

# HW1: report

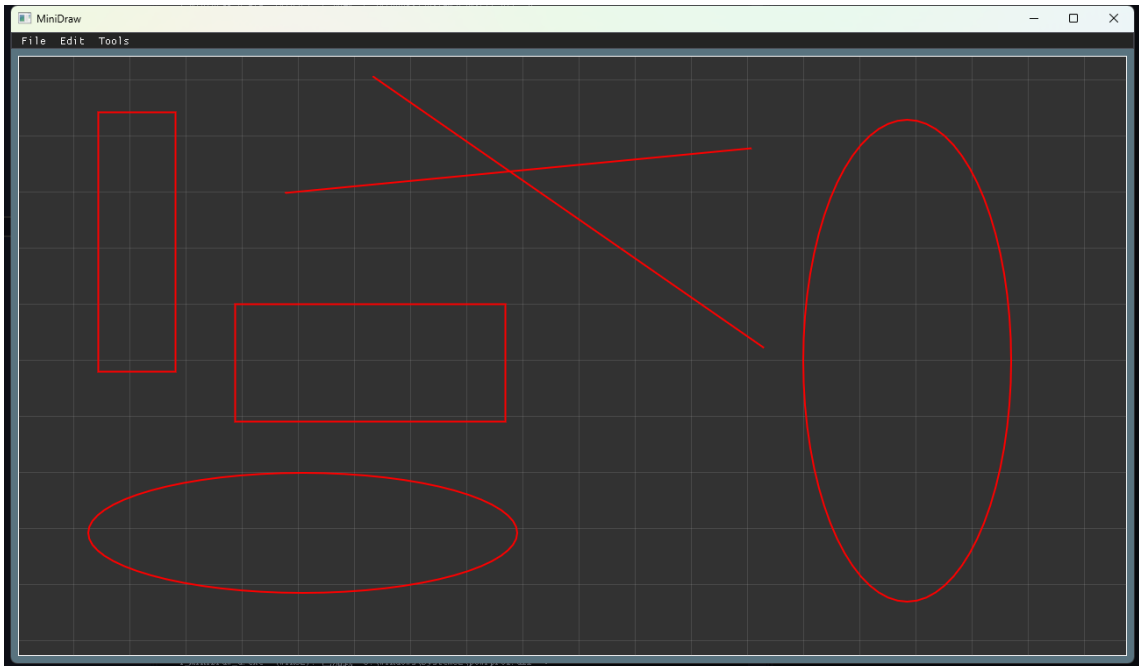
姓名：杜朋澈

ID: 68

学号：PB21050988

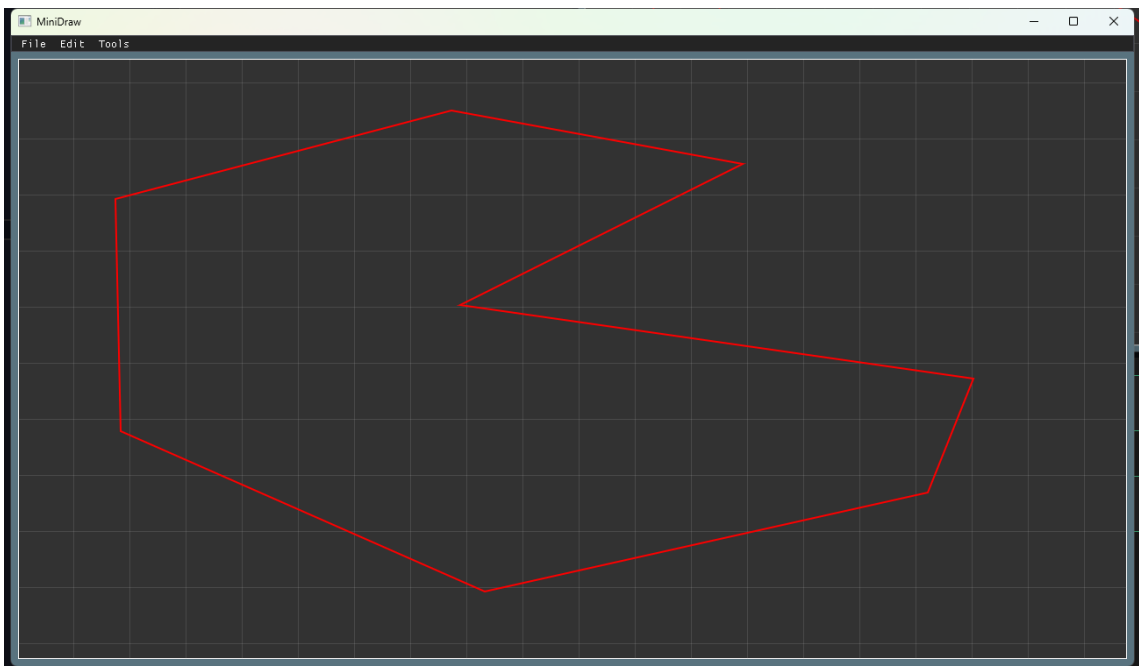
## 演示

- 绘制基本图形(Line, Rect, Ellipse)



