

浙江大学

程序设计专题

大程序报告



1. 姓名： 郑德钦 学号： 3190101117 电话： 18150960639
2. 姓名： 黄嘉欣 学号： 3190102060 电话： 19922051831
3. 姓名： 袁致远 学号： 3190106085 电话： 19817864896

指导老师： 吴春明

2019~2020 春夏学期 2020 年 6 月 19 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
1.1	背景及意义	4
1.2	目标要求	4
1.3	术语说明	4
2	功能需求分析.....	4
3	程序开发设计.....	5
3.1	总体架构设计	5
3.2	功能模块设计	5
3.3	数据结构设计	6
3.4	源代码文件组织设计.....	6
3.5	函数设计描述.....	11
4	部署运行和使用说明	28
4.1	编译安装	28
4.2	运行测试	28
4.3	使用操作	29
5	团队合作.....	32
5.1	任务分工	32
5.2	开发计划	32
5.3	编码规范	33
5.4	合作总结	33
5.5	收获感言	38
6	参考文献资料.....	39

小型算法流程图绘制工具大程序设计

1 大程序简介

1.1 选题背景及意义

流程图，顾名思义，是用来直观描述一个工作具体步骤图形表达方式。一张简明的流程图，能够帮助我们理清思路、查漏补缺，还能促进开发者、决策者等之间的交流，避免功能流程、逻辑上出现遗漏或错误，从而确保流程的正确性。通过绘制流程图，我们可以让思路更清晰，逻辑更清楚，有助于程序的逻辑实现，从而有效解决问题。

基于上述原因，我们小组决定自己动手设计一款算法流程图绘制工具，用简洁的形式、完整的框架、多元的功能帮助使用者完成流程图的绘制。相信，通过这种思维上的整理与外化，使用者能够获得一个更加轻松、愉悦的工作体验，能够事半功倍地完成流程项目。

1.2 目标要求

- 1、具有完整的软件结构和功能（包括工具栏、快捷键、背景音乐）
- 2、能够实现简单流程图的绘制
- 3、能对已有流程图进行编辑
- 4、文件、数据结构规范，能提供文件的保存和读取功能

1.3 术语说明

Easyx: EasyX Library for C++ 是针对 VC 的一套绘图库,接口简单易用,用起来很像 TC 的 graphics.h 绘图。

2 功能需求分析

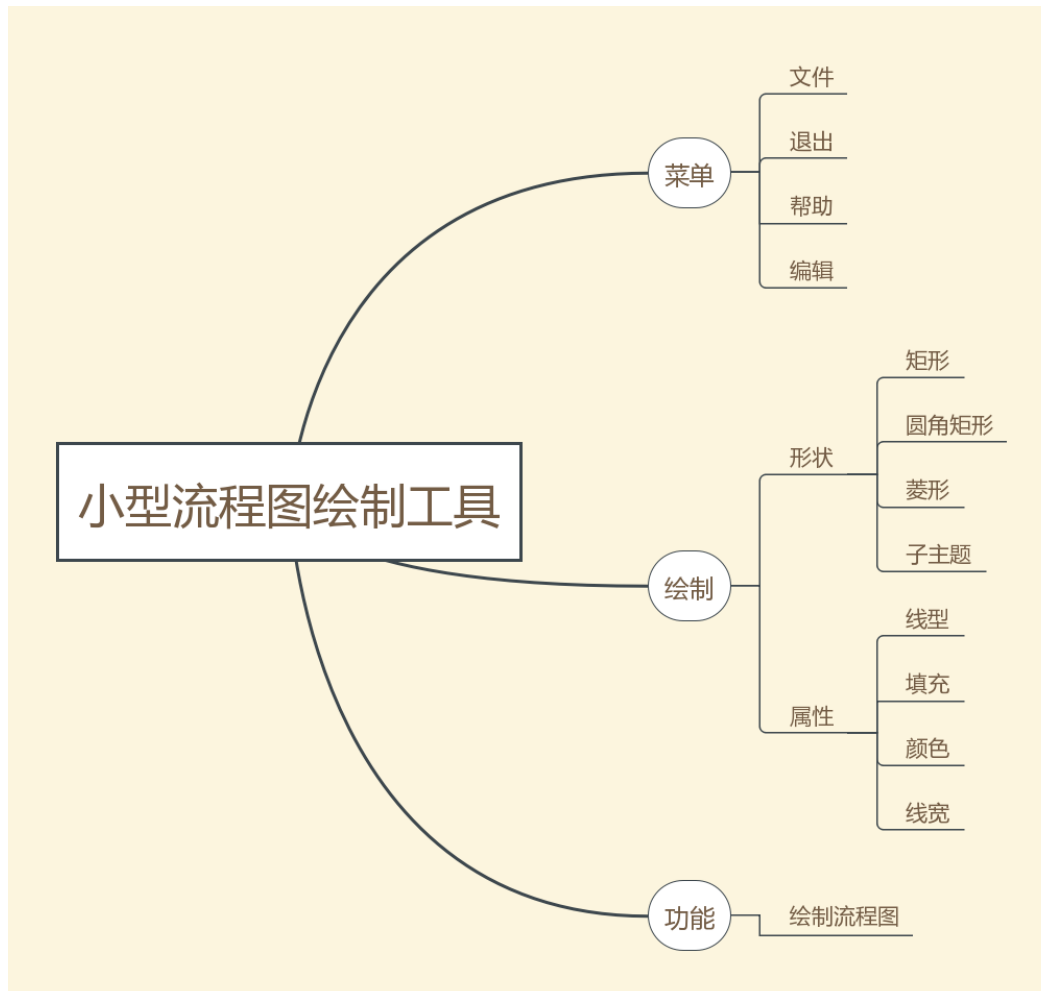
功能具备：绘制程序设计的流程图，助力初学者更好地学习程序语言。同时，在日常生活中，可以使用此软件绘制简单的流程图。

功能要求：多种图形选择，多元图形属性确定，对已经绘制出的图形，可以进行简单的编辑。

性能要求：简洁美观、方便易用。

3 程序开发设计

3.1 总体架构设计



3.2 功能模块设计

功能具备：绘制并保存流程图，具备流程图应有的基础的图形，可在需要的基础上自由发挥，如颜色、线型、线宽等等。对已经绘制出的图形，可以进行简单编辑，如旋转、放缩等。

3.3 数据结构设计

文件指针: file *fp;

链表: struct rec_coor {

int lx;

int ly;

int rx;

int ry;

int style; //1 为矩形, 2 为圆角矩形, 3 为菱形, 4 为箭头

int chcol; //颜色参数

int chwid; //线宽参数

int chosl; //线型参数

int chofl; //填充颜色

char str[500]; //输入字符串

struct rec_coor* next;

};

3.4 源代码文件组织设计

<文件目录结构>

1) 文件函数结构

头文件:

1、necessity.h:

基本库文件的引用, 内容:

```
#include <graphics.h>
```

```
#include <conio.h>
```

```
#include <windows.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <io.h>
```

```
#include <string.h>
```

2、Drawing_tool.h:

基本自定义函数的引用, 内容:

```
#include "necessity.h"
```

```
int position(int x, int y);
```

```
int file_pos(int x, int y);
```

```
void File(int x, int y);
```

```
void Edit(int x, int y);
```

```
void Paint(int x, int y);
```

```
void Help(int x, int y);
void redo(int x, int y);
void revoke(int x, int y);
```

源文件：（函数具体功能见 3.5 函数设计描述）

1、coor_find.cpp:

定义函数 `rec_coor* coor_find(int x, int y)`

2、copy_move and do.cpp:

定义函数 `int copy_do(copyxy* xy, rec_coor* copy)`、
`rec_coor* copy_move(copyxy* xy, rec_coor* copy)`

3、delet_do and paint.cpp:

定义函数 `int delet_do(rec_coor* delet)`、`rec_coor* move(rec_coor* m)`、
`void paint(rec_coor* d)`

4、Edit.cpp:

定义函数 `void Edit(int x, int y)`

5、Edit_Choose.cpp:

定义函数 `void Edit_Choose(int x, int y)`

6、Edit_Copy.cpp:

定义函数 `void Edit_Copy(int x, int y)`

7、Edit_Delet.cpp:

