

Temă Laborator 1

OpenGL

Nume : Pentelescu Claudiu Gabriel

Grupa : 3132a

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. OpenGL este portabilă, fiind disponibilă pentru pe o varietate de sisteme cum ar fi PC, MacIntosh, Silicon Graphics, UNIX, Linux, Irix, Solaris, HP-UX. OpenGL și rulează pe principalele sisteme de operare: MacOS, OS/2, UNIX, Windows, Linux, OPENStep, Python și BeOS. Pe parte de implementare, OpenGL poate fi apelat din limbajele de programare C, C++, Java, FORTRAN și Ada.

Din punctul meu de vedere, cred că tehnologiile grafice au fost influențate în mare măsură de progresele înregistrate în viteza calculatorului și întrucât accelerația grafică implică aspectele toate ale unui computer (aspectul arhitecturii electronice, aspectul OS, aspectul CPU, aspectul de compatibilitate API), implică și o muncă de inginerie foarte densă și detaliată, necesitând resurse enorme, astfel încât aceste tehnologii să nu poată fi eliberate, ci să evolueze și să fie "ușoare" pentru dezvoltatorii de jocuri și/sau motoare 3D.

Având o proiecție asupra ceea ce face OpenGL (cu excepția cazului în care este o implementare software, desigur) ,identific simplul fapt de a lua tot ce-i spui să facă și să împingă toate datele, toate remiterile, aproape totul pe placa grafică și să spună GPU-ului să-și facă treaba. GPU-ul este creat special pentru acest tip de serviciu. Înmulțirea numerelor cu puncte în virgulă (ceea ce faceți mult atunci când desenați o scenă 3D) și executați shadere. Și acest lucru în paralel. Doar pentru a vă simți cât de repede este GPU, gândiți-vă la o scenă simplă în 3D în ecran complet cu 1920x1080 pixeli. Acestea sunt înmulțite, 2.073.600 de pixeli pentru a desena. Pentru fiecare pixel , unitatea GPU va rula cel puțin o dată shader-ul de fragmente, de cele mai multe ori de mai multe ori.

Având în vedere o creație de context OpenGL este un proces destul de complex și având în vedere că acesta variază între sistemele de operare, crearea automată a contextului OpenGL a devenit o caracteristică comună a mai multor biblioteci de dezvoltare de jocuri și interfețe utilizator, inclusiv SDL, Allegro, SFML, FLTK, și Qt. Câteva biblioteci au fost proiectate exclusiv pentru a produce o fereastră compatibilă cu OpenGL. Prima astfel de bibliotecă a fost OpenGL Utility Toolkit (GLUT), ulterior înlocuit de freeglut. GLFW este o alternativă mai nouă. Aceste seturi de instrumente sunt concepute pentru a crea și gestiona ferestrele OpenGL și pentru a gestiona intrarea, dar puțin dincolo de asta. GLFW - O fereastră multiplatforma și un manipulator de tastatură-mouse-joystick; este mai orientat spre joc freeglut - O fereastră

multiplatformă și un manipulator de tastatură-mouse; API-ul său este un superset al API-ului GLUT și este mai stabil și actualizat decât GLUT OpenGL Utility Toolkit (GLUT) - Un vechi handler de fereastră, care nu mai este întreținut. Mai multe „biblioteci multimedia” pot crea ferestre OpenGL, în plus față de intrare, sunet și alte sarcini utile pentru aplicații de tip joc

Allegro 5 - O bibliotecă multimedia pe mai multe platforme cu un API C axat pe dezvoltarea jocurilor

Simple DirectMedia Layer (SDL) - O bibliotecă multimedia pe mai multe platforme cu un API C

SFML - O bibliotecă multimedia pe mai multe platforme cu un API C ++ și alte legături multiple la limbi precum C #, Java, Haskell și Go

Seturi de instrumente

Widget FLTK - O mică bibliotecă de widget-uri C ++ pe mai multe platforme

Qt - Un set de instrumente cu platforme C ++ pentru widgeturi. Oferă multe obiecte de asistență OpenGL, care chiar abstrag diferența dintre desktop GL și OpenGL ES

wxWidgets - Un set de instrumente pentru widget C ++.

Vulkan, denumit anterior „Next Generation OpenGL Initiative” (glNext), este un efort de reproiectare a motivării pentru a unifica OpenGL și OpenGL ES într-un singur API comun care nu va fi compatibil cu versiunile OpenGL existente.

Ce ofera OpenGL:

primitive geometrice: puncte, linii si poligoane;

primitive rastu;

mod de lucru RGBA;

modelarea si vizualizarea transformarilor de geometrie;

eliminarea muchiilor ascunse (invizibile);

antialiasing - tehnica netezirii muchiilor crestate prin modificarea culorii si intensitatii pixelilor

pentru ca muchiile sa apara netede, continue;

maparea texturilor - aplicarea de texturi 2D pe obiect 3D;

efecte speciale - fum, ceata si dizolvare;

evaluatori polinomiali - pentru reprezentari NURBS;

operatii cu pixeli;

stencil planes - lucrul cu portiuni de ecran;

double-buffering - lucrul simultan cu doua rânduri de imagini, una care este afisata si una care este

pregatita pentru afisare.

Daca am incerca sa reducem intregul API de OpenGL la mari concepte acestea ar fi:

•date •stari •shadere

Starile reprezinta un concept mai larg, OpenGL fiind de fapt un mare automat finit cu o multime de stari si posibilitati de a trece dintr-o stare in alta.