

UPBGE

Camera follow

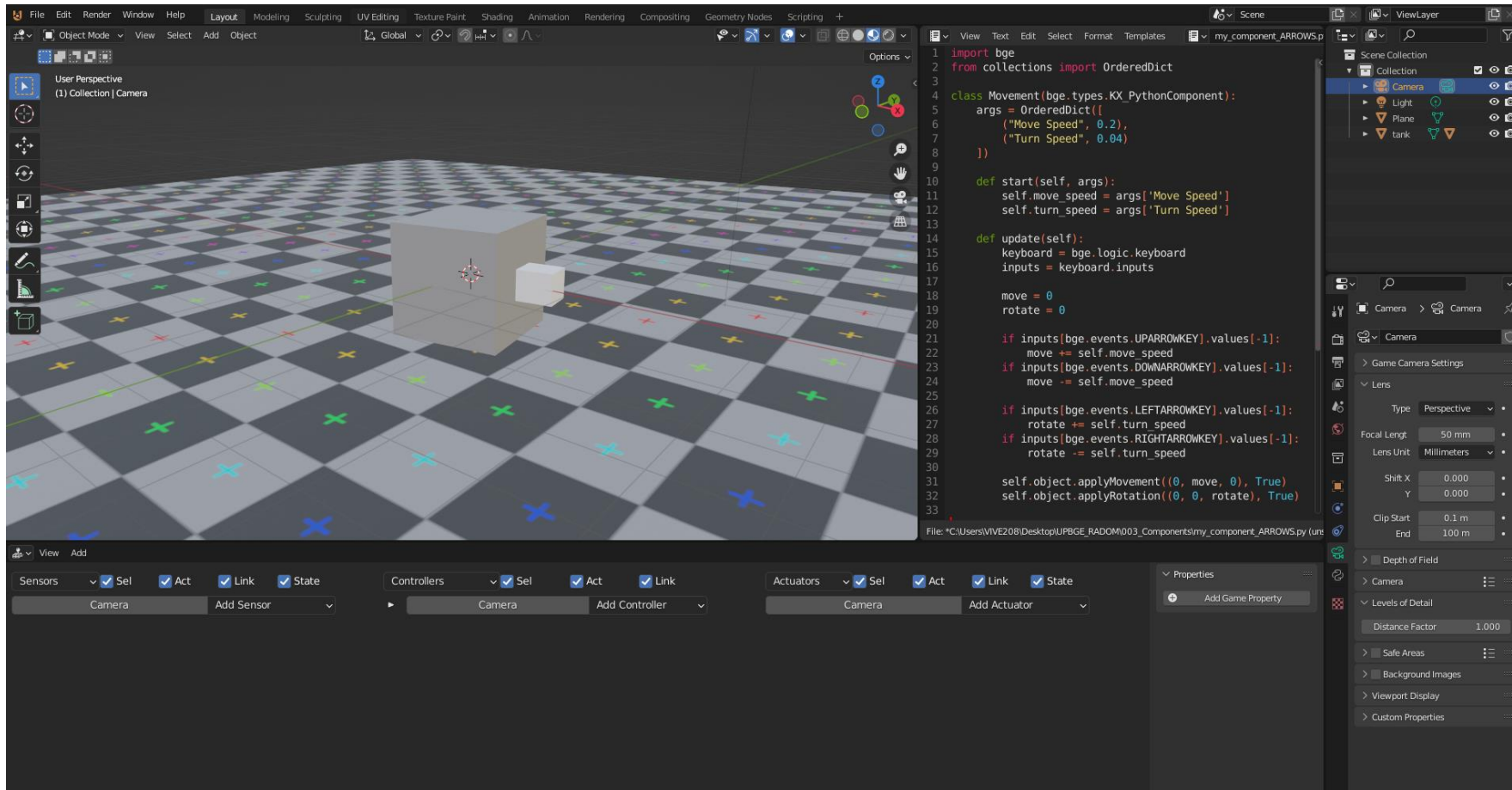


**Co-funded by
the European Union**



2024-1-PL01-KA220-VET-000243150

WE ARE OPENING AN PREVIOUS FILE

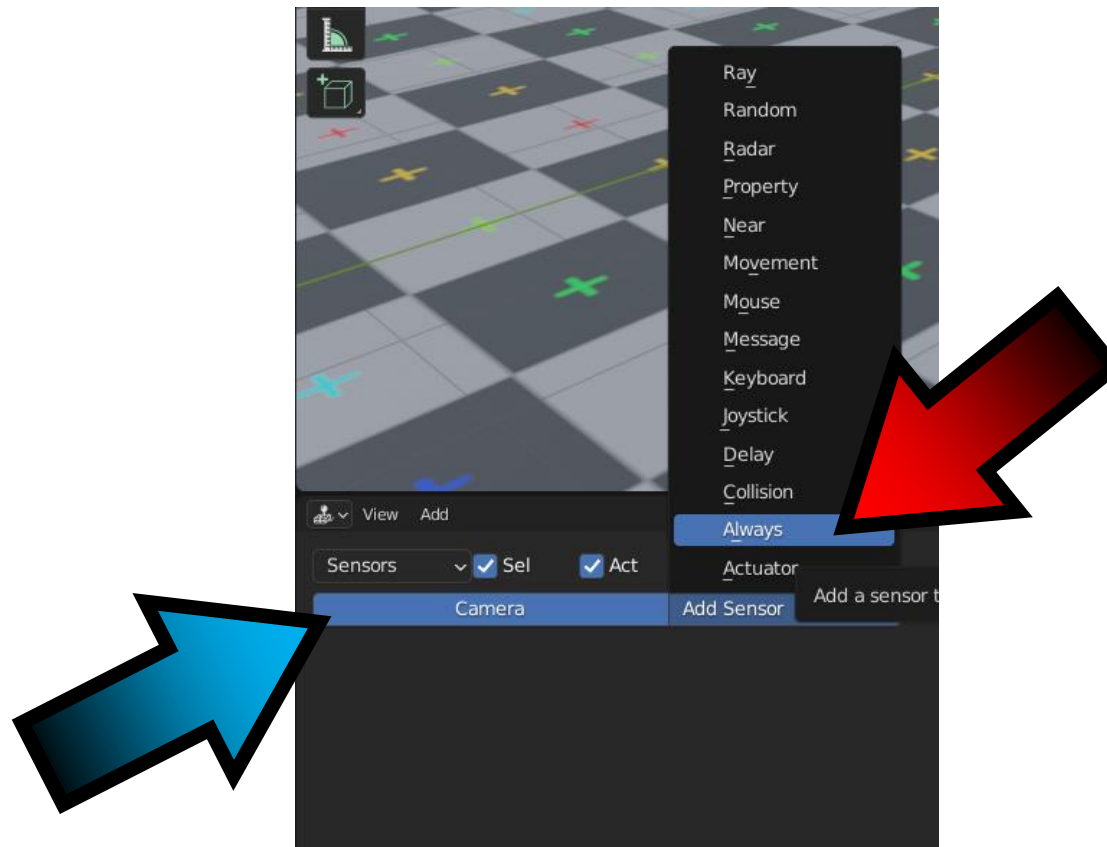


The screenshot displays the Blender 2.80 interface. The central 3D viewport shows a perspective view of a scene with a grey and white checkered floor. A large grey cube is positioned in the center, with a smaller white cube attached to its right side. The floor is marked with various colored 'x' symbols (yellow, green, cyan, blue, red). The interface includes a top menu bar, a left sidebar with toolshelves, a central code editor, and a right sidebar with the Outliner and Properties panels. The code editor contains the following Python code:

