



8th Advanced School on **SCIENTIFIC** VISUALIZATION

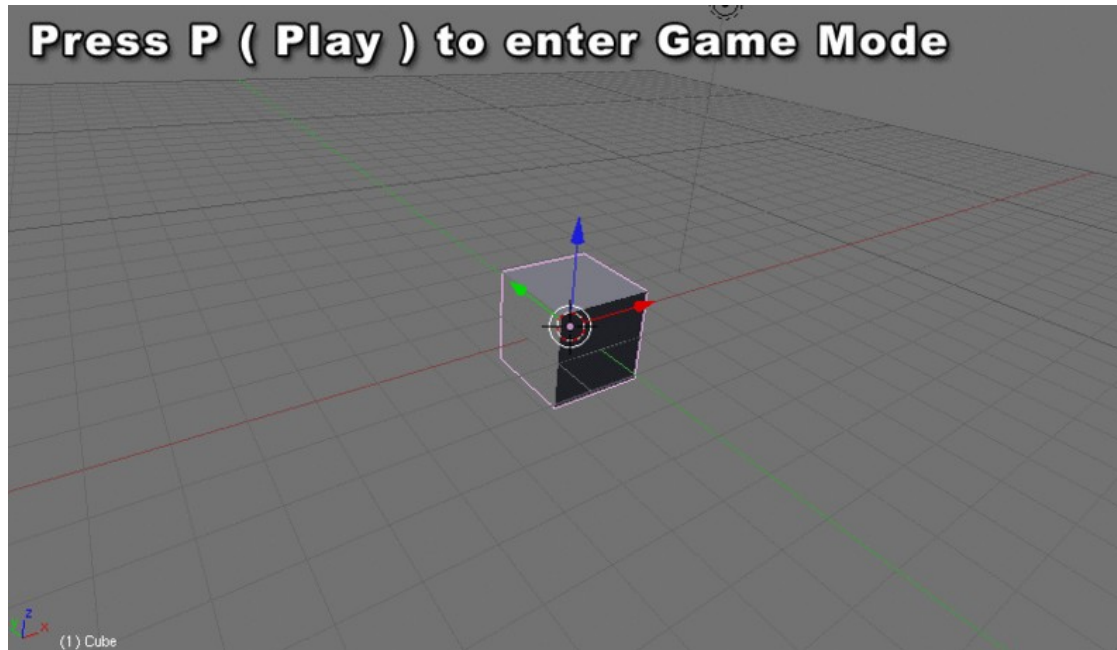
Blender Game Engine with Blender 2.6

Daniele De Luca - [d.deluca@cineca.it](mailto:d.deluca@ Cineca.it)
Ufficio Tecnico - Cineca





Blender Game Engine - Play

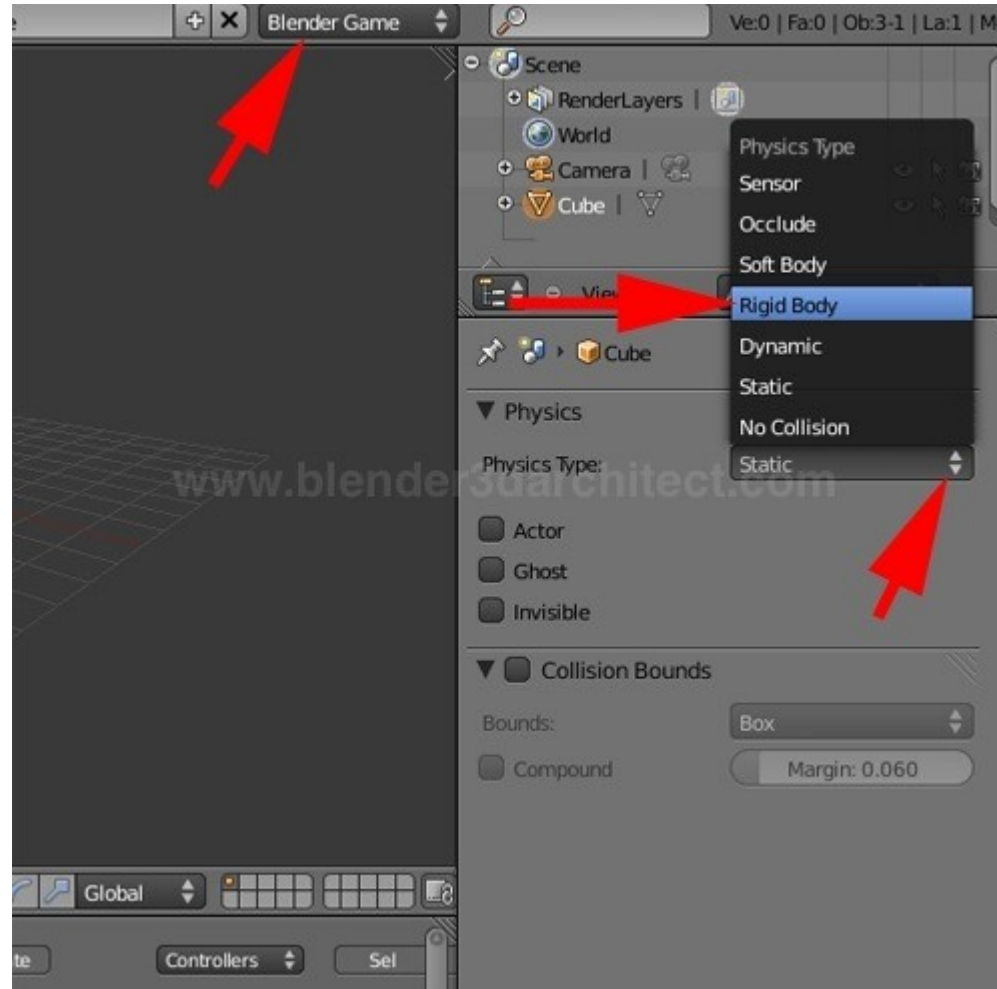


Blender has its own **built in Game Engine** that allows you to create **interactive 3D applications**.

