The Software-Based Cyber-Physical Interface for ICS/SCADA: Delivering High Quality Cyber Training, Testing, and Mission Rehearsal using Gaming Interfaces

Scott Thompson
CACI
Clarksville, TN
scott.thompson@caci.com

Rembrandt Bukowski
CACI
San Antonio, TX
rembrandt.bukowski@caci.com

ABSTRACT

Cyber training, testing, and mission rehearsal for Operational Technology (OT) is quite different than traditional IT because physical representations of industrial processes (known as a cyber-physical interface) are needed to convey the status of an Industrial Control System (ICS). Cyber-physical interfaces are quite challenging for organizations to build. Industrial equipment is expensive and requires a lot of floor space, demanding high costs for the initial build and reoccurring costs associated with leased facilities. Large cyber-physical interfaces demand dedicated technicians to maintain them and require routine and corrective maintenance. High-energy systems present significant safety challenges. Users often need to travel to the site of the cyber-physical interface, increasing travel costs and presenting roadblocks when travel is restricted because of pandemic concerns.

These challenges can be addressed by offering cybersecurity researchers and trainees with a completely software-based cyber-physical interface that provides a visualization of the physical process and reacts in a realistic manner to changes in the ICS/SCADA system.

A software-based cyber-physical interface is a scalable, modular platform that offers users the ability to model any industrial environment they desire. Training or testing audiences can be anywhere in the world, remotely connected with the ICS/SCADA environment and seeing the impact of cyber effects against cyber-physical systems using a web-based video feed served by the software. Software-based cyber-physical interfaces provide an engaging, immersive experience for the user through vivid and realistic representations of physical systems that can be delivered through 3D video or through AR/VR headsets. The software-based nature of the system allows organizations to deliver cost-effective remote training, testing, or mission rehearsals when travel is restricted by COVID-19.

ABOUT THE AUTHORS

Scott Thompson is a Program Manager from CACI specializing in Industrial Control Systems, Operational Technology, and visualization using gaming engines like Unity. He was awarded a MS degree in Cyber Forensics from Stevenson University. Scott has worked for CACI for approximately 5 years. He is also a retired Navy Lieutenant Commander with 25 years of service. His most notable projects include the “Bricks-in-the-Loop” ICS/SCADA training environment and the Information Warfare Integration Platform (IWIP) for the United States Air Force. Both projects rely heavily on physical models and Unity visualization to make for a more meaningful training experience. Scott is also the Program Manager for CACI’s City Block Research and Development project that combines ICS/SCADA with Unity visualizations.

Rembrandt Bukowski is an Electrical Engineer at CACI. He obtained his MS degree in Electrical Engineering from The University of Texas at San Antonio, where he researched renewable energy and Super Critical Carbon Dioxide integration into the electrical grid. Over his career he has developed innovative modelling and simulation systems for DoD training and exercises that replicate Industrial Control Systems (ICS) and Operational Technology (OT). He helped develop CACI live, virtual, and constructive information warfare systems that integrate various simulation platforms and physical entities into a single visualization of the battlefield for training and exercise audiences.

The Software-Based Cyber-Physical Interface for ICS/SCADA: Delivering High Quality Cyber Training, Testing, and Mission Rehearsal using Gaming Interfaces

Scott Thompson
CACI
Clarksville, TN
scott.thompson@caci.com

Rembrandt Bukowski
CACI
San Antonio, TX
rembrandt.bukowski@caci.com

INTRODUCTION

ICS/SCADA stands for Industrial Control Systems and Supervisory Control and Data Acquisition and deals with a set of computer technologies that control much of the world's critical infrastructure such as electrical power grids, wastewater systems, and oil refineries. The cybersecurity threats against American ICS/SCADA have grown considerably over the last few decades, putting the nation's critical infrastructure at risk. Government has an urgent need to understand ICS/SCADA technologies. Military organizations that depend on industrial technologies must determine the best methodologies to protect ICS/SCADA networks from cyber-attacks. The DHS is the government's leading department for protecting the 16 sectors of critical infrastructure in the United States. Many of these sectors utilize ICS/SCADA technologies that can be exploited by hostile and foreign actors. The DoE has a similar role as the DHS but focuses on the electric power grid and the safe delivery of nuclear power.