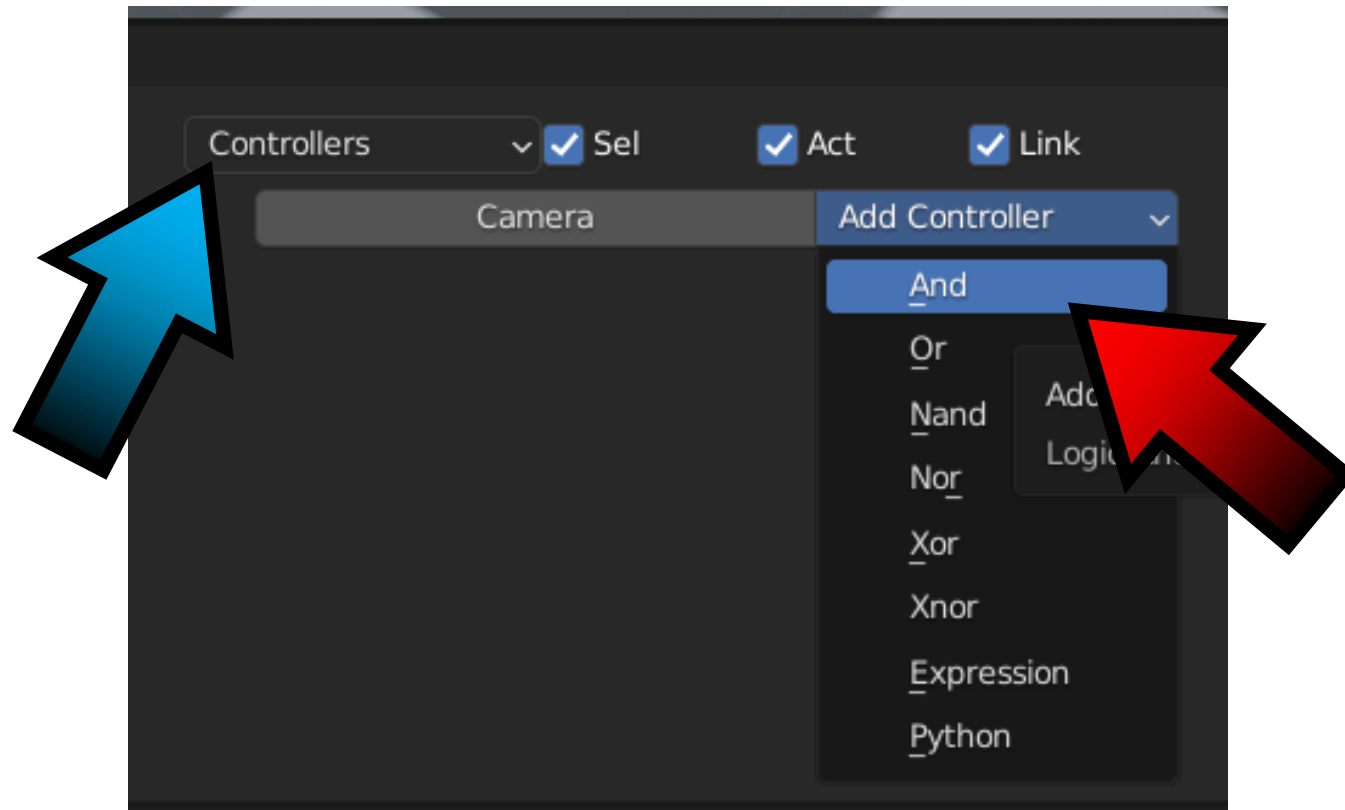
```
1 import bge
2 from collections import OrderedDict
3
4 class Movement(bge.types.KX_PythonComponent):
5     args = OrderedDict([
6         ("Move Speed", 0.2),
7         ("Turn Speed", 0.04)
8     ])
9
10 def start(self, args):
11     self.move_speed = args['Move Speed']
12     self.turn_speed = args['Turn Speed']
13
14 def update(self):
15     keyboard = bge.logic.keyboard
16     inputs = keyboard.inputs
17
18     move = 0
19     rotate = 0
20
21     if inputs[bge.events.UPARROWKEY].values[-1]:
22         move += self.move_speed
23     if inputs[bge.events.DOWNARROWKEY].values[-1]:
24         move -= self.move_speed
25
26     if inputs[bge.events.LEFTARROWKEY].values[-1]:
27         rotate += self.turn_speed
28     if inputs[bge.events.RIGHTARROWKEY].values[-1]:
29         rotate -= self.turn_speed
30
31     self.object.applyMovement((0, move, 0), True)
32     self.object.applyRotation((0, 0, rotate), True)
33
```

The Properties panel on the right shows the Camera settings, including Type (Perspective), Focal Length (50 mm), Lens Unit (Millimeters), Shift X (0.000), Y (0.000), Clip Start (0.1 m), and End (100 m). The bottom of the interface shows the Logic Editor with Sensors, Controllers, and Actuators for the Camera.

