

```
'Draw Bool Sprites by Bluatigro
'version 20-may--2010
'bugfix Gordon and extra rotate sub (6 aug 2010)

''use of this code is free
''created sprites to
''as long as you mention this code and its creator in your credits
```

```
WindowWidth = DisplayWidth
WindowHeight = DisplayHeight
global height , pi , winx , winy , sprx , spry
global black , red , green , yellow
global blue , magenta , cyan , white
global pink , purple , gray , orange
global block , dot , dot6 , chaos , marble
block = 1
dot = 2
dot6 = 3
marble = 4
chaos = 5
    winy = WindowHeight
    winx = WindowWidth
    pi = atn( 1 ) * 4
    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 )
nomainwin
open "Draw Bool Sprite" for graphics as #m
#m "trapclose [quit]"
''the folowing sprite-draw-comands are there
''up = andcolor
''down = orcolor
''for a 'normal' sprite up = black
''----- 2D -----
''clear spritewidth , spriteheight
''drawline x1 , y1 , z2 , y2 , thick , down , up
''ellipse x , y , dx , dy , thick , down , up
```

```
'ellipsefil x , y , dx , dy , down , up
'box x1 , y1 , x2 , y2 , thick , down , up
'boxfil x1 , y1 , x2 , y2 , down , up
'the begin and end are in degrees
'and can be different settings for same results
'arc x , y , dx , dy , begin , end , thick , down , up
'pie x , y , dx , dy , begin , end , thick , down , up
'piefil x , y , dx , dy , begin , end , down , up
'blezier x1,y1 , x2,y2 , x3,y3 , x4,y4 , thick,down,up
'tri x1 , y1 , x2 , y2 , x3 , y3 , down , up
'd = down u = up
'tri2 x1,y1,d1,u1,x2,y2,d2,u2,x3,y3,d3,u3
'rotate byref k, byref l, deg
'----- 3D -----
'sphere x , y , z , d , k1
'q is in degrees and rotates the spheres Yas
'mat = { chaos , dot , dot6 , block , marble }
'sphere2 x , y , z , d , k11 , k12 , q,mat
'egg x1,y1,z1,d1 , x2,y2,z2,d2 , dm , k1
'save spr$
''

''===== BEGIN SPRITE DRAW CODE

'paste example code here
'or invent something yourself

''===== END SPRITE DRAW CODE
wait
function nr$( no , max )
nr$ = right$( "00000000" + str$( no ) , max )
end function
[quit]
close #m
end

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 save spr$
#m "getbmp bmp 0 0 " ; sprx ; " " ; spry * 2
```

```
    bmpsave "bmp", DefaultDir$ + "\BMP\" _
    + spr$ + ".bmp"
end sub
sub clear x , y
    #m "cls"
    #m "color white"
    #m "backcolor white"
    #m "goto 0 0"
    #m "down"
    #m "boxfilled " ; x ; " " ; y
    #m "up"
    #m "goto 0 " ; y
    #m "down"
    #m "color black"
    #m "backcolor black"
    #m "boxfilled " ; x ; " " ; 2 * y
    #m "up"
    sprx = x
    spry = y
end sub
sub tri x1 , y1 , x2 , y2 , x3 , y3 , down , up
if y1 = y2 then y1 = y1 - 1e-10
if y2 = y3 then y3 = y3 + 1e-10
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
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
call drawline a , i , b , i , 1 , down , up
next i
end sub
sub swap byref a , byref b
```

```
h = a : a = b : b = h
end sub
sub pie x , y , h , w , b , e , t , down , up
#m "size " ; t
call setcolor up
#m "goto " ; x ; " " ; y
#m "down"
#m "piefilled " ; w ; " " ; h ; " " ; b ; " " ; e
#m "up"
call setcolor down
#m "goto " ; x ; " " ; y + spry
#m "down"
#m "piefilled " ; w ; " " ; h ; " " ; b ; " " ; e
#m "up"
end sub
sub piefil x , y , h , w , b , e , down , up
call setcolor up
#m "goto " ; x ; " " ; y
#m "down"
#m "piefilled " ; w ; " " ; h ; " " ; b ; " " ; e
#m "up"
