

```
'version 20 aug 2010
'demonstration of scripting of
'a simple scene
'whit a animation
'and a jointed lim
on error goto [error]
dim mv( 64 , 5 ) , sk( 64 , 2 ) , cam( 6 ) , pen( 5 )
dim pal( 255 )
global number , pi , frame , cubo$
pi = atn( 1 ) * 4
frame = 0
WindowWidth = DisplayWidth
WindowHeight = DisplayHeight
global black , red , green , yellow
global blue , magenta , cyan , white
global pink , orange , gray , purple
black = rgb( 000 , 000 , 000 )
red = rgb( 255 , 000 , 000 )
green = rgb( 000 , 255 , 000 )
yellow = rgb( 255 , 255 , 000 )
blue = rgb( 000 , 000 , 255 )
magenta = rgb( 255 , 000 , 255 )
cyan = rgb( 000 , 255 , 255 )
white = rgb( 255 , 255 , 255 )
pink = rgb( 255 , 127 , 127 )
orange = rgb( 255 , 127 , 000 )
gray = rgb( 127 , 127 , 127 )
purple = rgb( 127 , 000 , 127 )
'load freeshape object sript
cubo$ = loadscript$( "lino-cubo.txt" )
nomainwin
open "lino3D" for graphics as #m
print #m , "trapclose [quit]"
timer 250 , [tmr]
wait
[tmr]
#m "fill black"
frame = frame + 3
q = sin(rad(frame)* 2)*200
'set place and orientation of camara
call camara 0,0,0 , 0,0,0 , 1
call setpen 0,0,0 , frame,0,0
'set index colors of script
call palette 1 , red
call palette 2 , red
call palette 3 , red
```

```
call palette 4 , magenta
''run script and scale it
call freeshape cubo$ , 1, 1, 1
call setpen 0,0,0 , frame,0,0
call movepen 150,0,0 , 0,frame*2,0
''set index colors of script
call palette 1 , blue
call palette 2 , blue
call palette 3 , blue
call palette 4 , cyan
''run script and scale it
call freeshape cubo$ ,.5,.5,.5
print #m , "flush"
wait
''3d engine stuf
''=====
sub setpen x , y , z , xz , yz , xy
pen( 0 ) = x
pen( 1 ) = y
pen( 2 ) = z
pen( 3 ) = xz mod 360
pen( 4 ) = yz mod 360
pen( 5 ) = xy mod 360
end sub
sub movepen x , y , z , xz , yz , xy
call rotate x , y , pen( 5 )
call rotate y , z , pen( 4 )
call rotate x , z , pen( 3 )
pen( 0 ) = x + pen( 0 )
pen( 1 ) = y + pen( 1 )
pen( 2 ) = z + pen( 2 )
pen( 3 ) = xz + pen( 3 )
pen( 4 ) = yz + pen( 4 )
pen( 5 ) = xy + pen( 5 )
end sub
sub link no , x , y , z , xz , yz , xy , p
if no < 1 or no > 64 then exit sub
if p < 0 or 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 )
number = no
end sub
sub child no , x , y , z , lim , p
if lim < 1 or 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 , pen( 5 )
call rotate y , z , pen( 4 )
call rotate x , z , pen( 3 )
if cam( 6 ) = 0 then cam( 6 ) = 1
x = ( x + pen( 0 ) - cam( 0 ) )
y = ( y + pen( 1 ) - cam( 1 ) )
z = ( z + pen( 2 ) - cam( 2 ) )
call rotate x , z , 0 - cam( 3 )
call rotate y , z , 0 - cam( 4 )
call rotate x , y , 0 - cam( 5 )
x = x * cam( 6 )
y = y * cam( 6 )
z = z * cam( 6 )
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 < 1 or no > 64 then exit sub
if ax < 0 or ax > 2 then exit sub
sk( no , ax ) = deg
end sub
sub rotate byref k , byref l , deg
s = sin( rad( deg mod 360 ) )
c = cos( rad( deg mod 360 ) )
kh = k * c - l * s
lh = k * s + l * c
k = kh
```

```
l = lh
end sub
function rad( x )
rad = x * pi / 180
end function
'''=====
'''end 3d engine stuf
'''graphics
function loadscript$( file$ )
file$ = DefaultDir$;"\scripts\";file$
open file$ for input as #in
txt$ = input$( #in , lof( #in ) )
close #in
loadscript$ = txt$
end function
[error]
close #in
notice Err$
end
sub palette no , clr
pal( no ) = clr
end sub
sub freeshape obj$ , sx , sy , sz
if word$( obj$ , 1) <> "lino3D" then exit sub
