Tema 1 - Fiordurile Norvegiene

- **Responsabili:** Cristi Lambru, Anca Bălușoiu, Alex Grădinaru, Alex Dinu
- **Lansare:** 25 noiembrie
- **Termen de predare:** 15 decembrie 2020, ora 23:55
- **Regulament:** https://ocw.cs.pub.ro/courses/spg/regulament_general [https://ocw.cs.pub.ro/courses/spg/regulament_general]
- **Notă: Orice informație ce nu a fost acoperită în acest document este la latitudinea voastră!**

În cadrul acestei teme trebuie să realizați o simulare a fiordurilor din Norvegia, prin care se plimbă o bărcuță. Un posibil rezultat al implementării îl puteți observa în filmulețul de mai jos:

Mediul Înconjurător

În cadrul implementării trebuie să aveți un râu principal, care la un moment dat se sparge în minim trei brațe. Între râurile nou formate trebuie să generați munți.

- Generarea Apei

Atât râul principal, cât și brațele sale se vor genera folosind suprafețe de translație bazate pe curbe Bezier. Punctele de control pe care le veți folosi pentru a construi curba Bezier vor fi generate aleator. Curba rezultată se află la jumătatea lățimii râului.

Pentru a putea uni râul principal cu cele trei brațe ale sale veți considera că ultimul punct de control folosit pentru generarea râului principal este în același timp și primul punct de control al celor trei brațe, urmând ca celelalte puncte de control ale fiecărui braț să fie generate aleator.

- Reflexia Apei

Din moment ce suprafețele de translație generate reprezintă apa unui râu va fi necesar să aplicați o textură sugestivă, care să imite aspectului apei (puteți folosi ce textură doriți). Atenție cum alegeți coordonatele de textură astfel încât să se mapeze textura corespunzător pe suprafață.

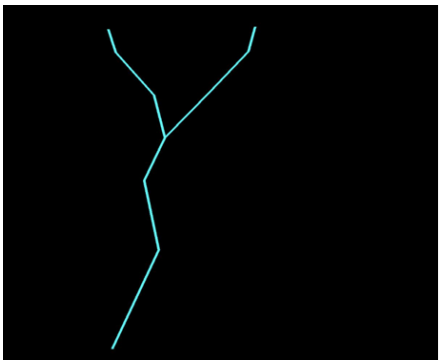
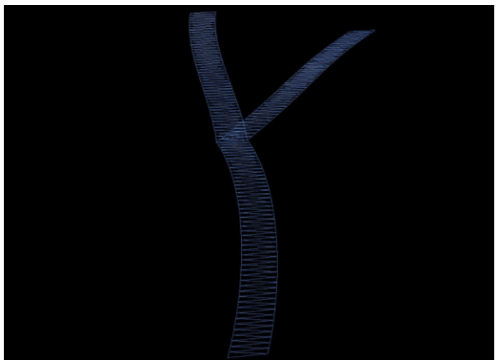
Mai mult, știm că în realitate mediul înconjurător se reflectă pe suprafața apei. Acest lucru va trebuit implementat și în cadrul acestei teme. Pentru simplificare însă, nu o să fiți nevoiți să mapați tot mediul înconjurător, ci doar cerul.

Ca să aplicați atât textura apei, cât să reușiți să mapați și reflexia mediului înconjurător pe suprafețele generate, puteți folosi funcția mix (https://www.khronos.org/registry/OpenGL-Refpages/gl4/html/mix.xhtml [https://www.khronos.org/registry/OpenGL-Refpages/gl4/html/mix.xhtml]).

