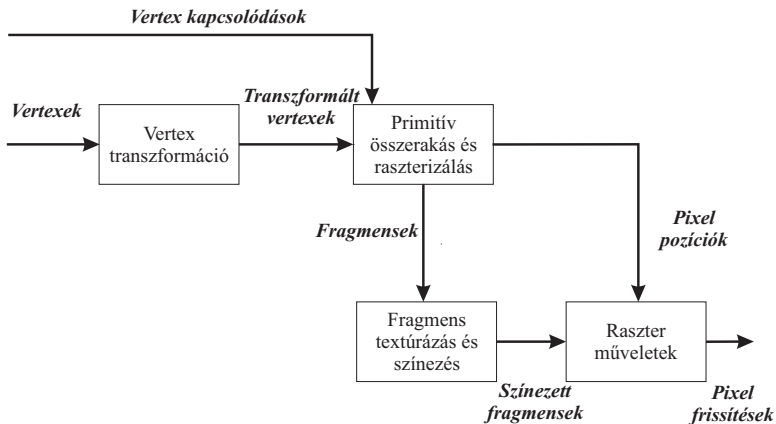


Grafikus csővezeték és az OpenGL függvénykönyvtár

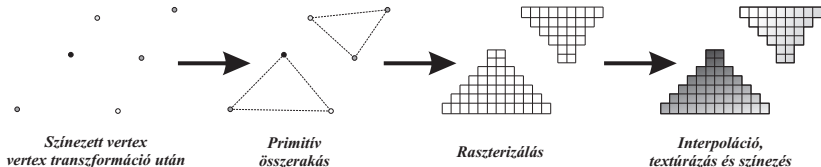
- 3D-s színtér objektumainak leírása primitívekkel:
 - pontok,
 - élek,
 - poligonok.
- Primitívek szögpontjait vertexeknek nevezzük
- Adott sorrendben végrehajtott műveletek segítségével áll elő a 2D-s kép
- Műveletek sorrendjét grafikus csővezetéknek nevezzük
 - Rögzített műveleti sorrendű grafikus csővezeték
 - Programozható grafikus csővezeték

Rögzített műveleti sorrendű grafikus csővezeték

Rögzített műveleti sorrendű grafikus csővezeték



- Vertex transzformációk
 - Matematikai műveletek sorozata
 - Primitívek szögpontjainak meghatározása a képernyőn
 - Vertex attribútumok átadása

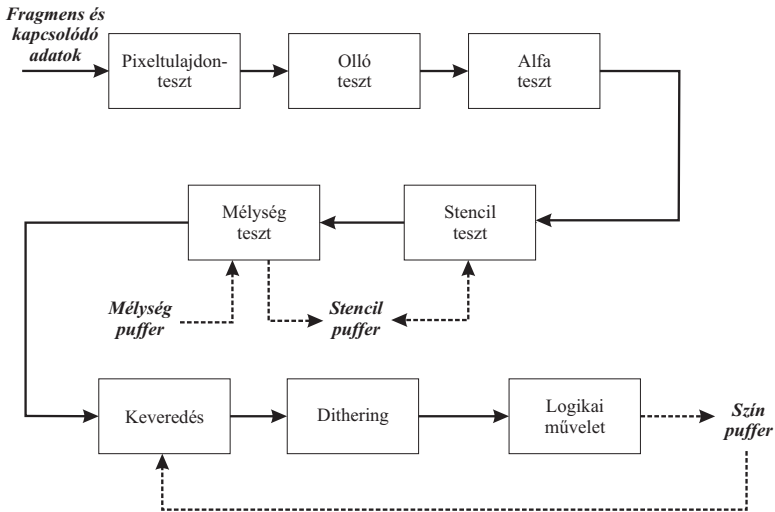


- Primitív összerakás és raszterizálás
 - A vertexek primitívekbe vannak szervezve kapcsolódási információk alapján
 - Vágás
 - A 3D-s színtér látható térfogata
 - Alkalmazás által definiált vágósíkok
 - Raszterizáló eldobhat poligonokat
 - Geometriai primitívek lefedése
 - Fragmensek

- Fragmens textúrázás és színezés
 - Mindegyik fragmensre textúrázás és matematikai műveletek végrehajtása
 - Transzformált vertexekből származó interpolált szín
 - Interpolált textúra koordináták
 - Fragmenshez tartozó texel kinyerése
 - Fragmens szín

