

**程 序 设 计 专 题**

**大 程 序 报 告**



**队名： Pac-Miners**

1. **学生姓名 ： 焦笑然 学号： 3180105869**
2. **学生姓名 ： 张知宇 学号： 3180105087**
3. **学生姓名 ： 常博宇 学号： 3180102943**

**教师： 张引**

**2018~2019春夏学期 2019 年 6 月 9 日**

# 题目描述和题目要求

“黄金矿工”是一款益智类小游戏。

现要求基于libgraphics库，设计带有图形界面的“黄金矿工”游戏，并要求运用菜单、按钮、鼠标、键盘、文件与链表相关操作。**菜单：**玩家可以通过鼠标点选或者快捷键的形式来控制游戏设置。**按钮：**玩家通过点击游戏按钮来切换游戏界面。**鼠标：**实现玩家鼠标与程序的交互。**键盘：**实现玩家键盘与程序的交互。**文件：**使用外置文件的方式保存游戏数据。**链表：**采用链表的形式组织数据结构。

“黄金矿工”的游戏方式为游戏主体在画面中上方摆动，玩家通过释放游戏主体的方式，以碰撞的形式来抓取目标，进行分数的累加。在此基础上实现时间、排行榜、背景音乐等功能。

# 需求设计

**设计启动界面。**游戏开始后播放一小段动画，展现出浙大特色。

**设计欢迎界面。**欢迎界面由“开始游戏”、“游戏帮助”、“排行榜”、“退出游戏”和“游戏设置”等按键构成。当点击“开始游戏”时，进入用户信息登记界面，随后进入游戏界面。点击“游戏帮助”，弹出游戏帮助。点击“排行榜”可以显示

**设计游戏界面。**玩家通过按键“↓”和“↑”或鼠标左右键单击来控制左右摆动的“吃豆人”进行游戏操作。“↓”可以“释放”吃豆人，即使其直线飞出并抓取“生命结晶”，获取一定的积分。“↑”可以“收回”吃豆人。

当“生命结晶”被抓取完毕后，游戏将进行时间的累积计算。游戏以竞速的方式进行，玩家可以通过在最短的时间内抓取所有“能量物质”完成游戏来取得排行榜的优异成绩。

游戏界面还应当具有菜单栏，玩家可以通过鼠标选择或者快捷键的方式来进行游戏的操作。

游戏界面在保持美观度之外，还需要有关卡难度和情景模式的选择余地，玩家可以根据个人喜好选择不同的游戏难度和游戏界面风格。

**设计帮助界面。**玩家通过帮助界面了解游戏的操作。

**设计结束界面。**当玩家抓取完所有“生命结晶”之后，进入结束界面，显示游戏完成情况和玩家的总体排行榜。

# 3、总体设计

## 3.1 功能模块设计

**菜单功能：**设置3个子选项，每个子选项分别设置若干下拉项。当玩家在游戏界面中用鼠标点选相关选项或用快捷键时，呼出相应功能。

**状态栏功能：**状态栏位于游戏界面下方，实时显示游戏主体当前状态及触发效果，便于玩家理解游戏进程。

**计分板、计时板功能：**位于游戏右上角，实时显示当前关卡开始后的分数累计与时间累积情况。

**用户名输入功能：**游戏开始前，玩家可以输入自定义的英文字母或数字昵称，用于排行榜的姓名显示。玩家也可以使用随机显示的默认用户名。

**难度与风格选择功能：**玩家可以根据个人喜好选择游戏进行的难度与游戏的色调风格。

**排行榜功能：**游戏结束展示游戏结束界面后，展示排行榜。排行榜仅展示前二十名，排行依据为分数高者名次靠前，分数相同者用时短者名次靠前。

**图片读取功能：**游戏可以从程序所在文件夹的txt文件中读取处理后的图片信息，并显示在游戏界面的指定位置。

**背景音乐功能：**程序运行后，播放指定的背景音乐文件；当游戏通关后，能停止背景音乐，并播放通关音效，随后继续播放背景音乐。

**游戏帮助功能：**玩家可以翻页查看关于游戏背景介绍、游戏操作说明以及游戏物品功能介绍、游戏攻略。

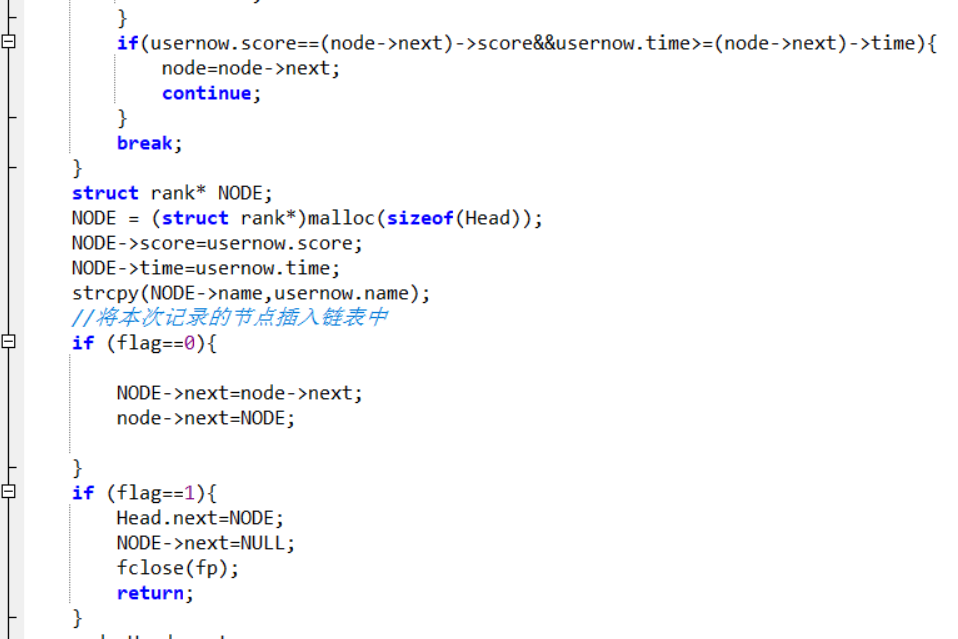
## 3.2 数据结构设计

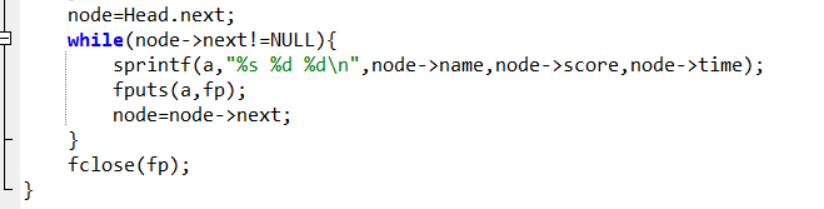
**链表：**在使用文件来保存排行榜的数据时，采用链表的数据储存形式。当玩家开始进行游戏时，程序将读取当前目录下的排行榜信息，并以如图所示的方式为每一条记录建立一个节点，并将这些节点以带有空数据头结点的单向链表保存在全局中。当玩家游戏关卡正常结束时，将产生一个新的节点，并按分数高优先、时间短优先的原则从头结点遍历到尾节点，找到对应位置并插入链表中。



*建立新的链表并初始化操作*





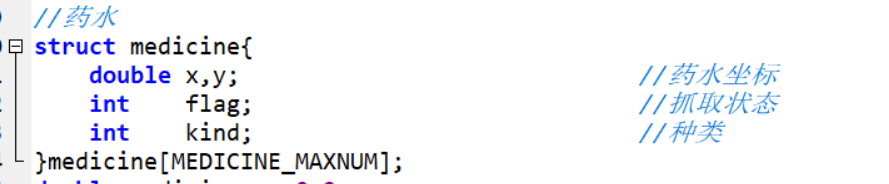


*节点的遍历与插入*

**结构数组：**因为在每一个游戏关卡中，能够显示的物块数量是有限的，因此在管理游戏过程中被抓取的物块时，我们采用了结构数组来记录每个物块的x、y坐标、有效抓取判定半径r、对应分数score、抓取状态flag以及物块特效kind。



*抓取物体的相关结构数组构建*



*抓取药水的相关结构数组构建*

## 3.3 函数功能描述

**1、状态栏绘制函数**

函数原型：void drawStage()

