orxSPAWNER structure

Summary

```
[SpawnerTemplate]
Object           = ObjectTemplate
WaveSize         = <int>
WaveDelay        = <float>
ActiveObject     = <int>
TotalObject      = <int>
AutoReset        = <bool>
CleanOnDelete    = <bool>
ObjectSpeed      = <vector>
Position         = <vector>
Rotation         = <float>
Scale            = <vector> | <float>
Interpolate      = <bool>
Immediate        = <bool>
UseAlpha         = <bool>
UseColor         = <bool>
UseRotation      = <bool>
UseScale         = <bool>
UseRelativeSpeed = <bool>
UseSelfAsParent  = <bool>
IgnoreFromParent =
none|all|rotation|scale|position|position.rotation|position.scale|position.p
osition
```

Details

Here's a list of the available properties for an `orxSPAWNER` structure:

- `Object`: Defines the name of the object (ie. its config section name) that will be spawned.
- `WaveSize`: Defines the size of a wave (ie. number of objects spawned in a single wave). Its default value is 0 which means the automated wave spawning system isn't used. The wave system only works for spawners that have been created and therefore owned by an orxOBJECT.
- `WaveDelay`: Time ellapsed (in seconds) between two waves. Its default value is 0, which means there are no delay between two waves (ie. continuous spawning). If a negative value is provided, the wave system is deactivated. The wave system only works for spawners that have been created and therefore owned by an orxOBJECT.
- `ActiveObject`: Number of allowed active object at the same time. When this limit is reached the spawner can't spawn objects anymore till some of the spawned objects are deleted. By default, its value is 0 which means there's no limit.
- `TotalObject`: Number of total objects the spawner is allowed to create. When this limit is reached the spawner will deactivate itself and send an event [1]. By default, its value is 0 which means there's no limit.
- `AutoReset`: If this is set to `true`, when the spawner reaches its total limit, it'll be reset instead of deactivated. An event [2] will be sent to notify the user. After being reset, the spawner will

continue spawning as if it was brand new with the current active and total object counters back to 0. By default this property is set to `false`.
- `CleanOnDelete`: If this is set to `true` all the spawned objects still leaving will be deleted as soon as the spawner gets deleted. Defaults to `false`.
- `ObjectSpeed`: If this property is defined, it'll give an initial speed to every spawned object that will override any speed setting of the object itself. By default, there's no speed override.
- `Position`: Position of the spawner in its parent space, in pixels. Defaults to (0, 0, 0).
- `Rotation`: Rotation of the spawner in its parent space, in degrees. Defaults to 0°.
- `Scale`: Scale of the spawner in its parent space (Z component being ignored). By default its value is 1.0. [3)]
- `Interpolate`: When interpolate is set to true, sub-frame precision spawning will be emulated if needed. Defaults to false.
- `Immediate`: If set, the delay will be ignored for the first wave after creation or reset; Defaults to false.
- `UseAlpha`: This only works if the spawner is linked to a parent orxOBJECT. If so and its value is set to `true`, it will transmit the parent's alpha to any spawned objects. By default its value is `false`.
- `UseColor`: This only works if the spawner is linked to a parent orxOBJECT. If so and its value is set to `true`, it will transmit the parent's color to any spawned objects. By default its value is `false`.
- `UseRotation`: This only works if the spawner is linked to a parent orxOBJECT. If so and its value is set to `true`, it will transmit the parent's rotation to any spawned objects. By default its value is `true`.
- `UseScale`: This only works if the spawner is linked to a parent orxOBJECT. If so and its value is set to `true`, it will transmit the parent's scale to any spawned objects. By default its value is `true`.
- `UseRelativeSpeed`: If set to `true` and the spawner has a value for `ObjectSpeed`, the applied speed will be relative to the spawner's current rotation and scale. By default its value is `false`.
- `UseSelfAsParent`: If set to `true`, all spawned object will have the spawner as parent (in the orxFRAME hierarchy sense). This means that every spawned object will move together with its spawner parent: it will adapt to its position, rotation and scale. If you only want a spawned object to adopt the current orientation of the Spawner, see ObjectSpeed and UseRotation. By default this property's value is set to `false`.
- `IgnoreFromParent`: none|all|rotation|scale|position|position.rotation|position.scale|position.position; NB: Defines which parts of the parent's transformation will be ignored when transmitted to the spawner. Defaults to none.

Latest config settings for the Development Version

We endeavor to keep the config properties on this page up to date as often as possible. For up to the minute config information for the latest version of Orx, check the most recent published at:

CreationTemplate.ini and

SettingsTemplate.ini

Additionally these files can be found under your orx source tree in the `orx/code/bin` folder.

[1)]

Type = orxEVENT_TYPE_SPAWNER, ID = orxSPAWNER_EVENT_EMPTY

[2)]

Type = orxEVENT_TYPE_SPAWNER, ID = orxSPAWNER_EVENT_RESET

[3)]

ie. (1.0, 1.0, 0.0)

From:
https://orx-project.org/wiki/ - **Orx Learning**

Permanent link:
https://orx-project.org/wiki/en/orx/config/settings_structure/orxspawner?rev=1533738008

Last update: **2025/09/30 17:26 (14 months ago)**