Challenges in real-time rendering

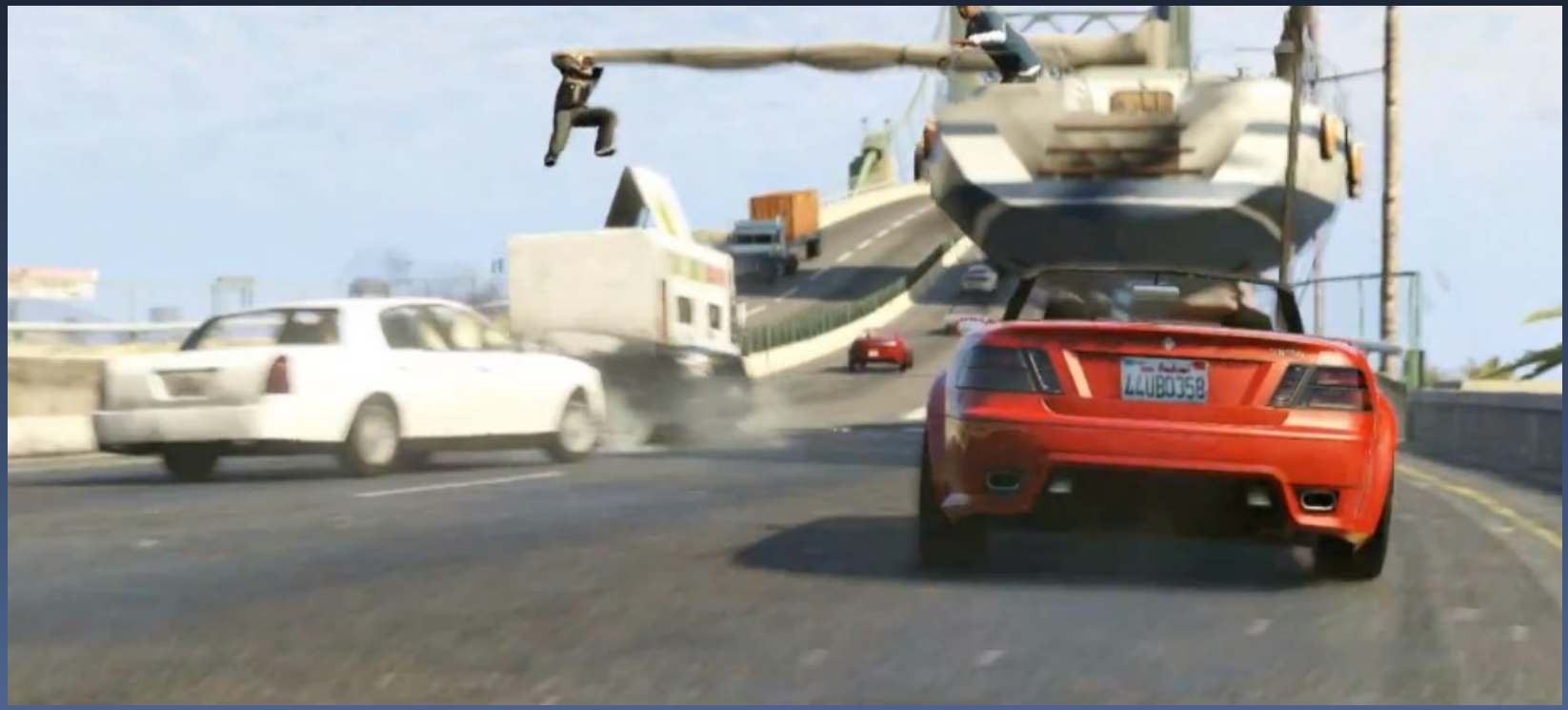
Overview

- Which problems should be solved?
- What is the aim?(on which problems are we focussing and what do we want to achieve)

The 5 challanges

- 1. Cinematic Image Quality
- 2. Illumination
- 3. Programmability
- 4. Costs
- 5. Scaling

1. Cinematic image quality



Cinematic image quality

- The goal is to achieve cinematic image quality
- Same smooths and rich pictures that computer graphics movies have
- Need improvements to GPU primary visibility
  - Antialiasing
  - Transparency
  - Defocus blur
  - Motion blur

