

**程 序 设 计 专 题**

**大 程 序 报 告**



**2019~2020春夏学期 2020 年 5 月 26 日**

**报告撰写注意事项**

1. 图文并茂。文字通顺，语言流畅，无错别字。
2. 书写格式规范，排版良好，内容完整。
3. 存在拼凑、剽窃等现象一律认定为抄袭；0分
4. 蓝色文字为说明，在最后提交的终稿版本，请删除这些文字。

**目 录**

[1 大程序简介 4](#_Toc32875113)

[1.1 背景及意义 4](#_Toc32875114)

[1.2 目标要求 4](#_Toc32875115)

[1.3 术语说明 4](#_Toc32875116)

[2 功能需求分析 4](#_Toc32875117)

[3 程序开发设计 5](#_Toc32875118)

[3.1 总体架构设计 5](#_Toc32875119)

[3.2 功能模块设计 5](#_Toc32875120)

[3.3 数据结构设计 7](#_Toc32875121)

[3.4 源代码文件组织设计 1](#_Toc32875122)1

[3.5 函数设计描述 1](#_Toc32875123)3

[4 部署运行和使用说明 2](#_Toc32875124)2

[4.1 编译安装 2](#_Toc32875125)2

[4.2 运行测试 2](#_Toc32875126)3

[4.3 使用操作 2](#_Toc32875127)5

[5 团队合作 2](#_Toc32875128)8

[5.1 团队分工](#_Toc32875129) 28

[5.2 开发计划 2](#_Toc32875130)8

[5.3 编码规范 2](#_Toc32875131)8

[5.4 合作总结 2](#_Toc32875132)8

[5.5 收获感言 3](#_Toc32875133)0

[6 参考文献资料 3](#_Toc32875134)0

小型算法流程图绘制工具大程序设计

# 大程序简介

## 选题背景及意义

设计算法是程序设计的核心。而用流程图的形式表示算法，直观形象，易于理解。美国国家标准化协会ANSI曾规定了一些常用的流程图符号，为世界各国程序工作者普遍采用。

在此程序中，我们基于libgraphics，设计实现简单流程图的输入、编辑功能，并提供文件保存和读取功能，为算法流程图的绘制提供更方便、规范的环境与平台。

## 目标要求

1. 设计实现基本流程图的图形绘制
2. 设计实现基本流程图的图形编辑
3. 设计实现文件读取以及相关操作：包括新建、打开、保存、关闭、退出等
4. 设计链表，用于系统数据的组织
5. 设计实现背景音乐支持

# 功能需求分析

1. **菜单系统：**

创建四个菜单，包含“文件、编辑、绘图、帮助”四大模块。 其中“文件”模块包含“新建、打开、保存、关闭、退出”功能；“编辑”模块包含“选择、复制、粘贴、删除”功能；“绘图”模块包含“矩形、圆角矩形、菱形、箭头线条”等功能；“帮助”模块包含“关于本软件、使用方法”等功能；

1. **图标工具栏：**

设置菜单中常用功能/命令，包括保存、打开、删除、剪切、复制、粘贴、旋转、箭头形态、填充颜色、字体颜色

1. **图形绘制工具栏：**

绘制图形，包括“矩形、圆角矩形、菱形、箭头、线条”等多种图形的绘制，包括填充、颜色等属性，图形框架支持文字输入编辑。

1. **快捷键**：

设置常用操作的快捷键，文件的三个操作对应相应快捷键：“新建|

Crtl-N”， “打开| Ctrl-O”，“保存| Ctrl-S”。图形编辑三个操作对应响应快捷键：“剪切| Crtl-X”， “复制| Ctrl-C”，“粘贴| Ctrl-V”

1. **状态信息栏：**

窗口底部，显示填充状态、填充颜色、字体颜色、背景音乐、文本编辑、选中状态等

1. **链表：**

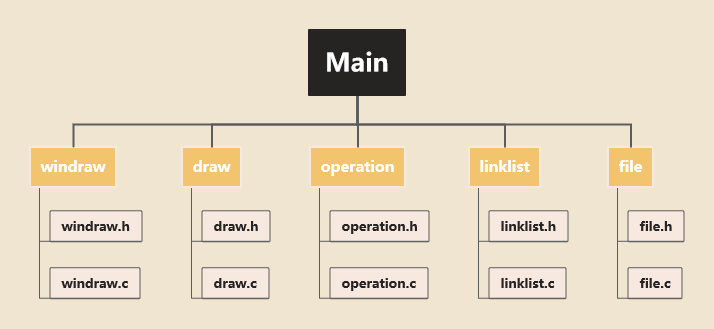
设计图形元素链表来储存每一种图形元，每个节点分别代表对应的一个图元

1. **文件：**

采用多文件组织，设计保存图元信息到文件；设计从文件中读取图元

# 程序开发设计

## 总体架构设计



## 功能模块设计

1. **菜单系统**

1.1 文件（File）

* + 1. 新建（New） 新建空白文件，对其进行绘制、编辑操作
    2. 打开（Open） 打开用户选中文件，对其进行绘制、编辑操作
    3. 保存（Save） 覆盖之前的文件
    4. 关闭（Close） 关闭当前文件
    5. 退出（Exit） 退出程序

