

# A. Tasainen pyörimisliike

# Pyörimisliikkeen suureet

**Pyörähdysaika  $T$**

yksikkö : 1 s

**Pyörimistaajuus  $f = \frac{1}{T}$**

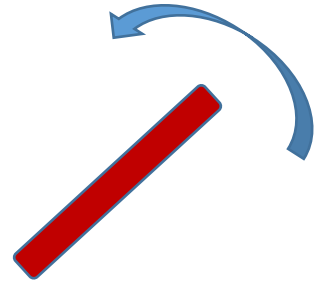
yksikkö : 1 Hz = 1 RPS

tai 1 RPM ( 1 RPS = 60 RPM)

**Kulmataajuus  $\omega = \frac{\Delta\varphi}{\Delta t}$**

kulman muutos / aikaväli

**Kulmataajuus  $\omega = 2\pi f = 2\pi/T$**  yksikkö 1 rad/s

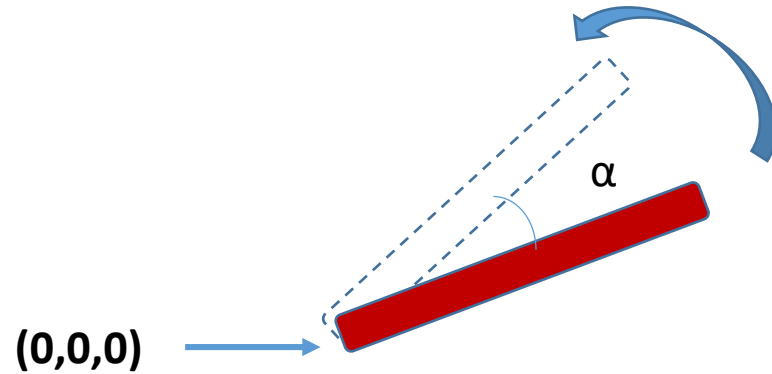


**Käytämme koodissa kulmanopeudesta symbolia  $w$**

Esim. Jos pyörähdysaika  $T = 20\text{s}$ , niin kulmataajuus  $\omega = 0.314 \text{ rad/s}$   
Vpythonissa voidaan määritellä

$w = 2*\text{pi}/20$

# Grafiikkaobjektin rotaatio VPythonissa



Esim. keppi, jonka toinen pää on origossa.

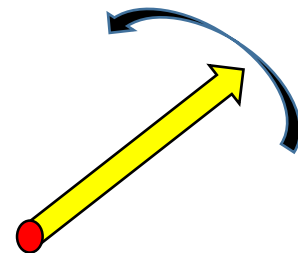
Rotaatio origon ympäri  $20^\circ$  siten että pyörimisakselina on z- akseli

```
keppi.rotate(angle=radians(20), axis=vec(0,0,1))
```

Rotaatio kepin keskipisteen (tässä (1.5,0.5,0) ympäri  $20^\circ$  siten että pyörimisakselina on z- akseli

```
keppi.rotate(angle=radians(20), axis=vec(0,0,1), origin=vec(1.5,0.5,0))
```

# Tasainen pyöriminen kulmanopeudella $w$



Esimerkki: pyörivä nuoli

```
scene.range=8
nuoli=arrow(pos=vec(0,0,0), axis=vec(4, 3, 0),shaftwidth=0.5, color=color.yellow)
T= 1          # pyorahdysaika sekunteina
w=2*pi/T      #kulmanopeus rad/s
dt=0.01       #aikavali
scene.pause()
while True:
    rate(100)
    nuoli.rotate(angle=w*dt, axis = vec(0,0,1))
```

*Kokeile muuttaa viimeinen rivi muotoon:*

```
keppi.rotate(angle=w*dt, axis = vec(0,0,1),axis=vec(-3,4,0))
```

*Kokeile vaihtaa kulmanopeuden  $w$  etumerkki*

Pyörimisliike toteutetaan antamalla kiertoaika  $T$ , josta lasketaan kulmanopeus  $w$ .

