

POWER OF AR AND VR

UPBGE

Komponenty



**Co-funded by
the European Union**

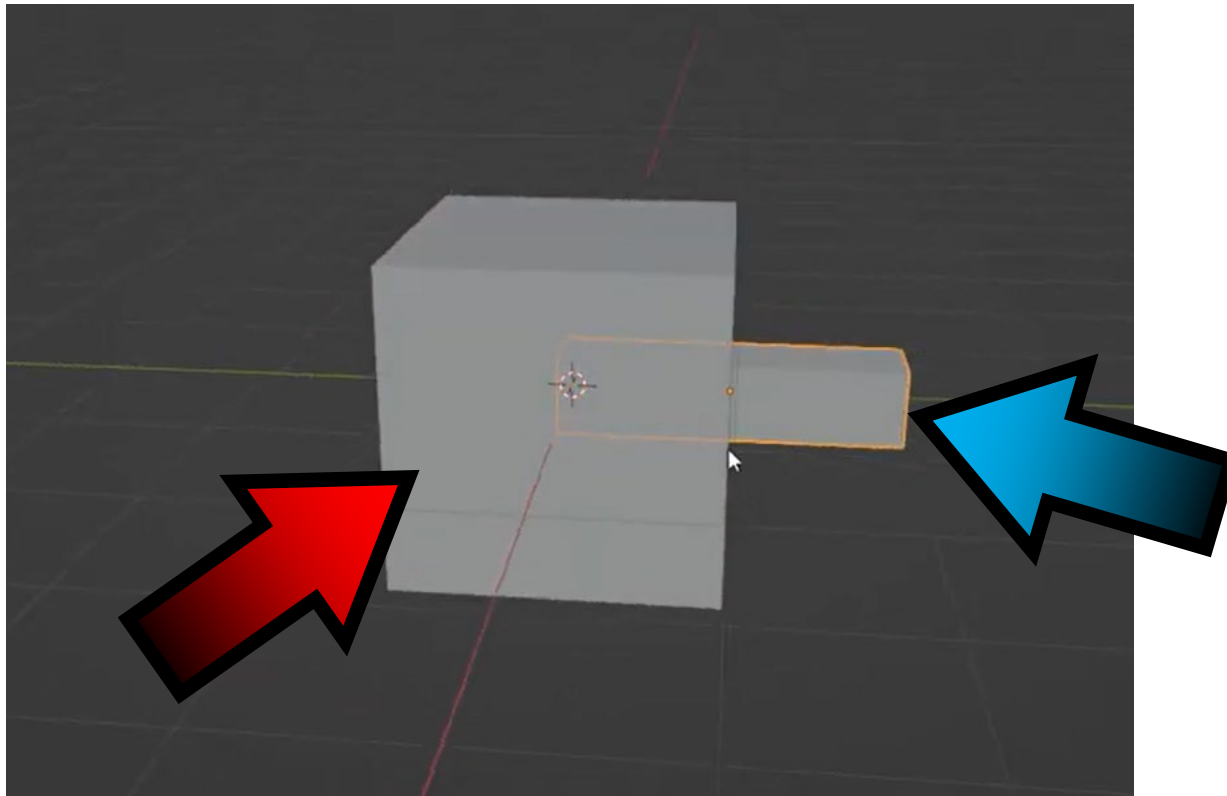


2024-1-PL01-KA220-VET-000243150

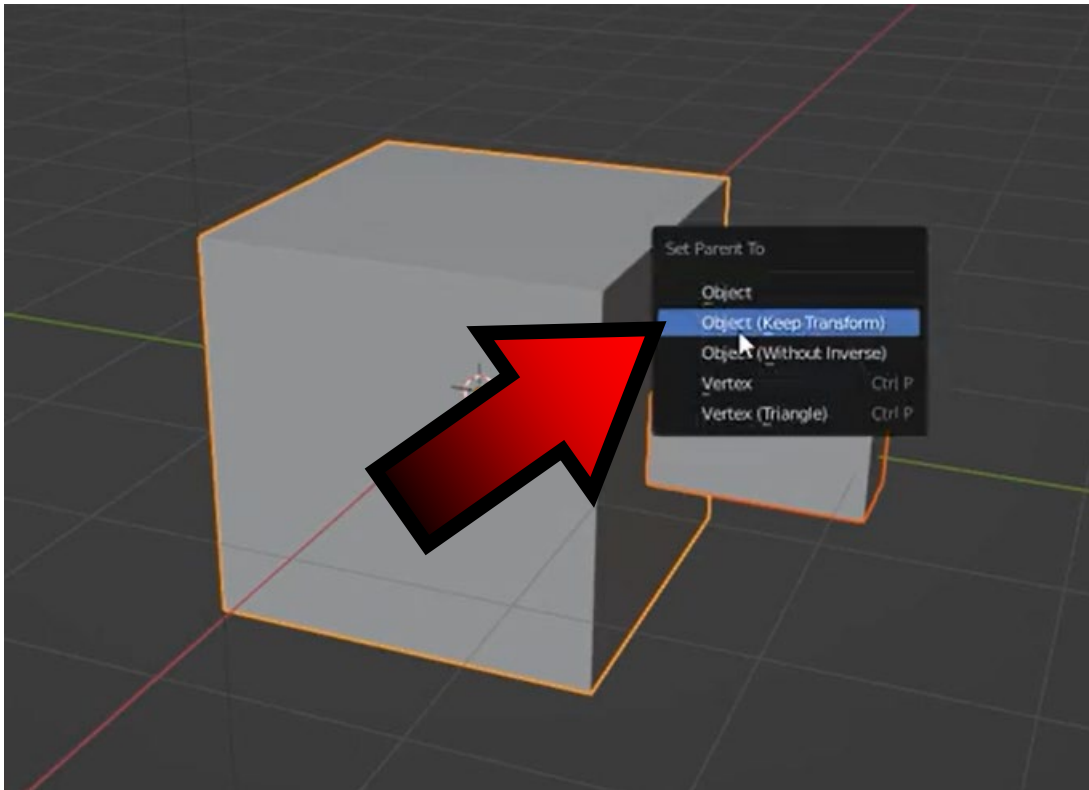
the European Union
co-funded by

2024-1-PL01-KA220-VET-000243150

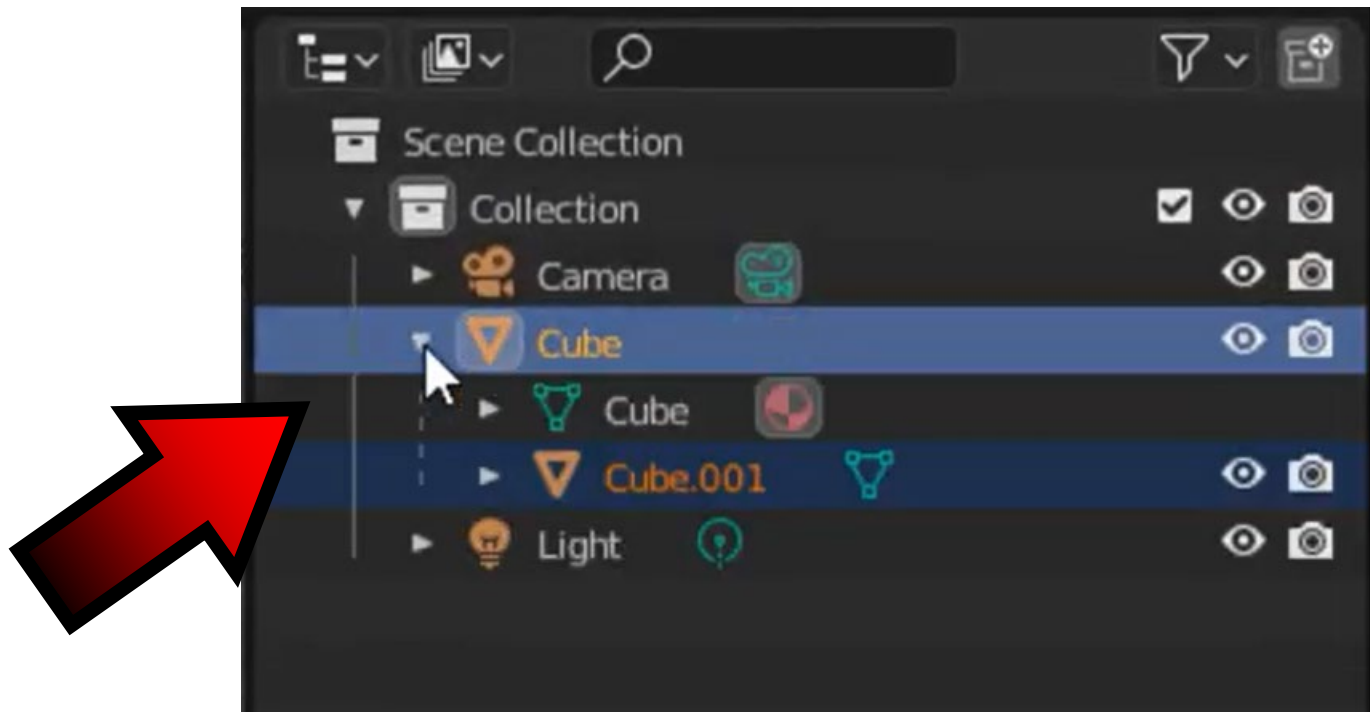
STWÓRZ TAKI MODEL



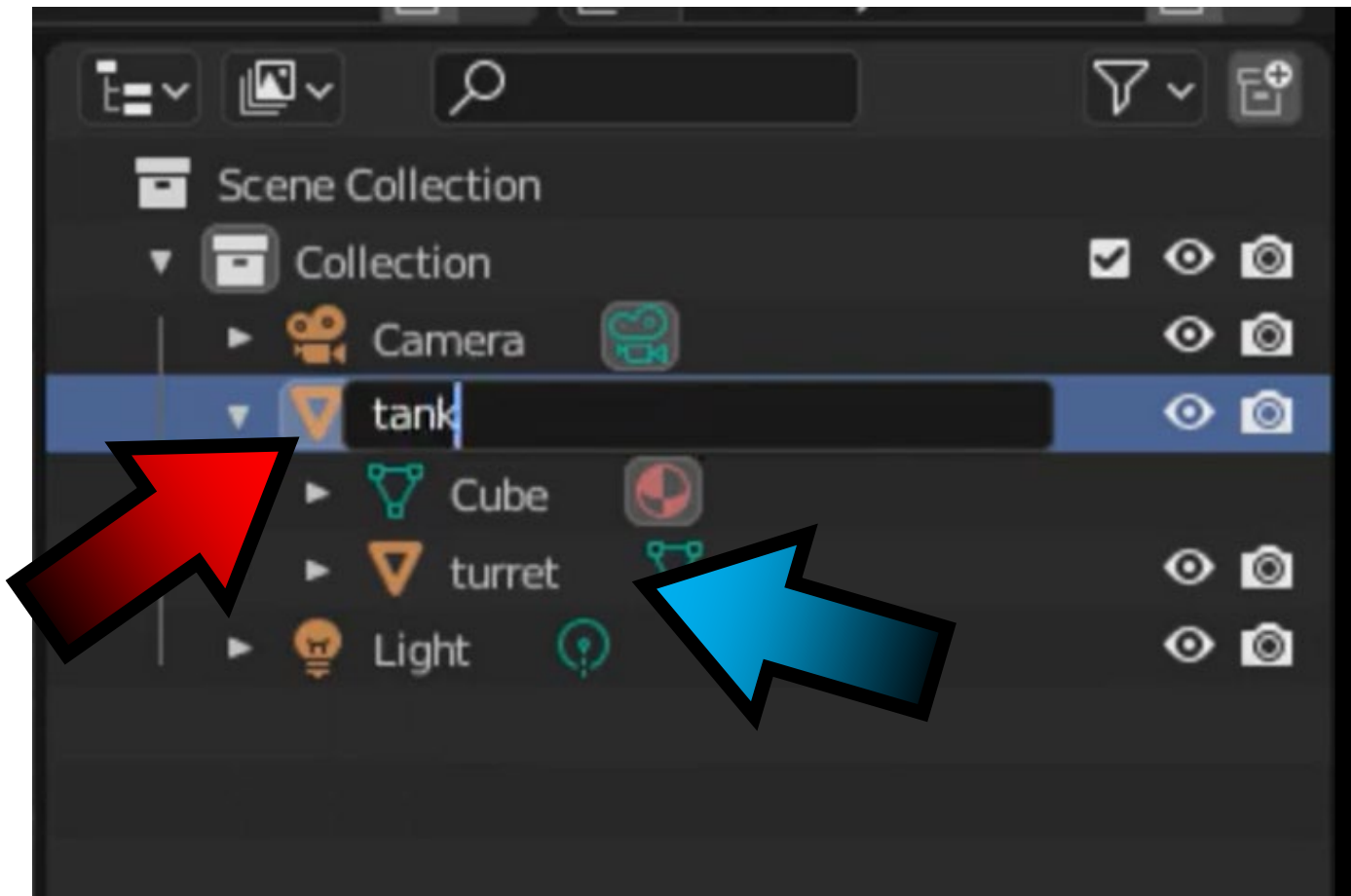
POŁĄCZ ZA POMOCĄ **CTRL+P**



