Final lecture:
Competition + lessons learned

EDA075
Mobile Graphics

Michael Doggett

Department of Computer Science

Lund University

iPhone Competition

- Esteemed jury:
  - Tomas Akenine-Möller, Intel Lund & Lund University
  - Carl Loodberg, Illusion Labs
  - Jacob Munkberg, Intel Lund & Lund University
  - Henrik Ohlsson, ST Ericsson
  - Paul Rosengren, Netville

iPhone Projects

- **3D Wall**
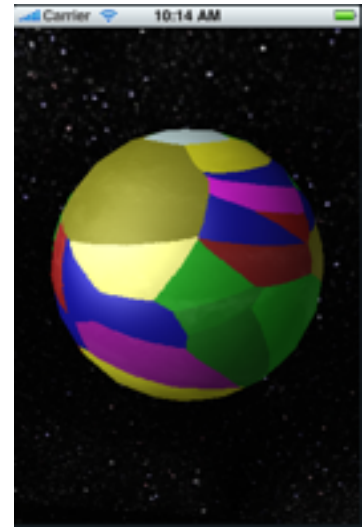 - Henrik Tydesjö and Marc Klefter

  - **3D Space Invaders**
    - Christofer Bach & Bereket

- **Tower Defense**
  - Simon Thunberg and Christian Simonson

3
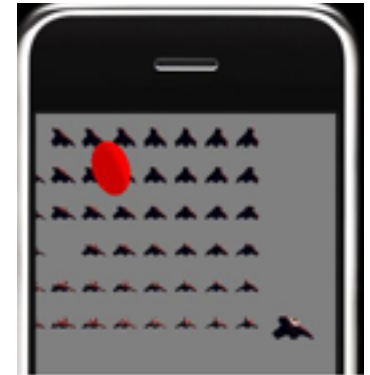
iPhone Projects

- **PlanetMenu**
  - Oskar Gustafsson and Katharina Schade

- **Risk**
  - Pontus Lindberg-Parker and Johan Teleman

- **Pacman 3D**
  - Erik Iverot and Robert Åkemalm

4

iPhone Projects

- **Defense**
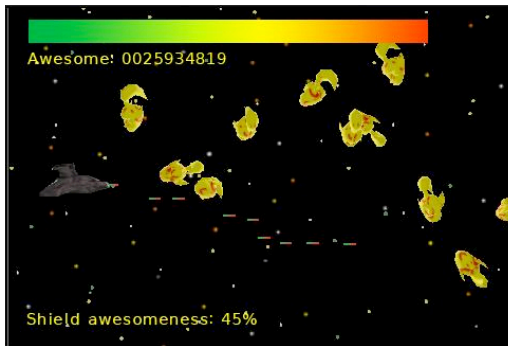 - Martin Haetta Evertsen and David Fabian

- **Texas Long Road Massacre**
  - Patrik Andersson and Jonas Oscarsson

- **The Awesomeness Game**
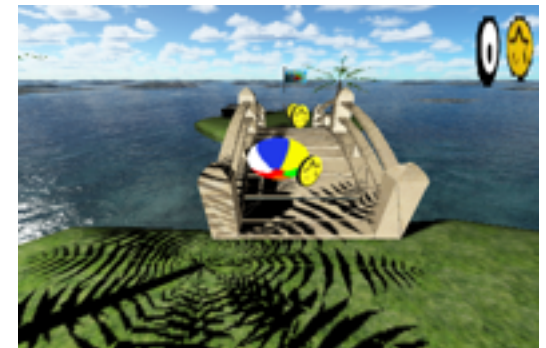 - Dennis Laks, Magnus Winter and Gustav Peterz

iPhone Projects

- **Poo fighter**
  - Eric Fransson and Robert Magnusson

- **Warp Ball**
  - Erik Bäckström and Kristian Sylwander

The jury will leave us now...
They'll be back in a little while with the results!

Graphics Hardware Optimization

- Rules: use 3072 bytes of onchip memory for reducing bandwidth usage as much as possible
  - Implement algorithms from lectures
  - Invent your own stuff
- Render at 320x240 pixels
- Scene #5
  - Multitexturing...

