Js元素拖拽功能实现

# 需要解决的问题

最近项目遇到了一个问题，就是用户某个操作需要弹出一个自定义的内容输入框，但是有个缺点，当浏览太大的时候没办法点击确认和取消按钮，应为这个弹出框是采用绝对定位的，取消和确认按钮都被挡住了。

# 拖拽的原理及实现

首先将元素设置为绝对定位，还用到鼠标的三个事件（mousedown、mousemove和mouseup），当用户按下鼠标触发mousedown事件设置被拖拽的元素为拖拽对象，然后移动鼠标连续触发mousemove事件，判断拖拽对象不为空时重新设置拖拽对象的位置，直到松开触发mouseup事件将拖拽对象置空。

# 遇到的问题

## 消抖问题：

刚开始没有计算元素的坐标与点击事件位置之间的差值，直接将鼠标移动的位置赋值给元素的位置（top和left属性），这样用户会感觉到鼠标跳动了一下，并且随着鼠标的移动，元素一直在鼠标的右下方。

这个问题的解决办法是：在触发mousedown的时候除了设置被拖拽的事件，也要记录拖拽对象的位置与事件发生的位置的水平和垂直之间的差值，触发mousemove事件的时候将拖拽对象的位置设置为事件发生的水平和垂直位置对应减去差值。

## 设计问题

本想尝试着为处理事件的函数传入一个参数（dom对象或者jQuery对象），为这个参数添加拖动事件（三个鼠标监听事件），但是效果不好，触发mouseup的时候有问题，后来将mouseup事件放在了document上，这样又发现了之前未发现的问题，在任何地方都可以触发拖动事件让元素被拖动，这样也不好。

源代码：

function dragEvent:(ele){

var drag = null,

diffx = 0,

diffy = 0,

$e = null;

if($.isNull(ele)) return;

if(!($e = $(ele))) return;

$(document).bind("mousedown",handleEvent);

$(document).bind("mousemove",handleEvent);

//$(document).bind("mouseup",handleEvent);

$.muphy.addEevent(document,"mouseup",handleEvent);

function handleEvent(event){

event = event || window.event;

var target = $e[0];

switch(event.type){

case "mousedown":

if($.Nvl(target.className).indexOf("draggable") > -1){

drag = target;

diffx = event.clientX - target.offsetLeft;

diffy = event.clientY - target.offsetTop;

}

break;

case "mousemove":

if(drag !== null){

drag.style.left = (event.clientX - diffx) + "px";

drag.style.top = (event.clientY - diffy) + "px";

}

break;

case "mouseup":

drag = null;

break;

}

}

}

# 效果好的解决方案

除了设置被拖动的元素为绝对定位之外，再为其添加一个类class=”draggable”，不需要给拖拽事件传入元素，将拖动事件的三个鼠标监听事件都放在document上，当触发mousedown事件时判断目标元素上是否有draggable类，有的话就将这个元素赋值给拖动对象，其他鼠标事件不变。

这个方式有一点需要注意，当拖动对象有内容的时候要为其设置padding属性以便能在边缘触发拖动事件。

源代码：

function dragEvent1(){

var drag = null,

diffx = 0,

diffy = 0;

$(document).unbind("mousedown",handleEvent);

$(document).unbind("mousemove",handleEvent);

$(document).unbind("mouseup",handleEvent);

$(document).bind("mousedown",handleEvent);

$(document).bind("mousemove",handleEvent);

$(document).bind("mouseup",handleEvent);

function handleEvent(event){

event = event || window.event;

var target = event.target || event.srcElement;

switch(event.type){

case "mousedown":

if($.nvl(target.className).indexOf("draggable") > -1){

drag = target;

diffx = event.clientX - target.offsetLeft;

diffy = event.clientY - target.offsetTop;

}

break;

case "mousemove":

if(drag !== null){

drag.style.left = (event.clientX - diffx) + "px";

drag.style.top = (event.clientY - diffy) + "px";

}

break;

case "mouseup":

drag = null;

break;

}

}

}

# 为拖放添加自定义事件

为拖动事件添加三个事件：dragstart、drag和dragend，分别表示拖动开始，正在拖动，拖动结束。

源代码：

function dragEvent:(){

var drag = null,

ce = new muphy.costomEvent();

diffx = 0,

diffy = 0;

$(document).unbind("mousedown",handleEvent);

$(document).unbind("mousemove",handleEvent);

$(document).unbind("mouseup",handleEvent);

$(document).bind("mousedown",handleEvent);

$(document).bind("mousemove",handleEvent);

$(document).bind("mouseup",handleEvent);

function handleEvent(event){

event = event || window.event;

var target = event.target || event.srcElement;

switch(event.type){

case "mousedown":

if($m.nvl(target.className).indexOf("draggable") > -1){

drag = target;

diffx = event.clientX - target.offsetLeft;

diffy = event.clientY - target.offsetTop;

event.type = "dragstart";

ce.fire(event)

}

break;

case "mousemove":

if(drag !== null){

drag.style.left = (event.clientX - diffx) + "px";

drag.style.top = (event.clientY - diffy) + "px";

event.type = "drag";

ce.fire(event)

}

break;

case "mouseup":

if(drag != null){

drag = null;

event.type = "dragend";

ce.fire(event)

}

break;

}

}

return ce;

