Considerations for a Distributed 3D Virtual Environment

G. Jamie Cope
Senior Research Engineer
Val G. Hemming Simulation Center

Evolution of the WAVE

- Challenges
  - Create a software backbone
  - Create a content pipeline
  - Provide a dynamic user experience
  - Improve visual quality

Challenge: Create a software backbone

- Key pieces
  - Cluster synchronization
  - Rendering
  - Audio

WAVE

?

Challenge: Create a software backbone

- The goal:

Challenge: Create a software backbone

Cluster Sync	Complete	Extensible	Low-Latency	Scalable	Support
Hardware	✗	✗	✓	-	-
Chromium	✗	-	✓	✓	✓
Flatland	✓	✓	-	✗	✗
VR Juggler	✓	✓	-	-	✓

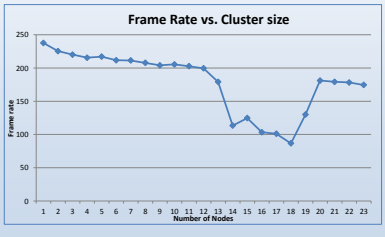
- Highly configurable and platform independent
- Network-driven software sync
  - Sync barriers for input, update, and render (buffer swap)
- Established and active user base

WAVE

VR Juggler

Challenge: Create a software backbone

- Testing: VR Juggler
  - Latency and scalability
    - Designed for clustering, but can it handle ~50 nodes?



Challenge: Create a software backbone

Rendering	Complete	Extensible	Low-Latency	Scalable	Support
Crysis/Unreal	✓	×	✓	×	✓
Ogre	✓	✓	✓	-	✓
OpenSG	×	✓	✓	✓	-
OpenSceneGraph	✓	✓	✓	✓	✓

- Modular code with callbacks and solid foundation classes
- Plugins for particles and animation
- Efficient state sharing & graph optimization
- Dynamic culling

Challenge: Create a software backbone

- Testing: OSG
  - Speed and quality

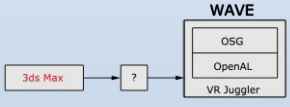
Mass Casualty Scene
- 1.8 Million faces
- 37 animated bipeds
- Animated debris
- Better than 60 FPS!

Challenge: Create a software backbone

- Audio
  - OpenAL Toolkit
  - Spatialized sound rendering
  - Simple scene graph integration

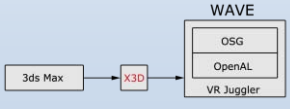
Challenge: Create a content pipeline

- Requirements
  - From design to render
  - Reliable and easy to use
  - Solid yet extensible
- Starting point: 3ds Max (www.autodesk.com)
  - Industry standard rendering package
  - Already used in-house



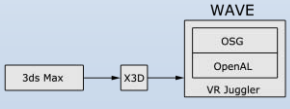
Challenge: Create a content pipeline

- X3D file format
  - eXtensible 3D! – Everything we need, plus room to grow
  - Custom 3ds Max X3D export plugin
  - Synchronization script to push files to cluster
  - Custom OSG import plugin



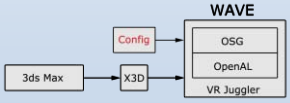
Challenge: A dynamic experience

- Scripted scenarios become predictable and easily ignored.
- Much of the impact on training is lost.
- Solution
  - Triggerable events
  - Randomization
  - On-demand scene changes



Challenge: A dynamic experience

- Triggerable events
  - Events: animations, sounds, physical
  - Triggers: keyboard, tracking, on load, other events
  - Configuration file (XML)
    - Set up events (sound parent, animation loop, etc.)
    - Tie events to triggers
    - Unit animations
      - Supported through 3ds Max note tracks
    - Scene properties



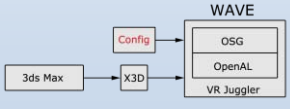
Challenge: A dynamic experience

- Triggering events

Challenge: A dynamic experience

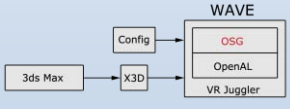
- Randomization
  - Event start, event frequency
- On-demand scene changes

  - Pre-load scene graph per-scenario
  - Reset environment on switch
  - Lesson Learned: Flush OpenGL memory!