1.2 编辑（Edit）

1.2.1 剪切（Cut） 剪切选中图形

1.2.2 复制（Copy） 复制选中图形

1.2.3 粘贴（Paste） 粘贴所复制的图形

1.2.4 删除（Delete） 删除选中的图形

1.3 绘制（Draw）

1.3.1 矩形（rectangle） 绘制矩形

1.3.2 圆角矩形（roundrec） 绘制圆角矩形

1.3.3 菱形（diamond） 绘制菱形

1.3.4 直线（line） 绘制直线

1.3.5 单向箭头（directional connection） 绘制单向箭头

1.3.6 双向箭头（bidiretional connection）绘制双向箭头

1.3.7 虚线（dashline） 绘制虚线

1.3.8 平行四边形（process） 绘制平行四边形

1.3.9 圆（circle） 绘制圆

1.3.10 椭圆（oval） 绘制椭圆

1.4 帮助（Help）

1.4.1 关于我们（About us）

团队与程序基本介绍，可选择前往github界面获得更多内容

1.4.2 快捷键列表（Short cuts list）

给出copy、paste相关快捷键，可以选择前往主页查看其余快捷键

1.4.3 其他（Guard） 当用户无法使用本程序时，通过前往主页解决相关问题

**2. 图标工具栏：菜单常用操作**

2.1保存（Save）

2.2 打开（Open）

2.3 删除（Delete）

2.4 剪切（Cut）

2.5 复制（Copy）

2.6 粘贴（Paste）

2.7 顺时针旋转（Clockwise Rotate）

2.8 逆时针旋转（AntiClockwise Rotate）

2.9 箭头形态（Arrowstyle） 箭头形态，有直线箭头和图形箭头可供选择

2.10 填充颜色（Fill colors） 选择或者改变填充颜色，默认为黑色

2.11 字体颜色（Front colors） 选择或者改变字体颜色，默认为黑色

**3.图形绘制工具栏**

3.1 矩形（rectangle） 绘制矩形

3.2 圆角矩形（roundrec） 绘制圆角矩形

3.3 菱形（diamond） 绘制菱形

3.4 直线（line） 绘制直线

3.5 单向箭头（directional connection） 绘制单向箭头

3.6 双向箭头（bidiretional connection）绘制双向箭头

3.7 虚线（dashline） 绘制虚线

3.8 平行四边形（process） 绘制平行四边形

3.9 圆（circle） 绘制圆

3.10 椭圆（oval） 绘制椭圆

**4.状态信息栏**

4.1 填充状态（fill&unfill） fill为填充 unfill不填充 默认不填充

4.2 背景音乐支持（silent&music on） silent静音 music on有音乐 默认静音

4.3 文本编辑处 编辑所选中图形的响应文本

4.4 填充颜色（Fill color）显示当前图形的填充颜色

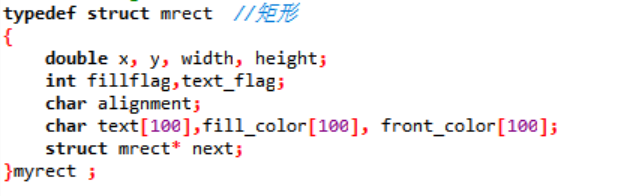
4.5 字体颜色（Front color）显示当前图形的字体颜色

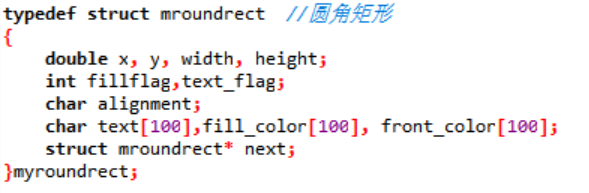
4.6 操作状态 显示当前操作状态，用于观察是否选中

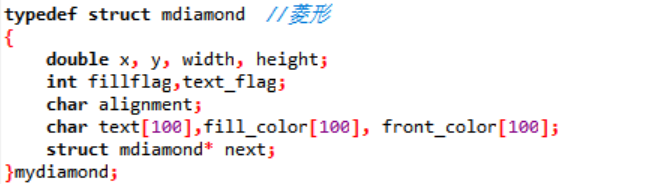
## 数据结构设计

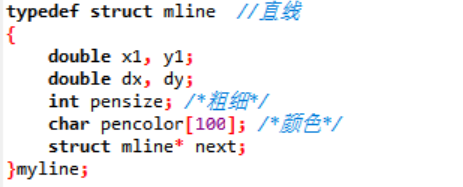
**一、图形相关**

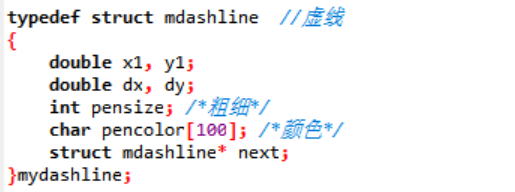
1. 图形结构变量

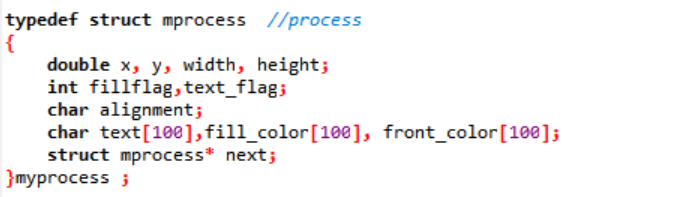


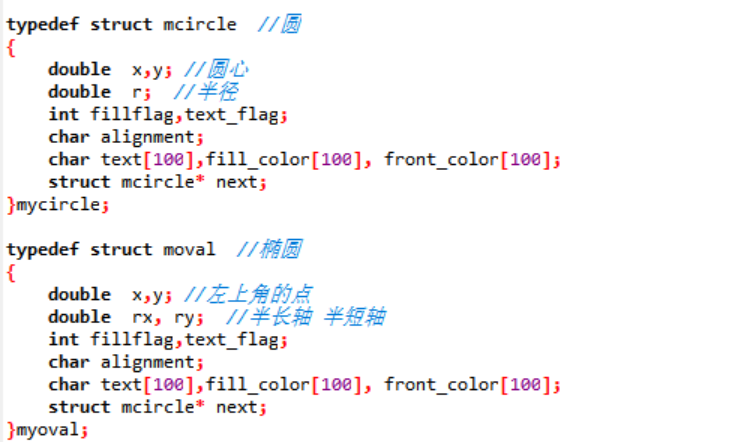


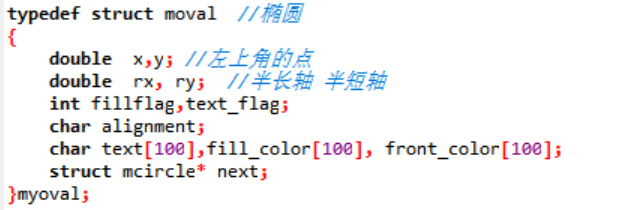












1. 图形链表头指针（以矩形为例，其余图形类似）



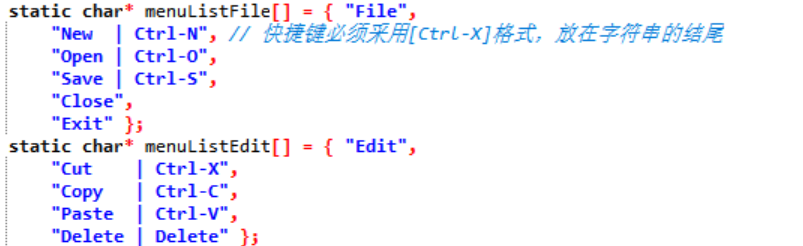
1. 选中图形指针变量（以矩形为例，其余图形类似）

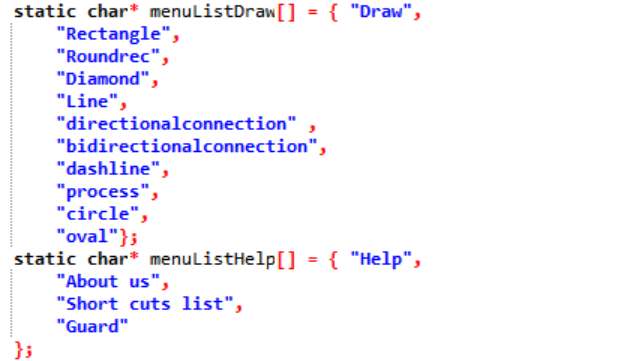


1. 文件变量



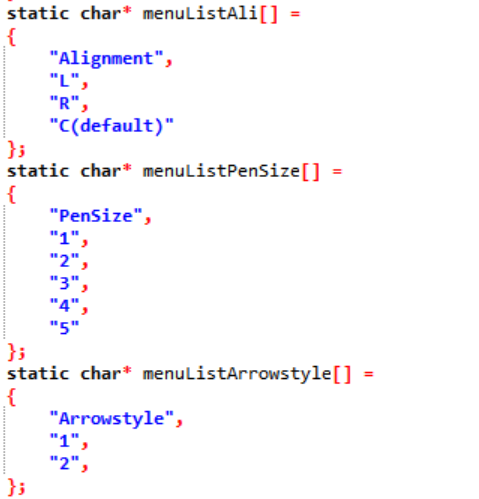
1. **菜单相关（包括各类菜单）**





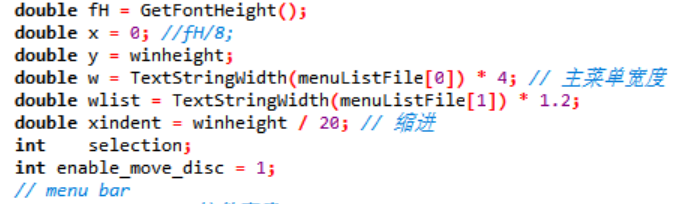








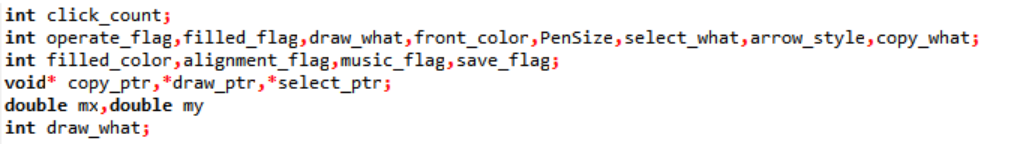
1. **初始化相关参数设定**



1. **交互式相关变量**



1. **绘制相关**



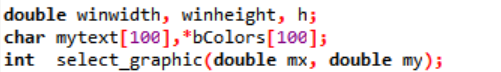
## 源代码文件组织设计

<文件目录结构>

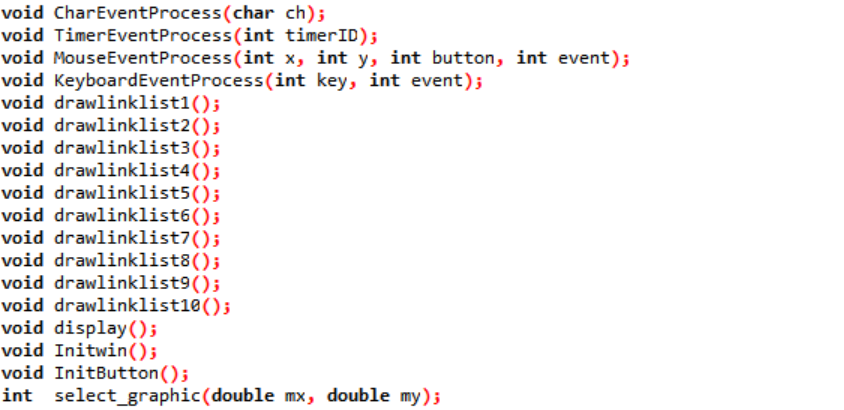
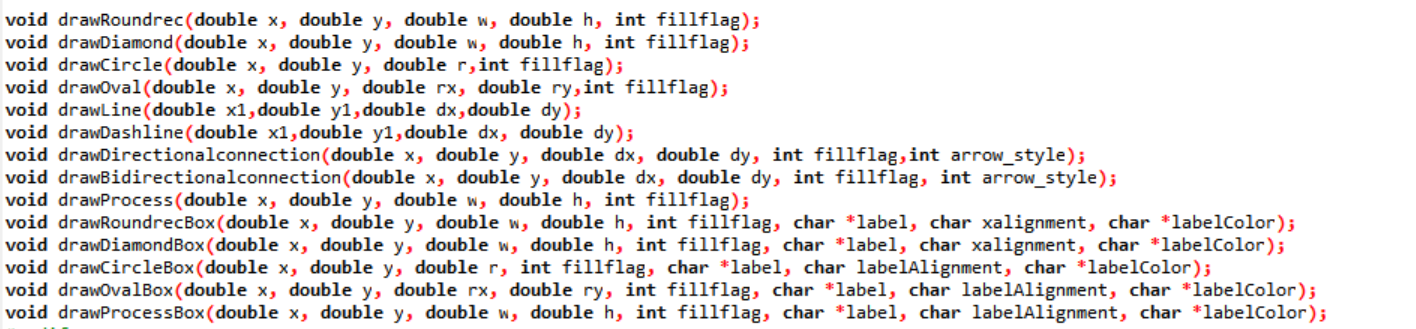
1. **文件函数结构**

