

Hello community,

Today seems to be the good day to present my project of doing 3D game engine named UltimateOpenGL.

The project is still at beta state but I think it's good enough to actually show it and show some code and example.

Moreover having some return on the way of how it works can be a good point to improve it.

So let's dive inside :

UltimateOpenGL is a 3D motor used to make 3D games made by a beginner. It's designed to be "code only". it mean you don't have (at this time) any GUI to help you create scene with your mouse.

At this time, UOGL allow Scene creation, insertion of multiple 'GameObject'(Shapes, Lights), insertion of multiple camera per scene. Model loading and rendering with Assimp

https://github.com/Xemuth/UltimateOpenGL_V3

Here is some screenshot of some example I made :

here you have some code example of how it's working :

```
/**All basic code to set Up glfw is coming before see my exemple or Readme of my
GITHUB**//
context.AddTexture("sand",TransformFilePath("/Textures/sand.jpg")); //Load sand texture

Scene& myScene = context.AddScene("main");
myScene.AddCamera("main");
myScene.SetBackgroundColor(context.TransformRGBToFloatColor(40,180,200));

Mesh m; //Use to simulate the flat float under the model
m.GetVertices().Add().SetPosition(glm::vec3(-20.0f, 0.0f, -20.0f)).SetTexCoords(glm::vec2(0.0f,
20.0f));
m.GetVertices().Add().SetPosition(glm::vec3(20.0f, 0.0f,
-20.0f)).SetTexCoords(glm::vec2(20.0f, 20.0f));
m.GetVertices().Add().SetPosition(glm::vec3(20.0f, 0.0f, 20.0f)).SetTexCoords(glm::vec2(20.0f,
0.0f));
m.GetVertices().Add().SetPosition(glm::vec3(20.0f, 0.0f, 20.0f)).SetTexCoords(glm::vec2(20.0f,
0.0f));
m.GetVertices().Add().SetPosition(glm::vec3(-20.0f, 0.0f, 20.0f)).SetTexCoords(glm::vec2(0.0f,
0.0f));
```

```

    m.GetVertices().Add().SetPosition(glm::vec3(-20.0f, -0.0f,
-20.0f)).SetTexCoords(glm::vec2(0.0f, 20.0f));

    Object3D& floor = myScene.CreateGameObject<Object3D>("floor",m); //Lets create floor
    floor.BindTexture("sand");

    Object3D& modele = myScene.CreateGameObject<Object3D>("modele");
    modele.LoadModel("C:\\Upp\\myapps\\ExempleUltimateOpenGL_V3\\obj upp\\upp.obj");
    //Loading of model

    modele.GetTransform().SetNewPosition(glm::vec3(0,4,0)); //Set new position
    modele.GetTransform().ScaleNewValue(glm::vec3(0.05f,0.05f,0.05f)); //Scale the model

    modele.SetOnDrawFunction([](GameObject& gm){ //Bind event on draw
        double rotation = glm::cos(context.GetEllapsedTime())/100;
        gm.GetTransform().RotateFromEulerAngles(context.GetDeltaTime() * 2,glm::vec3(0,1,0) );
    });

    myScene.Load();
    while(!glfwWindowShouldClose(window)) {
        processInput(window);

        glfwSetWindowTitle(window, "UltimateOpenGL V3 - " +AsString(context.GetFPS()) + " FPS");
        try{
            context.Draw(); //Draw the context
        }catch(UGLException& e){
            LOG(e.what());
        }

        glfwSwapBuffers(window);
        glfwPollEvents();
    }
    glfwTerminate();

```

I will not dive deeply in this example by the time. But if you are interested, you can find more information and some code example in thoses both github page :

UltimateOpenGL
Exemple using it

Lot of feature are coming, Animation, Physics, Sound...

Let me know in coment what you though about it :)

Subject: Re: UltimateOpenGL [BETA] 3D motor
Posted by [koldo](#) on Mon, 18 Nov 2019 18:53:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Very cool! :)

It would be great if you could include it in Bazaar, including all dependencies to be set in plugin folder :roll:

Subject: Re: UltimateOpenGL [BETA] 3D Engine
Posted by [Xemuth](#) on Mon, 18 Nov 2019 19:06:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

koldo wrote on Mon, 18 November 2019 19:53Very cool! :)

It would be great if you could include it in Bazaar, including all dependencies to be set in plugin folder :roll:

Yeah, I agree, also It can be interesting to connect UltimateOpenGL with Upp OpenGL Control instead of using GLFW !
I will take a look on it :d

Subject: Re: UltimateOpenGL [BETA] 3D motor
Posted by [Klugier](#) on Mon, 18 Nov 2019 21:42:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Xemuth & Koldo,

I am not sure we should put every more complex open source project to Bazaar. Some times ago we discussed to abandon baazar, but we kept status quo. I think links to repositories are fine alternative for us. I plan to add the possibility to add remote packages (e.g. github repo) as a package.

