# CROWDS

**User manual**

**Ver. Beta 0.65**

# 1 Plugins Types

The control system of Crowds system is made up of a series of different plugins, each of which has specific purposes:

**Crowds Generator:** this plugin allows you to create groups of agents.

**Crowds Obstacle:** allows you to define an object obstacle

**Crowds Behaviours:** contains all of the behaviors that are available for simulations.

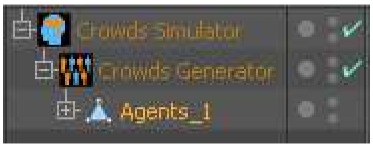
**Crowds Blender:** allows the mixing of the various behaviors.

**Crowds Simulator:** this solver is the end of the simulation, each Generator object must be son of this plugin to be active in the simulation

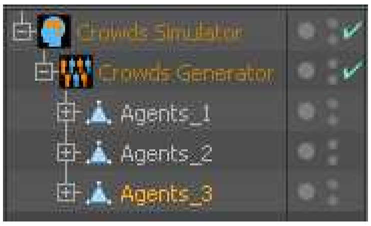
# 2 Crowds Generator

This plugin allows you to create groups of agents from one or more models of the crowd you want to generate. The basic model is to be inserted as a child of the same plugin, then the program will think themselves to clone it as many times as you want. It is currently not possible to differentiate, except for a few parameters as a group of agents, so to have a more heterogeneous multitude convine put multiple copies of the same agent and changed out of phase synchronization in the animation, so you have a larger group so realistic.

Each plugin of this type needs to be inserted as son of an object Simulator.
**All Generator plugin inserted as children of the same object Simulator belong to the same simulation and can interact with each other!**

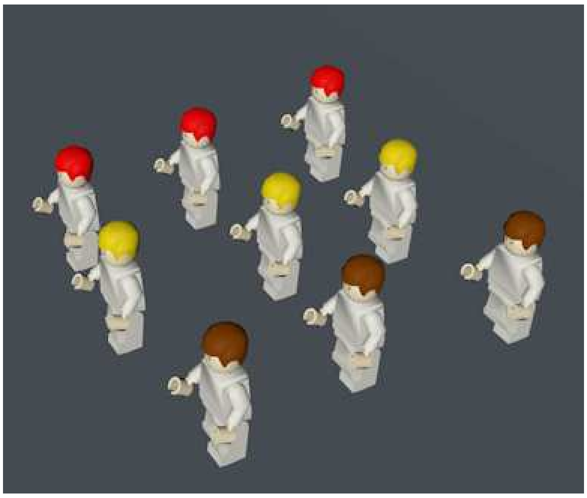


This image shows how a scene should be structured to create a group formed by type models Agents_1.



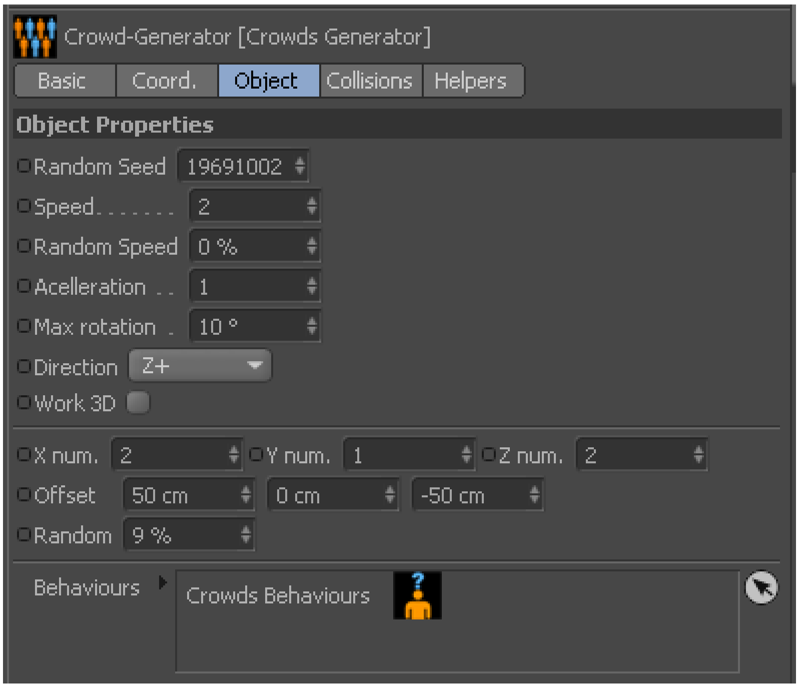
If the plugin has multiple child objects, then alternate each agent base to create a clone of the group. Currently, the program does not provide a randomization of agents children then their distribution is uniform.