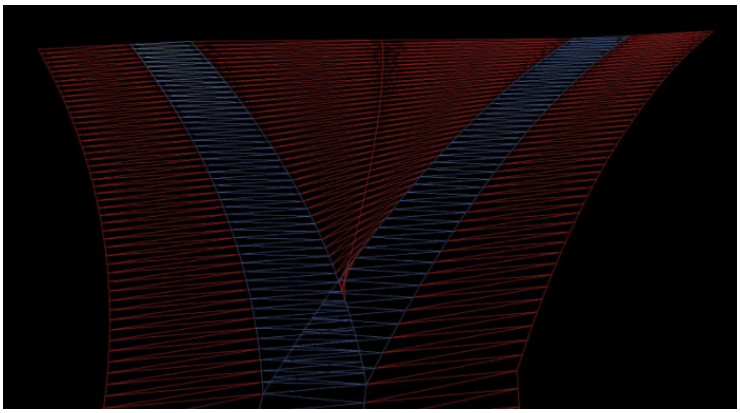
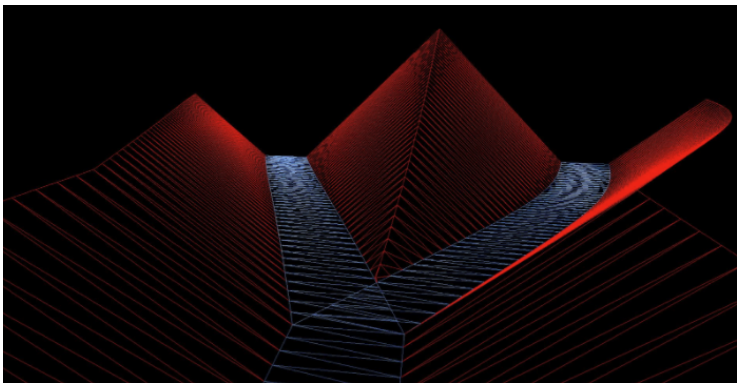
- Generarea Munților

Între oricare două brațe ale râurilor va trebui să generați câte un munte. De asemenea, și pe partea exterioară a râurilor vor trebui generați munți. Munții, la rândul lor, se vor genera folosind suprafețe de translație.

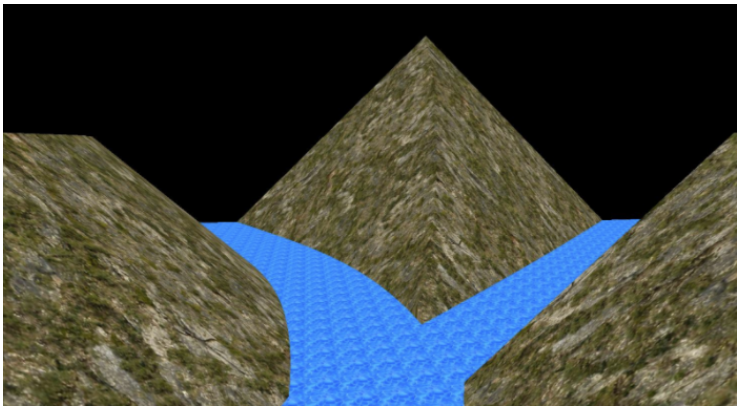
Ca să generați un munte aflat între două râuri, este mai întâi nevoie să vizualizați cum se generează aceste râuri. În imaginile de mai jos puteți vedea procesul pas cu pas. Mai întâi, puteți vedea curbele pe baza cărora veți construi râurile. Pe urmă puteți observa suprafețele de translație generate. A se observa faptul că în punctul în care se unesc râurile există o zonă comună (în a treia poză se poate vedea cu ușurință intersecția râurilor).


Ca să generați muntele, va trebui să generați două suprafețe. Fiecare dintre acestea va merge de-a lungul marginii unui râu. Pentru a calcula punctul de start, va trebui să calculați coordonatele intersecției marginilor celor două râuri (a se observa în imaginile de mai jos) (Hint: https://stackoverflow.com/questions/563198/how-do-you-detect-where-two-line-segments-intersect/565282#565282 [https://stackoverflow.com/questions/563198/how-do-you-detect-where-two-line-segments-intersect/565282#565282]). Restul punctelor vor fi calculate folosind media aritmetică dintre punctele folosite pentru generarea râurilor. Pentru a da diferite înălțimi munților, puteți varia coordonata y.

Nu uitați să aplicați textura pe suprafața astfel generată!



Munții exteriori se vor genera într-un mod similar. Nu va mai fi însă nevoie să determinați intersecția între două suprafețe. Puteți să realizați o copie a marginii râului.

- Cerul

În implementare va trebui să randați și cerul. Acest lucru se poate implementa folosind un cubemap.