pointer = 2
while word$( obj$ , pointer ) <> "end"
comand$ = word$( obj$ , pointer )
select case comand$
case "lino"
x1 = val( word$( obj$ , pointer + 1 ) )
y1 = val( word$( obj$ , pointer + 2 ) )
z1 = val( word$( obj$ , pointer + 3 ) )
x2 = val( word$( obj$ , pointer + 4 ) )
y2 = val( word$( obj$ , pointer + 5 ) )
z2 = val( word$( obj$ , pointer + 6 ) )
clr$ = word$( obj$ , pointer + 7 )
if val( clr$ ) < 0 then
c = pal( abs( val( clr$ ) ) )
else
c = dec( clr$ )
end if
call lino x1*sx , y1*sy , z1*sz _
, x2*sx , y2*sy , z2*sz , 3 , c
pointer = pointer + 8
case "rotate"
pan = val( word$( obj$ , pointer + 1 ) ) * frame
```

```
tilt = val( word$( obj$ , pointer + 2 ) ) * frame
rol = val( word$( obj$ , pointer + 3 ) ) * frame
call setpen pen(0),pen(1),pen(2),pan,tilt,rol
pointer = pointer + 4
case else : exit sub
end select
wend
end sub
function dec( h$ )
''i m testing this in justbasic so this has to be
if len( h$ ) > 6 then dec = 0
if len( h$ ) < 1 then dec = 0
som = 0
for i = 1 to len( h$ )
digit = instr( "123456789abcdef" , mid$( h$ , i , 1 ) )
som = som + digit * 16 ^ ( len( h$ ) - i )
next i
dec = som
end function
sub cubo mx , my , mz , dx , dy , dz , dik , kl
call lino mx - dx , my - dy , mz - dz _
, mx + dx , my - dy , mz - dz , dik , kl
call lino mx - dx , my + dy , mz - dz _
, mx + dx , my + dy , mz - dz , dik , kl
call lino mx - dx , my - dy , mz + dz _
, mx + dx , my - dy , mz + dz , dik , kl
call lino mx - dx , my + dy , mz + dz _
, mx + dx , my + dy , mz + dz , dik , kl
call lino mx - dx , my - dy , mz - dz _
, mx - dx , my + dy , mz - dz , dik , kl
call lino mx + dx , my - dy , mz - dz _
, mx + dx , my + dy , mz - dz , dik , kl
call lino mx - dx , my - dy , mz + dz _
, mx - dx , my + dy , mz + dz , dik , kl
call lino mx + dx , my - dy , mz + dz _
, mx + dx , my + dy , mz + dz , dik , kl
call lino mx - dx , my - dy , mz - dz _
, mx - dx , my - dy , mz + dz , dik , kl
call lino mx + dx , my - dy , mz - dz _
, mx + dx , my - dy , mz + dz , dik , kl
call lino mx - dx , my + dy , mz - dz _
, mx - dx , my + dy , mz + dz , dik , kl
call lino mx + dx , my + dy , mz - dz _
, mx + dx , my + dy , mz + dz , dik , kl
end sub
sub okto x , y , z , dx , dy , dz , t , kl
```

```
call lino x,y+dy,z,x,y,z+dz,t,kl
call lino x,y,z+dz,x,y-dy,z,t,kl
call lino x,y-dy,z,x,y,z-dz,t,kl
call lino x,y,z-dz,x,y+dy,z,t,kl
call lino x+dx,y,z,x,y,z+dz,t,kl
call lino x,y,z+dz,x-dx,y,z,t,kl
call lino x-dx,y,z,x,y,z-dz,t,kl
call lino x,y,z-dz,x+dx,y,z,t,kl
call lino x+dx,y,z,x,y+dy,z,t,kl
call lino x,y+dy,z,x-dx,y,z,t,kl
call lino x-dx,y,z,x,y-dy,z,t,kl
call lino x,y-dy,z,x+dx,y,z,t,kl
end sub
sub opo x , y , z , d , sides , t , kl
if sides < 3 then sides = 3
if sides > 24 then sides = 24
for i = 0 to sides
a=i*pi*2/sides
b=(i+1)*pi*2/sides
call lino sin(a)*d+x , cos(a)*d+y , z _
, sin(b)*d+x , cos(b)*d+y , z _
, t , kl
next i
end sub
sub lino 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
#m "size "; dik
#m "color "; r ;" "; g ;" "; b
#m "down"
#m "line "; ax ;" "; ay ;" "; bx ;" "; by
#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( k1 / 256 ) and 255
b = int( k1 / 256 / 256 ) and 255
#m "color "; r ;" "; g ;" "; b
#m "backcolor " ; 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
#m "size " ; dik
#m "go "; x ;" "; y
#m "down"
#m "ellipse " ; d ; " " ; d
#m "up"
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
```