

30

"Plans fail without good advice."

Scribble

Scribble is a CHR file creator developed with graphics with BGI. It will be a good example of coding style, using mouse routines, graphics with BGI, library & project file creation and file format.

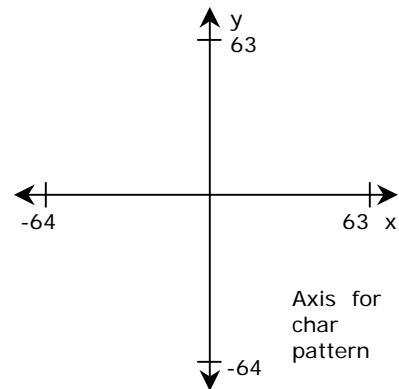
30.1 Prelude

CHR files are used for generating fonts in Turbo C's graphics programs. Except for default font, we need the corresponding CHR file to display respective fonts. For example, in order to display 'Gothic' fonts, we need GOTH.CHR file. Scribble is a CHR (or font) file creator. When I developed this utility, I thought that there is no utility to create CHR files. But later I came to know that Borland also provides 'Font Editor' to create CHR file. When you compare 'Scribble' and Borland's 'Font Editor', you can find that the mouse support in Borland's Font Editor is worse! When I developed Scribble, I thought that CHR file format is undocumented. And so I cracked the CHR file format. But later I came to know that it is documented. So my view about CHR file format may slightly differ from Borland's official documentation. I suggest you to have a glance at the CHR file format on file format collection.

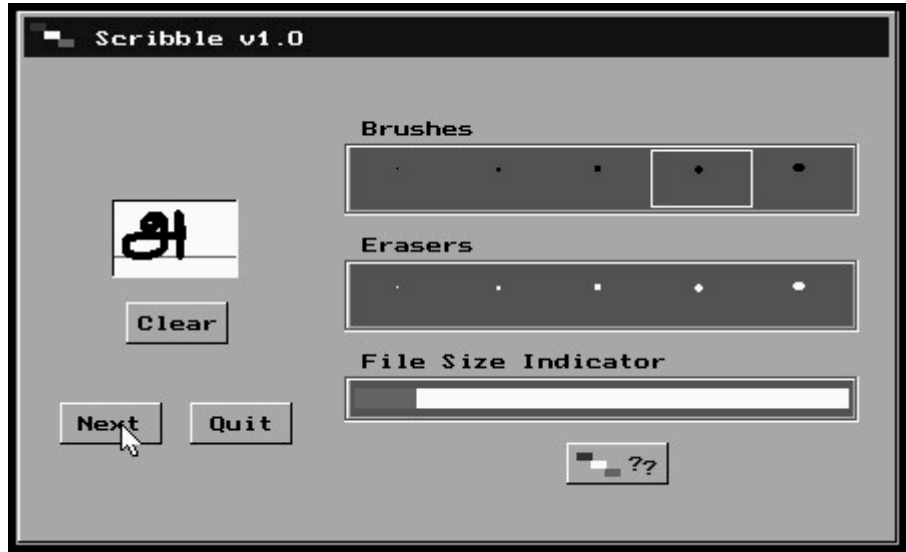
30.2 Storing Fonts

Borland's CHR file structure saves a character pattern as a set of lines with X, Y coordinates stored in corresponding bytes. The coordinate values are stored in 7-bits of a byte and they all are signed. So the existing values can be -64 to 63 for X and Y coordinates. The last bit (7th bit) of the X-Y values holds the command. The command can be any one of the following 3 commands: *Move/Scan character*, *Draw line from current location* or *End of character definition*.

You can see that the X values can even be in negative. But for the sake of brevity, I have avoided negative values in Scribble.



30.3 Scribble screenshots



30.4 Mouselib.lib

30.4.1 Mouselib.h

```
#ifndef __MOUSELIB_H
```

```

#define LFTCLICK (1)

int InitMouse( void );
void ShowMousePtr( void );
void MoveMousePtr( int x, int y );
void RestrictMousePtr( int x1, int y1, int x2, int y2 );
void HideMousePtr( void );
void GetMousePos( int *mbutton, int *x, int *y );
void ChangeMousePtr( int *shape );

#endif

```

30.4.2 Mouselib.c

```

#include "mouselib.h"

#pragma inline

/*-----
    InitMouse - Initializes Mouse.
                Returns 0 for success.          */

int InitMouse( void )
{
    asm {
        MOV AX, 0;
        INT 33h;
    }
    return;
} /*--InitMouse( )---*/

/*-----
    ShowMousePtr - Shows Mouse Pointer.          */

void ShowMousePtr( void )
{
    asm {
        MOV AX, 1h;
        INT 33h;
    }
} /*--ShowMousePtr( )----*/

/*-----
    HideMousePtr - Hide Mouse Pointer.          */