⇒ Rotaatiokulma on  
 $\Delta\varphi = w*dt$

NUOLEN KIERTOLIIKE VOIDAAN TOTEUTTAA MYÖS ILMAN ROTATE FUNKTION KÄYTTÖÄ SITEN, ETTÄ SILMUKASSA PÄIVITETÄÄN NUOLEN SUUNTAKULMAA JA KEPIN AKSELIA. (animaatio hieman hidas, silmä erottaa kuvien vaihtumisen)

GlowScript 2.9 VPython

scene.range=10

scene.title='Pyoriva nuoli, klik = aloitus'

**kulma=0**

keppi=arrow(pos=vec(0,0,0), **axis=vec(5\*cos(kulma), 5\*sin(kulma), 0)**, shaftwidth=0.5, color=0.4\*color.yellow)

T= 1 # pyorahdysaika sek

w=2\*pi/T #kulmanopeus rad/s

dt=0.01

scene.pause()

while True:

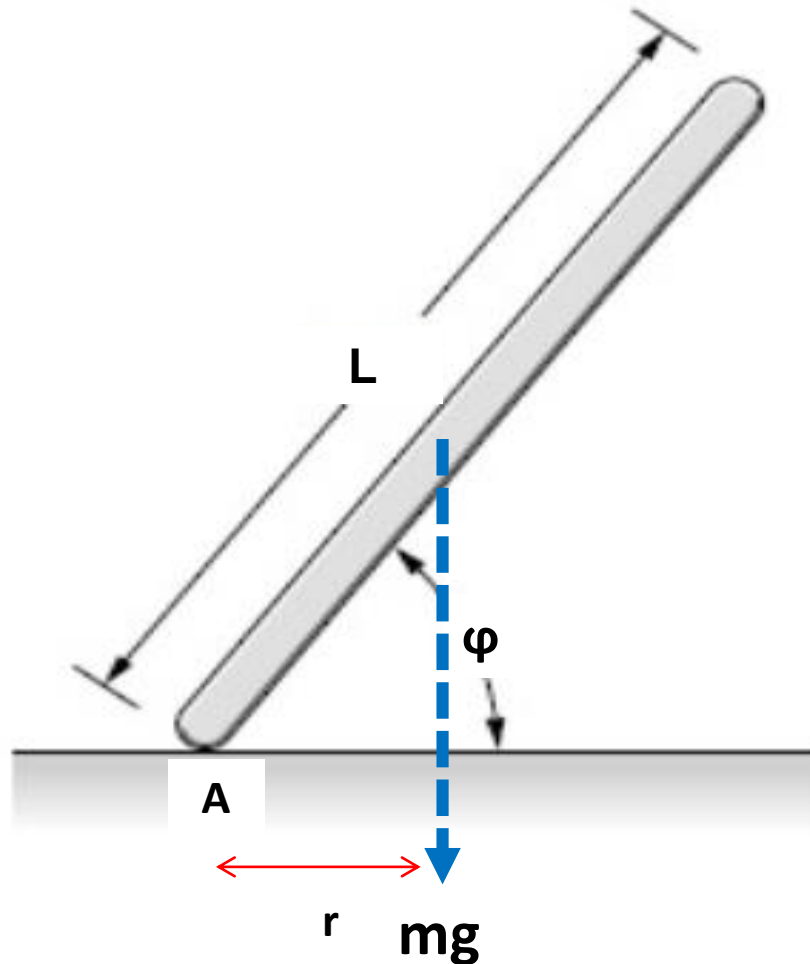
rate(100)

**kulma+=w\*dt**

**keppi.axis=vec(5\*cos(kulma),5\*sin(kulma),0)**

## B. Pylvään kaatumisen fysiikkaa

Pyörimisliikkeen kuvaamisessa etenevän liikkeen suuret korvataan pyörimisliikkeen suureilla. Muuten yhtälöt näyttävät hyvin samanlaisilta.



