Topics on VR

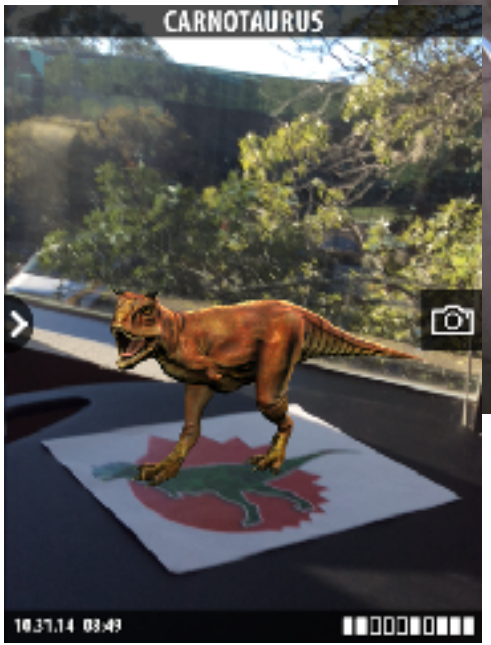
VR Today

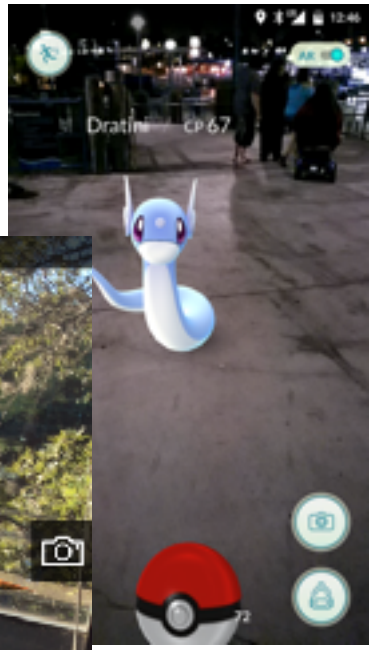
What is VR?

