

VPython

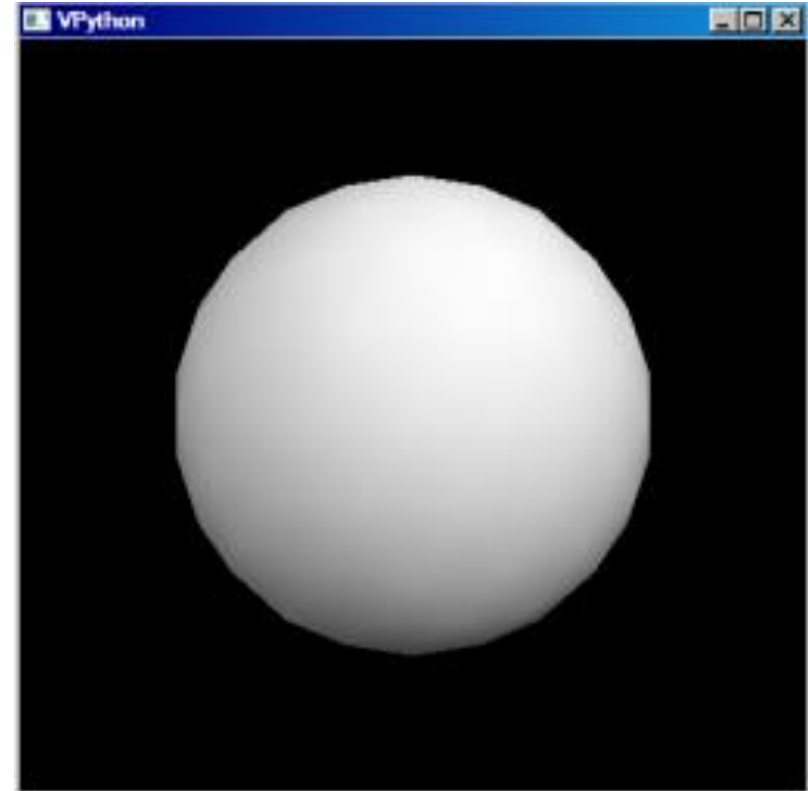
Dr Wojciech Palubicki

VPython

- Moduł pythonowski stworzony do interaktywnego modelowania
- Bardzo łatwy do nauczenia

Pierwszy krok

- `from visual import *`
- `sphere()`



Zmiana atrybutów kuli

- Kolor
 - `sphere(color=color.red)`

Zmiana atrybutów kuli

- Kolor
 - `sphere(color=color.red)`
- Radius
 - `sphere(radius=0.5,color=color.red)`

Zmiana atrybutów kuli

- Kolor
 - `sphere(color=color.red)`
- Radius
 - `sphere(radius=0.5,color=color.red)`
- Nazwa
 - `kula = sphere(radius=0.5,color=color.red)`

Zmiana atrybutów kuli

- Kolor
 - `sphere(color=color.red)`
- Radius
 - `sphere(radius=0.5,color=color.red)`
- Nazwa
 - `kula = sphere(radius=0.5,color=color.red)`
- Pozycja
 - `kula = sphere(pos=(0,2,0),radius=0.5,color=color.red)`

Zmiana atrybutów kuli

- Kolor
 - `sphere(color=color.red)`
- Radius
 - `sphere(radius=0.5,color=color.red)`
- Nazwa
 - `kula = sphere(radius=0.5,color=color.red)`
- Pozycja
 - `kula = sphere(pos=(0,2,0),radius=0.5,color=color.red)`
- Zmiana pozycji
 - `kula.pos = (1,2,3)`

