Texturare

Mihai-Sorin Stupariu

Sem. al II-lea, 2022- 2023

Etape

Maparea texturilor presupune realizarea următoarelor patru etape:

Etape

Maparea texturilor presupune realizarea următoarelor patru etape:

- crearea unei texturi sau a unui obiect textură și specificarea caracteristicilor texturii;

Etape

Maparea texturilor presupune realizarea următoarelor patru etape:

- crearea unei texturi sau a unui obiect textură și specificarea caracteristicilor texturii;
- indicarea modului în care textura este aplicată pe pixelii imaginii;

Etape

Maparea texturilor presupune realizarea următoarelor patru etape:

- crearea unei texturi sau a unui obiect textură și specificarea caracteristicilor texturii;
- indicarea modului în care textura este aplicată pe pixelii imaginii;
- activarea texturării;

Etape

Maparea texturilor presupune realizarea următoarelor patru etape:

- crearea unei texturi sau a unui obiect textură și specificarea caracteristicilor texturii;
- indicarea modului în care textura este aplicată pe pixelii imaginii;
- activarea texturării;
- reprezentarea scenei, indicând atât coordonatele de texturare (s, t, r, q), cât și cele geometrice (x, y, z, w).

Texturi 1D - context geometric

În cazul 1-dimensional, coordonatele de texturare sunt situate în intervalul $[0.0, 1.0]$ (0.0 reprezintă începutul texturii, iar 1.0 sfârșitul acesteia). În momentul în care este apelată o funcție de tipul

```
glTexCoord* (coord);
```

urmată de

```
glVertex* (pt);
```

coordonata `coord` a texturii este asociată vârfului `pt`.

Texturi 1D - funcții asociate

Definirea unei texturi 1D se face apelând funcția

```
glTexImage1D (.........);
```

având parametrii

- `target`: tipul de textură (`GL_TEXTURE_1D`);
- `level`: nivelul de texturare (0);
- `intformat`: numărul valorilor (de culoare) utilizate pentru fiecare pixel; în cazul codului RGBA acest număr este 4;
- `width`: lățimea este o putere a lui 2; trebuie să fie coerentă și consistentă cu modul în care a fost definită textura;
- `border`: este un număr egal cu 0,1 sau 2 și indică numărul pixelilor de pe frontieră;
- `format`: poate fi o constantă simbolică de forma `GL_RGB`; `GL_RGBA`;
- `type`: tipul este indicat printr-o constantă simbolică `GL_UNSIGNED_BYTE`; `GL_BYTE`;
- `texels`: se indică unde este localizată textura.

1. Crearea unui obiect textură și specificarea proprietăților acestuia

Obiectele textură memorează date referitoare la textură, făcându-le accesibile imediat. Fiecărui obiect îi este asociată o singură textură. Sunt parcurși câțiva pași intermediari, descriși mai jos, împreună cu funcțiile OpenGL asociate.

1a. Generarea numelor obiectelor textură

:

```
glGenTextures (n, *texNames);
```

Prin această funcție sunt *rezervați n* identificatori de textură în vectorul `texNames`, declarat explicit în procedura de inițializare `init`. Rezervarea are ca semnificație faptul că numele din `texNames` sunt marcate ca fiind utilizate, ele primesc caracteristici efective atunci când sunt prima dată legate cu funcțiile specifice. **Observație.** De menționat că funcția OpenGL cu efect contrar celei descrise este funcția de ștergere

```
glDeleteTextures (n, *texNames);
```

1b. Crearea și utilizarea obiectelor textură

```
glBindTexture (target, id);
```

Aici `target` este parametrul care descrie dimensiunea texturii; fiind vorba de o textura 2-dimensională este `GL_TEXTURE_2D`, iar `id` este numele texturii. Prin această funcție este *creat* un nou obiect textură, cu numele `id`. La apelarea lui `glBindTexture (...);` în cadrul funcției de desenare, textura este legată de obiectul desenat, ceea ce este corelat cu corespondența dintre coordonatele de texturare și cele de modelare. De fapt, `glBindTexture (...);` precizează obiectul textură activ.

1c. Specificarea caracteristicilor texturii

```
glTexImage2D (.........);
```

având parametrii asemănători cu cei din cazul 1-dimensional

- `target`: tipul de textură (`GL_TEXTURE_2D`);
- `level`,`intformat`: similar cu cazul 1D;
- `width, height`: lățimea și înălțimea texturii sunt de forma $2^w + b_w$, respectiv $2^h + b_h$ (altfel spus, puteri ale lui 2 la care se adaugă dimensiunile frontierei);
- `border`: dimensiunea frontierei (b_w, b_h);
- `format`, `type`, `texels`: similar cu cazul 1D.