unity

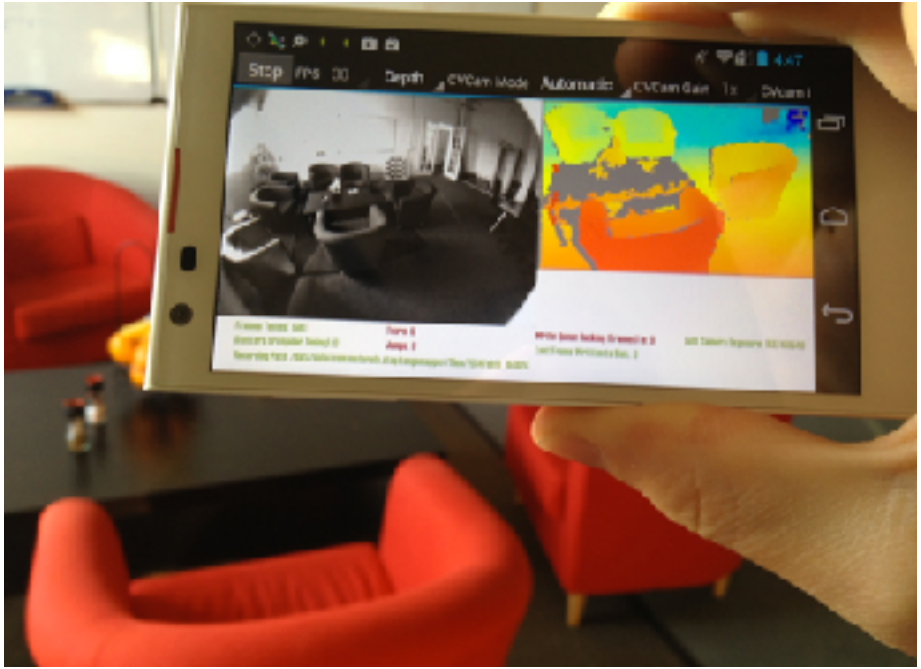
Augmented Reality

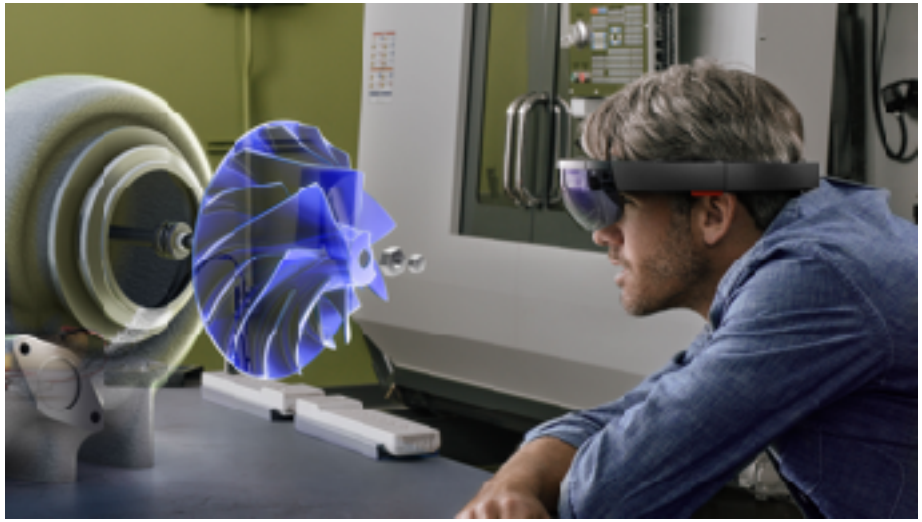
unity





Mixed Reality





unity

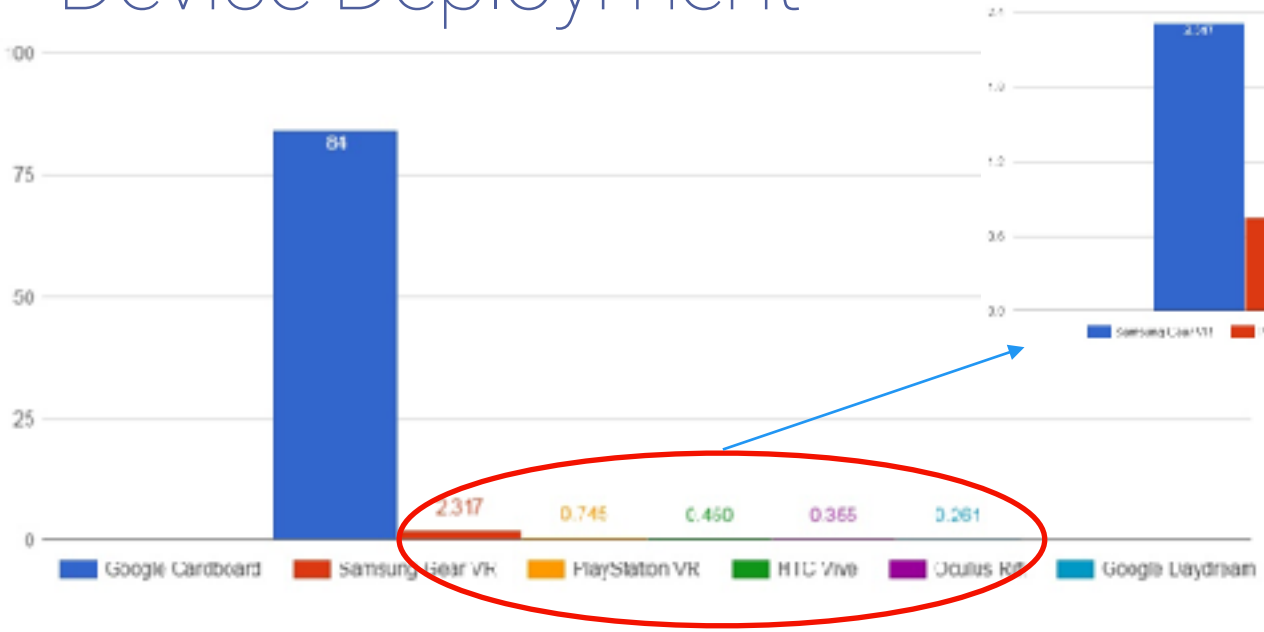
360° Images & Video: The Jury's Still Out