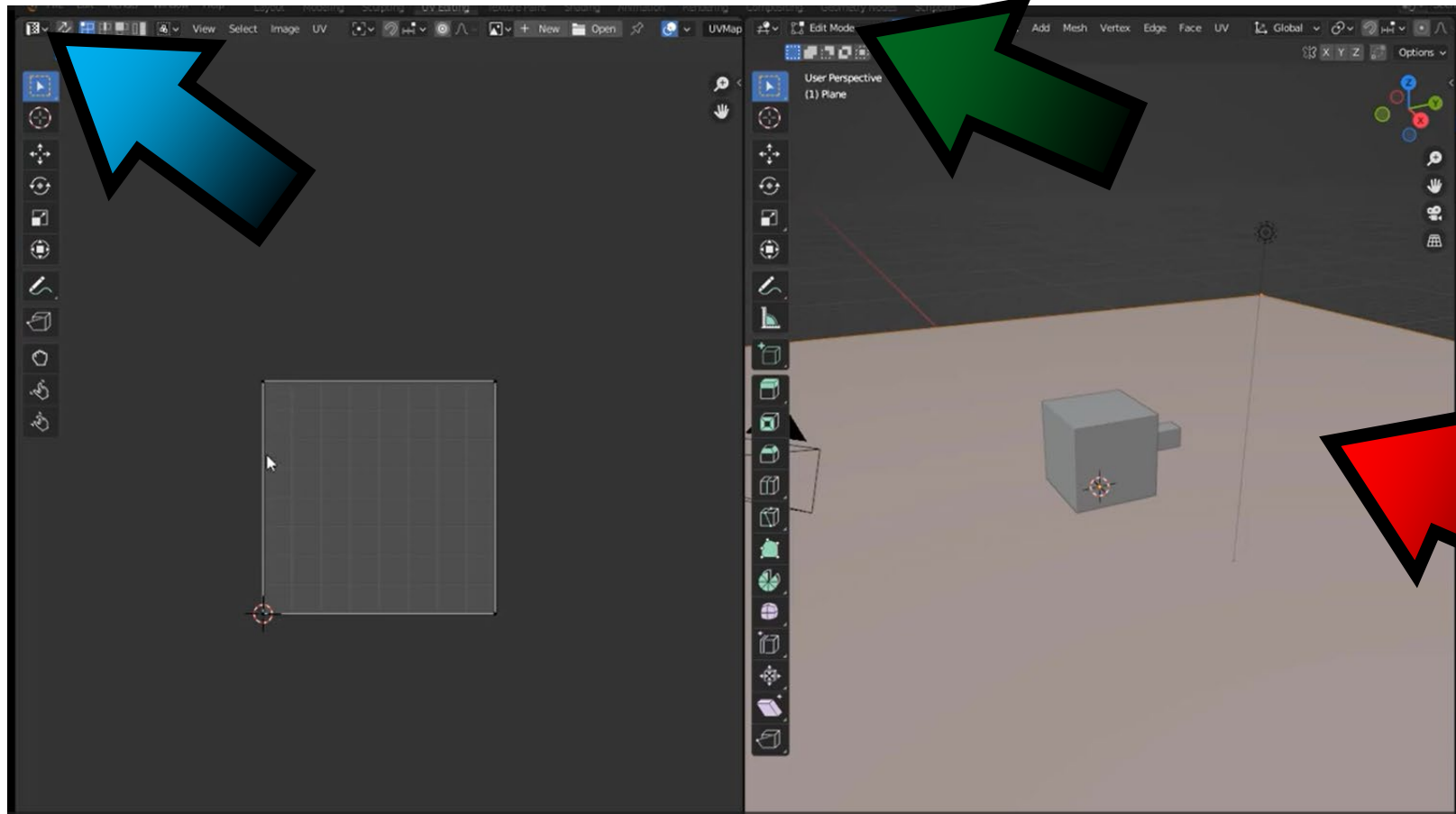
WEJDŹ DO **OUTLINER**



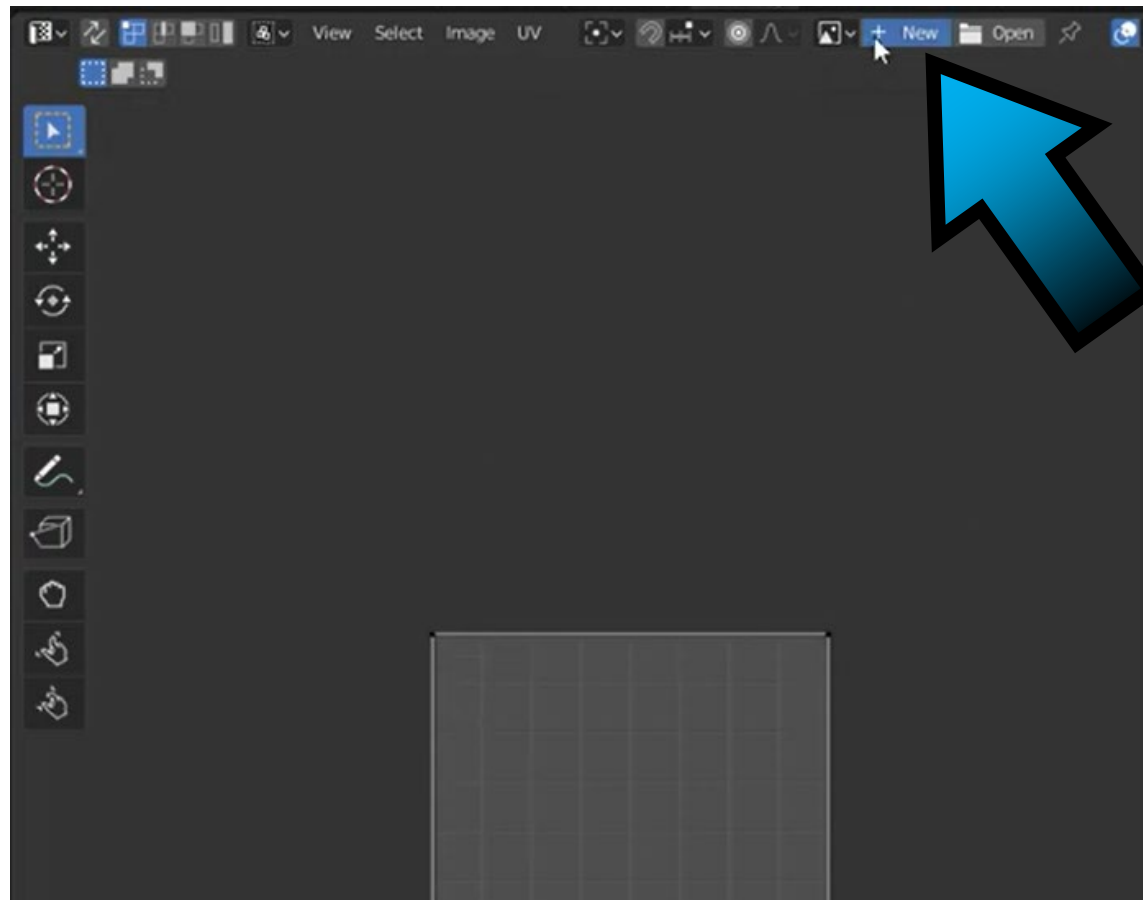
ZMIENŃ NAZWĘ



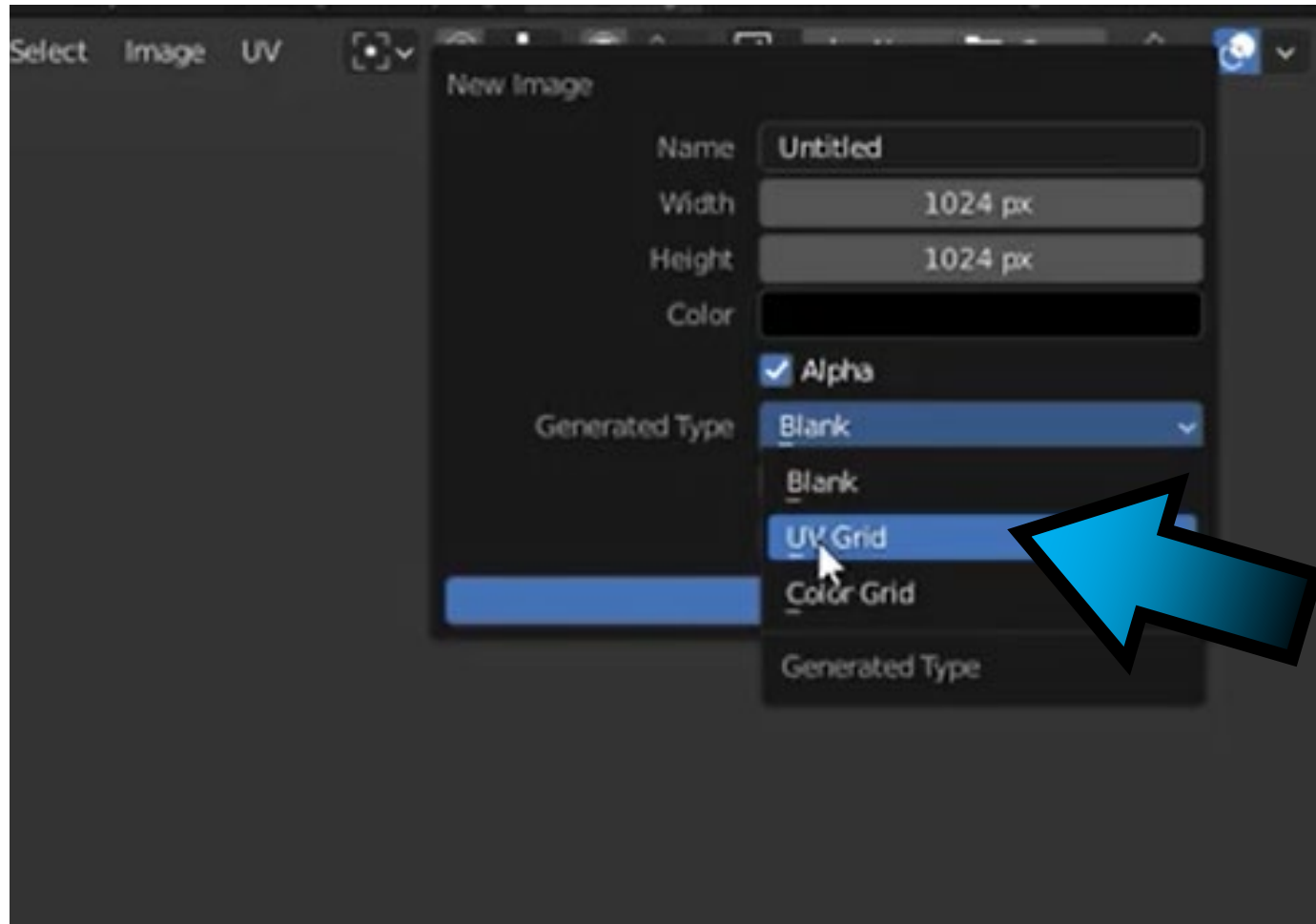
WSTAW **PLANE** I WEJDŹ DO **EDIT MODE** PODZIEL OKNO I WYBIERZ **UV EDITOR**



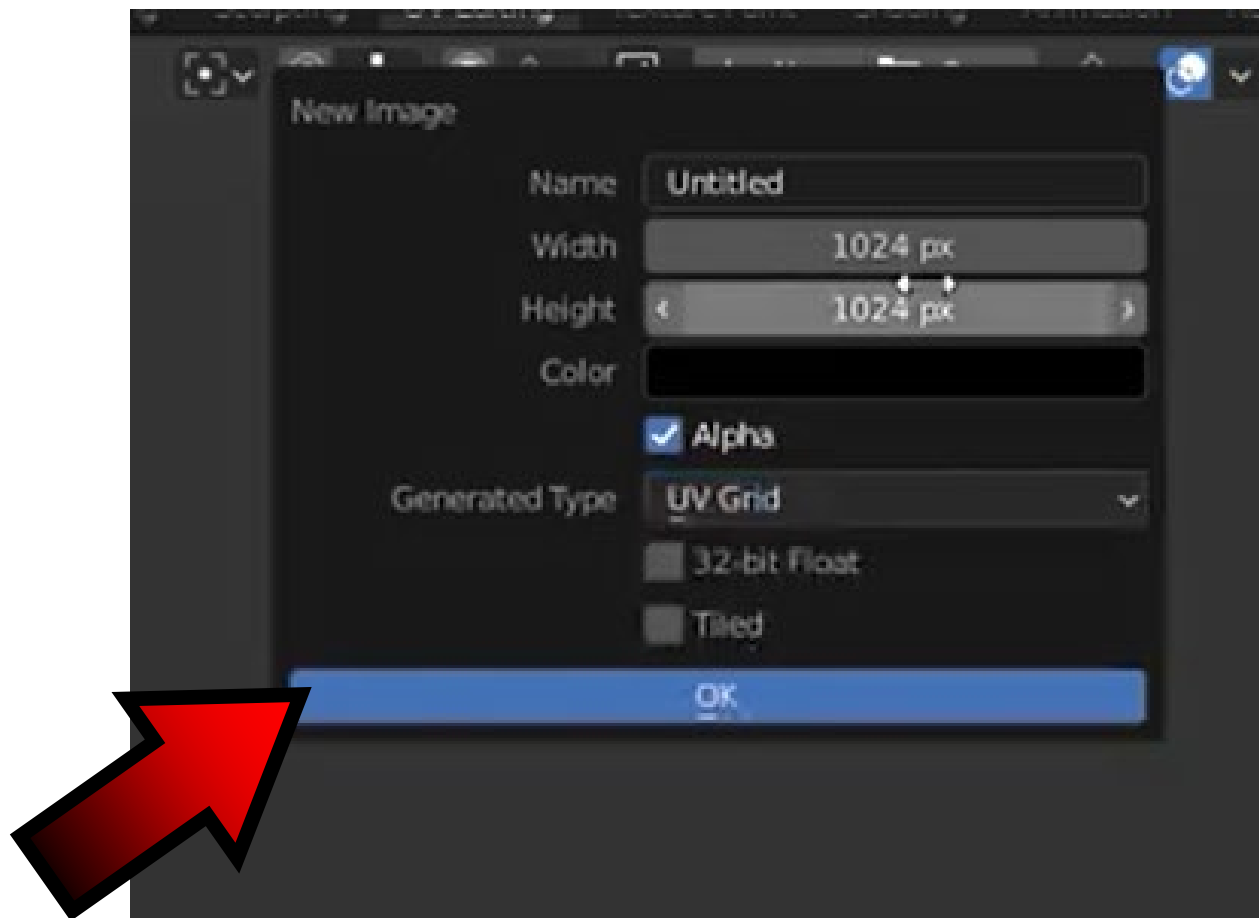
KLIKNIJ NA **NEW**



WYBIERZ UV GRID

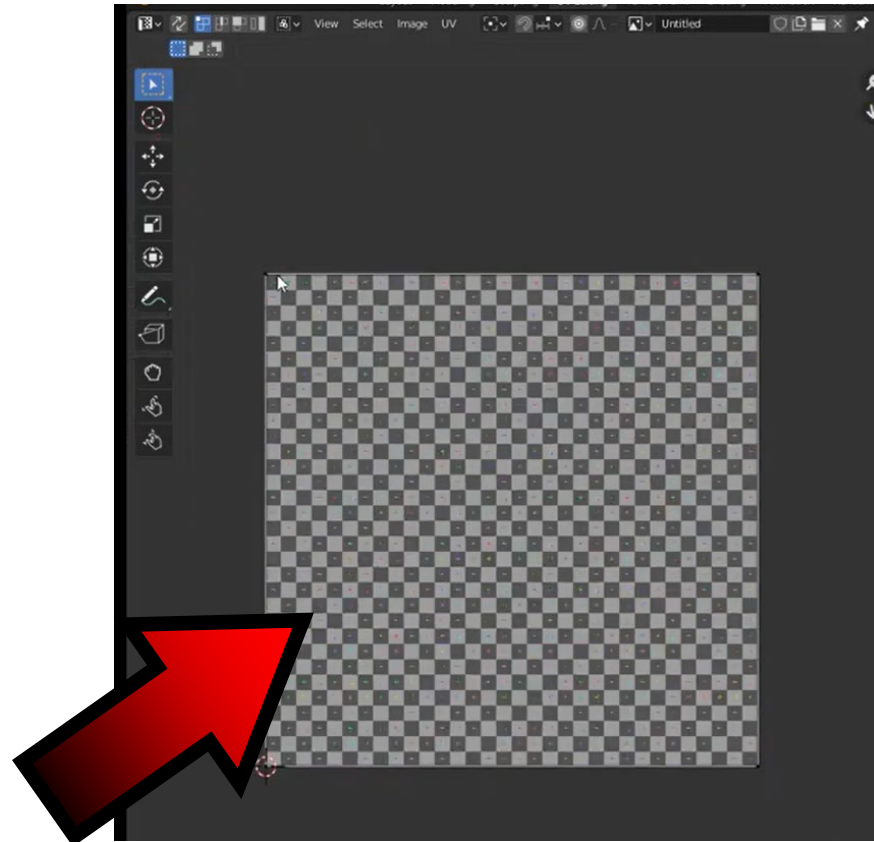


