5 Major Challenges in Interactive Rendering

Johan Andersson
DICE
Overview

• What are the major challenges for us in the next 5-10 years?
  – Interactive rendering for games as well as other interactive areas

• Which problems do we want to solve?

• What do we want to achieve & focus on?

• Based on own thoughts & feedback from people in the industry
THE 5 CHALLENGES
(in no particular order)
Challenge #1

CINEMATIC IMAGE QUALITY
Challenge #1 - Cinematic image quality

• Want to get to that smooth visual feel CG movies have
  – Consumers are viewing them on same device as they play games on

• Visual realism of real-time is still far from offline CG
  – 33 ms vs ~60 minutes per frame

• Areas with big quality difference:
  – Aliasing
  – Motion blur & depth of field
  – Transparency
  – Geometry
Aliasing – we have it!
Aliasing

• Aliasing is one of the biggest visual artifacts
  – Most games have lots of it 😞
  – Eyes distracted by flickering
  – Aliasing within a frame is typically very variable

• Multiple current antialiasing techniques, but no complete solution yet
  – MSAA does not scale well (storage, bandwidth)
  – Post-effect based techniques only solve part of the aliasing problem

• How can we get to a pipeline that scales up to much higher quality AA?
  – Without breaking performance, memory storage or bandwidth
Everybody loves bokeh
Motion blur and Depth of Field

• Important visual cues to perceive depth, focus & motion!
  – The movie people know this

• Games only have post-process based versions
  – Lots of artifacts
  – Not possible to implement complete effect as post-process

• Ideal to have a rendering pipeline that can naturally support motion blur and depth of field
  – Stochastic rasterization?
  – Raytracing?
  – REYES?
  – Other?
Transparency

• Order-dependent transparency has always been a big limitation for content creators & developers
  – Restrictive art pipeline: no glass houses
  – Even windows on cars & buildings can be painful
  – Restrictive interaction between objects & effects
    • Meshes vs particles vs volumetrics

• Order-independent transparency is must going forward
  – Big challenge! Gradual process
Micropolygon rendering

• Massively detailed geometry
  – Render directly instead of simplify to normalmaps
  – True silhouettes & no faceted edges

• **DX11 tessellation** is a good step forward
  – But still quite complex full pipeline in practice
  – Shading quad efficiency issues with small triangles

• We’re getting closer!
Challenge #2

ILLUMINATION
Challenge #2 - Illumination

Global Illumination

Shadows

Reflections
Global Illumination
Global Illumination

- **Key visual component**
  - Build mood, ground environment
  - Current generation almost always static (or non-existing)

- **Need dynamic GI solution(s)**
  - Dynamic environments & for quick iteration times
  - Starting to see real-time dynamic solutions!
  - Multiple types of algorithms & levels of pre-computation

- **Interesting & difficult example use cases:**
  - Large-scale destructible environment
  - Single frame instant muzzle flash
Shadows

• **Shadowmaps** are still not a completely solved problem
  – Non-trivial implementations:
  – Aliasing, resolution-matching, filtering, management, culling
  – Translucent shadows

• Oh and where are the *penumbras*?
  – Area light source shadows are more pleasing
  – Variable penumbra & overlapping casters
Massive local shadowing

- We can light with 1000 light sources
  - Without shadows! 😞
  - Or only a few with shadows
  - Handled separately

- Next step: have 100s of lights with shadows
  - Requires rethinking and much improved culling & dispatch efficiency
Reflections
Reflections

• Currently only have 2 methods:
  – Envmaps
    • Scale up to 100s of dynamic envmaps? Similar to point light shadow problem
    • Doesn’t solve concave or large flat surfaces
  – Planar reflections
    • Good for big flat single/few surfaces
    • But restricted to that as well, how to handle slopes & multiple walls/windows?

• Need solutions for local reflections on arbitrary surfaces
  – Both glossy & perfect reflections
  – Not that much research in this area except with raytracing?

