Computer Graphics

MTAT.03.015

Raimond Tunnel

Study IT in .ee
Course Information

- **Course Page** [https://courses.cs.ut.ee/2019/cg/fall](https://courses.cs.ut.ee/2019/cg/fall)

- **Mailing List**
  
aine.ati.arvutigraafika@lists.ut.ee

- **Learning Environment (CGLearn)**
  
  [https://cglearn.eu](https://cglearn.eu)
Instructors

- **Raimond Tunnel** – jee7@ut.ee
  Lectures, Basic I, Basic II, JavaScript, C++

- **Jaanus Jaggo** – jjaggo@ut.ee
  Blending, Procedural Generation, Noise,
  Game Engines, Blender

- **Ats Kurvet** – akurvet@ut.ee
  Game Engines, Unreal Engine 4
Organization

• 6 credit course
  
  - **Lectures**: $15 \cdot 1.5h = 22.5h = \sim 0.9$ credits
  
  - **Practice Sessions**: $16 \cdot 1.5h = 24h = \sim 0.9$ credits
  
  - **Homework**: $2.3$ credits $= \sim 60h = 15 \cdot 4h$

  - **Project**: $40h = 1.5$ credits

  - **Exam**: $2h + 8h = \sim 0.4$ credits
Organization

• **Grade**

  • **Homework:** 40p
    Solve the tasks in CGLearn
  
  • **Project:** 30p
    Do something fun in a 2-3 person team

  • **Exam:** 30p
    General knowledge from the lectures and CGLearn
Organization

- **Grade**
  - **Homework:** 40p
    Solve the tasks in CGLearn
  - **Project:** 30p
    Do something fun in a 2-3 person team
  - **Exam:** 30p
    General knowledge from the lectures and CGLearn

You can earn more when you solve more tasks.
Organization

• Grade
  
  • **Homework:** 40p
    Solve the tasks in CGLearn
  
  • **Project:** 30p
    Do something fun in a 2-3 person team
  
  • **Exam:** 30p
    General knowledge from the lectures and CGLearn
  
• **Bonus Points** – *eg* APT GG organized game jams
Organization

• **Lectures** – Tuesday, 16:15, r403

• **Practice Sessions**
  • Basic I (JS), Basic II – Thursday, 12:15, r003
  • Basic I (C++), Game Engines – Thursday, 14:15, r003

• **Project Presentation** – In the exam session. TBA.

• **Exam** – In the exam session. TBA
Organization

• **Lectures** – Tuesday, 16:15, r403

• **Practice Sessions**
  • Basic I (JS), Basic II – Thursday, 12:15, r003
  • Basic I (C++), Game Engines – Thursday, 14:15, r003

• **Project Presentation** – In the exam session. TBA.

• **Exam** – In the exam session. TBA

NB! The exam session is in January!
Project

• Deadlines and instructions are here:
  https://courses.cs.ut.ee/2019/cg/fall/Main/Projects

• Can be any graphical application, prototype or a research project (e.g. replicating results from some paper).

• Extra ideas & time in the Computer Graphics Seminar (MTAT.03.305) and Computer Graphics Project (MTAT.03.328) courses.

Modules

• Basic I
  Geometry, transformations, projection, lighting, texturing and blending.

• Basic II
  Environment mapping, curves, procedural generation, ray tracing, global illumination, shadows.

• Game Engines
  Graphics in Unreal Engine 4 and 3D modelling in Blender
• You should finish two of the modules to the end
• You can also attend the other group and do their tasks for extra points (earning ~60p total).
Homework

• Each week we will cover topics in the practice sessions.

• Deadlines
  • Basic I – 27.10.2019
  • Basic II, Game Engines – Week before the exam

• It is recommended, however, to do the homework weekly, because:
  • You won't fall behind
  • You will get the points sooner
Homework

- We start doing the tasks together in the practice.

- You can probably get more up to speed with the tasks when attending the practice session.

- If you miss the practice, then there are instructional videos for the Basic I and Basic II tasks.

- Instructional videos for Game Engines TBA.
Material

- **CGLearn** – [https://cglearn.eu](https://cglearn.eu)
  - Material with interactive examples
  - Tasks (homework)

- In the lectures we will cover similar topics that are in CGLearn (but not 100% overlapping)
- In the practice sessions we will discuss the concepts in tasks and start implementing them.

CGLearn is my Master thesis work. There might be bugs, let me know immediately if you find some. :)

Material

- **Fundamentals of Computer Graphics**  
  P. Shirley, M. Ashikhmin, S. Marschner, 2009

- Provides a good and systematic approach to many topics we cover.

- Explains the math behind the topics.

