Extending the Virtual World Framework for Mobile Training

Howard Mall
Engineering and Computer Simulations
Orlando, FL
howardmall@ecsorl.com

Ronald Punako, Jr.
Concurrent Technologies Corporation
Johnstown, PA
punakor@ctc.com

ABSTRACT

The Virtual World Framework (VWF) is a browser-based technology suite created by the Office of the Secretary of Defense for Personnel and Readiness (OSD/PR). The current prototype supports efforts to create collaborative, 3-dimensional spaces leveraging new web standards such as HTML5, Web Sockets, and WebGL that work in browsers without the installation of plugins or other software. As stated in 2011 (Boyd,Smith), "as the full VWF ecosystem emerges, an open and accessible architecture approach will allow users and third parties to add customized content, behaviors, and technologies to create new applications and create a virtuous cycle of value directly benefiting all users of the system." At this nascent stage, there have been few operational demonstrations created that tests the VWF's capabilities and showcases its possibilities for instruction.

Joint Knowledge Online (JKO) has a very specific mission to educate the Joint Warfighter, and took on the challenge of piloting a VWF application that would support their training objectives for their required Combating Trafficking in Persons (CTIP) course. Additionally, JKO wanted to push the VWF envelope to augment the Warfighter’s mobile training initiative. This paper will describe the efforts taken by CTC and ECS, Inc., to create this pilot for JKO.

We will describe several technical approaches to delivering immersive world content within the VWF to mobile platforms (namely iOS and Android). We will describe the challenges, successes, and methodologies developed to deliver this capability for mobile browsers. We will include performance analyses, descriptions of experiments conducted employing 3D, 2D, and 2.5 D techniques, an explanation of how the final product was built and delivered as a fielded product for JKO, along with future work we see in this area.

ABOUT THE AUTHORS

Howard Mall Howard Mall is Vice President of Engineering at Engineering and Computer Simulations, Inc. He has spent the last nine years at ECS building various kinds of training systems. He has lead efforts for the Navy to develop training solutions deployed on cell phones and hand-held computers. For the Army, he delivered the Tactical Combat Casualty Care (TC3) Simulation used by combat medics to learn triage and medical decision-making on a virtual battlefield. He has led several virtual world and game-based simulation projects for a myriad of federal agencies and commercial concerns. He currently oversees multiple engineering efforts at ECS both mainstream and on the fringe of the state-of-the-art.

Ronald Punako, Jr. Ron Punako Jr. is a Senior Software Engineer developing advanced learning technologies and operations simulations for the DoD, state and federal governments. He is developer of four game-based learning titles, two training exercise frameworks and a collaborative multi-intelligence geospatial simulation. He is a developer of seven content management systems and is an experienced software usability and accessibility design practitioner. Ron is an active speaker at education and training conferences where he presents topics such as the game development process, avatar kinesics and experiential learning game design. With 15 years of experience in the IT industry, he has worked for CTC supporting education and training for ten years. Ron holds a Master's Degree in Computer Information Systems from Nova Southeastern University and is currently a student in the Ph.D. Computing Technology in Education program at Nova.

Extending the Virtual World Framework for Mobile Training

Howard Mall
Engineering and Computer Simulations
Orlando, FL
howardmall@ecsorl.com

Ronald Punako, Jr.
Concurrent Technologies Corporation
Johnstown, PA
punakor@ctc.com

INTRODUCTION

This paper will discuss research & development (R&D) in pursuit of prototypes designed to demonstrate usages of the Virtual World Framework (VWF) for mobile training. The goals of the prototypes were to help inform of what technological, content and methodological avenues were available in pursuit of mobile training using native VWF technology, content and methods as a priority and then by extending technical and methodical scope, as necessary.

The prototype work was conducted for the Joint Staff DDJ7 Joint & Coalition Warfighting (JCW) Support Team with a period of performance November 2012 to March 2013.

Definitions

Experiential Learning – The process whereby knowledge is created through the transformation of experience. Knowledge results from the combination of grasping and transforming experience [4].

Machinima - the process of making real-time animated films by utilizing the 3-D graphics technology of computer games [3].

Sprites – A sprite is a two-dimensional image or animation that is integrated into a larger scene. Initially including just graphical objects handled separately from the memory bitmap of a video display, this now includes various manners of graphical overlays [6].