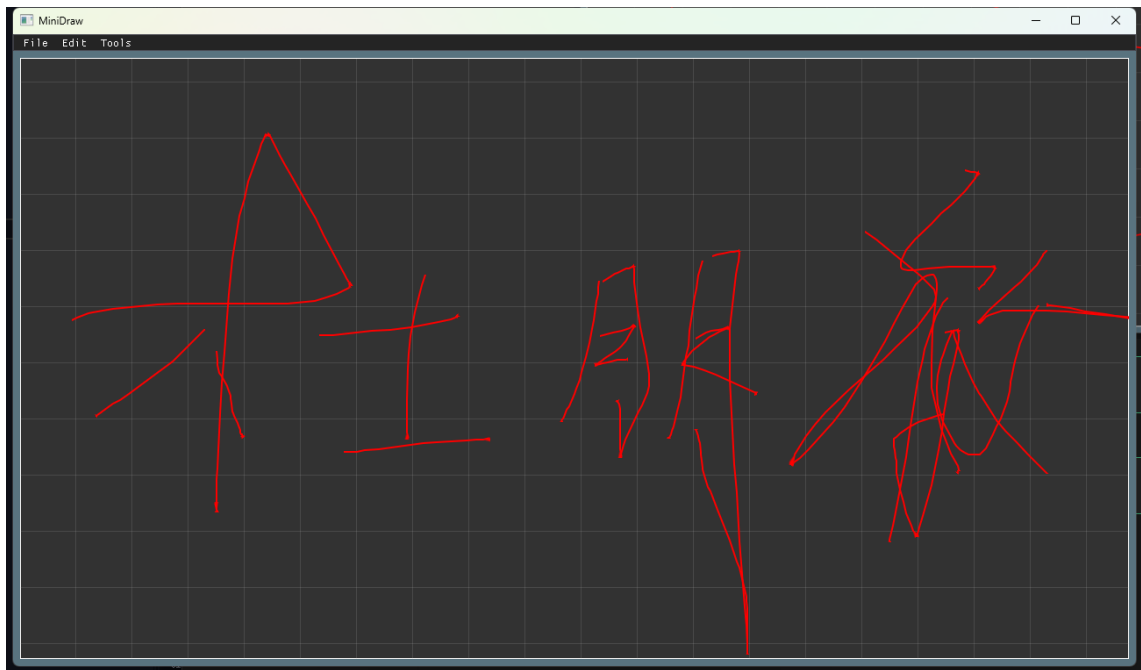
拖动左键开始绘制，松开左键停止绘制。

- 绘制多边形