Backing to Xemuth :) I think you made grate work here and probably you had a lot of fun writing this application. I am glad you decided to publish it as an open source. We definitely need more initiative like this. Thanks and good luck in developing this project and other projects in the future!

I definitely agree that you should try to use OpenGL control from Upp :) If something is not possible using this control you should report the bug or contribute to Upp to overcome issue. Please notice that recently we added the possibility to contribute via PR on GitHub!

At the end, I would like to give you some power tips for the future. I notice that you use following code in your header files (.h):

using namespace Upp;

This should be avoided at all cost, because when somebody includes that file it will also inherit this namespace shortcut. To overcome that issue you can put your class into Upp namespace or directly use namespace Upp::. You can find more information on <https://stackoverflow.com/questions/5849457/using-namespace-in-c-headers>.

Sometimes you pass string as a copy when you can pass it by const reference to avoid copy:

```
GameObject::GameObject(String _name){ // <- Copy
    name = _name; // <- Copy one more time
}
```

Should be replace with

```
GameObject::GameObject(const String& _name){ // <- No copy, address (reference) is passed
    name = _name; // <- Copy
}
```

As you may see with const& we avoid one extra copy that is not need in above case.

Sincerely,
Klugier

Subject: Re: UltimateOpenGL [BETA] 3D motor
Posted by [koldo](#) on Tue, 19 Nov 2019 07:10:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Klugier

UltimateOpenGL would be a great new part of Bazaar, and I invite Xemuth to include it there, giving my personal support if needed.

GitHub is very interesting, and my main personal project, BEMRosetta, is there. However I cannot include BEMRosetta in Upp Bazaar because I have released it with GPLv3. However Xemuth has released UltimateOpenGL in a BSD like license so it would fit perfectly in Bazaar.

In addition, the advantage of Bazaar over GitHub, especially in projects like UltimateOpenGL, is that the developer can match sources and dependencies so well that it is very easy for a new developer to run a sample. When in GitHub, a new developer has to arrange it all by hand, and UltimateOpenGL include at least GLM, Glad, Assimp, Glfw, and Bullet3.

I agree with Klugier that it would be great if you could integrate it in OpenGL control. In addition I would ask you to separate (if it is not already done), if possible, the physics from the visualization,

as some applications would not require to add Bullet3. This would be great as I think a thin scene graph manager for U++ would be necessary.

Subject: Re: UltimateOpenGL [BETA] 3D motor
Posted by [koldo](#) on Tue, 19 Nov 2019 08:06:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Other detail, totally personal. I would remove "Ultimate" from source files.

I know in other projects (Qt, Kde) they love to put their acronyms everywhere. However in my humble opinion that is ugly :roll:

Subject: Re: UltimateOpenGL [BETA] 3D Engine
Posted by [Xemuth](#) on Tue, 19 Nov 2019 20:32:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Koldo & Klugier,

Klugier wrote on Mon, 18 November 2019 22:42
Backing to Xemuth :) I think you made grate work here and probably you had a lot of fun writing this application. I am glad you decided to publish it as an open source. We definitely need more initiative like this. Thanks and good luck in developing this project and other projects in the future!

Yeah I had lot of fun writing it :d

Klugier wrote on Mon, 18 November 2019 22:42
At the end, I would like to give you some power tips for the future. I notice that you use following code in your header files (.h):

using namespace Upp;

This should be avoided at all cost, because when somebody includes that file it will also inherit this namespace shortcut. To overcome that issue you can put your class into Upp namespace or directly use namespace Upp::. You can find more information on <https://stackoverflow.com/questions/5849457/using-namespace-in-c-headers>.

you are right, it's now fixed !