Virtual World Framework – Virtual World Framework is a fast, light-weight, web-based architecture for creating and distributing, scalable, collaborative, and component-based virtual spaces [5].

Prototypes Subject Matter

The Department of Defense Instruction, Combating Trafficking in Persons establishes policy in regard to all DoD Component heads to provide an annual awareness training program to all component staff and to provide data to the Office of the Secretary of Defense for Personnel and Readiness (OSD P&R) for its annual report [1]. Department of Defense policy states that CTIP is designed to oppose prostitution, forced labor, and any related activities such as coercion, commercial sex act, debt bondage, involuntary servitude and sex trafficking that may contribute to the phenomenon of TIP as inherently harmful and dehumanizing. [2].

Prototypes Audience

The Urban Explorer Prototype was designed as an operational link for students to access while taking the CTIP computer-based training course located on the JKO Web site.

Prototypes Learning Methodologies

The prototypes are designed to provide experiential learning as the primary means of learning while employing didactic learning experiences to validate knowledge of associated terms and concepts and to reconcile the learners understanding of their own experience within the prototype to the experiences designed to be imparted by the prototypes. The prototypes enable learners to visualize realistic physical environments and interact within them to create the illusion that they are experiencing the environments in real-life [4].

Virtual World Framework Technology Overview

The VWF is a new multi-user and collaborative web-based server that utilizes evolving and emerging web standards including Javascript, WebSockets, and WebGL. The VWF employs server and next generation browser technology that enables developers to quickly create 3-D scenes and allows multiple users to collaborate within the scenes in real-time. It is designed for users to share 2-D and 3-D visualizations and interactions through web browsers. Javascript is a popular programming language that enables rich user experiences and client-side calculations within Web browsers. WebSockets enable web browsers to employ bi-directional, real-time communication between two remote processes. WebGL is an emerging standard for graphics rendering inside of Web browsers that allows for complex visualizations for 3-D games and virtual worlds. As a renderer, WebGL is very new for desktop browsers. Support for it within mobile Web browsers is currently very poor and in most cases does not yet exist.

Requirements

The prototype designs were based upon the following contract requirements:

1. Design a 3D virtual environment for the student to experience a virtual vignette relating to the lesson content of the CTIP course that is interactive with the student. This optional training module should be an interactive vignette of the lesson content.

2. The Contractor shall develop the interactive 3d virtual training module that will last between 5-8 minutes.

3. The course shall run inside a browser, requiring no browser plug-ins (other than Adobe Flash), and load/stream with minimal time and effort.

4. The CTIP mobile course will link to an optional virtual training module hosted by the developer at the developer site.

Two prototype systems were developed in support of these priorities. The primary prototype was designed to enable learners to explore a virtual urban environment for environmental, physical and behavioral signs of CTIP. This system shall be referred to as the “Urban Explorer Prototype” and was developed to satisfy all requirements. The supplemental prototype was designed to enable learners to monitor a sequence of interactions between avatars for signs physical and behavioral signs of CTIP. This system shall be referred to as the “Sprite Avatar Prototype” and was developed to satisfy requirements 1, 3 and 4.

URBAN EXPLORER PROTOTYPE DESIGN

The Urban Explorer prototype was designed to address two goals. The first was to provide an interactive educational experience that would augment the Combatting Trafficking In Persons (CTIP) course and the second was to create an operational technology demonstration of the Virtual World Framework (VWF) capabilities in mobile browsers.

Experimental Designs Overview

The following sections detail the methods, designs and results of each demonstration constructed for the prototype. The learning goals of the prototype were as follows:

1. Identify environmental signs of CTIP
2. Identify physical signs of CTIP
3. Identify behavioral sings of CTIP

The prototype was designed to provide an interactive, immersive environment for exercising and reinforcing lessons learned using an open world venue in which learners would look for clues of human trafficking. Learners would apply the lessons learned in the CTIP mobile course in assessing a situation they might witness while working abroad and then demonstrate that they understood the proper course of action to take when faced with a possible case of trafficking in persons.