The picture shows a group of three different agents.



## 2.1 Object

The main properties of the Crowd plugin-Generator can be found in the Object tab.



Object Properties

Random Seed 19691002
Speed. . . . . . . 2
Random Speed 0 %
Acelleration . . 1
Max rotation . 10 °
Direction Z+
Work 3D □

X num. 2 Y num. 1 Z num. 2
Offset 50 cm 0 cm -50 cm
Random 9 %

Behaviours Crowds Behaviours

**Random Seed**: This is the seed for the random numbers management group. It should be changed from generator to generator to obtain different distributions.

**Speed**: This parameter specifies the maximum speed of the agent.

**Random Speed**: is a percentage value that allows you to generate agents with different speeds. With 0% all agents have the same speed, with 100% agents speed will have a much different, but that will always have a maximum value that was added to Speed.

**Acelleration**: The agents starts at 0 speed and reach to cruise through the acelleration expressed by this parameter. This parameter identify the deceleration also, which is equal to three times its value.

**Max rotation**: It 's the rotation in degrees maximum that an agent can have in each frame. Very high values result in a jerky behavior during the simulation phase, while very low values generate movements very retarded.

**Direction**: Specify the direction in which to move the agents. It is possible to select Z + X + Z- and X-, Y-axis direction instead are not managed.

**Work 3d**: Allows you to define whether the agent will simply move on the two-dimensional plane ZX, or instead can move freely in three-dimensional space ZXY.

**X num.**: Defines the number of clones to be created on the X axis

**Y num.**: Defines the number of clones to be created on the Y axis

**Z num.**: Defines the number of clones to be created on the Z axis

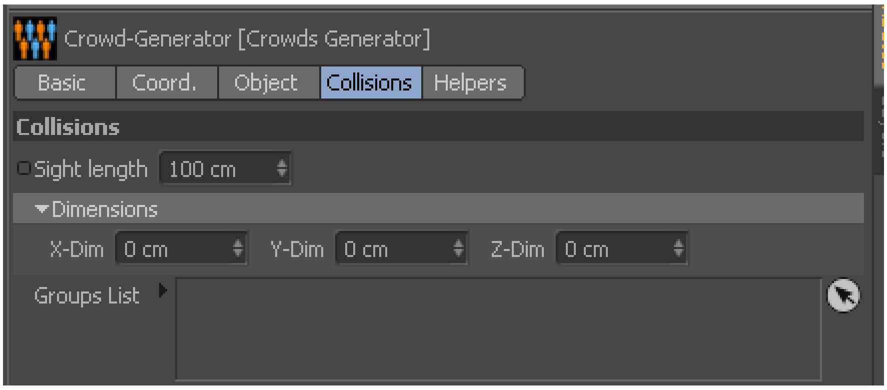
**Offset**: These three parameters are used to define an offset respectively for X, Y, Z to be used when creating the group.

**Random**: This parameter allows you to specify the value of randomness (from 0 to 100%) to be used for the initial formation of the group. With low values tend to have a rectangular distribution, while high values emphasize a chaotic distribution. At the moment there is no procedure managed to prevent the overlapping of agents then very chaotic distributions also require abundant offset values.

**Behaviours**: This box must be inserted only objects of type Behaviours or Blender. Basically it is a list of behaviors that govern the simulation of the group. These behaviors will be developed for each frame of animation, starting from the top of the list until you reach the top down.

## *2.2 Collisions*

In this section you can define basic  parameters in the management of collisions.



Crowd-Generator [Crowds Generator]

Collisions

Sight length 100 cm

Dimensions

X-Dim 0 cm Y-Dim 0 cm Z-Dim 0 cm

Groups List

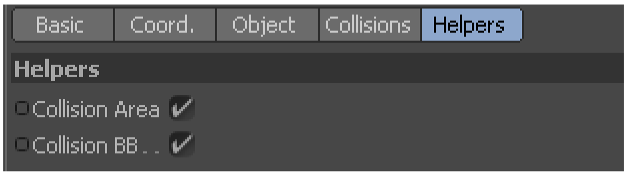
**Sight length**: Specifies the length of the vector view of the agent. This vector can be thought of as the stick of the blind man, is the main tool that the agent has available to deal with the outside world.