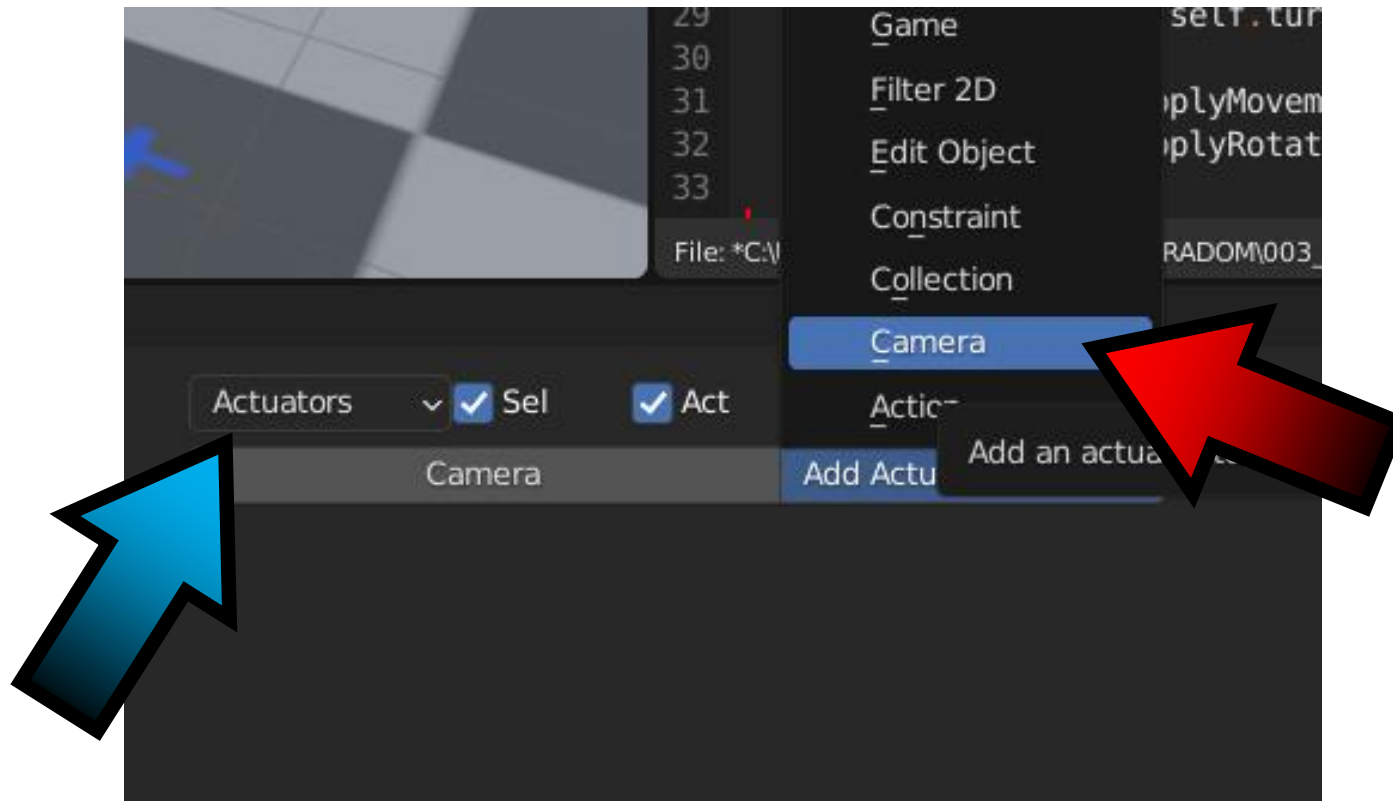
**FOR THE SELECTED CAMERA WE
CHOOSE SENSOR ALWAYS**



THEN FOR **CONTROLLERS** WE SELECT
AND



FOR ACTUATORS WE CHOOSE CAMERA





POWER OF AR AND VR



WE **CONNECT** AS BEFORE

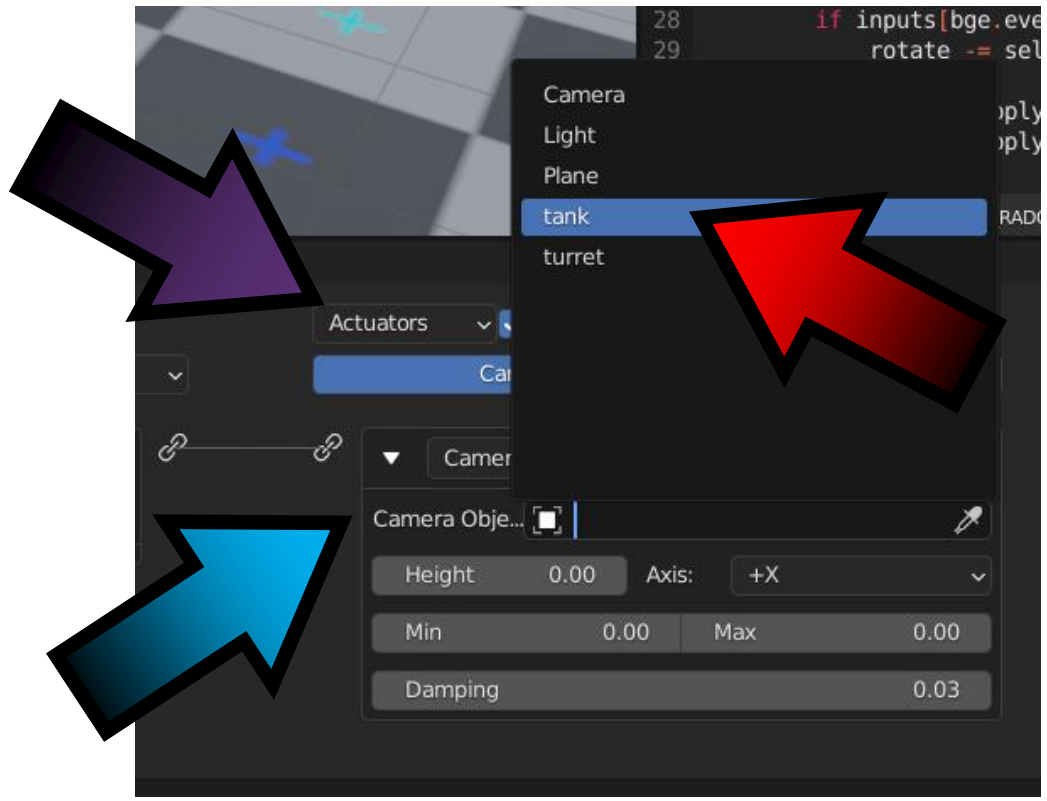
The screenshot displays a game engine's logic editor with three main sections: Sensors, Controllers, and Actuators. Each section has a 'Camera' component selected. The Sensors section includes a dropdown menu with 'Always' selected and a 'Link' checkbox checked. The Controllers section includes a dropdown menu with 'And' selected and a 'Link' checkbox checked. The Actuators section includes a dropdown menu with 'Camera' selected and a 'Link' checkbox checked. Two large red arrows point upwards from the bottom towards the connection points between the Camera sensor and the Camera controller, and between the Camera controller and the Camera actuator.

UPBGE



POWER OF AR AND VR

IN ACTUATORS FOR CAMERA OBJECT WE CHOOSE TANK





POWER OF AR AND VR



WE SET THE CAMERA VIEW

The screenshot displays the Blender 2.80 interface. The central 3D Viewport shows a perspective view of a simple scene with a grey and white checkered floor plane and a small camera object positioned above it. A red arrow points from the bottom left towards the camera object. The right side of the interface features a Text Editor window with Python code for a custom movement class. The bottom of the interface shows the Logic Editor with a sequence of three bricks: a Sensor, a Controller, and an Actuator, all named 'Camera'.

```
1 import bge
2 from collections import OrderedDict
3
4 class Movement(bge.types.Klass):
5     args = OrderedDict({
6         ("Move Speed", 0.2),
7         ("Turn Speed", 0.04)
8     })
9
10    def start(self, args):
11        self.move_speed = args["Move Speed"]
12        self.turn_speed = args["Turn Speed"]
13
14    def update(self):
15        keyboard = bge.logic.keyboard
16        inputs = keyboard.inputs
17
18        move = 0
19        rotate = 0
20
21        if inputs[bge.events.LEFT]:
22            move += self.move_speed
23        if inputs[bge.events.RIGHT]:
24            move -= self.move_speed
25
26        if inputs[bge.events.UP]:
27            rotate += self.turn_speed
28        if inputs[bge.events.DOWN]:
29            rotate -= self.turn_speed
30
31        self.object.applyMovement(self.move_speed * move, self.turn_speed * rotate)
32        self.object.applyRotation(rotate)
33
```

