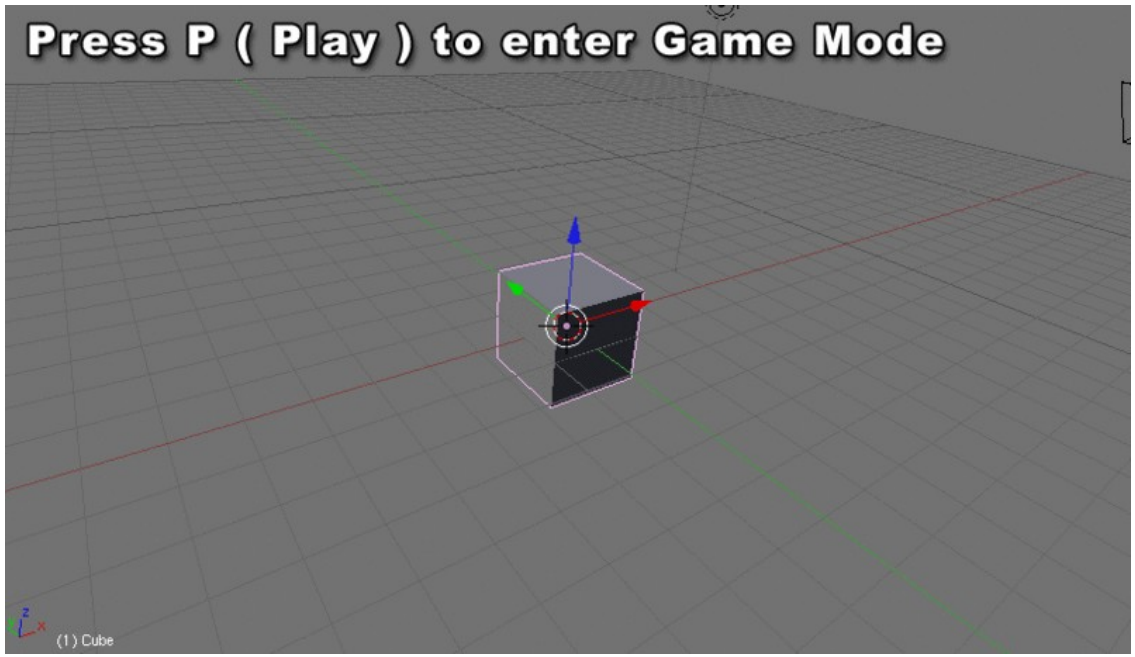
8th Advanced School on SCIENTIFIC VISUALIZATION

Titolo del corso

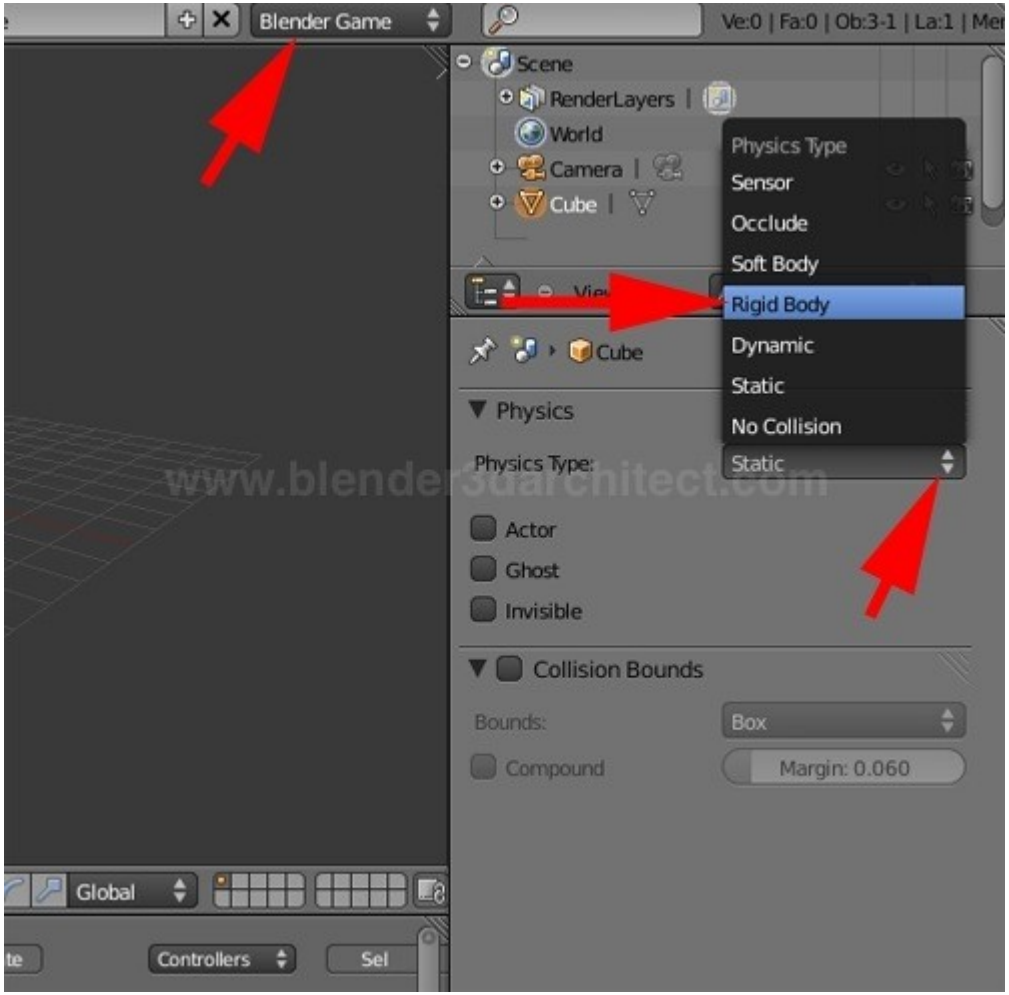
Nome Cognome – mail
SuperComputing Applications and Innovation Department

CINECA

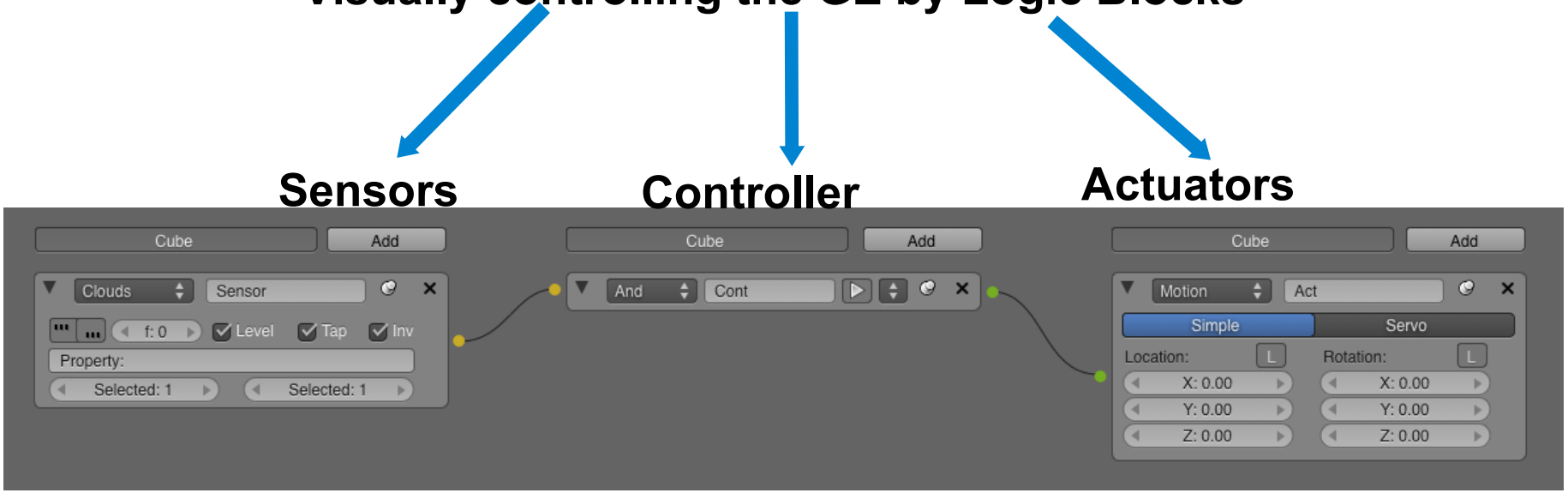


Blender has its own **built in Game Engine** that allows you to create **interactive 3D applications.**

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 archaeological tours or educational physics research.


