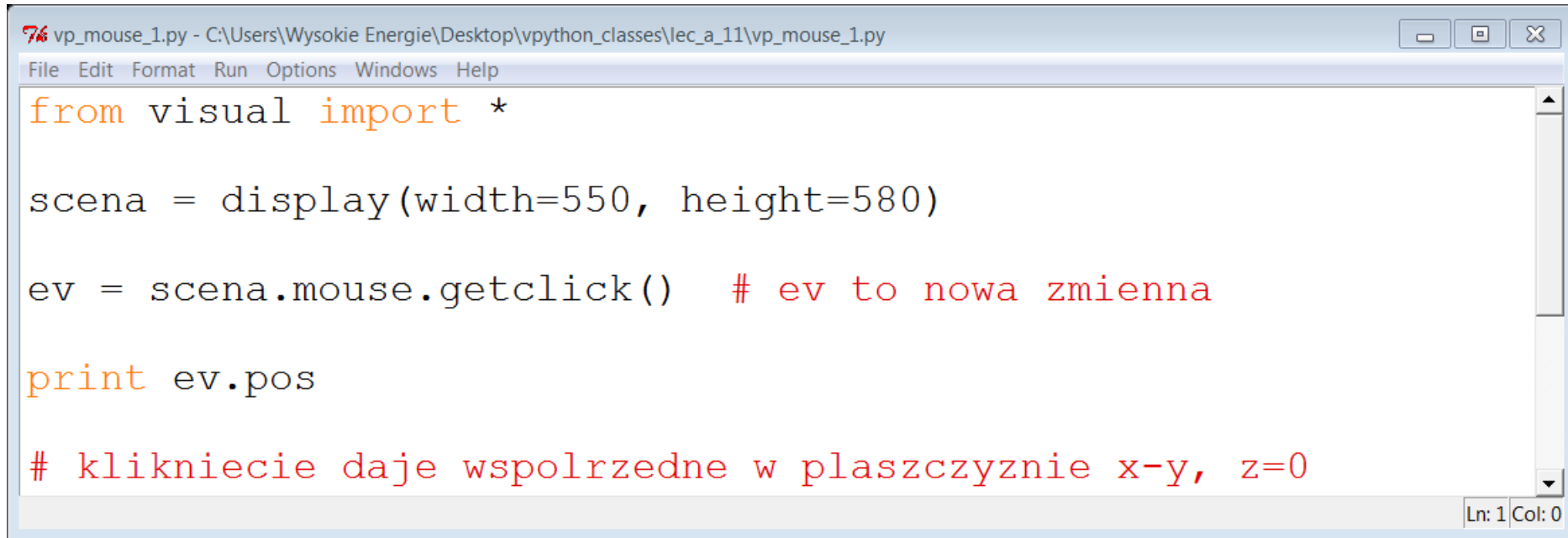


# VPython - symulacje fizyczne z grafiką 3D dla każdego

## wykład 12

Dr hab. Adam Bzdak, prof. AGH

# Mysz, współrzędne kliknięcia



```
76 vp_mouse_1.py - C:\Users\Wysokie Energie\Desktop\vpypython_classes\lec_a_11\vp_mouse_1.py
File Edit Format Run Options Windows Help

from visual import *

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

ev = scena.mouse.getclick() # ev to nowa zmienna

print ev.pos

# klikiecie daje wspolrzedne w plaszczyznie x-y, z=0

Ln: 1 Col: 0
```

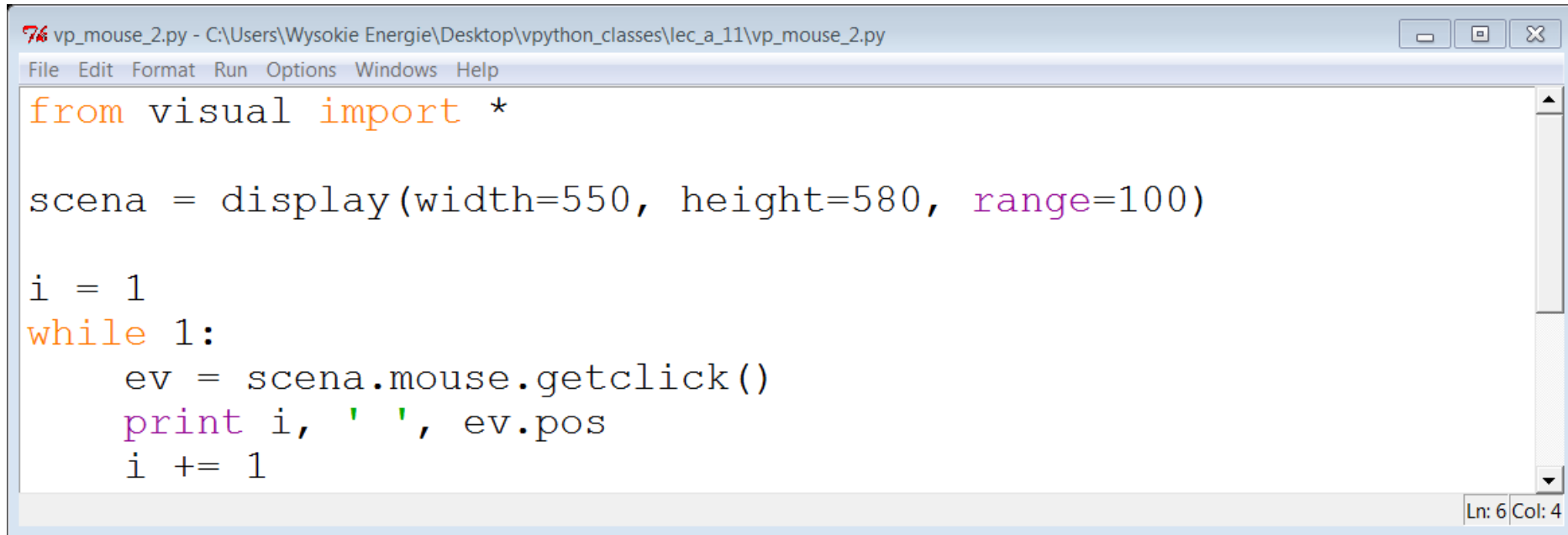
lub ogólniej, punkt jest w płaszczyźnie równoległej do ekranu

**scene.waitfor('click')** dobre w VPython 6 i VPython 7

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

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

# Mysz, współrzędne kliknięcia



The image shows a screenshot of a Python IDE window. The title bar indicates the file is 'vp\_mouse\_2.py' located at 'C:\Users\Wysokie Energie\Desktop\vpypython\_classes\lec\_a\_11\vp\_mouse\_2.py'. The menu bar includes 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The code editor contains the following Python code:

```
from visual import *

scena = display(width=550, height=580, range=100)

i = 1
while 1:
    ev = scena.mouse.getclick()
    print i, ' ', ev.pos
    i += 1
```

The status bar at the bottom right shows 'Ln: 6 Col: 4'.

