Development and Implementation of an Enhanced Virtual Reality Development System Into the Computer Assisted Rehabilitation Environment (CAREN)

Jacob A. VanDehy[1], Vrajeshri Ordek[1], Harvey Edwards[2], Christopher Smalt[2], Chris Gengler[2], Rasmus Grunnet-Jepsen[1], Chris Montagne[3], Kyle Fitzgibbons[4], Doug Brungart[3], Sarah Kruger[4], Pinata H. Sessoms[1]

Author Contributions: PS, SK, DB, and HE developed the idea for the proposed work. HE, PS, and JV performed initial hardware integration development. HE and JV performed initial software integration development. HE, JV, VO, KF and SK performed additional development to refine the system to work at different sites with different projection and motion capture systems. DB, SK, HE, CS, CG and CM developed the AAVS application. JV, VO, and RG-J drafted the initial report. JV, VO, SK, and PS revised the report. PS and SK led project development needs and execution.

[1]Warfighter Performance Department Naval Health Research Center 140 Sylvester Rd San Diego, CA 92106-3521	**[2]Lincoln Laboratory Massachusetts Institute of Technology 244 Wood St Lexington, MA 02421**	**[3]National Military Audiology and Speech Center, Walter Reed National Military Medical Center 4954 North Palmer Rd Bethesda, MD 20889-5630**	**[4]National Intrepid Center of Excellence, Walter Reed National Military Medical Center 4860 South Palmer Rd Bethesda, MD 20889-4649**

Report No. 22-90 was supported by the Congressionally Directed Medical Research Programs Psychological Health/Traumatic Brain Injury Research Program under work unit no. N1801 (funding year: FY16; award number: CDMRPL-17-0-PT160118). This material is based upon work supported by the Department of the Navy under Air Force Contract No. FA8702-15-D-0001. The views expressed in this article are those of the authors and do not necessarily reflect the official policy or position of the Department of the Navy, Department of Defense, nor the U.S. Government.

The study protocol was approved by the Naval Health Research Center Institutional Review Board in compliance with all applicable Federal regulations governing the protection of human subjects. Research data were derived from an approved Naval Health Research Center Institutional Review Board protocol, number NHRC.2019.0002.

EXECUTIVE SUMMARY

U.S. Department of Defense (DoD) health and research systems have several rehabilitation platforms. These platforms, which once represented the peak of computer and hardware technology, now lack necessary support from the original manufacturer to keep pace with advances in graphic systems and third-party biometric measurement hardware. To conduct the most effective scientific research and maintain a competitive edge, an upgrade to a modifiable development system with access to modern graphical inputs is necessary.

This report details the DoD, Naval Health Research Center (NHRC), and partners' development and testing of an updated gaming-based system using the Unity system for simulation development and hardware integration with one such platform: the Computer Assisted Rehabilitation Environment (CAREN), a large-scale immersive virtual reality system. From 2017 to 2019, engineers from NHRC, National Intrepid Center of Excellence (NICoE), and Massachusetts Institute of Technology Lincoln Laboratory (MIT LL) collaboratively developed a complete hardware–software system that integrates seamlessly with existing simulation systems and utilizes the current standard for simulation software development to create a development platform that is robust, flexible, and meets the needs of an evolving military mission.

TABLE OF CONTENTS

INTRODUCTION

Over the past decade, the Military Health System (MHS) has made substantial investments in the development of sophisticated multimodal virtual reality (VR) systems that can address the unique challenges faced by patients suffering from vestibular and sensory issues related to traumatic brain injury (TBI). These VR systems are used for rehabilitation and can be used as an independent modality or as an adjunct to traditional therapies. They have been shown to improve outcomes in patients with co-occurring clinical presentations of multisensory dysfunction (Gottshall & Sessoms, 2015; Sessoms et al., 2015; Sessoms et al., 2021). However, utilization of these new VR-based rehabilitation strategies is restricted due to a limited number of large, expensive, fixed systems which has constrained both the overall number of patients who have access to these therapies as well as the amount of time that individual patients can use these systems for their rehabilitation regimens.

One example of a sophisticated VR-based system that is used as a modality for rehabilitation is the Computer Assisted Rehabilitation Environment (CAREN; Motekforce Link, Amsterdam, The Netherlands). The CAREN system utilizes interactive virtual environments (VEs) that incorporate visual, auditory, and motion (treadmill embedded on a 6 degrees of freedom [6DOF] motion platform) inputs that can challenge a patient's cognition, vision, balance, and motor functions, individually or simultaneously, if desired. The system utilizes VEs that allow clinicians to easily modify the difficulty level and task requirements (both physical and/or cognitive) in a systematic fashion to target patient-specific needs as they progress through therapy. Of the approximately 30 CAREN systems in the world, five are currently installed in U.S. Department of Defense (DoD) medical and research facilities: Naval Health Research Center (NHRC), Walter Reed National Military Medical Center (WRNMMC), National Intrepid Center of Excellence (NICoE), Center for the Intrepid at San Antonio Military Medical Center, and at Massachusetts Institute of Technology Lincoln Laboratory (MIT LL). Within the MHS, the CAREN has been successfully utilized for wounded warrior rehabilitation as well as to investigate novel research questions. However, its native software, D-Flow, has been limiting for VE development as it has been maintained solely by Motekforce Link, which has lagged in both graphics processing unit (GPU) and code advancement. Graphics are limited to '90s-era polygon counts, with a deprecated file converter required for custom work. The codebase is closed source and runs on the Lua engine, which is rarely supported by software development kits (SDKs) associated with research hardware. Thus, to maintain the cutting-edge capability of this investment, a new simulation development platform had to be integrated into the DoD CAREN systems.

To that end, development of a new simulation platform was undertaken as part of a 4-year multisite research project established in fiscal year 2018 under funding from the Congressionally Directed Medical Research Programs Psychological Health/Traumatic Brain Injury Research Program (Award Number: CDMRPL-17-0-PT160118). This project, known as Virtual Reality for Vestibular Physical Therapy (VR4VPT), was spearheaded by NHRC, NICoE, and other partners and will directly serve the wounded warrior community and their beneficiaries by providing advanced rehabilitation and assessment to patients throughout DoD. As part of this undertaking and technological advancement, it was necessary to upgrade from the base CAREN functionality (D-Flow software) to a commercial-grade simulation development platform using the Unity 3D game engine.