Blender Game
Ve:0 | Fa:0 | Ob:3-1 | La:1
Scene
RenderLayers
World
Camera
Cube
Physics Type
Sensor
Occlude
Soft Body
Rigid Body
Dynamic
Static
No Collision
Cube
Physics
Physics Type:
Static
Actor
Ghost
Invisible
Collision Bounds
Bounds:
Box
Compound
Margin: 0.060
Global
Controllers
Sel

Visually controlling the GE by 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.

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

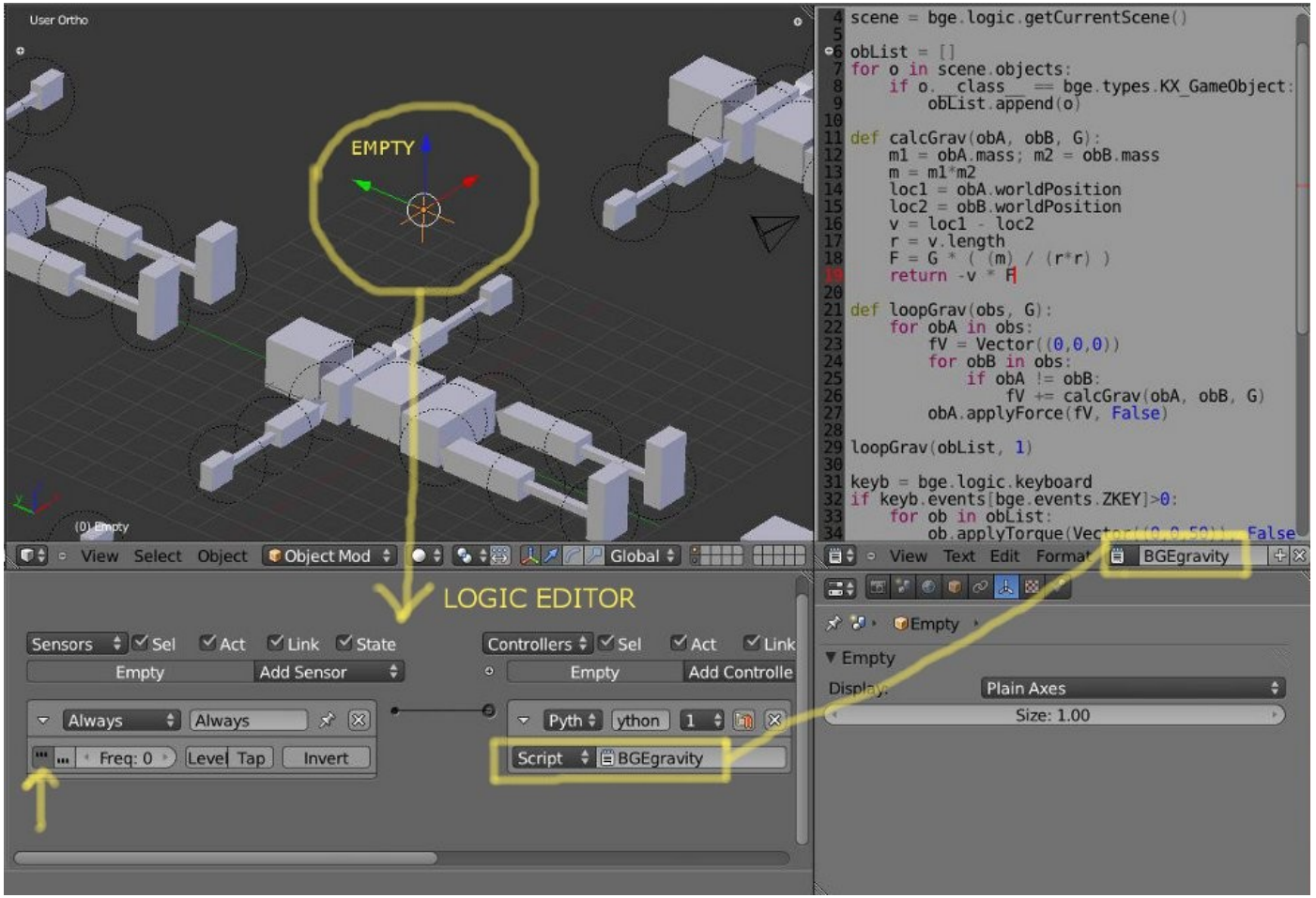
Controllers are used to **link Sensors to Actuators**.

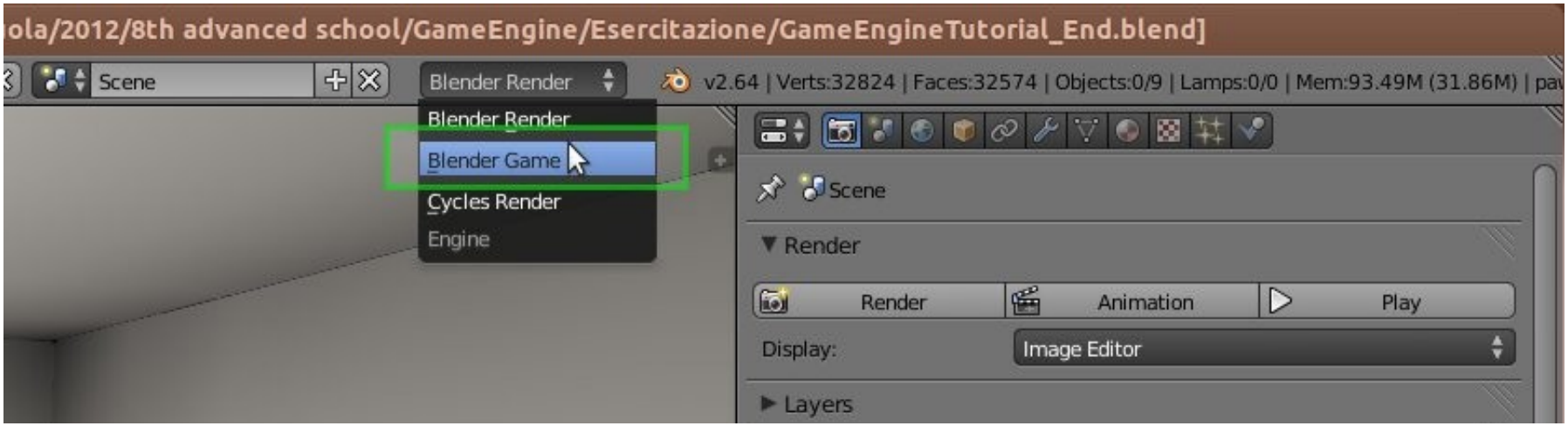
They allow for some more complex control over how sensor and actuators interact with each other.

