

Measuring QoS in Web-Based Virtual Worlds: an Evaluation of Unity 3D Web Builds

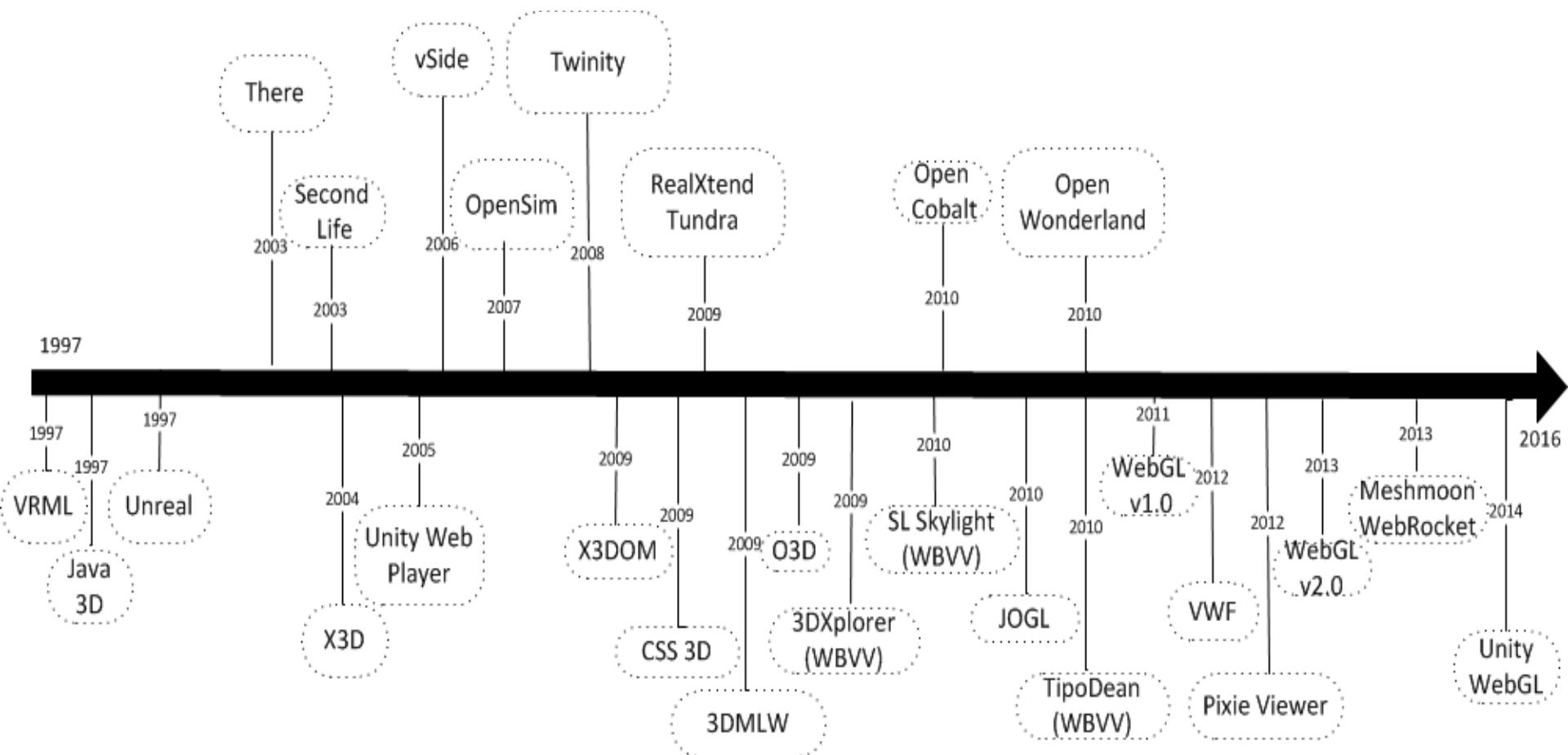
Hussein Bakri and Colin Allison
University of St Andrews

openvirtualworlds.org

multi-user and web-based virtual worlds

- traditional multi-user virtual worlds e.g. Second Life, OpenSim, Open Wonderland, have enabled engaging interactive environments
 - immersive learning and cultural heritage
- but: a dedicated stand-alone client/viewer needed, which can be a barrier to accessibility, adoption or use
- regular (2D) web is becoming more suitable for hosting interactive 3D content -> the emerging 3D web?
- can browsers replace MUVW viewers?
- we compare two web-oriented virtual world 3D builds
 - Unity Web Player
 - Unity WebGL

