南开大学 计算机学院

姓名 林语盈

学号 2012174

班级 2020级计算机卓越班

2021年5月3日

高级语言程序设计

实验报告

目录

[高级语言程序设计大作业实验报告 2](#_Toc70960787)

[**一.** **作业题目** 2](#_Toc70960788)

[**二.** **开发软件** 2](#_Toc70960789)

[**三.** **课题要求** 2](#_Toc70960790)

[**四.** **主要流程** 2](#_Toc70960791)

[**1． UI设计** 2](#_Toc70960792)

[**2． 用例设计** 3](#_Toc70960793)

[**（1） 添加元件 3**](#_Toc70960794)

[**（2）添加结点 5**](#_Toc70960795)

[**（3）添加连线 5**](#_Toc70960796)

[**（4）拖动元件 8**](#_Toc70960797)

[**（5）更改元件标记值 8**](#_Toc70960798)

[**（6）更改元件方向 10**](#_Toc70960799)

[**(7) 删除元件 12**](#_Toc70960800)

[**（8）保存 13**](#_Toc70960801)

[**3. 类设计** 15](#_Toc70960802)

[（1）mainwindow 15](#_Toc70960803)

[（2）View 16](#_Toc70960804)

[（3）Storage 16](#_Toc70960805)

[**4. 流程设计(算法及技术要点)** 18](#_Toc70960806)

[（1）新建元件 19](#_Toc70960807)

[（2）元件移动 20](#_Toc70960808)

[（3）元件属性更改 23](#_Toc70960809)

[（4）元件删除 24](#_Toc70960810)

[（5）新建线条 25](#_Toc70960811)

[**五、单元测试** 27](#_Toc70960812)

[测试案例 27](#_Toc70960813)

[测试结果 27](#_Toc70960814)

[**六、收获** 27](#_Toc70960815)

[1. **基础算法是函数库还是自建** 27](#_Toc70960816)

[**2. 程序设计与变更** 27](#_Toc70960817)

[**3. 调试** 28](#_Toc70960818)

[**4. 版本管理** 28](#_Toc70960819)

[**5. 测试** 28](#_Toc70960820)

高级语言程序设计大作业实验报告

1. **作业题目**

电路图绘制软件。

代码仓库链接：[林语盈/CircuitDiagram (gitee.com)](https://gitee.com/lin-yuying/circuit-diagram)