功能描述：在游戏界面下方绘制状态栏，实时显示游戏主体所处状态。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：在调用此函数时，在游戏界面下方以DrawLine(),StartFilledRegion(),EndFilledRegion()函数绘制白色状态栏，并用通过switch()语句判断当前状态，以DrawTextString()表达状态信息。

**2、菜单栏绘制函数**

函数原型：void drawMenu()

功能描述：在游戏界面上方绘制菜单栏。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：函数首先获取窗口信息，随后调用drawMenuBar()、menuList()函数确定菜单栏的子选项和下拉框的显示状态，并绘制相应界面。

**3、游戏初始化函数**

函数原型：void init\_game()

功能描述：初始化游戏使用的各个状态参数。

参数描述： 无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：对全局状态参数进行初始化，并调用物体初始化函数。

**4、物体初始化函数**

函数原型：init\_object()

功能描述：初始化物体的位置与分数。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：调用函数后，对Gamemode进行switch分支判断，产生一个循环变量的阈值。当循环变量小于阈值时，为每个object成员变量赋值。

**5、颜色初始化函数**

函数原型：init\_color()

功能描述：定义了不同游戏模式下游戏背景的色彩。

参数描述： 无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：以函数DefineColor()定义了游戏背景所需的若干色彩，完成了游戏界面调色工作。

**6、按钮绘制函数**

函数原型：void drawButtons()

功能描述：在开始界面绘制操作按钮，供玩家进行选择。

参数描述： 无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量

重要局部变量用途描述：无重要局部变量。

函数算法描述：获取当前窗口大小信息，并在指定位置绘制欢迎界面的按钮。

**7、排行榜绘制函数**

函数原型：void drawRank()

功能描述：绘制排行榜界面。

参数描述： 无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：获取当前窗口大小信息，并在游戏左上角绘制“返回”按钮，在游戏界面绘制排行榜画面。

**8、游戏帮助绘制函数。**

函数原型：void drawHelp()

功能描述：绘制游戏帮助界面。

参数描述： 无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：获取当前窗口大小信息，并在窗口左上角绘制“返回”按钮，并在游戏界面中部绘制游戏帮助信息。

**9、难度与情景选择函数**

函数原型：void drawUser()

功能描述：绘制难度选择与情景模式选择按钮。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：调用drawLabel()、button()等函数绘制相应按钮。

**10、界面刷新函数**

函数原型：void display()

功能描述：核心函数，重新绘制游戏画面，实现刷新。

参数描述： 无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：首先调用DisplayClear()函数清除画面内容，继而以switch()分支判断界面显示状态，并分别调用相应函数绘制相应界面。

**11、背景绘制函数**

函数原型：void drawBackground()

功能描述：绘制情景模式的背景。

参数描述： 无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：以switch()分支判断玩家选择的游戏情景，并绘制相应背景。

**12、抓取状态函数**

函数原型：void judgeGet()

功能描述：判断游戏主体是否与物体相接，并修改游戏主体状态与物体状态。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：double x,y,distance;

重要局部变量用途描述：时刻改变并记录游戏主体的中心坐标，进行边缘抓取判断。

函数算法描述：以x=hookx+cos(hookangle\*PI/180)\*hooklength、y=hooky+cos(hookangle\*PI/180)\*hooklength获取游戏主体实时位置，并计算物体中心点坐标与游戏主体中心坐标的距离来判断是否抓取成功。

**13、游戏主体伸出函数**

函数原型：void stretchHook()

功能描述：进行游戏主体伸出的状态修改。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：调用函数后首先判断游戏主体是否正常伸出，继而判断伸出长度是否超过限制。若未超过限制，则对状态参量hooklength进行增加slength（单位时间内伸长距离）的操作。

**14、游戏主体收缩函数**

函数原型：void shrinkHook()

功能描述：进行游戏主体收缩的状态修改。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：调用函数后首先判断游戏主体是否处于未成功抓取物体的状态，继而判断当前偏离平衡位置hooklength是否大于零。若大于零则减小，若为零则修改游戏主体运动状态标记为SWING。若判断剩余未抓取物体数目为零，则游戏结束关卡。

**15、游戏主体绘制函数**

函数原型：draw\_hook()

功能描述：核心函数，进行游戏主体的绘制。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：double dx,dy,centrex,centrey

重要局部变量用途描述：dx,dy用于确定游戏主体偏离平衡中心位置的横向、纵向偏移量，centrex、centrey用于确定游戏主体在画布上的横纵坐标。

函数算法描述：调用函数后首先获取dx，dy，centrex，centrey的值，然后switch分支判断当前游戏主体的运动状态（SWING、STRETCH、SHRINKGET、SHRINKFALL），并根据相应状态绘制对应的游戏主体图案。

**16、游戏主体摆动函数**

函数原型：void swingHook()

功能描述：绘制游戏主体处于SWING状态的摆动效果。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：调用后读取游戏主体处于SWING状态下的当前角度，然后调用draw\_hook()函数绘制相应图片。

**17、待抓取物体绘制函数**

函数原型：DrawAllObject()

功能描述：绘制所有可抓取物体，包括特效物体与计分物体。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：调用函数后用循环变量遍历储存物体信息的结构数组，判断是否抓取，并调用DrawObject()与DrawMedicine()函数绘制相应的计分物体与特效物体。

**18、用户信息输入界面**

函数原型：void drawEditText()

功能描述：提供一个输入框供玩家输入自定义的昵称（支持英文、数字）

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：调用函数后，在画面中央绘制textbox输入框，并将输入信息存放在游戏初始化产生的节点的name成员变量中。

**19、计分、计时板绘制函数**

函数原型：void drawBoard()

功能描述：在游戏界面右上角绘制一个计分板，当物体抓取瞬间，进行积分累计。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：调用函数后，进行时间变动的累计、抓取分数的累计，并用drawBox()函数绘制相应图案。

**20、字符串转化函数**

函数原型：void GetDoubleStr(double value)

功能描述：将传入的double类型的时间数据转化为可输出的字符串形式。

参数描述：double value

返回值描述：无返回值。

重要局部变量定义：int i,index

重要局部变量用途描述：i用于逐个读取数组中每个字符时计步，index用于确定字符串结尾定位。

函数算法描述：调用函数后，先用sprintf(stringsoretime,”%8.f”,value)将clock()函数传入的时间信息保存在字符数组中，然后len=strlen(stringscoretime)获取长度，最后用i从6+len到5的递减循环遍历数组直到读取”.”作为时间字符串的结束。

**21、像素点绘制函数**

函数原型：void DrawDot(double red, double green, double blue, double x, double y, double size)

功能描述：在指定的(x,y)坐标处绘制边长为size，RGB颜色为(red, green, blue)的像素点。

参数描述：red, green, blue 为0-1的浮点数，表示RGB参数。x,y表示绘制像素点左下角的坐标。size表示绘制像素点的边长。当边长足够小时，可以视为绘制了一个点。

返回值描述：无返回值。

重要局部变量定义：static int flag, static char \*a[]

重要局部变量用途描述：在绘制像素点时，需要自定义颜色，a[]中保存了一些随机颜色名称，flag用于名称的切换。

函数算法描述：a[]是一个保存了三个字符串的指针数组，在每次调用函数时，根据flag的值选择为传入的（R，G，B）定义一个颜色名称，作为DefineColor（）的参数。然后调用SetPenColor（）函数更改画笔颜色，调用StartFilledRegion（）、MovePen（）、DrawLine（）、EndFilledRegion（）绘制一个指定颜色的像素点。