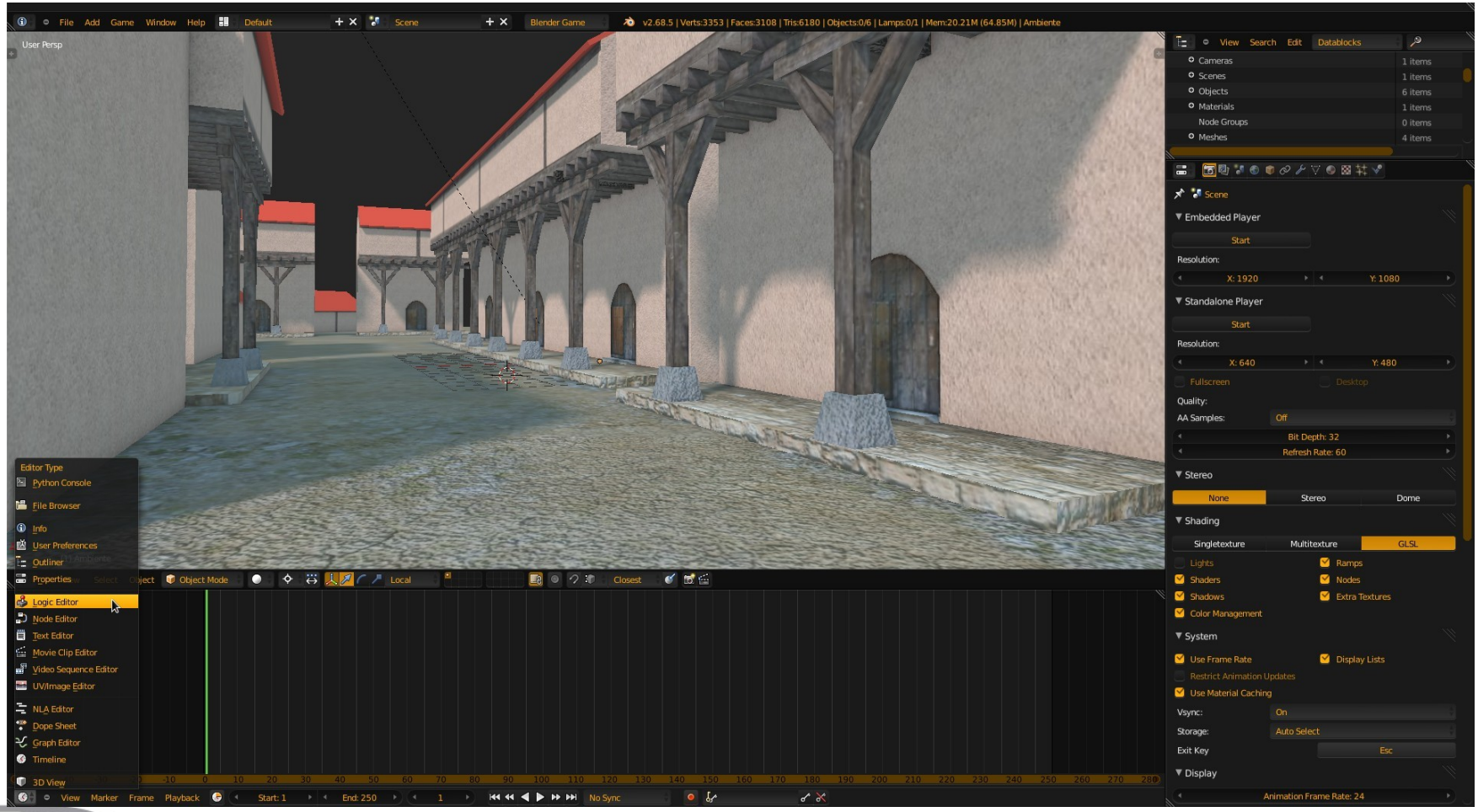
Actuators

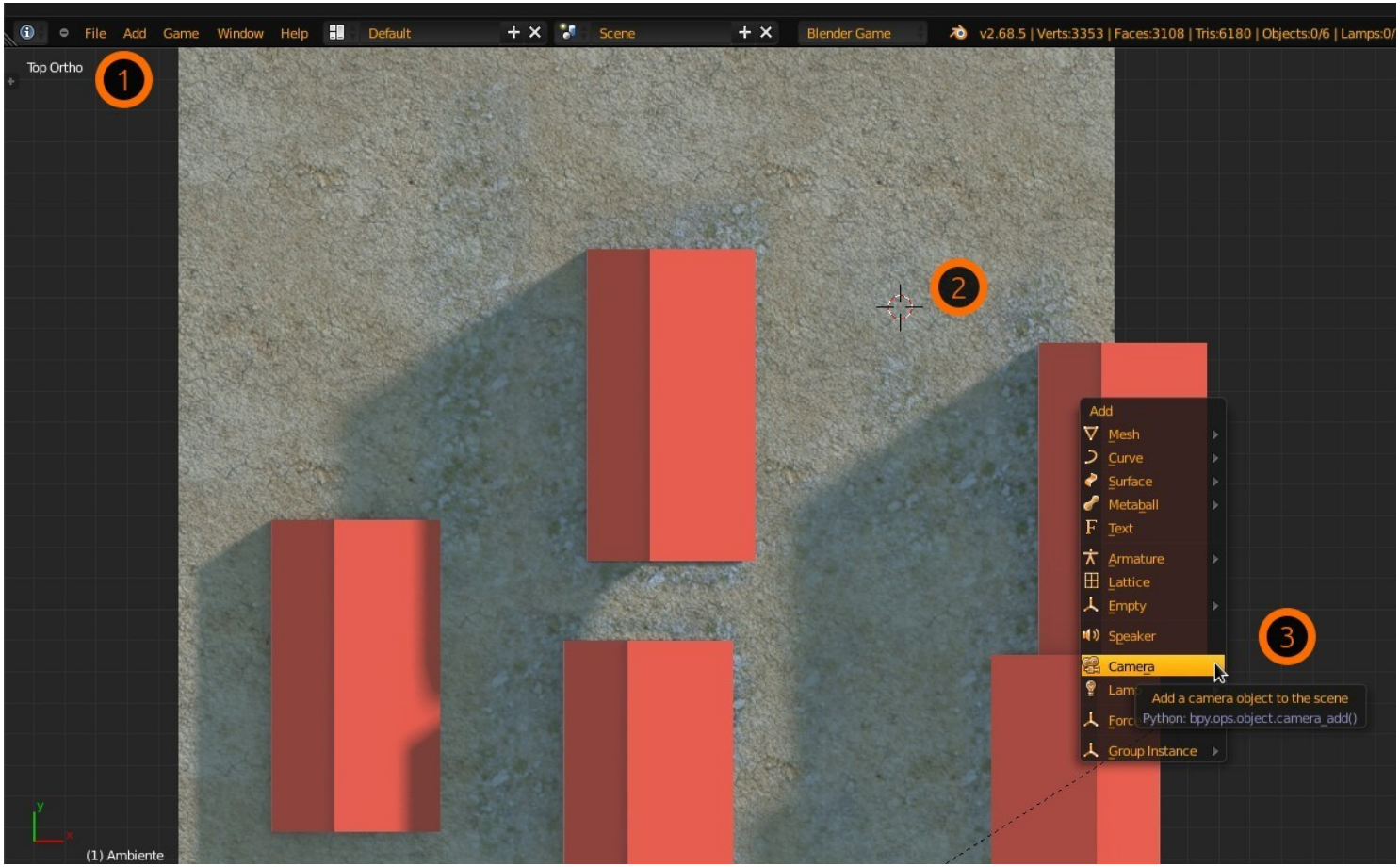
An Actuator will actually **carry out an action** within the game. This can include **moving** an object within a scene, playing an **animation**, or playing a **sound** effect.

