COMMON CONDITIONS

View online: https://www.construct.net/en/make-games/manuals/construct-3/plugin-reference/common-features/common-conditions

The following conditions are common to several plugins.

Angle

Note angles in Construct start with 0 degrees facing right and increment clockwise.

Is between angles

True if the object's current angle is between the two given angles in degrees. The first and second angles must be in clockwise order. That is, *Is between 0 and 45 degrees* is true if the object is in a 45 degree area, but *Is between 45 and 0 degrees* is true if the object is in the 315 degree area from 45 degrees through 0 degrees.

Is clockwise from

True if the object's current angle is clockwise from the given angle in degrees. Invert to test if the object is anticlockwise from the given angle. For example, an object at 45 degrees is clockwise from 0 degrees, but an object at 0 degrees is anticlockwise from 45 degrees.

Is within angle

True if the object's current angle is within a number of degrees of another angle. This is more reliable than testing if the object's angle exactly equals an angle, e.g. *Is within 0.5 degrees of 90 degrees* is probably better than *Angle equals 90 degrees*, since there are many cases an object can be very close to, but not exactly, 90 degrees.

Appearance

Compare opacity

Compare the object's current opacity, from 0 (transparent) to 100 (opaque).

Is effect enabled

Test if one of the effects added to the object is currently enabled.

Is visible

True if the object is currently visible. Invert to test if invisible. This only tests the visibility set by the *Set visible* action; it is not affected by the object being offscreen, having 0 opacity, or

being on an invisible layer.

Collisions

Is overlapping another object

Is overlapping at offset

True if any instance is overlapping any instance of another object. Collision polygons are taken in to account (if any), as well as the object's size and rotation. The *offset* variant will test for an overlap at an offset from the first object. For example, testing for an overlap at an offset of (100, 0) will temporarily move the object to the right 100 pixels, test for the overlap, then move it back again.

On collision with another object

Triggered upon the first time any instance starts overlapping any instance of another object. The collision polygons are taken in to account if set, as well as the object's size and rotation.

Hierarchy

These conditions are available for plugins that support the *scene graph* feature, allowing objects to be connected together so they move, rotate and scale as if they were one large object.

Compare child count

Compare the number of children that are currently attached to the object (using the *Add child* action).The *Which* option can be set to:

- *Own*: only the object's direct children are compared
- *All*: all children of the object are compared, including children-of-children, all the way to the bottom of the hierarchy

---

Has children

True if any children have been attached to the object (i.e. the child count is greater than 0).

Has parent

True if this object is currently attached to another object.

On hierarchy ready

Triggered for the root (top parent) instance in a hierarchy after *On created* has triggered for all instances in the hierarchy. For a specific instance in a hierarchy, when *On created* triggers, it is not certain that *On created* has yet triggered for other instances in the hierarchy, which

can complicate initializing hierarchies. This trigger gives you an opportunity to initialize a hierarchy once *On created* has been triggered for all instances in the hierarchy.

Pick children

Pick the children of a given object type or family attached to this object. The *Which* option can be set to:

- *Own*: only the object's direct children are picked
- *All*: all children of the object are picked, including children-of-children, all the way to the bottom of the hierarchy
- *Bottom*: only children at the bottom of the hierarchy from this object are picked, i.e. children with no further children of their own.

Pick nth child

Pick a specific child of this object at a given zero-based index. This only picks from the object's own children (children-of-children are excluded). When *Index type* is *all*, the index is in to the list of all children of any type. The object type or family of the child must still be specified; if it is the wrong type, it won't be picked even if there is a child at the given index. When *Index type* is *filtered*, the index is in to the list of children of the specified object type only. For example if a parent has three children of object types *Leg*, *Arm*, and *Leg*, and the condition is used to pick child *Leg* at index 1, index type *all* will not pick anything (as there is an *Arm* child there instead), whereas index type *filtered* will pick the second *Leg* instance (as that is index 1 in to all *Leg* children).

Pick parent

Pick the parent of a given object type or family this object is attached to. The *Which* option can be set to:

- *Own*: only the object's direct parent is picked
- *All*: all parents of the object are picked, all the way to the top of the hierarchy
- *Top*: only the parent at the top of the hierarchy from this object is picked, i.e. the parent not attached to any other parent.

HTML element

These conditions are available for some plugins in the *Form controls* category, like Button and Text Input. These objects are HTML elements placed on top of the canvas.

Is enabled

True if the element is currently enabled, and can be interacted with.

Is focused

True if the element is currently focused, meaning it will receive keyboard input. Typically this also involves a visual indication of focus as well.

Is visible

True if the element is currently visible. Otherwise the element still exists and preserves its contents, but is hidden.

Instance variables

Compare instance variable

Compare the current value of one of the object's number or text type instance variables.

Is boolean instance variable set

Test if one of the object's boolean instance variables is set to *true*. Invert the condition to test if false.

Pick highest/lowest

Pick the single instance with the highest or the lowest instance variable value of all the instances. Note this still only picks a single instance even if multiple instances have the same highest or lowest value; in this case an arbitrary instance is selected.

Misc

Has Tags

Check if the instance has all of the specified tags, provided by a space-separated string. Note that instance tags are case insensitive.

On created

On destroyed

Triggered for each instance that is created or destroyed during the running of the game. *On created* is also trigged for each object already on a layout when the layout starts. For example, a one-shot particle effect could be spawned every time an object is created, and an

explosion created every time the object is destroyed. These conditions are analogous to *constructors* and *destructors* in traditional programming languages (commands which run at the creation and destruction of an object). Be careful not to create an object of the same type in an *On created* event (e.g. *On Sprite2 created: create Sprite2*) since this will create an infinite loop and cause the game to hang.

On signal

Triggered when the *Signal* action is run with a matching tag (case insensitive).

Pick

Pick by unique ID

Pick the instance matching a given unique ID (UID) number.

Pick nearest/furthest

Pick the instance either nearest or furthest from a given position in the layout.

Size & Position

Compare width

Compare height

Compare the object's current size, in pixels.

Compare X

Compare Y

Compare the object's current position in the layout, in pixels. Note that objects can be positioned between pixels, e.g. at (5.5, 10.33333). Because of this it's usually a bad idea to rely on an object being at an exact position.

Is on-screen

True if any part of the object's bounding box is within the screen area. This is not affected by the object's visibility or opacity.

Is outside layout

True if the entire object's bounding box is outside the layout area.

Z Order

Compare Z elevation

Pick instances according to their elevation on the Z axis.

Is on layer

Pick all instances on a given layer, specified either by its name or zero-based index.

Pick top/bottom

Pick either the top-most or bottom-most instance, taking in to account layers and Z index. For example, the instance at the front of the top most layer is the top instance.