ZATWIERDŹ **OK**



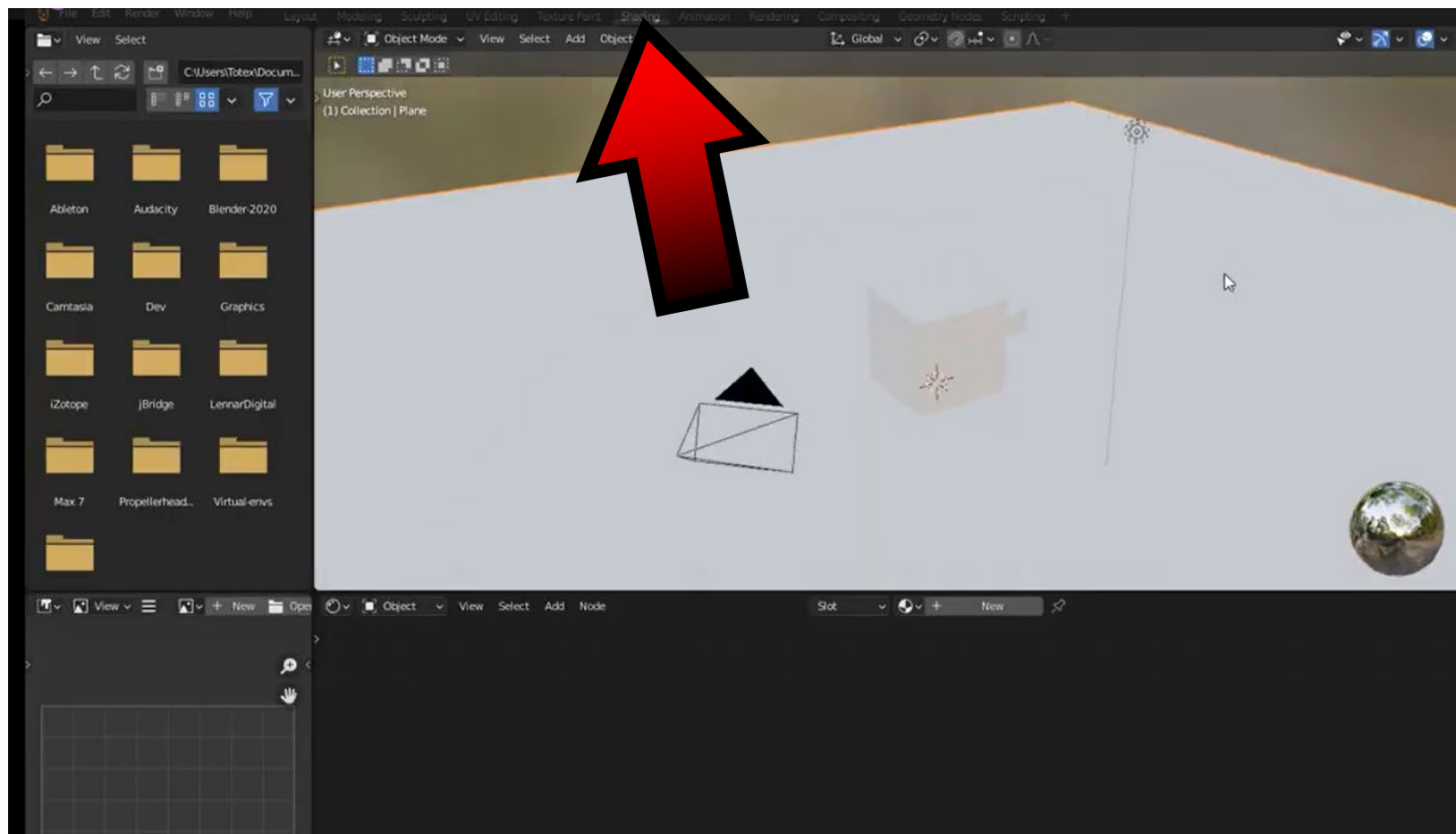
POWER OF AR AND VR

WIDOK PLANE W UV EDITOR

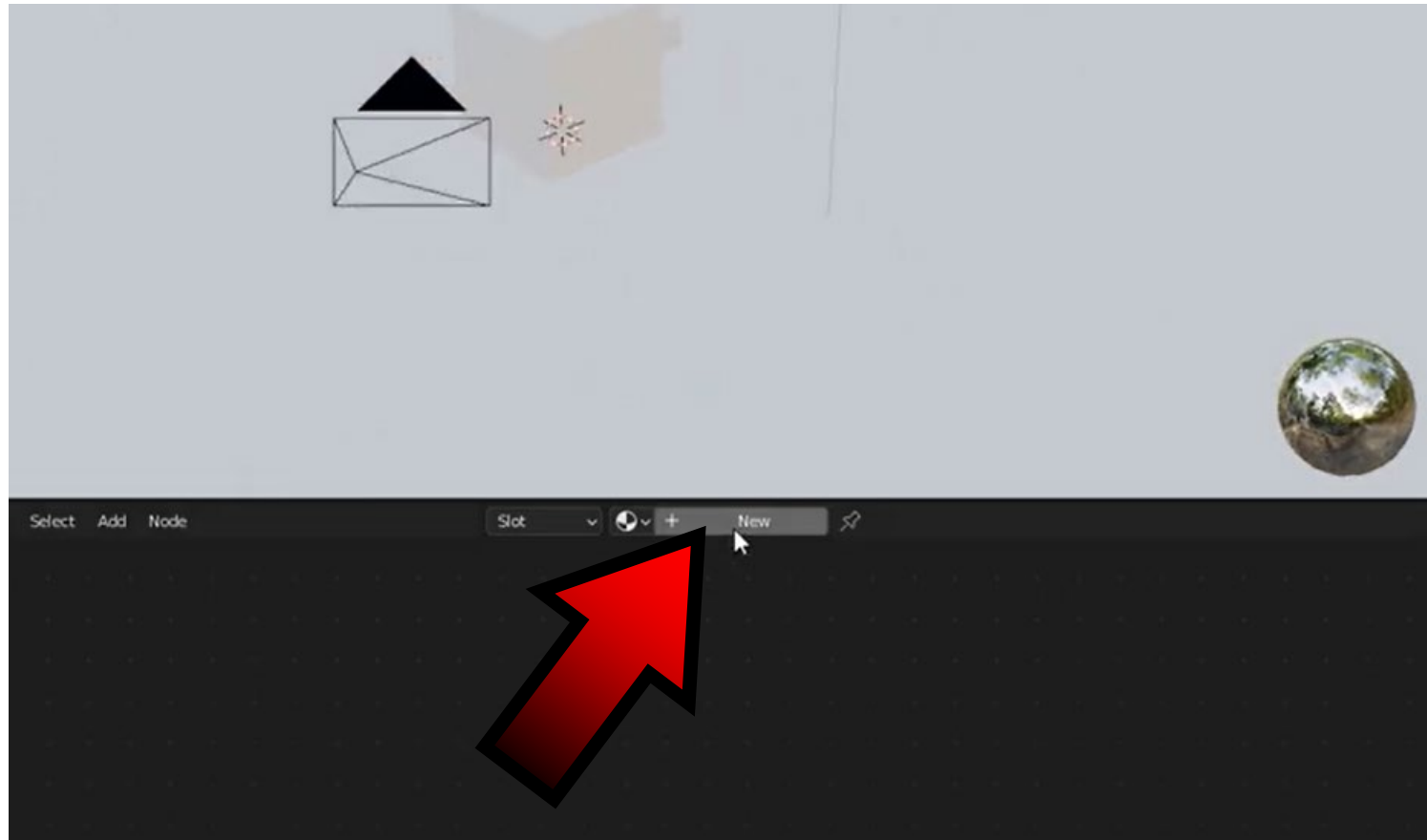


UPBGE

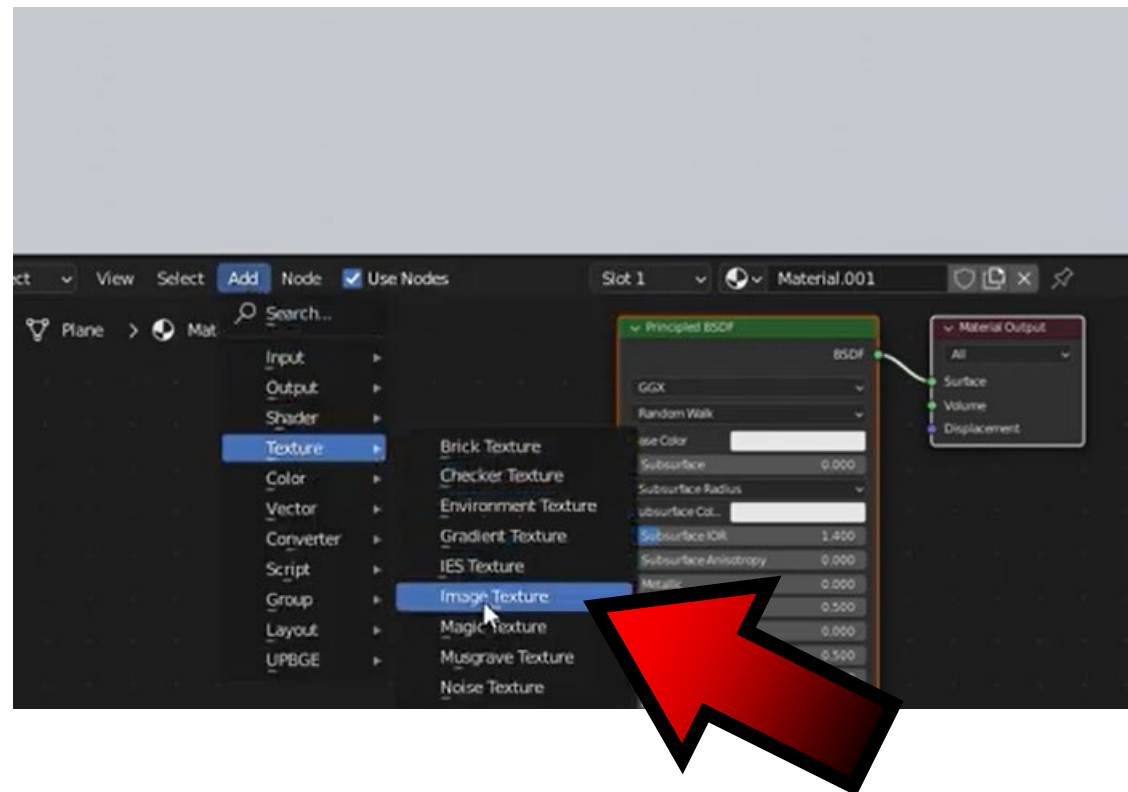
IDŹ DO WIDOKU **SHADING**



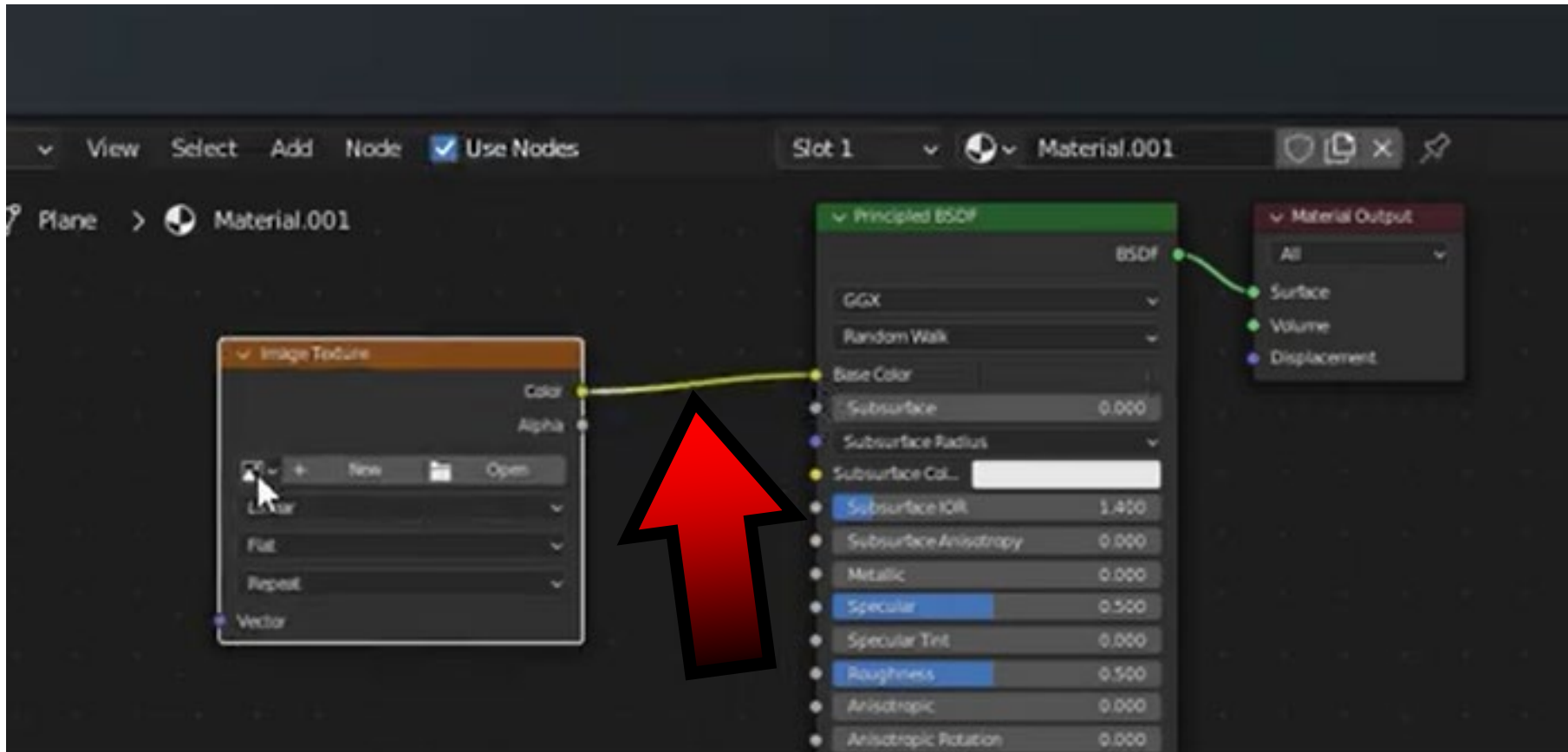
KLIKNIJ NA **NEW**



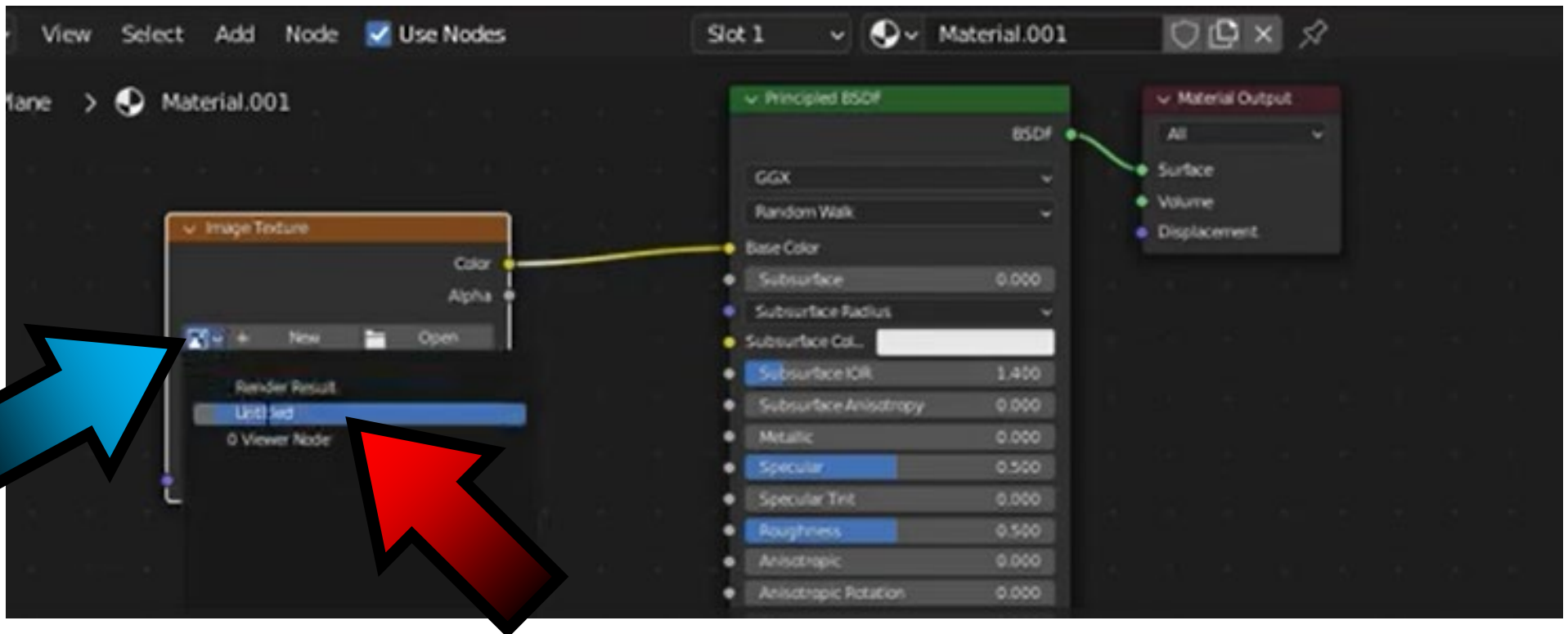
NACIŚNIJ **SHIFT+A** I DODAJ **IMAGE TEXTURE**



