```
''version 12 aug 2010
''x/0 stil there !!
[part_declarations]
WindowWidth = DisplayWidth
WindowHeight = DisplayHeight
global winx , winy
winx = WindowWidth
winy = WindowHeight
global trimax , tritel , pi , frame
global black , red , green , yellow
global blue , magenta , cyan , white
global pink , purple , gray , orange
black = rgb( 0 , 0 , 0 )
red = rgb( 255 , 0 , 0 )
green = rgb( 0 , 255 , 0 )
yellow = rgb( 255 , 255 , 0 )
blue = rgb( 0 , 0 , 255 )
magenta = rgb( 255 , 0 , 255 )
cyan = rgb( 0 , 255 , 255 )
white = rgb( 255 , 255 , 255 )
pink = rgb( 255 , 127 , 127 )
orange = rgb( 255 , 127 , 0 )
gray = rgb( 127 , 127 , 127 )
purple = rgb( 127 , 0 , 127 )
trimax = 1000
pi = atn( 1 ) * 4
dim h3( trimax , 16 ) , ry( trimax ) , pnt( 256 , 2 ) , cam( 6 )
global boxx , boxy , boxz , boxdx , boxdy , boxdz
global frame
''al var's whit o... are for OOP like programing
''points
global ox1,oy1,oz1,ox2,oy2,oz2,ox3,oy3,oz3
ox1 = 0 : oy1 = 1 : oz1 = 2
ox2 = 3 : oy2 = 4 : oz2 = 5
ox3 = 6 : oy3 = 7 : oz3 = 8
''zpoint normal color
global oxz,oyz,ozz,oxn,oyn,ozn,oclr
oxz = 9 : oyz = 10 : ozz = 11
oxn = 12 : oyn = 13 : ozn = 14
oclr = 15
''for sorting objects
for i = 0 to trimax
ry( i ) = i
next i
nomainwin
graphicbox #m.g , 0 , 0 , winx , winy
```

```
open "VR" for window as #m
#m "trapclose [quit]"
#m.g "setfocus"
#m.g "when characterInput [key]"
call scene
''remove rem by the folowing after saving
''rem is there so you can see something whitout x/0
'' timer 500 , [tmr]
wait
[part_game_flow]
[tmr]
frame = ( frame + 5 ) mod 360
call scene
wait
sub scene
''in examples i wil only list this sub
#m.g "fill black"
call clear
call camara 0,0,0 , 0,frame,0 , 1
call punt 0 , 0 , 0 , 0
call punt 1 , -100 , -100 , 0
call punt 2 , -100 , 100 , 0
call punt 3 , 100 , 100 , 0
call punt 4 , 100 , -100 , 0
'' call d3 0 , 1 , 2 , cyan
'' call d3 0 , 2 , 3 , yellow
''draw triangle whit abs color
caLL d3 0 , 1 , 2 , 0-green
''draw triangle whit shaded color
call d3 0 , 3 , 4 , green
call draw
#m.g "flush"
end sub
[key]
if right$( Inkey$ , 1 ) _
= chr$( _VK_ESCAPE ) then
close #m
end
end if
wait
[quit]
close #m
end
[part_graphics]
sub clear
''clear al triangles
```

```
tritel = 0
for i = 0 to trimax
for d = 0 to 15
h3( i , d ) = 0
next d
next i
end sub
sub punt no , x , y , z
''fil a point in the swarm
if no < 0 or no > 256 then exit sub
pnt( no , 0 ) = x
pnt( no , 1 ) = y
pnt( no , 2 ) = z
end sub
sub d3 p1 , p2 , p3 , clr
''create triangle of points of swarm
''get points
x1 = pnt( p1 , 0 )
y1 = pnt( p1 , 1 )
z1 = pnt( p1 , 2 )
x2 = pnt( p2 , 0 )
y2 = pnt( p2 , 1 )
z2 = pnt( p2 , 2 )
x3 = pnt( p3 , 0 )
y3 = pnt( p3 , 1 )
z3 = pnt( p3 , 2 )
''put points trou shape to world sub
call spot x1 , y1 , z1
call spot x2 , y2 , z2
call spot x3 , y3 , z3
''store points
h3( tritel , ox1 ) = x1
h3( tritel , oy1 ) = y1
h3( tritel , oz1 ) = z1
h3( tritel , ox2 ) = x2
h3( tritel , oy2 ) = y2
h3( tritel , oz2 ) = z2
h3( tritel , ox3 ) = x3