Rögzített műveleti sorrendű grafikus csővezeték

Raszterműveletek



- Pixeltulajdon-teszt
 - Képernyő adott pixelére írhatunk-e
- Olló teszt
 - Olló téglalapra korlátozott kirajzolás
- Alfa teszt
 - Fragmens alfa értékének összehasonlítása egy előre megadott értékkel
 - Adott reláció mellett kapott hamis értéknél a fragmens eldobódik

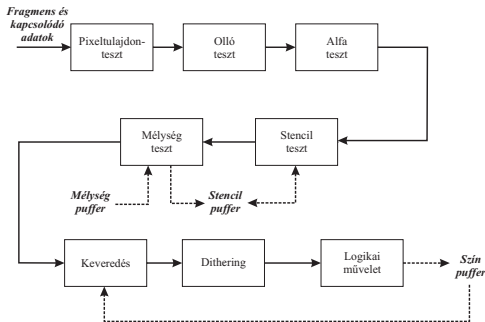
- Stencil-teszt
 - Fragmens pozíciójának megfelelő stencilpuffer érték összehasonlítása egy előre megadott értékkel
 - Ha az összehasonlítás eredménye hamis, akkor a fragmens eldobódik
 - Stencilpuffer értékének módosítása
 - Műveletek megadása sikeres és sikertelen stencil teszt esetén
 - Sikeres stencil teszt és a mélység tesztől függő művelet megadása

- Mélység teszt
 - Fragmens mélység értékének összehasonlítása a mélységpufferben levő értékkel
 - Sikeres teszt esetén frissül a színpuffer és a mélységpuffer
 - Alap esetben a nézőponthoz közelebbi fragmens fog bekerülni a színpufferbe

Rögzített műveleti sorrendű grafikus csővezeték

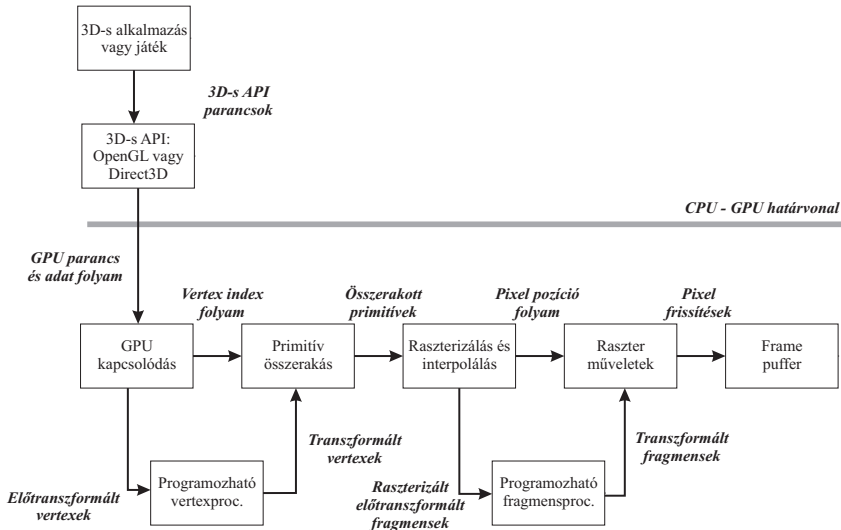
Raszterműveletek

- Keveredés: végső fragmens és pixelek egyesítése
- Dithering: a színmélység javítása a térbeli felbontás rovására
- Logikai műveletek: OR, XOR vagy INVERT