定义函数 `void Edit_Delet(int x, int y)`

8、Edit_Drag.cpp:

定义函数 `void Edit_Drag(int x, int y)`、`int drag_do(rec_coor* a, copyxy* b)`

9、edit_pos.cpp:

定义函数 `int edit_pos(int x, int y)`

10、Edit_Rotate.cpp:

定义函数 `void Edit_Rotate(int x, int y)`

11、Edit_Zoom.cpp:

定义函数 `void Edit_Zoom(int x, int y)`

12、Ekeybd.cpp:

定义函数 `int Ekeybd(int x, int y)`

13、rotate_do.cpp:

定义函数 `int rotate_do(copyxy* xy, rec_coor* rotate)`

14、`zoom do and move.cpp`:

定义函数 `int zoom_do(copyxy* xy, rec_coor* zoom)`

15、`File.cpp`:

定义函数 `void File(int x, int y)`

16、`File_Close.cpp`:

定义函数 `void File_Close(int x, int y)`

17、`File_New.cpp`:

定义函数 `void File_New(int x, int y)`

18、`File_Open.cpp`:

定义函数 `void File_Open(int x, int y)`

19、`file_pos.cpp`:

定义函数 `int file_pos(int x, int y)`

20、`File_Save.cpp`:

定义函数 `int File_Save(int x, int y)`

21、`Fkeybd.cpp`:

定义函数 `int Fkeybd(int x, int y)`

22、`loadpast.cpp`:

定义函数 `void loadpast(char* s)`

23、`Screen_File.cpp`:

定义函数 `void Screen_File(char* s)`

24、`Help.cpp`:

定义函数 `void Help(int x, int y)`

25、`Help_about.cpp`:

定义函数 `void Help_about(int x, int y)`

26、`help_pos.cpp`:

定义函数 `int help_pos(int x, int y)`

27、`Help_way.cpp`:

定义函数 `void Help_way(int x, int y)`

- 28、Hkeybd.cpp:
 定义函数 int Hkeybd(int x, int y)

- 29、main.cpp:
 定义函数 int main()

- 30、Mkeybd.cpp:
 定义函数 int Mkeybd(int x, int y)

- 31、music.cpp:
 定义函数 void music(int x, int y)

- 32、position.cpp:
 定义函数 int position(int x, int y)

- 33、redo.cpp:
 定义函数 void redo(int x, int y)

- 34、revoke.cpp:
 定义函数 void revoke(int x, int y)

- 35、stop_singing.cpp:
 定义函数 void stop_singing(int x, int y)

- 36、at_pos.cpp:
 定义函数 int at_pos(int x, int y)

- 37、bs_pos.cpp:
 定义函数 int bs_pos(int x, int y)

- 38、color.cpp:
 定义函数 void color(int x, int y)

- 39、color_pos.cpp:
 定义函数 int color_pos(int x, int y)

- 40、fill.cpp:
 定义函数 void fill(int x, int y)

- 41、fill_pos.cpp:
 定义函数 int fill_pos(int x, int y)

- 42、linestyle.cpp:
 定义函数 void linestyle(int x, int y)

- 43、Paint.cpp:
 定义函数 void Paint(int x, int y)

- 44、Paint_Arrow.cpp:
 定义函数 void Paint_Arrow(int x, int y)

- 45、Paint_Att.cpp:
 定义函数 void Paint_Att(int x, int y)

- 46、Paint_Bashp.cpp:
 定义函数 void Paint_Bashp(int x, int y)

- 47、Paint_Diamond.cpp:
 定义函数 void Paint_Diamond(int x, int y)

- 48、paint_pos.cpp:
 定义函数 int paint_pos(int x, int y)

- 49、Paint_Rectangle.cpp:
 定义函数 void Paint_Rectangle(int x, int y)

- 50、Paint_Rorec.cpp:
 定义函数 void Paint_Rorec(int x, int y)

- 51、Pkeybd.cpp:
 定义函数 int Pkeybd(int x, int y)

- 52、save_coor.cpp:
 定义函数
 void save_coor(int lx, int ly, int rx, int ry, int style, int chcol, int chwid, int chosl,
 int chofl, char* str)

- 53、style_pos.cpp:
 定义函数 int style_pos(int x, int y)

- 54、width.cpp:
 定义函数 void width(int x, int y)

- 55、width_pos.cpp:
 定义函数 int width_pos(int x, int y)

2) 多文件构成机制

在编写过程中，我们小组将功能函数独立出来，从而可使多个函数共同调用一个函数。对主要的图形数据，我们设置了多个全局变量以保存其属性、位置信息，例如 `chcol`（颜色）、`chwid`（线宽）等等，并将每个图形的这些参数值储存在一个链表 `head` 里，从而实现了数据的共享和多文件的互动。

3.5 函数设计描述

函数原型：int main()

功能描述：显示主页面，包括各个主选项按钮（文件、编辑、绘图等），以及音乐的播放、暂停，图标工具栏和主选项键盘快捷键

参数描述：无参数

返回值描述：返回值 0

重要局部变量定义：

- ① `int event = -1;`
- ② `RECT Ri (i=1, 2.....13)`
- ③ `int b = Mkeybd(mouse.x , mouse.y)`
- ④ `MOUSEMSG mouse`

重要局部变量用途描述：

- ① `event`：
用于接收 `position` 函数的返回值，判断鼠标移动、点击在哪个选项框或图标上。
- ② `Ri`：
保存 13 个选项框、图标或工具栏的坐标信息。
- ③ `b`：
接收 `Mkeybd` 函数的返回值，判断用户输入键盘消息时，是否按下了快捷键以及按下的快捷键为何键。
- ④ `mouse`：
接收鼠标事件返回的消息

函数算法描述：`main` 函数首先根据预先设置的坐标信息，绘制好整个程序的图形界面。在函数当中，采用了 `while(TRUE)` 死循环，在每一次循环里，函数将根据用户的鼠标坐标信息和键盘事件消息，判断出用户点击或按下的选项，从而进入相应的子函数当中，如 `File`、`Paint` 等，并实时更新下方状态栏，待子函数结束返回后，自动进入下一次循环。

函数原型：int Mkeybd(int x, int y)

功能描述：判断用户按下的键盘消息，并返回对应的整数值

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：整型返回值，根据不同的键盘消息，返回设定好的对应值

重要局部变量定义：char s

重要局部变量用途描述：s:用 getch() 函数，读取用户的按下的键钮，并将其保存到 s 当中，方便后面 switch 函数的选择。

函数算法描述：Mkeybd 函数通过 getch，读取用户键盘按下的字母，并为每个字母设定相应的返回值（如 'F' 或 'f' 对应 0）。经过 switch 函数的判断选择后，将该值返回到 main 函数中。

函数原型：void music(int x, int y)

功能描述：播放音乐

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：main 函数根据鼠标点击音乐键的次数，选择播放或暂停音乐。当选择播放音乐时，music 函数启动，其调用 PlaySound 函数，进行指定音乐的播放。

函数原型：void stop_singing(int x, int y)

功能描述：停止播放音乐

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：main 函数根据鼠标点击音乐键的次数，选择播放或暂停音乐。当选择暂停音乐时，stop_singing 函数启动，其调用 PlaySound 函数，取消对指定音乐的播放。

函数原型：int position(int x, int y)

功能描述：判断鼠标所处位置在哪个主选项框或图标之上

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：整型返回值，根据鼠标所处的不同位置，返回设定好的对应值

重要局部变量定义：int i

重要局部变量用途描述：i:既表示第 i 个选项框，也表示返回值 i（如第 0 个选项框的返回值对应为 0）。若不在选项框或图标上，则 i = -1，并作为返回值返回。

函数算法描述：position 函数通过 for 循环+if 判断，将鼠标的坐标信息与 main 里设置好的选项框坐标进行比较，由此判断出鼠标正在哪个选项框之上，并返回相应的 i 值。

函数原型：void revoke(int x, int y)

功能描述：撤销键，可撤销上一次绘图动作

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：revoke 函数中引用了外部全局变量 last_lx, last_ly, last_rx, last_ry（上一次画图时的绘图区坐标信息），在调用 revoke 函数时，其会根据上一次画图时产生的图形信息，在上次绘图区域使用 clearrectangle 函数，进行指定区域的图形清除，并将其在链表和文件中的内容擦除。

