3D CAMERA SCRIPT INTERFACE

The `I3DCameraObjectType` interface derives from IObjectClass to add APIs specific to the 3D Camera plugin.

Note this class derives from the object class interface, not the instance interface. Its default name is not a valid JavaScript identifier as it starts with a number, meaning it needs to be accessed with the syntax `runtime.objects["3DCamera"]`. You may wish to rename the object to a valid JavaScript identifier such as *Camera3D*, so that it can be accessed more conveniently with the syntax `runtime.objects.Camera3D`.

3D Camera APIs

---

lookAtPosition(camX, camY, camZ, lookX, lookY, lookZ, upX, upY, upZ)

Set the position and orientation of the 3D Camera using a camera position, a position for the camera to look towards, and an up vector. The camera and look-at positions are given as 3D co-ordinates. The up vector is a 3D vector specifying which way is up, as this is also necessary to determine how to orient the camera towards the look-at position. The default up vector is (0, 1, 0), i.e. up is the positive direction on the Y axis, suitable for a top-down view.

---

lookParallelToLayout(camX, camY, camZ, lookAngle)

Set the position and orientation of the 3D Camera using a camera position and a camera angle in radians. Whereas the default view is top-down, this sets a camera position looking along the layout, such that the layout appears as the floor at the bottom of the screen. This is a shortcut for using the *Look at position* action looking towards a 2D angle with an up vector of (0, 0, 1).

---

restore2DCamera()

Restore the camera to its default 2D behavior, using the standard scrolling features to move the view.

---

moveAlongLayoutAxis(distance, axis, which)

Move the camera position, the look position, or both, a distance along an axis relative to the layout. The distance can be negative to move in the opposite direction to the given axis. *axis* must be a string of `"x"`, `"y"` or `"z"`. *which* must be a string of `"camera"`, `"look"` or `"both"`.

> *Note the scale on the Z axis is different to the X and Y axes.*

---

moveAlongCameraAxis(distance, axis, which)

Move the camera position, the look position, or both, a distance along an axis relative to the current camera orientation. The distance can be negative to move in the opposite direction to the given axis. *axis* must be a string of "forward", "up" or "right". *which* must be a string of `"camera"` , `"look"` or `"both"` .

> *Note the scale on the Z axis is different to the X and Y axes.*

---

rotateCamera(rotateX, rotateY, minPolar, maxPolar)

Moves the camera look-at position according to X and Y rotations in radians. Note that a 3D Camera must first have been enabled using the `lookAtPosition()` or `lookParallelToLayout()` methods, since these also define the starting orientation for the camera to rotate around. Typically the rotation values will be provided by the mouse movement for "mouse look". The min/max polar values are also in radians and limit the Y rotation, stopping the camera rotating too close to directly upwards or downwards.

---

getCameraPosition()

Return an array of `[x, y, z]` with the current 3D position of the camera.

---

getLookPosition()

Return an array of `[x, y, z]` with the current 3D position of the position the camera is pointing at.

---

getLookVector()

Return an array of `[x, y, z]` with the current vector of the direction the camera is pointing in, including camera rotation.

---

getForwardVector()

Return an array of `[x, y, z]` with a 3D unit vector pointing in the direction of the camera.

> *Note this does not include camera rotation. Use `getLookVector()` to get the vector of the direction the camera is pointing in including camera rotation.*

---

getRightVector()

Return an array of `[x, y, z]` with a 3D unit vector pointing to the right of the camera, perpendicular to the forward vector.

----

getUpVector()

Return an array of `[x, y, z]` with the camera up vector, which helps determine the camera orientation. Note this is recomputed from the given camera and look positions, so may not be exactly the same as the up vector given in *lookAtPosition()*.

----

zScale

A read-only number with the number of pixels per unit on the Z axis. See the *Z scale* property in the 3D Camera manual entry for more details.

----

fieldOfView

Set or get the angle of the field of view in radians. Note this only has an effect when *Z Axis scale* is set to *Regular* in Project Properties. The default field of view is also set in Project Properties.