**Dimensions**: The three values of X-Dim, Dim Y-, Z-Dim, allow you to define an ellipse containing the agent. This ellipse is used to define the space occupied by the same agent, it is very important especially when working with the verification of collisions between agents (see Collision Avoidance Behaviours plugin).

**Group List**: This list can include other objects of type Generator. This list is used to identify groups of simulation that you want to occur when in the  “Collision avoidance” panel is selected the verification between groups. Although the simulation contain many other groups, the program will be verified only ones on this list.

## *2.3 Helpers*

This box contains parameters that are used to activate functions in order to understand how they have been set some variables. These parameters are for informational purposes only and do not alter in any way the simulation results.



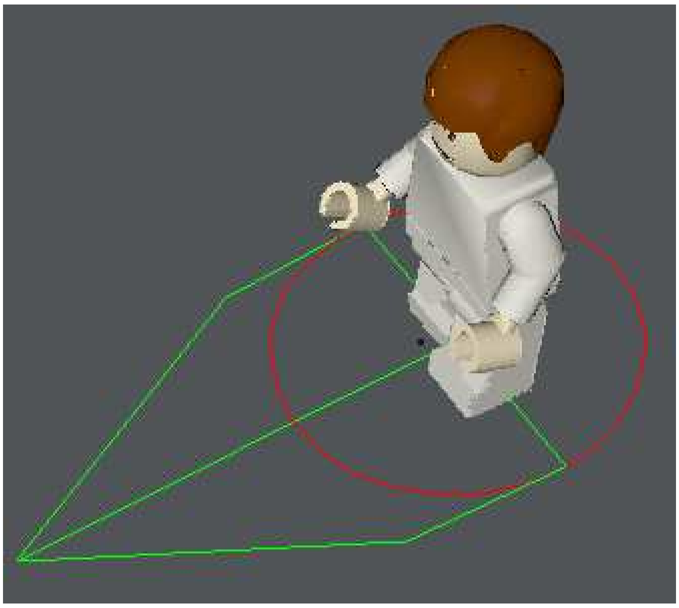
Helpers

○ Collision Area ☑

○ Collision BB . . ☑

**Collision Area**: se attivato il programma visualizza la zona di collisione definita per gli agenti. Questa zona ha un senso solo se ci si trova in un gruppo che lavora sul piano (parametro Work 3d = False), in caso contrario il programma utilizzerà esclusivamente il vettore definito dal parametro Sight length per gestire le collisioni.
Comunque la forma di questa zona è una semiellisse, con gli assi definiti dalla lunghezza del parametro **Sight length** e dal parametro **Dimension** (utilizzando tra le dimensioni X e Z, quella opposta alla direzione selezionata per gli agenti).
Per velocizzare la visualizzazione il programma invece di disegnare una semi ellisse perfetta ne mostra una stilizzata.

**Collision BB**: Visualizza l’elissoide generato dai valori del parametro **Dimesion**, che permettono di specificare l’ingombro dell’agente.

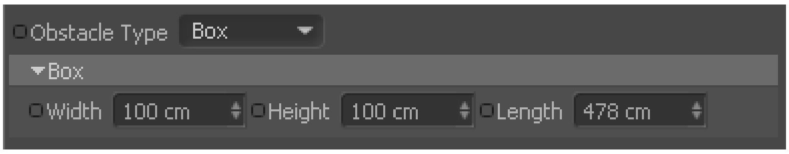


Esempio di visualizzazione attiva, con **Collision area** in verde e **Collision BB** in rosso.

# 3 Crowds Obstacle

This plugin has the task of defining the obstacles to simulation. For obstacle we mean an area belonging to the space of the simulation that can not be occupied by the agents.
The program allows you to manage three different types of obstacles: Box, Sphere and Mesh.

