen = false;              isblue = false;              break;          case 'g':          case 'G':              isred = false;              isgreen = true;              isblue = false;              break;          case 'b':          case 'B':              isred = false;              isgreen = false;              isblue = true;              break;          case 'c':          case 'C':              isClear = true;              break;          case VK\_CONTROL:              isbiger = true;              break;          case VK\_ESCAPE:              running = false;              break;          }      }      if (msg.message == WM\_KEYUP) // 松开按键处理      {          switch (msg.vkcode)          {          case VK\_CONTROL:              isbiger = false;              break;          }      }  } |
| 消息的处理与接受 |

|  |
| --- |
| if (isClear)  {      cleardevice();      isClear = false;  }  // 绘图  if (isred)      setfillcolor(RED);  else if (isgreen)      setfillcolor(GREEN);  else if (isblue)      setfillcolor(BLUE);  if (isyuan && !issqure)  {      if (!isbiger)          solidcircle(x, y, r); // 绘制一个圆      else          solidcircle(x, y, 2 \* r);  }  else if (!isyuan && issqure)  {      if (!isbiger)          solidrectangle(x - edge / 2, y - edge / 2, x + edge / 2, y + edge / 2);      else          solidrectangle(x - edge, y - edge, x + edge, y + edge);  }  FlushBatchDraw(); // 刷新批量绘图  DWORD endTime = GetTickCount();          // 记录循环结束时间  DWORD elapsedTime = endTime - beginTime; // 计算循环耗时  if (elapsedTime < 1000 / 120)            // 按每秒60帧进行补时      Sleep(1000 / 120 - elapsedTime); |
| 绘制操作 |

|  |
| --- |
| EndBatchDraw(); // 关闭批量绘图  closegraph();   // 关闭绘图窗口 |
| 结束绘图 |

**运行效果见图6**

|  |
| --- |
|  |
| Figure 6 画图效果展示 |

**案例二、弹球程序：**

|  |
| --- |
| struct BALL  {      int r;      int x;      int y;      int vx;      int vy;  } ball;  struct BLOCK  {      int x;      int y;      int length;      int height;      bool isright;      bool isleft;  } block;  void isGuangdong(BALL &b)  {      if (b.x <= b.r)      {          b.vx \*= -1;          b.x = b.r;      }      else if (b.x + b.r >= 800)      {          b.x = 800 - b.r;          b.vx \*= -1;      }      else if (b.y <= b.r)      {          b.y = b.r;          b.vy \*= -1;      }      else if ((b.y + b.r + block.height >= 600 && b.x >= block.x && b.x <= block.x + block.length))      {          b.y = 600 - b.r - block.height;          b.vy \*= -1;          numScore++;      }  } |
| 构建球结构体，板砖结构体，碰撞检测函数 |

|  |
| --- |
| srand(time(0));  bool running = true; // 主循环控制参数  ExMessage msg;       // 消息对象  string ss = "score : ";  ball.x = 400, ball.y = 300;  ball.r = 10;  ball.vx = 10 + rand() % 10;  ball.vy = 10 + rand() % 10;  block.x = 350, block.y = 590;  block.height = 10, block.length = 100;  block.isleft = false, block.isright = false;  initgraph(800, 600); // 初始化绘图窗口  BeginBatchDraw();    // 开启批量绘图  settextcolor(YELLOW);  settextstyle(20, 0, "微软雅黑");  ll timecnt = 0; |
| 初始化程序所需变量，绘图窗口参数 |

|  |
| --- |
| DWORD beginTime = GetTickCount64(); // 记录循环开始时间  timecnt++;  isGuangdong(ball);  ball.x += ball.vx / 2;  ball.y += ball.vy / 2;  while (peekmessage(&msg))  {      if (msg.message == WM\_KEYDOWN) // 按下按键处理      {          switch (msg.vkcode)          {          case 'a':          case 'A':          case VK\_LEFT:              block.isleft = true;              break;          case 'd':          case 'D':          case VK\_RIGHT:              block.isright = true;              break;          case VK\_ESCAPE:              running = false;              break;          }      }      if (msg.message == WM\_KEYUP) // 松开按键处理      {          switch (msg.vkcode)          {          case 'a':          case 'A':          case VK\_LEFT:              block.isleft = false;              break;          case 'd':          case 'D':          case VK\_RIGHT:              block.isright = false;              break;          }      }  }  if (block.isleft && !block.isright)      block.x -= 10;  else if (block.isright && !block.isleft)      block.x += 10;  if (ball.x > 820 || ball.x < -20 || ball.y < -20 || ball.y > 620)      running = false;  solidcircle(ball.x, ball.y, ball.r);  solidrectangle(block.x, block.y, block.x + block.length, block.y + block.height);  ostringstream stringScore;  stringScore << numScore;  string s;  s = ss + stringScore.str();  int scorewidth = textwidth(s.c\_str());  outtextxy(400 - scorewidth, 50, s.c\_str());  FlushBatchDraw(); // 刷新批量绘图  DWORD endTime = GetTickCount64();        // 记录循环结束时间  DWORD elapsedTime = endTime - beginTime; // 计算循环耗时  if (elapsedTime < 1000 / 60)             // 按每秒60帧进行补时      Sleep(1000 / 60 - elapsedTime);  cleardevice(); |
| 主循环 |

|  |
| --- |
| EndBatchDraw(); // 关闭批量绘图  closegraph(); // 关闭绘图窗口 |
| 结束绘图 |

程序效果见图

|  |
| --- |
|  |
| **Figure 7 弹球程序效果图** |

# 2 项目开发报告

## 2.1 项目简介

东方名校幻想乡(Eastern\_Famous\_School\_Fantasyland)为刘晨旭(刘程序/Acidbarium)与2024年暑期基于visual studio和easyx开发的类东方stg弹幕射击类游戏。项目于7-13号进行了公测，虽然并没有很多人来玩，但还是取得了一些成果，制作者对此已经比较满意了。

游戏发生在一个名叫东方名校幻想乡的世界，在这里各路名校拉帮结派，勾心斗角，共同竞争东方名校之王的称号。在这里，玩家需要操控大连理工大学，躲避敌人子弹的同时消灭敌人收集尽可能多的金币和分数，最终成为东方最强大学。

## 2.2 需求分析

### 2.2.1 功能模块

|  |
| --- |
|  |
| **Figure 8功能模块图** |

### 2.2.2 需求说明

**角色选择：**

* 玩家可以在游戏开始时自由选择角色。

**角色操作：**

* 移动：使用方向键进行角色移动。
* 潜行模式：按住Shift键进入潜行模式。
* 发射子弹：按Z键发射子弹。
* 技能释放：按X键释放技能。

**音乐与音效，在以下情况下将产生不同的音乐和音效：**

* 发射子弹时
* 中弹和死亡时
* 释放技能时
* 不同敌人出现时

**敌人设计：**

* 在不同的时间段会出现不同类型的敌人。
* 每种敌人都有独特的攻击效果和移动模式。

**BOSS战**

* 每一流程的最后会出现BOSS。
* BOSS具有华丽的攻击效果和较高的血量。

**伤害判定：**

* 角色与敌人受到子弹攻击后会减少血量。

**背景移动**

* 背景将产生移动效果，营造出角色在移动的感觉。

**得分机制**

* 拾取掉落物可以增加分数。

**复活机制**

* 当主角的生命值为0时，玩家可以选择复活，消耗50点分数即可复活。

**榜单显示**

* 游戏结束后，将根据玩家的得分显示榜单。

## 2.3 系统设计

### 2.3.1 界面设计（UI）

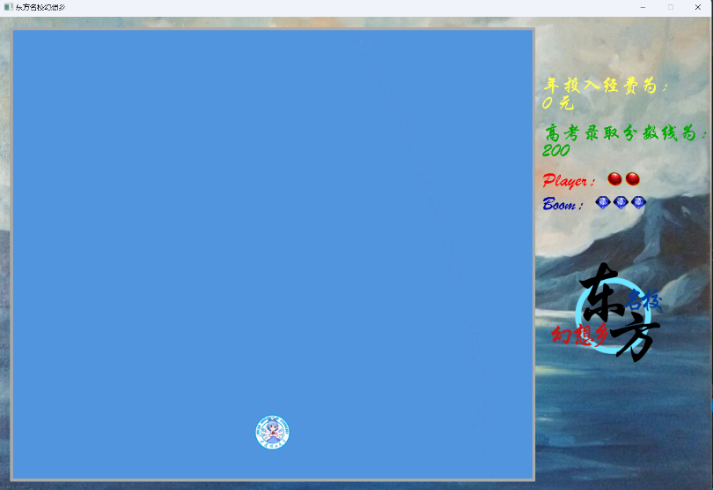
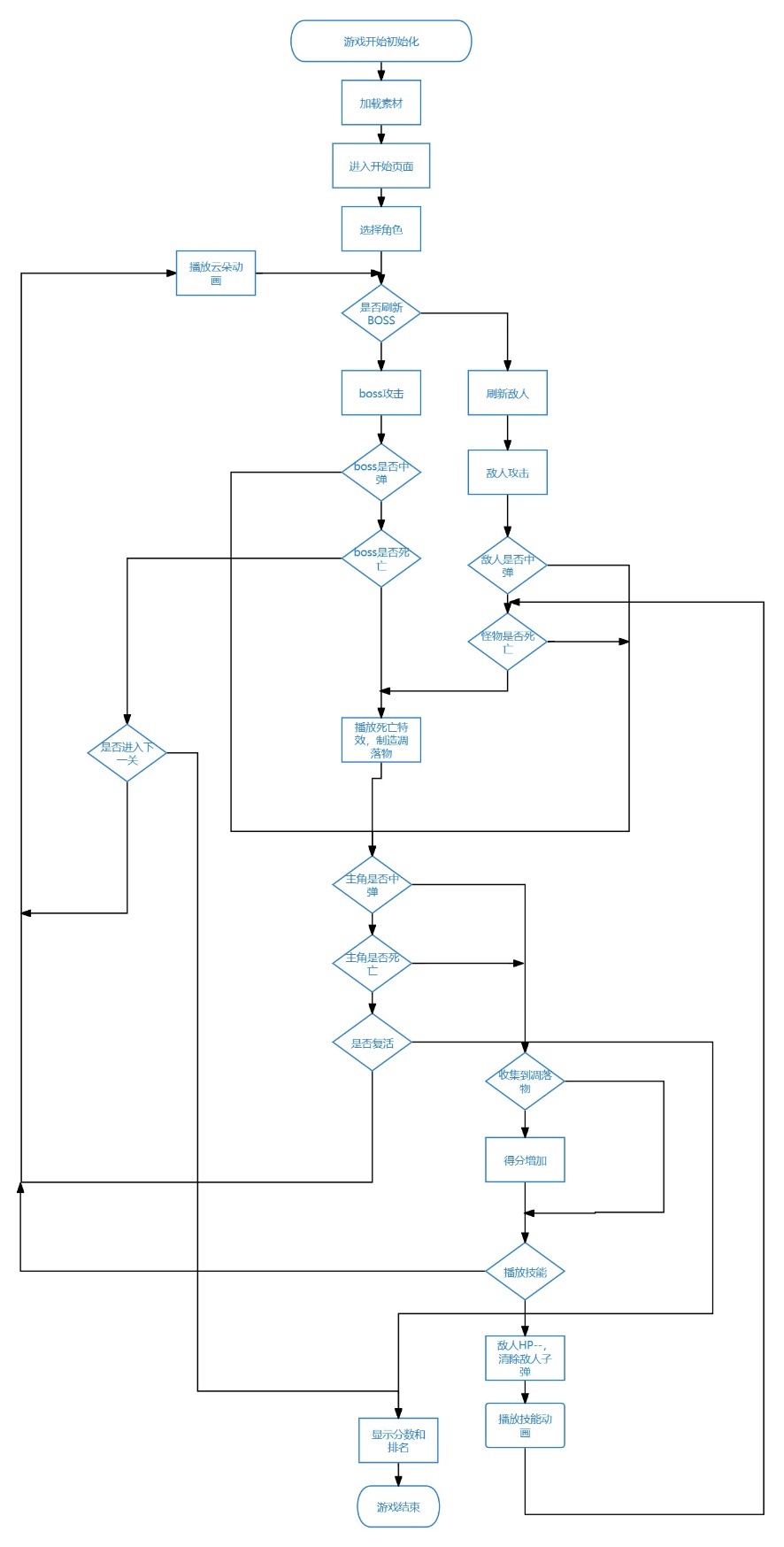


