OBJECT INSTANCES

View online: https://www.construct.net/en/make-games/manuals/construct-3/project-primitives/objects/instances

Object **instances** are the actual objects you see in a game: an *instance* of an object type. For example, if there are four *TrollEnemys* in a layout, those are four *instances* of the TrollEnemy *object type*.

It is **instances** which have a position, angle and size in the layout. Object types do not have these properties - they simply define a 'class' of object.

Instances can be created at runtime in events (typically by the System *Create object* action and the Sprite *Spawn an object* action). They can also be pre-arranged in layouts with the Layout View to design levels, menus and title screens. Instances can also be individually animated in timelines.

Selecting an instance in the Layout View shows its properties in the Properties Bar. These are a mix of properties in common with all (or most) objects, and plugin-specific properties. The common properties are described below, and plugin-specific properties are described for each plugin in the reference section.

Common instance properties

The following properties are common to most objects, depending on their capabilities.

Name

The name of the associated object type.

Global

By default, all instances are destroyed when the layout ends (e.g. when going to the next layout). If enabled, none of the instances of this object type will be destroyed when switching layouts.

Plugin Read-only

A reminder of the plugin this object is based on.

Position

The X and Y co-ordinates in the layout in pixels. This is measured to the object's origin. This can also be altered by moving the instance in the Layout View.

Size

The width and height of the instance in pixels. This can also be altered by dragging the resize handles in the Layout View.

Angle

The angle in degrees the instance is oriented at. This can also be altered by rotating the object in the Layout View by clicking and dragging just outside the resize handles.

Opacity

The instance opacity (or semitransparency), from 0% (transparent) to 100% (opaque).

Color

A color tint to apply to the instance. This works by normalizing each color component in the 0-1 range, and multiplying it with the object's color. This means a white color (with 1 for each color component) displays the original color of the object. Choosing another color will tint the object, e.g. choosing red will preserve only the red color component of the object's image.

Layer

The layer the instance is placed on. In the case the selected instance is from from a global layer in a different layout to the one currently active, the dropdown will show first the layers of the layout the instance is really coming from, followed by the layers of the layout which is currently active.

Z elevation

The instance's elevation on the Z axis. By default the camera is at Z = 100, and looking down to Z = 0. The default Z elevation is 0. Increasing it will move it upwards (towards the camera) and decreasing it will move it downwards (away from the camera).

Z elevation only affects the appearance of the object. It does not affect collisions - everything else continues to work in 2D as if its Z elevation was still 0.

Z elevation takes precedence over Z order. In other words, using Send to top of layer will not make an object appear on top of an object that has a higher Z elevation.

Z index Read-only

Indicates the zero-based Z index of the instance on its layer relative to all the other instances on the layer. A value of 0 means it is the bottom instance, and increasing values mean it is closer to the top of the layer. The Z index can be modified using the Z Order Bar Paid plans only.

UID Read-only

Every instance in the project has a unique number assigned, called its unique ID or UID. This value is displayed in the editor so you can view the UID for specific instances. You can use conditions like *Pick by unique ID* in events to pick specific instances by their UID.

It may be more convenient to use instance tags instead of UIDs, as they can be more descriptive.

Tags

A space separated list of string tags to identify an instance. The first tag of an instance is used in some places in the editor to better distinguish specific instances. There are associated common ACEs and a scripting interface to work with tags. Note that instance tags are case-insensitive.

Edit variables

Open the **Object Instance Variables dialog**.

Edit behaviors

Open the **Object Behaviors dialog**.

Edit effects

Open the **Effects dialog**.

Container

Group a set of object types together so they create, destroy and pick in events together. See the dedicated section on Containers for more information.

Template

A set of properties for managing templates, which allow conveniently updating properties of instances across the entire project. See the dedicated section on Templates for more information.

Visible in editor

An editor-only property, determines if the instance is visible or not. Can also be set through the Instances Bar.

Locked

An editor-only property, determines if the instance can be interacted with. Can also be controlled through the Instances Bar.

Index IDs (IIDs)

As well as unique IDs (UIDs, described above), all instances are also assigned an Index ID (IID). This is the zero-based index of the instance within its own object type. The first instance created for each object type is assigned an IID of 0, and subsequent instances are assigned incrementing numbers. Unlike UIDs, IIDs can change: if an instance is destroyed, all the object type's instance's IIDs are reassigned so they are continuous (i.e. 0, 1, 2, 3... N with no gaps). Therefore an IID does not persistently refer to one instance - use UIDs for that purpose. However IIDs can be useful for advanced users taking advantage of object expression indexing, the *Pick Nth instance* system condition, or the *IID* expression.