

Welcome to Game Programming

Ole Fogh Olsen

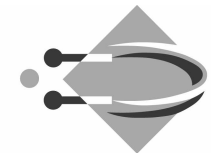
Kim Steenstrup Pedersen

Thomas Riisbjerg

Jesper Juul

A guest lecturer from industry

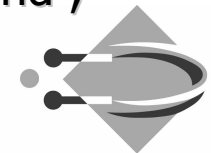
www.itu.dk/courses/SSPG/F2005



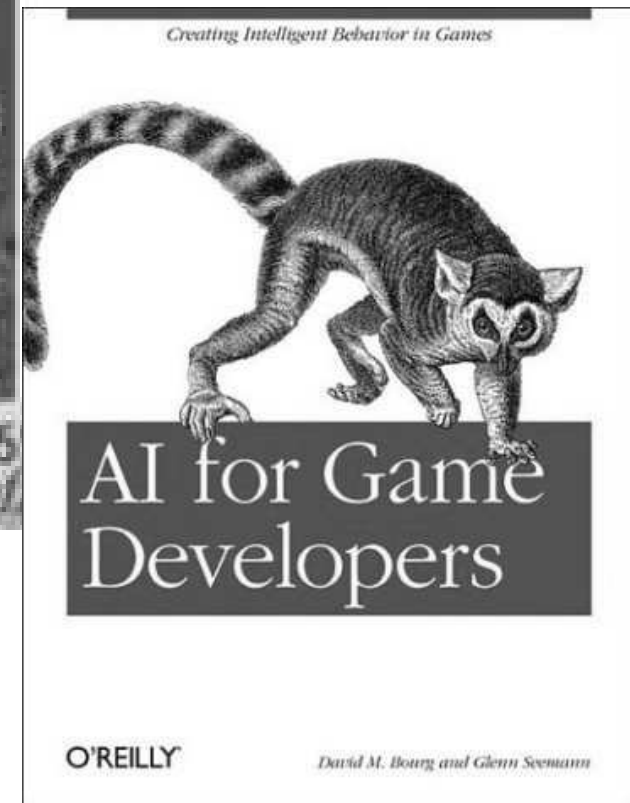
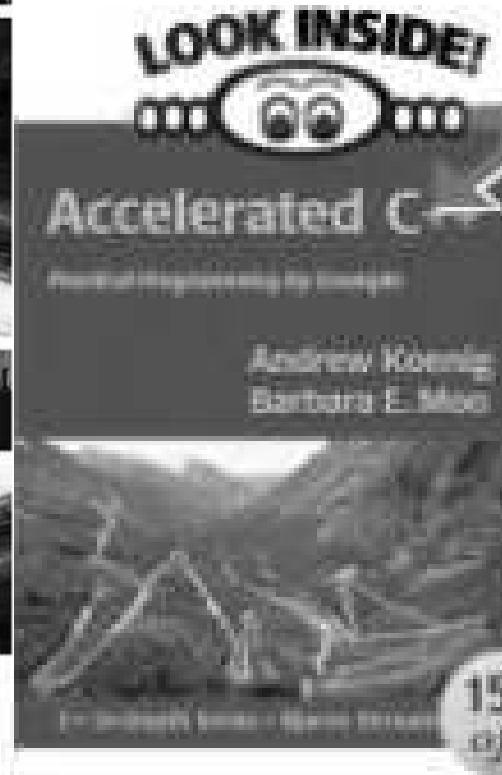
The **IT** University
of Copenhagen

Tentative Lecture Plan

- 1 Course Content, Game development, Game architecture, Software Libraries, c++, Compiler, linking, Panda, Python (4/2) OFO/KSP
- 2 C++, Data structures, intro graphics, scene graphs (11/2) KSP
- 3 Program design, Game Engines, Game development (18/2) KSP
- 4 Graphic engines, event queues, I/O (25/2) KSP
- 5 Game Objects, Animation (4/3) KSP
- 6 Physics, simulation, Collision detection (11/3) OFO
- 7 AI for car races (18/3) OFO
25/3 Easter break –
- 8 MAI (1/4) OFO
- 9 What makes a good game, How Game design influences the programming (8/4) JJ
- 10 Real time aspects, Synchronisation, multimedia programming, Sound , video (15/4) OFO
- 11 Network Games, Wrapping up, Exam talk (29/4) OFO



The Books + some copies.



Exercises and Assignments

Final production: CAR RACE

Production (report+cd) is part of the exam to be submitted at the exam office 29.april before 3pm.

Sub assignments support the production but are not part of the exam. 1-2 weeks for each subassignments plus a few ordinary exercises.

Groups of 4

Groups with 1 or 2 will not have their assignments corrected.



Game development philosophy

Rules of thumb

Every principle has a range of validity.

Iterate Start small. Get bigger through small, incremental steps.

Avoid presenting single solutions to critical tasks.

If something does not work, stop doing it.
Acknowledge your mistakes. Learn from them.

If something works, keep doing it.

Avoid repeating things you do wrong. Avoid having to redo things you've already done right.

Weigh pros and cons for all potential solutions. Avoid being dogmatic.



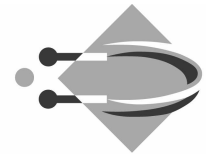
Hacker's charter

Claim : What game programmers do is different to what other programmers do.

Conclusion : Structures and disciplines evolved elsewhere in the computing industry can not be used.

Consequence: Therefore any practice or lack of it goes.

Is this really true!



So what is different?

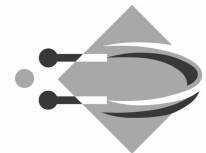
Speed : Updates within 50 Hz

Control : Interactive and Responding

Robustness : Never crash

Commercial reality: Delivering on time within budget.

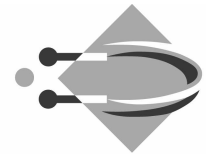
But this is not special for game development.



The IT University
of Copenhagen

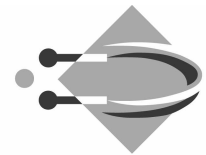
Some differences

- Open-ended design
The possibility of shifting items between core, required and desired features is unique.
- Heuristic content
Balance of core content and player expectation.
- Artistic content
Still images, video, special effects, 3D models.
- Control methodology
User-interface design
- Complexity reduction
It must feel right but it does not have to be a model of reality.
- Combinations of the above might be unique.



Conclusion

Game development is no more unique than any other branch of software development



The **IT** University
of Copenhagen