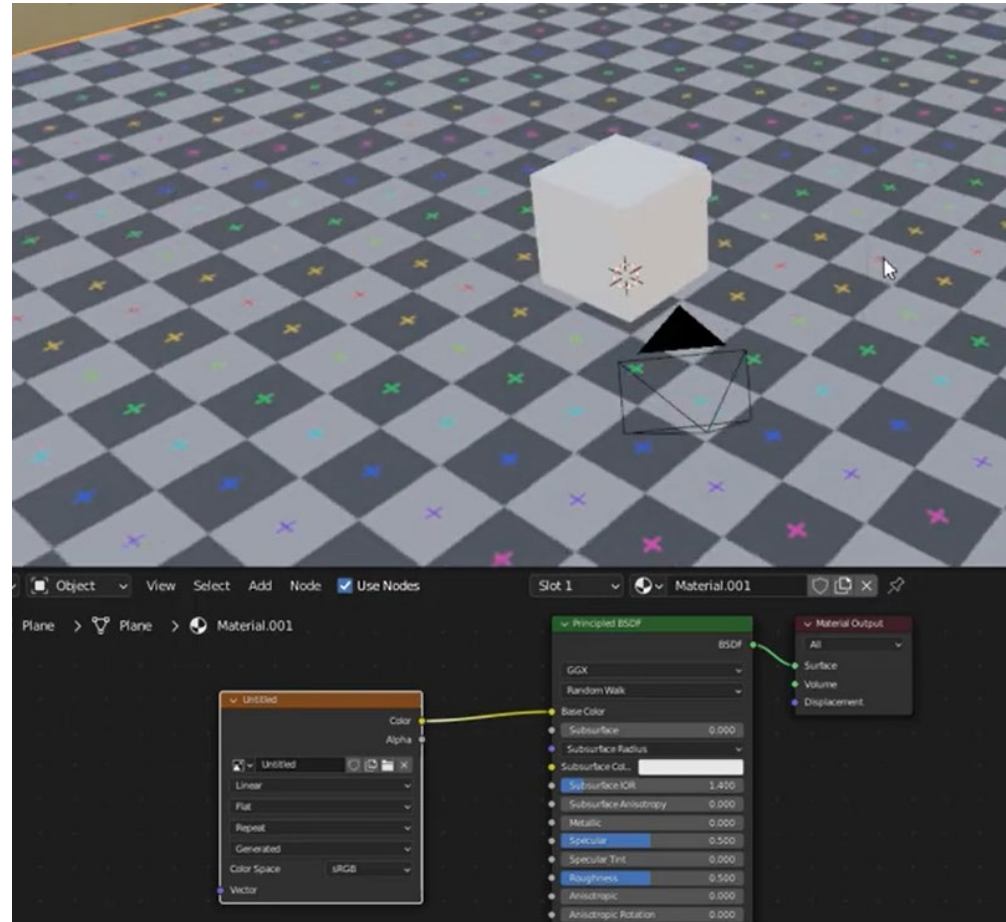
POŁĄCZ NODS



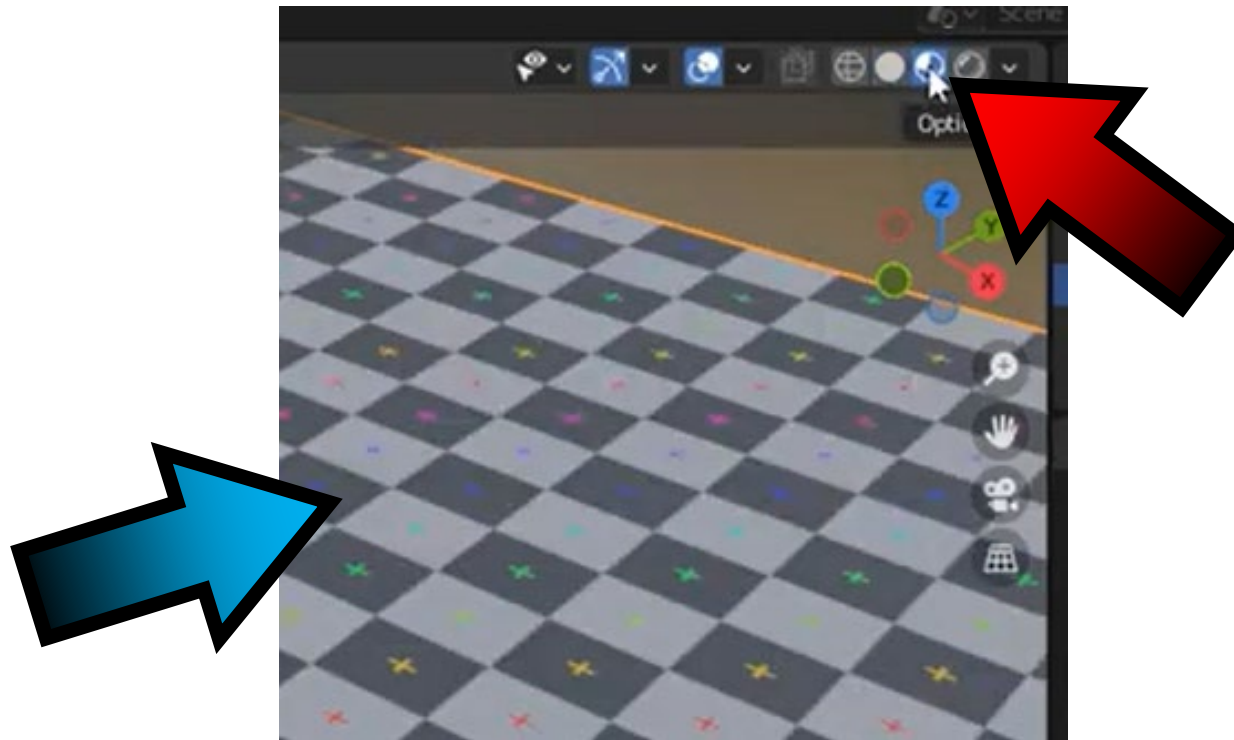
WYBIERZ TEXTURE Z UV EDITOR



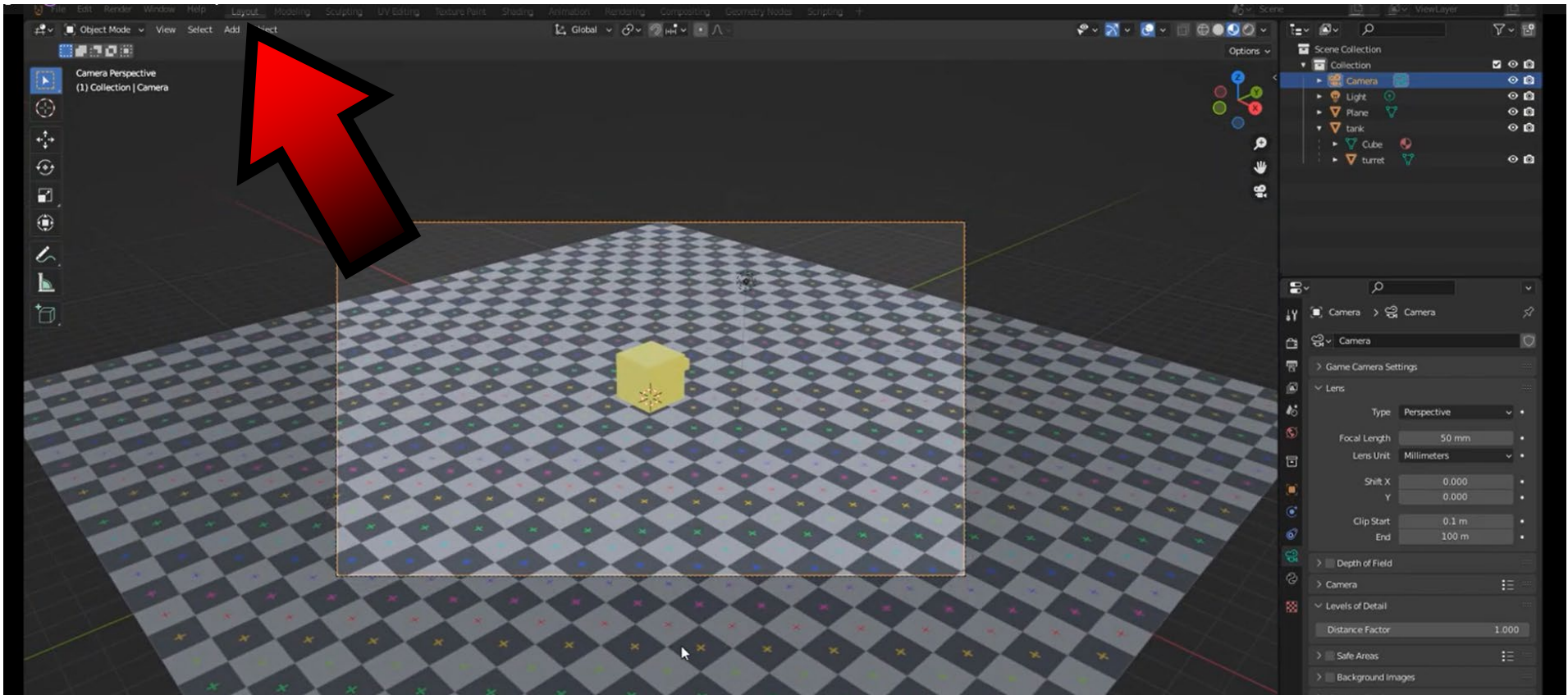
WIDOK W SHADING



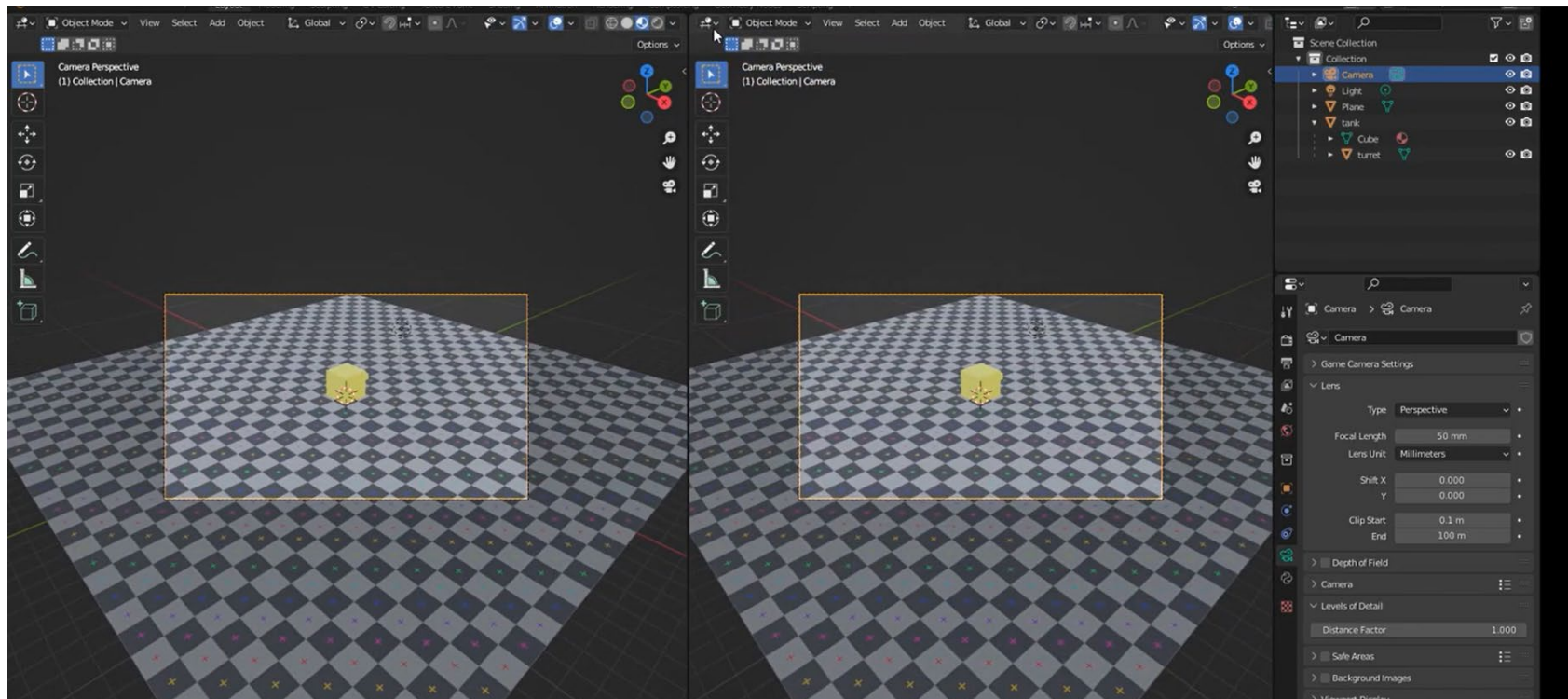
JEŚLI NIE WIDZISZ GRAFIKI KLIKNIJ W MATERIAL PREVIEW