METHODS

The Development Process

Selection of Unity Simulation Development Platform

Limitations of VE development with CAREN D-Flow are twofold: an outmoded graphics engine and closed scripting capability, both of which are solely supported by Motekforce Link. The graphics suite provided by Motekforce Link has assets with polygon counts in the low thousands, which is two orders of magnitude lower than 2022 industry capability. Creating and converting custom graphic assets from Autodesk (Autodesk Inc., San Rafael, CA) or Blender® software (Stichting Blender Foundation, Amsterdam, The Netherlands) is possible to created different visual scenes, but scene files can only through the OGRE 3DS Max exporter, an open-source system developed in 2010. VE development for the CAREN platform is handled through D-Flow, Motek's proprietary simulation development platform. The D-Flow editor is a graphics-based interface, wherein the user must connect inputs and outputs from module icons that control different aspects of the system like the treadmill and motion capture hardware, as well as various assets from the scene file like objects and animations. While the editor allows nonprogrammers to develop VEs for the CAREN, each module has a limited number of accessible features and there is no backend access that would allow for modification. Additionally, the incorporation of new hardware required the creation of new modules, which must be completed by Motekforce Link and billed accordingly. They created a heart rate module to support certain models of Polar® and Zephyr™ heart rate monitors, but it has not always been reliable and may not meet the requirements of DoD systems. In 2014, scripting capability, written in Lua, was added through a script module. While Lua is open source and compatible with most C++ systems, it does not integrate with Microsoft's C# suite without the use of secondary compilers and converters, thereby introducing a new developmental delay.

While modern simulation and gaming studios have proprietary engines for VE development, there are two popular engines that are available for small-developer teams with off-the-shelf company support: Unreal Engine (Epic Games, Cary, NC) and Unity 3D (Unity Technologies, San Francisco, CA). Unreal Engine has been used for over 20 years for game development and is prized for its efficient graphics capabilities with a C++ scripting backend. Unity 3D began development in 2004 and is designed with a C# scripting backend, lower graphic capability but higher flexibility, offering automatic conversion to platforms for any consumer-grade device that supports game development. This includes head-mounted display (HMD) VR systems such as the HTC VIVE® (HTC Corporation, Taoyuan City, Taiwan) or Meta Quest 2 (Meta, Menlo Park, CA) and augmented reality (AR) systems such as Magic Leap (Magic Leap, Inc., Plantation, FL) and HoloLens 2 (Microsoft Corporation, Redmond, WA). Ultimately, Unity 3D was chosen because of its support for C# scripting, which is used in most SDKs for new biomedical technologies, and for its platform conversion functionality to ensure that new simulations developed for the CAREN can be pushed to VR and AR HMD devices.

Systems to Recreate in the Unity Development Platform

For the new Unity 3D-based platform to work successfully, it needed to be integrated with existing hardware associated with the CAREN system. Several key features required some redevelopment: motion capture streaming, treadmill and motion base control, and multi-projector image blending.

Motion Capture Streaming. The CAREN system is capable of receiving streaming optical motion capture data from a variety of systems. Vicon™ (Vicon Motion Systems Ltd UK, Oxford, UK), OptiTrack™ (NaturalPoint Inc., Corvallis, OR), Cortex™ (Motion Analysis Corporation, Rohnert Park, CA), and Qualisys (Göteborg, Sweden) systems currently can communicate with D-Flow. Streamed data, on a shared network from the motion capture system to D-Flow, can be used to drive participant interaction with the system or control progression through each VE (e.g., self-paced treadmill capabilities). However, streaming directly from the capture source to the simulation system was determined to be the optimal design to minimize lag in participant input.

Treadmill and Motion Base Control. The CAREN system has a 6DOF motion base and split-belt treadmill that can be synchronized to the optical flow in the visual projection. Controlling both is a priority for maintaining system functionality. Because the motion base (Moog Inc., Elma, NY) has strict controls on messaging latency to and from the motion base's real-time control computer, D-Flow was maintained as an intermediary system to avoid small changes in latency and base disruption. Streaming directly to the networked motion base computer should also be considered for further development.

Image Blending. The immersive visuals for each VE require multiple projectors to create the approximately 180-degree visual projection. These projectors are programmatically blended to produce a seamless blend, a single image across the entirety of the screen. This was previously maintained by Motekforce Link warping and blending hardware components. During the course of this project, work was done at NHRC, MIT LL, and Motekforce Link to enable use of Scalable blending (Scalable Display Technologies, Cambridge, MA), which is now the current standard for warping and blending with the CAREN system. Maintaining a blend on both the Unity and D-Flow systems concurrently, without arduous hardware switching, is a necessary deliverable.

Hardware Integration

Extron Video Switch

An Extron CrossPoint XTP 1600 (Extron, Anaheim, CA) is used to switch between video streams from different sources. This allows the visual projection output to switch between using images from the D-Flow image generators (IGs) and images from the Unity system. Programming the CrossPoint is facilitated using the Extron DataViewer program. Secondary connection methods are available through Ethernet command, but they must be set up using DataViewer.

Presets are buttons that correspond to the input line and are available on the front panel of the Extron video switch (Figure 1). Pushing the Preset button, followed by the correctly programmed button (which will alight in red), followed by Enter changes the preset and allows for switching between the input lines (e.g., from D-Flow to Unity or vice versa).

