

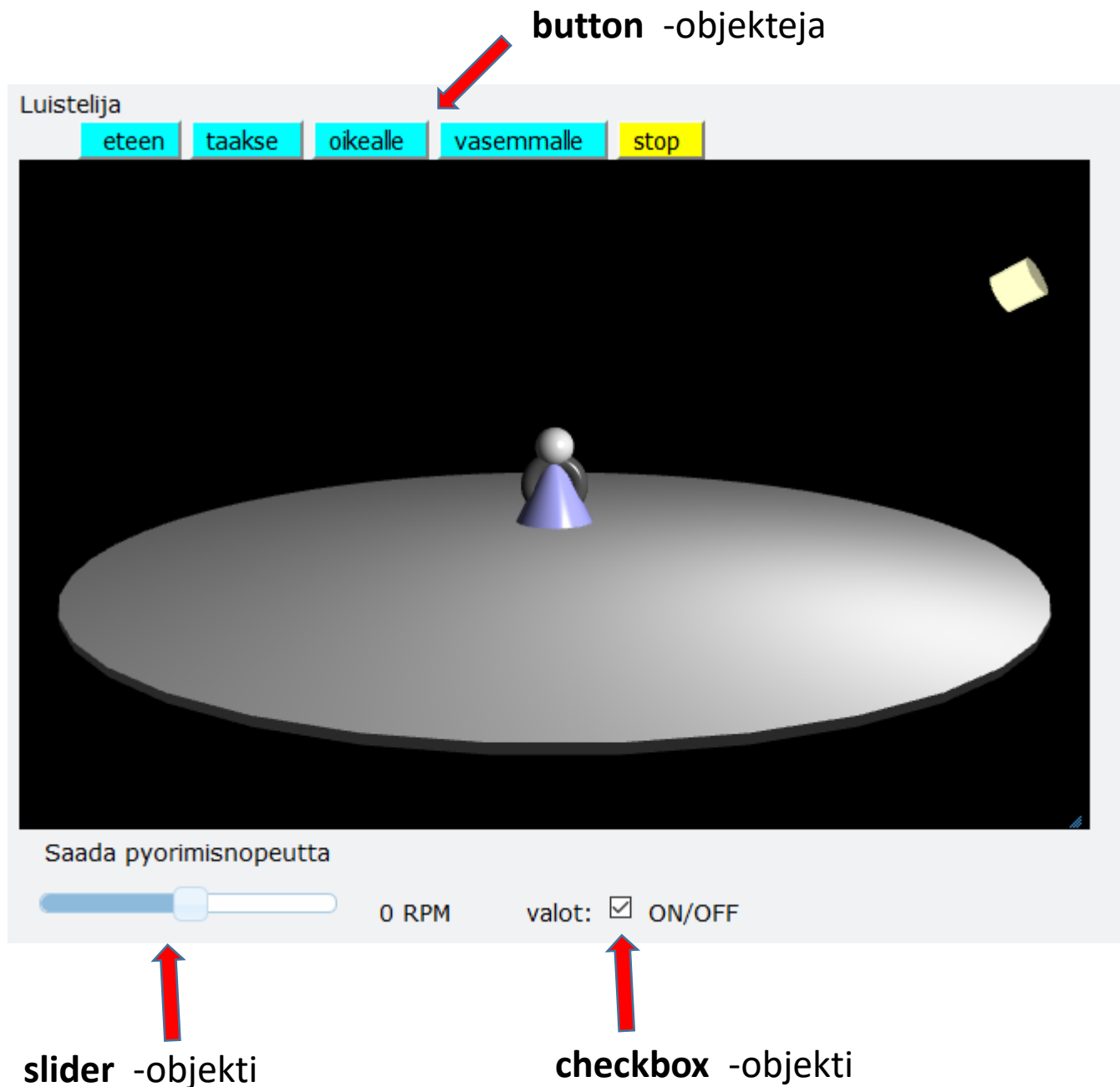
Interaktiivisuutta käyttäjäliityntään

VPython Controls:

- **Button**
- **Slider**
- **Checkbox**

Esimerkkiohjelmanä on **"TAITOLUISTELIJA"** , jossa luistelijaa ohjataan painikkeilla.

