9.6.3 Physics - Particles - Particle Physics

Particle Physics

Introduction

The movement of particles may be controlled in a multitude of ways. With particles physics: there are five different systems:

None (No Physics)
: It doesn’t give the particles any motion, which makes them belong to no physics system.

Newtonian
: Movement according to physical laws.

Keyed
: Dynamic or static particles where the (animated) targets are other particle systems.

Boids
: Particles with limited artificial intelligence, including behavior and rules programming, ideal for flocks of birds or schools of fishes, or predators vs preys simulations.

Fluid
: Movement according to fluid laws (based on Smoothed Particle Hydrodynamics technique).

Additional ways of moving particles:

- By softbody animation (only for Hair particle systems).
- By forcefields and along curves.
- By lattices.

Here we will discuss only the particle physics in the narrower sense, i.e. the settings in the Physics panel.

Velocity

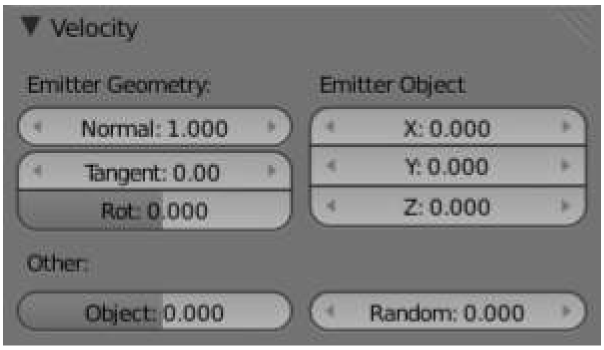


Image 3: Initial velocity.

The initial velocity of particles can be set through different parameters, based on the type of the particle system (see Particle System tab). If the particle system type is Emitter or Hair, then the following parameters give the particle an initial velocity in the direction of...

Emitter Geometry

Normal
: The emitter's surface normals (i.e. let the surface normal give the particle a starting speed).

Tangent
: Let the tangent speed give the particle a starting speed.

Rot
: Rotates the surface tangent.

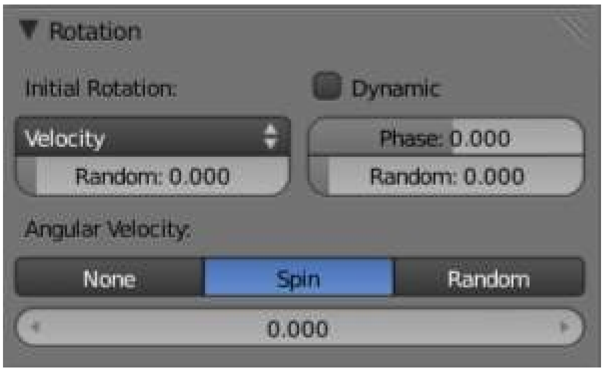
Emitter Object

Align X,Y,Z
: Give an initial velocity in the X, Y, and Z axes.

Object
: The emitter objects movement (i.e. let the object give the particle a starting speed).

Random
: Gives the starting speed a random variation. You can use a texture to only change the value, see Controlling Emission, Interaction and Time).

Rotation



Particles rotation settings.

These parameters specify how the individual particles are rotated during their travel. To visualize the rotation of a particle you should choose visualization type Axis in the Visualization panel and increase the Draw Size.

Initial Rotation Mode

Sets the initial rotation of the particle by aligning the x-axis in the direction of:

None
: the global x-axis.

Normal
: Orient to the emitter's surface normal, the objects Y axis points outwards.

Normal-Tangent
: As with normal, orient the Y axis to the surface normal. Also orient the X axis to the tangent for control over the objects rotation about the normal. requires UV coordinates, the UV rotation effects the objects orientation, currently uses the active UV layer. This allow deformation without the objects rotating in relation to their surface.

Velocity

the particle's initial velocity.

Global X/Global Y/Global Z

one of the global axes

Object X/Object Y/Object Z

one of the emitter object axes.

Random

Randomizes rotation.

Dynamic

If enabled, only initializes particles to the wanted rotation and angular velocity and let's physics handle the rest. Particles then change their angular velocity if they collide with other objects (like in the real world due to friction between the colliding surfaces). Otherwise the angular velocity is predetermined at all times (i.e. set rotation to dynamic/constant).