Etenevän liikkeen suuret	Rotaatiosuuret
Paikkavektori $r=(x,y)$	Kulma $\varphi$ (radiaaneina)
Nopeus $v= \Delta r/\Delta t$	Kulmanopeus $\omega = \Delta\varphi/\Delta t$
Kiihtyvyys $a = \Delta v/\Delta t$	Kulmakiihtyvyys $\alpha = \Delta\omega/\Delta t$
<b>Paikan ja nopeuden päivitys</b>	<b>Kulmasuureiden päivitys</b>
$r = r_0+ v_0 \, dt + \frac{1}{2} a \, dt^2$	$\varphi= \varphi_0+ \omega_0 \, dt + \frac{1}{2} \alpha \, dt^2$
$v = v_0+ a \, dt$	$\omega = \omega_0+ \alpha \, dt$
<b>Dynamiikan peruslaki</b>	<b>Laki rotaatioliikkeessä</b>
Voima $F$	Vääntömomentti $M = F \, r$
massa $m$	Hitausmomentti $J = \int r^2 \, dm$ (taulukoitua eri kappaleille)
$a = F/m$	$\alpha = M/J$

# Newtonin 1. ja 2. laki rotaatiossa

1. Kappale on levossa tai pyörii tasaisella kulmanopeudella, jos siihen vaikuttava vääntömomentti = 0

2. Jos kappaleeseen vaikuttaa vääntömomentti  $M$ , se joutuu kiihtyvään pyörimisliikkeeseen. Kulmakiihtyvyyden kaava on

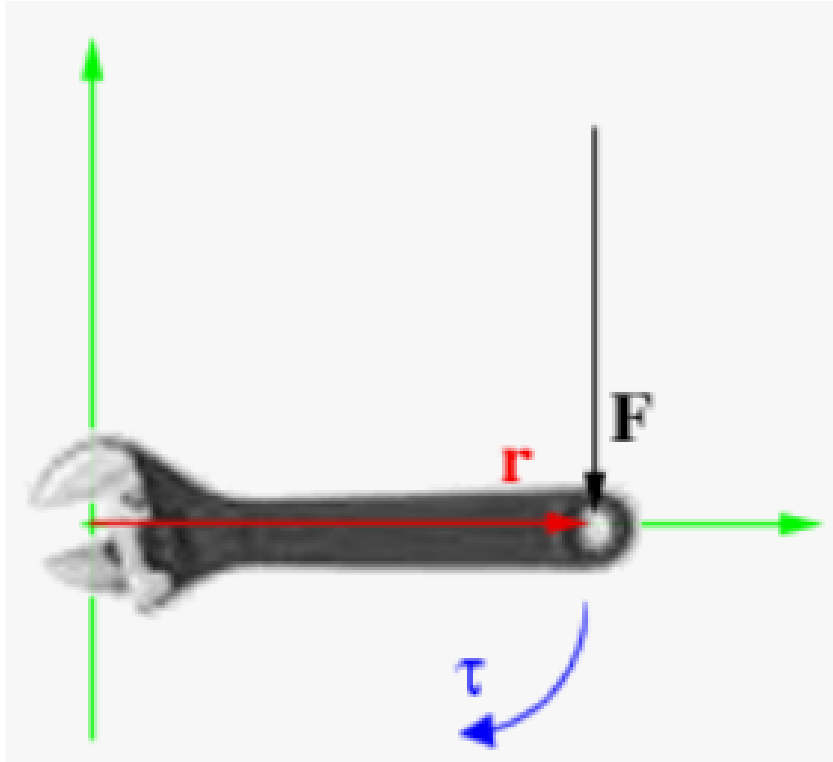
$$\alpha = \frac{\Delta\omega}{\Delta t} = \frac{M}{J}$$