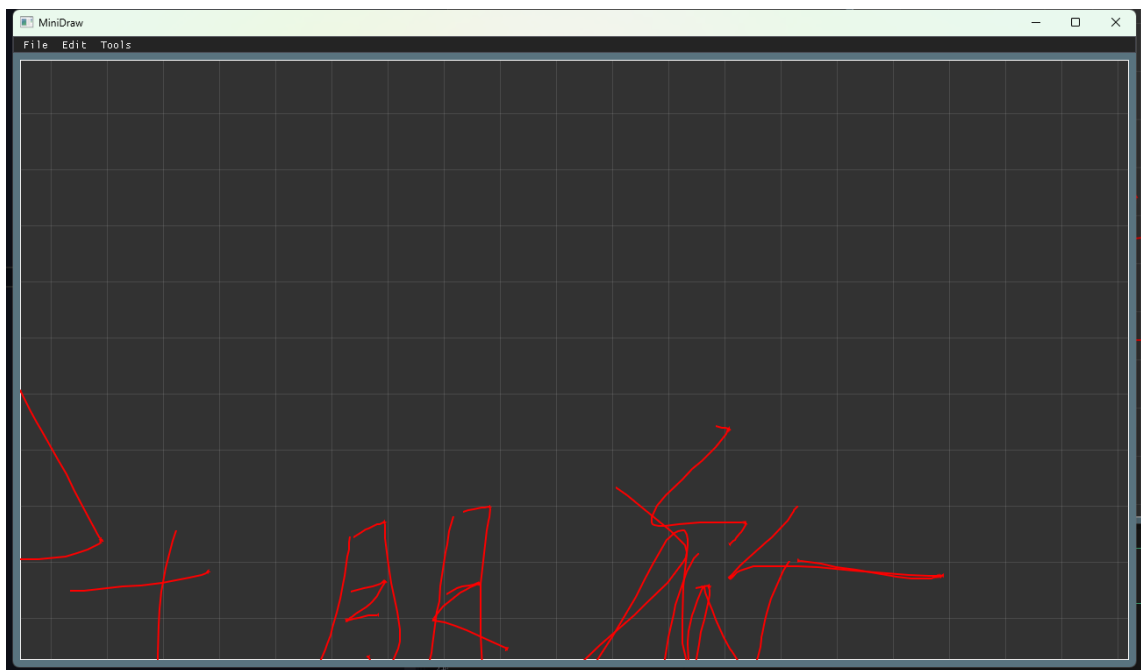
左键添加顶点，右键停止绘制。

- 自由绘制



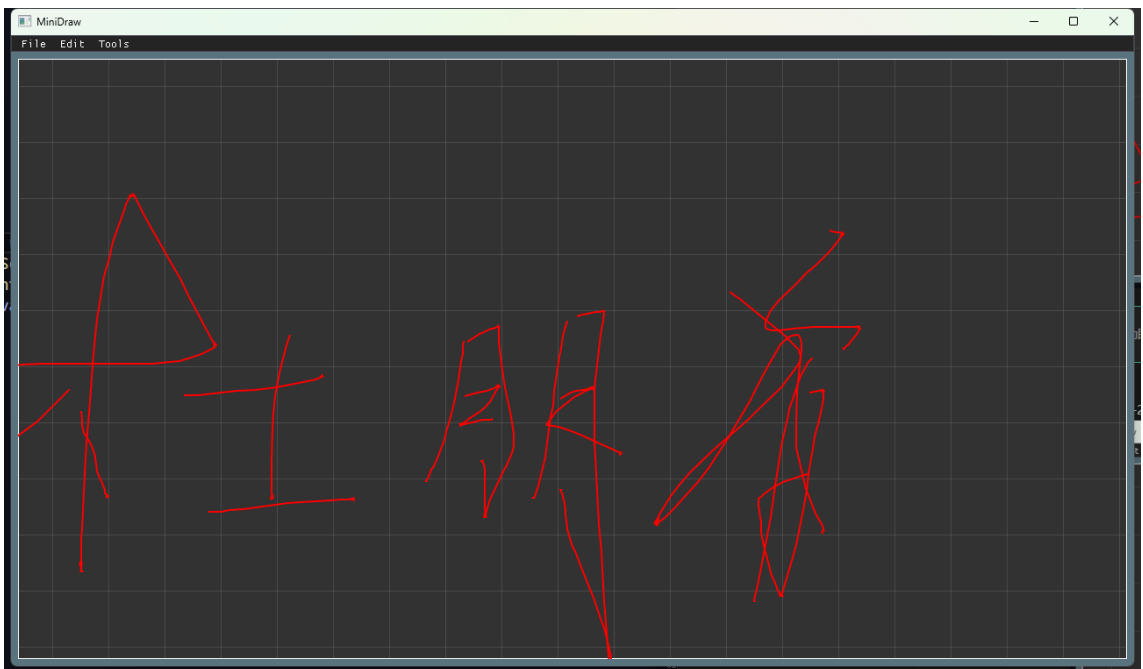