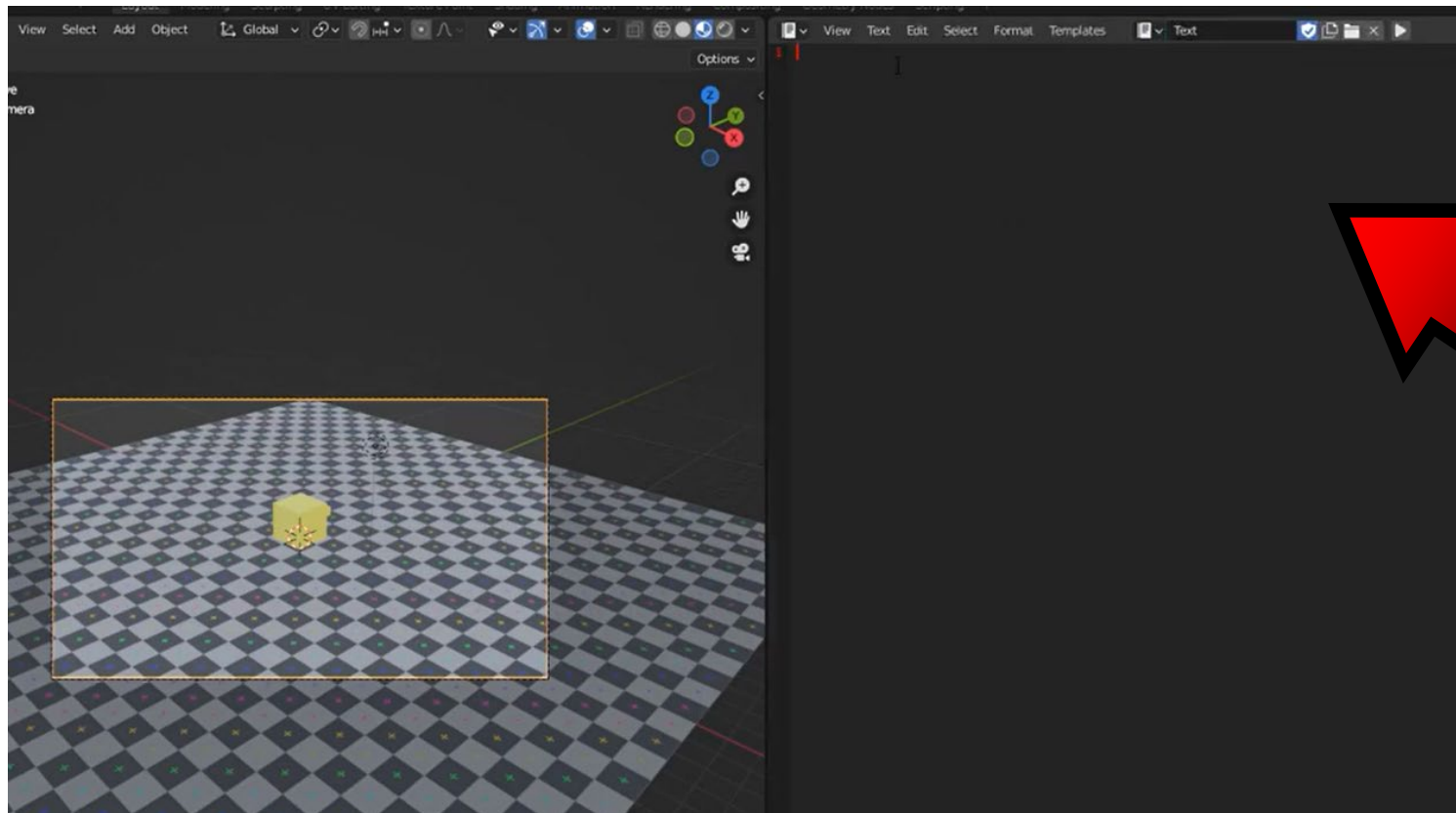
IDŹ DO **LAYOUT**



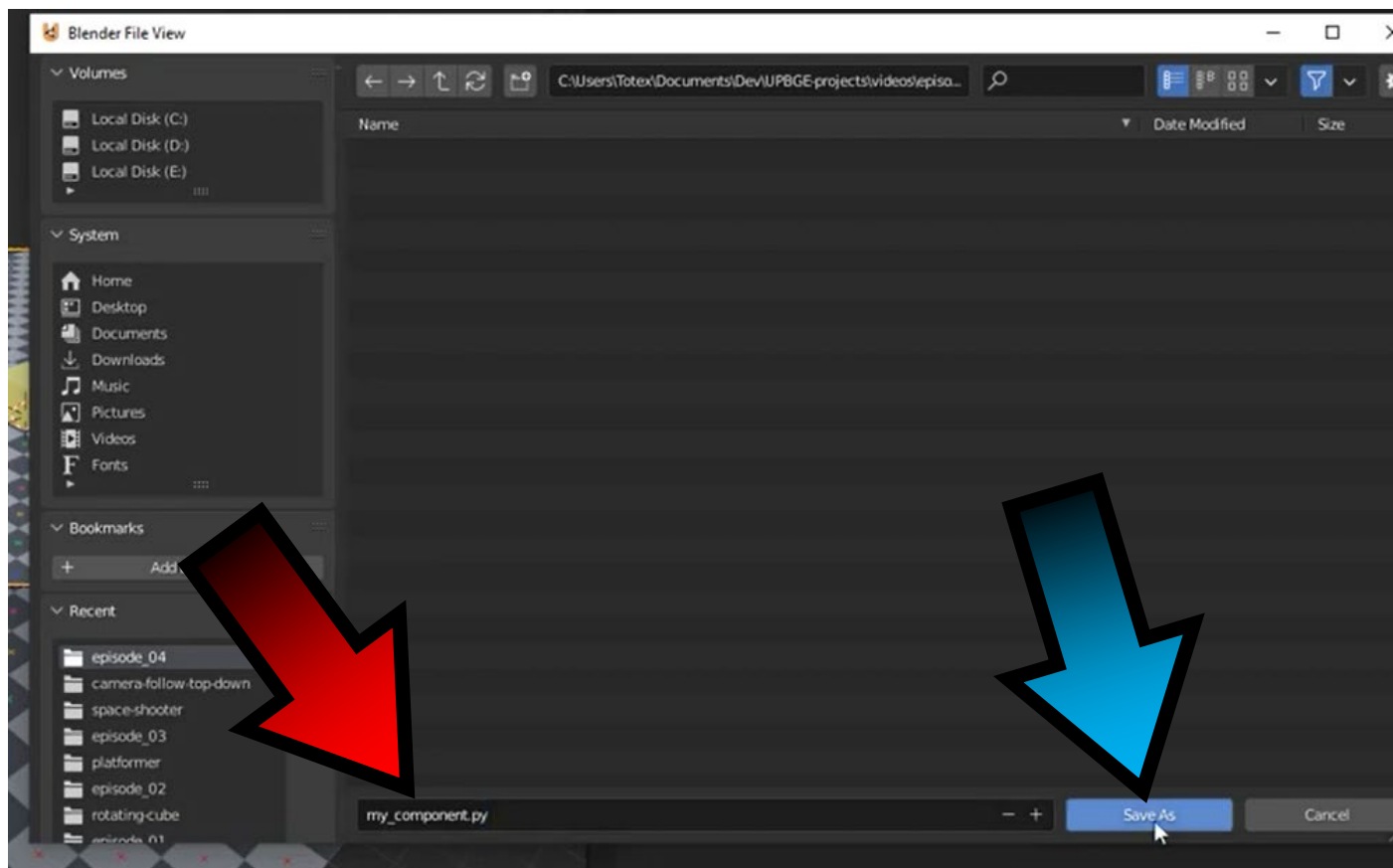
PODZIEL OKNO



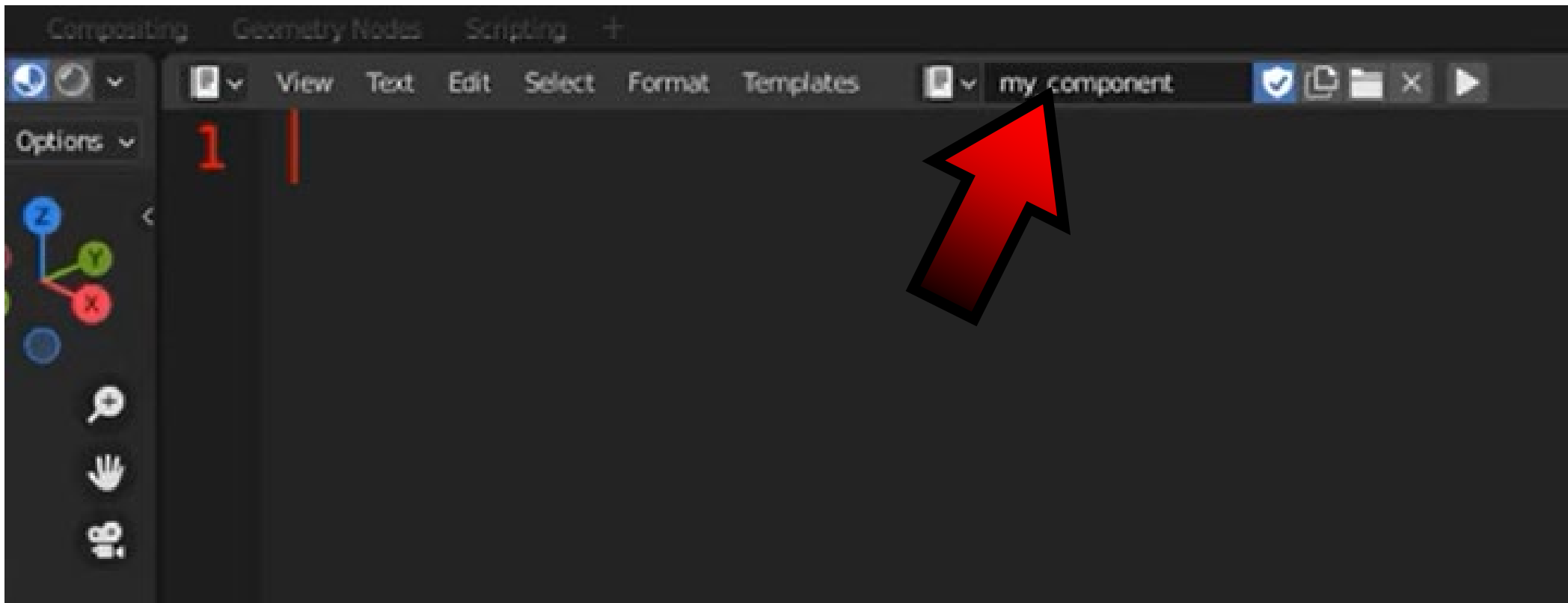
WYBIERZ **TEXT EDIT**



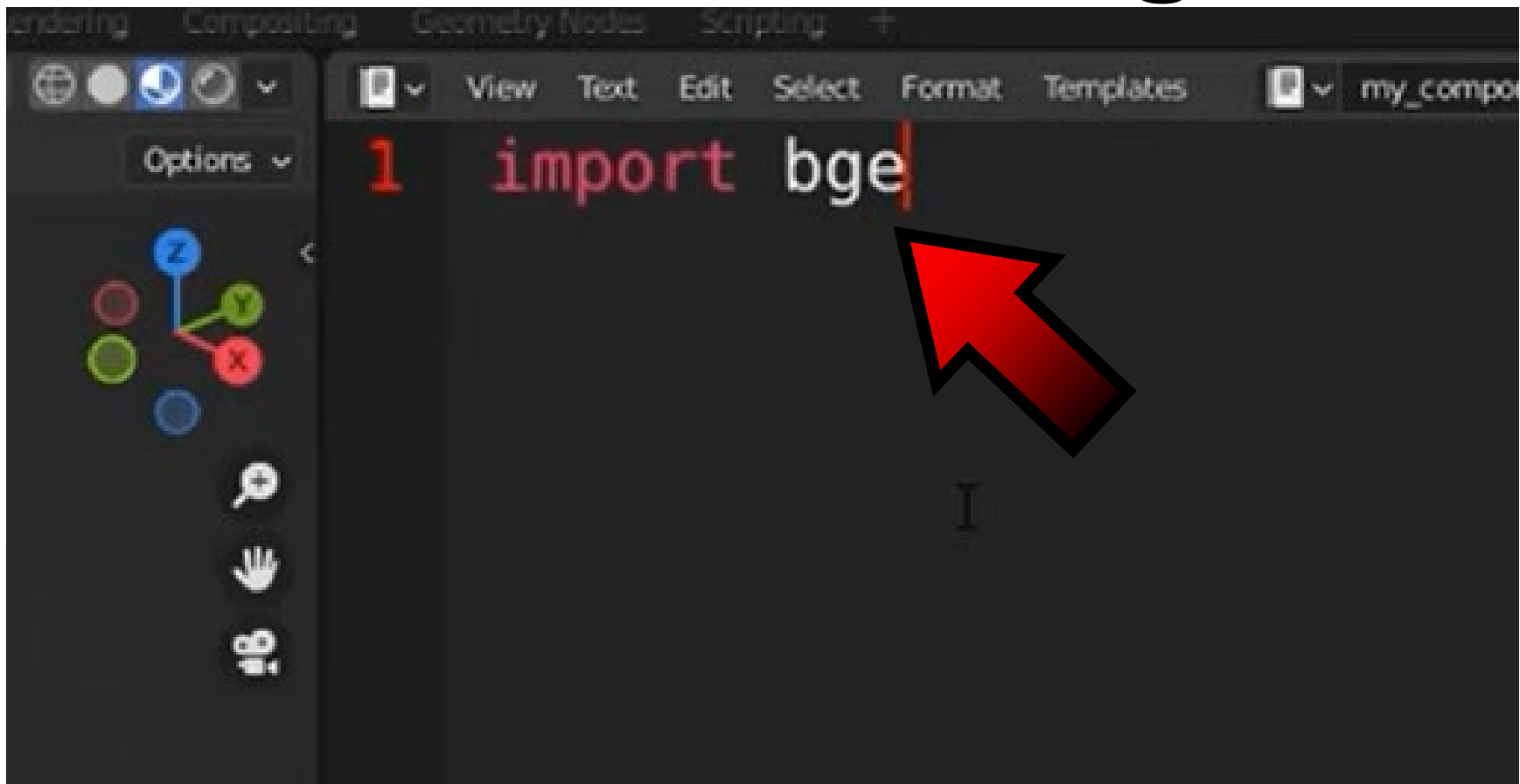
ZAPISZ SKRYPT NAZWA JEST BARDZO WAŻNA



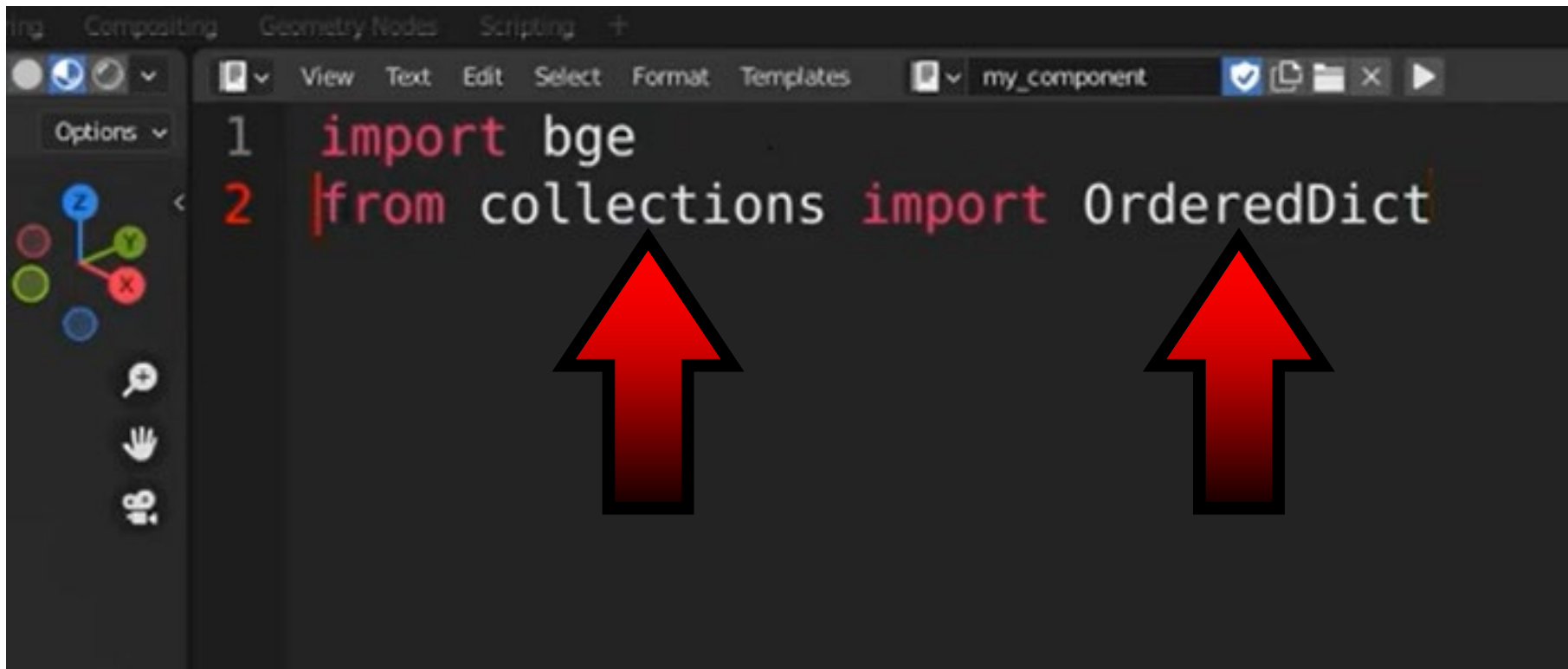
WPROWADŹ TUTAJ NAZWĘ SKRYPTU



IMPORT Z **BGE** **B**lender **G**ame **E**ngine



IMPORT Z COLLECTIONS



```
1 import bge
2 from collections import OrderedDict
```

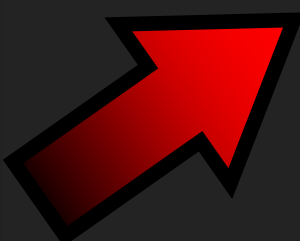
The screenshot shows a code editor window with a dark theme. The code is as follows:

```
1 import bge
2 from collections import OrderedDict
```

Two large red arrows with black outlines point upwards from the bottom of the code editor. The first arrow points to the word 'collections' in line 2. The second arrow points to the word 'OrderedDict' in line 2. The editor interface includes a menu bar with 'View', 'Text', 'Edit', 'Select', 'Format', and 'Templates'. The file name 'my_component' is visible in the top right corner. On the left side, there is a sidebar with a tree view showing a node labeled 'z' and other nodes, and a search bar.