Klugier wrote on Mon, 18 November 2019 22:42
Sometimes you pass string as a copy when you can pass it by const reference to avoid copy:

```
GameObject::GameObject(String _name){ // <- Copy
```

```
name = _name; // <- Copy one more time
}
```

Should be replace with

```
GameObject::GameObject(const String& _name){ // <- No copy, address (reference) is passed
name = _name; // <- Copy
}
```

As you may see with const& we avoid one extra copy that is not need in above case.

Yeah I know, I just sometime forget it :roll:

Subject: Re: UltimateOpenGL [BETA] 3D Engine
Posted by [Xemuth](#) on Tue, 19 Nov 2019 20:43:28 GMT
[View Forum Message](#) <> [Reply to Message](#)

koldo wrote on Tue, 19 November 2019 08:10Hello Klugier

UltimateOpenGL would be a great new part of Bazaar, and I invite Xemuth to include it there, giving my personal support if needed.

GitHub is very interesting, and my main personal project, BEMRosetta, is there. However I cannot include BEMRosetta in Upp Bazaar because I have released it with GPLv3. However Xemuth has released UltimateOpenGL in a BSD like license so it would fit perfectly in Bazaar.

In addition, the advantage of Bazaar over GitHub, especially in projects like UltimateOpenGL, is that the developer can match sources and dependencies so well that it is very easy for a new developer to run a sample. When in GitHub, a new developer has to arrange it all by hand, and UltimateOpenGL include at least GLM, Glad, Assimp, Glfw, and Bullet3.

I agree with Klugier that it would be great if you could integrate it in OpenGL control. In addition I would ask you to separate (if it is not already done), if possible, the physics from the visualization, as some applications would not require to add Bullet3. This would be great as I think a thin scene graph manager for U++ would be necessary.

I did some test today with GLCtrl and it Worked !

Let me show you how I did it :

```
#include <CtrlLib/CtrlLib.h>
#include <GLCtrl/GLCtrl.h>
```

```

#include <UltimateOpenGL_V2/UltimateOpenGL.h>
using namespace Upp;

struct OpenGLExample : GLCtrl {
    Vector<float> CubeVertices{
        //Here is a vertice of cube
    };

    UltimateOpenGL_Context context;
    bool isLoaded = false;

    virtual void GLPaint() {
        if(!isLoaded){

            glEnable(GL_DEPTH_TEST);
            glEnable(GL_MULTISAMPLE); // Anti aliasing

            Scene& presentation = context.AddScene("presentation"); //Create Scene
            unsigned int camera = presentation.AddCamera(); //Adding camera to the scene
            Object3D& cube = presentation.CreateGameObject<Object3D>("cube",CubeVertices);
//Adding the cube
            cube.AddTextures("upp","/texture/upp.png",SAMPLE_RGB);
            cube.BindTexture("upp",0.64f);

            cube.GetTransform().SetNewPosition(glm::vec3(0.0f,0.0f,-2.0f)); // move the cube forward the
camera
            presentation.Load(); //Loading the scene

            cube.SetOnDrawFunction([](GameObject& myGameObject){
                myGameObject.GetTransform().RotateFromEulerAngles(0.008f,glm::vec3(1.0f,1.0f,1.0f)); //
rotating of 0.001 degree every frame
            });

            GameObject& sun = presentation.CreateGameObject<GameObject>("sun");

            sun.AddDirLight("lightSun",DirLight(glm::vec3(0.0f,-1.0f,0.0f),glm::vec3( 0.5f, 0.5f,
0.5f),glm::vec3(0.9f, 0.9f, 0.9f) ,glm::vec3(0.5f, 0.5f, 0.5f)));
            sun.GetTransform().SetNewPosition(glm::vec3(0.0f,3.0f,0.0f));

            isLoaded=true;
        }
        glClearColor(0.4f,0.5f,0.8f, 1.0f); //définie la couleur de fond dans la fenetre graphique
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT); //nétoie la fenetre graphique
et la regénère
        context.Draw(); //Drawing the context (active scene)
        Refresh();

    }
}

```

```
};

GUI_APP_MAIN
{
    Ctrl::GlobalBackPaint();
    TopWindow win;
    OpenGLExample gl;
    gl.SetFrame(InsetFrame());
    win.Add(gl.HSizePos(10, 10).VSizePos(10, 10));
    win.Sizeable().Zoomable();
    win.Open();
    win.Run();
}
```

To use glew instead of glad.h I did this in UltimateOpenGL:

```
#ifndef flagUPPGL
#include <glad/glad.h> //Glad is used to init OpenGL
#else
#include <GLCtrl/GLCtrl.h>
#endif
```

If you are wondering why I did an initialisation in the drawing loop and not in window constructor, it's because at window constructor the GL context is not started yet so I can't load and compile thing as shader. Is it a probleme ? I dont know ?