Figure 9游戏界面图

### 2.3.2 流程设计



**Figure 10系统流程图**

## 2.4 系统实现

### 2.4.1 实现效果

|  |
| --- |
|  |
| **Figure 11 游戏开始界面** |

|  |
| --- |
|  |
| **Figure 12 选择角色界面** |

|  |
| --- |
|  |
| **Figure 13 游戏转场画面** |

|  |
| --- |
|  |
| **Figure 14 游戏画面** |

|  |
| --- |
|  |
| **Figure 15 BOSS战** |

|  |
| --- |
|  |
| **Figure 16 第二场BOSS战** |

|  |
| --- |
|  |
| **Figure 17 游戏结算画面** |

### 2.4.2 核心代码

|  |
| --- |
| 显示特殊文字和特殊图片 |
| void outTextCenter(int x, int y, string ss) // 中心显字法  {      int Textwidth = textwidth(ss.c\_str());      int Textheight = textheight(ss.c\_str());      outtextxy(x - Textwidth / 2, y - Textheight / 2, ss.c\_str());  }  // 显示透明图片  void putimage\_alpha(int x, int y, IMAGE \*img)  {      int w = img->getwidth();      int h = img->getheight();      AlphaBlend(GetImageHDC(NULL), x, y, w, h, GetImageHDC(img), 0, 0, w, h, {AC\_SRC\_OVER, 0, 255, AC\_SRC\_ALPHA});  } |

|  |
| --- |
| 背景移动云类 |
| class CLOUD  {      ll cnt;      IMAGE imgShow;      int x;  public:      void chushihua()      {          x = 0;          cnt = 0;      }      void show()      {          cnt++;          if (cnt % 1200 == 0)          {              cnt = 0;              int option = rand() % 3;              if (option == 0)                  imgShow = imgcloud1;              else if (option == 1)                  imgShow = imgcloud2;              else if (option == 2)                  imgShow = imgcloud3;              x = rand() % 380 - 530;          }          putimage\_alpha(x, cnt - 300, &imgShow);          // circle(x, cnt, 5);      }  } cloud; |

|  |
| --- |
| 展示伤害效果类 |
| class shangHaiXiaoguo  {  public:      int x;      int y;      int cnt;      bool isOK; // true 表示这个效果可以用      shangHaiXiaoguo(int xx, int yy)      {          x = xx;          y = yy;          cnt = 0;          isOK = true;      }      void chushihua(int xx, int yy)      {          x = xx;          y = yy;          cnt = 0;          isOK = true;      }      bool show()      {          if (cnt < 25)          {              cnt++;              setlinecolor(RGB(200, 200, 200));              setlinestyle(1, 1);              solidcircle(x, y, 10 + cnt);          }          else              isOK = false;          return isOK;      }  };  vector<shangHaiXiaoguo> hurtXiaoGuoList; |

|  |
| --- |
| 主角子弹类 |
| class mainBULLET  {  public:      int x;      int y;      int v;      int option; // 发射模式，子弹方向  1是单发，2是右上发，3是左上，4是右右上，5是左左上      int r;      // 子弹大小      void chushihua(int xx, int yy, int optionn)      {          r = 20; // 子弹大小          v = 20;          x = xx;          y = yy;          option = optionn;      }      void calshot() // 1是单发，2是右上发，3是左上，4是右右上，5是左左上      {          if (option == 1)              y -= v;          else if (option == 2)          {              x += v \* cos(PI / 18.0 \* 8.0);              y -= v \* sin(PI / 18.0 \* 8.0);          }          else if (option == 3)          {              x -= v \* cos(PI / 18.0 \* 8.0);              y -= v \* sin(PI / 18.0 \* 8.0);          }          else if (option == 4)          {              x += v \* cos(PI / 18 \* 7.0);              y -= v \* sin(PI / 18 \* 7.0);          }          else if (option == 5)          {              x -= v \* cos(PI / 18 \* 7.0);              y -= v \* sin(PI / 18 \* 7.0);          }      }      bool isNeedDelete() // 飞出屏幕则返回1      {          if (x >= 300 + r || x <= -580 - r)              return true;          if (y <= 20 || y >= 780 + r)              return true;          return false;      }      bool pengZhuangJianCe(int againstX, int againstY, int againstR)      {          if (sqrt(1.0 \* (x - againstX) \* (x - againstX) + 1.0 \* (y - againstY) \* (y - againstY)) <= (againstR + r))              return true;          return false;      }      void show()      {          putimage\_alpha(x - r, y - r, &imgMineEnergyBall);      }  };  vector<mainBULLET> mainBulletList; // 主角子弹列表 |

|  |
| --- |
| 主角技能类 |
| class BOOMMMMM  {  public:      int x;      int y;      ll timeCNT;      void chushihua()      {          x = 300;          timeCNT = 120;          y = 20;      }      void calAndMove()      {          timeCNT--;          if (x >= -250)              x += (-24);          else if (x >= -580)              x += (-5);          else              x += (-24);          putimage\_alpha(x, 20, &imgYUANSHEN);      }      bool isNeedDelete()      {          if (timeCNT > 0)              return false;          return true;      }  };  vector<BOOMMMMM> boomList; |

|  |
| --- |
| 主角类 |
| class mainRole // 主角类，DUT  {  private:      ll shootZhen;  public:      int x;      int y;      int vx;      int vy;      int HP;      int Boom;      int GkLuquScore;      ll MoneyEveryYear;      int r;      int moveXmodel;      int moveYmodel;      ll dutGolenBodyCnt; // 金身效果      struct dir      {          bool left;          bool right;          bool up;          bool down;      } direction;      void chushihua() // 初始化      {          shootZhen = 0; // 射击帧          direction.down = false; // 上下左右          direction.up = false;          direction.right = false;          direction.left = false;          x = -140; // 坐标          y = 700;          vx = 7; // 速度          vy = 7;          moveXmodel = 0; // 移动模式 1左，2右，1上，2下, 0不动          moveYmodel = 0; // 移动模式 1左，2右，1上，2下, 0不动          r = 10;          dutGolenBodyCnt = 120; // 人物初始2s的金身，在计算移动里进行增减      }      void pengZhuangJianCe() // 碰撞检测      {          if (x + r >= 300)              x = 300 - r;          if (x - r <= -580)              x = -580 + r;          if (y - r <= 20)              y = 20 + r;          if (y + r >= 780)              y = 780 - r;      }      void claMovemode() // 计算角色位置      {          if (dutGolenBodyCnt)              dutGolenBodyCnt--; // 一直把金身效果减到0停止。          if (direction.up && direction.down)              moveYmodel = 0;          else if (direction.up)              moveYmodel = 1;          else if (direction.down)              moveYmodel = 2;          else              moveYmodel = 0;          if (direction.right && direction.left)              moveXmodel = 0;          else if (direction.left)              moveXmodel = 1;          else if (direction.right)              moveXmodel = 2;          else              moveXmodel = 0;      }      void move(int optionx, int optiony) // 1左，2右，1上，2下, 0不动  //改变xy      {          int vxtemp = vx, vytemp = vy;          if (optionx == 1)              vxtemp \*= -1;          else if (optionx == 2)              vxtemp \*= 1;          else if (optionx == 0)              vxtemp \*= 0;          if (optiony == 1)              vytemp \*= -1;          else if (optiony == 2)              vytemp \*= 1;          else if (optiony == 0)              vytemp \*= 0;          if (optionx && optiony)          {              x += vxtemp / 1.414;              y += vytemp / 1.414;          }          else          {              x += vxtemp;              y += vytemp;          }      }      void show() // 展示角色图像      {          putimage\_alpha(x - 30, y - 30, &imgChar);          // circle(x, y, r);                           //显示判定区      }      void getHert()      {          // 还没做的角色死亡效果          HP--;      }      void useTechnology()      {          if (Boom > 0)          {              mciSendString("play BGMxiaoZhang from 0", 0, 0, 0);              Boom--;              BOOMMMMM temp;              temp.chushihua();              boomList.push\_back(temp);          }      }      void shoot() // 发射子弹      {          if (shootZhen == 0)              mciSendString("play BGMshoot0 from 0", 0, 0, 0); // 播放音效          else if (shootZhen == 5)              mciSendString("play BGMshoot1 from 0", 0, 0, 0);          else if (shootZhen == 10)              mciSendString("play BGMshoot2 from 0", 0, 0, 0);          shootZhen++;          shootZhen %= 15;          if (MoneyEveryYear >= 0 && !(shootZhen % 5))          {              mainBULLET temp;              temp.chushihua(x, y - 20, 1);              mainBulletList.push\_back(temp);          }          if (MoneyEveryYear >= 25000 && !(shootZhen % 5))          {              mainBULLET temp2, temp3;              temp2.chushihua(x, y - 20, 2);              temp3.chushihua(x, y - 20, 3);              mainBulletList.push\_back(temp2);              mainBulletList.push\_back(temp3);          }          if (MoneyEveryYear >= 50000 && !(shootZhen % 5))          {              mainBULLET temp4, temp5;              temp4.chushihua(x, y - 20, 4);              temp5.chushihua(x, y - 20, 5);              mainBulletList.push\_back(temp4);              mainBulletList.push\_back(temp5);          }      }  } DUT; |

