

VPython - symulacje fizyczne z grafiką 3D dla każdego

wykład 10

Dr hab. Adam Bzdak, prof. AGH

scene.objects

76 vp_scene_1.py - C:/Users/Wysokie Energie/Desktop/VPython/lec_a_10/vp_scene_1.py

File Edit Format Run Options Windows Help

```
from visual import *

scene = display(width=550, height=580, range=5)

ba1 = sphere(pos=(0,0,0), radius=1)
ba2 = sphere(pos=(2.5,0,0), radius=1)

wa1 = box(pos=(0,2,0))
wa2 = box(pos=(0,4,0))

sleep(2)

wa2.visible = False

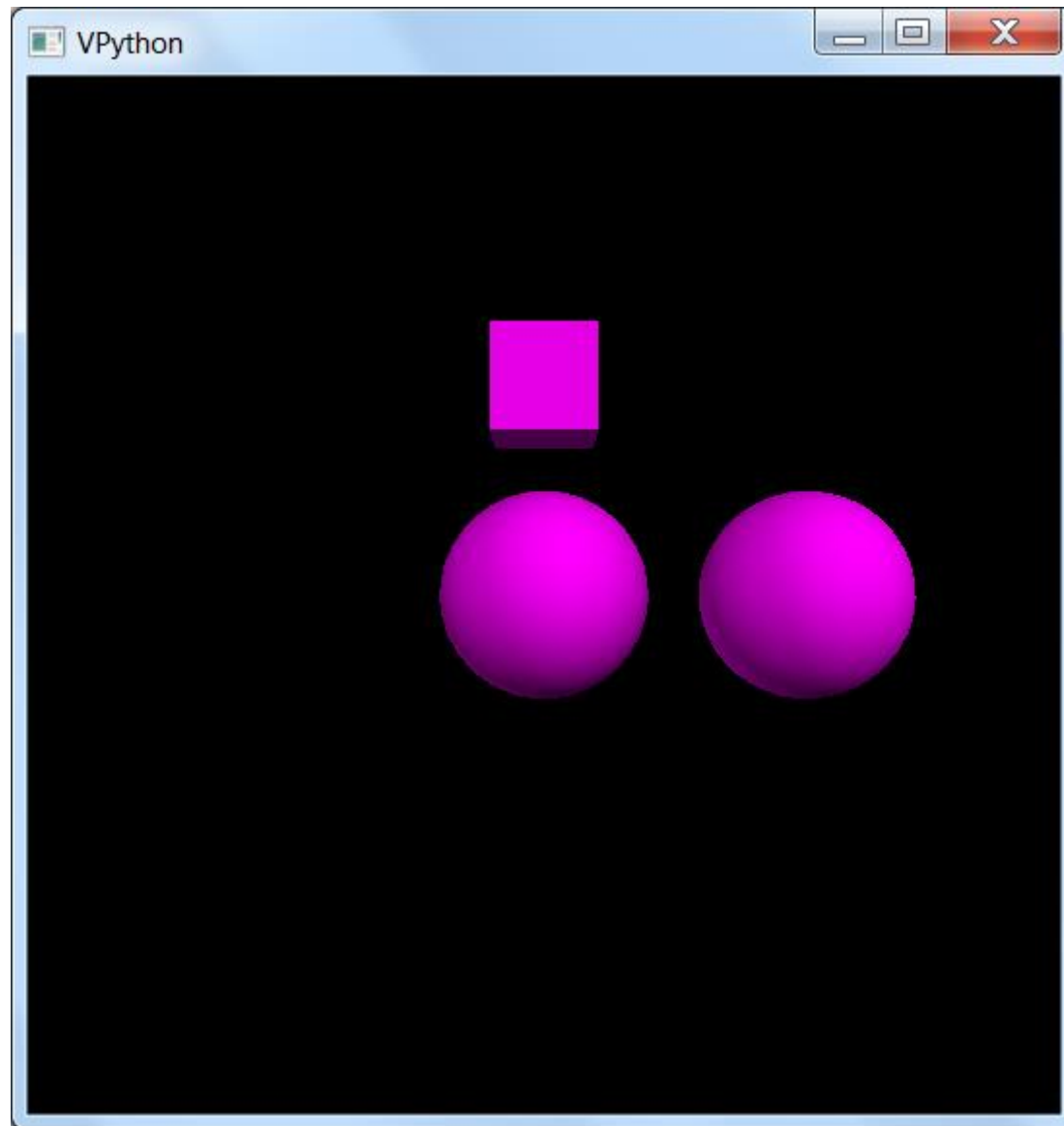
for k in scene.objects:
```

wszystkie widoczne obiekty

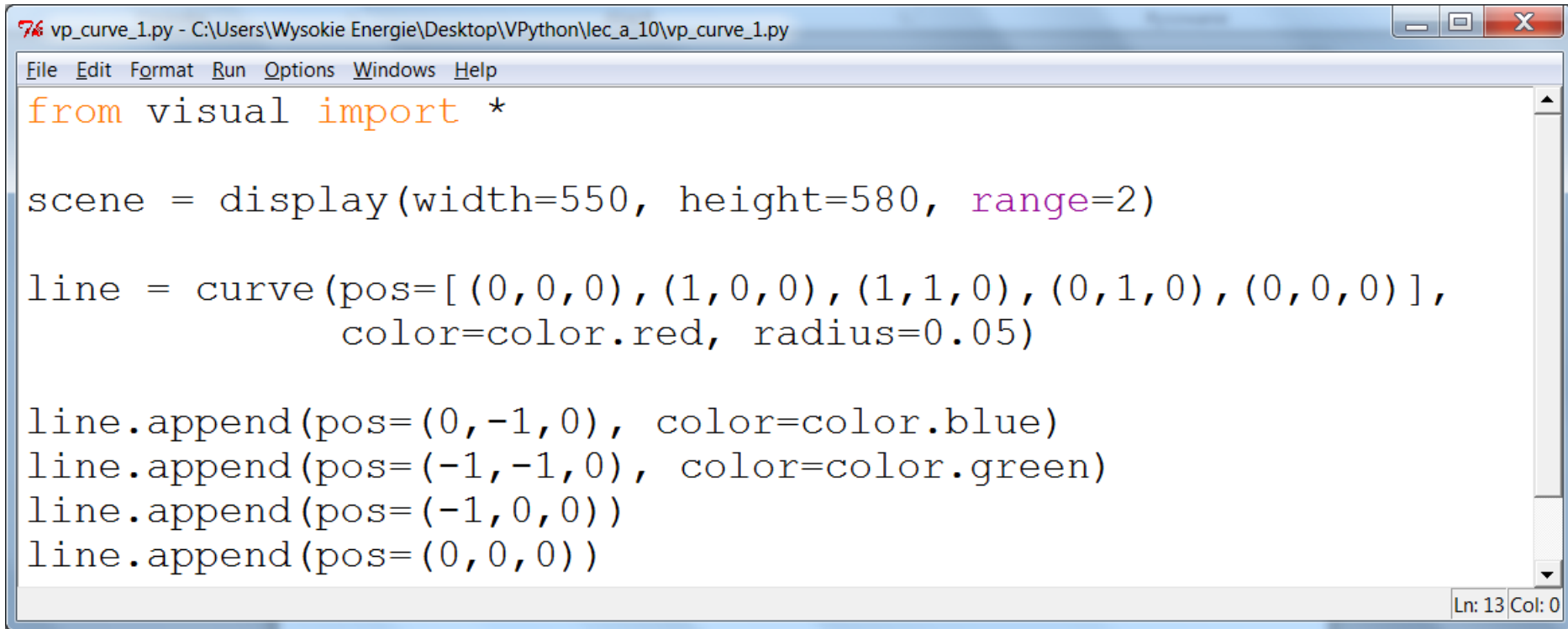
```
<0, 0, 0>
<2.5, 0, 0>
<0, 2, 0>
```

Ln: 20 Col: 0

Ln: 30 Col: 4



krzywa (curve)

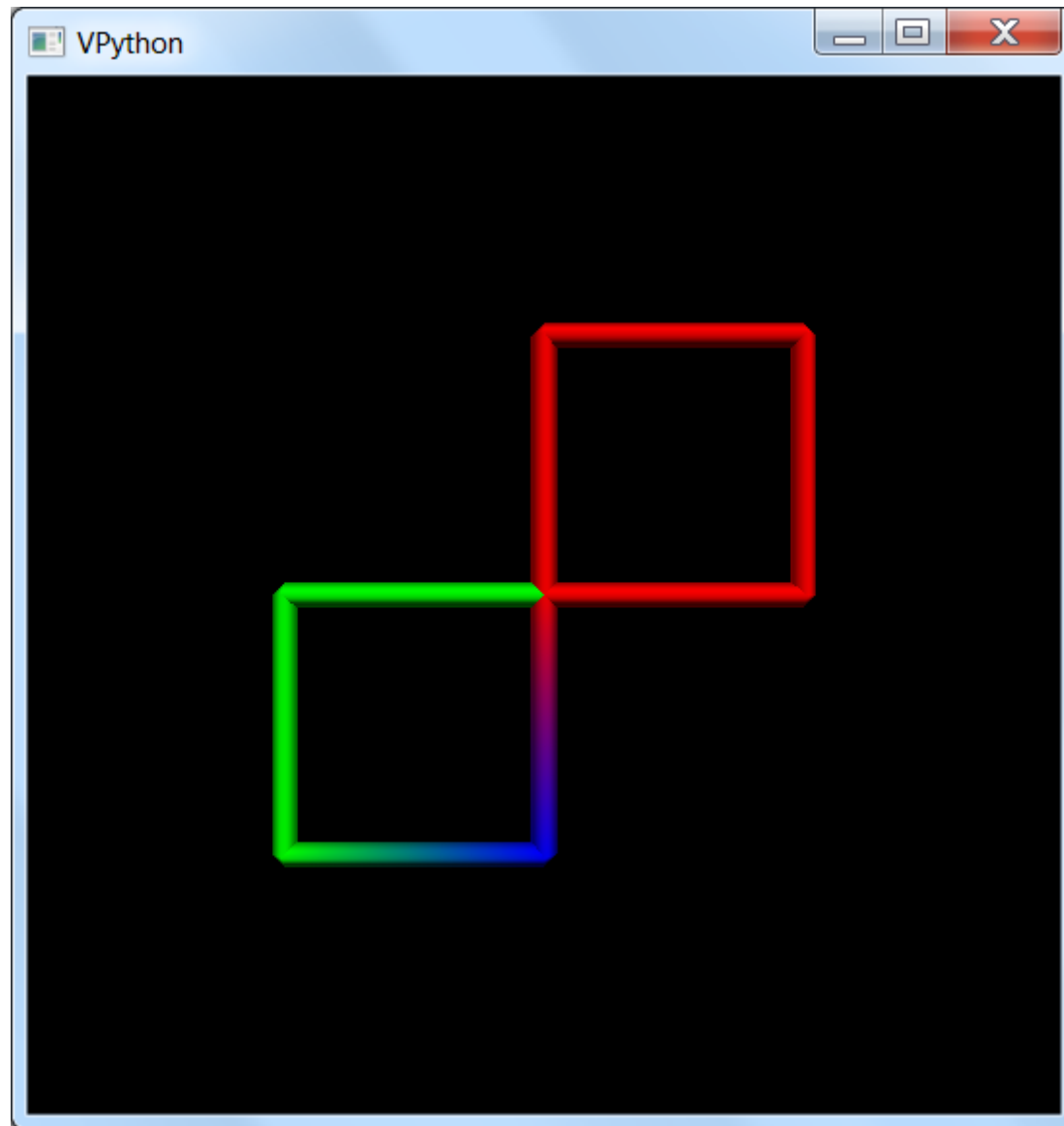
A screenshot of a VPython script editor window. The title bar shows the file path: "vp_curve_1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_a_10\vp_curve_1.py". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The code area contains the following Python code:

```
from visual import *  
  
scene = display(width=550, height=580, range=2)  
  
line = curve(pos=[(0,0,0), (1,0,0), (1,1,0), (0,1,0), (0,0,0)],  
             color=color.red, radius=0.05)  
  
line.append(pos=(0,-1,0), color=color.blue)  
line.append(pos=(-1,-1,0), color=color.green)  
line.append(pos=(-1,0,0))  
line.append(pos=(0,0,0))
```