# Mysz, tworzenie obiektów

76 vp\_mouse\_3.py - C:\Users\Wysokie Energie\Desktop\vpypython\_classes\lec\_a\_11\vp\_mouse\_3.py

File Edit Format Run Options Windows Help

```
from visual import *

scena = display(width=550, height=580, range=10)

while 1:
    ev = scena.mouse.getclick()
    wsp = ev.pos # wspolrzedna klikniecia
    print wsp
    sphere(pos=wsp, radius=1)

# uzywajac wspolrzednej klikniecia mozna tworzyc obiekty
# w tym punkcie

# aby utworzyc obiekt poza plaszczyzna x-y, nalezy obroc
# kamerke za pomoca prawego przycisku myszy

# mozna oddalic kamerke i tez dziala
```

Ln: 1 Col: 0

# Mysz, test programu

76 vp\_mouse\_5.py - C:\Users\Wysokie Energie\Desktop\vpypython\_classes\lec\_a\_11\vp\_mouse\_5.py

File Edit Format Run Options Windows Help

```
from visual import *

# scena.mouse.getclick() jest wygodne w sprawdzaniu
# programu

scena = display(width=550, height=580, range=10)

kula = sphere(pos=(0,0,0), radius=2)

ev = scena.mouse.getclick()
# program czeka az klikniemy myszka

while 1:
    rate(500)
    kula.pos += vector(1,0,0)*0.01
```

Ln: 5 Col: 0

# make\_trail, ślad

vp\_trail.py - C:\Users\Wysokie Energie\Desktop\vppython\_classes\lec\_a\_11\vp\_trail.py

File Edit Format Run Options Windows Help

```
from visual import *

scena = display(width=550, height=580, range=10)

kula = sphere(pos=(0,5,0), make_trail=True,
               trail_type='points', interval=200, retain=150)

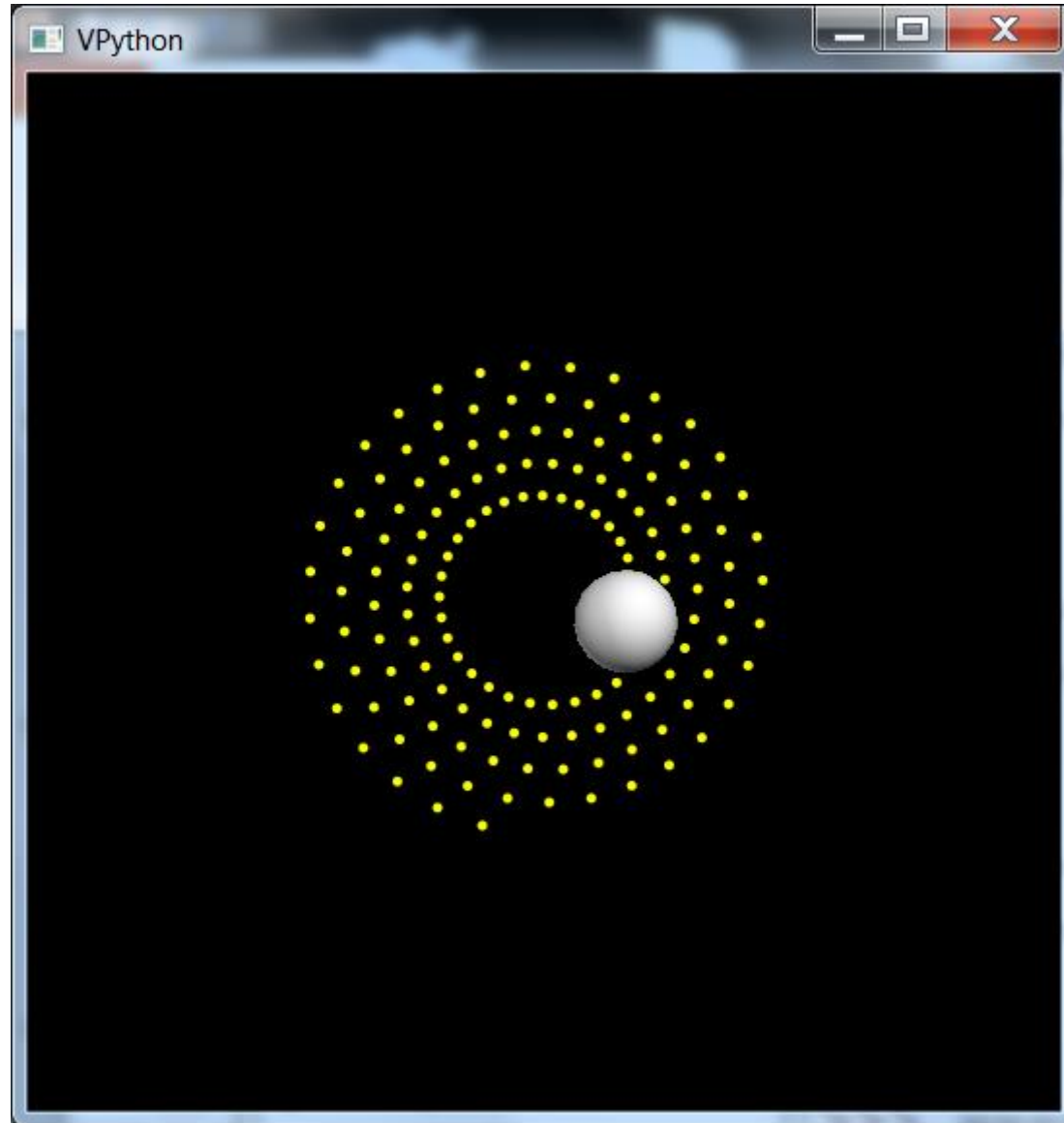
# trail_type = 'curve' / 'points'
# interval=X, punkt jest dodany co X krokow
# retain=Y, Y ostatnich punktow jest trzymany w sladzie

kula.trail_object.color = color.yellow

t = 0
while 1:
    rate(5000)
    A = 5-0.1*t
    kula.pos = (A*sin(t), A*cos(t), 0)
    t += 0.001
```

Ln: 19 Col: 14

make\_trail, ślad



# factorial, combin

vp\_factorial.py - C:\Users\Wysokie Energie\Desktop\vpypython\_classes\lec\_a\_11\vp\_factorial.py

File Edit Format Run Options Windows Help

```
from visual import *
import visual.factorial as vf

print vf.factorial(10)
print vf.combin(10,2)  # 10!/(2!*(10-2)!)

print

print vf.factorial(1000)
print vf.combin(1000,998)

# pozwala na liczenie  $a!/(b!*(a-b)!)$  dla dużych liczb
```

Ln: 8 Col: 0

```
>>> ===== RESTART =====
```

```
>>>
```

```
3628800.0
```

```
45.0
```

```
inf
```

```
499500.0
```