函数原型：void redo(int x, int y)

功能描述：重做键，可重做上一次撤销的图形

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：redo 函数中引用了外部全局变量 last_lx, last_ly, last_rx, last_ry（上一次画图时的绘图区坐标信息），在调用 redo 函数时，其会根据上一次画图时产生的图形、属性信息，通过 switch 函数判断，在撤销绘图区域重新绘画该图形，并将坐标、属性信息保存到链表及文件当中。

函数原型：void File(int x, int y)

功能描述：小功能选项—文件，在用户选择文件按钮后显示文件功能选项框，并支持快捷键操作

参数描述: int x, int y:两个整型参数, 用于接收鼠标的坐标信息

返回值描述: 无返回值

重要局部变量定义:

- ① int fevent = -1;
- ② RECT Ri (i=1, 2.....5)
- ③ int b = Fkeybd(mouse.x, mouse.y)
- ④ bool flag = 1;

重要局部变量用途描述:

- ① fevent:
用于接收 file_pos 函数的返回值, 判断鼠标移动、点击在哪个选项框上。
- ② Ri:
保存 5 个选项框的坐标信息。
- ③ b:
接收 Fkeybd 函数的返回值, 判断用户输入键盘消息时, 是否按下了快捷键以及按下的快捷键为何键。
- ④ flag:
保存用户是否有按下鼠标键, 用于控制 while 循环。

函数算法描述: 当用户在 main 界面选择文件选项后, File 函数启动, 绘制出文件选项下的子选项, 包括新建、保存、打开等。在函数当中, 采用了 while 循环, 用 flag 控制循环次数。当用户点击了子选项框或按下相应快捷键时, 程序将进入 File 下的子函数, 若点击别处, 将退出 File, 返回到 main 函数。

函数原型: void File_Close(int x, int y)

功能描述: 关闭当前界面的文件

参数描述: int x, int y:两个整型参数, 用于接收鼠标的坐标信息

返回值描述: 无返回值

重要局部变量定义: int re

重要局部变量用途描述: re: 若界面中含有文件且该文件未被保存时, 程序会弹出询问消息框。re 用于保存弹出消息框的返回结果。

函数算法描述: 当用户在文件子选项框中选择了关闭按钮时, File_Close 函数启动。其首先会根据外部全局变量 sign 判断当前页面是否有文件, 若无, 则报错。在有文件的情况下, 系统会弹出消息框, 询问是否保存文件。re 保存消息框返回结果, 并通过 if 语句的判断, 进入对应的程序当中, 如: 保存文件、不保存或取消, 并实时更新下方状态栏。

函数原型: void File_New(int x, int y)

功能描述: 新建文件

参数描述: int x, int y:两个整型参数, 用于接收鼠标的坐标信息

返回值描述: 无返回值

重要局部变量定义: int re

重要局部变量用途描述: re: 若界面中含有文件且该文件未被保存时, 程序会弹出询问消息框。re 用于保存弹出消息框的返回结果。

函数算法描述: 当用户在文件子选项中选择了新建按钮时, File_New 函数启动。其首先会根据外部全局变量 sign 判断当前页面是否有文件。若有, 系统会弹出消息框, 询问是否保存文件。re 保存消息框返回结果, 并通过 if 语句的判断, 进入对应的程序当中, 如: 保存文件、不保存或取消, 并实时更新下方状态栏, 重置上一个文件操作遗留下的全部信息。若无, 则绘制画图区域, 令 sign = 1, 并更新状态栏。

函数原型: void File_Open(int x, int y)

功能描述: 打开文件

参数描述: int x, int y:两个整型参数, 用于接收鼠标的坐标信息

返回值描述: 无返回值

重要局部变量定义: int re

重要局部变量用途描述: re: 用于保存弹出消息框的返回结果。

函数算法描述: 当用户在文件子选项中选择了打开按钮时, File_Open 函数启动。其首先会根据外部全局变量 sign 判断当前页面是否有文件。若有, 系统会弹出消息框, 询问是否保存文件。re 保存消息框返回结果, 并通过 if 语句的判断, 进入对应的程序当中, 如: 保存文件、不保存或取消, 并实时更新下方状态栏, 弹出消息框, 输入需打开的文件地址。若该文件存在, 则调用 loadpast 函数, 将文件中的图元信息全部绘画到绘图区, 否则报错。若当前界面没有文件, 则直接进入文件打开程序。

函数原型: int file_pos(int x, int y)

功能描述: 判断鼠标所处位置在哪个子选项框上

参数描述: int x, int y:两个整型参数, 用于接收鼠标的坐标信息

返回值描述: 整型返回值, 根据鼠标所处的不同位置, 返回设定好的对应值

重要局部变量定义: int i

重要局部变量用途描述: i: 既表示第 i 个选项框, 也表示返回值 i (如第 0 个选项框的返回值对应为 0)。若不在选项框或图标上, 则 i = -1, 并作为返回值返回。

函数算法描述: file_pos 函数通过 for 循环和 if 判断, 将鼠标的坐标信息与 File 里设置好的选项框坐标进行比较, 由此判断出鼠标正在哪个选项框之上, 并返回相对应的 i 值。

函数原型：int File_Save(int x, int y)

功能描述：保存当前页面的文件

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：整型返回值，表示文件是否保存成功

重要局部变量定义：int re

重要局部变量用途描述：re：用于保存弹出消息框的返回结果。

函数算法描述：当用户在文件子选项中选择了保存按钮时，File_Save 函数启动。其首先会根据外部全局变量 sign 判断当前界面是否有文件。若无，则报错。若有，当该文件由 open 打开且未做更改时，无提示窗口，其余情况下，若文件未被保存过或已被保存过但已做过修改，则弹出消息框，输入存储路径。在路径可用的情况下，若保存路径重复，则弹出消息框，并用 re 保存选择，分别进入到对应程序中，如：覆盖、不覆盖、取消；若保存路径不重复，则直接调用 Screen_File 函数将图元信息保存。若保存路径不可用，则报错。与此同时，File_Save 函数将保持状态栏的即时状态显示。

函数原型：int Fkeybd(int x, int y)

功能描述：判断用户按下的键盘消息，并返回对应的整数值

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：整型返回值，根据不同的键盘消息，返回设定好的对应值

重要局部变量定义：char s

重要局部变量用途描述：s:用 getch() 函数，读取用户的按下的键钮，并将其保存到 s 当中，方便后面 switch 函数的选择。

函数算法描述：Fkeybd 函数通过 getch，读取用户键盘按下的字母，并为每个字母设定相应的返回值（如 'N' 或 'n' 对应 0）。经过 switch 函数的判断选择后，将该值返回到 File 函数中。

函数原型：void loadpast(char* s)

功能描述：将文件中的图元信息重新绘画到屏幕上

参数描述：char* s: 保存需打开的文件存储路径

返回值描述：无返回值

重要局部变量定义：struct rec_coor* p, *tail

重要局部变量用途描述：

① p:

链表结点，将文件中的图元信息依次保存到链表当中

② tail:

链表结点，遍历时将链表中的信息依次取出，按照数字、字符串的对应规则重新绘制图形到屏幕上

函数算法描述：当 File_Open 调用 loadpast 函数时，loadpast 将保存图元信息的链表清空，并以只读形式打开对应文件，依次将文件中的图元信息保存到程序的链表当中。然后，遍历链表，将链表中每一个结点所包含的属性、坐标、字符串信息按照既有的对应规则转换成变量值，并根据图形种类画出，最后将上一个文件操作遗留下的信息全部重置。

函数原型：void Screen_File(char*s)

功能描述：将图元信息保存到文件当中

参数描述：char* s：文件的存储路径

返回值描述：无返回值

重要局部变量定义：rec_coor* p

重要局部变量用途描述：p:链表结点，保存图形的图元信息

函数算法描述：当 File_Save 调用 Screen_File 函数时，Screen_File 将以二进制只读形式打开或新建文件，并遍历保存图元信息的链表，依次将图元信息即时写入到文件当中。

函数原型：void Help(int x, int y)

功能描述：小功能选项一帮助，在用户选择帮助按钮后显示帮助功能选项框，并支持快捷键操作

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：

- ① int Hevent = -1;
- ② RECT Ri (i=1, 2)
- ③ int b = Hkeybd(mouse.x, mouse.y)
- ④ bool flag = 1;

