Assessment - Visualization and Illumination - 2018/19

The assessment for this course aims at getting students to increase research skills, exploring particular topics. It is based on two components:

- A practical assignment (groups up to 3 students)
- Writing an essay in article format (individual).

Each component is equally weighted in the final grade, with a minimum grade of 10 in each component. The theme selected for one component must not overlap the topic for the second component.

Practical Assignment

The practical assignment deliveries are:

- An implementation of an algorithm
- A written report on the topic, focused on the selected algorithm, justifying the group's options and detailing the algorithm, ideally comparing it to other alternatives.
- A public presentation of the work (10 mins + 5 mins for questions)

The main goal is not the implementation per se, but rather its analysis. Students may use code found in the Internet, but must be able to detail the pros and cons of the selected solution, and justify its implementation, both from a software engineering and computer graphics perspective. In this context both the written report and the presentation have a very relevant role.

Note that when referring to "code found in the Internet" it is not to be understood that the implementation can be just a copy of some work on github. The students are supposed to develop the core of the algorithm but may use snippets and routines from Internet sources.

Some possible themes are:

- Screen space ambient occlusion
- Terrain generation
- GPU Procedural Material Generation
- Non-Photorealistic Rendering
- Occlusion Queries
- Drawing Geometry with OpenGL

Students may present alternative topics, checking with the lecturer the suitability of their proposal prior to starting their work.

Essay

The essay's main goal is to provide an introduction to research in CG. The essay should have between 8 and 10 pages following ACM template for articles (http://tog.acm.org/authors.html).

The essays should focus on the state of the art for a particular topic relevant to real-time CG. Some topic suggestions are:

- Hybrid rendering (rasterization + ray tracing)
- Rendering 3D roads
- Water shading or geometric simulation
- Buoyancy
- L-Systems
- Particle Systems
- Boids or Crowd behaviour
- Clouds
- Lightning

As in the practical assignment, students may present alternative topics, checking with the lecturer the suitability of their proposal prior to starting their work.

Support materials:

SSAO:

- http://www.gamerendering.com/category/lighting/ssao-lighting/
- http://www.gamedev.net/page/resources/_/technical/graphics-programming-and-theory/a-simple-and-practical-approach-to-ssao-r2753
- http://www.john-chapman.net/content.php?id=8

Terrain Generation:

- http://vterrain.org/Elevation/Artificial/
- Algorithms and Approaches for Procedural Terrain Generation - A Brief Review of Current Techniques (LINK só a partir da UM)
- A Survey of Procedural Terrain Generation Techniques using Evolutionary Algorithms (LINK)

Buoyancy:

- One Thousand Ships with Real-time Buoyancy and NVIDIA WaveWorks (LINK)
- Water interaction model for boats in video games (LINK)

Clouds:

- Physically Based Sky, Atmosphere and Cloud Rendering in Frostbite (LINK)
- Convincing Cloud Rendering (LINK)

Occlusion Queries:

- https://vertostudio.com/gamedev/?p=177
- http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter06.html

Drawing Geometry with OpenGL

- http://www.openglsuperbible.com/2013/10/16/the-road-to-one-million-draws/
- Opengl specification

Procedural Textures/Materials

- http://www.upvector.com/?section=Tutorials&subsection=Intro%20to%20Procedural%20Textures
- http://liu.diva-portal.org/smash/get/diva2:618262/FULLTEXT02.pdf
- http://www.davidcornette.com/glsl/gallery.html

Non-Photorealistic Render

- http://www.red3d.com/cwr/npr/
- http://www.mrl.nyu.edu/publications/npr-course1999/

Boids

- http://www.red3d.com/cwr/boids/

L-Systems

- http://algorithmicbotany.org/papers/abop/abop-ch1.pdf

Procedural City Generation

- http://www.citygen.net/
- https://graphics.ethz.ch/Downloads/Publications/Papers/2001/p_Par01.pdf

Particle Systems

- https://www.khanacademy.org/computing/computer-programming/programming-natural-simulations/programming-particle-systems/a/intro-to-particle-systems

OpenGL ES

- https://www.khronos.org/opengles/

General

- OpenGL Superbible
- OpenGL Programming Guide
- Real Time Rendering Resources (http://www.realtimerendering.com/)
- Slides Siggraph 2012 (http://blog.selfshadow.com/2012/08/11/siggraph-2012-links/)
- Slides Siggraph 2013 (http://blog.selfshadow.com/2013/07/24/siggraph-2013-links/)
- Slides Siggraph 2014 (http://blog.selfshadow.com/2014/08/14/siggraph-2014-links/)
- Slides Siggraph 2015 (http://blog.selfshadow.com/2015/08/15/siggraph-2015-links/)

- Slides Siggraph 2016 (http://blog.selfshadow.com/2016/07/31/siggraph-2016-links/)
- Slides Siggraph 2017 (https://blog.selfshadow.com/2017/08/01/siggraph-2017-links/)
- Slides Siggraph 2018 (https://blog.selfshadow.com/2018/08/16/siggraph-2018-links/)