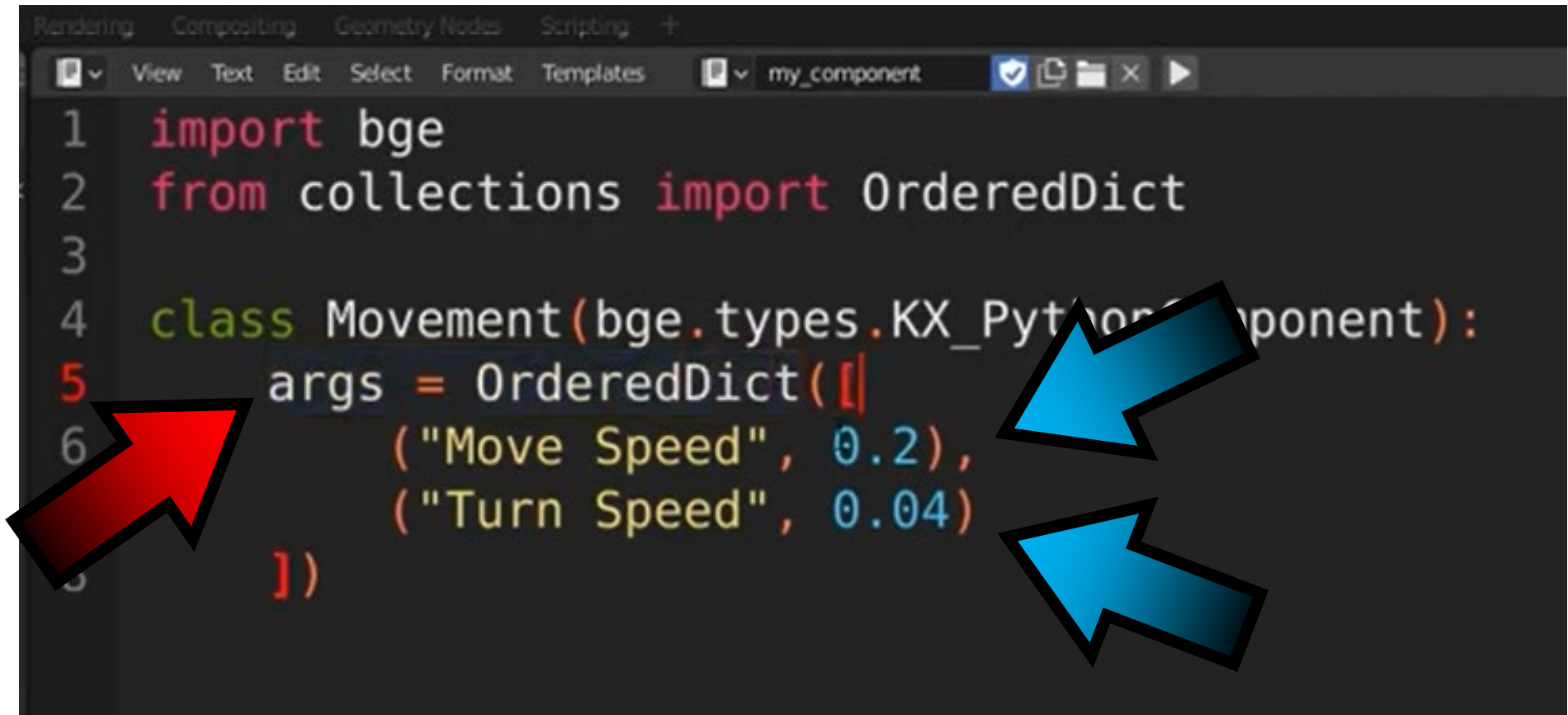
ODNIESIENIE DO **CLASS MOVEMENT**

```
Animation Rendering Compositing Geometry Nodes Scripting +
View Text Edit Select Format Templates my_component
1 import bge
2 from collections import OrderedDict
3
4 class Movement(bge.types.KX_PythonComponent):|
```




USTALAMY ARGUMENTY

```
Rendering Compositing Geometry Nodes Scripting +
View Text Edit Select Format Templates my_component
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
```



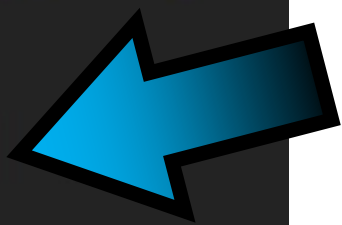

USTALAMY TRYB STARTOWY

```
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']
```




USTALAMY TRYB AKTUALIZACJI

```
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
```



USTALAMY JAK DANE BĘDĄ WPROWADZANE

```
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
```



Mouse Keys

```
bge.events.LEFTMOUSE  
bge.events.MIDDLEMOUSE  
bge.events.RIGHTMOUSE  
bge.events.WHEELUPMOUSE  
bge.events.WHEELDOWNMOUSE  
bge.events.MOUSEX  
bge.events.MOUSEY
```

Alphabet keys

```
bge.events.AKEY  
bge.events.BKEY  
bge.events.CKEY  
bge.events.DKEY  
bge.events.EKEY  
bge.events.FKEY  
bge.events.GKEY  
bge.events.HKEY  
bge.events.IKEY  
bge.events.JKEY  
bge.events.KKEY  
bge.events.LKEY  
bge.events.MKEY  
bge.events.NKEY  
bge.events.OKEY  
bge.events.PKEY  
bge.events.QKEY  
bge.events.RKEY  
bge.events.SKEY  
bge.events.TKEY  
bge.events.UKEY  
bge.events.VKEY  
bge.events.WKEY  
bge.events.XKEY  
bge.events.YKEY  
bge.events.ZKEY
```

Number keys

```
bge.events.ZEROKEY  
bge.events.ONEKEY  
bge.events.TWOKEY  
bge.events.THREEKEY  
bge.events.FOURKEY  
bge.events.FIVEKEY  
bge.events.SIXKEY  
bge.events.SEVENKEY  
bge.events.EIGHTKEY  
bge.events.NINEKEY
```

Numberpad Keys

```
bge.events.PAD0  
bge.events.PAD1  
bge.events.PAD2  
bge.events.PAD3  
bge.events.PAD4  
bge.events.PAD5  
bge.events.PAD6  
bge.events.PAD7  
bge.events.PAD8  
bge.events.PAD9  
bge.events.PADPERIOD  
bge.events.PADSLASHKEY  
bge.events.PADASTERKEY  
bge.events.PADMINUS  
bge.events.PADENTER  
bge.events.PADPLUSKEY
```

Modifiers Keys

```
bge.events.CAPSLOCKKEY  
bge.events.LEFTCTRLKEY  
bge.events.LEFTALTKEY  
bge.events.RIGHTALTKEY  
bge.events.RIGHTCTRLKEY  
bge.events.RIGHTSHIFTKEY  
bge.events.LEFTSHIFTKEY
```

Arrow Keys

```
bge.events.LEFTARROWKEY  
bge.events.DOWNARROWKEY  
bge.events.RIGHTARROWKEY  
bge.events.UPARROWKEY
```

Function Keys


```
bge.events.F1KEY  
bge.events.F2KEY  
bge.events.F3KEY  
bge.events.F4KEY  
bge.events.F5KEY  
bge.events.F6KEY  
bge.events.F7KEY  
bge.events.F8KEY  
bge.events.F9KEY  
bge.events.F10KEY  
bge.events.F11KEY  
bge.events.F12KEY  
bge.events.F13KEY  
bge.events.F14KEY  
bge.events.F15KEY  
bge.events.F16KEY  
bge.events.F17KEY  
bge.events.F18KEY  
bge.events.F19KEY
```

Other Keys

```
bge.events.ACCENTGRAVEKEY  
bge.events.BACKSLASHKEY  
bge.events.BACKSPACEKEY  
bge.events.COMMAKEY  
bge.events.DELKEY  
bge.events.ENDKEY  
bge.events.EQUALKEY  
bge.events.ESCKEY  
bge.events.HOMEKEY  
bge.events.INSERTKEY  
bge.events.LEFTBRACKETKEY  
bge.events.LINEFEEDKEY  
bge.events.MINUSKEY  
bge.events.PAGEDOWNKEY  
bge.events.PAGEUPKEY  
bge.events.PAUSEKEY  
bge.events.PERIODKEY  
bge.events.QUOTEKEY  
bge.events.RIGHTBRACKETKEY  
bge.events.ENTERKEY  
bge.events.SEMICOLONKEY  
bge.events.SLASHKEY  
bge.events.SPACEKEY  
bge.events.TABKEY
```



ZEROWANIE SKŁADNIKÓW

```
13
14     def update(self):
15         keyboard = bge.logic.keyboard
16         inputs = keyboard.inputs
17
18         move = 0
19         rotate = 0
```



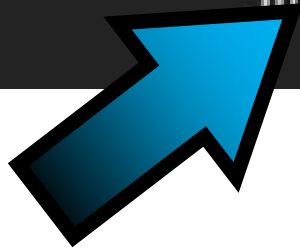

PRZEMIESZCZANIE KLAWISZEM W

```
18     move = 0
19     rotate = 0
20
21     if inputs[bge.events.WKEY].values[-1]:
22         move += self.move_speed
```




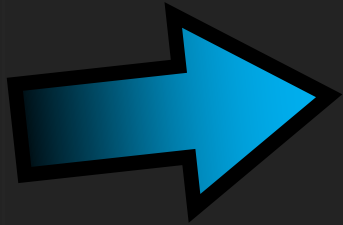
PRZEMIESZCZANIE KLAWISZEM S

```
21     if inputs[bge.events.WKEY].values[-1]:  
22         move += self.move_speed  
23     if inputs[bge.events.SKEY].values[-1]:  
24         move -= self.move_speed
```



OBRÓT KLAWISZAMI A I D

```
21     if inputs[bge.events.WKEY].values[-1]:
22         move += self.move_speed
23     if inputs[bge.events.SKEY].values[-1]:
24         move -= self.move_s
25
26     if inputs[bge.events.AKEY].values[-1]:
27         rotate += self.turn_speed
28     if inputs[bge.events.DKEY].values[-1]:
29         rotate -= self.turn_speed
```