Phase

Initial rotation phase

Random

Rand allows a random variation of the Phase.

Angular Velocity

The magnitude of angular velocity, the dropdown specifies the axis of angular velocity to be

None

a zero vector (no rotation).

Spin

the particles velocity vector.

Random

a random vector.

If you use a Curve Guide and want the particles to follow the curve, you have to set Angular Velocity to Spin and leave the rotation on Constant (i.e. don't turn on Dynamic). Curve Follow does not work for particles.

Common Physics Settings

Size

Sets the size of the particles.

Random Size

Give the particles a random size variation.

Mass

Specify the mass of the particles.

Multiply mass with particle size

Causes larger particles to have larger masses.

No Physics

At first a Physics type that makes the particles do nothing could seem a bit strange, but it can be very useful at times. None physics make the particles stick to their emitter their whole life time. The initial velocities here are for example used to give a velocity to particles that are affected by a harmonic effector with this physics type when the effect of the effector ends.

Moreover, it can be very convenient to have particles at disposal (whose both Unborn and Died are visible on render) to groom vegetation and/or ecosystems using Object, Group or Billboard types of visualization.

Field Weights

The Field Weight Panel allows you to control how much influence each type of external force field, or effector, has on the particle system. Force fields are external forces that give dynamic systems motion. The force fields types are detailed on the *Force Field Page*.

Effector Group
: Limit effectors to a specified group. Only effectors in this group will have an effect on the current system.

Gravity
: Control how much the Global Gravity has an effect on the system.

All
: Scale all of the effector weights.

Force Fields

The Force Field Settings Panel allows you to make each individual act as a force field, allowing them to affect other dynamic systems, or even, each other.

Self Effect
: Causes the particle force fields to have an effect on other particles within the same system.

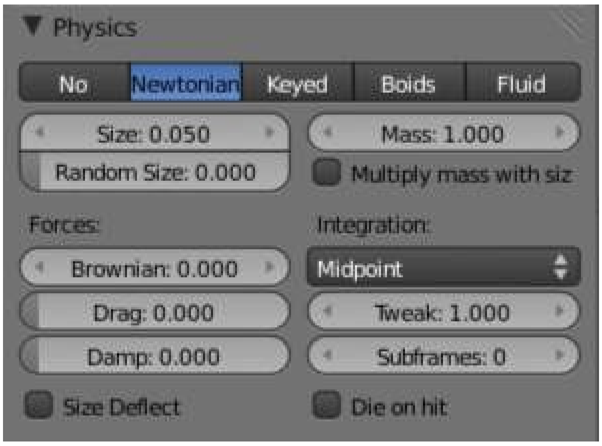
Amount
: Set how many of the particles act as force fields. 0 means all of them are effectors.

You can give particle systems up to 2 force fields. By default they do not have any. Choose an effector type from the dropdowns to enable them. Settings are described on the *Force Field Page*.

Newtonian Physics

These are the “normal” particle physics. Particles start their life with the specified initial velocities and angular velocities, and move according to Newtonian forces. The response to environment and to forces is computed differently, according to any given integrator chosen by the animator.

Forces



Newtonian Physics Settings.

Brownian

Specify the amount of Brownian motion. Brownian motion adds random motion to the particles based on a Brownian noise field. This is nice to simulate small, random wind forces.

Drag

A force that reduces particle velocity in relation to it's speed and size (useful in order to simulate Air-Drag or Water-Drag).

Damp

Reduces particle velocity (deceleration, friction, dampening).

Collision

Size Deflect

Use the particle size in deflections.

Die on Hit

Kill particle when it hits a deflector object.

Integration

Integrators are a set of mathematical methods available to calculate the movement of particles. The following guidelines will help to choose a proper integrator, according to the behavior aimed at by the animator.

Euler

Also known as "Forward Euler". Simplest integrator. Very fast but also with less exact results. If no dampening is used, particles get more and more energy over time. For example, bouncing particles will bounce higher and higher each time. Should not be confused with "Backward Euler" (not implemented) which has the opposite feature, energies decrease over time, even with no dampening. Use this integrator for short simulations or simulations with a lot of dampening where speedy calculations is more important than accuracy.

