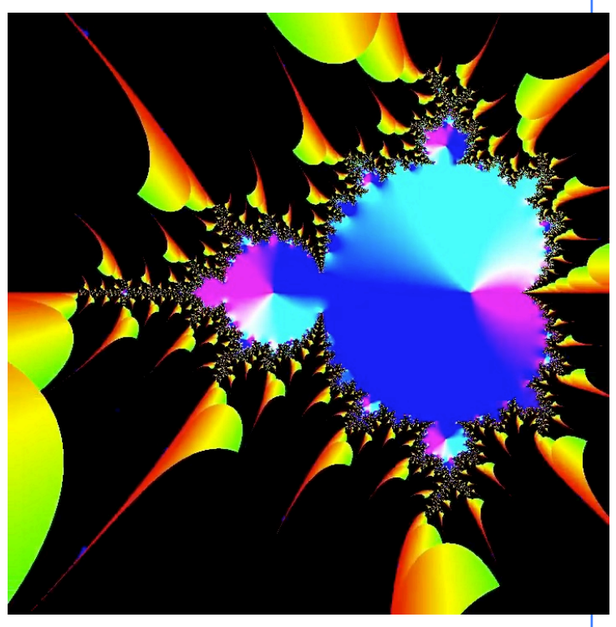
TSBK07/TSBK11

Computer Graphics

Ingemar Ragnemalm, ISY



TSBK07 Computer Graphics
TSBK11 Fundamental Computer Graphics

Spring 2025

Course leader/examiner/lecturer:
Ingemar Ragnemalm
ingemar.ragnemalm@liu.se / ingemar@ragnemalm.se

Course home page:

https://computer-graphics.se/TSBK07/

This lecture:

Course plan and overview

Projects

Graphics systems

Who am I?

- Lecturer/assistant professor
- Researcher
- Game & graphics programmer
- Game designer
- Hacker
- Geocacher
- Gamer (boardgames, computer games, RPGs)

Who are you?

- 2nd year on 3 year program
- 4th-5th year on 5 year programs
- International students

MOST of you have no prior experience of CG
SOME of you have some experience of CG
A FEW of you have extensive experience of CG

All of you know some programming. (Some are “wizards”)

People:

Examiner/lecturer:
Ingemar Ragnemalm

Lab assistants:
Susanne Ragnemalm
Martin Clason
David Berntsson
Robert Cranston

Changes and conclusions from last year:

Labs only for C++

Revision of the course book

Major change: Dugga tests on labs!

Changes this year:

Same duggas on both courses but with different grading limits

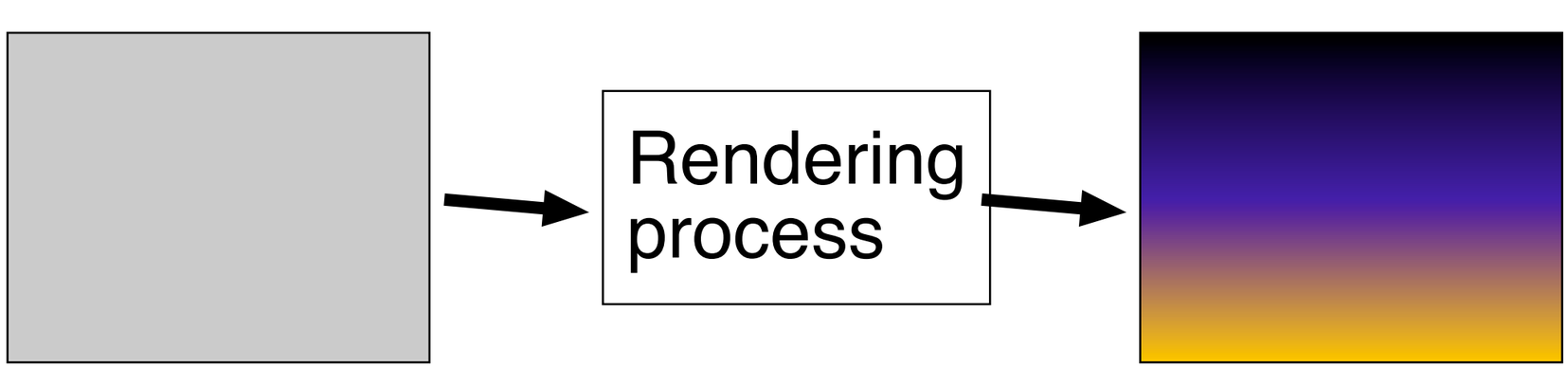
Some lab changes. Lab 4 easier for TSBK11.