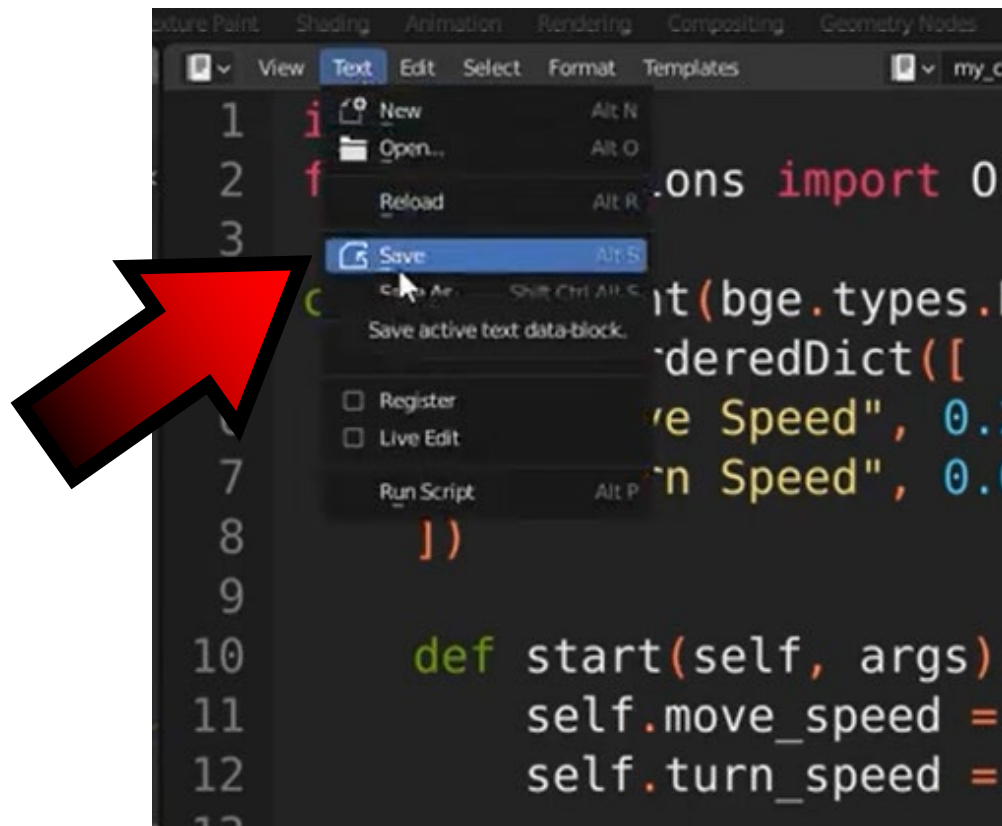
POWER OF AR AND VR

PRZEMIESZCZANIE W OSIACH LOKALNYCH

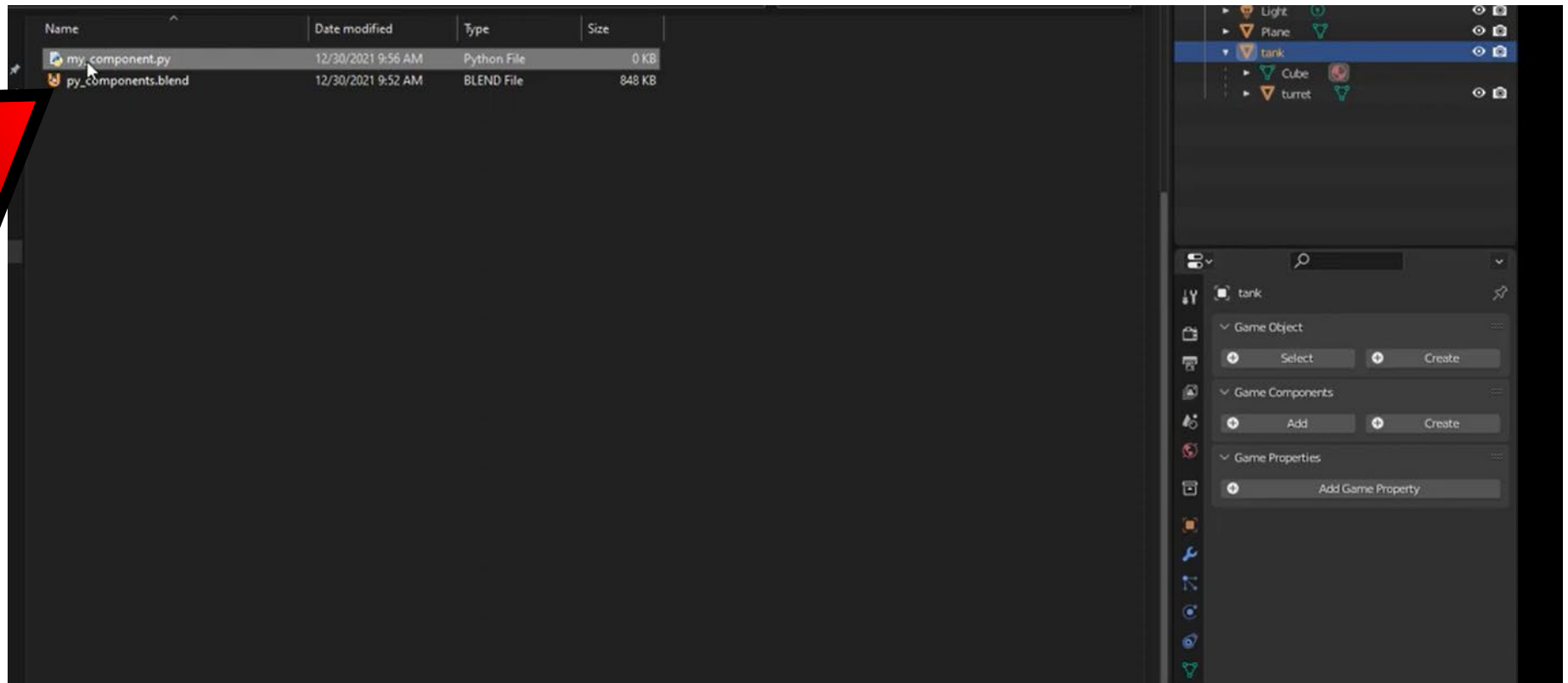
```
25  
26     if inputs[bge.events.AKEY].values[-1]:  
27         rotate += self.turn_speed  
28     if inputs[bge.events.DKEY].values[-1]:  
29         rotate -= self.turn_speed  
30  
31     self.object.applyMovement((0, move, 0), True)
```



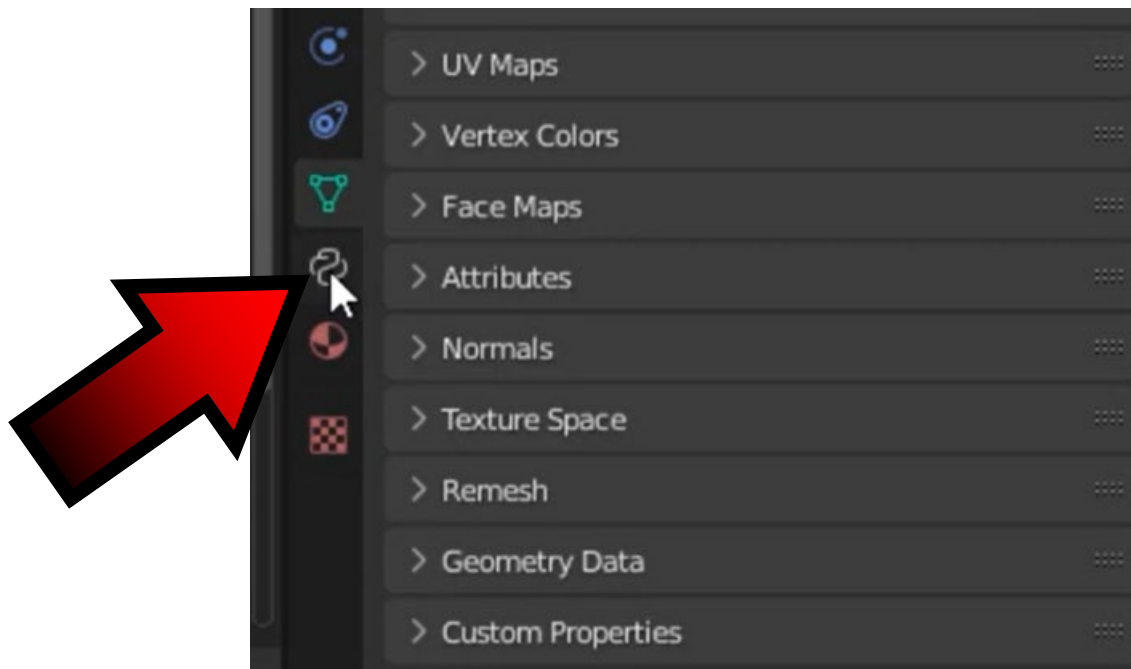
ZAPISZ SKRYPT



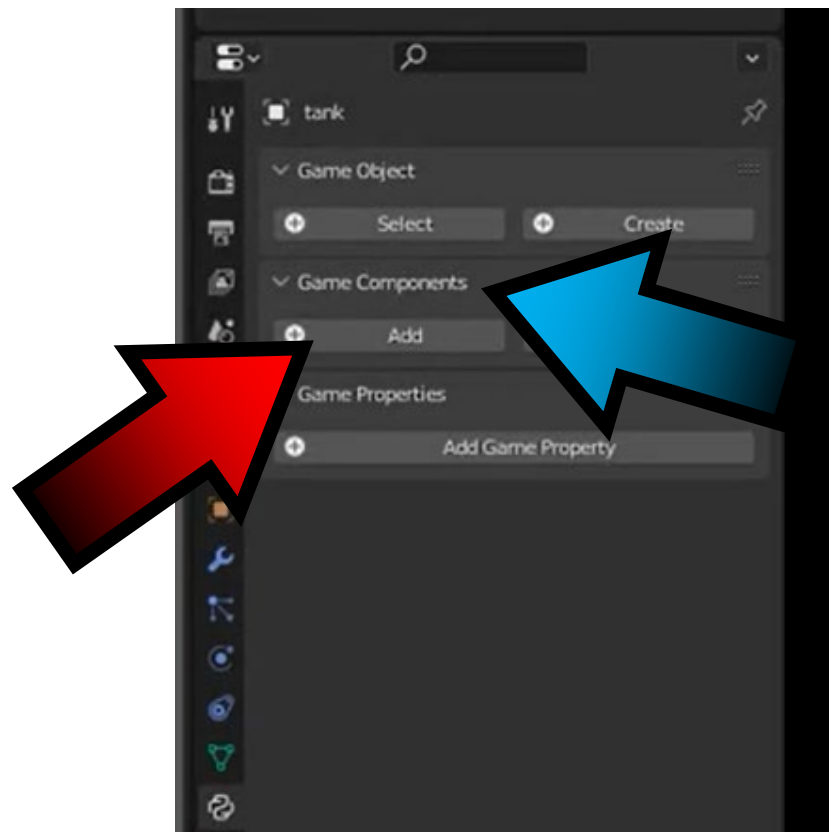
NIE ZMIENIAJMY NAZWY



WPROWADZAMY WŁAŚCIWOŚCI OBIEKTÓW GRY



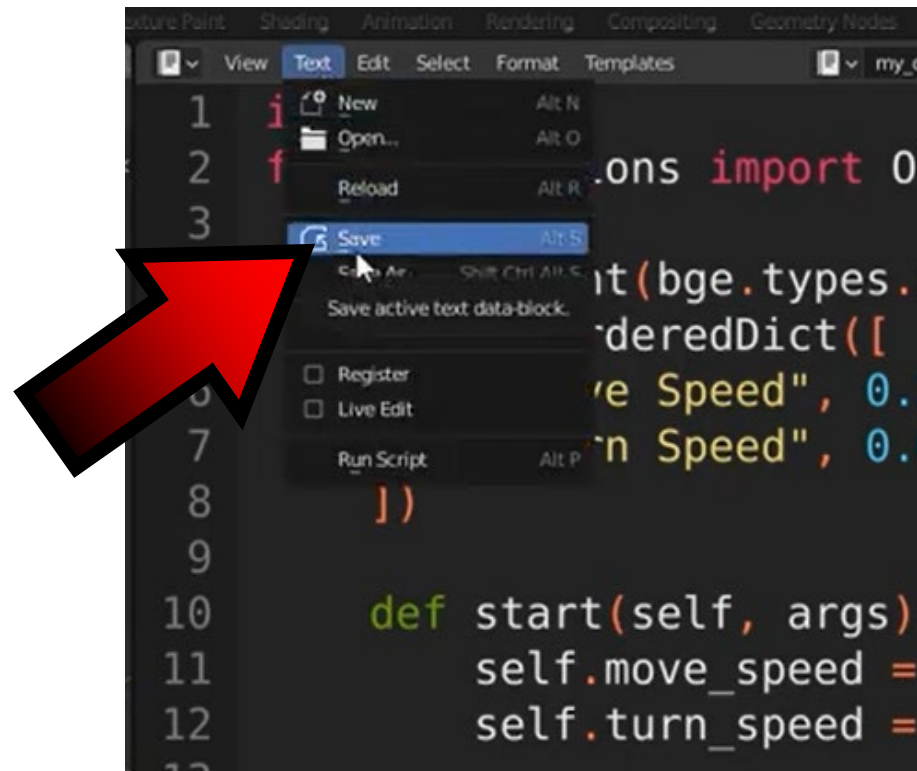
KLIKNIJ NA **ADD** W **GAME COMPONENTS**



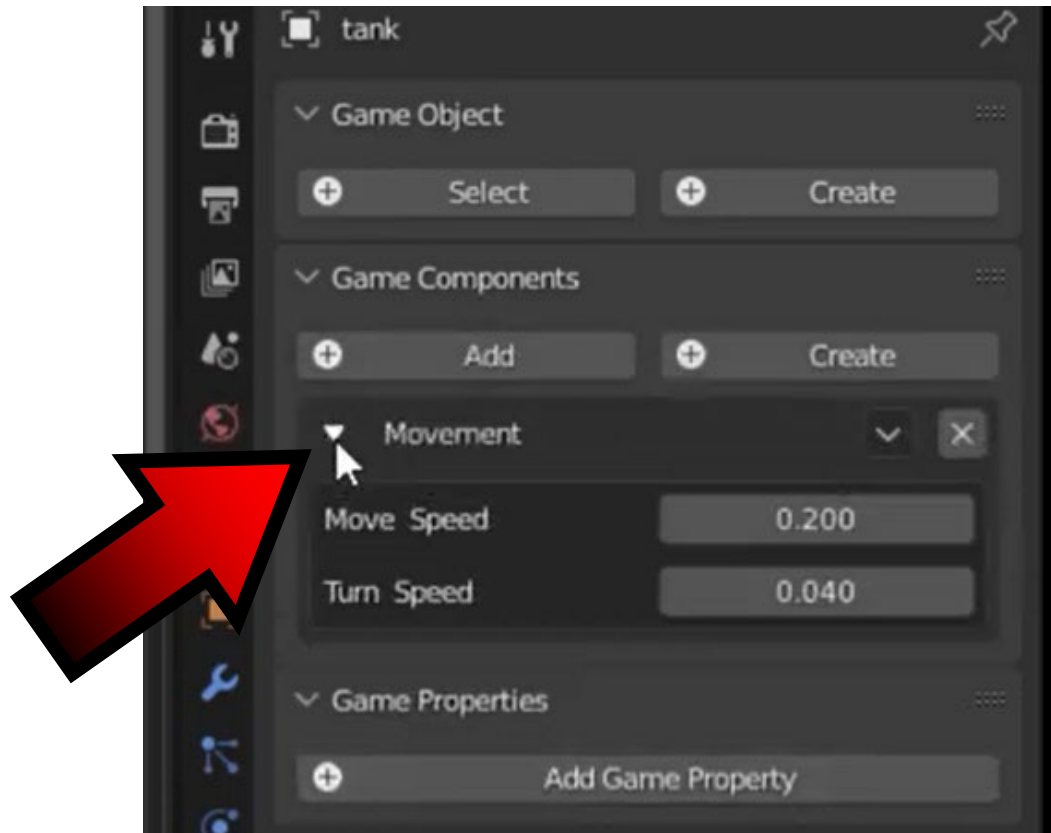
WPROWADŹ NAZWĘ PLIKU I PO KROPCE NAZWĘ KLASY



ZAPISZ SKRYPT POWNOWNIE

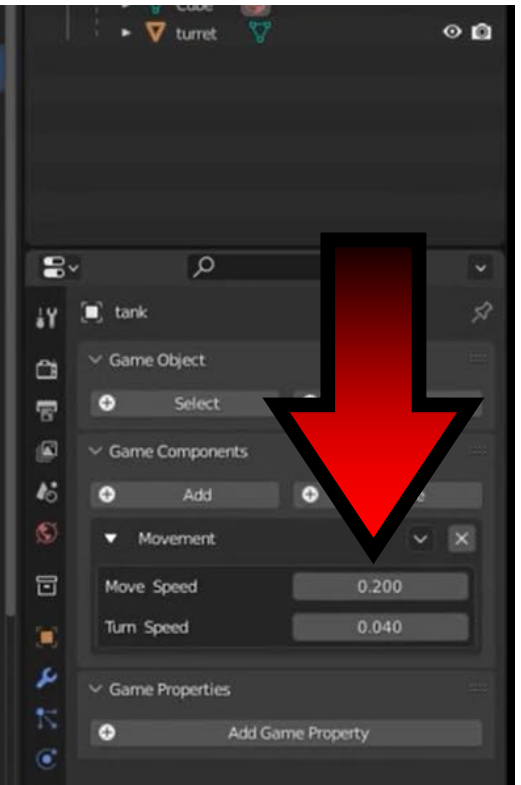


OTWÓRZ MOVEMENT



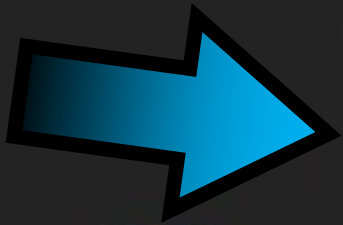
TE DWA ARGUMENTY POBIERAMY Z TEGO MIEJSCA

```
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.WKEY].values[-1]:
```