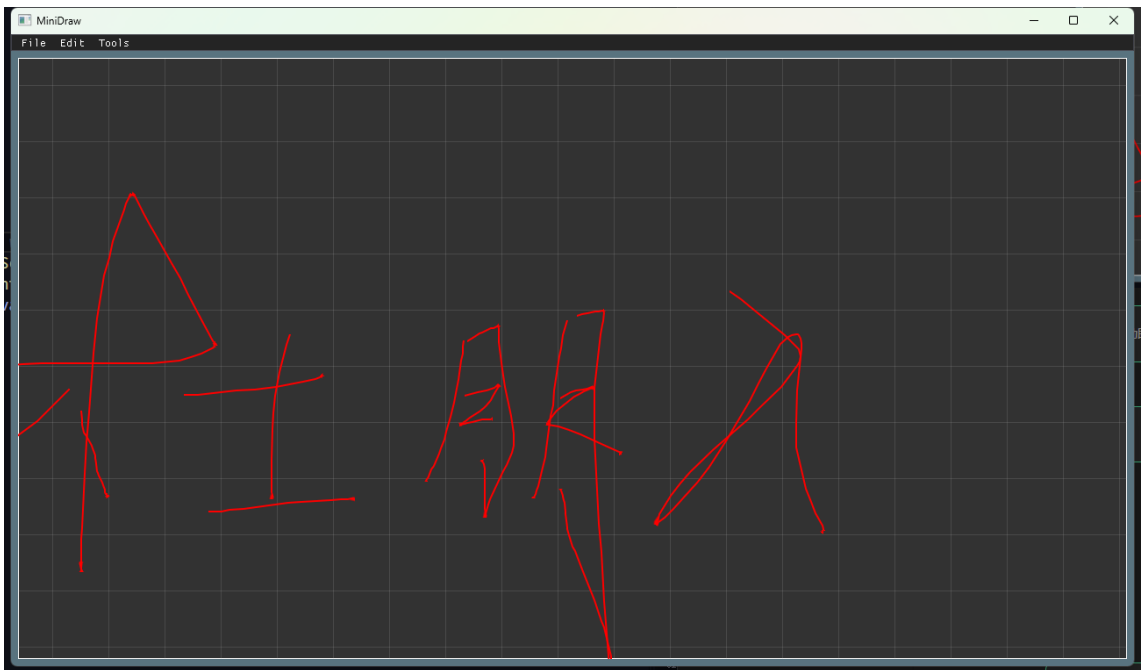
拖动左键开始绘制，松开左键停止绘制。

