

2. Primitive OpenGL

2.1 Suport teoretic

Primitivele de desenare ale bibliotecii OpenGL (punct, linie, triunghi, patrulater, poligon) pot fi reprezentate în diferite moduri selectabile (cu diferite atribute: culoare, stil linie, grosime linie, etc.). Există moduri specifice punctelor, liniilor, și poligoanelor. Fiecare setare poate fi realizată independent, fără a afecta celelalte setări (deși multe setări pot interacționa pentru a determina ceea ce se redă în framebuffer). Se specifică toate setările corespunzătoare unei primitive, apoi se specifică primitiva, iar OpenGL va reprezenta primitiva cu setările corespunzătoare.

Spre exemplu:

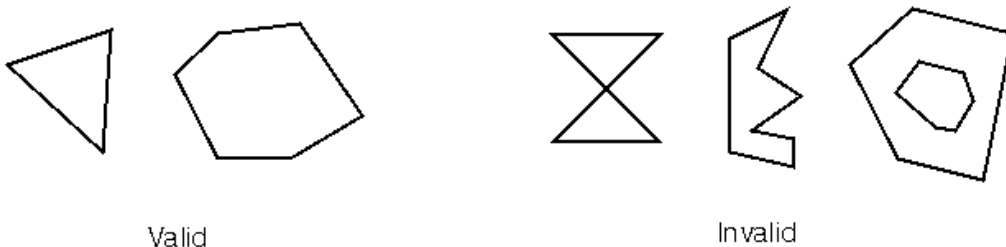
```
glPointSize(5);           // dimensiunea pixelului 5
glColor3f(1.0, 0, 0);     // culoare roșie
glBegin(GL_POINTS);
    glVertex2f(0.0, 0.0);
glEnd();
```

Acest cod va desena un punct roșu în originea sistemului de coordonate și de dimensiune 5x5 pixeli.

Primitivele sunt descrise de un grup ce conține unul sau mai multe vârfuri (vertex-uri). Un vârf definește un punct, capetele unui segment de linie sau un „colț” al unui poligon (locul unde două linii se întâlnesc). Datele asociate unui vârf sunt: coordonate carteziene, culori, normale la suprafață, coordonate de texturare. Fiecare vârf este procesat independent, în ordine și în același mod. Singura excepție de la regulă este cazul în care un grup de vârfuri trebuie decupate față de volumul de vizualizare. În această situație datele asociate vârfurilor pot fi modificate și se creează noi vârfuri.

Restricții pentru vârfurile poligoanelor:

- ordinea în care se dau vârfurile este sensul trigonometric;
- toate vârfurile trebuie să fie în același plan;
- poligoanele concave și figurile răsucite nu sunt recunoscute.



Primitivele și vârfurile lor se vor specifica într-un corp `glBegin()/glEnd()`;

```
glBegin( );
    .
    .
glEnd( );
```

Parametrul funcției `glBegin()` este numele primitivei care se reprezintă și care poate fi:

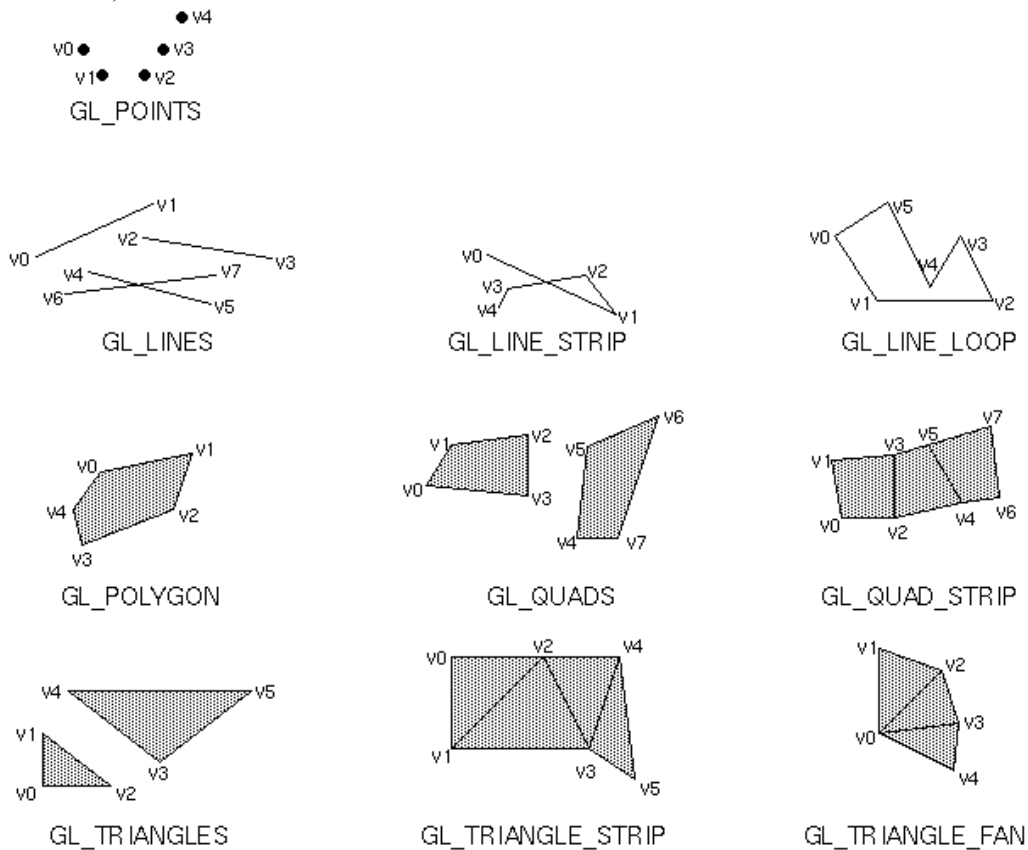
1. Pentru primitive fără atribut de umplere:

