

## Despre LoadShaders.cpp

- O noua versiune, care permite afisarea erorilor de compilare.
- Poate fi instalata o extensie a MVS care sa indice eventuale erori de sintaxa in shadere

<https://marketplace.visualstudio.com/items?itemName=DanielScherzer.GLSL>

- Este recomandat sa fie apelata in functia de initializare.

## Despre createVBO ( )

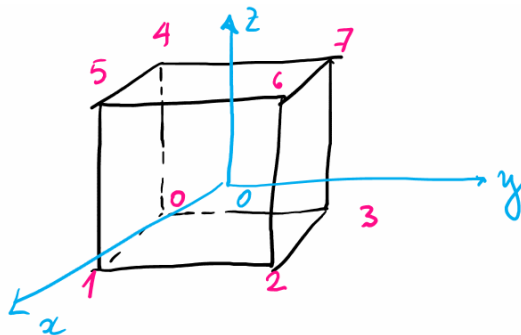
- Apelata in functia de initializare. Daca nu functioneaza, `glBindBuffer ( )` trebuie apelate inainte de functia de desenare.

- Structura:

- vectori cu varfuri, indici
- Generare nume ptr. buffer-objects: [`glGenBuffers \( \)`](#)
- Transfer date in buffer: [`glBufferData \( \)`](#)
- "Legare buffer" (eventual apelata inainte de functia de desenare): [`glBindBuffer \( \)`](#)
- Activarea lucrului cu atribute, indicarea locatiilor – vor fi utilizate in shader-ul de varfuri: [`glEnableVertexAttribArray \( \)`](#) ; [`glVertexAttribPointer \( \)`](#)

## 08\_01\_desenare\_cub.cpp

- Coordonatele varfurilor si indicii:



- Testul de adancime este esential pentru a reprezenta corect un obiect 3D
- Alte comentarii: functiile care “transmit” numele locatiei (`glGetUniformLocation`) pot fi indicate si in functia de initializare, DAR valoarea transmisa trebuie modificata dupa fiecare actualizare prin functii de tipul `glUniform*`. Numele unei variabile uniforme utilizate in *main* poate fi diferit de numele utilizat in shader.

## 08\_02\_survolare\_cub.cpp

- Observatorul se misca pe o sfera de raza `float dist` (variabila) in jurul punctului de referinta (centrul cubului).
- Verticala din planul de vizualizare este (0,0,-1).
- Este implementata reprezentarea sferei folosind unghiurile alpha si beta (latitudine, respective logitudine).

## 08\_03\_instanced\_rendering.cpp

- Sunt desenate mai multe instante ale aceluasi obiect, fiecare dintre ele avand culorile proprii ale varfurilor si o pozitie proprie.
- In `createVBO()` apar o serie de elemente specific randarii instantiate:
  - Coordonatele varfurilor sunt indicate separate de culorile/matricele de pozitie ale instantelor. Pentru fiecare instanta este precizata pozitia, pe o curba de forma  $(r(t)*\sin(t), r(t)*\cos(t))$ ; in plus, fiecare instanta este rotita (in spatiu).  

```
glm::translate(glm::mat4(1.0f),  
glm::vec3(80 * n * sin(10.f * n * 180 / PI), 80 * n * cos(10.f * n * 180 / PI), 0.0)) *  
glm::rotate(glm::mat4(1.0f), n * PI / 8, glm::vec3(n, 2 * n * n, n / 3));
```
  - O functie specifica randarii instantiate este `glVertexAttribDivisor()`. Aceasta indica rata cu care are loc distribuirea atributelor per instanta (de testat cazul cand al doilea parametru este 0, 2, 5, `INSTANCE_COUNT`).
  - In cazul atributului “pozitie a instantei” trebuie tinut cont ca este indicata o matrice 4x4, sunt alocate 4 atribute, corespunzator celor 4 coloane.