h3( tritel , oy3 ) = y3
h3( tritel , oz3 ) = z3
h3( tritel , oclr ) = clr
''zpoint
h3( tritel , ozz ) = ( z1 + z2 + z3 ) / 3
''take sure that in sub draw no x/0 happen
if z1 > -900 and z2 > -900 and z3 > -900 then
tritel = tritel + 1
```

```
end if
''normal = crosproduct( p2-p1 , p3-p1 )
h3(tritel,oxn)=(y2-y1)*(z3-z1)-(y3-y1)*(z2-z1)
h3(tritel,oyn)=(z2-z1)*(x3-x1)-(z3-z1)*(x2-x1)
h3(tritel,ozn)=(x2-x1)*(y3-y1)-(x3-x1)*(y2-y1)
end sub
sub d4 p1 , p2 , p3 , p4 , clr
''create quadangle whit points of swarm
x = (pnt( p1 , 0 ) + pnt( p2 , 0 ) _
+ pnt( p3 , 0 ) + pnt( p4 , 0 ))/4
y = (pnt( p1 , 1 ) + pnt( p2 , 1 ) _
+ pnt( p3 , 1 ) + pnt( p4 , 1 ))/4
z = (pnt( p1 , 2 ) + pnt( p2 , 2 ) _
+ pnt( p3 , 2 ) + pnt( p4 , 2 ))/4
call punt 255 , x , y , z
call d3 255 , p1 , p2 , clr
call d3 255 , p2 , p3 , clr
call d3 255 , p3 , p4 , clr
call d3 255 , p4 , p1 , clr
end sub
sub box mx , my , mz , dx , dy , dz
''fil boinding box
boxx = mx
boxy = my
boxz = mz
boxdx = dx
boxdy = dy
boxdz = dz
end sub
sub cube l , d , f , r , u , b
''create a cube-like shape whit 6 colors
''first fil the swarm
''use boindingbox for dimensions
call punt 0 , boxx - boxdx , boxy - boxdy , boxz - boxdz
call punt 1 , boxx - boxdx , boxy - boxdy , boxz + boxdz
call punt 2 , boxx - boxdx , boxy + boxdy , boxz - boxdz
call punt 3 , boxx - boxdx , boxy + boxdy , boxz + boxdz
call punt 4 , boxx + boxdx , boxy - boxdy , boxz - boxdz
call punt 5 , boxx + boxdx , boxy - boxdy , boxz + boxdz
call punt 6 , boxx + boxdx , boxy + boxdy , boxz - boxdz
call punt 7 , boxx + boxdx , boxy + boxdy , boxz + boxdz
''then use swarm for quadangles
call d4 0 , 1 , 3 , 2 , l
call d4 7 , 6 , 4 , 5 , r
call d4 0 , 2 , 6 , 4 , f
call d4 7 , 5 , 1 , 3 , b
```

```
call d4 0 , 1 , 5 , 4 , d
call d4 7 , 6 , 2 , 3 , u
end sub
sub draw
''first sort triangles on z of zpoint
for h = 1 to tritel
for l = 0 to h - 1
if h3( ry( h ) , ozz ) _
< h3( ry( l ) , ozz ) then
q = ry( h )
ry( h ) = ry( l )
ry( l ) = q
end if
next l
next h
''calcutate screen coordinates
''the + 2000 is there to not do x/0
for i = 0 to tritel
x1 = winx / 2 + h3( ry( i ) , ox1 ) _
/ ( h3( ry( i ) , oz1 ) + 2000 ) * 2000
y1 = winy / 2 - h3( ry( i ) , oy1 ) _
/ ( h3( ry( i ) , oz1 ) + 2000 ) * 2000
x2 = winx / 2 + h3( ry( i ) , ox2 ) _
/ ( h3( ry( i ) , oz2 ) + 2000 ) * 2000
y2 = winy / 2 - h3( ry( i ) , ox2 ) _
/ ( h3( ry( i ) , oz2 ) + 2000 ) * 2000
x3 = winx / 2 + h3( ry( i ) , ox3 ) _
/ ( h3( ry( i ) , oz3 ) + 2000 ) * 2000
y3 = winy / 2 - h3( ry( i ) , ox3 ) _
/ ( h3( ry( i ) , oz3 ) + 2000 ) * 2000
clr = h3( ry( i ) , oclr )
''shading of triangle if clr = positif
if clr > 0 then
angle = gamma( ry( i ) , 0,1,0 )
clr = mix( clr _
, angle / pi / 2 + .5 , black )
else
clr = abs( clr )
end if
call tri x1 , y1 , x2 , y2 , x3 , y3 , clr
next i
end sub
sub tri x1 , y1 , x2 , y2 , x3 , y3 , clr
''draw a triangle
''first split color
r = clr and 255
```

```
g = int( clr / 256 ) and 255
b = int( clr / 256 ^ 2 ) and 255