Ln: 12 Col: 4



# frame, rama

```
74 vp_frame_1.py - C:\Users\Wysokie Energie\Desktop\vp_python_classes\lec_a_11\vp_frame_1.py
File Edit Format Run Options Windows Help

from visual import *

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

f = frame()
# automatycznie pos=(0,0,0), axis=(1,0,0)

kula1 = sphere(frame=f, pos=(-1,0,0), radius=0.4)
kula2 = sphere(frame=f, pos=(1,0,0), radius=0.4)

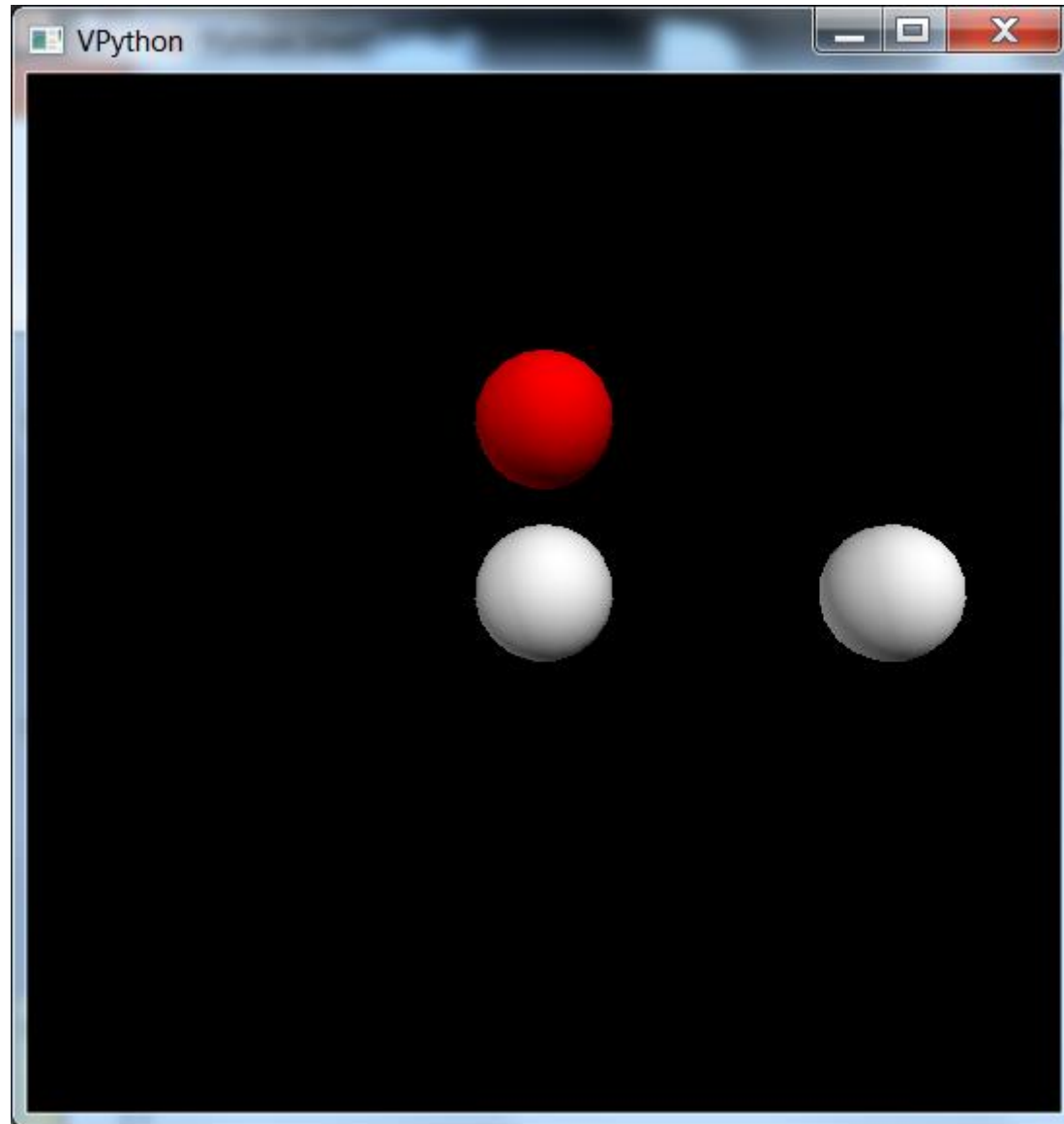
kula3 = sphere(pos=(0,1,0), radius=0.4, color=color.red)

sleep(2)          brak frame dla kula3

f.pos = (1,0,0)   # zmiana polozenia frame
```

Ln: 1 Col: 0

efekt końcowy



# frame

vp\_frame\_2.py - C:\Users\Wysokie Energie\Desktop\vp\_python\_classes\lec\_a\_11\vp\_frame\_2.py

File Edit Format Run Options Windows Help

```
from visual import *

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

f = frame()
# automatycznie pos=(0,0,0), axis=(1,0,0)

kula1 = sphere(frame=f, pos=(-1,0,0), radius=0.4)
kula2 = sphere(frame=f, pos=(1,0,0), radius=0.4)
# pozycja kul wzgledem pozycji frame !

sleep(2)

f.pos = (1,0,0) # zmiana polozenia frame

print kula1.pos # nie zmienia sie !
print kula2.pos # nie zmienia sie !
```

Ln: 17 Col: 0

```
>>>
<-1, 0, 0>
<1, 0, 0>
```

Lp: 5 Col: 0

# frame, visible

76 vp\_frame\_3.py - C:\Users\Wysokie Energie\Desktop\vppython\_classes\lec\_a\_11\vp\_frame\_3.py

File Edit Format Run Options Windows Help

```
from visual import *

scena = display(width=550, height=580, range=4)

f = frame(pos=(2,0,0))
# automatycznie axis=(1,0,0)

L = []
for i in [-1,0,1]:
    for j in [-1,0,1]:
        L.append(sphere(frame=f, pos=(i,j,0), radius=0.4))

sleep(2)

f.visible = False

sleep(1)

f.pos = (-2,0,0)

f.visible = True
```

# frame, objects

vp\_frame\_4.py - C:\Users\Wysokie Energie\Desktop\vp\_python\_classes\lec\_a\_11\vp\_frame\_4.py

File Edit Format Run Options Windows Help

```
from visual import *

scena = display(width=550, height=580, range=4)

f = frame(pos=(2,0,0))
# automatycznie axis=(1,0,0)

L = []
for i in [-1,0,1]:
    for j in [-1,0,1]:
        L.append(sphere(frame=f, pos=(i,j,0), radius=0.4))

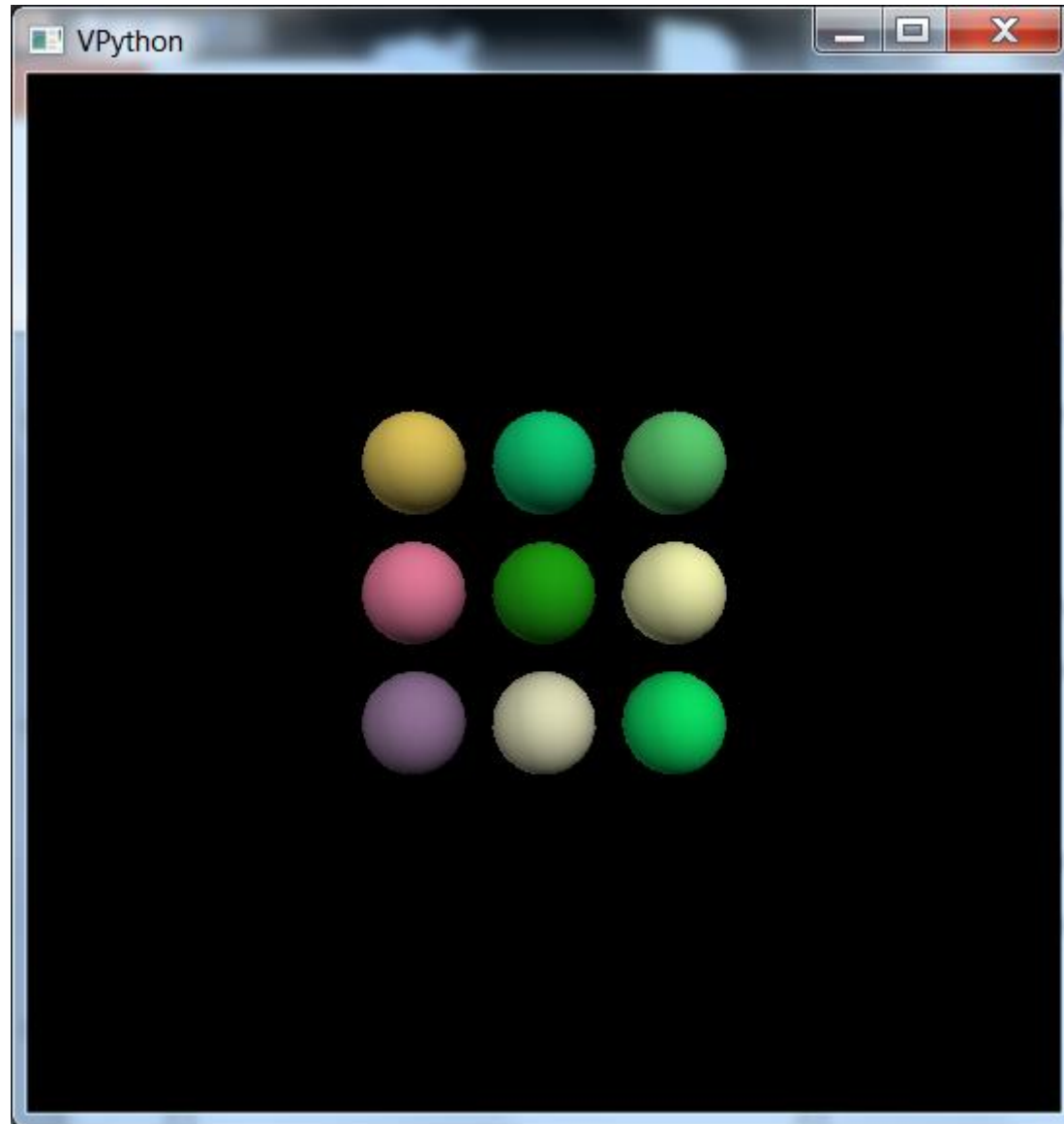
sleep(2)

f.pos = (0,0,0)

for obj in f.objects:
    obj.color = random.uniform(0,1,3) # 3 liczby od 0 do 1
```

