C++ Graphics for Windows using WINBGIm

Download and install the WINBGIm devpak file "WinBGIm-6.0-1g17l.DevPak".

www.onecore.net/winbgim-graphics.htm

Open Dev C++ and create a new project.

File -> New -> Project...
Choose the WinBGIm tab.
Choose either with our without console.
Give your project a name.
The newly created project will contain source code.
Save the cpp file with any name you choose.
Put it in the same folder as the project.
Compile the project.

The project folder now will contain 9 files:
(Using the project name Project1, and the file name file1)
Project1.ico – a default icon for the exe file, and application frame.
File1.cpp – Your C++ source code.
Makefile.win – Specifies files to use in project, compiler to use...
Project1.dev – Dev C++ Project file. Double click this to open project.
Project1_private.h – Header file that would hold complicated stuff if we had any.
Project1_private.rc – Precompiled resource file, has name of icon file.
File1.o – Output file from compiling File1.cpp
Project1.exe – Linked executable for entire project.
Project1_private.rc – Compiled resource file.

The source code

```
#include <iostream>
#include <graphics.h>
using namespace std;

int main( )
{
     initwindow( 640 , 480 , "WinBGIm" );
     outtextxy( 0 , 0 , "Hello WinBGIm!" );
     bgiout << "Hello WinBGIm-Stream!" ;
     outstreamxy( 0 , 15 );
     while( !kbhit() );
     closegraph( );
     return( 0 );
}
```

void initwindow(int width, int height, string title)

The initwindow() function opens a window that we can make graphics calls to. You can specify the width, height, and title.

void outtextxy(int x, int, y, string text)

The outtextxy() function displays text in the window. You can specify the x,y coordinate and the text.

The closegraph() function ends the graphics session and closes the window.

The cleardevice() function clears the window to the current background color.

Examples:

```
int main( )
{
      initwindow( 640 , 480 , "Graphics" );
      outtextxy( 50 , 50 , "The Internet is Tubes!" );
      system("pause");
      closegraph( );
      return( 0 );
}

int main( )
{
      initwindow( 640 , 480 , "Timing" );
      outtextxy( 50 , 50 , "The Internet is Tubes!" );
      delay(2000);
      outtextxy( 100 , 100 , "The Internet is Not a Big Truck!" );
      delay(2000);
      closegraph( );
      return( 0 );
}

int main( )
{
    initwindow( 640 , 480 , "Repetition" );
    while(true)
    {
          outtextxy( 50 , 50 , "The Internet is Tubes!" );
          delay(2000);
          cleardevice();
          outtextxy( 100 , 100 , "The Internet is Not a Big Truck!" );
          delay(2000);
          cleardevice();
     }
      closegraph( );
      return( 0 );
}
```

Drawing shapes

`void setcolor(int color)`

Sets the drawing (outlining) color.
The argument can be one of the integers in the table to the right, or a new color can be created using the COLOR function.

BLACK	0
BLUE	1
GREEN	2
CYAN	3
RED	4
MAGENTA	5
BROWN	6
LIGHTGRAY	7
DARKGRAY	8
LIGHTBLUE	9
LIGHTGREEN	10
LIGHTCYAN	11
LIGHTRED	12
LIGHTMAGENTA	13
YELLOW	14
WHITE	15

`int COLOR(int red, int green, int blue)`

Creates a new RGB color and represents it as an integer.

`void setbkcolor(int color)`

Sets the background color.

`void setfillstyle(int pattern, int color)`

Sets the fill pattern and the fill color.
The table to the right shows allowable fill constants.

