robodoc

J.v.Weert and F.N.C.Slothouber

COLLABORATORS			
	TITLE : robodoc		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	J.v.Weert and F.N.C.Slothouber	August 4, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

robodoc

1.1 safeclip.doc

1.2 Autodoc/safeclip.c

```
Autodoc/safeclip.c

NAME
    safeclip.c
```

```
FUNCTION
    Interface to system rendering routines, but with clipping.

AUTHOR
    Peter Knight: All programming.
    <pak@star.sr.bham.ac.uk>

CREATION DATE
    24-Mar-96

COPYRIGHT
    No restriction (Public Domain).
    Use these routines as you wish in your programs (a
    mention in the credits would be nice, but it's up to
    you).
```

1.3 safeclip.c/SafeAreaEnd

```
safeclip.c/SafeAreaEnd

NAME
   SafeAreaEnd

SYNOPSIS
        SafeAreaEnd (rp)
   VOID SafeAreaEnd (struct RastPort *)

FUNCTION
   Process buffer AreaDraw() instructions

INPUTS
   rp - Pointer to RastPort on which to draw.

RESULTS

SEE ALSO
   SafeAreaDraw


 (struct RastPort *rp)

  if (CLP_nvert)
    {
      WORD i;
      SafeAreaDraw (CLP_vert[0][0], CLP_vert[0][1]);  /* ensure shape is closed */
      CLP_nvert = Clip2d (CLP_nvert);
      AreaMove (rp, CLP_vert[0][0], CLP_vert[0][1]);
      for (i = 1; i < CLP_nvert; i++)
        AreaDraw (rp, CLP_vert[i][0], CLP_vert[i][1]);
      AreaEnd (rp);
      CLP_nvert = 0;
    }
```

1.4 safeclip.c/SafeClose

```
safeclip.c/SafeClose

NAME
   SafeClose

SYNOPSIS
        SafeClose ()
   VOID SafeClose (VOID)

FUNCTION
   Free memory allocated by SafeInit()

INPUTS

RESULTS

SEE ALSO
   SafeInit


 (VOID)

  if (CLP_wrk)
    {
      FreeVec (CLP_wrk);
      CLP_wrk = 0;
    }
  if (CLP_vert)
    {
      FreeVec (CLP_vert);
      CLP_vert = 0;
    }
```

1.5 safeclip.c/SafeDraw

```
safeclip.c/SafeDraw

NAME
   SafeDraw

SYNOPSIS
        SafeDraw (rp, x, y)
   VOID SafeDraw (struct RastPort *, LONG, LONG)

FUNCTION
   Move from previous point to new point and draw line.

INPUTS
   rp - Pointer to RastPort on which to draw.
   (x,y) - Coordinates of point to draw to
```

```
RESULTS

SEE ALSO
   SafeMove


 (struct RastPort *rp, LONG x, LONG y)

  CLP_vert[0][0] = CLP_lastx;
  CLP_vert[0][1] = CLP_lasty;
  CLP_vert[1][0] = x;
  CLP_vert[1][1] = y;
  if (ClipLine ())
    {
      Move (rp, CLP_vert[0][0], CLP_vert[0][1]);
      Draw (rp, CLP_vert[1][0], CLP_vert[1][1]);
    }
  CLP_lastx = x;
  CLP_lasty = y;
```

1.6 safeclip.c/SafeInit

```
safeclip.c/SafeInit

NAME
   SafeInit

SYNOPSIS
   res = SafeInit (nvertmax)
   ULONG SafeInit (ULONG)

FUNCTION
   Initialise clipping routines

INPUTS
   nvertmax - Maximum number of vertices per polygon that you will use

RESULTS
   res - FALSE if everything was OK, otherwise TRUE

SEE ALSO
   SafeClose


 (ULONG nvertmax)

  CLP_nvertmax = nvertmax;
  if (!(CLP_vert = (LONG (*)[2]) AllocVec (8 * CLP_nvertmax, MEMF_PUBLIC)))
    return 1;
  if (!(CLP_wrk = (LONG (*)[2]) AllocVec (8 * CLP_nvertmax, MEMF_PUBLIC)))
    {
      FreeVec (CLP_vert);
      CLP_vert = 0;
```

```
      return 2;
    }
  return 0;
```