重要局部变量用途描述：

- ① Hevent:
用于接收 help_pos 函数的返回值，判断鼠标移动、点击在哪个选项框上。
- ② Ri:
保存 2 个选项框的坐标信息。
- ③ b:
接收 Hkeybd 函数的返回值，判断用户输入键盘消息时，是否按下了快捷键以及按下的快捷键为何键。

④ flag:

保存用户是否有按下鼠标键，用于控制 while 循环。

函数算法描述：当用户在 main 界面选择帮助选项后，Help 函数启动，绘制出帮助选项下的子选项框，包括关于本软件、使用方法等。在函数当中，采用了 while 循环，用 flag 控制循环次数。当用户点击了子选项框或按下相应快捷键时，程序将进入 Help 下的子函数，若点击别处，将退出 Help，返回到 main 函数。

函数原型：void Help_about(int x, int y)

功能描述：关于本软件

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：若用户在帮助子选项中选择了关于本软件，Help_about 函数启动。其将弹出消息框，显示本软件的相关信息。

函数原型：void Help_way(int x, int y)

功能描述：弹出使用方法消息框

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：若用户在帮助子选项中选择了帮助，Help_way 函数启动。其将弹出消息框，显示本软件的使用方法。

函数原型：int help_pos(int x, int y)

功能描述：判断鼠标所处位置在哪个子选项框上

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：整型返回值，根据鼠标所处的不同位置，返回设定好的对应值

重要局部变量定义：int i

重要局部变量用途描述：i:既表示第 i 个选项框，也表示返回值 i（如第 0 个选项框的返回值对应为 0）。若不在选项框或图标上，则 i = -1，并作为返回值返回。

函数算法描述：help_pos 通过 for 循环和 if 判断，将鼠标的坐标信息与 Help 函数里设置好的选项框坐标进行比较，由此判断出鼠标正在哪个

选项框之上，并返回相对应的 i 值。

函数原型：int Hkeybd(int x, int y)

功能描述：判断用户按下的键盘消息，并返回对应的整数值

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：整型返回值，根据不同的键盘消息，返回设定好的对应值

重要局部变量定义：char s

重要局部变量用途描述：s:用 getch() 函数，读取用户的按下的键钮，并将其保存到 s 当中，方便后面 switch 函数的选择。

函数算法描述：Hkeybd 函数通过 getch，读取用户键盘按下的字母，并为每个字母设定相应的返回值（如 'A' 或 'a' 对应 0）。经过 switch 函数的判断选择后，将该值返回到 Help 函数中。

函数原型：void Paint(int x, int y)

功能描述：小功能选项—绘图，在用户选择绘图按钮后显示绘图功能选项框，并支持快捷键操作

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：

- ① int Pevent = -1;
- ② RECT Ri (i=1, 2)
- ③ int b = Pkeybd(mouse.x, mouse.y)
- ④ bool flag = 1;

重要局部变量用途描述：

- ① Pevent:
用于接收 paint_pos 函数的返回值，判断鼠标移动、点击在哪个选项框上。
- ② Ri:
保存 2 个选项框的坐标信息。
- ③ b:
接收 Pkeybd 函数的返回值，判断用户输入键盘消息时，是否按下了快捷键以及按下的快捷键为何键。
- ④ flag:
保存用户是否有按下鼠标键，用于控制 while 循环。

函数算法描述：当用户在 main 界面选择绘图选项后，Paint 函数启动，绘制出绘图选项下的子选项框，包括基本形状、属性等。在函数当中，采用了 while 循环，用 flag 控制循环次数。当用户点击了子选项框或按下相应快捷键时，程序将进入 Paint 下的子函数，若点

击别处，将退出 Paint，返回到 main 函数。

函数原型：int Pkeybd(int x, int y)

功能描述：判断用户按下的键盘消息，并返回对应的整数值

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：整型返回值，根据不同的键盘消息，返回设定好的对应值

重要局部变量定义：char s

重要局部变量用途描述：s:用 getch() 函数读取用户的按下的键钮，并将其保存到 s 当中，方便后面 switch 函数的选择。

函数算法描述：Pkeybd 函数通过 getch()，读取用户键盘按下的字母，并为每个字母设定相应的返回值(如 'B' 或 'b' 对应 0)。经过 switch 函数的判断选择后，将该值返回到 Paint 函数中。

函数原型：int paint_pos(int x, int y)

功能描述：判断鼠标所处位置在哪个子选项框上

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：整型返回值，根据鼠标所处的不同位置，返回设定好的对应值

重要局部变量定义：int i

重要局部变量用途描述：i:既表示第 i 个选项框，也表示返回值 i (如第 0 个选项框的返回值对应为 0)。若不在选项框或图标上，则 i = -1，并作为返回值返回。

函数算法描述：paint_pos 通过 for 循环和 if 判断，将鼠标的坐标信息与 Paint 函数里设置好的选项框坐标进行比较，由此判断出鼠标正在哪个选项框之上，并返回相对应的 i 值。

函数原型：void Paint_Att(int x, int y)

功能描述：打开子选项框一属性，作为 color、width、linestyle、fill 函数的入口

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：

- ① int Aevent = -1;
- ② RECT Ri (i=1, 2, 3, 4)
- ③ int flag = 1;

重要局部变量用途描述：

- ① Aevent:

用于接收 at_pos 函数的返回值，判断鼠标移动、点击在哪个选项框上。

② Ri:

保存 4 个选项框的坐标信息。

③ flag:

保存用户是否有按下鼠标键，用于控制 while 循环。

函数算法描述：当用户在绘图选项中选择了属性选项后，Paint_Att 函数启动，绘制出属性子选项框，包括颜色、宽度、线型、填充等。在函数当中，采用了 while 循环，用 flag 控制循环次数。当用户点击了子选项框时，程序将进入 Paint_Att 下的子函数，若点击别处，将退出 Paint_Att，返回到 main 函数。

函数原型：void Paint_Bashp(int x, int y)

功能描述：打开子选项框—基本形状

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：

① int Bevent = -1;

② RECT Ri (i=1, 2, 3, 4)

③ int flag = 1;

重要局部变量用途描述：

① Bevent:

用于接收 bs_pos 函数的返回值，判断鼠标移动、点击在哪个选项框上。

② Ri:

保存 4 个选项框的坐标信息。

③ flag:

保存用户是否有按下鼠标键，用于控制 while 循环。

函数算法描述：当用户在绘图选项中选择了基本形状选项后，Paint_Bashp 函数启动，绘制出基本形状子选项框，包括矩形、圆角矩形等。在函数当中，采用了 while 循环，用 flag 控制循环次数。当用户点击了子选项框时，程序将进入 Paint_Bashp 下的子函数，若点击别处，将退出 Paint_Bashp，返回到 main 函数。

函数原型：int at_pos(int x, int y)、int bs_pos(int x, int y)

功能描述：判断鼠标所处位置在哪个子选项框上

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：整型返回值，根据鼠标所处的不同位置，返回设定好的对应值

重要局部变量定义：int i

重要局部变量用途描述：i:既表示第 i 个选项框，也表示返回值 i（如第 0 个选项框的返回值对应为 0）。若不在选项框或图标上，则 i

= -1, 并作为返回值返回。

函数算法描述: at_pos、bs_pos 通过 for 循环和 if 判断, 将鼠标的坐标信息与 Paint_Att 函数和 Paint_Bashp 函数里设置好的选项框坐标进行比较, 由此判断出鼠标正在哪个选项框之上, 并返回相对应的 i 值。

函数原型: void color(int x, int y)

功能描述: 选择绘图时将使用的颜色

参数描述: int x, int y: 两个整型参数, 用于接收鼠标的坐标信息

返回值描述: 无返回值

重要局部变量定义:

①②③同 Paint_Bashp

重要局部变量用途描述:

① ②③同 Paint_Bash

函数算法描述: 用户选择颜色选项后, color 函数会绘制出颜色选择框。在函数当中, 采用了 while 循环, 用 flag 控制循环次数。当用户点击了某个颜色时, 变量 chcol 会更改成相应的值, 同时调用 setlinecolor 函数改变绘线的颜色, 若点击别处, 将退出 color, 返回到 main 函数。

