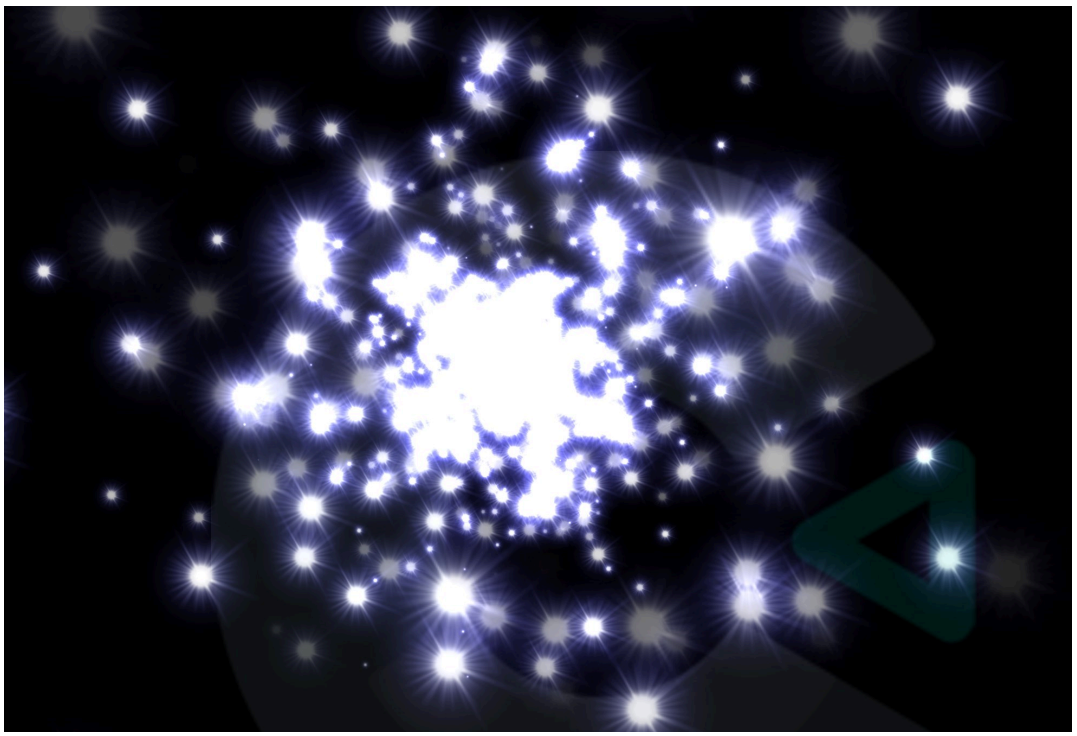
PARTICLES

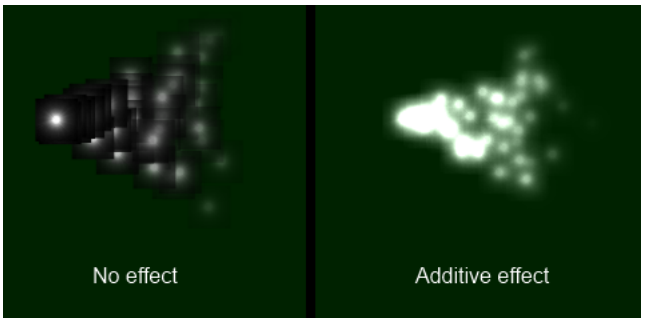
View online: https://www.construct.net/en/make-games/manuals/construct-3/plugin-reference/particles

The **Particles** object can quickly create visual effects by creating and moving many small images independently. It is a versatile object capable of many different kinds of visual effects. There are several examples in the Example Browser, ranging from fire to fountains; search for *Particles* to find them. The image below shows an example of one of the particle effects possible with the object.



The Particles object has many parameters to change the behavior of each particle. Also, it requires a texture used to draw each particle. Often a simple white spot on a black background is sufficient.

The **Additive** blend mode works especially well with the Particles object. It makes each particle brighten the background rather than pasting its image over the background, and allows particles to blend in to each other as well rather than simply overlapping. This makes particles look more like light sources. The below image shows what the effect does when the texture is a white spot on a black background.



Using Additive blend on particles

Colored effects can be created using colored particle images. Note that since the Additive effect brightens the background towards white, any objects using an Additive effect will not show up on a white background. The effect works best on dark backgrounds.

