**Java图形用户界面分析报告**

指导老师：陈俊侠

院 系：计算机学院

专 业：计算机

班 级：2015级计算机2班

学 号：20152100027

姓 名：詹萍

**一、程序框架及功能**

1. **程序框架**

主窗口

画板

菜单

Edit

Add Circle

Add Triangle

Add Rectangle

Add Pentagon

Add Square

Add Text

Add Line

File

Save Image

Load Image

Save txt

Load txt

Load Object

Save Object

工具栏

Add

Circle

Rectangle

Triangle

Square

Pentagon

Line

Text

No

Delete

删除组合框选中的图形

Edit

修改组合框选中的图形

的数据

Save

Save Image

Save txt

Save Object

Load

Load Image

Load txt

Load Object

**2.功能描述**

这是一个图形程序项目。主窗口由三部分组成：①菜单、②工具栏、③画板。

**菜单：**菜单分为两个功能模块。

1. File：这个模块主要是用于保存图形和载入图形。各分为三种方式，图片的存取、部分数据的存取、整个对象的存取。
2. Edit：这个模块主要用于添加各种图形。

**工具栏：**主要分为五个模块。

1. Add：通过按钮选择图形种类来添加图像。
2. Delete：通过组合框来选择要删除的图形。
3. Edit：通过组合框来选择要修改数据的图像。
4. Save：保存图形到文件中，分为三种保存方式，图片形式、部分数据、整个对象的保存。
5. Load：从文件中加载图形，分为三种加载方式，图片形式、部分数据、整个对象的加载。

**画板：**添加或者修改后的图形会显示在画板上，删除的图形会从画板上消失。加载的图形、图片也会显示在画板上。

**二、程序结构的分析**

**1．程序包**

整个程序主要分为三个程序包：

1. **main：**这个程序包放着主窗口GraphApp，是程序运行的入口。
2. **Graph：**这个程序包主要包含各个图形类:Circle、Rectangle、Triangle、Square、Pentagon、Line、Text、以及他们的父接口MyShape、图形保存向量类MyGraph、文件对话框FileIO类。
3. **UI：**这个程序包放着各个图形的对话框：CircleInputDialog、RectangleInputDialog、TriangleInputDialog、SquareInputDialog、PentagonInputDialog、LineInputDialog、TextInputDialog、以及画板DrawPanel和工具栏Toolbar。
4. **各个类、接口之间的相互调用关系**
5. **图形类：**

类Circle、Rectangle、Triangle、Square、Pentagon、Line和Text均继承了接口MyShape，MyShape定义了各个图形类共有的方法paint方法和toShortString方法。

类MyGraph主要定义了保存各个图形对象的父接口MyShape类型的向量，其次是对这些图形的一些操作。有以下各个方法：

displayAll()画出所有图形、

add()添加新图形、

remove ()删除选中图形、

get()获得指定图形、

size()获得已有图形的数目、

Savetxt()将图形数据保存至文件中、

Readtxt()从文件中读出图形数据、

SaveObject()将整个图形对象保存至文件中、

ReadObject()从文件中读出整个图形对象。

1. **图形对话框类：**

类CircleInputDialog、RectangleInputDialog、TriangleInputDialog、SquareInputDialog、PentagonInputDialog、LineInputDialog和TextInputDialog。各个图形对话框类通过调用图形类的构造方法，从对话框上获取数据来新建图形。

1. **画板类：**

主要是构造方法、setImage()和paintComponent()方法。构造方法中有可以初始化画板类中图形graph成员的方法。setImage()可以用外部传入的图片来初始化画板类中的image数据成员。paintComponent()方法调用图形类MyGraph的方法displayAll()在画板上画出所有图形。

1. **工具栏类：**
   1. **添加图形的按钮：**

通过图形选择的按钮组来选择要添加的图形，点击Add按钮则添加图形，相应的动作事件会选择好图形，通过调用相应的图形对话框类的get图形方法，根据对话框的数据来添加图形。

* 1. **删除图形的按钮：**

通过复选框选定图形，点击Delete按钮，调用图形类MyGraph的方法remove ()删除选中的图形。

* 1. **编辑图形的按钮：**

通过复选框选定图形，点击Edit按钮，调用指定图形类的get方法和set方法，通过对话框来修改指定图形的数据。

* 1. **保存图形的按钮：**

三种保存方法，分别调用类MyGraph的保存图形的方法以及文件对话框类的方法来保存图形数据。

* 1. **加载图形的按钮：**

三种加载方法，分别调用类MyGraph的加载图形的方法以及文件对话框类的方法来载入图形或者图片。

1. **文件对话框类：**

文件对话框主要用来获取保存或者加载文件的对话框。在文件对话框类中包含有以下方法：getFileExt()、getImageFile()、getFile()。

getFileExt()方法是用于获取文件的后缀名。getImageFile()方法用于获得一个图片文件，在此方法中调用getFileExt()方法来获得文件名的后缀，来判断获得的文件是否为图片文件。getFile()方法则可以获得多种类型的文件（不局限于图片文件）。

1. **部分功能的具体分析**
2. **添加图形功能Add：**

点击某个图形的按钮时，其动作事件设置变量select的值，不同的值代表不同的图形，选择要添加的图形的种类，再点击Add按钮，通过不同的select的值来选择要添加的图形，调用不同的Creat方法，分别调用不同的图形对话框的get图形的方法，由对话框上的数据来生成相应的图形。

1. **编辑图形功能Edit：**

先通过组合框选定要修改数据的图形，当选中的图形不为空时，可以修改图形的数据。

先通过instanceof来判断图形的种类，然后通过不同图形get图形方法，打开对应的不同对话框，用目前获得的图形的数据初始化对话框上的文本框的数据值，在对话框上修改图形的数据，然后在对话框的Enter按钮对应的动作事件方法内部用文本框内修改完的数据值去初始化图形的值，并将其返回给调用者。

此时返回的修改完数据的图形用于建立一个相同类型的新的图形的初始化。再用这批新的数据调用图形类的get和set方法去修改要修改数据的图形的相应数据。

1. **保存/载入图片功能SaveImage、LoadImage：**

保存和加载图片通过文件对话框类来获得一个图片的文件。

在保存SaveImage方法中，通过getSize()方法来获得画板的长和宽，建立一个BufferedImage对象来保存画板上的图片信息，并将这些信息用于建立一个Graphics2D图片，再将其paint在获得的图片文件上。

在加载LoadImage方法中则只需创建一个Image对象来保存获得的图片文件。

1. **保存/载入图形数据的功能Savetxt、Readtxt：**

在Savetxt方法中创建一个使用默认大小输出缓冲区的缓冲字符输出流，调用图形类对应的toShortString()方法获取图形的信息并以字符的形式写入文件中。同时使用newLine()方法写入行分隔符。再刷新该流的缓冲。直至写入全部图形的信息。

在Readtxt方法中同样创建一个使用默认大小输出缓冲区的缓冲字符输出流，用readLine()方法逐行读入图形信息，每读入一个图形信息，判断其类型，调用与toShortString()相对应的图形方法，分割图形数据，建立图形。直到读入为空则结束。