Here is the result :

The gif is really slow compared to the reality

however the scene is ugly compared to test on GLFW.
it seems like AntiAliasing is inexistant and texture seems really blurring at some time.
I dont see from what it could come.

Subject: Re: UltimateOpenGL [BETA] 3D Engine
Posted by [Xemuth](#) on Tue, 19 Nov 2019 20:44:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

koldo wrote on Tue, 19 November 2019 09:06Other detail, totally personal. I would remove "Ultimate" from source files.

I know in other projects (Qt, Kde) they love to put their acronyms everywhere. However in my humble opinion that is ugly :roll:

Done :lol:

Hello !

Here you can find an exemple of UltimateOpenGL working with GLCtrl of Upp :

```
#include <CtrlLib/CtrlLib.h>
#include <GLCtrl/GLCtrl.h>
#include <UltimateOpenGL_V2/UltimateOpenGL.h>
using namespace Upp;

struct OpenGLExample : GLCtrl {
    Point point;

    String TransformFilePath(String FilePath){
        String FilePathBuffer = String(__FILE__);
        String FilePathTemporary = "";
        FilePathBuffer = Replace(FilePathBuffer,Vector<String>{"\\"},Vector<String>{"/"});

        Replace(FilePath,Vector<String>{"\\"},Vector<String>{"/"});
        FilePathTemporary = FilePathBuffer.Left(FilePathBuffer.ReverseFind("/") + FilePath);
        if(!FileExists(FilePathTemporary)){
            return (GetExeFolder() +FilePath);
        }
        return FilePathTemporary;
    }

    Vector<float> CubeVertices{
        -0.2f, -0.2f, -0.2f, 0.0f,0.0f,-0.4f, 0.0f, 0.0f,
        0.2f, -0.2f, -0.2f, 0.0f,0.0f,-0.4f, 1.0f, 0.0f,
        0.2f, 0.2f, -0.2f, 0.0f,0.0f,-0.4f, 1.0f, 1.0f,
        0.2f, 0.2f, -0.2f, 0.0f,0.0f,-0.4f, 1.0f, 1.0f,
        -0.2f, 0.2f, -0.2f, 0.0f,0.0f,-0.4f, 0.0f, 1.0f,
        -0.2f, -0.2f, -0.2f, 0.0f,0.0f,-0.4f, 0.0f, 0.0f,

        -0.2f, -0.2f, 0.2f, 0.0f,0.0f, 0.4f, 0.0f, 0.0f,
        0.2f, -0.2f, 0.2f, 0.0f,0.0f, 0.4f, 1.0f, 0.0f,
        0.2f, 0.2f, 0.2f, 0.0f,0.0f, 0.4f, 1.0f, 1.0f,
        0.2f, 0.2f, 0.2f, 0.0f,0.0f, 0.4f, 1.0f, 1.0f,
        -0.2f, 0.2f, 0.2f, 0.0f,0.0f, 0.4f, 0.0f, 1.0f,
        -0.2f, -0.2f, 0.2f, 0.0f,0.0f, 0.4f, 0.0f, 0.0f,