- 移动画布



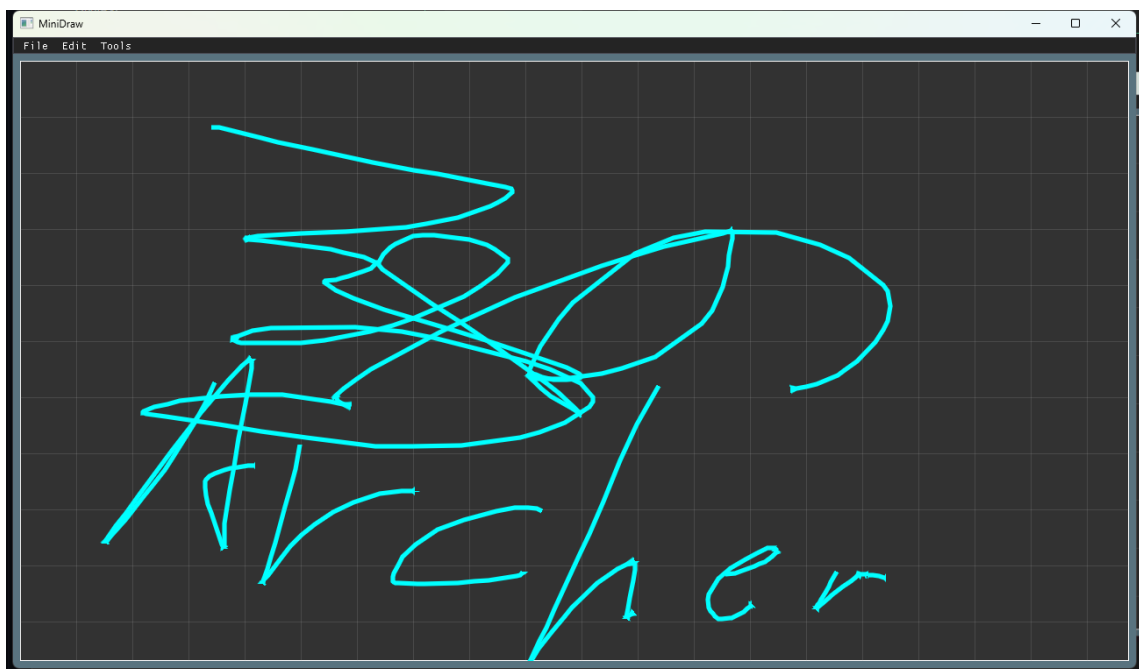
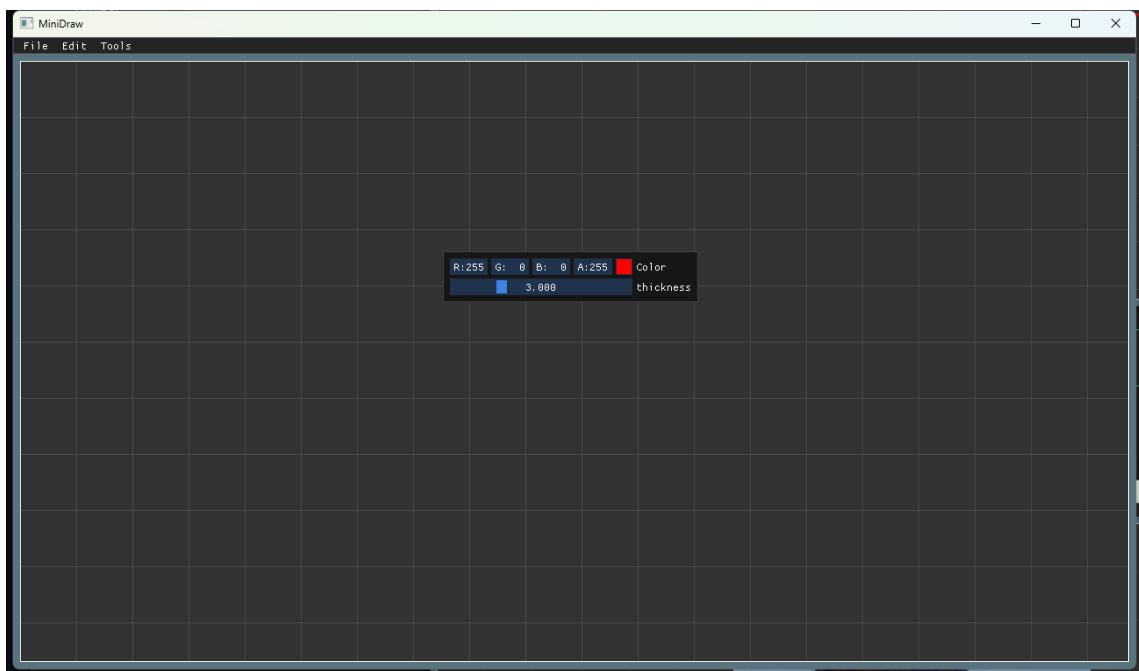
右键拖动即可

