

HPPS Epilogue

Troels Henriksen (athas@sigkill.dk)

DIKU
University of Copenhagen

Course Evaluation

Exam Hints

This Year's Exam

The numbers

The words

With some editing and selection to pick representative comments.

Relevance

- Koncepterne i forelæsninger havde en klar rød tråd til den kommende aflevering
- Rigtig god forbindelse fra øvelser til afleveringer. Det gav mening hvorfor man havde lavet øvelser, når man gik i gang med opgaverne
- Struktureret undervisning - det var nemt at følge med i hvilket emne vi var i gang med og hvad jeg burde lære.
- Der var for mange øvelser til øvelsestimerne, man endte tit med at lede efter de mest afleveringsnære.
- Jeg synes ikke det har været specielt udbytterigt at komme til forelæsningerne for at man kunne løse opgaverne. Derudover synes jeg især at flere af forelæsningerne virkede decideret irrelevante for det vi lavede og i højere grad som om nogen havde en tjekliste af ting vi skulle igennem end et meningsfuldt pensum.

Jeg var virkelig nervøs for det her kursus og regnede med, at det ville blive meget svært for mig, så jeg var i baglås, inden det overhovedet begyndte. Den ærlige, lidt ironiske og lette tilgang i den første introduktionsforelæsning hjalp mig enormt meget med at være rolig, og det er en rød tråd, som har trukket sig igennem hele kurset. Øvelserne bygger op i et tilpas tempo, så man er tilstrækkeligt udfordret, uden at det føles helt overvældende og umuligt. Den gode sammenhæng mellem øvelser og afleveringer er meget tilfredsstillende og motiverende. Forelæsningerne er spændende, udbytterige og underholdende, og det føles, som om der er lagt kærlighed i det her kursus. Det endte med at være et af mine yndlings.

Assignments

- Afleveringer var lidt overvældende - mere generel introduktion til øvelses time ville havet hjulpet
- Interessante opgaveformuleringer
- Kodemæssigt svære afleveringer med den nye syntaks. A5 ligger for sent i forhold til eksamen, enten fjern eller flyt til tidligere.
- Som sagt, opgaveformuleringerne var interessante. Fedt at arbejde med kNN, store databser osv. Men omvendt var de 5 opgaver også meget tunge hver især. Måske kunne man skære lidt fra, eller forsimple dem en smule så en opgaveformulering f.eks. ikke er 10-15 sider lang
- Afleveringerne var for store så havde ikke tid til også at lave øvelserne, øv.
- 1-2 assignments less. Having assignments every week meant that there wasn't time to read the provided material and solving the weekly exercises.
- The assignments were often very difficult to understand and difficult just to get started on.

GitHub

- + Super godt. Elsker github.
- + Jeg syntes github fungerede fint og gav et godt overblik, men nogle kan selvfølgelig godt stejle over, at Absalon ikke er roden til al viden.
- + Synes det nice med github, ville dog være rart at kunne downloade hele foldere til øvelserne, så man ikke skal download en fil ad gangen.
- + We barely used it [Absalon] due to GitHub. Made sense, given the course.
- + Fantastisk at benytte github!
- Det kunne være rart, hvis forelæsningerne også kom ind på absalon og ikke kun GitHub, da jeg synes jeger lidt svært nogen gange ved at finde slides.
- Lidt øv at I ikke kunne holde jer til det system, men skulle finde på jeres eget
- jeg foretrækker absalon frem for github (og discord), så alt er samlet ét sted

Exercise classes

- lidt mere aktive TA's, så man får mere ud af øvelsestimerne for dem som synes det er svært.

Live coding

- Virkelig gode live code sessions
- god livecoding
- Live coding til forelæsninger var rigtig udbytterigt.
- especially the live-coding was very helpful
- Live coding - noget af de bedste live coding jeg har oplevet gennem uddannelsen
- godt med live kodning
- Jeg synes at 'live coding/testing' fungerer rigtig godt.

Der var lidt problemer med slides her og der, kunne godt optimeres, hvis jeg skal finde på noget :).