The Blender Game Engine (**BGE**) is a powerful **high-level programming tool**. Its main focus is Game Development, but can be used to create any interactive 3d software, such as interactive 3d architectural tours or educational physics research.

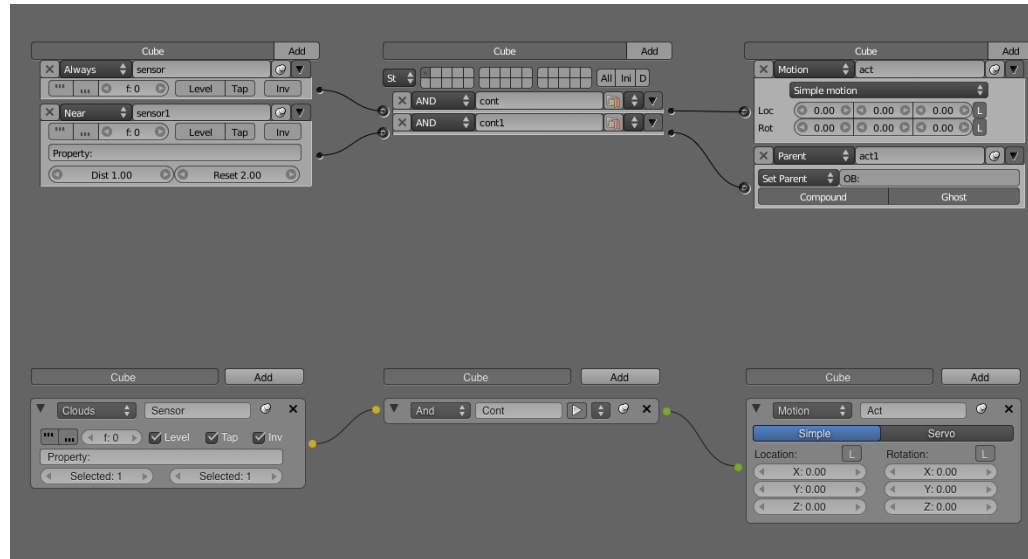


Blender Game Engine - Physics





Blender Game Engine - Logic



Visually controlling the GE - Sensors, Actuators, Controller Logic Blocks

The GE system uses **Logic Blocks** as a visual way to set up **interactions** within the game. These logic blocks can be connected together visually to allow for complex game actions to take place.

There are three different types of Logic Blocks - **Sensors**, **Controllers** and **Actuators** - each with a number of different sub-types.



Blender Game Engine - Logic

Sensors

A Sensor will detect some form of **input**. This input could be anything from a **keypress**, a **joystick** button, or a **timer** that triggers every screen update (or frame) of the game. By default, Blender calculates 60 game frames every second. You can change the frame rate, mist settings and the gravity in the world buttons, physics panel (Physics).

Controllers

Controllers are used to **link Sensors to Actuators**.

They allow for some more complex control over how sensor and actuators interact with each other.

Actuators

An Actuator will actually **carry out an action** within the game. This can include **moving** an object within a scene, playing an **animation**, or playing a **sound** effect.



Blender Game Engine - Scripting

The image shows the Blender Game Engine interface. At the top, a 3D scene is displayed in 'User Ortho' view, featuring several grey rectangular blocks arranged in a line. A yellow circle highlights an 'EMPTY' object in the scene, with a yellow arrow pointing down to the 'LOGIC EDITOR' panel. The 'LOGIC EDITOR' panel shows a logic brick chain starting with an 'Always' sensor, followed by a 'Script' controller. The 'Script' controller is highlighted with a yellow box, and a yellow arrow points from it to the Python code editor on the right. The Python code editor shows a script named 'BGEgravity' with the following code:

```
4 scene = bge.logic.getCurrentScene()
5
6 obList = []
7 for o in scene.objects:
8     if o.class_ == bge.types.KX_GameObject:
9         obList.append(o)
10
11 def calcGrav(obA, obB, G):
12     m1 = obA.mass; m2 = obB.mass
13     m = m1*m2
14     loc1 = obA.worldPosition
15     loc2 = obB.worldPosition
16     v = loc1 - loc2
17     r = v.length
18     F = G * (m) / (r*r)
19     return -v * F
20
21 def loopGrav(obs, G):
22     for obA in obs:
23         fV = Vector((0,0,0))
24         for obB in obs:
25             if obA != obB:
26                 fV += calcGrav(obA, obB, G)
27             obA.applyForce(fV, False)
28
29 loopGrav(obList, 1)
30
31 keyb = bge.logic.keyboard
32 if keyb.events[bge.events.ZKEY]>0:
33     for ob in obList:
34         ob.applyTorque(Vector((0,0,50)), False)
```

