TRAILBLAZING ROBLOX VIRTUAL SYNTHETIC TESTBED DEVELOPMENT FOR HUMAN-ROBOT TEAMING STUDIES

Felix R. Raimondo[a], Alexandra T. Wolff[a], Alexander J. Hehr[a], Matthew A. Peel[a], Margaret E. Wong[a], Erin K. Chiou[a], Mustafa Demir[b], Nancy J. Cooke[a, b]
[a] Human Systems Engineering, Arizona State University, Mesa, Arizona, USA.
[b] Center for Human, Artificial Intelligence, and Robot Teaming, Global Security Initiative, Arizona State University, Tempe, Arizona, USA.

Virtual testbeds are fundamental to the success of research on cognitive work in safety-critical domains. A testbed that can meet researchers' objectives and create a sense of reality for participants positively impacts the research process; they have the potential to allow researchers to address questions not achievable in physical environments. This paper discusses the development of a synthetic task environment (STE) for Urban Search and Rescue (USAR) to advance the boundaries of Human-Robot Teams (HRTs) using Roblox. Virtual testbeds can simulate USAR task environments and HRT interactions. After assessing alternative STE platforms, we discovered Roblox not only met our research capabilities but also would prove invaluable for research teams without substantial coding experience. This paper outlines the design process of creating an STE to meet our research team's objectives.

INTRODUCTION

Humans and robots are seeing more use operating as a team in numerous fields and contexts, especially as urban search and rescue tools in response to environmental catastrophes (Burke et al., 2004; Murphy, 2004; Murphy, 2019). Human Robot Teams (HRTs) emphasize the use and inclusion of machine agents (e.g., AI-enabled robots) with advanced decision-making capabilities to be considered as a teammate (Demir et al., 2017). Synthetic Task Environments (STEs) can be used to study how future HRTs can work together without waiting for the development of a physical robot. STEs bridge the gap between controlled laboratory tasks and uncontrolled field studies, allowing for the ability to systematically study the complex team interactions that may take place (Cooke & Shope, 2004).

STEs are especially useful for studying how humans and robots can team together in complex and high-stakes work environments that may be extremely difficult or not yet possible to test in the real world. One example of such an environment is in the field of urban search and rescue (USAR) (Murphy, 2014). Rescue robots have the ability to navigate dangerous environments, mitigating the risk exposed to rescuers. Current rescue robots are advanced computers teleoperated by humans, and as these tools develop more capabilities and autonomy, research is needed on the abilities of future HRTs. The impacts of team performance, trust, situation awareness, and workload in a team of humans and autonomous robots require additional research (Nourbakhsh et al., 2005).

Previously, Minecraft was used to simulate a USAR testbed that studied team performance of robot communication strategies (Bartlett & Cooke, 2015; Lematta et al., 2019). It was found that information exchange from robots needed to balance explanations and transparency to increase team trust and team performance (Chiou et al., 2021). Minecraft studies have successfully studied different HRT strategies (Chiou et al., 2021; Ezenyilimba et al., nd); however, because of a lack of visual quality and feature complexity that was limited by the Minecraft USAR environment, we considered developing an alternative USAR using the Roblox gaming platform.

EXPERIMENT OVERVIEW

The STE in this paper was developed to study the effects of transparency and explanations in HRTs within a USAR virtual synthetic task environment. These studies use the Wizard of Oz (WoZ) paradigm, where participants are led to believe it is an AI-enabled autonomous agent and told after an experimenter virtually controlled the teammate (Kelley, 1984; Riek, 2012). To effectively implement the WoZ methodology, participants must have no indication that the robot they are interacting with is a human. This design allows researchers to design, deploy, and observe human-robot interactions without the time and cost constraints of developing or obtaining a real agent.

For the current study, the research investigates how different presentations of robot communication (text versus graphics) and the presence of confidence information impact trust, workload, situation awareness, and performance in HRTs (Peel et al., 2021). The study was designed to have experimenters playing the role of a robot, dubbed the *Battery Rescue Autonomous Vehicle Explorer* (BRAVE), and the role of an *incident commander*. Participants are onboarded to perform the role of *navigator* within the team. As navigator, the participant is responsible for routing the robot through the USAR environment, allocating medical resources, and reporting the victim's location. BRAVE and the IC do not communicate directly; all communication is filtered through the navigator. BRAVE and the navigator communicate through text-chat only, and the IC and navigator communicate verbally only. To accomplish this design, the research team determined that the STE needed to accommodate certain "behind the scenes" actions. The study design led the research team to consider alternatives to Minecraft as the proposal involved an entirely new STE, and leveraging the newest technology that would also be valuable in future studies was imperative. Additional versatility of the STE was difficult to accomplish in

Minecraft. These new research requirements were met by the Roblox platform and confirmed the decision to create a new USAR STE.

