Abstract

3D Adventures este un scene-oriented 3D rendering engine, pe baza paradigmei programarii orientate pe obiecte si implementat in C++ si OpenGL(3.3+).

Principalul scop al aplicatiei il reprezinta realizarea jocurior, dar poate fi folosit pentru orice proiect care necesita rendare in 3D (simulatoare, filme de animatie etc.). Acest lucru este posibil datorita arhitecturii pe componente, parte integrata a engine-ului.

Pe partea de rendare, engine-ul dispune de numeroase efecte de inalta calitate ca Screen-Space Ambient-Occlusion, High-Dynamic-Range, Fast-ApproXimate-AntiAliasing s.a. realizate cu ajutorul shaderlor scrise in GLSL.

Engine-ul poate renda modele in format .obj, .3ds, .dae, .blend etc. , artistilor fiindu-le foarte usor sa exporte materiale pentru realizarea scenelor. De asemenea engine-ul accepta si heightmap-uri de tip .raw pentru rendarea de terenuri. (pentru texturare sunt folosite tehnicile splatting si bump mapping).

Arhitectura aplicatiei

Folosind programarea pe obiecte am implementat un sistem modular de management al nodurilor engine-ului. Practic, fiecare clasa care mosteneste interfata modulara poate accesa datele din oricare nod al engine-ului (UI, Pipeline etc.)

De asemenea, engine-ul este scene-oriented. Fiecare scena este compusa din mai multe entitati caracterizate prin anumite setari(environment settings, filter settings). Entitatiile sunt compuse din componente, existand o interfata dedicata adaugarii de noi componente.

Engine-ul importa foarte usor entitati noi convertindu-le din modele 3D. Procesul de adaugare este foarte simplu, presupunand doar un drag&drop.

Pentru a implementa numeroasele functionalitati de care engine-ul dispune, am integrat librarii, atat open source, cat si proprietary:

Open source:

-GLEW pentru extension loading

-GLFW pentru window management

-FreeImage pentru image loading

-Assimp pentru convertirea modelelor de modele 3D in formatul engine-ului

-CEGUI pentru UI

-Boost.Filesystem si Boost.PropertyTree pentru file management si xml parsing

Proprietary:

Granny SDK pentru incarcarea si rendarea de modele.

(Granny SDK a fost folosita in peste 2600 de jocuri si prin folosirea ei am vrut sa demonstrez capacitatea de integrare a engine-ului).

Prezentarea aplicatiei

Engine-ul este compus din 2 parti care pot fi folosite pentru dezvoltarea unei aplicatii. Prima este editorul care faciliteaza procesul crearii unei scene, punand la dispozitie o interfata grafica usor de inteles si folosit. Cea de-a doua parte este codul engine-ului care poate fi usor extins prin adaugarea si modificarea de module.

Astfel, utilizatorul creaza entitatii pe care le integreaza intr-un scenariu, ca apoi sa implementeze dinamica necesara aplicatiei prin intermediul codului.

Folosiind CEED (program creat de cei care au facut libraria CEGUI) utilizatorul poate adauga layout-uri de interfata grafica pentru a completa procesul de dezvoltare al aplicatiei.

Engine-ul vine si cu o documentatie generata cu Doxygen pentru a-i usura utilizatorului procesul de intelegere a functionalitatiilor acestuia.

Puncte forte

Consider ca principalele puncte forte ale engine-ului meu sunt versatilitatea adusa de sistemul modular(dezvoltatorul putand sa adauge orice fel de modul si sa-l conecteze de cele deja existente) si aspectul general al scenelor rendate, engine-ul folosind numeroase shadere pentru un look cat mai placut si mai interesant.

Personal, prefer acest engine in detrimentul altora din acelasi segment(ex. Ogre3D, Irrlicht) deoarce doresc sa-l folosesc pentru a realiza jocuri si nu vreau sa fiu dependent de feature-urile implementate de engine-urile mentionate. Cu alte cuvinte, nu voi fi nevoit sa astept integrarea feature-ului X de catre alte engine-uri pentru a realiza o idee.

Ghid de instalare

Instalarea engine-ului este foarte simpla, fiind necesara doar copierea folderului cu sursele si editorul in locatia dorita.

Justificarea folosirii tehnologiilor alese

Justificarea folosirii limbajului C++ in realizarea engine-ului este una simpla: **viteza!** Majoritatea AAA-urilor sunt scrise in C++, iar viteza este principalul motiv.

Urmatoarea controversa este legata de folosirea unei implementatri a specificatiei OpenGL. Cu alte cuvinte: De cu nu am folosit DirectX? Raspunsul e simplu: am avut nevoie de o solutie **cross-platform**, iar OpenGL indeplineste acest criteriu si in acelasi timp are performante asemanatoare cu DirectX.

Am folosit GLFW pentru window management deoarece este o solutie cross-platform din ce in mai populara printre dezvoltarorii de aplicatii ce implica folosirea OpenGL.

Freeimage este solutia folosita de Unity3D pentru image loading asa ca am ales sa le urmez exemplul si sa integrez si in engine-ul meu libraria.

Assimp este una dintre cele mai bune librarii open source de 3D model loading. Suporta zeci de formate de modele si este perfecta pentru nevoile engine-ului.

CEGUI este populara printre dezvoltatorii de jocuri si a fost folosita in realizarea jocului Torchlight.

Am folosit solutiile Boost pentru file management deoarece sunt stabile, cross-platform si foarte usor de folosit.

Pentru a-i oferi utilizatorului posibilitatea de a cripta anumite date am preluat aceasta implementare a algoritmului Blowfish. [http://www.codeproject.com/Articles/1400/A-C-ImplemenHYPERLINK "http://www.codeproject.com/Articles/1400/A-C-Implementation-of-the-Blowfish-Encryption-Decr"tation-of-the-Blowfish-Encryption-Decr](http://www.codeproject.com/Articles/1400/A-C-Implementation-of-the-Blowfish-Encryption-Decr)

**Ideea ce sta la baza engine-ului 3D Adventures este aceea de a pune la dispozitie uneltele necesare dezvoltarii unei aplicatii grafice complexe si consider ca utilitatea lui sta in eficienta cu care reuseste acest lucru.**

References**:**

<https://www.youtube.com/user/ThinMatrix>

<https://www.youtube.com/user/thebennybox>

<http://web.archive.org/web/20140209181347/http://www.arcsynthesis.org/gltut/>

<http://ogldev.atspace.co.uk/>

<http://www.opengl-tutorial.org/>

<http://www.videotutorialsrock.com/>

De asemea integrarea Granny SDK a fost facuta cu ajutorul documentatiei si a sample-urilor oferite de RAD Game Tools

SSAO-ul a fost realizat cu ajutorul <http://john-chapman-graphics.blogspot.ro/2013/01/ssao-tutorial.html>

HDR-ul a fost realizat cu ajutorul <http://frictionalgames.blogspot.ro/2012/09/tech-feature-hdr-lightning.html>