

```
"draws a rotating ? line cube
"this is a test of my 3dengine in liberty
"what i wil need for animations
"see spheres
dim mv( 64 , 5 ) , sk( 64 , 2 ) , cam( 6 )
global nummer , pi , frame
pi = atn( 1 ) * 4
frame = 0
WindowWidth = DisplayWidth
WindowHeight = DisplayHeight
global zwart , rood , groen , geel
global blauw , magenta , cyaan , wit
global roze , oranje , grijs , paars
zwart = rgb( 000 , 000 , 000 )
rood = rgb( 255 , 000 , 000 )
groen = rgb( 000 , 255 , 000 )
geel = rgb( 255 , 255 , 000 )
blauw = rgb( 000 , 000 , 255 )
magenta = rgb( 255 , 000 , 255 )
cyaan = rgb( 000 , 255 , 255 )
wit = rgb( 255 , 255 , 255 )
roze = rgb( 255 , 127 , 127 )
oranje = rgb( 255 , 127 , 000 )
grijs = rgb( 127 , 127 , 127 )
paars = rgb( 127 , 000 , 127 )
global wereld , lijf , beenl , beenr
global kniel , klier , staart
global arml , armr , ell , elr , handl , handr
global duiml , duimr , vingerl , vingerr , frame
wereld = 0 : lijf = 1
arml = 2 : armr = 3 : ell = 4 : elr = 5
handl = 6 : handr = 7
duiml = 8 : duimr = 9
vingerl = 10 : vingerr = 11
beenl = 15 : beenr = 20
kniel = 25 : knier = 30
staart = 35
a = 0 : b = 1 : c = 3
print a , b , c
call link 1 ,0,0,0 , 30,0,0 , 0
call spot a , b , c
print a , b , c
wait
a = 0 : b = 1 : c = 3
print
print a , b , c
```

```
call link 1 ,0,0,0 , -30,0,0 , 0
call spot a , b , c
print a , b , c
end
"nomainwin
timer 40 , [timer_tick]
open "lines" for graphics as #m
print #m , "trapclose [quit]"
wait
[timer_tick]
print #m , "cls"
frame = frame + 5
call camara 0,0,0 , 0,0,0 , 1
call link 2 , 0,0,0 , 0,0,0 , 0
call kubus 0,0,0 _
, 100,100,100 , 5 , rainbow( frame )
print #m , "flush"
wait
"3d engine spul
"hier onder niet aanzitten
"tenzij je weet wat je doet !!
"=====
sub link no , x , y , z , xz , yz , xy , p
if no 64 then exit sub
if p 64 then exit sub
if n = p then exit sub
call rotate x , y , mv( p , 5 )
call rotate y , z , mv( p , 4 )
call rotate x , z , mv( p , 3 )
mv( no , 0 ) = x + mv( p , 0 )
mv( no , 1 ) = y + mv( p , 1 )
mv( no , 2 ) = z + mv( p , 2 )
mv( no , 3 ) = xz + mv( p , 3 )
mv( no , 4 ) = yz + mv( p , 4 )
mv( no , 5 ) = xy + mv( p , 5 )
nummer = no
end sub
sub child no , x , y , z , lim , p
if lim 64 then exit sub
call link no , x , y , z _
, sk( lim , 1 ) _
, sk( lim , 0 ) _
, sk( lim , 2 ) , p
end sub
sub spot byref x , byref y , byref z
call rotate x , y , mv( nummer , 5 )
```

```
call rotate y , z , mv( nummer , 4 )
call rotate x , z , mv( nummer , 3 )
if cam( 6 ) = 0 then cam( 6 ) = 1
x = ( x + mv( nummer , 0 ) - cam( 0 ) ) * cam( 6 )
y = ( y + mv( nummer , 1 ) - cam( 1 ) ) * cam( 6 )
z = ( z + mv( nummer , 2 ) - cam( 2 ) ) * cam( 6 )
call rotate x , z , -cam( 3 )
call rotate y , z , -cam( 4 )
call rotate x , y , -cam( 5 )
end sub
sub rotate byref k , byref l , deg
s = sin( rad( deg ) )
c = cos( rad( deg ) )
kh = k * c - l * s
lh = k * s + l * c
k = kh
l = lh
end sub
sub camara x , y , z , xz , yz , xy , zoom