Nawigacja

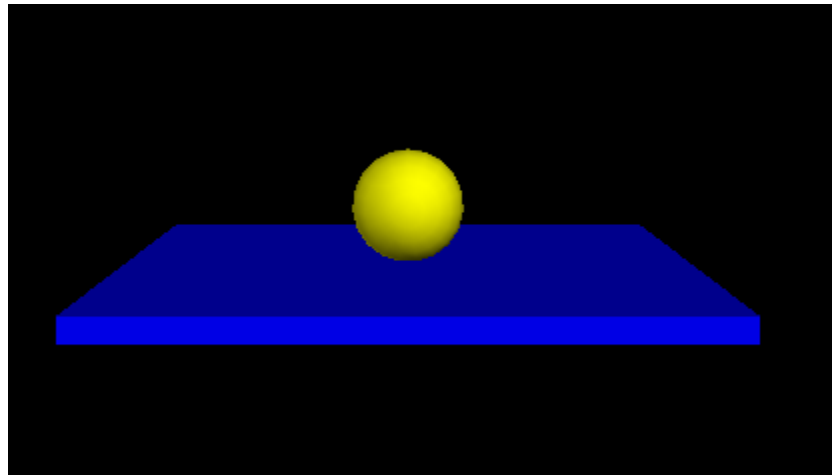
- Zoom
 - Trzymaj środkowy przycisk myszki i poruszaj myszką
 - Lub, trzymaj lewy i prawy przycisk naraz
- Obrót
 - Trzymaj prawy przycisk i poruszaj myszka

Prosta scena

```
from visual import *
```

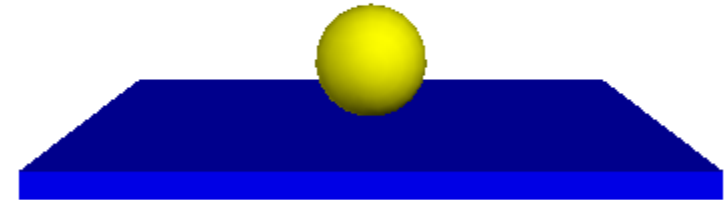
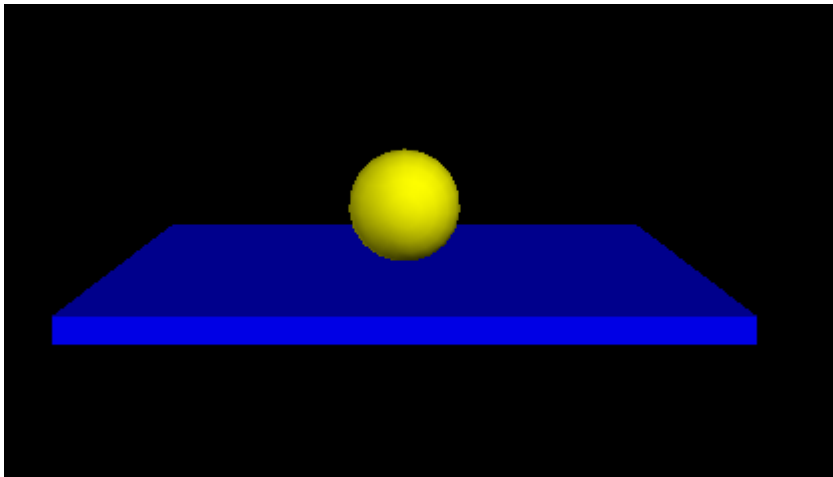
```
kula = sphere( pos=(0,-1.9,0), radius=0.5, color=color.yellow )
```

```
pudelko = box( pos = (0,-2,0), size = (5,0.2,5), color=color.blue )
```



Display()

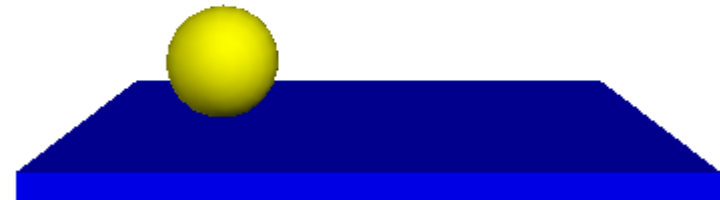
- `scena = display(title='Kula', width=800, height=600, range=(7, 7, 7))`



`scena.background = color.white`

Animacja

- Za pomocy pętli while, np.
 l=1
 while l == 1:
 rate(n) #odświeżanie obrazu
 #zmień pozycje kuli