```

156 A to Z of C

```
void HideMousePtr( void )
{
    asm {
        MOV AX, 2h;
        INT 33h;
    }
} /*--HideMousePtr( )-----*/

/*-----
    MoveMousePtr - Move Mouse Pointer
                  to (x, y).                               */

void MoveMousePtr( int x, int y )
{
    asm {
        MOV AX, 4h;
        MOV CX, x;
        MOV DX, y;
        INT 33h;
    }
} /*--MoveMousePtr( )-----*/

/*-----
    RestrictMousePtr - Restrict Mouse Pointer
                     to the specified coordinates          */

void RestrictMousePtr( int x1, int y1, int x2, int y2 )
{
    asm {
        MOV AX, 7h;
        MOV CX, x1;
        MOV DX, x2;
        INT 33h;
        MOV AX, 8h;
        MOV CX, y1;
        MOV DX, y2;
        INT 33h;
    }
} /*--RestrictMousePtr( )-----*/

/*-----
    GetMousePos - Gets Mouse position & mouse button value. */

void GetMousePos( int *mbutton, int *mx, int *my )
{
    asm {
        MOV AX, 3h;
```

```

        INT 33h;

        MOV AX, BX;
        MOV BX, mbutton;
        MOV WORD PTR [BX], AX;

        MOV BX, mx;
        MOV WORD PTR [BX], CX;

        MOV BX, my;
        MOV WORD PTR [BX], DX;
    }
} /*--GetMousePos( )-----*/

```

30.4.3 Mouselib.lib

Using the above Mouselib.c file compile it to library file for Small memory model, you will get Mouselib.lib file. You can use the library – Mouselib.lib in your projects.

30.5 Scribble.h

```

/*-----
                                Scribble Declarations
                                scribble.h
*-----
*/

/* PC bios data area pointer to incrementing unsigned long int */
#define BIOSTICK (*(volatile unsigned long far *) (0x0040006CL))

typedef int BOOLEAN;

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

#define MAXCMBUTTON    (7)
#define CLEAR          (0)
#define NEXT           (1)
#define QUIT           (2)
#define ABOUT          (3)
#define OKBUTTON      (4)
#define NOBUTTON       (5)
#define YESBUTTON      (6)

```

158 A to Z of C

```
struct ButtonStatus
{
    int x1;
    int y1;
    int x2;
    int y2;
};
#define MAXBRUSH (10)
#define THANKS (1)
#define FSIZEERR (2)

typedef int WORD;
typedef char BYTE;

#define EOFCHAR1 (0)
#define EOFCHAR2 (0)
#define CHARSCAN1 (0)
#define CHARSCAN2 (1)
#define DRAWCHAR1 (1)
#define DRAWCHAR2 (1)

typedef struct tagFILEHEADER
{
    BYTE fId[4];
    BYTE copyRight[111];
    BYTE copyRightEnd;
    WORD headerOffset;
    BYTE fntName[4];
    WORD fntSize;
    BYTE fntVersion[4];
    BYTE fntHeader;
    WORD noOfChars;
    BYTE undefined1;
    BYTE startChar;
    WORD defOffset;
    BYTE fillFlag;
    BYTE dCapital;
    BYTE dBase;
    BYTE dBottomDescender;
    BYTE undefined2[5];
//    WORD charOffset[noOfChars];
//    BYTE widthTbl[noOfChars];
} FILEHEADER;

typedef struct tagFONTINFO
{
    unsigned int y : 7;
```

```

    unsigned int op2 : 1;
    unsigned int x   : 7;
    unsigned int op1 : 1;
} FONTINFO;

/* File header for Scribble */
FILEHEADER scriFh = {
    'P', 'K', 8, 8,
    "Scribble v1.0 for DOS ,2001 by R. Rajesh Jeba Anbiah, "
    "Web page: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx, "
    "Thank you Jesus! ",
    0x1A,
    128,          /* headerOffset */
    "????",      /* fName - To be changed */
    0,           /* fntSize - To be changed */
    1, 0, 1, 0,
    '+',
    1,           /* noOfChars - To be changed */
    0,          /* undefined ?? */
    ' ',        /* startChar */
    0,          /* defOffset - To be changed */
    0,
    25,
    0,
    -9,
    "          /* undefined ?? */
};

/* Store brushe types */
char *Pixel_Mask[16] =
{
    "11000000",
    "11000000",
    "00000000",
    "00000000",

    "11100000",
    "11100000",
    "11100000",
    "00000000",

    "01100000",
    "11110000",
    "11110000",
    "01100000",

    "01111000",

```

160 A to Z of C

```
        "11111100",
        "11111100",
        "01111000"
    };
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 PutPoint( int x, int y, int btype );
void ScribbleLine ( int x1, int y1, int x2, int y2, int btype );
void ScribbleInfo( void );
void InitScribble( void );
void GWindow( int x1, int y1, int x2, int y2, char *title );
void SetScreen( void );
void GetFontName( char *str );
void FileSizeIndicator( void );
void Clear( void );
void CmdButton( int cmdno, int status );
int CmdButtonVal( int x, int y );
void BrushBox( int brushno, int status );
int BrushVal( int x, int y );
void MsgWindow( char *fontname, int msgno );
int X4CenteredMsg( char *str );
void MakeFontProcedure1( void );
void MakeFontProcedure2( void );
void MakeFontProcedure3( void );
void CloseScribbleFiles( void );
```

30.6 Scribble.c

```
/*-----
                                Scribble
                                ( CHR file creator )
                                by
                                R. Rajesh Jeba Anbiah,

    File name: Scribble.c
    Written: March-April, 2001
    Copyright (c) 2001, R. Rajesh Jeba Anbiah
    All Rights Reserved.

*---
*/
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <time.h>
#include <alloc.h>
```



```

#include <dir.h>
#include <graphics.h>
#include "mouselib.h"
#include "scribble.h"

struct ButtonStatus But_Stat[MAXCMBUTTON], Brush_Stat[MAXBRUSH];

FONTINFO fInfo;
WORD charoffset;
BYTE charwidth;
FILE *chOffFp, *wthFp, *chInfoFp, *scriFp;

/*-----
   MyOuttextxy - 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 ↑, lowcolor for ↓.
                 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( )-----*/

/*-----
   PutPoint - Point with a specified
              pattern ( brush type ).
              Pattern is stored in *Pixel_Mask[]
              It's for Brush effect.
*/
void PutPoint( int x, int y, int btype )
{