Varlet

Very fast and stable integrator, energy is conserved over time with very little numerical dissipation.

Midpoint

Also known as "2nd order Runge-Kutta". Slower than Euler but much more stable. If the acceleration is constant (no drag for example), it is energy conservative. It should be noted that in example of the bouncing particles, the particles might bounce higher than they started once in a while, but this is not a trend. This integrator is a generally good integrator for use in most cases.

RK4

Short for "4th order Runge-Kutta". Similar to Midpoint but slower and in most cases more accurate. It is energy conservative even if the acceleration is not constant. Only needed in complex simulations where Midpoint is found not to be accurate enough.

Frame Settings

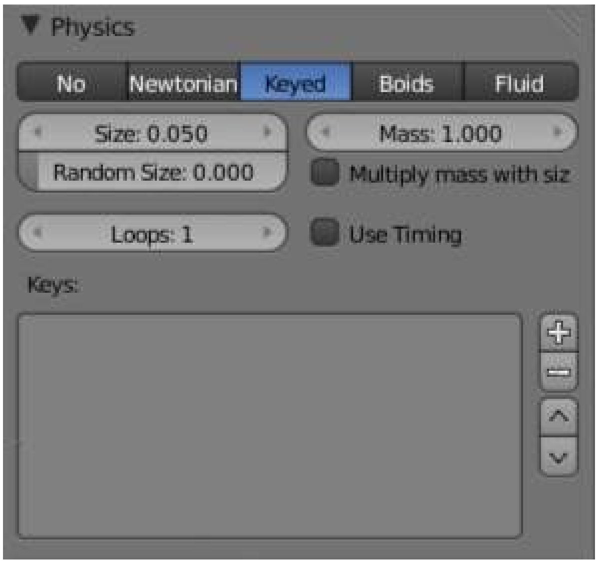
Timestep

The simulation time step per frame.

Subframes

Subframes to simulate for improved stability and finer granularity in simulations. Use higher values for faster moving particles.

Keyed Particles



Keyed Physics Settings.

The particle paths of keyed particles are determined from the emitter to another particle system's particles. This allows creation of chains of systems with keyed physics to create long strands or groovy moving particles. Basically the particles have no dynamics but are interpolated from one system to the next at drawtime.

Setup

To setup Keyed particles you need at least two particle systems.

The first system has keyed physics, and it needs the option First activated. This will be the system thats is visible.

- The second system may be another keyed system but without the option First,

or a normal particle system. This second system is the target of the keyed system.

Loops
    Sets the number of times the keys are looped. Disabled if *Use Timing* is enabled.

Keys

Key Targets

    You have to enter the name of the object which bears the target system and if there are multiple particle systems the number of the system.

    Click the `Plus` to add a key, then select the object.

If you use only one keyed system the particles will travel in their lifetime from the emitter to the target. A shorter lifetime means faster movement. If you have more than one keyed system in a chain, the lifetime will be split equally. This may lead to varying particle speeds between the targets.

