

iGraphics: A Wrapper For OpenGL in 2D

S. M. Shahriar Nirjon
Lecturer, Dept. of CSE, BUET
email: nirjon@cse.buet.ac.bd
Date: Feb 26, 2008

Abstract

iGraphics.h header file contains some drawing functions that can be used to draw basic graphical shapes in Visual C++. These functions are implemented in OpenGL. Users of iGraphics do not need any knowledge of OpenGL to use it. Simply calling the drawing functions a user can draw any 2D shape on screen. This library also provides easy ways for animation, keyboard and mouse event handling.

1. Setup

Just copy the iGraphics folder in your PC. The folder contains the following files- GLUT.H, GLUT32.LIB, GLUT32.DLL, iGraphics.h, iMain.cpp and iDoc.pdf.

2. Description of iMain.cpp

Users of iGraphics only have to edit, compile and run iMain.cpp using Visual C++ 6. See the listing of iMain.cpp with comments.

```
# include "iGraphics.h"

/*
    Function iDraw() is called again and again by the system.
*/
void iDraw()
{
    //place your drawing codes here
}

/*
    Function iMouseMove() is called mouse is dragged while pressed.
    (mx, my) is the position where the mouse pointer is.
*/
void iMouseMove(int mx, int my)
{
    //place your codes here
}

/*
    Function iMouse() is called when mouse is pressed/released.
    (mx, my) is the position where the mouse pointer is.
*/
void iMouse(int button, int state, int mx, int my)
{
    if(button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
    {
        //place your codes here
    }
    if(button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
    {
        //place your codes here
    }
}

/*
    Function iKeyboard() is called whenever a key is hit.
    key- holds the ASCII value of the key pressed.
*/
void iKeyboard(unsigned char key)
{
    if(key == 'q')
    {
        exit(0);
    }
}
```

```
    }
    //place your codes for other keys here
}

int main()
{
    //place your own initialization codes here.
    iInitialize(400, 400, "demooo");
    return 0;
}
```

Fig: iMain.cpp

3. Functions in iGraphics.h

void iInitialize(int width=500, int height=500,char* title="iGraphics")
Description: Creates a window of specified size and title. Parameters: width- Width of the window. height- Height of the window. title- Title of the window.
void iClear()
Description: Clears the screen. Parameters: none
void iSetColor(double r, double g, double b)
Description: Sets current drawing color. Parameters: r- Red component of color. g- Green component of color. b- Blue component of color.
void iDelay(int sec)
Description: Pauses the program for some seconds. Parameters: sec- Pause time in seconds.
void iPoint(double x, double y, int size=0)
Description: Draws a ppoint(x, y) on screen with current color. Parameters: x, y- Coordinates of the point. size- (Optioal)Size of the point.
void iLine(double x1, double y1, double x2, double y2)
Description: Draws a line on the screen with current color. Parameters: x1, y1- Coordinates of one end point. x2, y2- Coordinates of other end point.
void iCircle(double x, double y, double r, int slices=100)
Description: Draws a circle on the screen with current color. Parameters: x, y- Coordinates of center. r- Radius of circle. slices- Number of line segments used to draw the circle.
void iFilledCircle(double x, double y, double r, int slices=100)
Description: Draws a filled-circle on the screen with current color. Parameters: x, y- Coordinates of center. r- Radius of circle. slices- Number of line segments used to draw the circle.
void iEllipse(double x, double y, double a, double b, int slices=100)
Description: Draws an ellipse on the screen with current color. Parameters: x, y- Coordinates of center. a, b- Sizes of major and minor axes. slices- Number of line segments used to draw the circle.
void iFilledEllipse(double x, double y, double a, double b, int slices=100)
Description: Draws a filled-ellipse on the screen with current color.

Parameters: x, y- Coordinates of center. a, b- Sizes of major and minor axes. slices- Number of line segments used to draw the circle.
void iRectangle(double left, double bottom, double dx, double dy) Description: Draws a rectangle on the screen with current color. Parameters: left- x-coordinate of lower-left corner of the screen. bottom- y-coordinate of lower-left corner of the screen. dx- width of rectangle. dy- height of rectangle.
void iFilledRectangle(double left, double bottom, double dx, double dy) Description: Draws a filled-rectangle on the screen with current color. Parameters: left- x-coordinate of lower-left corner of the screen. bottom- y-coordinate of lower-left corner of the screen. dx- width of rectangle. dy- height of rectangle.
void iPolygon(double x[], double y[], int n) Description: Draws a polygon on the screen with current color. Parameters: x- x coordinates of vertices of a polygon. y- y coordinates of vertices of a polygon. n- Number of vertices.
void iFilledPolygon(double x[], double y[], int n) Description: Draws a filled-polygon on the screen with current color. Parameters: x- x coordinates of vertices of a polygon. y- y coordinates of vertices of a polygon. n- Number of vertices.
void iText(GLdouble x, GLdouble y, char *str, void* font=GLUT_BITMAP_8_BY_13) Description: Displays a string on screen. Parameters: x, y- coordinates of the first character of the string. str- The string to show. font- (Optional)Specifies the font type. Values could be any one of the following- {GLUT_BITMAP_8_BY_13, GLUT_BITMAP_9_BY_15, GLUT_BITMAP_TIMES_ROMAN_10, GLUT_BITMAP_TIMES_ROMAN_24, GLUT_BITMAP_HELVETICA_10, GLUT_BITMAP_HELVETICA_12, GLUT_BITMAP_HELVETICA_18}
void iShowBMP(int x, int y, char filename[]) Description: Shows a 24-bit .bmp file on screen. The size must be an integral power of 2. Parameters: x, y- coordinates of lower-left corner of .bmp file. filename- The path of the .bmp file.
void iSettimer(int msec, void (*f)(void)) Description: Schedules a task to perform after some prespecified time interval. There can be at most 10 timers in your program. Once started these timers cannot be stopped. This function must be called before the iInitialize() function call. Parameters: msec- Time interval in mili-seconds. f- The function that will be called automatically by the system again and again after the specified time interval.

4. Why iEverything() ?

All the function names in iGraphics library starts with the character ‘i’. This is done- so that, functions YOU write in your program don’t match with any library function.

5. Conclusion

iGraphics is still under development. In future versions, more new and exciting features will be introduced.