

```
"version 20/3/2011
WindowWidth = DisplayWidth
WindowHeight = DisplayHeight
global winx , winy , frame , pi
global scrnx , scrny , eye , you
global balx , baly , balz , baldx , baldy , baldz
global humanx , humany , humanz , humandz
global state , speed , frame , anascript$
dim sk( 20 , 3 )
global leg , knee , enkle , sholder , elbow , wrist , right
leg = 1
knee = 2
enkle = 3
sholder = 4
elbow = 5
wrist = 6
right = 8
winx = WindowWidth
winy = WindowHeight
pi = atn( 1 ) * 4
scrnx$ = str$( 350 )
prompt "Screen width in mm =" ; scrnx$
scrnx = val( scrnx$ )
scrny$ = str$( 280 )
prompt "Screen height in mm =" ; scrny$
scrny = val( scrny$ )
you$ = str$( 350 )
prompt "You - screen in mm =" ; you$
you = val( you$ )
eye = 70 "pupil distance in mm
dim pen( 6 ) , cam( 6 )
nomainwin
open "anaglyph 3D pong" for graphics as #m
#m "trapclose [quit]"
#m "when charaterInput [key]"
#m "fill black"
#m "rule "; _R2_MERGEOPEN
humanz = 50
call scene
timer 10000 , [tmr]
wait
end
[quit]
close #m
end
[tmr]
```

```
call scene
wait
[key]
select case right$( Inkey$ , 1 )
case "1"
state = state + 5
speed = -15
case "2"
speed = -30
case "3"
state = state - 5
speed = -15
case "4"
state = state + 5
speed = 0
case "5"
" humandy = 20
case "6"
state = state - 5
speed = 0
case "7"
state = state + 5
case "8"
speed = 30
case "9"
state = state - 5
case else
speed = 0
end select
wait
sub scene
#m "fill black"
q$ = ""
w = rnd(0)*6+6
for i = 0 to w
x = rnd(0)*winx/2-winx/4
y = rnd(0)*winy/2-winy/4
z = rnd(0)*winx/2-winx/4
q$=q$;x;" ";y;" ";z;"| "
next i
call bspline q$
#m "flush"
end sub
sub man x , y , z , pan , fase , amp
call setpen x,y,z , pan,0,0
call lino 0,0,0 , 0,50,0 , 3
```

```
call lino -10,0,0 , 10,0,0 , 3
call sphere 0,70,0 , 20 , 3
call lino -20,50,0 , 20,50,0 , 3
call setpen x,y,z , pan,0,0
call movepen -20,50,0 , 0,sin(rad(fase+180))*amp,0
call lino 0,0,0 , 0,-30,0 , 3
call lino 0,-30,0 , 0,-30,-30 , 3
call setpen x,y,z , pan,0,0
call movepen 20,50,0 , 0,sin(rad(fase))*amp,0
call lino 0,0,0 , 0,-30,0 , 3
call lino 0,-30,0 , 0,-30,-30 , 3
call setpen x,y,z , pan,0,0
call movepen -10,0,0 , 0,sin(rad(fase))*amp,0
call lino 0,0,0 , 0,-30,0 , 3
call movepen 0,-30,0 , 0,-30-cos(rad(fase))*amp,0
call lino 0,0,0 , 0,-30,0 , 3
call lino 0,-30,0 , 0,-30,-20 , 3
call setpen x,y,z , pan,0,0
call movepen 10,0,0 , 0,sin(rad(fase+180))*amp,0
call lino 0,0,0 , 0,-30,0 , 3
call movepen 0,-30,0 , 0,-30-cos(rad(fase+180))*amp,0
call lino 0,0,0 , 0,-30,0 , 3
call lino 0,-30,0 , 0,-30,-20 , 3
end sub
function tox( x , y , z , rl )
"catch x/0 error
if z + you = 0 then tox = 0
"offset red or blue
o = ( eye / 2 ) / ( z + you ) * you - ( eye / 2 )
o = o * rl
"offset z + perspertif
a = ( x + o ) / ( z + you ) * you
"from mm to pixels
tox = winx / 2 + a * winx / scrnx
end function
function toy( x , y , z )
"catch x/0 error
if z + you = 0 then toy = 0
"offset z + perspectif
a = y / ( z + you ) * you
"from mm to pixels
toy = winy / 2 - a * winy / scrny
end function
function lenght( x , y , z )
lenght = sqr( x^2 + y^2 + z^2 )
end function
```

```

sub bezier x1,y1,z1 , x2,y2,z2 , x3,y3,z3 , x4,y4,z4 , t
if lenght(x1-x2,y1-y2,z1-z2) > 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
