29 VB Controls

"People with understanding want more knowledge."

Using graphics with BGI, we can create VB like controls: Forms, textboxes, command buttons etc. In this chapter let us see how to create few VB like controls.

29.1 Paintbrush

The following program is a Demo Paintbrush program. This program uses: command buttons, Windows and Frame. Paintbrush coders usually find difficulty in implementing mouse drawings. Here, I give you few guidelines.

29.1.1 Restricting Mouse Pointer

When the mouse is clicked on the drawing area, you must restrict it so that outside of the drawing should not be affected.

29.1.2 Hiding/Showing Mouse Pointer

You must properly hide/show mouse pointer. When you want to paint on the drawing box using `putpixel( )` or anything else, first of all hide the pointer, paint (using `putpixel( )`) and then do not forget to 'show' mouse pointer! I could see, even the commercial software—Adobe's *Instant Artist* fails to use this logic! So the logic is *hide-paint-show*.

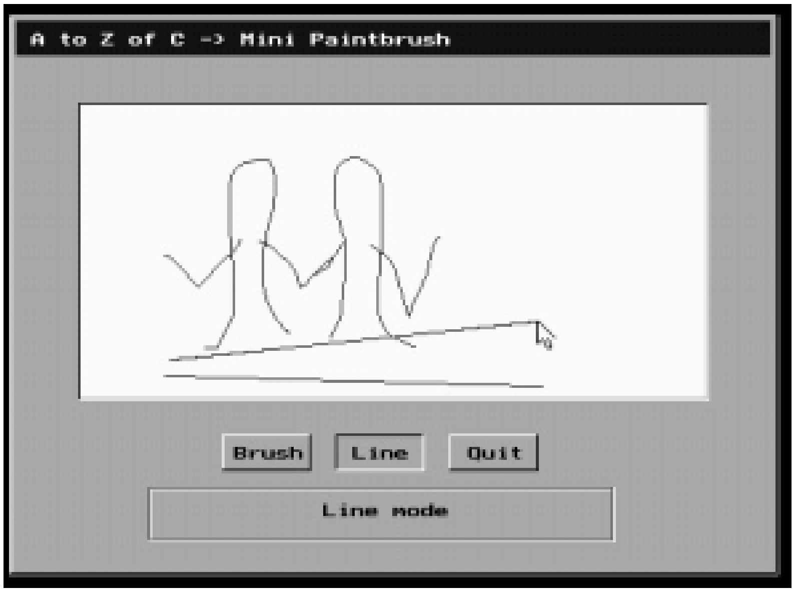
29.1.3 Avoiding Flickering of Mouse Pointer

When you would hide and show the pointer repeatedly, it usually starts flickering. So use '*hide-paint-show*' logic, only when the current mouse position is not equal to previous mouse position. If the current mouse position is equal to previous mouse position, don't do anything!

29.1.4 Using `setwritemode( )` function

When you draw line with the so called 'rubber-band technique', you may find that the existing images will get erased. We can avoid such 'erasing' with `setwritemode(XOR_PUT)`. As we know XOR is used for 'toggling', we can utilize it to avoid 'erasing'.