## *3.1 Box obstacle type*

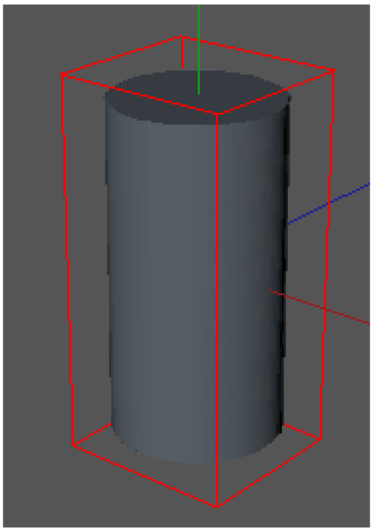


○ Obstacle Type Box

Box

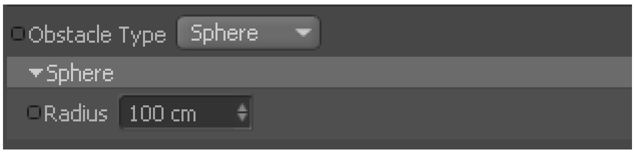
○ Width 100 cm ○ Height 100 cm ○ Length 478 cm

With this type you create a box the size defined by the parameters Width, Height and Length. The created object can then be moved to build on areas that should be avoided by the agents.
One important thing is that the object obstacle can be moved and rotated, but the scale of operation should be avoided because the size must be defined only by the parameters of the object.



In this picture you can see an object of type BOX obstacle that approximates a cylinder.

## 3.2 Sphere obstacle type



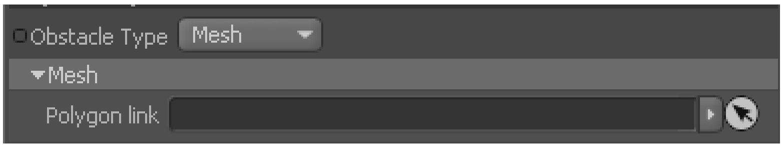
Obstacle Type Sphere

Sphere

Radius 100 cm

This type of obstacle is a spherical type and its dimensions are defined by the Radius parameter.

## 3.3 Mesh obstacle type



Obstacle Type Mesh

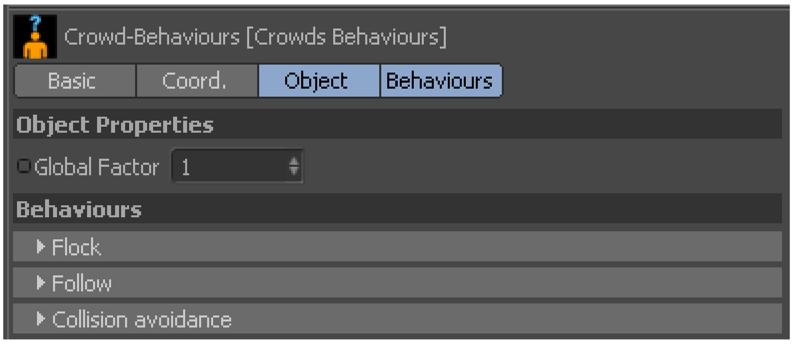
Mesh

Polygon link

This typology allows to use any polygonal object of Cinema4D as an obstacle.
The object is specified in the box **Polygon link**.
Unlike the other two types, this requires a longer processing, then you should use it sparingly, and still try to use simple objects (composed of few polygons) and maybe that approximate the real object that should be avoided by agents.

# 4 Crowds Behaviours

This plugin contains all the behaviors that each agent can use to run a simulation.
Every behavior has a parameter of type Factor that determines the intensity. It 's very important to understand the meaning of the parameter because it is the main factor responsible for how they mixed the various behaviors within the program.
In practice all the behaviors enabled within a single instance Behaviours will be mixed with one another via the corresponding parameter Factor, then those with Factor largest will have greater significance than those with Factor lowest. The same thing happens then between the various instances of Behaviours included in the list of the behaviors of an object Generator, indeed any resulting final is blended with others via the **Global parameter Factor**, visible in the Object folder.



