QTVR and Lingo

Using QTVR within Director involves layering Lingo over the top of QTVR’s capabilities and setting up communication between the two. Within QTVR’s authoring environment, nodes and hotspots are created. Nodes are separate QTVR panoramas that can be stitched together like a series of rooms connected in space. Each hotspot has an ID, a type, and a bounding rectangle that is set using QTVR authoring tools. The various properties of nodes and hotspots can be read by Director and used to activate Lingo commands through rollovers and mouse clicks. While the tilting, panning, and zooming of the QTVR image through user dragging/clicking is typically QTVR-controlled, Lingo can be used to measure and readjust these values through nudging (discontinuous movement in a single direction), swinging (continuous movement to a new set of values), or resetting individual values.

To determine whether a cast member or sprite is a QTVR digital video, test the `isVRMovie` property, which returns a `TRUE` or `FALSE`.

```
member(whichCastMember).isVRMovie
isVRMovie of member whichCastMember
sprite(whichSprite).isVRMovie
isVRMovie of sprite whichSprite
```

To control how Director passes mouse clicks on a QuickTime sprite, set the `mouseLevel` sprite property to one of the following values:

- `#controller`—Passes clicks only on the movie controller to QuickTime. Director responds only to mouse clicks that occur outside the controller. This is the standard behavior for QuickTime sprites other than QTVR.
- `#all`—Passes all mouse clicks within the sprite’s bounding rectangle to QuickTime. No clicks pass to other Lingo handlers.
- `#none`—Does not pass any mouse clicks to QuickTime. Director responds to all mouse clicks.
- `#shared`—Passes all mouse clicks within a QTVR sprite’s bounding rectangle to QuickTime and then passes these events to Lingo handlers. This is the default value for QTVR.

```
sprite(whichQuickTimeSprite).mouseLevel
the mouseLevel of sprite whichQuickTimeSprite
```

**Example.** This frame script checks to see if the name of the QuickTime sprite in channel 5 contains the string “QTVR.” If it does, this script sets `mouseLevel` to `#all`; otherwise, it sets `mouseLevel` to `#none`.

```
on prepareFrame
  if the name of member (the memberNum of sprite 5) ¬
  contains “QTVR” then
    set the mouseLevel of sprite 5 = #all
  else
    set the mouseLevel of sprite 5 = #none
  end if
end
```

To set the codec quality to use when the user clicks and drags a QTVR sprite, set the `motionQuality` sprite property to `#minQuality`, `#maxQuality`, or `#normalQuality`.

```
motionQuality of sprite whichQTVRSprite
```
To specify the codec quality to use when a QTVR panorama image is static, set the `staticQuality sprite property` to `#minQuality`, `#maxQuality`, or `#normalQuality`.

`staticQuality of sprite whichQTVRSprite`

To enable or disable the specified hotspot, use the `enableHotSpot` command.

`enableHotSpot(sprite whichQTVRSprite, hotSpotID, trueOrFalse)`

To find the approximate bounding rectangle for a specific hotspot in a QTVR sprite, use the `getHotSpotRect` function. If the hotspot doesn’t exist or isn’t visible on the Stage, this function returns `rect(0, 0, 0, 0)`. If the hotspot is partially visible, this function returns the bounding rectangle for the visible portion.

`getHotSpotRect(sprite whichQTVRSprite, hotSpotID)`

To specify the name of the handler that runs when the pointer enters a QTVR hotspot that is visible on the Stage, set the `hotSpotEnterCallback`. This property contains the name of the handler that runs when the cursor enters a QTVR hot spot that is visible on the Stage. The QTVR sprite receives the message first. The message has two arguments: the `me` parameter and the ID of the hot spot that the cursor entered. To clear, set to 0.

`hotSpotEnterCallback of sprite whichQTVRSprite`

To find the ID of the hotspot, if any, that is at a specific point on the Stage, use the `ptToHotSpotID` function. If there is no hot spot, the function returns 0.

`ptToHotSpotID(sprite whichQTVRSprite, point)`

To specify the name of the handler that runs when the user clicks a hotspot in a QTVR sprite, set the `triggerCallback` sprite property. The handler is sent two arguments: the `me` parameter and the ID of the hot spot that the user clicked. The value that the handler returns determines how the movie processes the hot spot. If the handler returns `#continue`, the QTVR sprite continues to process the hot spot normally. If the handler returns `#cancel`, the default behavior for the hot spot is canceled. Set this property to 0 to clear the callback. The QTVR sprite receives the message first.

