“Throughout the two and a half years Asger worked for us, he consistently came across as a highly- competent developer, as well as an amicable person. We threw a wealth of different projects at him, based on an equal amount of different technologies and architectures, and never saw him waiver or hesitate - neither at the thought of learning new subject matter, languages, and frameworks nor at having to grasp the structure of unknown (sometimes vast) amounts of existing source code.”

—Tobias Hinnerup
Owner
Hinnerup Net

“Asger is awesome!”

—Lars Kroll
Technical Director
Unity Studios

Asger Vejen Hoedt
+45 6051 3669 - asgerhoedt@gmail.com
DK-2635 Ishøj
www.asgerhoedt.dk

Summary

Computer Graphics Software Developer. Accomplishments include:

- designed the MobiGlobe architecture for 48 different visualizations and led a team of 4 developers implementing a flexible visualization datamodel for the installations in Autostadt, as well as the subsequent creation of three frontends with touch and gestures as input.
  The MobiGlobe project won the *RedDot: Grand Prix award for Communication Design* in Berlin, 2013.
- researched and combined monte carlo integration, density estimation, image space noise reduction and material approximations in a renderer used for producing interactive and reliable previews of global illumination in immersive environments for computer aided engineering with scenes starting at 10 million polygons. The resulting backend ran at interactive framerates on a high-end workstation.
- implemented a physically based shading framework for visualizing plausible materials and integrated it into multiple different rendering backends, spanning real-time rendering via rasterization, an in-house developed interactive baked global illumination solution and a progressively refining global illumination renderer.
- structured software projects of 500’000 lines of code or more by focusing on modularity and applying software patterns such as Visitor, Model-View-Controller and Entity-Component to construct reliable and modular architectures.

Career Driver

Learning, utilizing, and enhancing state-of-the-art algorithms in graphics and image processing.

Special Skills

- Shader and GPGPU Languages.
- Linear Algebra, Linear Programming, and Convex Sets
- Global illumination rendering techniques.
- Programming Data Parallel Solutions on the GPU
- The Usual Buzzwords: C/C++, CUDA, Thrust, Scrum, Data-Parallel Programming, SIMT, C#, OpenGL, GLSL, CG, Unity3D, OptiX, OpenCL, GCC, CMake, Emacs, Visual Studio

Employment History

R&D Software Developer - ESI Software Germany GmbH
Stuttgart, Germany, 05/2013 - present
Worked on-site in Stuttgart for the first two years. Homeoffice since 07/2015.

- implemented framework for physically plausible materials and integrated it into existing rendering backends, enabling consistent visualizations of those materials for differing rendering approaches, such as real-time, plausible or offline global illumination renderings.
- researched, conceptualized, and implemented a real-time global illumination approximation used for computer aided engineering in interactive immersive environments.
- achieved a factor two speed-up in the path tracing backend by exploiting knowledge of the underlying GPU architecture and allowing that knowledge to influence the random number generator used for integration.

Lead/GFX/Game Programmer - Unity Studios
Aarhus, Denmark, 05/2011 - 04/2013

- developed interactive 3D applications using the Unity 3D Engine, optimizing visual quality, resource management, and rendering performance.
- optimized a model loading system which needed to on demand stream models with over a million vertices and 6MB textures while seamlessly blending between four levels of details for said models.
- researched, co-designed and implemented - over the course of three months - the full body interaction for a Kinect application running on a 150" screen, which supported simultaneous pointing and clicking, object rotations, and button dragging.

Freelance Developer - Hinnerup Net
Aarhus, Denmark, 11/2008 - 05/2011

- implemented new website features for multiple customers using various web and database technologies, debugging both new and legacy codebases.
- provided accurate time/material cost estimates for proposed solutions.

Student Programmer - CAVI, "Aarhus By Light" Project
Aarhus, Denmark, 02/2008 - 11/2008

- extended camera tracking and physics to improve the complexity of the interaction between onlookers and the virtual creatures on a 180 m2 mediafacade, which resulted in interactions beyond simply pushing the creatures.
- developed a sandbox environment, where features of the facade could be "drawn" into the creatures' environment, allowing the creatures to interact with the facade's window and other prominent features.

Sparetime Projects

Architect and developer - Cogwheel, Open Source 3D Framework
09/2015 - present

- a unified cross-platform framework for my sparetime experiments, combining my previous minor sparetime projects into a single solution, allowing quick prototyping of future projects.
- exploring the use of data-driven programming to achieve a decoupled and efficient codebase design.

Open Source Developer - OpenEngine, Open Source 3D Engine Framework
09/2007 - 04/2013

- co-developed 3D engine framework for use in courses at Aarhus University. The engine is now being used in a professional business environment.
- structured software comprised of 500,000 lines of code by focusing on modularity and applying software patterns such as Visitor, Model-View-Controller and Component to create a reliable and flexible architecture.
- improved speed and visual quality of rendering through continuous development of the resource system; added post process effects and reworked geometry structure to facilitate switch from OpenGL's immediate mode to rendering with vertex buffers.

Ray Tracing Enthusiast - DACRT, Divide and Conquer Ray Tracing
11/2011 - 9/2012
Recent research had illustrated methods to abandon the usual acceleration structures in favor of constructing partial ones for each ray iteration. 'Classical' ray tracing uses splitting methods which classified geometric primitives, inspired by kd-trees and BVHs. However, since an iteration can consist of millions of rays and usually only hundreds-of-thousands of triangles, it seemed promising to focus on classifying rays instead to achieve early-ray-termination and avoid allocating rays to several threads in a multithreaded environment.

- combined DACRT and 5D hyperrays on a CPU to successfully eliminate standard acceleration structures.
- adapted the algorithm for the GPU to speed up the DAC ray tracer, uncovering problems with performing DACRT completely data parallel - such as undeterministic ray or geometry storage size.

Education

- M.Sc., Computer Science - Aarhus University
  Aarhus, Denmark, 2008 - 2011
  Thesis: Efficient Ray Tracing of Dynamic Scenes on the GPU
- B.Sc., Computer Science - Aarhus University
  Aarhus, Denmark, 2005 - 2008

Languages

- Danish - Native
- English - Proficient
- German - Professional working proficiency

Personal Information

- born 4. March 1985
- married
- father of 3 year old son
- hobbies include; playing with my son, reading fantasy books and articles, sparetime projects, building LEGO (not just with Junior) and skiing.