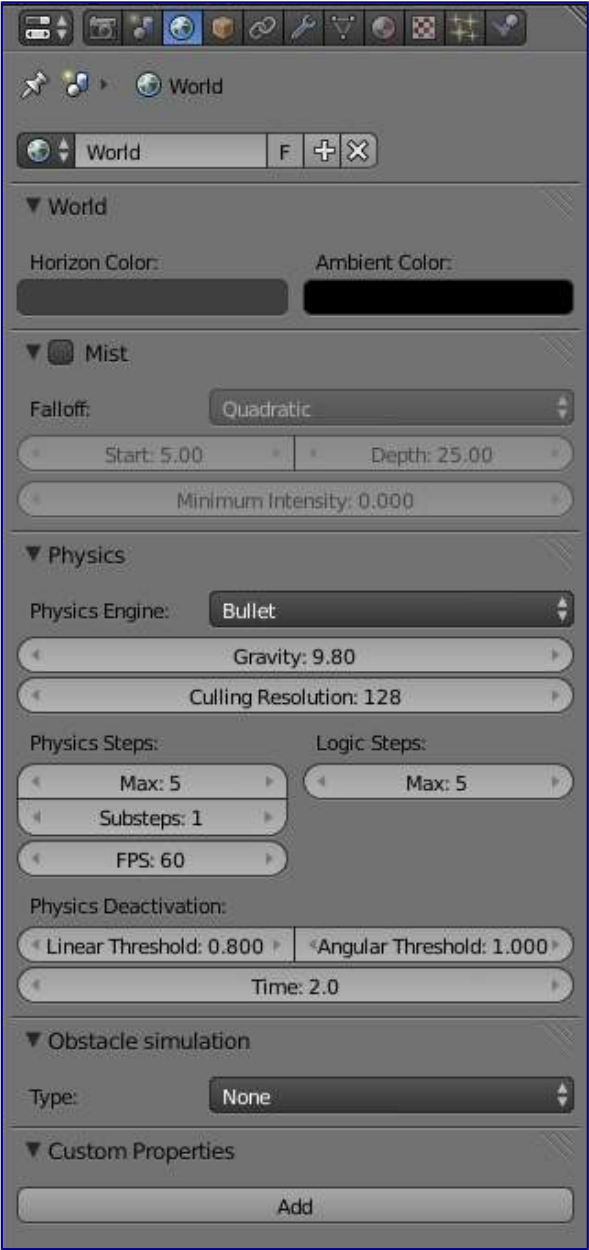


## 12.6.2 Game Engine - Physics - World Physics

World Physics.....	1
World.....	2
Mist.....	2
Game Physics.....	2
Obstacle Simulation.....	3

### World Physics



BGE World Panel (fully expanded)

World settings enable you to set some basic effects which affect all scenes throughout your game, so giving it a feeling of unity and continuity. These include ambient light, depth effects (mist) and global physics settings.

These effects are a limited subset of the more extensive range of effects available with the Blender internal or cycles renderer.

### Tip

While world settings offer a simple way of adding effects to a scene, *compositing nodes* are often preferred, though more complex to master, for the additional control and options they offer. For example, filtering the Z value (distance from camera) or normals (direction of surfaces) through compositing nodes can further increase the depth and spacial clarity of a scene.

## World

These two color settings allow you to set some general lighting effects for your game.

### Horizon Color

The RGB color at the horizon; i.e. the color and intensity of any areas in the scene which are not filled explicitly.

### Ambient Color

Ambient light mimics an overall background illumination obtained from diffusing surfaces (see *Ambient Light*, *Exposure* and *Ambient Occlusion*). Its general color and intensity are set by these controls.

## Mist

Mist can greatly enhance the illusion of depth in your rendering. To create mist, Blender makes objects farther away more transparent (decreasing their Alpha value) so that they mix more of the background color with the object color. With Mist enabled, the further the object is away from the camera the less it's alpha value will be. For full details, see *Mist*.

### Mist

Toggles mist on and off

### Falloff

Sets the shape of the falloff of the mist.

### Start

The starting distance of the mist effect. No misting will take place for objects closer than this distance.

### Depth

The depth at which the opacity of objects falls to zero.

### Minimum intensity

Overall minimum intensity of the mist

## Game Physics

The Game Physics located in the World panel determine the type of physical rules that govern the game engine scene, and the gravity value to be used. Based on the physics engine selected, in physics simulations in the game engine, Blender will automatically move *Actors* in the downward (-Z) direction. After you arrange the actors and they move as you wish, you can then bake this computed motion into keyframes (see *Digging Deeper* for more info).

## Physics Engine

Set the type of physics engine to use.

### Bullet

The default physics engine, in active development. It handles movement and collision detection. The things that collide transfer momentum to the collided object.

### None

No physics in use. Things are not affected by gravity and can fly about in a virtual space. Objects in motion stay in that motion.

### Gravity

The gravitational acceleration, in units of meters per squared second ( $m \cdot s^{-2}$ ), of this world. Each object that is an actor has a mass and size slider. In conjunction with the frame rate (see *Render* section), Blender uses this info to calculate how fast the object should accelerate downward.

### Culling Resolution

The size of the occlusion culling buffer in pixel, use higher value for better precision (slower). The optimized Bullet DBVT for view frustum and occlusion culling is activated internally by default.

### Physics Steps

#### Max

Sets the maximum number of physics steps per game frame if graphics slow down the game. higher value allows physics to keep up with realtime.

#### Substeps

Sets the number of simulation substeps per physics timestep. Higher value give better physics precision.

#### FPS

Set the nominal number of game frames per second. Physics fixed timestep =  $1/\text{fps}$ , independently of actual frame rate.

### Logic Steps

Sets the maximum number of logic frame per game frame if graphics slows down the game, higher value allows better synchronization with physics.

### Physics Deactivation

These settings control the threshold at which physics is deactivated. These settings help reducing the processing spent on Physics simulation during the game.

#### Linear Threshold

The speed limit under which a rigid bodies will go to sleep (stop moving) if it stays below the limits for a time equal or longer than the deactivation time (sleeping is disabled when deactivation time is set to 0).

#### Angular Threshold

Same as linear threshold, but for rotation limit (in rad/s)

#### Time

The amount of time in which the object must have motion below the thresholds for physics to be disabled (0.0 disables physics deactivation).

## Obstacle Simulation

Simulation used for obstacle avoidance in the Game Engine, based on the RVO (Reciprocal Velocity Obstacles) principle. The aim is to prevent one or more actors colliding with obstacles.

See Path finding and steering behaviors for more details.

## **Type**

### **None**

obstacle simulation is disabled, actors aren't able to avoid obstacles

### **RVO (cells)**

obstacle simulation is based on the RVO method with cell sampling.

### **RVO (rays)**

obstacle simulation is based on the RVO method with ray sampling

## **Level height**

Max difference in heights of obstacles to enable their interaction. Used to define minimum margin between obstacles by height, when they are treated as those which are situated one above the other i.e. they doesn't influence to each other.

## **Visualization**

Enable debug visualization for obstacle simulation.