

程序代写代做 CS编程辅导

CMT100 Visual Computing



I.4 Introduction to OpenGL

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

Xianfang Sun

QQ: 749389476

<https://tutorcs.com>

School of Computer Science & Informatics

Cardiff University

Overview

➤ Introduction to OpenGL

- What is OpenGL
- OpenGL History
- OpenGL Pipeline
- OpenGL Compc
- Java OpenGL (Jogl)

程序代写代做 CS编程辅导



WeChat: cstutorcs

- Installation of JOGL on Eclipse

➤ OpenGL Programming

Assignment Project Exam Help

- Basic OpenGL Coding Framework
- OpenGL Geometric Primitives
- A Simple OpenGL Program

Email: tutorcs@163.com

QQ: 749389476

- In C
- In Java

<https://tutorcs.com>

What is OpenGL?

- **OpenGL**: Open Graphics Library
 - Originally IRIS GL (Integrated Raster Imaging System Graphics Library) by Silicon Graphics
- OpenGL is **NOT** a **library**, it is
 - a **software interface** to graphics hardware
 - a graphics **programming library**
 - a **standard** for 3D graphics
- At the lowest possible level it still allows device independence
 - OpenGL is partly implemented in software and partly in hardware depending on the device
 - No high-level modelling operations, etc.

程序代写代做CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

OpenGL History

OpenGL Release	GLSL Release	Year	Features
1.0	---	1992	Fixed-function Pipeline
1.1 ~ 1.5	---	1997 ~ 2003	
2.0	1.10	2004	vertex shaders and fragment shaders
2.1	1.20	2006	
3.0 ~ 3.2	1.30 ~ 1.50	2008 ~ 2009	Deprecated features; Geometry shaders from 3.2.
3.3	3.30	2010	
4.0, 4.1	4.00, 4.10	2010	Tessellation shaders
4.2	4.20	2011	
4.3	4.30	2012	Compute shaders
4.4	4.40	2013	
4.5	4.50	2014	
4.6	4.60	2017	

程序代写代做 CS编程辅导



WeChat: cstutorcs

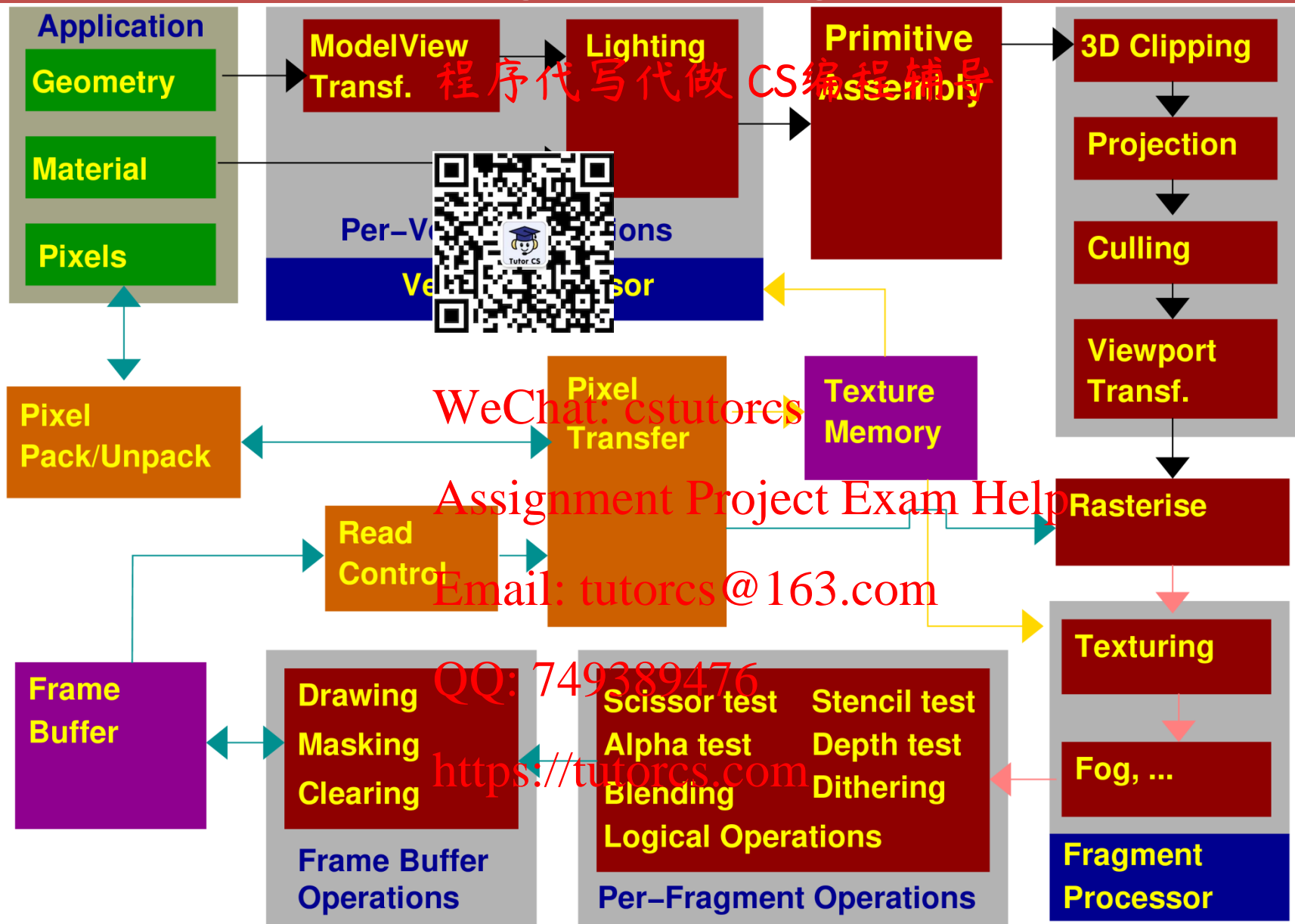
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