- 撤销与重做



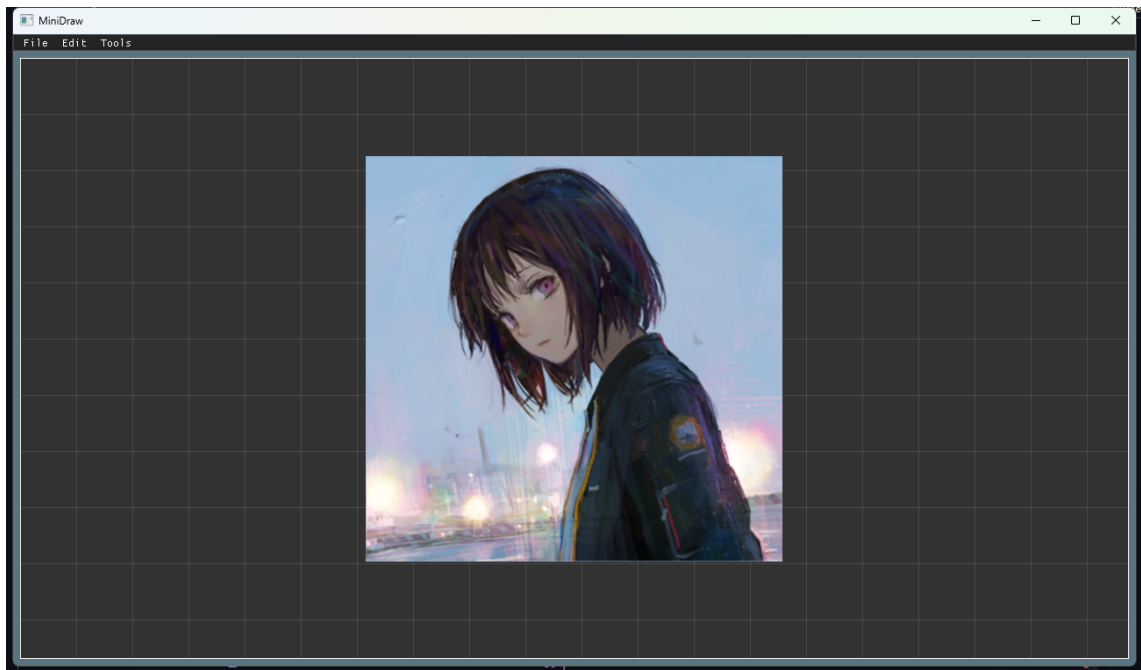
<Ctrl-Z> <Ctrl-Y> 或通过 Edit > Undo & Redo 操作均可

- 调整画笔颜色和粗细



鼠标中键弹出配置窗口

- 读取图片



## 功能设计

### 多边形与自由绘制

```
1 draw_list->AddPolyline(  
2     temp.data(),  
3     temp.size(),  
4     ImGui::ColorConvertFloat4ToU32(color),  
5     true,  
6     thickness);
```

```
1 draw_list->AddPolyline(  
2     temp.data(),  
3     temp.size(),  
4     ImGui::ColorConvertFloat4ToU32(color),  
5     false,  
6     thickness);
```

分别调取正确的内置API即可。

自由绘制时需要将draw flag设置为false，否则多边形会封口。

### 移动画布

修改主渲染循环：

```

1 void Canvas::draw()
2 {
3     //get io...
4
5     mouse_poll_event();
6
7     draw_background();
8     draw_shapes();
9     draw_context();
10
11     // some update...
12 }