**三、项目测试**

1. **项目的优点与不足**
2. **项目的优点**

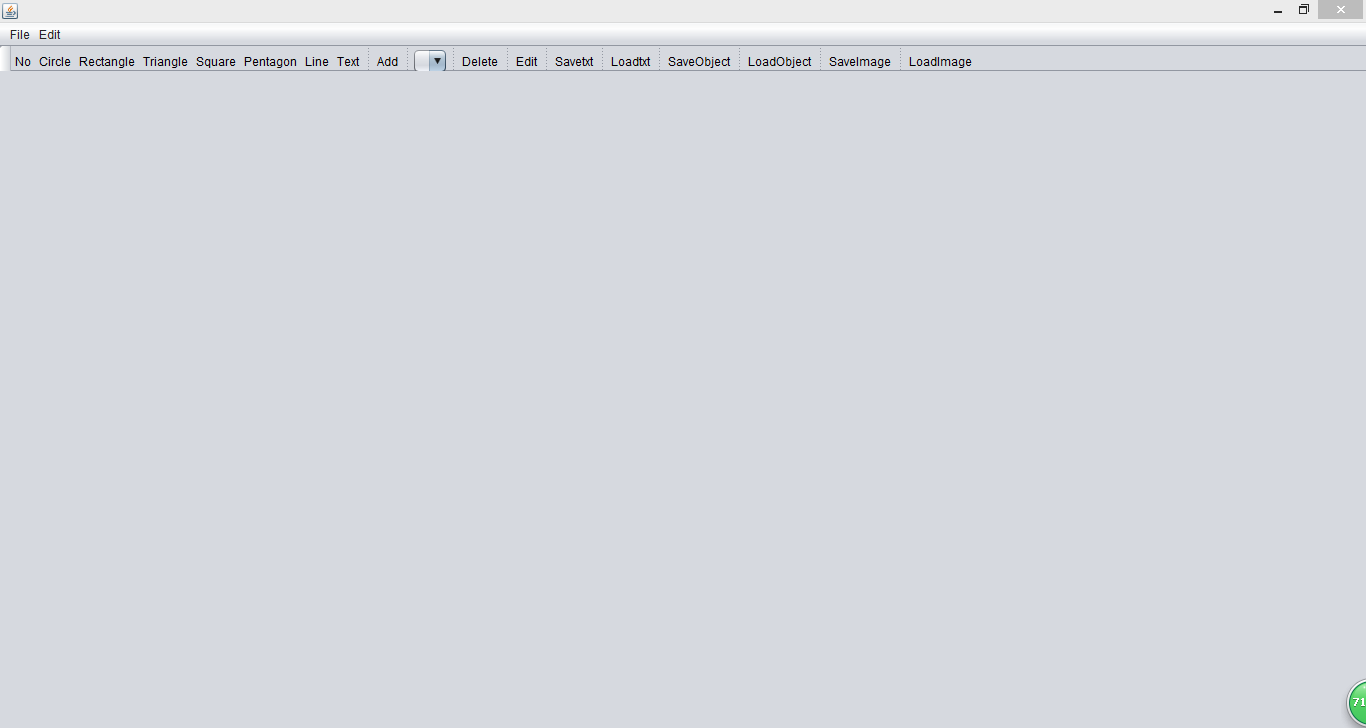
这是一个简易图形程序，操作简便、可供用户添加、删除和修改多种图形的数据、组合框的使用，使得删除和修改的图形数据变得直观、不同的保存和加载的方式使得图形程序多样化。

1. **项目的不足之处**

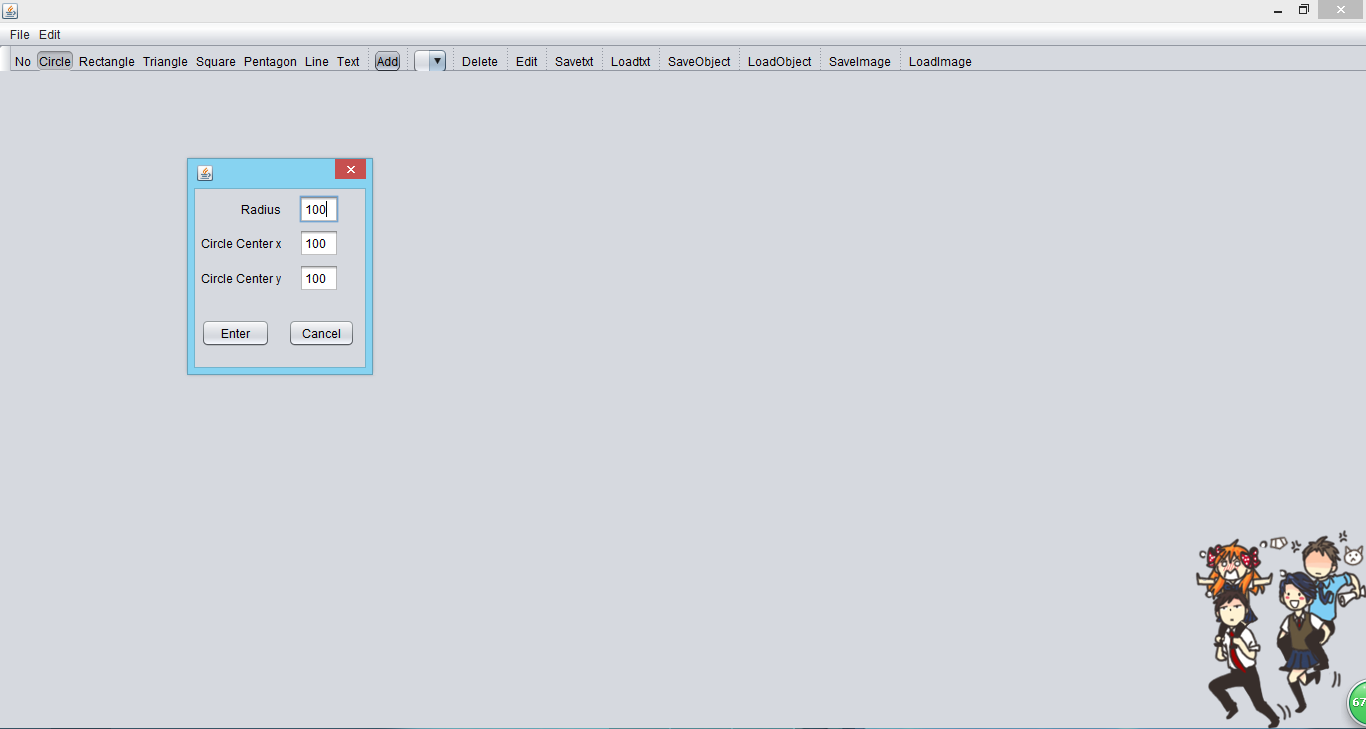
目前此项目尚且不能实现用户自由绘画的功能、无法指定图形的颜色、图形之间显示会出现相互覆盖的现象，不能调整图形的显示顺序、功能较少且比较单一。