The OpenGL Pipeline



The OpenGL Pipeline (Ver < 2.0)

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

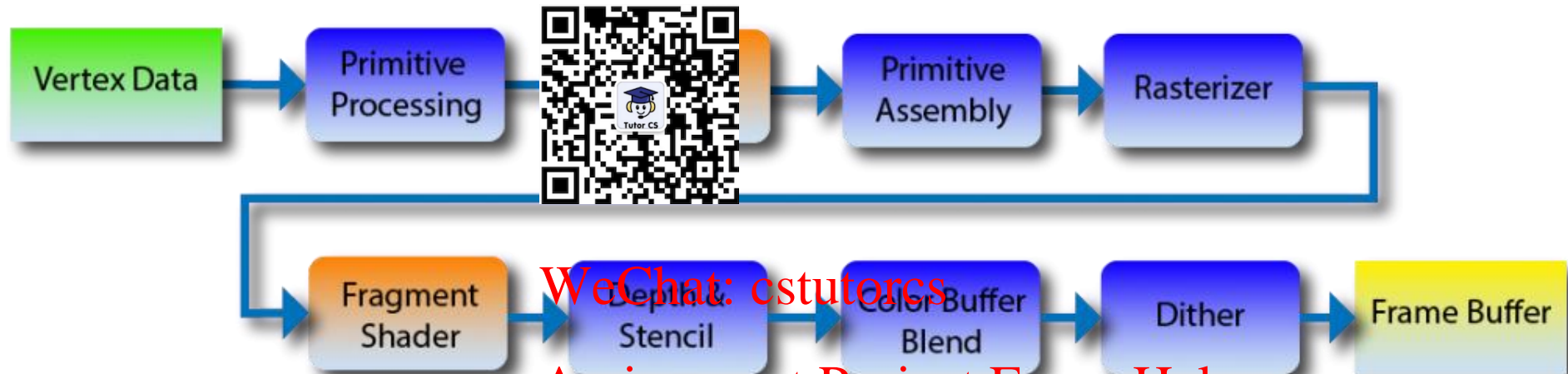
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