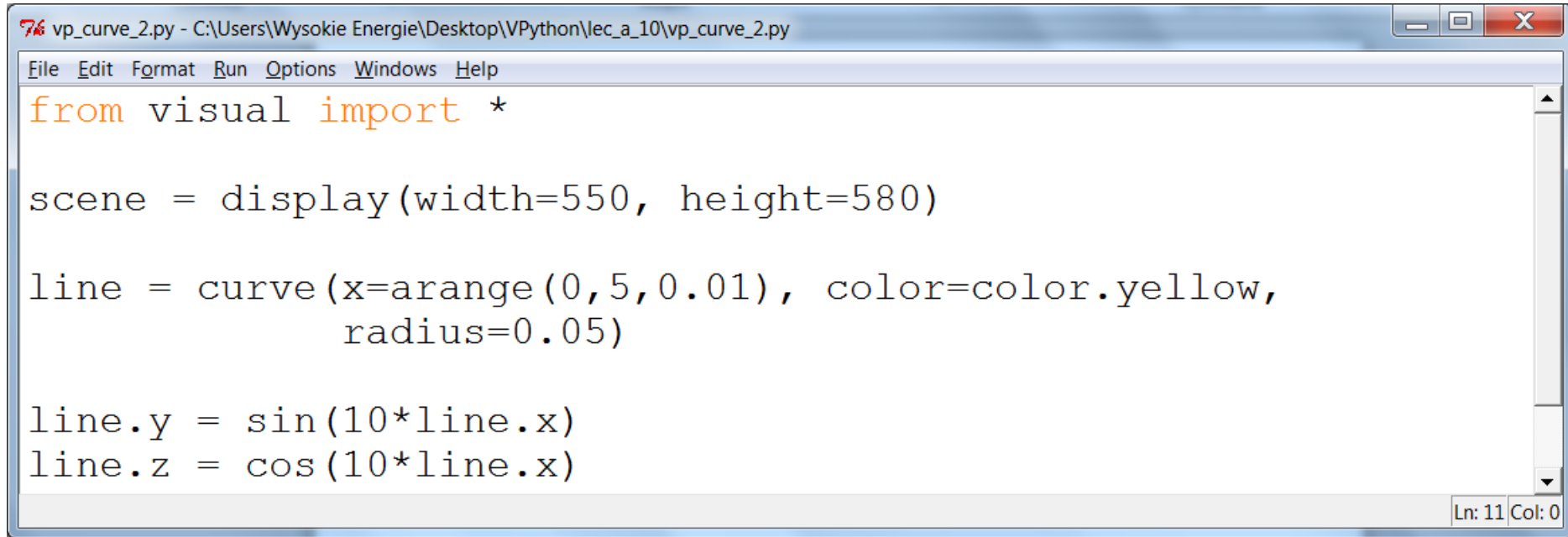
The status bar at the bottom right indicates "Ln: 13 Col: 0".

Jak nie podamy promienia to rysuje cienką linię

$(0,0,0) \rightarrow \text{vector}(0,0,0)$ w VPython 7



curve - x, y, z



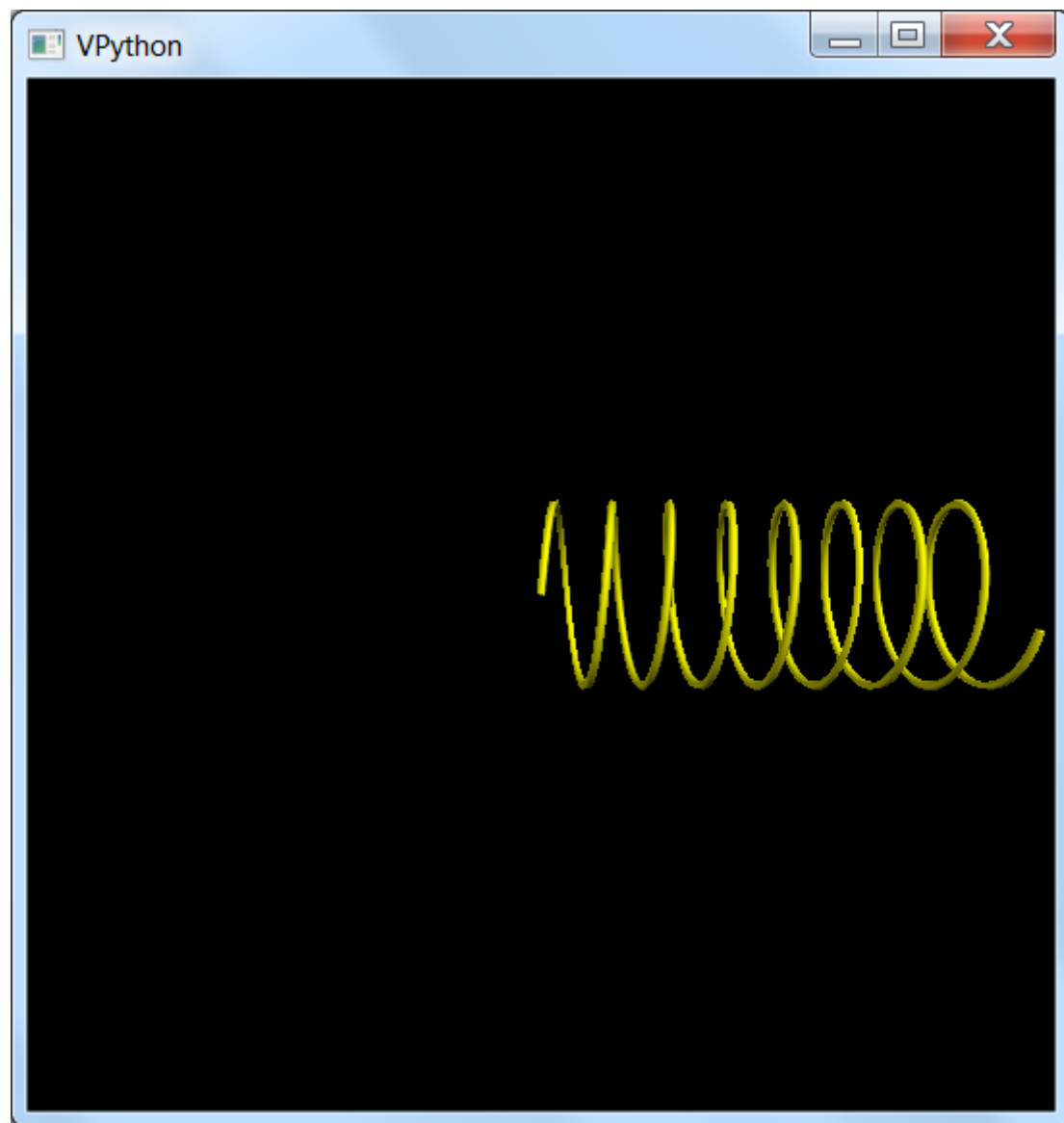
The image shows a screenshot of a Python IDE window titled "vp_curve_2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_a_10\vp_curve_2.py". The window contains a Python script that uses VPython to create a 3D scene and a curve. The script defines a scene with a width of 550 and a height of 580. It then creates a curve with a yellow color and a radius of 0.05, using a range of x values from 0 to 5 with a step of 0.01. The curve's y and z coordinates are defined as $\sin(10 \cdot x)$ and $\cos(10 \cdot x)$ respectively. The status bar at the bottom right indicates "Ln: 11 Col: 0".

```
76 vp_curve_2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_a_10\vp_curve_2.py
File Edit Format Run Options Windows Help
from visual import *

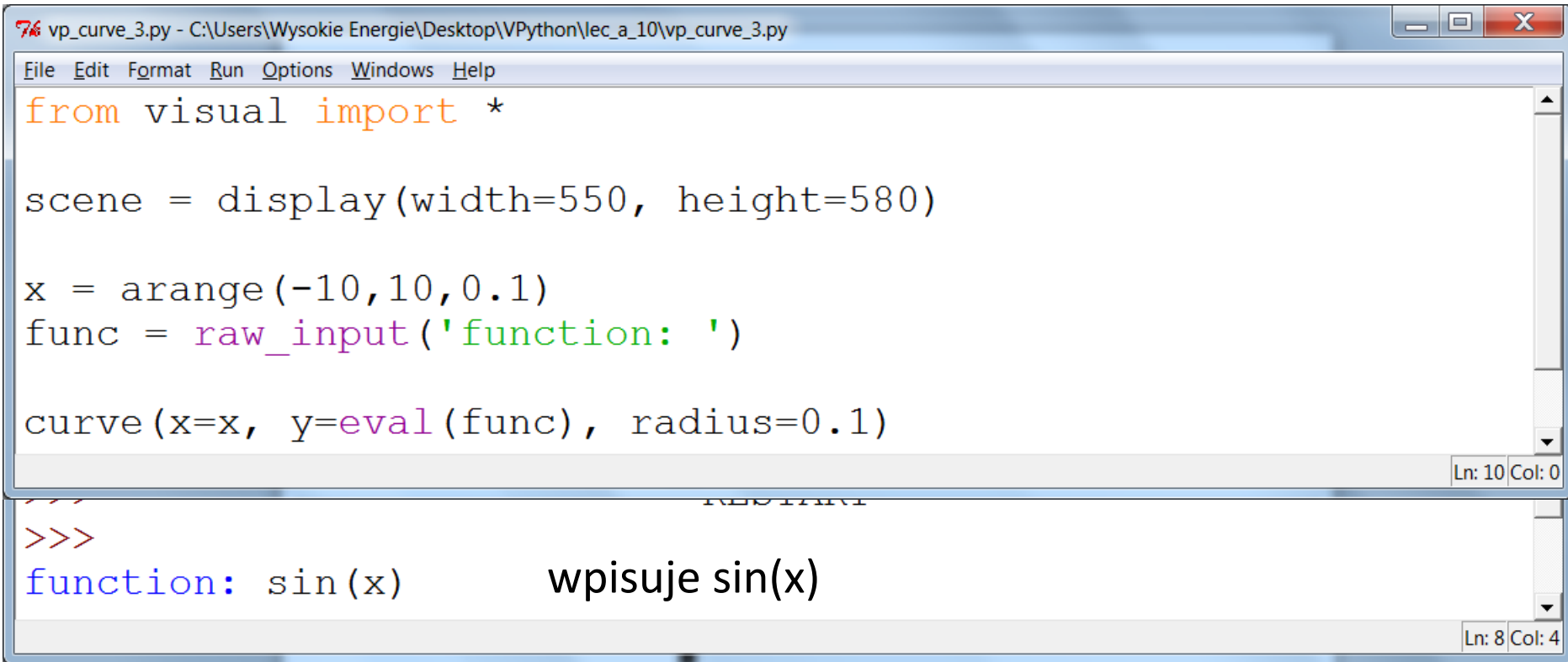
scene = display(width=550, height=580)

line = curve(x=arange(0,5,0.01), color=color.yellow,
             radius=0.05)

line.y = sin(10*line.x)
line.z = cos(10*line.x)
Ln: 11 Col: 0
```



