

Draw Sprites Yourself

- [bluatigro](#) Mar 29, 2010

this is a program to draw your
sprites yourself
it does this whit comands

the folowing sprite-draw-comands are there
up = andcolor
down = orcolor
for a 'normal' sprite up = black
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 setiings for same results
arc x , y , dx , dy , begin , end , 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
pixel x , y , down , up
pxls no , kl\$
save spr\$

```
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
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 )
nocolor = -1
nomainwin
open "Boolean Sprites Draw" for graphics as #m
#m "trapclose [quit]"
call clear 64 , 64

for i = 0 to 360
call drawline 31 , 31 _
, sin( rad( i ) ) * 27 + 31 _
, cos( rad( i ) ) * 27 + 31 _
, 3 , rainbow( i ) , black
next i
wait
function nr$( no , max )
nr$ = right$( "00000000" + str$( no ) , max )
end function
[quit]
close #m
end
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 + 1
#m "down"
#m "color black"
#m "backcolor black"
#m "boxfilled " ; x ; " " ; 2 * y
#m "up"
sprx = x
spry = y
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 , 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
#m "down"
#m "set " ; x ; " " ; y
#m "up"
next i
```

```
call setcolor down
for i = b to e step 50 / ( dx + dy )
x = sin( rad( i ) ) * dx + mx
y = cos( rad( i ) ) * dy + my
#m "down"
#m "set " ; x ; " " ; y + spry
#m "up"
next i
end if
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 ellipse 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
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 , kl,thick
call bezier c1x,c1y , b1x,b1y , bx,by , x4,y4 , kl,thick
end if
end sub
sub pixel x , y , down , up
#m "goto " ; x ; " " ; y
call setcolor up
#m "down"
#m "set " ; x ; " " ; y
#m "up"
#m "goto " ; x ; " " ; y + spry
call setcolor down
#m "down"
#m "set " ; x ; " " ; y + spry
#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
end if
end sub
```