Timing

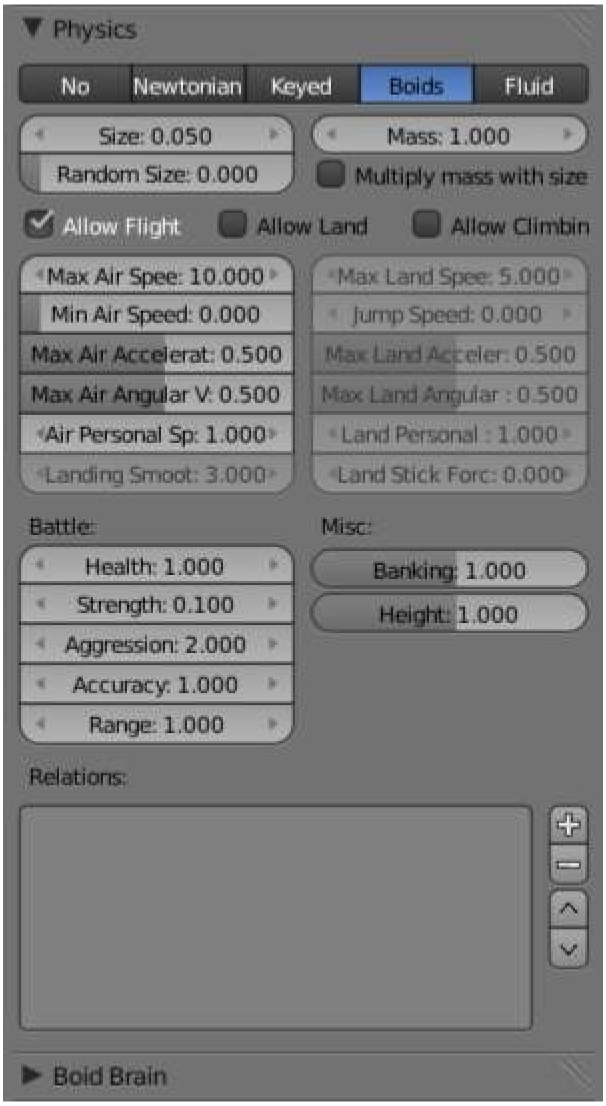
Use Timing

Timing works together with the Time slider for the other keyed systems in a chain. The Time slider allows to define a fraction of particle lifetime for particle movement.

An example: let’s assume that you have two keyed systems in a chain and a third system as target. The particle lifetime of the first system shall be 50 keys. The particles will travel in 25 frames from the first keyed system to the second, and in further 25 frames from the second system to the target. If you use the Timed button for the first system, the Time slider appears in the second systems panel. It’s default value is 0.5, so the time is equally split between the systems. If you set Time to 1, the movement from the first system to the second will get all the lifetime (the particles will die at the second system).

If you set Time to 0 the particles will start at the second system and travel to the target.

Boids



Boid Physics Settings.

Boids particle systems can be set to follow basic rules and behaviors. They are useful for simulating flocks, swarms, herds and schools of various kind of animals, insects and fishes. They can react on the presence of other objects and on the members of their own system. Boids can handle only a certain amount of information, therefore the sequence of the Behaviour settings is very important. In certain situations only the first three parameter are evaluated.

To view the subpanel to the right, add a *Particle System* of type *Emitter* and look in the middle area of the *Particle System* tab.

Physics

Boids try to avoid objects with activated Deflection. They try to reach objects with positive Spherical fields, and fly from objects with negative Spherical fields. The objects have to share one common layer to have effect. It is not necessary to render this common layer, so you may use invisible influences.

Boids can different physics depending on whether they are in the air, or on land (on collision object)

Allow Flight
: Allow boids to move in the air.

Allow Land
: Allow boids to move on land.

Allow Climbing
: Allow boids to climb goal objects.

Max Air Speed
: Set the Maximum velocity in the air.

Min Air Speed
: Set the Minimum velocity in the air.

Max Air Acceleration
: Lateral acceleration in air, percent of max velocity (turn). Defines how fast a boid is able to change direction.

Max Air Angular Velocity
: Tangential acceleration in air, percent 180 degrees. Defines how much the boid can suddenly accelerate in order to fulfill a rule.

Air Personal Space
: Radius of boids personal space in air. Percentage of particle size.

Landing Smoothness
: How smoothly the boids land.

Max Land Speed
: Set the Maximum velocity on land.

Jump Speed
: Maximum speed for jumping

Max Land Acceleration
: Lateral acceleration on land, percent of max velocity (turn). Defines how fast a boid is able to change direction.

Max Land Angular Velocity
: Tangential acceleration on land, percent 180 degrees. Defines how much the boid can suddenly accelerate in order to fulfill a rule.

Land Personal Space
: Radius of boids personal space on land. Percentage of particle size.

Land Stick Force
: How strong a force must be to start effecting a boid on land.

Banking

Amount of rotation around velocity vector on turns. Banking of (1.0 == natural banking).

Pitch
: Amount of rotation around side vector.

Height
: Boid height relative to particle size.

Battle

Health
: Initial boid health when born.

Strength
: Maximum caused damage per second on attack.