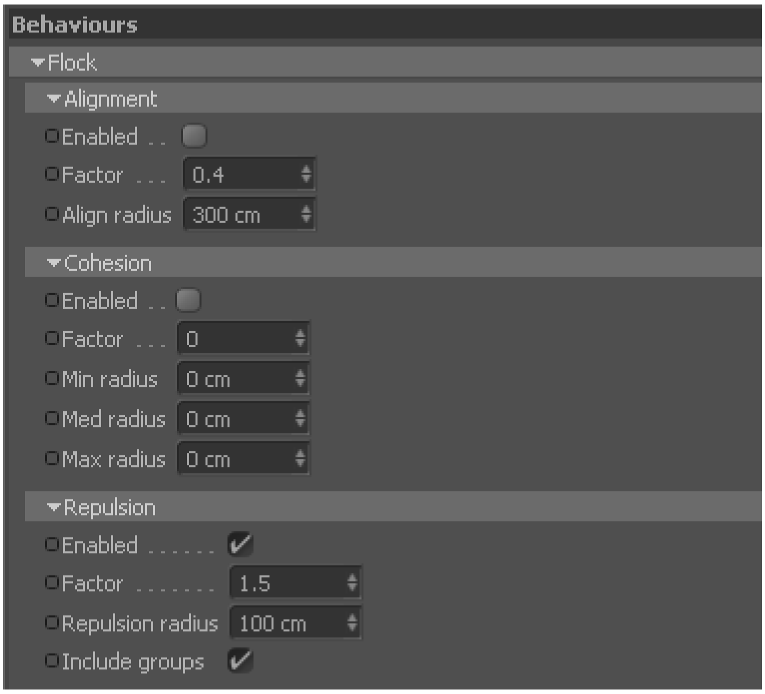
Crowd-Behaviours [Crowds Behaviours]

Object Properties

○ Global Factor 1

Behaviours

## *4.1 Flock behaviours*



Behaviours

Flock

Alignment

Enabled . . □

Factor . . . 0.4

Align radius 300 cm

Cohesion

Enabled . . □

Factor . . . 0

Min radius 0 cm

Med radius 0 cm

Max radius 0 cm

Repulsion

Enabled . . . . . . ☑

Factor . . . . . . . 1.5

Repulsion radius 100 cm

Include groups ☑

The Flock Behaviours, contains three actions: Alignment, Cohesion and Repulsion.

### 4.1.1 Alignment

This behavior serves to find a direction vector average between the directions of all the other agents of a group. Ultimately allows to maintain the orderly progress of a group.

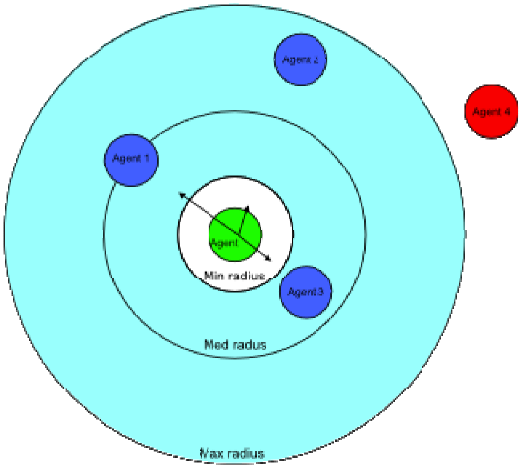
**Enabled** Enables the use of behavior

**Factor** determines the relative importance of the behavior

**Align radius** specifying a value greater than zero, the behavior is verified only for agents that are within a distance equal to the radius indicated. With a radius equal to zero instead will be considered all agents.

### 4.1.2 Cohesion

This behavior has the purpose of maintaining a cohesive group of agents and at a certain distance from each other, so as to prevent that these are dispersed within the simulation space.
This procedure takes into account only the agents that are located within an imaginary circular surface of radius equal to **Max radius-Min radius**, outside of these extremes the relative vector is considered invalid. Among all agents that meet this requirement the procedure will give the greatest importance to those who stannao perfectly on the line of circumference of radius **Med radius**, moving away from this circumference their weight decreases in a linear manner up to zero.



*In this picture you can see how it affects this behavior. The agent 1 will determine the vector of greater importance for the orientation of the new direction with respect to agents 2 and 3, the agent 4 (in red) will not be considered because instead outside the surface of verification (blue color).*

**Enabled** Enables the use of behavior

**Factor** determines the relative importance of the behavior

**Min radius** if the distance between the two agents compared is less than this value then their component is ignored by the calculation procedure.

**Med radius** the distance ideal to maintain, between an agent and the group.

**Max radius** represents the maximum distance over which compose the orientation vector is ignored.

### 4.1.3 Repulsion

Allows to remove an agent from all those that are located within a radius arbitrary.

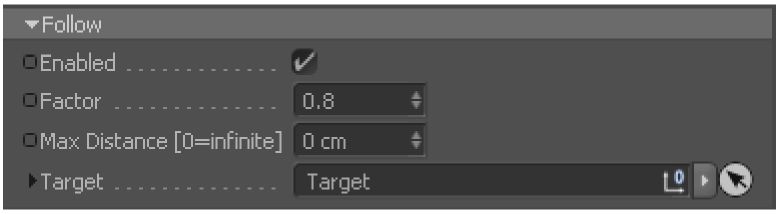
**Enabled** Enables the use of behavior

**Factor** determines the relative importance of the behavior

**Repulsion radius**: determines the range within which an agent is considered too close.

**Includes groups** enabling this box, the calculation of repulsion will be extended to all agents of the groups in the group list.