• Hybrid rasterization/raytracing pipeline?
Challenge #3

PROGRAMMABILITY
Programmability

• **Graphics pipeline** is fast but fixed
  – No conservative rasterization
  – No programmable blending
  – No flexible texture filtering (min/max/derivative)

• **Compute pipeline** can’t simulate full graphics pipeline efficiently today

• How do we get to a hybrid/reconfigurable pipeline?
  – What are the actual use cases & requirements?
Examples of what we like to solve

- **Irregular workloads** / user-mode scheduling
  - Key building block for many advanced techniques

- **GPU-based scene culling & rendering**
  - GPU feeding itself (on a high-level)
  - For performance and flexibility.

- **Half-res rendering without depth artifacts**
  - Depth test per sample, shade per quad, upsample to pixel

- **For more use cases:**
  - *Bending the Graphics Pipeline* at 11:45 am
Challenge #4

PRODUCTION COSTS
Production costs

• We are increasing quality & richness in all areas, but can’t continue to increase costs at same rate
  – Turn the trend of more & more expensive content creation!
  – Not as sexy, but single most important challenge in practice for many (game) developers

• Linked with iteration times which is critical for both quality, quantity & low costs
Importance of this challenge

• An example:
  – We’ve spent **20 man years** on improving workflows, iteration times & reducing production costs for our next game engine - Frostbite 2
Production cost reductions

• Improvements:
  – Faster workflows & tools
  – Procedural content amplification / generation
  – Fewer custom pipelines, techniques & solutions
  – More sharing & reuse of content

• Giant important topic but only something we cover indirectly in the course
Procedural foliage distribution
Challenge #5

SCALING UP
Scaling up

• GPU model has become quite flexible
  – Main problem is often not capabilities but performance/bw

• Want more of everything:
  – Performance, bandwidth & memory
  – Content, detail & quality

• Scalability without performance cliffs
  – Assumed reasonableness in fixed pipelines
  – Graceful performance degradation strongly preferred
Scaling up - Techniques

• Some techniques break down when scaling up, for example:
  – Deferred shading with +4x MSAA
  – Quad-based forward shading with ~1 pixel triangles
  – GPU dispatch APIs when doing 100s of small scene renders for shadows & reflections (batch counts)
Scaling up - Worlds

• Move from static worlds to immersive interactive worlds
  – Unique
  – Detailed
  – Changeable / Destructible
  – (Procedural)

• Massive scenes are a challenge for many industries
  – Movies, Scientific, Games
  – Esp. interaction with memory & performance cliffs
The Challenges

1. Illumination
2. Cinematic Image Quality
3. Programmability
4. Production costs
5. Scaling up
Conclusions

• Real-time rendering is far from a solved problem

• We need major improvements to the real-time rendering pipeline(s) and programming model
Thanks for all the feedback!

Christina Coffin (@christinacoffin)
Colin Barré-Brisebois (@ZigguratVertigo)
Daniel Collin (@daniel_collin)
Flavius Alecu (@flawe)
Sander van Rossen (@logicalerror)
Rob Jones (@bobvodka)
Stephen Tovey (@nonchaotic)
Colin Riley (@domipheus)
Joe Tidmarsh (@mrjovis)
Stephen Hill (@self_shadow)
Federico B P (@nocturndragon)
Stefan Boberg (@bionicbeagle)
Noel Llopis (@snappytouch)
Björn Knafla (@bjoernknafla)
Andrew Richards (@codeandrew)
Juan Manuel Alvarez (@the_naicigam)

David Luebke (@davedotluebke)
Jonathan Ragan-Kelley (@jrk)
Pat Wilson (@pat_wilson)
Rachel Blum (@groby)
Brian Karis (@briankaris)
Matt Collins (@matt_c_)
Sam Martin (@palgorithm)
Aaron Lefohn
Andrew Lauritzen
Luca Fascione
Steve Anichini
Simon Taylor
Matt Swoboda
Cody Ritchie
Mattias Kylen
Oscar Carlén

7/29/2010
Beyond Programmable Shading Course, ACM SIGGRAPH 2010
Contact details

email: repi@dice.se
blog: http://repi.se
twitter: @repi

For more DICE talks:
http://publications.dice.se
Beyond Programmable Shading Course
ACM SIGGRAPH 2010

BONUS
Surface lighting

• More advanced surface shading & lighting
  – Sub-surface scattering
  – Hair
  – Foliage
  – Arbitrary / generalized BRDFs?

• How can they work with deferred rendering?