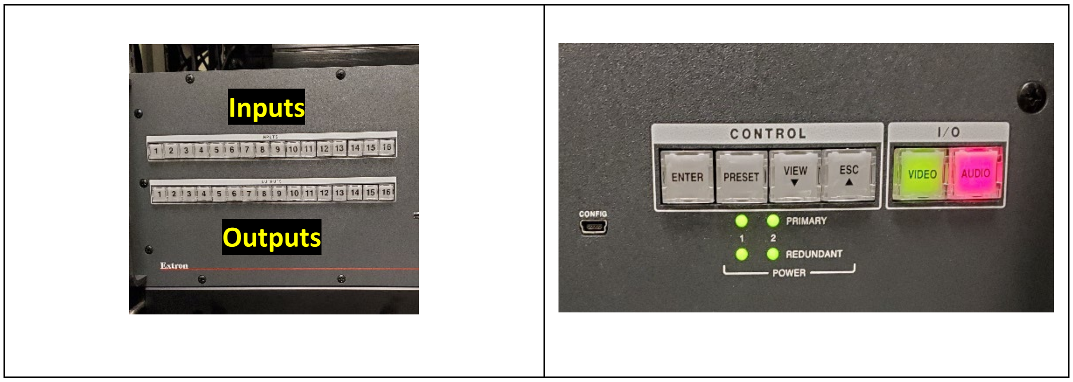


Figure 1. Extron switch inputs and outputs on the left and control functionality on right.

Graphics Processing Unit Requirement

The selected GPU must have the power to run a minimum of four outputs (three blended projectors and an operator computer monitor) in tandem. Unity 3D defaults to driving a display from its weakest GPU; using two different GPUs to drive projector and operator screen output is not recommended.

Additional Recommended Hardware for the Unity Tower

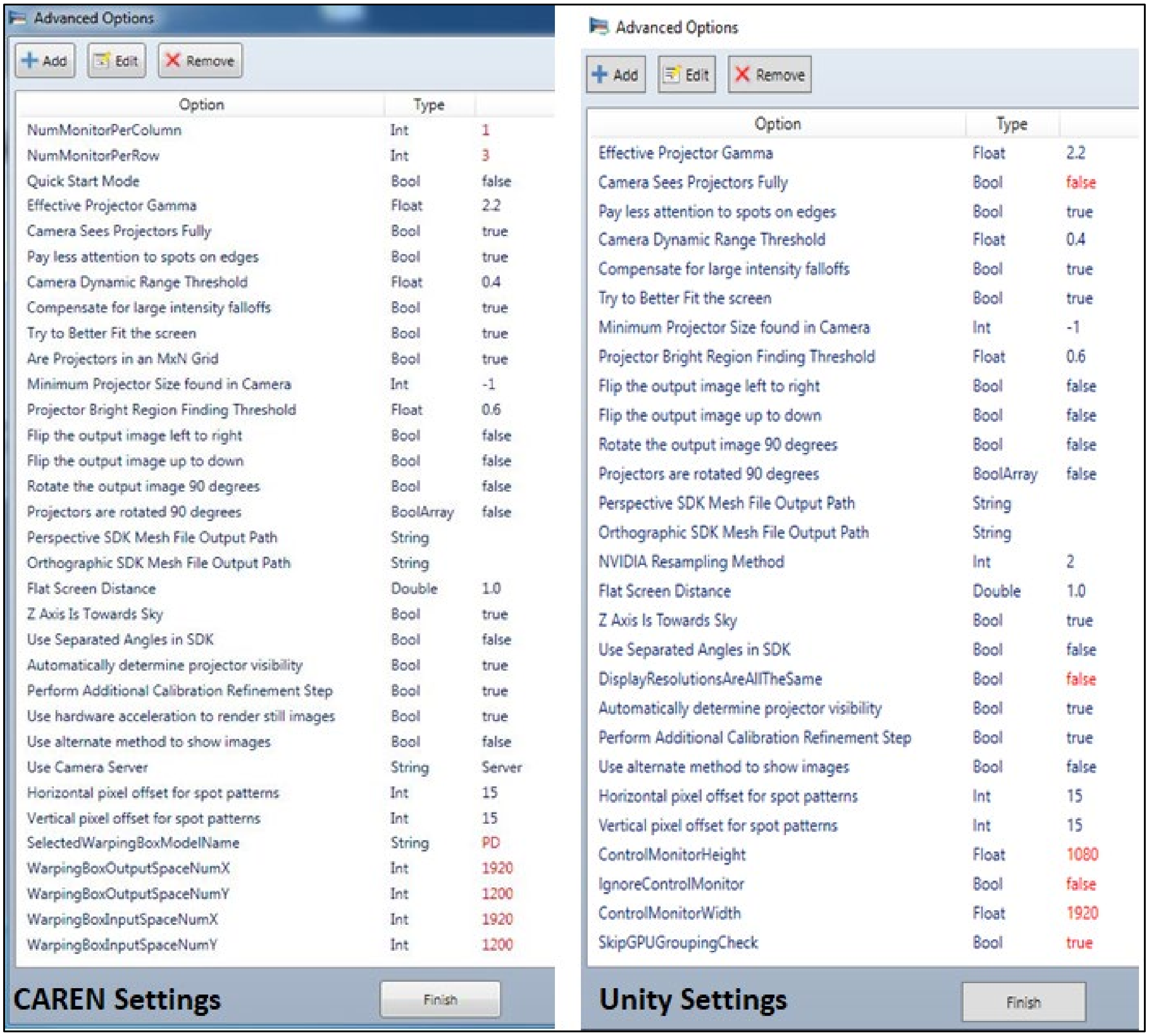
An Intel® Core™ i7-5000K (Intel Corporation, Santa Clara, CA) or above processor is recommended to effectively run programs without slowdown (as of Unity version 2021.1). Two network interface cards, one for Wi-Fi connectivity and one for Ethernet connectivity into the CAREN network, are recommended. A 500GB memory card is recommended for holding built programs and integrating data collection.

Software Integration

Loading Scalable Onto the D-Flow System

Scalable version 3.10.080 is currently supported within the D-Flow 3.1 infrastructure for handling warping and blending. Unlike earlier D-Flow versions, with Scalable 3.1, both warping and blending can occur within the IG without the need for additional blending hardware.