函数原型: void width(int x, int y)

功能描述: 选择绘图时将使用的线宽

参数描述: int x, int y: 两个整型参数, 用于接收鼠标的坐标信息

返回值描述: 无返回值

重要局部变量定义:

①②③似 Paint_Bashp

重要局部变量用途描述:

① ②③似 Paint_Bash

函数算法描述: 用户选择线宽选项后, width 函数会绘制出线宽选择框。在函数当中, 采用了 while 循环, 用 flag 控制循环次数。当用户点击了某个线宽时, 变量 chwid 会更改成相应的数值, 若点击别处, 则退出 width, 返回到 main 函数。

函数原型: void linestyle(int x, int y)

功能描述: 选择绘图时将使用的线型

参数描述: int x, int y: 两个整型参数, 用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：

①②③似 Paint_Bashp

重要局部变量用途描述：

①②③似 Paint_Bash

函数算法描述：用户选择线型选项后，linestyle 函数会绘制出线型选择框。在函数当中，采用了 while 循环，用 flag 控制循环次数。当用户选中了某个线型时，变量 chosl 会更改成相应的数值，若点击别处，则退出 linestyle，返回到 main 函数。

函数原型：void fill(int x, int y)

功能描述：选择绘图时将使用的线型

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：

①②③似 Paint_Bashp

重要局部变量用途描述：

①②③似 Paint_Bash

函数算法描述：用户选择填充色选项后，fill 函数会绘制出线型选择框。在函数当中，采用了 while 循环，用 flag 控制循环次数。当用户选中了某个填充色时，变量 chofl 会更改成相应的数值，若点击别处，则退出 fill，返回到 main 函数。

函数原型：int color_pos(int x, int y)、int width_pos(int x, int y)、int style_pos(int x, int y)、int fill_pos(int x, int y)

功能描述：判断鼠标所处位置在哪个选项框上

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：整型返回值，根据鼠标所处的不同位置，返回设定好的对应值

重要局部变量定义：int i

重要局部变量用途描述：i:既表示第 i 个选项框，也表示返回值 i（如第 0 个选项框的返回值对应为 0）。若不在选项框或图标上，则 i = -1，并作为返回值返回。

函数算法描述：color_pos、width_pos、style_pos、fill_pos 通过 for 循环和 if 判断，将鼠标的坐标信息与 color、width、linestyle、fill 函数里设置好的选项框坐标进行比较，由此判断出鼠标正在哪个选项框之上，并返回相对应的 i 值。

函数原型：void Paint_Rectangle(int x, int y)

功能描述：根据用户鼠标按下和松开位置绘画矩形

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：

- ① int lx, ly, rx, ry
- ② bool is_down = FALSE
- ③ bool is_draw = FALSE
- ④ int flag = 1

重要局部变量用途描述：

- ① lx, ly, rx, ry:
保存绘制矩形时的鼠标按下位置坐标和鼠标松开位置坐标，作为矩形的左上角和右下角两点。
- ② is_down:
判断鼠标是否处于按下状态
- ③ is_draw
保存绘图状态信息（0 表示停止画图，1 表示正在画图）
- ④ flag:
控制 while 循环的次数

函数算法描述：当用户在基本图形选项框中选择了矩形后，即可开始画图。首先，函数会根据外部全局变量 sign 的值判断当前界面是否有文件，若无，则报错；若有，则进入 while 循环，可正常画图。若鼠标点击了图标工具栏如撤销、重做等，则程序会进入该函数；若鼠标在绘图区域按下，is_down 变为 TRUE，若此时鼠标发生移动，则 is_draw 变为 TRUE。当鼠标松开时，若 is_draw 为 TRUE，则函数会根据 chcol、chwid 等变量的值，设定绘图属性，并调用 rectangle 函数，绘画矩形，再等待字符串输入。当检测到输入 ESC 时，输入结束，函数调用 save_coor 函数储存图元信息。若用户鼠标点击了退出绘图区，则 flag 变为 0，循环结束。

函数原型：void Paint_Rorec(int x, int y)

功能描述：根据用户鼠标按下和松开位置绘画圆角矩形

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：①②③④同 Paint_Rectangle

重要局部变量用途描述：①②③④同 Paint_Rectangle

函数算法描述：当用户在基本图形选项框中选择了圆角矩形后，即可开始画图。首先，函数会根据外部全局变量 sign 的值判断当前界面是否有文件，若无，则报错；若有，则进入 while 循环，可正常画图。若鼠标点击了图标工具栏如撤销、重做等，则程序会进入该函数；

若鼠标在绘图区域按下, is_down 将变为 TRUE, 若此时鼠标发生移动, 则 is_draw 变为 TRUE。当鼠标松开时, 若 is_draw 为 TRUE, 则函数会根据 chcol、chwid 等变量的值, 设定绘图属性, 并调用 roundrect 函数, 绘画圆角矩形, 再等待字符串输入。当检测到输入 ESC 时, 输入结束, 函数调用 save_coor 函数储存图元信息。若用户鼠标点击了退出绘图区, 则 flag 变为 0, 循环结束。

函数原型: void Paint_Diamond(int x, int y)

功能描述: 根据用户鼠标按下和松开位置绘画菱形

参数描述: int x, int y: 两个整型参数, 用于接收鼠标的坐标信息

返回值描述: 无返回值

重要局部变量定义: ①②③④同 Paint_Rectangle

重要局部变量用途描述: ①②③④同 Paint_Rectangle

函数算法描述: 当用户在基本图形选项框中选择了菱形后, 即可开始画图。首先, 函数会根据外部全局变量 sign 的值判断当前界面是否有文件, 若无, 则报错; 若有, 则进入 while 循环, 可正常画图。若鼠标点击了图标工具栏如撤销、重做等, 则程序会进入该函数; 若鼠标在绘图区域按下, is_down 将变为 TRUE, 若此时鼠标发生移动, 则 is_draw 变为 TRUE。当鼠标松开时, 若 is_draw 为 TRUE, 则函数会根据 chcol、chwid 等变量的值, 设定绘图属性, 并根据已经设定好的坐标关系, 绘画菱形, 再等待字符串输入。当检测到输入 ESC 时, 输入结束, 函数调用 save_coor 函数储存图元信息。若用户鼠标点击了退出绘图区, 则 flag 变为 0, 循环结束。

函数原型: void Paint_Arrow(int x, int y)

功能描述: 根据用户鼠标按下和松开位置绘画箭头

参数描述: int x, int y: 两个整型参数, 用于接收鼠标的坐标信息

返回值描述: 无返回值

重要局部变量定义: ①②③④同 Paint_Rectangle

重要局部变量用途描述: ①②③④同 Paint_Rectangle

函数算法描述: 当用户在基本图形选项框中选择了箭头后, 即可开始画图。首先, 函数会根据外部全局变量 sign 的值判断当前界面是否有文件, 若无, 则报错; 若有, 则进入 while 循环, 可正常画图。若鼠标点击了图标工具栏如撤销、重做等, 则程序会进入该函数; 若鼠标在绘图区域按下, is_down 将变为 TRUE, 若此时鼠标发生移动, 则 is_draw 变为 TRUE。当鼠标松开时, 若 is_draw 为 TRUE, 则函数会根据 chcol、chwid 等变量的值, 设定绘图属性, 并根据已经设定好的坐标关系, 绘画箭头, 再等待字符串输入。当检测到输入 ESC 时, 输入结束, 函数调用 save_coor 函数储存图元信

息。若用户鼠标点击了退出绘图区，则 flag 变为 0，循环结束。

函数原型: void save_coor(int lx, int ly, int rx, int ry, int style, int chcol, int chwid, int chosl, int chofl, char* str)

功能描述：将图元信息保存到链表当中

参数描述：

- ① int lx, int ly, int rx, int ry: 鼠标按下和松开的两点的纵横坐标
- ② int style: 图形类型
- ③ int chcol, int chwid, int chosl, int chofl: 所选颜色、线宽、线型、填充颜色
- ④ char* str: 输入图形的字符串

返回值描述：无返回值

重要局部变量定义：struct rec_coor* tail

重要局部变量用途描述：tail: 新建链表结点，保存形参传回的图元信息

函数算法描述：调用 save_coor 函数后，其首先会将已有链表遍历一次，找到链表的尾结点，当 tail 对应保存好形参传递回来的图元信息时，tail 会连接到链表的尾部，从而将图元信息存入链表。

