Information on the oral exam from the course Computer Graphics for Game Development (NPGR033)

Summer semester 2016/17

Jaroslav Křivánek, Jakub Gemrot, and Martin Kahoun

24/05/2017

General information

The exam is oral and consists of two questions on the material covered in the lectures and will be selected from the list given below. The exam starts by the student drawing the two questions; the student then has up to 30 minutes for preparation, which is then followed by the oral examination itself.
To succeed at the exam, it is essential that the student shows a true understanding of the material from the lectures (as opposed to just repeating buzzwords and formulas).

Questions

1. **Transformations.** Vectors, matrices, homogeneous coordinates. Translation, scale, shear. Rotation: Euler angles, gimbal lock, quaternions. Coordinate spaces (model, world, view, clip, NDC). Viewing (LookAt) and perspective projection transformations.

2. **Graphics pipeline.** General GPU architecture. Types of graphics primitives. Textures, sampling, and mip-mapping. The processing chain of the graphics pipeline: vertex processing (vertex, tessellation, and geometry shaders), viewport transform, primitive assembly, rasterization, fragment processing.

3. **Shaders.** Specific data types and constructs of shading languages (vectors, matrices, swizzling, etc.). Types of shaders, their role in the processing pipeline, example use: vertex shader, tesselation shader, geometry shader, fragment shader, compute shaders.

4. **Game engine architecture.** What is the difference between a game engine, game framework, and game middleware? Game engine component layers and its functionalities. What are common core and platform independence subsystems? How are memory and resources handled in game engines? Single- and multi-threaded loop types in a game engine.

5. **Basic rendering theory.** Mathematical description of material appearance – the BRDF. Correspondence between the surface appearance and the BRDF lobe shape. The reflection equation.

6. **Fundamental Monte Carlo approaches to rendering**. General principle of Monte Carlo integration, Monte Carlo estimator of a definite integral. Application of Monte Carlo integration to reflection equation (hemisphere sampling) and to the rendering equation (recursive ray/path tracing). General idea behind bidirectional path tracing and photon mapping.

7. **Function representation in linear bases**. Function representation as a linear combination of basis functions. Examples of basis functions in 1D. Finding the representation coefficients. Orthonormal basis functions and consequences for calculating the projection coefficients. Product integral in an orthonormal basis and its importance for rendering.

8. **Spherical harmonics**. Definition of real spherical harmonics. Organization of the SH basis functions into bands and their indexing. Order of SH representation, number of basis functions in n-th order representation. General idea behind the use of spherical harmonics in real-time rendering.

9. **Image-based lighting**. Problem definition: what is being computed (in terms of the reflection equation), general Monte Carlo approach. HDR environment maps and their acquisition. Different representations (mappings) used for environment maps (cube map, lat-long, dual paraboloid map, etc.)

10. **Basic real-time methods for image-based lighting**. Environment map prefiltering for diffuse and glossy (Phong) BRDFs. SH-based irradiance environment mapping: What is an irradiance environment map? Converting a SH-based representation of the environment map into an irradiance environment map, efficient shader code.

11. **Real-time methods for image-based lighting of arbitrary BRDFs.** Spherical harmonics based arbitrary BRDF shading: Expression of the reflection equation in the spherical harmonics basis; BRDF lobe representation; overall rendering algorithm. Filtered importance sampling: MC sampling of the reflection equation and its associated problems; basic idea of sample filtering and its implementation using MIP-mapping; suitable representation (mapping) of the environment map for the algorithm.

12. **Pre-computed (baked) global illumination in games**. General idea; illumination maps; advantages and limitations. Combination with normal mapping (radiosity normal mapping): What is a normal map? What is the issue involved in combining pre-computed GI and normal mapping? Application of SH-based irradiance environment mapping to the problem. More efficient solutions (H-basis, Half-life basis).

13. **Screen-space shading**. Deferred shading: idea, pros and cons. Ambient occlusion: definition, importance for rendering, general MC algorithm for computing AO, real-time screen-space AO.

14. **Sub-surface scattering**. What is sub-surface scattering and when is it important? BSSRDF and its difference from the BRDF. Generalized reflection equation. Practical dipole-based BSSRDF model, the diffusion profile. Off-line rendering with BSSRDFs: Monte Carlo sampling of the diffusion profile, point-based SSS solution. Real-time rendering with BSSRDFs: Gaussian approximation of the diffusion profile, texture-space filtering, scree-space filtering.

15. **Shadow algorithms I.** Why are shadows important? Shadow calculation in ray tracing. Shadow mapping: advantages and problems; field of view, shadow acne/biasing. Shadow map aliasing and its solutions; cascaded shadows maps. Shadow map filtering; percentage closer filtering.

16. **Shadow algorithms II**. Why are shadows important? Shadow calculation in ray tracing. Deep shadow maps. Shadow Volumes.

17. **Procedural content creation for games**. Top-down and bottom-up approach to procedural modeling. Procedural terrain modeling: terrain representation, use of Perlin noise for terrain modeling. Texturing terrains, tri-planar texture mapping. Procedural vegetation: L-systems and their use for plant modeling. Limitations of procedural modeling.