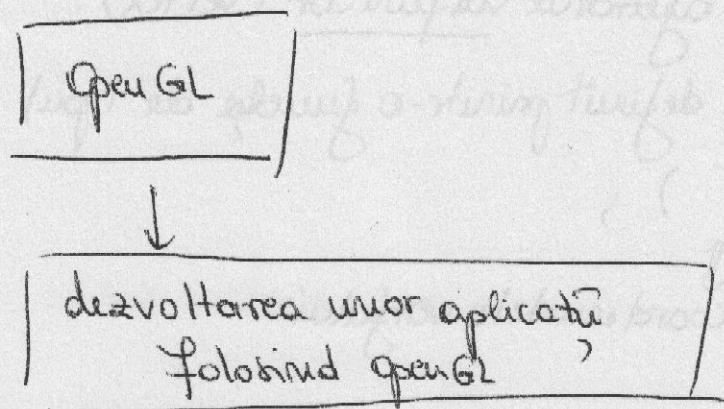


uc. geom. în gr. pe calc



Evaluare - verificare (ușă săpt.)

Punctaj: 10p of.
40p test oricărui final -> verificare
50p pe parcursul sem.
(laborator: proiecte, teste)

Laborator: 2h la săpt. (si > sp) → 7 lab

Biblioteci utilizate de OpenGL → funcții asociate

- biblioteca fundamentală ("core library")

→ independentă de platformă pe care lucrăm

→ funcții corespunzătoare cu prefixul gl

(ex.: glVertex(); glColor(); glBegin());

- OpenGL Utility (GLU): proceduri / funcții legate de proiecte
cuadrice, conice

funcțiile asociate cu prefixul glu

- pentru a realiza fereastra de vizualizare

biblioteca dependentă de sistem

(ex.: OpenGL Utility Toolkit GLUT) → poate interacționa
cu orice sistem de operare

Există și Apple GL (AGL) etc.

care utilizează ferestre
de vizualizare

* → funcții asociate cu prefixul: glut

Primitive grafice. Atribute ale primitiveelor grafice

P.g. sunt realizate cu ajutorul vârfurilor (vertex)

în OpenGL un vîrf este definit printr-o funcție de tipul

glVertex_n*();
 ↓ ↑
 prefix coordonatete vîrfului

*: 3 informații

• dimensiunile spațiului în care considerăm vîrfel
 $n \in \{2, 3, 4\}$

ex.: $n=2$ (2D) (3, 8)

$n=3$ (3D) (2, 4, 9) ↓ ⚡ (3, 8, 0) ⌄

$n=4$ (2, 4, 9, 1) ; (3, 8, 0, 1)

Δ într-un $n=4$, implicit: a 3^a componentă este 0.0

$a_{4^a} \rightarrow n \rightarrow 1.0$

• tipul de date utilizat: integer, float, double

• (poartă) utilizare a formei vectoriale

Ex.:

glVertex2i (80, 120);
 ↓

int p[] = {80, 120};

glVertex2iv(p);

Vârfurile le mut asociate:

- coordonate (fac parte din definire)
- culoare (optional)
- dimensiune (latură)

- coordinate de texturare, }
 - normale (iluminare) } \rightarrow later

Culori: (prezintă înainte de vârful respectiv!)

glColor * (compcolor),

* = suffix ascunzător celui de la glVertex();

n=3: R, G, B

red green blue

n=4: R, G, B, A

\hookrightarrow factorul alfa (transparență)

[afinteger] : R, G, B $\in \{0, 1, \dots, 255\}$

f, d R, G, B $\in [0.0, 1.0]$

abon() - combinație de bază
 (Cum obțin galben?)

0 0 0	- negru
1 1 1	- alb
1 0 0	- roșu
0 1 0	- verde
0 0 1	- bleu

Regulă: vârfurile sunt utilizate pt trasearea primutivelor grafice
 în cadrul uneor structuri de formă

\hookrightarrow tipul primutivelii

glBegin(" ");

— — — \leftarrow vârfuri

glEnd();

① Punctul | Puncte

glBegin(GL_POINTS);

\leftarrow vârfurile coresp punctelor

glEnd();

Atribute ale punctului

- culoarea (datorită de culoarea varfurii)
- dimensiunea $\text{glPointSize}(\text{dimensiune})$;

Δ Aceste funcții trebuie apelate înainte de apelarea primitivei $\text{glBegin(GL_POINTS)}$, pt a avea efectul dorit

② Segmente de drepte / linii

GL_LINES → segmente care unește punctul $2k+1$ cu $2k+2$
(le unește 2 căi)
nu împreună ultimul nomenosc punct

GL_LINE_STRIP → linie foarte îngustă în care punctul i este unit cu $(i+1), +i$

GL_LINE_LOOP → linie foarte îngustă în care punctul i este unit cu $(i+1), +i$ și ultimul punct e unit cu primul

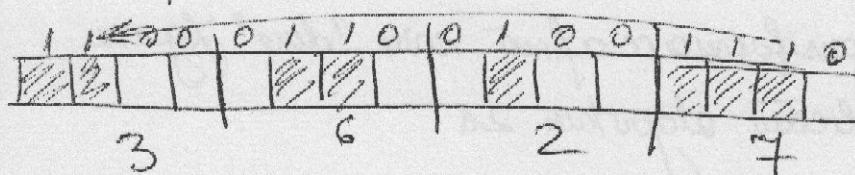
Atribute:

- culoare: Δ dacă varfurile au culori diferite, culorile punctelor unui segment sunt "calculate" folosind interpolare (afine)
Modul de desenare se controlează cu funcția glShadeModel
 - $\text{glShadeModel(GL_FLAT)}$; → NU interpolare
 - $\text{glShadeModel(GL_SMOOTH)}$; → defau et
- grauieea: $\text{glLineWidth(width)}$;
 ↑ float
Δ Ab. apelată înainte de glBegin(GL_LINES) ;
etc.