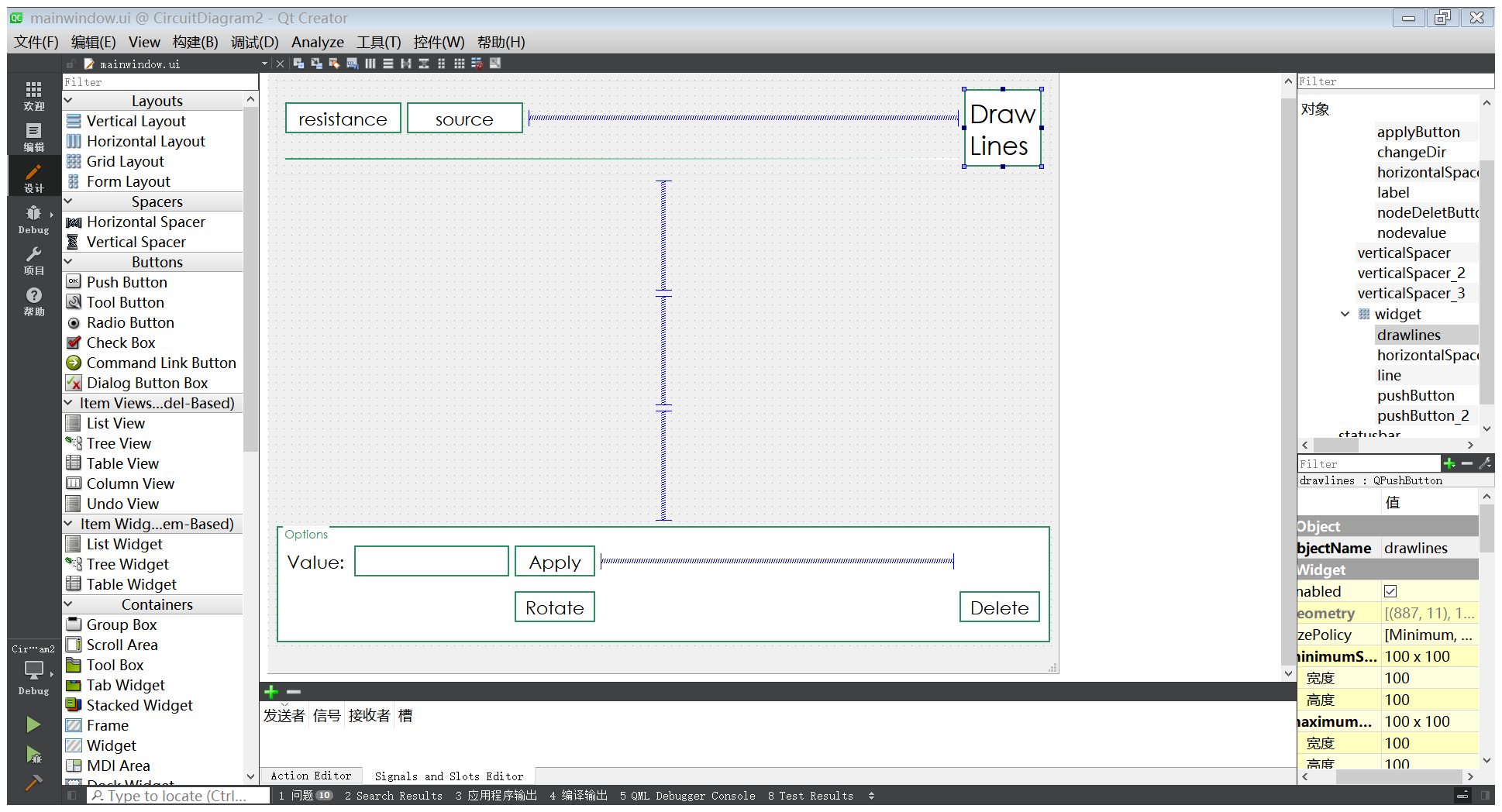
1. **开发软件**

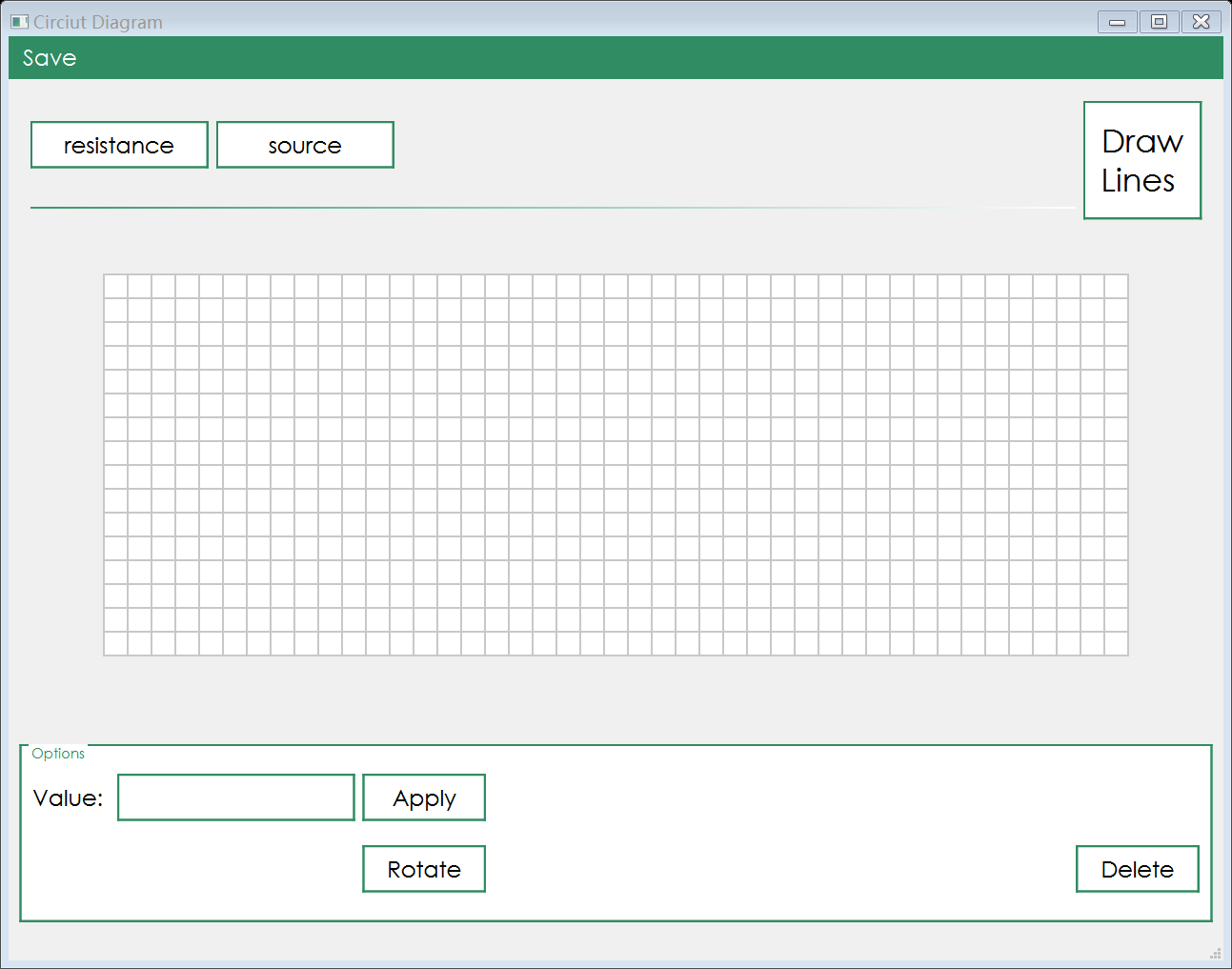
QT 6.0.1

QT creator 4.14.0

编程语言：C++

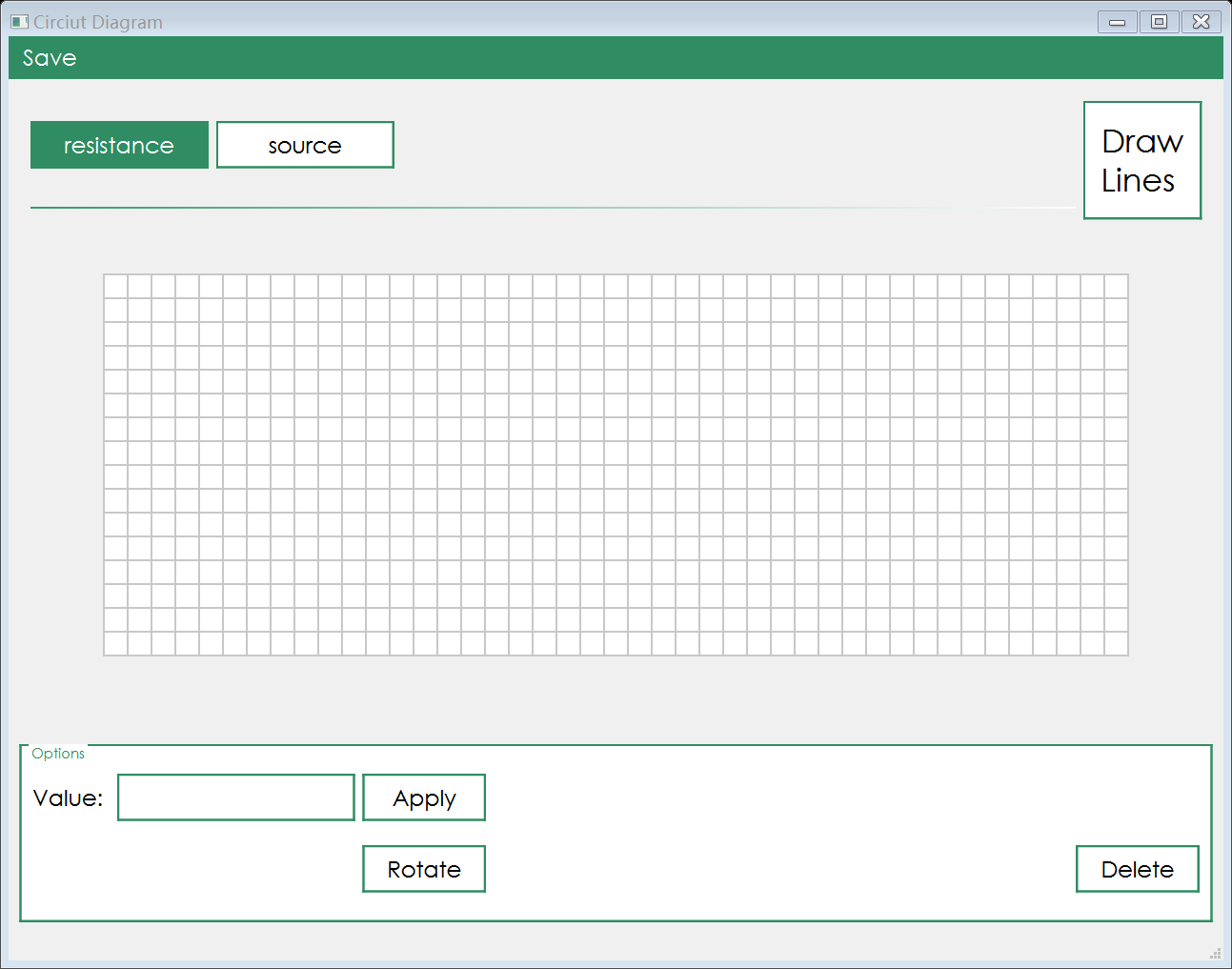
1. **课题要求**
2. 面向对象。
3. 单元测试。
4. 模型部分
5. 验证
6. **主要流程**
   1. **UI设计**