The OpenGL Pipeline (Ver = 2.0)

程序代写代做 CS编程辅导



WeChat: cstutorcs
Assignment Project Exam Help

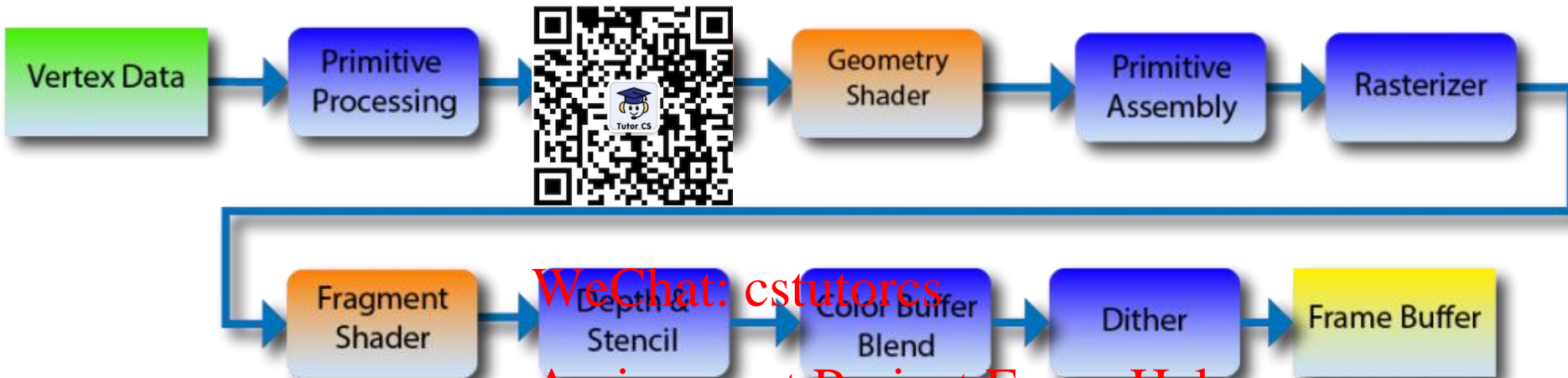
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

The OpenGL Pipeline (Ver = 3.2)

程序代写代做 CS编程辅导



WeChat: cstutores

Assignment Project Exam Help

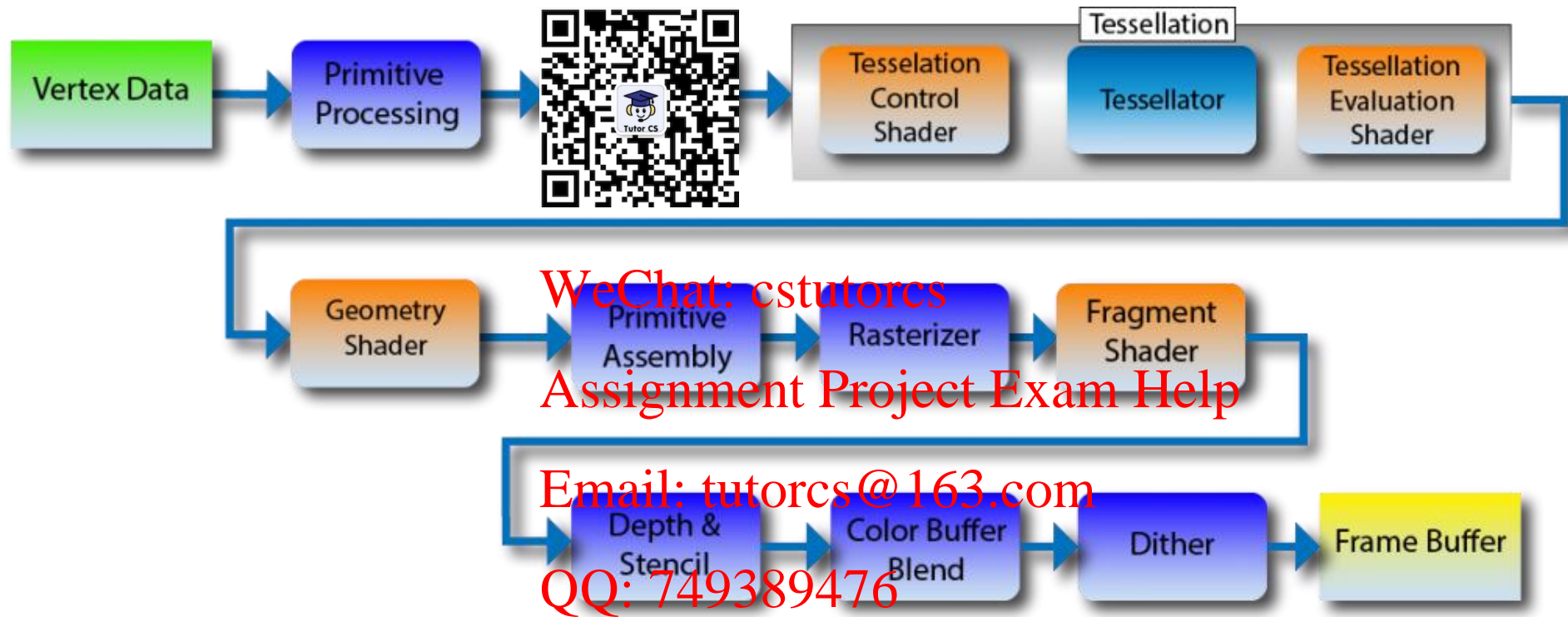
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

