**Documentație**

**proiect**

Disciplina: **Sisteme de prelucrare grafică**

Nume și prenume student: Gorghe Teodor

Grupa: 1310B

An universitar: 2022-2023

1. **Descriere proiect**

Aplicația cuprinde următoarele funcționalități:

* Desenarea curbelor de tip Coons/Bezier.
* Modificarea punctelor pentru curbele existente

Pentru desenarea curbelor, se apasă cu click dreapta pe o zonă liberă din ecran.

Pentru schimbarea tipului de curbă, se apasă pe tasta **SPACE**.

Pentru modificarea coordonatelor punctelor, se apasă pe un punctul respectiv (cerc) și se mută cu click dreapta, în zona dorită.

A screenshot of a computer

Description automatically generated with low confidence

A picture containing screenshot, line, diagram

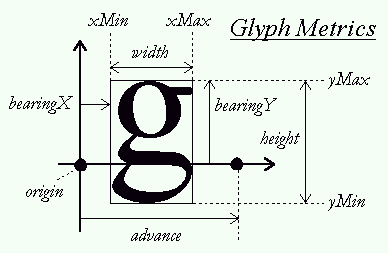
Description automatically generated

1. **Implementare**

S-a utilizat FreeGlut pentru gestiunea ferestrei/evenimente tastatură/mouse.

Ce este important de menținut:

- pentru randarea text-ului, s-a folosit biblioteca **FreeType**, care încarcă fișierul de tip .tff, returnând șiruri de octeți reprezentând textura și dimensiunile pentru fiecare caracter.



- pentru randarea cerc-urilor, s-a folosit pentru parametrul glDrawArrays, opțiunea **GL\_TRIANGLE\_FAN**, care formează triunghiurile din primul element din vector și grupări de 2 vertex-uri din ce a mai rămas din buffer (grupări 012, 023, ...), fără a mai fi necesar un **ELEMENT BUFFER OBJECT**. Valorile sunt calculate la inițializare și la schimbarea unui parametru.

- pentru randarea curbelor, s-a folosit un **VERTEX BUFFER OBJECT**, cu opțiunea **DYNAMIC DRAW**, pentru a putea modifica dimensiunea buffer-ului.

- pentru afișarea liniilor punctate, s-a folosit la funcția glDrawArrays, **GL\_LINES** și în fragment shader, se determină porțiunile unde linia e punctată și se folosește **discard**.

- s-a ținut cont de ordinea de desenare a curbelor. Întâi se desenează curbele, apoi liniile punctate și apoi marcajul pentru puncte (cercuri albastre).

- pentru fiecare tip de obiect, cerc, curbă, linie, text, este încapsulat într-o clasă, fiecare obiect având un vbo, vao și un shader (fragment și vertex). Ca o optimizare, s-ar putea instanția per clasă (global) și nu per obiect, dar fiind o aplicație didactică, nu am dorit să complic codul.

- pentru evenimentele freeGlut, s-a folosit funcții lambda, apelând metode din clasa **window**, pentru a păstra membrii necesari în clasă.

Modificări de la ultima prezentare:

- adăugare linii și cercuri, pentru o mai bună vizualizare a parametrilor

- remediere text atunci când se redimensionează fereastra

- adăugare funcționalitate modificare curbe

- adăugare contor referință VBO și VAO, atunci când se apelează destructorul, pentru curățare resurse după ștergere.

1. **Biblioteci folosite**

**-** FreeGLUT

- OpenGL v3.3

- FreeType (pentru font-uri)

1. **Referințe**

- Laborator SPG nr. 12

- <https://learnopengl.com/In-Practice/Text-Rendering>

- <https://github.com/VictorGordan/opengl-tutorials>

- <https://www.khronos.org/opengl/wiki/Primitive>

- <https://www.opengl.org/resources/libraries/glut/spec3/node50.html>

- <https://www.opengl.org/resources/libraries/glut/spec3/node49.html>

- <https://www.opengl.org/resources/libraries/glut/spec3/node48.html>