函数原型: void edit (int x, int y)

功能描述：编辑的枢纽

参数描述：int x, int y: 两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：编辑功能框的实现和枢纽。

函数原型: void edit_choose(int x, int y) rec_coor* coor_find(int x, int y)

功能描述：实现选择功能

参数描述：int x, int y: 两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值 后者返回 rec_coor* 类型链表

重要局部变量定义：无，choice

重要局部变量用途描述：choic

函数算法描述：以前者为枝干，后者为叶。全局变量的 choic 负责把选择的图形

数据输送到其他函数。后者是实现的重要部分，方法是通过对鼠标所在的坐标和链表 head 中的数据，匹配出所要选择的图形进行标记。并把数据传回存储在 chice 里面。

函数原型： `void edit_choose(int x, int y) rec_coor*
coor_find(int x, int y)`

功能描述：实现选择功能

参数描述：int x, int y:两个整型参数，用于接收鼠标的坐标信息

返回值描述：无返回值 后者返回 rec_coor*类型链表

重要局部变量定义：无，choice

重要局部变量用途描述：choice

函数算法描述：以前者为枝干，后者为叶。全局变量的 choic 负责把选择的图形数据输送到其他函数。后者是实现的重要部分，方法是通过对鼠标所在的坐标和链表 head 中的数据，匹配出所要选择的图形进行标记。并把数据传回存储在 chice 里面。

函数原型: `void edit_copy(int x, int y) rec_coor* copy_move
(copyxy* xy, rec_coor* copy),int copy_do (copyxy* xy,
rec_coor* copy)`

功能描述：实现复制功能

参数描述：void edit_copy(int x, int y) : int x, int y:两个整型参数，用于接收鼠标的坐标信息

rec_coor* copy_move (copyxy* xy, rec_coor* copy), copyxy* xy, rec_coor* copy 两个结构指针。用于传输数据。

rec_coor* copy_move (copyxy* xy, rec_coor* copy), copyxy* xy, rec_coor* copy 两个结构指针。用于传输数据。

返回值描述：无返回值 后者两者返回 rec_coor*类型链表

重要局部变量定义：无

重要局部变量用途描述：copyxy* xy: 储存鼠标的 x, y 信息。

函数算法描述：第一个函数为枝干两个函数分别实现各自功能，画图思路为：将坐标信息传输过去，同时将图形中心作为参照物进行计算。最后得出结果。

第二个函数是每一次都返回刚画好的图形的函数，在停留 15 毫秒后消失，给使用者提示。

最后一个函数是确定复制的函数，复制之后将所画的图形对应链表存储到 head 中。

4 部署运行和使用说明

4.1 编译安装

下载并解压后，点击 ConsoleApplication5.vproj，打开后点击[本地 Windows 调试器]即可使用。

注意事项：如果没有安装过 easyx，需要点击 EasyX_20200315(beta) 安装 easyx。

4.2 运行测试



最终测试情况一切良好，只要不是飞速使用，不会出现卡机现象。

在测试过程中，我们进行过多次试验，例如对复制后产生的图形进行选择和复制。开始测试时，因为返回值错误导致无法多次选中，此外选择对每个图形都是单次有效，这样的模式减弱了选择的意义，于是小组对选择函数内部的逻辑关系进行了梳理，找到问题所在，并对其进行改动和完善。

4.3 使用操作

在 readme 里面有比较详细的介绍，在帮助界面内也有相关说明。

尊敬的用户：

您好！

欢迎使用小型算法流程图绘制工具，这是此软件使用过程中的特别说明，请您仔细阅读，希望您能用的愉快！

一、菜单系统

本软件提供背景音乐功能，如果您希望关闭它，打开软件后，请点击窗口右上角的“|”符号，音乐将自动关闭。当然，如果您希望再打开它，请再次点击该处的实心三角符号，音乐会重新开始播放。音乐自带一首，如果要听其他音乐可以将文件夹中的音乐换掉，记得把音乐名字改成“song”，只支持 wav 格式。

本软件菜单包括文件、编辑、绘图、帮助、退出五个板块，且皆支持键盘快捷键。当您需要打开选项时，可以使用鼠标点击，也可以按下相应括号里的英文字符并稍微移动鼠标。对于图标工具栏，本软件支持撤销、重做、新建、打开、保存等五个快捷图标。

特别需要注意的是，在您的使用过程中，鼠标移动不宜太快，并请避免在已点开某一个选项框之后用鼠标点击此选项框之外的其它选项上，从而避免黑框的产生。如果已经发生了这种情况，请您重复上述过程，黑框会自动消除。

黑框产生：



黑框去除：



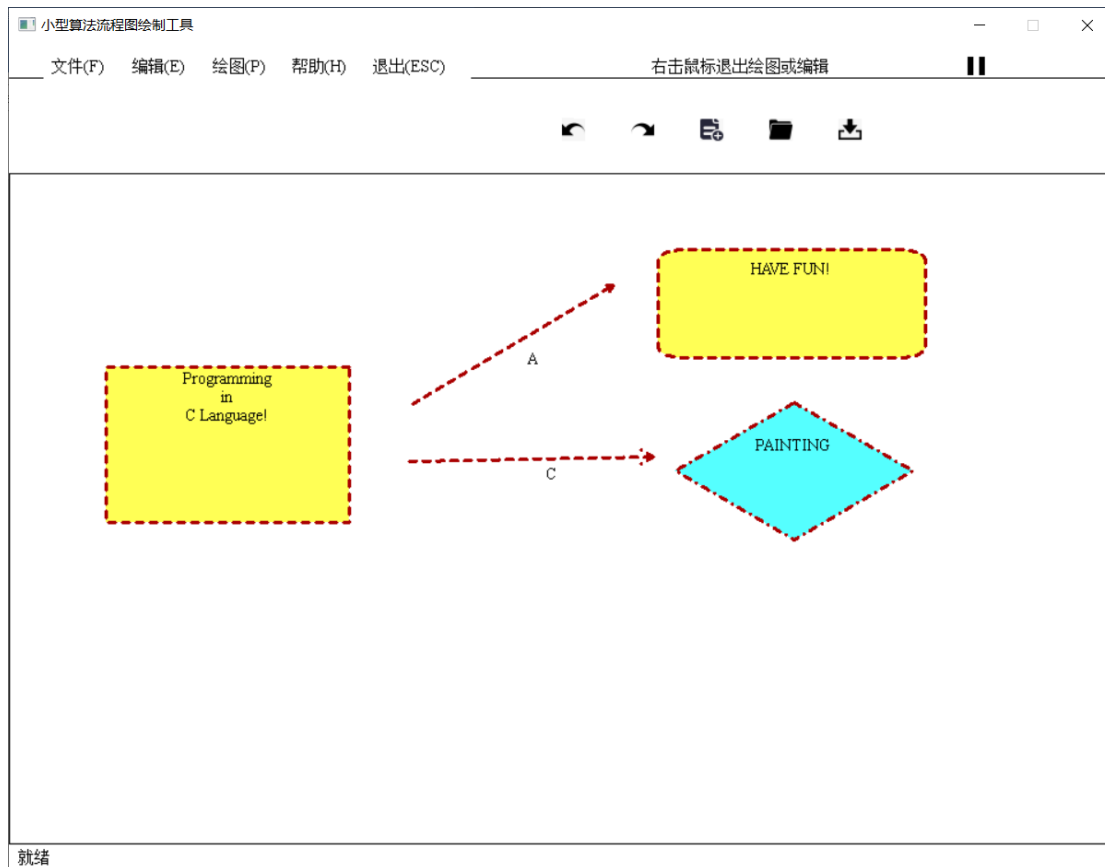
二、绘图

在您选定基本图形并绘制前，请先选择您所需要的属性，其中，图形颜色默认为黑色，线宽默认为细线，线型默认为实线，填充色默认为白色。在您的绘图过程中，请遵守流程图绘制规则，避免图形的交叉或重叠。当您在绘制出的图形中输入英文（只支持英文）时，系统设定为一切其余操作皆不可执行。输入功能支持删除键，您可以自由删除当前输入的结尾字符。当您输入完毕后，请按下 **ESC** 键，以退出输入。当您绘图完毕后，或者需要更换基本图形、属性，请按下鼠标右键，便可退出绘图模式。