**Main.c** 定义：

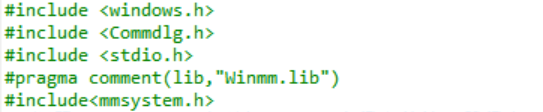
**windraw.h** 包含：windraw.c中定义的变量及函数的声明



**windraw.c** 定义：

**file.h**  包含：

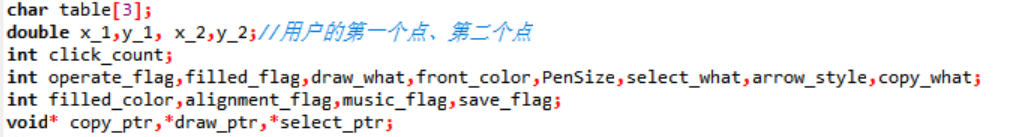


及file.c定义函数的声明

**file.c**  定义：

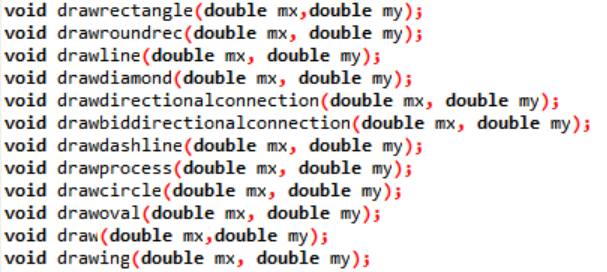


**draw.h**  包含：相关参数定义



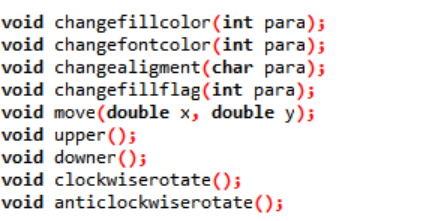
以及draw.c定义函数的声明

**draw.c** 定义：



**operation.h** 包含：operation.c定义函数的声明

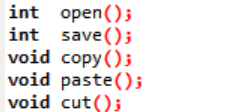
**operation.c** 定义：与图形编辑相关函数

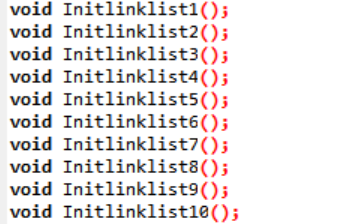


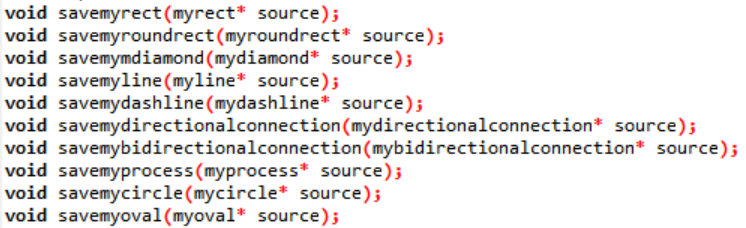
**linklist.h** 包含：不同图形元素结构变量的定义、图形链表头指针定义、选中图形指针变量定义、文件指针定义（详见数据结构模块1、2、3）

以及linklist.c所定义的函数的声明

**linklist.c** 定义：







（以矩形为例，其他图形元素类似）

**2）多文件构成机制**

1. 所有头文件中均使用#define保护，如draw.h中：



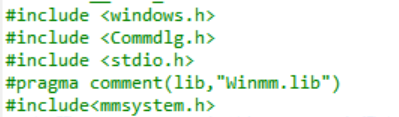


1. 部分头文件采用其他头文件包含

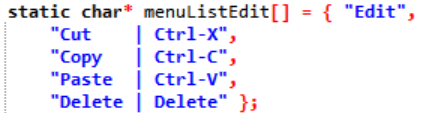
如operation.h中包含linklist.h

linklist.h与windraw.h中包含图形库文件

file.h中包含



1. 菜单栏采用static声明变量，保护其不被其他文件引用 如：



## 函数设计描述

1. **文件相关函数：**

**A）文件打开函数：**

函数原型：int FileOpenDialog(char\* path);

功能描述：弹出文件会话框，使用户打开文件

**参数描述：char\* path**

**返回值描述：int整型变量，0和1**

重要局部变量定义：OPENFILENAME ofn;

重要局部变量用途描述：用于指代所选中的文件，进行打开相关操作

函数算法描述：将结构大小、路径、路径大小、文件类型等赋值给ofn相关参数，并引用GetOpenFileName函数，来获取返回值，打开成功返回1，打开失败 返回0

**B）文件保存函数：**

函数原型：int FileOpenDialog(char\* path);

功能描述：弹出文件会话框，使用户保存文件

**参数描述：char\* path**

**返回值描述：int整型变量，0和1**

重要局部变量定义：OPENFILENAME ofn;

重要局部变量用途描述：用于指代所选中的文件，进行打开相关操作

函数算法描述：将结构大小、路径、路径大小、文件类型等赋值给ofn相关参数，并引用GetSaveFileName函数，来获取返回值，打开成功返回1，打开失败 返回0

**2.图形绘制相关函数**

**A）图形绘制入口函数：**

函数原型：void draw(double mx,double my)

功能描述：传入图形入口相关参数，判断所需绘制的图形类型，

并调用相应函数绘制图形

**参数描述：double mx,double my**

**返回值描述：无**

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：通过select\_what的值判断所选中图形的类型，并通过switch分支结构传递位置参数给相应的绘图函数，以此进行图形的绘制。

**B）图形绘制出口函数：**

函数原型： void drawing(double mx, double my)

功能描述：作为draw函数的出口，实时显示图形

**参数描述：double mx,double my**

**返回值描述：无**

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：作为draw函数的出口，通过select\_what的值判断所选中图形的类型，并通过switch分支结构传递位置参数给相应的绘图函数，实时显示绘制的图形。

**C）图形参数传入函数：（以矩形为例，其他图形类似）**

函数原型：void drawrectangle(double mx,double my)

功能描述：传入图形相关参数

参数描述：double mx,double my

返回值描述：无

重要局部变量定义：myrect\* a=malloc(sizeof(myrect))

重要局部变量用途描述：指向所绘制的图形

函数算法描述：通过传入参数给图形结构变量，再使用 pushrect函数将其压入链表中。

**3.图形编辑相关函数**

**A）图形移动函数：**

函数原型：void move(double x, double y)

功能描述：移动选中的图形到指定位置

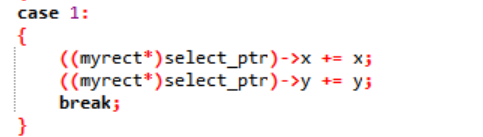
**参数描述：double x, double y**

**返回值描述：无**

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：通过select\_what的值判断所选中图形的类型，并通过switch分支结构进行不同图形的移动，主要通过改变图形的位置参数实现，以下为移动矩形的例子



**B）图形放大函数：**

函数原型：void upper()

功能描述：等比例放大图形

**参数描述：无**

**返回值描述：无**

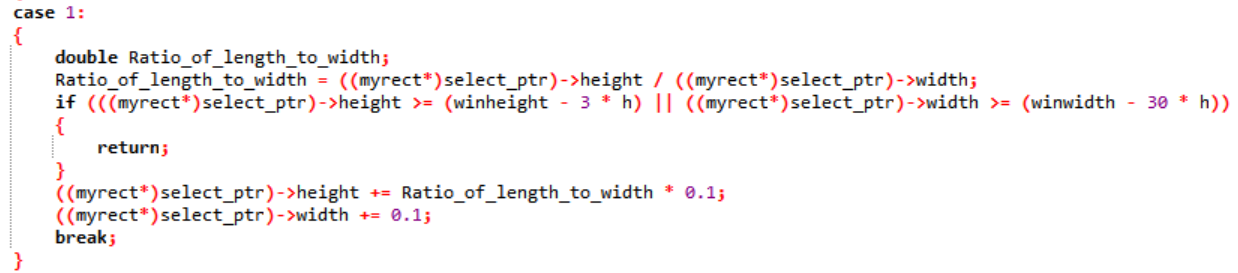
重要局部变量定义：double Ratio\_of\_length\_to\_width;

double Ratio\_of\_dy\_to\_dx;

double Ratio\_of\_ry\_to\_rx;

重要局部变量用途描述：用来控制图形的相关比例，使其等比例放大

函数算法描述：通过select\_what的值判断所选中图形的类型，并通过switch分支结构进行不同图形的放大，主要通过等比例改变图形与大小相关的参数来实现，不同图形参数不同，以下为放大矩形的例子



**C）图形缩小函数：**

函数原型：void downer()