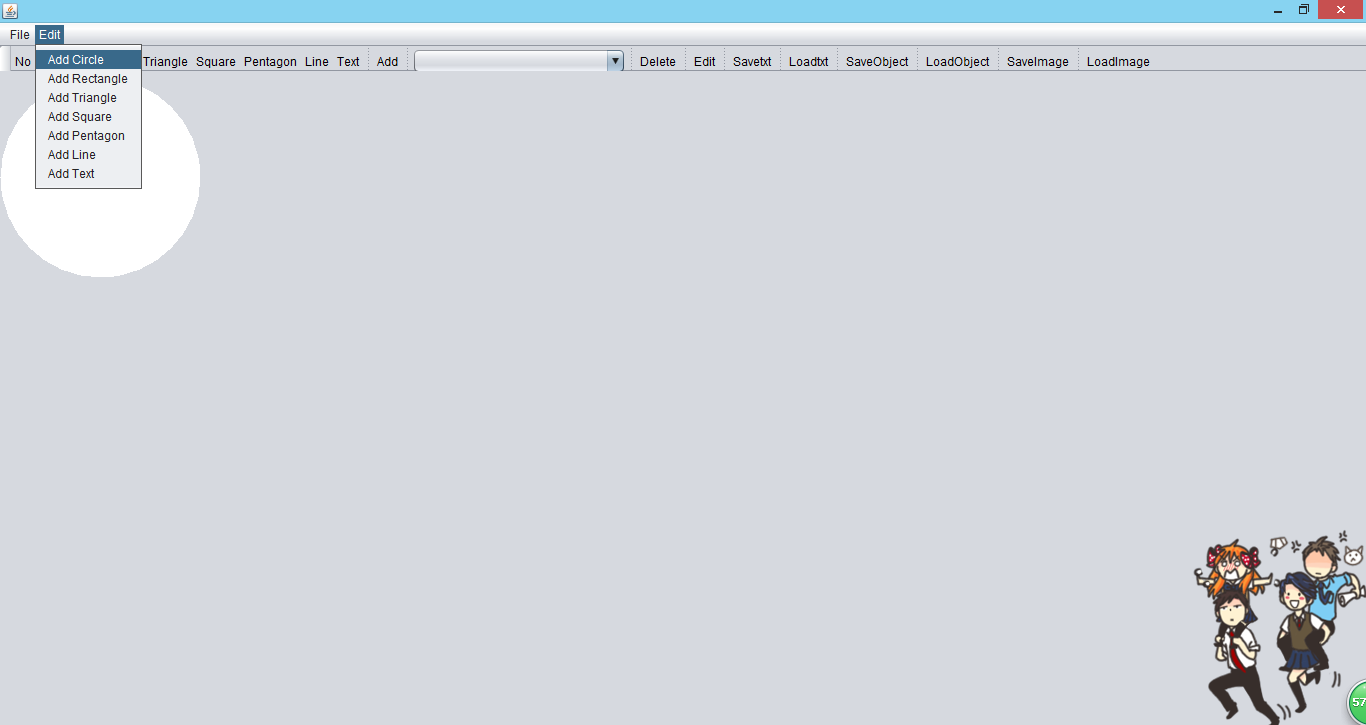
1. **测试结果**

界面：

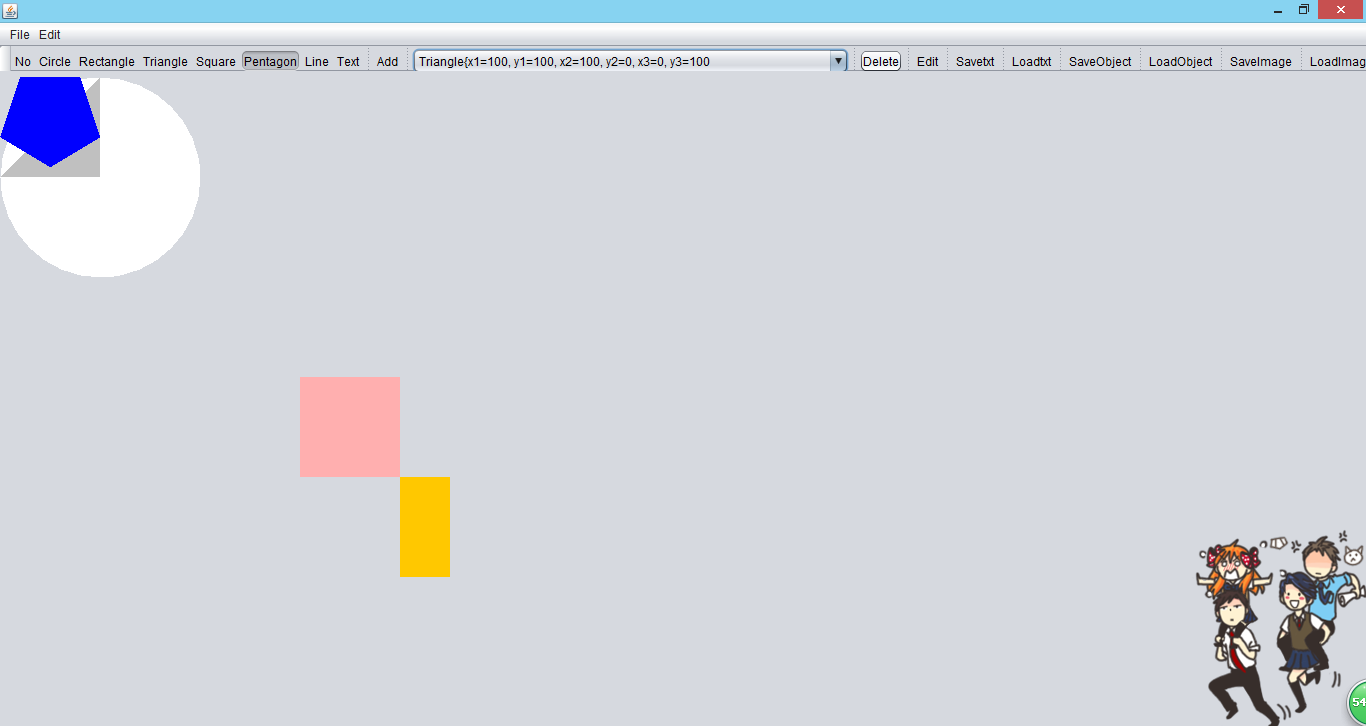


添加图形：（两种方式）



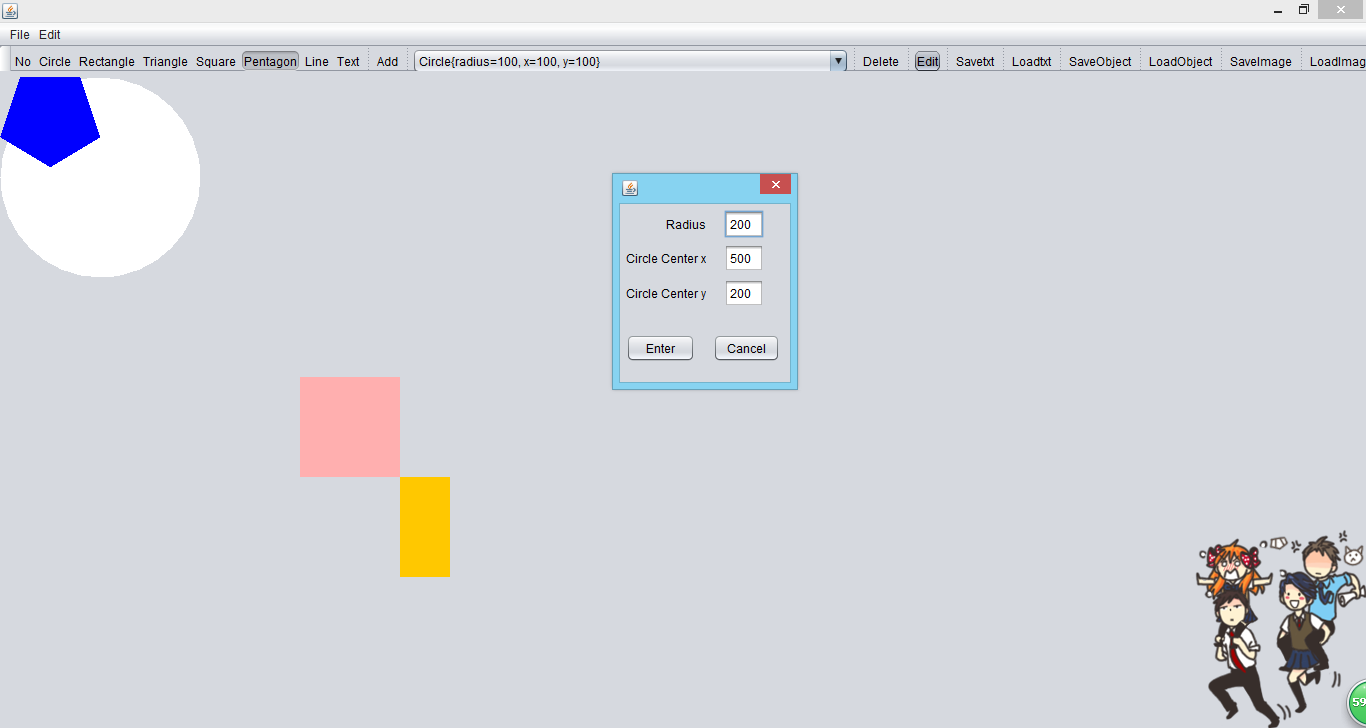


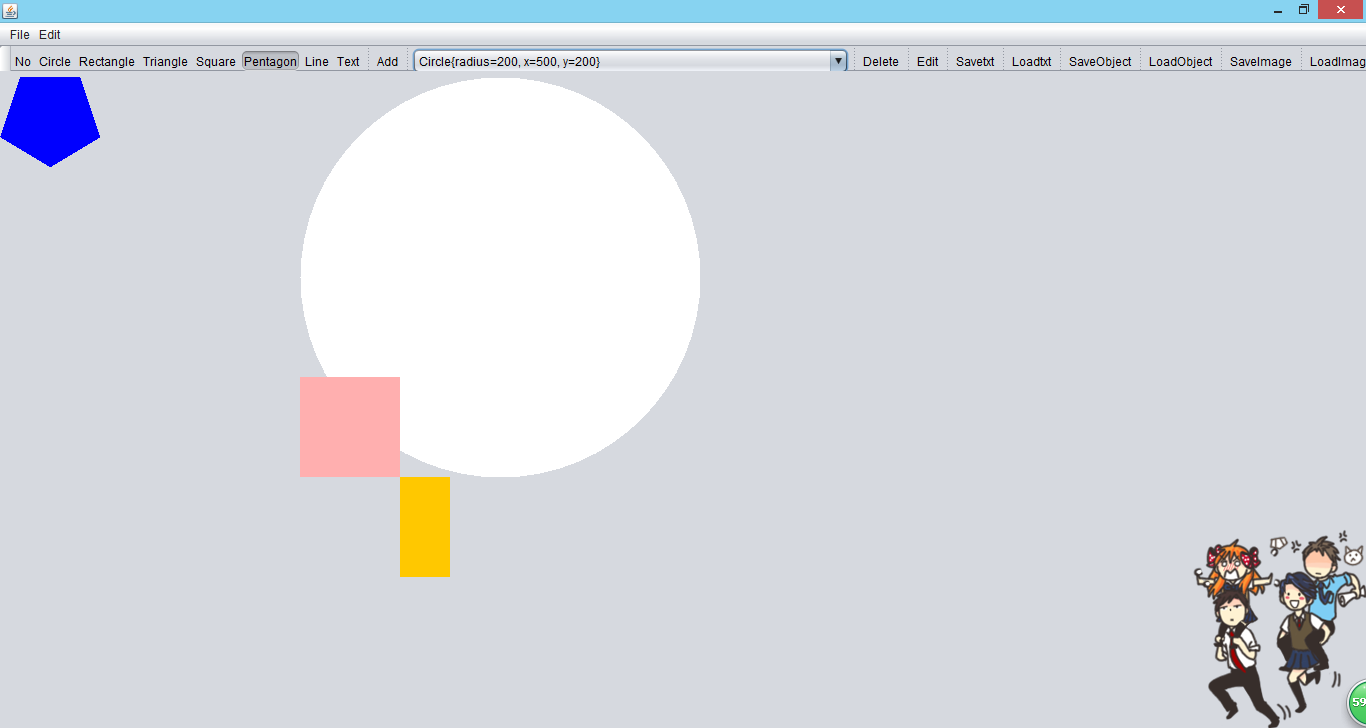
删除图形：





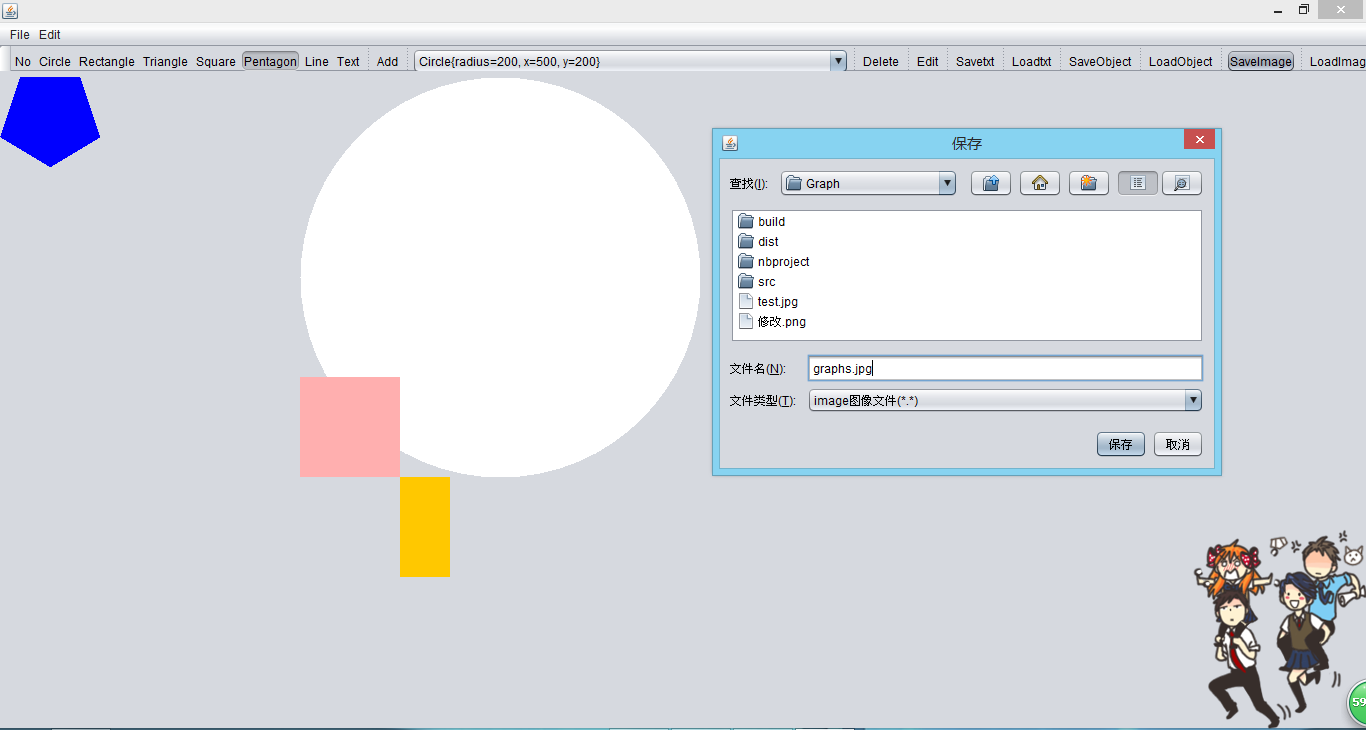