Multi-User Virtual Worlds and the 3D Web



Timespan Longhouse (openvirtualworlds.org)

- OpenSim
 - GRID URI <http://apollo.cs.st-andrews.ac.uk:8002/>
 - Login page <http://openvirtualworlds.org/splash>
- WebGL
 - <http://openvirtualworlds.org/unity-static/webgl-timespan-longhouse1/>
- Unity Web Player
 - <http://openvirtualworlds.org/unity-static/Timespan-Longhouse2-web/Timespan-Longhouse2-web.html>

Timespan Longhouse, Unity Web Player (Firefox plug-in)

ity Web Player | St Rules



virtualworlds.org/unity-static/Timespan-Longhouse2-web/Timespan-Longhouse2-web.html



Search



Unity Web Player | St Rules



Timespan Longhouse, Unity Web Player (Firefox plug-in)



Unity WebGL (Firefox, no plug-in)



Unity WebGL (Firefox, no plug-in)



Longhouse Inside



A Still Inside a Croft



QoS and QoE metrics

Quality of Experience

- Frames per Second (FPS) and Frame Time (FT)
 - In an OpenSim viewer each frame should complete in approximately 18.18ms (55 frames per second). “If total frame time is greater than this then simulator performance will be degraded”
 - Similarly, in Unity3D web applications, 60 frames per second is recommended
- Initial Download Time (IDT)
 - Virtual World equivalent of (2D) Web Page Load Time (PLT)

Browser performance in terms of CPU and Graphics processors

- CPU Load and physical memory usage
- GPU Load and GPU memory usage

Methodology

- Compare two builds for the same virtual world
 - Unity Web Player (UWP) – requires plug-in
 - Unity WebGL – no plug-in needed for most up-to-date versions of popular browsers
- Avatar mobility models
 - Standing: Avatar remains standing still with continuous yawing for 2 minutes (Yaw is the change in avatar orientation or the change in the “look at” view).
 - Random Walking: Avatar randomly walks for 3 minutes in different directions (from non-dense to dense areas) and with a constant speed
- Client side measurements
 - Client connected to server on campus where minimum bandwidth is 100 Mb/s.

