

5. Ascunderea suprafețelor. Eliminarea fețelor în funcție de orientare. Animarea obiectelor. Reuniunea dintre două corpuri

5.1 Ascunderea suprafețelor

Obiectele 3D sunt ordonate în adâncime după z. Pentru a se reprezenta corect obiectele în funcție de adâncime trebuie activat algoritmul de ascundere al suprafețelor. OpenGL utilizează algoritmul z-buffer (metoda buffer-ului de adâncime). Pentru aceasta buffer-ul de cadru trebuie să conțină și buffer-ul de adâncime. În funcția `auxInitDisplayMode()`, la crearea ferestrei se va specifica și opțiunea `AUX_DEPTH`.

Funcția `glEnable(GL_DEPTH_TEST)` activează testul de adâncime. La ștergerea buffer-elor aferente ferestrei se va șterge și buffer-ul de adâncime.

```
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
```

Pentru vizualizarea modului de ascundere a suprafețelor se dă sursa `TEST_ADANCIME.c`. Programul afișează două triunghiuri care se ascund reciproc.

Aplicația 1. Să se facă reprezentarea fără activarea testului de adâncime. În acest caz să se schimbe ordinea specificării triunghiurilor. Mai contează ordinea specificării în cazul în care se activează testul de adâncime?

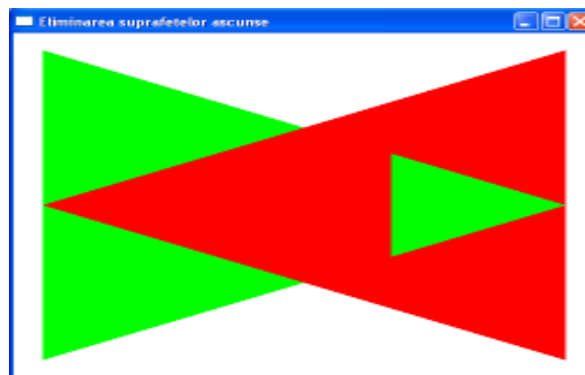


Fig. 1

5.2 Eliminarea fețelor funcție de orientare

OpenGL permite ca să fie eliminate (tăiate) fețele. Fiecare poligon are o față FRONT și o față BACK. În mod explicit, fețe FRONT sunt cele pentru care vârfurile sunt specificate în sens invers rotirii acelor ceasornicului. Orientarea implicită poate fi schimbată din `GL_CCW` în `GL_CW` (în sensul rotirii acelor de ceasornic).

Pentru eliminarea fețelor se folosesc funcțiile:

```
glEnable(GL_CULL_FACE)      //activează eliminarea fețelor;  
glCullFace(GL_BACK)         //sunt eliminate fețele spate;  
glCullFace(GL_FRONT)        //sunt eliminate fețele față.
```

Ordinea directă a vârfurilor se schimbă cu `glFrontFace` având parametrii - `GL_CW`, `GL_CCW`.

```
glFrontFace(GL_CW)      // orientare GL_CW ca orientare directă;  
glFrontFace(GL_CCW)     // orientare GL_CCW ca orientare directă.
```

Aplicația 2. Se dă sursa `FETE_ELIMINATE.c`. Programul afișează suprafețele fără activarea eliminării fețelor. Cu orientarea directă implicită se vor schimba fețele care se elimină adică, cele față sau cele spate, cu funcția `glCullFace()`.

Să se modifice sursa astfel încât să permită vizualizarea ambelor poligoane chiar dacă se activează tăierea fețelor cu funcția `glCullFace()`.

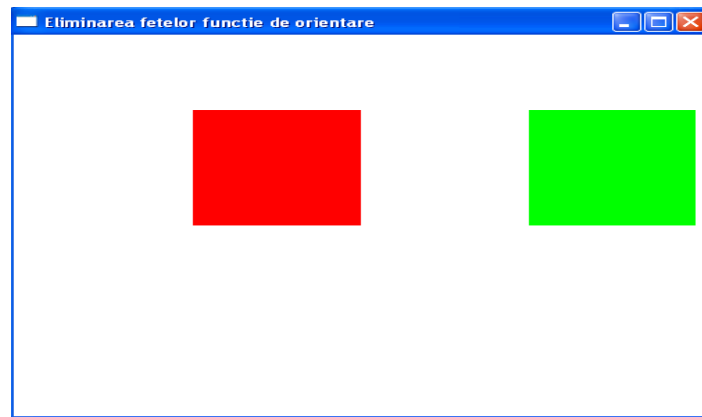


Fig.2

5.3 Animarea obiectelor

Varianta 1:

Pentru rotirea unui obiect se înregistrează în sistemul de operare o funcție de tip `CALLBACK-IdleFunction()` și care va fi apelată de acesta când aplicația este în așteptare.

```
auxIdleFunc (IdleFunction);
```

Funcția `IdleFunction` se va programa. Un exemplu este dat în continuare.

```
void CALLBACK IdleFunction(void)  
{  
    glRotatef(30,1,1,1);  
    display();  
    Sleep(300);  
}
```

Această variantă este cea recomandată pentru programarea animației deoarece permite interacțiunea utilizatorului cu aplicația fără blocaje.

Varianta 2:

```
void CALLBACK display (void)  
{
```

```

        for( i=0;i<=500;i++)
        {
            glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);
            glPushMatrix();
            glRotatef(i,1,1,1);
            auxWireCube(100);
            glPopMatrix();
            Sleep(70);
            glFlush();
        }
    }

```

Aplicația 3. Se dă sursa `cub.c`. Realizați următoarele :

- descoperiți eroarea de modelare a cubului;
- activați testul de ascundere;
- tăiați fețele față;
- faceți o funcție de desenare pentru cub;
- rotiți cubul folosind una dintre cele două variante.

5.4 Reuniunea a două corpuri

Pentru a se realiza reuniunea dintre două corpuri care au zone comune în spațiu, este suficient să se activeze ascunderea suprafețelor.

Aplicația 4. În aplicația 3 se va adăuga și sfera. Se vor reprezenta obiectele fără activarea tăierii fețelor, cu activarea tăierii fețelor și tăind fețele față. Se utilizează funcțiile `auxWireSphere()` sau `auxSolidSphere(0.6)`.

Indicație: pentru ca imaginea să nu clipească se folosesc două buffere de culoare, unul pentru scriere și unul pentru afișare în memoria video. În funcția de inițializare a ferestrei se va pune opțiunea `AUX_DOUBLE` în loc de `AUX_SINGLE`. Pentru comutarea celor două buffere, se utilizează funcția `auxSwapBuffers()` la sfârșitul rutinei `Display()`.

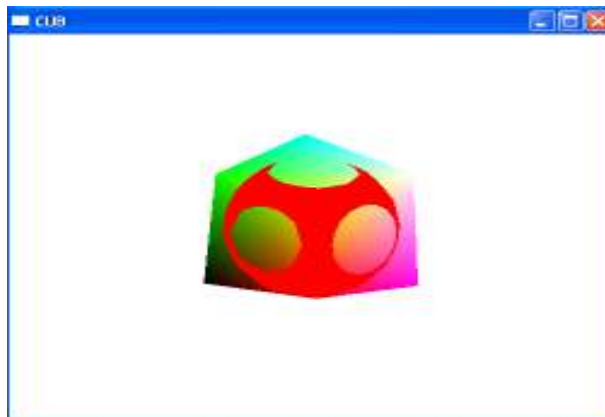


Fig. 3