Current modelling and simulation systems that have been built for ICS/SCADA training and testing are large and cumbersome. They are often full-size models of electric power grids or other industrial systems that are expensive to operate and can cost millions of dollars to build and maintain. Government have strived to virtualize its training solutions for both cost savings and accessibility. ICS/SCADA training environments have also been part of this virtualization, but they normally lack cyber-physical interfaces or high-resolution visualizations. This lack of visualization deprives the training audience of understanding the consequences of security failures in an ICS/SCADA network. For instance, a trainee never fully appreciates the impact an attack against ICS/SCADA can have on a military mission because no visualization exists that shows effects to the targeted ICS or the second and third order effects to systems supported by it.

The objective of this research and development effort is to combine ICS/SCADA with a Unity software cyber-physical interface to create a state-of-the-art ICS/SCADA evaluation, training, and mission rehearsal environment. Unity software enables developers and engineers to create and operate real-time, interactive 3D content. By offering a software-based cyber-physical interface built in Unity, audiences receive immediate feedback on effects to the entire system when a cyber-attack is executed against an ICS/SCADA network. Software-based cyber-physical interfaces eliminate the need to invest in large system infrastructures and the large facilities that house them. When Unity is used as a cyber-physical interface, customers gain the ability to greatly scale the systems controlled by ICS/SCADA. The Unity based cyber-physical interface allows for rapid re-configuration of ICS/SCADA systems, allows for portability, and allows for easy remote network access to the ICS/SCADA range and its supporting cyber-physical interface using the internet or a private network.

MOVING FROM PHYSICAL SYSTEMS TO SOFTWARE

The development team's first cyber-physical interface was built using LEGO™ building sets with installed 5vdc lights. LEGO™ DC motors were used to create movement of components within the model. The team created three initial scenarios simulating power distribution and public safety systems. The LEGO™ models were powered with a 5vdc source and connected to Programmable Logic Controllers (PLCs) using wiring harnesses. A PLC is a small, modular solid-state computer with customized instructions for performing a particular task that are used extensively in ICS/SCADA systems. Users of the system received immediate feedback in the LEGO™ model when cyber effects were executed against the ICS/SCADA system (see *Figure 1*).

In January 2020, the team transitioned from exclusively using a LEGO™ model to using both a LEGO™ model and a Unity 3D scene based on the LEGO™ model as a cyber-physical interface. Both the LEGO™ model and the Unity model demonstrated changes in the ICS/SCADA system in the same manner and at the same time. The Unity 3D environment depended on a translator that polled the ICS/SCADA system PLCs in real-time and then translated the system status into information that Unity used to replicate system states.

The ICS/SCADA system for this development project was maintained in a 14U Pelican case. The Pelican case included a 10U DIN rack which provided support for several Raspberry Pi computers running OpenPLC software and a myBOX hardware/software Human Machine Interface (HMI). An HMI allows operators to monitor the status of the industrial process and can send commands to a PLC. The DIN rack also housed a 24vdc power supply needed for some of the ICS/SCADA components and a 5vdc power supply needed for the LEGO™ model.

A network switch and a server were housed in the Pelican case. The server maintained a full Information Technology (IT) network that interfaced with the ICS/SCADA network and assisted with testing NIST 800-82R2 ICS/SCADA configurations. It helped users understand how IT and ICS/SCADA infrastructures interact with each other. The Unity software cyber-physical interface was hosted on a high-end, 3U, rack mounted computer system that was transported in a 4U Pelican case. The computer acted as the primary control interface for the Unity software. When Virtual Reality was desired by the user, the system was equipped with a VIVE Pro VR headset. Steam VR software and the Unity game engine were both hosted on this 3U computer as well as the Unity middleware. The middleware was proprietary CACI software, built and maintained by CACI engineers and interns. The middleware made it possible for the Unity software to demonstrate real-time changes in the ICS/SCADA environment.

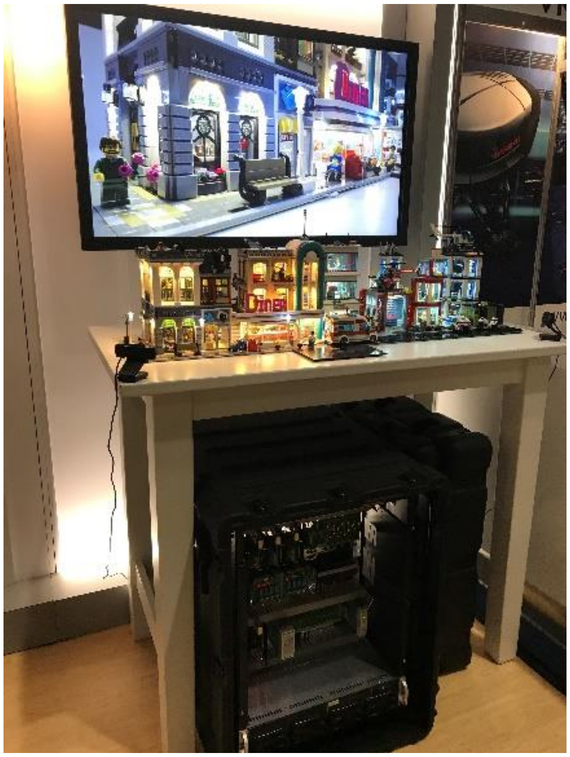

Figure 1. CACI's "City Block" displayed at I/ITSEC 2019. City Block is based on this research project.