Opgrader nogle af jeres slides, plsss :)

At eksamen ikke ligger i sidste undervisningsuge, da vi mindste en hel uges undervisning i det andet fag vi har samtidig, da eksamen ligger oven I det

Det ville være rart hvis de programmer og funktioner man skulle bruge var tilgængelige for all uanset computer eller system. Her tænker jeg specifikt på problemerne med mac, openmp og gcc

Det ville være rart hvis de programmer og funktioner man skulle bruge var tilgængelige for all uanset computer eller system. Her tænker jeg specifikt på problemerne med mac, openmp og gcc

Can someone remind me why it's actually cool & good
that Apple disables OpenMP in their gcc-
impersonating C compiler?

Det ville være rart hvis de programmer og funktioner man skulle bruge var tilgængelige for all uanset computer eller system. Her tænker jeg specifikt på problemerne med mac, openmp og gcc

Can someone remind me why it's actually cool & good that Apple disables OpenMP in their gcc-impersonating C compiler?

what a weird way of putting this from someone who iirc actually works on compilers. you might as well be asking why apple disabled fortran for aarch64.
like there's a turnkey perfect openmp implementation that we just spitefully set on fire

openmp is a massive language extension / library.
the current openmp spec is considerably longer than
the c standard. there is some existing work in
clang to support it – i did most of the code review
on that, actually – but it is not simple to enable

Anything I missed?

Behind the scenes

<https://sigkill.dk/writings/teaching.html>

Course Evaluation

Exam Hints

This Year's Exam

What is the purpose of an exam?

- A way for you to demonstrate your achievement of course learning goals.
- <https://kurser.ku.dk/course/ndab20001u/2025-2026>

You are graded on your demonstration of these learning goals.

How this happens in practice

The HPPS exam is a take-home programming problem where you have to demonstrate your *skills* through programming, and your *knowledge* by analysing your code and reflecting upon its properties, by answering specific questions.

- Form is quite similar to assignments.
- A mixture of tasks.
 - ▶ Something about reading/writing file formats.
 - ▶ Something about a basic sequential algorithm.
 - ▶ Something about parallelising it.
 - ▶ And perhaps surprises.
 - ▶ Some tasks are tightly constrained and come with existing skeleton code (see the exercises).
 - ▶ Other tasks are more open and will require you to design programs from scratch.

Implications for questions

- These are not trick questions, although some may be subtle.
- *Always* asked in the context of your code—these are *not* general questions about course material.
 - ▶ But not just about your runtime results—some questions may be hypothetical (“why *might* X be (in)efficient?”).
- Some questions are guaranteed to be present:
 - ▶ “How did you parallelise X?”
 - ▶ “What is the temporal and spatial locality of Y?”
 - ▶ “Show speedup and report scalability of Z.”
- Answer questions precisely and succinctly.
 - ▶ Nonsense raises doubts about your mastery of the learning goals, *even if you also say the right thing along the way*.
 - ▶ You don’t need to restate parts of the course curriculum, you just need to use it.
 - ▶ **Let us look at a reference solution.**

The code is also important

Forms the basis for your questions.

You are evaluated both on correctness, performance, and style.

Regarding style

- Use consistent indentation.
- Don't leave in blocks of commented-out code.
- Don't leave in noisy debug prints.
- Don't leave in unnecessary files.
- Cleanly deallocate memory.

Style is not the most important part, but it *will* have an impact on your assessment.