**M** = kappaleeseen vaikuttava vääntömomentti

**J** = kappaleen hitausmomentti (riippuu kappaleen massasta ja sen etäisyydestä pyörimisakselista)



# Voiman momentti $M = F r$



Pyörimisliikkeessä yhtä tärkeää kuin voiman suuruus  $F$  on voiman varsi  $r$  (ts. voiman vaikutussuoran etäisyys akselipisteestä)

Pyörimisliikkeessä käytetäänkin suuretta, jota sanotaan momentiksi:

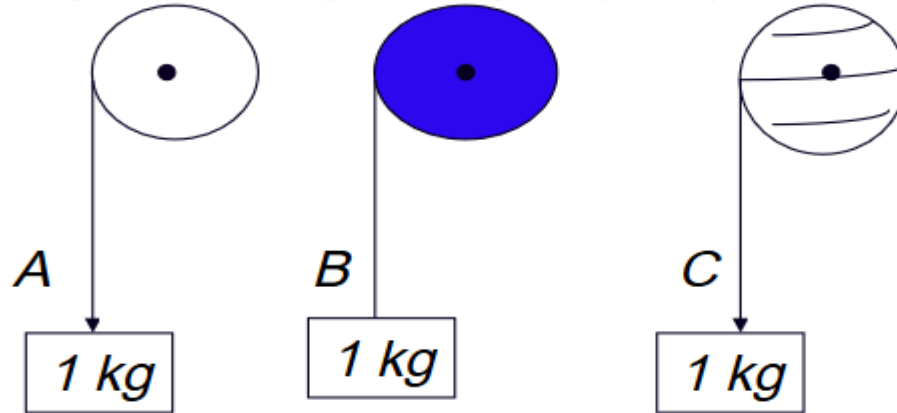
$$M = F r$$

Esim. mutterin aukaisemisen onnistuminen riippuu yhtä paljon käytetystä voimasta  $F$  ja jakoavaimen varren pituudesta  $r$  (ns. vääntövarresta).

# Hitausmomentti J

## Koe 2

*samanmassaiset ja – säteiset A) ontto sylinteri, B) umpinainen sylinteri ja C) umpinainen pallo*

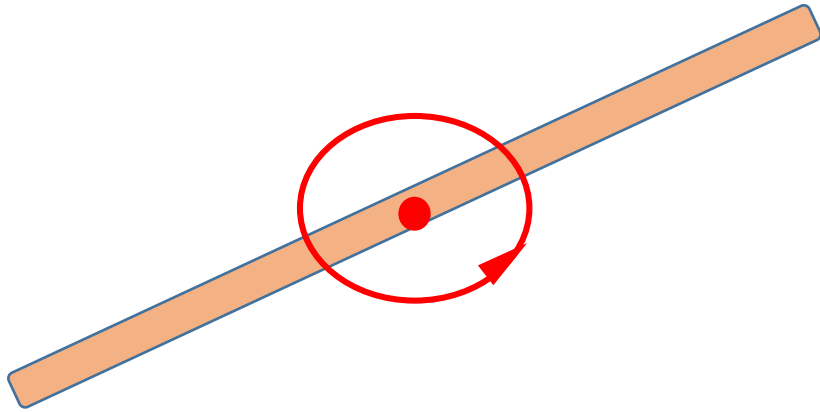


*Kysymys: Missä järjestyksessä punnukset putoavat alas?*

*Vastaus: Ensin umpinainen pallo, sitten umpisylinteri, viimeksi ontto sylinteri.*

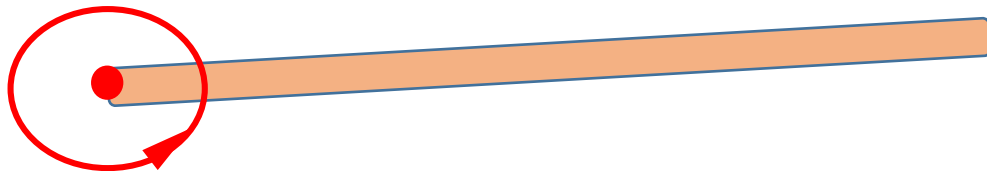
Samanmassaisilla ja -säteisillä kappaleilla kulmakiihtyvyys riippuu siitä, kuinka kaukana massa on keskimäärin pyörähdysakselista. Mitä kauempana massa on akselista, sen suurempi on kappaleen hitausmomentti  $J$  ja sitä pienempi on kulmakiihtyvyys.