`GL_POINTS`
`GL_LINES`
`GL_LINE_STRIP`
`GL_LINE_LOOP`

2. Pentru primitive cu atribut de umplere:

`GL_TRIANGLES`
`GL_TRIANGLE_STRIP`
`GL_TRIANGLE_FAN`
`GL_QUADS`
`GL_QUADS_STRIP`
`GL_POLYGON`

Explicitarea primitivelor enumerate mai sus (vârfurile sunt date în sensul acelor de ceasornic):



În corpul `glBegin()/glEnd()` pot fi utilizate funcțiile:

- `glVertex#()` - setează coordonatele carteziane;
- `glNormal#()` - indică coordonatele normalei la suprafață;
- `glColor#()` - indică culoarea de desenare;
- `glTexCoord#()` - setează coordonatele pentru texturare.

Caracterul „#” desemnează numărul parametrilor funcțiilor și tipul acestora.

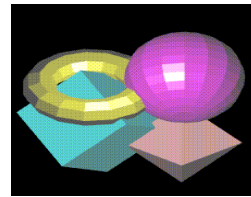
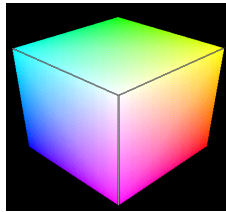
Exemplu: `glVertex2f(x, y)` – se dau 2 coordonate de tip float;
`glVertex3d(x, y, z)` – se dau 3 coordonate de tip double;
`glVertex4v` – se dă un pointer spre un tablou ce conține 4 elemente (coordonate omogene).

De reținut că OpenGL lucrează cu punctele implementate ca vectori [x y z w].

Funcția `glPolygonMode(GL_FRONT_AND_BACK, GL_FILL)` setează modul de redare al poligoanelor(`GL_FILL`– cu atribut de umplere, `GL_POINTS`– conturat cu puncte, `GL_LINES`– conturat cu linii). Primul parametru arată căror poligoane li se aplică setarea:

- poligoanelor față: `GL_FRONT`;
- poligoanelor spate: `GL_BACK`;
- poligoanelor față-spate: `GL_FRONT_AND_BACK`.

Exemple de obiecte modelate în OpenGL:



2.2 Desfășurarea lucrării

Utilizând primitivele bibliotecii OpenGL modelați următoarele corpuri:

Aplicație 1. Desenați un cerc folosind succesiv ca primitive punctul, linia și poligonul (se folosesc 360 de puncte).

Exemplu:



Aplicație 2. Modelați un cilindru (prin centrul cercului trece axa y, iar cilindrul este înfășurat de dreptunghiuri). Se va utiliza primitiva `GL_QUAD_STRIP`.

Aplicație 3. Modelați un con (vârful conului este un punct aflat pe una din axe). Se va utiliza primitiva `GL_TRIANGLE_FAN`.

Aplicație 4. Desenați un paralelipiped folosindu-vă de cilindru (din cele 360 de puncte se aleg numai 4 puncte aflate la 90 grade unul față de celălalt).