**Tehnologia OpenGL**

OpenGL sau Open Graphics Library este un API (Application Programming Interface) creat pentru randarea graficii 2D si 3D, API-ul interactioneaza cu GPU-ul pentru a obtine o randare a graficii accelerata de hardware. Acest API ofera dezvoltatorilor un set de functii si comenzi pentru a comunica direct cu GPU-ul, permitand randarea obiectelor grafice, texturilor si efectelor vizuale complexe. OpenGL este versatil si poate fi folosit cu multiple limbaje de programare si pe diverse sisteme de operare precum Windows, Linux si MacOS. Acesta este de obicei folosit pentru jocuri video(dezvoltatorii de jocuri folosesc OpenGL alaturi de game engine-uri precum Unity sau Unreal Engine), modelare stiintifica, CAD, virtual reality, simulatoare de zbor, etc.

OpenGL prezinta multe calitati atractive, cateva exemple sunt:

* Este un API deschis si standardizat sustinut de OpenGL Architecture Review Board ceea ce confera actualizari constante si compatibilitate.
* Este compatibil cu multe limbaje de programare cum ar fi C++(original dezvoltat doar pentru C++), Java, Python etc.
* Permite accesul direct la GPU oferind performante ridicate
* Este extrem de popular si cunoscut cu o comunitate larga si nenumarate biblioteci, resurse si tutoriale.

OpenGL prezinta si unele calitati mai putin dorite cum ar fi:

* Este considerat usor imbatranit comparativ cu DirectX si Vulkan iar design-ul sau API este considerat invechit si mai putin intuitiv comparat cu alte solutii.
* Performanta si stabilitatea depind de drivele GPU.
* Lipsa functionalitatilor moderne, desi OpenGL primeste actualizari acestea nu se ridica la nivelul altor tehnologi in ceea ce priveste functii avansate de control direct al hardware-ului si optimizarea resurselor.

Arhitectura OpenGL se bazează pe un model client-server. Un program de aplicație scris pentru a utiliza API-ul OpenGL este „clientul” și rulează pe CPU. Implementarea engine-ului grafic OpenGL (inclusiv programele de shader GLSL) este „serverul” și rulează pe GPU. Geometria și multe alte tipuri de atribute sunt stocate în buffere numite Vertx Buffer Objects (sau VBO). Aceste buffere sunt alocate pe GPU și umplute de programul CPU. Modelarea, randarea și interacțiunea sunt un proces de cooperare între programul de tip “client” CPU și cel de tip “server” GPU scris în GLSL.

# 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 stari finite este un sistem care poate fi intr-o singura stare la un moment dat si care trece intre diverse stari pe baza unor evenimente sau intrari specifice. In cazul OpenGL, aceste stari sunt reprezentate de configurarile active la un moment dat: modurile de shading, texturile, buffer-ele, modurile de blending, modurile de culling, etc.

Automatului cu stari finite din OpenGL influenteaza procesul de randare 3D prin faptul ca fiecare etapa a pipeline-ului grafic depinde de starile curente (cum ar fi shaderele active, texturile, matricele de transformare), iar schimbarea acestor stari modifica modul in care obiectele 3D sunt procesate si afisate pe ecran, fara a pastra un istoric al starilor anterioare.

# Concluzie

In final opinia mea despre tehnologia OpenGL este una foarte pozitiva pentru mai multe motive. Apreciez foarte mult faptul ca a pus bazele graficii 2D/3D la inceputul comercializarii calculatoarelor pentru publicul larg si a oferit o modalitate accesibila pentru crearea a nenumarate jocuri video si aplicatii grafice complexe. Fara OpenGL multiple tipuri de aplicatii(mentionate mai sus) ar fi stagnat sau folosit tehnici inferioare pentru randarea graficii.

Bibliografie :

<https://en.wikipedia.org/wiki/OpenGL#Vulkan>

<https://people.eecs.ku.edu/~jrmiller/Courses/OpenGL/Architecture.html>