

 triangle/README.md

# Hello Triangle!

---

În această parte a laboratorului codul GLSL (pentru şhadere) a fost mutat în fişiere sursă (vezi `shaders/fragment.glsl` şi `shaders/vertex.glsl` ). Metodele de citire şi de asamblare a programului de rendering sunt în `main.cpp` .

Verificaţi şablonul primit. Dacă este nevoie, faceţi modificările necesare ca să ruleze (necesare de ex, din cauza sistemului de operare sau a mediului de dezvoltare). Din directorul laboratorului

```
cd triangle/
```

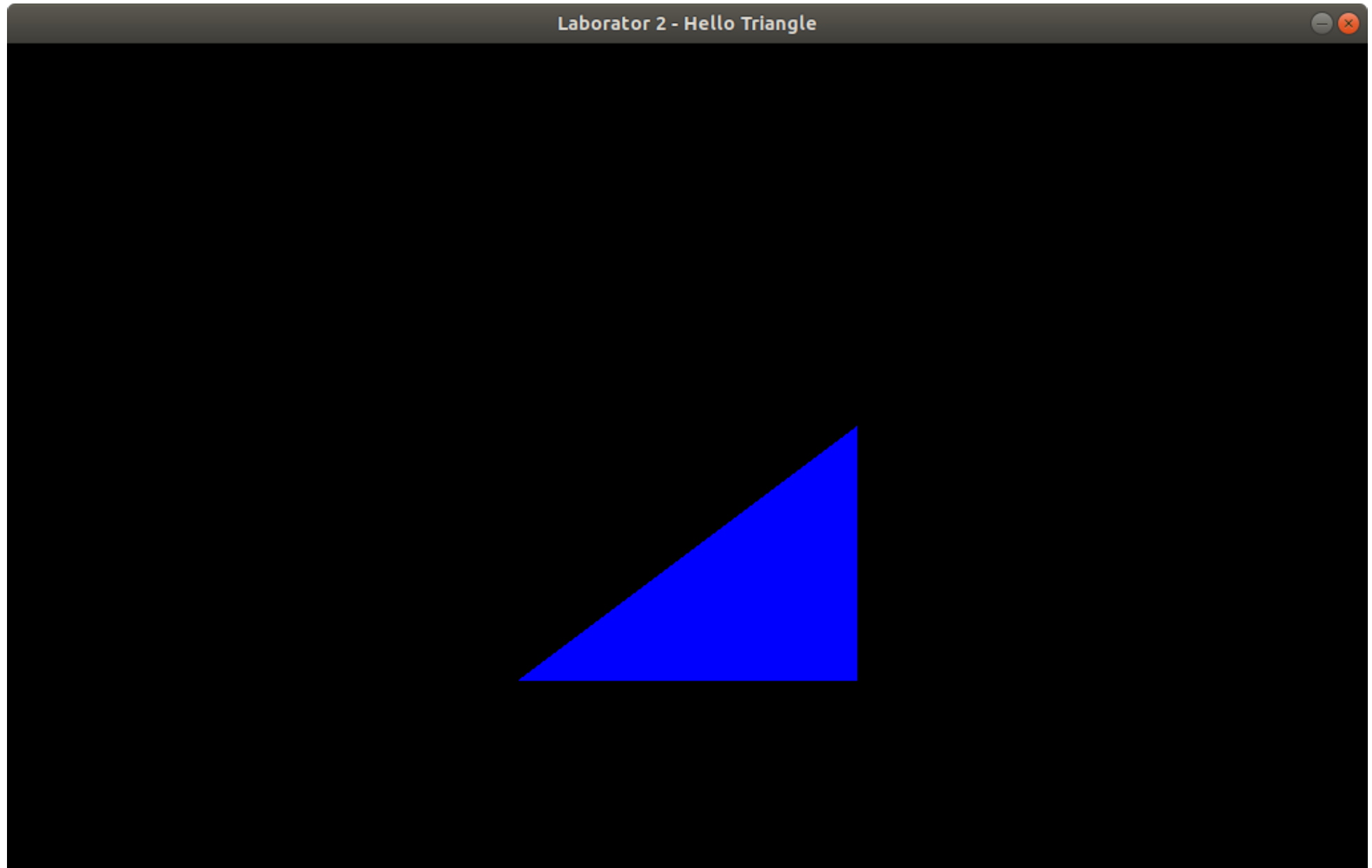
## Compilaţi

```
make
```

Apoi rulaţi aplicaţia. Sub linux:

```
./triangle
```

Veți obține un triunghi:



# Cerințe

---

1. Mică refactorizare: Mutați codul de citire și asamblare a șaderelor ca metode statice în clasa `utils` din directorul `common/`.
2. Provocați câte o eroare în `vertex.glsl` și `fragment.glsl` și verificați mesajul de eroare primit.
3. Adăugați un alt triunghi în scenă. Asta înseamnă că pipelineul trebuie activat pentru 6 puncte: `glDrawArrays(GL_TRIANGLES, 0, 6)` iar în `vertex.glsl` vom verifica id-ul pentru încă 3 puncte.
4. Faceți primul triunghi să se miște pe axa `Ox` (cea orizontală). Pentru asta putem adăuga variabile globale în aplicația `c++`

```
float x = 0.0f;    // pozitia triunghiului pe axa Ox
float inc = 0.01f; // pasul cu care mutam triunghiul
```

care să fie actualizate în metoda `display(..)` astfel:

```
x += inc; // modifică poziția triunghiului
if (x > 1.0f)
    inc = -0.01f; // schimbă direcția spre stânga
if (x < -1.0f)
    inc = 0.01f; // schimbă direcția spre dreapta
```

apoi, transmiteți poziția triunghiului, folosind o variabilă uniformă

```
GLuint offsetLoc = glGetUniformLocation(renderingProgram, "offset");
```

```
glProgramUniform1f(renderingProgram, offsetLoc, x);
```

iar în vertex.glsl

```
uniform float offset;  
...  
gl_Position = vec4( ... + offset, ..., 0.0, 1.0);
```

5. Schimbă culoarea pixelilor. Cei din stânga ecranului vor avea culoarea roșu iar cei din dreapta, albastru. Pentru asta, în fragment.glsl puteți folosi:

```
if (gl_FragCoord.x < ...)   
    color = vec4(1.0, 0.0, 0.0, 1.0);  
else ...
```