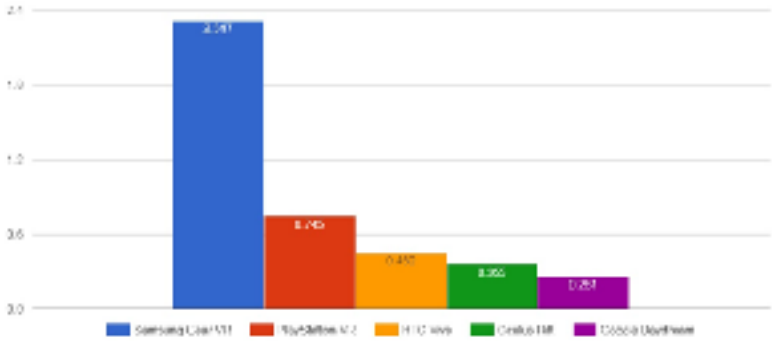
- Capture a scene using an array of cameras
  - either stationary or moving
- Rendered images
  - *omni-directional Stereo* (ODS)
  - *equirectangular* projection

theta360.com

unity

Device Deployment





unity

Rendering for VR

Frame Rate

- Video games target 60 Hz refresh
  - single view, very high resolution (UHD / 4K)
  - 16.6 ms/frame
- VR is different:
  - two eyes
  - high resolution (HD / 1080)
  - need time for compositing
- We'd like 120 Hz (8.3 ms/frame)
  - settle for 90 Hz (11 ms/frame)