编辑图形：



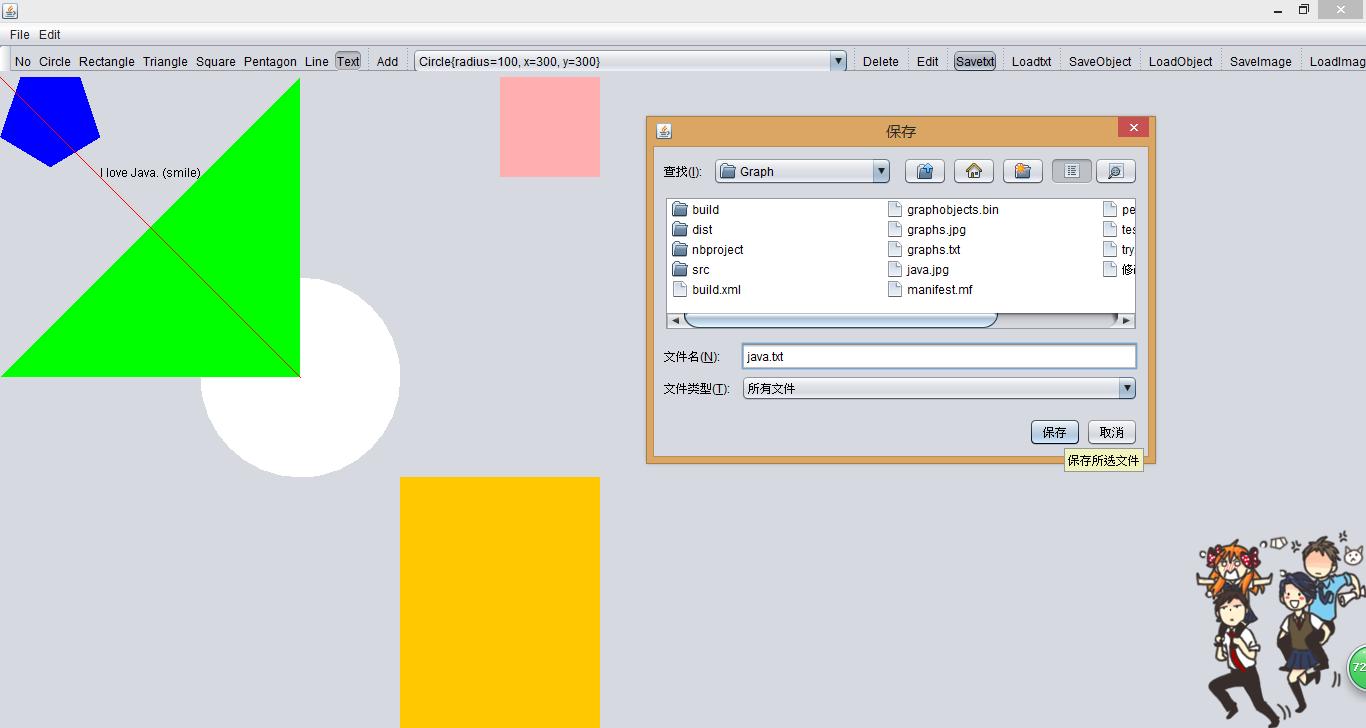


保存图形/图片：

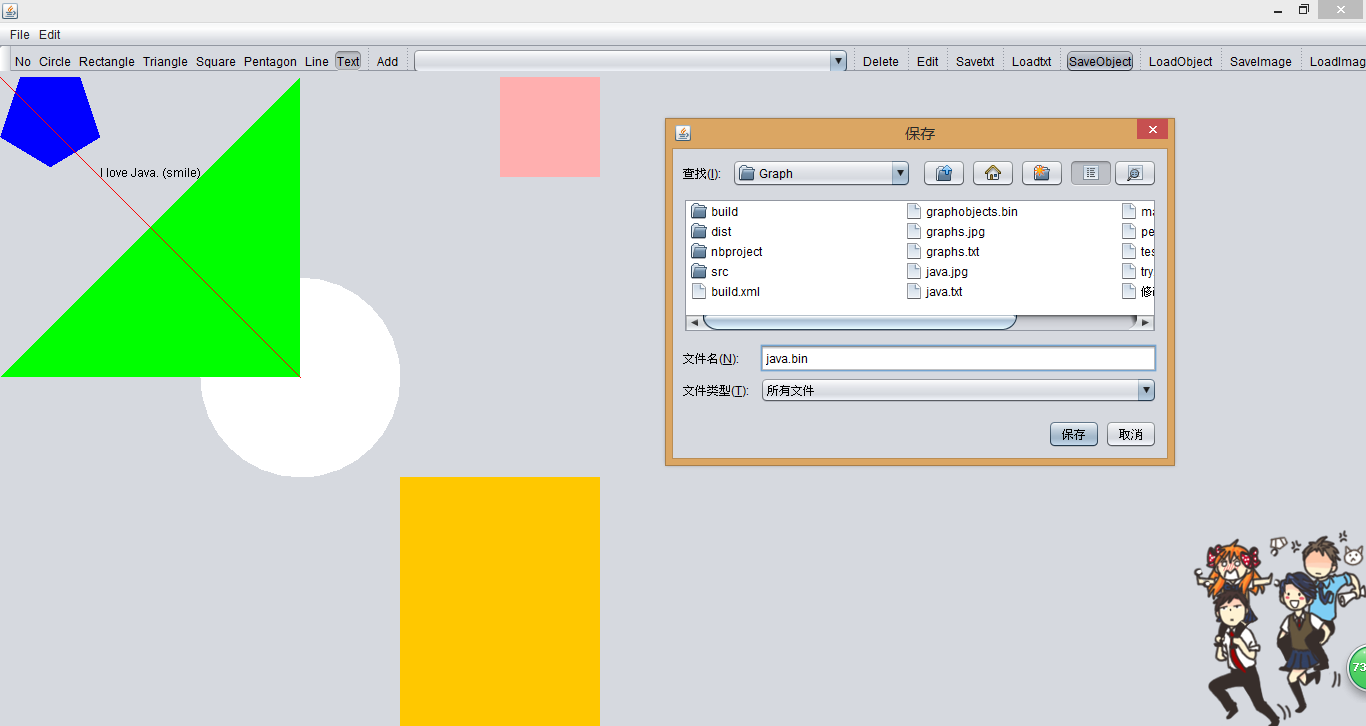
* 1. 图片形式：



保存txt文件：

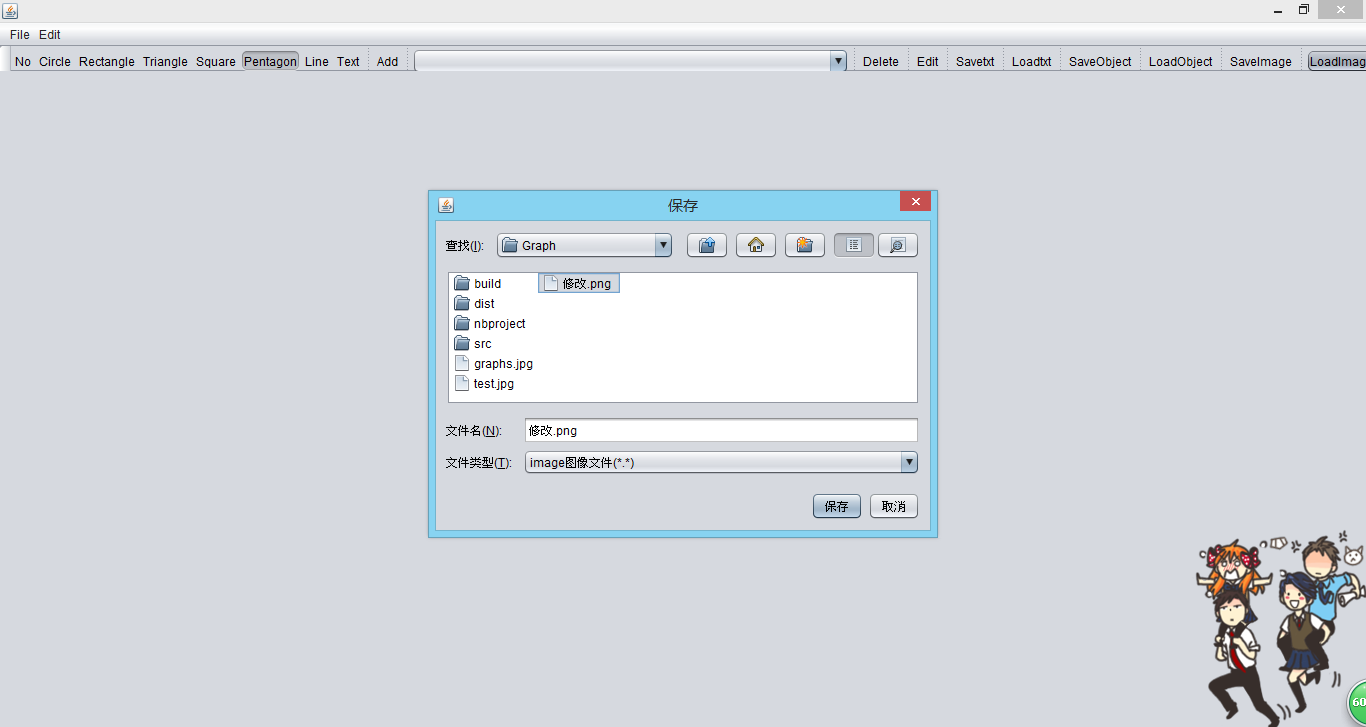


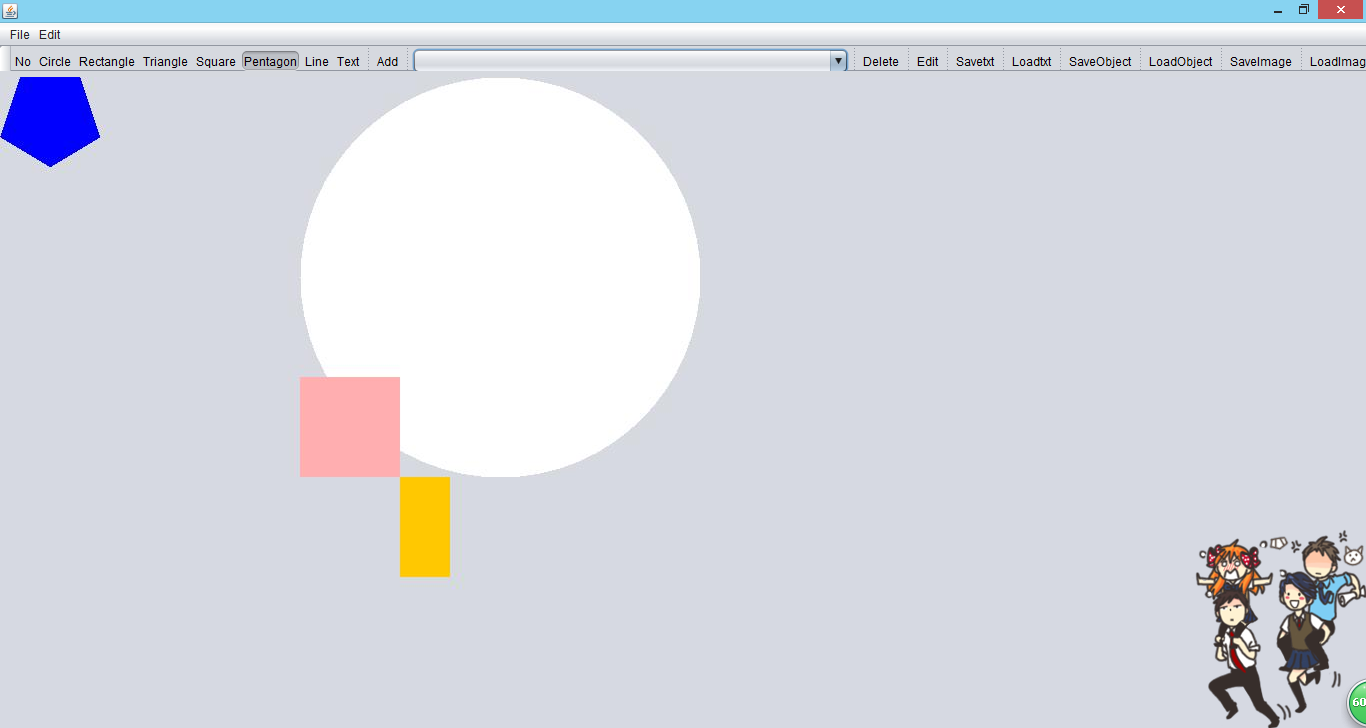
保存bin（整个对象）文件：



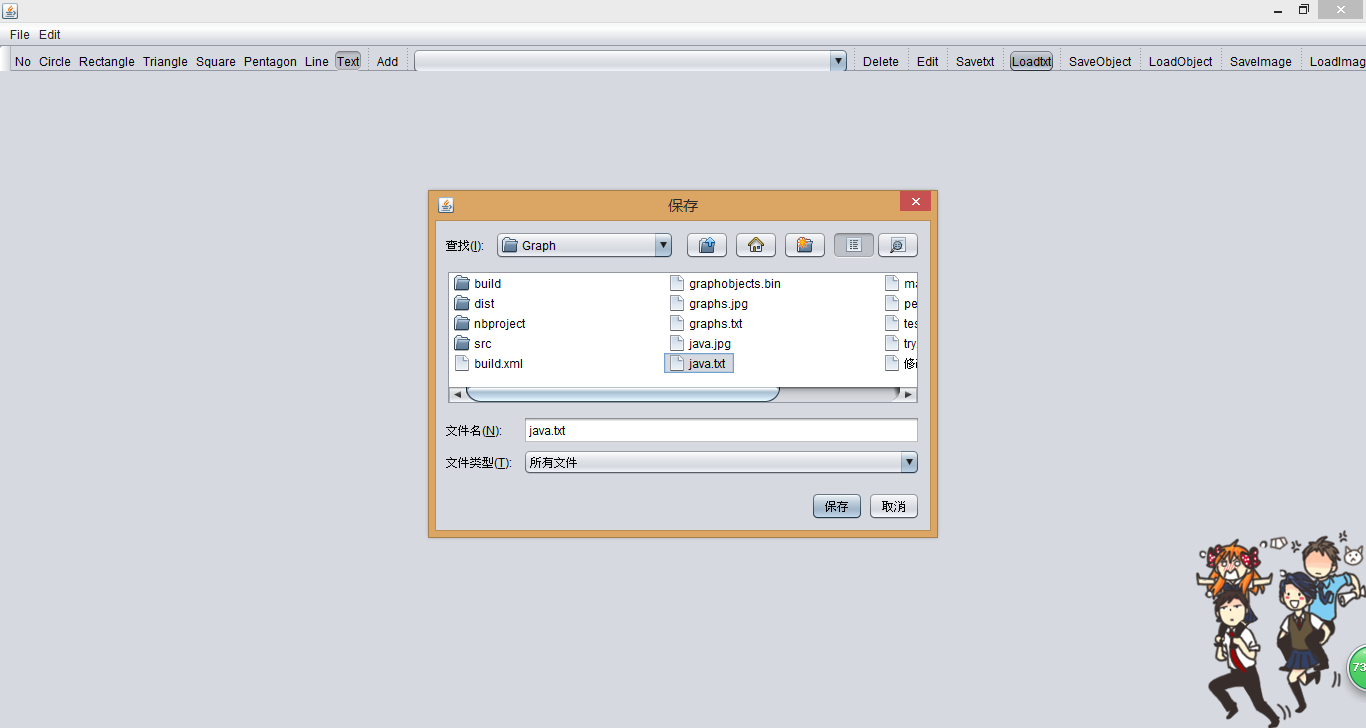