Aggression
: Boid will fight this times stronger than enemy.

Accuracy
: Accuracy of attack.

Range
: Maximum distance of which a boid can attack.

Alliance

The relations box allows you to set up other particle systems to react with the boids. Setting the type to *Enemy* will cause the systems to fight with each other. *Friend* will make the systems work together. *Neutral* will not cause them to align or fight with each other.

Deflectors and Effectors

As mentioned before, very much like Newtonian particles, Boids will react to the surrounding deflectors and fields, according to the needs of the animator:

Deflection: Boids will try to avoid deflector objects according to the Collision rule’s weight. It works best for convex surfaces (some work needed for concave surfaces). For boid physics, Spherical fields define the way the objects having the field are seen by others. So a negative Spherical field (on an object or a particle system) will be a predator to all other boids particle systems, and a positive field will be a goal to all other boids particle systems.

When you select an object with a particle system set on, you have in the Fields tab a little menu stating if the field should apply to the emitter object or to the particle system. You have to select the particle system name if you want prey particles to flew away from predator particles.

Spherical fields: These effectors could be predators (negative Strength) that boids try to avoid or targets (positive Strength) that boids try to reach according to the (respectively) Avoid and Goal rules’ weights. Spherical’s effective Strength is multiplied by the actual relevant weight (e.g. if either Strength or Goal is null, then a flock of boids won’t track a positive Spherical field). You can also activate Die on hit (Extras panel) so that a prey particle simply disappears when “attacked” by a predator particle which reaches it. To make this work, the predator particles have to have a spherical field with negative force, it is not sufficient just to set a positive goal for the prey particles (but you may set the predators force strength to -0.01). The size of the predators and the prey can be set with the Size button in the Extras panel.

Boid Brain

The Boid Brain panel controls how the boids particles will react with each other. The boids' behavior is controlled by a list of rules. Only a certain amount of information in the list can be evaluated. If the memory capacity is exceeded, the remaining rules are ignored.

The rules are by default parsed from top-list to bottom-list (thus giving explicit priorities), and the order can be modified using the little arrows buttons on the right side.

The list of rules available are:

Goal

Seek goal (objects with Spherical fields and positive Strength)

Predict
Predict target's movements

Avoid

Avoid "predators" (objects with Spherical fields and negative Strength)

Predict
Predict target's movements

Fear Factor
Avoid object if danger from it is above this threshold

Avoid Collision

Avoid objects with activated Deflection

Boids
Avoid collision with other boids

Deflectors
Avoid collision with deflector objects

Look Ahead
Time to look ahead in seconds

Separate
Boids move away from each other

Flock
Copy movements of neighboring boids, but avoid each other

Follow Leader

Follows a leader object instead of a boid

Distance
Distance behind leader to follow

Line
Follow the leader in a line

Average Speed

Maintain average velocity.

Speed
Percentage of maximum speed

Wander
How fast velocity's direction is randomized

Level

How much velocity’s Z component is kept constant

Fight

Move toward nearby boids

Fight Distance

Attack boids at a maximum of this distance

Flee Distance

Flee to this distance

Rule Evaluation

There are three ways control how rules are evaluated.

Average

All rules are averaged.

Random

A random rule is selected for each boid.

