Computer Graphics Seminar

MTAT.03.305

Spring 2015
Contact Information

- Raimond Tunnel – jee7@ut.ee
- Konstantin Tretjakov – kt@ut.ee
Organizational Information

- 16 seminars:
  - 3 introductory lectures
  - 11 student presentations
  - 2 project demonstrations
We hope that...

- 15 seminars
  Attendance: 22.5h = 0.85 credits

- 1 seminar
  Preparation: 16h = 0.6 credits
  Conducting: 1.5h = 0.05 credits

- Project
  Everything: 40h = 1.5 credits
... but it may happen that ...

• 15 seminars
  Attendance: 22.5h = 0.85 credits

• 1 seminar
  Preparation: 56h = 2.1 credits
  Conducting: 1.5h = 0.05 credits

• Project
  Everything: 0h = 0 credits
What am I even doing here?
What do I see?
What about this one?
This should be easy...
Regular seminars

- Listen to your fellow student's awesome presentation
- Ask questions, discuss
- X > 1 heads are better than one
Your seminar

• Choose an interesting CG topic
• Make the seminar fun and interactive
• Present some applications / demos
• Workshop
Sidetrack: Gamma correction
Sidetrack: Gamma correction

Sidetrack: Bloom effect

Need for Speed: Most Wanted

Elephant's Dream

Hitman: Absolution

Warframe: https://www.youtube.com/watch?v=gYHxhlvEyHk
Back to the main track
How do I choose a topic?

• I just gave you two possibilities:
  • Shader effects (like the Bloom effect)
  • Gamma correction
• Read something and find interesting topics
  • OpenGL's Red Book
  • GPU Gems
  • More "sophisticated" literature
• Continue on some already discovered theme
  • My example: Procedural tree generation?
How do I choose a topic?

- Continue on some already discovered theme
How to choose a topic?

- OpenGL ver 3.0 & 3.1
- Practical
- Basic topics:
  - Viewing
  - Color
  - Lighting
  - Blending
  - Textures
  - Buffers
How to choose a topic?

- Advanced topics:
  - Display lists (perf.)
  - Tessellation
  - Quadrics
  - Evaluators (curves & surfaces)
  - NURBS
How do I choose a topic?

- Covers all topics already mentioned and more
- Math heavy, but most of it you should be at home with
# Extra conditions!

<table>
<thead>
<tr>
<th>First time participant</th>
<th>Returning participant</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSc, MSc</td>
<td>MSc, PhD</td>
</tr>
<tr>
<td>No additional requirements – you can choose any CG-related topic.</td>
<td>Your topic should be related to several scientific articles / a book. ACM SIGGRAPH (Special Interest Group on GRAPHics and Interactive Techniques): <a href="http://www.siggraph.org/">http://www.siggraph.org/</a></td>
</tr>
</tbody>
</table>
Previously...

PREVIOUSLY, ON SCRUBS...

Oh Wait, This Ain't Scrubs...
3D Scan Data

Pointcloud in MeshLab

Constructed model in Blender

http://ikuz.eu/2014/04/03/proof-of-concept-from-3d-scanner-to-animated-model/
Bump Mapping

• Change surface normals according to a surface height map.
• Surface normals are used in light calculations
Bump Mapping

- Use finite (central) difference to calculate the derivative from the bump map.
- Assemble the gradient, transform to correct space, add to the surface normal.

Finite difference schemes
From height map to normal map

Gradient

\[ x_{grad} = \text{pix}(x - 1, y) - \text{pix}(x + 1, y) \]

\[ y_{grad} = \text{pix}(x, y - 1) - \text{pix}(x, y + 1) \]

\[ \vec{n}' = \vec{n} + U \cdot x_{grad} + V \cdot y_{grad} \]
Bump Mapping

https://www.shadertoy.com/view/ldjSDW
Real-Time Water Surface Rendering

- Reflective (and refractive) surface
- Wavy surface
- Caustic effects
- Transparency
- ...
Real-Time Water Surface Rendering

- Reflection off the water surface
Real-Time Water Surface Rendering

- Waves modeled with sine waves

https://www.shadertoy.com/view/ld2SRy
Projections

• Orthographic
  • Used in engineering

• Oblique
  • Used in magazines

• Perspective
  • Used in games, movies
  • The way we see the world
Projections

- Clip space
Projections

- View volume \([-1, 1] \times [-1, 1]\)
Projections

- From view volume to screen space

\[
\begin{pmatrix}
\frac{x_{\text{screen}}}{2} \\
\frac{y_{\text{screen}}}{2} \\
\frac{z_{\text{canonical}}}{2}
\end{pmatrix} =
\begin{pmatrix}
\text{width} & 0 & 0 \\
0 & \text{height} & 0 \\
0 & 0 & 1
\end{pmatrix}
\cdot
\begin{pmatrix}
\text{width} & 0 & 0 \\
0 & \text{height} & 0 \\
0 & 0 & 1
\end{pmatrix}
\cdot
\begin{pmatrix}
x_{\text{canonical}} \\
y_{\text{canonical}} \\
z_{\text{canonical}}
\end{pmatrix}
\]

\[
\text{point}' = M_{\text{viewPort}} \cdot M_{\text{projection}} \cdot M_{\text{camera}} \cdot \text{point}
\]
Procedural Generation