Blender Game Engine - Logic

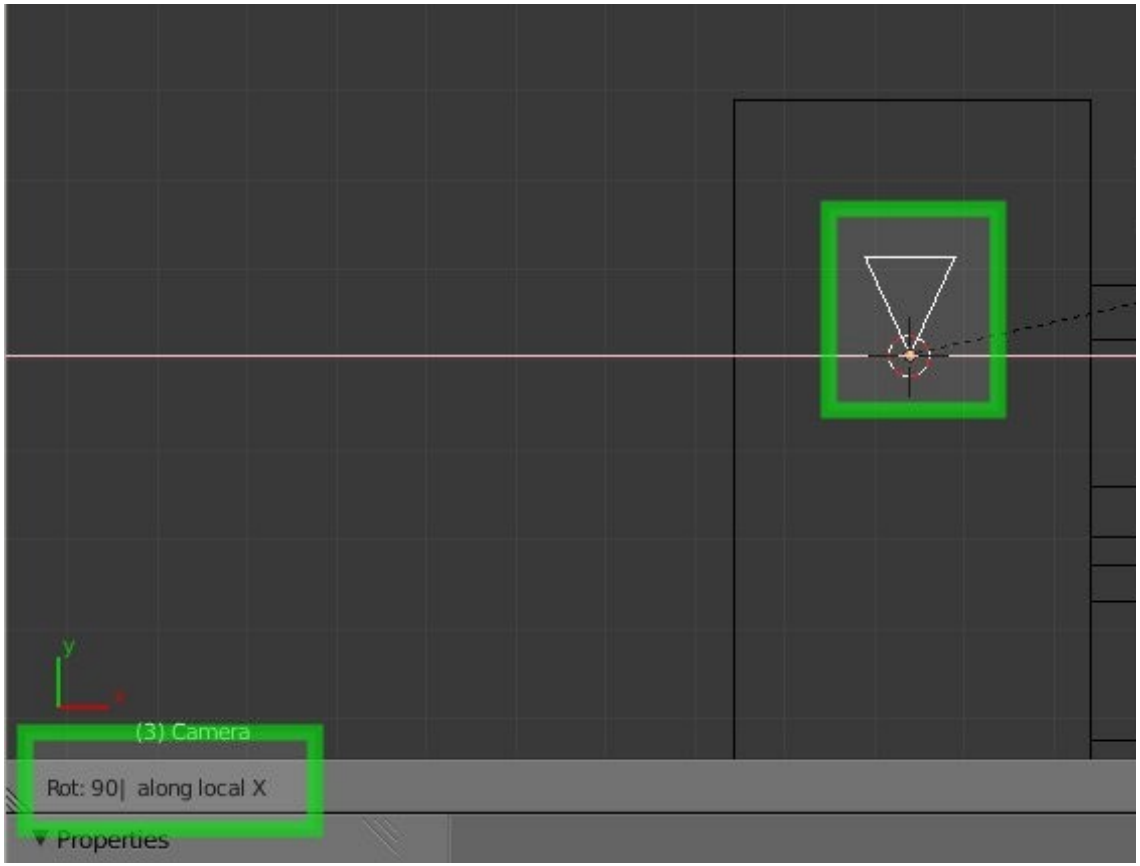



ola/2012/8th advanced school/GameEngine/Esercitazione/GameEngineTutorial_End.blend]
Scene
Blender Render
v2.64 | Verts:32824 | Faces:32574 | Objects:0/9 | Lamps:0/0 | Mem:93.49M (31.86M) | pa
Blender Render
Blender Game
Cycles Render
Engine
Scene
Render
Render
Animation
Play
Display:
Image Editor
Layers

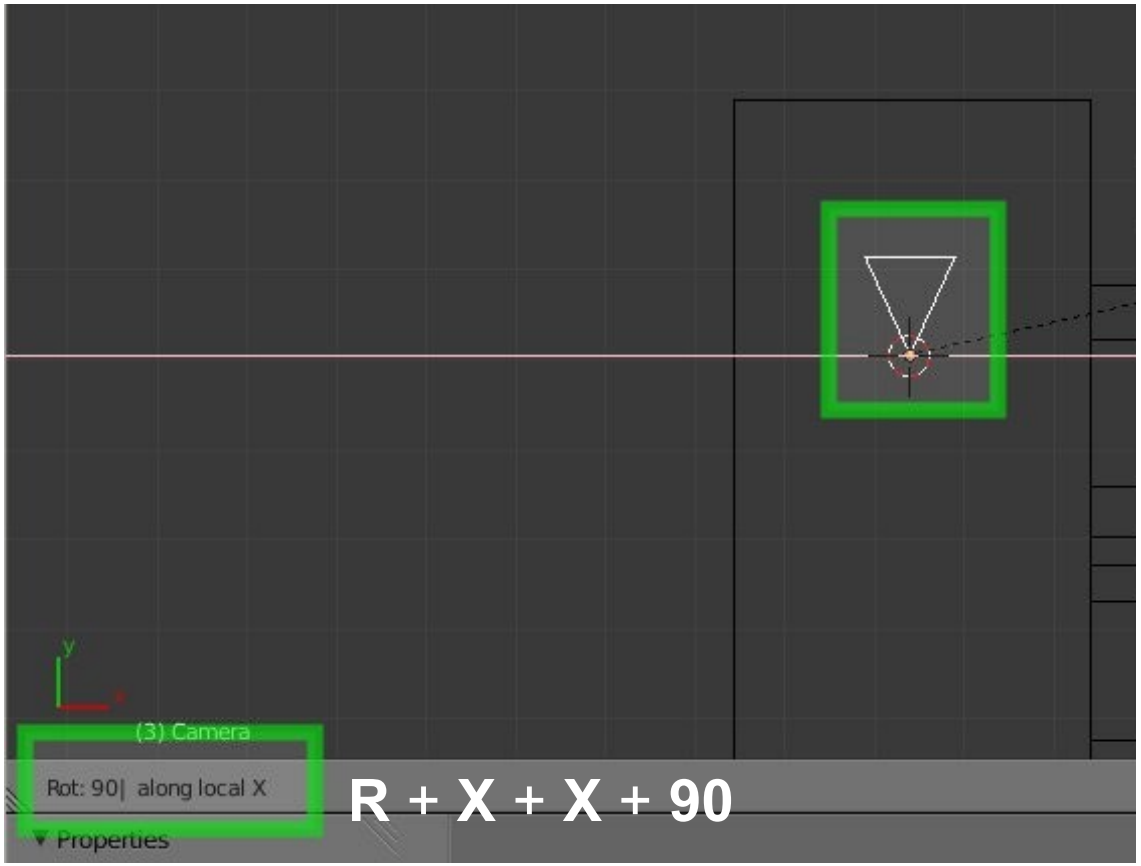

File Add Game Window Help Default Scene Blender Game v2.68.5 | Verts:3353 | Faces:3108 | Tris:6180 | Objects:0/6 | Lamps:0/1 | Mem:20.21M (64.85M) | Ambiente
User Persp
View Search Edit Datablocks
Cameras 1 items
Scenes 1 items
Objects 6 items
Materials 1 items
Node Groups 0 items
Meshes 4 items
Scene
Embedded Player
Start
Resolution:
X: 1920 Y: 1080
Standalone Player
Start
Resolution:
X: 640 Y: 480
Fullscreen Desktop
Quality:
AA Samples: Off
Bit Depth: 32
Refresh Rate: 60
Stereo
None Stereo Dome
Shading
Singletexture Multitexture GLSL
Lights Ramps
Shaders Nodes
Shadows Extra Textures
Color Management
System
Use Frame Rate Display Lists
Restrict Animation Updates
Use Material Caching
Vsync: On
Storage: Auto Select
Exit Key Esc
Display
Animation Frame Rate: 24
Editor Type
Python Console
File Browser
Info
User Preferences
Outliner
Properties
Logic Editor
Node Editor
Text Editor
Movie Clip Editor
Video Sequence Editor
UV/Image Editor
NLA Editor
Dope Sheet
Graph Editor
Timeline
3D View
Object Mode Local Closest
View Marker Frame Playback Start: 1 End: 250 1 No Sync



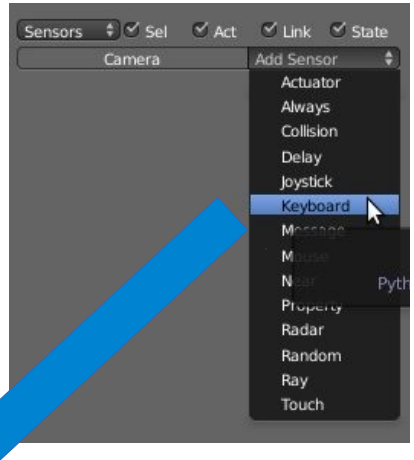
Add a camera at the cursor position



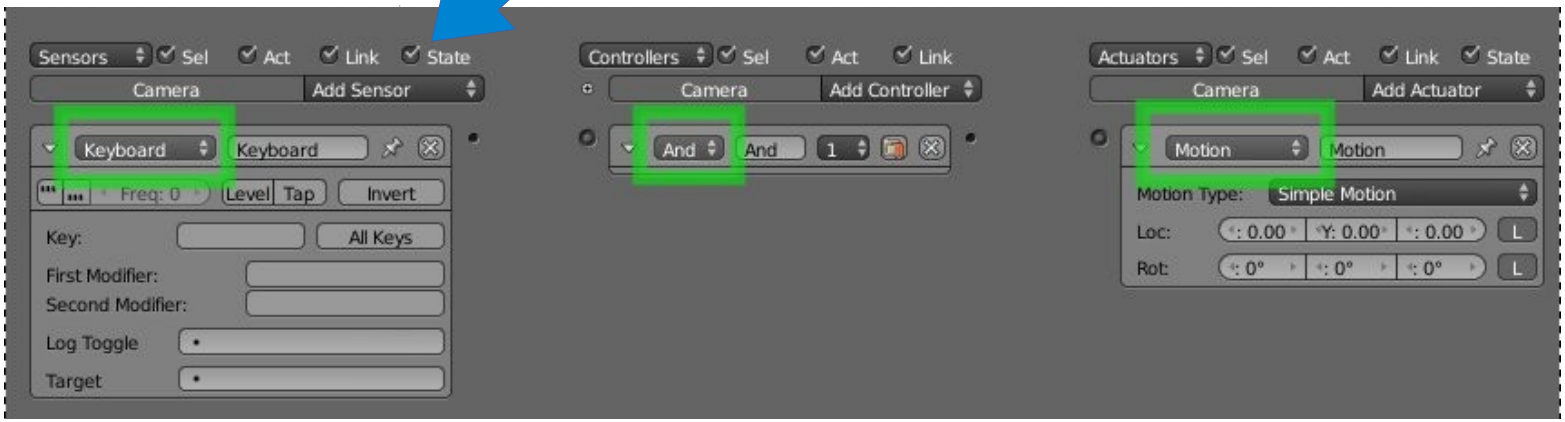
Rotate the camera
90 degrees on the local X axis