}

有一个小问题就是：当在某个父节点阻止了鼠标的mousedown、mousemove和mouseup事件的时候，拖放不能正常运行。

# 附件：

封装的所有代码（muphy-common.js）：

(function(window,$){

var muphy = Object.create({

isNull: function(data){

if(data === null || data === undefined){

return true;

}

if( typeof data === "String" && data.trim() ==='' ){

return true;

}

if(data instanceof Array && data.length == 0){

return true;

}

return false;

},

nvl: function(data,obj){

if(obj === 0) return 0;

return data || obj || '';

},

each: function(obj,fun){

for (var key in obj) {

if(fun.call(obj[key], key, obj[key]) === false){

break;

}

}

},

addEvent: function(ele,type,handler){

if(ele.addEventListener){

ele.addEventListener(type,handler,false);

} else if(ele.attachEvent){

ele.attachEvent("on" + type, handler);

} else {

ele["on" + type] = handler;

}

},

removeEvent: function(ele,type,handler){

if(ele.removeEventListener){

ele.removeEventListener(type,handler,false);

} else if(ele.detachEvent){

ele.detachEvent("on" + type, handler);

} else {

ele["on" + type] = null;

}

},

\_dragEvent: function(ele){

var drag = null,

diffx = 0,

diffy = 0,

$e = null;

if($.isNull(ele)) return;

if(!($e = $(ele))) return;

$(document).bind("mousedown",handleEvent);

$(document).bind("mousemove",handleEvent);

//$(document).bind("mouseup",handleEvent);

$.muphy.addEevent(document,"mouseup",handleEvent);

function handleEvent(event){

event = event || window.event;

var target = $e[0];

switch(event.type){

case "mousedown":

if($m.nvl(target.className).indexOf("draggable") > -1){

drag = target;

diffx = event.clientX - target.offsetLeft;

diffy = event.clientY - target.offsetTop;

}

break;

case "mousemove":

if(drag !== null){

drag.style.left = (event.clientX - diffx) + "px";

drag.style.top = (event.clientY - diffy) + "px";

}

break;

case "mouseup":

drag = null;

break;

}

}

},

dragEvent: function(){

var drag = null,

ce = new muphy.costomEvent();

diffx = 0,

diffy = 0;

$(document).unbind("mousedown",handleEvent);

$(document).unbind("mousemove",handleEvent);

$(document).unbind("mouseup",handleEvent);

$(document).bind("mousedown",handleEvent);

$(document).bind("mousemove",handleEvent);

$(document).bind("mouseup",handleEvent);

function handleEvent(event){

event = event || window.event;

var target = event.target || event.srcElement;

switch(event.type){

case "mousedown":

if($m.nvl(target.className).indexOf("draggable") > -1){

drag = target;

diffx = event.clientX - target.offsetLeft;

diffy = event.clientY - target.offsetTop;

event.type = "dragstart";

ce.fire(event)

}

break;

case "mousemove":

if(drag !== null){

drag.style.left = (event.clientX - diffx) + "px";

drag.style.top = (event.clientY - diffy) + "px";

event.type = "drag";

ce.fire(event)

}

break;

case "mouseup":

if(drag != null){

drag = null;

event.type = "dragend";

ce.fire(event)

}

break;

}

}

return ce;

},

\_costomEvent: function(){

var handlers = {};

this.addEvent = function(type, handler){

if(typeof handlers[type] === 'undefined'){

handlers[type] = [];

}

handlers[type].push(handler);

}

this.removeEvent = function(type, handler){

if(handlers[type] instanceof Array){

muphy.each(handlers[type], function(i){

if(this === handler){

handlers[type].splice(i,1);

return false;

}

});

}

}

this.fire = function(event){

if(!event.target){

event.target = this;

}

if(handlers[event.type] instanceof Array){

muphy.each(handlers[event.type],function(){

this(event);

})

}

}

},

costomEvent: function(){}

});

(function(muphy){

muphy.costomEvent.prototype = {

constructor: muphy.costomEvent1,

handlers:{},

addEvent: function(type, handler){

if(typeof this.handlers[type] === 'undefined'){

this.handlers[type] = [];

}

this.handlers[type].push(handler);

},

removeEvent: function(type, handler){

if(this.handlers[type] instanceof Array){

var handlers = this.handlers[type];

muphy.each(handlers, function(i){

if(this === handler){

handlers.splice(i,1);

return false;

}

});

}

},

fire: function(event){

if(!event.target){

event.target = this;

}

if(this.handlers[event.type] instanceof Array){

muphy.each(this.handlers[event.type],function(){

this(event);

})

}

}

}

})(muphy);

window.$m = muphy;

})(window,jQuery);

//以下是测试代码

<!DOCTYPE html>

<html>

<head>

<title>拖动事件处理</title>

<script type="text/javascript" src='./jquery.js'></script>

<script type="text/javascript" src='./Event.js'></script>

</head>

<body>

<div>

<div id="draggable" class="draggable" style="position: relative;width: 200px;height:100px;left: 300px;top: 200px;border: 1px solid red">

<p style="border: 1px solid green">

<span>哈哈哈哈哈</span>

</p>

</div>

</div>

</body>

<script type="text/javascript">

(function(window,$){

var d = $m.dragEvent();

var fun = function(e){

console.log(e.type);

}

d.addEvent("dragstart",fun);

d.addEvent("dragend",function(){

alert(123);

});

d.addEvent("dragend",fun);

d.removeEvent("dragend",fun);

// d.ennable1("#draggable")

})(window,$)

</script>

</html>