Testbed

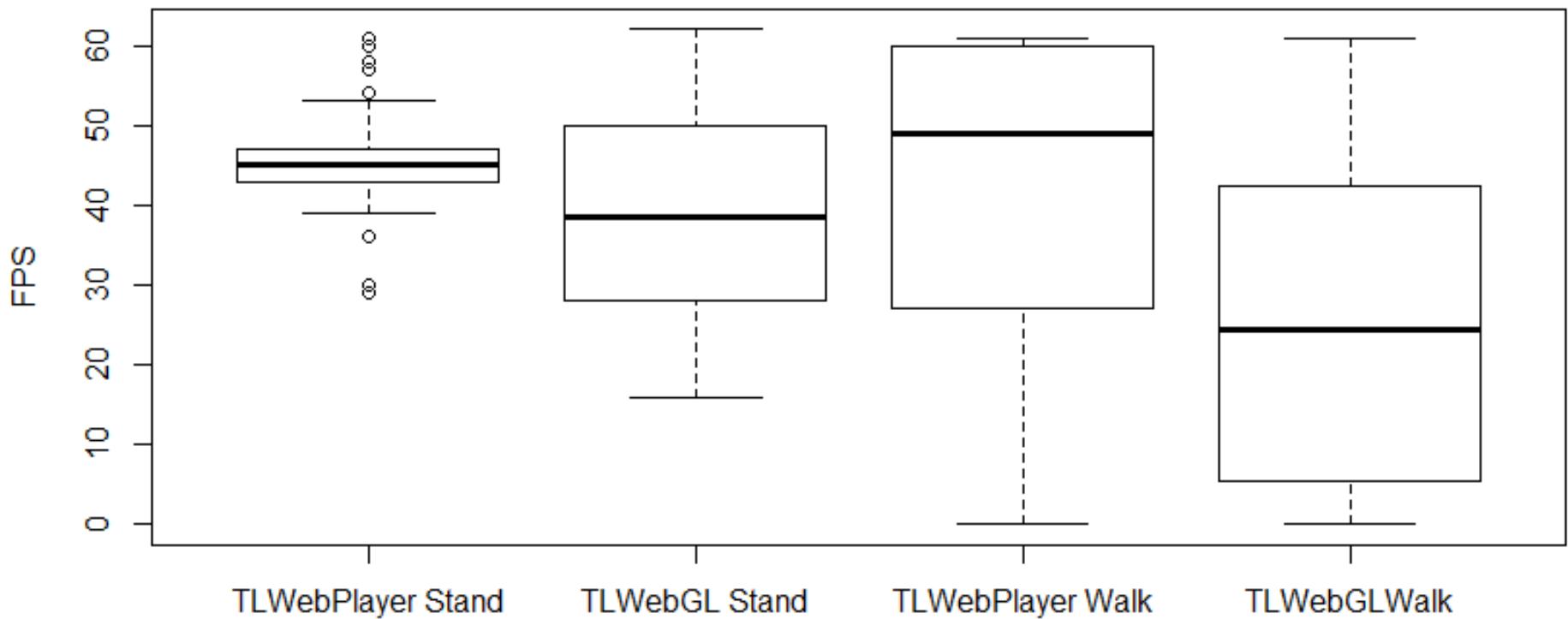
- Virtual World: Timespan Longhouse
- Unity 3D engine version 5.2.0f3
- Client specification
 - Intel Core i5-440- 3.10 GHz with 16GB 1067 MHz DDR3 RAM
 - NVIDIA GeForce GTX 970, 4GB Video RAM
 - Fresh installation of Windows 7 Enterprise 64 Bit Operating System with a minimal set of background processes running to avoid interference.
 - Builds generated by Unity 3D engine version 5.2.0f3
- Software tools
 - Chrome, Firefox, FRAPS, HWiINFO64, TechPowerUp GPU-Z, Powershell (for automation), Opera, Ccleaner,....