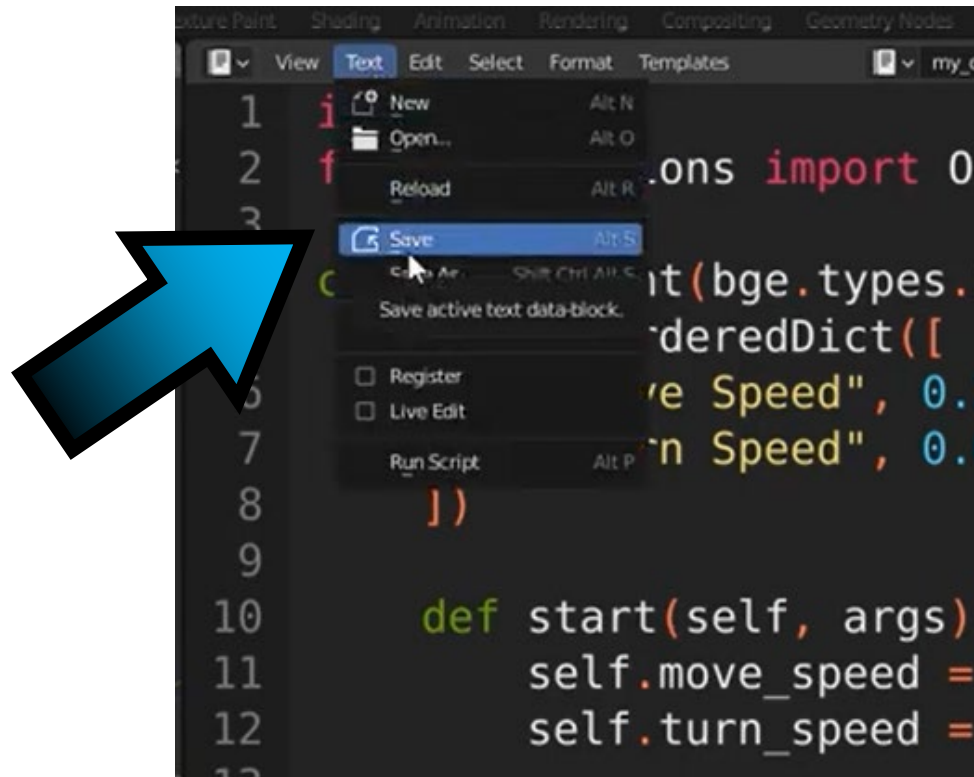


JEŚLI DODAMY TRZECI

```
args = OrderedDict([  
    ("Move Speed", 0.2),  
    ("Turn Speed", 0.04),  
    ("My Var", 5)  
])
```

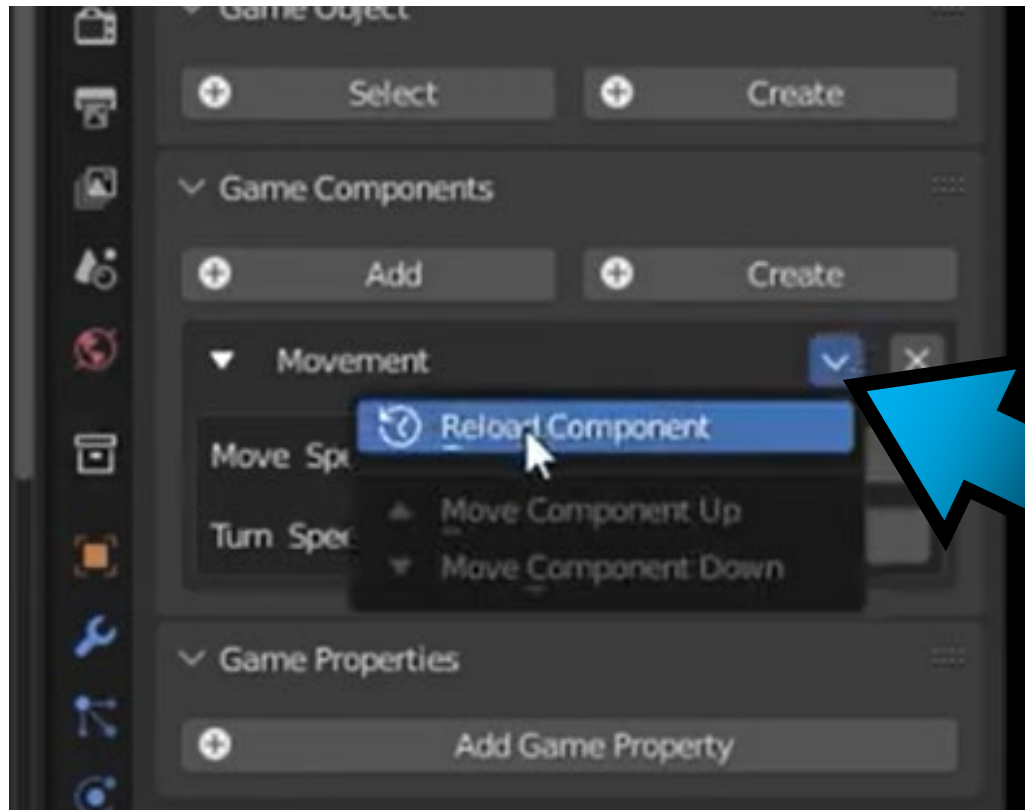


I ZAPISZEMY SKRYPT

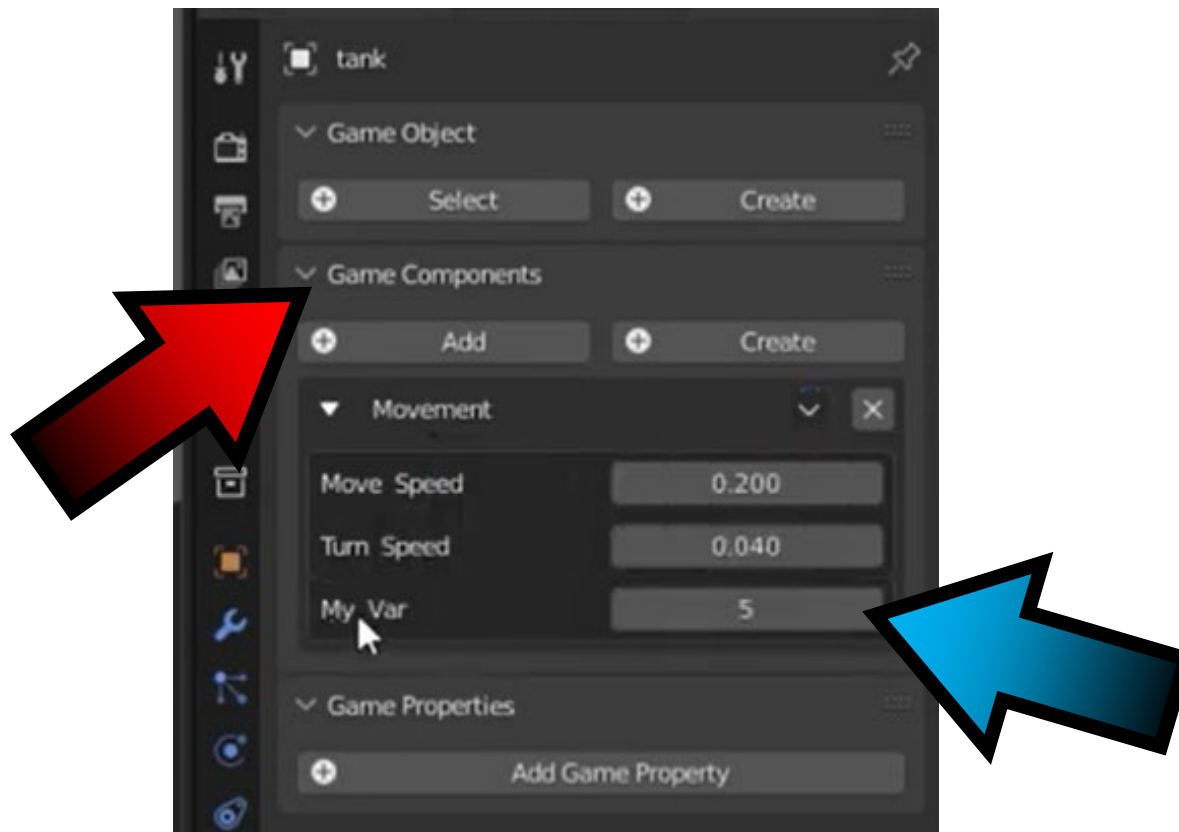


```
1 i New Alt N
2 f Open... Alt O
3 Reload Alt R
4 Save Alt S
5 Save active text data-block.
6 Register
7 Live Edit
8 Run Script Alt P
9
10 def start(self, args):
11     self.move_speed =
12     self.turn_speed =
```

PO ODŚWIERZENIU



TRZECIA ZMIENNA POKAŻE SIĘ W **GAME COMPONENTS**

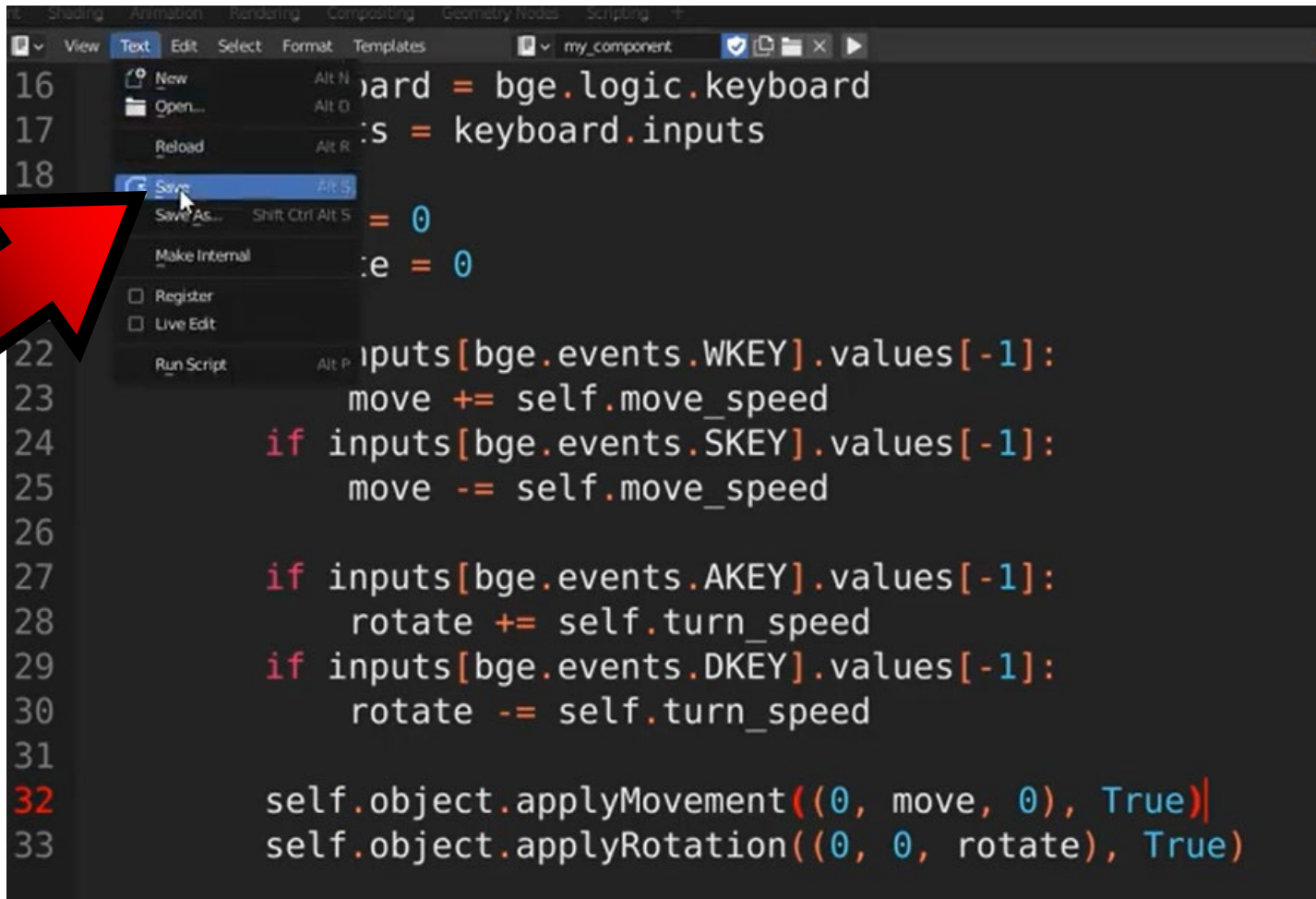


DODAJEMY LOKALNY OBRÓT VEKTORA **WOKÓŁ OSI Z**

```
31  
32 self.object.applyMovement((0, move, 0), True)  
33 self.object.applyRotation((0, 0, rotate), True)
```

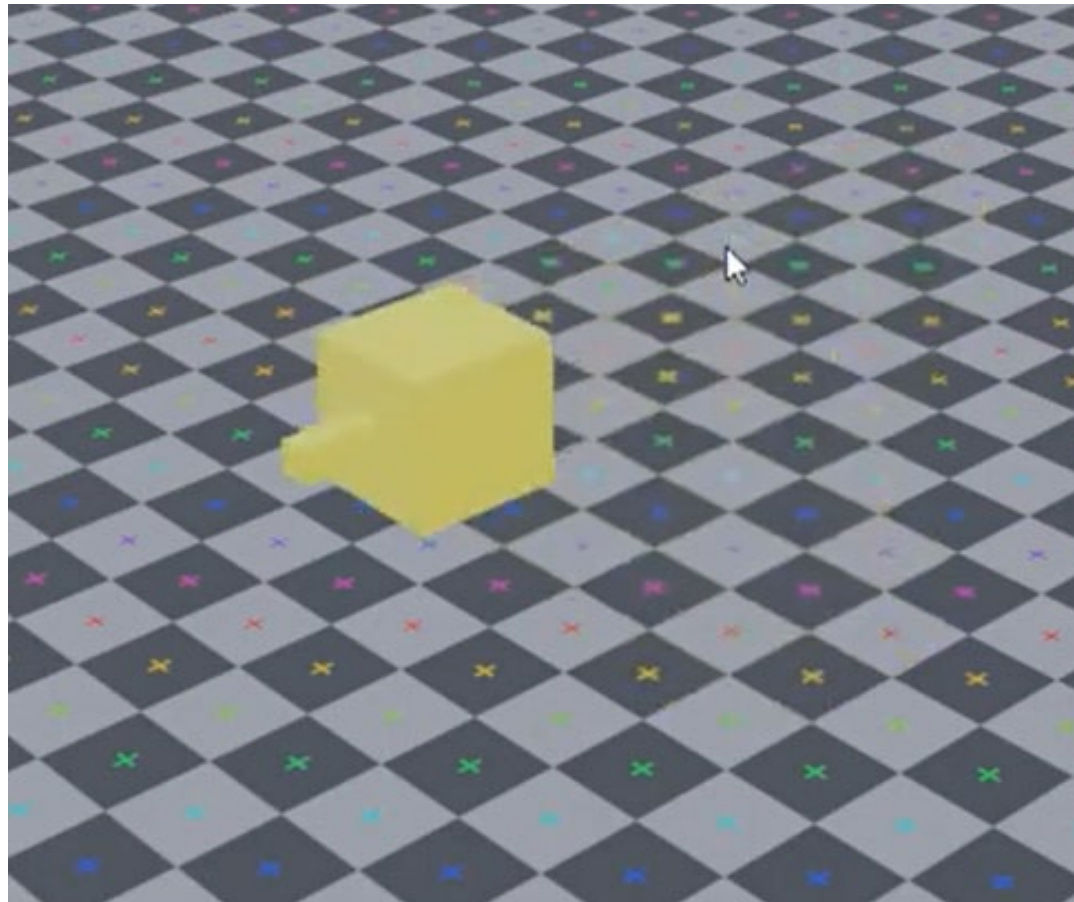


ZAPISZ SKRYPT



```
16 keyboard = bge.logic.keyboard
17 :s = keyboard.inputs
18
22 :e = 0
23
24 inputs[bge.events.WKEY].values[-1]:
25     move += self.move_speed
26
27 if inputs[bge.events.SKEY].values[-1]:
28     move -= self.move_speed
29
30 if inputs[bge.events.AKEY].values[-1]:
31     rotate += self.turn_speed
32
33 if inputs[bge.events.DKEY].values[-1]:
34     rotate -= self.turn_speed
35
36 self.object.applyMovement((0, move, 0), True)
37 self.object.applyRotation((0, 0, rotate), True)
```

URUCHOM GRĘ KLAWISZEM P



POWER OF AR AND VR

DZIĘKUJĘ ZA UWAGĘ



Co-funded by
the European Union



2024-1-PL01-KA220-VET-000243150

ημε Ενωθεν ηνιον
Co-funded by