call setcolor down
#m "goto " ; x ; " " ; y + spry
#m "down"
#m "piefilled " ; w ; " " ; h ; " " ; b ; " " ; e
#m "up"
end sub
sub drawline x1 , y1 , x2 , y2 , thick , down , up
#m "size " ; thick
call setcolor up
#m "color black"
#m "down"
#m "line " ; x1 ; " " ; y1 ; " " ; x2 ; " " ; y2
#m "up"
call setcolor down
#m "down"
#m "line " ; x1 ; " " ; y1 + spry ; " " ; x2 ; " " ; y2 + spry
#m "up"
end sub
sub arc mx , my , dx , dy , b , e , t , down , up
call setcolor up
for i = b to e step 50 / ( dx + dy )
x = sin( rad( i ) ) * dx + mx
y = cos( rad( i ) ) * dy + my
call ellipsefil x , y , t , t , down , up
next i
```

```
end sub
sub setcolor kl
r = int( kl and 255 )
g = int( kl / 256 ) and 255
b = int( kl / 256 / 256 ) and 255
#m "backcolor " ; r ; " " ; g ; " " ; b
#m "color " ; r ; " " ; g ; " " ; b
end sub
function rad( deg )
rad = deg * pi / 180
end function
function rainbow( deg )
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 )
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
sub box x , y , w , h , t , down , up
#m "size " ; t
#m "goto " ; x ; " " ; y
call setcolor up
#m "down"
#m "box " ; w ; " " ; h
#m "up"
```

```
#m "goto " ; x ; " " ; y + spry
call setcolor down
#m "down"
#m "box " ; w ; " " ; h + spry
#m "up"
end sub
sub boxfil x , y , w , h , down , up
#m "goto " ; x ; " " ; y
call setcolor up
#m "down"
#m "boxfilled " ; w ; " " ; h
#m "up"
#m "goto " ; x ; " " ; y + spry
call setcolor down
#m "down"
#m "boxfilled " ; w ; " " ; h + spry
#m "up"
end sub
sub ellipse x , y , dx , dy , t , down , up
#m "size " ; t
#m "goto " ; x ; " " ; y
call setcolor up
#m "down"
#m "ellipse " ; dx ; " " ; dy
#m "up"
#m "goto " ; x ; " " ; y + spry
call setcolor down
#m "down"
#m "ellipse " ; dx ; " " ; dy
#m "up"
end sub
sub ellipsefil x , y , dx , dy , down , up
#m "goto " ; x ; " " ; y
call setcolor up
#m "down"
#m "ellipsefilled " ; dx ; " " ; dy
#m "up"
#m "goto " ; x ; " " ; y + spry
call setcolor down
#m "down"
#m "ellipsefilled " ; dx ; " " ; dy
#m "up"
end sub
sub bezier x1,y1 , x2,y2 , x3,y3 , x4,y4 , t,d,u
call setcolor kl
if ( abs( x1 - x2 ) <= 1 ) _
```

```
and ( abs( y1 - y2 ) <= 1 ) then
call drawline x1 , y1 , x2 , y2 , t , d , u
else
ax = ( x1 + x2 ) / 2
ay = ( y1 + y2 ) / 2
bx = ( x3 + x4 ) / 2
by = ( y3 + y4 ) / 2
cx = ( x3 + x2 ) / 2
cy = ( y3 + y2 ) / 2
a1x = ( ax + cx ) / 2
a1y = ( ay + cy ) / 2
b1x = ( bx + cx ) / 2
b1y = ( by + cy ) / 2
c1x = ( a1x + b1x ) / 2
c1y = ( a1y + b1y ) / 2
call bezier x1,y1 , ax,ay , a1x,a1y , c1x,c1y ,t,d,u
call bezier c1x,c1y , b1x,b1y , bx,by , x4,y4 ,t,d,u
end if
end sub
sub tri2 x1,y1,d1,u1,x2,y2,d2,u2,x3,y3,d3,u3
if d1=d2 and d2=d3 _
and u1=u2 and u2=u3 then
call tri x1,y1,x2,y2,x3,y3,d1,u1
end if
if y1 = y2 then y1 = y1 - 1e-10
if y2 = y3 then y3 = y3 + 1e-10
if y1 > y3 then
call swap y1 , y3
call swap x1 , x3
call swap d1 , d3
call swap u1 , u3
end if
if y1 > y2 then
call swap y1 , y2
call swap x1 , x2
call swap d1 , d2
call swap u1 , u2
end if
if y2 > y3 then
call swap y2 , y3
call swap x2 , x3
call swap d2 , d3
call swap u2 , u3
end if
for y = y1 to y3
a = x1 + ( x3 - x1 ) * (y-y1) / ( y3 - y1 )
```

```
da = mix( d1 , (y-y1) / (y3-y1) , d3 )
ua = mix( u1 , (y-y1) / (y3-y1) , u3 )
if y < y2 then
b = x1 + ( x2 - x1 ) * (y-y1) / ( y2 - y1 )
db = mix( d1 , (y-y1) / (y2-y1) , d2 )
ub = mix( u1 , (y-y1) / (y2-y1) , u2 )