# Palkin hitausmomentin kaavat



Kun palkki, jonka massa on  $m$  ja pituus  $L$  pyörii keskipisteensä ympäri, sen hitausmomentti on

$$J = \frac{1}{12} mL^2$$



Kun se pyörii toisen pään ympäri, sen hitausmomentti on

$$J = \frac{1}{3} mL^2$$

## PYLVÄS VOIDAAN LUODA CYLINDER - OBJEKTINA

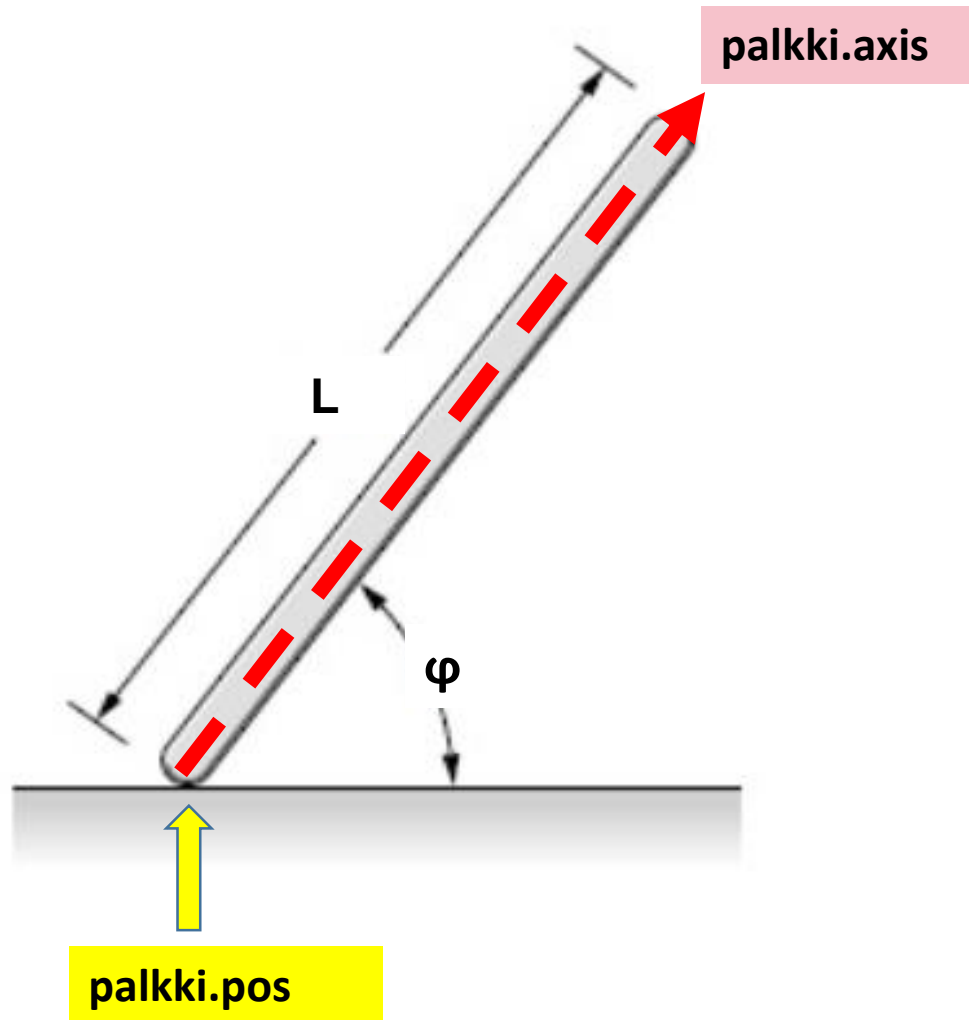
Tällöin

**palkki.pos** antaa palkin pään sijainnin

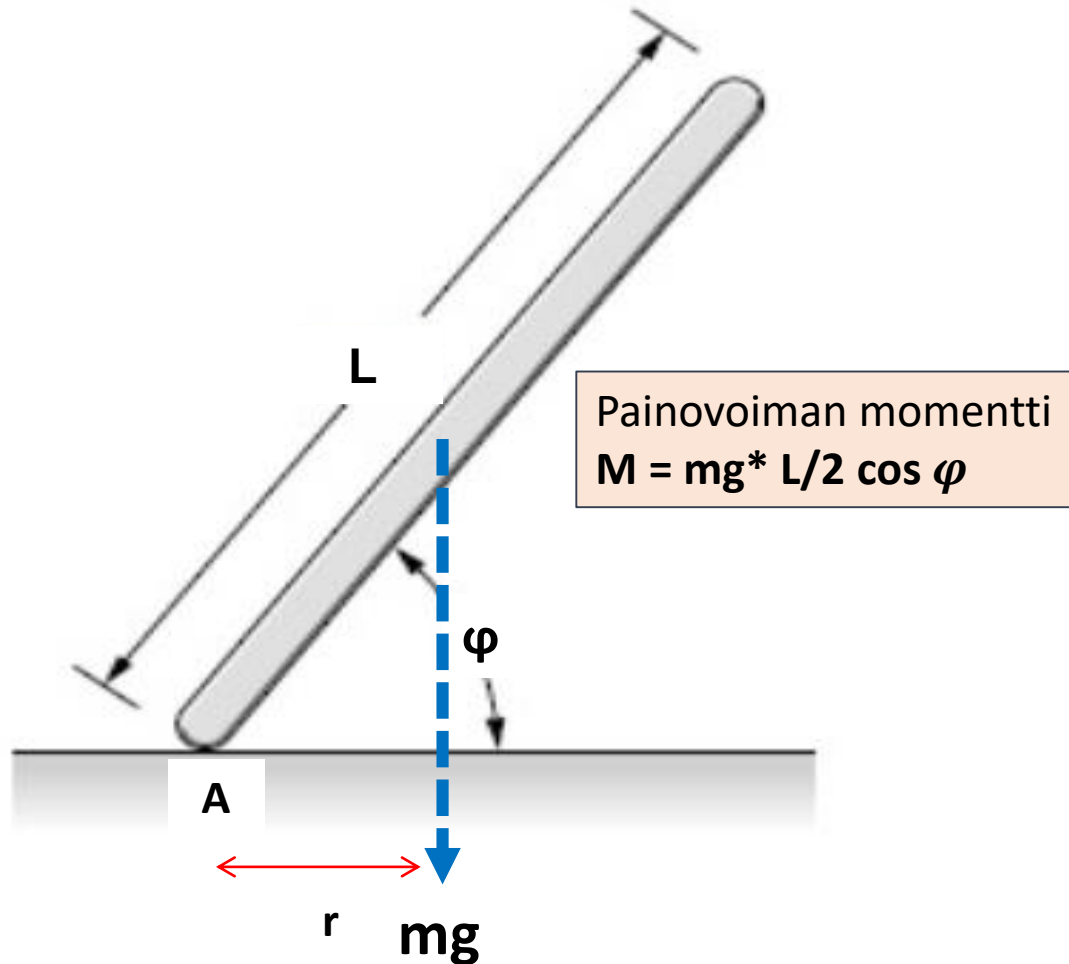
**palkki.axis** antaa palkin rungon vektorina, jonka  
pituus = palkin pituus