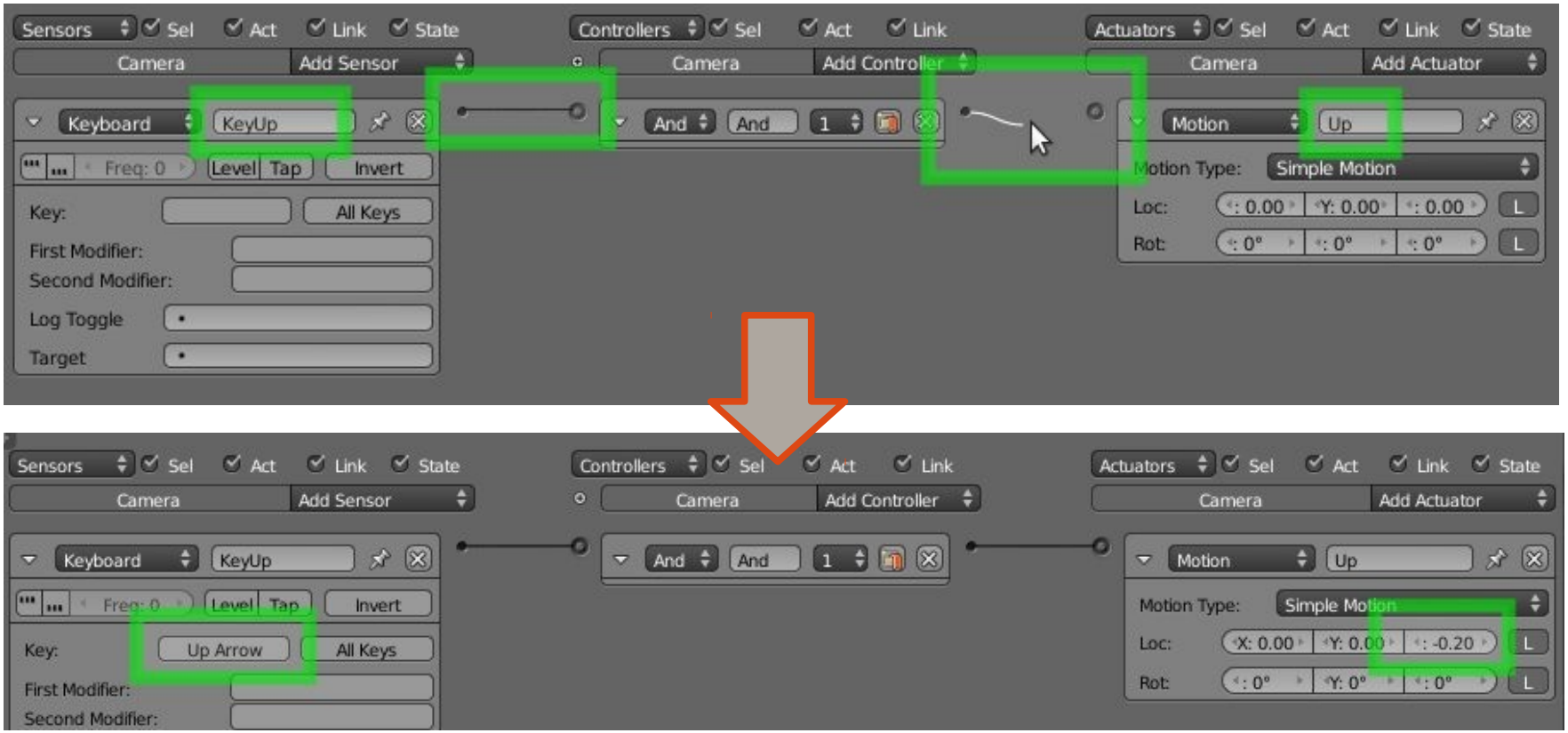
Rotate the camera
90 degrees on the local X axis