VIRTUAL TESTBED LANDSCAPE

This paper considered approximately four viable platforms that inexperienced developers could use to create similar STEs: Roblox, Minecraft, Core, and Fortnite Creative (Core, 2022; Fortnite, n.d.; Mojang Studios, 2022; Roblox, 2022). Table 1 highlights critical features of each platform to develop an effective USAR-STE. These features are based on the experiences of past studies (Chiou et al., 2021; Demir et al., 2020; Ezenyilimba et al., nd) and the requirements of the proposed study (Peel et al., 2021). Publicly available documentation from the platforms and notable developers was used to determine the degree of capabilities of each feature across the platforms. As seen in Table 1, Roblox is the overall best platform among these features, and Roblox offers the greatest performance across the listed features. The study team has previously used Minecraft as an STE platform for HRT studies; however, it requires third-party solutions to create the desired features and visual fidelity that is native to Roblox.

Table 1. Platform assessment

	Roblox	Core	Fortnite Creative	Minecraft
Fidelity of Environment	Photorealism	High Fidelity	Cartoonish	Block-Based Design
3D Modeling	One Viewport Available		No Viewport Available (In-game modeling only)	
Developer Resources	Extensive	Limited	Minimal	Extensive
Programming Language	Lua		In-Game Features	In-Game Binary
Content Moderation	Strict		User-generated content limited	
Collaborative Editing	Yes, Server Logged Changes	Yes, Server Logged Changes & GitHub Integration	Partially, Limited Functionality	
Server Distribution	Self-Hosting Not Allowed		Self-Hosting Recommended	
Exporting Data	User-defined export connections	No		Game Generated Log Files
Future Platform Updates	All versions supported	Not Disclosed		Updates not supported

DESIGN PROCESS

The design process of the testbed can be summarized as three phases: (1) requirements, (2) construction, and (3) final development (Haik et al., 2017). The product concept, solution concept, and embodiment design parts of the design process were condensed into the "construction" phase due to our development time constraints.

The first phase, the requirements of a testbed, involves documenting the necessary components and creating a development plan; a development plan allows for greater coordination among the research team. The second phase, the construction of the testbed, implemented an outcome-oriented construction. The third phase, final development, reflects and validates the completed design process.

Requirements

Before starting the testbed design, we described the desired components of the testbed identified by the research team using the following dimensions: necessary (N), advantageous (A), and extraordinary (E). The testbed design was grounded in the discussion of the "necessary" components, which became the design requirements. In total, there were nine requirements which are summarized in Table 2. Table 2 highlights the necessary design requirements and explains the considerations for each factor.

Table 2. Design requirements

Requirement (Type)	Description
USAR Environment (N)	Increasing perceived engagement by participants in the USAR virtual task environment. Akin to flight and driving simulators, a more realistic STE improves its external validity (Mulder et al., 2013; Philips & Morton, 2015).
Two Mission Design (N)	Incorporating a two-mission design. The flexibility of incorporating two independent missions in sequence was essential for testing interactions over time, as well as allowing the team to test for different communication modalities within subjects.
Remote Collection (N)	Ability to collect all data remotely. Due to the uncertainty of COVID-19, it was necessary to build the STE with virtual capabilities.
WoZ Paradigm (N)	WoZ functionality that allows experimenters to control the robot and the environment. HRTs are still in their infancy and require that experimenters "play" the role of an intelligent agent. WoZ functionality has been standard in all the previous STEs developed.
Victim Development (N)	Ability to build human "victims" in the STE Finding victims is central to the USAR task, and it was a priority to be able to create more physically representative figures in the STE.
Unexpected Events (N)	Ability to introduce unexpected events. Unexpected events in the STE were needed to better represent real-world task environments as well as to stimulate more dynamic team interactions.
Live Run Study (A)	Ability to share virtual screen within STE without limiting capabilities. Participants experience the environment firsthand, interacting with it live.
GUI. Functionality (A)	Ability to incorporate a Graphic User Interface (GUI) and display it in real time. The STE needed to have a certain level of flexibility and fidelity for displaying graphical information (i.e., data visualizations) to address the research objective.
User-Defined Data Collection (E)	Ability to set parameters and automate the data collection from the testbed.

