



链滴

## 绘画板 08——删除与清屏

作者: [crick77](#)

原文链接: <https://ld246.com/article/1474783146779>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

github地址: <https://github.com/wangyuheng/painter>

DEMO地址: <http://painter.crick.wang/>

## 删除

GlobalStatus增加isRecycle()方法, 用来判断删除按钮是否被点击

```
isRecycle: function(){
    return $("#tool_delete").hasClass("active");
},
```

在Element扩展的click方法中, 增加删除判断, 执行remove()方法。并在mouseover时修改鼠标图显示

```
(function() {

    SVG.extend(SVG.Doc, {
        transformPoint: function(event) {
            event = event || window.event;
            var touches = event.changedTouches && event.changedTouches[0] || event;
            var node = this.node;
            var point = node.createSVGPoint();
            point.x = touches.pageX - window.scrollX;
            point.y = touches.pageY - window.scrollY;
            var matrix = node.getScreenCTM().inverse();
            return point.matrixTransform(matrix);
        }
    });

    SVG.extend(SVG.Element, {
        pickable: function(enabled) {
            var _ele = this;
            GlobalStatus.pushElements(_ele);
            var color = _ele._stroke;
            var width = _ele.attr("stroke-width");
            _ele.on("mouseover", function() {
                if (GlobalStatus.isPicked()) {
                    _ele.stroke({
                        width: width * 2,
                        color: 'red'
                    });
                    $("#svgPanel").css("cursor", "pointer");
                } else if (GlobalStatus.isPreFilled()) {
                    $("#svgPanel").css("cursor", "url(style/img/cur/tool_fill.cur), auto");
                } else if (GlobalStatus.isRecycle()) {
                    $("#svgPanel").css("cursor", "url(style/img/cur/tool_delete.cur), auto");
                }
            });

            _ele.on("mouseout", function() {
```

```

        if (GlobalStatus.isPicked()) {
            _ele.stroke({
                width: width,
                color: color
            });
            $("#svgPanel").css("cursor", "default");
        } else if (GlobalStatus.isPreFilled()) {
            $("#svgPanel").css("cursor", "default");
        } else if (GlobalStatus.isRecycle()) {
            $("#svgPanel").css("cursor", "default");
        }
    });
    _ele.on("click", function() {
        if (GlobalStatus.isPreFilled()) {
            if ($("#fill_color").hasClass("active")) {
                _ele.fill(GlobalStatus.getFillColor());
                _ele.style("fill-opacity", GlobalStatus.getFillOpacity());
            } else {
                _ele.style("stroke", GlobalStatus.getFontColor());
            }
        }

        } else if (GlobalStatus.isPicked()) {
            if (!_ele.attr("picked")) {
                _ele.attr("picked", true);
                _ele.handleBorder = _ele.handleBorder || new HandleBorder(svgDoc);
                _ele.handleBorder.showShade(_ele);
                GlobalStatus.pushPicked(_ele);
            } else {
                _ele.attr("picked", null);
                _ele.handleBorder && _ele.handleBorder.hideShade(_ele);
                GlobalStatus.removePicked(_ele);
            }
        } else if (GlobalStatus.isRecycle()) {
            _ele.remove();
        }
    });
    return this;
}

});
})();

```

## 清屏

单击清屏按钮时，清空所有的element元素，不需要按钮被点击状态。

此时有一个思考，是遍历所有的元素执行删除操作，还是重新设置svg doc。

先用遍历方法执行。

将elementList抽象到GlobalStatus方法中管理，并在元素完成后执行添加操作。

```
elementList : [],
pushElements: function(o){
    return this.elementList.push(o);
},
removeElements: function(o) {
    return this.elementList.remove(o);
},
getAllElements: function(){
    return this.elementList;
}
```

监听清屏按钮，遍历删除element

```
$("#clear_all").on("click", function() {
    $(GlobalStatus.getAllElements()).each(function(){
        this.remove();
    });
});
```