The OpenGL Pipeline (Ver = 4.0)

程序代写代做 CS编程辅导



WeChat: cstutores
Assignment Project Exam Help

Email: tutorcs@163.com

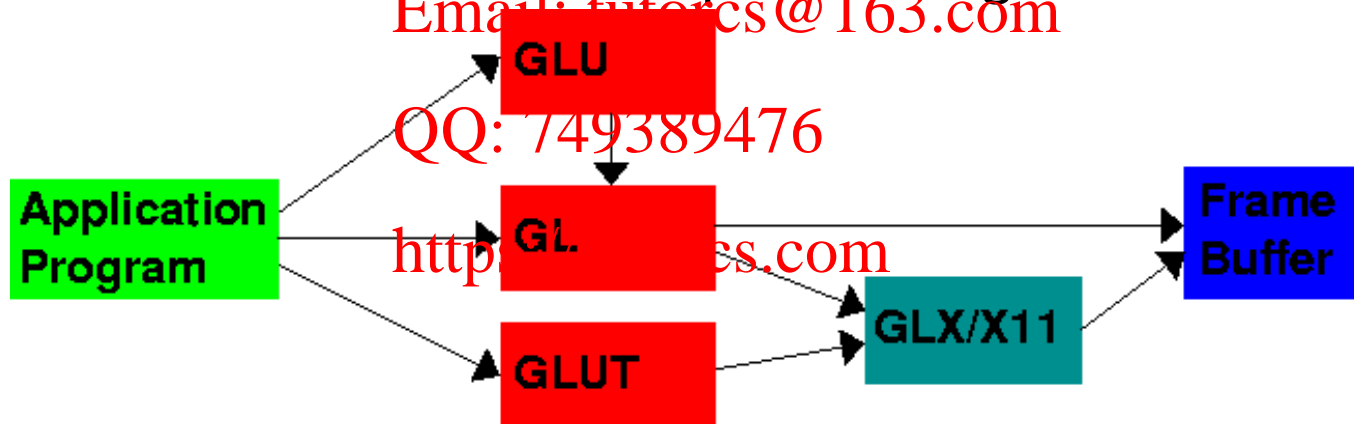
QQ: 749389476

<https://tutorcs.com>


OpenGL Components

➤ Components of the OpenGL interface:

- GL: core OpenGL functions
- GLU: graphics utility library
(a variety of graphics utility functions, e.g. gluLookAt)
- GLUT: OpenGL toolkit
(interface to windowing system via xlib; alternatives: glib+GTK, QT; helpers for creating common objects, e.g. spheres, the teapot)
- GLX: low-level interface to X11
(different interfaces for other platforms: glw for windows)