## 4.2 Follow



Follow

Enabled ☑

Factor 0.8

Max Distance [0=infinite] 0 cm

Target Target

This behavior allows you to select a target that agents should try to achieve.

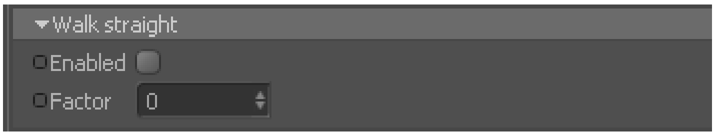
**Enabled** enables the use of behavior

**Factor** determines the relative importance of the behavior

**Max Distance** expresses the maximum distance within which the target must be located in order to be taken into consideration. If value is null then the target distance is not taken into consideration.

**Target** is the box that lets you specify the target to consider.

## 4.3 Walk straight



Walk straight

Enabled □

Factor 0

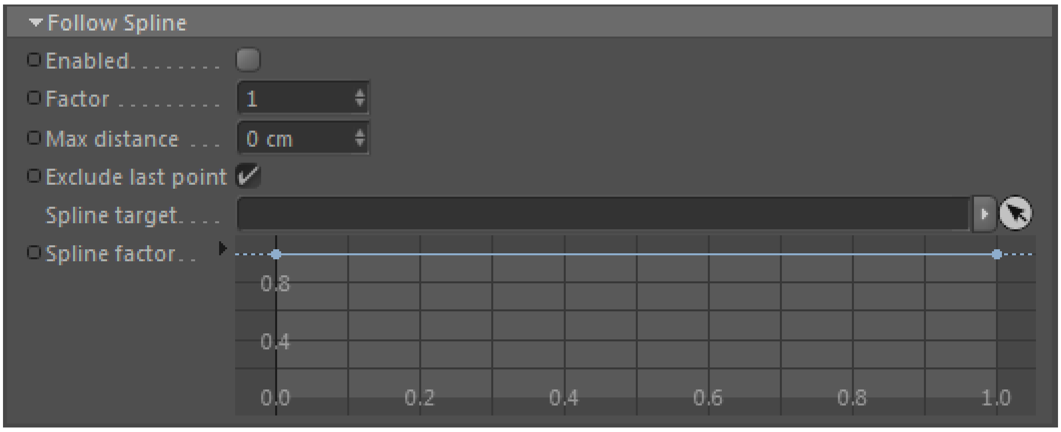
This behavior allows simply to do the agents move in the direction that he is facing.

**Enabled** Enables the use of behavior

**Factor** determines the relative importance of the behavior

## 4.4 Follow spline

This behavior allows you to use spline as a guide for agents. When the distance from the spline used is less than that specified in the threshold level, then the agent begins to move towards this.



Follow Spline

Enabled ☐

Factor 1

Max distance 0 cm

Exclude last point ☑

Spline target

Spline factor

**Enabled** toggles the use of behavior

**Factor** determines the relative importance of the behavior

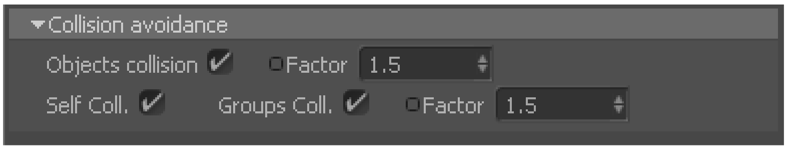
**Max distance** maximum distance beyond which the spline is no longer considered

**Exclude last point** allows to exclude the last point of the spline in order to avoid that the agent returns on his steps once he reaches the end of the segment. The same thing can be done in a more gradual rate using the Spline factor.

**Spline target** allows you to specify the spline object to use. The program also allow to operates with discontinuous spline, to create “rails” where agents can be aligned.

**Spline factor** permits modifying the relevance factor (factor parameter) at the sections of the spline used. The value factor is multiplied by the factor on the spline. For example, by setting to zero the value of the spline factor at the end of the spline, is obtained by a gradual neglect of the agents to follow the same spline.

