**Tehnologia OpenGL**

OpenGL (Open Graphics Library) este o bibliotecă grafică destinată randării grafice 2D și 3D, creată la începutul anilor '90 de Silicon Graphics. De-a lungul timpului, OpenGL a devenit un standard deschis, fiind menținut de consorțiul Khronos Group. Acesta este utilizat în mod extins pentru dezvoltarea graficii pentru jocuri, aplicații de simulare, design 3D și, în general, pentru aplicații care necesită o interfață grafică avansată.

Tehnologia OpenGL este recunoscută pentru portabilitatea sa – funcționează pe diferite sisteme de operare (Windows, macOS, Linux) și este susținută de o mare varietate de hardware grafic. Pe lângă OpenGL clasic, au apărut variante derivate, cum ar fi **OpenGL ES** (pentru dispozitive mobile și încorporate), **WebGL** (pentru randarea graficii 3D în browser), și mai recent, **Vulkan**, care preia o mare parte din conceptele OpenGL, dar oferă un control mai mare asupra procesului de randare, fiind mult mai aproape de hardware.

OpenGL are două scopuri principale:

* să mascheze (ascundă) complexitatea interfețelor cu diferite acceleratoare 3D, prin confruntarea programatorului cu un singur API uniform.
* să mascheze capabilitățile diferitelor platforme [hardware](https://ro.wikipedia.org/wiki/Hardware), prin cerința ca toate implementările să accepte OpenGL ca un set complet (cu ajutorul emulării de software, dacă este necesar).

# **Cum explicați modelul de automat cu stări finite al OpenGL și cum afectează acest lucru procesul de randare al scenei 3D de către biblioteca grafică/API?**

Un **automat cu stări finite (FSM)** este un model în care sistemul poate fi într-una dintre mai multe stări definite. În contextul OpenGL, aceste stări definesc comportamentul randării 3D și controlează cum sunt procesate obiectele în cadrul pipeline-ului grafic.

OpenGL gestionează procesul de randare printr-un set de **stări globale** (de exemplu, activarea testului de adâncime sau a blending-ului). Aceste stări afectează toate obiectele randate ulterior, fără a fi nevoie să le specifici individual. Astfel, OpenGL trece prin diverse stări ale pipeline-ului grafic, iar fiecare stare influențează modul în care sunt procesate obiectele (de exemplu, dacă sunt vizibile, transparente sau texturate).

**Cum afectează acest model procesul de randare?**

1. **Simplifică randarea**: Odată ce o stare este setată, aceasta se aplică automat tuturor obiectelor randate, economisind timp și reducând complexitatea.
2. **Impact asupra performanței**: Modificarea frecventă a stărilor poate afecta performanța, mai ales în aplicații complexe. Schimbările de stare impun recalcularea unor pași din pipeline, ceea ce poate duce la overhead.
3. **Control limitat**: Deși eficient, modelul FSM limitează controlul detaliat asupra fiecărei etape de randare. Pentru aplicații care necesită un control mai mare (de exemplu, jocuri moderne), API-uri precum **Vulkan** oferă mai multă flexibilitate.

**Concluzie**

Modelul de automat cu stări finite al OpenGL simplifică dezvoltarea aplicațiilor grafice, dar poate duce la pierderi de performanță dacă stările sunt schimbate frecvent. În aplicațiile moderne, un control mai detaliat al pipeline-ului poate fi obținut prin API-uri precum Vulkan.

Bibliografie:

<https://ro.wikipedia.org/wiki/OpenGL>