****

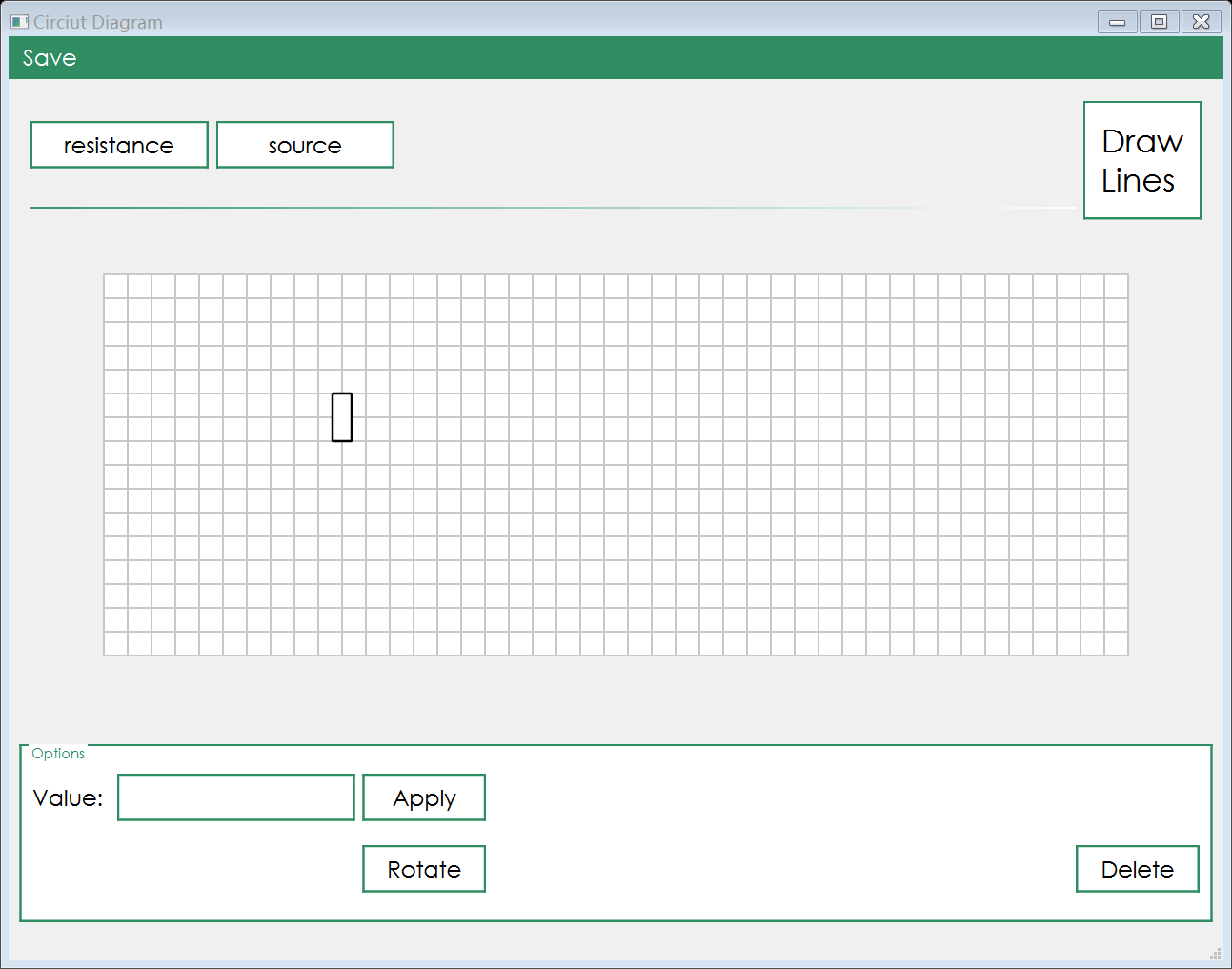
****

* 1. **用例设计**
     1. **添加元件**

1. 左键单击相应元件按钮，按钮进入已点击状态。
2. 点击元件位置，出现原件



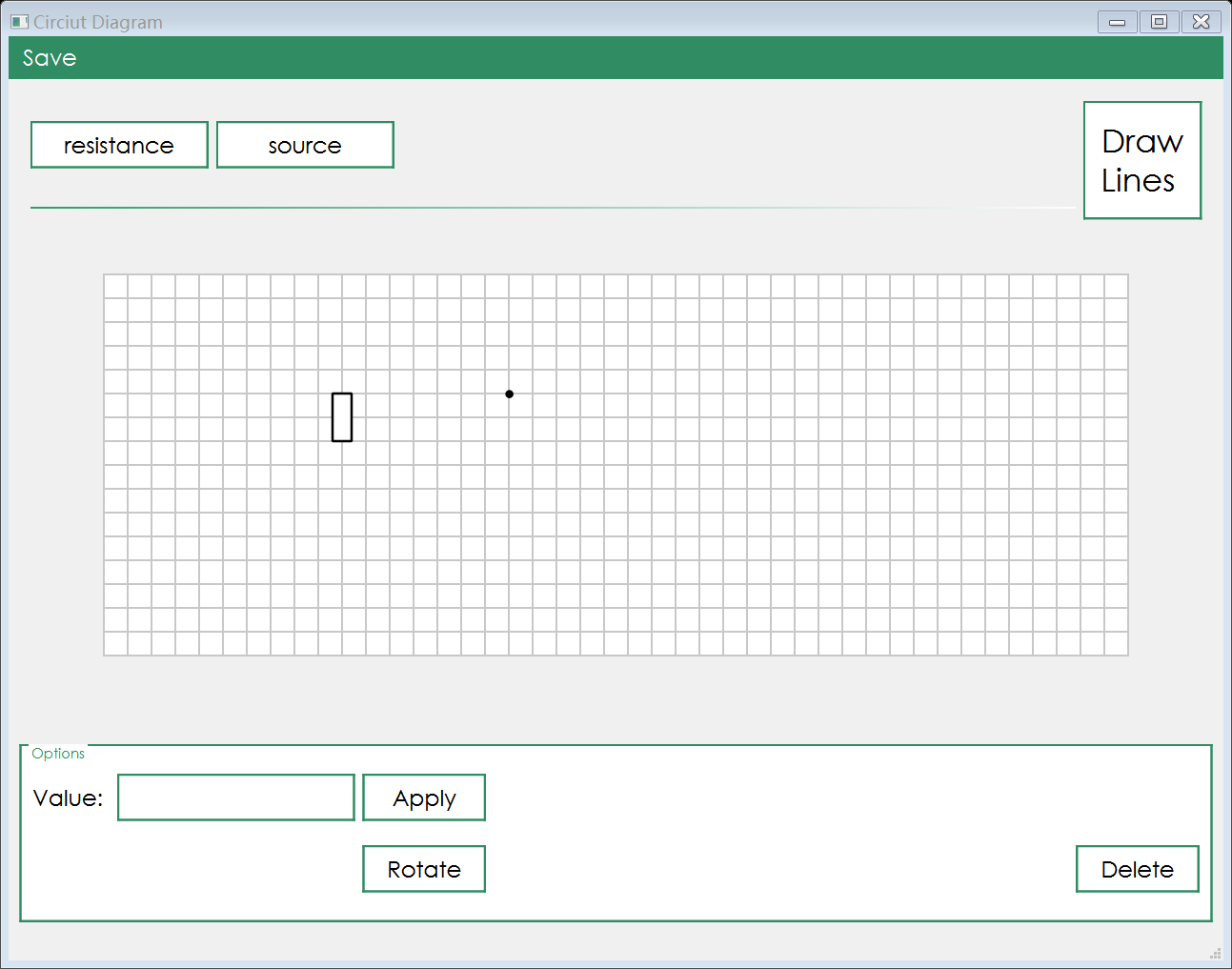
左键单击



左键单击

### (2)添加结点

①在画板区域，单击右键，添加节点。



右键单击

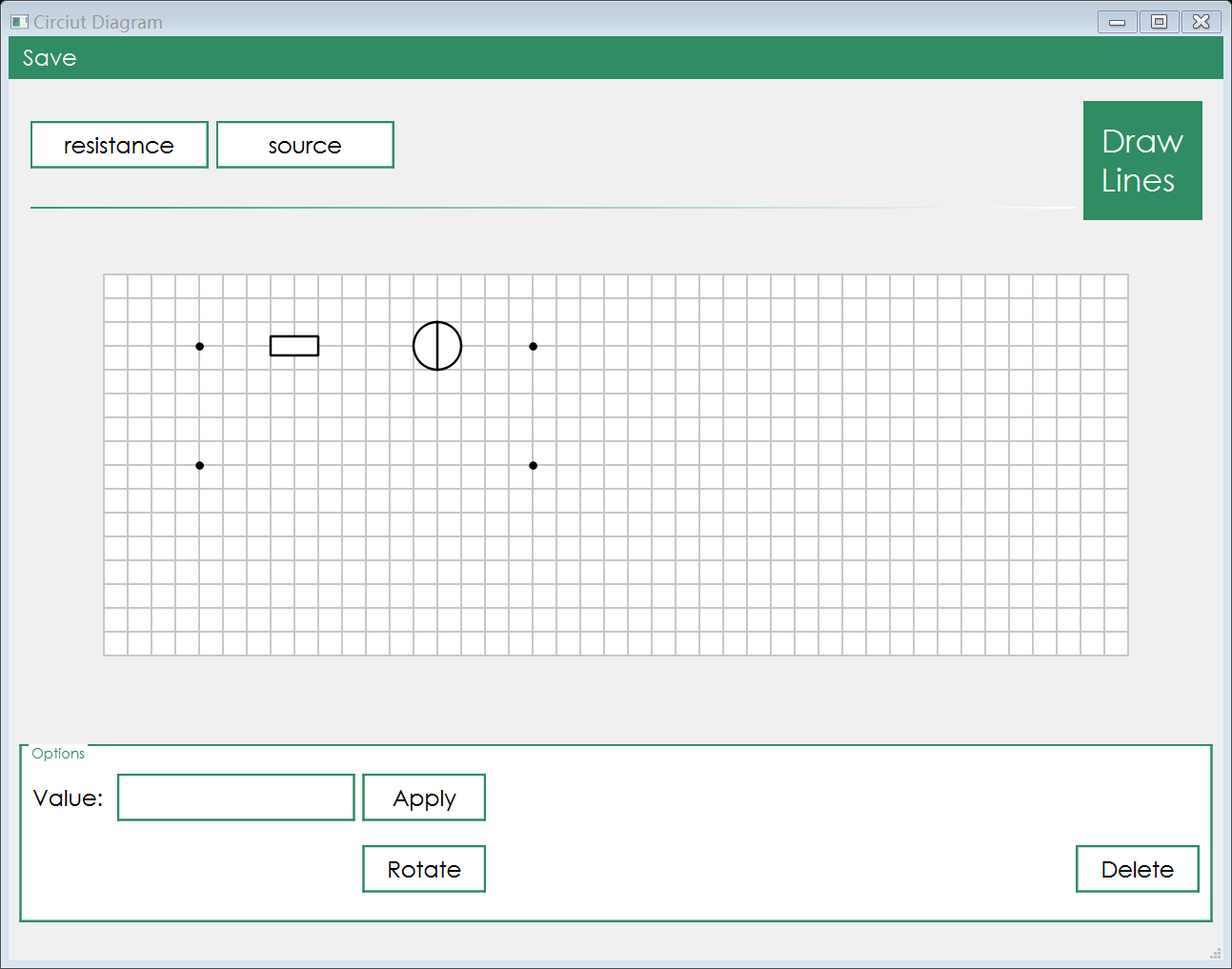
### （3）添加连线

①左键单击“draw lines”按钮，进入连线模式

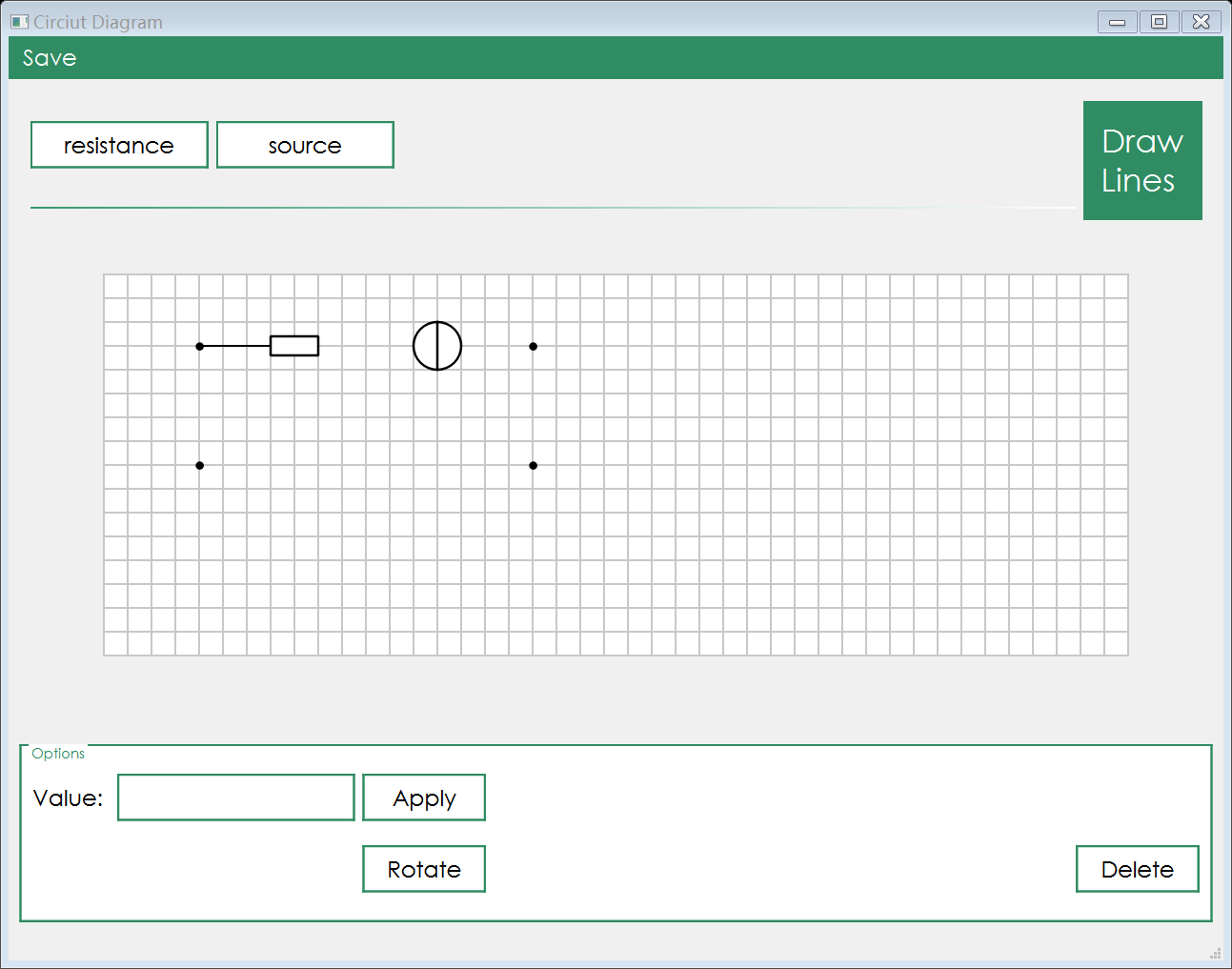
②依次点击结点或元件四周（连线点），连线

③重复上一步操作

④左键单击“draw lines”按钮，退出连线模式

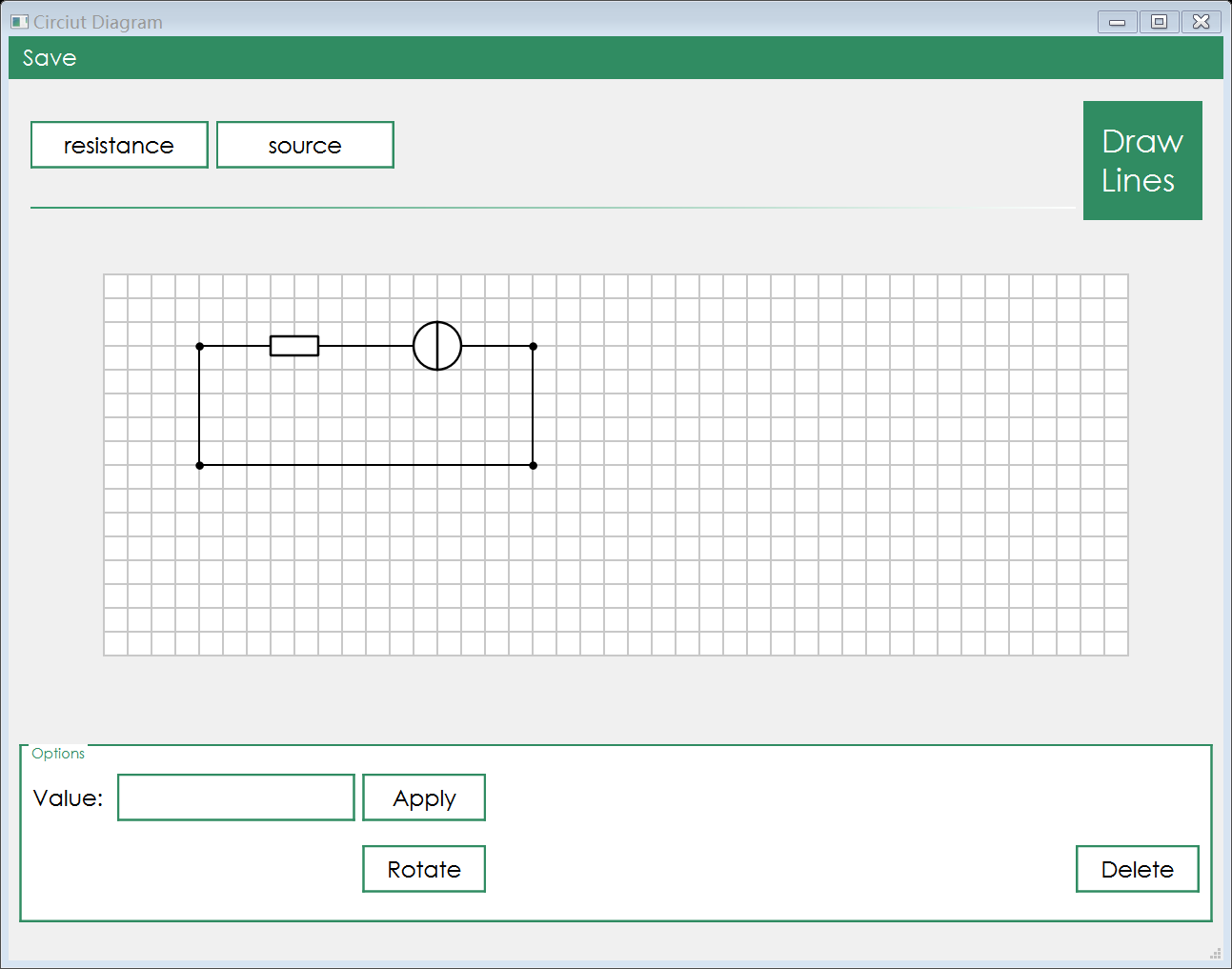


左键单击

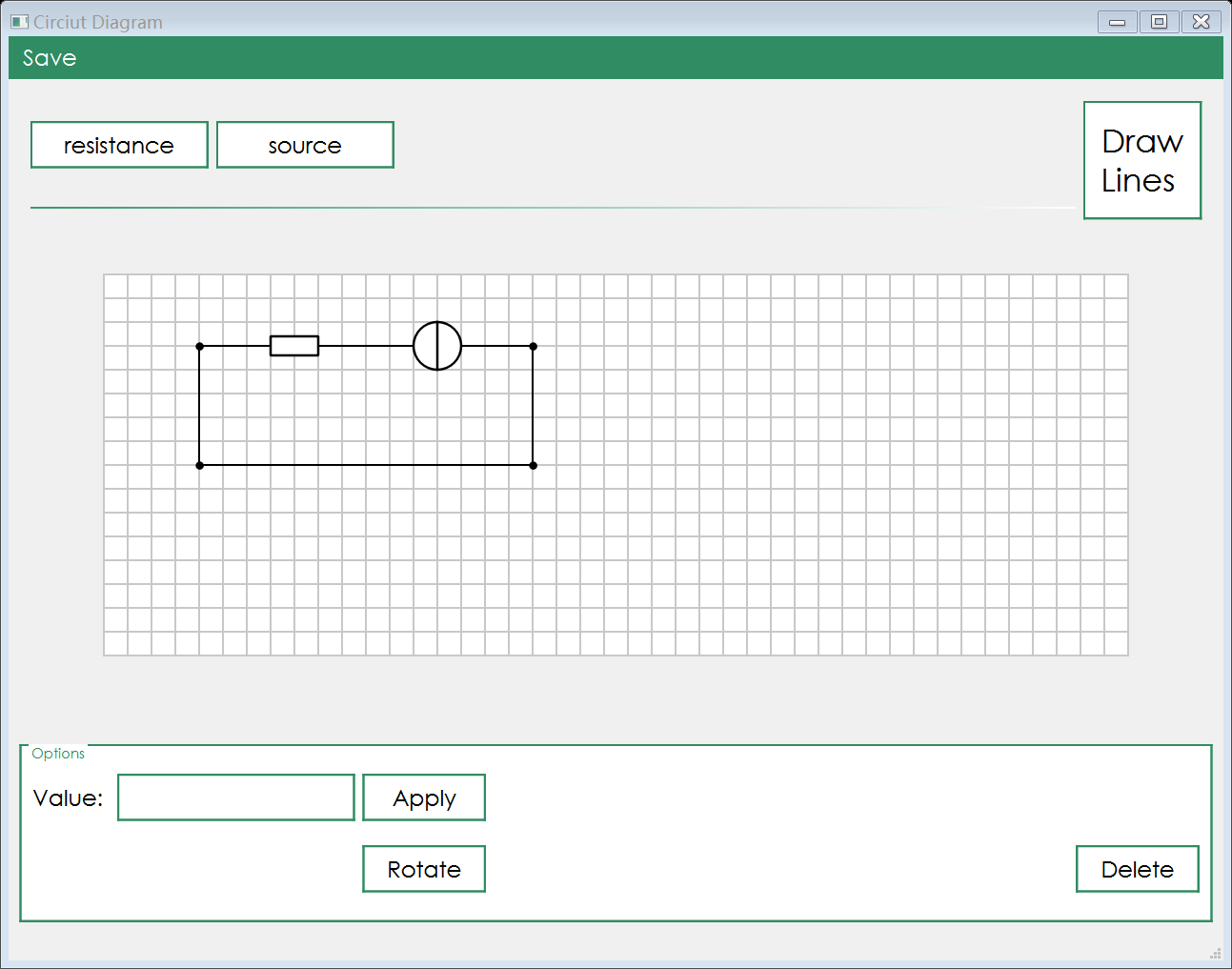