```

162 A to Z of C

```
int i, j, color = getcolor( );
if ( btype == 0 )
    putpixel( x, y, color );
else
    for ( i = 0 ; i<4 ; ++i )
        for ( j = 0; j<8 ; ++j )
            if ( Pixel_Mask [4*(btype-1)+i][j] == '1' )
                putpixel( x+j-(btype)/2, y+i-(btype)/2, color );
} /*--PutPoint( )-----*/

/*-----
    ScribbleLine - Draws line a specified
                    pattern ( brush type ).
    Logic: Bresenham's Line Algorithm.
    It's for Brush effect.                                */

void ScribbleLine ( int x1, int y1, int x2, int y2, int btype )
{
    int x, y, dx, dy, p, incrx, incry;
    dx = abs(x2 - x1);
    dy = abs(y2 - y1);
    incrx = (x2 >= x1)? 1 : -1;
    incry = (y2 >= y1)? 1 : -1;

    PutPoint( x1, y1, btype );
    x = x1;
    y = y1;
    if (dx > dy)
    {
        p = 2 * dy - dx;
        while( x != x2 )
        {
            x += incrx;
            if (p < 0)
                p += 2 * dy;

            else
            {
                y += incry;
                p += 2 * (dy - dx);
            }

            PutPoint( x, y, btype );
        }
    }
    else
    {
```


164 A to Z of C

```
"||-----|| \r\n"
);
textcolor( LIGHTGREEN );
cprintf(
"||   For any  Suggestions  Bug report           || \r\n"
"||                                Sending donations           || \r\n"
"||   visit Scribble's official page:           || \r\n"
"||   xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx           || \r\n"
"||-----|| \r\n"
"||   Copyright (c) April 2001, R. Rajesh Jeba Anbiah   || \r\n"
"||   All Rights Reserved.                             || \r\n"
"||-----|| \r\n"
);
textcolor( LIGHTBLUE+BLINK );
cprintf(
"||                               Press any Key...           || \r\n"
);
window( 31, 2, 65, 5 );
textcolor( WHITE );
cprintf( "■" );
textcolor( GREEN );
cprintf( "\r\n ■" );
getch( );
window( 10, 2, 75, 25 );
textcolor( GREEN );
_setcursortype( _NORMALCURSOR );
} /*--ScribbleInfo( )-----*/

/*-----
   InitScribble - Initializes Scribble.
                   ie, Checks errors.           */

void InitScribble( void )
{
  int gdriver = VGA, gmode = VGAHI, error = 0;
  registerfarbgidriver( EGAVGA_driver_far );
  if ( !InitMouse( ) )
  {
    cprintf( "Mouse support needed! \r\n\a" );
    error = 1;
  }

  if ( ( chOffFp = fopen( "~$scribl.raj", "wb+" ) ) == NULL )
  {
    cprintf( "Fatal Error(01): File cannot be created \r\n\a" );
    error |= 2;
  }
}
```

```

if ( ( wthFp = fopen( "~$scrib2.raj", "wb+" ) ) == NULL )
{
    fprintf( "Fatal Error(02): File cannot be created \r\n\a" );
    error |= 3;
}
if ( ( chInfoFp = fopen( "~$scrib3.raj", "wb+" ) ) == NULL )
{
    fprintf( "Fatal Error(03): File cannot be created \r\n\a" );
    error |= 4;
}
if ( error )
{
    CloseScribbleFiles( );
    exit( 1 );
}
initgraph( &gdriver, &gmode, "" );
error = graphresult( );
if ( error != grOk )
{
    CloseScribbleFiles( );
    closegraph( );
    fprintf( "Graphics error: %s \r\n\a", grapherrormsg( error ) );
    exit( 1 );
}
} /*--InitScribble( )-----*/

/*-----
    GWindow - Creates a Window with the given title.          */

void GWindow( 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 );
} /*--GWindow( )-----*/

/*-----
    SetScreen - Initializes Screen.                            */

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

    GWindow( 100, 125, 540, 390, "" );