New experimental labs (optional!)

Image-related technologies:

Image Processing
Image Editing
Computer Graphics
Image Analysis
Computer Vision
Image Coding
Image Compression
Graphic Arts
Tomography
etc…

Computer Graphics



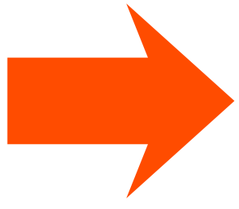
Description

Image

Computer Graphics:

Creating images from non-image data

Cube =
(10, 10, 10)
(10, 10, 20)
(10, 20, 10)
(10, 20, 20)
(20, 10, 10)
(20, 10, 20)
(20, 20, 10)
(20, 20, 20)
Camera =
(80, 60, 60) (15,15,15)



Lecture questions:

1. Which are the three primary colors?

2. What is a shader?

3. What is the typical format of a color pixel?

4. Why are 4 bytes better than 3?

5. What information is needed to describe a 3D scene?

Learning goals:

You should get:

- Experience of computer graphics programming (OpenGL and shader programming)

- Learn important concepts and methods for implementing computer graphics applicatons

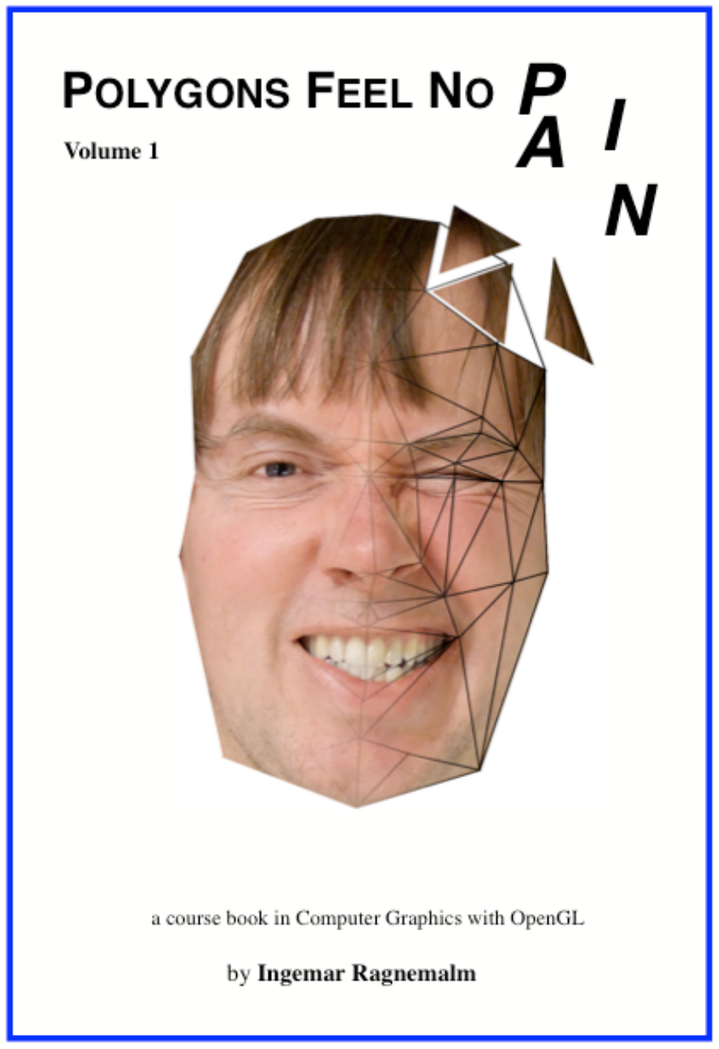
- Understanding of some important low-level graphics algorithms

Everybody should learn something new!

Course book:

Low-cost. Available at Bokakademin.

Also available on-line as PDF!



Time schedule

VT1:

Lectures (14)
Labs (4+)
Project specifications

VT2:

Project work
Project demonstrations
Reports
Lessons (2)
Written exam (if you need it)

Lecture plan

1. Introduction, graphics systems, API's
2. 2D transformations, OpenGL/GLSL introduction
3. 3D transformations, viewing, projection

4. Introduction to visible surface detection, light models, shading
5. Surface detail, texture mapping

6. More surface detail, more VSD
7. Rotation around arbitrary axis, normal matrix, multitexturing

8. Large worlds, high-level VSD, level of detail
9. Billboards, curves and surfaces

10. More curves and surfaces, animation
11. Collision detection, fractals and noise

12. Anti-aliasing
13. Ray-tracing and radiosity
14. Path tracing, low level algorithms

Changes in the later parts are likely!