功能描述：等比例缩小图形

**参数描述：无**

**返回值描述：无**

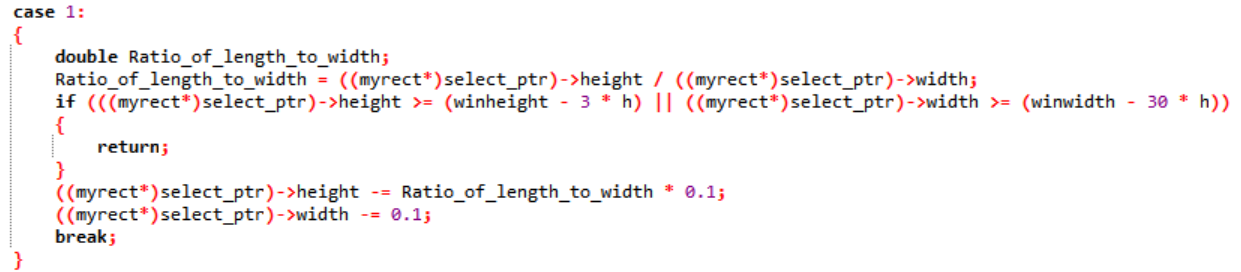
重要局部变量定义：double Ratio\_of\_length\_to\_width;

double Ratio\_of\_dy\_to\_dx;

double Ratio\_of\_ry\_to\_rx;

重要局部变量用途描述：用来控制图形的相关比例，使其等比例缩小

函数算法描述：通过select\_what的值判断所选中图形的类型，并通过switch分支结构进行不同图形的缩小，主要通过改变图形与大小相关的参数来实现，不同图形参数不同，以下为缩小矩形的例子



**D）图形颜色改变函数：**

函数原型：void changefillcolor(int para)

功能描述：改变图形颜色

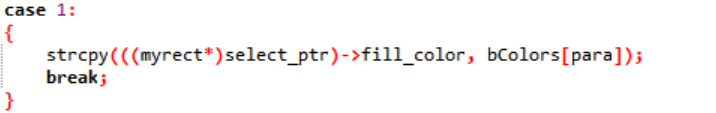
**参数描述：int para**

**返回值描述：无**

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：通过select\_what的值判断所选中图形的类型，并通过switch分支结构进行不同图形的颜色改变，主要通过改变图形与颜色相关的参数来实现，不同图形参数不同，以下为改变矩形颜色的例子



**E）字体颜色改变函数：**

函数原型：void changefontcolor(int para)

功能描述：改变字体颜色

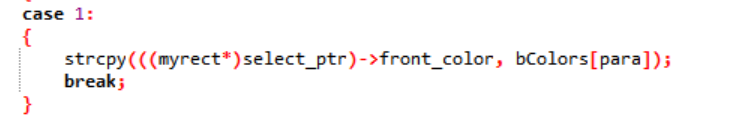
**参数描述：int para**

**返回值描述：无**

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：通过select\_what的值判断所选中图形的类型，并通过switch分支结构进行不同图形的字体颜色改变，主要通过改变图形与字体颜色相关的参数来实现，不同图形参数不同，以下为改变矩形字体颜色的例子



**F）字体对齐方式改变函数：**

函数原型：void changealigment(char para)

功能描述：改变字体对齐方式

**参数描述：char para**

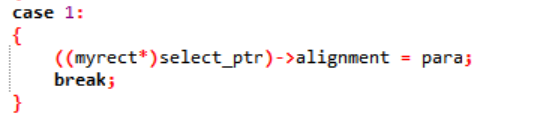
**返回值描述：无**

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：通过select\_what的值判断所选中图形的类型，并通过switch分支结构进行不同图形的字体对齐方式的改变，主要通过改变图形与字体对齐方式相关的参数来实现，不同图形参数不同，以下为改变矩形字体对齐方式的例子

\*对齐方式包括左对齐、右对齐、中间对齐



**G）填充状态改变函数：**

函数原型：void changefillflag(int para)

功能描述：改变图形填充状态

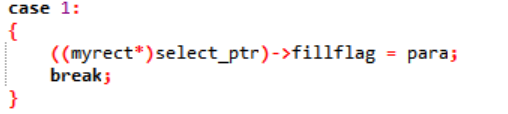
**参数描述：int para**

**返回值描述：无**

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：通过select\_what的值判断所选中图形的类型，并通过switch分支结构进行不同图形的填充状态的改变，主要通过改变图形与填充状态的相关的参数来实现，不同图形参数不同，以下为改变矩形填充状态的的例子



**H）图形顺时针旋转函数：**

函数原型：void clockwiserotate（）

功能描述：顺时针旋转图形

参数描述：无

返回值描述：无

重要局部变量定义： double angle = 10; double pi = 3.14159265;

double width0 = ((myrect\*)select\_ptr)->width;

double height0 = ((myrect\*)select\_ptr)->height;

重要局部变量用途描述：代表角度和圆周率，记录图形初始长宽

函数算法描述：根据select\_what进行分支，改变图形元长宽相关参数。

**I）图形逆时针旋转函数：**

函数原型：void anticlockwiserotate（）

功能描述：逆时针旋转图形

参数描述：无

返回值描述：无

重要局部变量定义： double angle = 10; double pi = 3.14159265;

double width0 = ((myrect\*)select\_ptr)->width;

double height0 = ((myrect\*)select\_ptr)->height;

重要局部变量用途描述：代表角度和圆周率，记录图形初始长宽

函数算法描述：根据select\_what进行分支，改变图形元长宽相关参数。

**4.窗口相关函数**

**A）初始化窗口函数：**

函数原型：void Initwin()

功能描述：初始化窗口

参数描述：无

返回值描述：无

重要局部变量定义：见**3.3数据结构设计：二、菜单相关**

**三、初始化相关参数设定**

重要局部变量用途描述：略

函数算法描述：设置初始化参数，运用menuList创建相关菜单，并在File、Edit、Draw、Help、填充颜色、字体颜色、对其、箭头样式等菜单下通过selection的值设计分支结构，对应相应的指令，调用相关函数实现功能。

**B）初始化按钮函数：**

函数原型：void InitButton()

功能描述：初始化按钮

参数描述：无

返回值描述：无

重要局部变量定义：见**3.3数据结构设计：三、初始化相关参数设定**

重要局部变量用途描述：略

函数算法描述：设置初始化参数，设置相关按钮，并通过图标工具栏、图形绘制工具栏、状态信息栏不同按钮的值设计分支结构，对应相应的指令，调用相关函数实现功能。并通过文本框输入函数实现文本输入。

**C）字符事件响应函数：**

函数原型：void CharEventProcess(char ch)

功能描述：编辑显示字符

参数描述：char ch

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：调用uiGetChar(ch)实现字符输入

调用display()刷新显示

**D）键盘事件响应函数：**

函数原型：void KeyboardEventProcess(int key, int event)

功能描述：键盘事件响应

参数描述：无

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：调用uiGetKeyboard(key, event)获取键盘

**E）鼠标事件响应函数：**

函数原型：void MouseEventProcess(int x, int y, int button, int event) 功能描述：鼠标事件响应

参数描述：int x, int y, int button, int event

返回值描述：无

重要局部变量定义：double mx, my;

重要局部变量用途描述：获取鼠标当前位置

函数算法描述：调用display()刷新显示，调用uiGetMouse(x, y, button, event)获取鼠标，并通过不同event和不同button设置双重分支结构，对应相应的指令，调用相关函数实现功能。并通过文本框输入函数实现文本输入。

**F）计时器时间响应函数：**

函数原型：void TimerEventProcess(int timerID)

功能描述：计时器时间响应

参数描述：int timerID

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：通过判断save\_flag 和timerID 的值来将save\_flag置0

便于后续程序退出时的保存判断

调用display()刷新显示

**G）图形链表函数（以矩形为例，其他图形类似）：**

函数原型：void drawlinklist1()

功能描述：循环输出所绘制的矩形图元

参数描述：无

返回值描述：无

重要局部变量定义：myrect\* p = rect\_head->next

重要局部变量用途描述：指向当前所要输出的图形

函数算法描述：调用SetPenColor（）、drawBox（）绘制当前所指向的图形元， 调用while循环输出所有矩形

**H）选中图形函数：**

函数原型：int select\_graphic(double mx,double my)

功能描述：判断并确定选中的图元

参数描述：double mx,double my

返回值描述：0为未选中， 1为选中

重要局部变量定义：myrect\* p = rect\_head->next

（以矩形为例，其他图形类似）

重要局部变量用途描述：指向图形链表中的元素，并不断循环

函数算法描述：根据鼠标所在位置，判断其是否在图元范围内，并调用while循环判断每张链表中是否有选中图形

**I）图元绘制函数：（以圆角矩形为例，其他图形类似）**

函数原型：void drawRoundrec(double x, double y, double w, double h,

int fillflag)

功能描述：绘制圆角矩形

参数描述：double x, double y, double w, double h,

int fillflag

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：调用MovePen、DrawLine、DrawEllipticalArc等相关函数绘制圆角矩形