The seven necessity, two advantageous, and one extraordinary component created a difficult design task for our STE development. The four platforms were assessed in their capabilities to satisfy our requirements. Table 3 shows the abilities of the platforms to achieve the design requirements. Roblox is the only platform to meet all requirements, with Core following in second. Roblox allows for custom images, in-game communication, and a live experience of the environment. These tasks were necessary for the proposed research and further reinforced the decision to move to Roblox. Additionally, Roblox has the capability to connect to a web service and transmit any information offsite. This is an extraordinary feature for a platform to support and was exclusively seen in Roblox.

Table 3. Platforms capabilities of research requirements

Requirement	Roblox	Core	Fortnite Creative	Minecraft
USAR Environment (N)	✓	✓	✓	✓
Two Mission Design (N)	✓	✓	✓	✓
Remote Collection (N)	✓	✓	✓	✓
WoZ Paradigm (N)	✓	✓	✓	✓
Victim Development (N)	✓	✓	✓	
Unexpected Events (N)	✓	✓	✓	✓
Live Run Study (A)	✓			
GUI. Functionality (A)	✓	✓		
User-Defined Data Collection (E)	✓			

Outcome Oriented Construction and Coding

The design process implemented an outcome-oriented approach for development and coding. An outcome-oriented approach can be defined as the fixation on the end-state of a system rather than on the process of achieving it (Schnell, 2019). This approach was adopted for our STE as there were significant time constraints.

This section discusses the implementation of the requirements in planning, building, and coding. The first step was to design a layout of the two missions. Based on the requirements, the design had to be an urban disaster environment with two distinct floors or zones and allowed for the placement of victims. There was ample consideration of different environments to depict: offices, homes, public transportation, stadiums, and more. Hotels were determined to be the ideal framework to satisfy the urban floor requirements and victim placements. Hotels are associated with large group events and leisure, and it was essential that participants felt the environment was realistic in a USAR scenario.

The construction phase was the most intensive phase and relied heavily on the outcome-oriented process. The finalized floor layouts were built in Roblox. This process of constructing a realistic environment with victims was laborious, despite efforts being outcome-oriented. If the environment *looked* right, then it *was* right. This meant in-accessible areas were left empty, stairs were cut off at the earliest possible position, and elevators were facades without functionality. The process of construction relied heavily on real-life designs. The construction team used their best judgment to imagine the destruction and built an environment accordingly. Figure 1 compares a previous Minecraft design, the tragic Chaplain Tower Collapse in Miami, and the new Roblox design (Schaff, 2021).

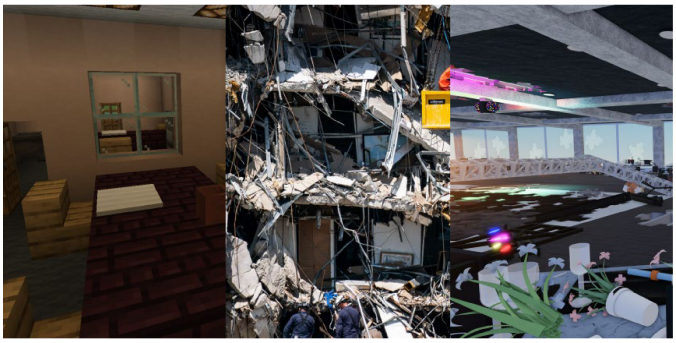

Figure 1. Fidelity Examples of Partially Collapsed Building from Left to Right: Minecraft (synthetic), Miami (real event), and Roblox (synthetic)

The result of the construction phase was the creation of a high-fidelity environment in a condensed time frame. This environment can be readily manipulated to change interior layouts and add different hazards. The current version has fire, electrical, gas, water, unstable floor, and collapse hazards. All hazards were built into the environment without software functionality. The coding process was simultaneous with construction.

The coding requirements were relatively straightforward, and the community support pages were utilized to accomplish them. Roblox has created a highly efficient platform in terms of coding functionality and dedicates much of the hardware capabilities to physics and rendering. The code in this STE is unnecessarily repetitive (as a function would have achieved the task) and as a result, changing one component requires changing it across many lines of code. A more efficient coding process would benefit the research team in terms of reduced coding time.

The planning process validated the required functionality within Roblox, which coordinated successfully with construction and coding operations. There were challenges faced that our team was able to overcome rapidly after development, thanks to the flexibility and functionality of the platform we chose. The final testbed resulted in the custom-built, usable, and engaging data collection method with the promise of rapidly designing and deploying future iterations.

Final Testbed

This section discusses the roles involved in the HRT study built into the testbed and the overall benefits uncovered. There are three roles in the testbed, representing the three members of the HRT: Incident Commander, (*Participant*) Navigator, and the Robot BRAVE. A researcher is also involved in onboarding the participant and debriefing the participant. There are 11 visual elements built in the testbed that display based on the

roles, and table 4 maps these visual elements to the roles that used them.