```

将IO事件与具体逻辑解耦。其中，`poll_event` 为:

```

1 void Canvas::mouse_poll_event()
2 {
3     if (is_hovered_ && ImGui::IsMouseClicked(ImGuiMouseButton_Left))
4     {
5         left_click_event();
6     }
7     if (is_hovered_ && ImGui::IsMouseClicked(ImGuiMouseButton_Right))
8     {
9         right_click_event();
10    }
11    if (is_active_ && ImGui::IsMouseDragging(ImGuiMouseButton_Left))
12    {
13        left_drag_event();
14    }
15    if (is_active_ && ImGui::IsMouseDragging(ImGuiMouseButton_Right))
16    {
17        right_drag_event();
18    }
19    if (is_hovered_ && ImGui::IsMouseReleased(ImGuiMouseButton_Left))
20    {
21        left_release_event();
22    }
23 }

```

订阅 `right_drag_event` 事件，完成如下逻辑即可:

```

1     ImGuiIO& io = ImGui::GetIO();
2     scrolling.x += io.MouseDelta.x;
3     scrolling.y += io.MouseDelta.y;
4     //逐一更新各个shape在view space的位置（基于canvas space的偏移）
5     for (const auto& shape : shape_list_)
6     {
7         shape->updateOffset(io.MouseDelta.x, io.MouseDelta.y);
8     }

```

## 撤销与重做

使用两个栈 `undo_stack` 和 `redo_stack` 来完成这个功能。

`undo_stack` 和 `redo_stack` 在操作过程中的维护：

```
1 //在新的shape绘制完成时：
2 void Canvas::on_draw_stop()
3 {
4     if (current_shape_)
5     {
6         shape_list_.push_back(current_shape_);
7         //记录操作和涉及的shape（操作只有Insert，时间匆忙并没有实现
delete，剪切粘贴等操作）
8         undo_stack.push({ current_shape_, Insert });
9         //注意在执行了新的操作后都必须将redo stack清空
10        while (!redo_stack.empty())
11        {
12            redo_stack.pop();
13        }
14        current_shape_.reset();
15    }
16 }
```

`undo` 和 `redo` 的具体操作：

```
1 void Canvas::undo()
2 {
3     if (undo_stack.empty()) return;
4     //弹出栈顶元素
5     Operation op = undo_stack.top();
6     undo_stack.pop();
7     //根据操作类型修改shape（此处并未真正将shape销毁）
8     switch (op.type)
9     {
10        case Insert:
11            op.shape->enable = false;
12            break;
13            //.....
14        default: break;
15    }
16    redo_stack.push(op);
17 }
18
19 //同理
20 void Canvas::redo()
21 {
22     if (redo_stack.empty()) return;
23     Operation op = redo_stack.top();
24     redo_stack.pop();
25     switch (op.type)
```

```

26     {
27         case Insert:
28             op.shape->enable = true;
29             //.....
30             break;
31     }
32     undo_stack.push(op);
33 }