**J）图元绘制函数（支持文本框输入）：（以圆角矩形为例，其他图形类似）**

函数原型：void drawRoundrecBox(double x, double y, double w,

double h, int fillflag, char \*label, char xalignment,

char \*labelColor);

功能描述：绘制圆角矩形，并支持文本框输入

参数描述：double x, double y, double w,

double h, int fillflag, char \*label, char xalignment,

char \*labelColor

返回值描述：无

重要局部变量定义：double fa = GetFontAscent()

重要局部变量用途描述：字体大小对应比例相关参数

函数算法描述：调用drawRoundrec绘制圆角矩形，根据labelAlignment的值设计字体对齐方式，并调用DrawTextString函数输出相应字符

**5.链表相关函数**

**A）压入图形函数：（以矩形为例，其他图形类似）**

函数原型：void pushrect(myrect\* source)

功能描述：将一个矩形压入链表

参数描述：myrect\* source

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：通过链表常规操作，将矩形压入链表

**B）初始化矩形链表函数：（以矩形为例，其他图形类似）**

函数原型：void Initlinklist1()

功能描述：初始化矩形链表

参数描述：无

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：链表初始化操作

**C）图元弹出函数：**

函数原型：void pop()

功能描述：将选中的图元从对应的链表中弹出

参数描述：无

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：通过删除链表节点的操作弹出对应图元

**D）图元链表保存函数：（以矩形为例，其他图形类似）**

函数原型：void savemyrect(myrect\* source)

功能描述：保存矩形图元链表到文件中

参数描述：myrect\* source

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：通过文件相关操作保存链表

**E）图形复制函数：**

函数原型：void copy()

功能描述：复制所选中的图形

参数描述：无

返回值描述：无

重要局部变量定义：myrect\* a = malloc(sizeof(myrect))等

重要局部变量用途描述：指代复制所得到的图形

函数算法描述：根据select\_what设计分支结构，用来复制不同的图元

**F）图形粘贴函数：**

函数原型：void paste()

功能描述：粘贴所复制的图形

参数描述：

返回值描述：无

重要局部变量定义：copy\_ptr

重要局部变量用途描述：指向粘贴的图元

函数算法描述：根据copy\_what设计分支结构，用来粘贴不同的图元

**G）图形剪切函数：**

函数原型：void cut()

功能描述：剪切选中图形

参数描述：无

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：略

函数算法描述：使用copy()函数复制后，使用pop()函数从将其链表中弹出

**H）文件打开函数：**

函数原型：int open()

功能描述：打开文件

参数描述：无

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：

函数算法描述：首先根据save\_flag的值判断是否保存文件，并询

问用户是否保存以及保存路径，再调用Initlinklist1()等函数初始化，

并运用文件函数，再加载所有图元链表。

**I）文件保存函数：**

函数原型：int save()

功能描述：保存文件

参数描述：无

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：

函数算法描述：调用文件相关函数，运用图元链表保存函数，将图元保存至文件中。

# 部署运行和使用说明

## 编译安装

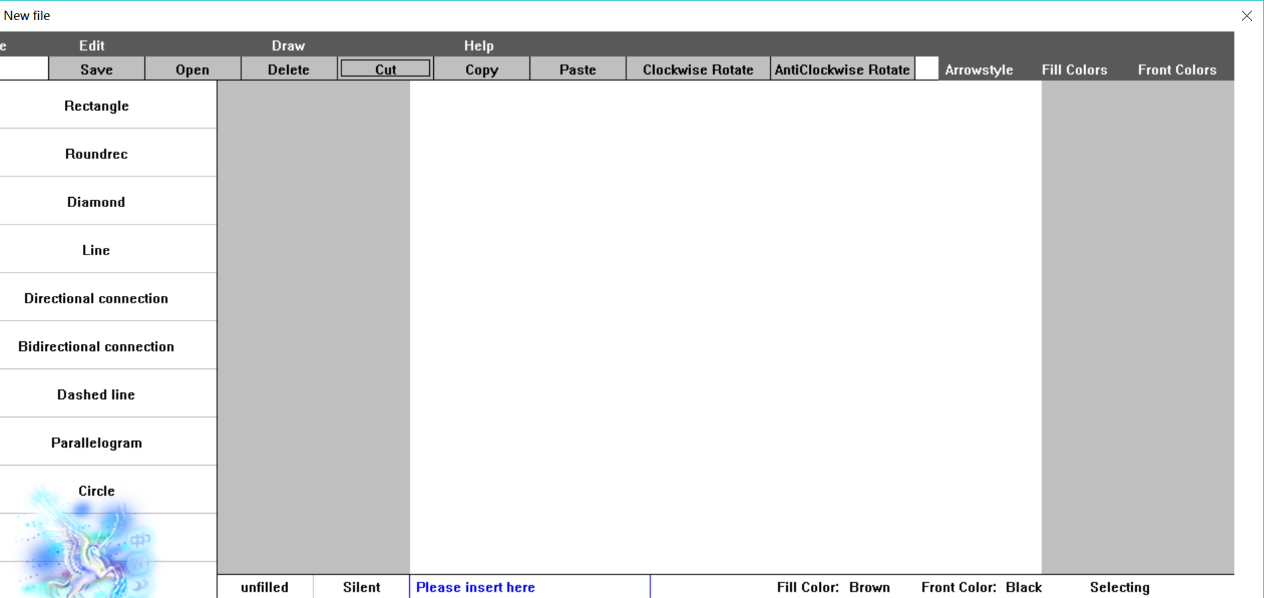
**vs2019：**  
最直接的方式是将项目源代码直接复制到democode文件夹中，然后打开老师所给得demoprj-vs2017,将解决方案中源代码栏目下所有文件删除，然后添加现有项，将所给得源代码加入到解决方案中即可编译运行。  
**dev:**  
将项目源代码复制到democode文件中，打开demoprj-devc文件中的demo.dev文件，右键移除项目管理中的demo.c，再右键demo选择添加，把复制到democode中的项目源代码全部添加到项目中。将如下位置的代码进行注释。  
windraw.c 557行、566行   
main.c 49行  
最后进行编译运行即可。

*注：dev需要使用utf-8格式，否则会出现乱码。若依然存在乱码，则用记事本打开乱码所处的文件后，全选粘贴到乱码文件进行保存即可。*

***注意：***运行时请注意将附带得音乐文件Soldout.mp3放在同一文件夹下  
请注意，源代码中使用了mciSendString函数用于播放背景音乐，该函数在dev环境下会找不到定义，因此相关代码已经做了注释处理。建议在vs环境下编译运行源代码，以便所有得功能得以展示。注释得代码得位置如下:  
windraw.c 557行、566行   
main.c 49行