```

166 A to Z of C

```
MyOuttextxy( 140, 135, "Scribble v1.0", WHITE );
/* Icons... */
MyOuttextxy( 107, 131, "■", RED );
MyOuttextxy( 114, 135, "■", WHITE );
MyOuttextxy( 121, 139, "■", GREEN );

MyRectangle( 148, 219, 210, 257, BLACK, WHITE);
Clear( );
settextstyle( DEFAULT_FONT, HORIZ_DIR, 4 );
MyOuttextxy( 150, 225, "!", BLACK ); /* starting character */
settextstyle( DEFAULT_FONT, HORIZ_DIR, 1 );

MyRectangle( 265, 192, 519, 225, WHITE, DARKGRAY );
MyRectangle( 264, 191, 520, 226, DARKGRAY, WHITE );
setfillstyle( SOLID_FILL, DARKGRAY );
bar( 267, 193, 517, 223 );
MyOuttextxy( 273, 180, "Brushes", BLACK );

MyRectangle( 265, 250, 519, 283, WHITE, DARKGRAY );
MyRectangle( 264, 249, 520, 284, DARKGRAY, WHITE );
bar( 267, 251, 517, 281 );
MyOuttextxy( 273, 238, "Erasers", BLACK );

MyRectangle( 265, 308, 519, 328, WHITE, DARKGRAY );
MyRectangle( 264, 307, 520, 329, DARKGRAY, WHITE );
bar( 267, 309, 517, 326 );
setfillstyle( SOLID_FILL, WHITE );
bar( 269, 313, 515, 322 );
MyOuttextxy( 273, 296, "File Size Indicator", BLACK );

for( i= 0, x = 267, y = 194 ; i < MAXBRUSH ; x += 50, ++i )
{
    Brush_Stat[i].x1 = x;
    Brush_Stat[i].y1 = y;
    Brush_Stat[i].x2 = x + 50;
    Brush_Stat[i].y2 = y + 28;
    if ( i==0 )
        BrushBox( i, PRESS );

    if ( i == MAXBRUSH/2-1 )
    {
        y = 252;
        x = 267-50;
    }
}

setcolor( BLACK );
```

```

for ( i=0, x=290 ; i<5 ; x += 50, ++i )
    PutPoint( x, 203, i );

setcolor( WHITE );
for ( i=0, x=290 ; i<5 ; x += 50, ++i )
    PutPoint( x, 262, i );

But_Stat[0].x1 = 155;
But_Stat[0].y1 = 270;
But_Stat[0].x2 = 205;
But_Stat[0].y2 = 290;
CmdButton( 0, NORMAL );

for( i= 1, x = 122, y = 320 ; i < 3 ; x += 65, ++i )
{
    But_Stat[i].x1 = x;
    But_Stat[i].y1 = y;
    But_Stat[i].x2 = x + 50;
    But_Stat[i].y2 = y + 20;
    CmdButton( i, NORMAL );
}

But_Stat[3].x1 = 375;
But_Stat[3].y1 = 340;
But_Stat[3].x2 = 425;
But_Stat[3].y2 = 360;
CmdButton( 3, NORMAL );

But_Stat[4].x1 = 290;
But_Stat[4].y1 = 335;
But_Stat[4].x2 = 340;
But_Stat[4].y2 = 355;

But_Stat[5].x1 = 270;
But_Stat[5].y1 = 270;
But_Stat[5].x2 = 320;
But_Stat[5].y2 = 290;

But_Stat[6].x1 = 330;
But_Stat[6].y1 = 270;
But_Stat[6].x2 = 380;
But_Stat[6].y2 = 290;

MyOuttextxy( 161, 277, "Clear", BLACK );
MyOuttextxy( 131, 327, "Next", BLACK );
MyOuttextxy( 197, 327, "Quit", BLACK );

```

168 A to Z of C

```
MyOuttextxy( 380, 345, "■", RED );
MyOuttextxy( 387, 349, "■", WHITE );
MyOuttextxy( 394, 353, "■", GREEN );

MyOuttextxy( 406, 347, "?", BLUE );
MyOuttextxy( 413, 349, "?", BLUE );
} /*--SetScreen( )-----*/

/*-----
   GetFontName - Gets Font Name &      checks
                 whether the file is already exist or not.
                 If exist, it prompt with Warning */

void GetFontName( char *str )
{
    int mx, my, mbutton, cmdno, prevcmdno, i,
        x = 382, y = 250, len = 4, cursorcolor = BLACK;
    unsigned int imgsize;
    volatile unsigned long nexttick = BIOSTICK;
    char cursor[2] = "|", ch[2] = " ", filename[10], tmpmsg[40];
    BOOLEAN stayin = TRUE;
    void far *buffer;
    struct ffbblk ffbblk;

    imgsize = imagesize( 150, 155, 490, 370 );
    if ((buffer = farmalloc(imgsize)) == NULL)
    {
        CloseScribbleFiles( );
        closegraph( );
        fprintf( "\r\nError: Not enough memory!\r\n\n\a" );
        exit(1);
    }
    getimage( 190, 200, 450, 300, buffer );

    while( stayin )
    {
        GWindow( 190, 200, 450, 300, "Happy Scribbling!" );
        MyOuttextxy( 213, 249, "Enter the Font Name", BLACK );
        MyOuttextxy( 213, 269, "( 4 Characters )", BLACK );
        setfillstyle( SOLID_FILL, WHITE );
        bar( 375, 245, 420, 260 );
        MyRectangle( 375, 245, 420, 260, BLACK, WHITE);
        i = 0;
        while( i<len )
        {
            if ( BIOSTICK > nexttick )
```



```

{
    MyOuttextxy( x, y, cursor, cursorcolor );
    cursorcolor ^= ( BLACK ^ WHITE );
    nexttick = BIOSTICK + 7L;
}
if ( kbhit( ) )
{
    MyOuttextxy( x, y, cursor, WHITE );
    ch[0] = toupper( getch( ) );
    if ( ch[0]==0 ) /* Ignore special characters */
        getch( );
    if ( i!=0 && ch[0]=='\b' )
    {
        ch[0] = str[--i];
        x -= textwidth( cursor );
        MyOuttextxy( x, y, ch, WHITE );
    }
    else if ( ch[0]!=' ' && ch[0]!='*' && ch[0]!='+'
        && ch[0]!='=' && ch[0]!='[' && ch[0]!=']'
        && ch[0]!='|' && ch[0]!='\\' && ch[0]!='\"'
        && ch[0]!=':' && ch[0]!=';' && ch[0]!='<'
        && ch[0]!=',' && ch[0]!='>' && ch[0]!='.'
        && ch[0]!='?' && ch[0]!='/'
        && !(iscntrl(ch[0])) )
    {
        str[i++] = ch[0];
        MyOuttextxy( x, y, ch, BLACK );
        x += textwidth( cursor );
    }
}
str[i] = '\0';
strcpy( filename, str );
strcat( filename, ".CHR" );
if ( findfirst( filename, &fblk, 0 ) == 0 ) /* File already
exist! */
{
    GWindow( 190, 200, 450, 300, "Warning!" );
    strcpy( tmpmsg, filename );
    strcat( tmpmsg, " already exist!" );
    MyOuttextxy( 213, 234, tmpmsg, RED );
    MyOuttextxy( 213, 248, "Overwrite existing file?", BLACK );
    CmdButton( NOBUTTON, NORMAL );
    CmdButton( YESBUTTON, NORMAL );
    MyOuttextxy( 289, 277, "No", BLACK );
    MyOuttextxy( 343, 277, "Yes", BLACK );
    x -= len * textwidth( cursor );
}