左键单击2

左键单击1



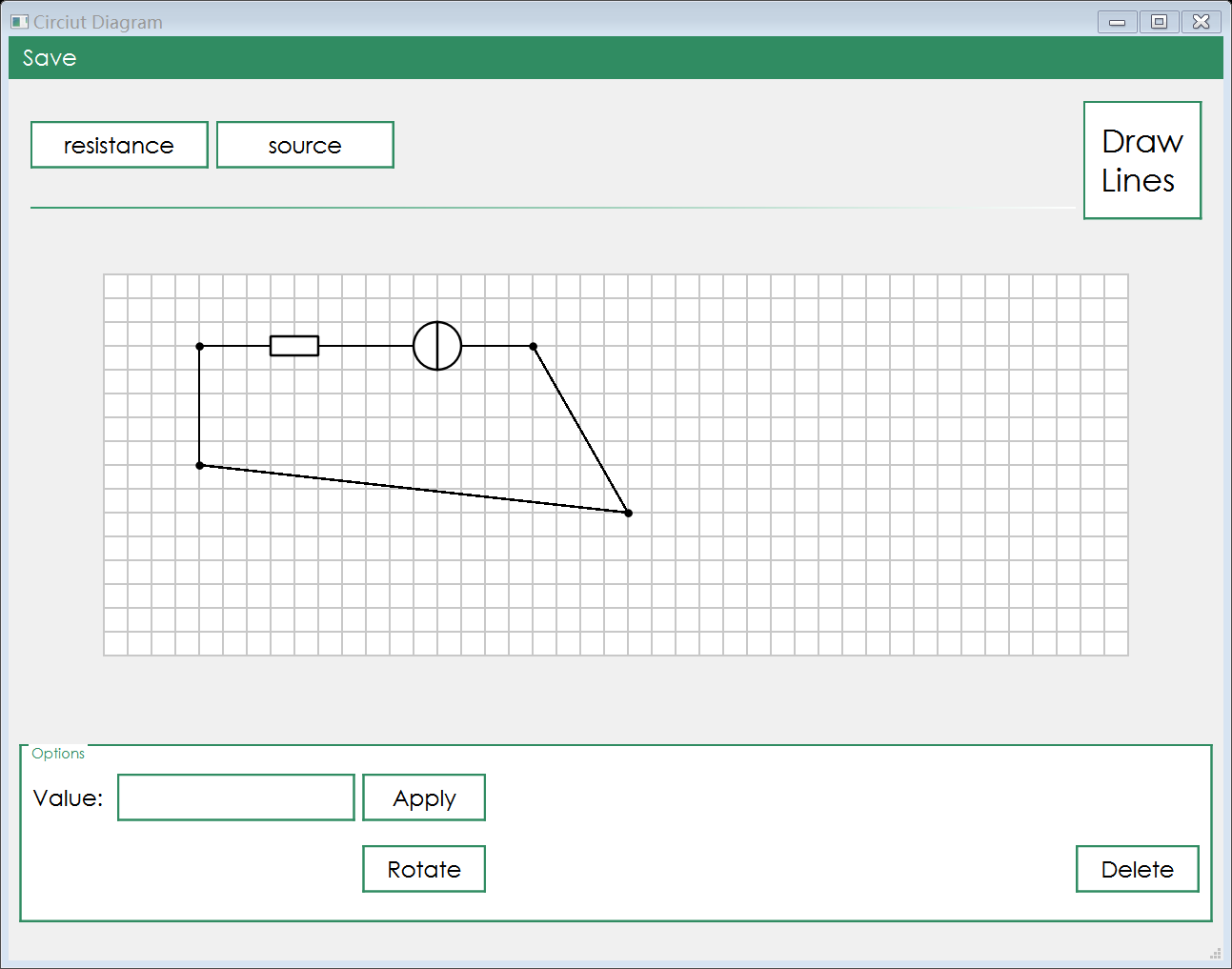
依次添加连线



左键单击，退出连线模式

### （4）拖动元件

①使用鼠标左键拖动元件，元件高亮，连线将自动跟随。



拖动

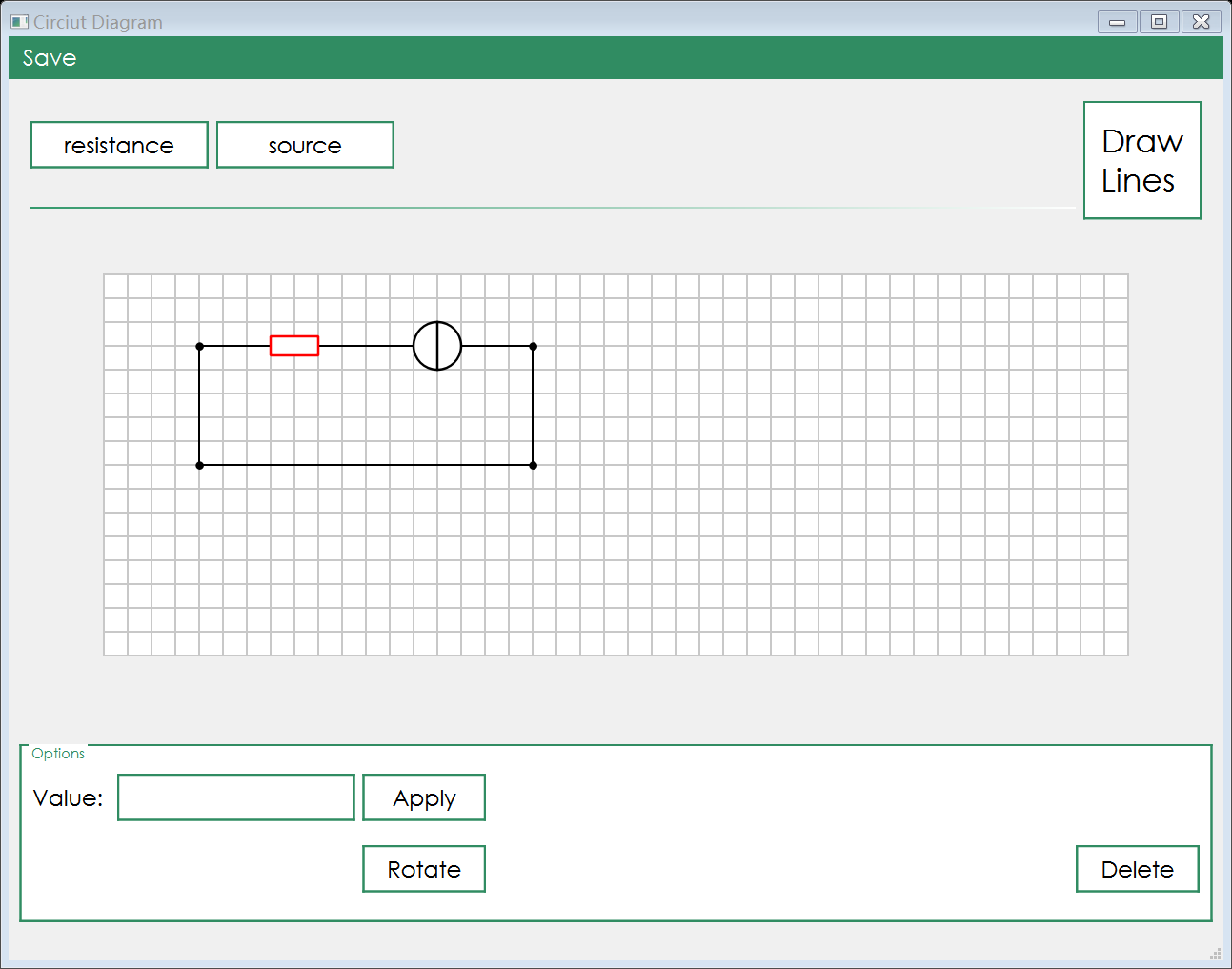
拖动

**（5）更改元件标记值**

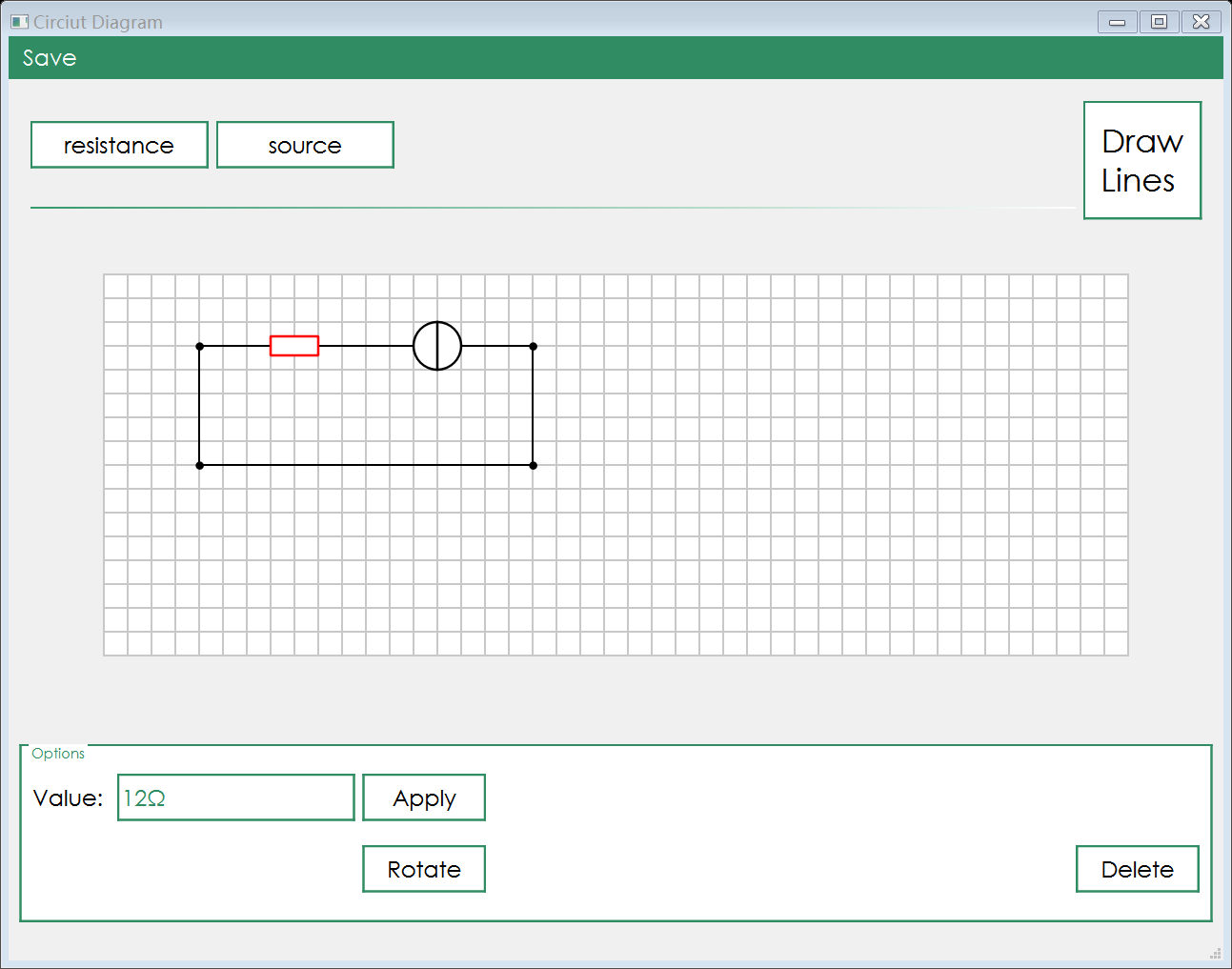
①左键单击选择元件，元件高亮显示

②在“Value”框中输入元件标记值（电阻阻值等）

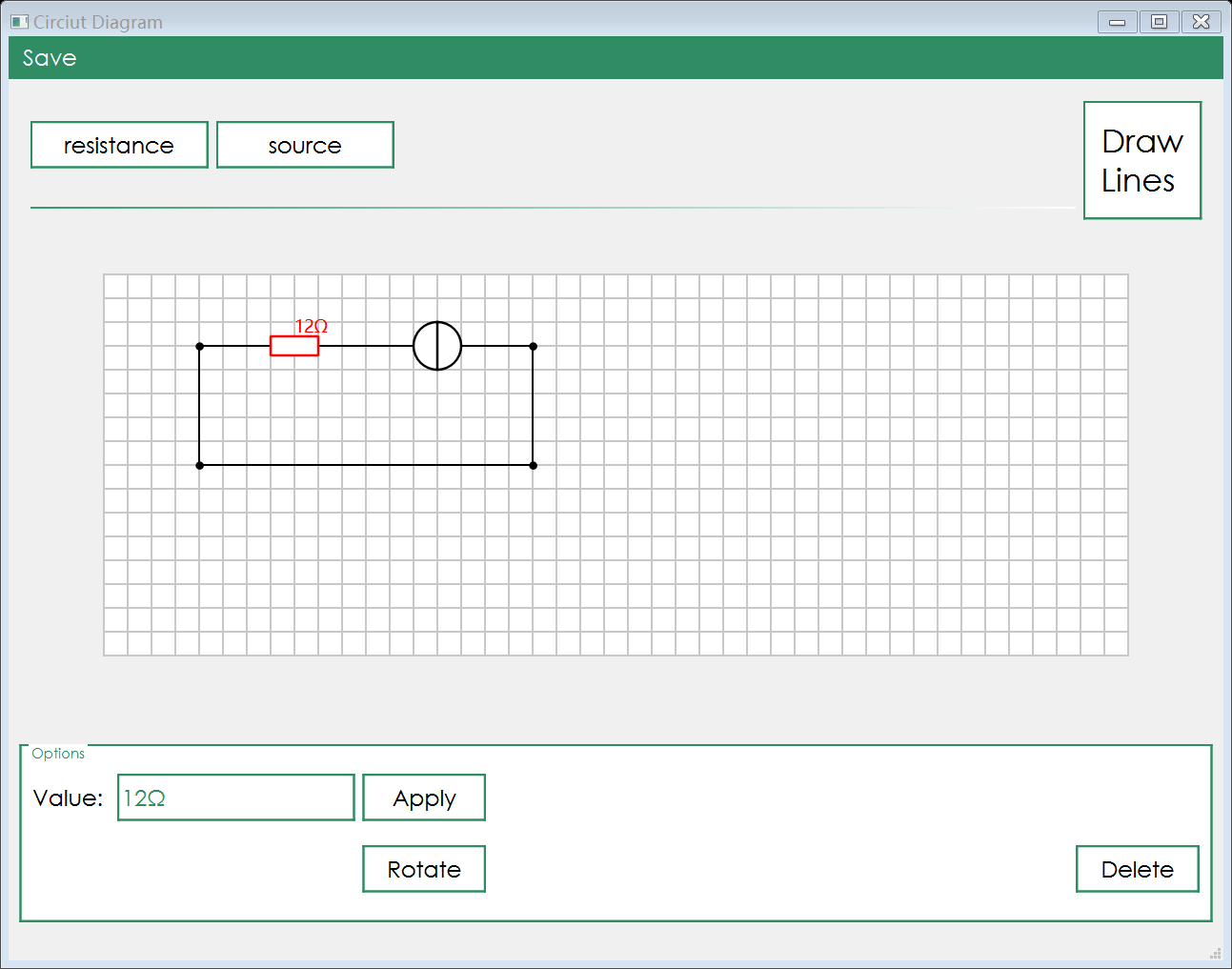
③左键单击“Apply”按钮，改变原件标记值。



左键单击



输入标记值

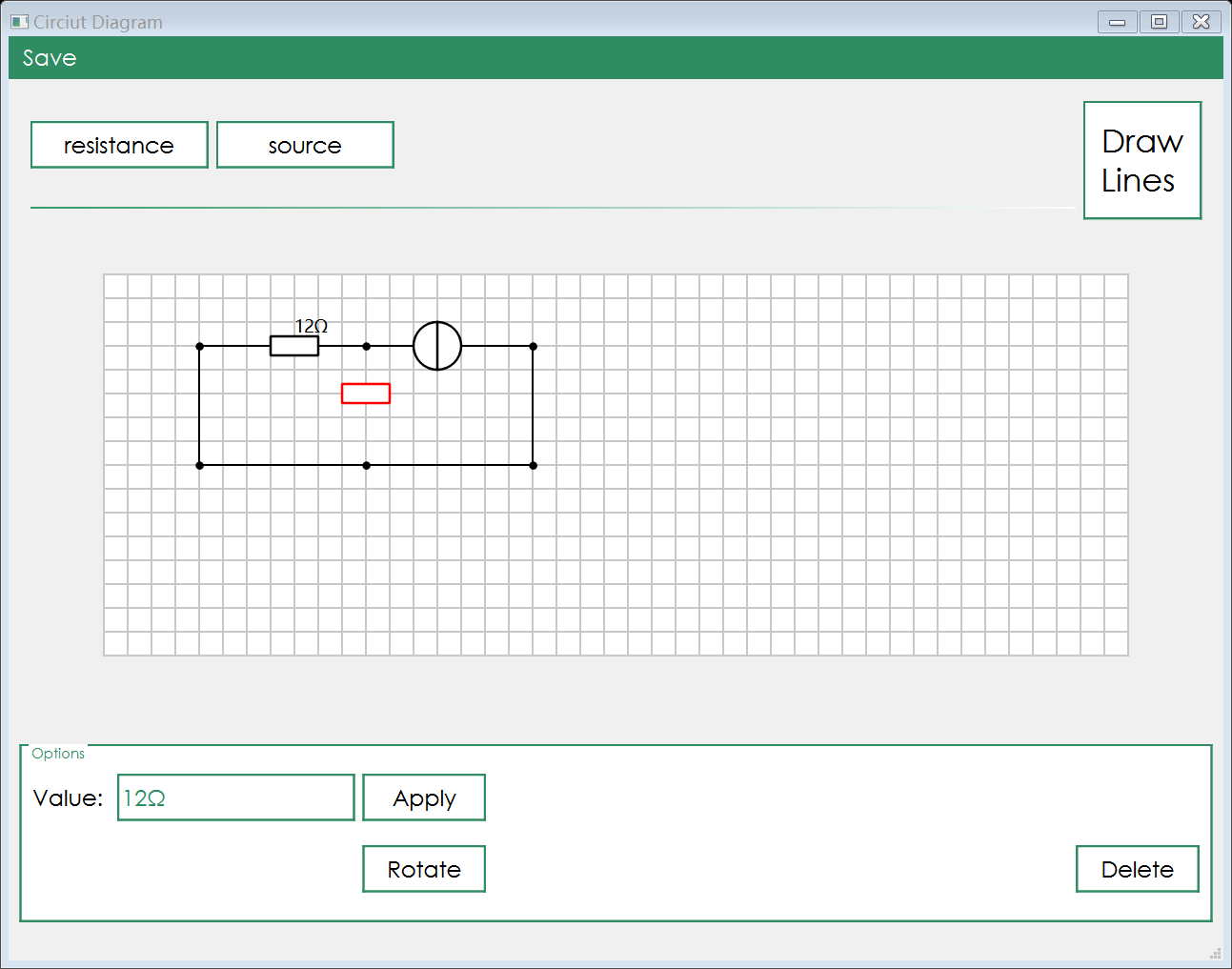


左键单击

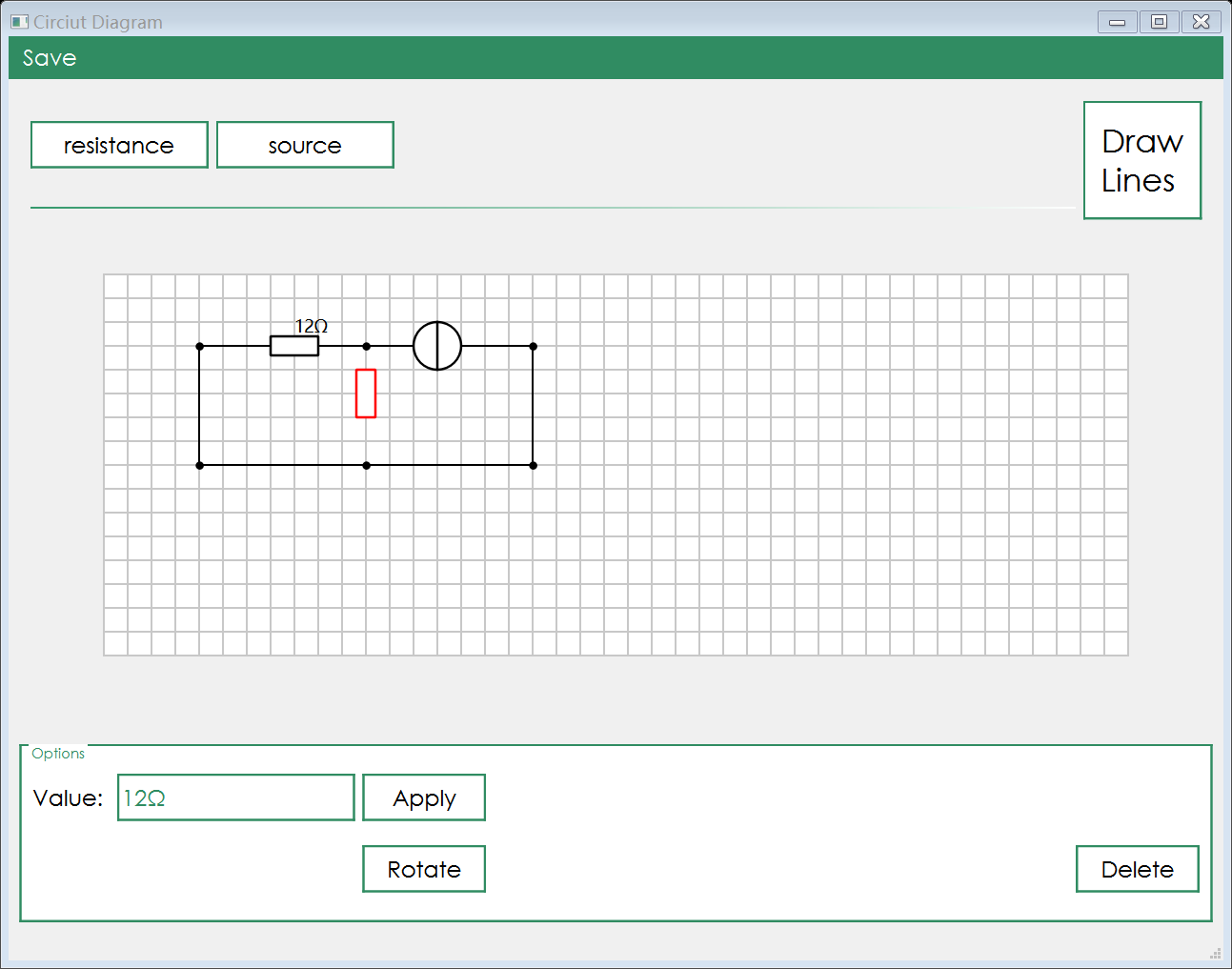
**（6）更改元件方向**

①左键单击选择元件，元件高亮显示。