Java OpenGL (JOGL)

- **Java OpenGL (JOGL)** is a wrapper library that allows OpenGL to be used in the Java programming.
程序代写代做 CS编程辅导
- JOGL 1.1.1 gives full access to the APIs in the **OpenGL 2.0** specification and full access to **GLU NURBS**, providing rendering of curves and surfaces via the traditional GLU APIs.
WeChat: cstutorcs

- JOGL 2.0 provides full access to the APIs in the **OpenGL 1.3 - 3.0, 3.1 - 3.3, ≥ 4.0, ES 1.x and ES 2.x** specification as well as nearly all vendor extensions.
Assignment Project Exam Help
Email: tutorcs@163.com
- Newest version (2.3.2) of JOGL can be downloaded from
QQ: 749389476
<http://jogamp.org/deployment/jogamp-current/archive/>
<https://tutorcs.com>

Installation of Jogl on IntelliJ

- Download and install IntelliJ.
- Download and install the latest Jogl api.
- Set up Jogl as a user library.
- Configure Jogl library for each OpenGL (Jogl) project.



- All downloads and installation instructions are free available from related official sites.
- More detail about installation can be found in the file available from learningcentral.

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Basic OpenGL Coding Framework

- **Configure OpenGL**
 - Create window, Display mode
- **OpenGL state initialization**
 - Set background color, View positions,
 - Compile and link shader programs
- **Set up Display Function**
 - Render the scene
- **Set up Reshape Function**
 - resize the view window and recompute projection matrices
- **Process Event loop**

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

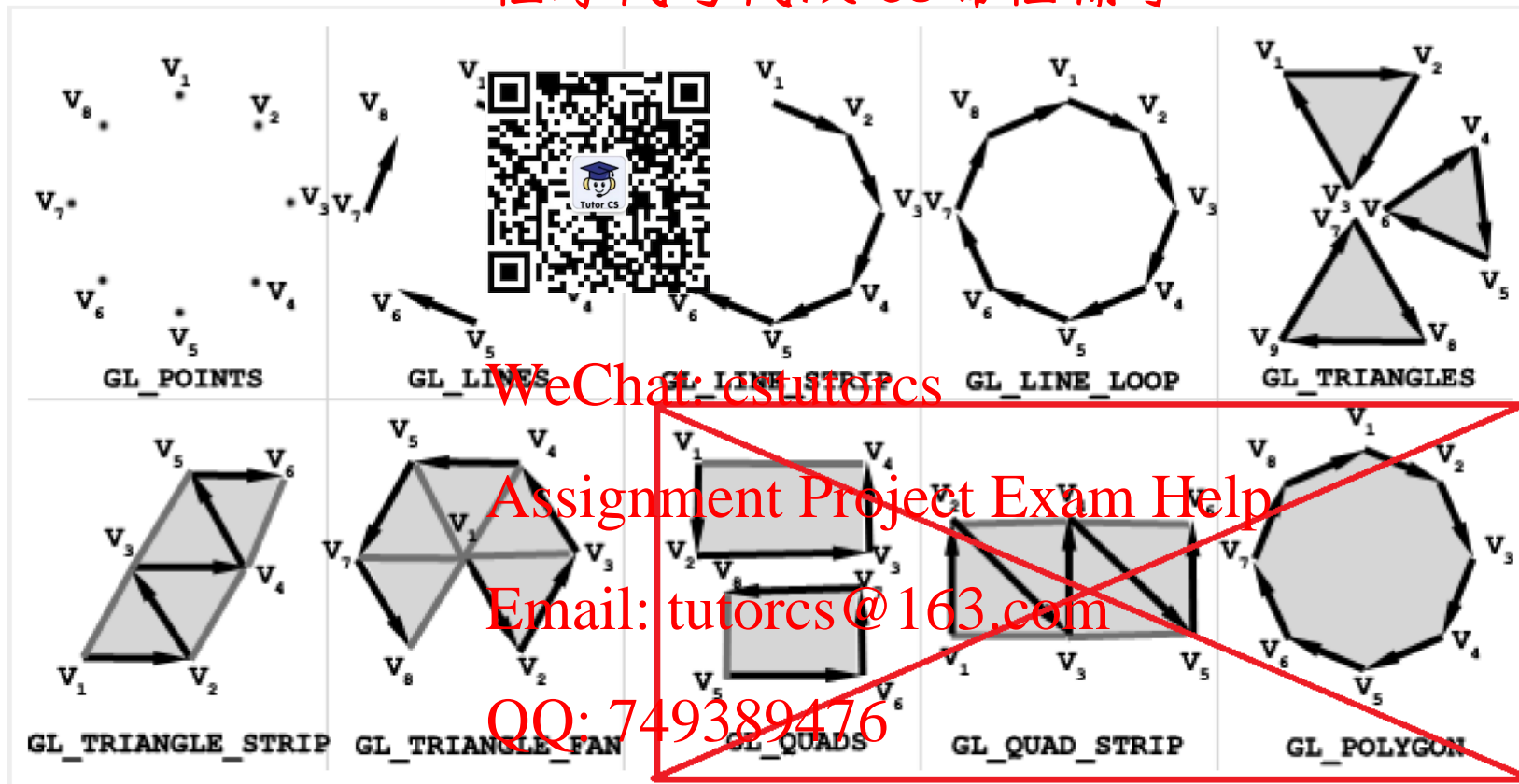
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

