Ονοματεπώνυμο	Νικόλαος Γέροντας
Αριθμός Μητρώου	1092813

Περιεχόμενα

Lab 01: Basics

Task 1 και Task 2

Construct a vertex buffer object that will contain 3 different colors. Assign this VBO attribute "1". Extend the vertex shader to accept the new attribute:

```
layout(location=1) in vec3 vertexColor;
```

Add an output variable:

```
out vec3 color;
```

Propagate the vertexColor to the output (in main):

```
// task propagate color to fragment shader
color = vertexColor;
```

Inside the fragment shader add an input variable that will accept the color from the vertex shader and assign the input color to the fragmentColor:

```
in vec3 color;
```

Use the following command and test the two modes.

```
// Draw wire frame triangles or fill: GL_LINE, or GL_FILL
glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
```

Απάντηση:

✓ Ολοκληρώθηκαν κατά την διάρκεια του εργαστηρίου.

Task 3

Try drawing with GL_LINE_STRIP or GL_LINES or GL_POINTS instead of triangles. Are the lines drawn as you expect? How big are the points by default?

Hint: Make use of:

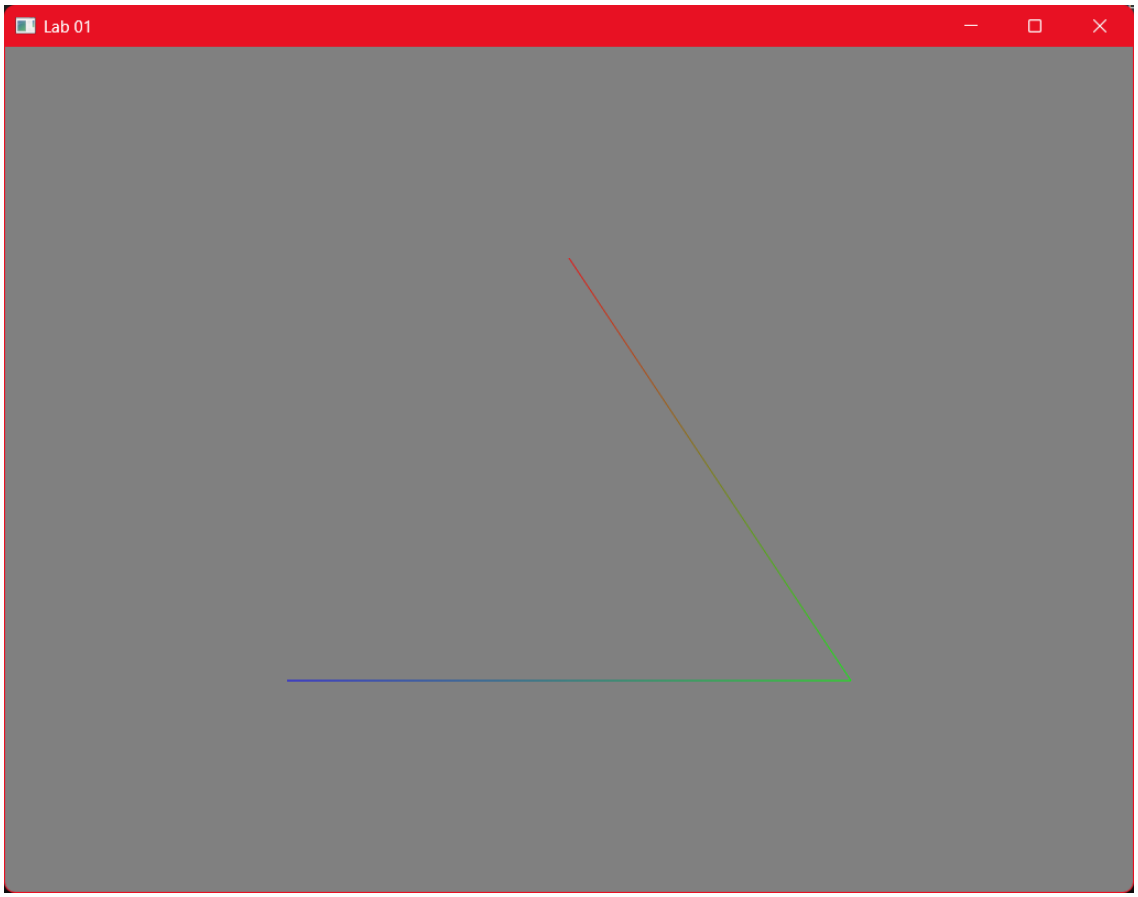
```
glDrawArrays(GL_POINTS, 0, 3);

glEnable(GL_PROGRAM_POINT_SIZE);

// point size in vertex shader (reserved attribute)
gl_PointSize = 10;
```