②点击“Rotate”按钮，改变元件方向，连线和标记值会自动更改位置。



左键单击



左键单击

左键单击

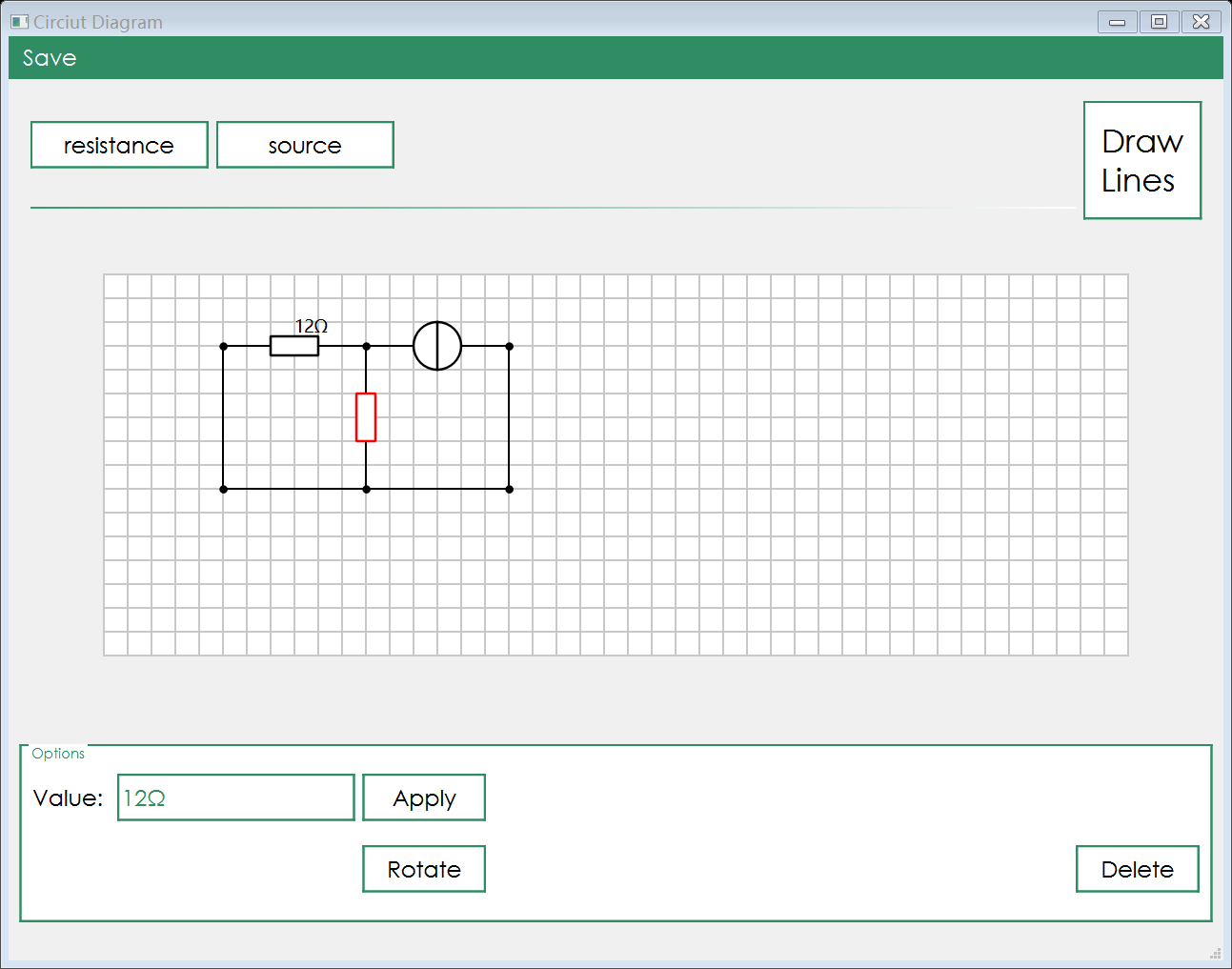
### (7)删除元件

①左键单击选择元件，元件高亮显示。

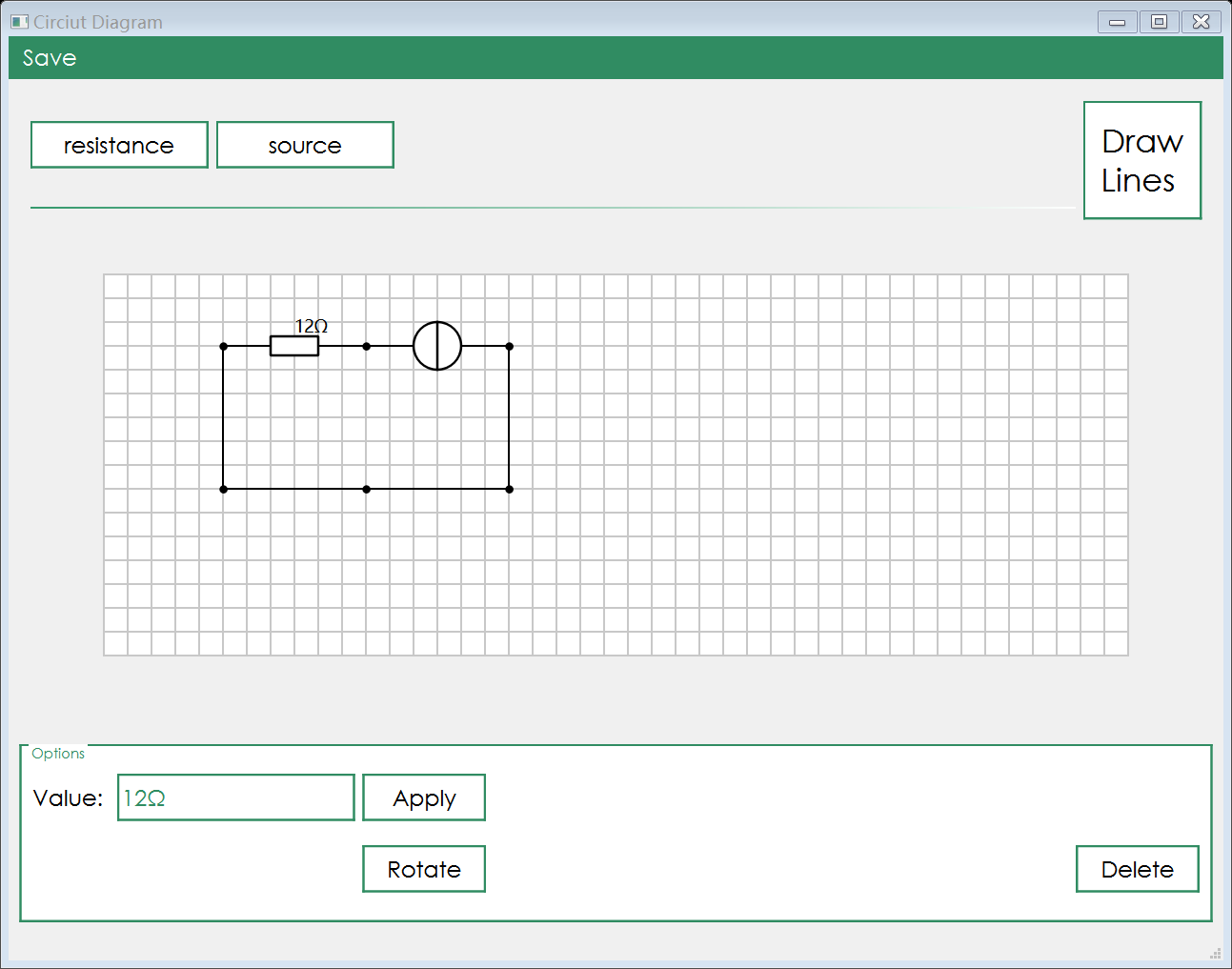
②点击“Delete”按钮，删除元件，相关的连线和标记值会自动被删除。

①左键单击选择元件，元件高亮显示。

②点击“Rotate”按钮，改变元件方向，连线和标记值会自动更改位置。



左键单击

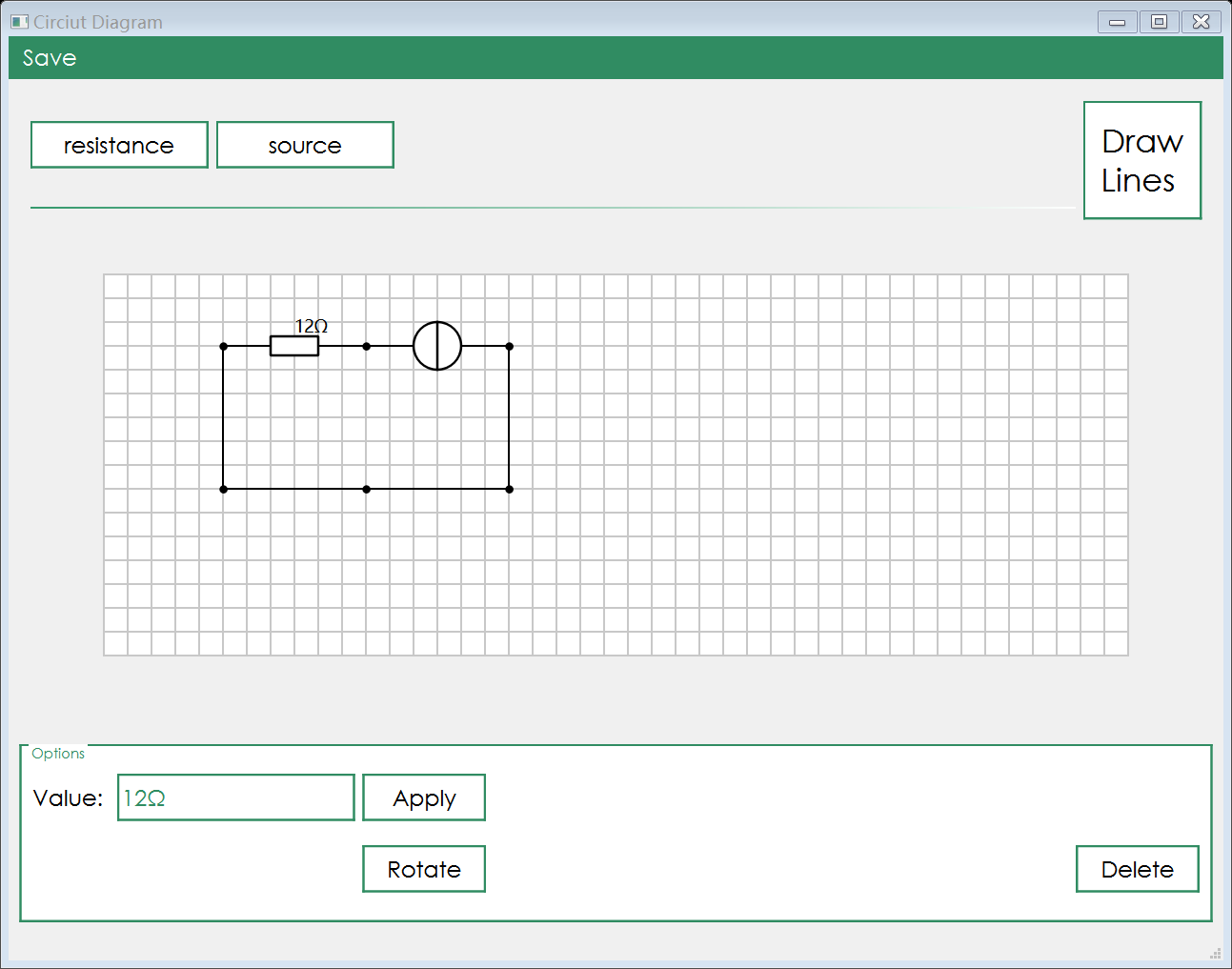


左键单击

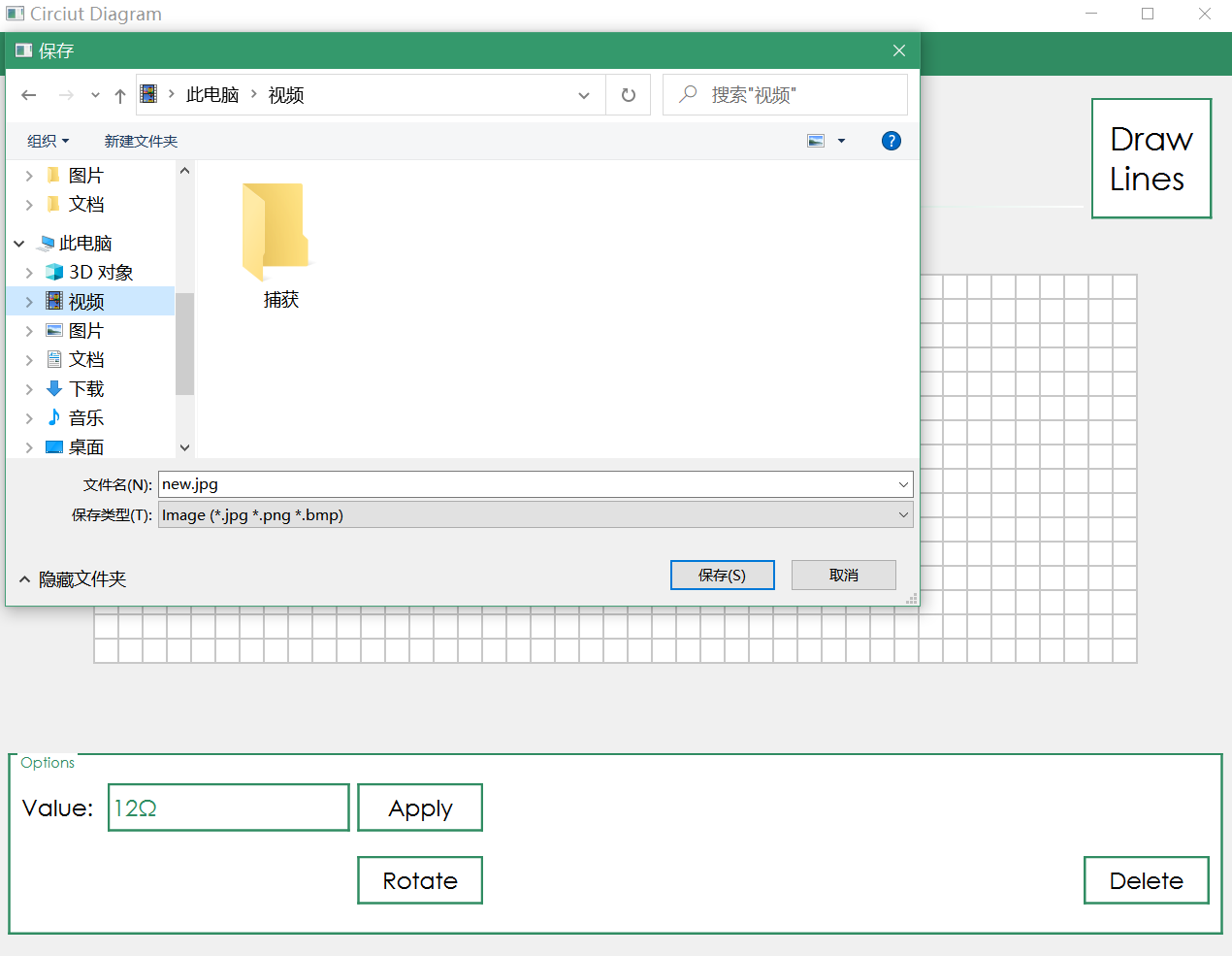
### （8）保存

①左键单击菜单栏“Save”按钮

②选择文件夹进行保存



左键单击

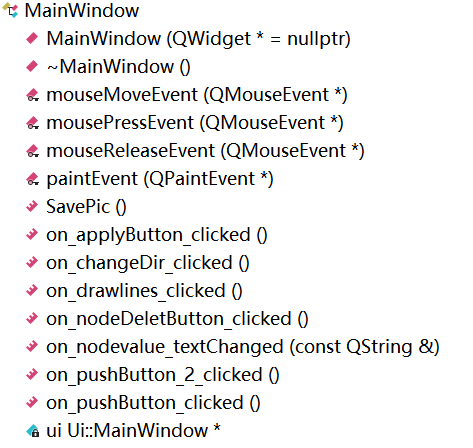


**3. 类设计**

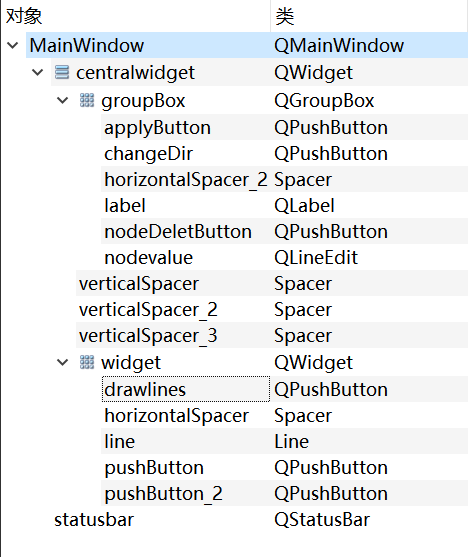
