DM809
Computer Game Programming I: Graphics

Rolf Fagerberg

Fall 2008

Goals for Todays Lecture

Introduction to course:

- Motivation
- Contents of course
- Formalities of course
- Textbook
- Tentative courseplan

Why Computer Game Programming?

- Fun, attraction, curiosity
- Career goal (in US, game industry twice as large as movie industry in sales)
- Great display of use of many Computer Science subjects and courses:
  - Programming (DM502, DM503, DM528)
  - Algorithms and data structures (DM507)
  - Linear algebra and other math (DM527, MM501, MM502, MM505)
  - Numerical analysis (MM518)
  - AI (eg. finite automatons from DM517)
  - Computer architecture (DM506)

Computer Game Development

- Large game company (100 persons, 20 M$/year turnover):
  - Game programmers: 30-40
  - Game artists, model designers: 30-40
  - Game level designers, testers: 10-30
  - Game designers: 2
  - Game producers: 3
  - Business and management persons: 5

Computer Game Development

- Large game company (100 persons, 20 M$/year turnover):
  - Game programmers: 30-40
  - Game artists, model designers: 30-40
  - Game level designers, testers: 10-30
  - Game designers: 2
  - Game producers: 3
  - Business and management persons: 5
- Casual game company (1-3 persons, ?? $/year turnover):
  - Each person has many roles.

Computer Game Development

- Large game company (100 persons, 20 M$/year turnover):
  - Game programmers: 30-40
  - Game artists, model designers: 30-40
  - Game level designers, testers: 10-30
  - Game designers: 2
  - Game producers: 3
  - Business and management persons: 5
- Casual game company (1-3 persons, ?? $/year turnover):
  - Each person has many roles.

Computer games in Computer Science: the study of

> Methods and principles of game programming

Course Sequence at Imada

Fall 2008, 3rd quarter:

Computer Game Programming I: Graphics

Fall 2008, 4rd quarter:

Computer Game Programming II: AI

Spring 2009, 1st quarter:

Computer Game Programming III: Physics

Spring 2009, 2nd quarter:

Computer Game Programming IV: Project

Course Sequence at Imada

Fall 2008, 3rd quarter:

Computer Game Programming I: Graphics

Fall 2008, 4rd quarter:

Computer Game Programming II: AI

Spring 2009, 1st quarter:

Computer Game Programming III: Physics

Spring 2009, 2nd quarter:

Computer Game Programming IV: Project

Other possibilities: DADIU, bachelorproject.

Subjects covered (all three courses)

- GPU pipeline
- 3D geometry (transformation, projection)
- Rendering (color, textures, lighting and shading)
- Polygonal techniques (culling, level of detail, surfaces)
- Image based techniques (skyboxes, billboards,. . .)
- Game AI (path finding, chasing and evading, fighting, flocking,. . .)
- Collision detection
- Physics modeling

Subjects not covered

- Graphics APIs (self-study)
- Software engineering, testing
- Game engines
- Level editors, scripting
- Modeling
- Animation
- Sound, music
- Gameplay, narrative, study of genres

Textbook

Real-Time Rendering, 3rd edition
By Tomas Akenine-Möller, Eric Haines, and Naty Hoffman
Published by A.K. Peters, 2008
ISBN 987-1-56881-424-7

Course Plan

Course plan for Computer Game Programming I:

Subject	Lectures	Chapters
Introduction	1	-
Graphics pipeline	1	2, 3
Transforms	2	4, A4, 1
Shading	2	5, 7.1–4, more?
Textures	2	6
Image-based effects	1	10
Polygonal techniques	2	12
Acceleration algorithms	3	14

If time should permit, we could also look at: surfaces (Ch. 13) and CPU architechture (Ch. 18).

Formal Course Description

Prerequisites:	Programming (DM502+DM503), algorithms and data structures (DM507), vectors and matrices (DM527, MM505), and matematical maturity (MM501+MM502)
Literature:	Textbook
Evaluation:	Implementation projects (pass/fail), oral exam (7-scale)
Credits:	5 ECTS
Course language:	Danish or English

Time and Place

- Mondays 12.15-14.00
- Fridays 12.15-14.00

All lectures in Imadas seminar room.

No examinatorier (API study and project takes up the time).

Project

Small project (in groups of 2–3) must be passed to attend the oral exam:

Implement a 3D visualization of a very simple game

Programming language and graphics API of own choice. Must run on either Imada machines (Linux), or on Windows XP or Vista.

Some suggestions for API and language:

- C++ and OpenGL
- Java and OpenGL-binding (e.g. JOGL)
- C++ and DirectX

Use the net. Contains many resources.

Disclaimer

- Includes reading quite a number of pages
- Includes actual math
- Includes programming
- Includes work on issues not taught in course (graphics APIs)

Rather heavy workload, some of it on your own.