加载图形/图片：

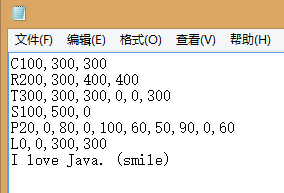
* 1. 图片形式：

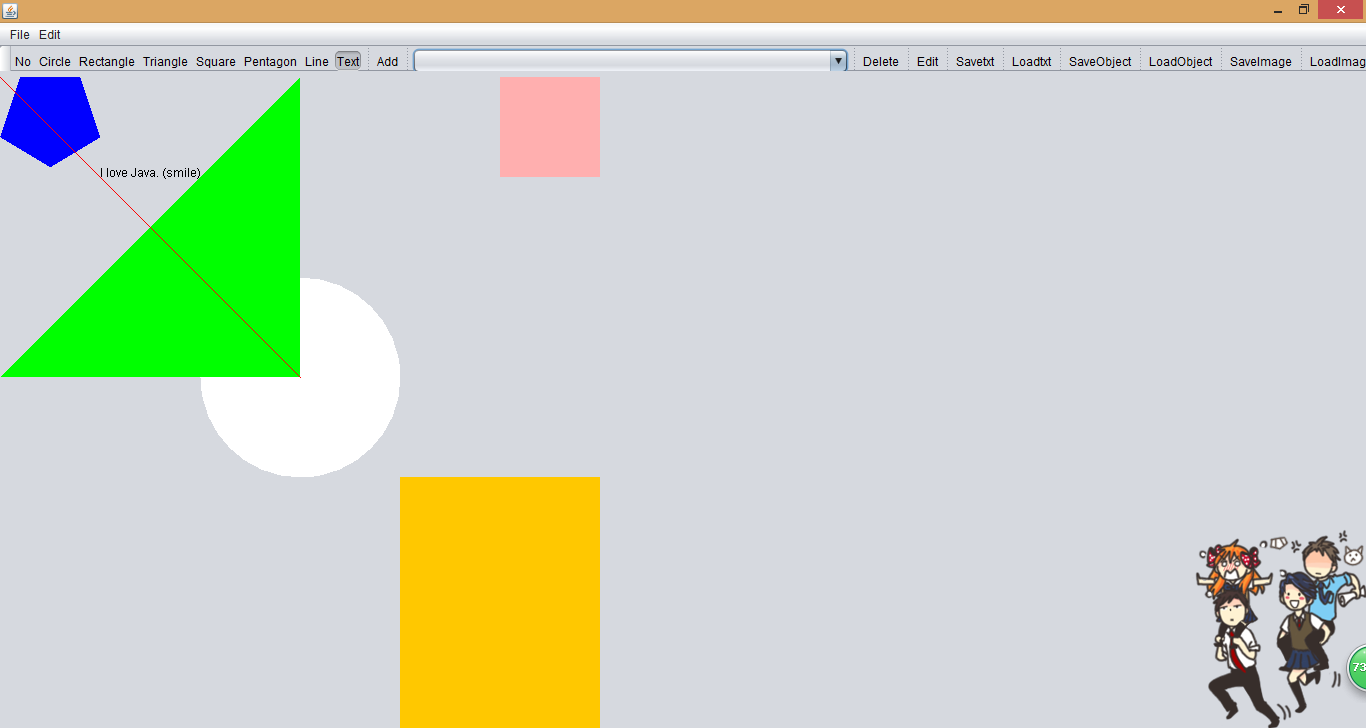




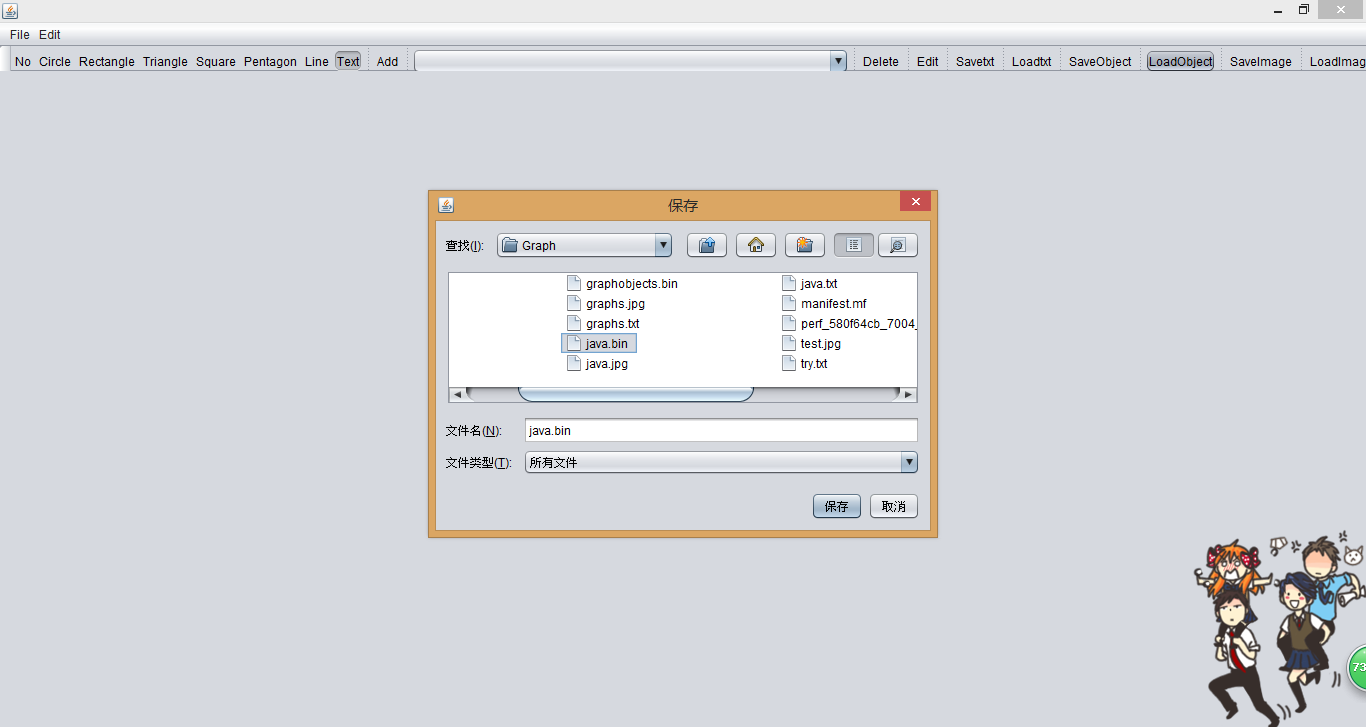
加载txt文件：

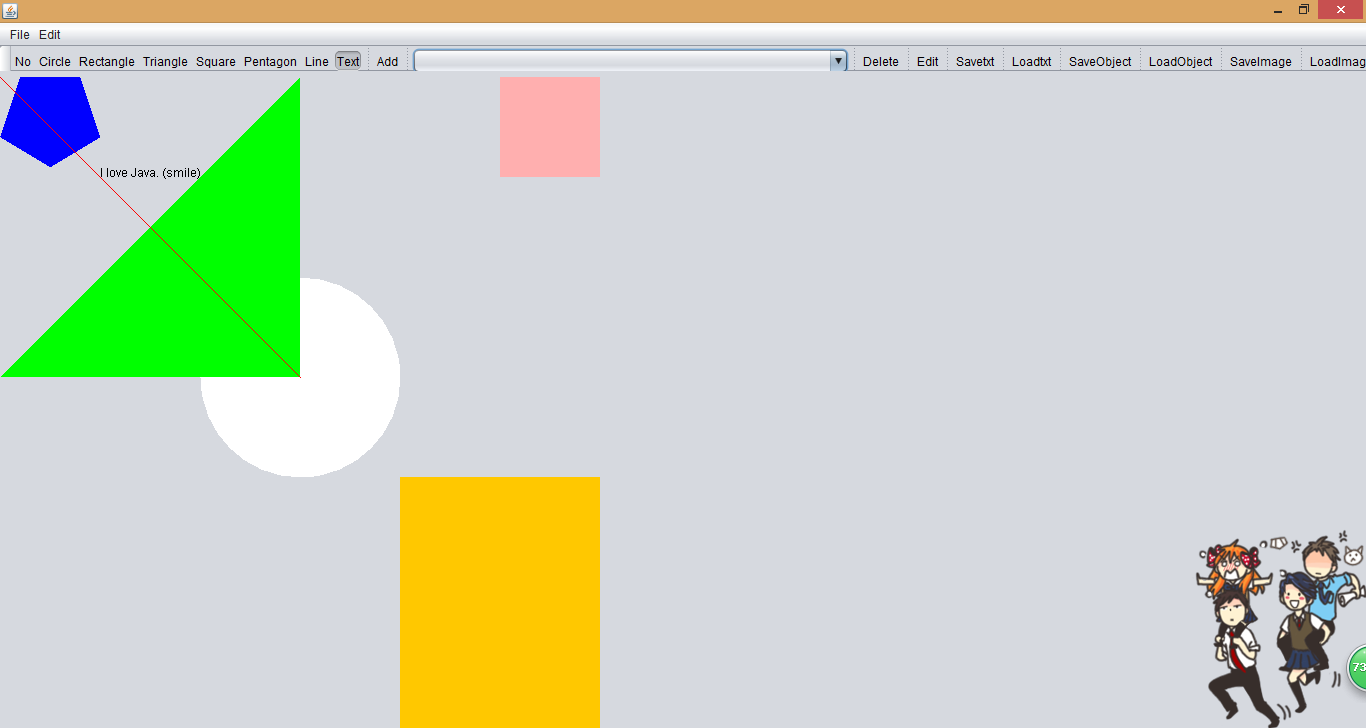


文本文件中的图形信息。



加载bin（整个对象）文件：