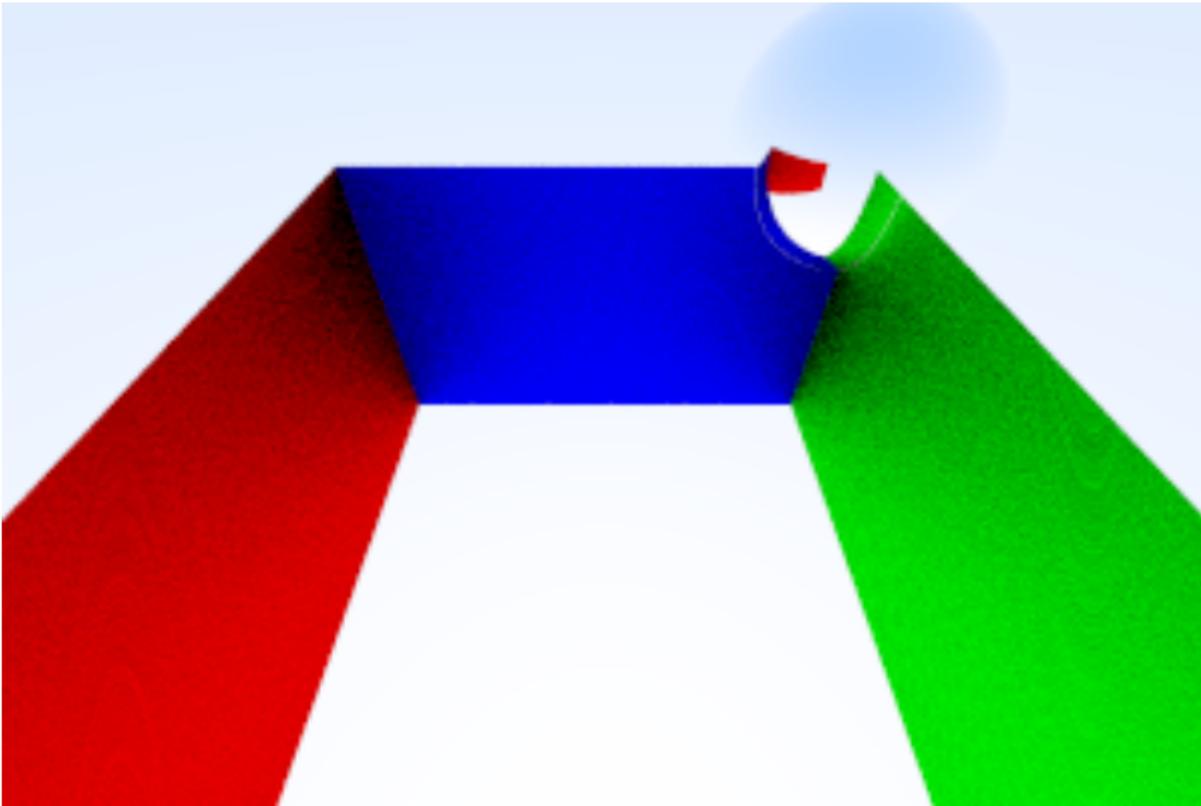
Course Evaluation

Exam Hints

This Year's Exam

Following information is non-normative.

