

**程 序 设 计 专 题**

**大 程 序 报 告**



大程名称： 《Free Geometry Tool》

**指导老师： 吴春明**

**2021~2022春夏学期 2022 年 06 月 05 日**

**目 录**

[1 大程序简介 3](#_Toc19565)

[1.1 选题背景及意义 3](#_Toc31844)

[1.2 目标要求 3](#_Toc2056)

[1.3 术语说明 3](#_Toc12273)

[2 需求分析 4](#_Toc16229)

[2.1 业务需求 4](#_Toc1137)

[2.2 功能需求 4](#_Toc69)

[2.3 数据需求 4](#_Toc31820)

[2.4 性能需求 4](#_Toc2140)

[3 程序开发设计 5](#_Toc14257)

[3.1 总体架构设计 5](#_Toc29529)

[3.2 功能模块设计 7](#_Toc22238)

[3.3 数据结构设计 11](#_Toc20102)

[3.4 函数设计描述 11](#_Toc4055)

[3.5 源代码文件组织设计 12](#_Toc26163)

[4 部署运行和使用说明 14](#_Toc6820)

[4.1 编译安装 14](#_Toc886)

[4.2 运行测试 14](#_Toc28800)

[4.3 用户使用手册 16](#_Toc17684)

[5 团队合作 21](#_Toc32175)

[5.1 开发计划 21](#_Toc21646)

[5.2 编码规范 21](#_Toc8911)

[5.3 任务分工 21](#_Toc160)

[5.4 个人遇到的难点与解决方案 21](#_Toc2527)

[5.4.1宋畅赢 21](#_Toc20666)

[5.4.2 邵晨昕 22](#_Toc13396)

[5.4.3 王子正 23](#_Toc310)

[5.4 合作总结 23](#_Toc4847)

[5.5 收获感言 24](#_Toc5726)

[6 参考文献资料 26](#_Toc15667)

**Free Geometry Tool程序设计项目**

# 大程序简介

## 选题背景及意义

选题以几何画板为背景，希望能够实现一个免费的基础几何绘制程序。

## 目标要求

能够实现基本的二维点、线以及基础几何图形如圆、多边形的绘制，并能够在基本形态之上进行适当拓展。且最终能够实现文件形式的导出与输入。

## 术语说明

编辑：对现有的点性质进行编辑（选择性将点进行隐藏与显示；改变所有点的点型；选择性改变颜色）。

构造：在现有点的基础上进行功能的实现。

# 需求分析

## 业务需求

使用基础的计算机图形学将计算机技术运用到几何领域，生动地展现几何图形

## 功能需求

1. 绘制点
2. 绘制线：直线、线段、射线
3. 圆相关：绘制圆、绘制圆弧
4. 绘制多边形
5. 绘制矩形
6. 能够对生成的点进行编辑
7. 通过已经生成的点构造:中点、直线、矩形
8. 绘制的图形通过文件保存或读取
9. 利用已有的功能，满足用户的绘图需求，构造复杂的图形