**四、回顾与总结**

**1.回顾：**

这是一个综合性实验。

一开始只是简单地从传递参数来建立图形，计算相应的面积和体积。然后逐步扩充其他内容。

图形种类增加。从简单的圆形，扩充了矩形，三角形，五边形，直线，甚至是文字（字符串）。

不同的构造方法可以满足不同输入的要求，当获得一串字符串，也可以通过字符串的分割来建立图形。但是需要指定输入格式。

通过不同的存储结构来存储图形。用向量来存储图形可以不必事先定下图形的数目。

加入对话框。通过不同层次的对话框来实现简单的菜单功能，从对话框上通过指定输入格式来获得图形的信息，并增加了删除和显示等功能。

增加了画板，可以把图形画在画板上。引入图形的绘画。

增加了对话框输入图形信息。引入图形用户界面，建立不同的图形可以通过不同的图形对应的对话框类来获取图形的数据。引入了事件与监听器的概念，不同按钮有对应的不同的事件，在方法体内加入具体操作可以实现按钮相应的功能。

增加了工具栏，将增加图形，删除图形，编辑图形的功能用工具栏按钮的形式实现，增加了组合框，可以用于选定要进行操作的相应图形。

增加了多种保存和加载图形信息的功能，以字符形式保存、保存整个图形对象以及保存为图片形式，其均有相对应的读入加载的方法。

增加菜单，将菜单，工具栏和画板整合到同一个窗口上，实现几个模块的衔接。

**2.总结：**

本次课程的学习我仍有许多问题没有弄懂，遇到问题也不能很快解决，仍然不具备独立完善项目的能力。但是我也对用java进行开发有了一定的理解，这次课程的学习帮我建立了很好的整局概念，可以说使我受益匪浅，这对我日后的学习也会有很大的帮助。

**参考文献**

**雍俊海.Java程序设计教程第三版.清华大学出版社.2013**