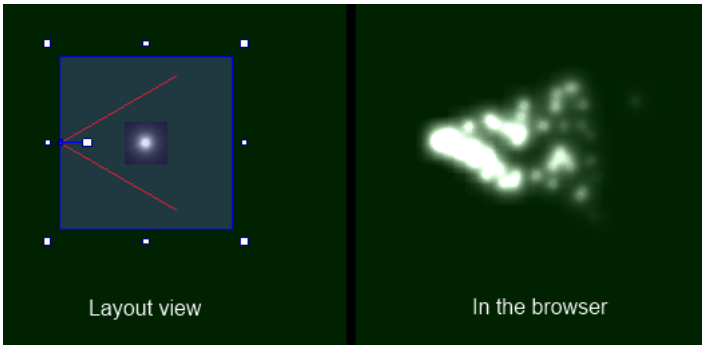
For more information about blend modes and effects, see the manual section on Effects.

Scripting

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

Particles in the Layout View

The Particles object is represented in the Layout View by two red lines which represent the spray cone (the angle through which particles are fired), with the particle texture in the middle. The Particle object's origin is where particles are created from. An example is shown below on the left, with the effect at runtime on the right.



The size of the particles object in the layout view is not important. The object will automatically size itself at runtime to fit all the particles it has created.

Previewing particle effects

Paid plans only Enable the **Preview** property to run the particle effect directly in the Layout View. As you edit the particle object's properties, the preview will update in real-time. This is a much faster way to get the exact effect you want compared to having to preview each time.

How particle effects work

The particle effect works similarly to using the Bullet behavior on each particle. Initially particles are fired forwards at a given speed and at an angle within the spray cone. Each particle is then individually controlled with different alterations to its speed, angle, opacity and size during its lifetime. The fact particles move independently of the others is often what makes the visual effect interesting. The various properties of the Particles object control exactly how the particles change over time and what random alterations are made. It is worth spending some time changing parameters to see the effect they have for yourself.

There are three different settings for when particles are destroyed, set by the **Destroy mode** property. The default *Fade to invisible* will fade each particle's opacity to zero over the *Timeout*, destroying the particle when it becomes invisible. *Timeout expired* will simply destroy the particle after an amount of time, without changing its opacity. *Particle stopped* will destroy the particle when its speed reaches zero, but you must take care to ensure particles slow down with a negative acceleration otherwise they will never be destroyed!

Advanced particle effects

By default, the Particles object draws particles using its own image. However you can specify an object instead by choosing another object, such as a Sprite, in the *Object* property. In this case the Particles object will then spawn and move instances of that object instead of drawing its own particles. This provides a similar visual result, but allows for much more flexibility. For example you can rotate the objects that are spawned, use behaviors and effects on them, or even test them for collisions.

> *Note that using objects for particles is slower than letting the Particles object draw its own particles.*

Optimisation

When the Particles object is drawing its own particles (i.e. not using an object), it is more efficient than creating the same effect with objects, but not by a large margin. Just like with sprites, you should be aware that creating a large number of particles can have a serious performance impact on your project. Use the **ParticleCount** expression to monitor how many particles are being created. Creating more than a few hundred particles may start to impact the framerate.

To reduce particle counts, try reducing the rate or shortening the timeout. To compensate, you can try making the particle size larger so the effect does not get thinner.

When using objects for particles, note that this is slower, and the performance can get worse if every object is different (e.g. using effects with different parameters) or mixed up in the Z order (e.g. other objects appear in between particles in the Z order). For best performance keep all the particle objects as similar as possible and ensure they aren't Z ordered individually.

Particle Properties

The Particles object has a relatively many properties, which are split in to three groups: particle spray properties (relating to the Particles object itself), initial particle properties (relating to the creation of each individual particle) and particle lifetime properties (relating to how particles behave after creation).

General

Rate

The number of particles created per second. If *Type* is *One-shot*, this is the total number of particles fired. Note that in *Continuous spray* mode, the overall particle count may be significantly more than the rate depending on the other properties. Also note that in *One-shot* mode, the rate can only be changed immediately after the object has been created; after the first tick, using the *Set rate* action will have no effect.

Spray cone

The number of degrees through which particles are fired. This is represented by the red lines in the Layout View. Use 360 to fire particles in all directions.