## 数据需求

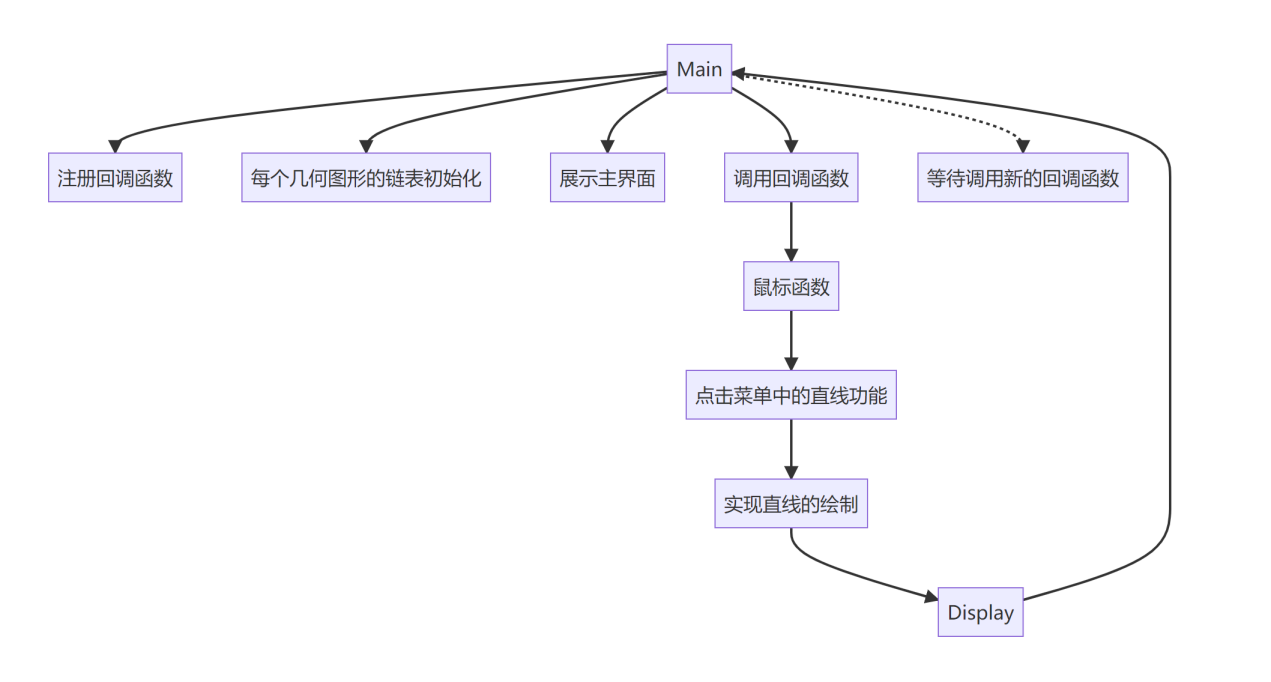
没有数据需求,所有的数据都是通过图形界面绘制生成。

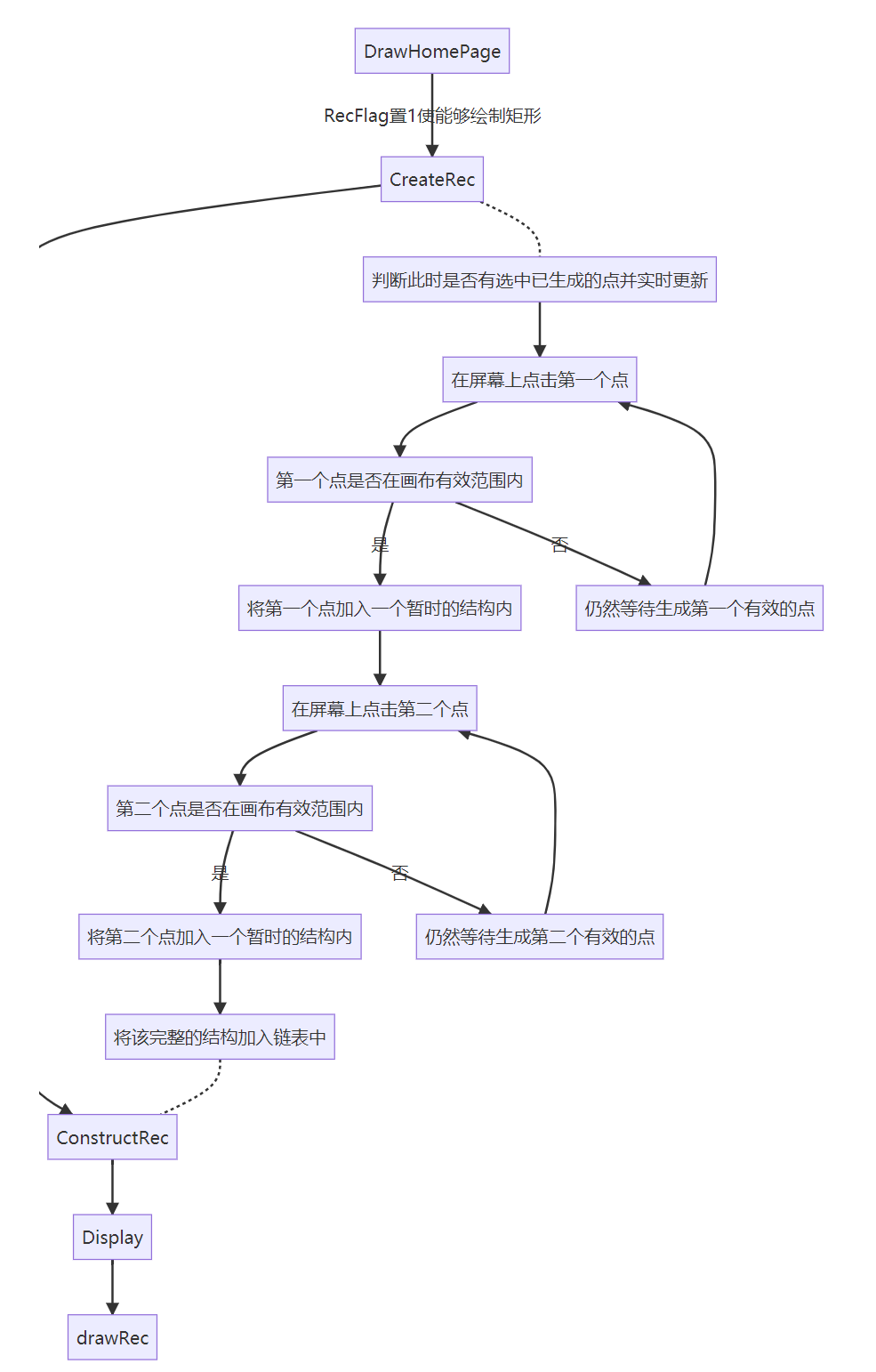
## 性能需求

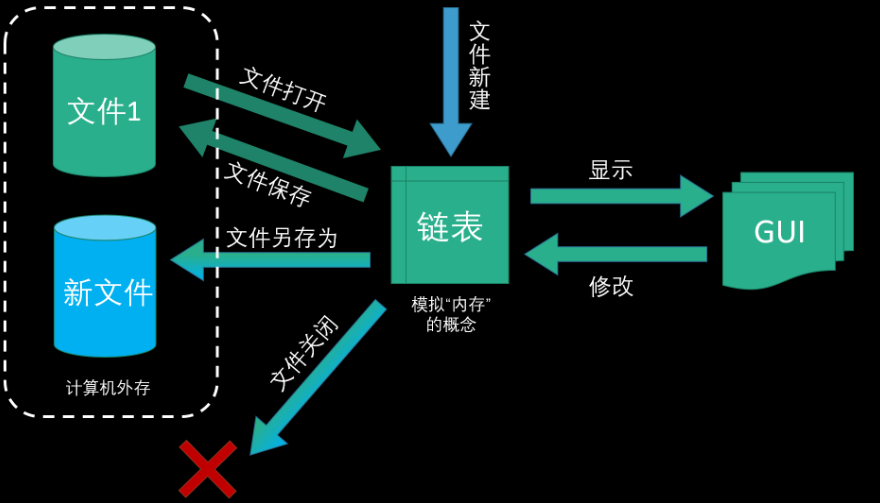
算法的空间复杂度和时间复杂度都是O(n)。

# 程序开发设计

## 总体架构设计

**

**



## 功能模块设计

1. 点功能：设计点以一个半径极小的圆表示，并把点的数据存储于含pointhead头结点的ListNodePoint链表中，数据中含有x横坐标、y纵坐标、pointnumber点的编号、chosenflag是否选择、hidenflag是否隐藏，画点时先MovePen(x+r, y)再DrawArc(r, 0, 360)。
2. 绘制点：在Display函数中使用drawPoint函数遍历链表获得数据，再使用drawthepoint函数把这些数据绘制成具体的点，若该点chosenflag等于1则使用不同颜色，若该点hidenflag等于1则不作出该点。
3. 创造点：在Display函数中使用CreatePoint函数，当DrawFlag等于Pointflag时开始创造点功能，点会跟随鼠标移动，如果鼠标移动至已有且hidenflag等于0的点，则点的半径增加至两倍，原有的点chosenflag变为1，鼠标自动吸附至该点，若此时按下鼠标则不会放置点，如果鼠标移开，则点的半径恢复，原有的点chosenflag变为0，若此时按下鼠标则使用CreatePointNode函数放置点，把点的数据作为新的结点加入链表中。
4. 选择点：在Display函数中使用choosepoint函数，当DrawFlag等于Mouseflag时，如果鼠标移动至已有且hidenflag等于0的点，则点的半径增加至两倍，原有的点chosenflag形式上暂时变为1，鼠标自动吸附至该点，若此时按下鼠标则chosenflag实质上发生改变，如果鼠标移开，则点的半径恢复，原有的点chosenflag形式上暂时变为0。
5. 线功能：把线的数据存储于含linehead头结点的ListNodePoint链表中，数据中含有p1第一个点的结构指针，p2第二个点的结构指针，linetype线的类型，以线段为例，画线时先MovePen(p1->x, p1->y)再DrawLine(p2->x - p1->x, p2->y - p1->y)。
6. 绘制线：在Display函数中使用drawLine函数遍历链表获得数据，根据线的类型利用不同的画法画出不同的线。
7. 创造线：以线段为例，在Display函数中使用CreateSegment函数，当DrawFlag等于Segmentflag时开始创造线段功能，点会跟随鼠标移动，如果鼠标移动至已有且hidenflag等于0的点，则点的半径增加至两倍，原有的点chosenflag变为1，鼠标自动吸附至该点，如果鼠标移开，则点的半径恢复，原有的点chosenflag变为0，若此时按下鼠标，则选取或创造该点作为第一个点，createsegmentpoint变为1，FinishFlag变为1。然后，点会跟随鼠标移动，同时绘制两点间的线段作为效果图，确定第二点时与之前相同，但如果第二个点与第一个点相同时不会以此作为第二个点，createsegmentpoint变为0，FinishFlag变为0，使用CreateLineNode函数放置线段，把线段的数据作为新的结点加入链表中。
8. 构造线：以线段为例，使用ConstructSegment函数，先在界面中选择两点，再点击构造>线段后，当chosenflag等于1的点有且仅有2个时，使用CreateLineNode函数，把线段的数据作为新的结点加入链表中。
9. 圆功能：把圆的数据存储于含circlehead头结点的ListNodeCircle链表中，数据中含有p圆心的结构指针，r圆的半径长度。画圆时先MovePen（p->x + r, p->y）再DrawArc（r，0，360）。
10. 绘制圆：在Display函数中使用drawCircle函数遍历链表获得数据，并绘制到画布之上。
11. 创造圆：在Display函数中使用CreateCircle函数，当DrawFlag等于Circleflag时，开始创造圆功能。点会跟随鼠标的移动，与创造线时选择第一点的方式相同，创造圆所选择第一点为圆的圆心，当第一点被确定之后，createcirclepoint和FinishFlag变为1。接着随着鼠标的移动，将会绘制以鼠标所在位置到第一点之间距离为半径的圆作为效果图。以同样的规则确定第二点之后，createcirclepoint和FinishFlag归于0，且调用CreateCircleNode函数将当前所得圆数据加入圆链表之中。
12. 圆弧功能：把圆弧的数据存储于含sectorhead头结点的ListNodeSector链表中，数据中含有p圆心的结构指针，r圆弧的半径长度，t1起始点的角度度数，t2终止点的角度度数。画圆弧时先MovePen（p->x + r\*cos（t1/180\*π）, p->y+r\*sin（t1/180\*π）），再DrawArc（r，t1，t2-t1），由于DrawArc函数中表述角的单位为角度，而在sin和cos函数中表述角的单位为弧度，所以需要对角的单位进行转换。
13. 绘制圆弧：在Display函数中使用drawSector函数遍历链表获得数据，并绘制到画布之上。
14. 创造圆弧：在Display函数中使用CreateSector函数，当DrawFlag与Sectorflag相等时，开始创造圆弧。与创造圆功能相似的思路，创造圆弧时选取第一点为圆心，选取圆心之后CreateSectorPoint1与FinishFlag均变为1。接着确定圆弧的起始点，在确定起始点之后，即可确定圆弧的半径大小，同时CreateSectorPoint2变为1。接着在鼠标移动过程中，会绘制出以鼠标所在位置为终止点的圆弧模拟图。为保证绘制方向始终一致，对于所得角度需要进行一定的变换再使用DrawArc函数。当确定第三点之后，圆弧绘制完成，CreateSectorPoint1，CreateSectorPoint2和FinishFlag回归于0，调用CreateSectorNode函数将当前圆弧相关数据存储进圆弧链表之中。
15. 文件功能：把链表中的数据以文本文件的方式存入。先通过调用<windows.h>头文件中的OPENFILENAME打开用户窗口下的文件选择，通过FileDialog(char \*path)读取后返回文件地址。
16. 读取功能：通过FileDialog(char \*path)返回的文件地址，先判断是否读取了文件，如果读取了就生成一个文件对象fp，然后使用数据块读取方式fread(p1,sizeof(node),1,fp)，将文本文件中的内容读取到以p1为头节点的链表中。这样通过循环将相应的数据读取到各个链表中。并且在循环中通过!feof(fp)判断是否读到了文件末尾。
17. 保存功能：通过FileDialog(char \*path)返回的文件地址，先判断是否打开了文件，如果打开了就生成一个文件对象fp，然后使用数据块写入方式fwrite(head,sizeof(node),1,fp)，将以head为头节点的链表中的数据分块的写入文件。这样通过循环将相应的数据写入到各个链表中，直到链表末尾没有数据。