```

## 调整画笔颜色和粗细

调用如下API即可。

```

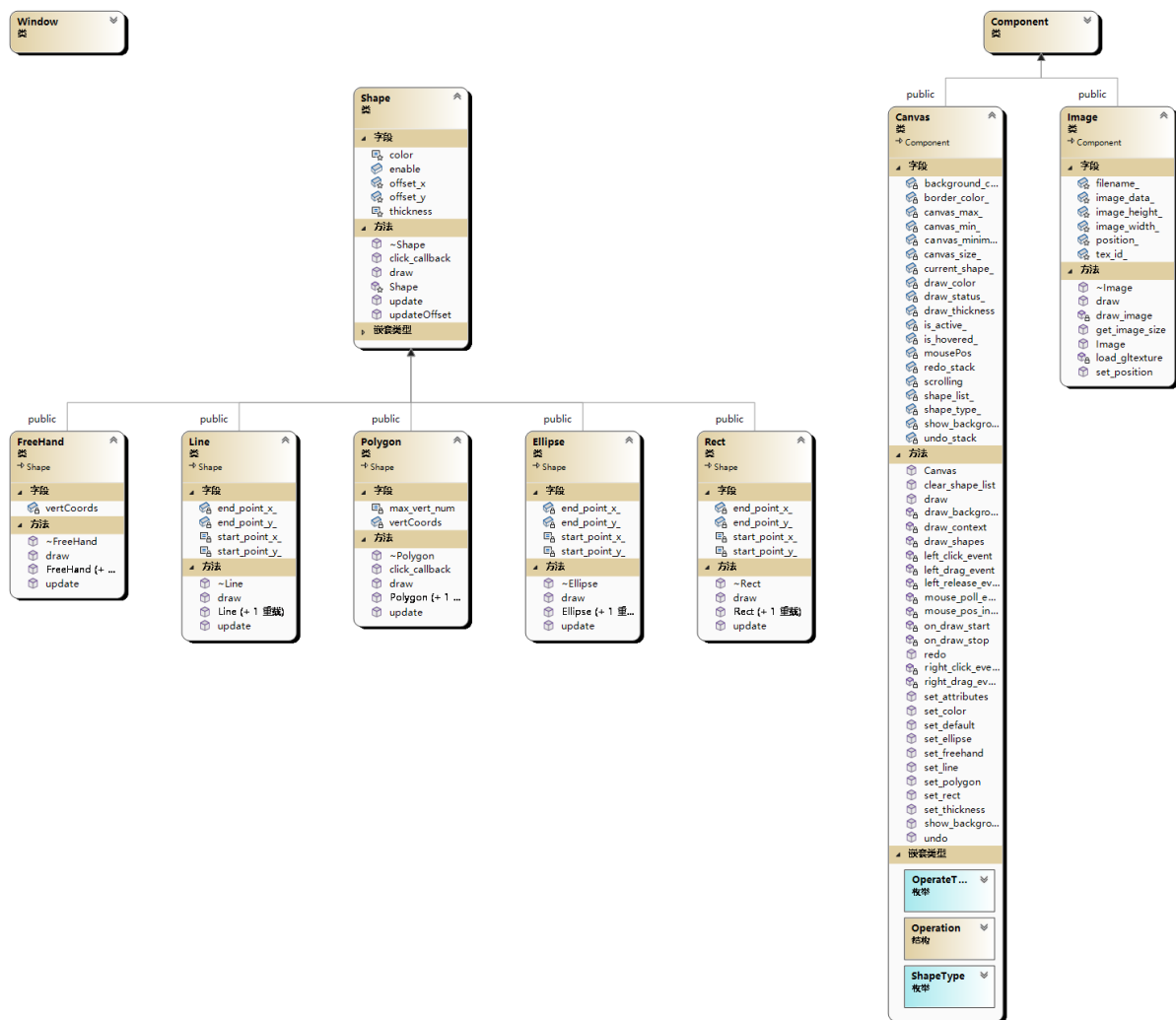
1  void Canvas::draw_context()
2  {
3      ImGui::OpenPopupOnClick(
4          "context", ImGuiPopupFlags_MouseButtonMiddle);
5      if (ImGui::BeginPopup("context"))
6      {
7          static ImVec4 colf = ImVec4(1.0f, 0.0f, 0.0f, 1.0f);
8          ImGui::ColorEdit4(
9              "color",
10             &colf.x,
11             ImGuiColorEditFlags_DisplayRGB |
12             ImGuiColorEditFlags_PickerHueBar |
13             ImGuiColorEditFlags_NoSidePreview);
14
15             static float thickness = 3.0f;
16             ImGui::SliderFloat("thickness", &thickness, 0.5, 10.0);
17
18             //在开启context菜单时不断更新颜色和粗细
19             draw_color = colf;
20             draw_thickness = thickness;
21
22             ImGui::EndPopup();
23     }
24 }

```

## UML

此处仅给出Target View的类图





和助教和老师说声抱歉，最近准备春招面试太忙啦，实验做的很赶，也没时间详细写报告