Sensors: Camera (Always) - Always - Skip 0 - Level Tap - Invert

Controllers: Camera (And) - And - Controller visible at: State 1

Actuators: Camera (Camera) - Camera - Camera Object: tank - Height 0.00 - Axis: +X - Min 0.00 - Max 0.00 - Damping 0.03

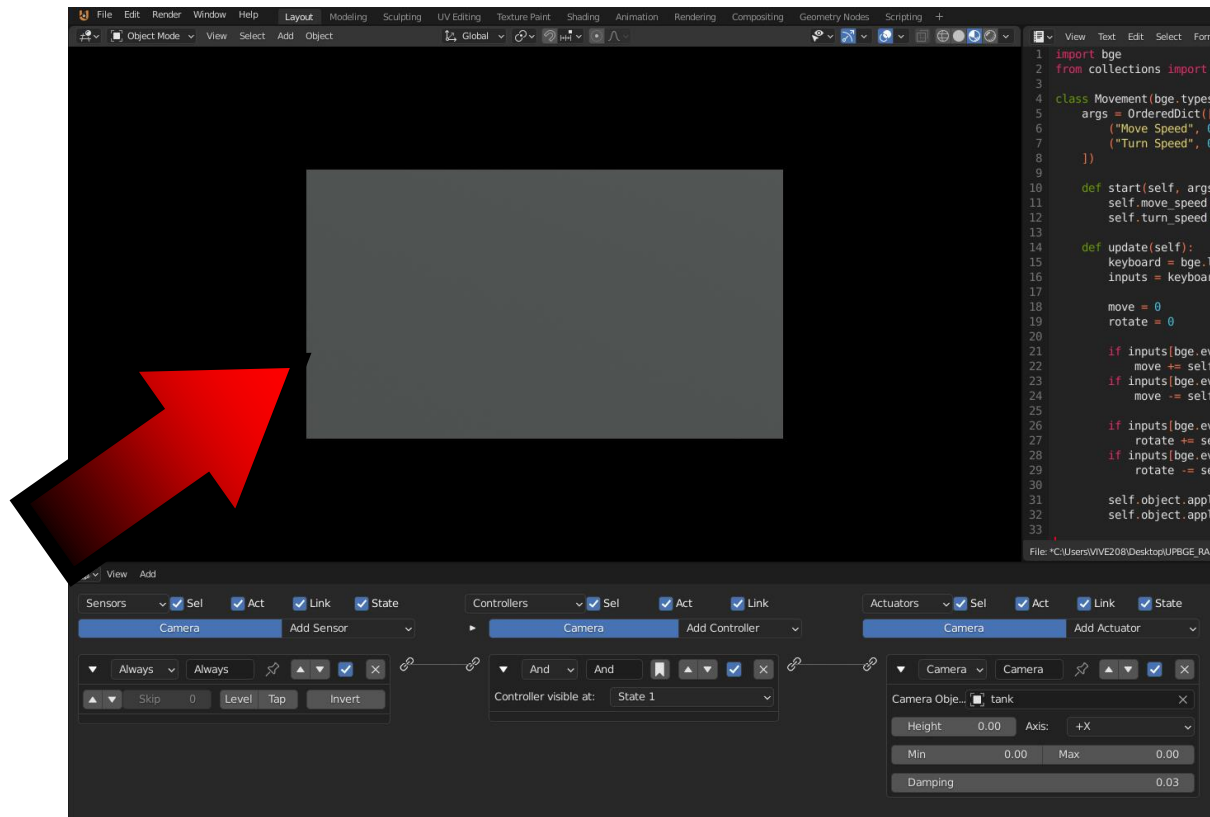
UPBGE



POWER OF AR AND VR

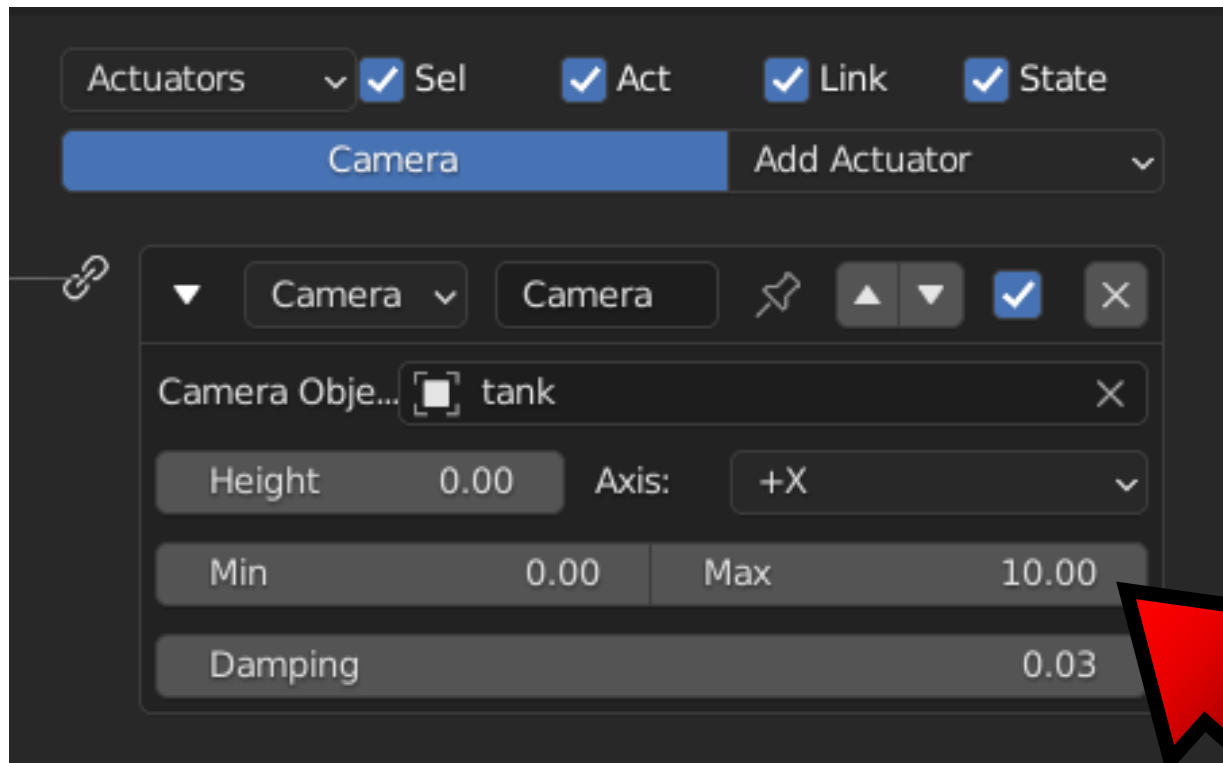


WE'RE STARTING THE GAME
WE'RE INSIDE THE TANK



UPBGE

LET'S CHANGE **MAX TO 10**