60 Hertz
30 Hurts

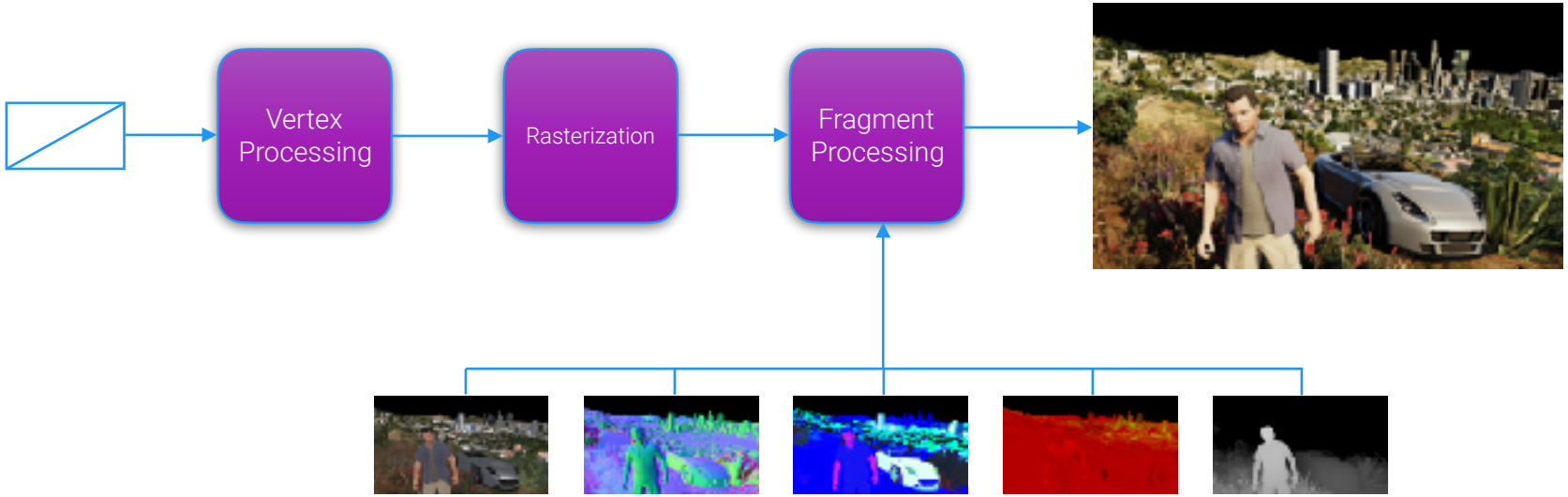
Deferred Rendering

http://www.adriancourreges.com/blog/2015/11/02/gta-v-graphics-study/

unity

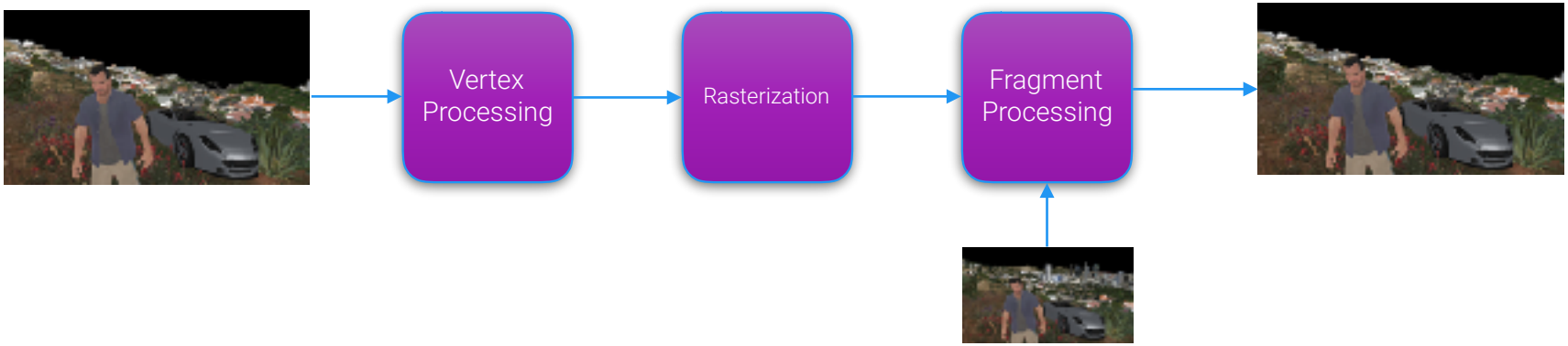
Deferred Rendering (Pass 2)