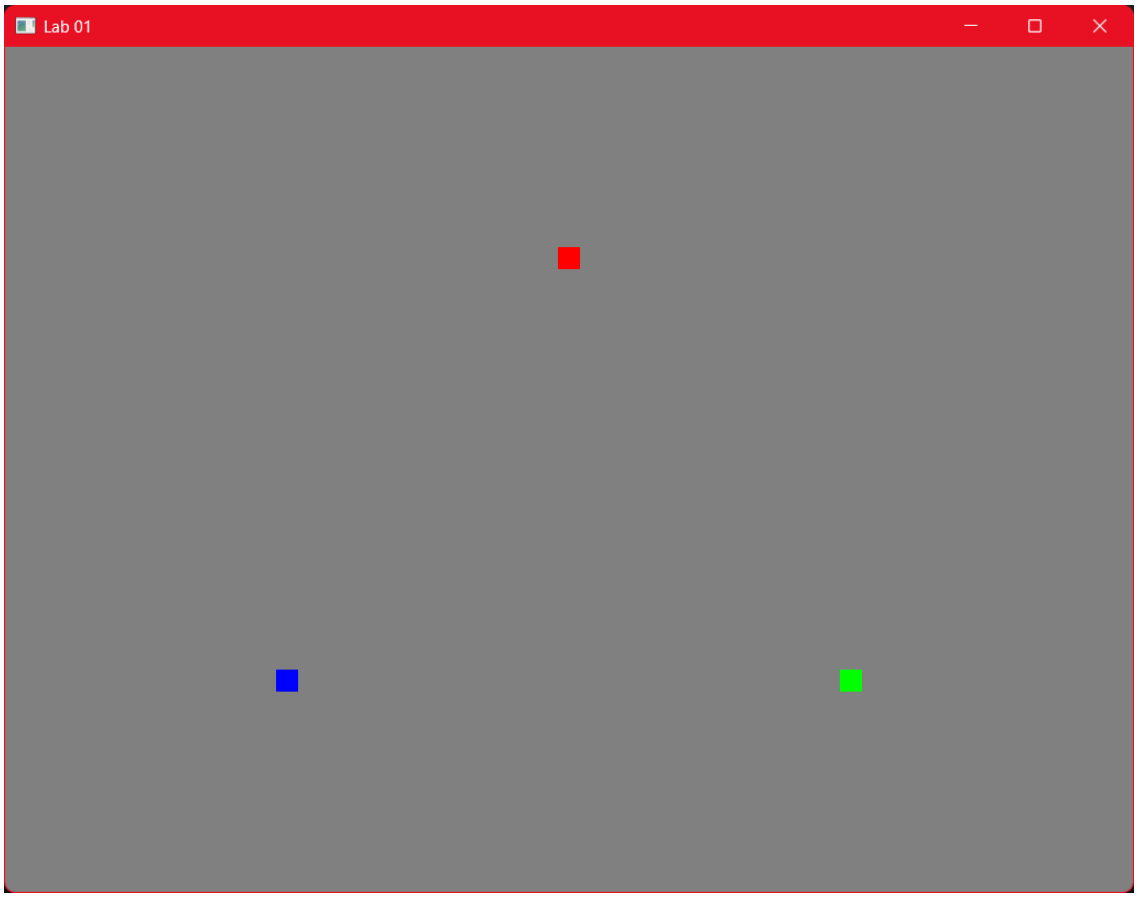
Απάντηση:

Αποτέλεσμα εκτέλεσης με **όρισμα GL_LINE_STRIP**:



Αποτέλεσμα εκτέλεσης με **όρισμα GL_LINES**:

Αποτέλεσμα εκτέλεσης με **όρισμα GL_POINTS** (και gl_PointSize = 20 ώστε να είναι ορατό το αποτέλεσμα):



Βάση του OpenGL Wiki (συγκεκριμένα: https://wikis.khronos.org/opengl/Primitive), έχουμε:

GL_LINES	Vertices 0 and 1 are considered a line. Vertices 2 and 3 are considered a line. And so on. If the user specifies a non-even number of vertices, then the extra vertex is ignored.
GL_LINE_STRIP	The adjacent vertices are considered lines. Thus, if you pass n vertices, you will get n-1 lines. If the user only specifies 1 vertex, the drawing command is ignored.