- modul de desenare (sablon / model)

④ este definit printr-o funcție de formă getlineStipple (repeat factor, pattern);

Ex: pattern: 0x7263



culoarea primăvei
 fond

ar mat / precizarea cătă cu se repetă pattern

16 biti Repetitie
în pixelarea

1 = pixel "on" (cu culoarea primăvei),
0 = pixel "off"

default: 0x7FFF

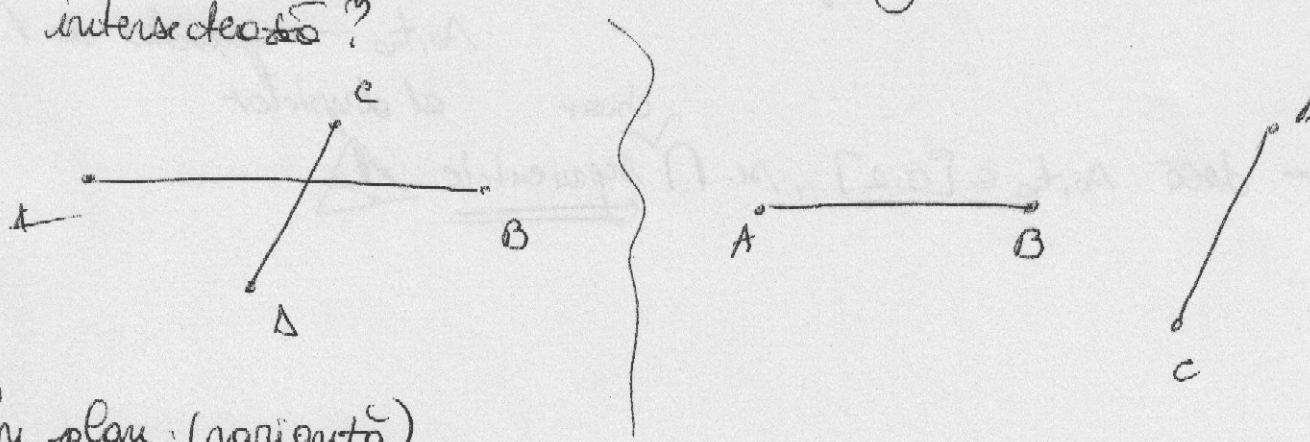
④ este activat / dezactivat prin:

getnable (GL_LINE_STIPPLE),

getable (GL_LINE_STIPPLE);

Intersecții de segmente

Întrebare: cum pot fi verificate dacă 2 segmente [AB] și [CD] se intersechetează?



În plan: (variantă)

④ [AB] și [CD] se intersechetează \Leftrightarrow A și B sunt de o parte și de altă a dreptei CD și C și D sunt de o parte și de altă a dreptei AB.

⑤ Cum verificăm că 2 puncte M și N sunt de o parte și de altă a unei drepte?



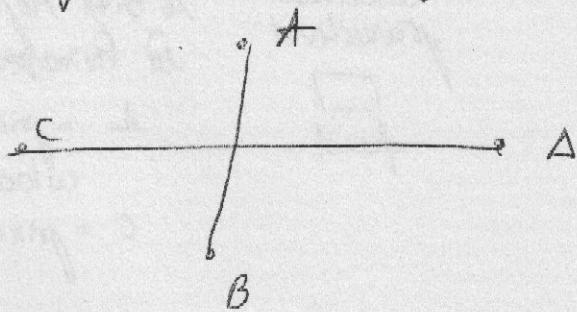
Seriem ecuație implicită a dreptei d : $f(x, y) = 0$

$M \neq N$ mult de o parte și de altă a lui d $\Leftrightarrow \underline{f(M) \cdot f(N) < 0}$

In spațiu : (pp ca A, B, C, D coplanare
→ de testat...)

Vari 1 : găsim o izometrie liniară formări afină care "dă" figura în planul $\mathbb{A} = 0$ și aplicăm algoritmul 2A.

Vari 2 :



$$AB = \{(1-t)A + tB \mid t \in \mathbb{R}\} = \text{dreapta } AB$$

$$c\Delta = \{(1-s)c + s\Delta \mid s \in \mathbb{R}\} = \text{dreapta } c\Delta$$

⑦ găsim $s \neq t \in \mathbb{R}$ qr. $(1-t)A + tB = (1-s)c + s\Delta$

(strenghid de coordonate: găsim 3 ecuații cu 2 necunoscute s_0, t_0)

- dacă sistemul e compatibil determinat soluție e unică

$s_0, t_0 \rightarrow$ punctul de Δ

- deci $s_0, t_0 \in [0, 1]$, $\Delta \cap$ segmentele chiar al dreptelor