```

170 A to Z of C

```
ShowMousePtr( );
do
{
    cmdno = 0;
    GetMousePos( &mbutton, &mx, &my );
    if ( mbutton==LFTCLICK )
    {
        cmdno = CmdButtonVal( mx, my );
        if ( cmdno==NOBUTTON || cmdno==YESBUTTON )
        {
            HideMousePtr( );
            CmdButton( cmdno, PRESS );
            ShowMousePtr( );
            prevcmdno = cmdno;
            do
            {
                GetMousePos( &mbutton, &mx, &my );
                cmdno = CmdButtonVal( mx, my );
            } while( mbutton==LFTCLICK&&cmdno==prevcmdno);
            HideMousePtr( );
            CmdButton( prevcmdno, NORMAL );
            ShowMousePtr( );
        }
    }
    } while( cmdno!=NOBUTTON && cmdno!=YESBUTTON );
    stayin = ( cmdno==NOBUTTON );
    HideMousePtr( );
}
else
    stayin = FALSE;
}
for ( i=0; i<len ; ++i )
    scriFh.fntName[i] = str[i];
putimage( 190, 200, buffer, COPY_PUT );
farfree( buffer );
} /*--GetFontName( )-----*/

/*-----
    FileSizeIndicator - Indicates the file
                        size limitation of 32KB      */
void FileSizeIndicator( void )
{
    int xmax = 269 + 0.007999 * (16 + 3*scriFh.noOfChars + ftell(
chInfoFp )) ;
    if ( xmax > 420 )
```

```

        setfillstyle( SOLID_FILL, RED );
    else
        setfillstyle( SOLID_FILL, GREEN );
    bar( 269, 313, xmax, 322 );

} /*--FileSizeIndicator( )-----*/

/*-----
    Clear - Clears the drawing box          */

void Clear( void )
{
    setfillstyle( SOLID_FILL, WHITE );
    bar( 149, 220, 209, 256 );
    setcolor( GREEN );
    line( 149, 247, 209, 247 );
} /*--Clear( )-----*/

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

void CmdButton( int cmdno, int status )
{
    if ( status==NORMAL )
        MyRectangle( But_Stat[cmdno].x1, But_Stat[cmdno].y1,
                    But_Stat[cmdno].x2, But_Stat[cmdno].y2, WHITE, BLACK );
    else
        MyRectangle( But_Stat[cmdno].x1, But_Stat[cmdno].y1,
                    But_Stat[cmdno].x2, But_Stat[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 > But_Stat[i].x1  &&  x < But_Stat[i].x2
                &&  y > But_Stat[i].y1 &&  y < But_Stat[i].y2);
    if ( found )