Ln: 7 Col: 0

# efekt końcowy



# frame, axis

76 vp\_frame\_5.py - C:\Users\Wysokie Energie\Desktop\vppython\_classes\lec\_a\_11\vp\_frame\_5.py

File Edit Format Run Options Windows Help

```
from visual import *

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

f = frame(pos=(2,0,0))
# automatycznie axis=(1,0,0)

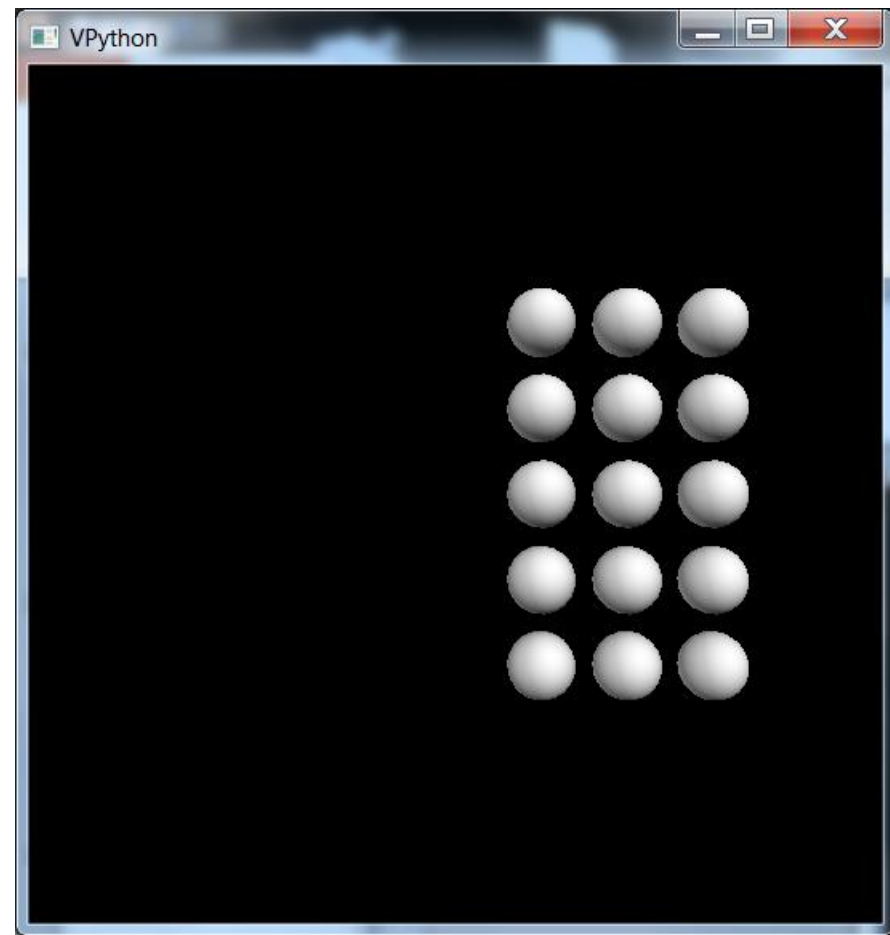
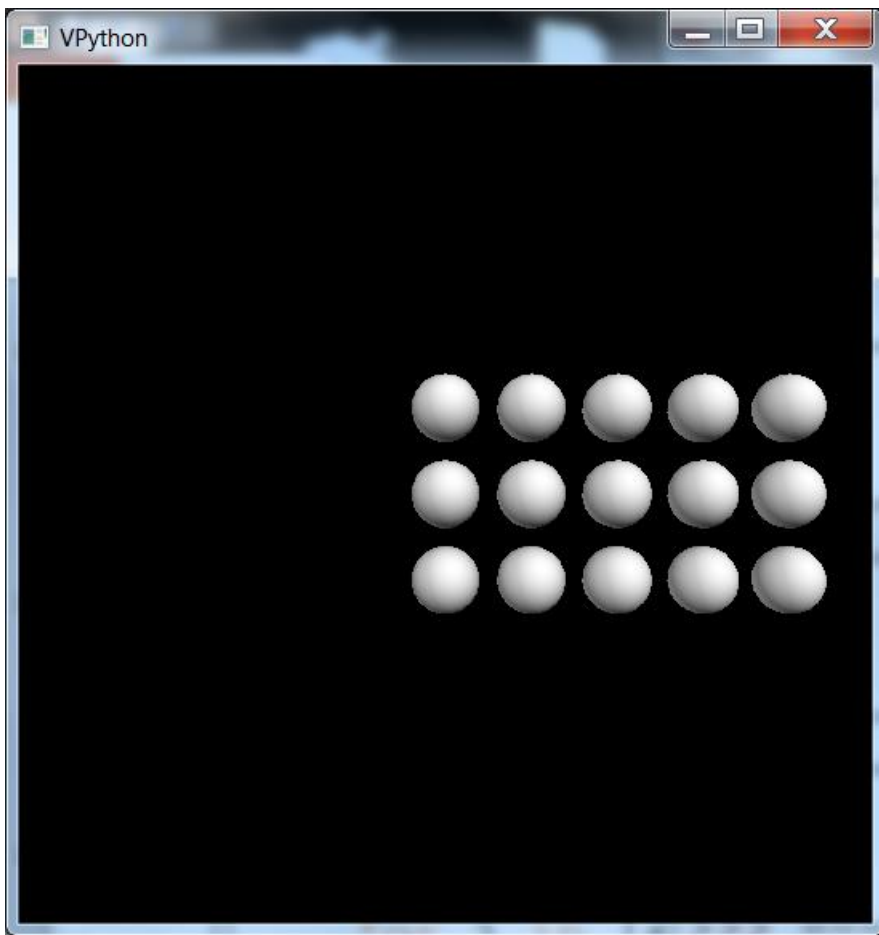
L = []
for i in [-2,-1,0,1,2]:
    for j in [-1,0,1]:
        L.append(sphere(frame=f, pos=(i,j,0), radius=0.4))

sleep(2)

f.axis = (0,1,0)
```