Table 4. Visual Elements for Specific Roles

Visual Elements	Participant	Incident Commander	BRAVE
Chat Window	✓	✓	✓
Time Remaining	✓	✓	
Medical Resources	✓	✓	✓
BRAVE's Medical Scan	✓	✓	✓
Map	✓		
Map of Victims		✓	✓
Timer Control		✓	
Camera Feed Control		✓	
Condition Control		✓	
Resource Control			✓
Position Control			✓

The participant works with BRAVE to save victims by properly allocating medical resources. The participant view can be seen in Figure 2, which shows; that the participant's view is mainly the camera feed of BRAVE. The top left of the screen is the chat window, allowing for live-run communication with BRAVE. The top-right is the status of available medical resources. The medical resources display shows three types of medical services – first aid kit, Emergency Medical Technician (EMT), and medical evacuation – and how many remain for each. In the center of the screen, the victim image appears when BRAVE approaches a victim and completes a scan—specific information in the image changes depending on the study condition. The bottom right is the dynamic map, where BRAVE's position is continuously updated. The Incident Commander is a fundamental teammate with external and internal experimental tasks.

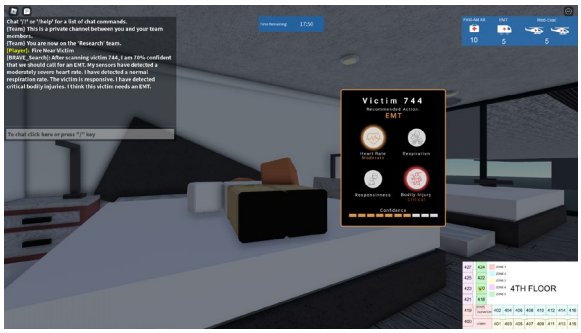

Figure 2. Screenshot of Participant View Example of Victim

The Incident Commander is a remote team member whose role is entirely stationary. The Incident Commander coordinates medical resources from the participant's recommendation.

BRAVE, or more precisely, the experimenter filling in the role of BRAVE, is responsible for the direct movement in the environment and scanning for victims. However, Participants/Navigators direct BRAVE's search path by asking for specific zones. The final testbed will continue to be implemented and adapted in future study designs.

DISCUSSION AND CONCLUSION

As this study and the final STE were an extension of previous research on HRTs in an USAR task environment, the research team successfully developed a new STE. Roblox was found to be the most viable option to achieve the research objectives within a limited timeframe, despite having to recreate a virtual environment from scratch.

The initial problem we faced was the lack of experience using Lua, the programming language used in game development within Roblox. The outcome-oriented development process resulted in the completed construction of a map layout, a 2-floor hotel building that experienced a natural disaster, and a research-ready testbed.

With Roblox's functionality and relative programming simplicity, this testbed will be of great use to future projects, particularly for teams without substantial coding background or experience. Coding functions are separated from visual representations, allowing ease of change in coding or environment design. This ability was a driving force behind the switch from Minecraft to Roblox. We hope that the functionality and fidelity that Roblox provided our team can also serve other HRT researchers and future USAR studies that need to select a platform to develop their testbeds.

Future Directions

The experience of this new STE was transformative in validating a new technology and trailblazing a path for future research. Roblox and its developer community appear to be an impactful tool and resource for research and HRT data collection. Through the development of this testbed, we noted features to consider in future studies. Roblox has the potential to run multiple experiments simultaneously; multiple independent (or interdependent) servers can effectively allow for hyper-data collection sessions.

There is also a potential for survey data collection through Roblox, affording participants not to leave the window and reducing the risk of incorrect study procedure sequences. One of the most remarkable features is the connection to external processes; similar to Roblox pushing information to Google Sheets; it can also pull information. Roblox can push information to Google Sheets, allow for the manipulation or calculation of information and pull real-time factors into the system's mechanics, resulting in a hyper-data collection system. An example of this could be observing a pattern of high-risk behaviors and creating an environment adaptation where the risk generated is proportional to the risk behavior detected.

Roblox is also trailblazing the development of virtual reality; it currently supports Oculus Rift, H.T.C. Vive, and Xbox control input for VR manipulation. Roblox's framework for VR is native, meaning even the current testbed supports VR; however, the design was not created for a VR scenario but easily could have been if decided earlier. The trajectory of Roblox's development is promising and should engage all researchers using Synthetic Task Environments.