```

172 A to Z of C

```
        --i;
    return( i );
} /*--CmdButtonVal( )-----*/

/*-----
    BrushBox - Draws Brush Box for
               specified status.
               status are NORMAL, PRESS      */

void BrushBox( int brushno, int status )
{
    if ( status==NORMAL )
        setcolor( DARKGRAY );
    else
        setcolor( WHITE );
    rectangle( Brush_Stat[brushno].x1, Brush_Stat[brushno].y1,
              Brush_Stat[brushno].x2, Brush_Stat[brushno].y2 );
} /*--BrushBox( )-----*/

/*-----
    BrushVal - Returns Brush value.          */

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

    for( i= 0; !found && i < MAXBRUSH ; ++i )
        found = ( x > Brush_Stat[i].x1  &&  x < Brush_Stat[i].x2
                &&  y > Brush_Stat[i].y1 &&  y < Brush_Stat[i].y2);

    if ( found )
        --i;
    return( i );
} /*--BrushVal( )-----*/

/*-----
    MsgWindow - Prompts with messages "Thank you!",
               "Error!", "About...".      */

void MsgWindow( char *fontname, int msgno )
{
    int mx, my, mbutton, cmdno = 0, prevcmdno, xx;
    unsigned int imgsize;
    char *message[ ] = { " ", "Thank you!", "Error!", "About..." };
    char title[15], tmpmsg[40];
    void far *buffer;
```

```

strcpy( title, message[msgno] );
strcpy( tmpmsg, fontname );
strcat( tmpmsg, " font has been created!" );

HideMousePtr( );
imgsize = imagesize( 150, 155, 490, 370 );
if ((buffer = farmalloc(imgsize)) == NULL)
{
    CloseScribbleFiles( );
    closegraph( );
    fprintf( "\r\nError: Not enough memory!\r\n\a" );
    exit(1);
}
getimage( 150, 155, 490, 370, buffer );

GWindow( 150, 155, 490, 370, title );

setfillstyle( SOLID_FILL, RED );
bar( 160, 185, 195, 200 );
setfillstyle( SOLID_FILL, WHITE );
bar( 190, 200, 225, 215 );
setfillstyle( SOLID_FILL, GREEN );
bar( 220, 215, 255, 230 );
CmdButton( OKBUTTON, NORMAL );
MyOuttextxy( 308, 341, "OK", BLACK );

settextstyle( DEFAULT_FONT, HORIZ_DIR, 3 );
MyOuttextxy( 230, 190, "Scribble", BLACK );
settextstyle( DEFAULT_FONT, HORIZ_DIR, 1 );

switch( msgno )
{
    case FSIZEERR:
        xx = X4CenteredMsg("Error:Cannot create more fonts!");
        MyOuttextxy( xx, 281, "Error: Cannot create more fonts!", RED );
        xx = X4CenteredMsg( "Reason: File size is limited" );
        MyOuttextxy( xx, 294, "Reason: File size is limited", RED );
        xx = X4CenteredMsg( "Suggestion: Try small fonts" );
        MyOuttextxy( xx, 307, "Suggestion: Try small fonts", RED );
        xx = X4CenteredMsg( "Quitting..." );
        MyOuttextxy( xx, 320, "Quitting...", RED );

    case THANKS:
        xx = X4CenteredMsg( "Thanks for using Scribble!" );
        MyOuttextxy( xx, 240, "Thanks for using Scribble!",
                    BLACK );
        xx = X4CenteredMsg( "May God bless you!" );

```

174 A to Z of C

```
MyOuttextxy( xx, 253, "May God bless you!", BLUE );
xx = X4CenteredMsg( tmpmsg );
MyOuttextxy( xx, 266, tmpmsg, BLACK );
break;
case ABOUT:
xx = X4CenteredMsg( "Version 1.0" );
MyOuttextxy( xx, 217, "Version 1.0", BLACK );
xx = X4CenteredMsg( "by" );
MyOuttextxy( xx, 235, "by", BLACK );
xx = X4CenteredMsg( "R. Rajesh Jeba Anbiah" );
MyOuttextxy( xx, 248, "R. Rajesh Jeba Anbiah", BLACK );
xx = X4CenteredMsg( "Tamil Nadu, South India" );
MyOuttextxy( xx, 261, "Tamil Nadu, South India", BLACK );
xx = X4CenteredMsg( "xxxxxxxxxx@yahoo.com" );
MyOuttextxy( xx, 274, "xxxxxxxxxx@yahoo.com", BLUE );
xx = X4CenteredMsg( "http://xxxxxxxxxxxxxxxxxxxx.com" );
MyOuttextxy( xx, 287, "http://xxxxxxxxxxxxxxxxxxxx.com",
              BLUE );

MyOuttextxy( 160, 308,
              "Copyright c 2001, R. Rajesh Jeba Anbiah", BLACK );
setcolor( BLACK );
circle( 243, 312, 5 );
MyOuttextxy( 160, 323, "All Rights Reserved.", BLACK );
}
ShowMousePtr( );
do
{
  GetMousePos( &mbutton, &mx, &my );
  if ( mbutton==LFTCLICK )
  {
    cmdno = CmdButtonVal( mx, my );
    if ( cmdno==OKBUTTON )
    {
      HideMousePtr( );
      CmdButton( cmdno, PRESS );
      ShowMousePtr( );
      prevcmdno = cmdno;
      do
      {
        GetMousePos( &mbutton, &mx, &my );
        cmdno = CmdButtonVal( mx, my );
      } while( mbutton==LFTCLICK && cmdno==prevcmdno );
      HideMousePtr( );
      CmdButton( prevcmdno, NORMAL );
      ShowMousePtr( );
    }
  }
}
```

```

    } while( cmdno!= OKBUTTON );
    HideMousePtr( );
    putimage( 150, 155, buffer, COPY_PUT );
    farfree( buffer );
    ShowMousePtr( );
} /*--MsgWindow( )-----*/

/*-----
    X4CenteredMsg - Returns X coordinate value
                   for the center justified message
                   in MsgWindow.

    Logic:( 150, y ) msg ( 490, y )

    To have centered msg,
    150 + ((490-150)-textwidth(msg))/2. */

int X4CenteredMsg( char *str )
{
    return( 150 + ( 340 - textwidth( str ) ) /2 );
} /*--X4CenteredMsg( )-----*/

/*-----
    MakeFontProcedure1 - Creates the first font
                       ie, ' ' ( space )      */

void MakeFontProcedure1( void )
{
    charoffset = ftell( chInfoFp );
    charwidth = 14;

    fInfo.x = 0;
    fInfo.y = charwidth;
    fInfo.op1 = CHARSCAN1;
    fInfo.op2 = CHARSCAN2;
    fwrite( &fInfo, sizeof( fInfo ), 1, chInfoFp );

    fInfo.op1 = EOFCHAR1;
    fInfo.op2 = EOFCHAR2;
    fwrite( &charoffset, sizeof( charoffset ), 1, chOffFp );
    fwrite( &charwidth, sizeof( charwidth ), 1, wthFp );

    fwrite( &fInfo, sizeof( fInfo ), 1, chInfoFp );
} /*--MakeFontProcedure1( )-----*/

/*-----
    MakeFontProcedure2 - Creates the fonts
                       and store the commands in a