## 数据结构设计

宏定义：



## 函数设计描述

函数原型：CreateRec()

功能描述；创造矩形的链表里的一个节点

参数描述： 无

返回值描述：无

重要局部变量定义：static int createrecpoint = 0;

重要局部变量用途描述：用于程序判断是第几个点

函数算法描述：通过嵌套的分支判断

与3.2中模块的对应关系：相仿

原型和定义保存在哪个.c/.h文件中：geo\_funcs03.h\geo\_funcs03.c

## 源代码文件组织设计

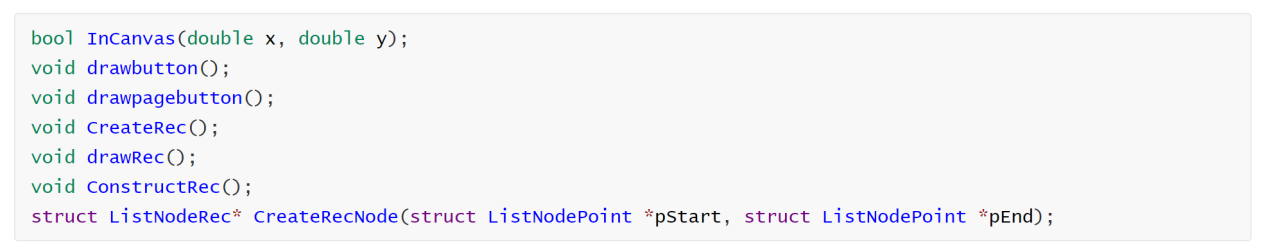
<文件目录结构>

1. 文件函数结构

