Game Engine Architecture

hannes@ru.is

Exam

- Date: 18 April
- Time: 09:00 - 12:00
- Location: V107

- No materials
  except calculator

Intended Learning Outcomes (1 of 2)

- Explain, compare and evaluate game engines
- Sketch the typical components of a runtime game architecture
- Use C++ development tools and apply best practice in object oriented C++ development
- Design and implement low-level engine systems that deal with:
  - start-up/shut-down, memory management, complex data types, engine configuration, file system, game resources, game loop, rendering loop and interface devices

Intended Learning Outcomes (2 of 2)

- Apply 3D math for solving game world problems
- Explain the core functionality of the rendering and ~~animation~~ system
- ~~Solve basic collision detection and use rigid body physics middleware~~
- ~~Explain the anatomy of a game world, game objects, data-driven game engines and the general construction of a runtime gameplay foundation system~~

Material

FOUNDATION

- Chapters 1-2: Introduction and Tools
- Chapter 3: Software Engineering
- Robert Galanakis: Tool Chain

LOW-LEVEL ENGINE

- Chapter 5: Engine Support Systems
- Chapter 6: Resources and the File System
- Chapter 7: The Game Loop and Real-Time Simulation
- Chapter 8: Human Interface Devices
- Harri Darri: Networks, Protocols and Distributed Systems

GRAPHICS AND MOTION

- Chapter 10: The Rendering Engine (Section 10.1)
- Páll Pálsson: Shaders in EVE Online

Also:
Labs and Problem Sets!

Exam Structure

- Explaining Concepts (20%)
- Algorithms / Calculations (20%)
- Implementations (Ogre) (20%)
- General Multiple-Choice (20%)
- Design Discussion (20%)