**22、图片绘制函数**

函数原型：void DrawPic(char a[], double x, double y)

功能描述：在坐标(x.y)处绘制指定名称的图片。

参数描述：a[]是一个字符指针，用于指定图片信息来源。x,y表示绘制图片的左上顶点坐标。

返回值描述：无返回值。

重要局部变量定义：int width, height, degree, red\_, green\_, blue\_

重要局部变量用途描述：width、height、degree用于读取图片的长宽坐标与颜色深度，red\_, green\_, blue\_用于读取色彩（R，G，B）。

函数算法描述：调用函数后，首先freopen(a,”r”,stdin)重定向数据输入，然后scanf(“(%d,%d,%d)”,&width, &height, &degree)读取图片信息。最后用两层循环遍历图片的宽高，调用DrawDot（）函数绘制图片每一个像素点。

**23、排行榜初始化函数**

函数原型：InitRankings()

功能描述：游戏开始时，初始化排行榜信息。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：调用函数后，从当前目录读取排行榜文件“rankngs.txt”，初始化用户分数、时间信息。继而读取用户名、分数、时间信息，建立单向链表。

**24、排行榜绘制函数**

函数原型：void drawRanking()

功能描述：绘制排行榜界面。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：struct rank\* node

重要局部变量用途描述：链表节点，用于遍历全体节点。

函数算法描述：调用函数后，创建一个节点指向表头，然后依次遍历每个节点，在每次遍历时读取相应用户的序号、昵称、分数、时间，然后逐行绘制在画布上。

**25、排行榜分数更新函数**

函数原型：rankChange()

功能描述：将本次游戏中玩家的相关信息（用户昵称、时间、分数）保存在文件中。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：struct rank\* node

重要局部变量用途描述：链表节点，用于插入节点时遍历。

函数算法描述：调用函数后，依据得分usernow.score和链表中各节点node.score相比较，分数相同者比较时间usernow.time。以分数高、时间短者优先。

**25、程序启动动画（一）绘制函数**

函数原型：void stretchBoot()

功能描述：程序启动时调用此函数，播放程序启动动画（一）。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：对全局变量bootlength进行累加。

**26、程序启动动画（二）绘制函数**

函数原型：void biggerBoot()

功能描述：程序启动时调用此函数，播放程序启动动画（二）。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：对全局变量bootsize进行累加。

**27、程序启动动画（三）绘制函数**

函数原型：void drawBootBackground ()

功能描述：程序启动时调用此函数，播放程序启动动画（三）。

参数描述：无传入参数。

返回值描述：无返回值。

重要局部变量定义：无重要局部变量。

重要局部变量用途描述：无重要局部变量。

函数算法描述：判断全局变量bootlength的值是否达到阈值winwidth，通过if-else分支判断结构选择执行drawBootBackground1()或者drawBootBackground2()，绘制动画。

## 3.4 程序文件结构

**1）文件函数结构**

**graphics.c&libgraphics.h:**

static void InitCheck(void);

static void InitGraphicsState(void);

static void InitDisplay(void);

static void InitDrawingTools(void);

static void DisplayExit(void);

static HWND FindConsoleWindow(void);

static BOOL CALLBACK EnumerateProc(HWND window, LPARAM clientData);

static void RegisterWindowClass(void);

static LONG FAR PASCAL GraphicsEventProc(HWND w, UINT msg,WPARAM p1, LPARAM p2);

static void CheckEvents(void);

static void DoUpdate(void);

void DisplayClear(void);

static void PrepareToDraw(void);

static void DisplayLine(double x, double y, double dx, double dy);

static void DisplayArc(double xc, double yc, double rx, double ry,

double start, double sweep);

static void RenderArc(double x, double y, double rx, double ry,

double start, double sweep);

static void DisplayText(double x, double y, string text);

static void DisplayFont(string font, int size, int style);

static int FindExistingFont(string name, int size, int style);

static void SetLineBB(RECT \*rp, double x, double y, double dx, double dy);

static void SetArcBB(RECT \*rp, double xc, double yc,

double rx, double ry, double start, double sweep);

static void SetTextBB(RECT \*rp, double x, double y, string text);

static void StartPolygon(void);

static void AddSegment(int x0, int y0, int x1, int y1);

static void DisplayPolygon(void);

static void AddPolygonPoint(int x, int y);

static void InitColors(void);

static int FindColorName(string name);

static bool StringMatch(string s1, string s2);

static bool PrefixMatch(string prefix, string str);

static int RectWidth(RECT \*rp);

static int RectHeight(RECT \*rp);

static void SetRectFromSize(RECT \*rp, int x, int y, int width, int height);

static double Radians(double degrees);

static int Round(double x);

static double InchesX(int x);

static double InchesY(int y);

static int PixelsX(double x);

static int PixelsY(double y);

static int ScaleX(double x);

static int ScaleY(double y);

static int Min(int x, int y);

static int Max(int x, int y);

void InitGraphics(void);

void MovePen(double x, double y);

void DrawLine(double dx, double dy);

void DrawArc(double r, double start, double sweep);

double GetWindowWidth(void);

double GetWindowHeight(void);

double GetCurrentX(void);

double GetCurrentY(void);

void DrawEllipticalArc(double rx, double ry,

double start, double sweep);

void StartFilledRegion(double grayScale);

void EndFilledRegion(void);

void DrawTextString(string text);

double TextStringWidth(string text);

void SetFont(string font);

string GetFont(void);

void SetPointSize(int size);

int GetPointSize(void);

void SetStyle(int style);

int GetStyle(void);

double GetFontAscent(void);

double GetFontDescent(void);

double GetFontHeight(void);

double GetMouseX(void);

double GetMouseY(void);

bool MouseButtonIsDown(void);

void WaitForMouseDown(void);

void WaitForMouseUp(void);

bool HasColor(void);

void SetPenColor(string color);

string GetPenColor(void);

void DefineColor(string name,

double red, double green, double blue);

void SetPenSize(int size);

int GetPenSize(void);

void SetEraseMode(bool mode);

bool GetEraseMode(void);

void SetWindowTitle(string title);

string GetWindowTitle(void);

void UpdateDisplay(void);

void Pause(double seconds);

void ExitGraphics(void);

void SaveGraphicsState(void);

void RestoreGraphicsState(void);

double GetFullScreenWidth(void);

double GetFullScreenHeight(void);

void SetWindowSize(double width, double height);

double GetXResolution(void);

double GetYResolution(void);

**random.c&random.h:**

void Randomize(void);

int RandomInteger(int low, int high);

double RandomReal(double low, double high);

bool RandomChance(double p);

**simptio.c&simptio.h:**int GetInteger(void);

long GetLong(void);

double GetReal(void);

string GetLine(void);

string ReadLine(FILE \*infile);

**strlib.c&strlib.h:**  
static string CreateString(int len);;

string Concat(string s1, string s2);

char IthChar(string s, int i);

string SubString(string s, int p1, int p2);

string CharToString(char ch);

int StringLength(string s);

string CopyString(string s);

bool StringEqual(string s1, string s2);

int StringCompare(string s1, string s2);

int FindChar(char ch, string text, int start);

int FindString(string str, string text, int start);

string ConvertToLowerCase(string s);

string ConvertToUpperCase(string s);

string IntegerToString(int n);

int StringToInteger(string s);

string RealToString(double d);

double StringToReal(string s);

static string CreateString(int len);

exceptio.c&exception.h:

static context\_block \*FindHandler(exception \*e);

void RaiseException(exception \*e, string name, void \*value);

bool HandlerExists(exception \*e);

static context\_block \*FindHandler(exception \*e);