Έτσι εξηγείται η εμφάνιση μόνο των δύο εκ των τριών αρχικών γραμμών (n = 3 vertices, άρα παίρνουμε 2 lines) κατά την χρήση του ορίσματος GL_LINE_STRIP.

Επίσης, με όρισμα το GL_LINES εμφανίζεται μόνο μία γραμμή, αυτή που συνδέει τα vertices 0 και 1 (καθώς το vertex 2 αγνοείται).

Παράλληλα, με το primitive GL_POINTS ως όρισμα, η OpenGL ερμηνεύει την κάθε ξεχωριστή κορυφή στο stream ως ένα σημείο.

Τέλος, **η τιμή της μεταβλητής gl_PointSize είναι undefined**, έως ότου οριστεί. Συγκεριμένα, βάση των δοκιμών μου με if - else λογική (κατέληξα στο ίδιο συμπέρασμα: Impossible to debug GLSL shaders) και της υλοποίησης/συμπεριφοράς του undefined του συστήματός μου, η μεταβλητή gl_PointSize δεν έχει αρχικοποιημένη θετική τιμή.

https://registry.khronos.org/OpenGL-Refpages/es3.1/html/gl_PointSize.xhtml

Task 4: Color modulation

Go back to drawing triangles instead of points. Add the following line to the bottom of the main() function inside the fragment shader:

```
fragmentColor *= vec3(abs(cos(length(color)*100.0)));
```

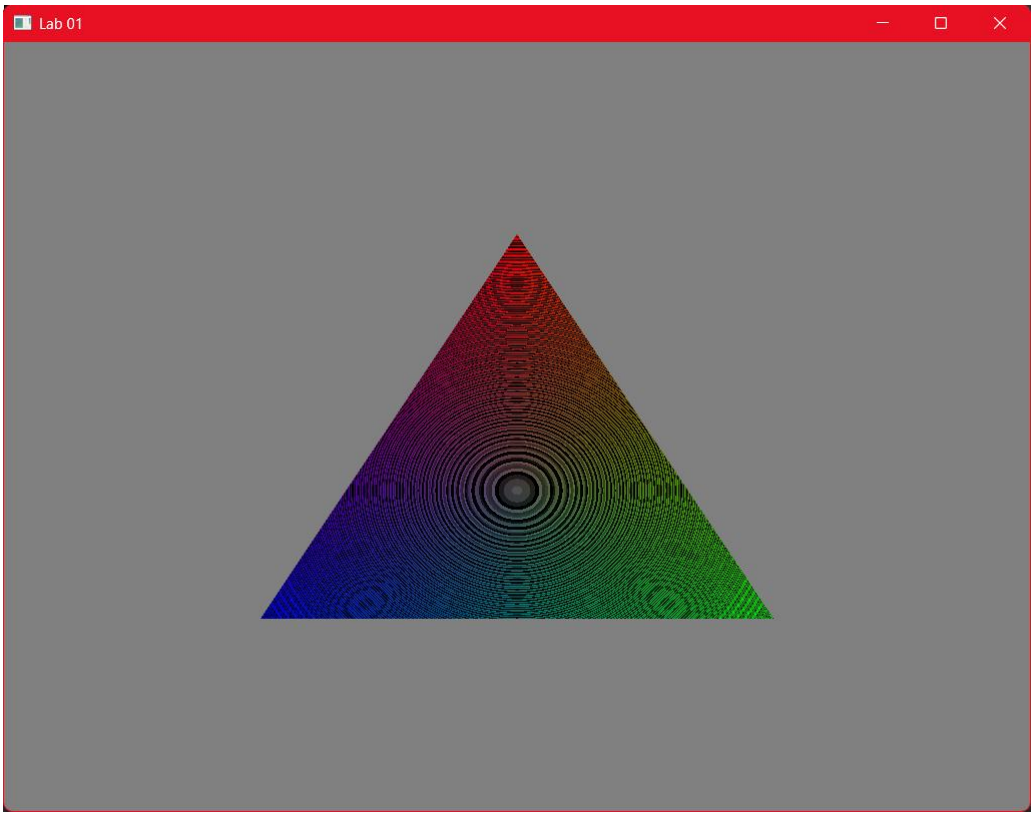
Replace the factor “100.0” with a much larger number. Try different values. What happens when the factor changes?