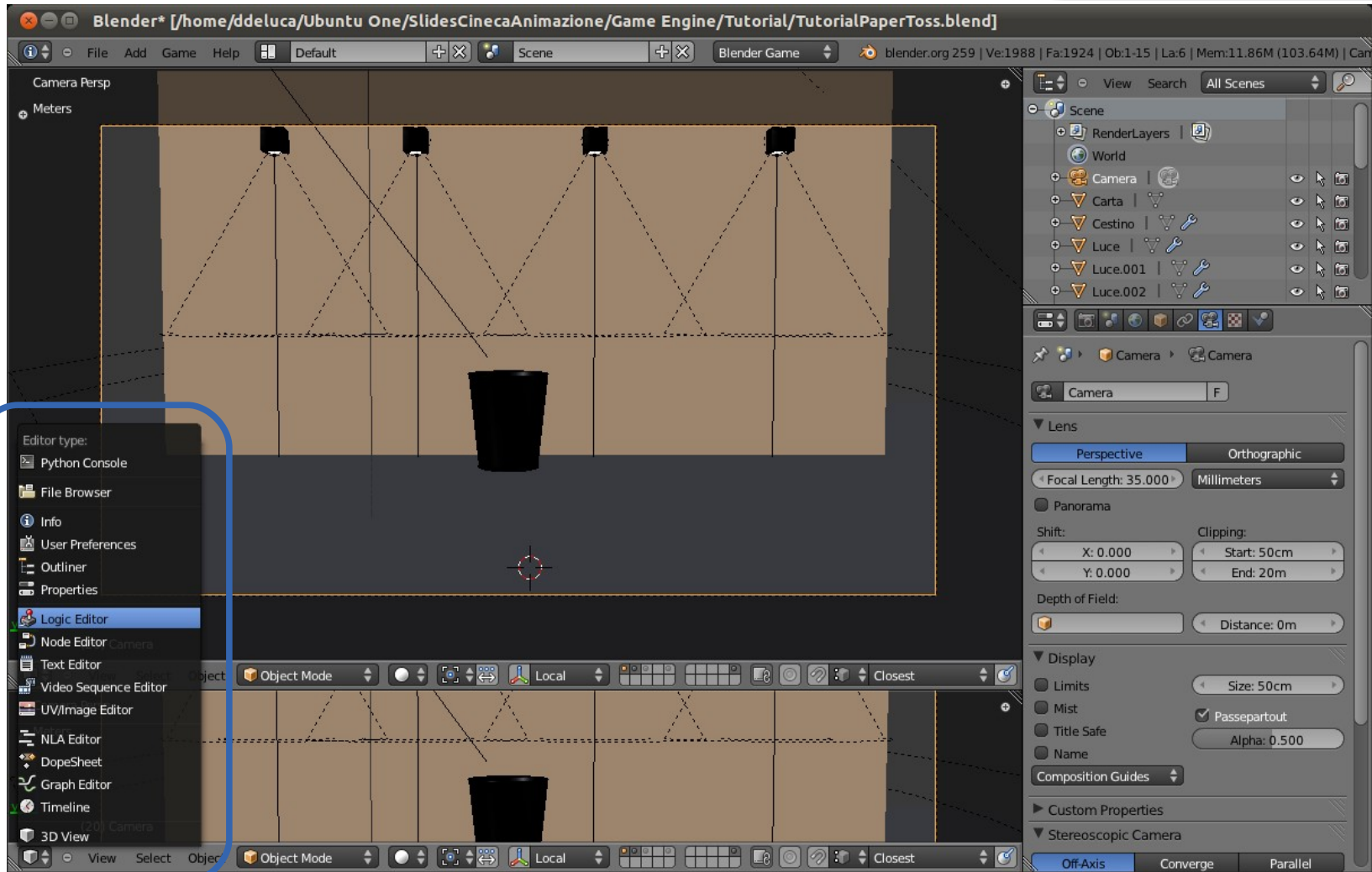


BGE - Tutorial



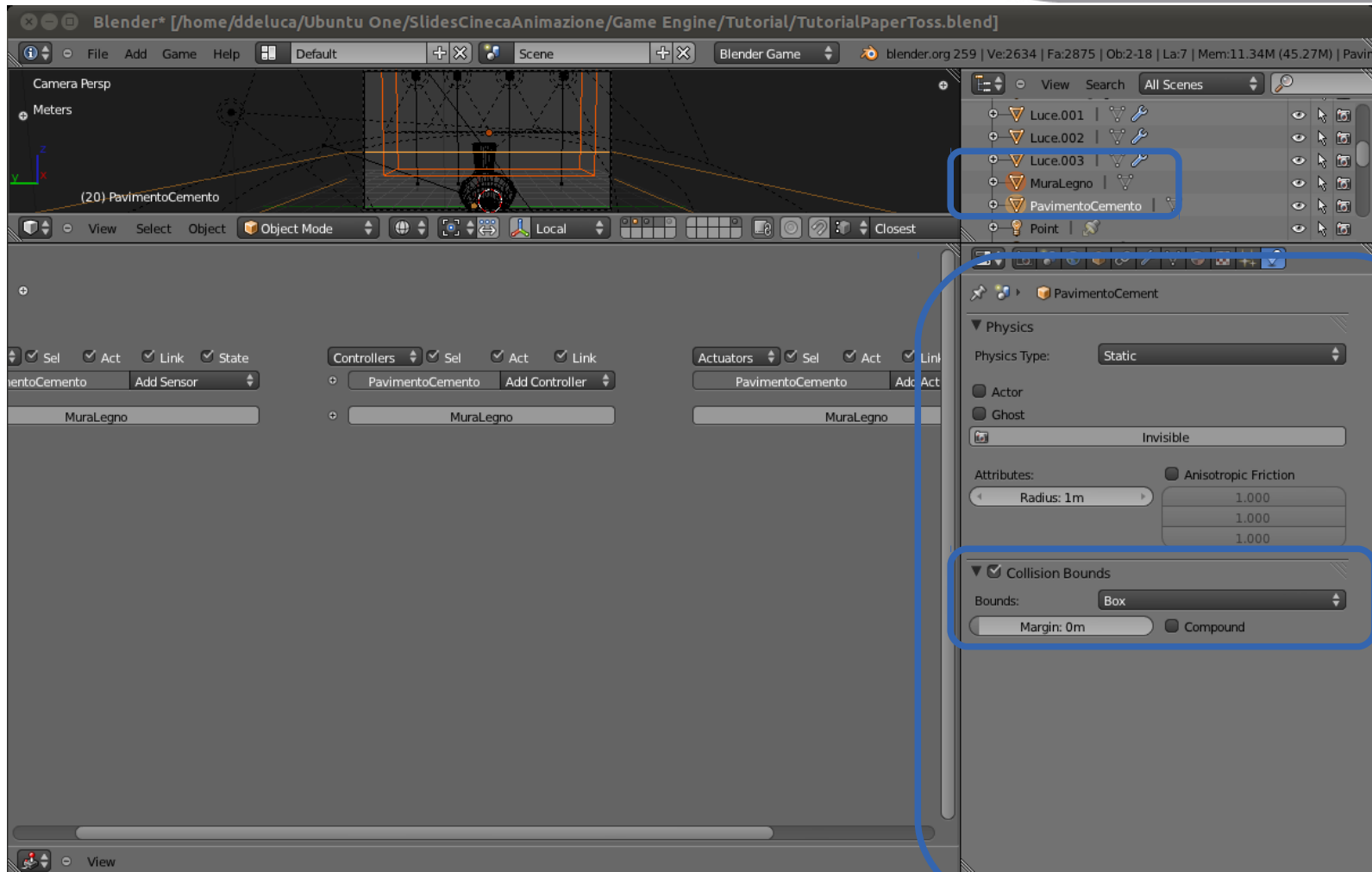


BGE - Tutorial



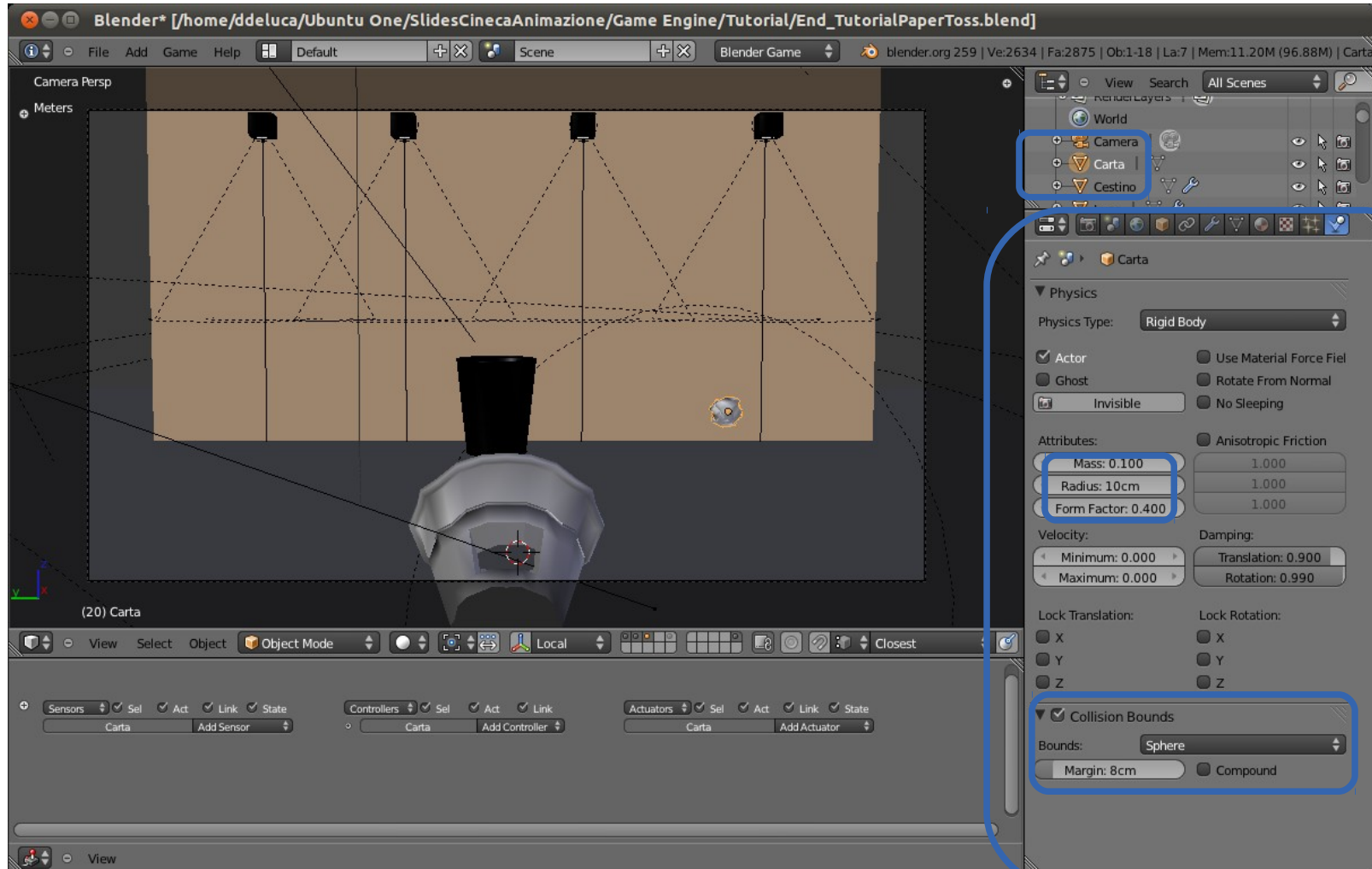