OpenGL Geometric Primitives

程序代写代做 CS编程辅导



<https://tutorcs.com>

A Simple OpenGL C Program (1)


```
#include <GL/glew.h>
#include <GL/freeglut.h>
// Define: number of Vertex Array Objects,
// number of Vertex Buffer Objects,
// number of Vert...

const GLuint numVAOs = 1;
const GLuint numVBos = 1;

// Specify the ids of points, buffers,
// and the vertex attribute position
// in the vertex shader program.
GLuint idPoint = 0, idBuffer = 0;
GLuint vPosition = 0;

// Declare VAOs and VBos
GLuint VAOs[numVAOs];
GLuint VBos[numVBos];
```

程序代写代做 CS编程辅导



WeChat: estutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

A Simple OpenGL C Program (2)

```
// Define: Vertex shader program, and Fragment shader program
```

```
const GLchar* srcVShader =
```

```
    "#version 330 core\n"
```

```
    "layout(location = 0) in vec4 vPosition;"
```

```
    "void main"
```

```
    "{"
```

```
    "
```

```
    "};";
```



```
    = vPosition;"
```

```
const GLchar* srcFShader =
```

```
    "#version 330 core\n"
```

```
    "out vec4 fColor;"
```

```
    "void main() "
```

```
    "{"
```

```
    "
```

```
    "};";
```

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

A Simple OpenGL C Program (3)

```
void init(void) // initialisation
{
    //Define vertices coordinates
    GLfloat vertices[numVertices][2] = {
        {0.0f, 0.0f},
        {0.0f, 1.0f},
        {1.0f, 1.0f},
        {1.0f, 0.0f}
    };

    //Generate vertex array objects (VAOs), and
    //Bind a VAO, i.e., initialise this VAO.
    // A second binding is needed later to use it
    glGenVertexArrays(1, &vao);
    glBindVertexArray(vao);

    //Generate vertex buffer objects (VBOs), and
    //Bind a VBO, i.e., initialise this VBO.
    glGenBuffers(1, &vbbo);
    glBindBuffer(GL_ARRAY_BUFFER, vbbo);
    //The Data is then pooled into the buffer
    glBufferData(GL_ARRAY_BUFFER, sizeof(vertices),
                 vertices, GL_STATIC_DRAW);

    //Specify the location and data format of the
    //array of vertex attributes for rendering
    glVertexAttribPointer(vPosition, 2, GL_FLOAT,
                          GL_FALSE, 0, (void*)(0));
    glEnableVertexAttribArray(vPosition);
}
```