Type

The Particles object can work in two modes:

- **Continuous spray** will create a constant spray of particles (the default).
- **One-shot** will create a single blast of particles, the total number set by *Rate*. Once all particles have been destroyed, the Particles object then destroys itself. This is useful for one-off effects like explosions or impacts.

Image

Click to open the Animations editor to edit the particle image. Try a spot on a transparent background, or on a black background with the Additive effect. Note the image is not used if an object is set instead.

Object

Create instances of an object for each particle instead of drawing the particle image. For more information see the section *Advanced particle effects* above. Note this mode is slower than using a particle image.

> *To unselect an already chosen object, open the object picker, click Clear selection, and*

then click OK.

Initially visible

Set whether the object is shown (visible) or hidden (invisible) when the layout starts.

Preview Paid plans only

Enable to run a preview of the particle effect directly in the Layout View. You can change the other particle properties and see their effect in real-time.

Initial particle properties

Speed

The initial speed each particle is fired at, in pixels per second.

Size

The initial size of each particle, in pixels. Particles are always shown as squares, but the shape can be customised with the particle image.

Opacity

The initial opacity of each particle, from 0 (transparent) to 100 (opaque).

Grow rate

The initial grow rate (change in size over time) for each particle, in pixels per second. 0 means the particle will always stay the same size. A positive value will make particles grow, and a negative value will make particles shrink.

X randomiser

Y randomiser

The initial offset to the particle's position. You can make particles created along a line or in a box with these properties.

Speed randomiser

A random adjustment to each particle's initial speed on creation. For example, a value of 100 will change each particle's initial speed by up to 50 pixels per second faster or slower.

Size randomiser

A random adjustment to each particle's size on creation. For example, a value of 20 will change each particle's initial size by up to 10 pixels larger or smaller.

Grow rate randomiser

A random adjustment to each particle's grow rate on creation. For example, a value of 10 will change each particle's initial grow rate by up to 5 pixels per second greater or less.

Particle lifetime properties

Acceleration

Change in particle speed over time, in pixels per second per second. A positive value will make particles speed up, and a negative value will make them slow down.

Gravity

The acceleration downwards caused by gravity, in pixels per second per second. Useful for making fountain or other falling particle effects. Set to 0 to prevent gravity having any effect on particle movement.

Angle randomiser

A random change to each particle's angle to apply during its lifetime. For example, set to 0 to prevent particles ever changing direction, or set to 10 to allow particles to randomly change direction a little over time.

Speed randomiser

A random change to each particle's speed to apply during its lifetime. For example, set to 0 to prevent the speed changing, or set to 100 to allow particles to speed up or slow down somewhat over time.

Opacity randomiser

A random change to each particle's opacity to apply during its lifetime. Useful for creating "twinkling" effects.

Destroy mode

How each particle is destroyed. There are three modes available:

- **Fade to invisible** will fade each particle's opacity to zero over the *Timeout*. When the particle becomes invisible, it is destroyed.

- **Timeout expired** simply destroys each particle after the *Timeout* has expired, without altering the opacity.

- **Particle stopped** destroys each particle when its speed reaches zero. You must take care to use a negative *Acceleration*, or particles will never be destroyed!

Timeout

The time in seconds particles last for before being destroyed, depending on the *Destroy mode*.

Particle conditions, actions and expressions

Most of the Particle object's actions and expressions just set or get the above properties. See the above properties for a reference. The other conditions, actions and expressions not relating to the above properties are documented below.

For features in common to other objects, see Common features.

Particle conditions

Is spraying

True if the particle spray is currently enabled.

Particle actions

Set spraying

Enable or disable the spray, when in *Continuous spray* mode. When disabled, no new particles are created.

Fast-forward

Skip ahead the particle effect by a time in seconds. For example fast-forwarding by 3 seconds will cause the Particles object to instantly spawn, move and destroy particles as if 3 seconds had gone by. This is useful for making sure particle effects appear ready immediately, rather than taking a few seconds to move their particles out from the spawn point.

Particle expressions

ParticleCount

The number of particles the Particles object currently has. This is important to ensure you are not creating too many particles and slowing the project down; see the *Optimisation* section above. Note that due to the way Construct expressions work, if you have multiple

Particle object instances, this will only return the particle count for one of the instances - use a *For Each* loop to count multiple instance's total particle count.