The Scalable Client.exe must be installed onto each of the IGs. This enables Scalable to utilize the IGs in the Warp and Blend process. In the Scalable program, choose File>Advanced Settings and input the settings as shown in Figure 2. Next, enter the Projectors tab and select the internet protocol (IP) addresses of the IGs in your system (three IGs are shown in Figure 3), selecting the respective values for IG 1, 2, and 3. For the remainder of work on completing the blend, please refer to the following section on Warp and Blend integration.



Advanced Options

Add | Edit | Remove

Option	Type	
NumMonitorPerColumn	Int	1
NumMonitorPerRow	Int	3
Quick Start Mode	Bool	false
Effective Projector Gamma	Float	2.2
Camera Sees Projectors Fully	Bool	true
Pay less attention to spots on edges	Bool	true
Camera Dynamic Range Threshold	Float	0.4
Compensate for large intensity falloffs	Bool	true
Try to Better Fit the screen	Bool	true
Are Projectors in an MxN Grid	Bool	true
Minimum Projector Size found in Camera	Int	-1
Projector Bright Region Finding Threshold	Float	0.6
Flip the output image left to right	Bool	false
Flip the output image up to down	Bool	false
Rotate the output image 90 degrees	Bool	false
Projectors are rotated 90 degrees	BoolArray	false
Perspective SDK Mesh File Output Path	String	
Orthographic SDK Mesh File Output Path	String	
Flat Screen Distance	Double	1.0
Z Axis Is Towards Sky	Bool	true
Use Separated Angles in SDK	Bool	false
Automatically determine projector visibility	Bool	true
Perform Additional Calibration Refinement Step	Bool	true
Use hardware acceleration to render still images	Bool	true
Use alternate method to show images	Bool	false
Use Camera Server	String	Server
Horizontal pixel offset for spot patterns	Int	15
Vertical pixel offset for spot patterns	Int	15
SelectedWarpingBoxModelName	String	PD
WarpingBoxOutputSpaceNumX	Int	1920
WarpingBoxOutputSpaceNumY	Int	1200
WarpingBoxInputSpaceNumX	Int	1920
WarpingBoxInputSpaceNumY	Int	1200

CAREN Settings | Finish

Advanced Options

Add | Edit | Remove

Option	Type	
Effective Projector Gamma	Float	2.2
Camera Sees Projectors Fully	Bool	false
Pay less attention to spots on edges	Bool	true
Camera Dynamic Range Threshold	Float	0.4
Compensate for large intensity falloffs	Bool	true
Try to Better Fit the screen	Bool	true
Minimum Projector Size found in Camera	Int	-1
Projector Bright Region Finding Threshold	Float	0.6
Flip the output image left to right	Bool	false
Flip the output image up to down	Bool	false
Rotate the output image 90 degrees	Bool	false
Projectors are rotated 90 degrees	BoolArray	false
Perspective SDK Mesh File Output Path	String	
Orthographic SDK Mesh File Output Path	String	
NVIDIA Resampling Method	Int	2
Flat Screen Distance	Double	1.0
Z Axis Is Towards Sky	Bool	true
Use Separated Angles in SDK	Bool	false
DisplayResolutionsAreAllTheSame	Bool	false
Automatically determine projector visibility	Bool	true
Perform Additional Calibration Refinement Step	Bool	true
Use alternate method to show images	Bool	false
Horizontal pixel offset for spot patterns	Int	15
Vertical pixel offset for spot patterns	Int	15
ControlMonitorHeight	Float	1080
IgnoreControlMonitor	Bool	false
ControlMonitorWidth	Float	1920
SkipGPUGroupingCheck	Bool	true

Unity Settings | Finish

Figure 2. Advanced Options panels for Scalable Display Manager (File>Support>Advanced Options).

Figure 3. Scalable Display Clients selection for configuring the image generators or graphics processing unit for the D-Flow blend or Unity blend, respectively.

Scalable 4.0+ Warp and Blend Integration on Unity System

After setting up the inputs to be used for the visual projection, the individual projections must be blended together into a single, seamless visual on the projection screen. This is performed by warping and blending the projections. C# libraries for Warp and Blend integration are currently supported for Scalable versions 4.0 and later. This allows automatic warping and blending using cameras pointed at the projection screen. Acquiring a good blend is a matter of proper hardware integration, camera placement, and screen-fitting process. Several attempts may be required, with different camera orientations and placements. If your setup supports it without obstruction, consider mounting your cameras permanently within the structure so that the calibration can be repeated whenever the blend may be off (which may occur if the projectors shift, bulbs dim, etc.). To achieve the best blend, position the initial unblended projectors so that the top and bottom of the images for the projectors are relatively aligned. The goal is to avoid tilt and offset between the projectors as much as possible. This will improve the overall warp at the end of the process.

Cameras and webcams are both supported for capturing screen calibration information. Current supported cameras are Canon (Canon Inc., Ōta, Tokyo, Japan) T1i, T2i, T3, T3i, T4i, T5, T5i, T6, and T6i, with EF-S 18–55mm, EF-S 10–22mm, or EF-S 10–18mm lenses. Supported webcams are the Logitech (Newark, CA) C910, C920, C922, C930e, or B910. The cameras must be positioned so the entirety of the screen is captured by each individual camera, or so that together two cameras capture greater than half of the total screen, with a 20% overlap in the center. This can be viewed in the Scalable Data Collection panel. As much as possible, it's desirable to fill the field of view of each camera with the screen view, minimizing the amount of non-screen space visible. When acquiring these data, ensure the room is as dark as possible by closing all doors and shades and turning off all lights and unnecessary monitors during the acquisition process.

Since not all CAREN systems have screens that are the same shape and/or size, it's essential to shape the geometry of the projection surface. Shaping the geometry of the screen requires moving six control points around the projected cylinder in the Scalable software, three at the top and three at the bottom, by using the mouse and arrow keys (Figure 4). These points should be set in the four corners of the cylinder and the center horizontal points. Points 1, 2, and 3 should be vertically aligned with points 4, 5, and 6, respectively. Points 1, 2, and 3 should align to the top of the screen, while points 4, 5, and 6 should align along the bottom of the screen. As you move control points in the Scalable software, the projected image will move and stretch on the screen accordingly with each new adjustment. Various "test images" are available to view the prospective warp and blend. The checkerboard pattern is particularly useful for viewing any egregious bubbling in the blend, where straight lines have slight bends that need to be smoothed out. If you cannot manipulate the six control points to achieve relatively straight lines, adjust the camera placement and take another round of data, then realign the points.