## 4.5 Collision avoidance



▾Collision avoidance

Objects collision ☑ □Factor 1.5

Self Coll. ☑ Groups Coll. ☑ □Factor 1.5

This behavior allows you to enable the verification of collisions. In the process will include all items included in the list of plugins Obstacles Simulator.
The program also allows you to enable the control of collisions between agents of the same group, or even with all the agents of the groups specified in the list of plugins Generator Group list. This type of collision control is very different from that obtainable through the behavior Repulsion, and can only be used if you working with a two-dimensional simulation.

**Object collison** active collisions checking agent / object

**Factor** determines the relative importance of the behavior Object collision.

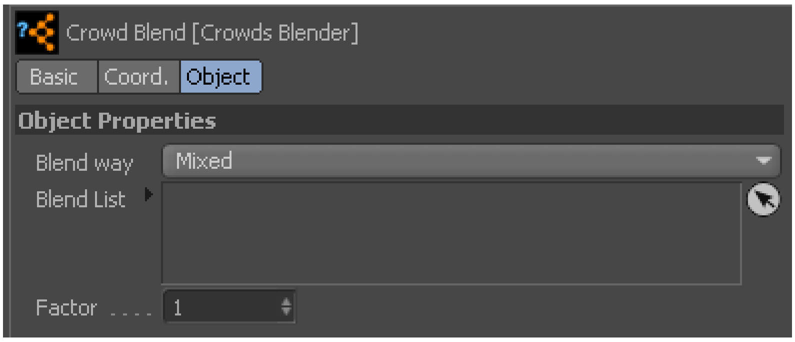
**Self Coll.** enable verification of collisions between agents in the group. **Can be used only in a two-dimensional simulation**.

**Groups Coll.** enable verification of collisions between agents in the group and those of all the groups included in the list object Generator Group list. **Can be used only in a two-dimensional simulation**.

**Factor** determines the relative importance of the behavior Self coll.. and Group coll.

# 5 Crowds Blender

This plugin is intended to mix, using various criteria, the result obtained from instances of type **Behaviours** or **Blender**.



Crowd Blend [Crowds Blender]

Object Properties

Blend way: Mixed

Blend List

Factor . . . . 1

**Blend way** Allows you to specify the policy of mixing. There are three different types: Mixed, First applicable, Strongest.

The **Mixed modality** make an average of the results of all instances and Behaviour Blender on the list.

**First mode applicable** modality instead stop the procedure of checking with the first result different from zero, namely the first behavior that returns a value of direction valid.

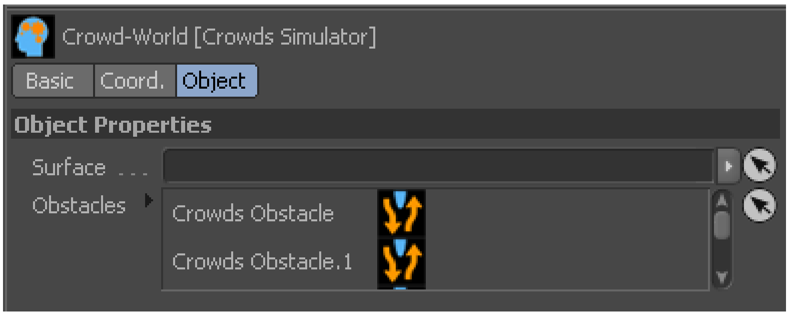
**Strongest modality** instead calculates all the components relating to instances in the list and then select only the one that returns the value with Factor greater.

**Blend List** This is the list that contains instances of the type or types Behaviour Blender. These are always verified in succession, starting from the first at the top.
As it is structured this plugin you could enter in a list instance that already contains the list itself, in this case to prevent an infinite loop, the program will proceed to ignore it in the testing phase.

**Factor** determines the relative importance of the behavior

# 6 Crowds Simulator

It is the plugin that is committed to solving the simulation set. All groups must interact with each other, must be children of the same plugin simulation.



Crowd-World [Crowds Simulator]

Object Properties

Surface . . .

Obstacles: Crowds Obstacle, Crowds Obstacle.1

**Surface** allows you to specify the area in which the agents belonging to the simulation will be placed. Currently, the program uses the pivot of the parent agent as a reference for the agent to stick the ground.
This function is active only for agents who belong to groups that are working two-dimensionally (3D Work = False)

**Obstacles** should be included in this list all objects Obstacles that must be verified in the process of obstacle avoidance.