另外，本软件提供撤销和重做功能，可以且仅可以撤销您绘画的前一个图形（或重做），重做完成后，重做出来图形的文字内容将不可被修改。

其次，由于本软件窗口大小限制，不建议您大量使用粗线，当然，也请您不要将图形绘制过小，以免出现文字内容不可见或文字遮盖图形的情况。

在您的使用过程中，如果出现选项边框显现的情况，请再点击一次边框即可。在绘制粗线的圆角矩形时，如果您觉得最终绘完的图形较细，请点击撤销，再点击重做，系统会将该图形自动加粗。



三、编辑

选择：按下编辑菜单中的选择后可以点击图形进行选择。在选择完毕后可以点击鼠标右键退出选择模式。

复制：按下编辑菜单中的复制后，如果没有选择图形，可以点击图形拖动复制，如果已经选择，可以直接拖动所要的距离进而复制图形。

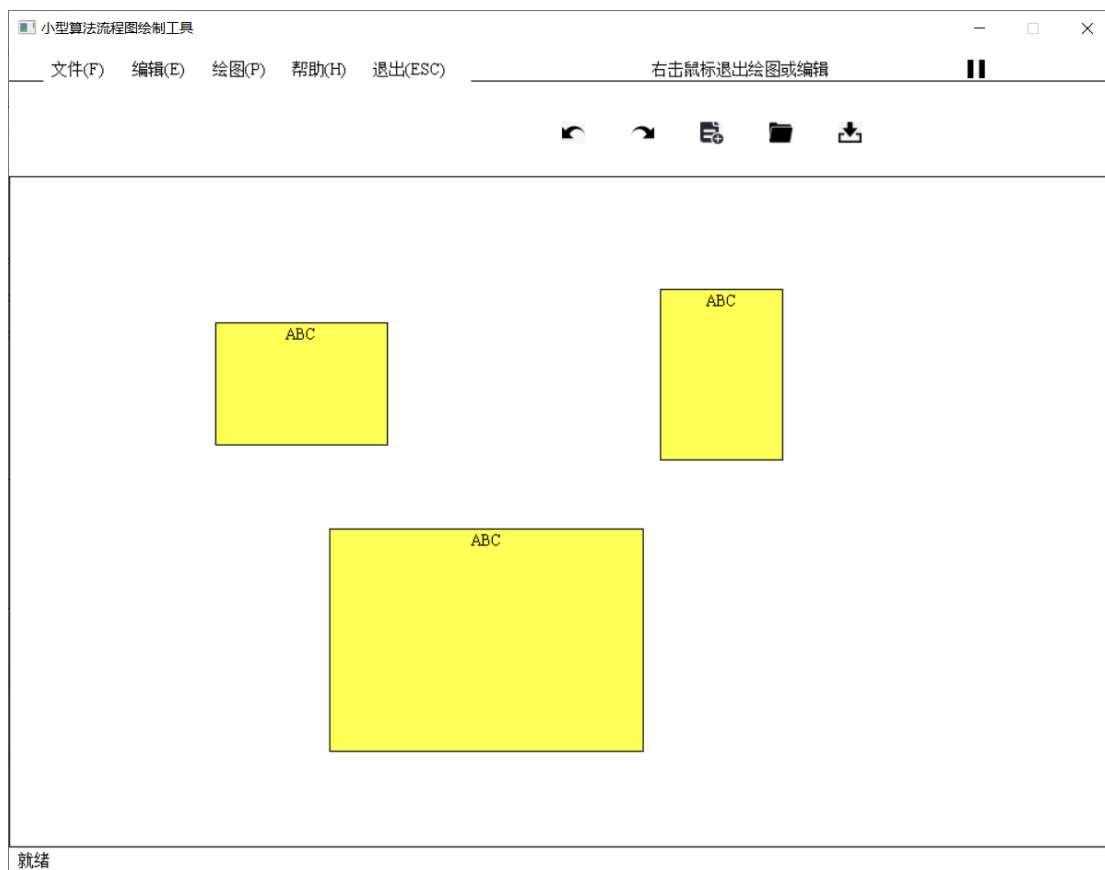
删除：按下编辑菜单中的删除后，如果没有选择图形，点击图形可以直接删除，如果已经选择可以直接删除。

拖拽：按下编辑菜单中的拖拽后，如果没有选择图形，可以点击图形拖动，如果已经选择，可以直接拖动所要的距离从而移动图形。

放缩：按下编辑菜单中的放缩后，如果没有选择图形可以点击图形放缩，如果已经选择，可以直接拖动放缩图形，再次点击确定放缩比例。

旋转：按下编辑菜单中的旋转后，如果没有选择图形，可以点击图形进行旋转。目前只有箭头可以自由旋转，其他都是十字旋转，如果已经选择，可以对已经选择图形进行旋转。

操作细则：进入功能后，鼠标单击图形正式进入功能实行，请勿松开（复制放缩，拖拽过程均有提示），在达到您的满意效果后，再松开鼠标左键以正式完成。因为选中多个目标后，复制过程如果在移动过程中清除残影，程序运行速度会明显下降，价值不大，因此我们将复制过程的残影予以保留，同时残影也可以起到一定的提示作用，不必担心残影会影响效果，在放开鼠标确认后残影会被自动清除。



5 团队合作

5.1 任务分工

郑德钦负责编辑部分，板块之间衔接优化，统筹组内成员修改的函数，报告技术补充与修改。

黄嘉欣负责程序菜单系统（包括快捷键）、撤销等图标工具栏功能、状态信息栏、文件部分以及绘图功能，程序衔接优化，报告修改与补充。

袁致远主要负责文秘工作，报告编写，团队时间协调，信息收集。

5.2 开发计划

4月7日—5月6日：选题公布，组员研究内容分工。

5月6日—5月26日：劳动节假日结束，框架构建，最后采用黄嘉欣写出的最完善的框架。郑德钦更换框架，开始学习 easyx，袁致远熟悉框架开始研究自己分工以内工作，黄嘉欣继续完善、拓展程序功能。