Figure shows the use of VB like controls in Paintbrush program



```
/*-----------------------------------------------------------------------
            Mini Paintbrush for VB Controls demo
 *---
 */

#include <dos.h>
#include <graphics.h>
#include "mouselib.h"

#define ESC       (27)
#define ISDRAWBOX(x, y)       ( x>141 && x<498 && y>131 && y<298 )

typedef int BOOLEAN;

#define FALSE     (0)
#define TRUE      (1)
#define PRESS     (0)
#define NORMAL    (1)

#define MAXCMDBUTTON    (3)
#define BRUSH           (0)
#define LINE            (1)
#define QUIT            (2)
```

```
struct RecButtonCoord
  {
     int x1;
     int y1;
     int x2;
     int y2;
  };

struct RecButtonCoord RecBut_Cd[MAXCMDBUTTON];

void far MyOuttextxy( int x, int y, char far *str, int color );
void MyRectangle( int x1, int y1, int x2, int y2, int upcolor, int
lowcolor );
void InitVB( void );
void InitScreen( void );
void VBForm( int x1, int y1, int x2, int y2, char *title );
void VBFrame( int x1, int y1, int x2, int y2 );
void VBDrawBox( int x1, int y1, int x2, int y2 );
void CmdButton( int cmdno, int status );
int CmdButtonVal( int x, int y );
void ShowStatus( int msgno );

/*-------------------------------------------------
      MyOttextxy - Prints text with
            specified color               */

void far MyOuttextxy( int x, int y, char far *str, int color )
{
   setcolor( color );
   outtextxy( x, y, str );
} /*--MyOuttextxy( )-----------*/

/*-------------------------------------------------
      MyRectangle - Rectangle with
            upcolor for Ú, lowcolor for Ù.
      It's for Command Button effect.           */

void MyRectangle( int x1, int y1, int x2, int y2, int upcolor, int
lowcolor )
{
   setcolor( upcolor );
   line( x1, y1, x2, y1 );
   line( x1, y1, x1, y2 );
   setcolor( lowcolor );
   line( x1, y2, x2, y2 );
   line( x2, y1, x2, y2);
} /*--MyRectangle( )-------------*/
```

```
/*-----------------------------------------------
      InitVB - Initializes VB.
            ie, Checks errors.            */

void InitVB( void )
{
   int gdriver = VGA, gmode = VGAHI, error;
   if ( !InitMouse( ) )
      {
       cprintf( "Mouse support needed! \r\n\a" );
       exit( 1 );
      }

   initgraph( &gdriver, &gmode, "c:\\tc\\bgi" );
   error = graphresult( );
   if ( error != grOk )
      {
      closegraph( );
      cprintf( "Graphics error: %s \r\n\a", grapherrormsg( error ) );
      exit( 1 );
      }
} /*--InitVB( )-------*/

/*-----------------------------------------------
      InitScreen - Initializes Screen.          */

void InitScreen( void )
{
   int i, x, y;

   VBForm( 100, 80, 540, 400, "A to Z of C -> Mini Paintbrush" );
   VBFrame( 180, 350, 445, 380 );
   VBDrawBox( 140, 130, 500, 300 );

   for( i= 0, x = 222, y = 320 ; i < 3 ; x += 65, ++i )
     {
      RecBut_Cd[i].x1 = x;
      RecBut_Cd[i].y1 = y;
      RecBut_Cd[i].x2 = x + 50;
      RecBut_Cd[i].y2 = y + 20;
      CmdButton( i, NORMAL );
     }
   /* Labels for Command Button... */
   MyOuttextxy( 229, 327, "Brush", BLACK );
   MyOuttextxy( 297, 327, "Line", BLACK );
   MyOuttextxy( 363, 327, "Quit", BLACK );
} /*--InitScreen( )------*/
```

```
/*-----------------------------------------------
      VBForm - Creates a Window with the given title.       */

void VBForm( int x1, int y1, int x2, int y2, char *title )
{
   setfillstyle( SOLID_FILL, LIGHTGRAY );
   bar( x1, y1, x2, y2 );
   setfillstyle( SOLID_FILL, BLUE );
   bar( x1+4, y1+3, x2-5, y1+22 );
   MyOuttextxy( x1+13, y1+10, title, WHITE );
   MyRectangle( x1+1, y1, x2-1, y2-1, WHITE, BLACK );
} /*--VBForm( )-----------*/

/*-----------------------------------------------
      VBFrame - Creates VB like Frame.    */

void VBFrame( int x1, int y1, int x2, int y2 )
{
   MyRectangle( x1+1, y1+1, x2, y2, WHITE, DARKGRAY );
   MyRectangle( x1, y1, x2+1, y2+1, DARKGRAY, WHITE );
} /*--VBFrame( )--------------*/

/*-----------------------------------------------
      VBDrawBox - Creates Drawing Box.    */

void VBDrawBox( int x1, int y1, int x2, int y2 )
{
   setfillstyle( SOLID_FILL, WHITE );
   bar( x1+1, y1+1, x2-2, y2-2 );
   MyRectangle( x1, y1, x2, y2, BLACK, WHITE);
} /*--VBDrawBox( )--------*/

/*-----------------------------------------------
      CmdButton - Draws Command Button for
            specified status.
            status are NORMAL, PRESS      */

void CmdButton( int cmdno, int status )
{
   if ( status==NORMAL )
      MyRectangle( RecBut_Cd[cmdno].x1, RecBut_Cd[cmdno].y1,
                 RecBut_Cd[cmdno].x2, RecBut_Cd[cmdno].y2, WHITE, BLACK
);
     else
      MyRectangle( RecBut_Cd[cmdno].x1, RecBut_Cd[cmdno].y1,
               RecBut_Cd[cmdno].x2, RecBut_Cd[cmdno].y2, BLACK, WHITE );
} /*--CmdButton( )----------*/
```

```
/*-----------------------------------------------