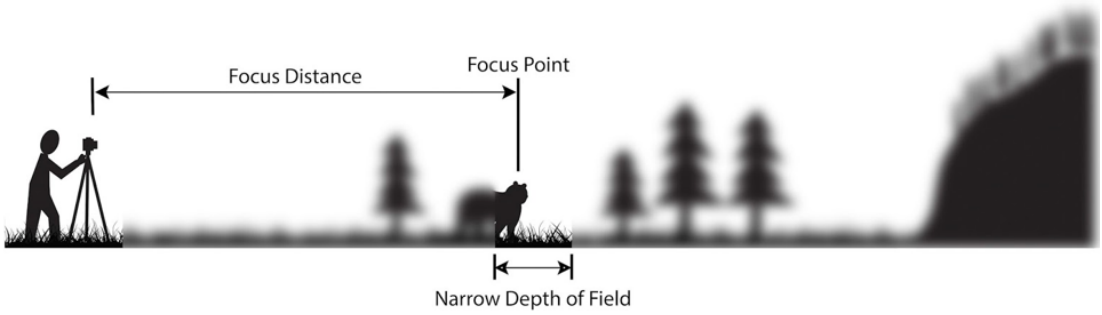
Bărcuța de pe Apă

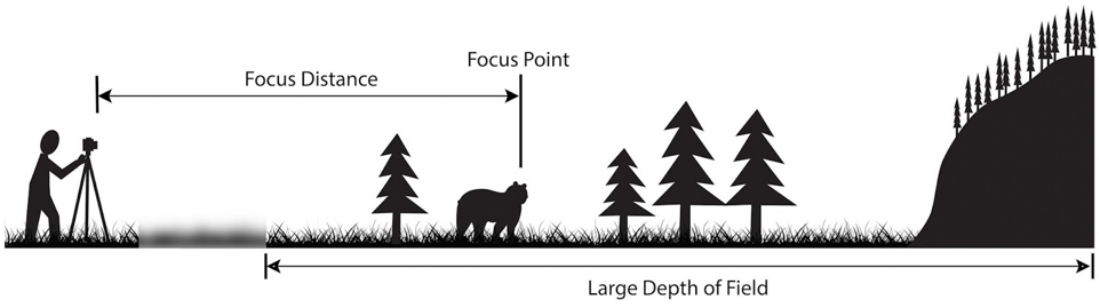
Pe râu se va plimba o bărcuță. Aceasta se va plimba aleator atât pe râul principal, cât și pe brațele acestuia la mijlocul latimii raului. Deplasarea se va face pe curbele Bezier utilizate pentru generarea râurilor. În mișcarea ei, bărcuța, evident, va fi rotită corespunzător (Hint: https://stackoverflow.com/questions/53365389/opengl-rotate-object-on-y-axis-to-look-at-another-object [https://stackoverflow.com/questions/53365389/opengl-rotate-object-on-y-axis-to-look-at-another-object]).

Elicea motorului bărcuței va arunca în urma sa apa. Pentru a simula acest lucru, puteți folosi un sistem de particule. La resetarea sistemului, aveți grijă să setați poziția și direcția particulelor într-un mod relativ aleator.

Efect de Post-Procesare: Profunzimea câmpului vizual (depth of field)

Profunzimea câmpului vizual reprezintă zona de la o anumită distanță față de o cameră de fotografiat în care aceasta are acuratețea maximă. În afara acestei zone obiectele devin neclare. O reprezentare vizuală se poate vedea mai jos:



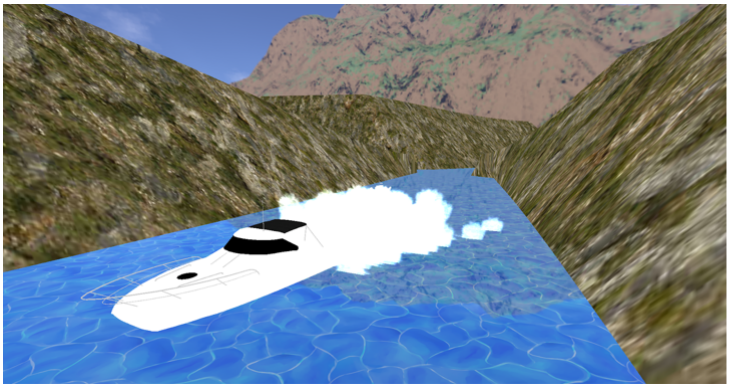
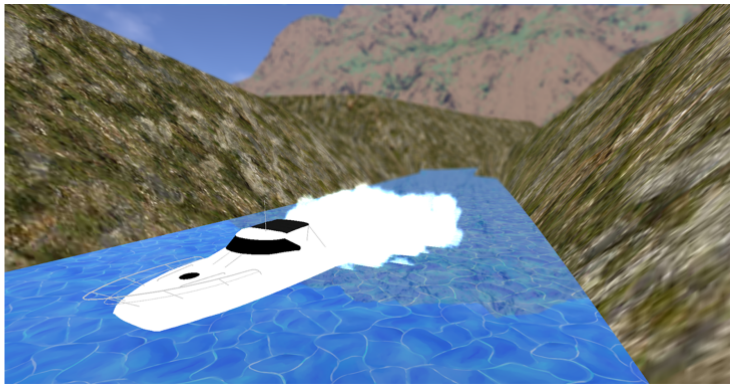