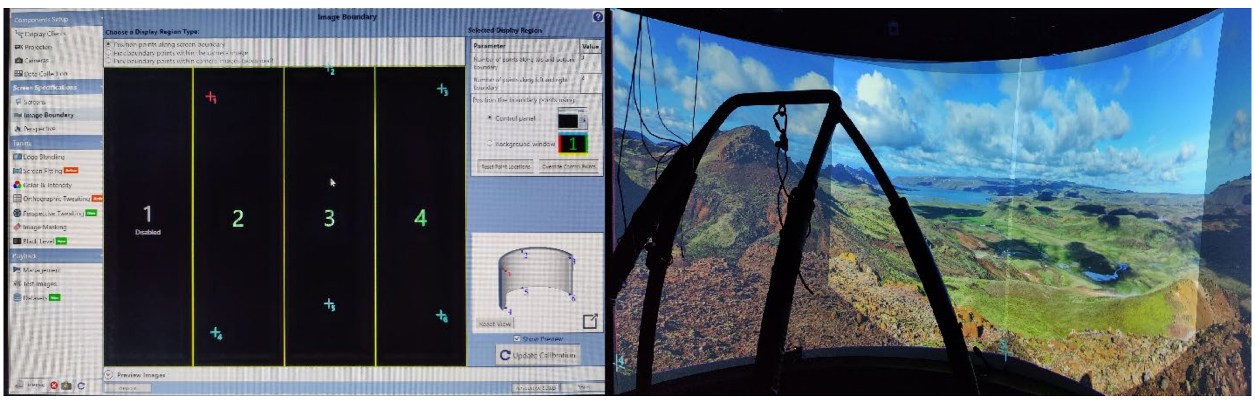


Figure 4. User interface (left) and physical display (right) of the geometry shaping for the warp and blend of a three-projection blend.

After the control points have been positioned in the software, minor adjustments in alignment can be made. Within the initial grid, points should be shifted in the software to achieve a better alignment with the physical projection. Subdivision can be increased, by adding more control points, to attain greater control over the horizontal and vertical appearance of the blend.

Once these points have been adjusted, generate the frustrum for each of the three projectors. These values are automatically saved to the "C:\Program Files\Scalable Display\DEI\LocalCalibration\" location, with files for ScalableData.o1, o2, and o3 for the respective projectors (more files will be needed if you have more projectors).

This blend value is applied to three virtual cameras in the Unity project. In all projects with blended screens the C# script DisplayMap.cs has three central functions: instantiating cameras for each of the ScalableData.o files; applying the initial rotation and warp to each; and applying frame-by-frame updates to the three based on the position of the master camera. Within

OnEnable, all previously generated cameras are reinstantiated after a scene transition. The script FindProjectionandRotation.cs reads the ScalableData.o files and produces projection matrices for the corners of each camera and rotation matrices for the camera positions. The PerspectiveOffCenter function is run through update and, frame by frame, updates the position of each of the three secondary cameras based on the position of the primary camera. Each time this cycle is run, the new camera becomes static at the position it is created. All views on the child camera are updated, frame by frame, using the PerspectiveOffCenter values, offsets to the parent will not be applied to the child cameras.

Installing Extra Lua Modules for Use in D-Flow

In order for the Unity software to communicate with the CAREN D-Flow computer, the Unity computer must be added to the D-Flow network via an Ethernet connection. On the Unity computer, from the Start panel, search for Network Status. Select the Ethernet connection and click the Change Adaptor Settings link. Right click the Ethernet connection and choose the Properties dropdown menu. Highlight Internet Control Protocol Version 4 (TCP/IPv4) and click the Properties button. Set a static IP address that is congruent to your current CAREN network computer, typically the D-Flow computer. For example, if your D-Flow computer is set to 192.168.90.1 with a subnet mask of 255.255.255.0, set the Unity static IP to 192.168.90.XXX, with a subnet mask of 255.255.255.0 and a default gateway of 192.168.90.1. **It is necessary to maintain a separate Ethernet connection, outside of the connection to the internet, to maintain this process.**

The LuaSocket module must be installed into the embedded D-Flow Lua scripting system to facilitate data transfer between the two computers. Download the open-source software as a zip file for Windows (http://files.luaforge.net/releases/luasocket/luasocket/luasocket-2.0.2). Unzip the files and place them in C:\\CAREN Resources\Scripts. If these are duplicate folders, drop the contents into the existing directories. Open the lua5.1.exe file and input the following command:

print("LUAMODULES:\n",(package.path:gsub("%;","\n\t")),"\n\nCMODULES:\n",(package.cpath:gsub("%;","\n\t")))

to check the installed pathway settings. The network pathway will need to be modified for the path and cpath values to find the .lua and .dll files, respectively. Lead your Lua scripts with:

package.path: ';.\?.lua;C:\\CAREN Resources\\Scripts\\lua\?.lua;C:\\CAREN Resources\\Scripts\\lua\\?\init.lua;C:\\CAREN Resources\\Scripts\\?.lua;C:\\CAREN Resources\\Scripts\\?\init.lua;C:\\Program Files (x86)\\Lua\\5.1\\lua\\?.luac'

package.cpath: '.\?.dll;C:\\CAREN Resources\\Scripts\\?.dll;C:\\CAREN Resources\\Scripts\\loadall.dll'

socket = require("Socket")

Place the TCP_Connection.lua script in each Application folder to facilitate Ethernet communication between the D-Flow and Unity control systems. Outputs from the D-Flow Lua submodule are then sent to either the Motion Base submodule to control roll, pitch, yaw, X, Y, and Z base motion or the Treadmill submodule to control treadmill speed parameters.