Demonstrations Design Approach

The Urban Explorer prototype resulted in two demonstrations using panoramas and machinima. Both demonstrations required a technical workaround as an alterative to using full 3D models to represent the immersive world environment. 3D models are based on meshes and are many times overlain with large textures, which created two problems for mobile devices. First, mobile devices didn't posses enough memory to store VWF 3-D assets. Second, mobile devices were incapable of processing large VWF scene files within an acceptable level of performance. The transfer of these files over Internet connections using WebSockets and to be processed by a mobile browser was also an impediment to performance. Some files were too large to be fully loaded without a browser timeout.

To address these performance problems another approach had to be taken. It was determined that full 3D rendering had to be abandoned to create an immersive CTIP virtual environment. The solution was to use a series of panoramas generated from an existing 3D environment for navigation. Panoramas in this context are a series of photos taken from a single location of a scene that are stitched together to create a concave spherical projection of a scene. Several panoramas were created at strategic locations in the 3-D model. A navigation system was created so that the user could move from panorama to panorama by clicking on arrows shown on the perceived ground. This technique is similar to the approach used in Google Streetview where users may access panoramas from a street map that have been created using trucks taking pictures with 360° cameras that follow driving routes all over the world.

Machinima were then incorporated into the immersive environment for learners to discover through exploration and identification and to evaluate as possible human trafficking situations. These movie clips were delivered using embedded Web video. The machinima were then followed by a series of questions that asked what kind of trafficking in persons situations they had witnessed, why they believed they were trafficking in persons situation, and finally what actions they should take in response. An evaluation of their performance was presented at the end using a scoring feedback mechanism and after-action review.

The panoramas and the machinima used the same 3-D models and the same rendering engine for capture. For example, a scene of children begging on the street was animated on a street corner from the environment that was used to generate the panoramas. The environment for the panoramas contained the same begging street children in its captured stills. This content cohesiveness helped create the illusion of immersion in spite of the lack of a 3D virtual environment.

Results

The two demonstrations developed for the Urban Explorer prototype behaved nearly identically in user experience, however their technical approach to delivering an immersive world differed. The first demonstration utilized an HTML5 panorama viewer that was incorporated into the VWF server. This demonstration became JKO's operational version linked to the CTIP mobile course as it worked across both Android and iOS web browsers. The second demonstration utilized a panorama viewer but one that was developed utilizing the VWF's native WebGL implementation. While not a requirement for the training, this demonstration had the added benefit of natively implementing the VWF's collaborative communication. This allowed for one mobile device to control what another mobile device was viewing or *vice versa*.

SPRITE AVATAR PROTOTYPE DESIGN

The final sprite avatar design was not determined at the outset of the prototype design effort and was instead arrived upon three (3) evolutionary iterative design experiments. The primary driver for the iterative experimentation was to develop an acceptable balance of visual and hardware performance while meeting compatibility requirements for the mobile browsers of choice. A fourth and tertiary design was also employed to present the learner with a more interactive and a usable experience.

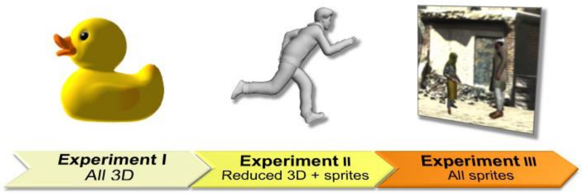

Figure 1. Design Experiment Evolution Timeline

Experiments Test Harness

Each experimental design utilized the following test harness to validate test results:

Devices:

Table 1 Experiment devices

	Name	Operating System	Hardware
Device 1*	MacBook Pro	OS X 10.8 running Windows 7 Parallels	2.7 GHz Intel Core i7 16 GB 1600 MHz DDR3 RAM NVIDIA GeForce GT 650M 1024 MB
Device 2	Nexus 7 Tablet	Android 4.2 Jelly Bean	NVIDIA® Tegra® 3 quad-core processor 1 GB RAM 7” 1280x800 HD IPS
Device 3	ASUS Tablet	Android 4.1 Jelly Bean	NVIDIA® Tegra® 3 T30L Quad Core 1 GB RAM 10.1” WXGA IPS
Device 4	Samsung Galaxy S3	Android 4.0 Ice Cream Sandwich	1.4 GHz quad-core Cortex-A9 16 GB RAM 4.8” HD Super AMOLED

*The desktop was used only to verify if VWF scenes worked on the desktop if they didn’t work on mobile as a way to determine what design approaches worked on mobile versus the desktop.