Tehtävä 5 liittyy TAITOLUISTELIJA – esimerkkiin, johon pitää koodata neljän painikkeen toiminnallisuus.



slider -objekti

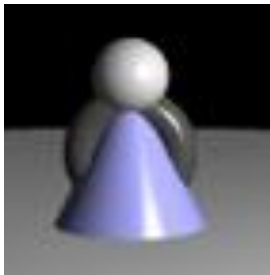
checkbox -objekti

← Tämä alue on **scene.title**

Siihen voi kirjoittaa otsikkotekstin seuraavasti:
scene.title = 'Luistelijä'
Jos haluat heti perään rivinvaihdon, kirjoita
scene.title = 'Luistelijä\n'

← Tämä alue on **scene.caption**

Siihen voi kirjoittaa tekstin seuraavasti:
scene.caption = 'Saada pyorimisnopeutta\n'



YHDISTELMÄOBJEKTI COMPOUND

VPython – kirjasto ei tue pelihahmojen luomista (pelejä varten ovat pygame ja panda)

VPythonin grafiikkaobjekteista voi rakentaa yhdistelmäobjektin (**compound** objekti), jota voi liikutella yhtenä kokonaisuutena.

Kuvan luistelijaa on muodostettu kartiosta (cone), renkaasta (ring) ja pallostasta (sphere), jotka muodostavat vartalon, kädet ja pään.

```
kadet=ring(pos=vec(0,4,0),radius=2.5,thickness=1,axis=vec(0,0,1),color=vec(0.3,0.3,0.3))
paa=sphere(pos=vec(0,8,0),radius=2,color=vec(0.8,0.8,0.8))
vartalo=cone(pos=vec(0,0.25,0),radius=4,axis=vec(0,8,0),color=vec(0.6,0.6,0.9))
```

```
luistelijaa=compound( [vartalo,kadet,paa] )
```

↑
osien yhdistäminen luistelijaa-objektiksi

(Yhdistelmäobjektin paikkaan viittaa vektori **luistelijaa.pos** , jonka arvoksi tulee jonkinlainen keskiarvo osien paikoista)

PAINIKKEIDEN LUOMINEN



Painike on **button** – objekti, jonka määreitä ovat mm. seuraavat:

pos = scene.title_anchor	(vaihtoehdot: scene.title_anchor tai scene.caption_anchor (oletus))
text = 'eteen'	(painikkeen teksti)
bind = eteen	(annetaan sen aliohjelman nimi, joka suoritetaan painiketta klikattaessa)
background = color.cyan	(painikkeen taustaväri)

Painikkeen b1 kutsuma aliohjelma ja painikkeen **muodostin**

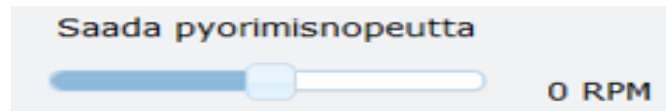
```
def eteen(b1):  
    global askel  
    askel=vec(0,0,0.1)  
  
b1=button(pos=scene.title_anchor,background=color.cyan,  
text=' eteen', bind=eteen)
```

Vasemmalla olevalla aliohjelmalla **eteen** on argumenttina painikkeen nimi (tässä b1).

Aliohjelma muuttaa pääohjelmassa esiintyvää vektorimuuttujaa **askel**

```
while True:      # pääohjelman silmukka  
    rate(100)  
    luistelij.pos+=askel  
    luistelij.rotate(angle=kulmanopeus, axis=vec(0,1,0))
```

SLIDERIN LUOMINEN



liukukytkin on **slider** – objekti, jonka määreitä ovat mm. seuraavat: (arvot luisteliija-esimerkistä)

min = -12	(palautusarvo s.value kun liukukytkin on vasemmassa päässä)
max = 12	(palautusarvo s.value kun liukukytkin on oikeassa päässä)
value = 0	(palautusarvo alkuhetkellä)
bind = saadanopeus	(aliohjelman nimi, jossa sliderin arvoa käytetään)

Sliderin s kutsuma aliohjelma ja sliderin muodostin

```
def saadanopeus(s):  
    global kulmanopeus  
    kulmanopeus=radians(s.value)  
    wt.text = '{:1.0f}'.format(s.value/6.28*60)+' RPM  '
```

```
s=slider(min=-  
12,max=12,value=0,length=200,bind=saadanopeus,  
left=20,top=15)  
wt = wtext(text='{:1.0f}'.format(s.value/6.28*60)+' RPM  ')
```

Vasemmalla olevalla aliohjelmalla **saadanopeus** on argumenttina sliderin nimi (tässä s).