curve



The image shows a screenshot of a Python IDE window titled "vp_curve_3.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_a_10\vp_curve_3.py". The window contains a Python script for plotting a curve using VPython. The script defines a scene, sets the width and height, creates an x-axis range, and defines a function to be plotted. The script is as follows:

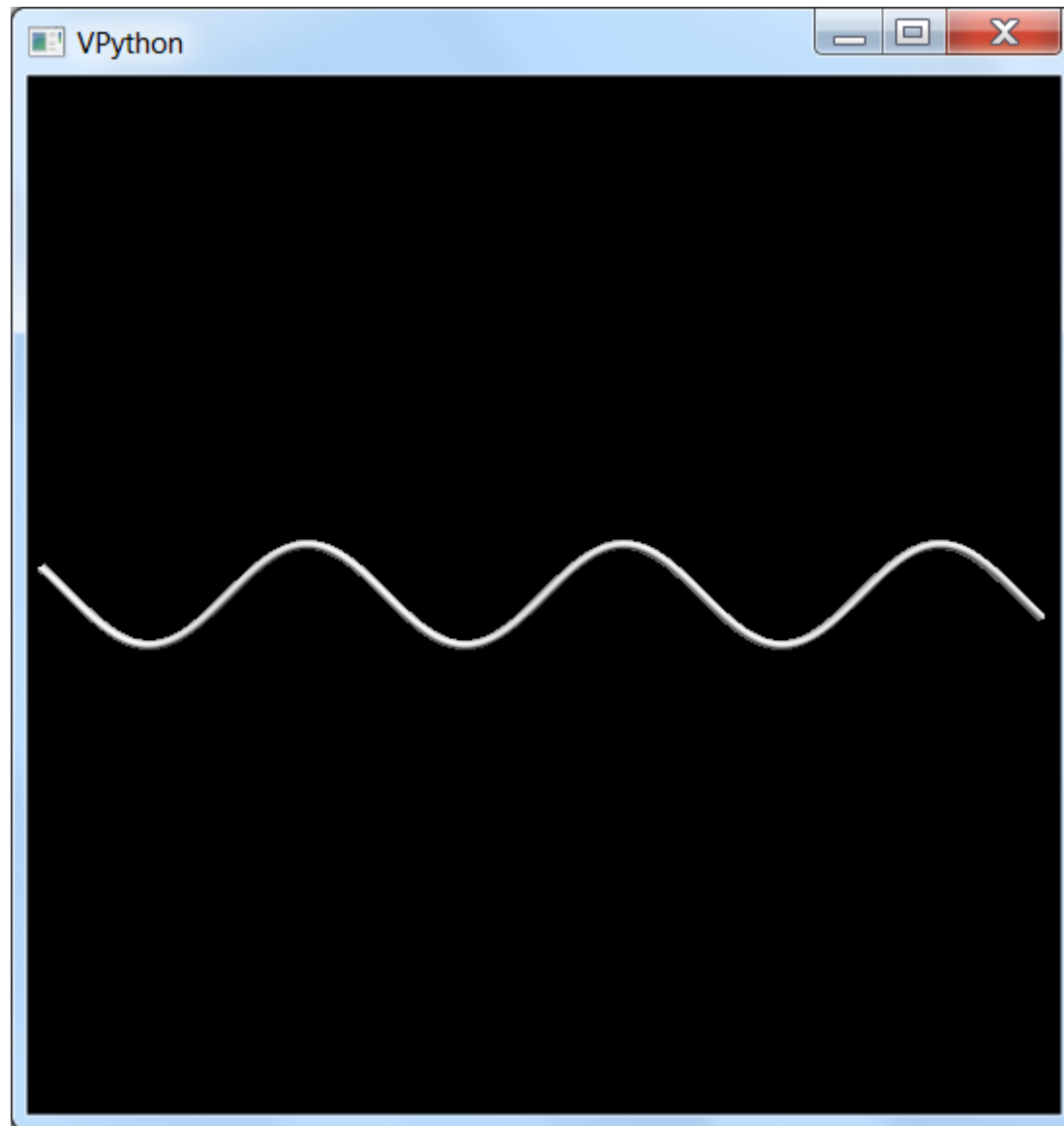
```
from visual import *

scene = display(width=550, height=580)

x = arange(-10,10,0.1)
func = raw_input('function: ')

curve(x=x, y=eval(func), radius=0.1)
```

The script is shown in a window with a menu bar (File, Edit, Format, Run, Options, Windows, Help) and a status bar (Ln: 10 Col: 0). Below the script, there is a prompt for the function name, and the user has entered "sin(x)". The prompt is "function: sin(x)" and the user input is "wpisuje sin(x)". The status bar for the input area shows "Ln: 8 Col: 4".



curve, ślad

76 vp_curve_4.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_a_10\vp_curve_4.py

File Edit Format Run Options Windows Help

```
from visual import *

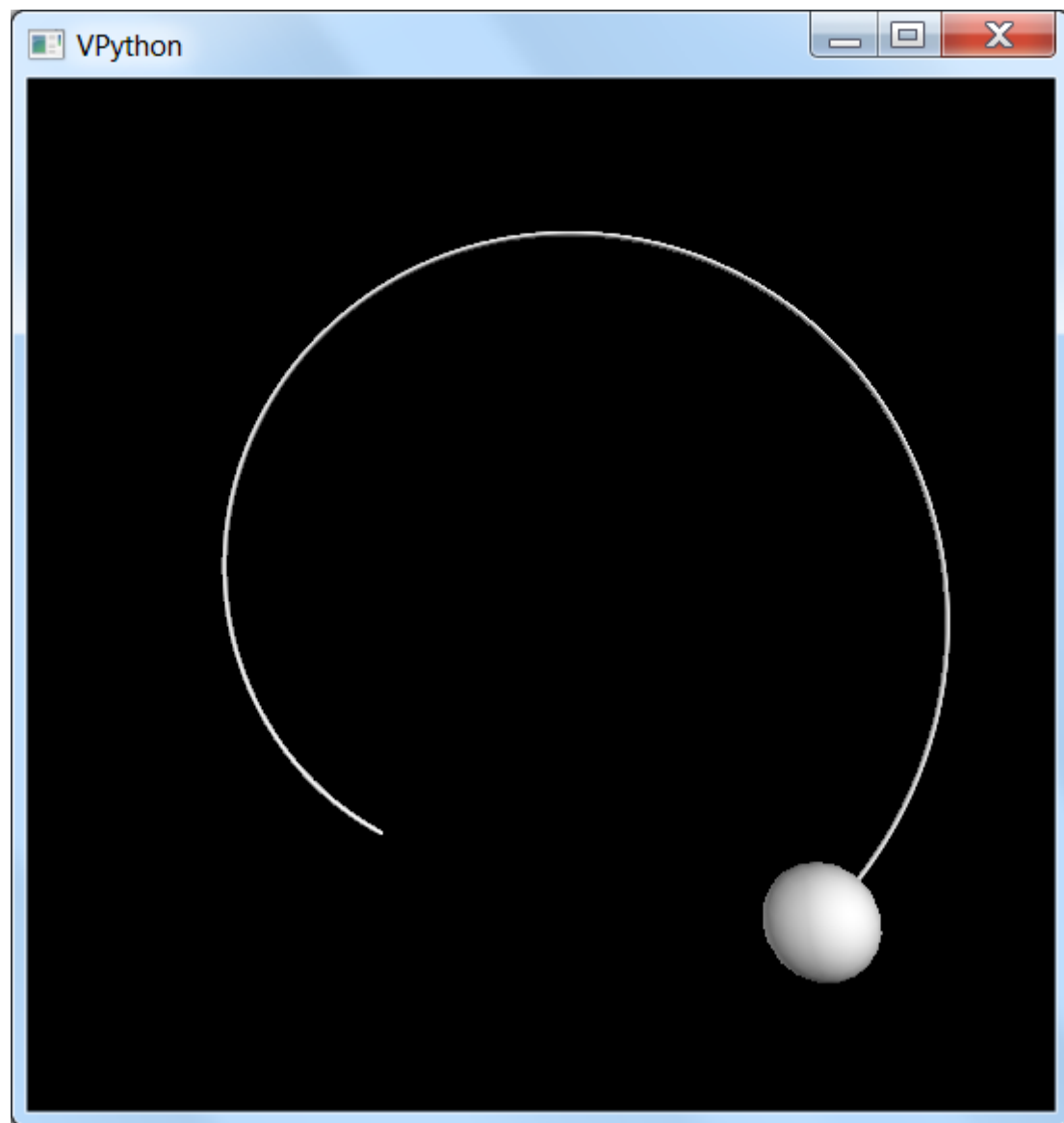
scene = display(width=550, height=580, range=9)

ba = sphere(pos=(0,5,0))
tra = curve(radius=0.05)

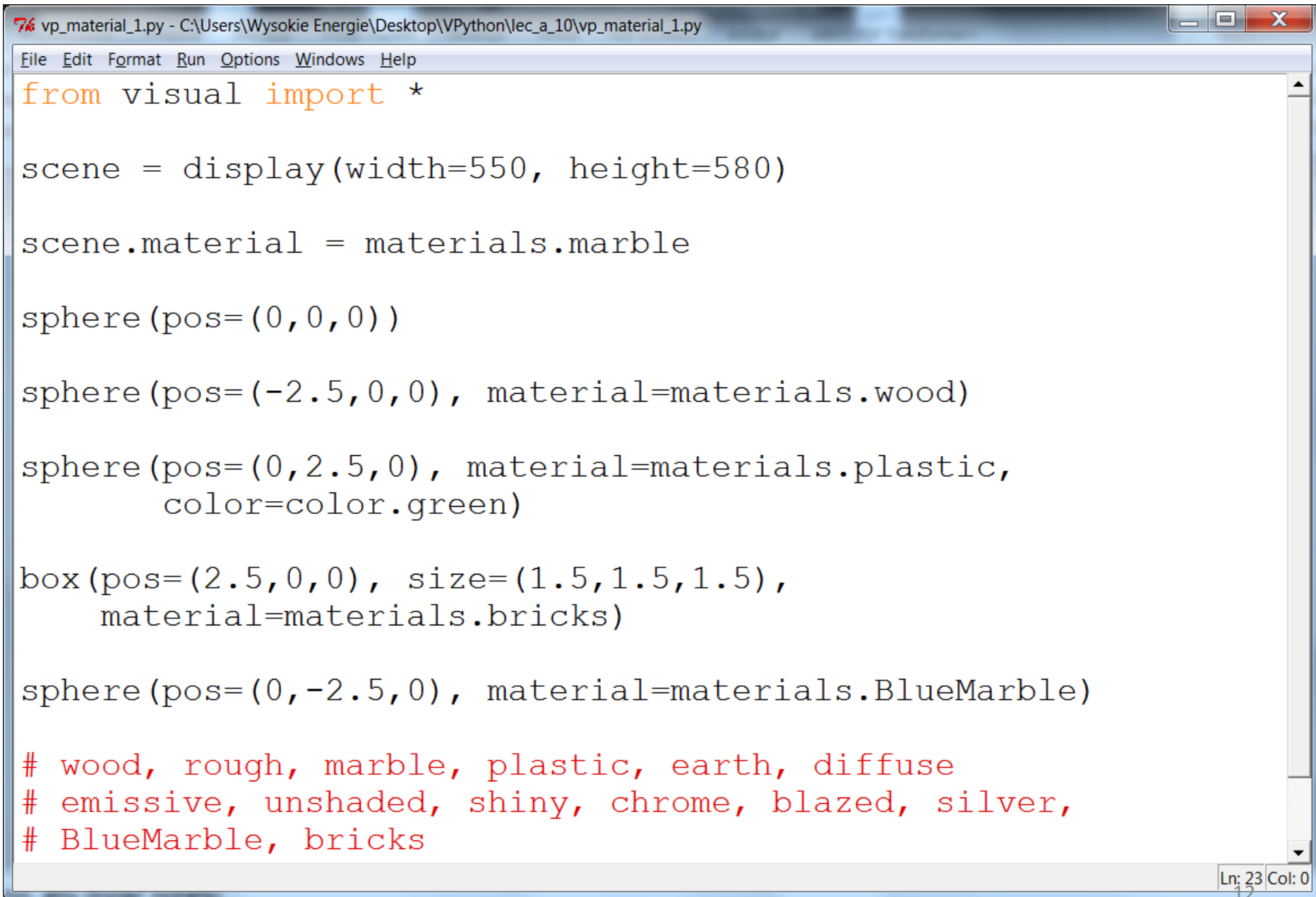
t = 0
while t<15:
    rate(200)
    ba.pos.x = 0.5*t*sin(t)
    ba.pos.y = 0.5*t*cos(t)
    tra.append(pos=ba.pos, retain=500)
    t += 0.01
```