Antialiasing

- Single most visible issue to improve on
  - Aliasing breaks the ilusion
  - Less aliasing: more pleasing and easier to see visuals
- Sources of aliasing
  - Geometric aliasing
  - Proxy geometry
  - Shader aliasing

Geometric aliasing

- Different solutions: MSAA, SSAA

(http://en.wikipedia.org/wiki/Multisample_anti-aliasing)

  - Fixed quality techniques, not adaptive
  - Problematic to scale up to very high quality
- 16x MSAA is good quality but expensive
  - Need higher rate if using coverage masks
- MSAA + deferred

Other alternatives

- Analytical antialiasing
- Pre-filtered Sparse Voxel Octrees
- Requires high resolution/large storage
- http://www.youtube.com/watch?v=lA1y_VPjeiY

Shader aliasing

- Shader aliasing becoming more problem
  - High-frequency specular highlights
  - High-frequency shadows
  - Amplified by HDR Bloom and Bokeh
  - (http://www.youtube.com/watch?v=jYAv5u6eQ5s)
- What is needed to make sure that shaders do not output aliased values?
  - Careful handling of derivatives when texture

Motion blur

- Important for sense of speed and direction
- Velocity vectors + post-process holds up quite well

Defocus blur

- Key visual cue to perceive depth and focus
  - Guide & emotional storytelling tool
- Sprite splatting is popular
  - Works great for out of focus background
  - Very sensitive to aliasing
  - Sharp edges on strong foreground blur

Illumination

- Challanges
  - Dynamic Global Illumination
  - Shadows
  - Reflections

Dynamic Global Illumination

- http://www.youtube.com/watch?v=nhQc_wo4-oM
- Key visual component
- Dynamic alternatives
  - Light Propagation Volume
  - Voxel cone tracing
  - Reflective Shadow Maps + VPLs
  - Geometry pre-compute based: Enlighten
- Major trade-offs depending on performance/memory/quality

The Many Shadow problem

- Want shadows of all lights
  - Easier to author
  - Doesn't limit content creators
  - Higher quality & more interactive
- Solutions
  - Efficient rasterization
  - Raytrace geometry
  - Cone trace into SVO

Reflections – categories

- Glossy reflections on arbitrary surfaces
- Perfect reflections on mostly-planar surfaces

Glossy reflections

- Most surfaces, rough metal
- Screen-space reflection
- Voxel Cone tracing

Perfect reflections

- Mostly planar surfaces: windows, water
- Render reflected view
- Raytracing
- Voxel Cone tracing

Programmability

- Graphics pipeline
  - No conservative rasterization
  - No programmable blending
  - No flexible texture filtering
- Gpu Compute
  - Use the graphics pipeline when possible
  - Need to enable building your own efficient GPU Compute pipelines

Costs

- Games/programs are getting bigger and more complex
  - More content
  - More variation
  - Higher quality/detail
  - More complex content production process

Costs

- If we had the ultimate real-time renderer that solves primary visibility and Illumination, how much artist time would we save?
  - Probably not much becuase the content creation is the biggest time sink
- What can save significant amount of time?
  - Scalable geometry representation
  - Procedural texturing
  - Procedural geometry
  - Content acquisition

Scaling

- Games and rendering use cases are needing more and more scaling. Both up and down!
  - Detail: mm to km
  - Resolution: 320x480 to 5760x1200(eyefinity)
  - Power: 1W to 300W
- Requires significant scaling in performance

Scaling: Detail

- How can we increase detail while building even larger interactive worlds?
  - Scalable geometry is difficult, discrete LODs are hard to handle
  - Can't author everything

Scaling: Resolution

- Some of the lowest powered devices have the highest resolution screens
  - Consumers->Happy
  - Developers->Unhappy
- Graphics pipeline need a more flexible decoupling of shading rate vs visibility rate!

Scaling: Power

- Marketplace is shifting from 100+ W to 1-45 W
- Developers typically don't care about power usage
- Need power efficient algorithms, techniques & pipelines

- Thank you!