- Creating content algorithmically
- Textures
Procedural Generation

- Objects, animations

Borderlands 2

Starbound

L-System tree

Overgrowth

https://www.youtube.com/watch?v=SAtwQa8t_3g
Procedural Generation

- Worlds

Binding of Isaac: Rebirth

Terraria

Minecraft
Post-Processing Effects

- Tone Mapping (High / Low Dynamic Range)
- Color Correction
- Anti-Aliasing
- Eye Adaption
- Bloom
- Motion Blur
- Depth of Field
- Lens Flare
### Post-Processing Effects

- **Bloom** – uses Gaussian blur

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
</tr>
<tr>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
</tr>
<tr>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
</tr>
</tbody>
</table>

**Ordinary Mean / Blur Kernel**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.08</td>
<td>0.12</td>
<td>0.08</td>
</tr>
<tr>
<td>0.12</td>
<td>0.20</td>
<td>0.12</td>
</tr>
<tr>
<td>0.08</td>
<td>0.12</td>
<td>0.08</td>
</tr>
</tbody>
</table>

**Gaussian Kernel**

[Gaussian Kernel Calculator](http://dev.theomader.com/gaussian-kernel-calculator/)
Re-Texturing

- Take a photograph of the red shirt on a robotic mannequin
- Generate the photographs for the blue and green shirts
Re-Texturing
WebGL, Three.js, Raphael.js

- Graphics technologies / libraries for the web
  - WebGL – 3D graphics
  - Raphael.js – Scalable Vector Graphics (SVG)

http://raphaeljs.com/tiger.html
WebGL, Three.js, Raphael.js

- Most often used as 3D library (on top of WebGL)
- CGLearn examples are done in Three.js
- Also: http://cgdemos.tume-maailm.pri.ee/
Virtual Reality

• Lot of high requirements for graphics:
  • Rotational accuracy < ¼ degrees
  • Translational accuracy < 1 mm
  • At least 90 FPS
  • Less than 20 ms latency
  • More than 1k resolution per eye
  • ...

Otherwise:
Virtual Reality

- We tried out a couple of demos on the Oculus Rift DK2:
  - Sightline
  - Windlands
  - Ats's Demo Game
Still confused?
You can...

- ... pick any topic from previous year
- ... pick some other CG related topic
World is a vast and mysterious place!

Ok, so I have a topic now...

- Look for materials
- Investigate, research
- Find examples
- Try it out yourself
- Present your findings
- Engage others
  - Discussion
  - Interactive demo
  - Workshop
What about the project?

- Interactive demo on the same topic as your seminar
What about the project?

- Advance something you've already done
What about the project?

- Can, of course, be a team / group effort!
What about the project?

- Do something fun and exciting

You can do it!
I didn't understand >70% of what you said...

• Don't worry about it!
• Pick a topic that suits your knowledge base
• Your topic may very well be:
  • Rasterization of triangles
  • Comparison of lighting models
  • How to do simple shadows?
  • Raytracing explained
  • etc
I don't even know where to start!

- There will be 2 more introductory lectures about the basics.

- Check out slides and exercises from Computer Graphics MTAT.03.015:
  https://courses.cs.ut.ee/2013/cg/fall
  https://courses.cs.ut.ee/2015/cg/spring

- Check out the slides from the previous seminar:
  https://courses.cs.ut.ee/2014/cg-seminar/spring/Main/Seminars
  https://courses.cs.ut.ee/2014/cg-sem/fall/Main/Seminars

- Find some online tutorial that seems manageable for you and try it out.
Questions?
List of some topics

1. **Color blending** – What happens when there are transparent objects in your scene?
2. **Lighting models** – What are the common models? Where and when are they used?
3. **Texturing** – How can one sample from a texture? What kinds of artefacts may appear?
4. **Curves** – Why are they important in CG? What about curved surfaces?
5. **Global illumination** – Pick one or compare different methods: Radiosity, path tracing, photon mapping.
6. **Realtime realistic rendering** – Provide an overview of the common methods or pick some effect (light, wetness, fog, fur / hair) and find out how it's rendered realistically in real time.
7. **Non-realistic rendering** – Where is it used and how is it achieved? Realtime vs prerendered?
8. **Tessellation** – How can this be done in OpenGL 4?
9. **Post-processing effects** – What effects are there? When and how are they used?
10. **Procedural generation** – Where and how is it used? How to apply procedural textures to procedurally generated meshes?