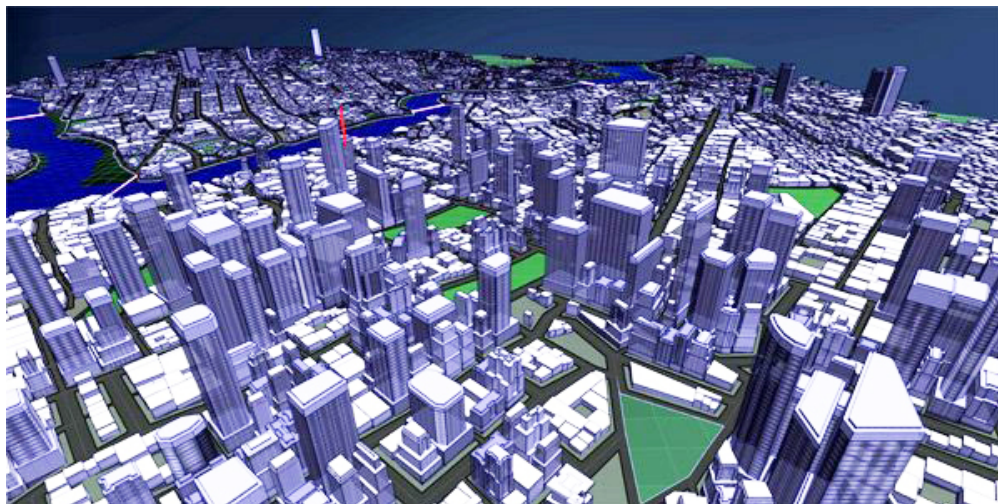


Tema 3

- **Responsabili:** Alex Gradinaru, Alex Dinu
- **Perioada de lucru:** 8/12/2018 – 13/01/2019
- **Termen de predare:** 13/01/2019, 23:55

New Vegas



Scopul acestei teme este de a stimula creativitatea prin generarea procedurală a unui oraș în 3D. Va veți folosi cunoștințele din domeniul graficii asistate de calculator, anume **generare de modele 3D, transformări geometrice, iluminare și texturare**.

Tema își propune un minim de complexitate, descris în cerințele ce urmează, însă va încuraja să veniți cu idei și abordări proprii, astfel încât să vă exercitați capabilitățile programatice și creativitatea, prin generare procedurală.

Utilitatea deprinderilor dobândite în urma creării acestui mediu virtual se regăsește în industrii precum: jocuri video (mai ales open world, e.g. No Man's Sky), filme (Hollywood CGI), retail (e.g. geometric patterns).

Generare procedurală

Generarea procedurală este procesul prin care entități ale unei clase sunt generate prin instanțierea repetată de obiecte (indivizi) de acea clasă, fiecărui individ atribuindu-se valori pseudoaleatoare, ghidate de un set de reguli.

Deși această definiție pare abstractă, avantajul este că nu sunteți constrânși în vreun fel de a genera clădirile și străzile, astfel că puteți utiliza aparate matematice la alegere pentru generarea geometriei și astfel puteți fi foarte creativi.

Orașul. Clădiri și străzi

Scopul generării procedurale în această temă este de a crea un oraș virtual. Elementele din acesta sunt:

- Clădiri
- Străzi
- Elemente decorative (e.g. elemente de trafic, iluminare stradală, **etc.**)

Clădirile și străzile vor fi generate **exclusiv procedural**.

Elementele decorative nu e nevoie să fie generate procedural, pot fi meshe importate de voi, însă plasarea

Ior in scena trebuie sa fie pur procedurala. Alegerea elementelor decorative este lasata la latitudinea voastra. Trebuie sa aveti **minim trei tipuri de elemente decorative**.

Nu sunteti constransi in a nu adauga elemente decorative pe cladiri, insa ferestrele cladirilor nu sunt elemente decorative.

Cerinte

Se va genera procedural o scena cu elementele mentionate mai sus.

Utilizatorul se poate plimba prin intreaga scena utilizand camera din laborator.

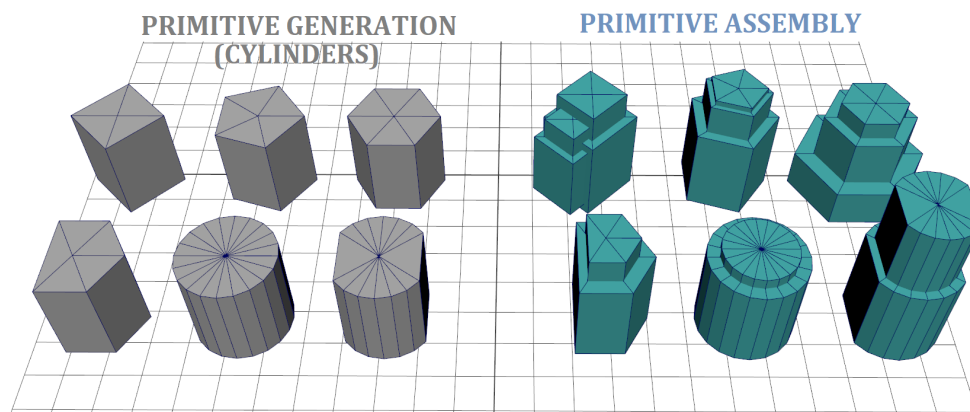
La fiecare noua rulare a aplicatiei, orasul va arata diferit.

Clădrile

Cladirile sunt formate prin asamblare de **primitive** cu geometrie alterata procedural. Asamblarea este controlata printr-un proces **pseudorandom** creat de voi. Primitivele pot fi, de exemplu, cilindri cu distanta nu neaparat uniforma intre oricare 2 puncte succesive de pe conturul cercului generator, cu numar variabil de puncte, etc. Pot fi paralelipipede, etc.