Final Image Generation



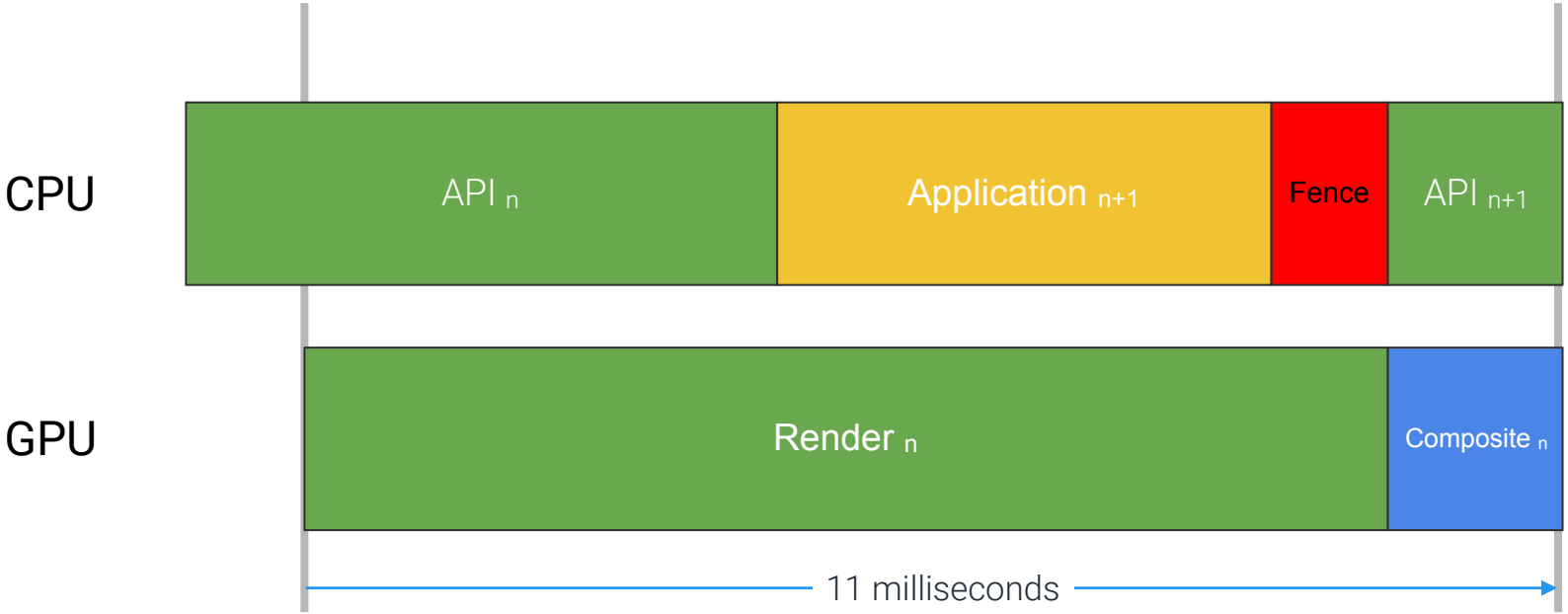
unity

Forward Rendering



unity

Frame Phases (@ 90Hz)



unity

Generating Stereo Pairs

- Image for each eye generated separately
  - same frustum separated by *inter-pupillary distance* (IPD)
- API extensions to "optimize" rendering
  - application provides an array of viewing transforms
  - driver executes each draw call twice, updating index into viewing transforms

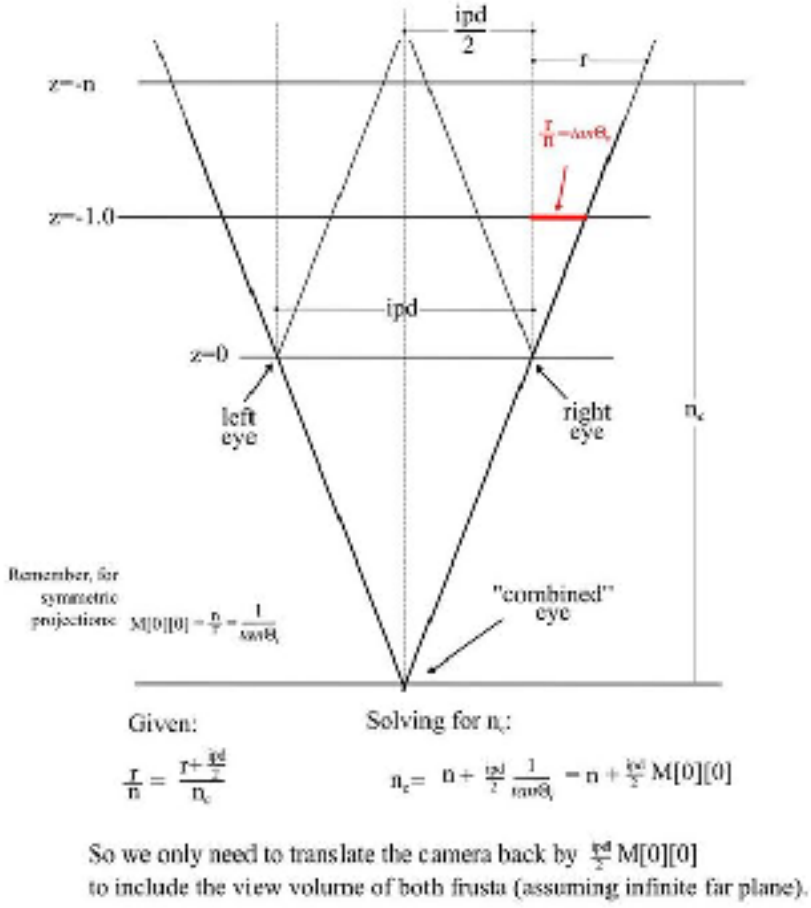
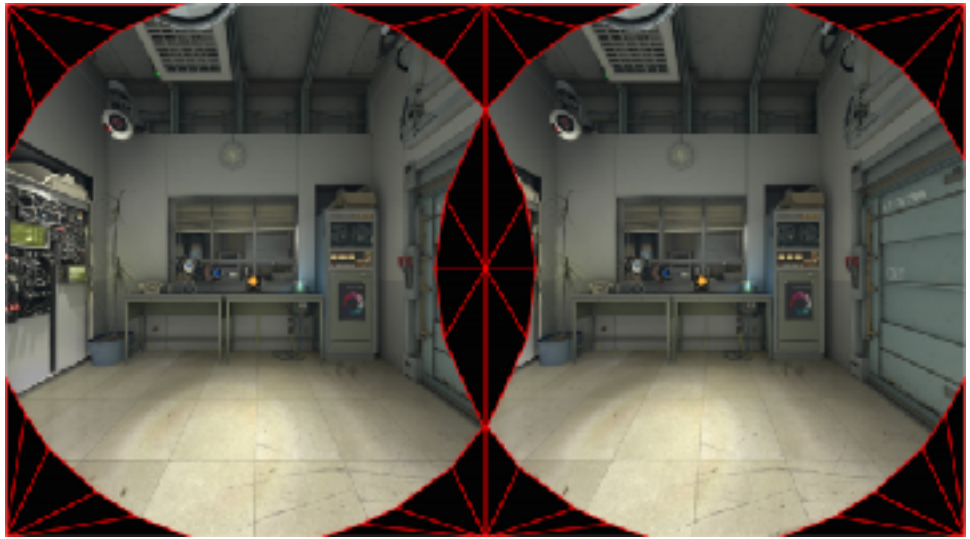