MIDDLEWARE APPROACH

Development started with the creation of middleware between the ICS/SCADA system and the Unity cyber-physical interface. This middleware translated a broad selection of ICS/SCADA protocols into data that could be used in Unity applications. *Figure 2* illustrates the translation of Modbus/TCP data received in a traffic light during two different states. Modbus/TCP is a common network protocol that is used across a wide range of ICS/SCADA devices. The first state is when the Main St. light is green and the second is when the Main St. light is yellow.

The state of the traffic light is received through Modbus/TCP polling and the middleware begins building a message for the Unity software. The data is received in binary and is padded with bits as necessary to convert the binary data to hexadecimal data. A tag is added to identify the control system from which the data was received. After the message building process is complete, it is sent using a UDP datagram to Unity. The middleware polls all the PLCs in the ICS/SCADA system using Modbus/TCP protocol every second for the status of the industrial systems.

The team chose to send messages from the middleware to the Unity software using UDP datagrams because UDP can be broadcast to many different locations at the same time. This allows many different Unity environments to be supported by the same middleware, creating a multi-user training environment if desired.

Once unity receives the datagram, it is deciphered, and action is taken on the proper system in the Unity cyber-physical environment. The Unity team created a structure for parsing the messages received through UDP and distributing the data to the proper destinations. Unity expects to receive an update on system status once per second. The team categorized the messages as follows:

- System type (Hangar, Water Treatment, Runway, etc.)
- System index (Hangar 0, 1, or 2…)
- System's updated states (a list of on/off Booleans)