|  |
| --- |
| 敌人子弹类 |
| class againstBullet  {  private:      double vx;      double vy;      double xa;      double ya;      IMAGE img;      float xReal;      float yReal;  public:      int x;      int y;      int r;      void chushihua(int xx, int yy, double vxx, double vyy, double xaa, double yaa, IMAGE \*imgg)      {          xReal = xx;          yReal = yy;          x = xx;          y = yy;          vx = vxx;          vy = vyy;          xa = xaa;          ya = yaa;          img = \*imgg;          r = 9;      }      void calMove()      {          vx += xa;          vy += ya;          /\*          if (vx > 0)              x += ceil(vx);          else if (vx < 0)              x += floor(vx);          if (vy > 0)              y += ceil(vy);          else if (vy < 0)              y += floor(vy);\*/          xReal += vx;          yReal += vy;          x = xReal;          y = yReal;      }      bool isNeedDelete() // 飞出屏幕则返回1      {          if (x >= 300 + r || x <= -580 - r)              return true;          if (y <= 20 || y >= 780 + r)              return true;          return false;      }      bool pengZhuangJianCe(int againstX, int againstY, int againstR)      {          if (sqrt(1.0 \* (x - againstX) \* (x - againstX) + 1.0 \* (y - againstY) \* (y - againstY)) <= (againstR + r))              return true;          return false;      }      void show()      {          putimage\_alpha(x - r, y - r, &img);      }  };  vector<againstBullet> againstBulletList; |

|  |
| --- |
| 敌人类 |
| class againstRolesmall  {  private:      int dieTimeCnt;      int HP;      double vx;      double vy;      double xa; // 加速度      double ya;      IMAGE imgRole;      ll saveTimeCnt;  public:      int x;      int y;      int r;      ll isShoot; // 判断小怪是否发射子弹,放在速度队列中，每帧加加，对某数取余来判断是否发射子弹      int shotModel;      void chushihua(int xx, int yy, int Vx, int Vy, double xaa, double yaa, IMAGE \*img, ll savetimecnt = llmax, int shotOption = 1, int hpp = 2) // 敌方小兵初始化      {          HP = hpp;          r = 22;          x = xx;          y = yy;          vx = Vx;          vy = Vy;          xa = xaa;          ya = yaa;          imgRole = \*img;          dieTimeCnt = 0;          isShoot = 0;          shotModel = shotOption;          saveTimeCnt = savetimecnt;      }      void calMove()      {          saveTimeCnt--;          if (saveTimeCnt >= 0)          {              vx += xa;              vy += ya;              x += vx;              y += vy;              isShoot++;              if (isShoot >= 1000000000)                  isShoot = 0;          }      }      bool isNeedDelet() // 飞出屏幕则返回1      {          if (x >= 300 + r || x <= -580 - r)              return true;          if (y <= 20 || y >= 780 + r)              return true;          return false;      }      // 小怪的话，目前只想做自瞄机，随机放子弹，加速度子弹      void shoot(int option) // 非常多的花里胡哨的射击模式，小怪的      {          if (option == 1) // 自瞄机          {              againstBullet temp;              int xCha = DUT.x - x;              int yCha = DUT.y - y;              double vxTemp = 2.0 \* xCha / sqrt(1.0 \* xCha \* xCha + 1.0 \* yCha \* yCha);              double vyTemp = 2.0 \* yCha / sqrt(1.0 \* xCha \* xCha + 1.0 \* yCha \* yCha);              temp.chushihua(x, y, vxTemp, vyTemp, 0, 0, &imgBlueBullet); // 蓝色自瞄子弹              againstBulletList.push\_back(temp);                          // 将自瞄子弹放到子弹列表中          }          else if (option == 2) // 随机          {              againstBullet temp[5];              rep(i, 0, 5)              {                  int xxxRand = rand() % 5 - 2;                  int yyyRand = rand() % 5 - 1;                  xxxRand = xxxRand ? xxxRand : 1;                  yyyRand = yyyRand ? yyyRand : 1;                  temp[i].chushihua(x, y, xxxRand, yyyRand, 0, 0, &imgRedBullet);                  againstBulletList.push\_back(temp[i]);              }          }          else if (option == 3)          {              againstBullet temp;              int xCha = DUT.x - x;              int yCha = DUT.y - y;              double vxTemp = 0.5 \* xCha / sqrt(1.0 \* xCha \* xCha + 1.0 \* yCha \* yCha);              double vyTemp = 0.5 \* yCha / sqrt(1.0 \* xCha \* xCha + 1.0 \* yCha \* yCha);              temp.chushihua(x, y, -vxTemp, -vyTemp, 0.15 \* vxTemp, 0.15 \* vyTemp, &imgPurpleBullet); // 蓝色自瞄子弹              againstBulletList.push\_back(temp);                                                      // 将自瞄子弹放到子弹列表中          }      }      bool gethurt() // 返回true直接打死了      {          HP--;          if (HP <= 0)              return true;          return false;      }      void show()      {          putimage\_alpha(x - r, y - r, &imgRole);      }      bool dieShow() // 死透了返回true      {          setlinestyle(1, 1);          if (dieTimeCnt < 10)          {              circle(x, y, r + dieTimeCnt);              return false;          }          return true;      }  };  vector<againstRolesmall> againstRoleLiveList; |