**genlib.c&genlib.**h:

void \*GetBlock(size\_t nbytes);

void FreeBlock(void \*ptr);

void ProtectBlock(void \*ptr, size\_t nbytes);

void Error(string msg, ...);

**imgui.c&imgui.h:**

void InitGUI();

void mySetPenColor(char \*color);

void setButtonColors(char \*frame, char\*label, char \*hotFrame, char \*hotLabel, int fillflag);

void setMenuColors(char \*frame, char\*label, char \*hotFrame, char \*hotLabel, int fillflag);

void setTextBoxColors(char \*frame, char\*label, char \*hotFrame, char \*hotLabel, int fillflag);

void usePredefinedColors(int k);

void usePredefinedButtonColors(int k);

void usePredefinedMenuColors(int k);

void usePredefinedTexBoxColors(int k);

void uiGetMouse(int x, int y, int button, int event);

void uiGetKeyboard(int key, int event);

void uiGetChar(int ch);

static bool inBox(double x, double y, double x1, double x2, double y1, double y2);

int button(int id, double x, double y, double w, double h, char \*label);

static int menuItem(int id, double x, double y, double w, double h, char \*label);

static char ToUpperLetter(char c);

static char shortcutkey(char \*s);

int menuList(int id, double x, double y, double w, double wlist, double h, char \*labels[], int n);

void drawMenuBar(double x, double y, double w, double h);

int textbox(int id, double x, double y, double w, double h, char textbuf[], int buflen);

void drawRectangle(double x, double y, double w, double h, int fillflag);

void drawBox(double x, double y, double w, double h, int fillflag, char \*label, char labelAlignment, char \*labelColor);

void drawLabel(double x, double y, char \*label);

**picture.c:**

void DrawDot(double red,double green,double blue,double x,double y,double size);

void DrawPic(char a[],double x,double y);

void drawWatermelon(double x,double y,double r);

void drawCandy(double x,double y,double r);

void drawBeer(double x,double y,double r);

void drawRice(double x,double y,double r);

void drawMilk(double x,double y,double r);

void drawMedicine(double x, double y, int kind);

**ranking.c&ranking.h:**

void InitRankings();

void drawRanking();

void changeScore(int score\_change);

void changeTime(int time\_change);

void initName;

void rankChange();

**main.c:**

void CharEventProcess(char ch);

void KeyboardEventProcess(int key, int event);

void MouseEventProcess(int x, int y, int button, int event);

void TimerEventProcess(int timerID);

void DrawAllObject();

void DrawObject(int index);

void DrawMedicine(int index);

void drawStage();

void drawMenu();

void drawEditText();

void drawSetting();

void drawUser();

void drawBoard();

void drawButtons();

void drawRank();

void drawBackground();

void drawGameBackground();

void drawBeginBackground();

void drawUserBackground();

void drawHelpBackground();

void drawTxtBackground();

void drawRankBackground();

void stretchBoot();

void biggerBoot();

void drawBootBackground();

void drawBootBackground1();

void drawBootBackground2();

void judgeEnd();

void drawEnd();

void drawHelp();

void draw\_hook();

void display();

void init\_game();

void init\_object();

void judgeGet();

void stretchHook();

void shrinkHook();

void rot\_hook();

void swingHook();

void GetDoubleStr(double value);

**2）多文件构成机制**

主程序源码main.c使用的编译预处理指令如下：

#include "graphics.h"

#include "extgraph.h"

#include "genlib.h"

#include "simpio.h"

#include "conio.h"

#include <stdio.h>

#include <stdlib.h>

#include <stddef.h>

#include <math.h>

#include <olectl.h>

#include <windows.h>

#include <mmsystem.h>

#pragma comment(lib,"Winmm.lib")

#include <wingdi.h>

#include <ole2.h>

#include <ocidl.h>

#include <winuser.h>

#include "imgui.h"

#include <time.h>

在编写ranking.h文件时采用

#ifndef \_\_\_\_ranking\_h\_\_\_\_

#define \_\_\_\_ranking\_h\_\_\_\_

#endif

来防止重复包含。

**外部变量：**ranking.c文件使用了来自main.c中的winwidth与winheight变量。

**外部函数：**main.c文件中对于界面绘制、游戏主体及物块绘制采用内部函数，其余的按钮绘制、排行榜显示调用外部函数。

# 4、部署与运行

**4.1 编译安装运行说明**

项目保存在“GoldMiner-Group23.rar”文件中，解压缩后如图。



*压缩包内各文件夹与文件*

各文件夹与文件说明：

**.git文件夹：**小组合作采用github仓库管理代码，提高了工作效率；同时分布式代码资源管理的系统有利于程序在不同版本之间切换。（备注：考虑到上传文件大小问题，并未将.git仓库上传）

**DevProject文件夹：**存放项目文件及编译相关附属文件。进行编译时，进入DevProject文件夹下，打开GoldMiner.dev文件，按F11即可进行游戏的编译。



*DevProject文件夹下文件*

**discussions文件夹：**存放小组讨论记录。



*Discussions文件夹下各文件*

**presentation文件夹：**存放了游戏展示的PPT与游戏讲解视频。

**filemanage文件夹：**存放自主编写的ranking.c,ranking.h,picture.c文件。

****

*Filemanage文件夹下各文件*

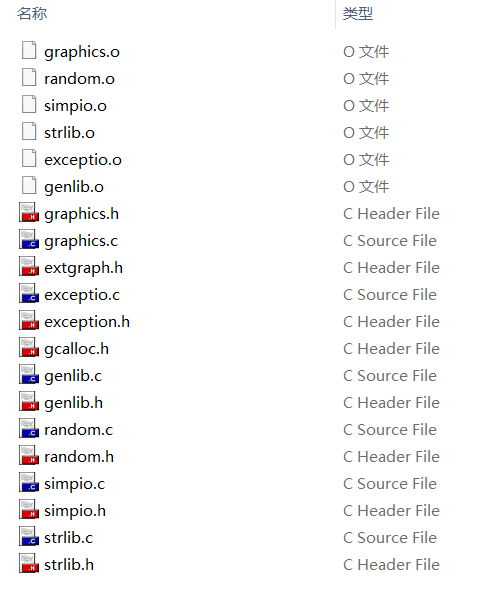
**Game文件夹：**存放游戏需要的文件。游戏的进行需要在此目录中。



*Game文件夹下各文件*

**备注：**游戏程序GoldMiner.exe必须和stage\_01.wav处于同一目录中，否则背景音乐无法正常加载；必须和win.wav处于同一目录中，否则背景音效无法加载；必须和rankings.txt处于同一目录中，否则玩家记录将会丢失。若实现“游戏关于”的全部效果，jxr.txt;cby.txt;zzy.txt必须处于同一目录中。

**libgraphics文件夹：**存放规定使用的libgraphics库函数文件。



*libgraphics文件夹下各文件*

**simpleGUI文件夹：**存放课程提供的面板绘制函数库文件。



*simpleGUI文件夹下文件*

**备注：**我们根据美观需求，对库函数文件进行了修改。

**SourceFiles文件夹：**存放main.c文件。

[**README.md文件**](README.md)**：**记录了各文件夹及文件说明。

[**README.pdf文件**](README.pdf)**：**考虑到部分计算机可能无法打开.md文件，特将README.md文件内容制作成pdf格式。

**程序设计专题大程序报告\_Pac-Miner\_Group23.docx文件：**即为本报告。

**4.2典型测试情况**

**Test1：**读取排行榜文件后输出游戏信息时，结尾总是附加多余的零。

解决办法：通过头结点初始化、定位链表建立函数发现，在使用foef（）判断文件是否结束时，文件指针不移动。因而在读取空文件时，第一个节点的usernow.score和usernow.time会被自动初始化为0.