A Simple OpenGL C Program (4)

```
//Create a shader program
GLuint program = glCreateProgram();
```

```
//Compile and link vertex shader
//into the program
GLuint shader = glCreateShader(GL_VERTEX_SHADER);
glShaderSource(shader, 1, &srcVShader, NULL);
glCompileShader(shader);
glAttachShader(program, shader);
glDeleteShader(shader);
```

```
//Compile and attach fragment shader
//into the program
shader = glCreateShader(GL_FRAGMENT_SHADER);
glShaderSource(shader, 1, &srcFShader, NULL);
glCompileShader(shader);
glAttachShader(program, shader);
glDeleteShader(shader);
```

```
//Link and use the shader program
glLinkProgram(program);
glUseProgram(program);
```

```
}
```

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutores@163.com

QQ: 749389476

<https://tutores.com>

A Simple OpenGL C Program (5)

```
// display the scene
void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);

    glPointSize(5);

    //Bind VAO again
    glBindVertexArray(VAOs[idPoint]);
    glDrawArrays(GL_POINTS, 0, numVertices);

    glutSwapBuffers();
}
```

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

A Simple OpenGL C Program (6)

```
int main(int argc, char** argv) {  
    // Initialise GLUT  
    glutInit(&argc, argv);  
    glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE );  
    glutInitWindowSize(512);  
    // Create display  
    glutCreateWindow("");  
    // OpenGL Version profile  
    glutInitContextVersion(3, 3);  
    glutInitContextProfile(GLUT_CORE_PROFILE);  
  
    // Deal with OpenGL extensions issues  
    glewExperimental = GL_TRUE;  
    if( GLEW_OK != glewInit() )  
        exit(EXIT_FAILURE);  
  
    init();  
  
    glutDisplayFunc(display);  
    glutReshapeFunc(reshape);  
    glutMainLoop(); // Start GLUT event loop  
    return 0;  
}
```

程序代写代做 CS编程辅导



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

A Simple OpenGL Java Program (1)

```
// Import some packages
```

```
import java.nio.FloatBuffer;
```

```
import com.jogamp.nativewindow.WindowClosingProtocol;
```

```
import com.jogamp.opengl.GLWindow;
```

```
import com.jogamp.opengl.GLAutoDrawable;
```

```
import com.jogamp.opengl.GLCapabilities;
```

```
import com.jogamp.opengl.GLEventListener;
```

```
import com.jogamp.opengl.GLProfile;
```

```
import com.jogamp.opengl.util.FPSAnimator;
```

```
// Import GL constant
```

```
import static com.jogamp.opengl.GL3.*;
```

程序代写代做 CS编程辅导



WeChat: cstutors

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

A Simple OpenGL Java Program (2)

```
public class Simple implements GLEventListener {  
    private GLWindow window; // Declare a window  
    final FPSAnimator animator=new FPSAnimator(60, true);  
    // Define: number of Vertex Array Objects,  
    // number of Buffer Objects,  
    // number of VAOs  
    // Specify number of points, buffers,  
    // and the attribute position  
    // in the vertex shader program  
    private int idPoint = 0, numVAOs = 1;  
    private int idBuffer = 0, numVBOS = 1;  
    private int vPosition = 0;  
  
    private final int numVertices = 1;  
  
    // Declare VAOs and VBOS  
    private int[] VAOs = new int[numVAOs];  
    private int[] VBOS = new int[numVBOS];
```



程序代写代做 CS编程辅导

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

A Simple OpenGL Java Program (3)

```
private String[] srcVShader =  
    { "#version 330 core\n"  
      + "layout(location = 0) in vec4 vPosition;\n"  
      + "void main() "\br/>      + "    _Position = vPosition;\n"  
      + "}"  
    };  
  
private String[] srcFShader =  
    { "#version 330 core\n"  
      + "out vec4 fColor;\n"  
      + "void main() "\br/>      + "{\n"  
      + "    fColor = vec4(1.0, 0.0, 0.0, 1.0);\n"  
      + "}"  
    };
```