|  |
| --- |
| BOSS类 |
| class BOSS  {  private:      ll oriHP;      int x;      int y;      int r;      double vy;      double ax;      double ay;      float xReal;      float yReal;      IMAGE imgRole;      ll cnt;      ll shootcnt;      int optionSevenCnt;      int optionEightCnt;  public:      ll HP;      double vx;      void chushihua(int xx, int yy, IMAGE \*img)      {          optionSevenCnt = 0;          optionEightCnt = 0;          shootcnt = 0;          cnt = 0;          vx = 2;          r = 40;          x = xx;          y = yy;          xReal = x;          yReal = y;          HP = 1500;          oriHP = HP;          imgRole = \*img;      }      bool pengzhuangjiance(int againstX, int againstY, int againstR)      {          if ((x - againstX) \* (x - againstX) + (y - againstY) \* (y - againstY) <= (againstR + r) \* (againstR + r))          {              HP--;              return true;          }          return false;      }      void shot(int option) // 1转圈，2类激光，3随便射击,4 同时一大圈,5 自瞄狙，6散弹枪      {          shootcnt++;          if (option == 1)          {              optionEightCnt = 0;              double vvvv = 4;              againstBullet temp, temp2;              double theta = 2.0 \* PI / 60.0 \* (cnt % 60);              temp.chushihua(x + 1.0 \* r \* cos(theta), y + 1.0 \* r \* sin(theta), vvvv \* cos(theta), vvvv \* sin(theta), 0, 0, &imgRedBullet);              temp2.chushihua(x - 1.0 \* r \* cos(theta), y - 1.0 \* r \* sin(theta), -vvvv \* cos(theta), -vvvv \* sin(theta), 0, 0, &imgRedBullet);              againstBulletList.push\_back(temp);              againstBulletList.push\_back(temp2);          }          else if (option == 2)          {              double vvvv = 3;              // 参数设定，每秒射击乘以 转一圈的时间，为cnt的取模数。              againstBullet temp1;              againstBullet temp2;              againstBullet temp3;              againstBullet temp4;              double theta = 2.0 \* PI / 90.0 \* (cnt % 90);              temp1.chushihua(x + 1.0 \* r \* cos(theta), y + 1.0 \* r \* sin(theta), vvvv \* cos(theta), vvvv \* sin(theta), 0, 0, &imgRedBullet);              againstBulletList.push\_back(temp1);              temp2.chushihua(x + 1.0 \* r \* cos(theta + PI / 2.0), y + 1.0 \* r \* sin(theta + PI / 2.0), vvvv \* cos(theta + PI / 2.0), vvvv \* sin(theta + PI / 2.0), 0, 0, &imgBlueBullet);              againstBulletList.push\_back(temp2);              temp3.chushihua(x + 1.0 \* r \* cos(theta + PI), y + 1.0 \* r \* sin(theta + PI), vvvv \* cos(theta + PI), vvvv \* sin(theta + PI), 0, 0, &imgPurpleBullet);              againstBulletList.push\_back(temp3);              temp4.chushihua(x + 1.0 \* r \* cos(theta + 3.0 \* PI / 2.0), y + 1.0 \* r \* sin(theta + 3.0 \* PI / 2.0), vvvv \* cos(theta + 3.0 \* PI / 2.0), vvvv \* sin(theta + 3.0 \* PI / 2.0), 0, 0, &imgYellowBullet);              againstBulletList.push\_back(temp4);          }          else if (option == 3)          {              againstBullet temp1;              againstBullet temp2;              int rand1 = (rand() % 7 - 3);              int rand2 = (rand() % 7 - 3);              int rand3 = (rand() % 7 - 3);              int rand4 = (rand() % 7 - 3);              rand1 = rand1 ? rand1 : 1;              rand2 = rand2 ? rand2 : -1;              rand3 = rand3 ? rand3 : -1;              rand4 = rand4 ? rand4 : -1;              temp1.chushihua(x, y, rand1, rand2, 0, 0, &imgRedBullet);              temp2.chushihua(x, y, rand3, rand4, 0, 0, &imgRedBullet);              againstBulletList.push\_back(temp1);              againstBulletList.push\_back(temp2);          }          else if (option == 4)          {              double vv = 4;              double thetaEv = 2 \* PI / 30;              IMAGE imgbullettemp;              againstBullet temp[30];              if (shootcnt % 4 == 0)                  imgbullettemp = imgRedBullet;              else if (shootcnt % 4 == 1)                  imgbullettemp = imgBlueBullet;              else if (shootcnt % 4 == 2)                  imgbullettemp = imgPurpleBullet;              else if (shootcnt % 4 == 3)                  imgbullettemp = imgYellowBullet;              rep(i, 0, 30)              {                  temp[i].chushihua(x + 1.0 \* r \* cos(thetaEv \* i), y + 1.0 \* r \* sin(thetaEv \* i), vv \* cos(thetaEv \* i), vv \* sin(thetaEv \* i), 0, 0, &imgbullettemp);                  againstBulletList.push\_back(temp[i]);              }          }          else if (option == 5)          {              double vv = 7;              double xcha = DUT.x - x;              double ycha = DUT.y - y;              againstBullet temp;              temp.chushihua(x, y, vv \* xcha / sqrt(xcha \* xcha + ycha \* ycha), vv \* ycha / sqrt(xcha \* xcha + ycha \* ycha), 0, 0, &imgBlueBullet);              againstBulletList.push\_back(temp);          }          else if (option == 6)          {              double ershidu = PI / 180.0 \* 20;              againstBullet temp[3];              temp[0].chushihua(x, y + r, 0, 5, 0, 0, &imgYellowBullet);              temp[1].chushihua(x + r \* sin(ershidu), y + r \* cos(ershidu), 7 \* sin(ershidu), 7 \* cos(ershidu), 0, 0, &imgYellowBullet);              temp[2].chushihua(x - r \* sin(ershidu), y + r \* cos(ershidu), -7 \* sin(ershidu), 7 \* cos(ershidu), 0, 0, &imgYellowBullet);              rep(i, 0, 3) againstBulletList.push\_back(temp[i]);          }          else if (option == 7) // 两个半圆          {              optionSevenCnt++;              double yidu = PI / 180.0;              againstBullet temp[95];              if (optionSevenCnt % 2)                  rep(i, 0, 95) temp[i].chushihua(-140 + 100 \* cos(yidu \* i), 30 + 100 \* sin(yidu \* i), 6 \* cos(yidu \* i), 6 \* sin(yidu \* i), 0, 0, &imgBlueBullet);              else                  rep(i, 0, 95) temp[i].chushihua(-140 + 100 \* cos(PI - yidu \* i), 30 + 100 \* sin(PI - yidu \* i), 6 \* cos(PI - yidu \* i), 6 \* sin(PI - yidu \* i), 0, 0, &imgRedBullet);              rep(i, 0, 95) againstBulletList.push\_back(temp[i]);          }          else if (option == 8)          {              double yidu = PI / 180.0;              optionEightCnt++;              againstBullet temp1, temp2;              temp1.chushihua(x + r \* cos(yidu \* optionEightCnt), y + r \* sin(yidu \* optionEightCnt), 7 \* cos(yidu \* optionEightCnt), 7 \* sin(yidu \* optionEightCnt), 0, 0, &imgYellowBullet);              temp2.chushihua(x + r \* cos(PI - yidu \* optionEightCnt), y + r \* sin(PI - yidu \* optionEightCnt), 7 \* cos(PI - yidu \* optionEightCnt), 7 \* sin(PI - yidu \* optionEightCnt), 0, 0, &imgYellowBullet);              againstBulletList.push\_back(temp1);              againstBulletList.push\_back(temp2);              optionEightCnt %= 80;          }      }      void calAndMoveAndShow(int option = 1)      {          cnt++;          ax = 0.0002 \* (-140 - x); // 弹簧振子的加速度和位移成正比          if (option == 1)              vx += ax; // 计算速度          else          {              if (x > -140)                  vx = -1;              else if (x < -140)                  vx = 1;              else if (x == -140)                  vx = 0;          }          xReal += vx; // 位移          yReal += vy;          x = xReal;          y = yReal;          if (y > 100)              vy = -1; // 使得敌人保持在100那条线上          if (y < 100)              vy = 1;          if (y == 100)              vy = 0;          putimage\_alpha(x - r, y - r, &imgRole); // 绘制          setlinecolor(RGB(200, 200, 200)); // 绘制boss血条          setlinestyle(0, 5);          line(-500, 25, 220, 25);          setlinecolor(RGB(255, 0, 0)); // 绘制boss血条          setlinestyle(0, 5);          line(-500, 25, -500.0 + 720.0 / oriHP \* HP, 25);      }  };  vector<BOSS> bossLiveList; |

|  |
| --- |
| 游戏资源加载函数 |
| void loadingManythings() // 图片等文件的缓冲加载区域  {      // 图片等文件的缓冲加载区域（开头）      loadimage(&imgStart, \_T("image\\start.jpg"));                           // 游戏开头图片      loadimage(&imgDUTbigger, \_T("image\\DUTbigger.png"));                   // 大连理工大学大头图片      loadimage(&imgDUTcartoon, \_T("image\\DUTqiLuNuo.png"));                 // 大连理工大学笨蛋9版      loadimage(&imgChosePartBackground, \_T("image\\blueSkyBackground.png")); // 选择英雄阶段的背景图片      loadimage(&imgTheGirlIsPraying, \_T("image\\theGirlIsPraying.jpg"));     // 少女折寿中的图片      loadimage(&imgLeftGame, \_T("image\\GameBackground1.jpg"));              // 战斗区域背景      loadimage(&imgleftGame2, \_T("image\\GameBackground2.jpg"));             // 第二张战斗区域的背景      loadimage(&imgBehindBK, \_T("image\\GameBehindBackground.jpg"));         // 后景，和信息栏一起的      loadimage(&LOGO, \_T("image\\logo.png"));                                // logo图片      loadimage(&imgHP, \_T("image\\redHP.png"));                              // 红宝石血量照片      loadimage(&imgBoom, \_T("image\\blueBoom.png"));                         // 蓝宝石技能条      loadimage(&imgChar1, \_T("image\\DUTchar1.png"));                        // 正常角色图像      loadimage(&imgChar2, \_T("image\\DUTchar2.png"));                        // 琪露诺角色图像      loadimage(&imgMineEnergyBall, \_T("image\\energyBall.png"));             // 主角的能量球      loadimage(&imgQingHuaBoss, \_T("image\\qingHuaBigBoss.png"));            // 清华大boss      loadimage(&imgBeiDaBoss, \_T("image\\BeiJingDaXueBigBoss.png"));         // 北大大boss      loadimage(&imgWuHanDaXueSmall, \_T("image\\smallWuHanDaXue.png")); // 小武汉大学      loadimage(&imgBeiJingHangKong, \_T("image\\BeiJingHangKong.png")); // 北京航空      loadimage(&imgBeiJingLiGong, \_T("image\\BeiJingLiGong.png"));      loadimage(&imgBeiJingShiFan, \_T("image\\BeiJingShiFan.png"));      loadimage(&imgDaLianHaiShi, \_T("image\\DaLianHaiShi.png"));      loadimage(&imgDianZiKeJi, \_T("image\\DianZiKeJi.png"));      loadimage(&imgDongBei, \_T("image\\DongBei.png"));      loadimage(&imgDongNan, \_T("image\\DongNan.png"));      loadimage(&imgFuDan, \_T("image\\FuDan.png"));      loadimage(&imgHaErBinGongCheng, \_T("image\\HaErBinGongCheng.png"));      loadimage(&imgHaErBinGongYe, \_T("image\\HaErBinGongYe.png"));      loadimage(&imgHuaKe, \_T("image\\HuaKe.png"));      loadimage(&imgHuaNanLiGong, \_T("image\\HuaNanLiGong.png"));      loadimage(&imgHuNanDaXue, \_T("image\\HuNanDaXue.png"));      loadimage(&imgJiLin, \_T("image\\JiLin.png"));      loadimage(&imgNanJing, \_T("image\\NanJing.png"));      loadimage(&imgNanKai, \_T("image\\NanKai.png"));      loadimage(&imgShanDong, \_T("image\\ShanDong.png"));      loadimage(&imgShangJiao, \_T("image\\ShangJiao.png"));      loadimage(&imgSiChuan, \_T("image\\SiChuan.png"));      loadimage(&imgTianJin, \_T("image\\TianJin.png"));      loadimage(&imgTongJi, \_T("image\\TongJi.png"));      loadimage(&imgXiaMen, \_T("image\\XiaMen.png"));      loadimage(&imgXiAnDianZiKeJi, \_T("image\\XiAnDianZiKeJi.png"));      loadimage(&imgXiAnJiaoTong, \_T("image\\XiAnJiaoTong.png"));      loadimage(&imgxiBeiGongYe, \_T("image\\xiBeiGongYe.png"));      loadimage(&imgZheJiang, \_T("image\\ZheJiang.png"));      loadimage(&imgZhengZhou, \_T("image\\ZhengZhou.png"));      loadimage(&imgZhongGuoRenMin, \_T("image\\ZhongGuoRenMin.png"));      loadimage(&imgZhongKe, \_T("image\\ZhongKe.png"));      loadimage(&imgZhongNan, \_T("image\\ZhongNan.png"));      loadimage(&imgZhongShan, \_T("image\\ZhongShan.png"));      loadimage(&imgcloud1, \_T("image\\cloud1.png")); // 背景白云      loadimage(&imgcloud2, \_T("image\\cloud2.png"));      loadimage(&imgcloud3, \_T("image\\cloud3.png"));      loadimage(&imgBlueBullet, \_T("image\\blueBullet.png"));     // 敌人的小蓝色子弹      loadimage(&imgRedBullet, \_T("image\\redBullet.png"));       // 红色子弹      loadimage(&imgPurpleBullet, \_T("image\\purpleBullet.png")); // 紫色子弹      loadimage(&imgYellowBullet, \_T("image\\yellowBullet.png")); // 黄色子弹      loadimage(&imgMoney, \_T("image\\money.png"));               // 钱的照片      loadimage(&imgPoint, \_T("image\\point.png"));               // 分数的图片      loadimage(&imgHPpoint, \_T("image\\pointHP.png"));           // 小的可以吃的血包      loadimage(&imgBoomPoint, \_T("image\\BoomPoint.png"));       // 小的可以吃的技能条      loadimage(&imgYUANSHEN, \_T("image\\ThisISDUT.png"));        // 元神      mciSendString("open audio\\arms01\\8bit\_laser1.mp3 alias BGM2", 0, 0, 0);               // 那个biubiu的选择用音效      mciSendString("open audio\\arms01\\8bit\_shoot1.mp3 alias BGMshoot0", 0, 0, 0);          // 主角子弹发送音效      mciSendString("open audio\\arms01\\8bit\_shoot1.mp3 alias BGMshoot1", 0, 0, 0);          // 主角子弹发送音效      mciSendString("open audio\\arms01\\8bit\_shoot1.mp3 alias BGMshoot2", 0, 0, 0);          // 主角子弹发送音效      mciSendString("open audio\\arms01\\chorus\_of\_angels1.mp3 alias BGMxiaoZhang", 0, 0, 0); // 校长的bgm      mciSendString("open audio\\arms01\\destruction1.mp3 alias BGMdie", 0, 0, 0);            // 角色生命值用完的音效      mciSendString("open audio\\arms01\\encounting.mp3 alias BGMboss", 0, 0, 0);             // boss放技能的音效      mciSendString("open audio\\arms01\\lightning.mp3 alias BGMbossOut", 0, 0, 0);           // boss出场的音效      mciSendString("setaudio BGMshoot0 volume to 150", NULL, 0, NULL); // 主角子弹发送音效   (但是调整音量）      mciSendString("setaudio BGMshoot1 volume to 150", NULL, 0, NULL); // 主角子弹发送音效      mciSendString("setaudio BGMshoot2 volume to 150", NULL, 0, NULL); // 主角子弹发送音效      // 图片等文件的缓冲加载区域（结尾）  } |