Image by Cass Everitt / Oculus

Compositing

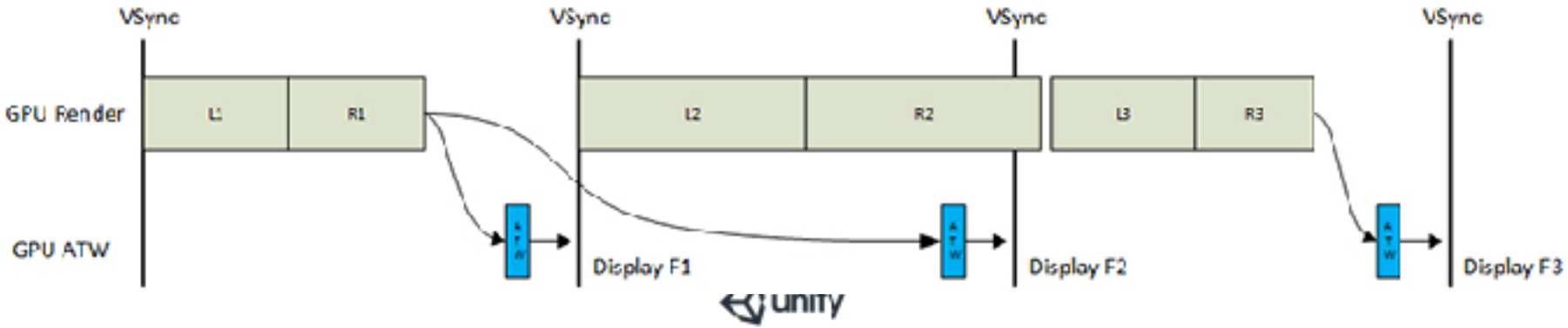
- Device drivers include a compositing facility
- Warp each generated eye texture to match device optics