## 运行测试

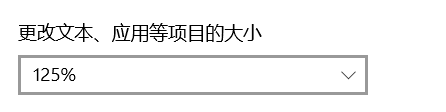
1. 程序在显示设置150%时出现如下情况：



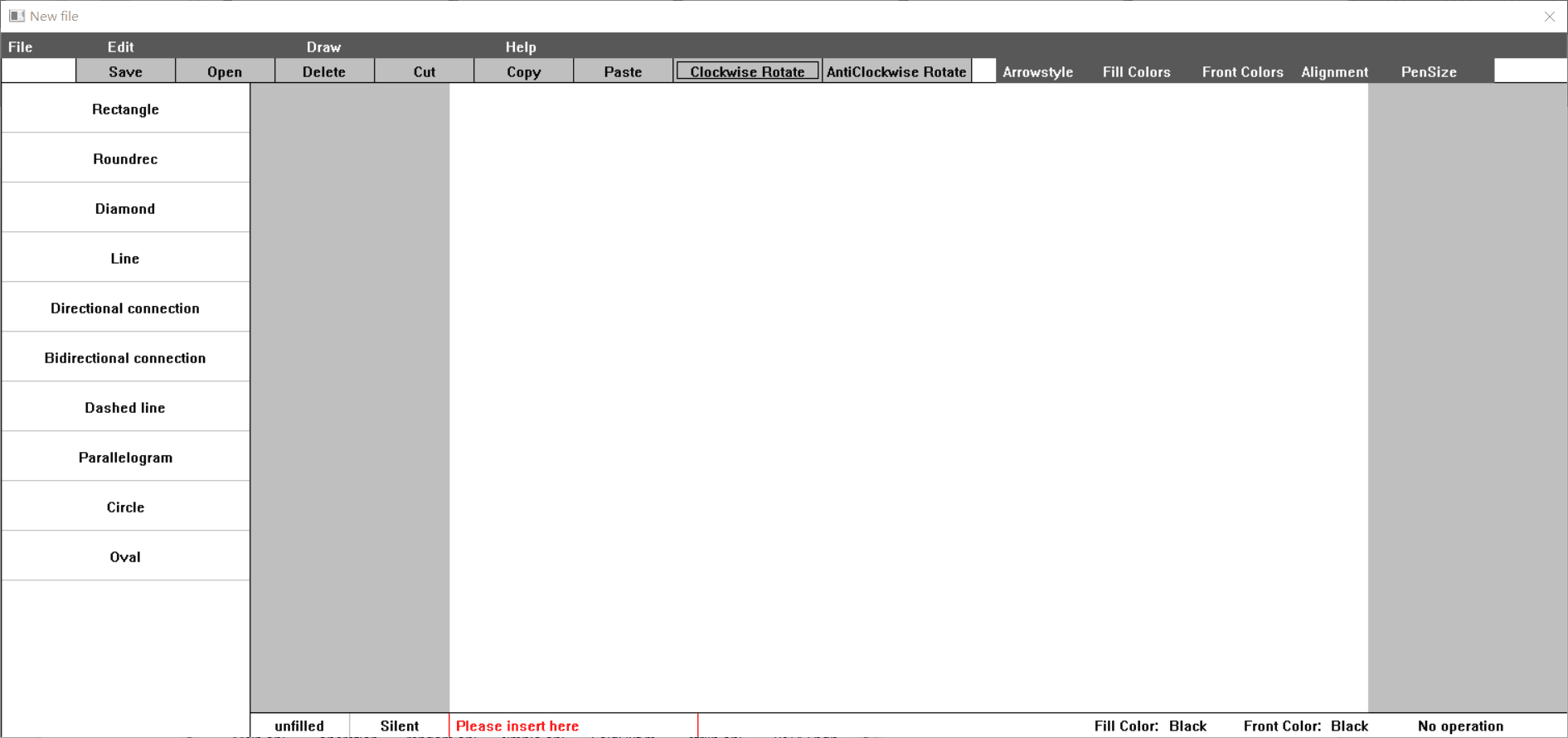
即图标工具栏最右侧有部分未显示

改进：需要调整电脑显示设置“更改文本、应用等项目的大小”等为125%

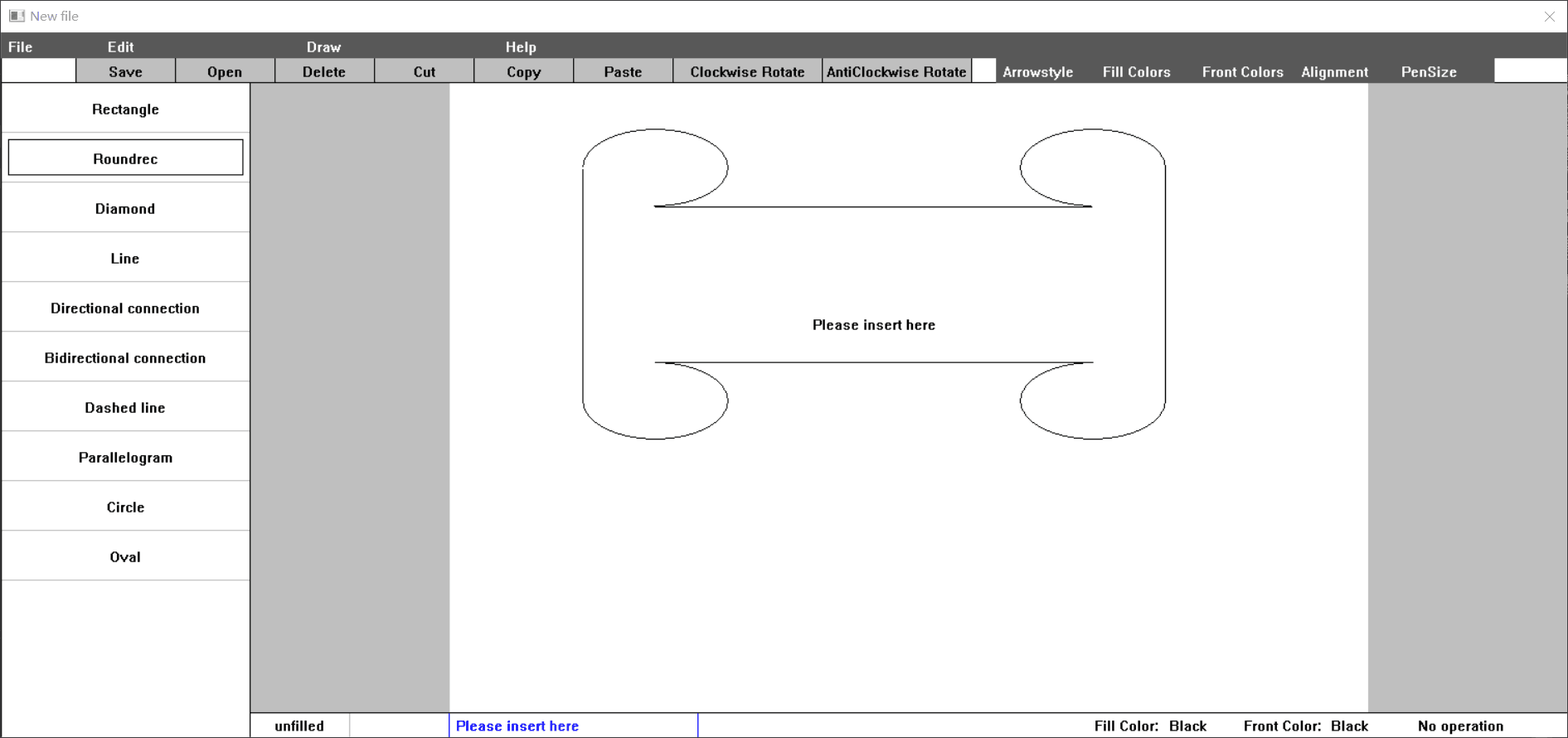
如下图：

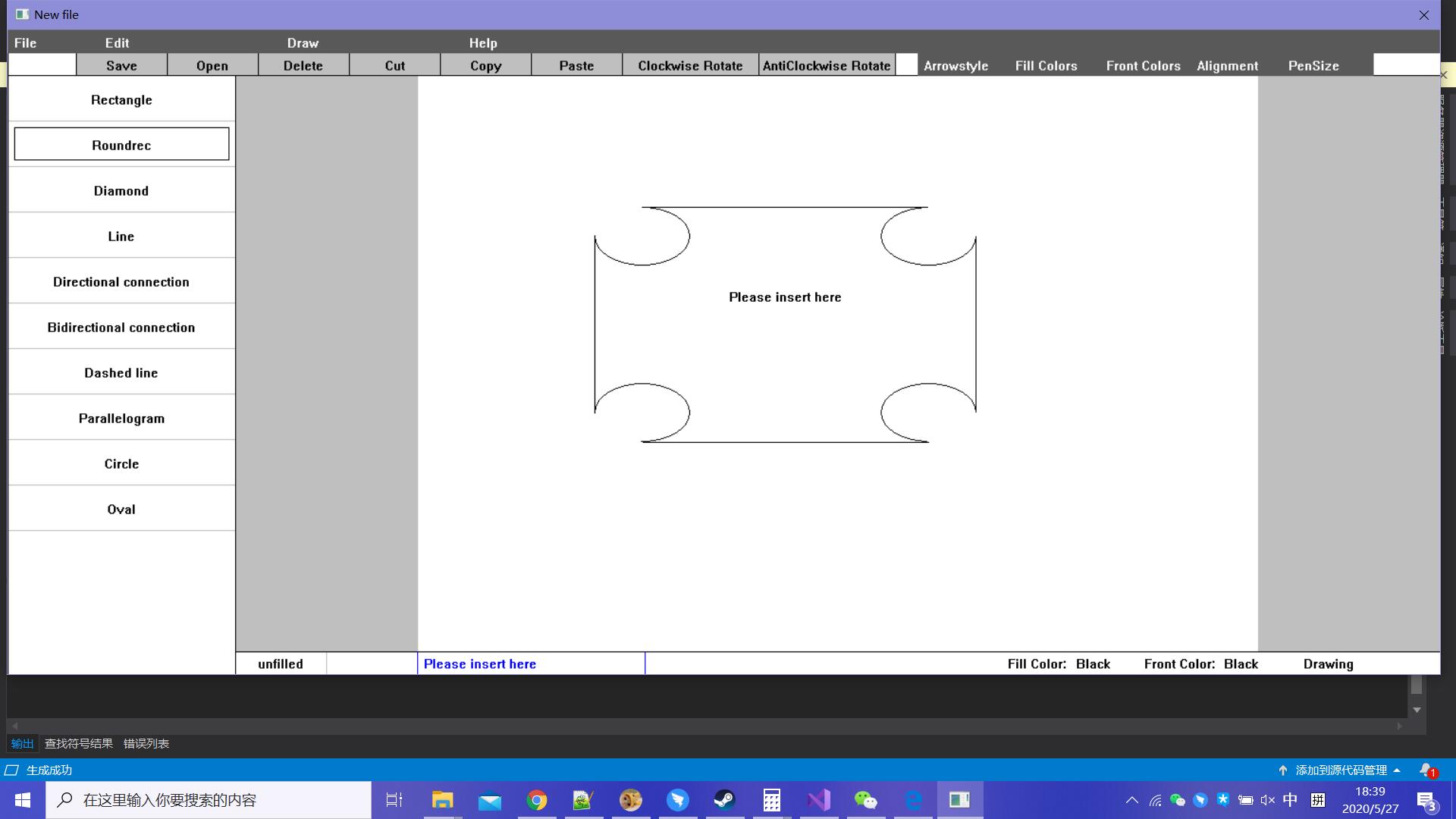


调整后如下：

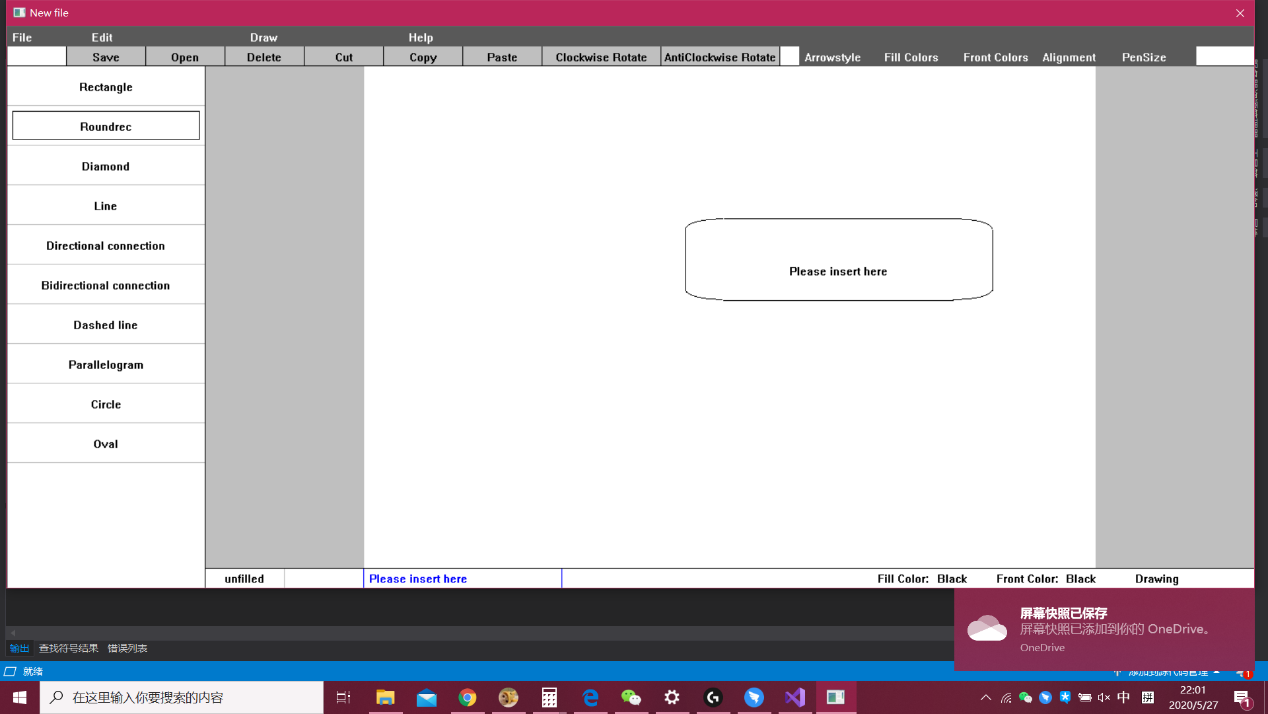


1. 绘制圆角矩形时出现如下情况：边角处不符合要求