Trebuie menționat faptul că pot fi definite și subtexturi, prin funcția

```
glTexSubimage2D (.........);
```

având parametrii similari

- `target`: tipul de textură (`GL_TEXTURE_2D`);
- `level`: similar cu funcția `glTexImage2D`;
- `xoffset, yoffset`: sunt numere întregi, pozitive, care precizează unde anume este așezată subtextura în structura de texeli (cu convenția că (0,0) este în stânga jos);
- `width, height`: lățimea și înălțimea subtexturii;
- `format`, `type`, `texels`: similar cu funcția `glTexImage2D`.

Semnificația acesti funcții este că se definește o structură prin care este înlocuită parțial / total o porțiune a texturii active (curente).

2. Precizarea unor reguli referitoare la modul în care texelii sunt așezați pe structura de pixeli

Două reguli sunt luate în considerare: *repetarea texturii* și *filtrare*. Ambele sunt legate de faptul că atât structura de texeli cât și cea de pixeli sunt discrete, iar scopul lor este de a indica modul în care se procedează atunci când nu există o corespondență 1:1 între cele două structuri. Forma generală a funcției asociate este

```
glTexParameter* (target, pname, value);
```

Parametrul `target` indică tipul de textură, fiind `GL_TEXTURE_2D`. Ceilalți parametri depind ca semnificație și valoare de regula la care se face referire.

2a. Repetarea texturii.

În acest caz (și ținând cont că suntem în context bidimensional) `pname` poate lua valorile

`GL_TEXTURE_WRAP_S`	`GL_TEXTURE_WRAP_T`

s, t sunt primele două coordonate ale texturii 2D, semnificația fiind că este precizată regula aplicată pentru coordonata indicată. În particular, pot fi utilizate reguli diferite pentru cele două coordonate de texturare. Parametrul `value` poate avea una din valorile

`GL_CLAMP`	`GL_REPEAT`

În cazul lui `GL_CLAMP`, dacă se parcurge toată textura de-a lungul coordonatei considerate, valoarea ultimului texel este folosită în continuare pentru completarea tuturor pixelilor. În cazul lui `GL_REPEAT`, dacă se parcurge toată textura de-a lungul coordonatei considerate, se repornește parcurgerea texturii de la început, aceasta fiind repetată de câte ori este necesar.

2b. Filtrare.

Aici apare explicit faptul ca se poate întâmpla ca la randarea unei scene sa nu existe o corespondență biunivocă între texeli și pixeli. Astfel, se poate ca un pixel sa corespundă unei porțiuni mici a structurii de texeli sau, invers, un pixel sa corespundă unei structuri de texeli, iar rolul regulii de filtrare este de a stabili ce texel îi este asociat pixelului considerat. Parametrul `pname` poate avea, corespunzător celor două situații, valorile