程序代写代做CS编程辅导



WeChat: estutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

A Simple OpenGL Java Program (4)

```
public Simple() {  
    GLProfile glp = GLProfile.getDefault();  
    GLCapabilities caps = new GLCapabilities(glp);  
    window = GLWindow.create(caps);  
    window.addKeyListener(this);  
    window.setDefaultCloseOperation(  
        WindowClosingProtocol.WindowClosingMode.  
            DISPOSE_ON_CLOSE); //Exit when click close  
    window.setSize(500, 500); // set the window size  
    window.setTitle("Simple Graphics"); // window title  
    window.setVisible(true); // Display the frame  
    animator.start();  
}
```



WeChat: estutores

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

A Simple OpenGL Java Program (5)

```
public void init(GLAutoDrawable drawable) {  
    // Get the GL implementation object  
    GL3 gl = drawable.getGL().getGL3();  
  
    //Define coordinates  
    float[] vertices = { 0.0f, 0.0f };  
    //wrap the vertex array into a FloatBuffer.  
    FloatBuffer vbos = FloatBuffer.wrap(vertices);  
  
    // Generate vertex array objects (VAOs), and  
    // Bind a VAO, i.e., initialise this VAO.  
    // A second binding is needed later to use it  
    gl.glGenVertexArrays(1, vaoIds, 0);  
    gl.glBindVertexArray(vaoIds[idPoint]);  
  
    // Generate vertex buffer objects (VBOs), and  
    // Bind a VBO, i.e., initialise this VBO.  
    // The Data is then pooled into the buffer  
    gl.glGenBuffers(1, vbos, 0);  
    gl.glBindBuffer(GL_ARRAY_BUFFER, vbos[idBuffer]);  
    gl.glBufferData(GL_ARRAY_BUFFER, vertices.length *  
        (Float.SIZE / 8), vertices, GL_STATIC_DRAW);  
  
    gl.glVertexAttribPointer(vPosition, 2, GL_FLOAT, false, 0, 0L);  
    gl.glEnableVertexAttribArray(vPosition);  
}
```



程序代写代做 CS编程辅导

WeChat: cstutores

Assignment Project Exam Help

Email: tutores@163.com

QQ: 749389476

https://tutores.com

A Simple OpenGL Java Program (6)

```
// Create a shader program
```

```
int program = gl.glCreateProgram();
```

```
// Compile and attach vertex shader into the program
```

```
int shader = gl.glCreateShader(GL_VERTEX_SHADER);
```

```
gl.glShaderSource(shader, 1, srcVShader, null);
```

```
gl.glCompileShader(shader);
```

```
gl.glAttachShader(program, shader);
```

```
gl.glDeleteShader(shader);
```

```
// Compile and attach fragment shader into the program
```

```
shader = gl.glCreateShader(GL_FRAGMENT_SHADER);
```

```
gl.glShaderSource(shader, 1, srcFShader, null);
```

```
gl.glCompileShader(shader);
```

```
gl.glAttachShader(program, shader);
```

```
gl.glDeleteShader(shader);
```

```
// Link and use the shader program
```

```
gl.glLinkProgram(program);
```

```
gl.glUseProgram(program);
```

```
}
```



WeChat: [tutorcs](https://tutorcs.com)

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

A Simple OpenGL Java Program (7)

```
public void display(GLAutoDrawable drawable) {  
    GL3 gl = drawable.getContext().getGL();  
  
    gl.glClear(GL_COLOR_BUFFER_BIT);  
    gl.glPointSize(POINTS, 0, numVertices);  
  
    gl.glDraw  
}
```



```
public void reshape(GLAutoDrawable drawable, int x, int y,  
    int width, int height) {  
}
```

```
public void dispose(GLAutoDrawable drawable) {  
    System.exit(0);  
}
```

```
public static void main(String[] args) {  
    new Simple();  
}
```

程序代写代做 CS编程辅导

WeChat: cstutores

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Summary

- What is the underlying model for the OpenGL library?
 - What are the components of OpenGL?
- Basic OpenGL programming with C++ or Java.
 - Describe the OpenGL programming framework.



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>