trzymamy ostatnie
500 punktów

Ln: 17 Col: 0



Materiał (material), działa w VPython 6



```
vp_material_1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_a_10\vp_material_1.py
File Edit Format Run Options Windows Help
from visual import *

scene = display(width=550, height=580)

scene.material = materials.marble

sphere(pos=(0,0,0))

sphere(pos=(-2.5,0,0), material=materials.wood)

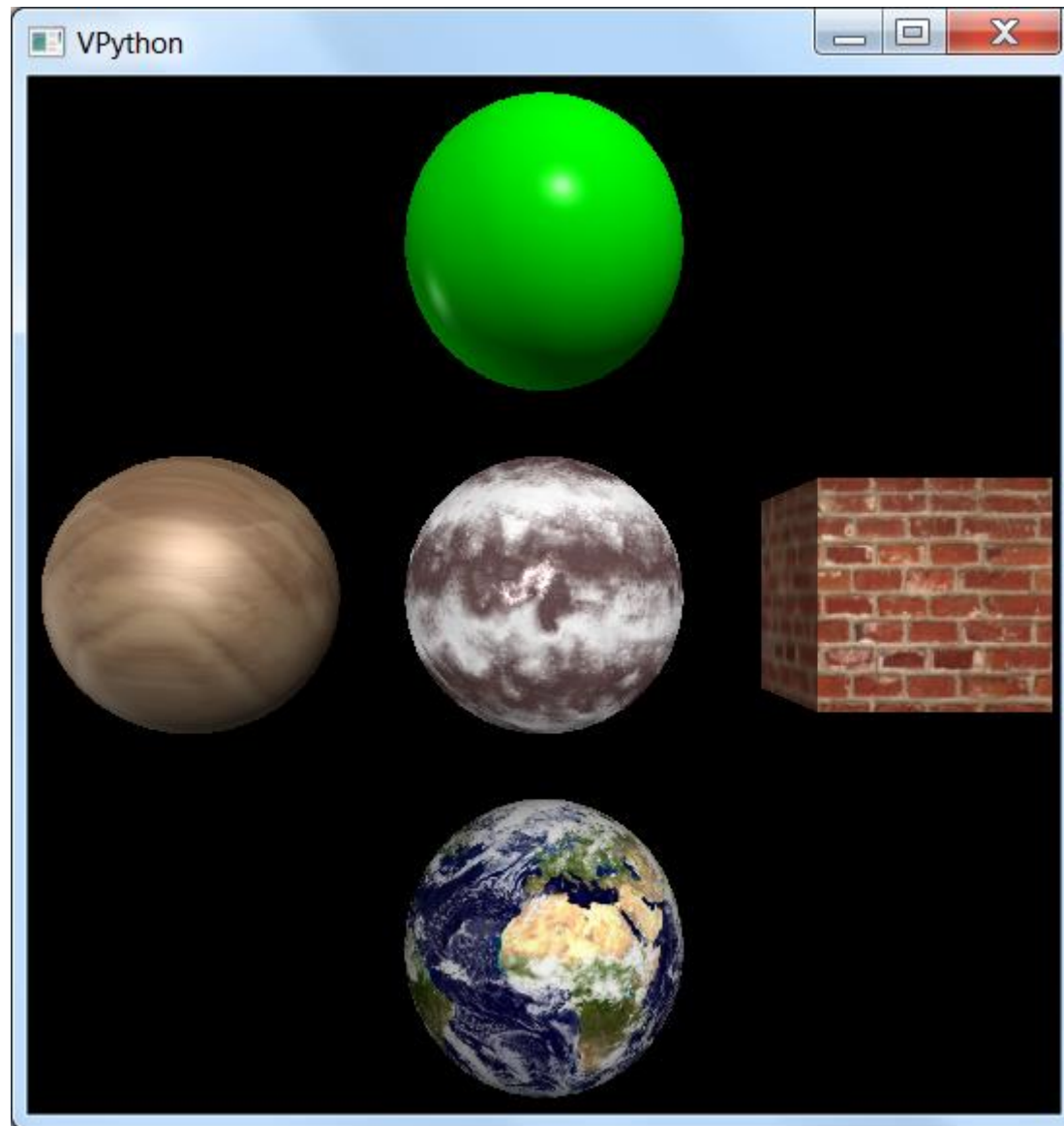
sphere(pos=(0,2.5,0), material=materials.plastic,
        color=color.green)

box(pos=(2.5,0,0), size=(1.5,1.5,1.5),
    material=materials.bricks)

sphere(pos=(0,-2.5,0), material=materials.BlueMarble)

# wood, rough, marble, plastic, earth, diffuse
# emissive, unshaded, shiny, chrome, blazed, silver,
# BlueMarble, bricks
```

Ln: 23 Col: 0



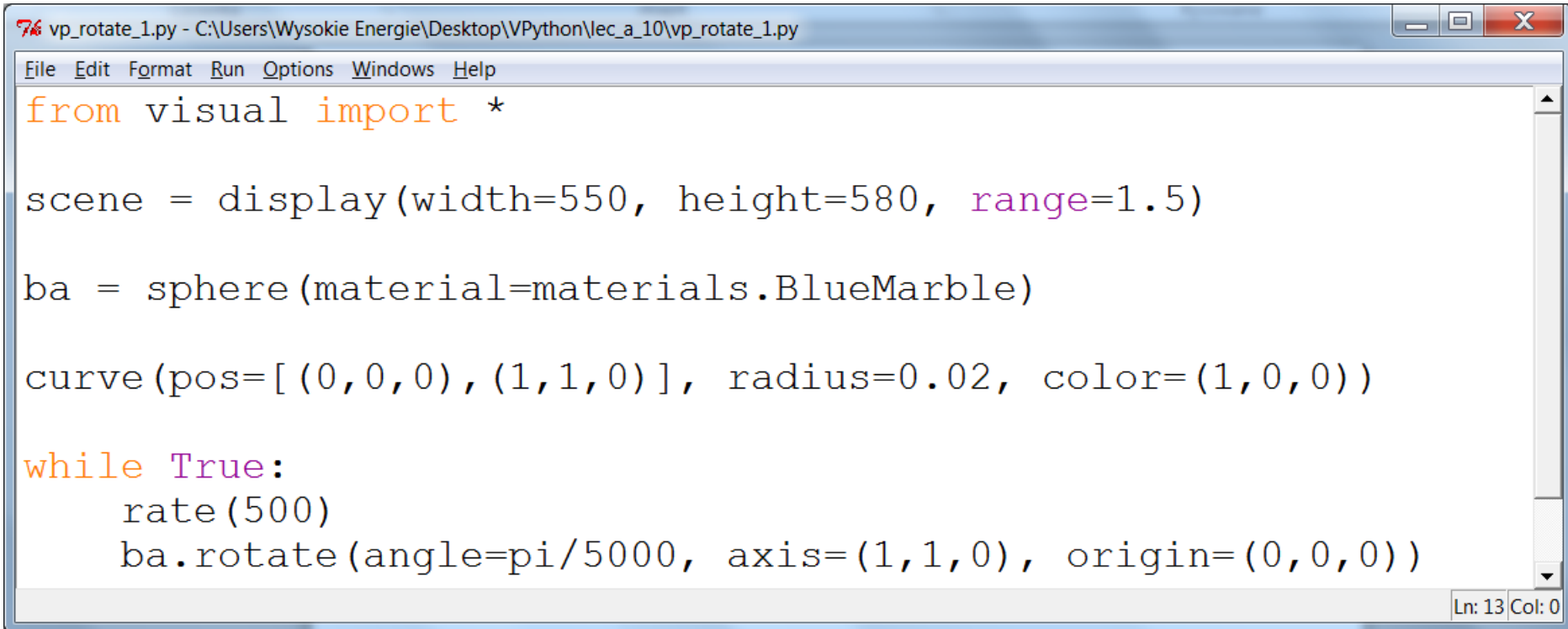
W VPython 7 mamy texture

<https://www.glowscript.org/docs/VPythonDocs/textures.html>

np.