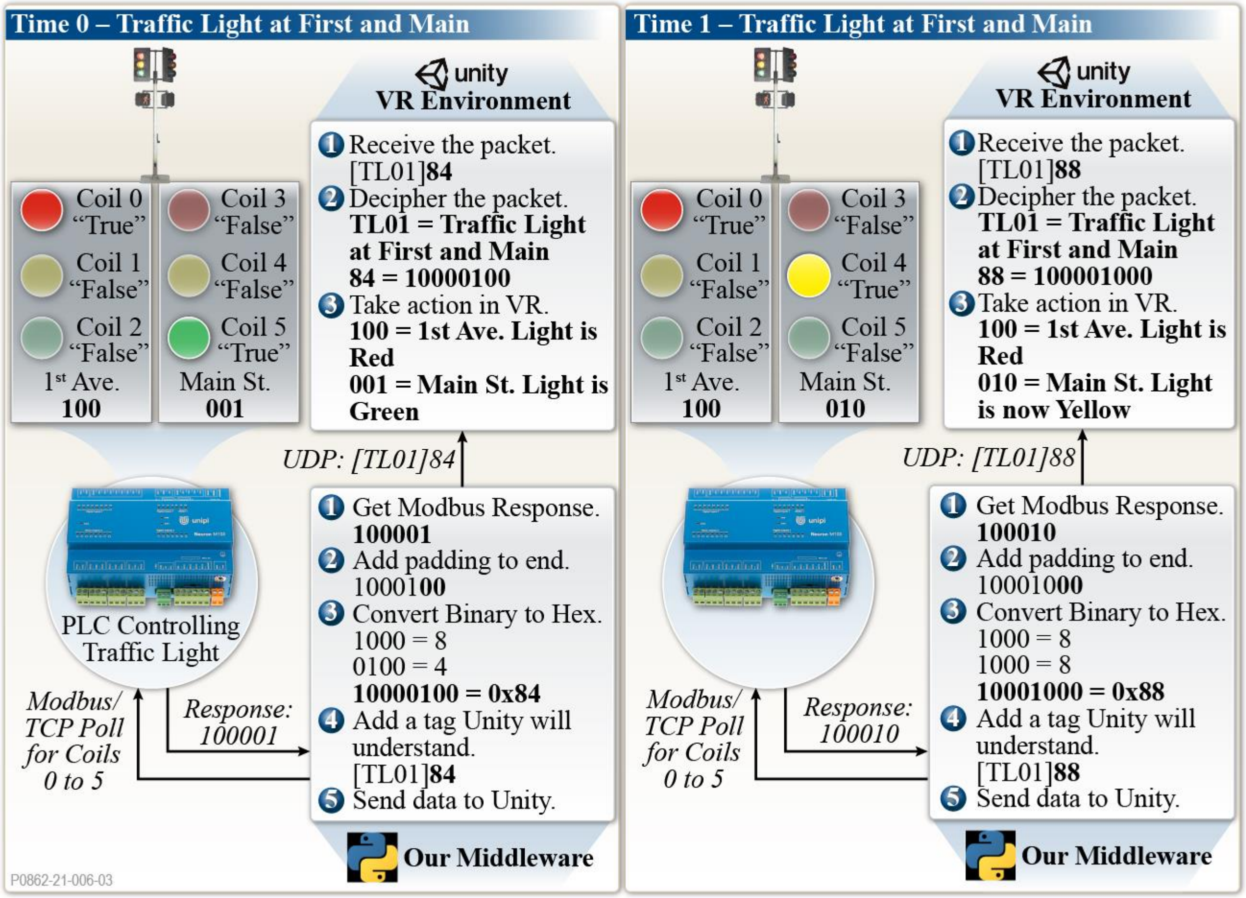

Figure 2. An example of CACI middleware developed for this project. In this case, Modbus/TCP protocol is translated into data that Unity understands.

Figure 2 shows how Unity accomplishes the parsing of data to create a visualization of the traffic light by distributing the data to the correct system. At time 0, the Unity VR environment receives a datagram containing the data "[TL01]84". "TL" is the system type of "Traffic Light". "01" corresponds with the traffic light system index of 01 corresponding with the traffic light at Main Street and 1st Avenue. The data "84" is converted back to binary, with the message "1 0 0 0 0 1 0 0". Unity will understand that only the first six bits will be used for a traffic light. The first three, "1 0 0" will correspond respectively with the red, yellow, and green lights on 1st Avenue. At time 0, the red light sets the first bit to "1" or True. The second three bits, "0 0 1" will correspond respectively with the red, yellow, and green lights on Main Street. At time 0, Main Street has a green light. Both lights are rendered properly in the Unity software environment at the intersection of Main and 1st.

BUILDING A SOFTWARE-BASED CYBER-PHYSICAL INTERFACE

The Forward Operating Base

The team wanted to demonstrate the scalability of the Unity-based cyber-physical interface by using a large model. A military Forward Operating Base, or FOB, was selected for this purpose. The FOB was also selected because it is an environment that is readily understood by military and government users. In addition to having a very large geographic area, the industrial systems supporting the base are very large. Presenting a traditional cyber-physical system for each industrial system on the base would be extremely expensive and would require a very large facility. The cost savings of using a Unity-based cyber-physical interface were immediately apparent.

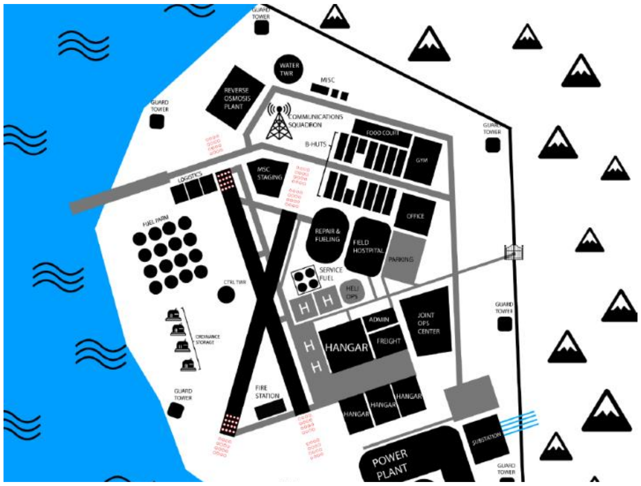

Figure 3. A map of the FOB as it was built in Unity.

The layout of the Forward Operating Base (FOB) included a hanger, power plant, fuel farm, and other infrastructure as depicted in *Figure 3*. To texture the runway without UV unwrapping the mesh, the Unity development team used a tri-planar shader created in Unity's Shader Graph. The functionality of the Shader Graph node editor to quickly create visual effects was the primary reason the team decided to use Unity's High-Definition Render Pipeline. This allowed a material to tile across a mesh using world space coordinates without texture stretching across its faces, making the Unity environment far more efficient. The shader also supported vertex color blending. This allowed the team to quickly "paint" other textures onto the runway like dirt or skid marks.