**Test2：**游戏物块的随机生成有时会重合，有时出现同时抓取情况。

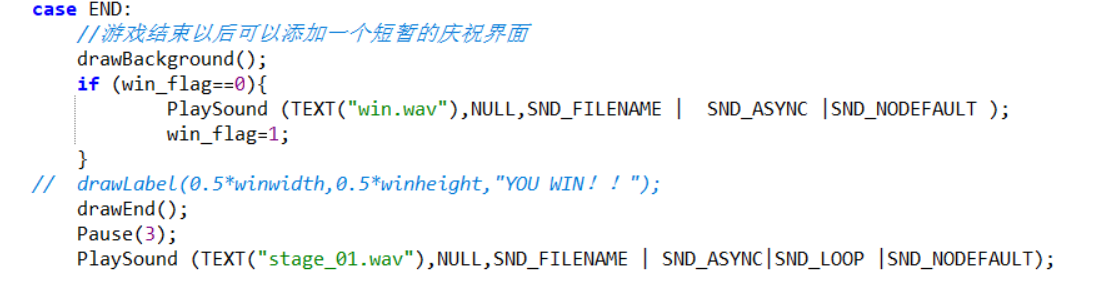
解决办法：设定静态全局变量判断是否已经抓取，若为抓取状态则关闭抓取功能；设定距离判断变量distance，判断图案是否重叠，若重叠，则重新产生一个随机化坐标。

**Test3：**若玩家删除背景音乐，则播放音乐时会反复出现错误警报。

解决办法：添加SND\_NODEFAULT参数，指定不采用默认音频。

**Test4：**游戏结束以后播放音乐会中断原有的背景音乐。

解决办法：异步调用音效文件，或者重新播放背景音乐。



*正在处于调试阶段的代码*

**Test5：**通过“游戏设置”来选择关卡难度随后进入游戏界面的方式与“开始游戏”重复，而且降低了游戏体验。

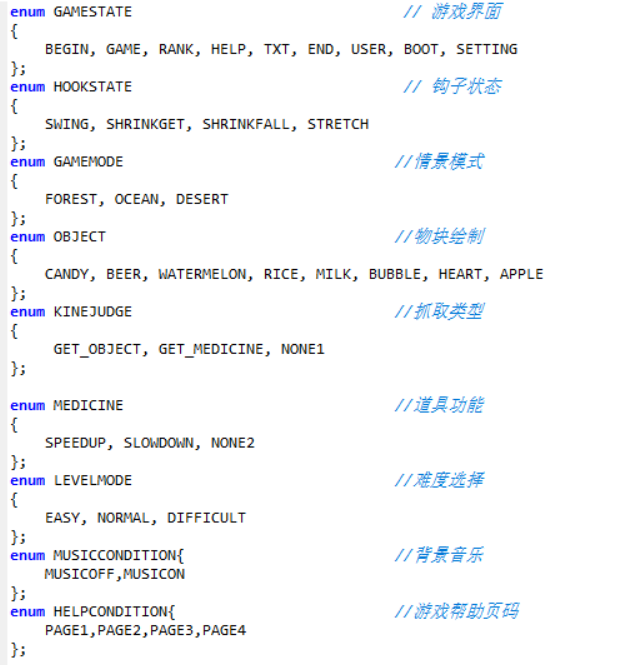
解决办法：将“游戏设置”置入“游戏开始”界面之中，并修改“游戏设置”的功能选项。

**Test6：**使用DrawTextString函数不便于确定画笔位置，而且每一次调用需要修改画笔位置，过于繁琐。

解决办法：使用课程提供的simpleGUI库中的drawLabel函数，并对其进行修改，实现了简介明了绘制文本的功能。

**Test7：**绘图擦除不干净，有残留像素点。

解决办法：问题的出现与浮点型数据在计算机中的保存方式有关，是存在一定误差的。因此换一种画图的思路，即不采用绘制-擦除，而采取状态-刷新模式。不仅便于功能的拓展，而且完美地解决擦除困难的问题。



*游戏状态的定义*

# 5.组内分工

**5.1 组内分工情况**

焦笑然：构建游戏状态管理系统；构建欢迎界面、排行榜界面；构建图片处理系统；协助修改基于链表的排行榜文件处理系统；构建计分板、计时器系统；统筹管理全组成员的工作进度；构建随机命名生成器；PPT制作与展示、游戏试玩展示、视频混剪。

张知宇：构建游戏主体状态函数；构建文本输入系统；构建基于结构数组的游戏对象管理系统；构建并封装基于libgraphics库的游戏主体绘画功能；实现按钮颜色与形状自定义；构建程序启动欢迎动画效果；构建基于游戏情景选择的多状态背景绘画功能。

常博宇：构建基于链表的排行榜文件处理系统；嵌入背景音乐播放系统；构建状态栏系统；协助修改图片读取系统；构建游戏帮助系统；构建多页面管理模式的游戏帮助功能；游戏试玩改进；协助完成绘图程序的封装；整理完成大程序报告。

**5.2个人实践过程中遇到的难点及解决方案**

**焦笑然：**

（1）在重定义已有颜色时，lib库中colorTable中三种色度存储的值会改变，而画出了的颜色却不会改变。

解决方案有二，一改库，而不再用一个颜色来存当前画图颜色，二是采取寄存器或者缓存的思想，开一个小数组或者多设几个变量来存储当前颜色，避免重定义后立即调用的问题。我们采取了后者。

（2）不规则的矿和长相生动的矿工要求我们必须可以调用图片，自己画的话效果会大打折扣。

解决方案：最初的想法是找库，我试用了xege库，了解了其中调用图片的细节。ege库中可以创建一种PIMAGE类型，有getimage()和putimage()两个函数用于读入图片和贴图。但是lib库没有留相应的接口。此外，ege库要求文件名后缀必须是.cpp，而lib库要求项目类型必须是c，否则就好出现bool类型的重定义问题。总而言之，这条路便走不通了。

**张知宇：**

（1）图片导入问题：libgraphics无法实现导入图片。

解决方案：采用matlab读入图片信息（各像素点的三原色），分割屏幕进行小点的绘制。但也存在缺点：读入速度慢，容易卡机，于是我们减小清晰度，用于了游戏帮助中（关于我们）照片展示。

（2）钩子：钩子为动态进行，拥有各种角度位置，不好进行图片读取。

解决方案：改变整体游戏风格为吃豆人，采用极坐标和三角函数直接在画布上作画。

（3）开机动画：在drawBootBackground()采用pause()函数发现进入死循环。

解决方案：开机动画改为利用全局变量的状态刷新。

（4）画图擦不干净问题：原采用用白色画笔再次绘制背景实现状态的转变，费时费力并且无法擦除干净

解决方案：全局设定一个定时器，通过改变Interface的值进行状态的刷新与跳转

**常博宇：**

（1）在调用排行榜文件储存数据时出现第一个节点usernow.time和usernow.score默认为0的现象。

解决方案：feof函数判断文件是否结束时不移动节点，因此添加一个flag判断是不是第一次读取。若是，则移动指针并跳过建立节点的过程。若不是，则正常建立节点。

（2）游戏结束以后播放音乐会中断原有的背景音乐。

解决办法：异步调用音效文件，或者重新播放背景音乐。

（3）drawPic函数绘制图片不方便，像素点过少则图片过小，像素点多则程序崩溃。

解决办法：折中两种思路，调整drawDot函数中size参数，将少量像素点绘制成大图片，牺牲清晰度换取较好的可视化效果。

# 6、合作纪要

**线下讨论：**

**2019年5月5日：**

基本敲定了游戏的宏观框架，并头脑风暴对细节的处理进行了讨论。

[C程序设计小组讨论记录表1](discussions/C程序设计小组讨论记录表1.docx)（位于discussions目录下）

**2019年5月15 日：**

对游戏的框架进行了进一步的交流讨论，确定了以排行榜的形式实现文件的读取，并设想了多关卡的游戏方式，商讨了游戏界面的按钮设定，构思了BGM的插入方式。

[C程序设计小组讨论记录表2](discussions/C程序设计小组讨论记录表2.docx)（位于discussions目录下）

**2019年5月22日：**

针对排行榜进行了讨论，按一定顺序输出用户的昵称、分数、时间，2） 游戏结束后，更新用户当前分数，并插入文件对应的名次位置。可以设置几个“NPC”占据排行榜，调动玩家的积极性。