        -0.2f, 0.2f, 0.2f, -0.4f, 0.0f, 0.0f, 1.0f, 0.0f,
        -0.2f, 0.2f, -0.2f, -0.4f, 0.0f, 0.0f,1.0f, 1.0f,
        -0.2f, -0.2f, -0.2f, -0.4f, 0.0f, 0.0f,0.0f, 1.0f,
        -0.2f, -0.2f, -0.2f, -0.4f, 0.0f, 0.0f,0.0f, 1.0f,
        -0.2f, -0.2f, 0.2f, -0.4f, 0.0f, 0.0f,0.0f, 0.0f,
        -0.2f, 0.2f, 0.2f, -0.4f, 0.0f, 0.0f,1.0f, 0.0f,
```

```

0.2f, 0.2f, 0.2f, 0.4f, 0.0f, 0.0f,1.0f, 0.0f,
0.2f, 0.2f, -0.2f, 0.4f, 0.0f, 0.0f,1.0f, 1.0f,
0.2f, -0.2f, -0.2f, 0.4f, 0.0f, 0.0f,0.0f, 1.0f,
0.2f, -0.2f, -0.2f, 0.4f, 0.0f, 0.0f,0.0f, 1.0f,
0.2f, -0.2f, 0.2f, 0.4f, 0.0f, 0.0f,0.0f, 0.0f,
0.2f, 0.2f, 0.2f, 0.4f, 0.0f, 0.0f,1.0f, 0.0f,

-0.2f, -0.2f, -0.2f, 0.0f, -0.4f, 0.0f,0.0f, 1.0f,
0.2f, -0.2f, -0.2f, 0.0f, -0.4f, 0.0f,1.0f, 1.0f,
0.2f, -0.2f, 0.2f, 0.0f, -0.4f, 0.0f,1.0f, 0.0f,
0.2f, -0.2f, 0.2f, 0.0f, -0.4f, 0.0f,1.0f, 0.0f,
-0.2f, -0.2f, 0.2f, 0.0f, -0.4f, 0.0f,0.0f, 0.0f,
-0.2f, -0.2f, -0.2f, 0.0f, -0.4f, 0.0f,0.0f, 1.0f,

-0.2f, 0.2f, -0.2f, 0.0f, 0.4f, 0.0f,0.0f, 1.0f,
0.2f, 0.2f, -0.2f, 0.0f, 0.4f, 0.0f,1.0f, 1.0f,
0.2f, 0.2f, 0.2f, 0.0f, 0.4f, 0.0f,1.0f, 0.0f,
0.2f, 0.2f, 0.2f, 0.0f, 0.4f, 0.0f, 1.0f, 0.0f,
-0.2f, 0.2f, 0.2f, 0.0f, 0.4f, 0.0f,0.0f, 0.0f,
-0.2f, 0.2f, -0.2f, 0.0f, 0.4f, 0.0f,0.0f, 1.0f
};

UltimateOpenGL_Context context;
bool isLoaded = false;

virtual void GLPaint() {
if(!isLoaded){
glEnable(GL_DEPTH_TEST);
glEnable(GL_MULTISAMPLE);
context.SetScreenSize(GetSize().cx,GetSize().cy);
Scene& presentation = context.AddScene("presentation"); //Create Scene
unsigned int camera = presentation.AddCamera(); //Adding camera to the scene
Object3D& cube = presentation.CreateGameObject<Object3D>("cube",CubeVertices);
//Adding the cube
context.AddTextures("upp",TransformFilePath("/texture/upp.png"),SAMPLE_RGB);
cube.BindTexture("upp",0.64f);

cube.GetTransform().SetNewPosition(glm::vec3(0.0f,0.0f,-1.0f)); // move the cube forward the
camera
presentation.Load(); //Loading the scene

cube.SetOnDrawFunction([](GameObject& myGameObject){
myGameObject.GetTransform(). RotateFromEulerAngles(0.5*
myGameObject.GetScene()->GetContext()->GetDeltaTime(),glm::vec3(1.0f,1.0f,1.0f)); // rotating
of 0.001 degree every frame1
});
};