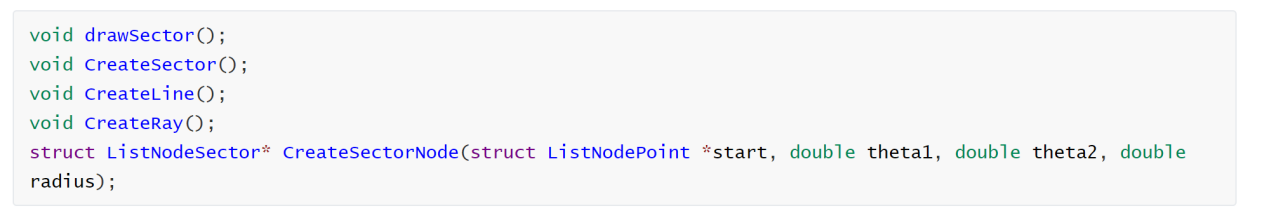
geo\_funcs01.c



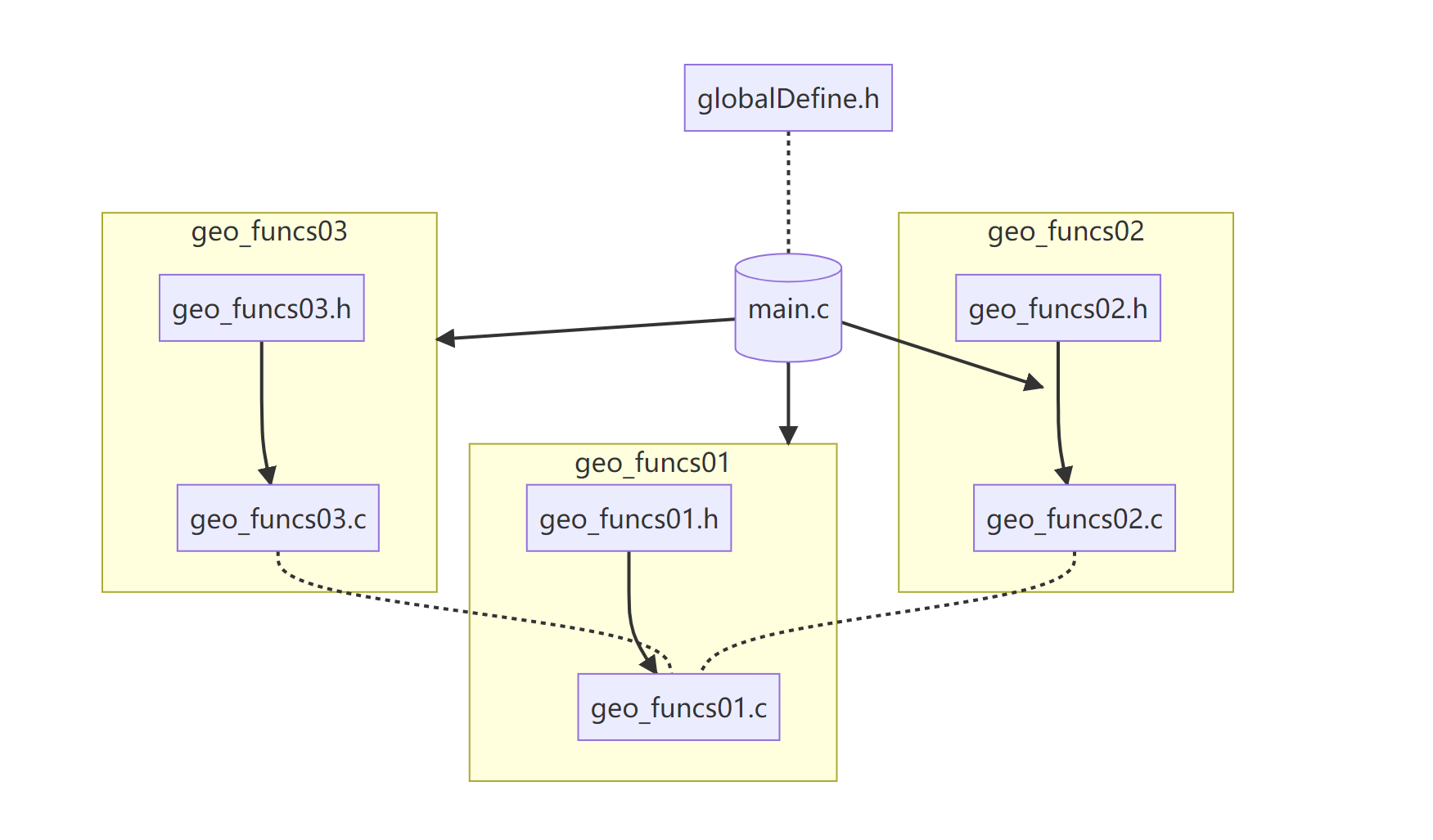
geo\_funcs03.c



geo\_funcs02.c



2）多文件构成机制

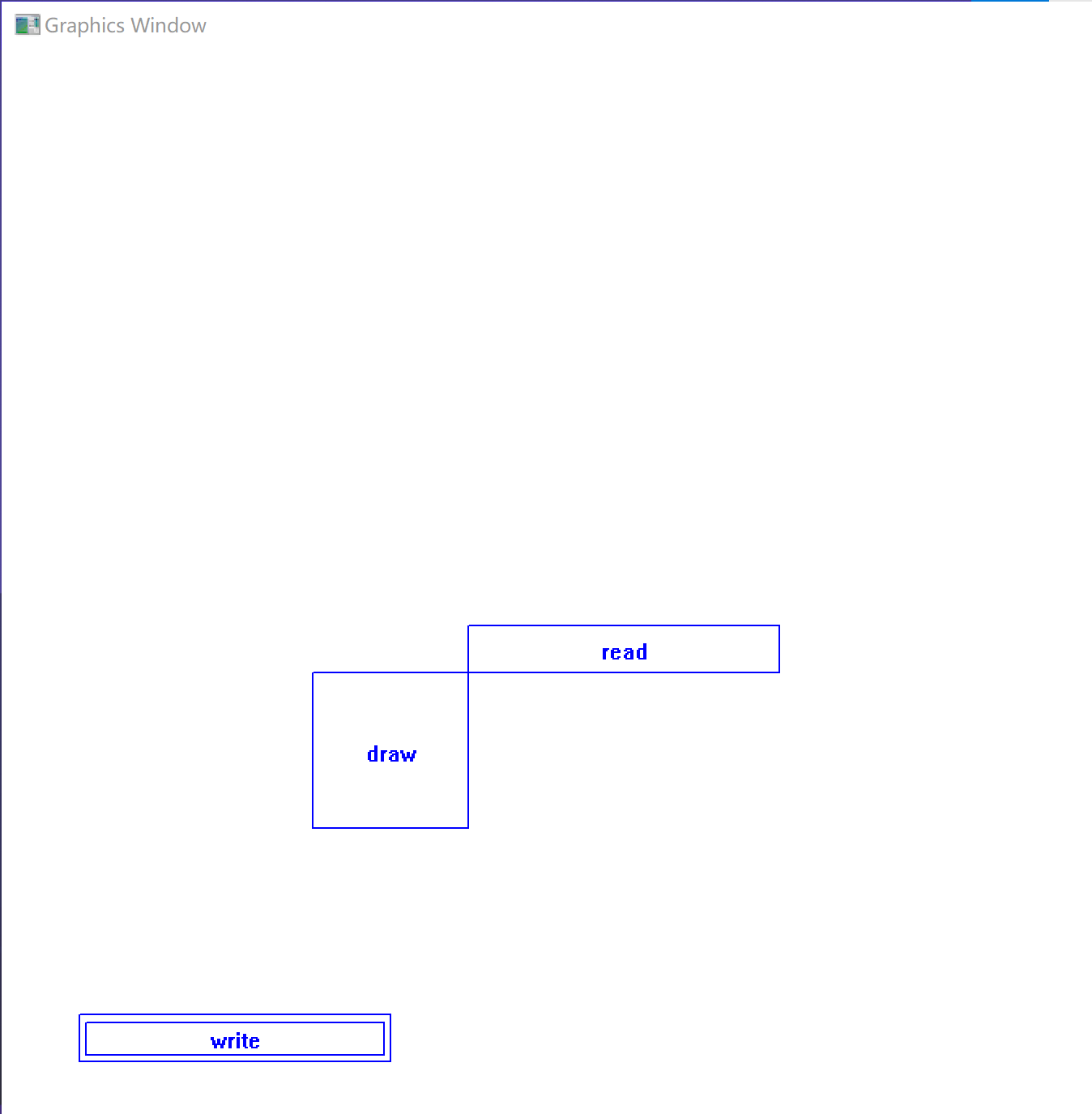


# 部署运行和使用说明

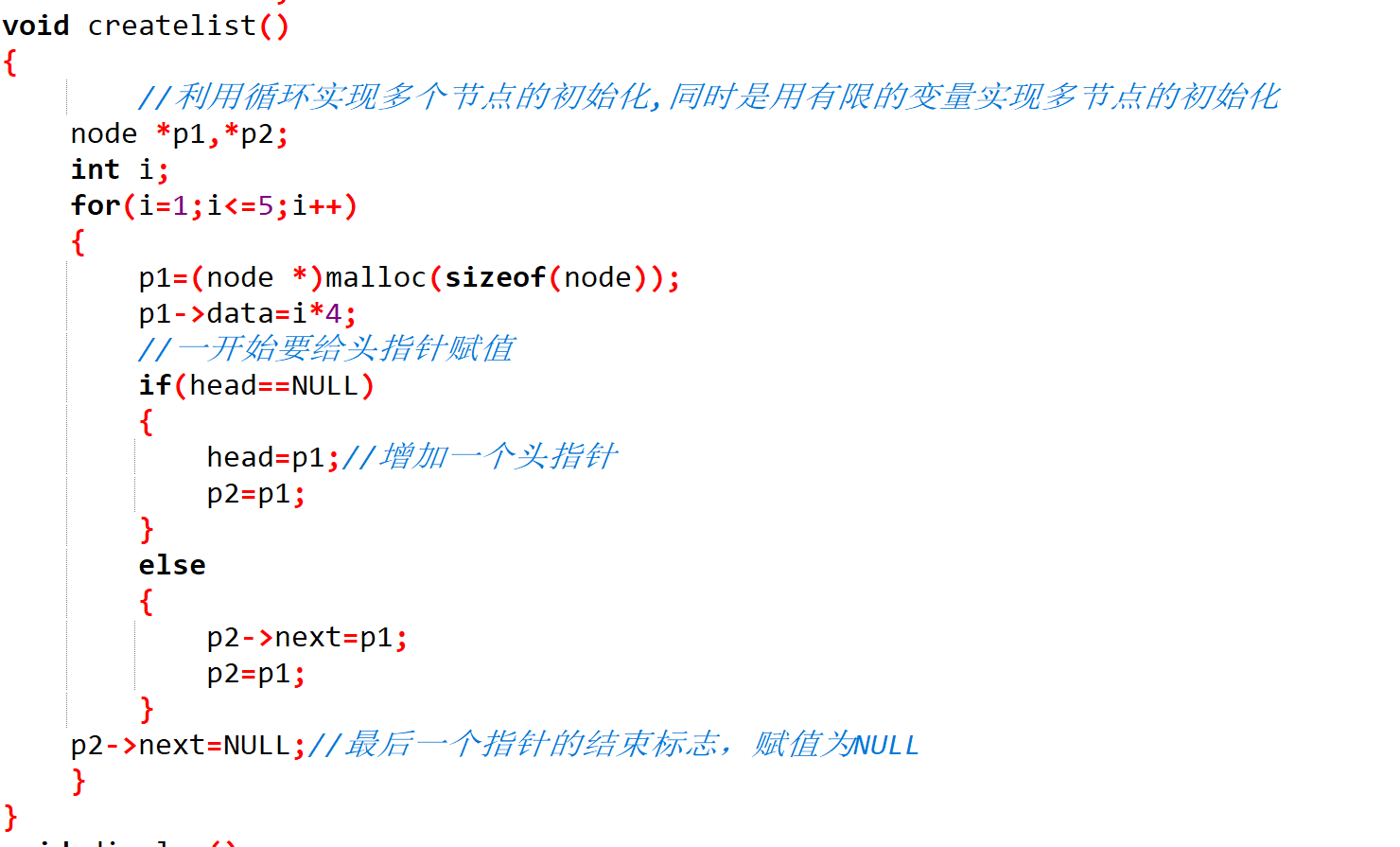
## 编译安装

将压缩包解压后，在..\devs中打开Free Geometry Tool.dev后按F11即可完成源代码编译，生成可执行文件。