Browser: Mobile Mozilla FireFox 18+

Pass/Fail Criteria:

In this context a status of “failed” indicates that the experimental VWF scene failed completely, partially or intermittently when executed on the specified mobile device.

Experimental Designs Overview

The following sections detail the methods, designs and results of each experimental design constructed for the prototype. The learning goals of the prototype were as follows:

1. Identify physical signs of CTIP.
2. Identify behavioral signs of CTIP.

The test method component of each section discusses the testing employed to determine performance of the design. The design component details the interactive and visual inputs and outputs developed for the experimental design. Finally, the results component reveals the outcomes of the experimental design.

Experimental Design 1

Experimental design 1 was conducted using desktop VWF function and performance as a baseline under the assumption that if a mobile version would be capable of obtaining desktop function and performance that no further experimentation be necessary on mobile as desktop functionality provides all interactive and spatial features necessary to create an immersion and experiential learning.

Test Method: Observe performance of mobile devices to execute native VWF nodes, meshes, materials in pursuit of a 3D environment and avatars, as commonly implemented in desktop browser-based VWF scenes.

Design: Visualization included the display of the VWF duck and Humvee assets from the VWF Web site. Interaction included the learner being able to click on the duck to activate its standard rotation translation and to click on constituent parts of the Humvee model in order to verify that the correct parts were clicked. The decision to pursue these display and interaction tests with models were to simulate what would happen if a CTIP scene were constructed in which avatars cold be clicked upon for inspection, interaction and/or knowledge check activation.

Results: Mobile devices would either not load the desired VWF scene or they would load part of a scene, but not the whole resulting in operational errors. In testing against the standard demos provided on the VWF site, similar results were found. It appeared execution of 3D mesh-based content on the devices was causing scene failure. It should be noted that the scene worked as expected within a desktop environment.

Table 2 Experimental Design 1 Results

	Device1	Device 2	Device 3	Device 4
Experiment 1	Passed	Failed	Failed	Failed

Experimental Design 2

Experimental design 2 was conducted to address performance problems encountered in the display of 3D mesh assets on mobile devices. To do so, it was determined that a gradual stepping-back from the use of all 3D meshes to a partial use with 2D sprites would allow the development team to determine what, if any, performance improvements occur. It would also help determine if sprites could be effectively used within VWF mobile environments.

Figure 2. Experimental Design 2 Screenshot

Test Method: Observe performance of mobile devices to run native VWF nodes, materials and reduced-volume 2D plane meshes mapped to 2D sprites in pursuit of memory efficient and higher-performing environments and avatars.

To do so required moving under the VWF abstraction layer and working directly with underlying GLGE graphics code.

Design: Visualization included the display of a red rectangle and a hidden sprite surface on a 2D plane facing the VWF camera. Interaction included clicking the red rectangle to reveal the sprite surface and to cycle a sprite animation of a running man. The decision to pursue this sprite test design was to simulate what would happen if a CTIP scene were constructed with 2D plane mapped avatars that could be clicked upon for inspection, interaction and/or knowledge check activation.

Results: The desired sprite-to-mesh mapping was successful and achieved synchronization across clients for viewing sprite animations when activated by learners. However, in attempting to animate the sprites for purposes of character locomotion we encountered significant flicker that diminished visual aesthetics. In addition, we continued to experience scene failure across mobile devices tested.

Table 3 Experimental Design 2 Results

	Device1	**Device 2**	**Device 3**	**Device 4**
Experiment 2	Passed	Failed	Failed	Failed

Experimental Design 3

Experimental design 3 was conducted to address performance problems encountered in the display of sprites on 2D meshes. In continuing to step-back from the number of 3D meshes used in our experiments to reach an acceptable level of visual performance the development team decided to remove all 3D meshes from the scene and to move to a completely sprite-based implementation to include avatars and environments.

Figure 3. Experimental Design 3 Screenshot

Test Method: Observe performance of mobile devices to run only VWF nodes, with no meshes or materials, while using sprites in pursuit of higher-performing environments and avatars. To do so required overlaying an HTML 5 (Hypertext Markup Language) canvas over the VWF canvas for painting of sprites.