1.7 safeclip.c/SafeRectFill

```
safeclip.c/SafeRectFill

NAME
   SafeRectFill

SYNOPSIS
        SafeRectFill (rp, x1, y1, x2, y2)
   VOID SafeRectFill (struct RastPort *, LONG, LONG, LONG, LONG)

FUNCTION
   Draw a filled rectangle

INPUTS
   rp - Pointer to RastPort on which to draw.
   (x1,y1) - Coordinates of upper left corner
   (x2,y2) - Coordinates of lower right corner

RESULTS

SEE ALSO


 (struct RastPort *rp, LONG x1, LONG y1, LONG x2, LONG y2)

  SafeAreaDraw (x1, y1);
  SafeAreaDraw (x2, y1);
  SafeAreaDraw (x2, y2);
  SafeAreaDraw (x1, y2);
  SafeAreaEnd (rp);
```

1.8 safeclip.h/SafeAreaDraw

```
safeclip.h/SafeAreaDraw

NAME
   SafeAreaDraw

SYNOPSIS
        SafeAreaDraw (x, y)
   VOID SafeAreaDraw (LONG, LONG)

FUNCTION
   Add a vertex to polygon vertex list.
```

```
INPUTS
   (x,y) - Coordinates of new vertex

RESULTS

SEE ALSO
   safeclip.c/SafeAreaEnd

 __inline VOID
 (LONG x, LONG y)

  extern LONG CLP_nvert, CLP_nvertmax, (*CLP_vert)[2];
  if (CLP_nvert < CLP_nvertmax)
    {
      CLP_vert[CLP_nvert][0] = x;
      CLP_vert[CLP_nvert][1] = y;
      CLP_nvert++;
    }
```

1.9 safeclip.h/SafeBltBitMapRastPort

```
safeclip.h/SafeBltBitMapRastPort

NAME
   SafeBltBitMapRastPort

SYNOPSIS
        SafeBltBitMapRastPort (srcbm, srcx, srcy,
                               destrp, destx, desty.
                               sizex, sizey, minterm)
   VOID SafeBltBitMapRastPort (struct BitMap *, LONG, LONG,
                               struct RastPort *, LONG, LONG,
                               LONG, LONG, UBYTE)

FUNCTION
   Blit a rectangle region from a BitMap to a RastPort.

INPUTS
   srcbm - Pointer to source BitMap
   (srcx,srcy) - Coordinates of upper left of source rectangle.
   destrp - Pointer to destination RastPort
   (destx,desty) - Coordinates of upper left of destination rectangle.
   sizex, sizey - Size of source rectangle
   minterm - Minterm for blitter to use during copy

RESULTS

SEE ALSO
   SafeBltMaskBitMapRastPort

 __inline VOID
 (struct BitMap *srcbm, LONG srcx, LONG srcy,
                        struct RastPort *destrp, LONG destx, LONG desty,
                        LONG sizex, LONG sizey, UBYTE minterm)
```

```
  extern LONG CLP_xmin, CLP_xmax, CLP_ymin, CLP_ymax;
  LONG xlo = destx, xhi = destx + sizex - 1;
  LONG ylo = desty, yhi = desty + sizey - 1;
  if (xlo < CLP_xmin) xlo = CLP_xmin;
  if (ylo < CLP_ymin) ylo = CLP_ymin;
  if (xhi > CLP_xmax) xhi = CLP_xmax;
  if (yhi > CLP_ymax) yhi = CLP_ymax;
  if (xlo <= xhi && ylo <= yhi &&
      xlo >= CLP_xmin && xhi <= CLP_xmax &&
      ylo >= CLP_ymin && yhi <= CLP_ymax)
    {
      BltBitMapRastPort (srcbm, srcx + (xlo - destx), srcy + (ylo - desty),
                         destrp, xlo, ylo,
                         (xhi - xlo) + 1, (yhi - ylo) + 1,
                         minterm);
    }
```