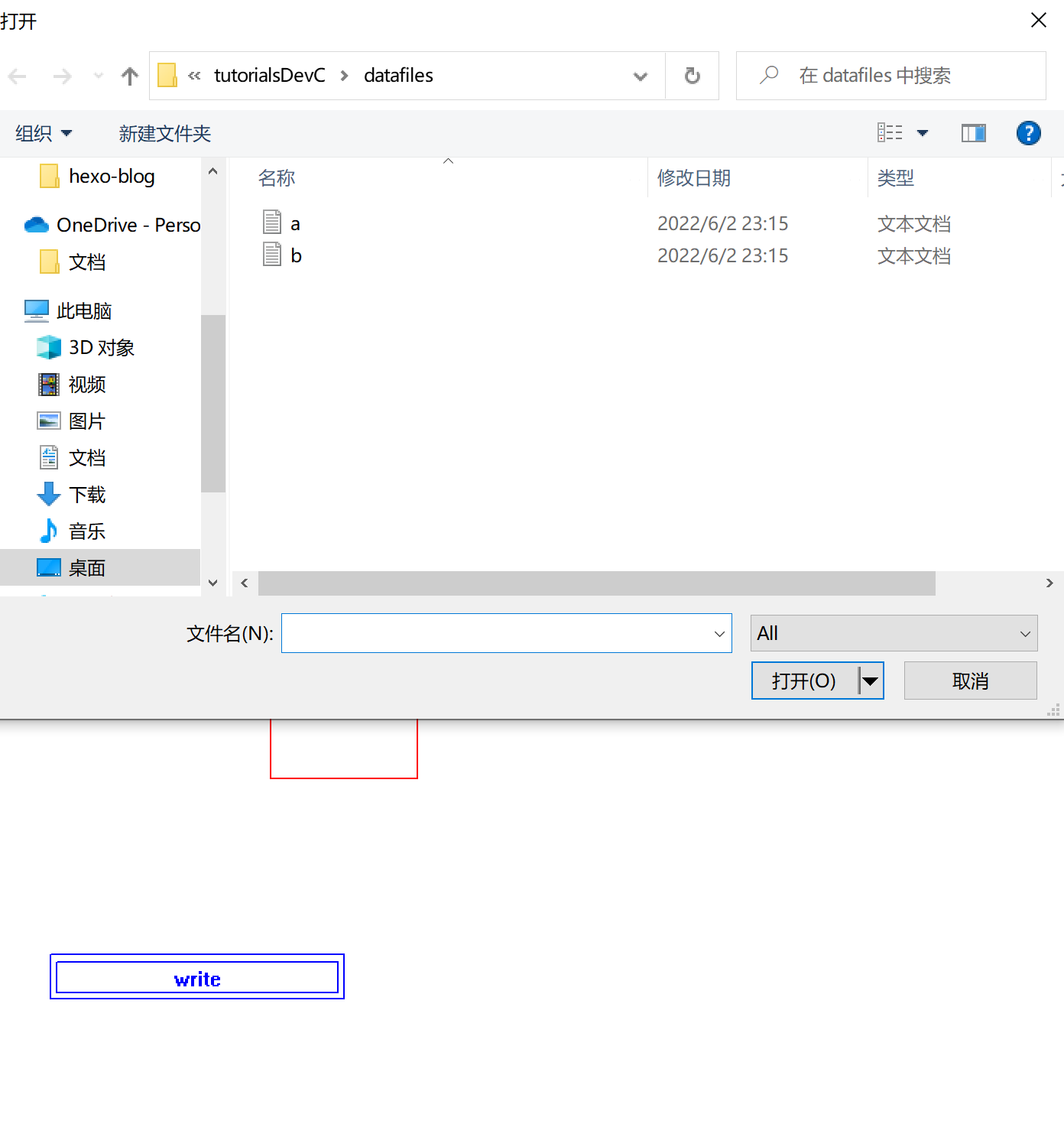
## 运行测试



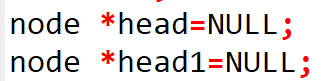
通过单独一个工程文件测试文件的保存和读取的实现



先写一个函数自动生成一个有内容的链表



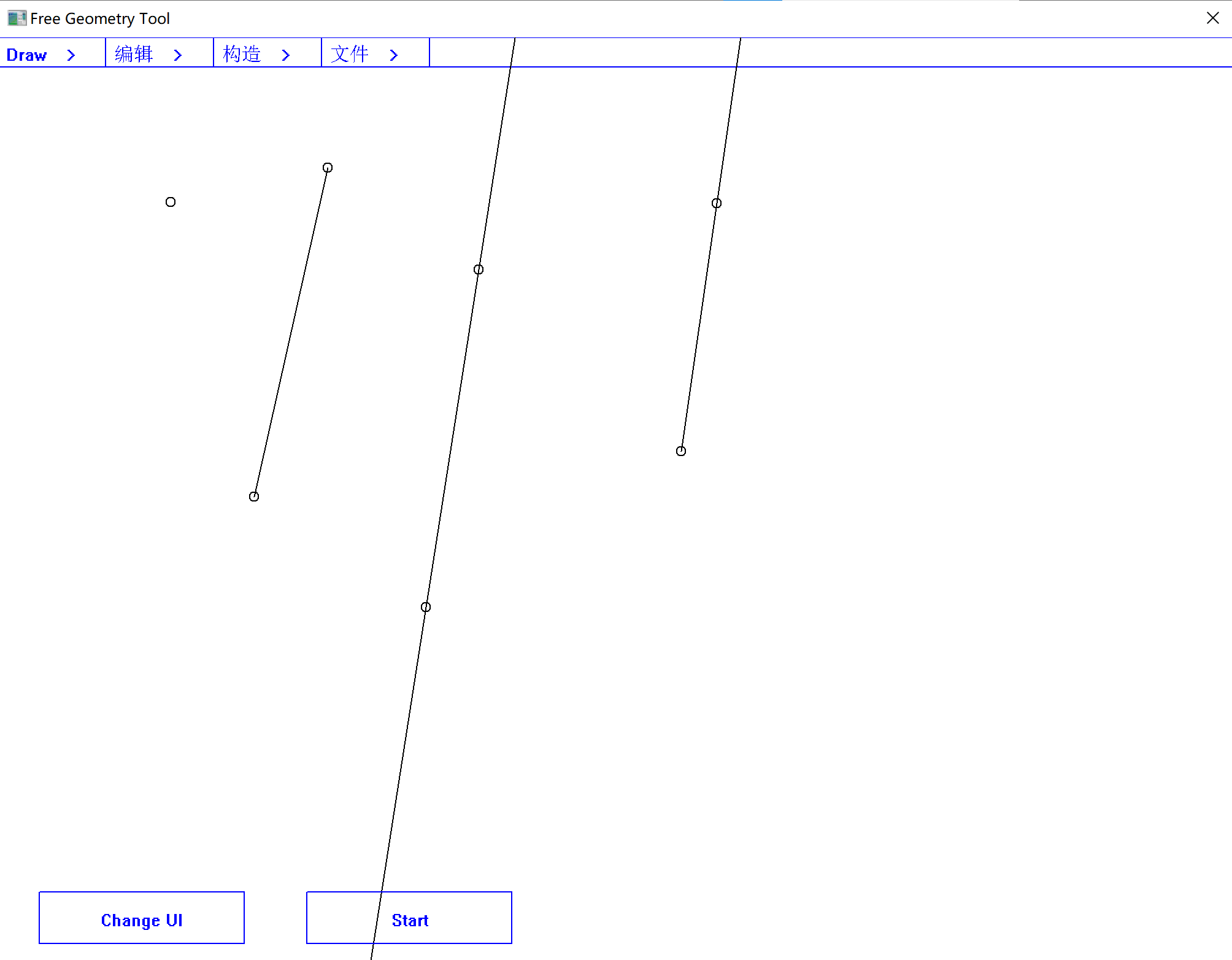
然后点击write按钮写入a文件



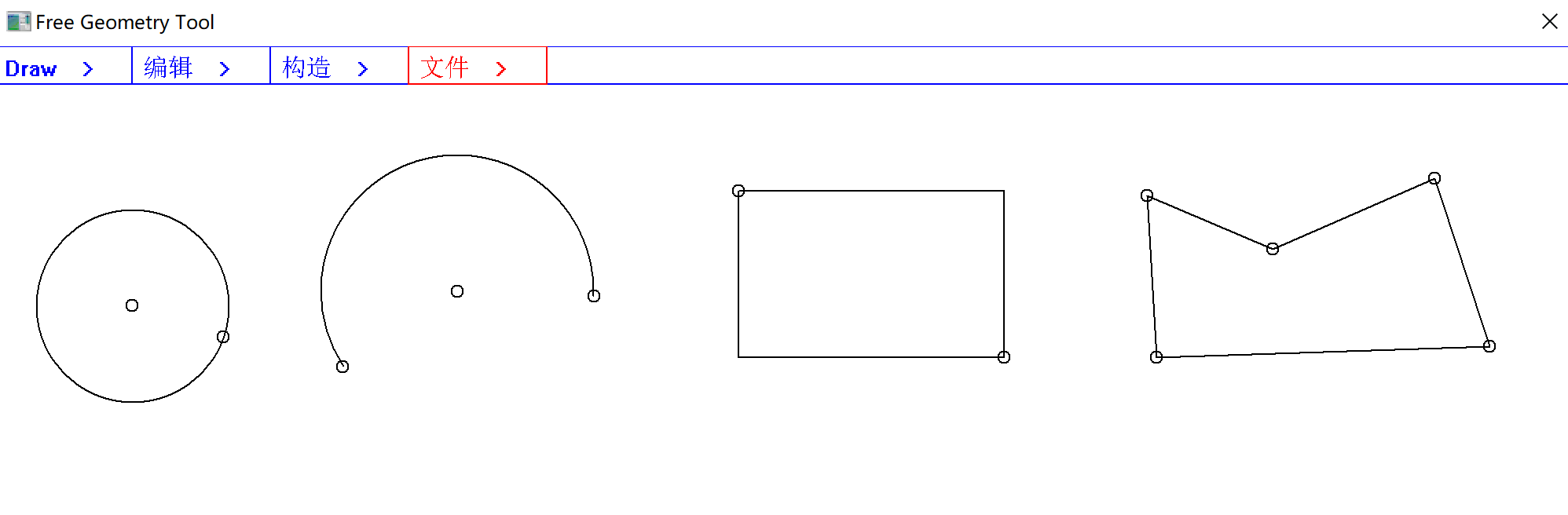
同理,能够点击read按钮从a文件中读取数据到以head1为头结点的链表中(head是自动生成的用来测试的链表的头结点)