- Desenarea în alt framebuffer

Toate obiectele din scenă se desenează inițial într-un framebuffer nou, diferit de cel default al ecrannului. Pentru a desena în final pe ecran se folosește un al doilea pas de desenare în care se rasterizează un quad pe tot ecranul. În acest fel se execută o instanță de fragment shader pentru fiecare pixel al ecranului. Acest al doilea pas primește texturile de culoare și de adâncime ale framebuffer-ului în care s-au desenat inițial obiectele scenei.

- Simularea efectului de profunzime al câmpului vizual

Pentru a simula acest efect se setează o distanță focală și dimensiunea zonei în care obiectele sunt preluate clar (large - narrow "depth of field" în imaginea de mai sus). Pentru fiecare pixel de pe ecran se extrage culoarea și adâncimea fragmentelor din framebuffer-ul în care au fost desenate obiectele. Pentru a putea fi folosită, adâncimea din depth buffer trebuie liniarizată (Hint: https://stackoverflow.com/a/6657284 [https://stackoverflow.com/a/6657284]). Toți pixelii sunt verificați dacă se află în câmpul de adâncime și pentru cei care se află în afară se aplică un filtru de netezire. Pentru a nu se face trecerea foarte rapid între clar și blurat se folosește o zonă de trecere în care se interpolează între varianta clară și cea blurată pe baza distanței.

Notare (150p)

- Mediul înconjurător 75p
  - Râul cu un braț principal și cel puțin 3 brațe secundare generate în shader 30p
  - Munții care delimitează râurile 40p
  - Cerul 5p
- Reflexie apă 15p
- Bărcuța 30p
  - Deplasare 15p
  - Sistem de particule 15p
- Depth of field 30p

Bonusuri Posibile

- Implementarea unui skydome în locul skybox-ului
- Implementarea ploii
- Valuri realiste
- Orice tehnică de a realiza un model de reflexie realistă pe apă (de exemplu Space Screen Reflections)
- Orice aduce realism și/sau dinamism scenei

Indicații Suplimentare

Tema va fi implementată în OpenGL și C++. Este indicat să folosiți framework-ul și Visual Studio.

Pentru implementarea temei, în folderul Source/Laboratoare/ puteți crea un nou folder, de exemplu Tema1, cu fișierele Tema1.cpp și Tema1.h (pentru implementare POO, este indicat să aveți și alte fișiere). Pentru a vedea fișierele nou create în Visual Studio în Solution Explorer, apăsați click dreapta pe filtrul Laboratoare și selectați Add→New Filter. După ce creați un nou filtru, de exemplu Tema1, dați click dreapta și selectați Add→Existing Item. Astfel adăugați toate fișierele din folderul nou creat. În fișierul LabList.h trebuie adăugată și calea către header-ul temei. De exemplu: #include <Laboratoare/Tema1/Tema1.h>

Arhivarea proiectului

- În mod normal arhiva trebuie să conțină toate resursele necesare compilării și rulării
- Înainte de a face arhiva asigurați-vă ca ați dat clean la proiect
  - Click dreapta pe proiect în **Solution Explorer** → **Clean Solution**, sau
  - Ștergeti folderul **/Visual Studio/obj**
- Ștergeți fișierul **/Visual Studio/Framework_SPG.sdf** (în caz că există)
- Ștergeți fișierul **/Visual Studio/Framework_SPG.VC.db** (în caz că există)
- Ștergeți folderul **/.vs** (in caz ca exista)
- Ștergeți folderul **/x64** sau **/x86** (in caz ca exista)
  - Executabilul final este generat în folderul **/x86** sau **/x64** la finalul link-editării în funcție de arhitectura aleasă la compilare (32/64 biti)
- În cazul în care arhiva tot depășește limita de 20MB (nu ar trebui), puteți să ștergeți și folderul **/libs** sau **/Resources** întrucât se pot adăuga la testare. Nu este recomandat să faceți acest lucru întrucât îngreunează mult testarea în cazul în care versiunea curentă a bibliotecilor/resurselor diferă de versiunea utilizată la momentul scrierii temei.

spg/teme/2020/01.txt · Last modified: 2020/11/26 15:22 by andrei.lambru