|  |
| --- |
| 凋落物类 |
| class dropout  {  private:      int vy;      int ay;  public:      int x;      int y;      int option; // 凋落物类型，1代表分数，2代表钱,3血包，3技能包      void chushihua(int xx, int yy)      {          x = xx;          y = yy;          vy = -15;          ay = 1;          option = rand() % 300;          if (option == 98)              option = 3;          else if (option == 99)              option = 4;          else          {              if (option % 2)                  option = 1;              else                  option = 2;          }      }      void MoveAndShow() // 一次计算位置并且播放图片      {          vy += ay;          if (vy >= 3)              vy = 3;          y += vy;          if (option == 2)              putimage\_alpha(x - 10, y - 10, &imgMoney);          else if (option == 1)              putimage\_alpha(x - 10, y - 10, &imgPoint);          else if (option == 3)              putimage\_alpha(x - 10, y - 10, &imgHPpoint);          else if (option == 4)              putimage\_alpha(x - 10, y - 10, &imgBoomPoint);      }      bool isNeedDelete()      {          if (y >= 780 + 10)              return true;          return false;      }      bool pengZhuangJianCe(int xx, int yy)      {          if (xx >= x - 25 && xx <= x + 25 && yy <= y + 25 && yy >= y - 25)              return true;          return false;      }  };  vector<dropout> dropOutThingList; |

|  |
| --- |
| 是否复活函数 |
| bool isGoOn()  {      bool GameMidlleIsGoOn = true;      bool isShang = true;      int temp = 200;      while (GameMidlleIsGoOn)      {          DWORD beginTime = GetTickCount64();          settextcolor(RGB(255, 255, 255));          settextstyle(40, 20, "黑体");          string shiFouJixu = "是否复活? 即使这会导致学校分数减少";          string shi = "是";          string fou = "否";          outTextCenter(0, 270, shiFouJixu);          outTextCenter(0, 340, shi);          outTextCenter(0, 400, fou);          while (peekmessage(&msg))          {              if (msg.message == WM\_KEYDOWN) // 按下按键处理，按下回车键下一步, 左右键选择              {                  switch (msg.vkcode)                  {                  case VK\_RETURN:                      GameMidlleIsGoOn = false;                      mciSendString("play BGM2 from 0", 0, 0, 0);                      break;                  case VK\_UP:                      isShang = true;                      mciSendString("play BGM2 from 0", 0, 0, 0);                      break;                  case VK\_DOWN:                      isShang = false;                      mciSendString("play BGM2 from 0", 0, 0, 0);                      break;                  }              }          }          setlinestyle(0, 0);          if (isShang)          {              setlinecolor(RGB(81, 149, 222));              rectangle(-25, 375, 25, 425);              setlinecolor(WHITE);              rectangle(-25, 315, 25, 365);          }          else          {              setlinecolor(RGB(81, 149, 222));              rectangle(-25, 315, 25, 365);              setlinecolor(WHITE);              rectangle(-25, 375, 25, 425);          }          DWORD endTime = GetTickCount64(); // 锁帧，60帧          DWORD elapsedTime = endTime - beginTime;          if (elapsedTime < 1000 / FRAME)              Sleep(1000 / FRAME - elapsedTime);          FlushBatchDraw();          Sleep(16);      }      return isShang;  } |