Kuvassa palkin pituus =  $L$  ja kulma x- akselin kanssa =  $\varphi$   
Tällöin palkin akseli määritetään seuraavasti

**palkki.axis =  $\text{vec}(L \cdot \cos \varphi, L \cdot \sin \varphi, 0)$**



# KAATUVAN PALKIN ANIMAATIOSILMUKKA



Palkin hitausmomentti sen pyöriessä pään ympäri:  
 $J = \frac{1}{3} mL^2$  (Maol:n taulukot)

Laske kulmakiikhtyvyys

$$\alpha = M/J = \frac{mgL/2\cos\varphi}{\frac{1}{3}mL^2} = \frac{3g}{2L}\cos\varphi$$

Päivitä palkin kulma

$$\varphi += \omega dt + 0.5 \alpha dt^2$$

Päivitä palkin kulmanopeus

$$\omega += \alpha dt$$

Päivitä palkin akseli uuteen asentoon

$$\text{palkki.axis} = \text{vec}(L * \cos \varphi, L * \sin \varphi, 0)$$

GlowScript 2.9 VPython

scene.width=600

scene.height=450

scene.range=12

scene.center=vec(0,4,0)

scene.title='Pylvaan kaatuminen, click to start'

L=8 #pylvaan pituus

kulma=radians(88) #pylvaan asento ennen kaatumista x- akseliin nähden 88°

w=0 #pylvaan kulmanopeus alkutilanteessa on 0

e=0.3 # elastisuus pylvaan tomahtaessa maahan

alusta=cylinder(pos=vec(0,-0.1,0),radius=10, axis=vec(0,0.2,0),color=0.4\*color.green)

pylvas=cylinder(pos=vec(0,0,0),radius=0.5,axis=vec(L\*cos(kulma),L\*sin(kulma),0),color=0.4\*vec(1,0.5,0.25))

g=9.81 #painovoiman kiihtyvyys

dt=0.01 #iteraatioaskel

scene.pause()

scene.title=L+ ' m korkea pylvas kaatuu' #ikkunan otsikko muutetaan

while kulma>-0.02: #toistetaan kun kulma on positiivinen

rate(100)