box(texture=textures.stucco)

obrót (rotate)

A screenshot of a Python IDE window titled 'vp_rotate_1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_a_10\vp_rotate_1.py'. The window contains a Python script that sets up a 3D scene and rotates a sphere. The script includes imports, scene creation, object creation, and a rotation loop.

```
from visual import *

scene = display(width=550, height=580, range=1.5)

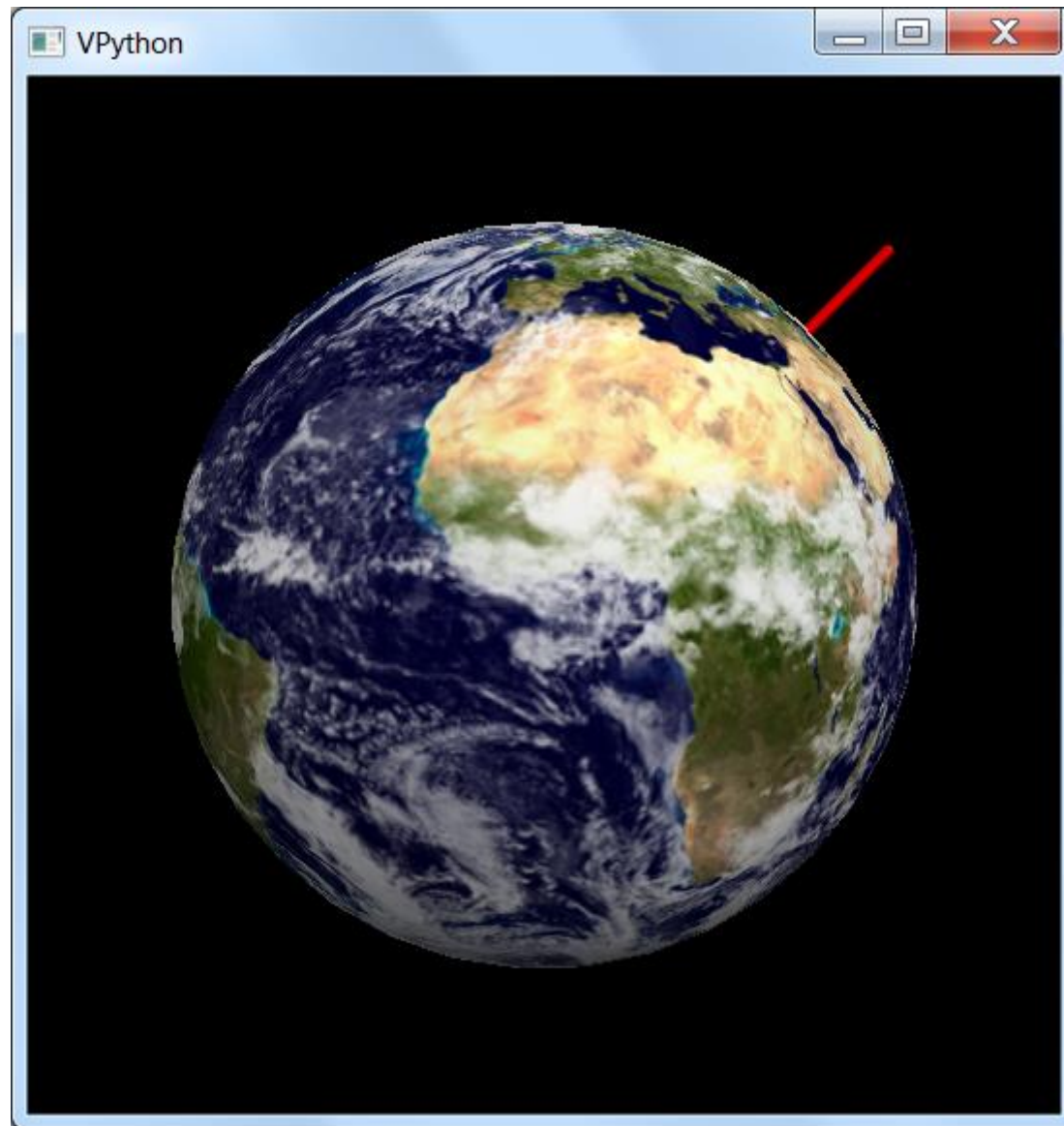
ba = sphere(material=materials.BlueMarble)

curve(pos=[(0,0,0), (1,1,0)], radius=0.02, color=(1,0,0))

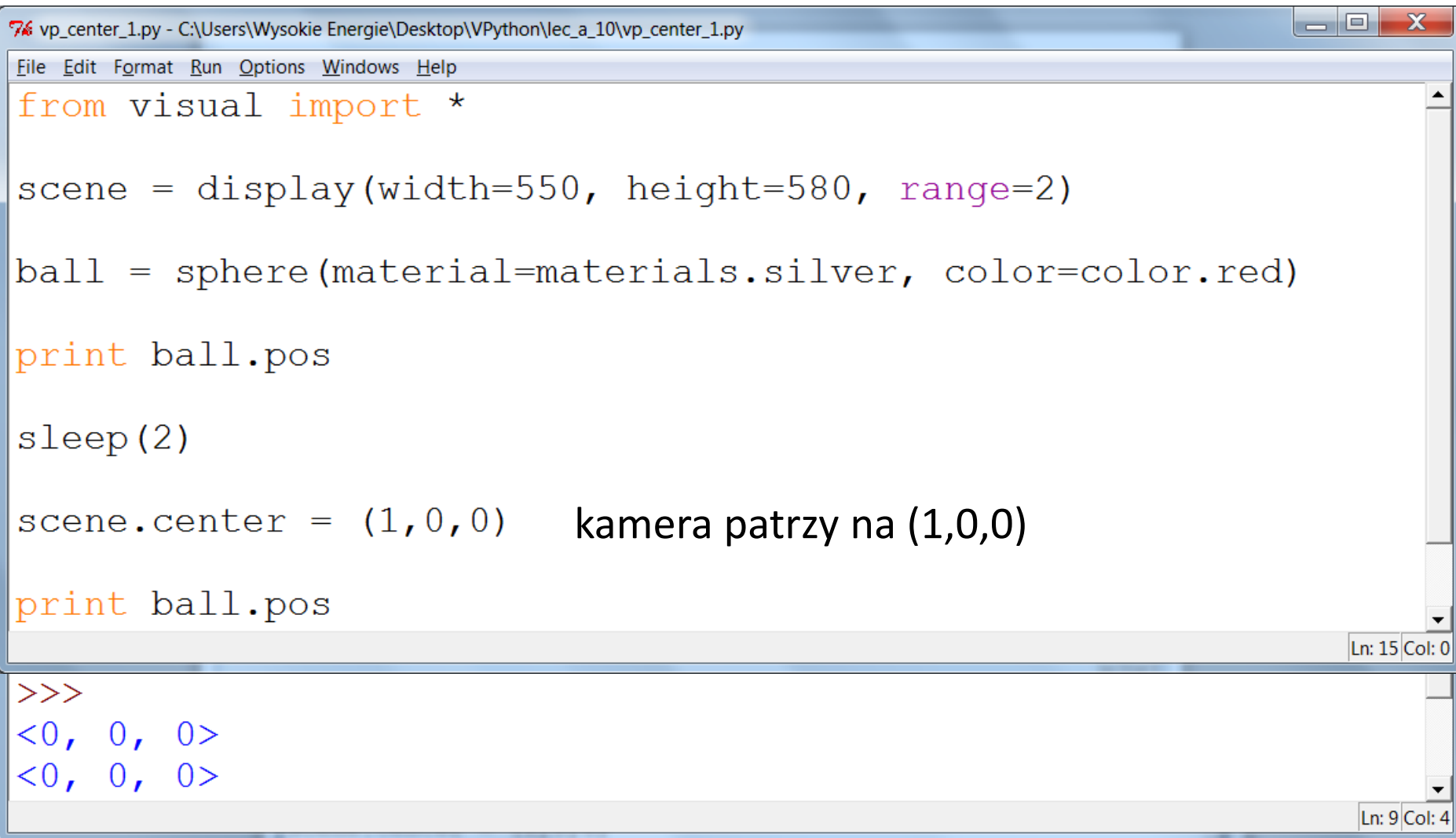
while True:
    rate(500)
    ba.rotate(angle=pi/5000, axis=(1,1,0), origin=(0,0,0))
```

The status bar at the bottom right indicates 'Ln: 13 Col: 0'.