Experiments

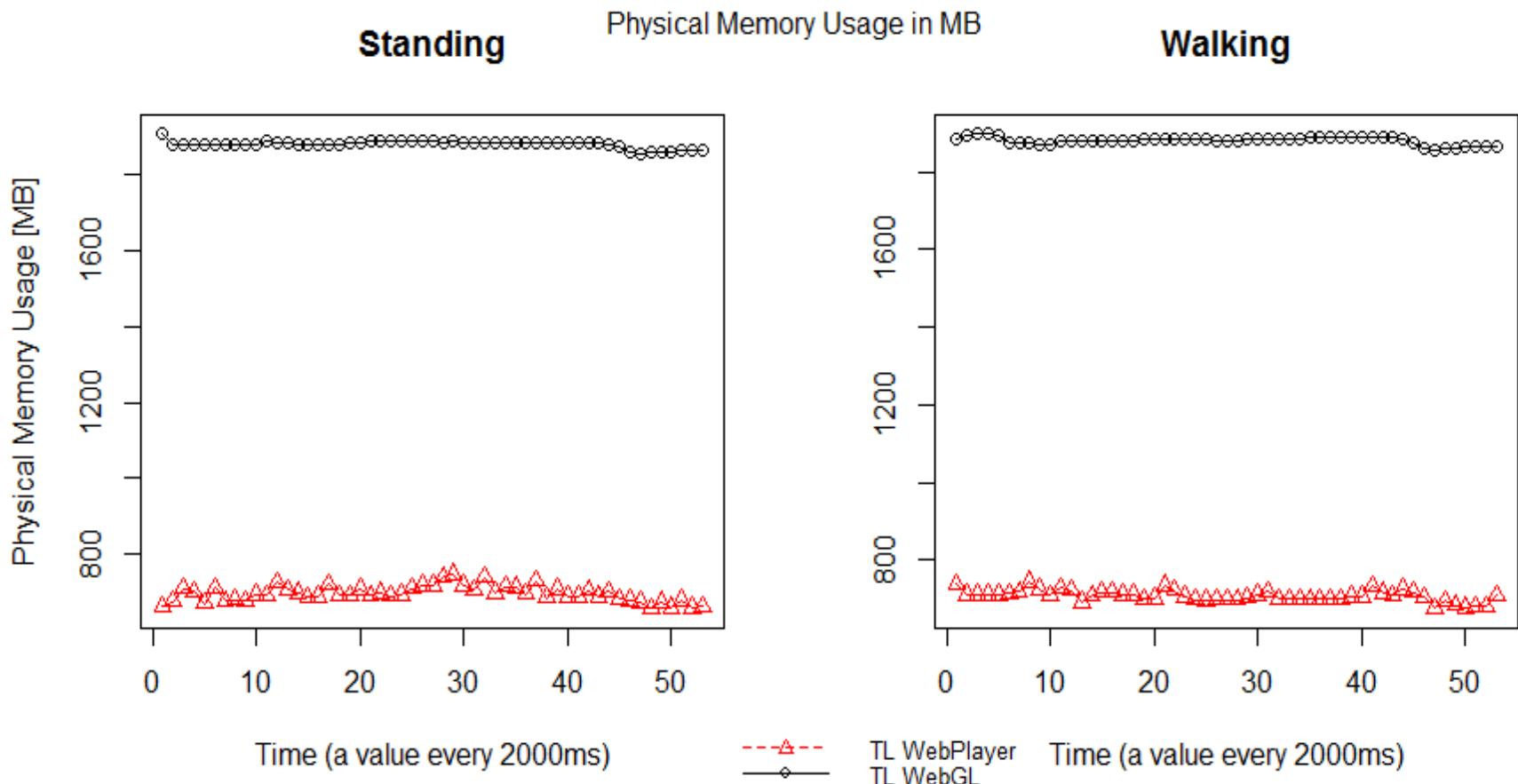
- **Experiment 1**
 - Google Chrome version 44.0.2403.125 and Fraps used to measure FPS & FT
- **Experiment 2**
 - HWiNFO64 and TechPowerUp GPU-Z used to measure:
 - Physical Memory used (MB),
 - Physical Memory load (%),
 - Total CPU usage (%),
 - GPU core load (%),
 - GPU Memory usage (%)
 - GPU D3D Memory usage (%).
- **Experiment 3**
 - *Initial Download Time* measured using a range of browsers, cache cleaning techniques and network monitoring tools
 - results from different sources were in complete agreement