cam( 0 ) = x
cam( 1 ) = y
cam( 2 ) = z
cam( 3 ) = xz
cam( 4 ) = yz
cam( 5 ) = xy
cam( 6 ) = zoom
end sub
sub angle no , ax , deg
if no < 64 then exit sub
if ax < 2 then exit sub
sk( no , ax ) = deg
end sub
function rad( x )
rad = x * pi / 180
end function
"=====
"einde 3d engine spul

"graphics
sub kubus mx , my , mz , dx , dy , dz , dik , kl
call lijn mx - dx , my - dy , mz - dz _
, mx + dx , my - dy , mz - dz , dik , kl
call lijn mx - dx , my + dy , mz - dz _
, mx + dx , my + dy , mz - dz , dik , kl
call lijn mx - dx , my - dy , mz + dz _
, mx + dx , my - dy , mz + dz , dik , kl
```

```
call lijn mx - dx , my + dy , mz + dz _
, mx + dx , my + dy , mz + dz , dik , kl
call lijn mx - dx , my - dy , mz - dz _
, mx - dx , my + dy , mz - dz , dik , kl
call lijn mx + dx , my - dy , mz - dz _
, mx + dx , my + dy , mz - dz , dik , kl
call lijn mx - dx , my - dy , mz + dz _
, mx - dx , my + dy , mz + dz , dik , kl
call lijn mx + dx , my - dy , mz + dz _
, mx + dx , my + dy , mz + dz , dik , kl
call lijn mx - dx , my - dy , mz - dz _
, mx - dx , my - dy , mz + dz , dik , kl
call lijn mx + dx , my - dy , mz - dz _
, mx + dx , my - dy , mz + dz , dik , kl
call lijn mx - dx , my + dy , mz - dz _
, mx - dx , my + dy , mz + dz , dik , kl
call lijn mx + dx , my + dy , mz - dz _
, mx + dx , my + dy , mz + dz , dik , kl
end sub
sub lijn x1 , y1 , z1 , x2 , y2 , z2 , dik , kl
r = int( kl ) and 255
g = int( kl / 256 ) and 255
b = int( kl / 256 / 256 ) and 255
dx = WindowWidth
dy = WindowHeight
call spot x1 , y1 , z1
call spot x2 , y2 , z2
ax = dx / 2 + x1 / ( z1 + 1000 ) * 1000
ay = dy / 2 - y1 / ( z1 + 1000 ) * 1000
bx = dx / 2 + x2 / ( z2 + 1000 ) * 1000
by = dy / 2 - y2 / ( z2 + 1000 ) * 1000
print #m , "size "; dik
print #m , "color "; r ; " "; g ; " "; b
print #m , "down"
print #m , "line "; ax ; " "; ay ; " "; bx ; " "; by
print #m , "up"
end sub
sub bol x , y , z , d , dik , kl
call spot x , y , z
d = d * cam( 6 )
r = int( kl and 255 )
g = int( kl / 256 ) and 255
b = int( kl / 256 / 256 ) and 255
print #m , "color "; r ; " "; g ; " "; b
d = d / ( z + 1000 ) * 1000
dx = WindowWidth
```

```
dy = WindowHeight
x = dx / 2 + x / ( z + 1000 ) * 1000
y = dy / 2 - y / ( z + 1000 ) * 1000
print #m , "go "; x ; " "; y
print #m , "down"
print #m , "circle "; d
print #m , "up"
end sub
sub insect kl
call kubus 0,0,0 , 30,10,60 , 1 , kl
for i = 0 to 5
call child beenl + i , 30 , 0 , -50 + i * 25 _
, beenl + i , lijf
call lijn 0,0,0 , 60,0,0 , 1 , kl
call child kniel + i , 60 , 0 , 0 _
, kniel , lijf
call lijn 0,0,0 , 0,-60,0 , 1 , kl
next i
end sub
function rainbow( x )
r = sin( rad( x ) ) * 127 + 128
g = sin( rad( x - 120 ) ) * 127 + 128
b = sin( rad( x + 120 ) ) * 127 + 128
rainbow = rgb( r , g , b )
end function
function rgb( r , g , b )
r = r and 255
g = g and 255
b = b and 255
rgb = r + g * 256 + b * 256 * 256
end function
[quit]
close #m
end
```