Oś obrotu jest zdefiniowana przez linię pomiędzy origin i origin+axis



scene.center



The image shows a screenshot of a Python IDE window titled "vp_center_1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_a_10\vp_center_1.py". The window contains a Python script and its output.

```
from visual import *

scene = display(width=550, height=580, range=2)

ball = sphere(material=materials.silver, color=color.red)

print ball.pos

sleep(2)

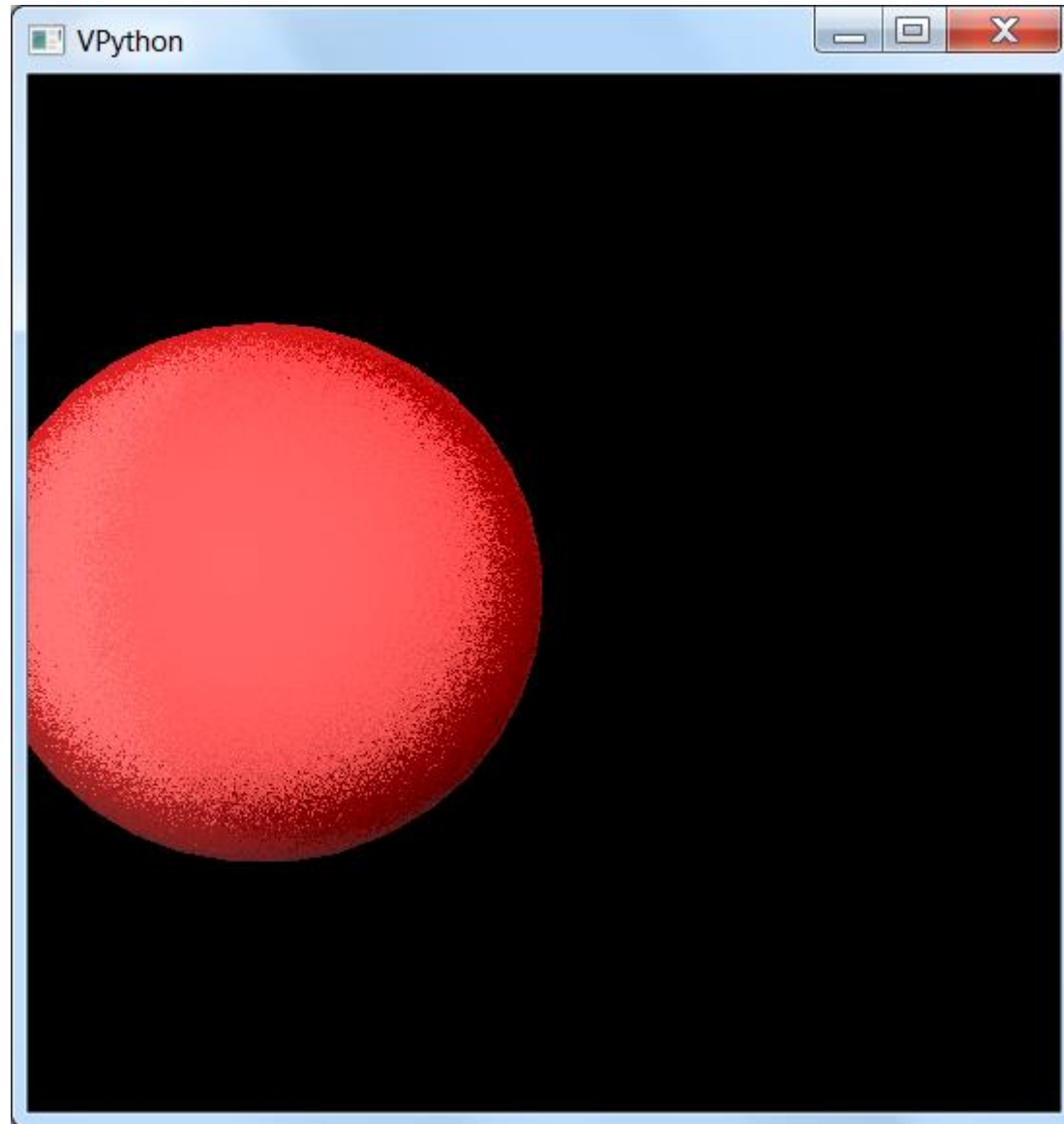
scene.center = (1,0,0)    kamera patrzy na (1,0,0)

print ball.pos
```

The output of the script is displayed in the bottom pane:

```
>>>
<0, 0, 0>
<0, 0, 0>
```

The script defines a scene with a width of 550 and a height of 580. It creates a red sphere with a silver material. The camera is positioned at the center of the scene, which is initially at (0,0,0). The script prints the position of the sphere, which is (0,0,0). The camera is then moved to (1,0,0) using the `scene.center` attribute. The script then prints the position of the sphere again, which is still (0,0,0).



center

```
76 vp_center_2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_a_10\vp_center_2.py
File Edit Format Run Options Windows Help
from visual import *

scene = display(width=550, height=580, range=5)

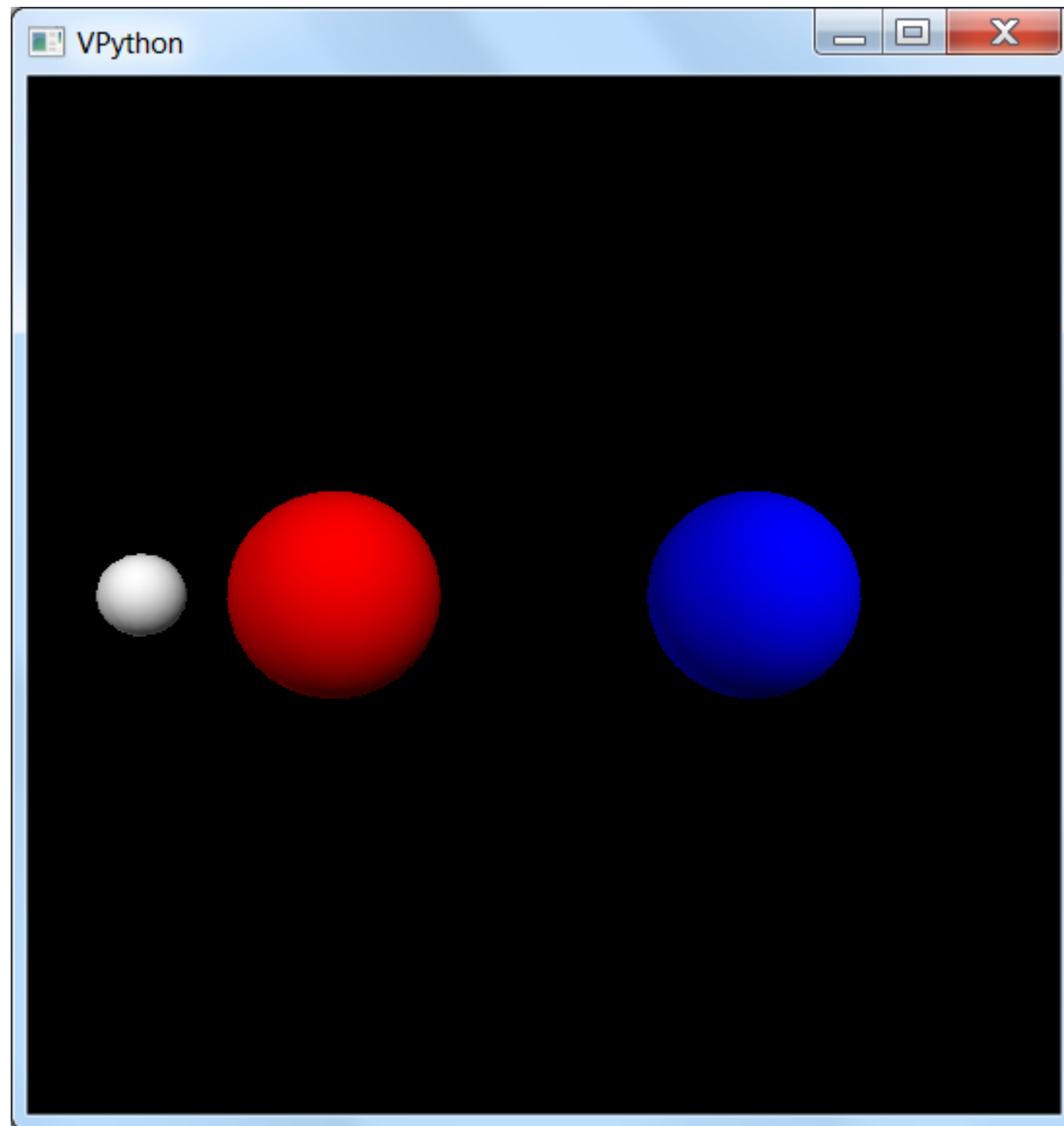
ba1 = sphere(pos=(-2,0,0), color=color.red)
ba2 = sphere(pos=(2,0,0), color=color.blue)
ba3 = sphere(pos=(0,0,0), radius=0.4)

scene.autoscale = False

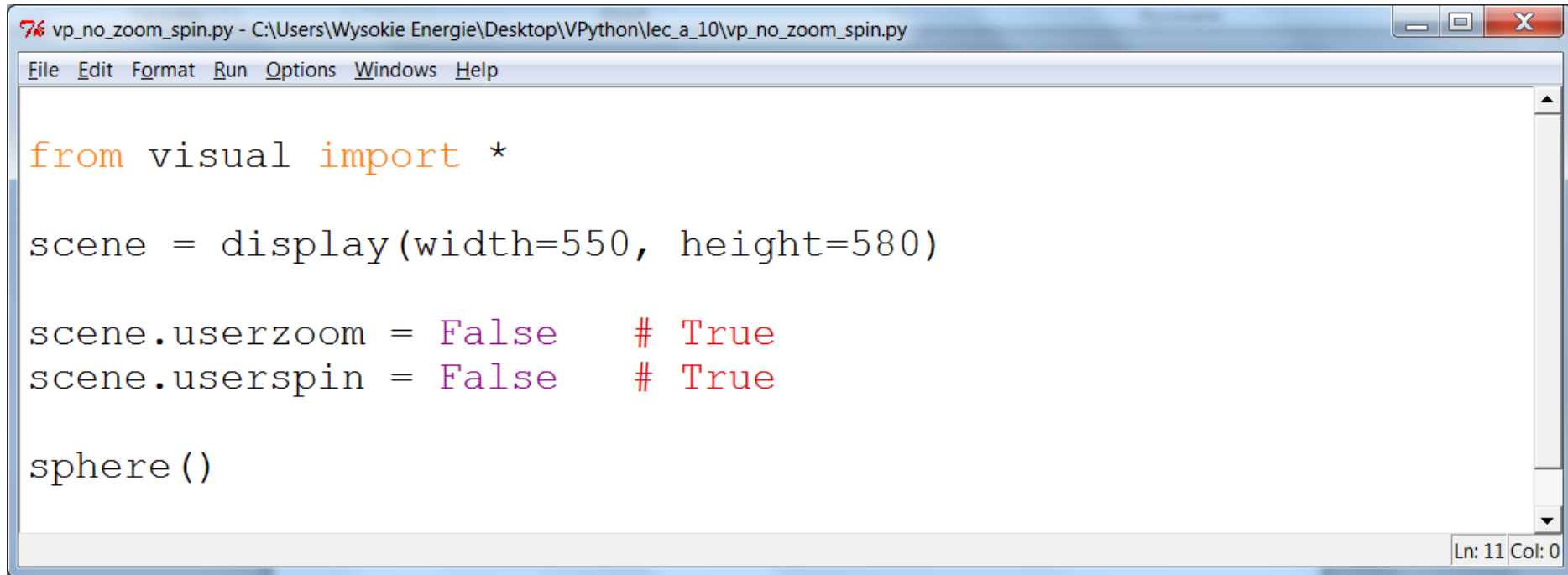
dt = 0.01
while True:
    rate(200)
    ba1.pos.x += 1*dt
    ba2.pos.x += 1*dt
    scene.center = (ba1.pos + ba2.pos)/2
```