## 用户使用手册

用户在绘图中可以选择不同的绘图功能，选定绘图功能后就会自动进入绘图的界面，用户可以随时点击屏幕下方的End按钮结束绘图状态。

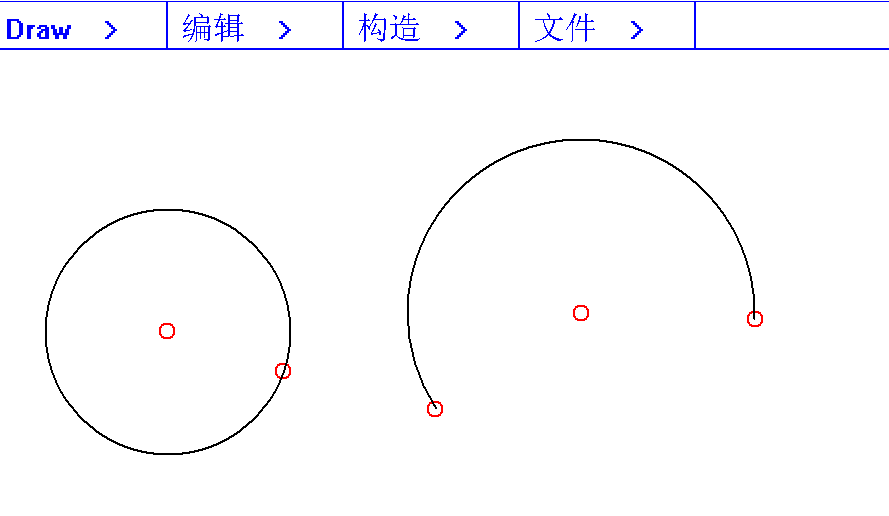


（上图从左到右，依次为点、线段、直线、射线的实现效果，小圆点代表鼠标点击过的地方，差不多是端点的作用）

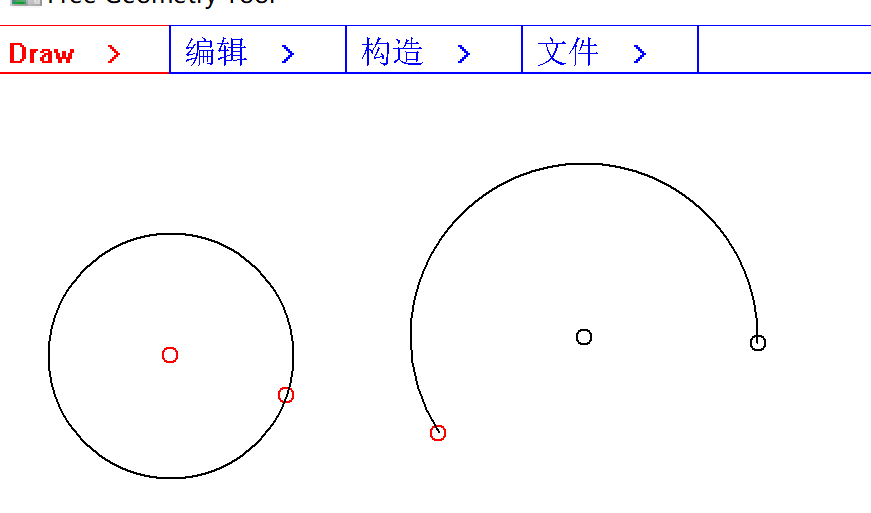


（上图从左到右，依次为圆、圆弧、矩形、多边形的实现效果）

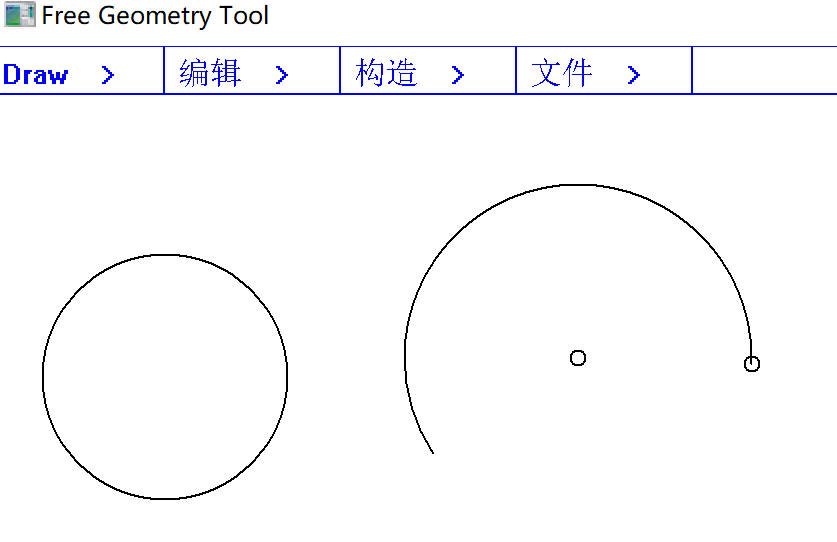
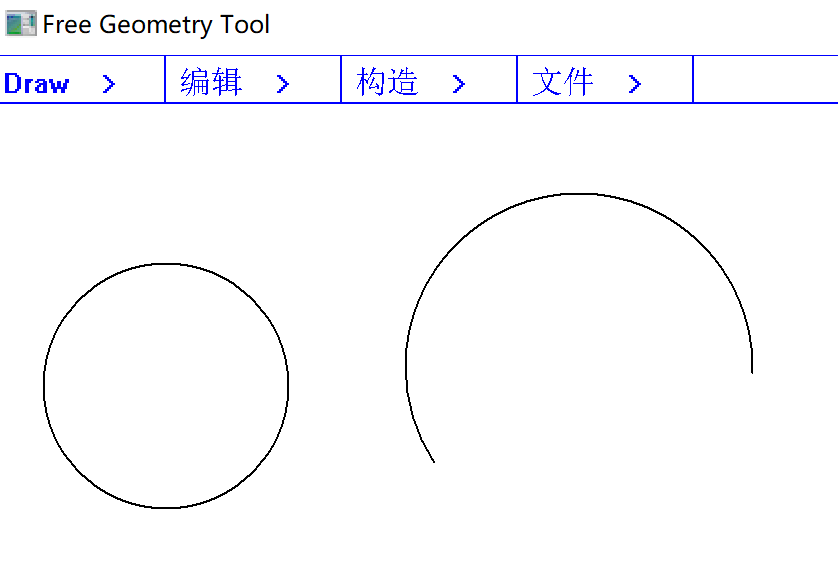
编辑：