`triggerCallback of sprite whichQTVRSprite`

Example. This statement sets the callback handler for a QTVR sprite to the handler named `MyHotSpotCallback` when the playback head first enters the sprite span. Every time that hot spot is triggered, the `MyHotSpotCallback` handler is executed.

```
property pMySpriteNum
on beginSprite
    set pMySpriteNum = the spriteNum of me
    set the triggerCallback of sprite pMySpriteNum = #MyHotSpotCallback
end

on MyHotSpotCallback me, hotSpotID
    put "Hotspot"&&hotSpotID&&"was just triggered"
end
```
on endSprite me
    set the triggerCallback of sprite pMySpriteNum = 0
end

To determine the name of the handler that runs when the pointer leaves a QTVR hotspot that is visible on the Stage, set the hotSpotExitCallback property. The QTVR sprite receives the message first. The message has two arguments: the me parameter and the ID of the hot spot that the cursor entered. To clear the callback, set this property to 0.

hotSpotExitCallback of sprite whichQTVRSprite

To specify the ID of the current node that a QTVR sprite displays, set the node QTVR sprite property.

node of sprite whichQTVRSprite

To specify the name of the handler that runs after the QTVR sprite switches to a new active node on the Stage, set the nodeEnterCallback QTVR sprite property. The message has two arguments: the me parameter and the ID of the node that is being displayed. The QTVR sprite receives the message first. To clear the callback, set this property to 0.

nodeEnterCallback of sprite whichQTVRSprite

To specify the name of the handler that runs when a QTVR sprite is about to switch to a new active node on the Stage, set the nodeExitCallback QTVR sprite property. The message has three arguments: the me parameter, the ID of the node that the movie is about to leave, and the ID of the node that the movie is about to switch to. The value that the handler returns determines whether the movie goes on to the next node. If the handler returns #continue, the QTVR sprite continues with a normal node transition. If the handler returns #cancel, the transition doesn't occur and the movie stays in the original node. Set this property to 0 to clear the callback. The QTVR sprite receives the message first.

nodeExitCallback of sprite whichQTVRSprite

To determine the type of node that is currently on the Stage, test the nodeType QTVR sprite property. Possible values are #object, #panorama, or #unknown. (#unknown is the value for a sprite that isn’t a QTVR sprite.)

nodeType of sprite whichQTVRSprite

To set the current pan of the QTVR sprite, set the pan QTVR sprite property. The value is in degrees.

pan of sprite whichQTVRSprite

To nudge a QTVR sprite in a specific direction, use the nudge command. Possible values for #direction are #down, #downLeft, #downRight, #left, #right, #up, #upLeft, and #upRight.

nudge(sprite whichQTVRSprite, #direction)
To specify the type of warping performed on the panorama of a QTVR sprite, set the `warpMode` QTVR sprite property. Possible values are `#full`, `#partial`, and `#none`. This property can be tested and set. When tested, if the values for the static and motion modes differ, the property’s value is the value set for the current mode. When set, the property determines the warping for both the static and motion modes.

`warpMode` of sprite `whichQTVRSprite`

To specify a QTVR sprite’s current field of view, set the `fieldOfView` QTVR sprite property (in degrees).

`fieldOfView` of sprite `whichQTVRSprite`

To swing a QTVR sprite to a specific pan, tilt, or field of view, set the `swing` sprite property. The `swing` is a smooth "camera dolly" effect.

- `whichQTVRSprite`—Sprite number of the sprite with the QTVR member.
- `pan`—New pan position, in degrees.
- `tilt`—New tilt, in degrees.
- `fieldOfView`—New field of view, in degrees.
- `speedToSwing`—Rate at which the swing should take place (1 to 10 (slow to fast)).

The function returns immediately, but the sprite continues to change view until it reaches the final view. The duration required to change to the final settings varies depending on machine type, size of the sprite rect, color depth of the screen, and other normal performance effecting parameters. To check if the swing has completed, check if the pan of sprite has arrived at the final value.

`swing` (sprite `whichQTVRSprite`, `pan`, `tilt`, `fieldOfView`, `speedToSwing`)

Also see the following set of hints and scripts:
- `http://www.geocities.com/SiliconValley/Heights/6791`