5月26日—6月3日：功能初步完成，进入 bug 修复和功能修正。

6月3日—6月18日：由袁致远撰写报告初稿，黄嘉欣进行技术性修改和总体补充，郑德钦完善整合完成互评版终稿。

6月18日—6月20日：接受意见，对 bug 进行力所能及的优化。

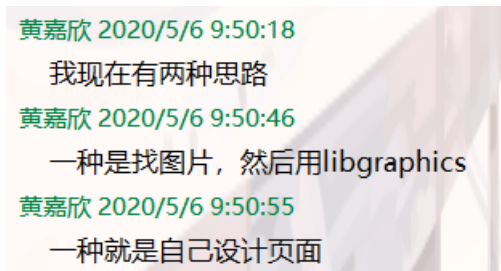
5.3 编码规范

- 1、将功能与枢纽分开，以功能名字定义多函数，可读性、可维护性和可变更性更强。
- 2、对变量名称尽量使用其功能或者指代物品的英文，可读性、可理解性更强。

5.4 合作总结

1.大程撰写：

选题定下，开始组内还没有确切思路。于是黄嘉欣和袁致远开始探索 easyx，郑德钦使用 simpleGUI。



黄嘉欣 2020/5/6 9:50:18

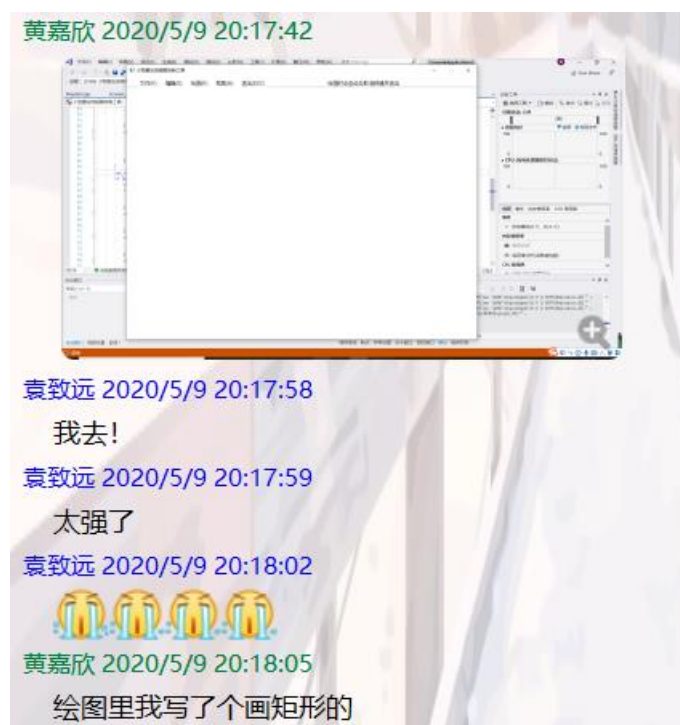
我现在有两种思路

黄嘉欣 2020/5/6 9:50:46

一种是找图片，然后用libgraphics

黄嘉欣 2020/5/6 9:50:55

一种就是自己设计页面



黄嘉欣率先写出了较为完善的第一版框架。于是小组在这个框架上开始编写。由郑德钦负责编辑功能，并提出了图形存储链表的功能性建议。



黄嘉欣在完成程序菜单系统（包括快捷键）、状态信息栏、绘图、文件机理等功能后，继续完成了图标工具栏功能以及背景音乐的播放。在袁致远编写文件保存遇到瓶颈时，与他交流文件读取实现思路，并进一步完成了文件图元信息的保存和文件中图形的读取。

袁致远 2020/5/20 22:25:29

问了一个老师，老师说用windows写很快

袁致远 2020/5/20 22:25:49

不是不是，就是我的file save功能写不出来

袁致远 2020/5/20 22:26:00

老师建议我换一个IDE

袁致远 2020/5/20 22:26:04

用Windows

小半 2020/5/24 9:45:05

致远，这个新版代码里我在链表中添加属性的四个int，因为你做打开的时候要还原属性。对于保存，直接按一定格式写入指定文件就好，然后打开的时候，如果打开文件里有东西，那个链表头我觉得应该要把文件遍历一遍，把文件里的每一个图形信息先放到链表里，然后再继续保存新画的图形，也就是说打开那里需要有一个if-else，然后图形的重绘，你看我写的redo，在一个while下，基本上直接复制就可以了。总体思路下来感觉执行起来应该不会很难

小半 2020/5/24 9:45:19

加油！

袁致远 2020/5/24 9:47:29



编写程序过程中，小组成员分工明确、团结合作、积极讨论，共同致力于大程作业的圆满完成。

郑德钦 2020/5/26 19:41:42

嘉欣

郑德钦 2020/5/26 19:43:00

之前我说让你把那个链表对应的什么图形做一个标记

郑德钦 2020/5/26 19:43:08

你怎么安排的呀

郑德钦 2020/5/26 19:43:11



黄嘉欣 2020/5/26 19:44:56

emm

黄嘉欣 2020/5/26 19:45:05

就是那个style

黄嘉欣 2020/5/26 19:45:17

1是矩形

郑德钦 2020/5/26 19:45:21

我找到了

郑德钦 2020/5/26 19:45:23

千辛万苦

黄嘉欣 2020/5/26 19:45:29



郑德钦 2020/5/27 15:11:13

怎么修复的

郑德钦 2020/5/27 15:11:14

?

郑德钦 2020/5/27 15:11:21

我复制的问题也是乱码

袁致远 2020/5/27 15:11:52

/emm/emm

黄嘉欣 2020/5/27 15:11:54

我发现如果判断最后一个是不是为 '\0' 是不起作用的

黄嘉欣 2020/5/27 15:12:08

所以把最后一位改成了 '#'

黄嘉欣 2020/5/27 15:12:26

再在后面加个 '\0'

黄嘉欣 2020/5/27 15:12:47

判断的时候就看倒数第二位是不是#号就ok了

互评过后开始分工，互相修复 bug。

郑德钦 2020/6/17 15:51:49

要不我把那一步换成设置颜色

郑德钦 2020/6/17 15:51:51

我试试

郑德钦 2020/6/17 15:56:05

不是这个问题

郑德钦 2020/6/17 15:56:19

。。。。换掉之后还在

郑德钦 2020/6/17 15:57:08

话说默认是黑色吧

袁致远 2020/6/17 15:57:15

嗯

黄嘉欣 2020/6/17 15:57:24

是的

黄嘉欣 2020/6/17 15:57:30

但是那也是我设置的

黄嘉欣 2020/6/17 15:57:43

你是咋设置换颜色的呀

袁致远 2020/6/17 15:57:43

缺口我之前也有过

郑德钦 2020/6/17 17:43:28

换成其他键可以吗

郑德钦 2020/6/17 17:43:41

#号需要shift和3一起按

郑德钦 2020/6/17 17:43:43

比较麻烦

黄嘉欣 2020/6/17 17:44:03

emm

黄嘉欣 2020/6/17 17:44:11

可以想一个不会输入的键

2.程序设计专题报告撰写：

袁致远同学完成第一稿简单的报告。

后续由黄嘉欣同学完成大部分函数的解释说明和报告的完善补充，并对报

告内容进行检查、修改。

郑德钦最后进行完善和空白填补如 5.团队合作，并对报告整体进行排版、纠错。

3.总结:

开发亮点:

- ① 初始界面颜色渐变;
- ② 背景音乐支持;
- ③ 程序简洁美观;
- ④ 应用功能全面。

挑战点:

- ① 图元信息的保存以及文件中图形的读取;
- ② 不同功能函数之间的有机协调;
- ③ 全局变量在各板块间的综合使用。

应用知识点总结:

- ① 链表: 包括链表结点的增加、链表的遍历和链表内容的读取;
- ② 图形库: 包括各类基本图形绘制函数、字符输出函数等, 以及鼠标消息结构体;
- ③ 基本库函数: 包括键盘输入检测函数 kbhit、延迟函数 Sleep 等等;

5.5 收获感言

袁致远: 很辛苦但是收获颇丰的一次经历, 很多地方很难, 大家也都曾经一筹莫展, 不过都积极的想办法解决, 很感谢自己一直的努力吧, 虽然帮不上什么大忙, 但是自己也算是给了一个交代, 学习是一个过程, 不是看结果的好坏, 只要尽力了就对得起自己, 感谢这段时间的曲折与磨砺, 当然最感谢其他同伴的助力, 在这段旅程中我也收获了友情, 合作让我们的情谊越来越深, 谢谢同伴的理解。我会更加努力, 在这段时光尽可能多的在团队中起到自己缺失的功能。希望可以圆满完成了这项任务, 谢谢大家。

郑德钦: 很开心可以自己写出一个程序, 写出一个个奇葩的功能还是很开心的, 虽然自己阉割了一些些功能, 但是也尽力了。没有很浮夸的期望, 只希望可以対得起队友, 对得起这段日子的学习, 大程这一门课学完也真的获得了一个技能了, 或许是把原本程序设计基础学到的东西化为一个技能吧, 原本那种输入框的程序实用价值太低了。以后自己也可以写一些小程序用了, 比如单词背诵的程序, 总之风雨兼程来, 不可踏空去。希望大家都可以有收获。希望可以圆满成功的完成这个任务, 感谢队友这段时间的互相帮助与支持。

黄嘉欣：“踏不平丘山，回头哪儿有岸。”进入大学以前，看到各种各样的软件，我总是会感叹它们功能的强大。直到自己亲身接触到编程这门课，才发现，原来别人可以做到的，我也可以。这一次大程作业，是试炼，也是锤炼。虽然做了很多工作：有框架，有文件，有绘图；虽然很累：多少天夜以继日地思索、实现，但看到程序正确地运行，看到成果摆在自己的眼前，原来付出的努力和心血，都是值得的。通过它，我检验了自己的学习能力、编程能力、合作能力和创新能力；我看到了自己某些能力的不足，也发现了自己的闪光点：或许我不够聪明，但努力能改变一切。最后，我想感谢各位组员，感谢大家的付出。这一次大程作业，将是我们学习道路上一个重要的里程碑，我们共同将它建筑，也将在它的激励下走得更远。

6 参考文献资料

Easyx 官网: <https://easyx.cn/>

百度: <https://www.baidu.com/>

CSDN 社区: <https://www.csdn.net/>