Design: Visualization included fluidly moving avatars within a simulated 2.5D perspective environment. Interaction included starting a vignette for physical and behavior observation. Learners started the vignette by clicking a button at the top of the screen and answered knowledge check questions by selecting and submitting them within a JQuery-styled modal popup window.

Results: This experiment demonstrated the best visual-aesthetic and hardware performance while again achieving synchronization across clients. However, tests continued to show scene failure across all, but a single device on which a consistent and useable experience was obtained.

Table 4 Experimental Design 3 Results

	Device1	**Device 2**	**Device 3**	**Device 4**
Experiment 3	Passed	Failed	Failed	Passed

Tertiary Experimental Design

Method: Develop components in support of temporal training narrative, character controls, sprite management and visualization, knowledge checks and after action review (AAR).

Results: An exercise event-sequencing engine was developed using JavaScript and YAML (YAML Ain't Markup Language) to enable sequences of linear or parallel-running narrative events to occur on a prescribed timeline. In the interest of time the engine was utilized to execute a linear narrative story instead of higher-order Immersive Multimedia Instruction (IMI) elements for choose-your-own-adventure styled play. The narrative playback worked properly using client synchronization. In order to use the event-sequenced narrative the learner was allowed to enter "Story Mode."

An avatar locomotion and animation component was developed to enable the learner to translate the antagonist character from the narrative around the VWF scene using the W, A, S and D keys. This component was provided purely as a demonstration of how to synchronize animation and translation of a 2D sprite character using JavaScript and YAML in the VWF. In order to use the character the learner was allowed to use "Freeplay Mode."

Because an efficient and consistently working sprite canvas could not be established using the VWF material node and texture in the second experiment, it was determined that development a sprite engine would be necessary to display sprites on an HTML 5 canvas. The engine was developed to keep track, display, translate and dispose of sprites. The engine was developed using JavaScript and YAML and are scheduled an executed using the exercise event-sequencing engine also developed as part of this tertiary experiment.

A knowledge check component was developed in order to test the learner's knowledge against what they had observed within the executed narrative. The component presented a linked-list of multiple-choice questions for the learner to answer. The knowledge check component was written in JavaScript and YAML and as HTML as part of the framing file that executes the VWF code itself. This framing file also served as a test harness in which HTML buttons were used to test individual aspects of the sprite engine and sequencing engine. These buttons were removed after testing for the final version of the prototype. An after-action review component was developed to show the learner a quantitative roll-up of their scores relative to the questions that they had answered in the knowledge check component. Passing answers were shown with a green checkmark graphic while missed answers were shown using a red X.

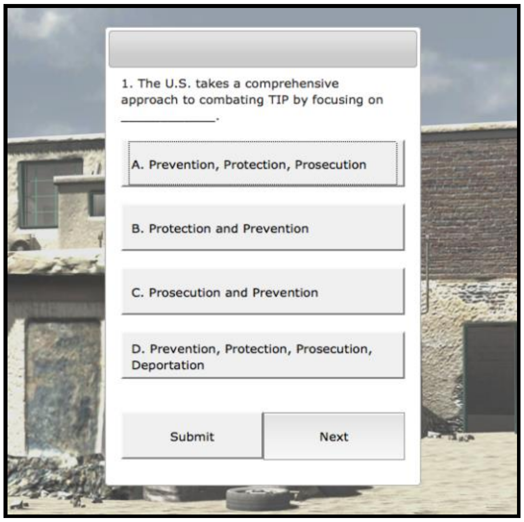

Figure 4. Knowledge Check Panel

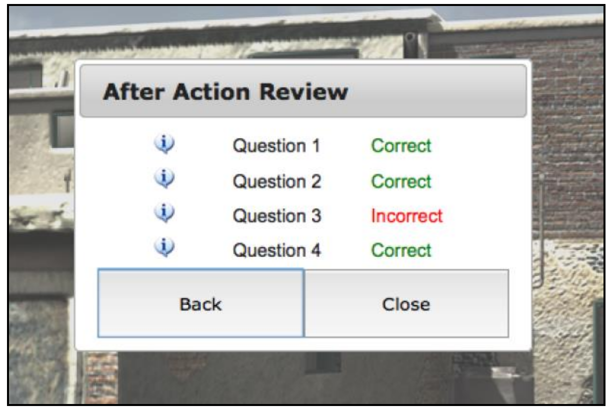

Figure 5. After Action Review Results Panel

Prototype Avatar Sprite Results Summary

The following table shows a summary of the results from across all three experimental designs. In this context a status of "failed" indicates that the experimental VWF scene failed completely, partially or intermittently when executed on the specified mobile device.