`GL_TEXTURE_MAG_FILTER`	`GL_TEXTURE_MIN_FILTER`

Parametrul `value` poate avea una din valorile

`GL_NEAREST`	`GL_LINEAR`

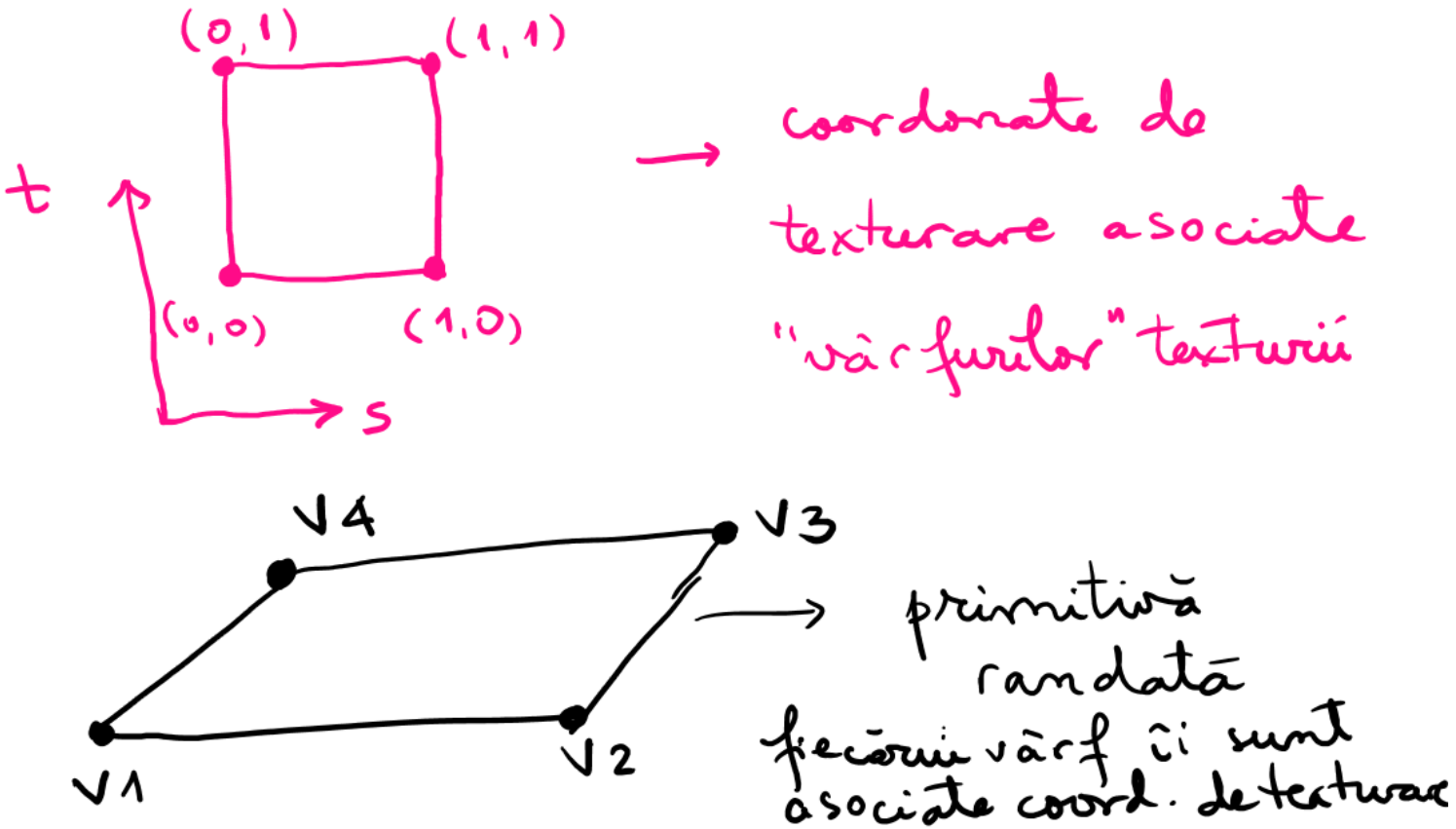
Semnificația valorilor este următoarea: pentru `GL_NEAREST` se alege texelul având coordonatele cele mai apropiate de centrul pixelului, în timp ce pentru `GL_LINEAR` este utilizată o tehnică de tipul *moving window*, fiind considerat un tablou de 2×2 texeli apropiați de centrul pixelului, după care se face o mediere.

Coordonate de modelare și coordonate de texturare

Coordonatele de modelare sunt cele utilizate în mod curent în funcția de desenare. Fiecărui vârf îi sunt asociate, așa cum se știe, cele trei coordonate spațiale (x, y, z). În cazul texturilor 2D coordonatele de texturare sunt s și t, ambele în intervalul $[0.0, 1.0]$. Cu alte cuvinte, coordonatele de texturare (teoretic) sunt reprezentate de pătratul $[0.0, 1.0] \times [0.0, 1.0]$. În cadrul funcției de desenare, fiecărui vârf îi sunt asociate anumite coordonate de texturare. Este suficient, pentru o primitivă grafică 2D, să fie indicate coordonatele de texturare pentru trei dintre vârfuri, coordonatele de texturare ale punctelor din interior rezultă automat. Motivație: dacă au fost fixate trei puncte necoliniare a, b, c în spațiul de texturare și trei puncte necoliniare A, B, C în spațiul de modelare, există o unică aplicație afină care transformă punctele a, b, c in A, B, respectiv C.

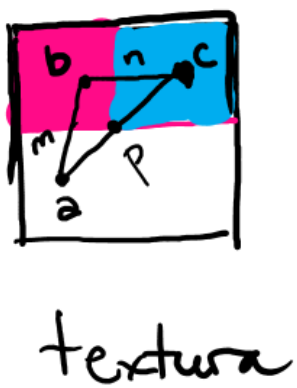
Despre aplicarea texturii

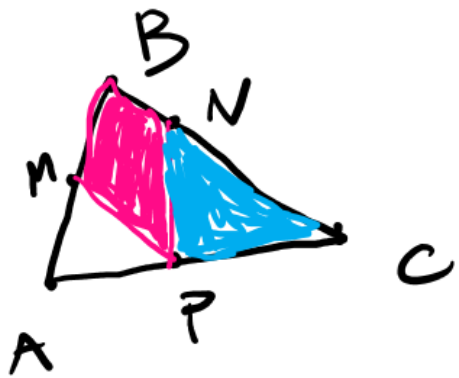
Coordonatele de texturare (teoretic) sunt reprezentate de pătratul $[0.0, 1.0] \times [0.0, 1.0]$.
În cadrul funcției de desenare, fiecărui vârf îi sunt asociate anumite coordonate de texturare.