|  |
| --- |
| 游戏第一关代码 |
| void GamePart1()  {      DWORD partoneBeginTime = GetTickCount64(); // 记录partOne开始的时间      bool isShot = false;      bool isBoom = false;      bool isShift = false;      ll boomToboomTime = 9999;      ll partOneCnt = 0;      int gongjiYuWAng = 200;      bool isBossOut = true;      int bossMoveModel = 1;      int bossShotModel = 1;      int bossPassCnt = 0;      ll isoverCnt = llmax;      while (running)      {          partOneCnt++;          boomToboomTime++;          DWORD beginTime = GetTickCount64();          // 消息处理          while (peekmessage(&msg))          {              if (msg.message == WM\_KEYDOWN) // 按下按键处理              {                  switch (msg.vkcode)                  {                  case VK\_SHIFT:                      isShift = true;                      break;                  case VK\_UP:                      DUT.direction.up = true;                      break;                  case VK\_DOWN:                      DUT.direction.down = true;                      break;                  case VK\_LEFT:                      DUT.direction.left = true;                      break;                  case VK\_RIGHT:                      DUT.direction.right = true;                      break;                  case 'Z':                  case 'z':                      isShot = true;                      break;                  case 'X':                  case 'x':                      if (boomToboomTime >= 120)                      {                          isBoom = true;                      }                      break;                  }              }              if (msg.message == WM\_KEYUP) // 松开按键处理              {                  switch (msg.vkcode)                  {                  case VK\_SHIFT:                      isShift = false;                      break;                  case VK\_UP:                      DUT.direction.up = false;                      break;                  case VK\_DOWN:                      DUT.direction.down = false;                      break;                  case VK\_LEFT:                      DUT.direction.left = false;                      break;                  case VK\_RIGHT:                      DUT.direction.right = false;                      break;                  case 'Z':                  case 'z':                      isShot = false;                      break;                  case 'X':                  case 'x':                      isBoom = false;                      break;                  }              }          }          if (isShift)          {              DUT.vx = 3;              DUT.vy = 3;              DUT.r = 5;          }          else          {              DUT.vx = 7;              DUT.vy = 7;              DUT.r = 10;          }          DUT.move(DUT.moveXmodel, DUT.moveYmodel); // 位置计算          DUT.claMovemode();                        // 角色运动模式计算          putimage(-screenWidth / 2, 0, &imgBehindBK);       // 后面的大图          putimage(-screenWidth / 2 + 20, 20, &imgLeftGame); // 左边游戏区域图          setlinecolor(RGB(170, 170, 170));                  // 游戏边框          setlinestyle(0, 5);          rectangle(-screenWidth / 2 + 20, 20, 300, 780);          string stringLuQuScore = "高考录取分数线为：";          string stringHP = "Player：";          string stringBoom = "Boom：";          string stringMoneyEveryYear = "年投入经费为：";          ostringstream stringTempToNumber;          stringTempToNumber << DUT.MoneyEveryYear << " 元";          settextstyle(35, 0, "华文行楷");          settextcolor(YELLOW);          outtextxy(315, 100, stringMoneyEveryYear.c\_str()); // 最上面的钱          outtextxy(315, 130, stringTempToNumber.str().c\_str());          settextcolor(GREEN);          outtextxy(315, 180, stringLuQuScore.c\_str()); // 分数线          stringTempToNumber.str("");          stringTempToNumber << DUT.GkLuquScore;          outtextxy(315, 210, stringTempToNumber.str().c\_str());          settextcolor(RGB(255, 0, 0));          outtextxy(315, 260, stringHP.c\_str()); // 显示血量          \_rep(i, 1, DUT.HP) putimage\_alpha(285 + textwidth(stringHP.c\_str()) + i \* 30, 260, &imgHP);          settextcolor(BLUE);          outtextxy(315, 300, stringBoom.c\_str()); // 显示技能数          \_rep(i, 1, DUT.Boom) putimage\_alpha(285 + textwidth(stringBoom.c\_str()) + i \* 30, 300, &imgBoom);          putimage\_alpha(315, 400, &LOGO); // 显示LOGO          cloud.show(); // 显示白云          DUT.pengZhuangJianCe(); // 角色碰撞检测          if (isShot)              DUT.shoot(); // 发射子弹，创建子弹，放入mainbulletList中          DUT.show();      // 角色位置移动显示          if (isBoom && boomToboomTime >= 120)          {              boomToboomTime = 0;              DUT.useTechnology();              if (!bossLiveList.empty())                  bossLiveList.begin()->HP -= 100;          }          DWORD partOneNowTime = GetTickCount64();          DWORD partOnePassTime = partOneNowTime - partoneBeginTime; // 计算游戏开始的时间          // 玩法区域          if (partOnePassTime > 2000 && partOnePassTime <= 20000)          {              gongjiYuWAng = 300;              if (!(partOneCnt % 90))              {                  int rand1 = rand() % 800 - 550;                  againstRolesmall temp;                  temp.chushihua(rand1, 40, 0, 5, 0, -0.03, &imgDaLianHaiShi, rand() % 30 + 90, 1); // 4只一起                  againstRoleLiveList.push\_back(temp);                  againstRolesmall temp2;                  temp2.chushihua(rand1 + 20, 40, 0, 5, 0, -0.03, &imgDongBei, rand() % 30 + 180, 1);                  againstRoleLiveList.push\_back(temp2);                  againstRolesmall temp3;                  temp3.chushihua(rand1 - 20, 40, 0, 5, 0, -0.03, &imgJiLin, rand() % 30 + 270, 1);                  againstRoleLiveList.push\_back(temp3);                  againstRolesmall temp4;                  temp4.chushihua(rand1 + 10, 30, 0, 5, 0, -0.03, &imgHaErBinGongCheng, rand() % 30 + 360, 1);                  againstRoleLiveList.push\_back(temp4);              }          }          else if (partOnePassTime <= 2000)          {          }          else if (partOnePassTime <= 32000)          {              gongjiYuWAng = 200;              if (!(partOneCnt % 30)) // 左下右上              {                  againstRolesmall temp;                  temp.chushihua(rand() % 800 - 550, 30, 5, 5, -0.1, -0.10, &imgDongNan, llmax, 1);                  againstRoleLiveList.push\_back(temp);                  againstRolesmall temp2;                  temp2.chushihua(rand() % 800 - 550, 30, -5, 5, 0.1, -0.10, &imgHaErBinGongCheng, llmax, 1);                  againstRoleLiveList.push\_back(temp2);              }          }          else if (partOnePassTime <= 48000)          {              gongjiYuWAng = 100;              if (!(partOneCnt % 25)) // 快速飞过，散弹              {                  againstRolesmall temp;                  temp.chushihua(-580, rand() % 300 + 20, 5, 0, -0.02, -0.03, &imgHaErBinGongYe, 120, 2);                  againstRoleLiveList.push\_back(temp);                  againstRolesmall temp2;                  temp2.chushihua(300, rand() % 300 + 20, -5, 0, 0.02, -0.03, &imgHuaDongShiFan, 120, 2);                  againstRoleLiveList.push\_back(temp2);              }          }          else if (partOnePassTime <= 72000)          {              gongjiYuWAng = 200;              if (!(partOneCnt % 50)) // 三只一起              {                  int rand1 = rand() % 800 - 550;                  againstRolesmall temp;                  temp.chushihua(rand1, 40, 0, 5, 0, -0.03, &imgDaLianHaiShi, rand() % 30 + 90, 1);                  againstRoleLiveList.push\_back(temp);                  againstRolesmall temp2;                  temp2.chushihua(rand1 + 20, 40, 0, 5, 0, -0.03, &imgDongBei, rand() % 30 + 180, 1);                  againstRoleLiveList.push\_back(temp2);                  againstRolesmall temp3;                  temp3.chushihua(rand1 - 20, 40, 0, 5, 0, -0.03, &imgJiLin, rand() % 30 + 270, 1);                  againstRoleLiveList.push\_back(temp3);              }          }          else if (partOnePassTime <= 84000)          {              gongjiYuWAng = 150;              if (!(partOneCnt % 30)) // 左下右上              {                  againstRolesmall temp;                  temp.chushihua(rand() % 800 - 550, 30, 5, 5, -0.1, -0.10, &imgNanJing, llmax, 1);                  againstRoleLiveList.push\_back(temp);                  againstRolesmall temp2;                  temp2.chushihua(rand() % 800 - 550, 30, -5, 5, 0.1, -0.10, &imgNanKai, llmax, 1);                  againstRoleLiveList.push\_back(temp2);              }          }          else if (partOnePassTime <= 96000)          {              gongjiYuWAng = 100;              if (!(partOneCnt % 20)) // 刷怪频率         直上直下              {                  againstRolesmall temp;                  temp.chushihua(-550, 800, 0, -8, 0, 0, &imgWuHanDaXueSmall, llmax, 1);                  againstRoleLiveList.push\_back(temp);                  againstRolesmall temp1;                  temp1.chushihua(270, 800, 0, -8, 0, 0, &imgBeiJingHangKong, llmax, 1);                  againstRoleLiveList.push\_back(temp1);              }          }          else if (partOnePassTime <= 108000)          {              gongjiYuWAng = 80;              if (!(partOneCnt % 30)) // 0.5s刷一个怪    双桥              {                  againstRolesmall temp;                  temp.chushihua(-580, 220, 7, -5, 0, 0.09, &imgBeiJingLiGong, llmax, 2);                  againstRoleLiveList.push\_back(temp);                  againstRolesmall temp1;                  temp1.chushihua(300, 220, -7, -5, 0, 0.09, &imgBeiJingShiFan, llmax, 2);                  againstRoleLiveList.push\_back(temp1);              }          }          else if (partOnePassTime <= 120000)          {              gongjiYuWAng = 100;              if (!(partOneCnt % 20)) // 刷怪频率       直上直下              {                  againstRolesmall temp;                  temp.chushihua(-550, 30, 0, 8, 0, 0, &imgShanDong, llmax, 1);                  againstRoleLiveList.push\_back(temp);                  againstRolesmall temp1;                  temp1.chushihua(270, 30, 0, 8, 0, 0, &imgShangJiao, llmax, 1);                  againstRoleLiveList.push\_back(temp1);              }          }          else if (partOnePassTime <= 145000)          {              gongjiYuWAng = 150;              if (!(partOneCnt % 30)) // 右下右上              {                  againstRolesmall temp;                  temp.chushihua(rand() % 800 - 550, 30, 5, 7, 0, -0.1, &imgNanJing, llmax, 2);                  againstRoleLiveList.push\_back(temp);                  againstRolesmall temp2;                  temp2.chushihua(rand() % 800 - 550, 30, -5, 7, 0, -0.1, &imgNanKai, llmax, 2);                  againstRoleLiveList.push\_back(temp2);                  againstRolesmall temp3;                  temp3.chushihua(rand() % 800 - 550, 30, rand() % 5, 7, 0.1 \* (rand() % 3), -0.1, &imgNanKai, llmax, 2);                  againstRoleLiveList.push\_back(temp3);              }          }          else if (partOnePassTime <= 155000)          {              gongjiYuWAng = 120;              if (!(partOneCnt % 20)) // 两右一左              {                  againstRolesmall temp;                  temp.chushihua(-577, 60, 7, 0, 0, 0, &imgFuDan, llmax, 3);                  againstRoleLiveList.push\_back(temp);                  againstRolesmall temp2;                  temp2.chushihua(290, 150, -7, 0, 0, 0, &imgHuaKe, llmax, 3);                  againstRoleLiveList.push\_back(temp2);                  againstRolesmall temp3;                  temp3.chushihua(-577, 270, 7, 0, 0, 0, &imgSiChuan, llmax, 3);                  againstRoleLiveList.push\_back(temp3);              }          }          else if (partOnePassTime <= 165000)          {              gongjiYuWAng = 120;              if (!(partOneCnt % 20)) // 两左一右              {                  againstRolesmall temp;                  temp.chushihua(290, 60, -7, 0, 0, 0, &imgTianJin, llmax, 1);                  againstRoleLiveList.push\_back(temp);                  againstRolesmall temp2;                  temp2.chushihua(-577, 150, 7, 0, 0, 0, &imgTongJi, llmax, 1);                  againstRoleLiveList.push\_back(temp2);                  againstRolesmall temp3;                  temp3.chushihua(290, 270, -7, 0, 0, 0, &imgXiaMen, llmax, 1);                  againstRoleLiveList.push\_back(temp3);              }          }          else if (partOnePassTime <= 175000)          {              gongjiYuWAng = 90;              if (!(partOneCnt % 20)) // 两右一左              {                  againstRolesmall temp;                  temp.chushihua(-577, 60, 7, 0, 0, 0, &imgXiAnDianZiKeJi, llmax, 2);                  againstRoleLiveList.push\_back(temp);                  againstRolesmall temp2;                  temp2.chushihua(290, 150, -7, 0, 0, 0, &imgXiAnJiaoTong, llmax, 2);                  againstRoleLiveList.push\_back(temp2);                  againstRolesmall temp3;                  temp3.chushihua(-577, 270, 7, 0, 0, 0, &imgxiBeiGongYe, llmax, 2);                  againstRoleLiveList.push\_back(temp3);              }          }          else if (partOnePassTime <= 180000)          {          }          else if (partOnePassTime <= 200000) // 200000          {              if (isBossOut)              {                  BGMpartOne(0);                  BGMpartOneBoss(1);                  mciSendString("play BGMbossOut from 0", 0, 0, 0);                  isBossOut = false;                  BOSS temp;                  temp.chushihua(-400, 0, &imgBeiDaBoss);                  bossLiveList.push\_back(temp);              }              if (!(partOneCnt % 3))                  if (!bossLiveList.empty())                      bossLiveList.begin()->shot(3); // 乱射          }          else          {              bossPassCnt++;              if (!(partOneCnt % 900))              {                  bossShotModel = (bossShotModel + 1) % 6;                  // bossShotModel = 4;                  bossPassCnt = 0;                  mciSendString("play BGMboss from 0", 0, 0, 0);              }              if (bossShotModel == 0)              {                  if (!(partOneCnt % 3))                      if (!bossLiveList.empty())                          bossLiveList.begin()->shot(1); // 双射              }              else if (bossShotModel == 1)              {                  if (!(partOneCnt % 30) && !bossLiveList.empty()) // 大圈                  {                      bossLiveList.begin()->shot(4);                      if (!(partOneCnt % 120))                          bossLiveList.begin()->shot(5); // 自瞄                  }              }              else if (bossShotModel == 2)              {                  if (!(partOneCnt % 2)) // 四射                      if (!bossLiveList.empty())                          bossLiveList.begin()->shot(2);              }              else if (bossShotModel == 3)              {                  if (bossPassCnt <= 600)                  {                      if (!(partOneCnt % 3) && !bossLiveList.empty())                          bossLiveList.begin()->shot(1); // 双射                  }                  else                  {                      if (!(partOneCnt % 9) && !bossLiveList.empty())                          bossLiveList.begin()->shot(6); // 三发                  }              }              else if (bossShotModel == 4)              {                  if (!(partOneCnt % 3) && !bossLiveList.empty())                      bossLiveList.begin()->shot(3); // 乱射              }              else if (bossShotModel == 5)              {                  if (bossPassCnt <= 450)                  {                      if (!(partOneCnt % 3) && !bossLiveList.empty())                          bossLiveList.begin()->shot(3); // 乱射                  }                  else                  {                      if (!(partOneCnt % 3) && !bossLiveList.empty())                          bossLiveList.begin()->shot(1); // 双射                  }              }          }          if (!bossLiveList.empty())              for (auto it = bossLiveList.begin(); it != bossLiveList.end();)              {                  if (it->HP > 0)                  {                      for (auto it2 = mainBulletList.begin(); it2 != mainBulletList.end();)                      {                          if (it->pengzhuangjiance(it2->x, it2->y, it2->r))                          {                              shangHaiXiaoguo temp2(it2->x, it2->y);                              hurtXiaoGuoList.push\_back(temp2);                              it2 = mainBulletList.erase(it2);                          }                          else                              it2++;                      }                      it->calAndMoveAndShow(bossMoveModel);                      it++;                  }                  else                  {                      it = bossLiveList.erase(it);                      BGMpartOneBoss(0);                      DUT.GkLuquScore += 150;                      for (auto it3 = againstBulletList.begin(); it3 != againstBulletList.end();) // 把子弹清了                      {                          it3 = againstBulletList.erase(it3);                      }                      isoverCnt = 180; // 把boss干死了就直接第一关结束                  }              }          isoverCnt--;          if (isoverCnt < 0)              return;          if (!againstRoleLiveList.empty())              for (auto it = againstRoleLiveList.begin(); it != againstRoleLiveList.end();)              {                  bool isMingZhong = false;                  if (!mainBulletList.empty())                  {                      for (auto it2 = mainBulletList.begin(); it2 != mainBulletList.end();)                      {                          if (it2->pengZhuangJianCe(it->x, it->y, it->r))                          {                              isMingZhong = true;                              if (it->gethurt()) // 打死了                              {                                  dropout dropTemp; // 生成凋落物                                  dropTemp.chushihua(it->x, it->y);                                  dropOutThingList.push\_back(dropTemp);                                  shangHaiXiaoguo temp((it->x + it2->x) / 2, (it->y + it->y) / 2);                                  hurtXiaoGuoList.push\_back(temp);                                  // againstRoleDIEList.push\_back(\*it);                                  it = againstRoleLiveList.erase(it);                              }                              it2 = mainBulletList.erase(it2);                              break;                          }                          else                              it2++;                      }                  }                  if (!isMingZhong)                      it++;              }          // 敌方子弹攻击效果          if (!againstBulletList.empty() && !DUT.dutGolenBodyCnt) // 如果主角没有金身效果就攻击              for (auto it = againstBulletList.begin(); it != againstBulletList.end();)              {                  bool isMingZhong = false;                  if (it->pengZhuangJianCe(DUT.x, DUT.y, DUT.r))                  {                      shangHaiXiaoguo temp(DUT.x, DUT.y);                      hurtXiaoGuoList.push\_back(temp);                      DUT.getHert();                      DUT.chushihua();                      it = againstBulletList.erase(it);                  }                  else                      it++;              }          if (!againstRoleLiveList.empty())              for (auto it = againstRoleLiveList.begin(); it != againstRoleLiveList.end();) // 当前存活的怪物列表              {                  if (it->isNeedDelet()) // 如果活着的越界了，就把它从活队中清除                  {                      it = againstRoleLiveList.erase(it);                  }                  else                  {                      int isShootRand = rand() % gongjiYuWAng;                      if (isShootRand == 1)                          it->shoot(it->shotModel);                      it->calMove();                      it->show();                      it++;                  }              }          if (!hurtXiaoGuoList.empty())              for (auto it = hurtXiaoGuoList.begin(); it != hurtXiaoGuoList.end();)              {                  if (it->show())                  {                      it++;                  }                  else                  {                      it = hurtXiaoGuoList.erase(it);                  }              }          // 凋落物移动，以及掉出检测，以及碰撞检测          if (!dropOutThingList.empty())              for (auto it = dropOutThingList.begin(); it != dropOutThingList.end();)              {                  if (it->isNeedDelete())                  {                      it = dropOutThingList.erase(it);                  }                  else if (it->pengZhuangJianCe(DUT.x, DUT.y))                  {                      if (it->option == 1)                          DUT.GkLuquScore++;                      else if (it->option == 2)                          DUT.MoneyEveryYear += 1000;                      else if (it->option == 3)                          DUT.HP++;                      else if (it->option == 4)                          DUT.Boom++;                      it = dropOutThingList.erase(it);                  }                  else                  {                      it->MoveAndShow();                      it++;                  }              }          // if(!againstRoleDIEList.empty())          // for (auto it = againstRoleDIEList.begin(); it != againstRoleDIEList.end();)                //正在播放去世动画的怪物列表          //{          //  if (it->dieShow())     //死透了会返回true          //  {          //      it = againstRoleDIEList.erase(it);          //死透了就把它删了          //  }          //  else          //  {          //      it++;          //  }          // }          if (!mainBulletList.empty())              for (auto it = mainBulletList.begin(); it != mainBulletList.end();)              {                  if (it->isNeedDelete()) // ATTENTION!! 后面的碰撞检测还没写                  {                      it = mainBulletList.erase(it);                  }                  else                  {                      it->calshot();                      it->show();                      it++;                  }              }          if (!againstBulletList.empty())              for (auto it = againstBulletList.begin(); it != againstBulletList.end();)              {                  if (it->isNeedDelete())                      it = againstBulletList.erase(it);                  else                  {                      it->calMove();                      it->show();                      it++;                  }              }          if (!boomList.empty())              for (auto it = boomList.begin(); it != boomList.end();)              {                  if (it->isNeedDelete())                  {                      it = boomList.erase(it);                  }                  else                  {                      it->calAndMove();                      for (auto it2 = againstBulletList.begin(); it2 != againstBulletList.end();)                      {                          it2 = againstBulletList.erase(it2);                      }                      for (auto it2 = againstRoleLiveList.begin(); it2 != againstRoleLiveList.end();)                      {                          dropout dropTemp; // 生成凋落物                          dropTemp.chushihua(it2->x, it2->y);                          dropOutThingList.push\_back(dropTemp);                          shangHaiXiaoguo temp(it2->x, it2->y); // 爆炸效果                          hurtXiaoGuoList.push\_back(temp);                          it2 = againstRoleLiveList.erase(it2);                      }                      it++;                  }              }          // 此处，ATTENTION ，改成退出并询问，（写一个bool函数询问是否继续）          if (DUT.HP < 0)          {              mciSendString("play BGMdie from 0", 0, 0, 0);              if (isGoOn())              {                  DUT.HP = 2;                  DUT.Boom = 3;                  DUT.GkLuquScore -= 50;              }              else              {                  isGameOver = true;                  return;              }          }          FlushBatchDraw();          DWORD endTime = GetTickCount64(); // 锁帧，60帧          DWORD elapsedTime = endTime - beginTime;          if (elapsedTime < 1000 / FRAME)              Sleep(1000 / FRAME - elapsedTime);      }  } |

