YUI Library: Drag & Drop

2011-3-21 **v2.9**

Simple Use Case: Making an Element Draggable

```
myDDobj = new YAHOO.util.DD("myDiv");
```
Makes the HTML element whose id attribute is "myDiv" draggable.

Constructor (YAHOO.util.DD, DDProxy, DDTarget)

```
YAHOO.util.DD(str | el ref target[, str group name,
   obj configuration]);
```

Arguments:

(1) **Element:** ID or elem. ref. of the element to make draggable; deferral is supported if the element is not yet on the page.

(2) **Group Name:** An optional string indicating the DD group; DD objects only "interact with" other objects that share a group.

(3) **Configuration:** An object containing name-value pairs, used to set any of the DD object's properties.

Properties & Methods of YAHOO.util.DragDrop

Properties:

available (b)
dragOnly (b)
useShim (b)
groups (ar)
id (s)
invalidHandle Classes (s[ ])
invalidHandleIds (obj)
isTarget (b)
maintainOffset (b)
padding (int[ ])
primaryButtonOnly (b)
xTicks (int[ ])
yTicks (int[ ])

Methods:

addInvalidHandle Class (s *cssClass*)
addInvalidHandleId (s *id*)
addInvalidHandle Type (s *tagName*)
addToGroup (s *groupName*)
clearTicks()
clearConstraints()
getDragEl()
getEl()
isLocked()
lock()
removeFromGroup(o *dd*, s *group*)
removeInvalid HandleClass(s *cssClass*)
removeInvalid HandleId(s *id*)
removeInvalidHandle Type (s *tagName*)
resetConstraints()
setDragElId(s *id*)
setHandleElId (s *id*)
setOuterHandleElId (s *id*)
setPadding(i *top*, i *right*, i *bottom*, i *left*)
setXConstraint(i *left*, i *right*, i *tick size*)
setYConstraint(i *up*, i *down*, i *tick size*)
unlock()
unreg()

Properties & Methods of YAHOO.util.DD & .DDProxy

Inherit from YAHOO.util.DragDrop and add the following:

YAHOO.util.DD Properties:

scroll (b)

YAHOO.util.DDProxy Properties:

centerFrame (b)
resizeFrame (b)

Interesting Moments in Drag & Drop

Moment	Point Mode	Intersect Mode	Event (e)
onMouseDown	e	e	mousedown
startDrag	x, y	x, y	n/a
onDrag	e	e	mousemove
onDragEnter	e, id	e, DDArray	mousemove
onDragOver	e, id	e, DDArray	mousemove
onDragOut	e, id	e, DDArray	mousemove
onDragDrop	e, id	e, DDArray	mouseup
onInvalidDrop	e	e	mouseup
endDrag	e	e	mouseup
onMouseUp	e	e	mouseup

These "moments" are exposed as events on your DD instances; they are methods of YAHOO.util.DragDrop. The table above identifies the arguments passed to these methods in Point and Intersect modes.

Solutions

Add a drag handle to an existing DD object:

```
myDDobj.setHandleElId('myDragHandle');
```

Set the "padding" or "forgiveness zone" of a DD object:

```
myDDobj.setPadding(20, 30, 20, 30); //units are
   pixels, top/rt/bt/left
```

Get the 'best match' from an onDragDrop event in Intersect Mode where the dragged element is over more than one target:

```
myDDobj.onDragDrop = function(e, DDArray) {
 oDDBestMatch =
   YAHOO.util.DragDropMgr.getBestMatch(DDArray);}
```

Override an interesting moment method for a DD object instance:

```
myDDobj = new YAHOO.util.DD("myDiv");
myDDobj.startDrag = function(x,y) {
  this.iStartX = x; this.iStartY = y;
}
```

Change the look and feel of the proxy element at the start of a drag event using YAHOO.util.DDProxy:

```
myDDobj.startDrag(x,y) {
  YAHOO.util.Dom.addClass(this.getDragEl(),
   "myCSSClass"); }
```

Lock Drag and Drop across the whole page:

```
YAHOO.util.DragDropMgr.lock();
```

Switch to Intersect Mode:

```
YAHOO.util.DragDropMgr.mode =
   YAHOO.util.DragDropMgr.INTERSECT;
```

Drag & Drop Manager: Properties

clickPixelThresh (i)
clickTimeThresh (i)
useShim (b)
mode either YAHOO.util.DragDropMgr.POINT or .INTERSECT
preventDefault (b)
stopPropagation (b)
useCache (b)

Drag & Drop Manager: Methods

oDD=instance of DragDrop object

getBestMatch(a [oDDs])
getDDById(s *id*)
getLocation(oDD)
getRelated(oDD, b *targets only*)
isDragDrop(s *id*)
isHandle(s *DDId*, s *HandleId*)
isLegalTarget(oDD, oDD *target*)
isLocked()
lock()
refreshCache()
swapNode()
unlock()

*__Note:__ YAHOO.util.DragDropMgr is a singleton; changes made to its properties (such as locking or unlocking) affect Drag and Drop globally throughout a page.

Dependencies

Drag & Drop requires the YAHOO object, DOM, and Event.