Actuators Sel Act Link State

Camera Add Actuator

Camera Camera

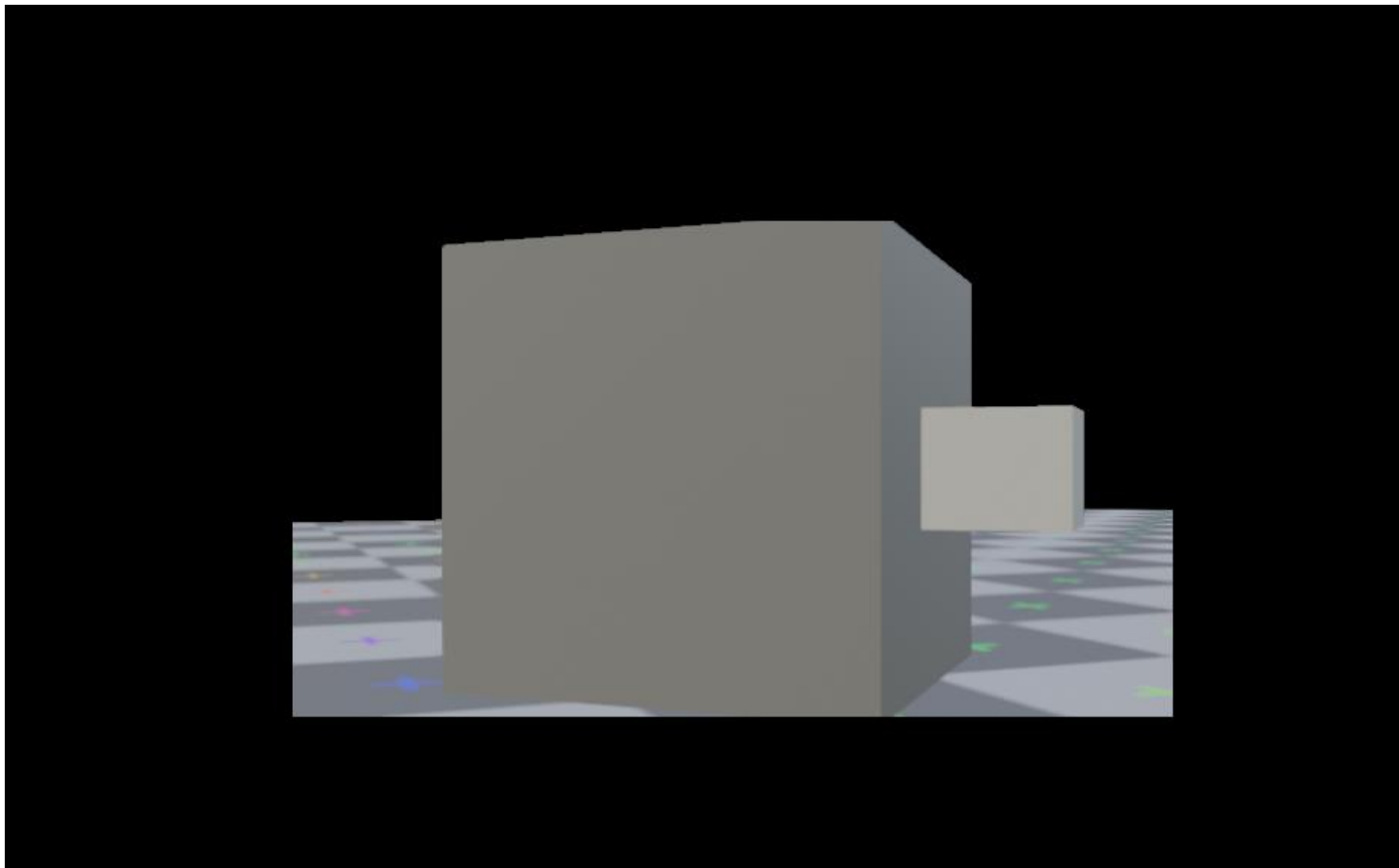
Camera Obj... tank

Height 0.00 Axis: +X

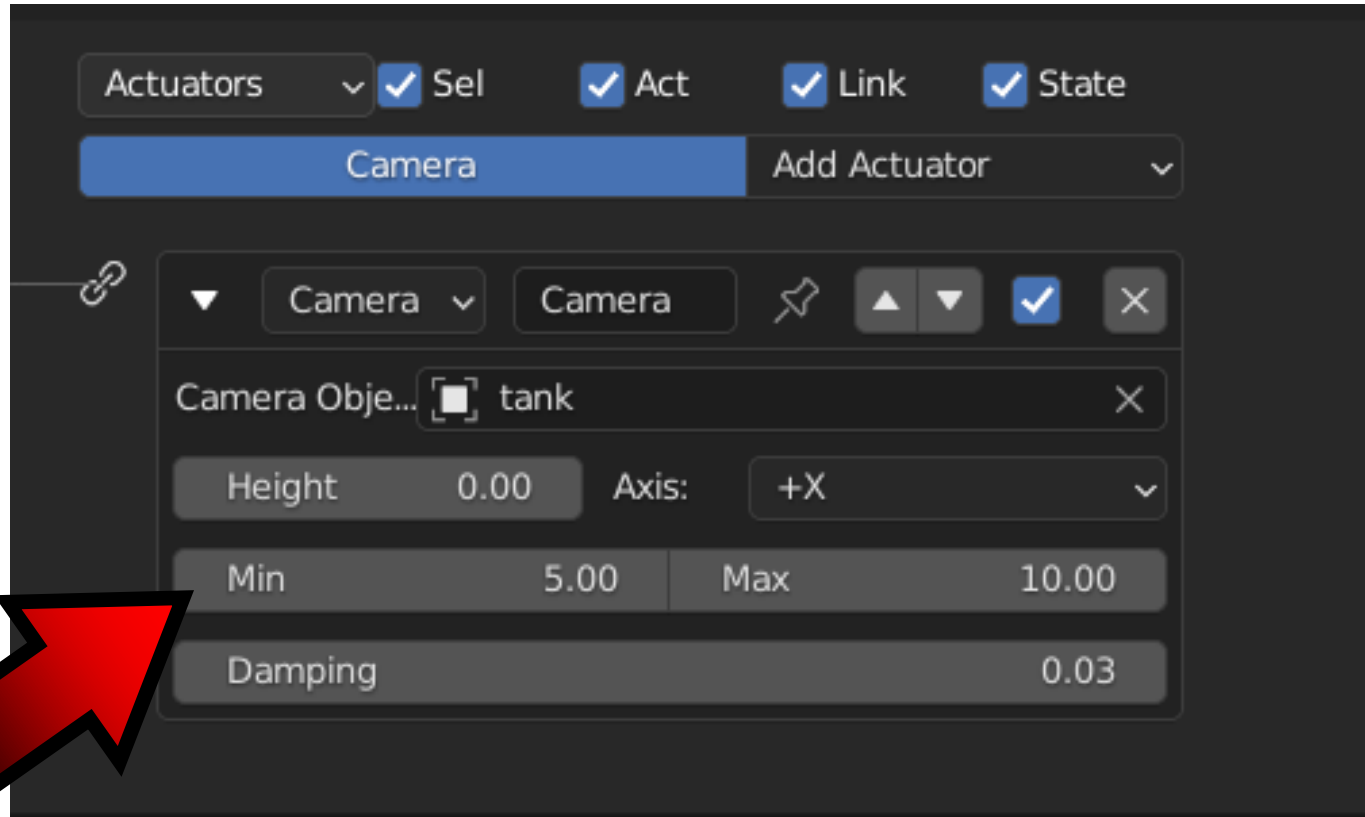
Min 0.00 Max 10.00

Damping 0.03

WE ARE STARTING THE GAME



SET MIN TO 5



Actuators Sel Act Link State

Camera Add Actuator

Camera Camera

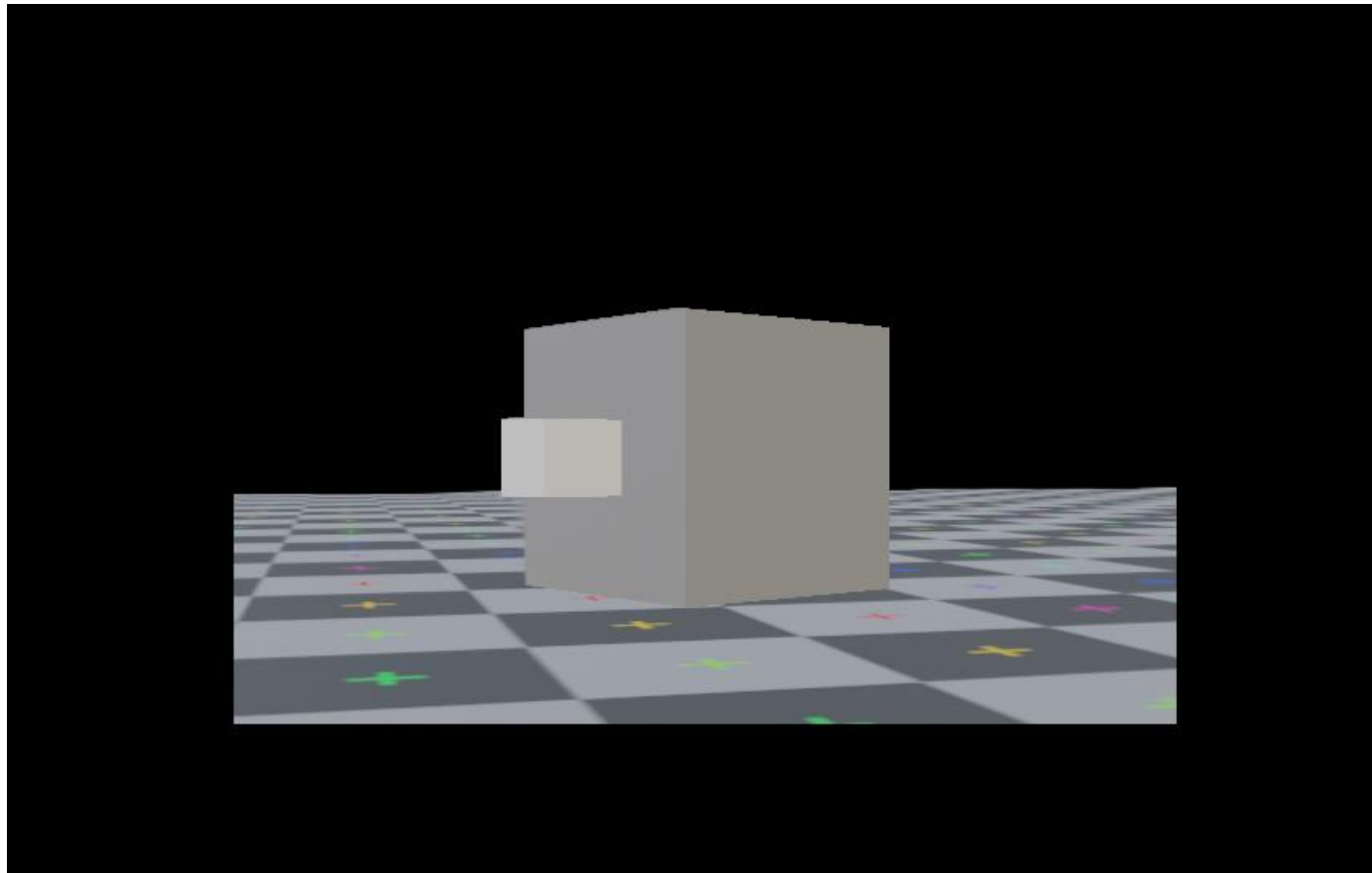
Camera Obj... tank

Height 0.00 Axis: +X