Despre aplicarea texturii

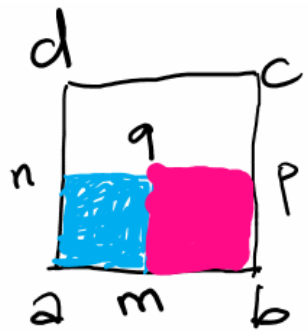
Este suficient, pentru o primitivă grafică 2D, să fie indicate coordonatele de texturare pentru trei dintre vârfuri, coordonatele de texturare ale punctelor din interior rezultă automat, folosind coliniaritatea și rapoarte de puncte coliniare.

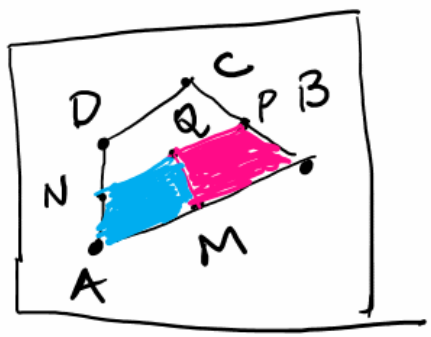




Despre aplicarea texturii

Este suficient, pentru o primitivă grafică 2D, să fie indicate coordonatele de texturare pentru trei dintre vârfuri, coordonatele de texturare ale punctelor din interior rezultă automat, folosind coliniaritatea și rapoarte de puncte coliniare.

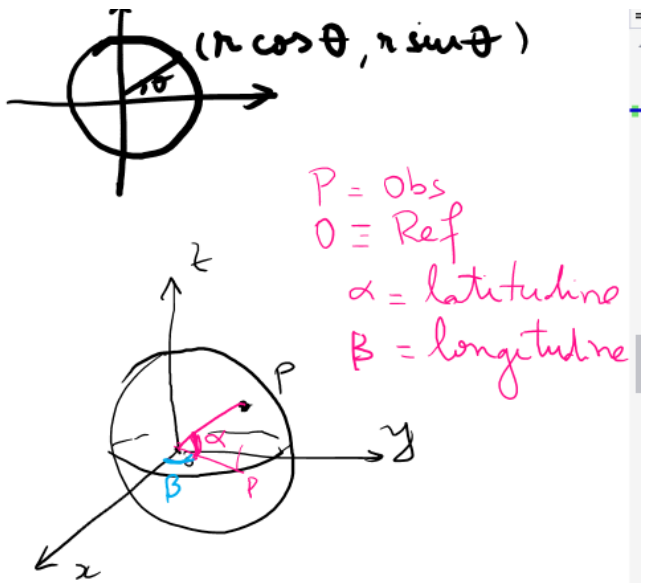




Comentariu - survolare (exemplul cu cub texturat)

Pentru a survola un obiect este folosită reprezentarea unei sfere cu centru și rază fixate (`Ref`, respectiv `dist` în codul de mai jos), folosind latitudinea și longitudinea (`alpha`, respectiv `beta`).

```
118     glEnable(GL_LIGHTING);
119     glEnable(GL_LIGHT0);
120     glShadeModel(GL_SMOOTH);
121     GLfloat pozitial0[] = { xs, ys, zs, ts };
122     glLightfv(GL_LIGHT0, GL_POSITION, pozitial0);
123
124     // Pozitia observatorului
125       //pozitia observatorului
126     Obsx = Refx + dist * cos(alpha) * cos(beta);
127     Obsy = Refy + dist * cos(alpha) * sin(beta);
128     Obsz = Refz + dist * sin(alpha);
129     glMatrixMode(GL_MODELVIEW);
130     glLoadIdentity();
131     gluLookAt(Obsx, Obsy, Obsz, Refx, Refy, Refz, 0.0f, 0.0f, 1.0f);;
```



Concluzie

În concluzie, o textură 2D este: (i) un tablou dreptunghiular de texeli având dimensiunile puteri ale lui 2; (ii) o mulțime înzestrată cu coordonate de texturare (pătratul standard). Atunci când tabloul de texeli are același număr de linii / coloane și atunci când textura este aplicată direct pe un pătrat, corespondența dintre pixeli și texeli este ușor de controlat și nu apar fenomene de deformare. În caz contrar (situație mai des întâlnită), trebuie manevrați în mod convenabil parametrii disponibili.