```

176 A to Z of C

```

                                temporary file                                */
void MakeFontProcedure2( void )
{
    int xmin, xmax, ymin, ymax, x, y, xt;

    ++scriFh.noOfChars;
    /* Scans the drawing box...
       To find character's xmin, xmax, ymin, ymax.
       Steps: ( top to bottom )
           top   : ----->
                : ---> ----->
           ...
           ...
           bottom: ---->
    */
    xmin = 209;    xmax = 149;
    ymin = 256;    ymax = 221;
    for ( y = 221 ; y<=256 ; ++y )
        for ( x = 149 ; x<=209 ; ++x )
            {
                if ( getpixel( x, y ) == BLACK )
                    {
                        if ( x<xmin )
                            xmin = x;
                        if ( y<ymin )
                            ymin = y;
                        if ( x>xmax )
                            xmax = x;
                        if ( y>ymax )
                            ymax = y;
                    }
            }
    /* Drawing box empty?
       ( No character? )
       check...
    */
    if ( xmin==209 && xmax==149 ) /* if no character */
        charwidth = 0;
    else
        charwidth = xmax - xmin + 4;
    fwrite( &charwidth, sizeof( charwidth ), 1, wthFp );
    if ( charwidth==0 )
        charoffset = 0;
    else
        charoffset = ftell( chInfoFp );
    fwrite( &charoffset, sizeof( charoffset ), 1, chOffFp );
}
```



```

/* Scans the character...
   To write character commands.
   Steps: ( top to bottom )
           top    : ----->
                : ---> ----->
           ...
           ...
           bottom: ---->
*/
for ( y = ymin ; y<=ymax ; ++y )
  for ( x = xmin ; x<=xmax ; ++x )
  {
    if ( getpixel( x, y ) == BLACK )
    {
      fInfo.x = 247 - y;
      fInfo.y = x - xmin;
      fInfo.op1 = CHARSCAN1;
      fInfo.op2 = CHARSCAN2;
      fwrite( &fInfo, sizeof( fInfo ), 1, chInfoFp );
      for ( xt=x ; getpixel( xt, y ) == BLACK ; ++xt )
        ;
      --xt;
      x = xt;
      fInfo.x = 247 - y;
      fInfo.y = xt - xmin;
      fInfo.op1 = DRAWCHAR1;
      fInfo.op2 = DRAWCHAR2;
      fwrite( &fInfo, sizeof( fInfo ), 1, chInfoFp );
    }
  }
if ( charwidth!=0 )
{
  fInfo.x = 0;
  fInfo.y = charwidth;
  fInfo.op1 = CHARSCAN1;
  fInfo.op2 = CHARSCAN2;
  fwrite( &fInfo, sizeof( fInfo ), 1, chInfoFp );
  fInfo.op1 = EOFCHAR1;
  fInfo.op2 = EOFCHAR2;
  fwrite( &fInfo, sizeof( fInfo ), 1, chInfoFp );
}
} /*--MakeFontProcedure2( )-----*/
/*-----
   MakeFontProcedure3 - Creates the final font
                       file with the headers & using the
                       stored commands from temporary file.      */

```

178 A to Z of C

```
void MakeFontProcedure3( void )
{
    scriFh.fntSize = 16 + 3 * scriFh.noOfChars + ftell( chInfoFp );
    scriFh.defOffset = 16 + 3 * scriFh.noOfChars;

    fseek( chOffFp, 0L, SEEK_SET );
    fseek( wthFp, 0L, SEEK_SET );
    fseek( chInfoFp, 0L, SEEK_SET );

    fwrite( &scriFh, sizeof( FILEHEADER ), 1, scriFp );

    while ( fread( &charoffset, sizeof( charoffset ), 1, chOffFp ) == 1 )
        fwrite( &charoffset, sizeof( charoffset ), 1, scriFp );

    while ( fread( &charwidth, sizeof( charwidth ), 1, wthFp ) == 1 )
        fwrite( &charwidth, sizeof( charwidth ), 1, scriFp );

    while ( fread( &fInfo, sizeof( fInfo ), 1, chInfoFp ) == 1 )
        fwrite( &fInfo, sizeof( fInfo ), 1, scriFp );
    CloseScribbleFiles( );
} /*--MakeFontProcedure3( )-----*/

/*-----
    CloseScribbleFiles - Closes all Scribble
                        files and then deletes the
                        temporary files.      */

void CloseScribbleFiles( void )
{
    fcloseall( );

    remove( "~$scrib1.raj" );
    remove( "~$scrib2.raj" );
    remove( "~$scrib3.raj" );
} /*--CloseScribbleFiles( )-----*/

/*-----
    main - Main of Scribble      */
int main( void )
{
    int mx, my, premx, premy,
        mbutton, cmdno, prevcmdno, bno, prevbno = 0, msgno = THANKS;
    long fontsize;
    char ch[2] = "!", fontname[10];
    BOOLEAN stayin = TRUE;

    ScribbleInfo( );
```

```

InitScribble( );
SetScreen( );
GetFontName( fontname );
strcat( fontname, ".CHR" );
if ( ( scriFp = fopen( fontname, "wb+" ) ) == NULL )
    {
        CloseScribbleFiles( );
        closegraph( );
        cprintf( "Fatal Error(04): File cannot be created \r\n\a" );
        exit( 1 );
    }

MakeFontProcedure1( );

FileSizeIndicator( );

ShowMousePtr( );
while( stayin )
    {
        GetMousePos( &mbutton, &mx, &my );
        if ( mbutton==LFTCLICK )
            {
                if ( mx>=149 && mx<=209 && my>=223 && my<=256 ) /* drawing
                                                                    box */
                    {
                        if ( prevbno>4 )
                            setcolor( WHITE );
                        else
                            setcolor( BLACK );
                        RestrictMousePtr( 150+(prevbno%5)/2,
                                         221+(prevbno%5)/2, 208-(prevbno%5)/2,
                                         255-((prevbno+1)%5)/2 );
                        premx = mx;
                        premy = my;
                        HideMousePtr( );
                        PutPoint( mx, my, prevbno%5 );
                        ShowMousePtr( );
                        do
                            {
                                GetMousePos( &mbutton, &mx, &my );

                                if ( premx!=mx || premy!=my )
                                    {
                                        HideMousePtr( );
                                        ScribbleLine( premx,premy,mx, my, prevbno%5 );
                                        ShowMousePtr( );
                                    }
                            }
                    }
            }
    }