Pentru fiecare triunghi generat, trebuie sa generati si **coordonatele de texturare** corespunzatoare, astfel incat puteti aplica o textura repetitiva (tileable, folositi **GL_REPEAT**) cu ferestre. Puteti gandi triunghiurile constituinte ale primitivelor cate 2, formand un quad, caruia sa-i dati coordonate de texturare in afara intervalului $(0,1) \times (0,1)$, astfel incat sa folositi **GL_REPEAT** corespunzator.

Asamblarea primitivelor constituinte unei cladiri consta in aplicarea de **transformari (scalare, translatie, rotatie)** acestora dupa un **set de reguli(constrangeri)** creat de voi astfel incat rezultatul arata ca o potentiala cladire.



Străzile

Algoritmul de generare a strazilor este lasat la latitudinea voastra, cu precizarea ca intre 2 rulari succesive ale aplicatiei, reseaua de drumuri arata **diferit** ca topologie si ca distanta intre strazi. Reteaua de drumuri trebuie sa fie **conexa**, adica exista un drum intre oricare 2 intersectii.

Strazile trebuie sa fie si ele **texturate** utilizand **GL_REPEAT**.

Geometriile cladirilor nu se vor suprapune peste geometriile strazilor.

Minimul de complexitate implica doar strazi pe verticala si pe orizontala. Cladirile vor fi plasate in celulele formate intre strazi. Minimul de complexitate implica o cladire per celula.

Elementele decorative

Orasul va contine elemente decorative plasate printr-un algoritm procedural, dupa un **set de reguli** ales de voi, astfel ca nu se intersecteaza cu alte elemente, iar pozitia, rotatia si dimensiunea acestora le incadreaza corespunzator in scena.

Elementele pot fi meshe alese de voi. O resursa buna on-line este TurboSquid [<https://www.turbosquid.com/>]
Trebuie sa aveti **minim 3** tipuri de astfel de elemente.

Luminile

Veti avea o **lumina directionala** (L constant pentru toate fragmentele iluminate) care va ilumina intreg orasul.

Fiecare cladire va avea un set de lumini de tip **spot** care o ilumineaza doar pe aceasta, cu directia aproximativ perpendiculara pe ground plane. Si acestea vor fi generate procedural.

Exemplu de Implementare

Sunteti incurajati sa va ghidati dupa implementarea de la acest link:

<https://youtu.be/-d2-PtK4F6Y> [<https://youtu.be/-d2-PtK4F6Y>]

Puteti folosi gramatici independente de context pentru construirea algoritmului de generare procedurala.

Resurse utile

- <http://pcg.wikidot.com/category-pcg-algorithms> [<http://pcg.wikidot.com/category-pcg-algorithms>]
- <https://medium.freecodecamp.org/how-to-make-your-own-procedural-dungeon-map-generator-using-the-random-walk-algorithm-e0085c8aa9a> [<https://medium.freecodecamp.org/how-to-make-your-own-procedural-dungeon-map-generator-using-the-random-walk-algorithm-e0085c8aa9a>]
- <https://stackoverflow.com/questions/155069/how-does-one-get-started-with-procedural-generation> [<https://stackoverflow.com/questions/155069/how-does-one-get-started-with-procedural-generation>]

Bonus

- Orice aduce un plus de realism scenei.
- Orice imbunatateste atmosfera scenei si/sau este captivant vizual.
- Animatii, dinamism.
- Trafic dinamic.

Barem

- **20p** - Generare procedurala a geometriei cladirilor
- **20p** - Generare procedurala a retelei de drumuri
- **20p** - Texturarea cladirilor
- **20p** - Texturarea strazilor
- **15p** - Iluminare scena
- **5p** - Readme&Comentarii

Arhivarea proiectului

- in mod normal arhiva trebuie sa contina toate resursele necesare compilarii si rularii
- inainte de a face arhiva asigurati-va ca ati dat clean la proiect
 - click dreapta pe proiect in **Solution Explorer** → **Clean Solution**, sau
 - stergeti folderul **/Visual Studio/obj**

- stergeti fisierul **/Visual Studio/Framework EGC.sdf** (in caz ca exista)
- stergeti fisierul **/Visual Studio/Framework EGC.VC.db** (in caz ca exista)
- stergeti folderul **/x64** sau **/x86** (in caz ca exista)
 - executabilul final este generat in folderul **/x86** sau **/x64** la finalul link-editarii in functie de arhitectura aleasa la compilare (32/64 biti)
- in cazul in care arhiva tot depaseste limita de 20MB (nu ar trebui), puteti sa stergeti si folderul **/libs** sau **/Resources** intrucat se pot adauga la testare. Nu este recomandat sa faceti acest lucru intrucat ingreuneaza mult testarea in cazul in care versiunea curenta a librariilor/resurselor difera de versiunea utilizata la momentul scrierii temei.

Observatii

Alte aspecte ce nu au fost prezentate raman la latitudinea voastra.

- Se poate utiliza framework-ul de laborator sau o implementare proprie.
- Documentatia pentru evenimentele de mouse se gaseste in fisierul `/Core/Window/InputController.h` [<https://github.com/UPB-Graphics/Framework-EGC/blob/master/Source/Core/Window/InputController.h>]

Punctajul este oferit doar pentru cerintele integrate in aceeasi scena.

Punctajul bonus se ofera doar daca cerintele obligatorii sunt implementate corect.

egc/teme/2018/03.txt · Last modified: 2018/12/10 20:38 by ovidiu.dinu