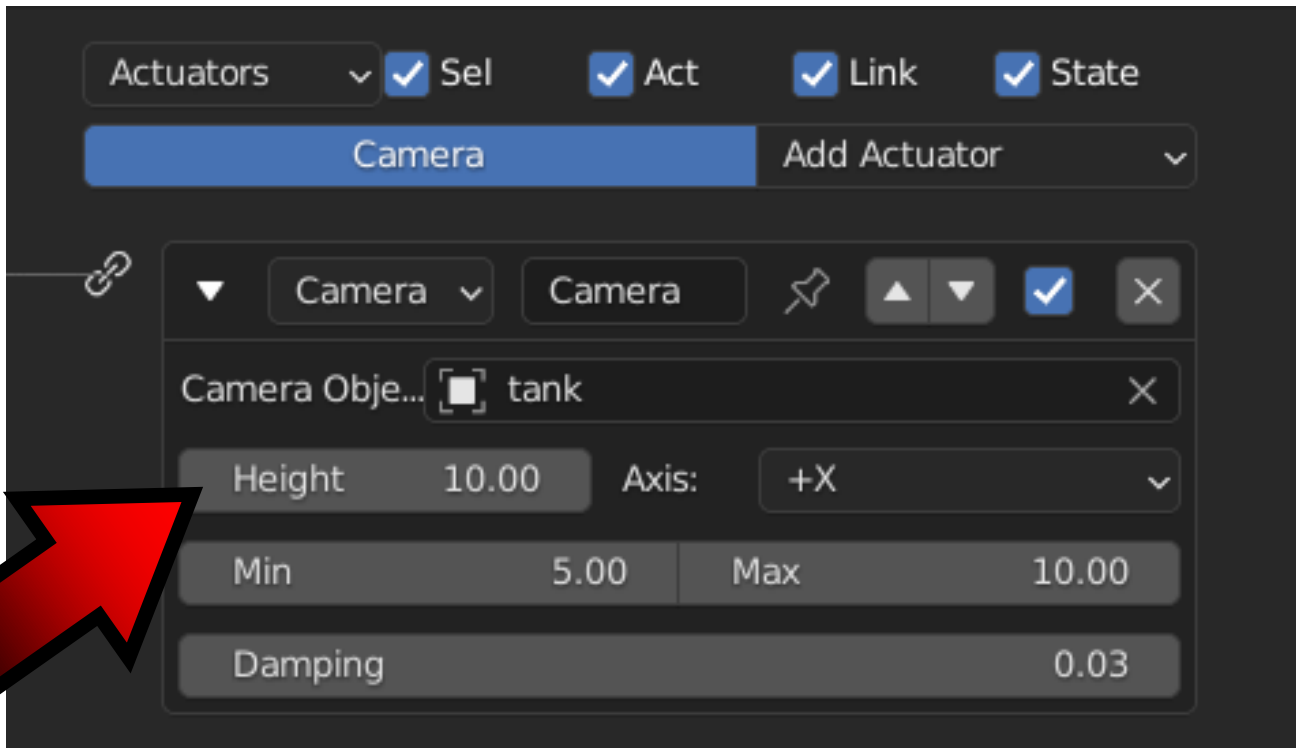
Min 5.00 Max 10.00

Damping 0.03

WE ARE STARTING THE GAME

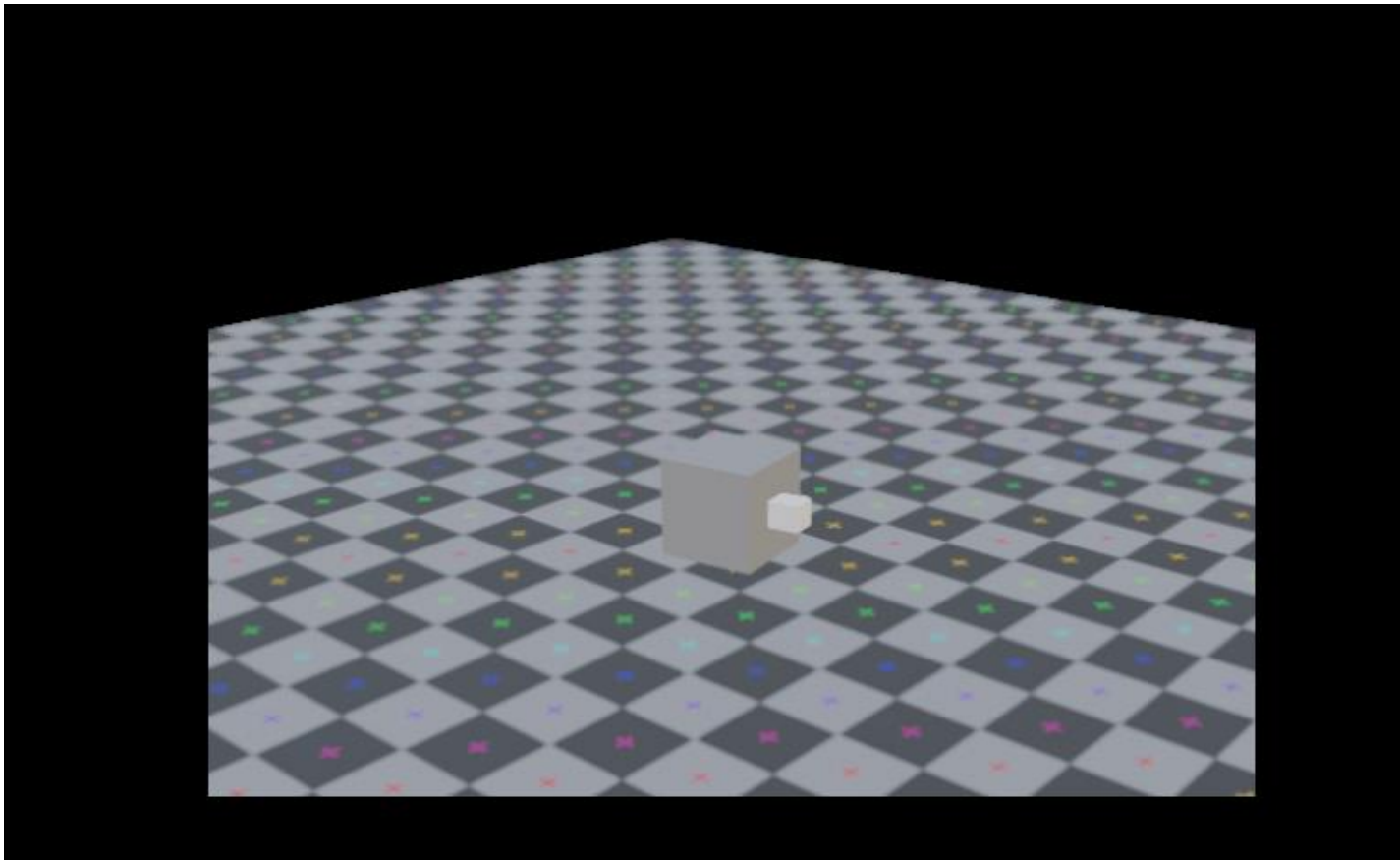


SET HEIGHT TO 10

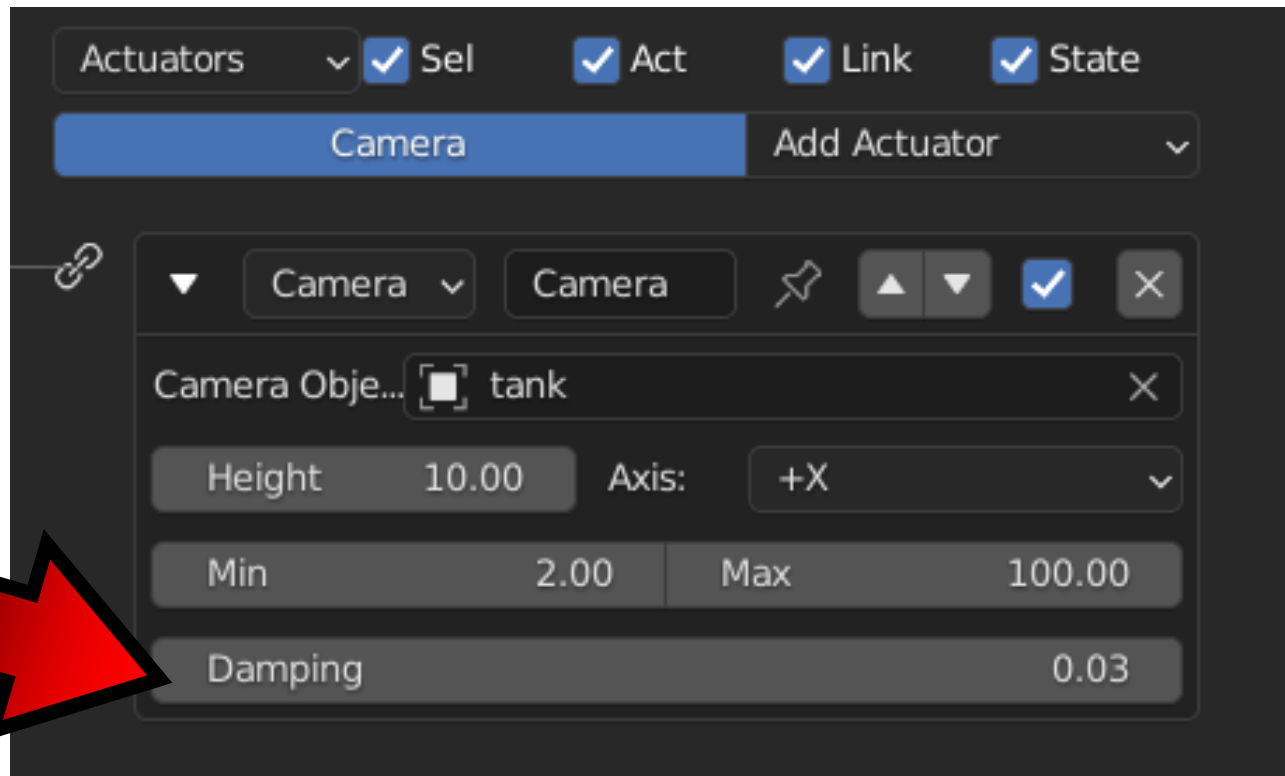


The screenshot shows a software interface for configuring a camera actuator. At the top, there are four checked checkboxes: 'Sel', 'Act', 'Link', and 'State'. Below them is a dropdown menu set to 'Camera' and an 'Add Actuator' button. The main