Add a **Keyboard** Sensor, an **And** Controller and a **Motion** Actuator



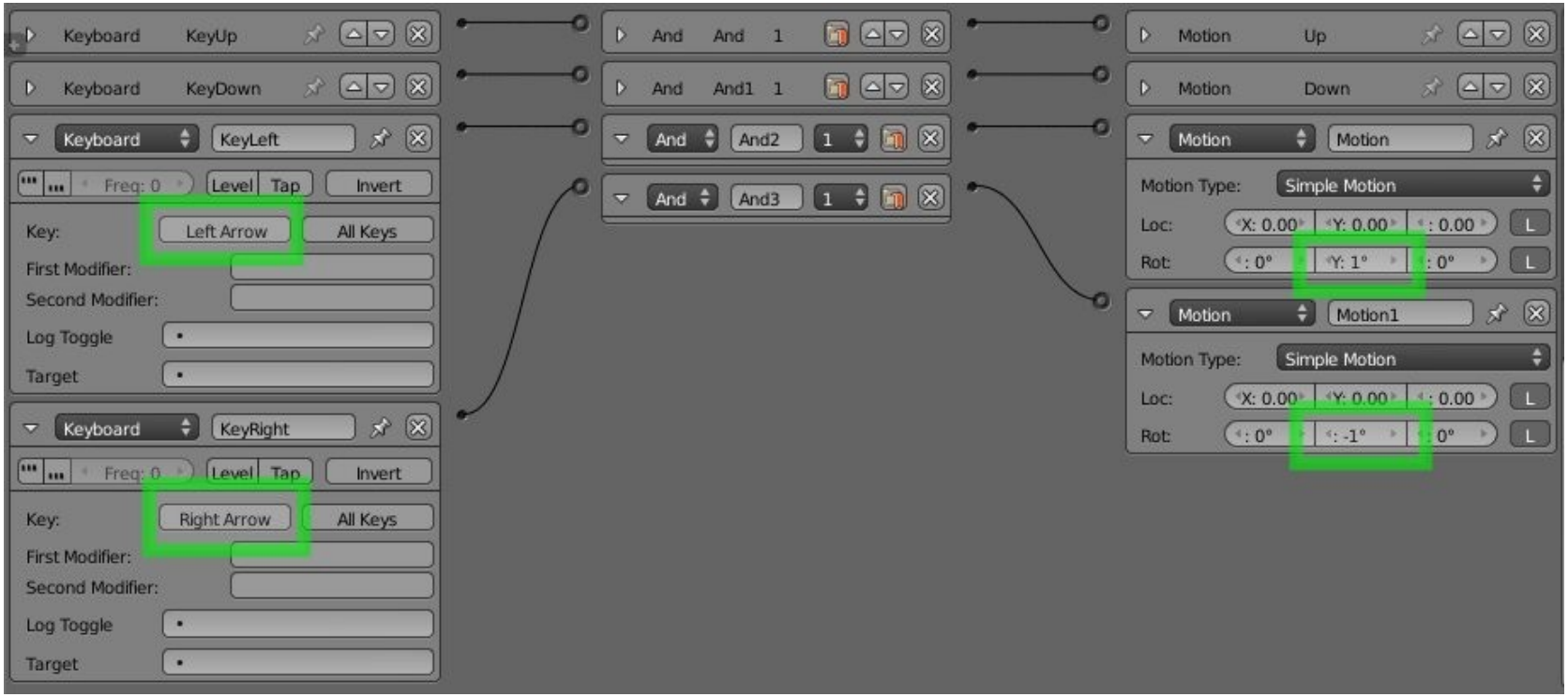
Connect and **name** those logic bricks



Set the **Up Arrow** and the **Z** movement

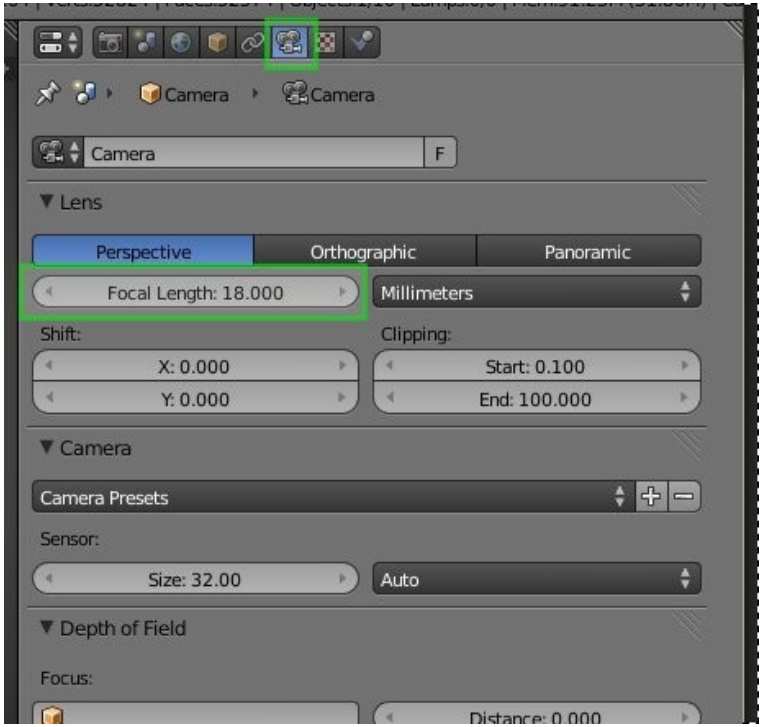
Why Z?

Complete the movement with a **KeyDown** logic brick and Add the **rotation** controls:

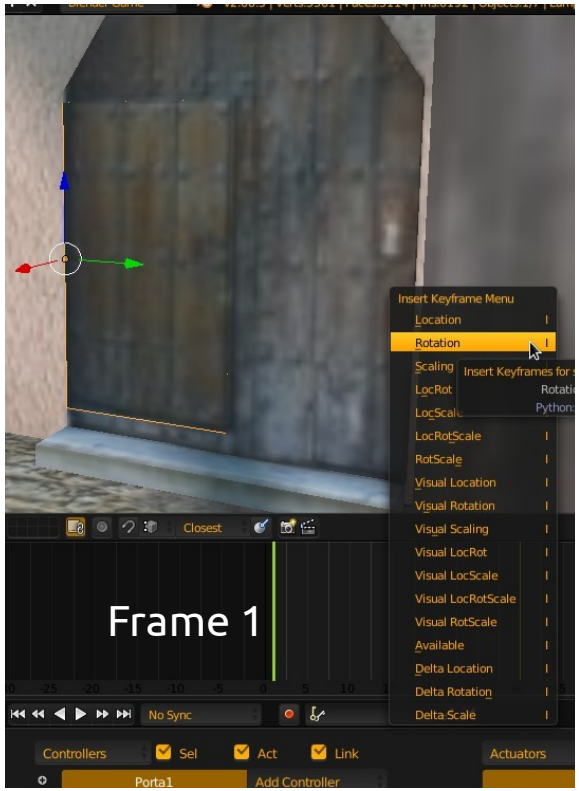


Why Y?

Tip
Change the **focal length** of your camera to view the spaces more confortably



Press **P** to **Play** your virtual tour

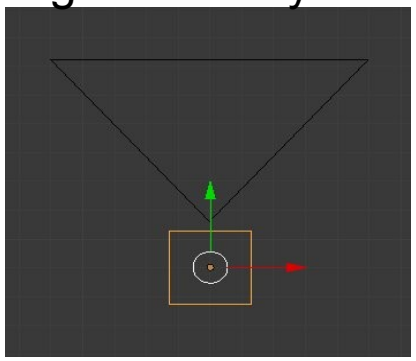


Select the “porta1” object and **add an Animation** for the open action of the door

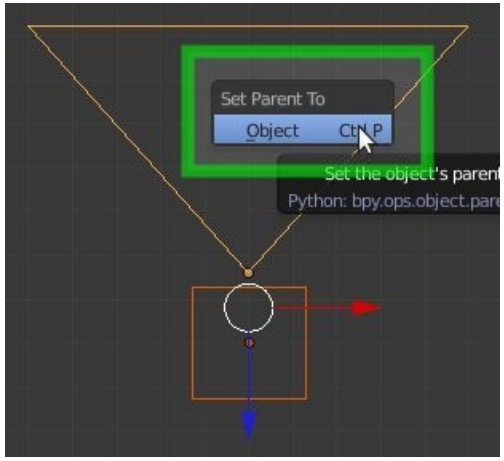
To add the animation, insert a **keyframe** (Key i) of the **Rotation** with the door **closed** at frame 1

And with the door **opened** at frame 30

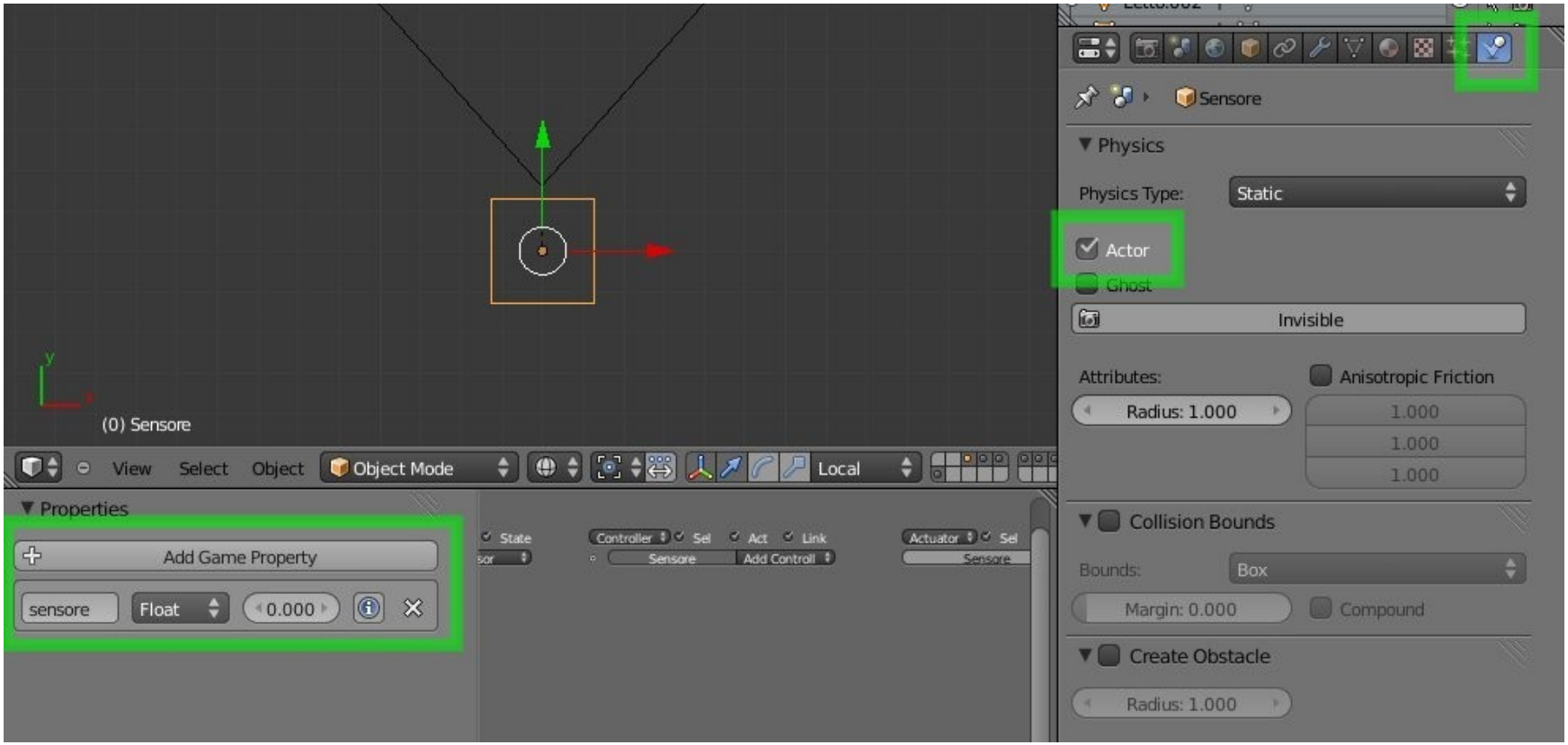
To add your interaction, you first need a **physic sensor:**
Add and **scale** a **Cube** right behind your camera