frames per second

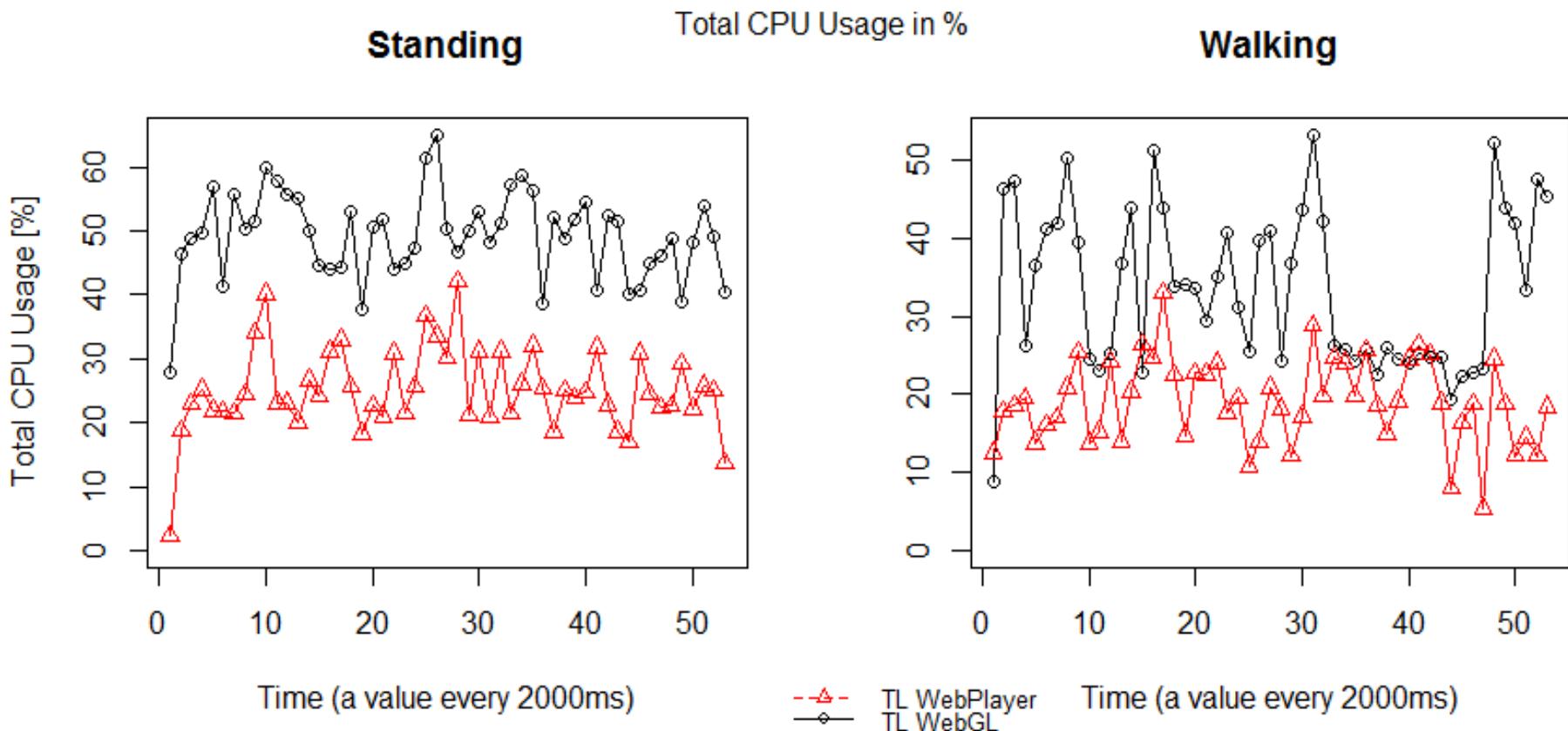
Distributions of FPS



physical memory usage

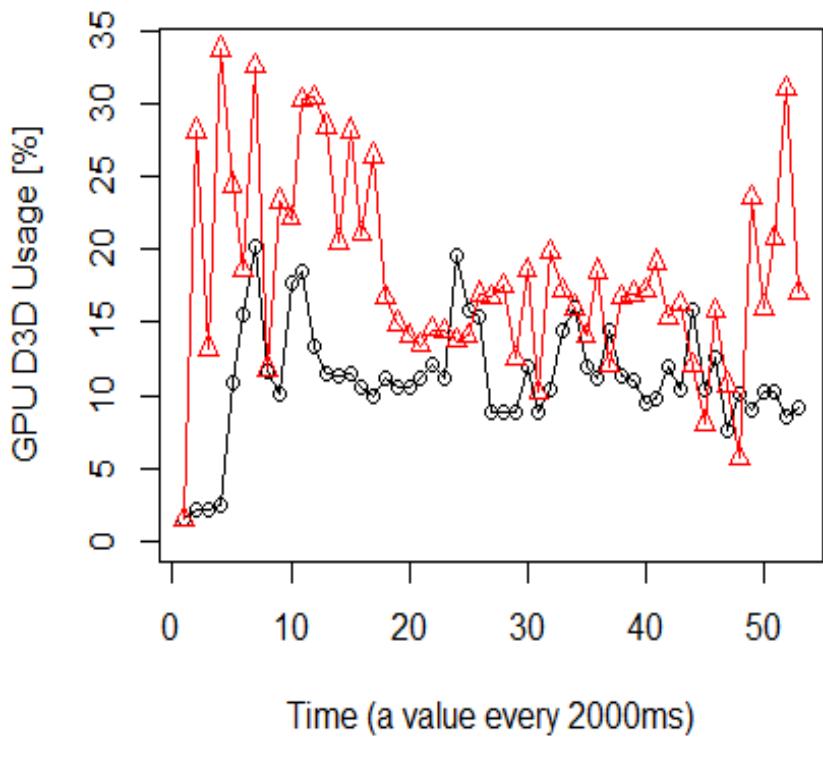


CPU Usage



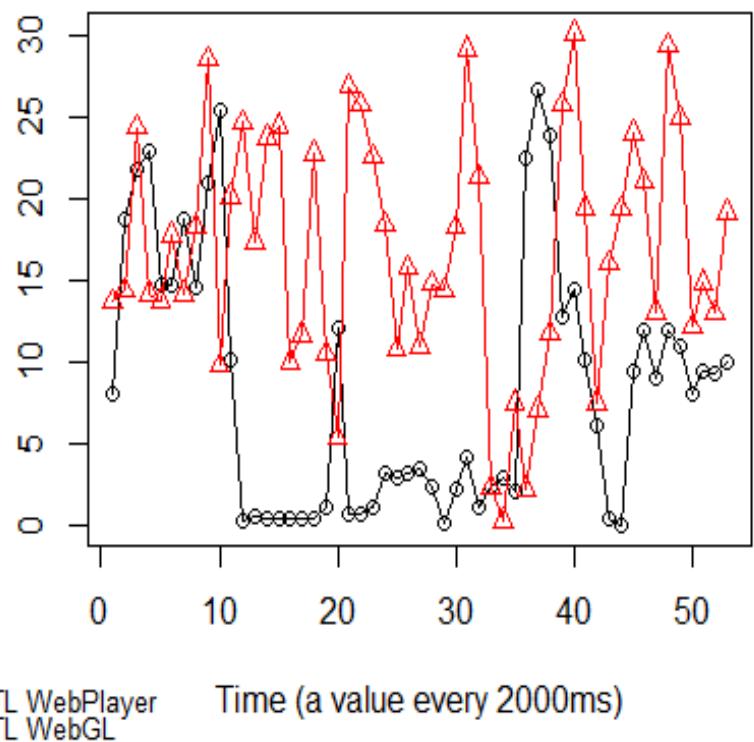
GPU D3D Usage

Standing

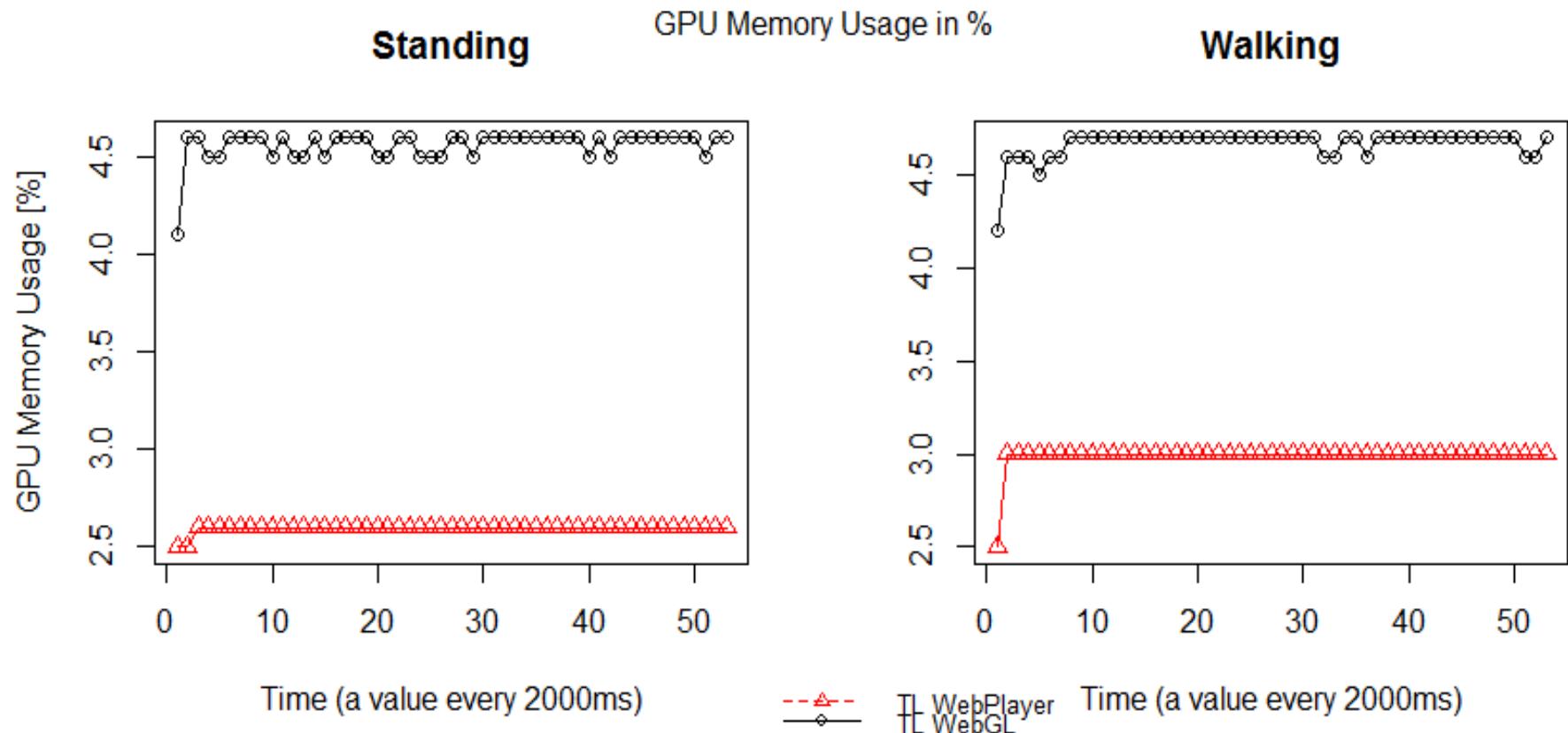


GPU D3D Usage in %

Walking



GPU Memory Usage



Initial Download Time (IDT)

Average RTT to Server	sizes of raw Files when downloaded (ftp, http)	Average downlink throughput
0.271 ms	40 MB (UWP file)	11.2 MB/s
	129MB (U-WebGL .data file)	10.4 MB/s

	Average IDT	Standard Deviation
UWP	3934.33 ms	91.258
U-WebGL	20191.67 ms	581.461

Limitations of study

- Only one virtual world used (Timespan Longhouse)
- Other worlds may have different characteristics