Ln: 1 Col: 0

frame, axis





# frame, axis

76 vp\_frame\_6.py - C:\Users\Wysokie Energie\Desktop\vppython\_classes\lec\_a\_11\vp\_frame\_6.py

```
from visual import *

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

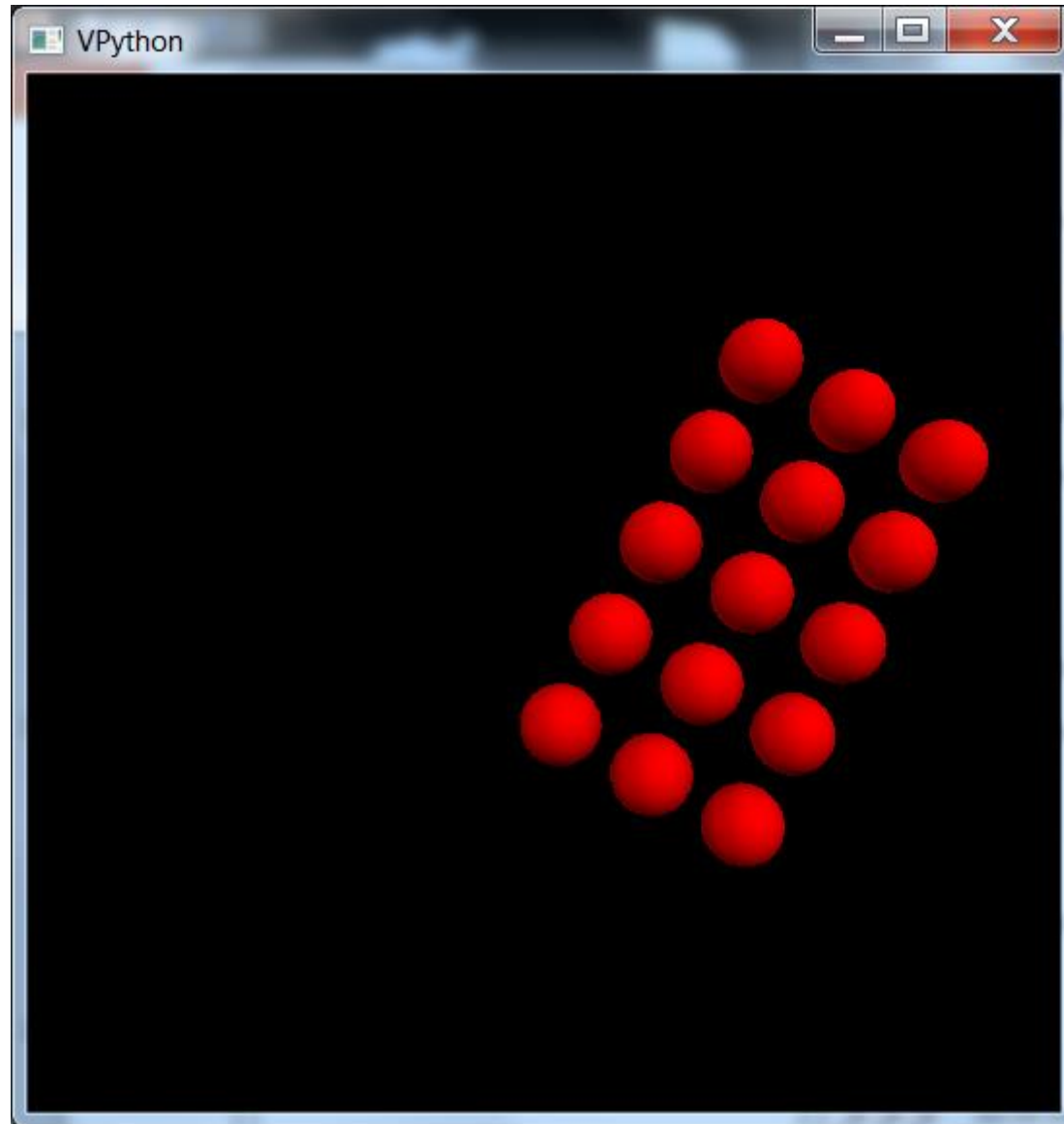
f = frame(pos=(2,0,0))
# automatycznie axis=(1,0,0)

L = []
for i in [-2,-1,0,1,2]:
    for j in [-1,0,1]:
        L.append(sphere(frame=f, pos=(i,j,0), radius=0.4,
                        color=color.red))

sleep(2)

t = 0
while 1:
    rate(1000)
    f.axis = (cos(t), sin(t), 0)
    t += 0.001
```

Ln: 16 Col: 5



# graph

76 vp\_graph\_1.py - C:\Users\Wysokie Energie\Desktop\vpypython\_classes\lec\_a\_11\vp\_graph\_1.py

File Edit Format Run Options Windows Help

```
from visual import *
from visual.graph import *

scena = gdisplay(x=50, y=10, width=700, height=400,
                 title='log(x)', xtitle='x',
                 ytitle='log(x)', foreground=color.yellow,
                 background=color.black,
                 xmin=-1, xmax=11, ymin=-3, ymax=3)

# wszystkie dane z gdisplay(...) można opuścić i napisać
# scena = gdisplay()

f1 = gcurve(color=color.cyan)

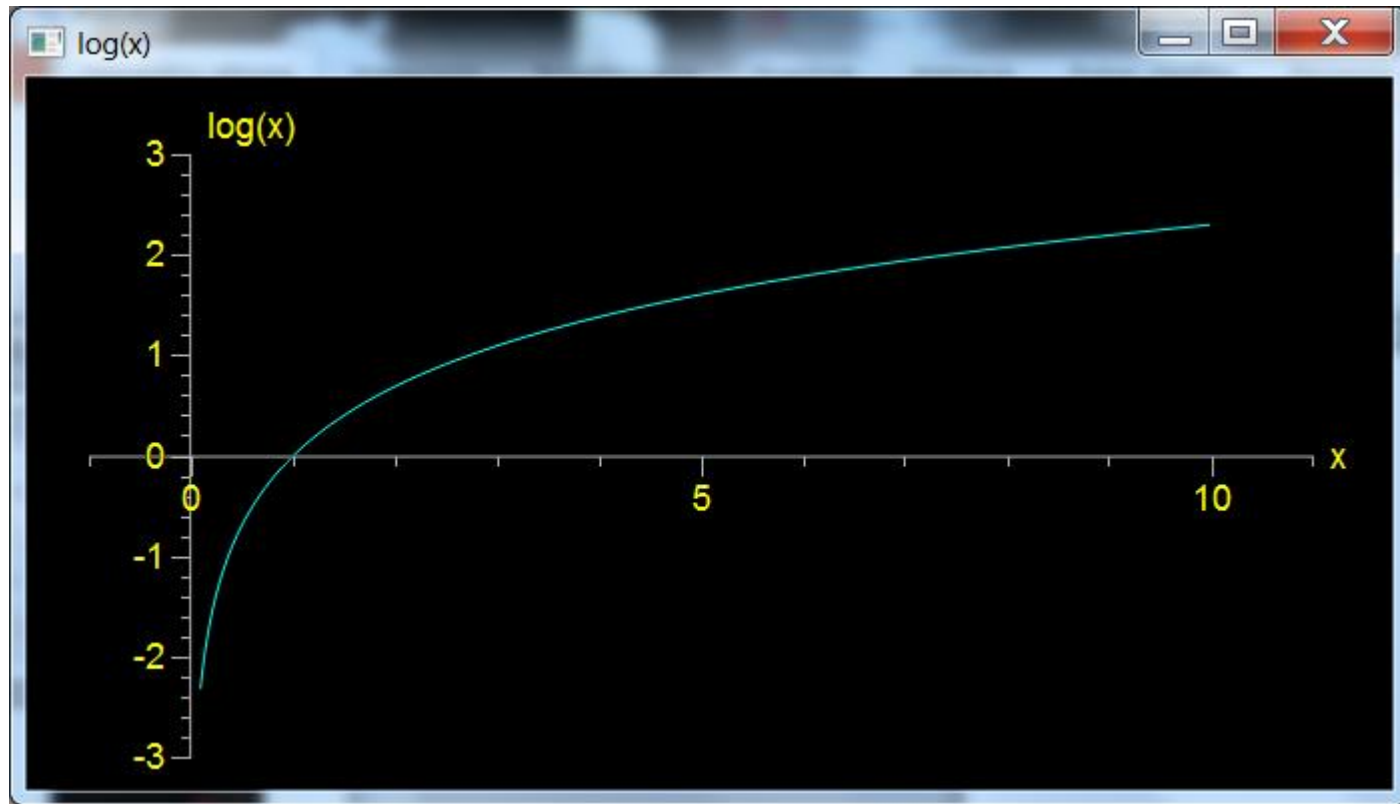
for x in arange(0.1, 10, 0.01):
    f1.plot( pos=(x, log(x)) )
```