|  |
| --- |
| 游戏环节代码 |
| void Game()  {      // 开始界面代码区（开头）      putimage(-screenWidth / 2, 0, &imgStart); // 显示开始界面的图像      BGMstart(1);                              // 播放开头音乐      string KeyWayStateString = "方向键移动，z键射击，x键技能";      string PressEnterToEnter = "按Enter键开始游戏";      setbkmode(TRANSPARENT); // 将字体背景设置为透明的      settextcolor(BLACK);      settextstyle(35, 17, "黑体");      outTextCenter(0, screenHeight - 40, PressEnterToEnter);      outTextCenter(0, screenHeight - 75, KeyWayStateString);      FlushBatchDraw();      while (running)      {          DWORD beginTime = GetTickCount64();          while (peekmessage(&msg))          {              if (msg.message == WM\_KEYDOWN) // 按下按键处理，按下回车键下一步              {                  switch (msg.vkcode)                  {                  case VK\_RETURN:                      running = false;                      mciSendString("play BGM2 from 0", 0, 0, 0);                      break;                  }              }          }          DWORD endTime = GetTickCount64(); // 锁帧，60帧          DWORD elapsedTime = endTime - beginTime;          if (elapsedTime < 1000 / FRAME)              Sleep(1000 / FRAME - elapsedTime);      }      running = true;      // 开始界面代码区（结尾）      //----------------------分割线-------------------------------------------------------------------------------------------------------------      // 选择人物界面（开头）      string leftCharString = "大连理工大学";      string rightCharString = "太车埋土大学";      string chosePartTitle = "请选择你的大学";      isCharNormal = true;           // 用于角色选择阶段的左右跳动以及后续的角色图片问题，true意味着原校徽，false则为琪露诺。      bool chosePartisright = false; // 选择方框位置 ,最开始在左边      bool chosePartisleft = true;      while (running)      {          DWORD beginTime = GetTickCount64();          while (peekmessage(&msg))          {              if (msg.message == WM\_KEYDOWN) // 按下按键处理，按下回车键下一步, 左右键选择              {                  switch (msg.vkcode)                  {                  case VK\_RETURN:                      running = false;                      mciSendString("play BGM2 from 0", 0, 0, 0);                      break;                  case VK\_RIGHT:                      chosePartisright = true;                      chosePartisleft = false;                      isCharNormal = false;                      mciSendString("play BGM2 from 0", 0, 0, 0);                      break;                  case VK\_LEFT:                      chosePartisleft = true;                      chosePartisright = false;                      isCharNormal = true;                      mciSendString("play BGM2 from 0", 0, 0, 0);                      break;                  }              }          }          cleardevice(); // 清除屏幕          putimage(-screenWidth / 2, 0, &imgChosePartBackground);          setlinecolor(YELLOW);          setlinestyle(0, 5);          if (chosePartisleft && !chosePartisright)          {              rectangle(-450, 150, -150, 450);          }          else if (!chosePartisleft && chosePartisright)          {              rectangle(150, 150, 450, 450);          }          putimage\_alpha(-450, 150, &imgDUTbigger); // 绘制左侧图案          putimage\_alpha(150, 150, &imgDUTcartoon); // 绘制右侧图案          outTextCenter(0, 40, chosePartTitle); // 请选择你的大学          outTextCenter(-300, 470, leftCharString);          outTextCenter(300, 470, rightCharString);          FlushBatchDraw();          DWORD endTime = GetTickCount64(); // 锁帧，60帧          DWORD elapsedTime = endTime - beginTime;          if (elapsedTime < 1000 / FRAME)              Sleep(1000 / FRAME - elapsedTime);      }      running = true;      // 选择人物界面（结尾）      // 少女折寿中以及关闭音乐（开头）      BGMstart(0);                                         // 关闭开始页面的BGM      putimage(-screenWidth / 2, 0, &imgTheGirlIsPraying); // 少女祈祷中的图片      settextcolor(WHITE);                                 // 游戏简介      settextstyle(26, 13, "黑体");      string stringTheGameContentIntroduction = "帮助你的大学击败若干高校，成为东方最强大学！";      outTextCenter(-300, 40, stringTheGameContentIntroduction);      if (isCharNormal)          imgChar = imgChar1;      else          imgChar = imgChar2;      FlushBatchDraw();      Sleep(2000);      // 少女折寿中以及关闭音乐（结尾）      //----------------------分割线---------------------------------------------------------------------------------------------------------      // 第一关 part1    （开头）      BGMpartOne(1); // 放音乐      GamePart1();      BGMpartOne(0);      BGMpartOneBoss(0);      if (isGameOver)          return;      // 第一关 part1    （结尾）      putimage(-screenWidth / 2, 0, &imgTheGirlIsPraying); // 少女祈祷中的图片      settextcolor(WHITE);                                 // 游戏简介      settextstyle(26, 13, "黑体");      stringTheGameContentIntroduction = "撑不住的时候，没关系，元神会出手的！";      outTextCenter(-300, 40, stringTheGameContentIntroduction);      FlushBatchDraw();      Sleep(2000);      // 第二关      BGMpartTwo(1);      GamePart2();      BGMpartTwo(0);      BGMpartTwoBoss(0);      // 第二关结尾      return;  } |