Table 5 Experimental designs results summary

	Device 1	Device 2	Device 3	Device 4
Experiment 1	Passed	Failed	Failed	Failed
Experiment 2	Passed	Failed	Failed	Failed
Experiment 3	Passed	Failed	Failed	Passed

DISCUSSION

Our experimental designs results indicate that the VWF implementation for 3D has not been optimized for mobile. The early application of VWF technology for mobile has created many challenges for our engineers to meet contract requirements. Results also show that use of non-native VWF HTML 5 elements in tandem with the native collaborative capabilities of the VWF are capable of providing a 2D, multiplayer solution for mobile.

Observations

WebSockets is an emerging standard that developers should continue to monitor. It's well known as a browser-based standard but can also be used by native applications. As libraries that implement WebSockets continue to improve, new possibilities in supporting communication between native and browser-based applications will emerge. Such improvements may allow native versions of the VWF client to be produced for mobile devices that are optimized to handle 3-D content.

Currently, mobile WebGL is only supported out-of-the-box on Firefox for Android. Chrome for Android may be configured to support WebGL and mobile Safari installed on iOS devices only makes WebGL available for advertisers using iAd.

Many web-based 3D engines are not optimized for mobile browsers. Native is still the method of choice when creating 3D immersive environments for mobile devices. There are several mature 3D engines and development environments that support mobile platforms such as Unity, Project Anarchy, and Unreal.

Finally, creation of IDEs supporting practical VWF application development tasks are desirable to include support for 2D and 3D graphical user interfaces development, music and sound integration, animation & modeling integration, learning objectives and learner missions development, event sequencing development, geospatial-based terrain integration, data feed integration, character AI integration, physics integration, collaboration support and gamification support.

Recommendations

In closing, we have formulated two recommendations for consideration in future development of the VWF for mobile usage based upon our experimental design results and observations. The recommendations include:

1. Augmentation of the WebGL VWF driver for optimal mobile browser operation; and/or

2. Development of a standalone application supporting VWF content and communication protocols.

We believe that implementation of these recommendations will assist in providing a user experience commensurate with the performance expectations of contemporary mobile applications and will help to increase the interest and utility of the VWF for the development community.

ACKNOWLEDGEMENTS

We would like to acknowledge Mr. Joseph Camacho and Cathy Moran of the JKO for providing the opportunity to conduct the experimental prototype designs and the CTC and ECS teams for carrying out the experiments. We would also like to acknowledge the efforts of the Advanced Distributed Learning (ADL) for volunteering technical advice throughout the effort.

REFERENCES

[1] Combatting Trafficking in Persons Web Site (2013), *Required Training Materials*, Retrieved June 10, 2013 from: http://ctip.defense.gov/Training.aspx

[2] Combatting Trafficking in Persons Web Site (2013), DODI 2200.01 Combating Trafficking in Persons (CTIP), Retrieved June 10, 2013 from:
http://ctip.defense.gov/Portals/12/Documents/TIP_DODI_220001p%5B1%5D.pdf

[3] Dictionary.com Web Site (2013), Machinima Definition, Retrieved June 28, 2013 from: http://dictionary.reference.com/browse/machinima

[4] Experience Based Learning Systems Web Site (2013), Experiential Learning Theory: Previous research and New Directions, Retrieved June 10, 2013 from:
http://learningfromexperience.com/media/2010/08/experiential-learning-theory.pdf

[5] Virtual World Framework Web Site (2013). *About, Virtual World Framework. Retrieved May 13, 2013 from: http://virtualworld framework.com/web/about.html*

[6] Wikipedia Web Site (2013). Sprites, *Wikipedia: The Free Encyclopedia.* Retrieved May 13, 2013 from:
http://en.wikipedia.org/wiki/Sprite_(computer_graphics)