Ln: 9 Col: 0

**gdisplay** → **graph** w VPython 7

Może też być `f1.plot( x, log(x) )`

graph



# graph, dodanie punktów

vp\_graph\_2.py - C:\Users\Wysokie Energie\Desktop\vppython\_classes\lec\_a\_11\vp\_graph\_2.py

File Edit Format Run Options Windows Help

```
from visual import *
from visual.graph import *

scena = gdisplay()

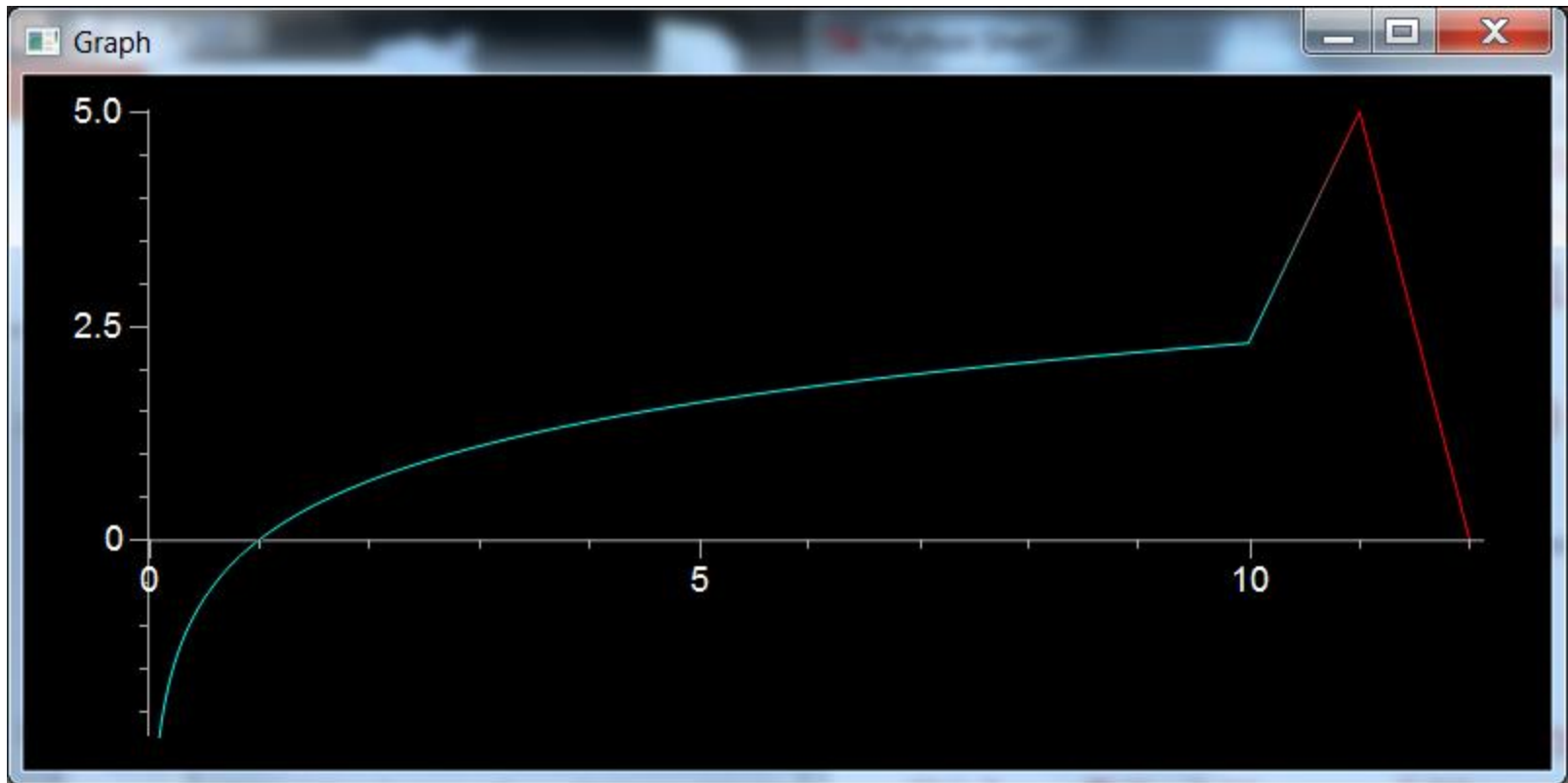
f1 = gcurve(color=color.cyan)

for x in arange(0.1,10,0.01):
    f1.plot( pos=(x,log(x)) )

f1.plot( pos=[(11,5),(12,0)], color=color.red )
```

Ln: 1 Col: 0

graph, dodanie punktów



# graph, krzywa, punkty, słupki

76 vp\_graph\_3.py - C:\Users\Wysokie Energie\Desktop\vppython\_classes\lec\_a\_11\vp\_graph\_3.py

File Edit Format Run Options Windows Help

```
from visual import *
from visual.graph import *

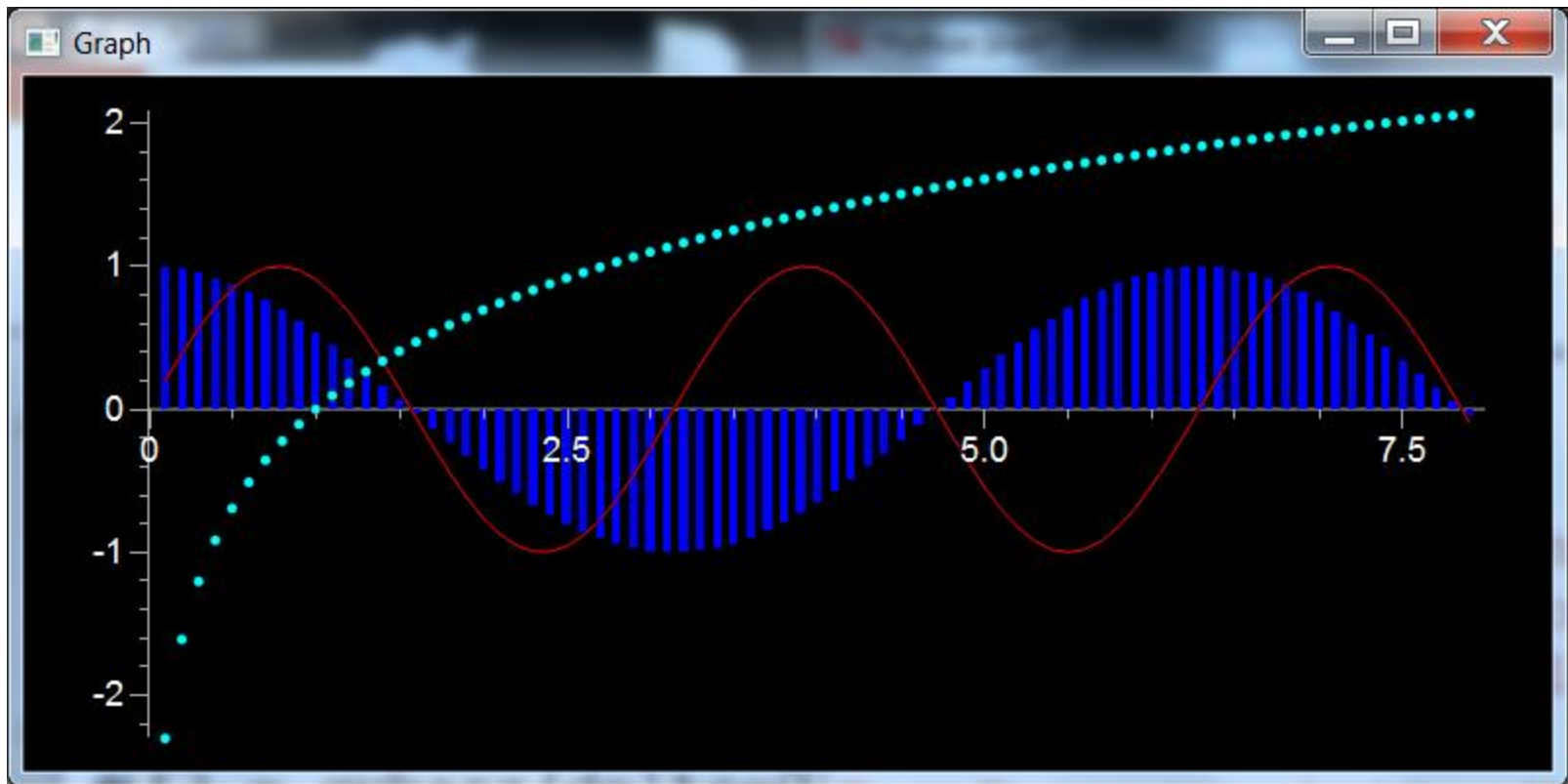
scena = gdisplay()

f1 = gcurve(color=color.red)    # linia
f2 = gdots(color=color.cyan)    # punkty
f3 = gvbars(delta=0.05, color=color.blue)    # słupki
# tez jest ghbars, poziome słupki

for x in arange(0.1, 8, 0.1):
    f1.plot( pos=(x, sin(2*x)) )
    f2.plot( pos=(x, log(x)) )
    f3.plot( pos=(x, cos(x)) )
```

