**GRAFICA PE CALCULATOR**

PROIECT 1

MOISEL RARES-IOAN

GRUPA 331

PREZENTAREA TEMEI:

Acest proiect folosește curbe Bezier de ordin variabil pentru a genera un traseu fluid și controlat, pe care un obiect îl poate parcurge. Scopul este de a permite obiectului să se deplaseze de-a lungul acestui traseu cu o mișcare naturală și precisă, utilizând proprietățile curbelor Bezier pentru a defini direcția și forma traiectoriei la fiecare punct.

DEFINIREA CURBELOR BEZIER:

Curbele Bezier sunt curbe parametrizate definite printr-un set de puncte de control, fiecare punct influențând forma finală a curbei. Numărul punctelor de control determină ordinul curbei.

În cazul unei curbe Bezier de ordin întâi, aceasta este definită prin formula:

Aceasta reprezintă o simplă interpolare liniară între două puncte, unde parametrul t este variabila care controlează poziția pe curba Bezier, variind de la 0 la 1. La t = 0, curba corespunde punctului de control inițial , iar la t = 1, curba ajunge la punctul de control final ​. Pe parcursul valorilor intermediare ale lui t, curba se deplasează fluid între aceste puncte de control.

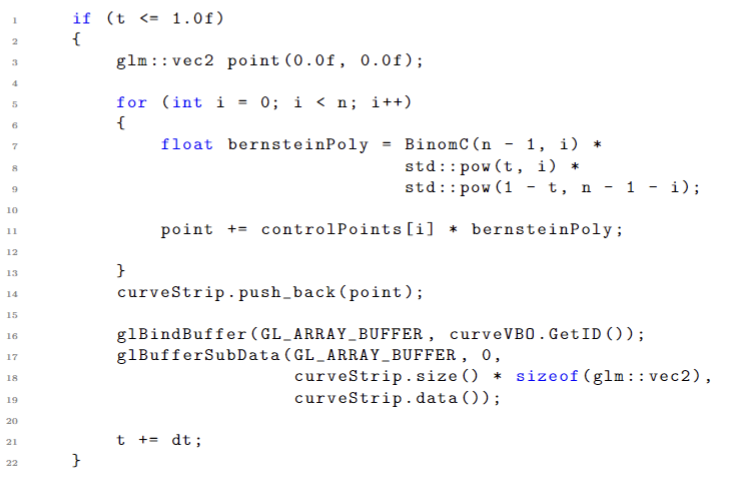
Pentru o curbă Bezier de gradul 2, formula devine:

În acest caz, calculăm mai întâi punctele intermediare ​ și ​ prin interpolare liniară între și ​, respectiv între și ​. Apoi, găsim punctul pe curbă B(t) prin interpolarea între ​ și ​, în funcție de parametrul t. Pentru curbele Bezier de ordin superior, formula este generalizată folosind polinoame Bernstein, astfel încât o curbă de ordin n este descrisă de relația:

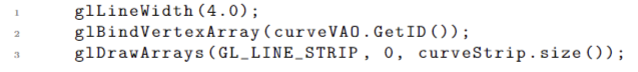
unde reprezintă punctele de control, iar primul termen al sumei este un coeficient binomial.

IMPLEMENTARE:

Realizarea curbelor Bezier se efectuează prin intermediul a două funcții din clasa Curve: CreateCurve() și DrawCurve(). Funcția CreateCurve() se ocupă cu popularea bufferelor necesare pentru reprezentarea curbei, în timp ce DrawCurve() este responsabilă pentru desenarea efectivă a acesteia.

CreateCurve():

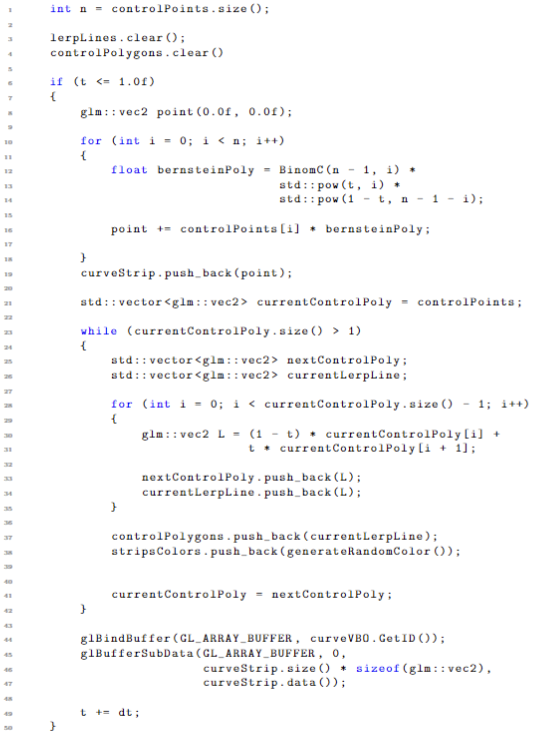


DrawCurve():