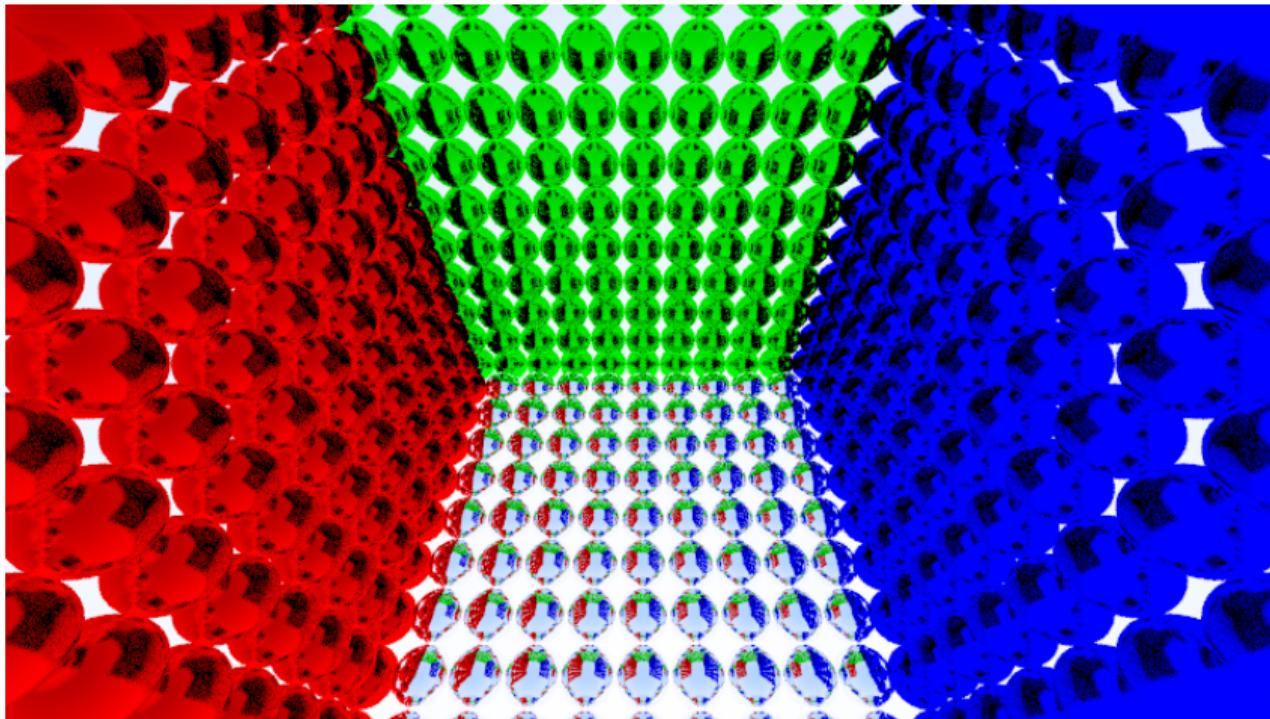
Ray tracing



Your starting point is a fully operational ray tracer

- Almost identical to the one from the exercises this week.
- You have to *optimise* it.
- You have to *analyse* the behaviour of the resulting program.
 - ▶ Empirically and theoretically.
- You have to implement *additional relevant tools*.

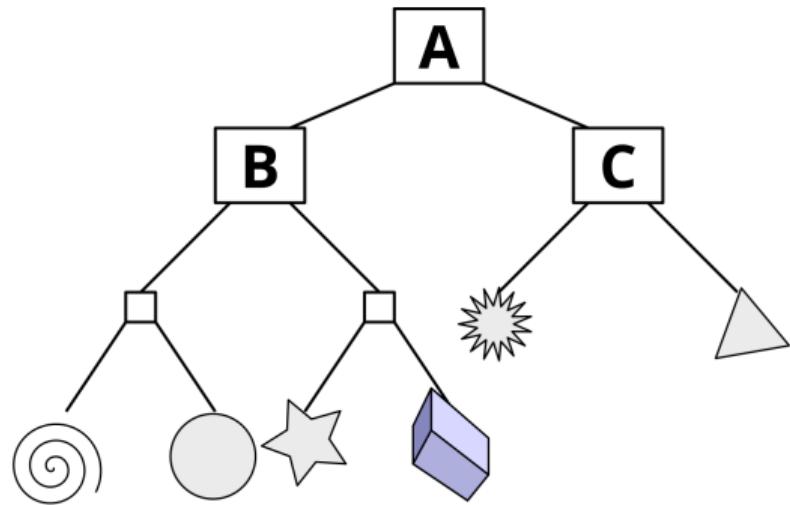
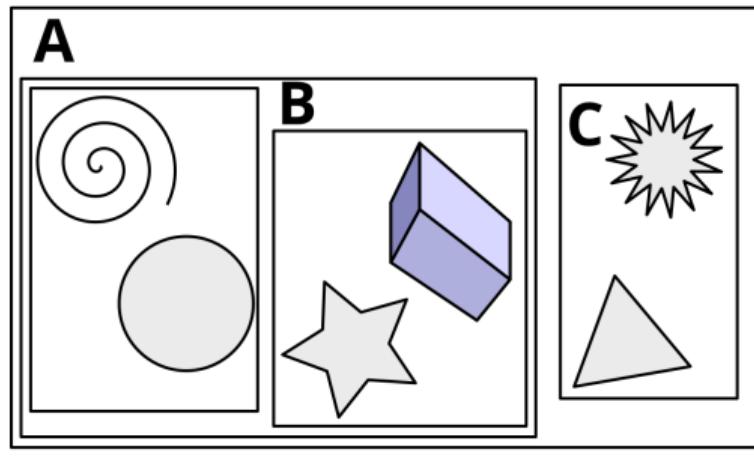
Scenes with many objects



The most costly part of ray tracing is finding ray-object collisions.

Bounding Volume Hierarchies

We optimise collision checking with a *bounding volume hierarchy* (BVH).



You have to implement one of these.

Not necessary for the exam

- Having an understanding of graphics or 3D geometry beyond the basic notion of what a 3D coordinate is.
- Having any kind of artistic sense—you do not need to create new scenes.

Questions?

Otherwise, good luck and see you next year.