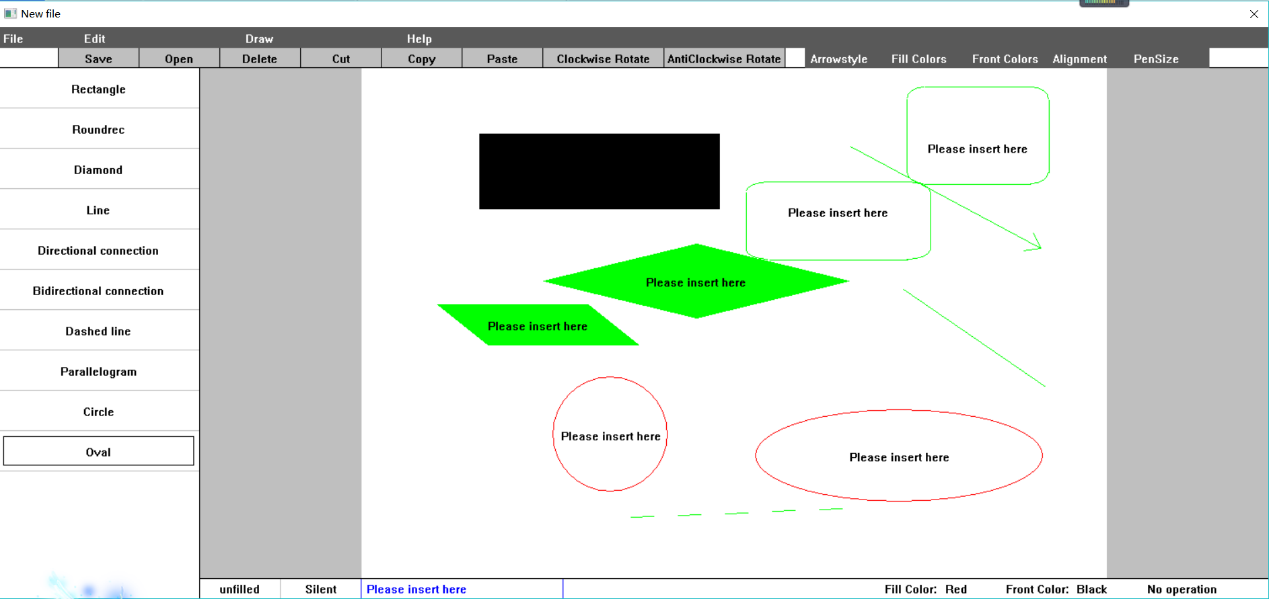




改进：分类讨论绘制边角后改进如下：



1. 典型数据测试

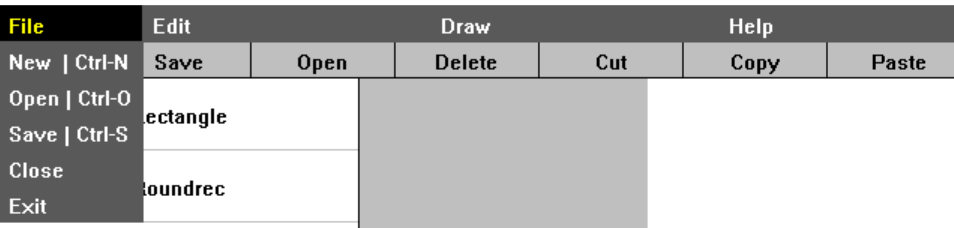


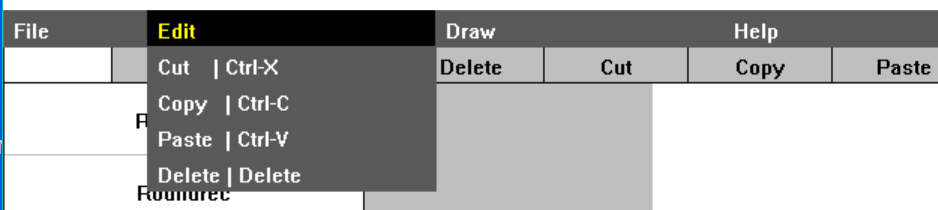
## 使用操作（详细版本见用户使用手册）

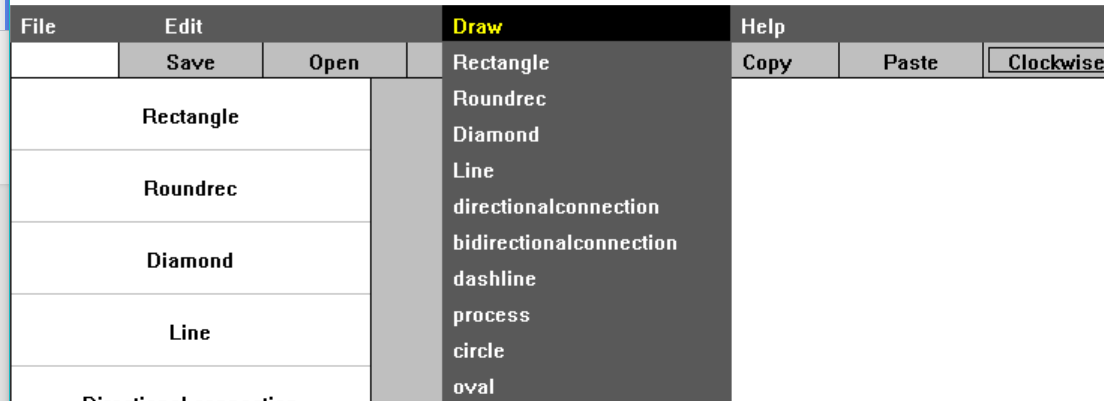
**1.菜单系统：**

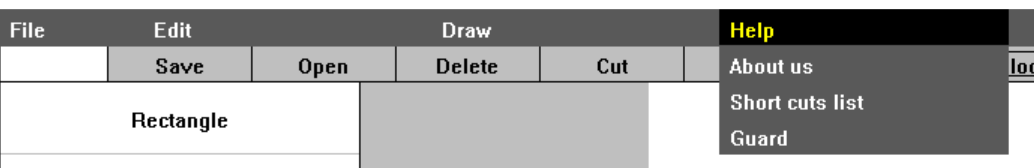
四个菜单，包含“文件、编辑、绘图、帮助”四大模块。 其中“文件”模块包含“新建、打开、保存、关闭、退出”功能；**鼠标左键点击**相关菜单即可。