[C程序设计小组讨论记录表3](discussions/C程序设计小组讨论记录表3.docx)（位于discussions目录下）

**2019年5月29日：**

针对图片进行了讨论。一是寻找可用的图片库，但是最初想法与实际情况产生了很大的冲突。经过慎重思考和讨论最后决定采取寄存器或者缓存的思想，开一个小数组或者多设几个变量来存储当前颜色，避免重定义后立即调用的问题。

[C程序设计小组讨论记录表4](discussions/C程序设计小组讨论记录表4.docx)（位于discussions目录下）

**2019年6月4日：**

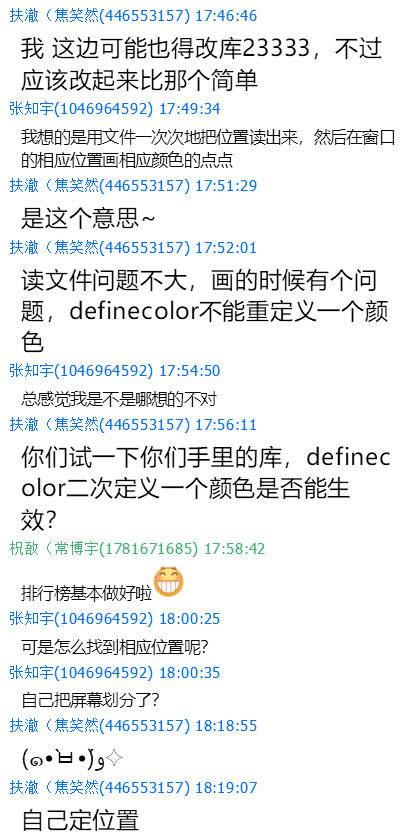
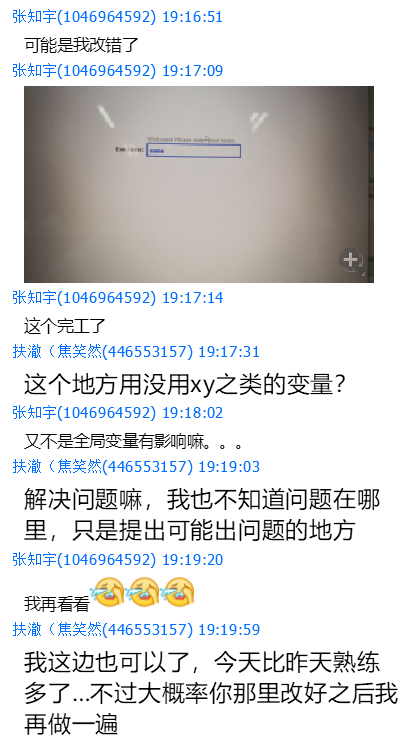
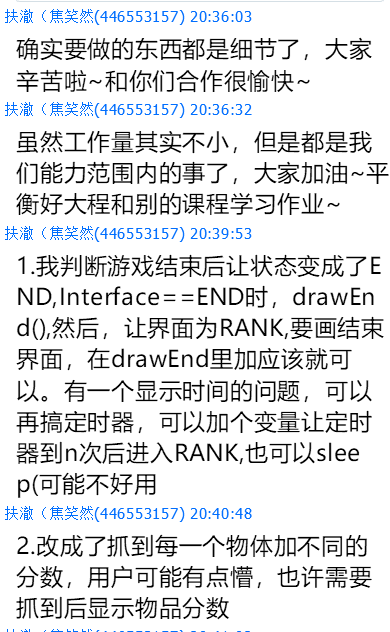
收尾工作讨论。界面优化方面，采取libgraphics画图而不是用直接画图；根据情景模式的选择自定义背景色的显示，将随机半径换为自定义半径。修复了钩子与背景不兼容、排行榜格式、物体相同分数的bug。

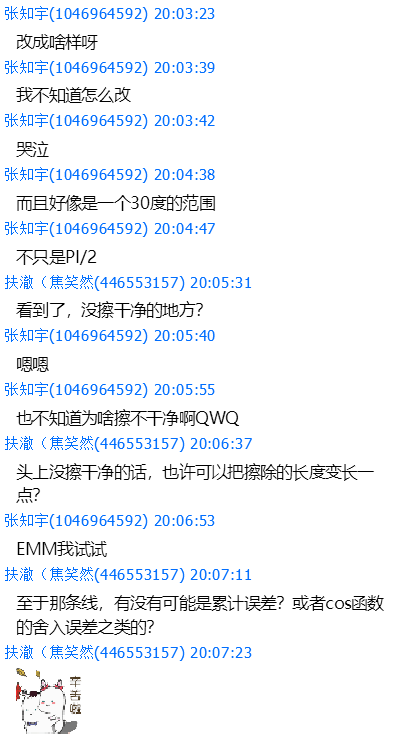
[C程序设计小组讨论记录表5](discussions/C程序设计小组讨论记录表5.docx)（位于discussions目录下）

**线上讨论：**



*蓝色背景标注日期为进行线上讨论日期*

*部分交流记录展示*

# 7、总结

**7.1程序亮点或创新之处**

（1）突破了传统“黄金矿工”的游戏形象设定，采取全新的游戏背景、游戏主体、物块绘图，增添游戏的趣味。

（2）增添了玩家输入自定义昵称功能，玩家可以保留自己的昵称，参与排行榜的竞争。另外，我们采用了随机数生成用户名的功能，提高了游戏的可玩性。

（3）增添了游戏难度选择（小白模式、普通模式、地狱模式）、游戏情景选择（森林模式、海洋模式、沙漠模式），突破了传统意义上的黄金矿工对游戏素材的约束，并且游戏显示的字体颜色会随着情景模式的改变而改变。

（4）增添了加速、减速道具药水，提高了游戏可玩性。

（5）游戏帮助最后展示了用彩蛋的形式展现了“贴图功能”。

**7.2应用知识点**

**（1）运用链表进行数据的管理。**

在管理排行榜文件时，采用单向链表的数据结构，将玩家每次记录保存在一个个节点中，按分数高者优先、分数相同时时间短者优先的方式进行排序。

**（2）运用文件保存游戏数据。**

使用“ranking.txt”保存游戏排行榜数据，运用了feof()、fopen()、fclose()等函数对文件进行操作。另外，我们将图片信息保存在txt文件中，实现了绘制任意图片到游戏界面的功能。

**（3）运用结构数组管理游戏物体。**

运用结构数组相关知识，管理游戏过程中出现的游戏物体的信息，简化了游戏操作，规范了游戏数据的保存样式，提高了游戏的健壮性。

**（4）运用枚举类型表示游戏状态。**

运用枚举类型enum相关知识，使用BEGIN、GAME、RANK、HELP、TXT、END、USER表示游戏状态，使用SWING、SHRINKGET、SHRINKFALL、STRECH表示游戏主体状态，使用FOREST、OCEAN、DESERT表示游戏模式，使用CANDY, BEER, WATERMELON, RICE, MILK, BUBBLE, HEART,APPLE表示物体种类，使用EASY, NORMAL, DIFFICULT表示游戏难度，使用SPEEDUP, SLOWDOWN, NONE2表示药水效果，使用MUSICOFF,MUSICON表示音乐开启状态，使用PAGE1,PAGE2,PAGE3,PAGE4表示游戏帮助界面的页码数。