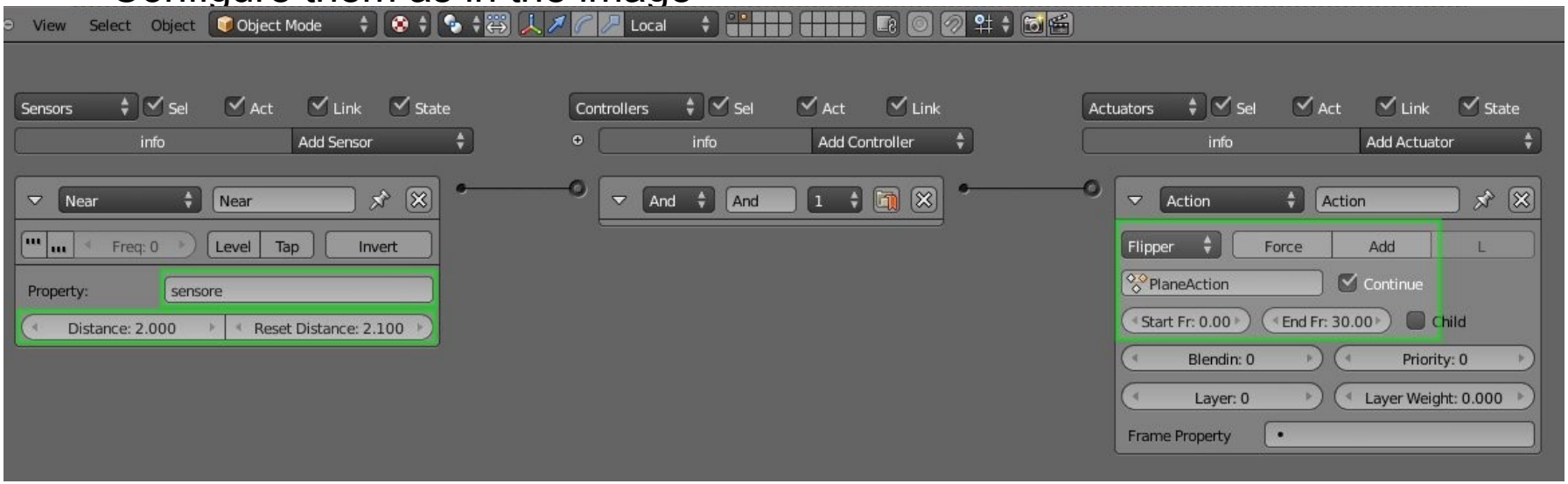
Then **parent** them: select the **Cube** and **then the Camera,** then press **CTRL +**



Add a Property to the Cube and **name** that Property **“sensore”**
Then in the **Physics** panel, **tick** the **Actor** button

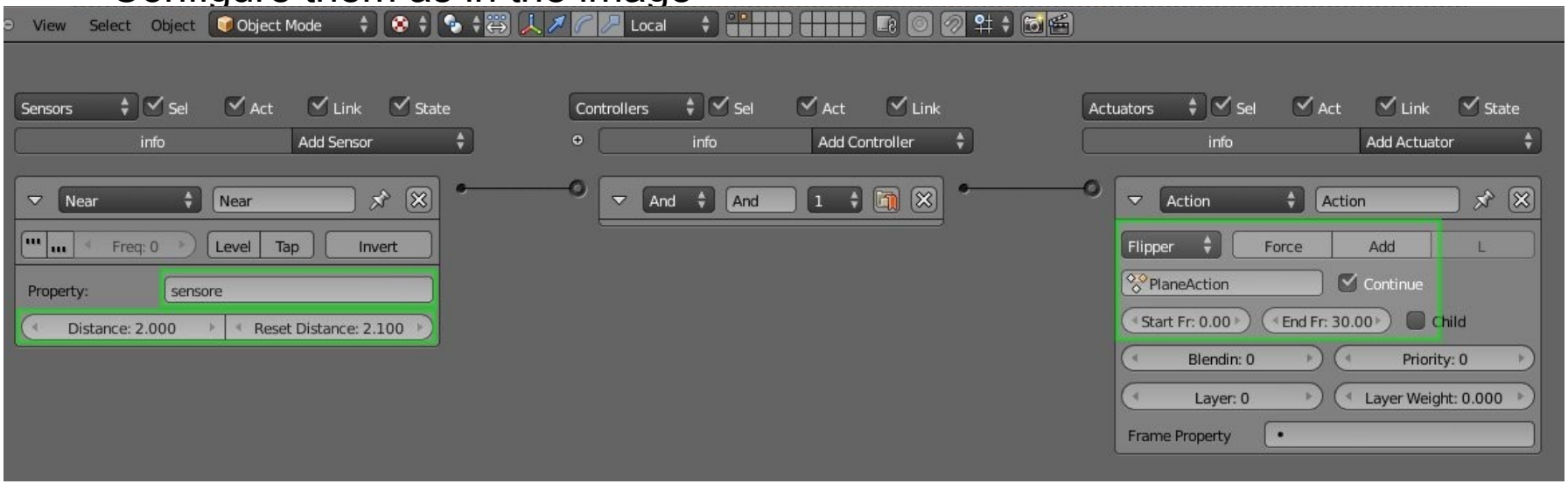


Select the “porta1” and add a **Near** Sensor, an **And** Controller and an **Action** Actuator.
Configure them as in the image



“sensore” in the Near Sensor, tells the door to look for the **cube proximity**

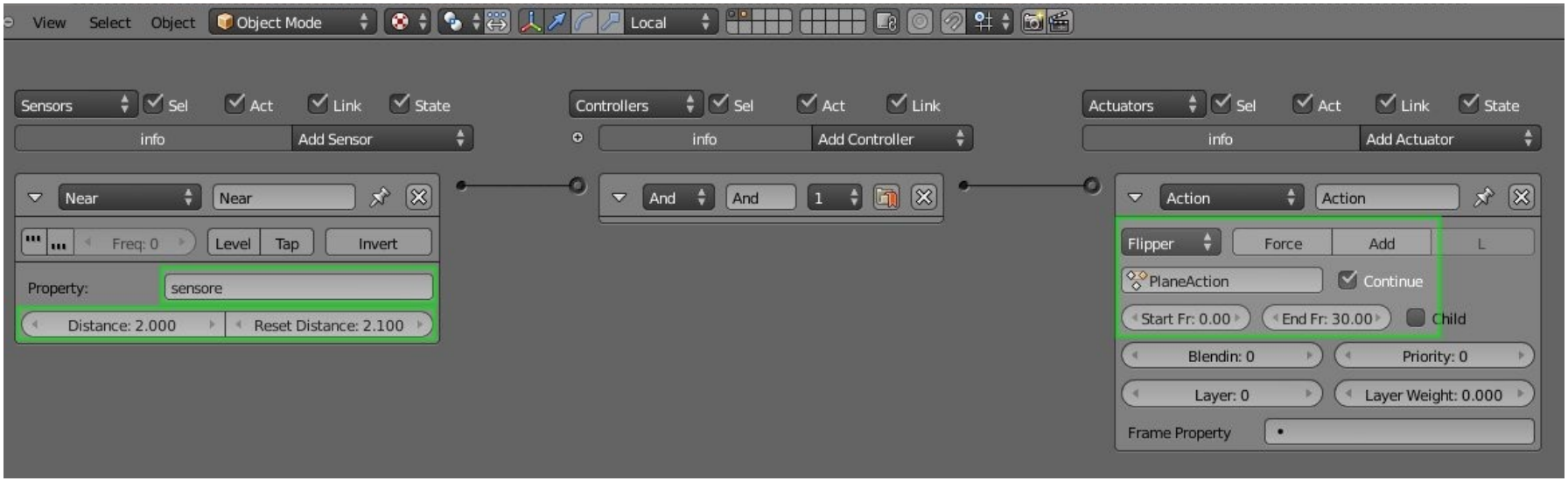
Select the “porta1” and add a **Near** Sensor, an **And** Controller and an **Action** Actuator.
Configure them as in the image