unity

Async Time Warp

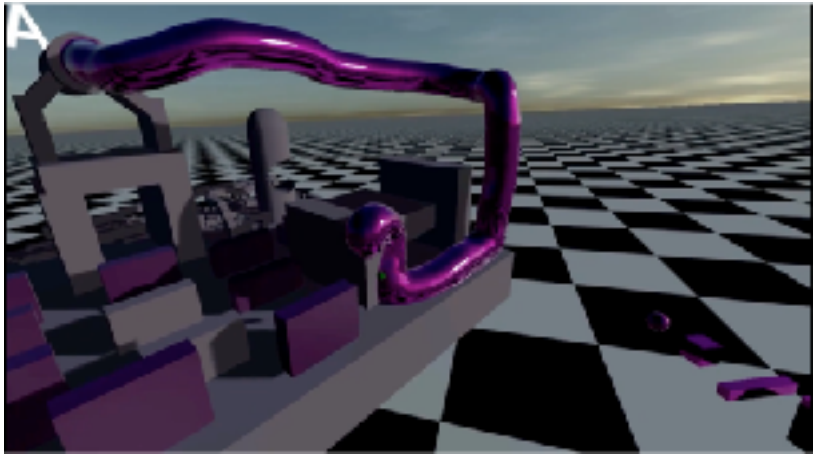
- Oculus technology for dealing with frame overruns
- Previous frame warped using updated pose information by compositor
- Helps prevent judder
- Requires GPU preemption



Triangles vs. SDFs

- Triangles aren't the (perhaps) the best solution in many cases
- *Signed Distance Fields* (SDFs) represent models as implicit surfaces
- rendered using *ray marching*
  - either rendered entirely in a compute shader
  - or full-screen quad and a helluva fragment shader

shadertoy.com



unbound.io

unity

Input, Gestures, & Haptics

Controlling the View: DOFs

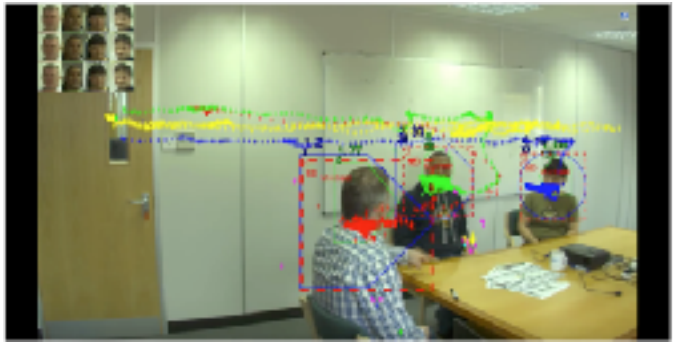
- Mice & keyboards are easy
  - 2D screen coordinates
  - binary mouse buttons & keyboard events
- Controllers provide much more data
  - *degrees of freedom* (DOFs)
  - 3-DOF
  - 6-DOF

unity

Events vs. Gestures

- Events are easy to put together
  - usually represented as some sort of state machine
  - clear state transitions based on discrete events
- Gestures can be more difficult
  - filter input event stream
  - constrain DOFs

Pose, Gaze, & Eye Tracking



- Computer vision and deep learning techniques can identify and track objects in real time
- Tobii - commercial eye-tracking
  - integrated into several laptops
  - similar technology being built into HMDs
- Apical - real-time computer vision IP
  - identifies and tracks objects in real-time
  - returns location and pose data

Foveated Rendering (Region of Interest)

- Your eye has more resolution in its central vision area (fovea)
- render more accurately in gaze direction
  - lower levels-of-detail or resolution outside gaze cone
- Extensions or hacks required on desktop
  - amazingly simple on mobile (tiled renderers)

Current Challenges

Wires vs. Latency

- Tethered headsets are a drag
- Wireless headsets are coming
  - unclear about constraints of latency and interference
- Pack entire system into a backpack
  - The Void

unity

Cloud vs. On-Device

- Battery-powered devices still have a ways to go with rendering performance
  - they can do pretty much everything a discrete GPU, just perhaps not as fast
- Can we render in the cloud and send to a device?
  - latency issues
    - pose information up
    - rendered framed down
  - image delivery
    - encode as H.265 (supports constant 90 Hz video encode / decode)

Thank you


unity