BGE - Tutorial





BGE - Tutorial





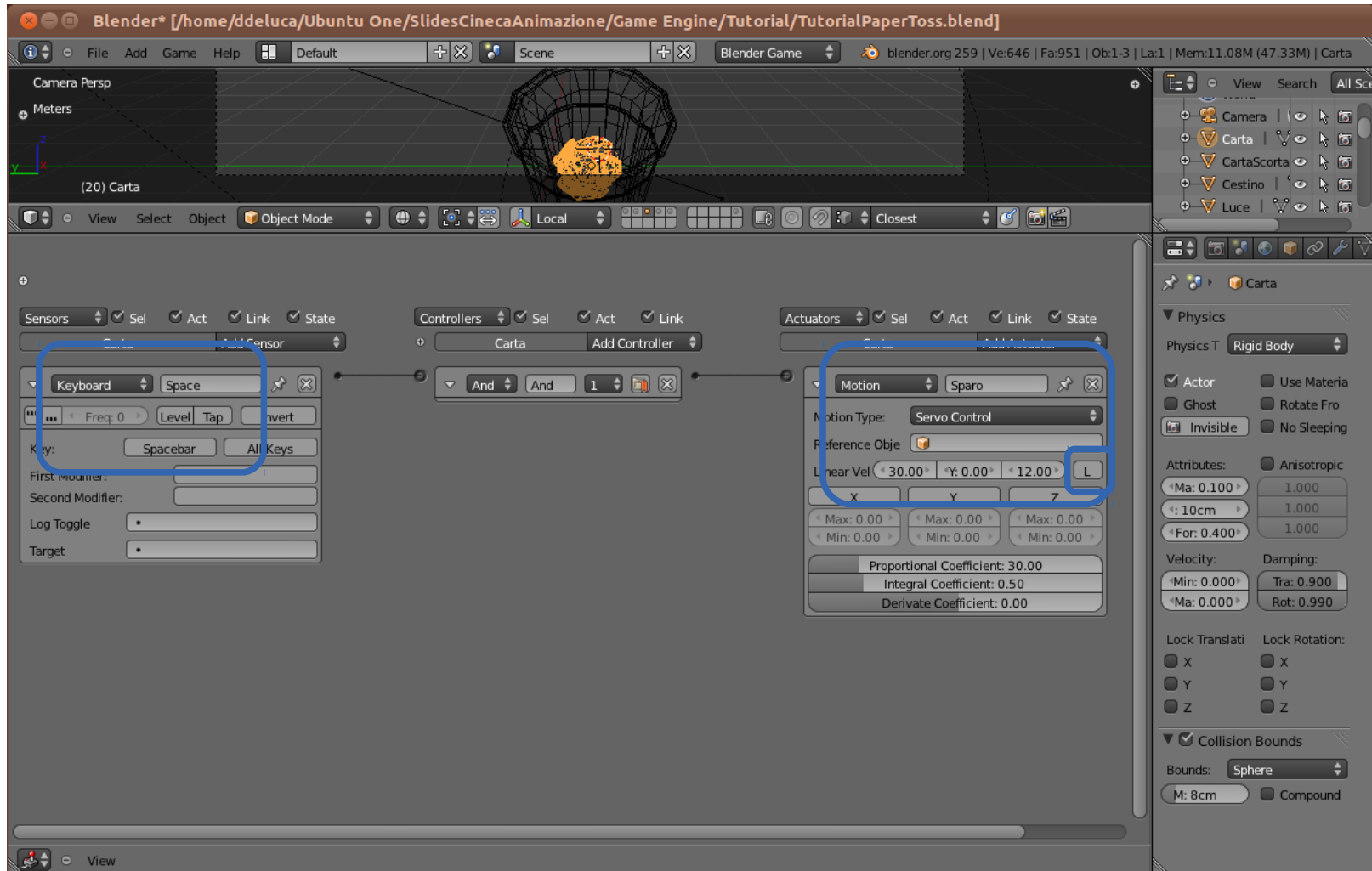
BGE - Tutorial

The screenshot shows the Blender Game Engine (BGE) interface. The top window title is "Blender* [/home/ddeluca/Ubuntu One/SlidesCinecaAnimazione/Game Engine/Tutorial/TutorialPaperToss.blend]". The interface is divided into several panels:

- 3D Viewport:** Shows a wireframe Sparacarta object in a perspective view. The top status bar indicates "blender.org 259 | Ver:646 | Fa:951 | Ob:1-3 | La:1 | Mem:11.12M (47.33M) | Sparacarta".
- Logic Editor:** Located at the bottom, it consists of three main columns: Sensors, Controllers, and Actuators. A blue box highlights the "Keyboard" sensor in the Sensors column and the "Simple Motion" actuators in the Actuators column. The "Keyboard" sensor is configured with "Key: Right Arrow" and "All Keys". The "Simple Motion" actuators are configured with "Motion Type: Simple Motion" and "Rot: 0°, -1°, 0°".
- Properties Panel:** On the right, it shows the "Physics" and "Collision Bounds" properties for the Sparacarta object. The "Physics" section is set to "Static" and "Invisible". The "Collision Bounds" section is set to "Box" with a radius of "1.000".

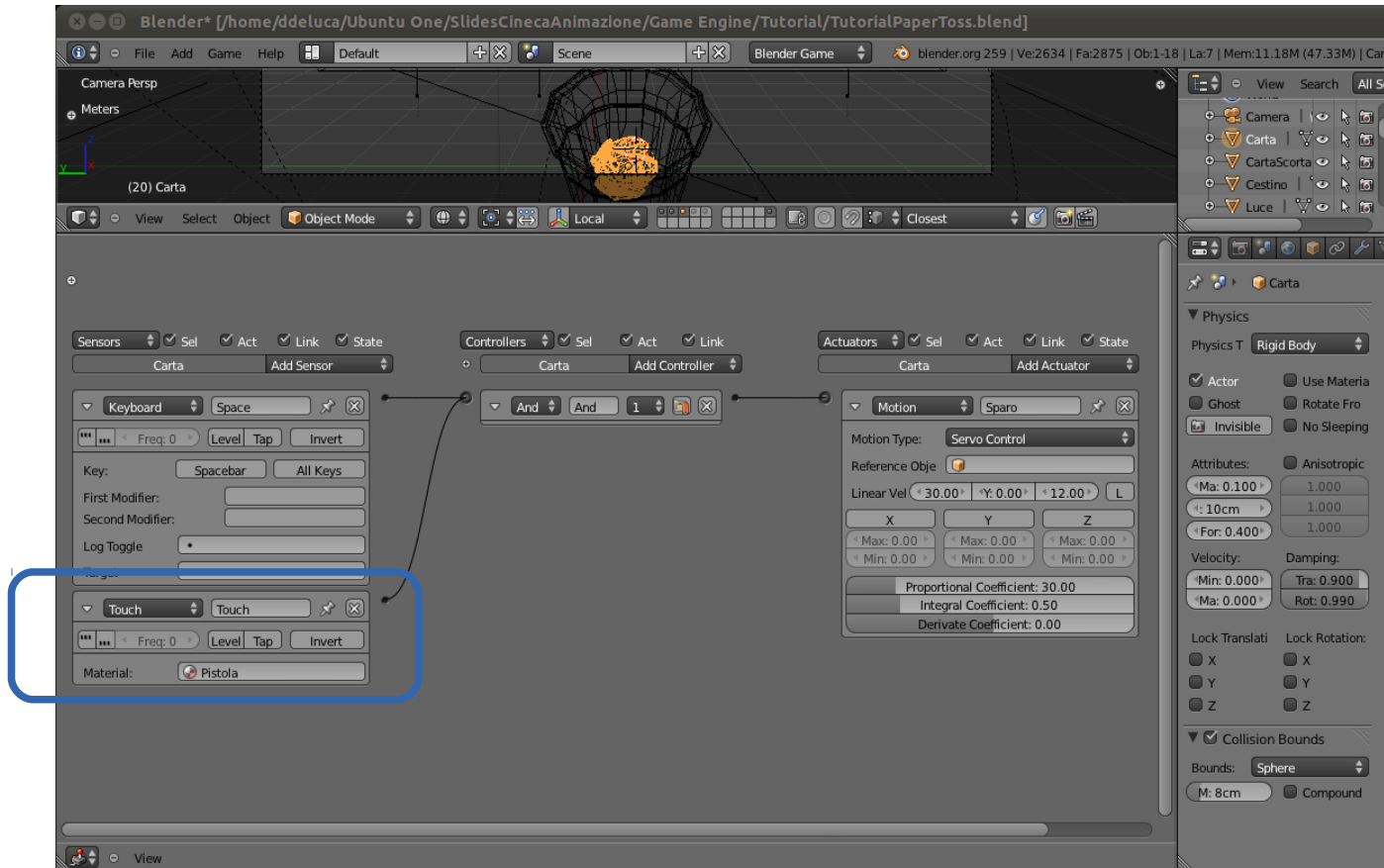


BGE - Tutorial





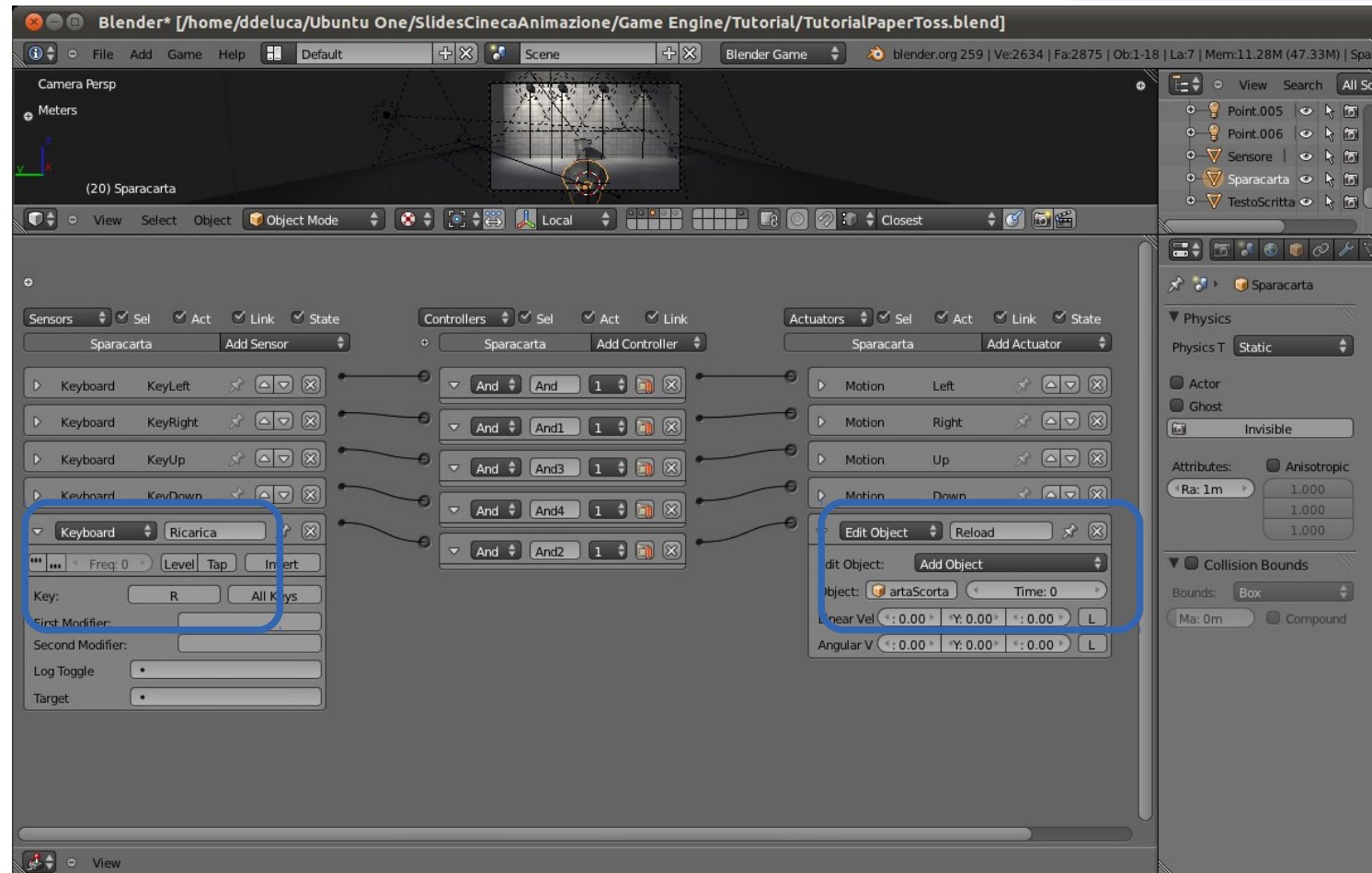
BGE - Tutorial



Now that you have added some actions to your paper ball, you could duplicate it by pressing **SHIFT+D** and “**Esc**” to deselect the duplicated moving ball. Now **hide** the new ball into an hidden layer by pressing **m** and selecting for instance the **layer four** (that is deselected)



BGE - Tutorial



Select the **duplicated** ball object into the edit object actuator, mine is called **CartaScorta**