Pentru funcția CreateCurve(), se menționează că variabilele t și dt sunt de tip float, fiind inițializate cu 0.0f și respectiv 0.1f. Funcția parcurge punctele de control pentru a determina punctul curent de pe curba Bezier, raportat la parametrul t actual, și actualizează bufferul asociat curbei, curveVBO, folosind funcția glBufferSubData().

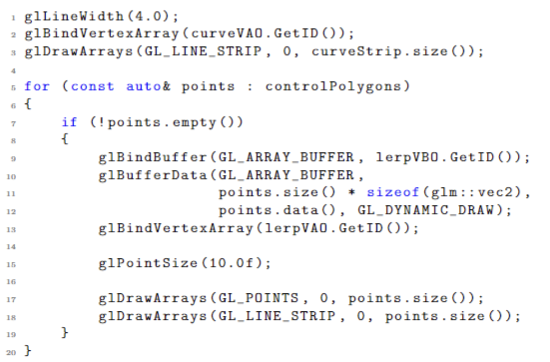
Pentru a crea o vizualizare completă a procesului de creare a unei curbe Bezier, funcțiile au fost modificate astfel:

CreateCurve():

În prima parte a funcției care creează curba Bezier, logica de bază rămâne aceeași Cu toate acestea, pentru a vizualiza întreag procesul de creere a curbei, trebuie să stocăm fiecare set de puncte interpolate pentru fiecare draw call. Această stocare se face în vectorul controlPolygons, care este un vector de vectori de vectori de tip vec2, reprezentând puncte.

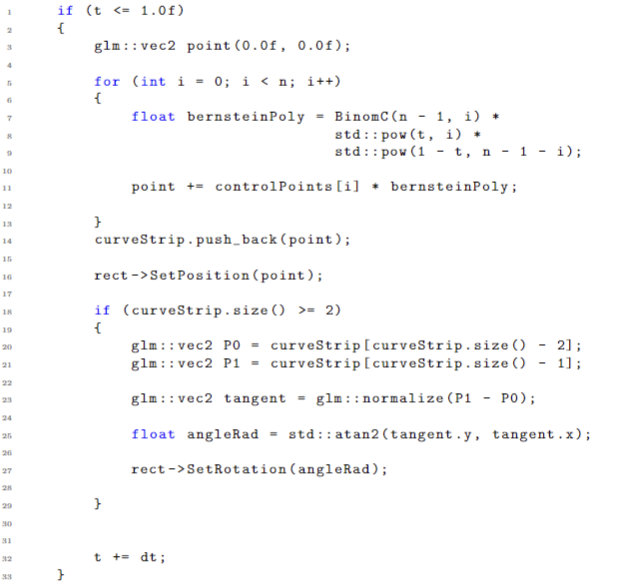
Containerul controlPolygons va conține:

* pe prima poziție, setul de puncte interpolate aflate pe segmentele dintre punctele de control;
* pe a doua poziție, punctele interpolate aflate pe segmentele dintre punctele interpolate din pasul anterior;
* și așa mai departe, până când setul curent conține un singur punct, caz în care ne aflăm la ultimul nivel de interpolare (adică atunci când currentControlPoly.size() > 1)

DrawCurve()



TRANSFORMARILE OBIECTULUI:

Pentru a actualiza corect poziția și direcția obiectului, la fiecare apel al funcției CreateCurve(), trebuie să setăm poziția curentă la punctul de pe curba Bezier care este desenat în acel moment. De asemenea, trebuie să rotim obiectul în jurul axei Z cu unghiul format de vectorul tangent la curbă în punctul curent.

Funcția atan2 este o variantă specială a funcției atan, care calculează unghiul dintre vectorul tangent la curbă și axa Ox, ținând cont de semnul componentelor vectorului tangent (adică direcția acestuia).





Codul sursă

<https://github.com/Barzoius/BezierCurves>

Observație

PDF-ul nu suportă formatul GIF, de aceea apar imagini gri. Pe GitHub am pus documentul în format .docx și GIF-urile separat.