|  |
| --- |
| 显示最终排名 |
| struct SCHOOL  {      int score;      string name;  } school[40];  bool cmp(const SCHOOL &a, const SCHOOL &b)  {      return a.score > b.score;  }  school[0].score = 690;  school[0].name = "清华大学";  school[1].score = 685;  school[1].name = "北京大学";  school[2].score = 680;  school[2].name = "浙江大学";  school[3].score = 677;  school[3].name = "上海交通大学";  school[4].score = 675;  school[4].name = "复旦大学";  school[5].score = 670;  school[5].name = "南京大学";  school[6].score = 669;  school[6].name = "中国科学技术大学";  school[7].score = 666;  school[7].name = "华中科技大学";  school[8].score = 662;  school[8].name = "武汉大学";  school[9].score = 658;  school[9].name = "西安交通大学";  school[10].score = 656;  school[10].name = "中山大学";  school[11].score = 655;  school[11].name = "北京航空航天大学";  school[12].score = 654;  school[12].name = "东南大学";  school[13].score = 651;  school[13].name = "北京理工大学";  school[14].score = 650;  school[14].name = "四川大学";  school[15].score = 647;  school[15].name = "哈尔滨工业大学";  school[16].score = 645;  school[16].name = "同济大学";  school[17].score = 642;  school[17].name = "中国人民大学";  school[18].score = 641;  school[18].name = "北京师范大学";  school[19].score = 640;  school[19].name = "天津大学";  school[20].score = 639;  school[20].name = "南开大学";  school[21].score = 637;  school[21].name = "山东大学";  school[22].score = 635;  school[22].name = "西北工业大学";  school[23].score = 632;  school[23].name = "厦门大学";  school[24].score = 630;  school[24].name = "中南大学";  school[25].score = 627;  school[25].name = "吉林大学";  school[26].score = 625;  school[26].name = "中国农业大学";  school[27].score = 624;  school[27].name = "华东师范大学";  school[28].score = 622;  school[28].name = "华南理工大学";  if (isCharNormal)      school[29].name = "大连理工大学";  else      school[29].name = "太车埋土大学";  school[29].score = DUT.GkLuquScore;  sort(school, school + 30, cmp);  BGMlast(1);  putimage(-screenWidth / 2, 0, &imgChosePartBackground);  settextcolor(BLACK);  settextstyle(50, 25, "黑体");  string lastPaiming = "东方大学排名";  string paiming = "排名";  string suxiao = "学校";  string luqufenshuxian = "录取分数线";  outTextCenter(40, 25, lastPaiming);  settextstyle(30, 15, "黑体");  outtextxy(-550, 120, paiming.c\_str());  outtextxy(-470, 120, suxiao.c\_str());  outtextxy(-180, 120, luqufenshuxian.c\_str());  outtextxy(50, 120, paiming.c\_str());  outtextxy(130, 120, suxiao.c\_str());  outtextxy(420, 120, luqufenshuxian.c\_str());  rep(i, 0, 15)  {      ostringstream ss;      ss << i + 1;      outtextxy(-550, 160 + i \* 40, ss.str().c\_str());      outtextxy(-470, 160 + i \* 40, school[i].name.c\_str());      ostringstream sss;      sss << school[i].score;      outtextxy(-150, 160 + i \* 40, sss.str().c\_str());  }  rep(i, 15, 30)  {      ostringstream ss;      ss << i + 1;      outtextxy(50, 160 + (i - 15) \* 40, ss.str().c\_str());      outtextxy(130, 160 + (i - 15) \* 40, school[i].name.c\_str());      ostringstream sss;      sss << school[i].score;      outtextxy(450, 160 + (i - 15) \* 40, sss.str().c\_str());  }  FlushBatchDraw(); |

## 2.5 系统测试

Table 1 系统测试

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用例编号 | 测试项 | 测试步骤 | 输入信号 | 预期结果 | 测试结果 |
| 1 | 开始游戏 | 按ENTER | 键盘 | 进入游戏，开始选择人物 | 通过 |
| 2 | 选择角色 | 方向键 | 键盘 | 选择框会按照键盘指示移动，播放音效 | 通过 |
| 3 | 选择角色2 | 对着选择框中的角色按  ENTER | 键盘 | 成功选择角色，进入游戏界面 | 通过 |
| 4 | 角色移动 | 方向键 | 键盘 | 跟随方向键移动 | 通过 |
| 5 | 角色边缘碰撞检测 | 方向键 | 键盘 | 角色无法越过边框 | 通过 |
| 6 | 发射子弹 | Z键 | 键盘 | 成功发射子弹，能对敌人造成伤害 | 通过 |
| 7 | 释放技能 | X键 | 键盘 | 清屏（对于BOSS造成50点伤害）技能条减少 | 通过 |
| 8 | 潜行 | SHIFT | 键盘 | 主角移动速度减慢，碰撞体积减小，判定体积减小 | 通过 |
| 9 | 受伤 |  |  | 受到伤害后生命值减少，生命值清零后询问是否复活 | 通过 |
| 10 | 复活选择是 | 方向键和ENTER | 键盘 | 角色满血复活，分数减少50，获得2s的金身效果 | 通过 |
| 11 | 复活选择否 | 方向键和ENTER | 键盘 | 进入游戏结算画面，显示排名 | 通过 |
| 12 | 敌人中弹 |  |  | 敌人中弹后声明值减少，生命值为0后播放死亡效果并产生掉落物 | 通过 |
| 13 | 掉落物拾取 |  |  | 靠近掉落物自动拾取，相应的分数增加 | 通过 |

## 2.6 项目总结

笔者作为一个刚入门不久的东方爱好者，我被东方STG的绚丽画面和高难度深深吸引。趁着暑假，我基于自己对东方STG的理解，完成了这个项目——东方名校幻想乡（Eastern Famous School Fantasyland）。游戏中的大部分素材来自网络，图像素材由我亲自处理，背景音乐选用了《东方永夜抄》和《东方幻想乡》的曲目。

在模仿原作的过程中，我尽了最大的努力。游戏中的敌人和我方的子弹均为圆形，而原作中的子弹形状更加多样，未能完全呈现出原作中华丽的攻击效果。普通小怪的射击模式由于时间关系只开发了三种，但我对它们的攻击路径进行了优化，使其更加规则和富有数学美感。BOSS的设计中，攻击路径有两种：一种参考了高中物理的简谐运动，另一种是在屏幕上方中间位置进行大规模攻击。BOSS的射击模式是本作中最复杂的部分之一，包含了九种攻击效果，通过排列组合可以产生更加绚丽的攻击效果，尽可能还原原作中BOSS战的视觉体验。

游戏中的角色、子弹和效果都较为复杂，代码也较长。在开发过程中，C++的类提供了很大的帮助。虽然封装效果不尽如人意，但9个类的使用大大简化了开发过程。游戏中包含许多判定逻辑，如子弹命中、出界和角色受伤等。由于同屏子弹和敌人数量较多，这些判定增加了不少难度。为了解决这个问题，我采用了动态数组，每次发射子弹等操作都会将对象放入相应的动态数组中，再通过遍历数组进行操作，当对象需要消失时则将其从数组中移除。这种方法在时间和空间开销方面都较为高效，较好地解决了问题。

整个游戏制作大约花费了一个多星期，完成后我立即在大工东方爱好者协会中分享了成果，并设立了挑战和奖励，希望大家能试玩并评价。来自计算机科学院的Reeinon同学仅用50分钟就通过了挑战，向他表示衷心的感谢和祝贺。大家对游戏的评价主要集中在画面有趣甚至搞笑和帧数不稳定两个方面。前者正是我期望达到的效果，希望能给玩家带来欢笑；对于后者，虽然大部分判定使用了动态数组，但具体操作仍是遍历元素（最坏情况下时间复杂度为O(n^2)），在图片较多时可能会出现掉帧情况（通常在技能画面，图片较大）。我希望未来能学习更专业的游戏引擎和更低复杂度的算法来改进这一问题。

游戏源代码已公开在我的GitHub上，有兴趣的朋友可以看看，如果能给我点个星那就更好了。

源代码链接：<https://github.com/AcidBarium/Eastern_Famous_School_Fantasyland>