**（5）使用静态数据类型。**

仅供文件内部使用的变量在定义时采用static类型声明，避免分工时出现的相同变量名称在不同文件中可能重复定义的现象。

**7.3 心得体会**

**总体描述**

我们的项目在2019年5月4日正式启动，一个多月以来，我们每周至少两次线下讨论，除了第一次以外，其余讨论所有成员全程参加。小组内气氛特别和谐，每次见面都有新的思路想法提出，进一步促进游戏的开展。游戏刚起步的时候我们还囿于传统意义上的“黄金矿工”，为“老爷爷”形象的画法纠结了一段时间，甚至因此还编写了从txt文本中读取图片信息并绘制的函数。

在这次长达一个多月的密切合作中，组长焦笑然在领导组员工作、解决关键性问题（比如编写图片读取函数、游戏刷新状态管理、游戏整体思路等）方面做出了重大贡献。组员张知宇主要负责利用数学知识进行游戏主体、物块以及按钮的绘制，工作量很大；在团队思路受到束缚时，张知宇总能“灵光一闪”，提出创新的想法，从而使难题迎刃而解。组员常博宇主要负责拓展功能的编写，比如文件保存排行榜数据、背景音乐功能实现；同时，常博宇往往能够发现游戏中不足之处，站在玩家的角度对游戏的可玩性提出了建议，提高了舒适度。

每次线下讨论都能对游戏的进展有很大帮助。其实在一开始我们也曾试过线上交流，不过限于通讯工具，交流起来没有线下那样方便。于是我们将原先课表中的正课与上机课时间拿来小组合作编程，遇到任何问题都可以交流讨论解决，大大减少了个人“钻牛角尖”的时间浪费。由于线上讨论的高效性以及计划安排的长远性，我们基本不因大程通宵熬夜。每次的线下讨论都是在愉悦气氛中进行，我们不仅从中学会了合作，而且也收获了友谊与难忘的回忆。

虽然交流的主要方式是线下讨论，但我们也没有因此偏废线上交流。当在线下讨论之外有什么新的想法思路，或者找到了什么好的网站（比如调色网），我们能及时分享。另外在管理整合代码时，线上文件传输压缩包也不失为一种好方法。线上交流的截图也能让队友及时了解最新进展，并提出自己的看法。最令人兴奋的是临近ddl前两天，我们的按钮、背景有了突破性进展，实现了从方形到圆角、从无背景到双侧颜色背景再到如今的吃豆人主题背景！

我们的工作并不是死板的按照分工来做，因为那样的话一人做不出来就会拖慢整组进度。我们在分工之后，会及时共享交流自己的编程情况，及时提出自己在写代码中遇到的问题。因为在具体工作开展之前每个人对工作性质和难度不是很了解，或者了解有偏差。当出现这种情况时，我们会采用“工作交换”的方法，将两人的工作暂时互换。这样一来不仅使得每个人都参与熟悉了队友的工作情况，提高了自己对大程序的了解，也能防止钻牛角尖后拖慢全组进度情况的出现。有时候代码出现的BUG自己不易发现，在这种情况下换换工作既能摆脱原来的郁结，又能擦出新思路的火花，对双方都大有裨益。

当然，我们的工作也有一些疏漏之处。在第二次线下讨论时，我们花费了大量时间来讨论一些细枝末节的问题，忽视了对游戏整体的把握。另外，我们在讨论时只是对讨论内容做了录音，尚未整理成文字材料，讨论时也忽略了讨论报告的整理，这使得工作进度的规划有一定的缓慢性；在整理讨论报告时，也因此而导致报告数量远小于讨论次数。在讨论分工时，我们一开始采取了先分模块写代码，最后一起整合的方式。但是在实际操作中发现，各个模块的编写难度是不同的，仅仅根据模块数量来分配工作任务是不合理的。这也导致了最初的三个星期里每个人都因模块编写无从下手而滞工。

后来组员常博宇提出按阶段分工的编写方式，即将游戏分为开始界面、游戏界面、结束界面三部分，在每个阶段中再进行分工，这样一来工作成效是随着工作开展就能即时看到的，也有利于提振信心。同时，组长焦笑然经过斟酌决定采用按状态管理游戏进程，使用全屏刷新的形式绘制游戏界面，这一理念的提出使得游戏的编写过程一下子明朗起来，为后续的拓展功能植入提供了非常好的条件。也正是在这一阶段，组员张知宇创新性地提出了游戏主体的绘制样式，不但解决了不易绘制的核心问题，而且使得游戏极具灵感与创造力。

随着游戏雏形的诞生，我们小组的工作进度更是如有神助。上午刚刚讨论出来的一个创意，往往是晚上就能用代码实现了。即便是后来发现了一些bug，我们也能够很快修复。在常博宇负责细节优化与创意提出、张知宇负责动画绘制与背景绘制、焦笑然负责逻辑修复与代码优化三强联合之下，我们的作品《Pac-Miner》不断推陈出新，用户体验不断提高。在此基础上，我们一鼓作气，不断开拓创新，使得游戏进入了蓬勃发展阶段。

其实，除了合作本身，我们小组在频繁的交流活动中也缔结了深厚的友谊。除了工作之外，我们时常约自习、约打卡、约聚餐，团队气氛始终保持在和睦的状态中，每个人都积极投入工作之中去，没有出现彼此推诿责任、争执不休的情况。到现在的阶段，对我们来说，作业本身已经不那么重要了，我们已经在这次合作中学会了如何与他人协作、如何管理团队工作中的事宜、如何将个人工作与合作紧密结合。我们相信，一份用心打造出来的产品，凝结着所有成员共同的心血，不论它收获了外界怎样的评价，都永远在我们心中熠熠生辉。

**个人心得**

**焦笑然：**选题之初，我们极少考虑实现的细节，只是想着选一个有趣的题目，就自己找了黄金矿工这个题目。当夏学期开始，我开始思考实现细节，才发现，不规则的矿和长相生动的矿工要求我们必须可以调用图片，自己画的话效果会大打折扣。于是，我开始了调用图片的探索之路。

最初的想法是找库，我试用了xege库，了解了其中调用图片的细节。ege库中可以创建一种PIMAGE类型，有getimage()和putimage()两个函数用于读入图片和贴图。那时，我的想法非常简单和幼稚，想用另一个图形库创建读取图片对象，用libgraphics库创建图形界面，将图片对象输出到lib库创建的图形界面上。助教的一席话点醒了我，lib库没有留相应的接口。且不说我们还遇到了许多问题，比如，ege库要求文件名后缀必须是.cpp，而lib库要求项目类型必须是c，否则就好出现bool类型的重定义问题。总而言之，这条路便走不通了。现在看来，当初放弃得有些早，后来了解到一点计算机图形学和windows图形编程，以及openGL库，在图书馆找到了相关书籍，发现绘图问题还是有很多种解决方案的。

后来在自定义颜色时，我发现DefineColor()函数是按RGB格式来定义颜色的，我就有了新的想法，既然大多数图片中信息都是按一个个像素点来存储的 ，那么，我们在画图的时候，也可以一个一个像素点画上去。我当时接触了openCV，就写了一段python代码读取png格式图片中每一个像素点的bgr信息，存储到一个txt文件中。（很多语言和软件都可以实现这一过程）*（参见附录注①）*

又在大程项目中封装了两个函数，drawDot和drawPic，drawDot的功能是画出一个指定颜色的像素点，drawPic的功能是逐一读取文件中每一个像素点的RGB信息，定义当前颜色并画出像素点。*（参见附录注②）*