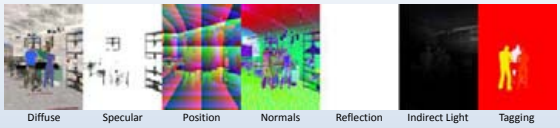
Challenge: Improve visual quality

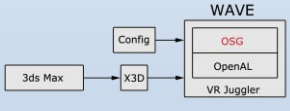
- Requirement: Maintain real-time frame rate
  - Transition from forward rendering to deferred
    - Geometry pass stores object info to buffer
    - Single per-pixel lighting pass decouples lighting from scene complexity.



Challenge: Improve visual quality

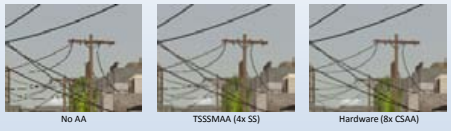
- Requirement: Maintain real-time frame rate
  - Deferred buffers:

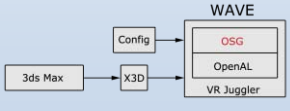




Challenge: Improve visual quality

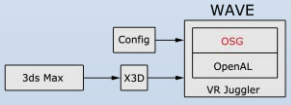
- Requirement: Maintain real-time frame rate
  - Transition from forward rendering to deferred
    - Challenge: Anti-aliasing
      - Several solutions – SMAA, FXAA, TSSSMAA





Challenge: Improve visual quality

- Requirement: Maintain real-time frame rate
  - Expand and refine use of hardware shaders
    - Use shaders for character animation and particles

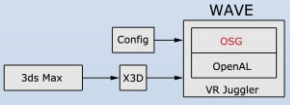



Challenge: Improve visual quality

- Requirement: Maintain real-time frame rate
  - Expand and refine use of hardware shaders
    - Create dynamic shader compositing system

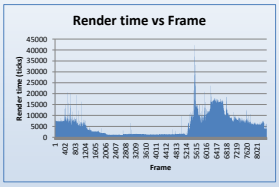

Diffuse + Specular + Normal

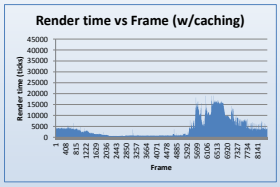

Diffuse + Opacity



Challenge: Improve visual quality

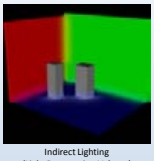
- Requirement: Maintain real-time frame rate
  - Lesson learned: Keep textures resident!

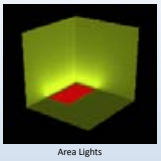




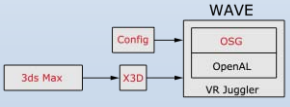
Challenge: Improve visual quality

- Requirement: 3ds Max == WAVE renderer
  - Support more features


Indirect Lighting
(Light Propagation Volume)

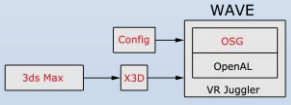

Area Lights


Real-Time Shadows



Challenge: Improve visual quality

- Requirement: 3ds Max == WAVE renderer
  - Support multiple development paths
  - Frequent artist/engineer collaboration
    - Custom 3ds Max shader prompts new feature integration and manages expectations



Lessons Learned

- Leverage the tools available…
  - … but stay flexible
- Keep track of graphics memory
  - Manage what's going in and out
- Close engineer/artist collaboration

Future work

- Remove dead nodes from cluster at run-time
- Make use of additional CPU cores
- Make use of additional GPUs with direct compute (CUDA)
- Improve character rendering (skin, hair)
- Implement cascaded versions of shadows and indirect lighting

Appendix A: Toolkits

- VR Juggler (www.vrjuggler.org)
  - Open source cluster-driven VR toolkit
  - Started in 1997 by Dr. Carolina Cruz-Neira and a team of students at Iowa State University's Virtual Reality Applications Center.
  - Currently developed and maintained by Priority 5 Holdings, llc.
- OpenSceneGraph (OSG) (www.openscenegraph.org)
  - OpenGL 3D graphics toolkit
  - Open source
  - Started in 1999 by Don Burns and Robert Osfield
- OpenAL
  - Open source audio toolkit
  - Started in 2000 by Loki Software
  - Currently developed and maintained by Creative Technology