

Hardware for computer graphics (NPGR019)

© 2003-2017 Josef Pelikán
CGG MFF UK Praha

pepca@cgg.mff.cuni.cz

<http://cgg.mff.cuni.cz/~pepca/>



Content and form

- ◆ **hardware support** for 3D graphics (GPUs)
 - ◆ follows loosely the CG I (NPGR003) course
 - ◆ math and data structures for 3D graphics
 - ◆ GPU capabilities, advanced techniques, data handling, ..
 - ◆ 3D scene rendering methods, advanced techniques
 - ◆ GPU programming (shaders), general computing GPGPU
 - ◆ concrete API (OpenGL, GLSL, CUDA, OpenCL ..)
- ◆ **2/1 C+Ex**
 - ◆ 90 min lecture every week, 90 min lab every other week
 - ◆ practical examples, homework tasks for the credit

Course brief I



◆ **introduction, history, architecture**

- ◆ graphic accelerator history, contemporary GPU arch.
- ◆ 3D graphics pipeline
- ◆ FFP and shader architecture (fixed & programmable pipeline)

◆ **mathematics for real-time graphics**

- ◆ homogeneous transforms, projections, coordinate syst.
- ◆ interpolation and approximation, perspective-correct
- ◆ quaternions and interpolation
- ◆ spline functions



Course brief II

◆ data for real-time 3D graphics

- ◆ boundary scene representations
- ◆ LoD (Level of Detail)
- ◆ hierarchical representations, billboards, imposters

◆ basic GPU techniques

- ◆ visibility, transparency
- ◆ texture mapping
- ◆ fog, ..



Course brief III

◆ **basic GPU programming**

- ◆ data handling, vertex-buffers, index-buffers
- ◆ textures and their combination, texture formats & compression
- ◆ double-buffering, triple-buffering

◆ **advances GPU programming**

- ◆ multi-pass methods, stencil-buffer, accumulation
- ◆ bump-mapping and environment-mapping
- ◆ shadow casting, lighting methods, NPR, ..

Course brief IV



◆ **programmable pipeline - shaders**

- ◆ basic PP architecture, vertex-, fragment-, geometry- and tessellation- shaders
- ◆ user data at vertices and in textures
- ◆ shader languages (GLSL, [Cg=HLSL])

◆ **concrete APIs and tools**

- ◆ OpenGL, [DirectX]
- ◆ practice (labs): OpenGL, OpenTK (C#), SDL (C[++]
- ◆ GPGPU tools: CUDA, OpenCL, [compute shaders in GLSL]



Literature I

- ◆ Tomas Akenine-Möller, Eric Haines: ***Real-time rendering, 3rd edition***, A K Peters, 2008, ISBN: 9781568814247
- ◆ OpenGL Architecture Review Board: ***OpenGL Programming Guide: The Official Guide to Learning OpenGL***, Addison-Wesley, latest edition (8th edition for the OpenGL 4.1)
- ◆ [Randima Fernando, Mark J. Kilgard: ***The Cg Tutorial***, Addison-Wesley, 2003, ISBN: 0321194969]



Literature II

- ◆ Ron Fosner: ***Real-Time Shader Programming***, Morgan Kaufmann, 2002, ISBN: 1558608532
- ◆ ed. Randima Fernando: ***GPU Gems***, Addison-Wesley, 2004, ISBN: 0321228324
- ◆ ed. Matt Pharr: ***GPU Gems 2***, Addison-Wesley, 2005, ISBN: 0321335597
- ◆ ed. Hubert Nguyen: ***GPU Gems 3***, Addison-Wesley, 2007, ISBN: 0321515269
- ◆ Randi J. Rost: ***OpenGL(R) Shading Language***, 3rd edition, Addison-Wesley, 2009, ISBN: 0321637631



Online sources

- ◆ **NVIDIA:** *<http://developer.nvidia.com/>*
- ◆ **ATI/AMD:** *<http://developer.amd.com/>*
- ◆ **OpenGL, OpenCL:** *<http://www.opengl.org/>*
- ◆ **DirectX:** *<http://msdn.microsoft.com/directx/>*
- ◆ **game development:** *<http://www.gamedev.net/>*
<http://www.gamasutra.com/>
- ◆ **Dave Eberly:** *<http://www.geometrictools.com/>*
- ◆ ...