Programozható grafikus csővezeték

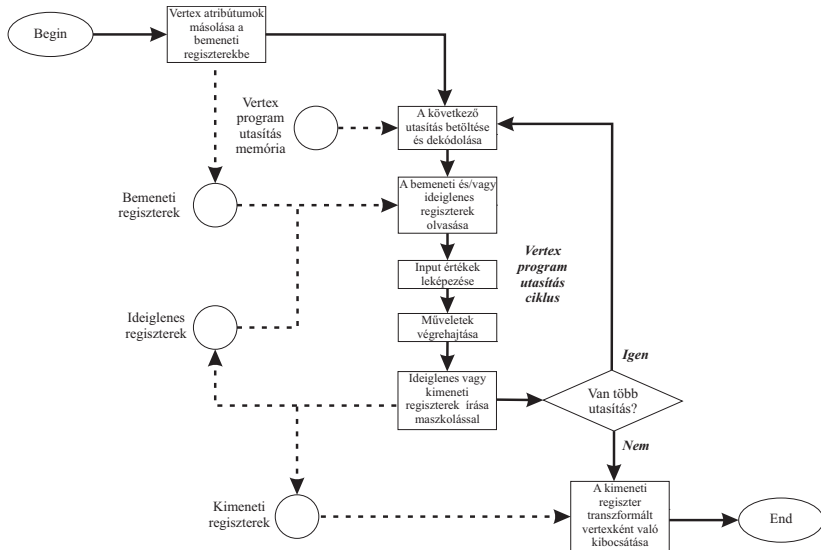
Programozható grafikus csővezeték



- Programozható vertex processzor
 - Vertex attribútumok betöltése a megfelelő regiszterekbe
 - Vektorokon végzett matematikai műveletek
 - Fejlettebb vertex processzorok a vezérlési szerkezeteket is támogatják
- Programozható fragmens processzor
 - Hasonló műveletek végrehajtása, mint a vertex processzorok esetén
 - A fragmens processzorok támogatják a textúra műveleteket is

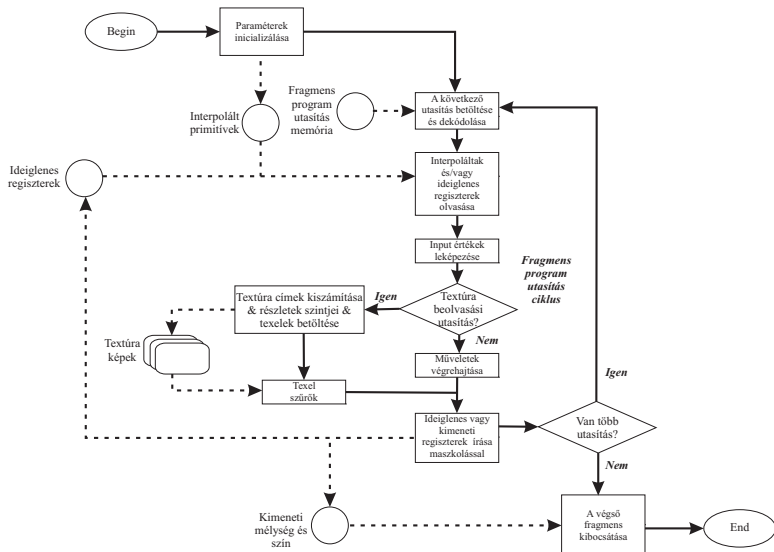
Programozható grafikus csővezeték

Programozható vertex processzor



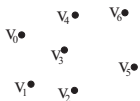
Programozható grafikus csővezeték

Programozható fragmens processzor

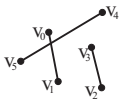


OpenGL függvénykönyvtár

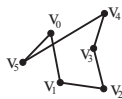
- Hordozható, 3D-s grafikus függvénykönyvtár
- Több száz függvényt és definíciót tartalmaz
- Egy szintér leírásához OpenGL függvény hívások sorozatát kell megadni
- Vertex
 - OpenGL primitívek szögpontjai
 - 2D-s és 3D-s pozíciók
 - Meghatározzák a primitív alakját és helyzetét



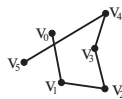
Pontok



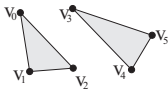
Vonalak



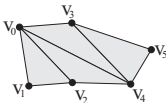
Vonal hurok



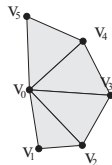
Töredezett vonal



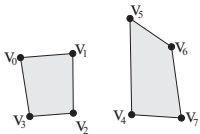
Háromszögek



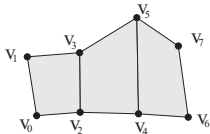
Háromszögsáv



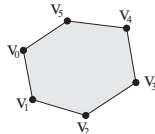
Háromszög-legyező



Négyszögek



Négyszögsáv



Poligon

- Az OpenGL függvénykönyvtár támogatja a
 - Megvilágítást
 - Árnyalást
 - Textúrázást
 - Keveredést
 - Átlátszóságot
 - Más speciális hatásokat és képességeket
- Az OpenGL függvénykönyvtár nem tartalmaz
 - Ablakkezelő függvényeket
 - Felhasználói interaktivitást és I/O műveleteket végrehajtó függvényeket
- Nincs OpenGL file formátum
 - A modellek tárolására
 - A virtuális színtér tárolására