Utilizing Motion Capture Modules Within Unity (Cortex and Vicon)

Motion capture data from the motion capture computer stream into Unity through a User Datagram Protocol (UDP) Ethernet connection to a separate computer. To prevent unintended firewall intercepts, users must create firewall exceptions in the Windows Defender module for the Ethernet connections. Both Cortex and Vicon systems stream spatial location (X, Y, Z) data of optical markers in millimeters. These location data are specified in the motion capture software and can be linked to objects within the Unity scene for both participant interaction and rudimentary kinematic recording.

Vicon Streaming Into Unity

Information from the Vicon software streams into Unity in a separate thread via a marker-by-marker method. X, Y, and Z positional information is sent from Vicon to Unity and is then stored in arrays. With these data, segment information can be recreated within Unity. For example, center of mass (CoM) location can be calculated based on the number of markers passed into Unity. Parsing this array into segments is necessary to assign data to the usable stream. These data are read at a frame rate specified by Unity. The rotational vector is determined by a vector from the calculated CoM to a point forward in the system using the Quaternion.LookRotation function in Unity.

Cortex Streaming Into Unity

Information from Cortex is streamed on a segment-by-segment basis, not a marker-by-marker basis like Vicon streaming. Segments are defined and tracked in the Cortex system and streamed to Unity. Game objects within Unity inherit these independently; one game object corresponds to one segment in Cortex and must be assigned before building. Segment CoM X, Y, and Z positional information is streamed in millimeters and rotation is streamed in degrees using Euler angles definitions. Casting to quaternions within the Unity application functionality is discouraged because of this limited definition.

D-Flow Communication With Unity

A basic Unity application is created to stream treadmill and motion base data into D-Flow and the motion base. This application contains the TCP_Connection.lua script, which receives the information and passes it to the treadmill and motion base modules. It is important to use this to filter the motion base data from Unity before being passed to the motion base. To connect to the Unity stream, run the D-Flow program, then launch the Unity program, then press the manual Connect button on the D-Flow program. This connection will be maintained for the entirety of the D-Flow application runtime; tripping the light gate will not disturb the connection from Unity. The Unity program can stream data for control of the visual flow and treadmill speed alongside with Roll, Pitch, Yaw, X, Y, and Z commands from the Unity environment.

Case Study: The Aurally Aided Visual Search (AAVS) Task

In the Aurally Aided Visual Search (AAVS) paradigm (Brungart et al., 2019; Perrott et al., 1996), the participant is asked to perform different localization (search) tasks and is required to locate either an audio target (pinkish noise) and/or a visual target consisting of a cluster with an odd number of white dots (one or three, as shown in Figure 5). To test the original version of the AAVS paradigm at NICoE, their CAREN system was modified for the study to include a 64-speaker (MX-4 speakers; Meyer Sound, Berkely, CA) array mounted on a scaffold (custom truss), directly behind the fabric screen. MATLAB (The MathWorks, Inc., Natick, MA) was used to communicate with the speaker array and code was generated to select and play a single speaker which provided an auditory cue at the chosen target location. When a visual target was also employed, it is positioned at one of 64 target locations on the screen within 180° of azimuth (−90° to 90°, where 0° is the forward direction when the person is standing on the treadmill looking at the screen), and within a vertical range of approximately 3 meters, the height of the cylindrical screen at NICoE.

There are several variations of this task, including conditions with visual-only cueing (just the visual target cluster displayed), audio-only cueing (no visual target, just a directional auditory cue of where the cluster should be), and audio-visual cueing (visual target displayed at the screen location corresponding with the directional sound cue). The audio-visual cueing condition can also include visual distractors (distractor clusters comprising an even number of dots, either two or four dots placed at regular intervals across the range of azimuth and elevation; see Figure 6B) to make the task more challenging. The visual-only variant of the AAVS task was developed to facilitate comparison of results across different CAREN systems and serve as a “visual baseline.” This variant displays a target cluster of one or three dots in a known location, directly in front of the participant, and records their reaction time to press button A or B. The audio-only variant includes a head-slaved cursor that appears on the screen, controlled by motion capture markers that track head position and orientation, such that the participant must turn their head to move the cursor to identify the location where they hear the sound. For the audio-visual variant, the participant is instructed to use a wireless game controller to identify the number of dots in the target cluster (i.e., button A when the target has one dot or button B when the target has three dots) once they have found the target. This ensures that the participant is faithfully completing the task rather than guessing or faking a response. Participant reaction times (i.e., the time required to find the target cluster and then press the button) are measured for the different variations of the task. These tasks are performed standing in the middle of the treadmill, as well as while walking on the treadmill.

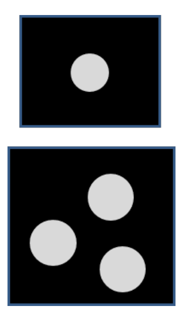


Figure 5. Visual target clusters of one (top) and three (bottom) dots.

After validation with the physical speaker array, a headphone-based virtual audio version was created. For that version, a virtual sound source is placed at the location of the target cluster, spatialized with a custom head-related transfer function (HRTF; Xie, 2013), and played over headphones to the participant. The HRTF contains interaural phase and level cues and monaural spectral cues that mimic the direction-dependent changes that occur in the left and right ears when the head is rotated relative to the direction of a real-world sound source. The interaural phase and level cues cause the pulsed-noise auditory cue used in the experiment to sound in front of the listener then the head is pointed directly at the target, to the left when the head is pointed to the right of the target, to the right when the head is pointed to the left of the target. The monaural spectral cues are less robust than the interaural cues, but they generally cause the target to sound above the horizon when the head is pointed below the target and below the horizon when the head is pointed above the target.

Figure 6A shows a participant on the MIT LL CAREN system during a trial of the audio-visual search task which includes visual distractors that fill up the remainder of the screen, as implemented in D-Flow. Figure 6B shows the same task as implemented in Unity 3D. In addition to headphones and as required for the version with the physical speaker array, the participant was also required to wear motion capture markers to track head position and orientation.