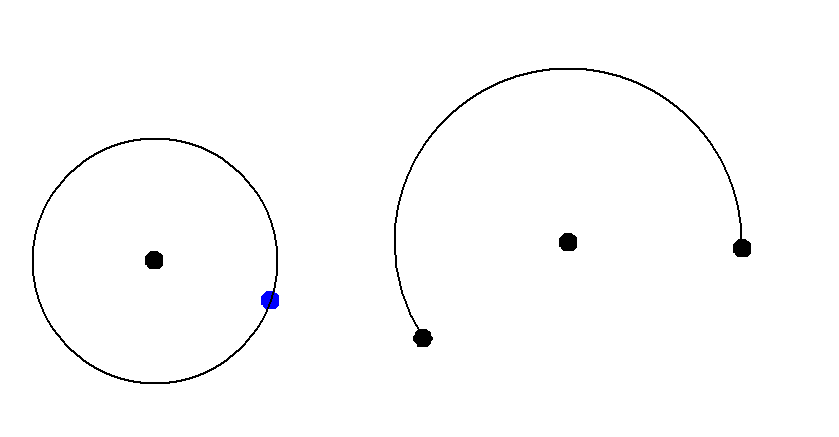
（全选功能能够选中当前画布中的所有点，这也是编辑中其他操作的基础）



（当然也可以通过鼠标点击，任意选取想要选取的点来进行编辑中的操作）

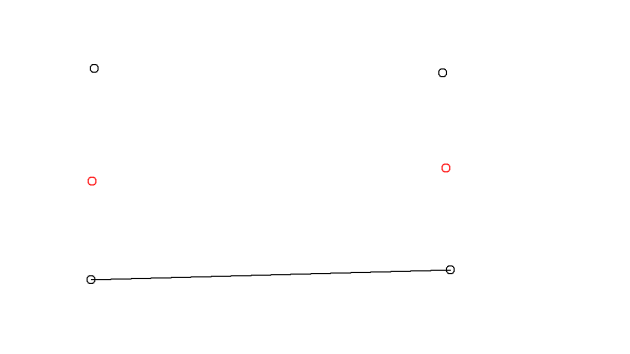
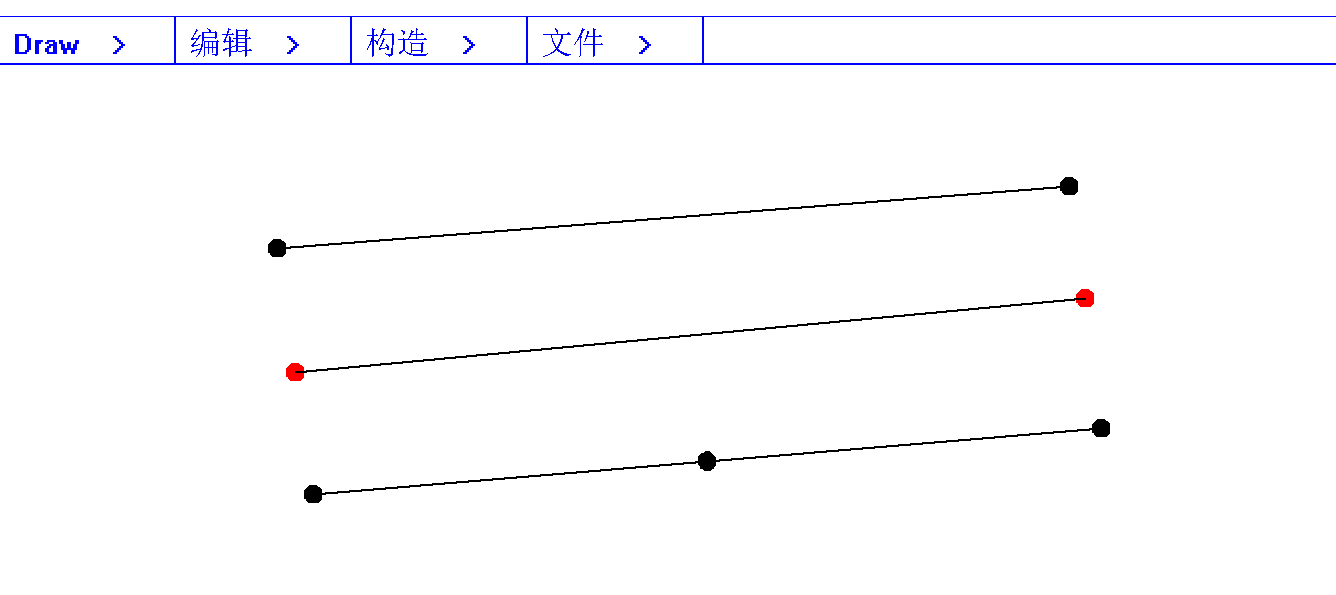


（点击隐藏后，被选中的点会被隐藏；点击显示，被隐藏的点就会显示出来；隐藏的点在操作中不会对正在绘图的功能有影响）



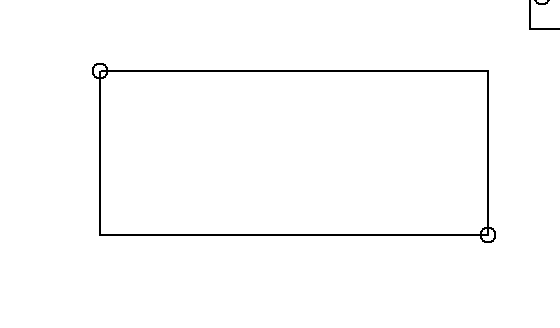
（点型的实心效果如上图所示，更改后，所有的点的样式都将是实心的，不论是否被选中；颜色功能能够自定义选中点之后，被选中的点呈现什么颜色）

构造：



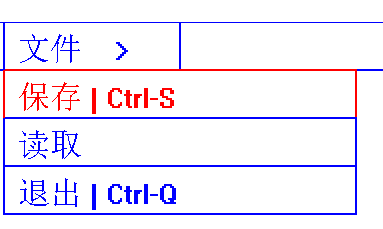
（从上到下分别代表了每一步骤的样式，从一开始的直线到最后的构造中点）

构造的使用方法：先选中画布上已经生成的两个端点，然后点击相应的构造功能，就能构造相应的几何图形



（如图是矩形的构造，通过选取两个点实现）

文件：



（保存能够将画布上的图形保存到文本文件中；读取能够从文本文件中读取数据；退出即退出程序）

# 团队合作

## 开发计划

利用libgraphics和simplegui，结合学校教授的C语言知识，开发一个类似于几何画板的Free Geometry Tool

## 编码规范

变量名采用“名词”或者“名词+形容词”的形式如：windowWidth

函数名采用动词的形式，以小写字母开头，后面穿插大写字母如：playBgm（）

## 任务分工

**甲:**做出初期的基本框架，构想了绘制点、线，编辑，构造的基本思路并得到实现，随后又完成圆、多边形功能，设计不同页面的切换，主页页面的顶部菜单，优化规范化了代码。

**乙:** 第一次分工选做功能模块时选择了圆和圆弧的部分，完成了相关函数与链表的工作；

第二次分工时选择将线的功能进行拓展，分化成直线，射线与线段；

第三次选择对界面的配色进行调整并共同撰写了实验报告。

**丙:**作为小组长,负责大家的分工和统筹;在前期规范了工程编写;在每周有序推进代码速度和实现思路。编写了画布范围函数、bgm播放函数、矩形绘制函数和网格线实现功能等。

## 个人遇到的难点与解决方案

### 5.4.1甲

1. 结束绘图时会在菜单处留点。

在界面底部加入“结束”按钮，点击按钮后DrawFlag变为Mouseflag，避免在菜单处留点。

1. 在按钮处仍然会留点。

把“结束”按钮置于画布之外，引入变量FinishFlag，开始作画时FinishFlag变为1，结束作画时FinishFlag变为0，只有当FinishFlag等于0时才能按下按钮。