Ln: 1 Col: 0

graph, krzywa, punkty, słupki





# graph, dwa okna, skala logarytmiczna

vp\_graph\_5.py - C:\Users\Wysokie Energie\Desktop\vppython\_classes\lec\_a\_11\vp\_graph\_5.py

File Edit Format Run Options Windows Help

```
from visual import *
from visual.graph import *

scena1 = gdisplay(x=10,y=10, logx=True)
scena2 = gdisplay(x=10,y=450, logy=True)

f1 = gcurve(gdisplay=scena1, color=color.red)
f2 = gdots(gdisplay=scena2, color=color.cyan)

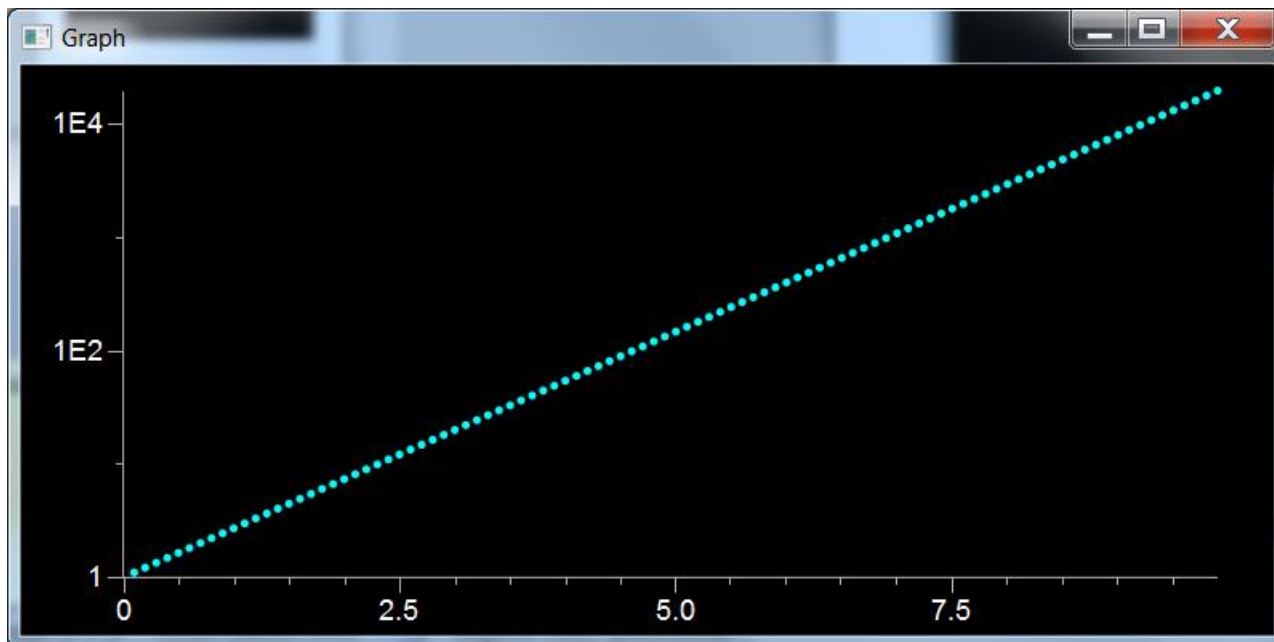
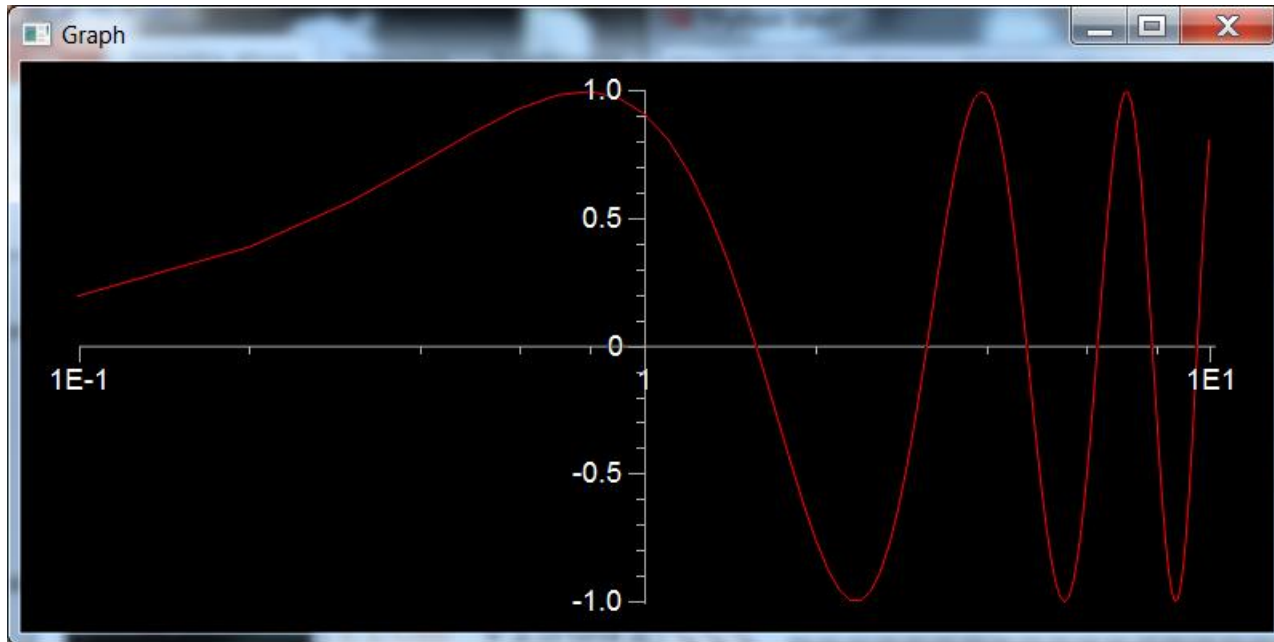
for x in arange(0.1,10,0.1):
    f1.plot( pos=(x,sin(2*x)) )
    f2.plot( pos=(x,exp(x)) )
```