其中也遇到一些困难，我还去研究过lib库中几个函数的实现方式。遇到最大的问题就是，我在重定义已有颜色时，lib库中colorTable中三种色度存储的值会改变，而画出了的颜色却不会改变，这令我非常郁闷。重定义颜色非常重要，否则，一张图片少说也有几万个像素点，每一个像素点定义一种新颜色，占有内存会非常大。以及，lib库中每次更改颜色都用的是一个名为FindColorIndex的线性搜索函数，如果不能重定义颜色，程序会越跑越慢。lib库的实现部分在我看来写得并不是很优雅，留的接口不多，改动空间不大。其中画颜色的部分竟全部是由一个PenColor的int类型全局变量来控制，而PenColor的值是cindex赋的，意义是在当前颜色在colorTable中的下标。感谢常博宇和张知宇同学积极的讨论，我们猜想是绘图部分在判断PenColor和oldColor相等后，即当前颜色在颜色表中的下标等于旧颜色的下标时，就不对画图颜色进行更改，不管此时colorTable中的RGB值是否改变。解决方案有二，一改库，而不再用一个颜色来存当前画图颜色，二是采取寄存器或者缓存的思想，开一个小数组或者多设几个变量来存储当前颜色，避免重定义后立即调用的问题。我们采取了后者的思想，常博宇同学改写并封装了DrawPic函数，问题得到了解决，可以优雅地在lib界面上绘制图片。

最终由于读取图片速度不够快，而我们的程序又采用的是整体刷新，我们摒弃了调用图片的想法，转而把注意力放在了画图优化界面上。

**张知宇：**两个月的大程让我真正领会到了C的魅力。无论是一开始的手足无措，还是最后关头的效率飞升，很感激我拥有着如此强大、包容的两个队友，愿意手把手地教会我基本的常识与技能，让我成长为可以不脱他们后腿的勉强合格的伙伴。

本次小组任务中，我主要承包了钩子和界面美化的工作，如何在现有的“简陋”条件下让我们的大程看起来不那么丢人成为了缠绕我整整半个学期的难题。做出最为粗糙的0.0.1，写出最为简陋的抓取伸缩判定函数，一步步地将一条直线换成吃豆人，将白板背景换成高贵暗纹，改按钮，做开机动画……每一次小组讨论都迸发出了无限的思想火花，哪怕只是观察队友的程序也头一次真正地将课堂所讲落地生根。若说这是一场踉跄行走的旅行，那么所踩下的每一个脚印都是发着光的。

谢谢常老师和焦焦肯拖着带我飞翔！大程，赛高！

**常博宇：在这次的大程编写中，我担任的主要是排行榜文件初步管理、背景音乐嵌入、报告整理以及代码规范化工作。对我来说，这次的分工合作不仅仅是对自己一年来学习的一个综合应用，还是一个同他人合作的绝佳机会。在一次次的讨论中，我收获了许多原来没有注意过的小知识，比如extern的使用位置、feof调用时不移动文件指针、enum、static、switch三连击等等。对于原来不怎么接触的链表也有了进一步的理解。**

**如果让我用一个词来表达我的体会，我会用“感谢”一词。感谢组长焦笑然高屋建瓴的程序设计思路，为我们的大程编写提供了灵活的框架，感谢组员张知宇妙手偶得的灵光一现，实现了游戏风格的全面创新，感谢她们两个人不顾自己的午睡，顶着烈日来到无空调的安中大楼二楼，积极参加每次小组讨论，感谢遇到两位开朗的队友，让每一次的线下讨论都是在一种欢快高效的气氛中进行。**

# 8、附录

注①：

import cv2

import numpy

import pylab

import matplotlib.pyplot as plt

img = cv2.imread("4.png")#+imgfile,cv2.IMREAD\_GRAYSCALE D:\CS\项目-goldminer\图片\

print("图像的形状,返回一个图像的(行数,列数,通道数):",img.shape)

print("图像的像素数目:",img.size)

print("图像的数据类型:",img.dtype)

#----------------------------------------------------------------------------

"""

In windows the COLOR->GRAYSCALE: Y = 0.299R+0.587G+0.114B 测试是否三个通道的值是相同的。

某些图像三通道值相同，可以直接读灰度图来代替读单一通道。

"""

# sum = 0

# ans = 0

# for i in range(562):

# for j in range(715):

# if not(img[i][j][0] == img[i][j][1] and img[i][j][1] == img[i][j][2]):

# sum += 1

# else:

# ans += 1

# print(ans)

# print(sum)

#-----------------------------------------------------------------------------

"""

将图片数据写入txt文件

格式:

基础信息

行号:

像素值

行号:

像素值

......

"""

fname = open("4.txt",'w')

fname.write(str(img.shape)+'\n')#----1

#fname.write("图像的像素数目:"+str(img.size)+'\n')#----2

#fname.write("图像的数据类型:"+str(img.dtype)+'\n')#----3

Xlenth = img.shape[1]#图片列数

Ylenth = img.shape[0]#图片行数

a = 1#----4

for i in range(Ylenth):

#fname.write(str(a) + ':'+'\n')#----5

for j in range(Xlenth):

fname.write(str(img[i][j][0])+' ')

fname.write(str(img[i][j][1])+' ')

fname.write(str(img[i][j][2])+' ')

a += 1#----6

#fname.write('\n')

fname.close()

#---------------------------------------------------------------------------

"""

将txt文件中的数据读取进blist

并显示为"test"图片框进行测试。

注意进行测试前需要注释掉数据写入模块

中标记的六行代码，要不然读取会出错误。

"""

# blist = []

# split\_char = ' '

# with open('C:/Users/Jake/Desktop/test01/'+txtfile, 'r') as bf:

# blist = [b.strip().split(split\_char) for b in bf]

#

##从txt文件读入进来的值类型是char需要转换为int

# for i in range(Ylenth):

# for j in range(Xlenth):

# blist[i][j] = int(blist[i][j])

#

# tlist = numpy.array(blist)

# plt.figure()

# plt.imshow(tlist)

# plt.axis('off') # 不显示坐标轴

# pylab.show()

#------------------------------------------------------------------------------

"""

将图片显示在'image'图片框

"""

cv2.imshow('image',img)

cv2.waitKey(0)

cv2.destroyAllWindows()

#----------------------------------------------------------------------

注②：

/\*

\* 函数名：DrawDot

\*

\* 用法：DrawDot(red,green,blue,x,y,size,n)

\*

\* 功能：在(x,y)处画一个边长为size的正方形。

\*

\* 备注：当size很小时，视为一个点。它的颜色是(R,G,B)。

\*

\*/

void DrawDot(double red,double green,double blue,double x,double y,double size)

{

static flag=0;

switch (flag){

case 0:{

flag++;

break;

}

case 1:{

flag++;

break;

}

case 2:{

flag=0;

break;

}

}

static char \*a[]={"a","b","c"};

DefineColor(a[flag],red,green,blue);

SetPenColor(a[flag]);

StartFilledRegion(1);

MovePen(x,y);

DrawLine(size,0);

DrawLine(0,size);

DrawLine(-size,0);

DrawLine(0,-size);

EndFilledRegion();

}

/\* 函数名：DrawPic

\*

\* 用法：DrawPic(name,x,y)

\*

\* 功能：以(x,y)为左上角画一个图片，这个图片的名字为name

\*

\* 备注：

\*

\*/

void DrawPic(char a[],double x,double y){

freopen(a,"r",stdin);

int i,j;

int width,height;

int num=0;

int degree;

int red\_,green\_,blue\_;

scanf("(%d,%d,%d)",&width,&height,&degree);

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

{

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

{

scanf("%d",&blue\_);

scanf("%d",&green\_);

scanf("%d",&red\_);

DrawDot(red\_/255.0,green\_/255.0,blue\_/255.0,x+i\*0.01,y-j\*0.01,0.01);

}

}

}

# 9、参考文献和资料

[1] 快捷调色 <https://color.hailpixel.com/>

[2] 色彩查询 <http://www.flatuicolorpicker.com/>

[3] 背景音乐 <https://blog.csdn.net/qq_21466543/article/details/60337040>

[4] 背景音乐<https://blog.csdn.net/lindorx/article/details/78724518>