1. 在选择点时，被选中的点chosenflag无法改变。

引入变量uselessflag、meaninglessflag与choosingflag，其中uselessflag用于初始化点结构q，meaninglessflag用于获取点的原有chosenflag，choosingflag用于存储点的原有chosenflag。

1. 在大多数绘图函数中需要创造点。

加入CreatePointNode函数，获得点的横坐标与纵坐标后，即可把该点数据加入链表，同时简化了大量代码，后续又利用相同原理修改了其余功能的代码。

1. 在设计初期，编译后常常闪退。

将链表的头结点做初始化处理，将其指向的下一个结点等于NULL。

1. 点击菜单或按钮后，需要鼠标或键盘做出动作后才能改变画面。

在每一次点击成功之后，加入Display函数，确保界面及时刷新。

### 5.4.2 乙

（1）对队友代码理解存在困难，导致难以利用队友写的函数。

询问队友具体函数语句的实现方式，加强交流，并添加相应的注释，绘制函数调用逻辑图加以理解。

（2）起始无论如何选择圆心与起始点，绘制的圆弧均为一个固定的小度数。

在头文件中查看绘制圆弧所调用函数的具体实现方式，发现数学函数如sin，cos，atan2函数等对于角的计算时均采用弧度制，而在图形库函数DrawArc中采用的是角度制，存在转换问题。于是引入pi的宏定义对角度进行转换。

（3）在圆弧绘制中发现只能绘制劣弧，优弧绘制时会变成从另一个方向绘制劣弧。

查看头文件中DrawArc函数的参数含义，参数范围，以及在计算角度时所引用的反三角函数使用规范。发现atan2比atan更能满足圆弧绘制的需要，更换反三角函数。

（4）在绘制过程中，透明菜单与按钮会透出后方图案。

对菜单和按钮进行纯色填充，盖住背后图案内容。

### 5.4.3 丙

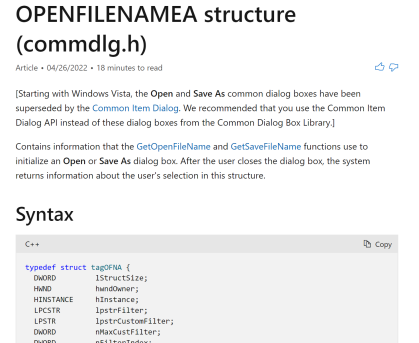
1. 已经使用的变量名与系统的某一个头文件产生了重复定义。

使用Ctrl+F将文件中的变量名一次全部替换。

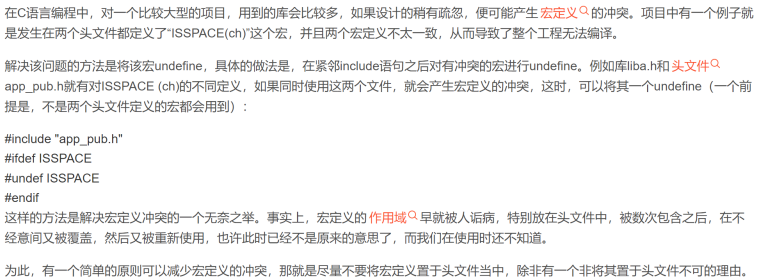
1. 一开始项目环境的配置。

去学习了什么是项目，一个项目可以有哪些设置，哪些是我们常用的，如何才能引用外部的库。

1. 文件的输入读取。
2. 开了一个新的项目独立调试
3. 参看了微软官方文档



1. 宏定义和全局变量冲突的问题



## 合作总结

2022.5.23

1.在Main中的malloc函数之后增加初始化，避免了编译后闪退

2.在geo\_funcs03.c中的drawbutton()函数中增加Display()，避免按下按钮后不发生界面变化

3.修改了ConstructMiddlePoint和ConstructLine的构造方式，避免一个或者多个点被选中的情况

4.修改了drawpoint中被选中的点的颜色改变方式，避免线段变色

5.修改了drawsector的作画方式，修改了CreateSector

画布的问题我想到一个方法，在右下角放一个按钮，按下之后选择一种作画方式，开始作画。如果想结束的话就必须让作画的步骤结束，比如说画线不能只画一个点就结束了，然后按一下按钮回到主菜单。

2022.5.25

1.修改了DrawSector中pStart2与pStart1重合的问题

2.修改了画布的问题，增加了start end按钮，改变了drawpage方式

2022.5.27

1.完成了多边形功能

2.随笔画功能可能做不出来

3.需要调节changeui的颜色配置，部分颜色会让点处于隐藏状态

2022.5.31

1.用宏定义DrawFlag改变了不同作画方式的选择

2.增加了drawthepoint函数，实现了实心点、空心点的选择与颜色的选择

3.修改了ListNodeLine的内容，把线段、直线、射线都归纳于其中

4.CreatePoint上做了一些小改变

5.实现了点的隐藏显示，是另外意义上的删除功能

2022.6.1

1.在点的选择时做了一些修改

2.实现了多边形开放与闭合的选择

2022.6.6

1.粗线细线的选择

2.StartPage的修改

3.配色选择

2022.6.10

程序中可以打开使用手册

## 收获感言

甲：几何画板是我从初中开始用的软件，但现在我却能用自己的能力用C语言做出一个类似的工程，而且我对这个项目很感兴趣，当我看到这是候选选题时，我便果断选择。我在设计大程序中学到了很多，由于课上关于图形库的知识很缺乏，我参加了朋辈辅学，在学长那里也获得了很多知识，还有翁恺老师的网课也让我收获颇丰。在小组中，我前端后端都有涉及，独自完成了前期开发，经过组员的帮助与指点，我进一步打磨了整个程序，解决了很多难题，不断增加新功能，整合并简化代码，尽可能地做到代码的规范化。但受到选题的限制，我还有很多编程领域没有触及，希望能通过继续学习丰富我的知识。

乙：从选题到之后的讨论，c大程的小组算是我大学以来合作最愉快的小组之一了，全程都以比较良好的节奏推进，两个队友配合度也都很高。从平时的小作业中链表和图形库学的三脚猫功夫，到现在可以轻松写链表，c大程用实践的方式让我学到了很多。看着几何画板逐渐在我们手中成形，成就感确实是理论课无法带来的。在写大程的过程中，慢慢掌握了读函数的技巧，明白了如何定义变量与函数才比较规范，逐渐也可以看头文件学习未见过的函数的使用方法……在写程序时遇到的困难，经过和队友讨论、向学长学姐讨论等，都能够较好地解决，也算是从一个例子上逐渐掌握了学习一门编程语言的方式。

丙：在《程序设计专题》这么课上，我第一次真正接触到合作写一个Project，我认为在这门课让人收获最多的地方是开发的过程。我认为自己在代码规范上做的还不是很好，以及关于Project的版本管理也不够。这门课让我学到写Project与写程序是完全不同的两件事，一个清晰的架构能够帮助我们顺利的写Project。在写大作业的过程中，我也会经常去复习前面学过的内容，往往能够从新的视角去掌握前面的知识和应该在哪里使用什么样的方法。遇到一些书上没有的问题，也尝试着如何去查找解决方案，该如何解决。虽然自己写的有些功能最后不能顺利融入Project，但这也就是因为前面说的没有规划好。最后还是很感谢队友们的合作和帮助。

# 参考文献资料

《C程序语言设计》 何钦铭 颜晖 主编

<http://paulbourke.net/geometry/>