      CmdButtonVal - Returns Command Button value.    */

int CmdButtonVal( int x, int y )
{
   BOOLEAN found = FALSE;
   int i;

   for( i= 0; !found && i < MAXCMDBUTTON ; ++i )
      found = ( x > RecBut_Cd[i].x1  &&  x < RecBut_Cd[i].x2
                  && y > RecBut_Cd[i].y1 && y < RecBut_Cd[i].y2);
   if ( found )
      --i;
   return( i );
} /*--CmdButtonVal( )----------*/

/*-----------------------------------------------
      ShowStatus - Display messages.            */

void ShowStatus( int msgno )
{
   char *message[] = {
                  "Brush mode",
                  "Line mode"
                };
   if ( msgno==0 || msgno==1 )
      {
       setfillstyle( SOLID_FILL, LIGHTGRAY );
       bar( 280, 360, 438, 370 );
       MyOuttextxy( 280, 360, message[msgno], BLACK );
      }
} /*--ShowStatus( )--------*/

/*-----------------------------------------------
      main - Main of VB             */

int main( void )
{
   int mx, my, x1, x2, y1, y2, mbutton, cmdno, prevcmdno=0;
   const int brushcolor = RED; /* choose default brush color */
   BOOLEAN stayin = TRUE;
   InitVB( );
   InitScreen( );

   CmdButton( BRUSH, PRESS );   /* Force <Brush> button to default */
   ShowStatus( BRUSH );
   ShowMousePtr( );
```

```
while( stayin )
  {
   /* if ESC is pressed, then quit! */
   if ( kbhit( ) )
         stayin = ( getch( )!=ESC );

   GetMousePos( &mbutton, &mx, &my );
   if ( mbutton==LFTCLICK )
       {
         cmdno = CmdButtonVal( mx, my );
         if ( cmdno!=MAXCMDBUTTON && cmdno != prevcmdno )
           {
               HideMousePtr( );
               CmdButton( cmdno, PRESS );
               CmdButton( prevcmdno, NORMAL );
               ShowStatus( cmdno );
               prevcmdno = cmdno;
               ShowMousePtr( );
               stayin = ( cmdno!=QUIT );
           }
         if ( ISDRAWBOX( mx, my ) )
           {
               RestrictMousePtr( 142, 132, 497, 297 );
               switch ( prevcmdno )
               {
                  case BRUSH:
                     x1 = mx;
                     y1 = my;
                     setcolor( brushcolor );
                     HideMousePtr( );
                     putpixel( mx, my, brushcolor );
                     ShowMousePtr( );
                     do
                       {
                          GetMousePos( &mbutton, &mx, &my );
                          if ( x1!=mx || y1!=my )
                           {
                              HideMousePtr( );
                              line( x1, y1, mx, my );
                              ShowMousePtr( );
                              x1 = mx;
                              y1 = my;
                           }
                       } while(mbutton==LFTCLICK);
                     break;
                  case LINE:
                     x2 = x1 = mx;
```

```
                              y2 = y1 = my;
                              /* Note! in XOR_PUT mode, you must
                                 setcolor to 'WHITE-brushcolor'
                              */
                              setwritemode( XOR_PUT );
                              setcolor( WHITE-brushcolor );
                              do
                                {
                                   GetMousePos( &mbutton, &mx, &my );
                                   if ( mx!=x2 || my!= y2 )
                                    {
                                       HideMousePtr( );
                                       line( x1, y1, x2, y2 );
                                       line( x1, y1, mx, my );
                                       ShowMousePtr( );
                                       x2 = mx;
                                       y2 = my;
                                    }
                                } while(mbutton==LFTCLICK);
                              setwritemode( COPY_PUT );
                              /* Note! in COPY_PUT mode, you must
                                 setcolor to 'brushcolor'
                              */
                              setcolor( brushcolor );
                              HideMousePtr( );
                              line( x1, y1, mx, my );
                              ShowMousePtr( );
                        }
                        RestrictMousePtr( 0, 0, 640, 480 );
                    }
                }
           }
   closegraph( );
   return( 0 );
} /*--main( )---------*/
```

29.2 Note

For mouse inputs, here I have used *request mode* and so it won't be much efficient. If you need more precision, use *event mode* to get mouse inputs.

A real VB control uses object-oriented concepts. So for the exact implementation, you have to go for C++.

Suggested Projects

1. Yet I haven't seen a full VB imitated controls library. If you could code all VB controls, you can even sell that library!