Ln: 19 Col: 0

Przydatne gdy kamera ma śledzić środek masy



kontrola myszką

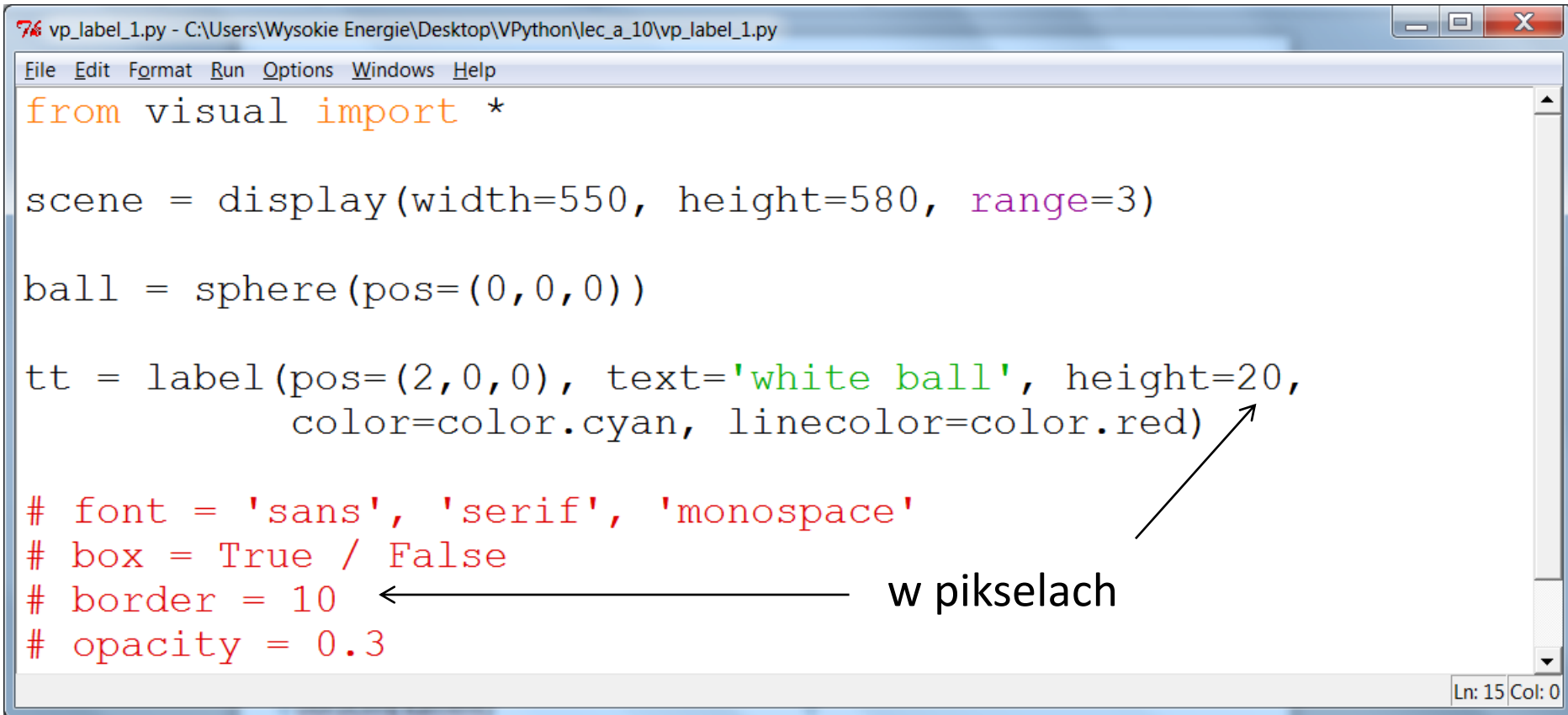


The image shows a screenshot of a Python IDE window. The title bar at the top reads "vp_no_zoom_spin.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_a_10\vp_no_zoom_spin.py". Below the title bar is a menu bar with the following options: File, Edit, Format, Run, Options, Windows, and Help. The main text area contains the following Python code:

```
from visual import *  
  
scene = display(width=550, height=580)  
  
scene.userzoom = False    # True  
scene.userspin = False    # True  
  
sphere()
```

At the bottom right of the window, the status bar indicates "Ln: 11 Col: 0".

label



```
vp_label_1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_a_10\vp_label_1.py
File Edit Format Run Options Windows Help

from visual import *

scene = display(width=550, height=580, range=3)

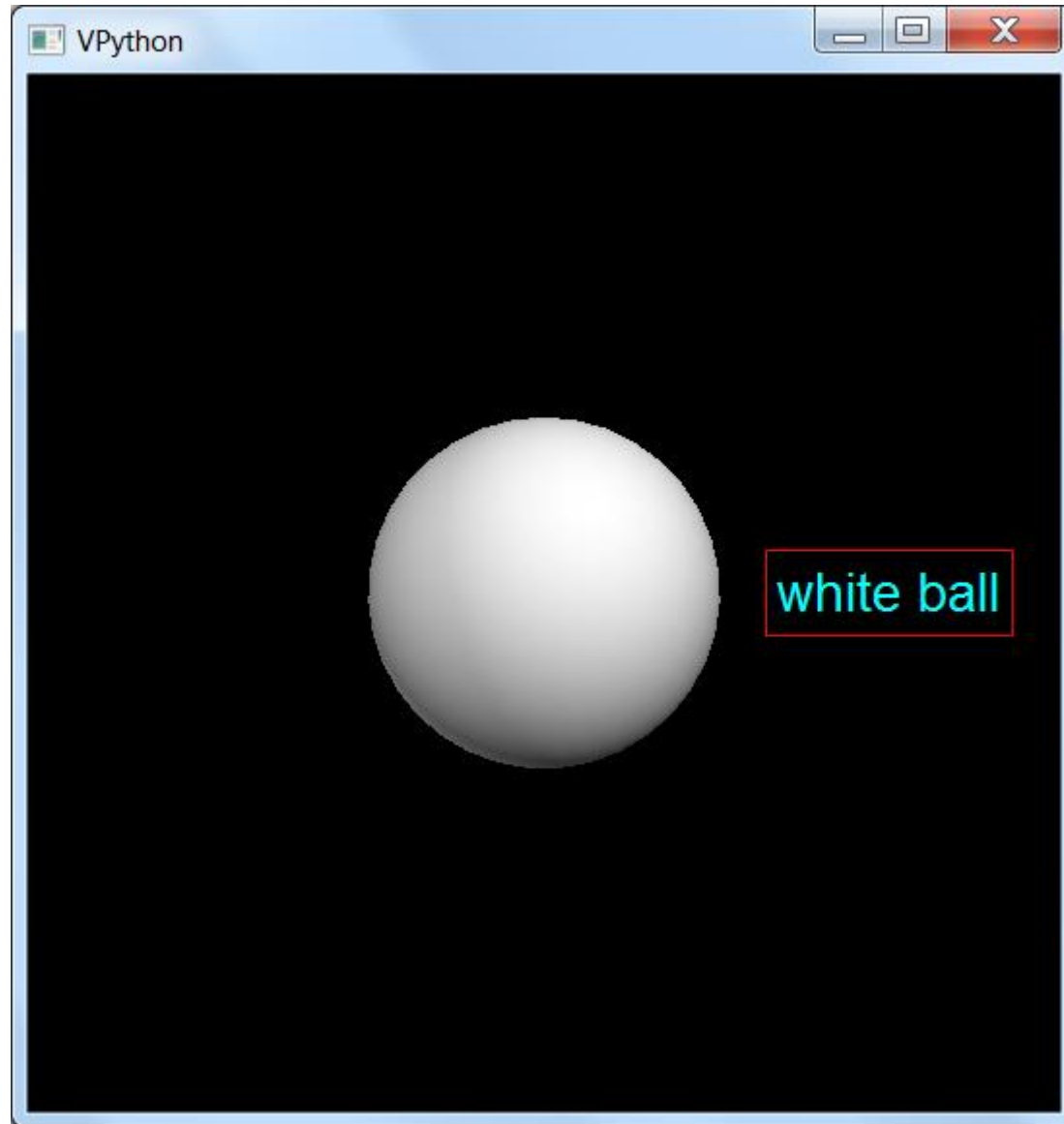
ball = sphere(pos=(0,0,0))

tt = label(pos=(2,0,0), text='white ball', height=20,
           color=color.cyan, linecolor=color.red)

# font = 'sans', 'serif', 'monospace'
# box = True / False
# border = 10 ← w pikselach
# opacity = 0.3
```

Ln: 15 Col: 0

label() zawsze patrzy się w ekran i rozmiar się nie zmienia gdy przybliżamy/obracamy kamerkę



label

76 vp_label_2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_a_10\vp_label_2.py

File Edit Format Run Options Windows Help

```
from visual import *

scene = display(width=550, height=580, range=3)

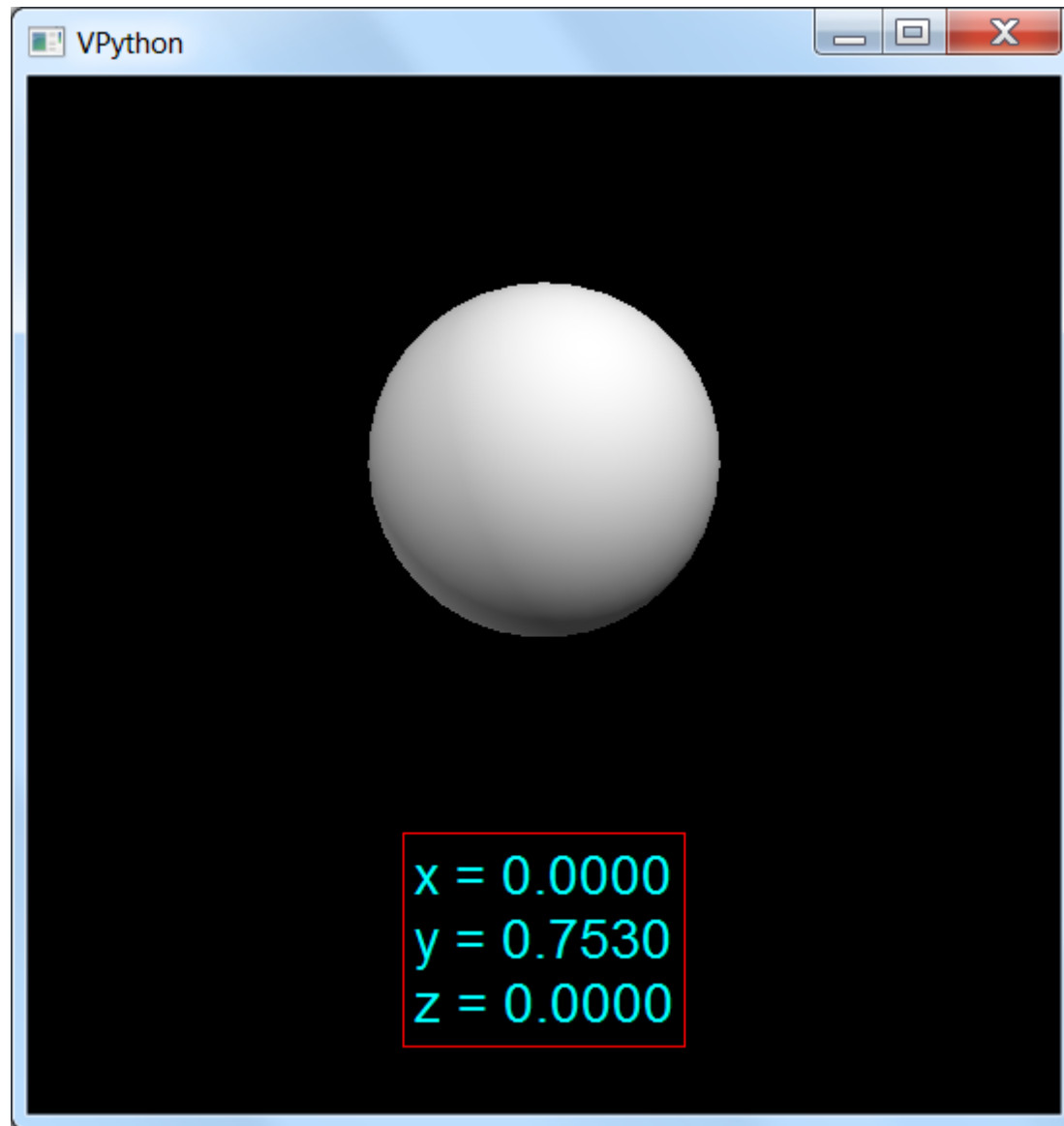
ball = sphere(pos=(0,0,0), radius=1)

tt = label(pos=(0,-2,0), text='ball', height=20,
           color=color.cyan, linecolor=color.red)

sleep(1)

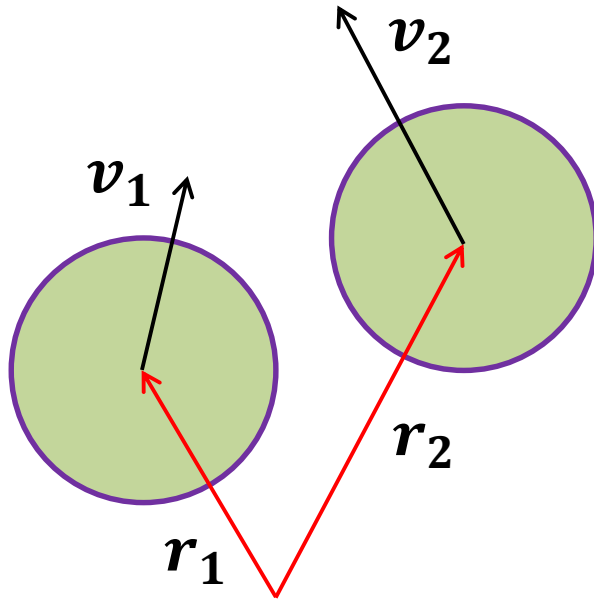
dt = 0.01
while 1:
    rate(100)
    ball.pos.y += 0.1*dt
    tt.text = 'x = ' + '%.4f'% (ball.pos.x) + '\n' + \
              'y = ' + '%.4f'% (ball.pos.y) + '\n' + \
              'z = ' + '%.4f'% (ball.pos.z)
```