#m.g "color " ; r ; " " ; g ; " " ; b
#m.g "backcolor " ; r ; " " ; g ; " " ; b
''sort coordinates
''these 2 lines are there to not get x/0
if y1 = y2 then y1 = y1 - 1e-5
if y2 = y3 then y3 = y3 + 1e-5
if y1 > y3 then
call swap y1 , y3
call swap x1 , x3
end if
if y1 > y2 then
call swap y1 , y2
call swap x1 , x3
end if
if y2 > y3 then
call swap y2 , y3
call swap x2 , y3
end if
''draw lines from border to border
for i = y1 to y3
a = x1 + ( x3 - x1 ) * (i-y1) / ( y3 - y1 )
if i < y2 then
b = x1 + ( x2 - x1 ) * (i-y1) / ( y2 - y1 )
else
b = x2 + ( x3 - x2 ) * (i-y2) / ( y3 - y2 )
end if
#m.g "down"
#m.g "line " ; a ; " " ; i _
; " " ; b ; " " ; i
#m.g "up"
next i
end sub
function gamma( trino , lx , ly , lz )
''calculate angle triangle.normal - light
tx = h3( trino , oxn )
ty = h3( trino , oyn )
tz = h3( trino , ozn )
dx = lx - tx
dy = ly - ty
dz = lz - dz
l = sqr( lx ^ 2 + ly ^ 2 + lz ^ 2 )
t = sqr( tx ^ 2 + ty ^ 2 + tz ^ 2 )
d = sqr( dx ^ 2 + dy ^ 2 + dz ^ 2 )
''1e-10 is there to not get x/0
```

```
test = ( d * d - l * l - t * t ) _
/ ( -2 * l * t + 1e-10 )
if test < -1 then test = -1
if test > 1 then test = 1
gamma = acs( test )
end function
sub swap byref a , byref b
''can not be used on array's
h = a : a = b : b = h
end sub
[part_3D_engine]
sub rotate byref k , byref l , deg
''rotating any coordinates
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 camara x , y , z , pan , tilt , rol , zoom
cam( 0 ) = x
cam( 1 ) = y
cam( 2 ) = z
cam( 3 ) = pan
cam( 4 ) = tilt
cam( 5 ) = rol
cam( 6 ) = zoom
end sub
sub spot byref x , byref y , byref z
''shape coordinates to world coodinates
''rotate coordinates
call rotate x , z , cam( 3 ) ''pan
call rotate y , z , cam( 4 ) ''tilt
call rotate x , y , cam( 5 ) ''rol
''use zoom
x = x * cam( 6 )
y = y * cam( 6 )
z = z * cam( 6 )
end sub
''future planing
sub link no , x , y , z , xz , yz , xy , p
''create a joint
end sub
sub child no , x , y , z , lim , p
''create a animated joint
end sub
```

```
sub angle lim , axel , deg
''animate a joint
end sub
[part_color]
function rainbow( deg )
''360o = rainbow-like coloring
rainbow = rgb( _
sin( rad( deg ) ) * 127 + 128 _
, sin( rad( deg - 120 ) ) * 127 + 128 _
, sin( rad( deg + 120 ) ) * 127 + 128 )
end function
function rgb( r , g , b )
rgb = ( r and 255 ) _
+ ( g and 255 ) * 256 _
+ ( b and 255 ) * 256 * 256
end function
function mix( kl1 , f , kl2 )
''mix 2 colors
if f < 0 then f = 0
if f > 1 then f = 1
r1 = int( kl1 and 255 )
g1 = int( kl1 / 256 ) and 255
b1 = int( kl1 / 256 / 256 ) and 255
r2 = int( kl2 and 255 )
g2 = int( kl2 / 256 ) and 255
b2 = int( kl2 / 256 / 256 ) and 255
dr = r2 - r1
dg = g2 - g1
db = b2 - b1
dr = dr * f
dg = dg * f
db = db * f
r = r1 + dr
g = g1 + dg
b = b1 + db
mix = rgb( r and 255 , g and 255 , b and 255 )
end function
[part_the_rest]
function rad( deg )
rad = deg * pi / 180
end function
function pend( fase , amp )
''for natural joint animations
pend = sin( rad( fase ) ) * amp
end function
```