

6. Decuparea față de un plan de decupare. Primitive raster în OpenGL. Liste de display. Fonturi în OpenGL.

6.1. Decuparea față de un plan de decupare

Un algoritm de decupare tridimensională identifică și salvează pentru reprezentare toate obiectele aflate de o parte a planului de decupare. Aceste obiecte vor fi apoi proiectate pe planul de vizualizare.

Decuparea poate fi realizată extinzând algoritmi de decupare a liniilor sau a suprafețelor. Se utilizează ecuația planului. În felul acesta se poate testa poziția relativă a unei linii față de plane și se pot localiza punctele de intersecție.

Prin substituirea în ecuația planului a coordonatelor punctului extrem al unei drepte se poate verifica dacă acest punct se află în interiorul sau în exteriorul planului. Un punct (x, y, z) se află în exteriorul planului de parametrii A, B, C, D dacă se verifică inegalitatea:

$$A * x + B * y + C * z - D > 0.$$

În cazul în care punctul se află în interiorul planului se verifică inegalitatea:

$$A * x + B * y + C * z - D < 0.$$

În situația segmentelor de dreaptă pentru care ambele extremități se află în interior, dreapta va fi eliminată. Dacă ambele extremități ale unui segment de dreaptă se află în semispațiul interior, segmentul va fi salvat pentru proiecție. Punctul de intersecție al unei drepte cu un plan se determină ca fiind punctul de coordonate (x_l, y_l, z_l) care verifică atât ecuația dreptei cât și egalitatea următoare:

$$A * x_l + B * y_l + C * z_l - D = 0.$$

Aplicația 1. Se folosește sursa `Decupare.c`.

```
glClipPlane (GL_CLIP_PLANE0, eqn0); //se asociază planul
GL_CLIP_PLANE0 cu coeficienții A, B, C, D furnizați în tabloul eqn0.
glEnable (GL_CLIP_PLANE0); //se activează planul GL_CLIP_PLANE0.
```

Un plan de decupare împarte spațiul în două semispații. Dacă ecuația `eqn0` conține coeficienții A, B, C, D se vor vedea obiectele din unul din cele două semispații. Dacă se înmulțesc coeficienții cu -1 se vor vedea obiectele din celălalt semispațiu.

$$\begin{aligned} A * x + B * y + C * z - D &= 0. / (-1) \\ -A * x - B * y - C * z + D &= 0. \end{aligned}$$

Să se decupeze sfera cu un plan orizontal care taie sfera în jumătăți. Să se vizualizeze odată calota superioară apoi calota inferioară.

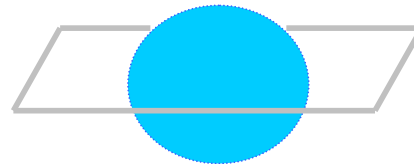


Fig.1

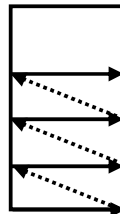
6.2. Primitive raster

În aplicația `Bitmap_E.c` este reprezentată o primitivă raster care modelează caracterul E.



Fig. 2

Într-un tablou de tip `Glubyte` sunt specificați octeții în hexazecimal. Fiecărui pixel îi corespunde un bit.



Ordinea în care se specifică pixelii este cea din figură. Funcția `glRasterPos2i()` furnizează poziția punctului raster curent în unități logice. Acest punct este punctul în care se va plasa originea bitmap-ului.

Reprezentarea bitmap-ului se face cu funcția `glBitmap()`. Parametrii funcției sunt lățimea și înălțimea bitmap-ului în pixeli, poziția originii, deplasarea poziției raster curente după desenarea bitmap-ului, tabloul care conține imaginea bitmap-ului. Tabloul va avea un număr par de octeți pe lățime. Spre exemplu, dacă lățimea bitmap-ului este de 13 pixeli se vor utiliza 2 octeți pe lățime. Biții de la 14 la 16 vor fi complecțați (padați) cu 0.

Aplicația 2. Să se reprezinte caracterul A.

6.3. Liste de display

Listele de display permit mărirea vitezei de afișare în situația în care un obiect se afișează de mai multe ori într-o scenă. Apelurile OpenGL dintr-o listă de display vor fi salvate cu valoarea calculată a parametrilor, din momentul executării listei de display. La apelarea listei se vor folosi funcțiile cu parametri calculați. Apelarea se face pe baza unui index (număr) asociat listei.

Aplicația 3. `Liste1.c` arată modul de definire și de executare al listelor de display. În aplicația `Liste2.c` se arată felul în care starea OpenGL modificată de liste nu afectează exteriorul listei. Să se identifice funcțiile pentru definirea listelor și funcțiile pentru apelarea listelor de display.

Să se compare cele 2 aplicații.

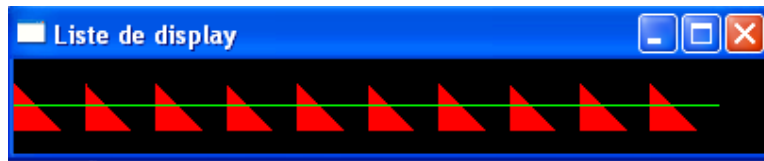


Fig. 3

6.4. Fonturi

Fontul este un set de caractere cu aspect grafic asemănător. Folosind listele de display și primitivele raster, în aplicația `Font.c` s-a construit un font. S-a construit o listă de display pentru fiecare caracter. Indexul fiecărei liste s-a dat ca fiind chiar codul ASCII al caracterului respectiv. Aceasta a permis scrierea unei funcții `PrintString()` pentru afișarea unui șir de caractere. Funcția primește ca parametru șirul de afișat. Va apela toate listele de display care au indexul egal cu codul ASCII al caracterului din șir.

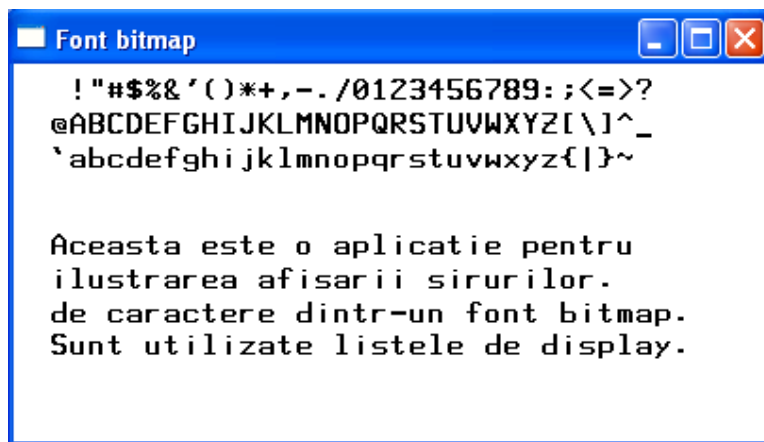


Fig. 4

Aplicația 4. Să se reprezinte o sferă. Sub sferă să se afișeze textul „Aceasta este o sferă”.