The team needed to render lights across the FOB, which included a few hundred runway, taxiway, mast, and approach lights that can be switched on or off at any time. *Figure 4* shows the extent of the lighting across the FOB. To achieve the large number of lights, the developers had to override Unity's maximum punctual light limit. Performance was optimized by creating lower level of detail (LOD) versions of the lights that can be dynamically switched at further distances from the camera

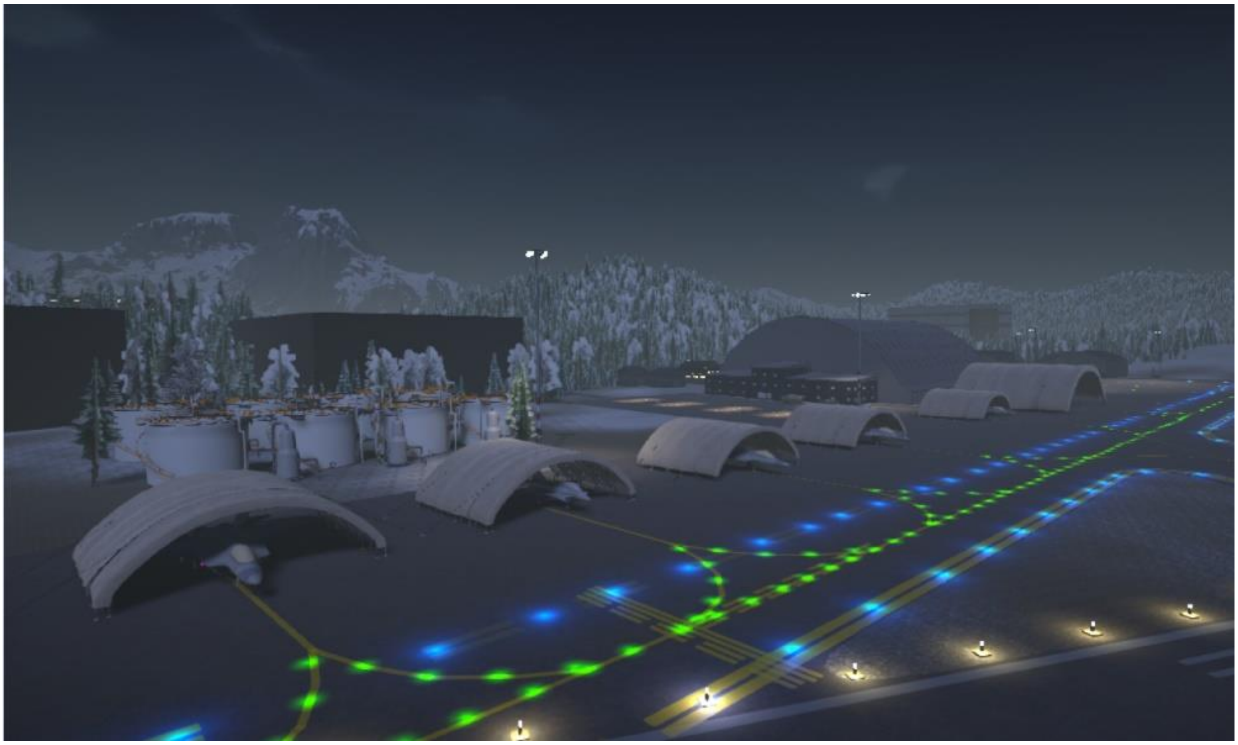

Figure 4. This picture of the runway and taxiway area of the FOB demonstrate the large number of lights needed for the Unity environment.

Airfield Lighting Control System

The first SCADA system, the Airfield Lighting Control System (*Figure 5*), was designed to mimic control systems that one would expect to find on a military airfield. The airfield lighting in the VR system, which included taxi, runway, and approach lighting, reacted to changes in the SCADA system using the same middleware strategy described in *Figure 2*. The runway lighting system also had dependencies with the base power system, where interruptions in base power are reflected within the Airfield Lighting Control System.

The electric substation model supplied primary power to the multiple major infrastructures inside the ICS/SCADA environment, driving almost all the SCADA simulations. The electrical and fire suppression systems in the FOB's hangar were designed to function based on power being supplied from the FOB's electric substation. Any aspect of the system which cannot obtain power from other means (i.e. battery power, valves, etc.) relies on power from the substation to function properly.