The Unity version of AAVS implements the same experimental features as the D-Flow version. It supports six variants of the task: audio-visual with distractors, audio-visual without distractors, visual-only with distractors, visual-only without distractors, audio-only, and the visual discrimination baseline. Using user-defined settings, visual and auditory stimuli are replicated in Unity to match as closely as possible to their visual counterparts in D-Flow and audio counterparts generated by MATLAB.

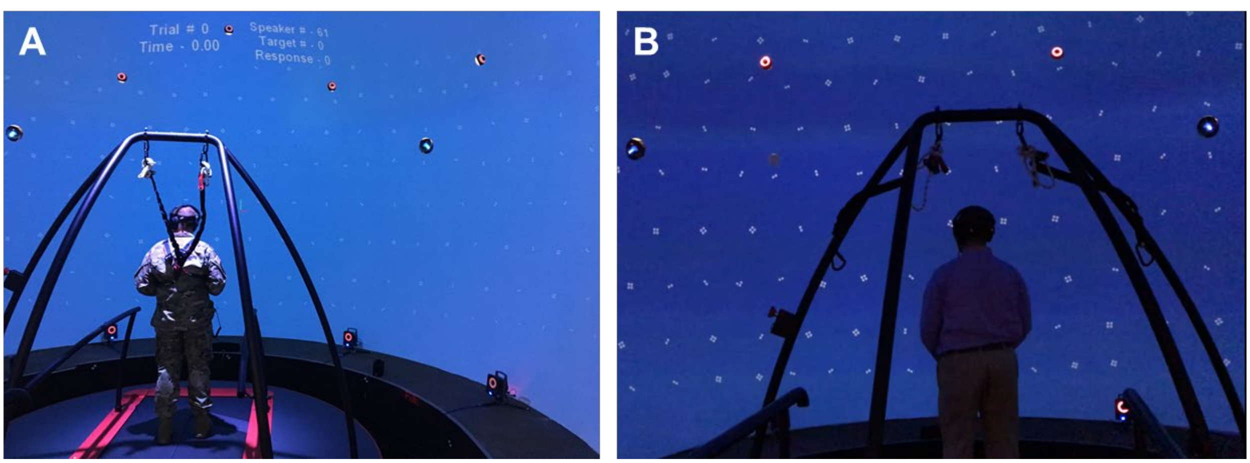


Figure 6. D-Flow and Unity versions of the Aurally Aided Visual Search (AAVS) task on the Computer Assisted Rehabilitation Environment (CAREN) at Lincoln Laboratory, Massachusetts Institute of Technology. (A) D-Flow version of AAVS in 2017. (B) Unity version of AAVS in 2019.

Design Limitations and Future Direction

The fundamental limitation of this endeavor is the D-Flow platform. To retain functionality of older CAREN VEs, the Motek's D-Flow proprietary program must be maintained, centralizing and retaining skills for future use. However, the most efficient method of communicating with the 6DOF motion base is through a direct UDP connection with the platform control computer

(Figure 7). A control protocol is available for this process, depending on the particular brand and model of motion base platform which varies across the DoD-based CAREN systems. A threaded direct communication with the platform computer could potentially allow Unity 3D to directly control the motion base.

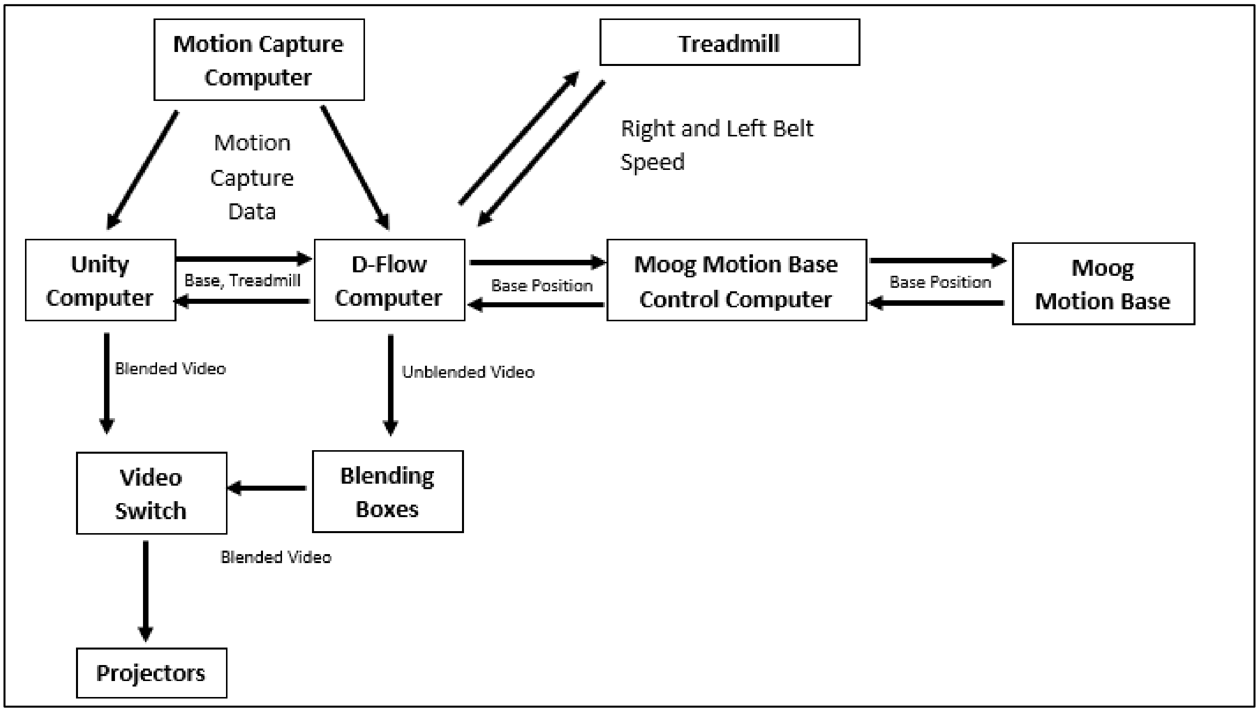


Figure 7. Flow diagram of data in the merged Unity–CAREN system. To remove the nexus of D-Flow, C# scripts would need to be written for both the treadmill and Moog controllers.