configuration area includes a link icon, a dropdown set to 'Camera', a text field containing 'Camera', and several control icons. Below this, there is a 'Camera Object' field with a small square icon and the text 'tank'. The 'Height' field is set to '10.00' and is highlighted by a red arrow. The 'Axis' dropdown is set to '+X'. Below the 'Height' field, there are two rows of fields: 'Min' (5.00) and 'Max' (10.00), and 'Damping' (0.03).

WE ARE STARTING THE GAME



PARAMETER DAMPING IS USED FOR DAMPING



Actuators Sel Act Link State

Camera Add Actuator

Camera Camera

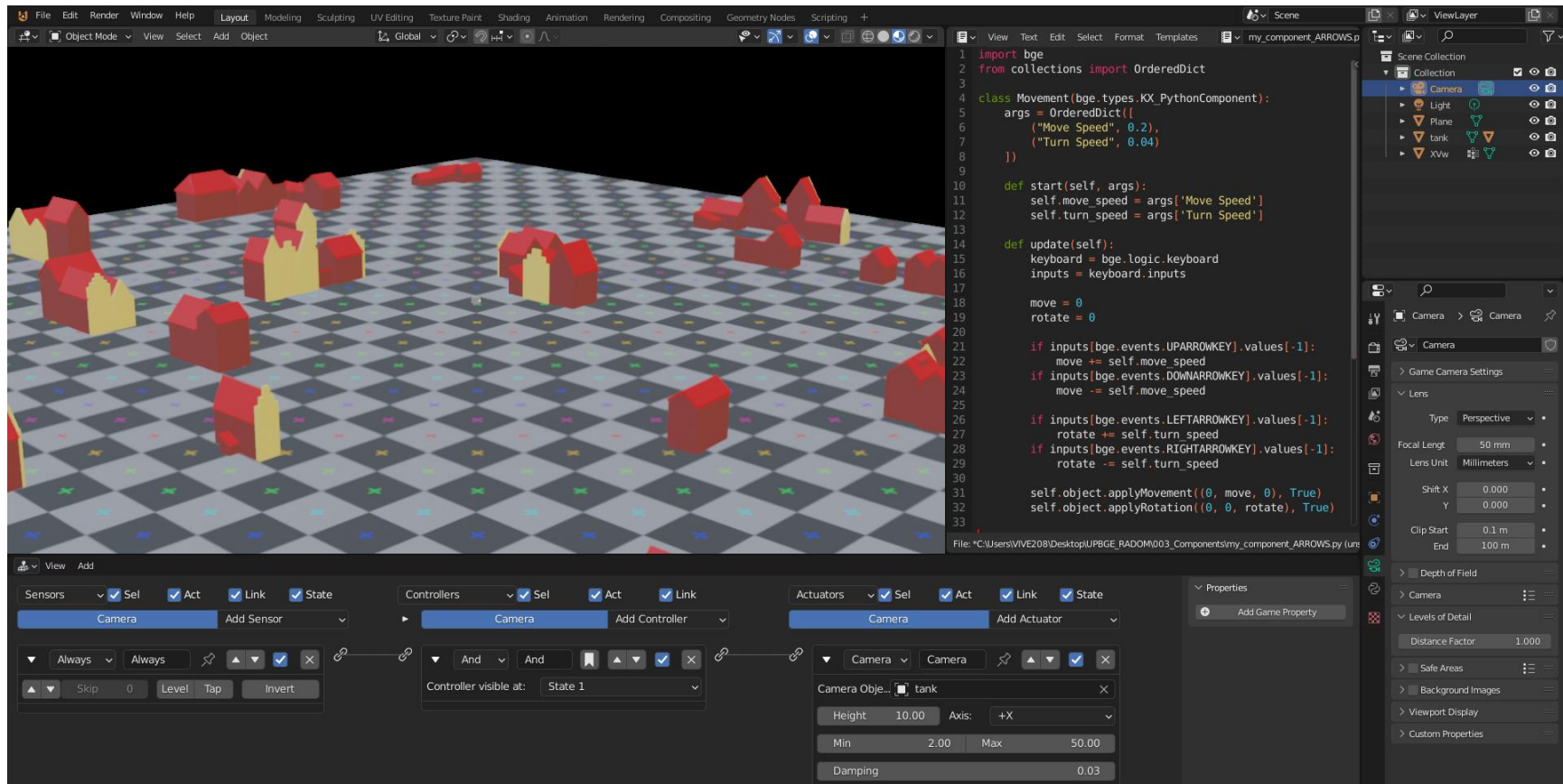
Camera Obj... tank

Height 10.00 Axis: +X

Min 2.00 Max 100.00

Damping 0.03

THIS IS HOW YOU CAN TEST DIFFERENT PROJECTS



POWER OF AR AND VR

**THANK YOU FOR
YOUR ATTENTION**



**Co-funded by
the European Union**



2024-1-PL01-KA220-VET-000243150

JACEK KAWAŁEK