

SPRITE SCRIPT INTERFACE

View online: <https://www.construct.net/en/make-games/manuals/construct-3/scripting/scripting-reference/plugin-interfaces/sprite>

The `ISpriteInstance` interface derives from `IWorldInstance` to add APIs specific to the `Sprite` plugin.

Sprite object types also define a `ISpriteObjectType` interface which derives from `IObjectClass` to add Sprite APIs that affect the object type, such as dynamic animations.

Sprite instance events

See [instance event](#) for standard instance event object properties.

"framechange"

Fired when the currently displayed animation frame changes during playback of an animation. The event object has the additional properties:

- `animationName` : a string of the name of the currently playing animation
- `animationFrame` : a zero-based index with the animation frame number of the new frame in its animation

"animationend"

Fired when playback of the current animation reaches the end. The event object has the additional properties:

- `animationName` : a string of the name of the animation that finished

Sprite instance APIs

animation

A reference to a `IAnimation` script interface representing the current animation, which can be used to access additional details such as the frames in the animation.

setAnimation(name, from = "beginning")

Set the current animation by a string of its name (case-insensitive). If the animation name does not exist, an exception will be thrown. `from` can be set to either `"current-frame"` to

switch to the same frame index in the new animation, or `"beginning"` to rewind to the first frame.

Note that if the animation being set is already playing, this method does nothing, even if set to play from the beginning. If you intend to restart the animation, use `startAnimation("beginning")` instead.

getAnimation(name)

Get an `Animation` for an animation in the Sprite object by a case-insensitive string of its name. Returns `null` if no animation is found.

animationName

A read-only string of the current animation name. Use the `setAnimation()` method to change the animation.

startAnimation(from = "current-frame")

Start playback of the current animation. `from` can be set to either `"current-frame"` to play from the existing frame, or `"beginning"` to play from the first frame.

stopAnimation()

Stop playback of the current animation.

animationFrame

The zero-based index of the current animation frame.

animationFrameTag

A string of the current animation frame tag (which is an empty string when not set). If assigned a new tag, and the current animation has multiple animation frames with the same tag, then it will use the first one.

animationSpeed

The current animation playback speed, in animation frames per second.

animationRepeatToFrame

The zero-based index of the animation frame to rewind to when repeating an animation.

imageWidth

imageHeight**getImageSize()**

Read-only numbers indicating the size of the current animation frame's source image, in pixels. The method allows getting both values at the same time.

getImagePointCount()

Return the number of image points on the current animation frame.

getImagePointX(nameOrIndex)**getImagePointY(nameOrIndex)****getImagePointZ(nameOrIndex)****getImagePoint(nameOrIndex)**

Return the location of an image point on the current animation frame in layout co-ordinates. The Z co-ordinate can be useful when using 3D mesh distortion. Image points are identified either by a case-insensitive string of their name, or their index. Note image point 0 is the origin, so index 1 is the first image point. If the image point is not found, this returns the origin instead. The `getImagePoint` variant returns `[x, y, z]`.

getPolyPointCount()

Return the number of collision polygon points on the current animation frame.

getPolyPointX(index)**getPolyPointY(index)****getPolyPoint(index)**

Return the location of a collision polygon point on the current animation frame in layout co-ordinates, by its zero-based index. The `getPolyPoint` variant returns `[x, y]`.

The first poly point is repeated again at the end (at the index `getPolyPointCount()`) since it makes it easier to iterate through each edge of the collision polygon.

setSolidCollisionFilter(isInclusive, tags)

Enable or disable collisions with the Solid behavior according to tags. Specify `tags` using either a string of space-separated tag names, or any iterable of strings such as an array or `Set`. If `isInclusive` is true, collisions are only enabled with solids that match any of the given tags; if no tags are specified, collisions are disabled with all solids. If `isInclusive` is false, collisions are disabled with solids that match any of the given tags; if no tags are specified, collisions are enabled for all solids (the default).

async replaceCurrentAnimationFrame(blob)

Replace the current animation frame image with the contents of a **Blob** representing an image file such as a PNG image. The blob can be locally generated or retrieved from a URL, for example:

```
// Loading an image from a URL
const response = await fetch(url);
const blob = await response.blob();
await spriteInst.replaceCurrentAnimationFrame(blob);
```

Sprite object type APIs

These **ISpriteObjectType** APIs are available on the object type class, e.g.

runtime.objects.MySprite, rather than instances. Note this means that any changes, such as to the animations or animation frames, will affect all instances.

getAnimation(name)

Return an **IAimation** representing the animation with the given name, or **null** if none exists.

getAllAnimations()

Return an array of **IAimation** representing all animations that the Sprite object type has.

addAnimation(animName)

Add a new animation with the given name and return an **IAimation** representing it. The name must be unique. The added animation will have a single transparent animation frame sized 100x100.

removeAnimation(animName)

Remove an animation with the given name. An exception will be thrown if an animation with the given name does not exist, or the specified animation is the last one, as Sprite objects must have at least one animation.

addAnimationFrame(animName, where)

Add an animation frame to the animation with the specified name. The new frame will be transparent and sized 100x100. The **where** parameter must be either a number for a zero-based index of where to insert the frame, and can be -1 to add to the end, or a string of an animation frame tag to insert relative to. When adding an animation frame not at the end, it is inserted just before the given frame. The method returns an **IAimationFrame** representing the added frame.

removeAnimationFrame(animName, where)

Remove an animation frame from the animation with the specified name. The last frame cannot be removed, as animations must have at least one frame. The `where` parameter must be either a number for a zero-based index of where to remove a frame, and can be -1 to remove from the end, or a string of an animation frame tag to remove.