Hints:

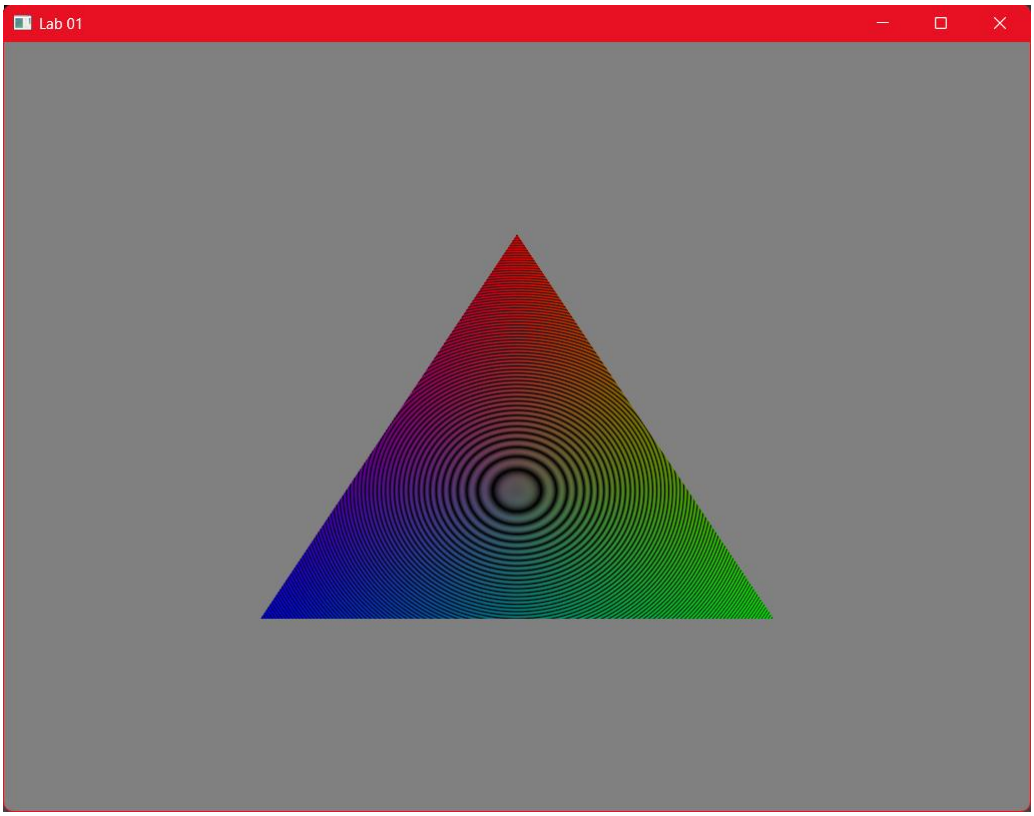
- The *length* function creates the circular shapes (because the locus of all points having the same distance to a constant point is a circle).
- The *cos* function creates a periodic sinusoidal behavior featuring increase followed by decrease in color.
- The *abs* function shifts the values of the cosine from [-1, 1] to [0, 1], as color is bound by those values.

Απάντηση:

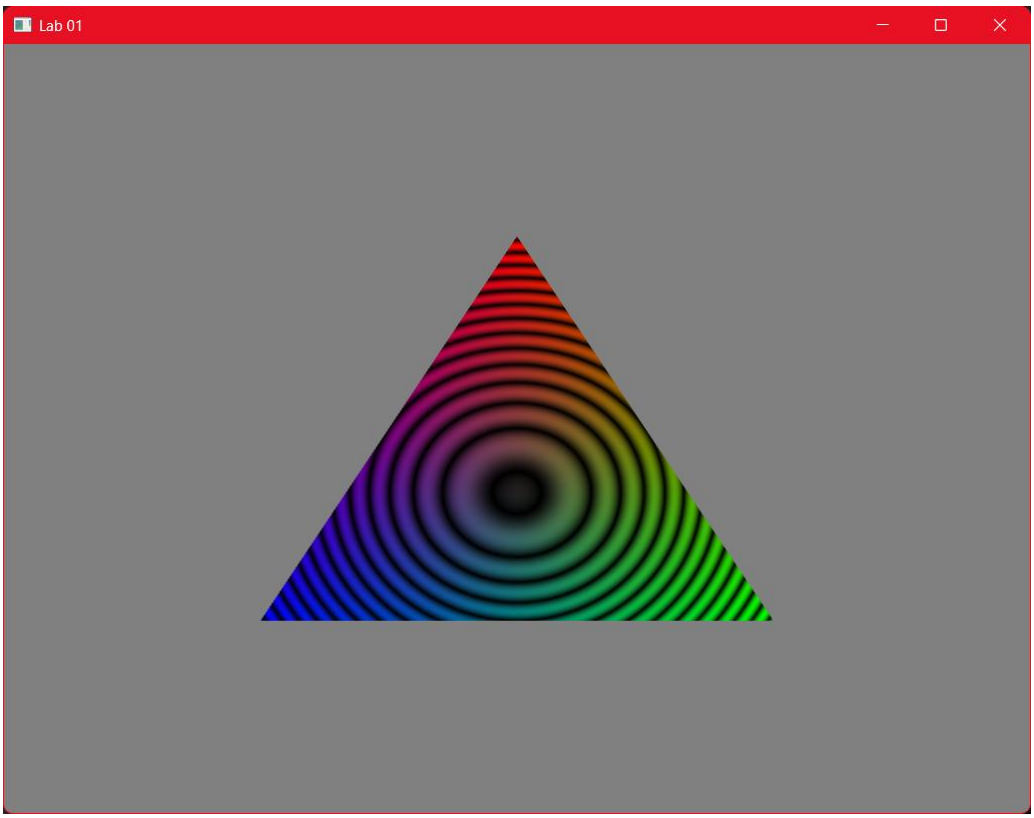
Αποτέλεσμα για **συντελεστή = 10^8**:



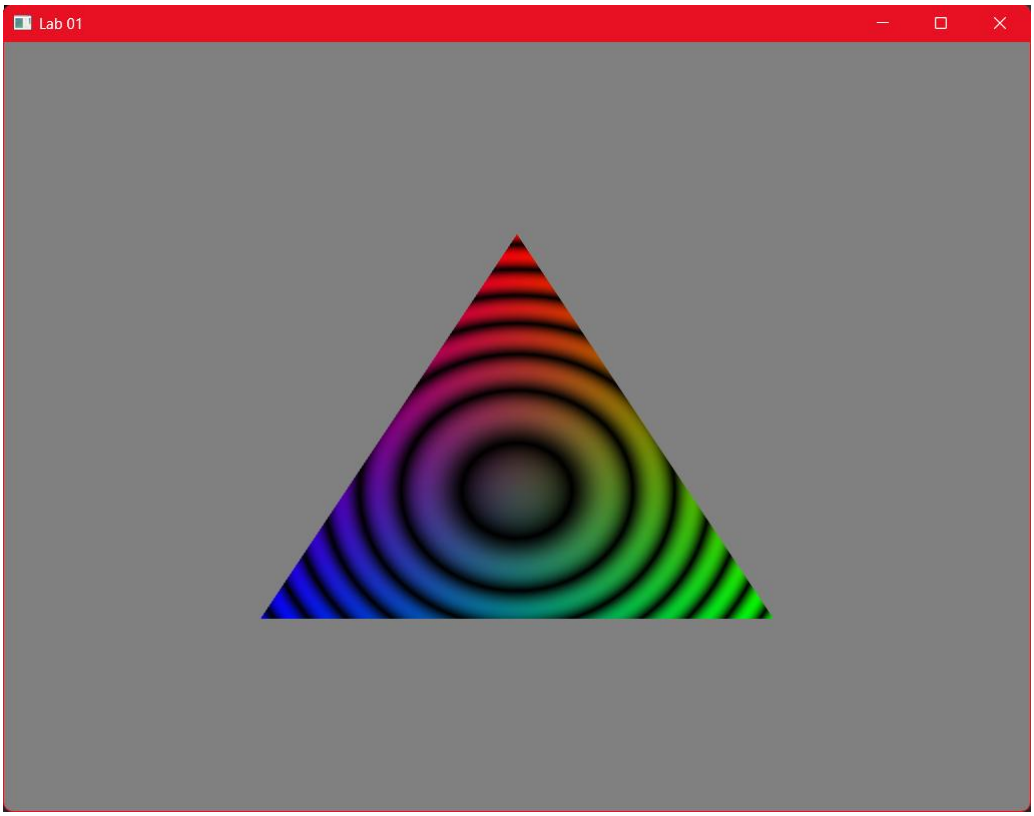
Αποτέλεσμα για **συντελεστή = 500.0**:



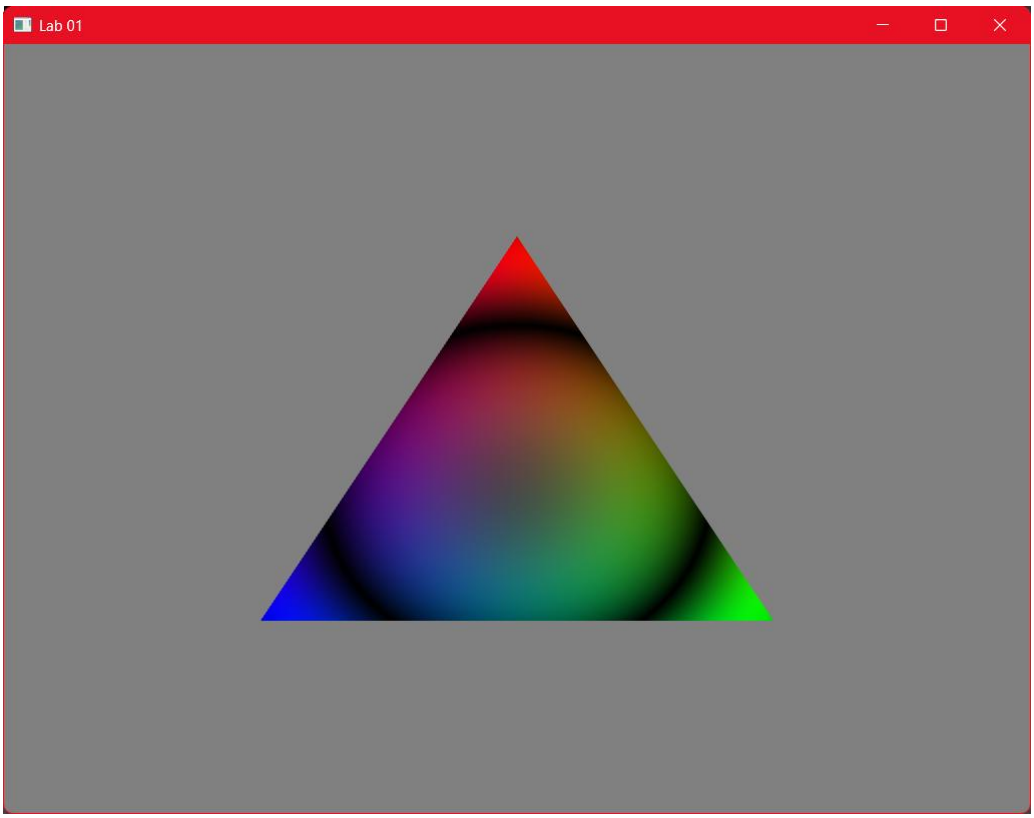
Αποτέλεσμα για **συντελεστή = 100.0**:



Αποτέλεσμα για **συντελεστή = 50.0**:



Αποτέλεσμα για **συντελεστή = 10.0**:



Παρατηρώ ότι, όταν αυξάνεται ο συντελεστής, το πλήθος των κυκλικών ζωνών αυξάνεται και αυτές γίνονται πιο πυκνές, δημιουργώντας ένα πιο λεπτομερές μοτίβο. Αντίθετα, για μικρότερες τιμές του συντελεστή, οι κυκλικές περιοχές εμφανίζονται πιο αραιές.

Στοιχεία εκτέλεσης:

```
GL Context Parameters:
Renderer: Intel(R) Iris(R) Xe Graphics
OpenGL version supported: 3.3.0 - Build 32.0.101.5542
GL_MAX_COMBINED_TEXTURE_IMAGE_UNITS 192
GL_MAX_CUBE_MAP_TEXTURE_SIZE 16384
GL_MAX_DRAW_BUFFERS 8
GL_MAX_FRAGMENT_UNIFORM_COMPONENTS 4096
GL_MAX_TEXTURE_IMAGE_UNITS 32
GL_MAX_TEXTURE_SIZE 16384
GL_MAX_VARYING_FLOATS 64
GL_MAX_VERTEX_ATTRIBS 16
GL_MAX_VERTEX_TEXTURE_IMAGE_UNITS 32
GL_MAX_VERTEX_UNIFORM_COMPONENTS 4096
GL_MAX_VIEWPORT_DIMS 16384 16384
-----------------------------
Compiling shader: simple.vertexshader
Compiling shader: simple.fragmentshader
Linking shaders...
Shader program complete.
```