



链滴

绘画板 11——拖拽

作者: [crick77](#)

原文链接: <https://ld246.com/article/1474873120353>

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/>

拖拽

使用svgjs官方推荐的拖拽插件<https://github.com/wout/svg.draggable.js> 实现元素的拖拽效果

暴露了4种事件监听

- beforedrag
- dragstart
- dragmove (you can call preventDefault on this one)
- dragend

可以根据需要结合。

在Element 的mouseover事件监听中, 执行 `_ele.draggable();` 方法, 实现元素的可拖拽效果

为了避免拖拽后, 触发click事件, 导致pick状态变更, 在这里监听了dragend事件, 并将pick状态设为未选中, 则click事件触发时, 一定会处理为选中。

```
(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() {  
        console.log("element mouseover");  
        if (GlobalStatus.isPicked()) {  
          _ele.stroke({  
            width: width * 2,  
            color: 'red'  
          });  
        }  
        $("#svgPanel").css("cursor", "pointer");  
      });  
    }  
  });  
});
```

```

    _ele.draggable();
    return false;
} 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.draggable(false);
});
_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()) {
        $("#svgPanel").css("cursor", "default");
    }
});
_ele.on("click", function() {
    console.log("click");
    if (GlobalStatus.isPreFilled()) {
        if ($("#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.fire("unPick");
        } else {
            _ele.fire("pick");
        }
    }

    } else if (GlobalStatus.isRecycle()) {
        _ele.remove();
    }
});
_ele.on("mousedown", function(event) {
    console.log("element mousedown");
    if (GlobalStatus.isPicked()) {
        event.preventDefault();
        event.stopPropagation();
        return false;
    }
});

```

```

    _ele.on("dragend", function(event) {
        console.log("element dragend");
        _ele.fire("unPick");
    });
    _ele.on("beforedrag", function(event) {
        console.log("element beforedrag");
    });
    _ele.on("pick", function() {
        console.log("pick");
        _ele.attr("picked", true);
        _ele.handleBorder = _ele.handleBorder || new HandleBorder(svgDoc);
        _ele.handleBorder.showShade(_ele);
        GlobalStatus.pushPicked(_ele);
    });
    _ele.on("unPick", function() {
        console.log("unPick");
        _ele.attr("picked", null);
        _ele.handleBorder && _ele.handleBorder.hideShade(_ele);
        GlobalStatus.removePicked(_ele);
    });
    return this;
}

});
})();

```

mouseover中的draggable();和 dragend事件中的fire("unPick");都不是很好的设计，需要重构。

bug修复

dragend事件中的fire("unPick")

针对dragend事件中的fire("unPick"); 修改设计思路，在element记录开始拖拽时的坐标点数据，和dragend进行比较，如果坐标改变，表示元素发生移动，则按拖拽处理，需要触发fire("unPick"); 否则，照click事件处理。

```

_ele.dragStartPoint = null;
_ele.on("dragstart", function(event) {
    console.log("element dragstart");
    _ele.dragStartPoint = event.detail.p;
});
_ele.on("dragend", function(event) {
    console.log("element dragend");
    if (_ele.dragStartPoint.x == event.detail.p.x && _ele.dragStartPoint.y == event.detail.p.y) {

    } else {
        _ele.fire("unPick");
    }
});

```