

# Tehnologia OpenGL

**OpenGL (Open Graphics Library)** este o specificație a unui standard care definește un API (*Application Programming Interface*) multiplatformă foarte utilizat pentru programarea componentelor grafice 2D și 3D ale programelor de calculator. Interfața constă în peste 250 de apeluri diferite care folosesc la a desena pe ecranul calculatorului scene 3D complexe din primitive (elemente simple). OpenGL a fost inițial dezvoltat de compania Silicon Graphics, Inc. (SGI) în 1992 și este foarte utilizat în grafică asistată de calculator, realitate virtuală, vizualizare științifică, simulări de zboruri sau jocuri pe calculator. Acest ultim domeniu este în strânsă competiție cu tehnologia DirectX de la Microsoft (compară OpenGL cu DirectX3D). Proiectul OpenGL este condus de compania Khronos Group, un consorțiu tehnologic non-profit.

Funcția de bază a OpenGL este de a accepta primitive, cum ar fi puncte, linii și poligoane, și de a le converti în pixeli. Acest lucru se face printr-o conductă grafică (graphics pipeline), cunoscută sub numele de mașină de stare OpenGL. Cele mai multe comenzi OpenGL primitive, sunt fie probleme la conducta grafică, sau configurarea felului în care aceste procese de conducte de primitive. Înainte de introducerea OpenGL 2.0, fiecare etapă din conductă efectua o funcție fixă și a fost configurabilă numai în limite restrânse. OpenGL 2.0 oferă mai multe etape, care sunt pe deplin programabile folosind GLSL.

OpenGL este un API procedural de nivel mic, care necesită ca un programator să impună măsurile exacte necesare pentru a face o scenă. Acest lucru contrastează cu alte API-uri, în care un programator are nevoie doar pentru a descrie o scenă și poate lăsa biblioteca să gestioneze detaliile redând finalul scenei. Un OpenGL de nivel mic impune programatorilor să aibă o bună cunoaștere a conduitei grafice dar oferă și o anumită libertatea de a pune în aplicare algoritmi noi de redare.

Un automat cu stări finite este un model matematic utilizat pentru a descrie un sistem care se află într-o stare specifică la un moment dat și care poate trece la alte stări pe baza unor intrări sau condiții specifice. În contextul OpenGL, FSM este utilizat pentru a gestiona stările care controlează toate aspectele de desenare printre care:

- Parametrii de stare: OpenGL menține un set de stări interne care includ informații despre culoare, tipuri de primitivă, configurații de blending, modul de iluminare, setările de texturare;
- Tranziții: Acestea sunt schimbările între diferitele stări, determinate de apelurile funcțiilor OpenGL (de exemplu, funcții de setare a culorii, activarea sau dezactivarea blending-ului, etc.);
- Alte operațiuni OpenGL (de exemplu, desenarea, iluminarea, texturarea) ce sunt influențate de setările de stare curente.

Modelul FSM din OpenGL este compus dintr-un număr mare de stări, organizate în mai multe categorii, fiecare având un rol esențial în procesul de randare:

Stări de geometrie și primitivă: Determină modul în care sunt desenate primitivele (puncte, linii, triunghiuri). Setările includ modul de poligonizare și culling.

- Stări de iluminare: Controlează efectele de iluminare asupra obiectelor, cum ar fi sursele de lumină, materialele și setările de reflecție.
- Stări de texturare: Permit definirea texturilor care vor fi aplicate pe obiectele 3D. Acestea implică filtrarea, aplicarea de texturi multiple și modurile de texturare;
- Stări de blending: Controlează modul în care culorile obiectelor se amestecă cu fundalul sau cu alte obiecte, utilizând funcții de blending pentru transparență și efecte speciale;
- Stări de matrice și transformare: Acestea sunt utilizate pentru a manipula poziționarea și orientarea obiectelor în scenă, utilizând matrici de modelare, vizualizare și proiecție;

În timpul procesului de randare, modelul FSM dictează ce se întâmplă în fiecare etapă, conform următorilor pași:

- Setarea Stărilor: La începutul procesului de randare, aplicația setează stările relevante, determinând astfel modul în care OpenGL va procesa obiectele. De exemplu, programul poate seta starea de iluminare, activând o sursă de lumină specifică, apoi va selecta textura și modul de texturare dorit.
- Executarea Comenzilor de Randare: OpenGL randează primitivele grafice pe baza stărilor curente. De exemplu, dacă starea de blending este activată, OpenGL va aplica funcțiile de blending pentru fiecare fragment al primitivei desenate. Orice schimbare de stare efectuată între desenarea obiectelor poate afecta modul în care următorul obiect este procesat.
- Eficiența și Optimizarea Randării: Din cauza modului de funcționare al FSM-ului, schimbarea frecventă a stărilor (de exemplu, de la o stare de texturare la alta) poate fi costisitoare în termeni de performanță, deoarece necesită o sincronizare a resurselor GPU. De aceea, este esențial să se minimizeze tranzițiile de stare pentru a optimiza randarea.

Principalele avantaje ale lui OpenGL reprezintă suportul cross-platform (Windows, macOS, Linux, dispozitive iOS și Android), compatibilitatea cu diferite dispozitive, o vastă documentație disponibilă, accesul direct la hardware pentru o performanță mai ridicată, pipeline programabil începând cu versiunea 3.0 (pentru modificarea și optimizarea efectelor precum reflexiile, lumina, umbra), suportul pentru extensii, compatibilitatea cu diferite limbaje de programare (C, C++, Python, Java), suportul pentru versiuni mai vechi.

Câteva dezavantaje ale lui OpenGL ar fi necesitatea unor cunoștințe despre matematica și grafica 3D, evoluția mai lentă comparativ cu competitorii acestuia (Vulkan sau DirectX), gestionarea ineficientă a resurselor GPU ceea ce necesită o responsabilitate în plus pentru dezvoltator, necesitatea unei sincronizări mai frecvente între CPU și GPU, nu oferă opțiuni de randare multithread ceea ce poate duce la o performanță redusă, OpenGL este considerat un API high-level care gestionează automat unele aspecte ale procesului de randare (din acest motiv nu oferă acces direct și explicit la hardware-ul GPU pentru un control mai precis).