

3D CAMERA

View online: <https://www.construct.net/en/make-games/manuals/construct-3/plugin-reference/3d-camera>

In Construct the standard view is restricted to 2D and scrolling to an X and Y co-ordinate. The **3D Camera** object can change the viewpoint of the game to anywhere in 3D, such as positioning the camera to an X, Y and Z co-ordinate, looking towards another X, Y and Z co-ordinate. See the [First-person shooter example](#) for a demonstration of what the 3D Camera object can do.

You can learn more about 3D Camera and how it works with other 3D features in the tutorial [Using 3D in Construct](#).

Scripting

When using JavaScript or TypeScript coding, the features of this object can be accessed via the `I3DCameraObjectType` script interface.

Using a 3D Camera

By default Construct will use the standard 2D view, using the standard scrolling features to pan the view in 2D. The *Look at position*, *Look parallel to layout* and *Set position* actions enable a 3D view, which overrides the standard 2D view - the standard scrolling features will no longer have any effect on the 3D view. The standard 2D view can be restored using the *Restore 2D camera* action.

The 3D Camera object will only apply to layers which have their **Rendering mode** property set to **3D**. Layers using a **2D** rendering mode will ignore the 3D Camera and remain using a standard 2D view using the standard scrolling features. This is useful for things like a 2D interface displayed on top of a 3D view.

3D Audio

When moving the camera in 3D, it can be useful to play positioned sounds which will adjust playback to make them sound like they are coming from a position in 3D space. This can be done by setting the listener orientation in the [Audio object](#), and playing positioned sounds either at objects (which will also use their Z elevation as the Z co-ordinate of the sound) or at a position which allows specifying the X, Y and Z co-ordinates of the sound.

3D Camera properties

The 3D Camera object has two properties: **Z scale** and **Default camera Z**. These are related and are influenced by the *Z axis scale* [project property](#). The properties are defined as follows.

Z scale

The number of pixels per unit on the Z axis. With a *Regular* Z axis scale, this is always 1, as the Z axis uses the same scale as the other axes.

Default camera Z

The default position of the camera on the Z axis, producing 2D rendering at 100% scale on the layout. With a *Normalized* Z axis scale, this is always 100.

The relationship between the properties and the *Z axis scale* property is as follows:

- With the **Normalized** Z axis scale (the default), the *Default camera Z* is set to 100. Then the *Z scale* tells you how many pixels each unit on the Z axis is equivalent to.
- With the **Regular** Z axis scale, the *Z scale* is always 1, as the Z axis uses the same scale as the X and Y axes. The *Default camera Z* is then adjusted to ensure 2D rendering appears at 100% scale. Note the *Field of view* project property also affects the *Default camera Z*.

The *Normalized* Z axis scale is designed for 2D projects using simple 3D features like Z elevation. If your project uses 3D camera to employ a fully 3D view, it will likely benefit from using the *Regular* Z axis scale.

3D Camera conditions

The 3D Camera object has no conditions.

3D Camera actions

Look at position

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.

Look parallel to layout

Set the position and orientation of the 3D Camera using a camera position and a camera angle in degrees. 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).

Move along camera axis

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, for example to move left when the specified axis is *Right*.

Note the scale on the Z axis can be different to the X and Y axes.

Move along layout axis

Move the camera position, the look position, or both, a distance along an axis relative to the layout. The distance can be negative, for example to move left when the specified axis is *X*.

Note the scale on the Z axis can be different to the X and Y axes.

Restore 2D camera

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

Rotate camera

Moves the camera look-at position according to X and Y rotations in degrees. Note that a 3D Camera must first have been enabled using the *Look at position*, *Look parallel to layout* or *Set position* actions, since these also define the starting orientation that this action rotates around. Typically the rotation values will be provided by the [Mouse](#) object's *MovementX* and *MovementY* expressions in an *On movement* trigger to achieve "mouse look" - see [First-person platformer](#) for an example of this.

Set field of view

Set the angle of the field of view in degrees. 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.

Set position

Set the camera position or the look position to a 3D co-ordinate. This can be used to control the camera or look positions independently without having to always specify both, such as to move the camera while using mouse look to control the look direction.

3D Camera expressions

CameraX

CameraY

CameraZ

Get the current 3D position of the camera.

LookX**LookY****LookZ**

Get the current 3D position of the position the camera is pointing at.

LookVectorX**LookVectorY****LookVectorZ**

Get the current vector of the direction the camera is pointing in, including camera rotation (i.e. changes applied with the *Rotate camera* action for purposes like mouse look).

ZScale

The number of pixels per unit on the Z axis. See *Z scale* under *3D Camera properties* for more details.

DefaultCameraZ

The default position of the camera on the Z axis, producing 2D rendering at 100% scale on the layout. See *Z scale* under *3D Camera properties* for more details.

FieldOfView

The current field of view in degrees. This is only applicable when *Z axis scale* is set to *Regular*.

CanvasToLayerX(layer, x, y, layerZ)**CanvasToLayerY(layer, x, y, layerZ)**

Transform a position in canvas co-ordinates to layer co-ordinates on a Z plane given by *layerZ*. This is similar to the [system expressions](#) of the same name, but working in 3D.

LayerToCanvasX(layer, x, y, z)**LayerToCanvasY(layer, x, y, z)**

Transform a position in 3D layer co-ordinates to 2D canvas co-ordinates. This is similar to the [system expressions](#) of the same name, but working in 3D.

LayerToLayerX(fromLayer, toLayer, x, y, z)**LayerToLayerY(fromLayer, toLayer, x, y, z)**

Calculate the 2D position on a second layer (*toLayer*) that corresponds to a 3D position given on a first layer (*fromLayer*). This is similar to the [system expressions](#) of the same name, but working in 3D.

ViewportBottomLeftX(layer)

ViewportBottomLeftY(layer)

ViewportBottomRightX(layer)

ViewportBottomRightY(layer)

ViewportTopLeftX(layer)

ViewportTopLeftY(layer)

ViewportTopRightX(layer)

ViewportTopRightY(layer)

Return the X and Y position in layer co-ordinates of the four corners of the visible viewport, taking in to account the layer's Z elevation. These expressions are similar to the viewport [system expressions](#), but when using a 3D camera the viewport area can be an irregular quadrilateral instead of a simple 2D rectangle, so these expressions provide four separate positions.

These expressions can return NaN (Not A Number) if a corner of the viewport does not intersect the layer plane.

CameraXRotation

CameraYRotation

Return the X and Y rotation of the camera in degrees, as set by the *Rotate camera* action.

ForwardX

ForwardY

ForwardZ

Returns a 3D unit vector pointing in the direction of the camera.

Note this does not include camera rotation. Use the LookVectorX/Y/Z expressions to get the vector of the direction the camera is pointing in including camera rotation.

RightX

RightY

RightZ

Returns a 3D unit vector pointing to the right of the camera, perpendicular to the forward vector.

UpX**UpY****UpZ**

Returns a 3D unit vector for 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 the *Look at position* action.