

IGLWidget v1.0

Crossplatform OpenGL class

**intended for simplify using OpenGL
on Linux and Windows
without code change**

CROSSPLATFORM EDITION

For Kylix2 & Delphi6

The IGLWidget is free software.
Copyright (C) 2002 Artur Bać All Rights Reserved.
REDA POLAND arturbac@poczta.onet.pl

1. Legal notice

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL ARTUR BAĆ BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Developed with Kylix2 Open Edition & Delphi6 Open Edition

**Piece of part of windows code is written on the base
of source CGLib project by Tom Nuygen**

2. Requirements

2.1. Linux users

a) Linux OS (ex. Mandrake 8.0 , Redhat 7.1)

b) Kylix2 instaled

For Mandrake 8.1 uses

I'm awaitng for patch for borlands qtinf library thus there is a problem

Kylix uses Qt library ver 2.3.0 , but Mandrake 8.1 Qt lib ver 2.3.1

As I read in Qt 2.3.1 notes they have to be compatible

But this qtinf lib make exceptions and will not work with qt 2.3.1

Problem we can obey with simple script that will start our compiled Application

Just remake copy of starkylix script to point your compiled App.

2.2. Windows users

Windows operating system with opengl.dll and glu.dll
 Most 3d cards installs appropriate opengl.dll for windows98
 Delphi 6 installed

3. Using

Using is identical for both Linux and Windows systems.

3.1. Constants & Types

For specifying requirements on new Rendering Context

```
glf_DoubleBuffer
glf_DepthBuffer
glf_Rgba
glf_AlphaChannel
glf_AccumBuffer
glf_StencilBuffer
glf_StereoBuffers
glf_DirectRendering
glf_Overlay
glf_SingleBuffer
glf_NoDepthBuffer
glf_ColorIndex
glf_NoAlphaChannel
glf_NoAccumBuffer
glf_NoStencilBuffer
glf_NoStereoBuffers
glf_IndirectRendering
glf_NoOverlay
```

If no setting are specified during setup these defaults will be used

```
DefaultGLFormat : EnumTIGLFormat = glf_DoubleBuffer
    or glf_DirectRendering
    or glf_DepthBuffer
    or glf_Rgba
    or glf_Overlay
    or glf_NoAlphaChannel
    or glf_NoStereoBuffers
    or glf_NoAccumBuffer
    or glf_NoStencilBuffer;
```

For specifying requirements on new Rendering Context

```
TIGLBits = record
    cColorBits: Byte;
    cDepthBits: Byte;
    cStencilBits: Byte;
    cRedBits: Byte;
    cGreenBits: Byte;
    cBlueBits: Byte;
    cAlphaBits: Byte;
    cAccumBits: Byte;
    cAccumRedBits: Byte;
    cAccumGreenBits: Byte;
    cAccumBlueBits: Byte;
    cAccumAlphaBits: Byte;
```

end;

If no setting are specified during setup these defaults will be used

Const

```
DefaultTIGLBits : TIGLBits =(cColorBits : 24;
                             cDepthBits : 24;
                             cStencilBits: 0; );
```

3.2. Variables

TIGLGraphicContext = class (TObject)

```
IGLFormat : EnumTIGLFormat;
bits      : TIGLBits;
```

For specifying requirements on new Rendering Context

```
major_glx,
minor_glx:integer;
```

**Major and minor version of glx extension present in the system
apply only to linux**

Methods

Constructor Create (panel : TPanel);

For creating glpanel
as param. give panel which have to be used for rendering context

Destructor Destroy;

Destroys class and cleans up Rendering contexts

Procedure GLSetup;

After specifying IGLFormat and bits initiates rendering context

Procedure XWaitForGL;

Procedure XWaitForX;

Additional methods for linux users refer to glx reference for glxWaitGL and glxWaitX
on <http://www.opengl.org>

Procedure MakeCurrent;

activates adequate rendering context for current class

Procedure SwapActiveBuffers;

Swaps buffers for adequate rendering context for current class
Use it with Paint events

Procedure SetViewport;

Sets ViewPort for adequate rendering context for current class
Use it with Resize events

3.3. Initialization

Initialization and use is the same for linux and windows users

example

```
.....
      { Public declarations of form class }
      GLO : TIGLGraphicContext;
.....
```

Creation of form class

```
.....
GLO := TIGLGraphicContext.Create(Panel1);
GLO.IGLFormat := glf_DoubleBuffer
               or glf_DirectRendering
               or glf_DepthBuffer
               or glf_Rgba
               or glf_NoOverlay
               or glf_NoAlphaChannel
               or glf_NoStereoBuffers
               or glf_NoAccumBuffer
               or glf_NoStencilBuffer;
with GLO.bits do begin
    cColorBits := 8; //Minimal of bits we can accept
    cDepthBits := 8;
    cAlphaBits := 1;
```

```

        end;
        glo.major_glx:=0;
        glo.minor_glx:=0;
        GLO.GLSetup;
        GLO.MakeCurrent;
        Caption:=Caption +' : Glx ver. ' +IntToStr(GLO.major_glx)+'.'+IntToStr(GLO.minor_glx);

        .....
//Redy to use

procedure TForm1.FormPaint(Sender: TObject);
        .....
        GLO.SwapActiveBuffers;
end;

Resize for Form or Panel
procedure TForm1.Resize(Sender: TObject);
begin
    GLO.SetViewPort;
    GLO.SwapActiveBuffers;
end;

In destructor of form or quit method
    GLO.Destroy;

Remember that you can create multiple Panels with different contexts
        GLO1 : TIGLGraphicContext;
        GLO2 : TIGLGraphicContext;

        .....
        GLO1 :=TIGLGraphicContext.Create(Panel1);
        GLO2 :=TIGLGraphicContext.Create(Panel2);
        GLO1.GLSetup;
        GLO2.GLSetup;
        GLO1.MakeCurrent;

        .....
        glcode for glo1
        .....
                GLO2.MakeCurrent;

        .....
        glcode for glo2
        .....

End

```