```

```

GameObject& sun = presentation.CreateGameObject<GameObject>("sun");

sun.AddDirLight("lightSun",DirLight(glm::vec3(0.0f,-1.0f,0.0f),glm::vec3( 0.5f, 0.5f,
0.5f),glm::vec3(0.9f, 0.9f, 0.9f) ,glm::vec3(0.5f, 0.5f, 0.5f)));
sun.GetTransform().SetNewPosition(glm::vec3(0.0f,3.0f,0.0f));

isLoaded=true;
}
this->GetTopWindow()->Title("UltimateOpenGL UPP GLCtrl - " + AsString(context.GetFPS()) + "
FPS");

context.Draw(); //Drawing the context (active scene)
Refresh();

glEnd();
}
};

GUI_APP_MAIN
{
Ctrl::GlobalBackPaint();
TopWindow win;
OpenGLExample gl;
gl.SetFrame(InsetFrame());
win.Add(gl.HSizePos(10, 10).VSizePos(10, 10));
win.NoSizeable();

win.SetRect(0, 0, 320, 320);
win.Open();
win.Run();
}

```

Here you can find package with texture and .exe :

File Attachments

1) [UltimateOpenGL and Upp GLCTRL.7z](#), downloaded 340 times

Subject: Re: UltimateOpenGL [BETA] 3D engine
Posted by [Xemuth](#) on Mon, 02 Dec 2019 13:56:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Update : Model Loading is now working. However still have some Texture problemes.
The Global repo for the project is now UltimateOpenGL_V3 :
https://github.com/Xemuth/UltimateOpenGL_V3

Code is now much cleaner and better architected

Subject: Re: UltimateOpenGL [BETA] 3D engine
Posted by [Xemuth](#) on Tue, 03 Dec 2019 22:42:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

Here is some screenshot of UltimateOpenGL loading 3D model:

You can also find a gif of the model rotating here : <https://i.imgur.com/QYLhb2J.mp4>
(The framerate of the gif is not the same as reality)

How to do it :

```
/**All basic code to set Up glfw is coming before see my exemple or Readme of my
GITHUB**//
context.AddTexture("sand",TransformFilePath("/Textures/sand.jpg")); //Load sand texture

Scene& myScene = context.AddScene("main");
myScene.AddCamera("main");
myScene.SetBackgroundColor(context.TransformRGBToFloatColor(40,180,200));

Mesh m; //Use to simulate the flat float under the model
m.GetVertices().Add().SetPosition(glm::vec3(-20.0f, 0.0f, -20.0f)).SetTexCoords(glm::vec2(0.0f,
20.0f));
m.GetVertices().Add().SetPosition(glm::vec3(20.0f, 0.0f,
-20.0f)).SetTexCoords(glm::vec2(20.0f, 20.0f));
m.GetVertices().Add().SetPosition(glm::vec3(20.0f, 0.0f, 20.0f)).SetTexCoords(glm::vec2(20.0f,
0.0f));
m.GetVertices().Add().SetPosition(glm::vec3(20.0f, 0.0f, 20.0f)).SetTexCoords(glm::vec2(20.0f,
0.0f));
m.GetVertices().Add().SetPosition(glm::vec3(-20.0f, 0.0f, 20.0f)).SetTexCoords(glm::vec2(0.0f,
0.0));
m.GetVertices().Add().SetPosition(glm::vec3(-20.0f, -0.0f,
-20.0f)).SetTexCoords(glm::vec2(0.0f, 20.0f));

Object3D& floor = myScene.CreateGameObject<Object3D>("floor",m); //Lets create floor
floor.BindTexture("sand");

Object3D& modele = myScene.CreateGameObject<Object3D>("modele");
modele.LoadModel("C:\\Upp\\myapps\\ExempleUltimateOpenGL_V3\\obj upp\\upp.obj");
//Loading of model

modele.GetTransform().SetNewPosition(glm::vec3(0,4,0)); //Set new position
modele.GetTransform().ScaleNewValue(glm::vec3(0.05f,0.05f,0.05f)); //Scale the model
```

```

modele.SetOnDrawFunction([](GameObject& gm){ //Bind event on draw
    double rotation = glm::cos(context.GetEllapsedTime())/100;
    gm.GetTransform().RotateFromEulerAngles(context.GetDeltaTime() * 2,glm::vec3(0,1,0) );
});

myScene.Load();
    while(!glfwWindowShouldClose(window)) {
        processInput(window);

    glfwSetWindowTitle(window, "UltimateOpenGL V3 - " +AsString(context.GetFPS()) +" FPS");
    try{
        context.Draw(); //Draw the context
    }catch(UGLException& e){
        LOG(e.what());
    }

    glfwSwapBuffers(window);
    glfwPollEvents();
}
glfwTerminate();

```

As you can see it's very simple !

Next step before digging the physic is to impletement animation and FBX support.
If you are interesting in 3D model of UPP Symbol, ask me !

Subject: Re: UltimateOpenGL [BETA] 3D engine
Posted by [koldo](#) on Thu, 02 Jan 2020 17:16:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

assimp package is up and running in Bazaar. This implies an automatic install of UltimateOpenGL dependencies (all documentation will be changed dramatically by the author soon).

Probably we will see soon a simple and strong UltimateOpenGL demo to be used just out of the box.

I send my thanks to Xemuth.

Subject: Re: UltimateOpenGL [BETA] 3D engine
Posted by [Xemuth](#) on Thu, 02 Jan 2020 17:22:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

That's a really great new ! Thanks a lot Koldo !