EMPTY_FILL	0	Fill with background color
SOLID_FILL	1	Solid fill
LINE_FILL	2	Fill with ---
LTSLASH_FILL	3	Fill with ///
SLASH_FILL	4	Fill with ///, thick lines
BKSLASH_FILL	5	Fill with \\\, thick lines
LTBKSLASH_FILL	6	Fill with \\\
HATCH_FILL	7	Light hatch fill
XHATCH_FILL	8	Heavy crosshatch fill
INTERLEAVE_FILL	9	Interleaving line fill
WIDE_DOT_FILL	10	Widely spaced dot fill
CLOSE_DOT_FILL	11	Closely spaced dot fill
USER_FILL	12	User-defined fill pattern

`void arc(int x, int y, int a1, int a2, int r)`

Draws an arc.
x and y are the center coordinates.
a1 and a2 are the starting and ending angles measures counterclockwise in degrees with zero degrees being in the positive x direction.
r is the radius of the circle defining the arc.

`void bar(int left, int top, int right, int bottom)`

Fills a rectangular area with the current fill color.

`void circle(int x, int y, int r)`

Draws a circle centered at x,y with radius r using the current drawing color.

`void floodfill(int x, int y, int boundcolor)`

Fills the area around x,y with the current fill color bounded by the argument boundcolor.

void line(int x1, int y1, int x2, int y2)

Draws a line from x1,y1 to x2,y2 using the current drawing color.

void pieslice(int x, int y, int a1, int a2, int r)

Draws an arc using the current drawing color and fills the arc using the current fill color.
x and y are the center coordinates.
a1 and a2 are the starting and ending angles measures counterclockwise in degrees with zero degrees being in the positive x direction.
r is the radius of the circle defining the arc.

void rectangle(int left, int top, int right, int bottom)

Draws a rectangle outline using the current drawing color.

Example:

```
int main( )
{
    initwindow( 640 , 480 , "Shapes" );
    int dcolor = COLOR(155, 0, 255);
    int fcolor = COLOR(255, 0, 155);
    int bcolor = COLOR(0, 100, 0);
    setcolor(dcolor);
    setfillstyle(SOLID_FILL, fcolor);
    setbkcolor(bcolor);
    cleardevice();
    arc(200, 200, 0, 90, 100);
    bar(20, 300, 300, 350);
    circle(100, 100, 100);
    circle(400, 400, 50);
    floodfill(400, 400, dcolor);
    line(20, 20, 400, 400);
    rectangle(300, 50, 400, 400);
    pieslice(200, 200, 0, 120, 50);
    system("pause");
    closegraph( );
    return( 0 );
}
```

Animation

```
int main( )
{
    int x = 250;
    int xv = 5;

    initwindow( 500 , 200 , "BouncingBall" );
    int dcolor = COLOR(155, 0, 255);
    int fcolor = COLOR(255, 0, 155);
    int bcolor = COLOR(0, 55, 0);
    setcolor(dcolor);
    setfillstyle(SOLID_FILL, fcolor);
    setbkcolor(bcolor);
    while(true)
    {
        x += xv;
        if(x > 300)
        {
            xv = - 10;
        }
        if(x < 200)
        {
            xv = 10;
        }

        cleardevice();
        circle(x, 100, 10);
        floodfill(x, 100, dcolor);
        delay(50);
    }
    closegraph( );
    return( 0 );
}
```

```
class Ball
{
private:
int x;
int xv;
int dcolor;

public:
Ball()
{
    x = 250;
    xv = 10;
    dcolor = COLOR(255, 0, 55);
    setcolor(dcolor);
    setfillstyle(SOLID_FILL, COLOR(155, 0, 55));
}
void paint()
{
    x += xv;
    if(x > 300)
    {
        xv = - 10;
    }
    if(x < 200)
    {
        xv = 10;
    }
    circle(x, 100, 10);
    floodfill(x, 100, dcolor);
}
};

int main( )
{
    initwindow( 500 , 200 , "BouncingBall" );
    Ball ball;
    int bcolor = COLOR(0, 55, 0);
    setbkcolor(bcolor);
    while(true)
    {
        cleardevice();
        ball.paint();
        delay(50);
    }
    closegraph();
    return(0);
}
```

For More WINBGIm:

http://www.cs.colorado.edu/~main/cs1300/doc/bgi/index.html