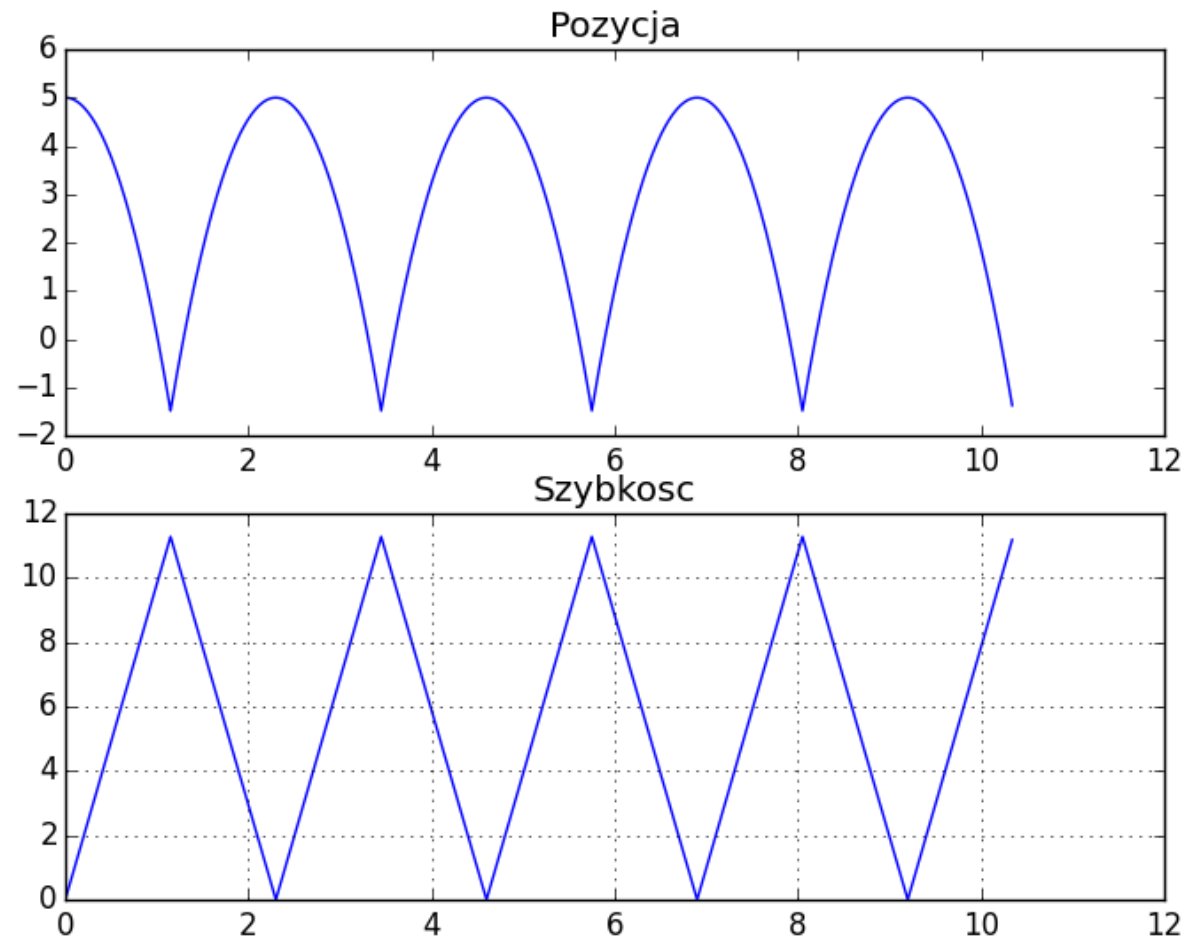
Symulacja spadającej kuli

- $v = \text{vector}(0,0,0)$
- p = pozycja kuli
- $a = \text{vector}(0,9.8,0)$ #przyspieszenie
- $\Delta t = 0.01$ #krok czasowy

- $v = a\Delta t$
- $\Delta p = v\Delta t - \frac{1}{2}a^2\Delta t$



Wykresy pozycji i prędkości



SciPy [Scientific Algorithms]

linalg	stats	interpolate
cluster	special	maxentropy
io	fftpack	odr
ndimage	sparse	integrate
signal	optimize	weave

NumPy [Data Structure Core]

fft	random	linalg
NDArray multi-dimensional array object		UFunc fast array math operations