Aliohjelma muuttaa pääohjelmassa esiintyvää skalarimuuttujaa **kulmanopeus**

```
while True:      # pääohjelman silmukka  
    rate(100)  
    luisteliija.pos+=askel  
    luisteliija.rotate(angle=kulmanopeus, axis=vec(0,1,0))
```

CHECKBOX:N LUOMINEN

valot: ☒ ON/OFF

Rasti ruutuun eli **checkbox** – objekti, jonka määreitä ovat mm. seuraavat: (arvot luistelij-esimerkistä)

bind= valot

text = 'ON/OFF'

(aliohjelma, jossa checkbox:n arvoa käytetään)

(checkboxin teksti)

checkboxin kutsuma aliohjelma **valot** ja checkboxin koodi

```
def valot(valokytin):  
    if valokytin.checked:  
        lamppu.visible=True  
    else:  
        lamppu.visible=False  
  
valokytin=checkbox(bind=valot, text='ON/OFF')
```

Vasemmalla olevalla aliohjelmalla **valot** on argumenttina checkboxin nimi (tässä **valokytin**).

Aliohjelma sytyttää ja sammuttaa kohdevalaisimen, jonka nimi on **lamppu**

#lamppu-objekti on määritelty ohjelmassa seur.:

```
lamppu = local_light(pos=vec(22,18,30),  
color=color.white, visible=False)
```

NÄYTTÄMÖN VALAISTUS

Oletusvalaistus voidaan poistaa näyttämöltä komennolla
`scene.lights=[]`

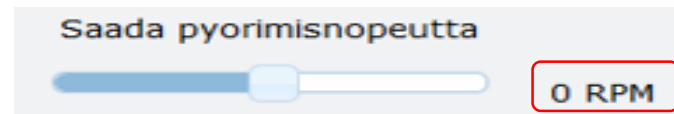
Tilalle voidaan tuoda **pistemäisiä valonlähteitä**, joille määritetään paikka, ja väri
`valo1 = local_light(pos=vec(22,18,30), color=color.white)`

Valo voidaan **sytyttää ja sammuttaa** määreellä `visible`
`valo1.visible = False`
`valo1.visible=True`

Itse valonlähde ei ole näkyvässä, vain sen aikaansaama valaistus. Jos halutaan näkyviin kirkas pistemäinen valo, pitää luoda esim. `sphere` -objekti.
Sen kirkkautta voidaan tehostaa **`emissive = True`** määreellä

`lamppu1 = sphere(pos=valo1.pos, color=color.white, emissive = True)`

DYNAAMISET TEKSTIT



Dynaaminen tekstikenttä on **wtext** – objekti (esimerkissä kierroslukunäyttö)

Muodostin näyttää seuraavalta:

```
wt = wtext(text='{:2.0f}'.format(s.value/6.28*60)+' RPM  ')
```

Sliderin arvo `s.value` liittaa rotaatiokulmaan, voi tulkita myös kulmanopeudeksi yksikössä rad/sek

Arvo jaetaan 2π :llä jolloin se muuttuu pyörimistaajuudeksi yksikkönä RPS (kierr./sek)

Arvo kerrotaan vielä 60:llä jolloin yksikkö on RMP (kierr./min)

Alla tekstiobjektin **wt** tekstin päivitys

```
wt.text = '{:2.0f}'.format(s.value/6.28*60)+' RPM  '
```

`{:1,0f}` on formaatti float (kokonaisosa 2, desimaaleja 0)

Huom!

Slider säättää kulmataajuutta ω ,
jonka yksikkö on rad/s

Käyttäjälle tutumpi suure on kierrostaajuus
yksikössä RPM.

Muunnos: $f(\text{RPM}) = \omega / 2\pi * 60$