```

180 A to Z of C

```
        premx = mx;
        premy = my;
    }
    } while(mbutton==LFTCLICK);
    RestrictMousePtr( 0, 0, 639, 479 );
}
bno = BrushVal( mx, my );
if ( bno!=MAXBRUSH && bno != prevbno )
{
    HideMousePtr( );
    BrushBox( prevbno, NORMAL );
    BrushBox( bno, PRESS );
    prevbno = bno;
    ShowMousePtr( );
}
cmdno = CmdButtonVal( mx, my );
if ( cmdno!=MAXCMDBUTTON && cmdno!= OKBUTTON
    && cmdno!=NOBUTTON && cmdno!=YESBUTTON )
{
    HideMousePtr( );
    CmdButton( cmdno, PRESS );
    ShowMousePtr( );
    prevcmdno = cmdno;
    do
    {
        GetMousePos( &mbutton, &mx, &my );
        cmdno = CmdButtonVal( mx, my );
    } while( mbutton==LFTCLICK && cmdno==prevcmdno );
    HideMousePtr( );
    CmdButton( prevcmdno, NORMAL );
    ShowMousePtr( );
    stayin = ( cmdno!=QUIT );
}
switch( cmdno )
{
    case CLEAR:
        Clear( );
        break;
    case NEXT:
        HideMousePtr( );
        MakeFontProcedure2( );
        FileSizeIndicator( );
        Clear( );
        ++ch[0];
        fontsize = 16 + 3*scriFh.noOfChars +
            ftell( chInfoFp );
        if ( fontsize >= 30000 )
```

```

        {
            msgno = FSIZEERR;
            stayin = FALSE;
        }
        else if ( ch[0]==0 )
            stayin = FALSE;
        if ( ch[0]!=0 && fontsize<30000 )
        {
            settextstyle(DEFAULT_FONT, HORIZ_DIR, 4 );
            MyOuttextxy( 150, 225, ch, BLACK );
            settextstyle(DEFAULT_FONT, HORIZ_DIR, 1 );
        }
        ShowMousePtr( );
        break;
    case QUIT:
        HideMousePtr( );
        MakeFontProcedure2( );
        ShowMousePtr( );
        break;
    case ABOUT:
        MsgWindow( fontname, ABOUT );
    }
}
}
MakeFontProcedure3( );
MsgWindow( fontname, msgno );
closegraph( );
return( 0 );
} /*--main( )-----*/

```

30.7 Scribble.prj

We use project (.PRJ) file to create standalone program. By the term *standalone*, we mean the EXE file that doesn't require any other (supporting) files for its execution.

Normally in BGI programming, we would supply the driver (BGI) files' directory with `initgraph()` function. If the corresponding BGI file is not found on that directory you would get error message. We get this error message because, the driver files are not added with our program. But if you have added the corresponding object (OBJ) file of the driver, to `graphics.lib` library, you won't get such error. You can use BGI OBJ utility to create object file for the driver (BGI & CHR) files.

```
C:\>BGI OBJ /F egavga
```

the /F switch is to get "far" object code.

Then you will get `Egavgaf.obj`. Similarly you can create object file for any CHR or BGI files. You can add the object file to `graphics.lib` using TLIB as:

182 A to Z of C

```
C:\> TLIB graphics + obj1 [+obj2...]
```

Adding object file to `graphics.lib` is not advisable as it would increase the compilation time. So the easy way is to add object file is through project file. For my Scribble project, I haven't used any CHR files, so I need to create object file only for `EGAVGA.BGI` driver. I have used the `registerfarbgidriver()` function to register the BGI driver so that it is being also added with our standalone EXE file.

Note
If you use other CHR files, just create object files for all the CHR files using <code>BGIOBJ</code> utility, then register them using <code>registerfarbgifont()</code> function.

Add the following files in `Scribble.prj`:

- i. `Mouselib.lib`
- ii. `Egavgaf.obj`
- iii. `Scribble.c`

Compile the `Scribble.prj` to get standalone `Scribble.exe` file.