Ln: 5 Col: 6

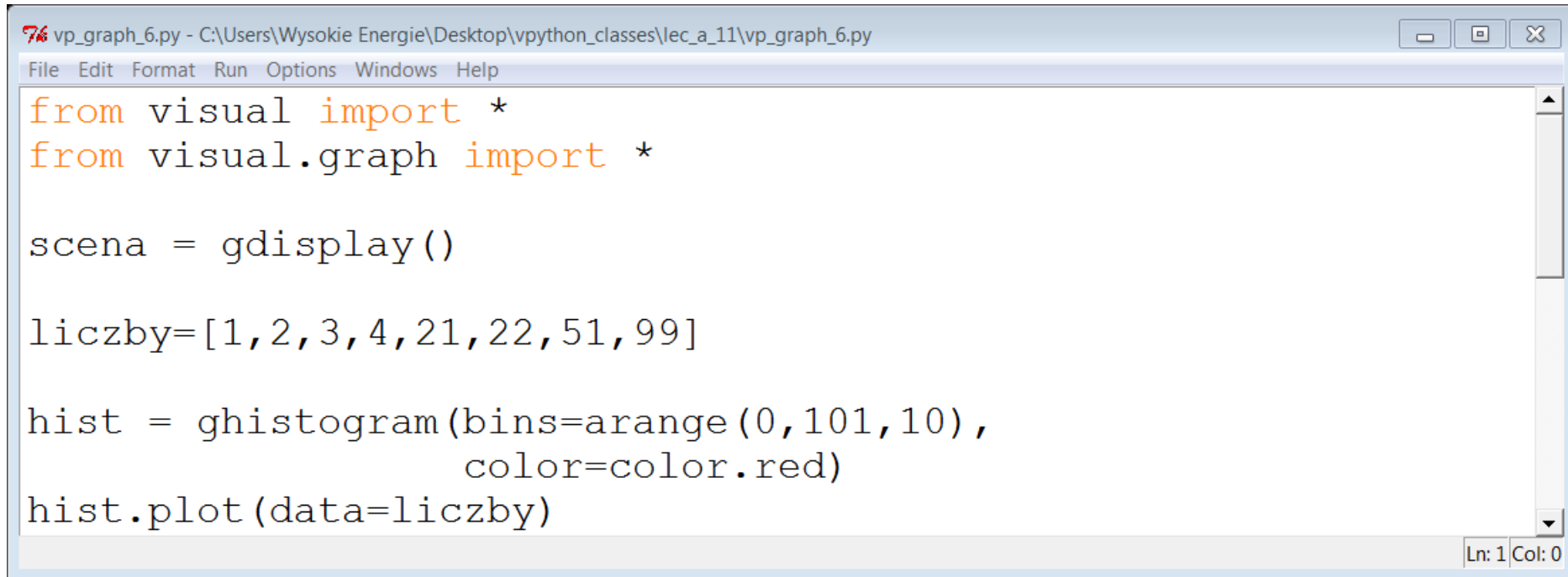
f1.plot( x, sin(2\*x) )

f2.plot( x, exp(x) )

graph, dwa okna



# graph, histogram



The screenshot shows a VPython 7 IDE window titled "vp\_graph\_6.py". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code in the editor is as follows:

```
from visual import *
from visual.graph import *

scena = gdisplay()

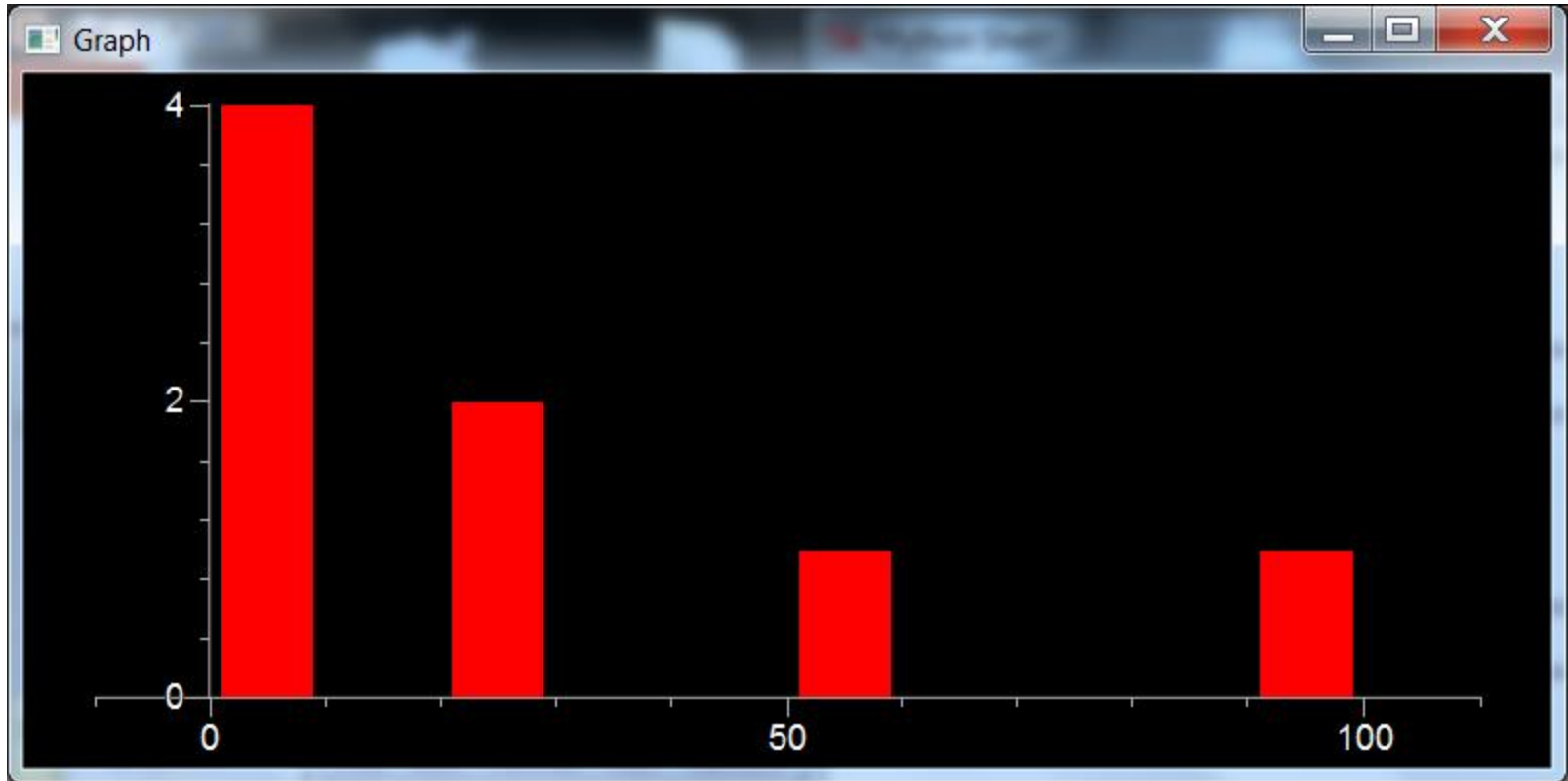
liczby=[1,2,3,4,21,22,51,99]

hist = ghistogram(bins=arange(0,101,10),
                  color=color.red)
hist.plot(data=liczby)
```

The status bar at the bottom right indicates "Ln: 1 Col: 0".

brak w VPython 7

graph, histogram



# graph, dynamiczny histogram

vp\_graph\_7.py - C:\Users\Wysokie Energie\Desktop\vppython\_classes\lec\_a\_11\vp\_graph\_7.py

File Edit Format Run Options Windows Help

```
from visual import *
from visual.graph import *

scena = gdisplay()

hist = ghistogram(bins=arange(0,101,10),
                  color=color.red)

while 1:
    rate(10)
    liczby = random.uniform(0,100,100)
    hist.plot(data=liczby)

# plot(..., accumulate=True) # dodaje slupki
# plot(..., accumulate=True, average=True)
# dodaje slupki i usrednia
```

Ln: 1 Col: 0