### （1）mainwindow

包括各按键点击事件的处理函数，鼠标事件的处理函数。

具体实现如下图所示：



### （2）View



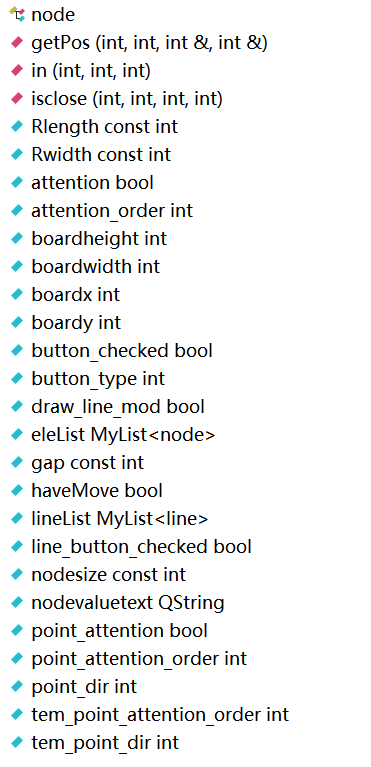
### （3）Storage

因为元件频繁的删除操作，利用链表进行存储

①元件类node：

包括元件的相关信息：坐标、大小、编号等

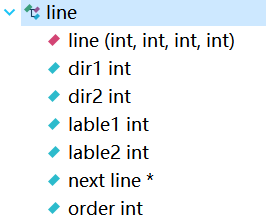
具体实现如下图所示。



②线段类line：

包括线段的相关信息：两端连接的结点编号等

具体实现如下图所示。

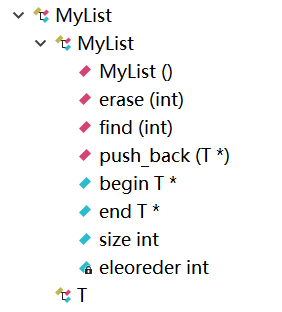


③元件及线段的存储模板类MyList：

包括链表的相关信息：首尾结点、存储个数等

函数包括：尾部插入新结点、删除结点、根据编号查找结点等。

具体实现如下图所示。



**4. 流程设计(算法及技术要点)**

### （1）新建元件

鼠标按下事件

已经有Button被按下