next i
end sub

sub dodeca x , y , z , d , dik
f = ( sqrt( 5 ) - 1 ) / 2
" (±1, ±1, ±1)
" (0, ±1/f, ±f)
" (±1/f, ±f, 0)
" (±f, 0, ±1/f)
call lino x + d , y + d , z + d , x , y + 1/f*d , z + f*d ,dik
call lino x + d , y + d , z + d , x + 1/f*d , y + f*d , z ,dik
call lino x + d , y + d , z + d , x + f*d , y , z + 1/f*d ,dik
call lino x - d , y - d , z - d , x , y - 1/f*d , z - f*d ,dik
call lino x - d , y - d , z - d , x - 1/f*d , y - f*d , z ,dik
call lino x - d , y - d , z - d , x - f*d , y , z - 1/f*d ,dik
call lino x+1/f*d,y+f*d,z,x+1/f*d,y-f*d,z,dik
call lino x-1/f*d,y-f*d,z,x-1/f*d,y+f*d,z,dik
call lino x,y+1/f*d,z+f*d,x,y+1/f*d,z-f*d,dik
call lino x,y-1/f*d,z-f*d,x,y-1/f*d,z+f*d,dik
call lino x-f*d,y,z-1/f*d,x+f*d,y,z-1/f*d,dik
call lino x+f*d,y,z+1/f*d,x-f*d,y,z+1/f*d,dik
call lino x+1/f*d,y+f*d,z,x+d,y+d,z-d,dik
call lino x-1/f*d,y-f*d,z,x-d,y-d,z+d,dik
call lino x+f*d,y,z+1/f*d,x-f*d,y,z+1/f*d,dik
call lino x-f*d,y,z-1/f*d,x+f*d,y,z-1/f*d,dik
call lino x-f*d,y,z+1/f*d,x-d,y+d,z+d,dik
call lino x+f*d,y,z-1/f*d,x+d,y-d,z-d,dik
call lino x+f*d,y,z-1/f*d,x+d,y+d,z-d,dik
call lino x-f*d,y,z+1/f*d,x-d,y-d,z+d,dik
call lino x-d,y+d,z+d,x,y+1/f*d,z+f*d,dik
call lino x+d,y-d,z-d,x,y-1/f*d,z-f*d,dik
call lino x-d,y+d,z+d,x-1/f*d,y+f*d,z,dik
call lino x+d,y-d,z-d,x+1/f*d,y-f*d,z,dik
call lino x+f*d,y,z+1/f*d,x+d,y-d,z+d,dik
call lino x-f*d,y,z-1/f*d,x-d,y+d,z-d,dik
call lino x+d,y-d,z+d,x,y-1/f*d,z+f*d,dik
call lino x-d,y+d,z-d,x,y+1/f*d,z-f*d,dik
call lino x+d,y+d,z-d,x,y+1/f*d,z-f*d,dik
call lino x-d,y-d,z+d,x,y-1/f*d,z+f*d,dik
call lino x+d,y-d,z+d,x+1/f*d,y-f*d,z,dik

```

```
call lino x-d,y+d,z-d,x-1/f*d,y+f*d,z,dik
end sub
sub cubo mx , my , mz , dx , dy , dz , thick
call lino mx+dx,my+dy,mz+dz,mx-dx,my+dy,mz+dz,thick
call lino mx+dx,my+dy,mz-dz,mx-dx,my+dy,mz-dz,thick
call lino mx+dx,my-dy,mz+dz,mx-dx,my-dy,mz+dz,thick
call lino mx+dx,my-dy,mz-dz,mx-dx,my-dy,mz-dz,thick
call lino mx+dx,my+dy,mz+dz,mx+dx,my-dy,mz+dz,thick
call lino mx+dx,my+dy,mz-dz,mx+dx,my-dy,mz-dz,thick
call lino mx-dx,my+dy,mz+dz,mx-dx,my-dy,mz+dz,thick
call lino mx-dx,my+dy,mz-dz,mx-dx,my-dy,mz-dz,thick
call lino mx+dx,my+dy,mz+dz,mx+dx,my+dy,mz-dz,thick
call lino mx+dx,my-dy,mz+dz,mx+dx,my-dy,mz-dz,thick
call lino mx-dx,my+dy,mz+dz,mx-dx,my+dy,mz-dz,thick
call lino mx-dx,my-dy,mz+dz,mx-dx,my-dy,mz-dz,thick
end sub
sub setpen x , y , z , pan , tilt , rol
pen( 0 ) = x
pen( 1 ) = y
pen( 2 ) = z
pen( 3 ) = pan mod 360
pen( 4 ) = tilt mod 360
pen( 5 ) = rol mod 360
end sub
sub movepen x , y , z , pan , tilt , rol
call rotate x , y , pen( 5 )
call rotate y , z , pen( 4 )
call rotate x , z , pen( 3 )
call setpen pen( 0 ) + x , pen( 1 ) + y , pen( 2 ) + z _
, pen( 3 ) + pan , pen( 4 ) + tilt , pen( 5 ) + rol
end sub
sub camara x , y , z , pan , tilt , rol
cam( 0 ) = x
cam( 1 ) = x
cam( 2 ) = x
cam( 3 ) = pan mod 360
cam( 4 ) = tilt mod 360
cam( 5 ) = rol mod 360
end sub
sub rotate byref k , byref l , deg
s = sin( rad( deg ) )
c = cos( rad( deg ) )
hk = k * c - l * s
hl = k * s + l * c
k = hk : l = hl
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 )
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 )
end sub
function rad( x )
rad = x * pi / 180
end function
```