1.10 safeclip.h/SafeBltMaskBitMapRastPort

```
safeclip.h/SafeBltMaskBitMapRastPort

NAME
   SafeBltMaskBitMapRastPort

SYNOPSIS
        SafeBltMaskBitMapRastPort (srcbm, srcx, srcy,
                                   destrp, destx, desty.
                                   sizex, sizey, minterm, bltmask)
   VOID SafeBltMaskBitMapRastPort (struct BitMap *, LONG, LONG,
                                   struct RastPort *, LONG, LONG,
                                   LONG, LONG, UBYTE, PLANEPTR)

FUNCTION
   Blit a rectangle region from a BitMap to a RastPort through
   a single plane mask.

INPUTS
   srcbm - Pointer to source BitMap
   (srcx,srcy) - Coordinates of upper left of source rectangle.
   destrp - Pointer to destination RastPort
   (destx,desty) - Coordinates of upper left of destination rectangle.
   sizex, sizey - Size of source rectangle
   minterm - Minterm for blitter to use during copy
   bltmask - Pointer to single plane mask

RESULTS

SEE ALSO
   SafeBltBitMapRastPort

 __inline VOID
 (struct BitMap *srcbm, LONG srcx, LONG srcy,
```

```
                            struct RastPort *destrp, LONG destx, LONG desty,
                            LONG sizex, LONG sizey, UBYTE minterm, PLANEPTR bltmask ↩
                                )

  extern LONG CLP_xmin, CLP_xmax, CLP_ymin, CLP_ymax;
  LONG xlo = destx, xhi = destx + sizex - 1, ylo = desty, yhi = desty + sizey - 1;
  if (xlo < CLP_xmin) xlo = CLP_xmin;
  if (ylo < CLP_ymin) ylo = CLP_ymin;
  if (xhi > CLP_xmax) xhi = CLP_xmax;
  if (yhi > CLP_ymax) yhi = CLP_ymax;
  if (xlo <= xhi && ylo <= yhi &&
      xlo >= CLP_xmin && xhi <= CLP_xmax &&
      ylo >= CLP_ymin && yhi <= CLP_ymax)
    {
      BltMaskBitMapRastPort (srcbm, srcx + (xlo - destx), srcy + (ylo - desty),
                             destrp, xlo, ylo,
                             (xhi - xlo) + 1, (yhi - ylo) + 1,
                             minterm, bltmask);
    }
```

1.11 safeclip.h/SafeMove

```
safeclip.h/SafeMove

NAME
   SafeMove

SYNOPSIS
         SafeMove (x, y)
   VOID SafeMove (LONG, LONG)

FUNCTION
   Move drawing pen to new position.

INPUTS
   (x,y) - Coordinates of new point.

RESULTS

SEE ALSO
   safeclip.c/SafeDraw

 __inline VOID
 (LONG x, LONG y)

  extern LONG CLP_lastx, CLP_lasty;
  CLP_lastx = x;
  CLP_lasty = y;
```

1.12 safeclip.h/SafeSetLimits

```
safeclip.h/SafeSetLimits

NAME
   SafeSetLimits

SYNOPSIS
        SafeSetLimits (x1, y1, x2, y2)
   VOID SafeSetLimits (LONG, LONG, LONG, LONG)

FUNCTION
   Set limits of clipping rectangle

INPUTS
   (x1,y1) - Coordinates of upper left of clipping rectangle
   (x2,y2) - Coordinates of lower right of clipping rectangle

RESULTS

SEE ALSO

 __inline VOID
 (LONG x1, LONG y1, LONG x2, LONG y2)

  extern LONG CLP_xmin, CLP_ymin, CLP_xmax, CLP_ymax;
  CLP_xmin = x1;
  CLP_ymin = y1;
  CLP_xmax = x2;
  CLP_ymax = y2;
```

1.13 safeclip.h/SafeSetRast

```
safeclip.h/SafeSetRast

NAME
   SafeSetRast

SYNOPSIS
        SafeSetRast (rp, pen)
   VOID SafeSetRast (struct RastPort *, UBYTE)

FUNCTION
   Set the entire clipping region to a certain colour.

INPUTS
   rp - Pointer to RastPort
   pen - Pen number to fill region with.

RESULTS

SEE ALSO
```

```
 __inline VOID
 (struct RastPort *rp, UBYTE pen)

  extern LONG CLP_xmin, CLP_ymin, CLP_xmax, CLP_ymax;
  UBYTE oldPen = rp->FgPen;   /* should use GetAPen() if v39+ */
  SetAPen (rp, pen);
  RectFill (rp, CLP_xmin, CLP_ymin, CLP_xmax, CLP_ymax);
  SetAPen (rp, oldPen);
```

1.14 safeclip.h/SafeWritePixel

```
safeclip.h/SafeWritePixel

NAME
   SafeWritePixel

SYNOPSIS
        SafeWritePixel (rp, x, y)
   VOID SafeWritePixel (struct RastPort *, LONG, LONG)

FUNCTION
   Set the colour of an individual pixel.

INPUTS
   rp - Pointer to RastPort
   (x,y) - Coordinates of pixel

RESULTS

SEE ALSO

 __inline VOID
 (struct RastPort *rp, LONG x, LONG y)

  extern LONG CLP_xmin, CLP_xmax, CLP_ymin, CLP_ymax;
  if (x >= CLP_xmin && x <= CLP_xmax && y >= CLP_ymin && y <= CLP_ymax)
    WritePixel (rp, x, y);
```