在list中添加结点

是

否

已经有右键被按下

否

是

### （2）元件移动

鼠标按下事件

鼠标位置与元件位置接近

将该结点设置为当前节点

是

否

已经有当前节点

获取鼠标位置

遍历List

是

鼠标移动事件

否

鼠标释放事件

已经有移动操作

获取鼠标位置

否

是

更改结点位置

### （3）元件属性更改

按钮被按下

已经有当前节点

获取更改属性

是

否

找到当前节点位置

更改当前节点属性

### （4）元件删除

### 

按钮被按下

已经有当前节点

是

否

找到当前节点位置

删除当前节点

### （5）新建线条

鼠标按下事件

计算可连线位置，分别验证

一般元件

获取鼠标位置

遍历List元件

结点

计算可连线位置

位置接近

否

是

添加连线

**五、单元测试**

## 测试案例

表 1：测试案例

|  |  |  |
| --- | --- | --- |
| **输入** | **输出** | **目的** |
| 生成元件 | 正确显示 | 元件显示在网格内 |
| 拖动元件 | 正确显示 | 元件位置正确变动，并且相关连线正确变动 |
| 生成连线 | 正确显示 | 连线到元件的一边或节点正中间 |
| 删除元件 | 正确显示 | 元件消失，相关连线消失 |
| 添加元件属性 | 正确显示 | 元件旁边出现标注 |
| 改变元件方向 | 正确显示 | 元件方向改变，相关连线的连线位置自动调整，标注值的位置自动调整 |

## 测试结果

通过测试。

**六、收获**

## 1.基础算法是函数库还是自建

在存储原件集合时，开始选用了QLIST类库，但由于并不是很熟悉，导致如对集合元素序号调整等一些操作产生误解，造成bug。在估计了深入学习类库与自己简单实现的代价后，快速用自己实现的链表替代类库。通过项目，我更加深入的体会到了类库的作用与代价。

**2. 程序设计与变更**

对于不同元件，因为实现方式类似，数目不多，所以开始未进行类继承的框架设计，程序可以运行，用课上所学的“基类-继承”等知识来设计反而觉得很麻烦。然而，当需要对不同部件添加不同属性时，在由此方式实现的程序，修改变得相当困难。