

Laborator 10

Cuprins

1. Obiective	2
2. Utilizarea texturilor în OpenGL	2
2.1 Introducere	2
2.2 Observații	3
3. Probleme de rezolvat	3

1. Obiective



- Lucrul cu API-ul OpenGL
- Lucrul cu biblioteci externe (OpenTK)
- Manipularea texturilor
- Texturarea obiectelor OpenGL
- Abordarea POO în generarea de obiecte 3D

2. Utilizarea texturilor în OpenGL

2.1 Introducere

Texturile sunt tablouri rectangulare de date ce pot reține, de exemplu, informații cu privire la culoare, la luminozitate, canale de transparență, etc. Valorile individuale din aceste tablouri se numesc *texeli*. Problema principală de care ne lovim în ceea ce privește texturarea o reprezintă maparea unei imagini rectangulare pe suprafețe care pot fi non-rectangulare; în acest caz rezultatele trebuie să fie acceptabile din punct de vedere grafic.

Pașii care trebuie urmați în ceea ce privește texturarea:

- crearea unui obiect de tip textură (cu textura asociată);
- indicarea modului în care textura va fi aplicată pe fiecare pixel;
- activarea opțiunii de texturare;
- desenarea scenei, furnizând atât textura cât și coordonatele geometrice.

Texturile pot fi 1D, 2D sau 3D. Un texel poate reține 1 până la 4 elemente, elemente ce pot fi constante, valori normalizate, tuple (R,G,B,A), etc.

Există 4 funcții posibile care pot calcula valoarea RGB finală pentru imaginea texturii ce va fi aplicată:

- *decal mode* (culoarea texturii este folosită ca atare, aceasta fiind "aplicată" deasupra obiectului texturat);
- *replace mode* (o variantă a decal mode);
- modulația/scalarea culorii fragmentului (folosită mai ales pentru combinarea efectelor de lumină cu textura aplicată);
- *blended mode* (o constantă de culoare este "amestecată" cu fragmentul de texturat, modificând valoarea fragmentului de textură).

Modul texturare trebuie activat înainte de a desena scena 3D. Această operațiune se face cu comanda (exemplul prevede texturarea 2D):

```
GL.Enable(EnableCap.Texture2D)
...
GL.Disable(EnableCap.Texture2D)
```

Textura creată trebuie aliniată la fragmentul de obiect 3D pe care se aplică. Acest lucru se face prin specificarea coordonatelor texturii precum și a coordonatelor vertexurilor ce definesc obiectul. Pentru texturile 2D coordonatele variază între 0.0 și 1.0 în ambele direcții (valorile peste 1.0 semnifică de obicei repetarea texturii pe fragmentul de obiect 3D, iar valorile mai mici ca 1.0

semnifică utilizarea parțială a texturii¹). Pentru texturarea unui *quad*, de exemplu, o imagine de textură se poate în general utiliza cu coordonatele (0,0), (1,0), (1,1) și (0,1).

Trebuie avută grijă la următoarele aspecte:

- Suprafețele adiacente ale unui obiect 3D trebuie să fie texturate cu imagini care se potrivesc.
- Texturarea suprafețelor neregulate sau non-rectangulare (sau volumetrice) poate duce la apariția de efecte nedorite de distorsiune sau la zone neacoperite de texturi. În acest caz trebuie luate măsuri speciale cu privire la imaginea texturii și a modului în care se face maparea.
-

2.2 Observații

1. ATENȚIE: pentru acest laborator utilizați versiunea 2.x a bibliotecii OpenTK!!!

3. Probleme de rezolvat



1. Urmăriți în cadrul aplicației **OpenTK_winforms_z03** funcționarea procesului de texturare. Testați ambele tipuri de obiecte 3D.
2. Urmăriți în cadru aplicației **OpenTK_winforms_z03** modul în care sunt stocate și încărcate vertexurile respectiv lista de vertexuri utilizate.
3. Modificați aplicația oferită astfel încât să realizați texturarea pe diverse obiecte 3D (volumetrice) sau pe suprafețe non-rectangulare. Ce observați?
6. **Răspundeți la următoarele întrebări:**
 - Utilizați pentru texturare imagini cu transparentă și fără. Ce observați?
 - Ce formate de imagine pot fi aplicate în procesul de texturare în OpenGL?
 - Specificați ce se întâmplă atunci când se modifică culoarea (prin manipularea canalelor RGB) obiectului texturat.
 - Ce deosebiri există între scena ce utilizează obiecte texturate în modul iluminare activat, respectiv dezactivat?

¹ 1.0 echivalează cu 100% din textură fiind folosită.