

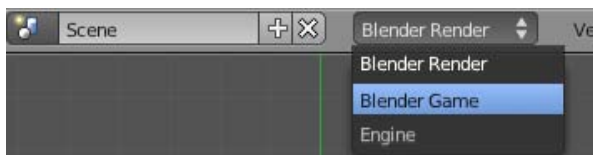
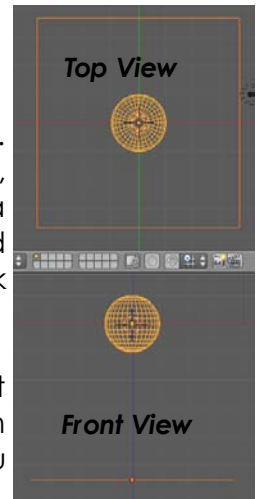
Chapter 21- Game Engine Basics

So how can you make 3D games with Blender? How can you use the physics in Blender to actually create animations for you? Have you tried to make dominoes fall realistically in Blender using traditional animation keys? It would be very difficult to do. Using the real-time features in Blender will do a much better job for you and with a lot less work. A blockbuster movie was just produced last year called 2012 that needed 3D animation of falling buildings and debris that looked real so they turned to the Bullet physics engine to do the work. Bullet is the same physics engine used for the real-time features in Blender. The Blender game engine uses a programming language called Python. Can you make nice games in Blender without knowing Python? The answer is "yes", but if you want to reach a more professional level, knowing Python is a definite. There is a lot of nice documentation on the web for learning Python.

Setting Up The Physics Engine

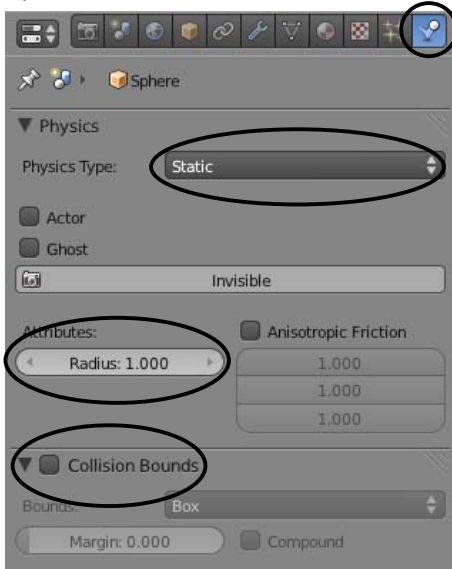
Let's say you want to use physics to make a ball bounce realistically. The 1st thing you need to do is set up the scene. For my sample scene, I have created a UV Sphere a few Blender units above a plane in a front view. Remember that this scene will be using gravity and reactions. If you make your scene in the top view laying flat, it will work just like real life.

It's now time to set up the real-time animation. When I first looked at Blender 2.5 I couldn't even figure out how to turn something into an actor because the interface changed so much! Here's what you need to do:



To enable the Game Engine physics, go to the top bar and find the box for the Render Engine. Change it from "Blender Render" to "Blender Game". This switches many of your property tool panels to game engine

options. We are interested in settings in 3 of these panels:



Physics Panel:

In the Physics panel, you control the Actors in your real-time animation. By default, everything is "Static", meaning that it doesn't react to the physics settings. It can still do things when logic blocks are applied to them, but do nothing otherwise. The other 2 main types we will discuss later are "Dynamic" and "Rigid Body" actors. You can also make something invisible here.

Two other important settings are "Radius" which controls the actor size and "Collision Bounds" which sets the shape of the actor. All of this will be addressed later.