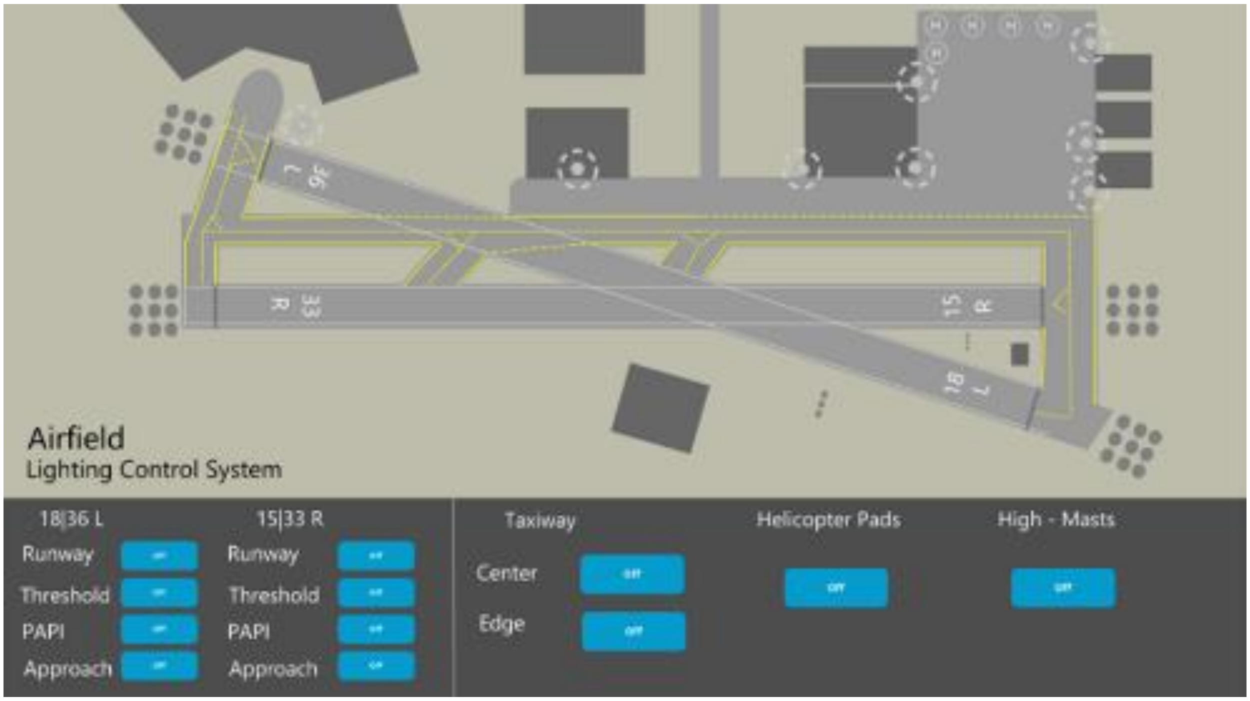

Figure 5. Realistic ICS/SCADA systems for the FOB. This HMI was designed for the Airfield Lighting Control System.

Water and Fuel Systems

The FOB included two ICS/SCADA systems that utilized tanks and pumps for their critical infrastructure. The first was a water treatment center that received wastewater from the base and processed it into potable water. The treatment center was driven by a python script that simulated the parameters of the process. The script drove tank levels and process values on a set of PLCs based on the position of valves and the status of pumps. The water treatment center was controlled from an HMI that was monitored by users of the training environment. The second was a simulation of the FOB fuel farm. In the fuel farm simulation, pumps were used to transfer fuel from storage tanks in a fuel farm to service tanks in a facility closer to the flight line. The FOB fuel farm simulation was controlled by a similar python script as the water treatment center.

The level of the water and fuel tanks was monitored in the Unity environment in two different ways. The user could use the HMI to see the tank value, or they could look at the tank itself. For the latter method, the tank level was painted on the tank using a different color. For instance, if a fuel tank was 50% full then half of the tank would be colored green. The developers also created a set of consequences for tanks that overfill. When tanks were in a high-level alarm status, their tank level would flash red. Fuel tanks that overfilled would spill over and had a chance of causing an explosion if they were left unattended. In cases where aggressor or red teams were used for ICS/SCADA cybersecurity events, the explosion provided a means of causing mission disruption on the base.

Electric Power and Fluid Flow

The Unity development team wanted to give the training audience feedback on the status of piping systems and electric power lines. Pulsating colors were used to indicate that an electric power line was energized or that a fuel or water pipe was part of the fluid flow path. The "open" or "shut" status of valves was drawn directly above the value in the Unity FOB. The status of emergency generators as "on" or "off" was prominently displayed above the generator. When an emergency diesel generator started, the generator exhaust would be dark but would clear over time as the fuel-air mixture became less rich. This animation made the start sequence look more realistic.