- Naive app with 2048 kB texture cache:
  - 20.8 +   5.9   + 11.0 + 118.4 = **156** MB in total
  - [depth + depth_clear + color + texture]

- If you're really clever (not), use 3072 kB texture cache:
  - 20.8 + 5.9 + 11.9 +**101.9** = **140** MB in total

Best algorithms and tricks...
Miscellaneous

- Tile-based rasterization
- Color buffer: RGB instead of RGBA
  - Reduction from 32 bits to 24...
  - This was the only color compression that could work

Best algorithms and tricks... Texturing

- Texture compression
  - S3TC/DXT1
  - reduced texture bandwidth down to <10MB from 133MB without
  - Reduction from 32 bits to 4 bits/texel
  - 8x increase in texture cache size
  - If compressed in cache, we can have very small texture cache (512 bytes)
- Replacement: Least-recently-used (+fully associative)
  - Why? Multitexturing kills current cache

Best algorithms and tricks...
Depth buffering

- Zmin culling
  - Easy to update.
  - Saves reading in Z for tile.
  - Could use as little as 8 bits
  - Could store the Zmin in external memory
- Zmax culling
  - Update of zmax is very expensive when whole tile has to be read.
  - One solution is to reduce the update frequency
  - Depth cache and depth compression needed
  - Could use as little as 8 bits

Depth buffering

- Depth compression
  - Tile table storing depth compression format
  - umax and umin per tile, plus offsets.
  - Depth offset: with 16 bits depth + 8 bits offsets -> 90% of tiles could be compressed
  - DPCM: 4x4 tiles -> good results
- Depth clear
  - Store 1 bit per tile to indicate tile is clear.
  - Only write out clear z value to tiles that don't get a new triangle's Z written to them.

How to pass the project?

- I'll read the reports and look at code
  - Might require updates of report and further explanations from you

- We will put all the .exe on the course website
  - If you have an updated exe please send it
  - If you don't want you exe on the web, let us know

Advanced Shading & Rendering

- Course in graphics at LTH

- Spring, VT2

- About

  – Photo-realistic rendering

  – Real-time shading

Real-time screenshot from another course: Advanced Shading And Rendering, VT2, LTH
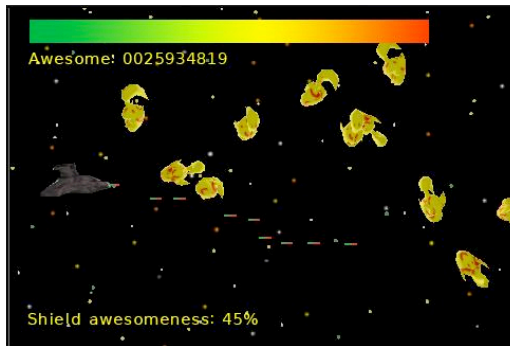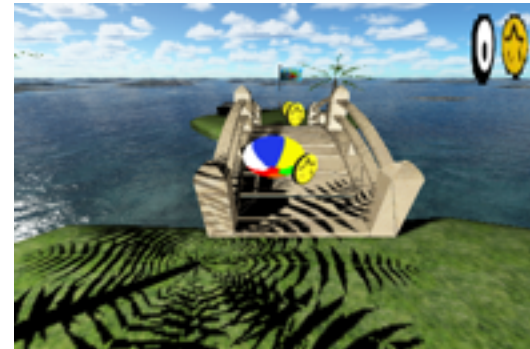

Courtesy of Paul Debevec

14

The end

Who won?

16

iPhone

- The word goes over to the jury

- 1st place

- **Warp Ball**
  - Erik Bäckström and Kristian Sylwander

- 2nd place

- **The Awesomeness Game**
  - Dennis Laks, Magnus Winter and Gustav Peterz

Graphics Hardware Optimization

- 2nd place:
  - Linus Mårtensson and Alexander Toresson
  - 48.6 MB

- 1st place:
  - Erik Hansson and Martin Sträng
  - 34.1 MB

The end... for real.