Fundamentals

1. Introduction, graphics systems, API's

2. 2D transformations, OpenGL/GLSL introduction

3. 3D transformations, viewing, projection

4. Introduction to visible surface detection, light models, shading

5. Surface detail, texture mapping

and a little bit more

6. More surface detail, more VSD

7. Rotation around arbitrary axis, normal matrix, multitexturing

Going big

8. Large worlds, high-level VSD, level of detail

9. Billboards, curves and surfaces

10. More curves and surfaces, animation

11. Collision detection, fractals and noise

12. Anti-aliasing

Global illumination part plus low-level

13. Ray-tracing and radiosity
14. Path tracing, low level algorithms

Laborations

Lab 1: Intro to modern OpenGL with GLSL
Lab 2: Models, texture, camera, simple light
Lab 3: Virtual world, skybox, advanced lighting with GLSL
Lab 4: Terrain rendering

Projects

Small projects! ≈ 1.5 week of full time work.

Lessons

2 lessons at the end, preparing for the exam (unless your duggas were good!)

The lab material is available on the course home page:

https://computer-graphics.se/TSBK07/

Lab location:

Asgård, Olympen, Egypten:
Entrance 25, upper floor

Asgård, and Egypten have the best GPUs!

Olympen: GTX 1080
Asgård and Egypten: 4000 series

New variants of the labs, using Vulkan

Newer API, improves performance.

Experimental! For advanced students!

Created by David Berntsson, assisted by him.

Late changes in the lab material are possible.

The current lab material should be perfectly usable

but

the deadline for the lab material is at the start of the lab, so I give myself the right to make changes.

Projects:

Define a course-related problem to go deeper into.

2-4 people in each group.

Consider the project now and then during lectures and labs

Project plans should be handed in at the end of VT1!

Implement in VT2

Allocate time!

How to form a group

Balance: Avoid groups with dramatic difference in experience!

Size: Larger groups need more project management

If a group doesn’t work, let me know. Better split/ reform early than late.

One-person projects allowed - but not recommended

Operating system, language...

In the lab: C++ with Linux

The lab material supports C and C++, Mac, Linux and Windows.

It is possible to use any language that can link with C but it might take some work.

Resources:

It is allowed to use most material that you come across, i.e. on the web, like 3D models, certain libraries.

It is not allowed to do that without documenting it! You must tell what you use and what you have added yourself!

Do not replace the lab material! Use the lab material plus the Packages and Demo pages.

Rules to follow:

- Proper references to anything you didn’t do yourself. This makes your results stronger, not weaker!

- Project report in your own words - no copy/paste!

- Nothing should be written with ChatGPT or similar AI tools.

- Lab solutions should be written by *you*, not someone else!

I am obliged to report cheating! So don't!

Project examination:

- Short presentation (3-4 groups at a time)
- Demonstration on some other suitable computer (often your own)
- Written report

This should take place in may, before the exam period.

Project ideas:

- 3D labyrinth
- Interactive solar system
- Robot with moveable limbs
- Driving simulator
- Flight simulator
- Enhanced terrain rendering
- Large virtual reality with frustum culling

Advanced project ideas:

- Real time ray-tracer or path tracer
- Demo with several scenes with fun effects.
- Working game with some nice visuals.
- Game/VR with (advanced?) collision detection/handling
- Terrains with geomipmapping
- Particle systems
- Fractal vegetation
- Advanced light effects
- That neat, advanced effect that I always wanted to do.

But don’t aim too high!

Different people have different style…

All project should not be similar!

- Some focus on specific algorithms
- Some are “applications”, like games
- Some are artistic, neat “demos”

Do it *your* way!

Grades

Your grade is determined by the written exam or duggas!

...but...

Particularly good projects may qualify for bonus points on the exam/dugga score!

- More complex

- Polished, “finished products”

- Based on recent research

The bonus can not help you pass, but may give you a higher grade! May need to be applied for.

Level of bonus: 5 points (1/2 grade step) for "good"
10 points (a full step) for "very good"

Grades

Your grade is determined by the written exam or duggas!

10 points per dugga = max 50 points

TSBK07:

20 points = grade 3
30 points = grade 4
40 points = grade 5

New this year: TSBK11 has the *same* duggas, with preliminary grading:

16 points = grade 3
24 points = grade 4
32 points = grade 5

Less confusion, easier to reward ambitious students