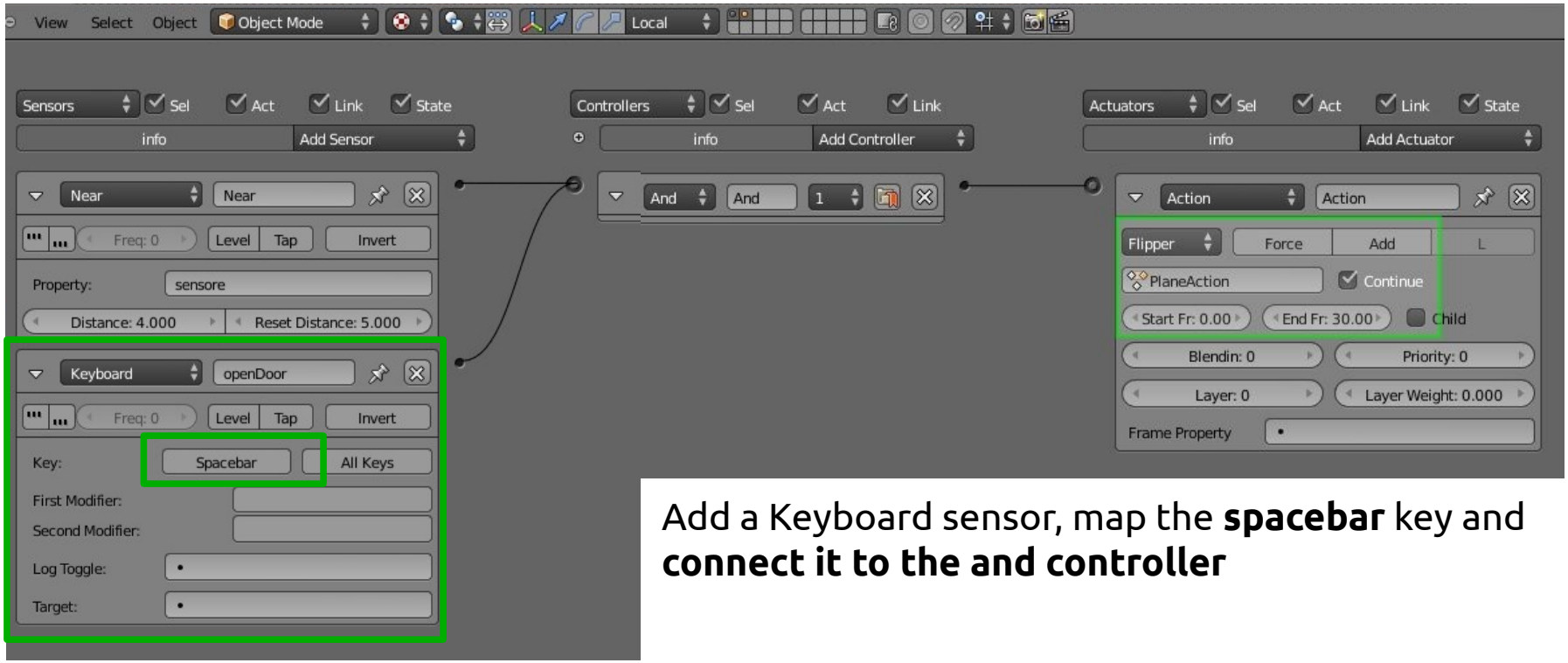
“sensore” in the Near Sensor, tells the door to look for the **cube proximity**

porta1Action is the animation of the door,
This animation starts in the **frame 0** and ends in the **frame 30**

Select the “porta2” and add a **Near** Sensor, an **And** Controller and an **Action** Actuator as before, configured as in the image

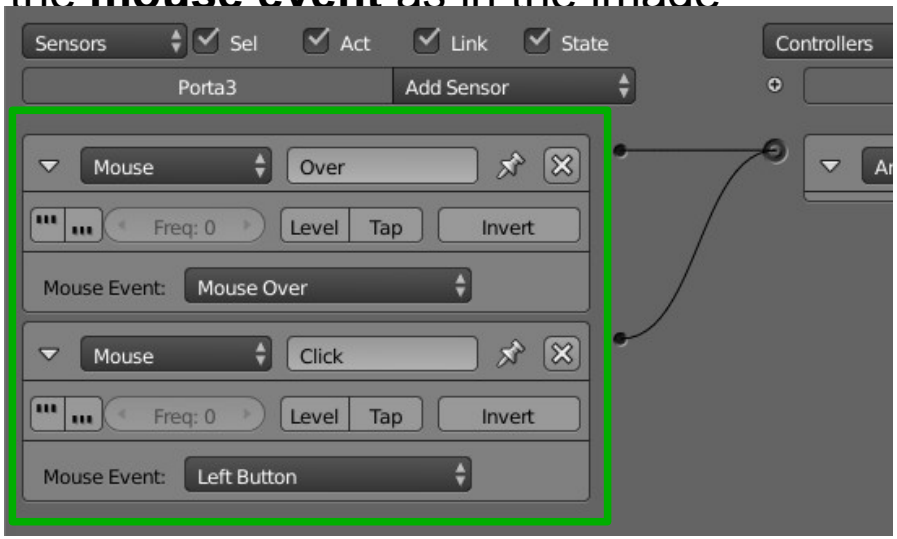


Select the “porta2” and add a **Near** Sensor, an **And** Controller and an **Action** Actuator as before, configured as in the image

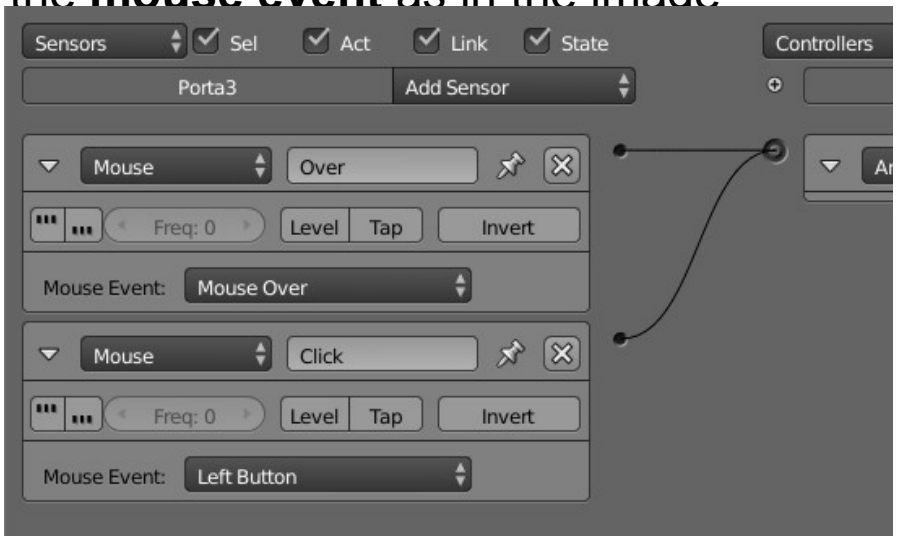


Add a Keyboard sensor, map the **spacebar** key and **connect it to the and controller**

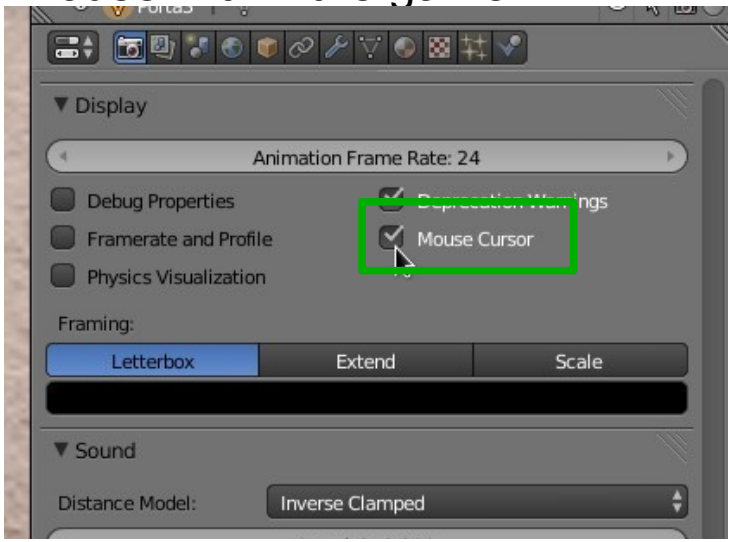
Select the “porta3” and add an **And** Controller and an **Action** Actuator as before, configured as in the image, now two M**ouse sensor** and configure the **mouse event** as in the image



Select the “porta3” and add an **And** Controller and an **Action** Actuator as before, configured as in the image, now two M**ouse sensor** and configure the **mouse event** as in the image



In the render properties, into the display panel, activate the **Mouse Cursor** to show the mouse within the game



Many thanks for your attention

Any questions?