Joint Operations Center

A parallel effort to the Unity FOB was the creation of a "Joint Operations Center" or JOC (Figure 6) where the user could monitor all SCADA systems throughout the FOB. The JOC was a central place where the user could monitor various stationary camera feeds from around the FOB and Unmanned Air System (UAS) feeds that showed a bird's eye view of the base. Initially, the JOC was a very simple white box building that had just one monitor inside. The monitor displayed a camera feed of the backside of the building. This was a proof-of-concept to demonstrate that the Unity developers could create a security system with cameras that functioned realistically and recorded in real-time what was happening throughout the FOB. The core technology behind this system was enabled by render textures, which mapped the output of in-scene cameras to other objects in the scene.

After a successful proof of concept, the team built multiple camera feeds for the JOC. The developers created a UI canvas for the monitor with one button to toggle between camera feeds and then added several cameras throughout the prototype scene. The JOC could either be used in Virtual Reality mode using a Vive Pro headset and paddles or in 3D mode where the user uses a monitor, a mouse and a keyboard. Unity developers wrote a script that, upon the press of the button, cycles through render materials (generated from render textures) for the monitor. In other words, the button was switching between camera feeds in real time.

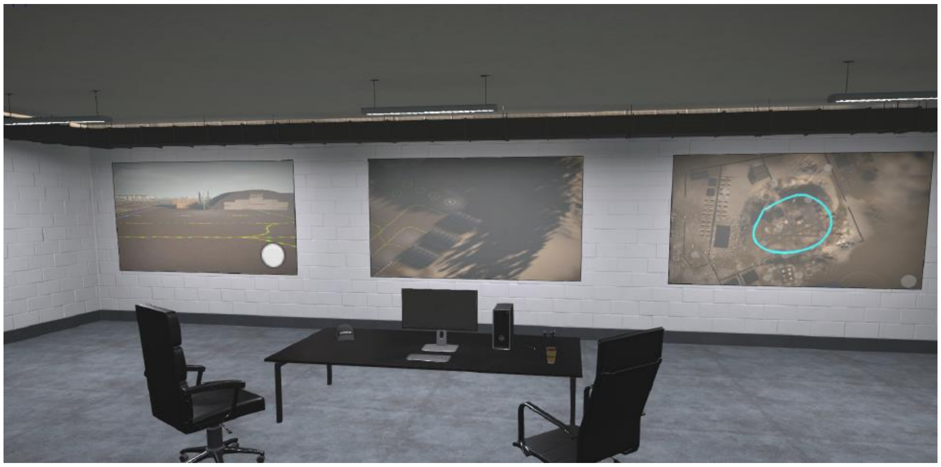

Figure 6. The Joint Operations Center (JOC) with multiple camera feeds.

Another visualization challenge was the Unmanned Aerial System (UAS) that monitored portions of the FOB from above. The first iteration of the UAS was a commercial UAS with a camera attached to its underside that was downloaded from the Unity Asset Store. The development team set up a series of nodes (bright red spheres whose transforms were accessed by the UAS) throughout the FOB scene that functioned as the destination points for the UAS and another screen that displayed the UAS camera feed. At runtime, the UAS would linearly interpolate (LERP) between those points and project its overhead view to the JOC.

As the team continued to research Unmanned Air Systems, they realized that commercial UAS and military UAS function in vastly different manners. Whereas commercial UAS have rotors that generate vertical propulsion for liftoff, can hover at a fixed location, and move quite slowly, military UAS typically require runways for takeoff and are constantly moving with horizontal propulsion. The developers decided that linear interpolation was not sufficient to emulate the kind of movement pattern that the FOB needed. The developers needed to generate curves that the

UAS could follow in a way that would reflect the momentum and handling of a military UAS. The team used a Unity asset pack to generate Bezier curves to simulate an aircraft taking off and landing at the FOB airstrip. The UAS flight path at this point looked realistic and paralleled real-world military UAS aerodynamics.

The last major mechanic that needed to be implemented was dynamic pathing to control the UAS flight path at runtime. Previously it was a hardcoded path, which is how the Bezier curves plugin was designed to be used.

