Cont GitHub: https://github.com/RealExo7

**OpenGL** (**Open Graphics Library**) este o specificație a unui standard care definește un [**API**](https://ro.wikipedia.org/wiki/API) (*Application Programming Interface*) [multiplatformă](https://ro.wikipedia.org/wiki/Multiplatform%C4%83) foarte utilizat pentru programarea componentelor grafice [**2D**](https://ro.wikipedia.org/wiki/2D) și [**3D**](https://ro.wikipedia.org/wiki/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 (din *primitives*, elemente simple). OpenGL a fost inițial dezvoltat de compania [**Silicon Graphics**, **Inc**.](https://ro.wikipedia.org/w/index.php?title=Silicon_Graphics,_Inc.&action=edit&redlink=1) (SGI) în 1992 și este foarte utilizat în grafică asistată de calculator, [realitate virtuală](https://ro.wikipedia.org/wiki/Realitate_virtual%C4%83), vizualizare științifică, simulări de zboruri sau jocuri pe calculator. Acest ultim domeniu este în strânsă competiție cu tehnologia [**DirectX**](https://ro.wikipedia.org/wiki/DirectX) de la [**Microsoft**](https://ro.wikipedia.org/wiki/Microsoft) (compară OpenGL cu [**Direct3D**](https://ro.wikipedia.org/w/index.php?title=Direct3D&action=edit&redlink=1)). Proiectul OpenGL este condus de compania Khronos Group, un consorțiu tehnologic non-profit.

Precum oricare altă tehnologie, **OpenGL** are atât puncte forte, cât și puncte slabe. Câteva puncte tari ale **OpenGL** sunt:

-**Portabilitatea**: OpenGL este proiectat să funcționeze pe o varietate de platforme, inclusiv Windows, macOS, Linux, iOS și Android. Acest lucru face ca aplicațiile dezvoltate cu OpenGL să fie ușor de portat între aceste medii.

-**Performanță**: OpenGL oferă un acces direct la resursele hardware ale GPU-ului, ceea ce înseamnă că poate fi foarte eficient în procesarea grafică intensivă. Acest lucru îl face potrivit pentru aplicații grafice de înaltă performanță, cum ar fi jocurile video și aplicațiile de modelare 3D.

-**Suport pentru 2D și 3D**: OpenGL este conceput pentru a funcționa atât pentru grafică bidimensională, cât și pentru cea tridimensională. Aceasta îl face extrem de versatil și potrivit pentru o gamă largă de aplicații.

-**Comunitate activă**: OpenGL are o comunitate mare și activă de dezvoltatori, ceea ce înseamnă că există o cantitate semnificativă de resurse, documentație și ajutor disponibile online.

La fel există și puncte slabe ale **OpenGL**:

-**Interfața de Nivel Scăzut**: OpenGL este o interfață de nivel scăzut, ceea ce înseamnă că necesită o cunoaștere detaliată a graficii computerizate pentru a fi folosit eficient. Acest lucru poate face dezvoltarea cu OpenGL mai dificilă și mai laborioasă în comparație cu tehnologii de nivel mai înalt.

-**Gestionarea Stării și a Resurselor**: În OpenGL, dezvoltatorul trebuie să gestioneze manual multe aspecte ale stării grafice, cum ar fi texturile, vertex array-urile și alte resurse. Aceasta poate fi o sursă de erori subtile și complexitate suplimentară.

-**Lipsa Funcționalităților Specifice Aplicației**: OpenGL furnizează un set de funcționalități de bază pentru randarea grafică, dar nu include un suport nativ pentru aspecte specifice aplicației, cum ar fi fizica, sunetul sau rețeaua. Dezvoltatorul trebuie să integreze aceste funcționalități separat.

-**Retrocompatibilitate și Modernizare**: OpenGL a suferit de-a lungul timpului diferite actualizări și modificări, iar unele funcționalități vechi pot să nu mai fie suportate pe dispozitive și drivere mai recente. Aceasta poate necesita o gestionare atentă a compatibilității între versiuni.

-**Concurență cu API-uri Concurate**: Pe măsură ce tehnologii precum Vulkan și DirectX12 au devenit mai populare, OpenGL a pierdut o parte din atractivitate, mai ales în industria jocurilor, datorită avantajelor lor în ceea ce privește performanța și controlul asupra hardware-ului.

Putem face o analogie pentru a înțelege modul în care funcționează OpenGL, folosind conceptul de automat cu stări finite:

**Stările OpenGL**: OpenGL funcționează într-o manieră "stare-dată". Acest lucru înseamnă că, în orice moment, există o anumită stare a motorului grafic. Această stare poate include, de exemplu, atributele materialelor, modul de desenare (puncte, linii, triunghiuri etc.), modul de umbrire (vertex shader, fragment shader etc.).

**Comenzile OpenGL**: Comenzile OpenGL sunt echivalentele tranzitiilor într-un automat cu stări finite. Acestea sunt apeluri de funcții prin care dezvoltatorul spune motorului grafic ce să facă. De exemplu, glDrawArrays este o comandă care spune OpenGL să deseneze primitive grafice (cum ar fi triunghiuri) utilizând datele dintr-un vertex array.

**Tranzitiile între Stări**: Atunci când dezvoltatorul face o apelare la o funcție OpenGL, aceasta poate schimba starea motorului grafic. De exemplu, dacă un program shader este activat prin glUseProgram, acest lucru schimbă modul în care OpenGL procesează geometria și fragmentele.

**Procesul de Rendare**: Procesul de randare în OpenGL poate fi văzut ca o serie de tranziții de stare și comenzi care manipulează datele geometrice și texturale pentru a obține imaginea finală care va fi afișată pe ecran. Acest lucru implică setarea stărilor, încărcarea datelor, activarea programelor shader, configurarea texturilor, precum și desenarea propriu-zisă.

**OpenGL** utilizează concepte similare cu cele ale unui automat cu stări finite pentru a gestiona procesul de randare 3D, oferind astfel dezvoltatorului un control detaliat asupra modului în care este generată grafica. Cu toate acestea, este afectat și procesul de randare al scenei 3D:

-**Controlul Finațat al Rendării**: Prin utilizarea stărilor și comenzilor, OpenGL oferă dezvoltatorului un control fin asupra modului în care sunt desenate obiectele în scenă. Acest lucru permite optimizări și efecte grafice complexe.

-**Complexitatea Dezvoltării**: Pe de altă parte, necesitatea gestionării stărilor și comenzilor poate face dezvoltarea în OpenGL mai complexă. Trebuie să se aibă grijă să se mențină starea corectă și să se folosească comenzile în ordinea potrivită pentru a obține rezultatele dorite.

-**Eficiență și Performanță**: Datorită controlului fin asupra stării, dezvoltatorii pot face optimizări pentru a obține performanțe maxime din hardware-ul grafic disponibil. Aceasta este crucială pentru aplicațiile 3D interactive și jocurile video.

În concluzie, **OpenGL** rămâne o tehnologie puternică și relevantă în domeniul graficii computerizate, dar alegerea sa depinde de cerințele specifice ale proiectului și nivelul de cunoștințe al dezvoltatorului.