

# 专题三：图形化编程

## 实验报告

报告小组：47 | 小组成员：郭书廷(3170104871),叶帆(3170102410) | 指导老师：张引，田沈磊，熊海辉

项目：CAD, RescueTheBall

### Project1-CAD

#### 问题描述

制作一个简易 CAD(PPT 中有展示动图)

#### 程序使用说明

操作方法：

- 1.选择颜色模式，显示 **LINE** 时表示调整边框线条颜色，在此模式下绘制矩形和椭圆为空心图形；显示 **FILL** 时表示调整填充颜色，在此模式下绘制矩形和椭圆为实心图形。
- 2.选择工具，拖动绘制或点击选择修改对象，选择空心图形和直线时，需点击边框 0.1 英寸范围内方可选中；选择实心图形时点击图形内部即可选中。
- 3.在 **LINE** 模式下输入文字会在选定的初始位置左对齐显示，在 **FILL** 模式下，输入的文字会以设定的初始位置为中心居中显示。
- 4.在绘制矩形和椭圆时按下 **Shift** 可以绘制正方形和正圆，在绘制直线时按下 **Shift** 会将直线角度对齐到 45 度，在修改一个图形大小时按住 **Shift**，拖动上下左右四个定位点可以按比例缩放（四个顶点不行）。
- 5.对于矩形、椭圆、直线、文字的清除可以用选择工具选中后按 **delete**，对于刷子工具，可以新建一个实心矩形，然后拖动来代替橡皮。
- 6.除刷子外，选中后可以重新设置颜色。
- 7.有四个按键分别是字体/画笔大小+5,-5,+1,-1

#### 程序设计

首先，对于 UI 的设计，我们希望用尽量少的按键，尽量方便地实现更多的功能，所以就有了上述黑体的功能。这些功能很多都是参照 **windows** 自带的画图工具，包括预置的 20 中颜色也来源于它。

1. 使用结构体 **Shape** 来存储图形，用一个 **type** 变量记录类型，使用左上角和右下角的坐标来记录每一个图形的位置，用两个三维向量分别记录边框颜色和填充颜色。
2. 使用链表来记录图形的重叠顺序，每次选中后的图形被移动到链表尾部，随后重新按链表顺序绘制即可将选中的图形移至顶层。
3. 重建坐标系，以方便绘制 UI（我们还使用了 **PS** 来预先设计界面）。
4. 各种画图模式的切换，使用两个变量 **Temp\_Mode** 和 **Mode**，分别表示松开鼠标按键后应该应用的模式和当前的模式。比如在一个按钮上按下鼠标左键时，设置 **Temp\_Mode** 为 **PAINT\_RECT**(画矩形)，设置 **Mode** 为 **NONE**(因为在按钮上点下去后移动鼠标的过程不需要绘制任何内容)，然后再松开鼠标后将 **Mode** 设置为 **Temp\_Mode**(也就是 **PAINT\_RECT**)。这个方法在画图时也十分好用，比如，在 **PAINT\_RECT** 模式下，按下左键，设置 **Temp\_Mode** 为 **PAINT\_RECT**,设置 **Mode** 为 **PAINT\_RECT\_ACTIVE**，随后在鼠标移动事件的处理时，发现模式为“活动的矩形模式”，绘制矩形。松开鼠标左键后 **Mode** 回复 **PAINT\_RECT**，再次移动鼠标，模式是“矩形模式”而非“活动的矩形模式”，所以不会继续绘制矩形。（详见代码注释）

Project2-RescueTheBall	
问题描述	点击开始后，会有一个小球随鼠标移动，球不能触碰墙壁，界面中有若干机关和障碍，需要移动鼠标躲避障碍、触发机关、吃金币，到达终点。 <b>(压缩包内有展示视频，release 文件夹中的 test32.exe 可以从任意位置开始，用来试玩)</b>
算法分析与 代码设计	<p>游戏机关介绍及玩法详见 ppt。</p> <p>代码技巧：</p> <ol style="list-style-type: none"><li>1.设计 4 种结构体，平移直线，旋转直线，矩形，圆弧。</li><li>2.每个图形原件有各自的速度（线速度/角速度），每个单位时间程序调用每个原件的 Move 函数，这些移动原件就可以自己动起来了。</li><li>3.判断圆和线段有没有相交，使用向量计算比解析式计算方便的多，方法如下，设圆心为 A，线段的端点为 B，C，用点积的方法来判断 <math>\angle ABC</math> 和 <math>\angle ACB</math> 有没有钝角，如果有钝角，则直接判断圆心和端点的距离，否则，用 <math> AB \times AC  /  BC </math> 计算圆心到直线的距离。</li><li>4.判断圆和圆弧相交的方法与上述方法类似。</li><li>5.为了优化程序性能，在每次鼠标移动时的检测只用于检测固定物体，移动的物体则按一定时间间隔来检测，部分障碍进行双重检测，用此方法来减少运算次数。</li></ol>