“编辑”模块包含“选择、复制、粘贴、删除”功能，**鼠标左键点击**，选中某一图形元后（右下角选中状态，选中则变为selecting），**左键点击**所要进行的操作菜单即可实现功能；“绘图”模块包含“矩形、圆角矩形、菱形、箭头线条”等功能，点击所需画的图形后在中央画布中绘制；“帮助”模块包含“关于本软件、使用方法”等功能；



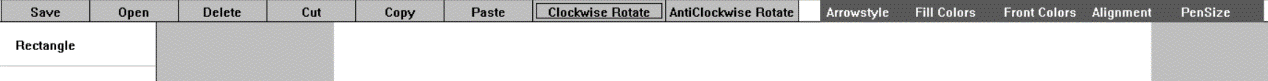






**2.图标工具栏：**

菜单中常用功能/命令，包括保存、打开、删除、剪切、复制、粘贴、旋转、箭头形态、填充颜色、字体颜色



**3.图形绘制工具栏：**

绘制图形，包括“矩形、圆角矩形、菱形、箭头、线条”等多种图形的绘制，包括填充、颜色等属性，图形框架支持文字输入编辑。绘制图形的时候：**鼠标首先左键**点击一个点，作为起始点，然后松开，拉到第二个点处**再次进行鼠标左键**点击，即可绘制成功。**左键点击**选中图形后，**滚动滚轮**可以进行缩放。



**4.快捷键**：

常用操作的快捷键，文件的三个操作对应相应快捷键：“新建|

Crtl-N”， “打开| Ctrl-O”，“保存| Ctrl-S”。图形编辑三个操作对应响应快捷键：“剪切| Crtl-X”， “复制| Ctrl-C”，“粘贴| Ctrl-V”

**5.状态信息栏：**

窗口底部，显示填充状态、填充颜色、字体颜色、背景音乐、文本编辑、选中状态等，可根据需要点击进行调整



**6.中央画布**

相关图形的绘制



# 团队合作

## 任务分工（互评版略）

## 开发计划

1.分析功能需求，制定分配相关任务

2.构建程序大框架

3细节处理

4.后期完善

## 编码规范

**可读性、可维护性和可变更性**

1.避免使用不易理解的数字，用有意义的标识来替代。

2.重要变量、函数处提供相应注释。

3.代码尽量简洁易懂

## 合作总结（注意匿名）

**开发亮点：**

1.背景音乐支持

2.箭头不同形态的选择

**挑战点：**

1.图形的旋转操作

2.圆角矩形的绘制

**应用知识点：**

1.图形编程知识

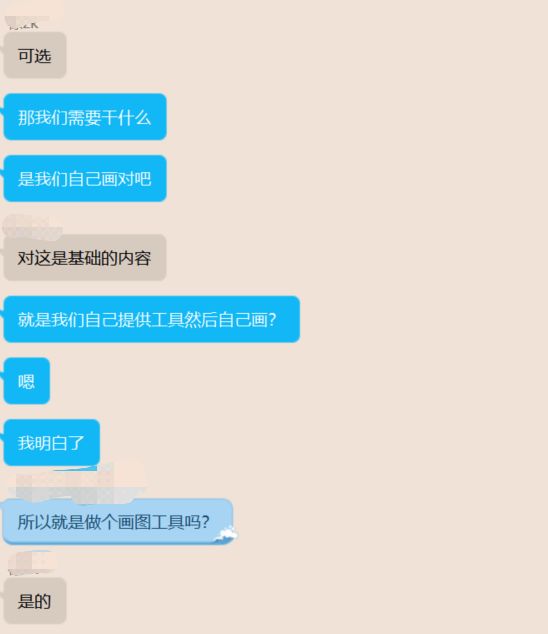
2.链表知识

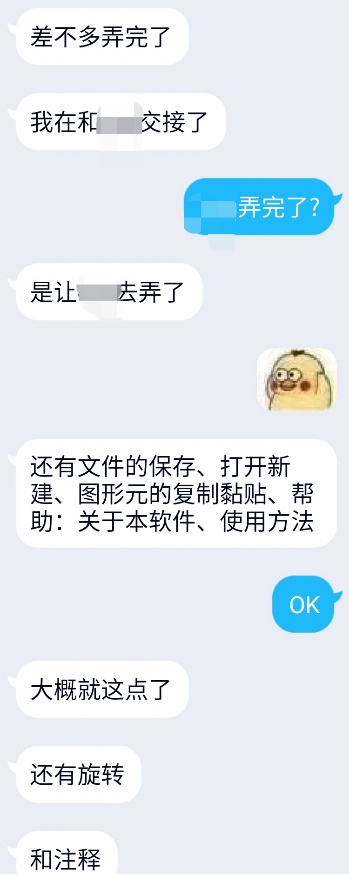
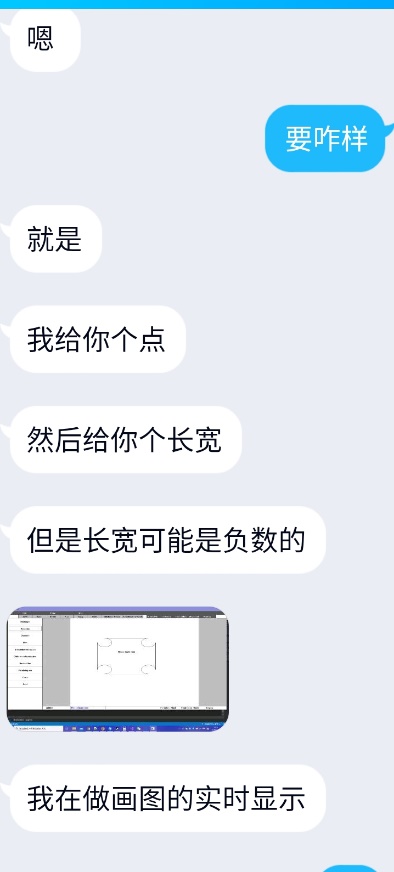
3.文件知识

**合作记录（附图）：**

1.2020年5月8日：小组会议，功能需求分析，进行大框架搭建

1. 2020年5月9日：大框架建立，后续任务分配完毕
2. 2020年5月15日：菜单系统等完成，图形绘制部分启动
3. 2020年5月21日：图形绘制基本完成，图形编辑启动
4. 2020年5月24日：图形编辑完成
5. 2020年5月27日：文件模块完成
6. 2020年5月30日：总体整合
7. 2020年6月3日：初稿完成





## 收获感言（注意匿名）

组长：我分配到的任务是在框架搭好之后再进行的，所以我在程序制作的前期并没有太过积极地参与其中。这就导致我对整个程序的组成的了解不是特别透彻，使我在完成我的任务时遇到一些困难。好在，有组内同学的帮助，对于我提出的一些问题（有些问题甚至略显白痴）都能够做出耐心详细的解答，也给我介绍了关于图形如何形成的设计原理，同时也帮我把我的粗糙的代码进行了多次调试和完善。真的很感谢这位同学！就我个人而言，我感觉我分配到的任务量似乎还是挺小的，大头都由组内同学帮忙完成了，感觉心有愧疚。不过，在这些任务的完成中，我也对文件的创建读写等相关操作进行了学习回顾，对图形旋转的原理有了一些认知，也对团队的合作分工有了一定的认识。总而言之，收获还是挺多的，再次感谢组内同学的付出与帮助!

组员1：我分配到的任务较少，主要负责图形的绘制。可以说是在前面那位同学的框架下进行。但我发现大程不同于pta，需要较强的整体意识。另外我们的选题算法流程，也需要我自行学习相关知识。万事开头难，一开始我无从下手，后来找到了方向，才逐渐发现了图形编程的乐趣。可以说在此过程中，我写的代码虽然不多，但是学到了很多知识。看到最后的也非常感谢另一位组员，耐心解答我的疑惑。可以说，我们小组间的沟通是非常高效的，大家的积极性都非常高。很高兴程设课给了我这样一次机会，让我遇到一起肝ddl的组员们！

组员2：作为组员，首先感谢全组另外两位同学积极承担责任，主动协调配合，使得我们整个过程即便遇到许多困难最终也非常流畅。我主要负责图形界面的编写调试的实现，与其他两位同学的任务有较大的不同，因此切身体会到了分工合作的独特意义。在大工程项目面前一个人的力量往往微不足道，但合而为一则能造就不同。

# 参考文献资料

1. <https://www.csdn.net/>
2. <https://github.com/>