

Examples of semestral project topics for the course **Computer Graphics for Game Development (NPGR033)**

Summer semester 2017/18

Jaroslav Křivánek, Jakub Gemrot, and Martin Kahoun

20/02/2018

General information

To get a credit for NPGR033 labs you are required to pick, implement and defend a semestral project. This document list examples of topic you can choose from. You are also free to come with your own topic; if you have some pet graphic / game project you are working on, feel free to work on that as your semestral project.

One semestral project should correspond to 60-80 hours of work (roughly). You can also work on single semestral project in a group of two. In this case, you have to clearly decide on responsibilities / divide the project in parts so it is clear that both of you contributed to the project equally.

Scoring sheet

Roman Borufka - Self-building structures	[report 1]	[report 2]	[report 3]
Bohus Brecka - Filtering of normal distribution	[report 1]	[-10 pt]	[-10 pt]
Julius Flimmel - Music to 3D	[report 1]	[report 2]	[report 3]
Pavel Halbich - Simplified voxel terrain rendering in UE for game Tau Ceti F 4	[-10 pt]	[-10 pt]	[-10 pt]
Filip Hauptfleisch - Pencil drawing shader for unity	[report 1]	[report 2]	[report 3]
Ondřej Lakomý - Rendering Countless Blades of Waving Grass	[report 1]	[report 2]	[report 3]
Lucia Magurová - Water movement on music	[-10 pt]	[-10 pt]	[-10 pt]
Marek Polák - UE shaders for simulating a specific artistic style	[report 1]	[report 2]	[report 3]
Luis Sanchez - Checkerboard rendering in UE4	[-10 pt]	[-10 pt]	[-10 pt]
Merve Tuncel - Stylized buildings in Unity	[report 1]	[report 2]	[report 3]
Marcel Hruška - Water rendering and object interaction			

Papers:

- Roman Borufka - [Illustrative Rendering in Team Fortress 2](#)
 - Bohuslav Brecka - [Microfacet-based Normal Mapping](#)
 - Julius Flimmel - [Assisted Texture Assignment](#)
 - Pavel Halbich - [Authoring Landscapes by Combining Ecosystem and Terrain Erosion Simulation](#)
 - Filip Hauptfleisch - [Animating 3D Vegetation in Real-time Using a 2D Approach](#)
 - Ondřej Lakomý - [Real-time cloth rendering with fiber-level detail](#)
 - Lucia Magurová - N/A
 - Marek Polák - [Real-time Pencil Rendering](#)
 - Luis Sanchez - [Time-varying Weathering in Texture Space](#)
 - Merve Tuncel - [Depixelizing Pixel Art](#)
-

Implementing & playing around with a graphics paper / a GPU gem

Find a paper / algorithm, whose result intrigues you. Reimplement it “from scratch”.

- NVidia GPU Gems:
<https://developer.nvidia.com/gpugems>
- List of more GPU books:
<http://www.realtimerendering.com/resources/shaderx/>
Note that many of those books are present in our faculty library; if you find some that is not, request us to buy it!
- Graphics papers online:
<http://kesen.realtimerendering.com/>
of interest especially:
 - I3D (Symposium on Interactive 3D Graphics and Games)
 - Specifically on what the conference title says
 - HPG (High Performance Graphics)
 - Much of it is GPGPU, but there are some good real-time rendering and game papers
 - EGSR
 - General conference on rendering, mostly offline, but also real-time
 - SIGGRAPH, SIGGRAPH Asia
 - Best in graphics and interactive techniques. One or two papers a year can be relevant.
- Real-time rendering resources:
<http://www.realtimerendering.com/>

Specific topic suggestions, Jarda

Real-time area light lighting, Option 1: Implement the “linearly transformed cosines” techniques into Unreal engine / may be Unity

- Linearly transformed cosines [[demo](#)]:
<https://labs.unity.com/article/real-time-line-and-disk-light-shading-linearly-transformed-cosines>

Real-time area light lighting, Option 2: Compare “linearly transformed cosines” and “analytic approximations”

- Linearly transformed cosines [[demo](#)]:
<https://labs.unity.com/article/real-time-line-and-disk-light-shading-linearly-transformed-cosines>
- Analytic approximations:
 - Most recent paper:
http://pascal.lecocq.home.free.fr/publications/lecocq_TVCG2017_analyticAreaLightShading.pdf
 - First author's page with some related work: <http://pascal.lecocq.home.free.fr/>
- Real-time particle systems:
https://docs.google.com/presentation/d/1w7PPmnDw26-zPe_5ePiX8SOQPM9SqjQyzUZW8zVAg8k/edit?ts=5a9147aa#slide=id.p
- Music to 3D:
https://docs.google.com/presentation/d/1bBA7ETzTzdMiTS6SqQe_nAXPgpBUUG1BS8CUicFPE7k/edit#slide=id.p
- SSAO, https://www.amazon.com/dp/0998822892?tag=realtimerenderin&pldnSite=1#reader_0998822892
- Real-time Bokeh effect,
https://www.amazon.com/dp/0998822892?tag=realtimerenderin&pldnSite=1#reader_0998822892

Specific topic suggestions, Martin

1. Shadow mapy - implementace kaskadovych shadow map:
 - Percentage Closer Filtering (PCF) - zakladni s pomoci SampleCompare ci vlastni pomoci Gather eventuelne Poisson samplovani pro jemne okraje stinu.
 - Jemne prechody mezi kaskadama.
 - Receiver plane depth bias.
 - Sample Distribution Shadow Maps (SDSM) - automaticke pocitani min/max hloubky sceny pomoci compute shaderu a nastaveni optimalniho near/far.
 - Variance Shadow Maps
 - Zdroje:
 - i. <https://mynameismjp.wordpress.com/2013/09/10/shadow-maps/>
 - ii. [https://msdn.microsoft.com/en-us/library/ee416307\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/ee416307(v=vs.85).aspx)
2. Vodni shader
 - Implementace vlneni a displacementu meshe vodni hladiny (jednoduche sinusoidy nebo Tessendorf pocitani v compute shaderu).
 - Pouziti detailni normalove mapy pro zlepšení hrubych normal povrchu displacement mapy.
 - Refrakce dna perturbovana normalami.
 - Volumetricky efekt vodniho sloupce včetne inscatterovaneho svetla
 - Odraz oblohy a okoli, budto: planar reflections, Screen Space Reflections (SSR), samlovani cube mapy s environmentem, v zakladu staci constantni barva oblohy
3. Particle rendering
 - Implementace GPU particles - update v compute shaderu
 - Rendering billboardu
 - Soft particles (http://developer.download.nvidia.com/whitepapers/2007/SDK10/SoftParticles_hi.pdf)
4. Renderovani vegetace
 - Muzou byt bud cele modely nebo jen billboardy (<https://hal.inria.fr/hal-00650120/file/article.pdf>)
 - Nemusi byt ale tak brutalni jako v tom clanku nahore, spousta tech efektu jde dobre hacknout:

- i. Opposition efekt
 - ii. Silverlining
 - iii. Normalova mapa billboardu stromu, aby vypadal jako 3D
 - Hashed AToC (http://cwyman.org/papers/tvcg17_hashedAlphaExtended.pdf), aby stromy se vzdalenosti nepruhlednely.
 - Stiny na vegetaci (ale uz trosku predpoklada pritomnost kodu pro stinove mapy), krasna uloha je vrhani stinu billboardem na billboard bez toho, aniz by ten stin prosekaval ten billboard. Inspirace lze vzit tady: https://developer.nvidia.com/gpugems/GPUGems3/gpugems3_ch04.html
-