Ln: 21 Col: 0

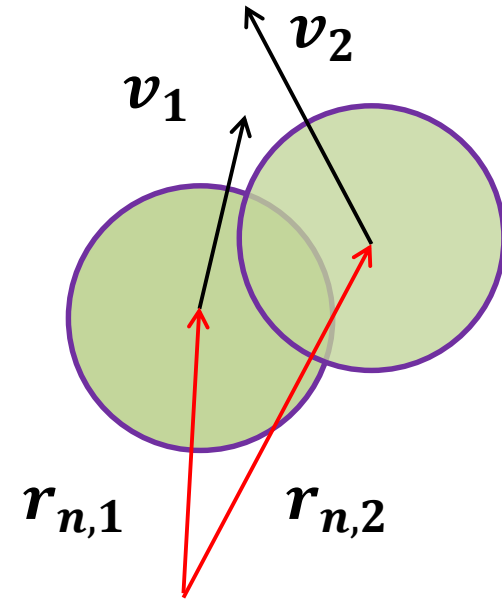


Zderzenia kul

czas: t



czas: $t + dt$



$$r_{n,i} = r_i + v_i \cdot dt$$

Jeśli $|r_{n,1} - r_{n,2}| < (R_1 + R_2)$ to mamy zderzenie

R_1 , R_2 to promienie kul

Wracamy w czasie o dt' tak aby kule dotykały się powierzchniami
łatwo policzyć dt' :

$$a = |\mathbf{v}_1 - \mathbf{v}_2|^2 \quad a > 0, c < 0$$

$$dt' = \frac{-b + \sqrt{\Delta}}{2a}$$

$$b = -2(\mathbf{r}_{n,1} - \mathbf{r}_{n,2}) \circ (\mathbf{v}_1 - \mathbf{v}_2)$$

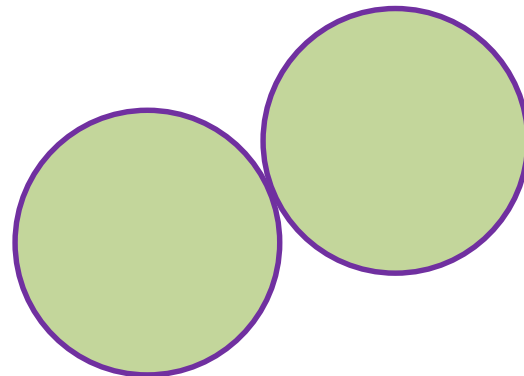
$$c = |\mathbf{r}_{n,1} - \mathbf{r}_{n,2}|^2 - (R_1 + R_2)^2$$

$$\Delta = b^2 - 4ac$$

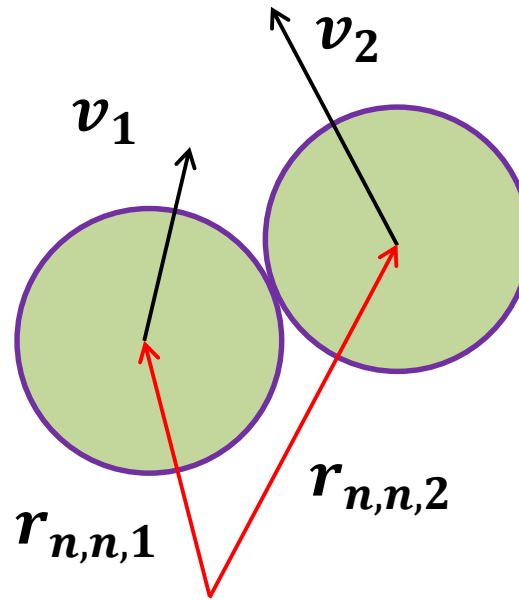
R_1 , R_2 to promienie

if $a=0$: continue

if $\Delta < 0$: continue

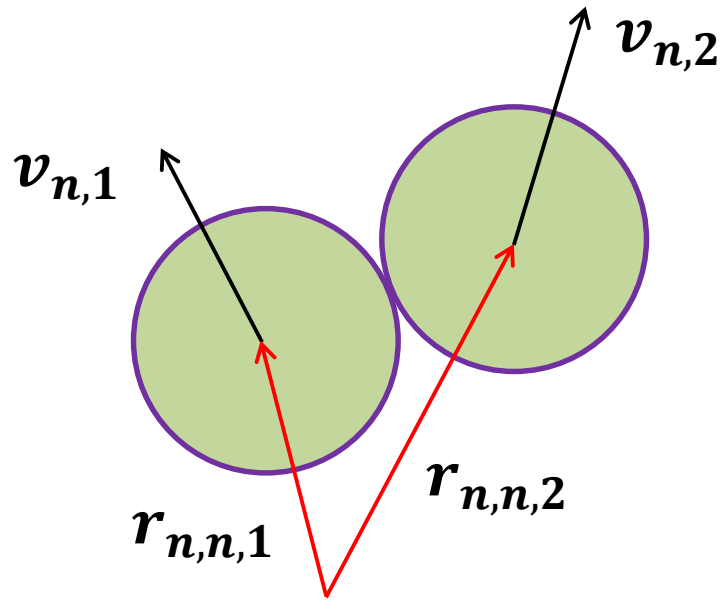


Po powrocie w czasie o dt' mamy:



$$r_{n,n,i} = r_{n,i} - v_i \cdot dt'$$

Zderzenie: liczymy nowe prędkości

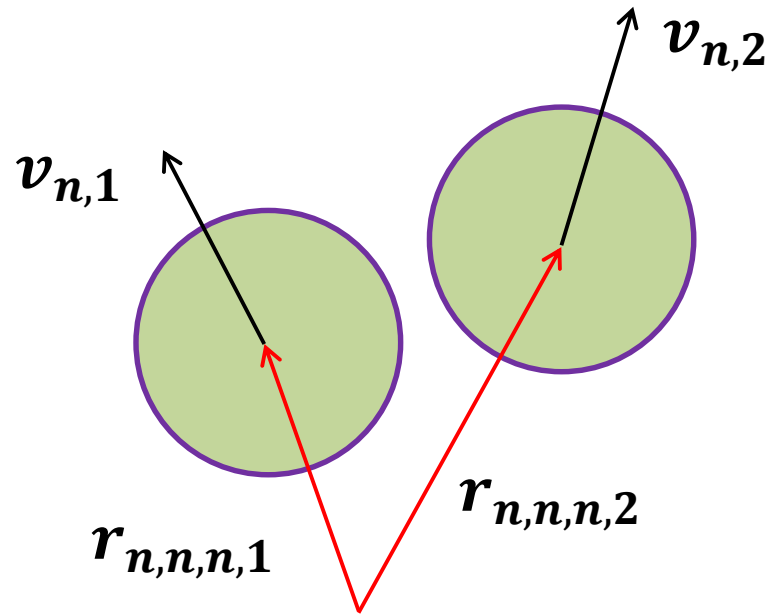


m_i – masa kuli

$$\mathbf{v}_{n,1} = \mathbf{v}_1 - 2 \frac{m_2}{m_1 + m_2} \left[(\mathbf{v}_1 - \mathbf{v}_2) \circ \frac{\mathbf{r}_{n,n,1} - \mathbf{r}_{n,n,2}}{|\mathbf{r}_{n,n,1} - \mathbf{r}_{n,n,2}|} \right] \frac{\mathbf{r}_{n,n,1} - \mathbf{r}_{n,n,2}}{|\mathbf{r}_{n,n,1} - \mathbf{r}_{n,n,2}|}$$

$$\mathbf{v}_{n,2} = \mathbf{v}_2 + 2 \frac{m_1}{m_1 + m_2} \left[(\mathbf{v}_1 - \mathbf{v}_2) \circ \frac{\mathbf{r}_{n,n,1} - \mathbf{r}_{n,n,2}}{|\mathbf{r}_{n,n,1} - \mathbf{r}_{n,n,2}|} \right] \frac{\mathbf{r}_{n,n,1} - \mathbf{r}_{n,n,2}}{|\mathbf{r}_{n,n,1} - \mathbf{r}_{n,n,2}|}$$

Idziemy do przodu o dt' (bo wcześniej się cofnęliśmy):



$$r_{n,n,n,i} = r_{n,n,i} + v_{n,i} \cdot dt'$$