OpenGL adattípus	Belső reprezentáció	C adattípusként definiálva
GLbyte	8 bites egész	signed char
GLshort	16 bites egész	short
GLint, GLsizei	32 bites egész	long
GLfloat	32 bites lebegőpontos	float
GLclampf	pont	
GLuint, GLenum, GLbitfield	32 bites előjel nélküli egész	unsigned long

<KÖNYVTÁR PREFIX><ALAP PARANCS><OPCIONÁLIS ARGUMENTUM
SZÁM><OPCIONÁLIS ARGUMENTUM TÍPUS>

gl Color 3 f

glColor3f(0.5, 0.5, 0.5);

- Platformfüggetlenség
 - Operációs rendszerekhez kapcsolódó feladatok
 - Ablakkezelés
 - Felhasználói interakciók kezelése
 - Felhasználó leütötte-e az Enter billentyűt?
- GLUT használata
 - Kezdetekben AUX (auxiliary) lib
 - Kereszt-platformos példák szemléltetése
 - Pop-up menük, ablakok, joystick támogatás, stb.
 - GUI programozása adott platformon

```
#include <GL/glut.h>

// a színtér rajzolása
void RenderScene(void)
{
// Az aktuális törlő színnel való ablak törlés
    glClearColor(GL_COLOR_BUFFER_BIT);

// Flush rajzoló parancs
    glFlush();
}

// A renderelési állapotok beállítása
void SetupRC(void)
{
//A színpuffer törlőszínének a beállítása
    glClearColor(0.0f, 0.0f, 1.0f, 1.0f);
}
```



```
// A program belépési pontja
int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
    glutCreateWindow("Egyszeru");
    glutDisplayFunc(RenderScene);
    SetupRC();
    glutMainLoop();
    return 0;
}
```

```
glutInit(&argc, argv);
```

- Továbbítja a parancssori paramétereket
- Inicializálja a GLUT függvénykönyvtárat

```
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
```

- Egyszeresen pufferezt ablak
- RGBA színmód

```
glutCreateWindow ( "Egyszeru" );
```

- Ablak létrehozása
- Címsorában az "Egyszeru" felirat

```
glutDisplayFunc ( RenderScene );
```

- RenderScene callback függvény regisztrálása
- Ablak újrarajzolása
 - Ablak első megjelenítésekor
 - Ablak előtérbe helyezésekor
- OpenGL renderelési függvények hívása

SetupRC () ;

- Renderelése előtti inicializálás
- OpenGL állapotok beállítása

glutMainLoop () ;

- GLUT eseménykezelő elindítása
- Vezérlés átadása a GLUT-nak
- Fő ablak bezárásig nem tér vissza
- Üzenetek feldolgozása

```
glClearColor(0.0f, 0.0f, 1.0f, 1.0f);
```

- Ablak törlésére használt szín megadása
- A színpuffer inicializálására használt szín beállítása

```
void glClearColor(GLclampf red, GLclampf green,  
                  GLclampf blue, GLclampf alpha);
```

- GLclampf: 0 és 1 közé leképezett float
- A szín vörös, zöld és kék összetevők keverékeként való megadása
- alpha: keveredés és speciális hatások

```
glClear (GL_COLOR_BUFFER_BIT);
```

- A színpuffer törlése
- Pufferek törlése

```
glFlush ();
```

- OpenGL parancssor ürítése
- Nem vár további utasításokra
- Beérkezett utasítások feldolgozásának folytatása

Grafikus csővezeték

- Műveletek meghatározott sorrendje
- Adatok áramlása egyik fázisból a másikba
 - Meghatározott típusú be- és kimenő adatok
 - SIMD
- Programozható grafikus csővezeték
 - Rögzített műveleti sorrendű grafikus csővezeték „kiegészítése”
 - Programozható vertex és fragmens processzor

OpenGL függvénykönyvtár

- 3D-s grafikus függvénykönyvtár
- Színtér leírása függvények meghívásával
- Rögzített műveleti sorrendű grafikus csővezeték
- GLUT
- Első program