CONCLUSION

The development team envisioned state-of-the-art ICS/SCADA training and testing environments through the construction of robust Unity-based cyber-physical interfaces that represented systems controlled by ICS/SCADA and replaced their cumbersome and expensive physical counterparts. The team started towards their goal with the LEGO™ environment, but then continued the quest by building a software-based cyber-physical interface that was far more scalable. The Unity environment allowed developers to build very large cyber-physical interfaces without the need for industrial hardware or excessive amounts of floor space. The team's developed environment is easily accessible from anywhere in the world using the internet and provides an inexpensive cyber-physical environment for superior ICS/SCADA training and education in cyber defense. This worldwide accessibility is particularly important during times of pandemic, when remote work and training becomes the norm.

FUTURE RESEARCH

The FOB developed by the research and development team has been demonstrated to and used by a variety of different customers throughout the Department of Defense. Feedback from those customers is driving the team's future efforts.

Playback

Although the live environment is rich and engaging, there is a need to be able to playback events for training and exercise audiences during debriefs. This is particularly important when dealing with audiences that lack the technical prowess to understand Tactics, Techniques, and Procedures (TTP) associated with cyber operations, when the visualization of cyber effects is needed to understand the impact on the battlefield. The team is utilizing an SQL database to store values rather than sending them straight to Unity using a UDP packet. Unity can maintain a live picture by selecting the most current entry from the SQL database. The audience can choose to playback a certain period by entering their time selection through the Unity Graphic User Interface (GUI). When playback is desired, Unity selects the proper entries from the SQL database as it follows the timeline selected. The entries inform the status of the models in the environment.

Additional SCADA Protocols

The research team began by using Modbus/TCP protocol because there was plenty of open-source material and software and because Modbus/TCP is ubiquitous across SCADA networks worldwide. There are many other SCADA protocols that merit future development for the FOB, but the research team is focusing on two. BACnet/IP is used universally in Building Automation Systems (BAS) such as lighting and HVAC. DNP3 protocol is known throughout the United States for its usage in both power and water systems.

ACKNOWLEDGEMENTS

This project has been supported over the last three years by a talented team of CACI employees and interns. Aaron Frost (Oregon State University) and Matthew Troutman (University of Utah) are two all-star Unity programmers that were instrumental in getting the Unity Forward Operating Base established. Emily Nguyen and Sana Wakko (University of Texas, San Antonio), Omar Nayfeh (Texas A&M, San Antonio), Christopher Dieck, Raymond Downing, Jacob Bryant, Carter Moore, Oler Dudley III, Husam Saoudi, and Nancy Hite (CACI) has all been important contributors.

REFERENCES

Akelian, C. J. (2015). Incorporating SCADA modules into Introductory Programmable Logic Controller Curriculum. *Proceedings of the ASEE Annual Conference & Exposition*, 1–14.

Baron, David. (2021). *Game Development Patterns with Unity 2021: Explore practical game development using software design patterns and best practices in Unity and C#.* (2nd Ed). Packt Publishing.

Bodungen, C. E., et al. (2017). *Hacking Exposed Industrial Control Systems ICS and SCADA Security Secrets & Solutions.* (1st Ed). McGraw Hill Education.

Faryal, A., Umer, F., Amjad, M., Rashid, Z., & Muhammad, A. (2021). Modelling and Simulation of SCADA and PLC System for Power System Protection Laboratory. *Electrical, Control & Communication Engineering*, *17*(1), 19–25. https://doi.org/10.2478/ecce-2021-0003

NIST SP 800-82R2. (2015). NIST Special Publication 800-82 Revision 2. Retrieved June 28, 2022, from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf

Santoso, H. B., Baroroh, D. K., & Darmawan, A. (2021). Future Application of Multisensory Mixed Reality in the Human Cyber-Physical System. South African Journal of Industrial Engineering, 32(4), 44–56. https://doi.org/10.7166/32-4-2551

Thomas, M. S., Kumar, P., & Chandna, V. K. (2004). Design, Development, and Commissioning of a Supervisory Control and Data-Acquisition (SCADA) Laboratory for Research and Training. IEEE Transactions on Power Systems, 19(3), 1582–1588. https://doi.org/10.1109/TPWRS.2004.826770

Wei, X., Zhang, Y. J., Toshniwal, D., Speleers, H., Li, X., Manni, C., Evans, J. A., & Hughes, T. J. R. (2018). Blended B-spline construction on unstructured quadrilateral and hexahedral meshes with optimal convergence rates in isogeometric analysis. *Computer Methods in Applied Mechanics & Engineering*, *341*, 609–639. https://doi.org/10.1016/j.cma.2018.07.013

This material consists of CACI International Inc general capabilities information that does not contain controlled technical data as defined within the International Traffic in Arms (ITAR) Part 120.10 or Export Administration Regulations (EAR) Part 734.7-10. (EXIM ID 5980)