else
b = x2 + ( x3 - x2 ) * (y-y2) / ( y3 - y2 )
db = mix( d2 , (y-y2) / (y3-y2) , d3 )
ub = mix( u2 , (y-y2) / (y3-y2) , u3 )
end if
t = 0
if a > b then
call swap a , b
call swap da , db
call swap ua , ub
end if
if a = b then b = b + 1
for x = a to b
d = mix( da , ( x - a ) / ( b - a ) , db )
u = mix( ua , ( x - a ) / ( b - a ) , ub )
call pixel x , y , d , u
next x
next y
end sub
sub pixel x , y , down , up
#m "goto " ; x ; " " ; y + spry
call setcolor down
#m "down"
#m "set " ; x ; " " ; y + spry
#m "up"
#m "rule " ; _R2_COPYPEN
#m "goto " ; x ; " " ; y
call setcolor up
#m "down"
#m "set " ; x ; " " ; y
#m "up"
end sub
sub pxls no , kl$
for i = 1 to len( kl$ )
q$ = mid$( kl$ , i , 1 )
select case q$
case "B"
call pixel i , no , black , black
case "R"
call pixel i , no , red , black
```

```
case "G"
call pixel i , no , green , black
case "Y"
call pixel i , no , yellow , black
case "b"
call pixel i , no , blue , black
case "M"
call pixel i , no , magenta , black
case "C"
call pixel i , no , cyan , black
case "W"
call pixel i , no , white , black
case "o"
call pixel i , no , orange , black
case "p"
call pixel i , no , pink , black
case "g"
call pixel i , no , gray , black
case "P"
call pixel i , no , purple , black
case else ''transparent
call pixel i , no , black , white
end select
next i
end sub
sub sphere x , y , z , d , k1
#m "rule " ; _R2_COPYPEN
if abs( height - y ) < d then
dd = sqr( d ^ 2 - ( height - y ) ^ 2 + .001 ) * 2
kl1 = mix( k1 , .5 - ( height - y ) / d / 2 , black )
call ellipsefil x + sprx / 2 _
, spry / 2 - height - z / 4 _
, dd , dd / 4 , kl1 , black
end if
end sub
sub sphere2 x , y , z , d , kl1 , kl2 , angle , mat
if kl1 = kl2 then
call sphere x , y , z , d , kl1
exit sub
end if
if mat < 0 then
#m "rule " ; _R2_MERGE PEN
else
#m "rule " ; _R2_COPYPEN
end if
mat = abs( mat )
```

```
if abs( height - y ) < d then
dd = sqr( d ^ 2 - ( height - y ) ^ 2 + .001 )
kl1a = mix( kl1 , .5 - ( height - y ) / d / 2 , black )
kl2a = mix( kl2 , .5 - ( height - y ) / d / 2 , black )
for i = 0 to pi * 2 step pi / dd / 4
px = sin( i ) * dd
py = height - z
pz = cos( i ) * dd
qx = cos( rad( angle ) ) * px - sin( rad( angle ) ) * pz
qz = sin( rad( angle ) ) * px + cos( rad( angle ) ) * pz
select case mat
case block
if py < 0 _
xor qx < 0 _
xor qz < 0 then
klb = kl1a
else
klb = kl2a
end if
case marble
q = sin( ( qx - py ) / 11 ) _
+ sin( ( qx - qz ) * 2 / 11 ) _
+ sin( ( py - qz ) * 3 / 11 ) _
+ sin( ( py - qx ) * 5 / 11 ) _
+ sin( ( qz - py ) * 7 / 11 ) _
+ sin( ( qz - px ) * 11 / 11 )
if q < 0 then
klb = kl1a
else
klb = kl2a
end if
case dot
if qz / d > 0-.8 then
klb = kl2a
else
klb = kl1a
end if
case dot6
if abs( qz / d ) > .8 _
or abs( py / d ) > .8 _
or abs( px / d ) > .8 then
klb = kl1a
else
klb = kl2a
end if
case else
```

```
if rnd( 0 ) < .5 then
klb = kl1a
else
klb = kl2a
end if
end select
call pixel x + px + sprx / 2 _
, spry / 2 - height - ( z + pz ) / 4 _
, klb , black
next i
end if
end sub
sub egg x1 , y1 , z1 , d1 , x2 , y2 , z2 , d2 , dm , kl
af = sqr( ( x1 - x2 ) ^ 2 + ( y1 - y2 ) ^ 2 + ( z1 - z2 ) ^ 2 )
dx = ( x2 - x1 ) / af
dy = ( y2 - y1 ) / af
dz = ( z2 - z1 ) / af
dd = ( d2 - d1 ) / af
dh = ( d1 + d2 ) / 2
for i = 0 to af
call sphere x1 + dx * i , y1 + dy * i , z1 + dz * i _
, d1 + dd * i + sin( i * pi / af ) * ( dm - dh ) , kl
next i
end sub
```