- Who is interested in borrowing it?
Material


Journals

- ACM SIGGRAPH
- IEEE Transactions on Visualization and Computer Graphics
- Computer Graphics Forum
- NVIDIA's GPU Technology Conference Presentations
- Ke-Sen Huang's Resources Page

Books

- The OpenGL Programming Guide (Red Book) (9th edition)
- GPU Gems (1, 2 and 3)
- GPU Pro series (1 to 7)
- GPU Zen series (1 to ...)
- Bézier and B-Spline Techniques – H. Prautzsch, W. Boehm, M. Paluszny
- Physically Based Rendering: From Theory to Implementation – G. Humphreys and M. Pharr (3rd edition)
- Vulkan Cookbook – P. Lapinski
- The Algorithmic Beauty of Plants – A. Lindonmayer and P. Prusinkiewicz
Lectures

HEY YOU

DON'T BE SO SAD,
I'M HERE FOR YOU

imgflip.com
Lectures

1) Me talking about the topics, Me answering questions (I do not ask you)

2) Me talking about the topics Class answering the questions (I ask the class)

3) Me talking about the topics You answering the questions (I ask people)

4) You talking about the topics (In small groups) Each time groups prepare presentations
Computer Graphics

- So, what is computer graphics? Applications?
Computer Graphics

• Games and entertainment

Deus Ex, 2000

Deus Ex: Human Revolution, 2011
Computer Graphics

- Games and entertainment

Deus Ex: Mankind Divided, 2016

Cyberpunk 2077, 2019?
Computer Graphics

- Graphical user interfaces (GUI)

Microsoft Word

WinDirStat

Photoshop
Computer Graphics

- Computer Aided Design (CAD)
Computer Graphics

- Scientific simulation visualization

HIV particle

Parker Solar Probe
Computer Graphics

- Data visualization

World ocean currents, NASA
http://deepbluehome.blogspot.com/2012/03/cool-cat-currents.html

World's Biggest Airlines
http://spatial.ly/2012/06/mapping-worlds-biggest-airlines/
Computer Graphics

- Simulations

Road Planer Simulator (Maantehöövli simulaator)
Estonian Road Museum (Eesti Maanteemuuseum)
http://muuseum.mnt.ee/

http://en.wikipedia.org/wiki/Flight_simulator
Computer Graphics

• Art

*Blithe Certainty* by Kerry Mitchell

See also: http://flam3.com/flame.pdf
And: https://screen.aptgg.ee/

Buddhabrot
https://en.wikipedia.org/wiki/Buddhabrot
Technologies

- What to use to make computer graphics?
- 2D graphics? 3D graphics?
- Any previous experience?
Technologies

Basic I

- Basic II
- Game Engines
Technologies

Lower level technologies and libraries

JavaScript

Three.js

Basic I

Basic II

Game Engines
Technologies

Lower level technologies and libraries

JavaScript

Three.js

C++

WebGL

Basic I

Basic II

Game Engines

Allegro

OpenGL

glm
Technologies

Lower level technologies and libraries

- JavaScript
  - Three.js
- C++
  - Allegro
  - OpenGL
  - glm

High level software

- Basic I
  - Game Engines
- Basic II
2D Technologies

Language
<canvas>
HTML Element
Browser

Language

Library
Allegro

Basic I, Practice Session Group 1

Basic I, Practice Session Group 2
OpenGL, WebGL

- API-s for communicating with the GPU drivers.

- WebGL is based on OpenGL ES 2 (Embedded Systems).

- They have GLSL (Shader Language) which is used to program code executed on the GPU. It is similar to the C language.

Different calls to GPU

GLSL code → Execute this code...
OpenGL, WebGL

Standard Graphics Pipeline

1. Construct geometry
   Define transformations
   Assign material properties
   ...

2. Vertex Transformations

3. Culling & Clipping
   Determine front-facing triangles
   Determine which vertices are visible

4. Rasterization
   Fill the triangle with fragments

5. Fragment Shading
   Calculate correct color values

6. Visibility Tests
   Is the fragment visible?
   Blend multiple fragments

7. Blending

---

Intel Core 2 Duo

NVIDIA GeForce GTX 1080
Standard Graphics Pipeline

- **Construct geometry**
- **Define transformations**
- **Assign material properties**

**Vertex Transformations**

- **Vertex Shader**
  - *Object’s local space → viewport space*

**Culling & Clipping**

- Determine front-facing triangles
- Determine which vertices are visible

**Rasterization**

- Fill the triangle with fragments

**Fragment Shading**

- Calculate correct color values

**Visibility Tests Blending**

- Is the fragment visible?
- Blend together multiple fragments
Geometry

- Everything starts with the geometry
- Geometry defines the 3D objects in the scene

What is depicted here?
Geometry

- Although a cube has only 6 faces, we define it as a set of 12 triangles (12 faces).
- Why triangles?
Geometry

- Triangle very useful scientific facts:
  - Triangle vertices always form a plane
  - Every polygon can be converted to triangles
  - Triangles are easy to rasterize
  - Triangles are convex and simple
  - Three vertices always form a convex and simple triangle
Geometry

• Convex polygon (*kumer hulknurk*) —
  1) All convex combinations of the vertices are inside the polygon
  2) Straight paths to all vertices from any point inside the polygon, are also inside the polygon
  3) The polygon is equal to its convex hull
  4) All interior angles are ≤ 180°
Geometry

- Simple polygon (*lihtne hulknurk*) – no intersecting edges

- What is the difference between those triangles?
Geometry

- Coordinate system can be left- or right-handed.
- We usually use the right-handed system.
Geometry

- In a right-handed coordinate system, the positive direction of an angle is counter-clockwise.

This means that the order of polygon vertices will define a front face and a back face.
What did you learn today?

What more would you like to know?

Next time: Geometry
(points, vectors, coordinate systems, ...)