- TL translated from OpenSim, so may be less than optimal
- WebGL not analyzed in detail for possibility of optimization

Postscript: RIP Unity Web Player (!)

- Although Unity Web Player format significantly outperformed the WebGL builds on almost every metric it is no longer supported in Unity 5.4 (March 2016)
- Why?
 - Chrome has dropped support for NPAPI plug-ins (v.45, Oct. 2015)
 - Edge (IE replacement in Windows 10) does not support UWP
 - Mozilla will also phase out support for plug-ins
 - General trend towards plugin-less browsers?
- Will the new versions of Unity generate higher quality WebGL?
 - “it is important to understand that WebGL is a different platform from the Web Player and does not match the feature set or performance of the Web Player. We are working closely with browser vendors to make sure this gap becomes as narrow as possible, but....” Unity, October 2015

QoS in Web-based virtual worlds: Summary

- Signs of convergence between traditional MUVWs and the 3D Web
- Desirable to deliver MUVW via standard browsers
 - With plug-in, or...
 - Even better, without plug-in (direct support for WebGL)
- Compared performance of two mobility models in two different Unity builds for browsers:
 - Web Player
 - WebGL
- Web Player build significantly outperformed WebGL build
- Unity has recently dropped UWP – will their WebGL builds improve?
 - We can check using the data sets and testbed!

Measuring QoS in Web-Based Virtual Worlds: an Evaluation of Unity 3D Web Builds

Questions?

openvirtualworlds.org