The creation of a basic front-end program for treadmill control, platform motion, and basic visual flow within Unity 3D would be an additional improvement. This could be used for pretrial setup and troubleshooting, basic testing, and demonstrations on the treadmill. This could completely remove the secondary user interface layer of the D-Flow connection. Though this is possible, the current system retains legacy D-Flow function.

SUMMARY

The present effort to develop and implement Unity 3D as a simulation and development platform was a success. The integration of an Extron video switch and a surround sound mixture allows for quick button presses to change the image and sound outputs between D-Flow and new Unity VEs. Software between both systems is mirrored for blending, automating the warping and blending process, and decreasing time to calibrate the projection. Streaming data from Unity to D-Flow allows for treadmill and platform motion, while direct streaming from motion capture to the Unity system maintains the nearly instantaneous interaction for participant feedback or rehabilitation in the environment.

As an additional part of the VR4VPT project, VEs developed for this Unity-based system on the CAREN platform have also been used to create mobile application versions that have been ported to VR and AR HMDs, such as the Magic Leap, HTC Vive, and Microsoft HoloLens 2.

Utilizing the suite of SDK functionality simplifies ports from full simulator development to deployment in limited environments.

Overall, the goal to integrate a Unity 3D development layer was achieved with this project. This has taken graphics to the current generation, improved processing power, and added additional capabilities for secondary biometric devices, like the Zephyr™ BioHarness™ (Medtronic Inc., Minneapolis, MN) and Xsens inertial measurement unit system (Xsens, Enschede, The Netherlands). The ease of development and port to secondary devices has improved ease of use and development for partners and made demonstration devices easier and more deployable overall. This has modernized the CAREN system and retained DoD's investment in this platform.

REFERENCES

Brungart, D., Kruger, S., Kwiatkowski, T., Heil, T., Highland, K. B., Cohen, J., Kokx-Ryan, M.. Schurman, J., Zaleski-King, A., & Zion, D. (2019). The effects of blast-related neurotrauma on aurally aided visual search while standing and walking. *Journal of Neurotrauma, 36*(16), 2443–2453. https://doi.org/10.1089/neu.2018.5630

Perrott, D. R., Cisneros, J., Mckinley, R. L., & D'Angelo, W. R. (1996). Aurally aided visual search under virtual and free-field listening conditions. Human factors, 38(4), 702-715.

Gottshall, K. R., & Sessoms, P. H. (2015). Improvements in dizziness and imbalance results from using a multi disciplinary and multi sensory approach to Vestibular Physical Therapy – a case study. *Frontiers in Systems Neuroscience, 9*, 106. https://doi.org/10.3389/fnsys.2015.00106

Sessoms, P. H., Fraser, J. J., Bodell, D. M., Haluch, K. S., Markham, A. E., Brookfield, J. S., Jameson, J., & Gottshall, K. R. (2021). Clinical effectiveness of virtual reality versus conventional clinic-based vestibular physical therapy on balance and function in active duty service members. A pilot randomized controlled trial. *Virtual Reality*, 1–14. https://doi.org/10.1007/s10055-021-00546-x

Sessoms, P. H., Gottshall, K. R., Collins, J. D., Markham, A. E., Service, K. A., & Reini, S. A. (2015). Improvements in gait speed and weight shift of persons with traumatic brain injury and vestibular dysfunction using a virtual reality computer-assisted rehabilitation environment. *Military Medicine, 180*(Suppl 3), 143–149. https://doi.org/10.7205/MILMED-D-14-00385

Xie, B. (2013). *Head-related transfer function and virtual auditory display* (2nd ed.). J. Ross Publishing.

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*: 09-12-2022

2. REPORT TYPE: Technical Report

3. DATES COVERED *(From - To)*: JAN2019-DEC2022

4. TITLE AND SUBTITLE
Development and Implementation of an Enhanced Virtual Reality Development System Into the Computer Assisted Rehabilitation Environment (CAREN)

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER: 2019.0002

5e. TASK NUMBER

5f. WORK UNIT NUMBER: N1801

6. AUTHOR(S)
Jacob A. VanDehy, Vrajeshri Ordek, Harvey Edwards, Christopher Smalt, Rasmus Grunnet-Jepsen, Sarah Kruger, Pinata H. Sessoms

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Commanding Officer
Naval Health Research Center
140 Sylvester Rd
San Diego, CA 92106-3521

8. PERFORMING ORGANIZATION REPORT NUMBER: 22-90

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Commanding Officer
Naval Medical Research Center
503 Robert Grant Ave
Silver Spring, MD 20910-7500

Chief, Bureau of Medicine and Surgery
(MED 00), Navy Dept
7700 Arlington Blvd
Falls Church, VA 22042-5113

10. SPONSOR/MONITOR'S ACRONYM(S): BUMED/NMRC

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A. Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
Do not cite, quote, or release until publication.

14. ABSTRACT
This report details the Department of Defense, Naval Health Research Center (NHRC), and partners' development and testing of an updated gaming-based system using the Unity system for simulation development and hardware integration with one such platform: the Computer Assisted Rehabilitation Environment (CAREN), a large-scale immersive virtual reality system. From 2017 to 2019, engineers from NHRC, National Intrepid Center of Excellence (NICoE), and Massachusetts Institute of Technology Lincoln Laboratory (MIT LL) collaboratively developed a complete hardware-software system that integrates seamlessly with existing simulation systems and utilizes the current standard for simulation software development to create a development platform that is robust, flexible, and meets the needs of an evolving military mission.

15. SUBJECT TERMS
augmented reality, virtual reality, Computer Assisted Rehabilitation Environment

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
UNCL	UNCL	UNCL	UNCL	16	Commanding Officer
					19b. TELEPHONE NUMBER *(Include area code)* COMM/DSN: (619) 553-8429

Reset