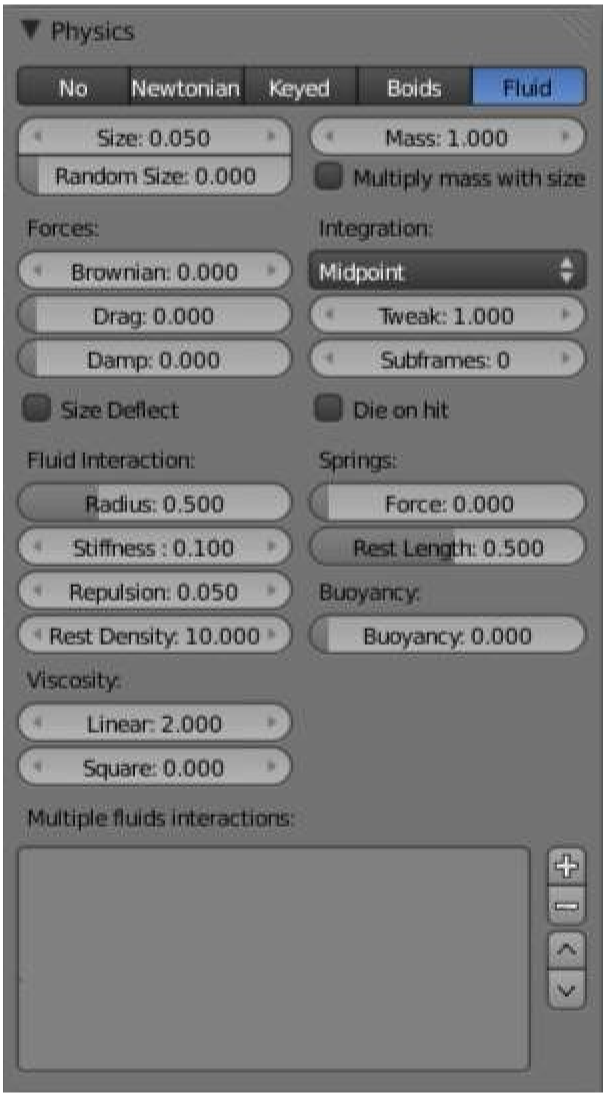
Fuzzy

Uses fuzzy logic to evaluate rules. Rules are gone through top to bottom. Only the first rule that effect above fuzziness threshold is evaluated. The value should be considered how hard the boid will try to respect a given rule (a value of 1.000 means the Boid will always stick to it, a value of 0.000 means it will never). If the boid meets more than one conflicting condition at the same time, it will try to fulfill all the rules according to the respective weight of each.

Please note that a given boid will try as much as it can to comply to each of the rules he is given, but it is more than likely that some rule will take precedence on other in some cases. For example, in order to avoid a predator, a boid could probably “forget” about Collision, Crowd and Center rules, meaning that “while panicked” it could well run into obstacles, for example, even if instructed not to, most of the time.

As a final note, the Collision algorithm is still not perfect and in research progress, so you can expect wrong behaviors at some occasion. It is worked on.

Fluid Physics



Fluid Physics Settings.

Fluid simulations are widely used in CG, and a very desired feature of any particle system, fluid particles are similar to newtonian ones but this time particles are influenced by internal forces like pressure, surface tension, viscosity, springs, etc. Blender particle fluids use the SPH techniques to solve the particles fluid equations.

Smoothed-particle hydrodynamics (SPH) is a computational method used for simulating fluid flows. It has been used in many fields of research, including astrophysics, ballistics, vulcanology, and oceanography. It is a mesh-free Lagrangian method (where the co-ordinates move with the fluid), and the resolution of the method can easily be adjusted with respect to variables such as the density.

From liquids to slime, goo to sand and wispy smoke the possibilities are endless.

Settings

Fluid physics share options with *Newtonian Physics*. These are covered on that page.

Fluid Properties

Stiffness
: How incompressible the fluid is.

Viscosity

Linear viscosity. Use lower viscosity for thicker fluids.

Buoyancy
Artificial buoyancy force in negative gravity direction based on pressure differences inside the fluid.

Advanced

Repulsion Factor
How strongly the fluid tries to keep from clustering (factor of stiffness). Check box sets repulsion as a factor of stiffness.

Stiff Viscosity
Creates viscosity for expanding fluid. Check box sets this to be a factor of normal viscosity.

Interaction Radius
Fluid’s interaction radius. Check box sets this to be a factor of 4*particle size.

Rest Density
Density of fluid when at rest. Check box sets this to be a factor of default density.

Springs

Force
Spring force

Rest Length
Rest length of springs. Factor of particle radius. Check box sets this to be a factor of 2*particle size.

Viscoelastic Springs
Use viscoelastic springs instead of Hooke’s springs.

Elastic Limit
How much the spring has to be stretched/compressed in order to change its rest length

Plasticity
How much the spring rest length can change after the elastic limit is crossed.

Initial Rest Length
Use initial length as spring rest length instead of 2*particle size.

Frames
Create springs for this number of frames since particle’s birth (0 is always).