BGE - Tutorial

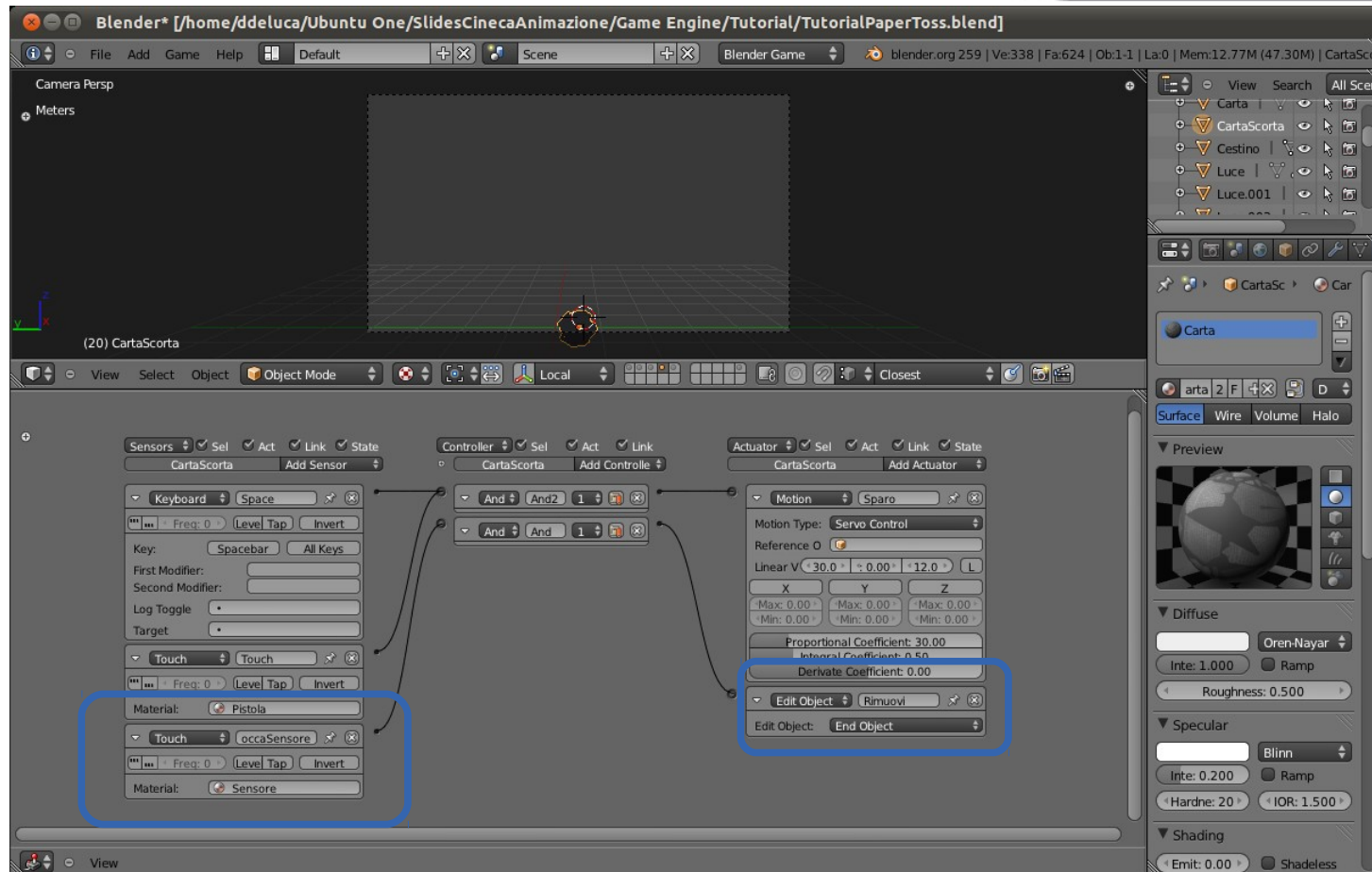
The screenshot displays the Blender Game Engine (BGE) interface. The main 3D viewport shows a large blue text object labeled "Great!". The interface includes a top menu bar, a left sidebar with "User Persp" and "Meters", a central toolbar, and a right sidebar with various panels. The "Physics" panel for the "TestoScritttaGreat" object is highlighted with a blue circle. The "Physics" panel shows the following settings:

- Physics Type: Rigid Body
- Actor:
- Ghost:
- Invisible:
- Use Material Force F:
- Rotate From Normal:
- No Sleeping:
- Anisotropic Friction:
- Mass: 1.000
- Radius: 1m
- Form Factor: 0.400
- Velocity: Minimum: 0.000, Maximum: 3.000
- Damping: Translation: 0.140, Rotation: 0.100
- Lock Translation: X, Y, Z (all unchecked)
- Lock Rotation: X, Y, Z (all unchecked)
- Collision Bounds: Box, Margin: 0m, Compound:

The bottom of the interface shows the Logic Editor with three columns: Sensors, Controllers, and Actuators. Each column contains a "TestoScritttaGreat" object and an "Add" button.



BGE - Tutorial



As you add sensor and actuators to your paper ball, you need to add the same sensors and actuators also to your **duplicated** ball.



BGE - Tutorial

The screenshot shows the Blender 2.59 Game Engine interface. The main 3D viewport displays a scene with a tiled wall, spotlights, and a paper toss mechanism. The logic editor at the bottom is configured with the following components:

- Sensors:** A 'Touch' sensor is highlighted with a blue box. It is set to 'Touch' with a frequency of 0 and 'Level' set to 'Tap'.
- Controllers:** Two 'And' controllers are connected in a sequence.
- Actuators:** An 'Edit Object' actuator is highlighted with a blue box. It is set to 'Edit Object' with 'Visibility' selected. The object is 'crittaGreat' and the time is set to 0. Linear and angular velocities are all set to 0.00.

On the right side, the Properties panel shows the 'Physics' settings for the selected object, with 'Physics T' set to 'Static' and 'Invisible' checked. The 'Collision Bounds' section shows 'Box' bounds with a radius of 10 cm.



Blender Game Engine

Resources

[Wiki Game_Engine](#)

[Schede-sensori-controller-attuatori](#)

[Scripting](#)

[in BGE](#)
[Dead Cyborg](#)

[Project Aftershock](#)

[Create Game with BGE - Italian Tut](#)

[orial](#)
[Mike Pan Files](#)

[Lucy And the Time Machine](#)

[BPC_EnvMap demo](#)

[Burster - Don't be flat, be 3D!](#)

[Blender Game Competition 2010](#)

Videos

[GLSL Demo](#)

[Megagolf](#)



Blender – Resources

[Blender 2.5 Cheat Sheet](#)

[Cineca MDC Blog](#)

[My Blog - Discoverylinux](#)

[Tips and Tricks](#)

Thanks!
And wish you Happy Blending!