alfa=-3\*g/(2\*L)\*cos(kulma) #lasketaan kulmakiihtyvyys  $\alpha = M/J$

kulma+=w\*dt+0.5\*alfa\*dt\*\*2 #kulman paivitys

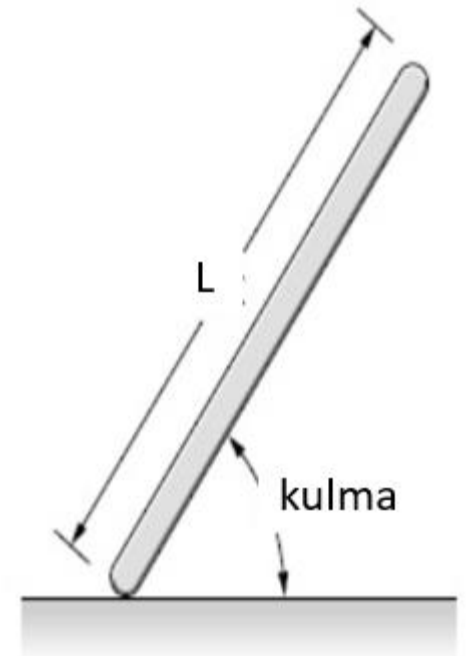
w+=alfa\*dt #kulmanopeuden paivitys

pylvas.axis=vec(L\*cos(kulma),L\*sin(kulma),0) #paivitetaan pylvaan suunta

if kulma<0.01 and w<0: # kun palkki osuu maahan, se pomppaa hieman

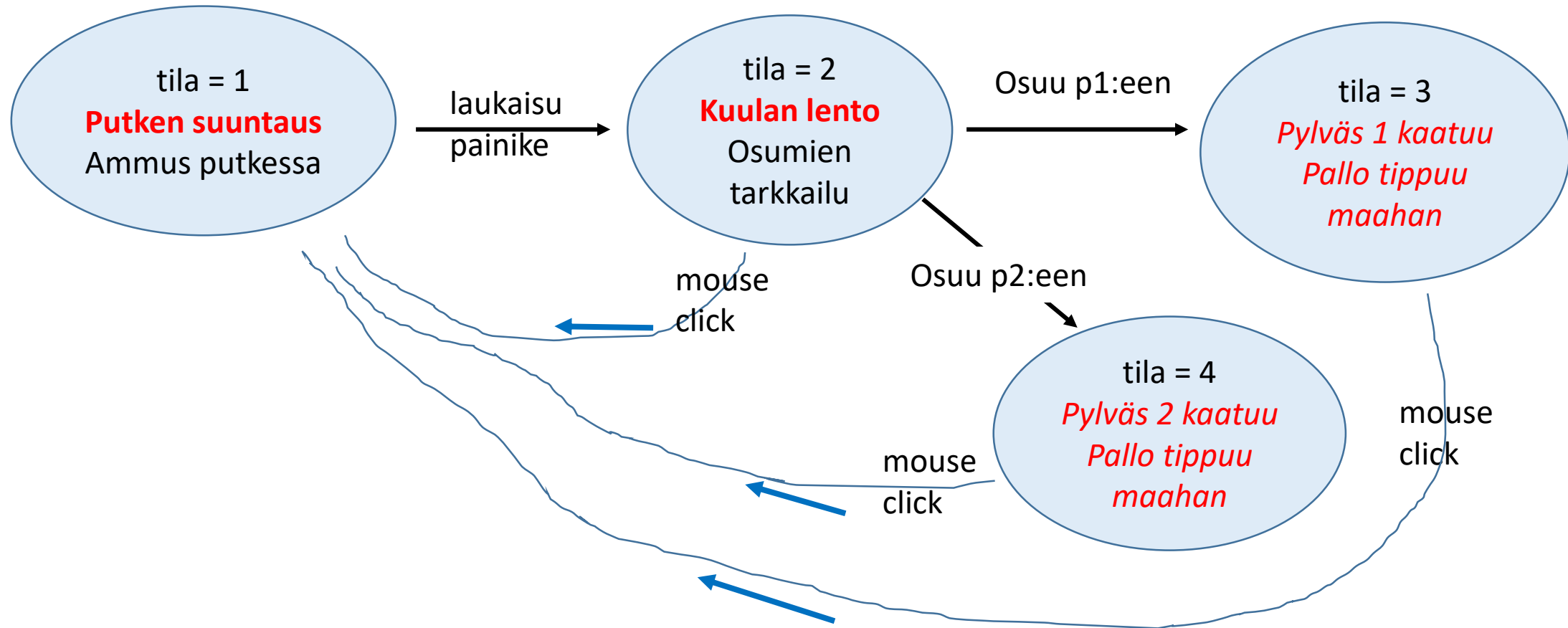
w=-e\*w

Viereisessä esimerkkiani-  
maatiassa pylväs kaatuu  
maahan



# Tykkisovelluksessa käytetään tilakonemallinnusta

(Kuvataan tilat ja triggerit, joilla siirrytään tilasta toiseen)



tila = 1

while tila < 5:

rate(100)

if tila == 1 :

*putken suuntaus*

*laukaisu => tila 2*

if tila == 2:

*kuulan lento*

*osumien tarkkailu*

*jos osuu => tila 3 tai 4*

*jos ohi = > klik:lla tilaan 1*

if tila == 3:

*palkki 1 kaatuu*

*pallo tippuu maahan*

*klik:lla tilaan 1*

if tila == 4:

*palkki2 kaatuu,*

*muuten sama kuin 3*

Silmukan rakenne: tilamuuttuja ohjaa  
mitä animaatioissa tapahtuu

Monimutkainen tapahtumaketju  
voidaan hallita tilakone -tekniikalla