Limitations

Roblox is strict on moderation; the platform is geared toward young children, so the tolerances are narrow. Any language, phrases, letters, or acronyms that can be considered inappropriate are censored in the chat. Additionally, the moderation levels confound with the lack of external servers. Roblox operates all of its servers and does not allow for the purchasing of private servers and deploying Roblox environments. This may raise concerns about potential participant data privacy or data ownership. This also creates a new reliance for the study team on external operations; Roblox can have a system issue and interfere with research. During development, Roblox experienced server crashes where our developers could not access the platform for several hours. These concerns are relatively unique to research applications leveraging Roblox software and collaborating more closely with Roblox Inc. could address these issues.

ACKNOWLEDGEMENT

This research was partially supported by a grant from the Air Force Office of Scientific Research [FA9550-18-1-0067]. This research would not have been possible without additional support; we appreciate the assistance of Madison Gilbert, Noella Campbell, and Shawaiz Bhatti.

REFERENCES

Bartlett, C. E., & Cooke, N. J. (2015). Human-Robot Teaming in Urban Search and Rescue. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 59(1), 250–254.

Burke, J. L., Murphy, R. R., Coovert, M. D., & Riddle, D. L. (2004). Moonlight in Miami: Field Study of Human-Robot Interaction in the Context of an Urban Search and Rescue Disaster Response Training Exercise. *Human–Computer Interaction*, 19(1–2), 85–116. https://doi.org/10.1080/07370024.2004.9667341

Chiou, E., Demir, M., Buchanan, V., Corral, C., Ensley, M., Lematta, G., Cooke, N., & McNeese, N. (2021). *Explanation based communication strategies in a simulated human-robot search and rescue team task.*

Cooke, N. J., & Shope, S. M. (2004). Designing a Synthetic Task Environment. In L. R. E. Schiflett, E. Salas, & M. D. Coovert (Eds.), *Scaled Worlds: Development, Validation, and Application* (pp. 263–278). Ashgate Publishing.

Core. (2022, February 22). *Play and Create Games for Free on Core*. Core Games. Retrieved February 22, 2022, from https://www.coregames.comDemir, M., McNeese, N. J., & Cooke, N. J. (2017). Team situation awareness within the context of human-autonomy teaming. *Cognitive Systems Research*, *46*, 3–12. https://doi.org/10.1016/j.cogsys.2016.11.003

Endsley, M. R. (2019, October). Explainable and Transparent AI. [Presentation to the HFES. Annual Meeting]. HFES. Annual Meeting, Seattle, WA.Ezenyilimba, A., Wong, M., Alexander, H., Demir, M., Wolff, A., Chiou, E. K., & Cooke, N. J. (n.d.). The impact of transparency and explanations on trust and situation awareness in human-robot teams.

Fortnite. (n.d.). *Fortnite Creative Documentation*. Epic Games. Retrieved February 22, 2022, from https://www.epicgames.com/fortnite/en-US/creative/docs

Haik, Y., Sivaloganathan, S., & Shahin, T. M. (2017). *Engineering Design Process* (3rd ed.). Cengage Learning.

Kelly, J. F. (1984) An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Office Information Systems*, *2*(1), 26-41. https://doi.org/10.1145/357417.357420

Lematta, G. J., Coleman, P. B., Bhatti, S. A., Chiou, E. K., McNeese, N. J., Demir, M., & Cooke, N. J. (2019). Developing Human-Robot Team Interdependence in a Synthetic Task Environment. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *63*(1), 1503–1507. https://doi.org/10.1177/1071181319631433

McNeese, N. J., Demir, M., Chiou, E. K., & Cooke, N. J. (2021). Trust and Team Performance in Human–Autonomy Teaming. *International Journal of Electronic Commerce*, *25*(1), 51–72. https://doi.org/10.1080/10864415.2021.1846854

Mojang Studios. (2022, January 21). *What is Minecraft?* Minecraft.Net. Retrieved February 22, 2022, from https://www.minecraft.net/en-us/about-minecraft

Mulder, M., Zaal, P., Pool, D. M., Damveld, H. J., & van Paassen, M. (2013). A Cybernetic Approach to Assess Simulator Fidelity: Looking back and looking forward. *AIAA Modeling and Simulation Technologies (MST) Conference*. 10.2514/6.2013-5225

Demir, M., McNeese, N. J., & Cooke, N. J. (2017). Team situation awareness within the context of human-autonomy teaming. *Cognitive Systems Research*, *46*, 3–12. https://doi.org/10.1016/j.cogsys.2016.11.003

Murphy, R. R. (2019). *Introduction to AI Robotics*. MIT Press.

Murphy, R. R. (2004). Activities of the rescue robots at the World Trade Center from 11-21 september 2001. *IEEE Robotics & Automation Magazine*, 11(3), 50–61. 10.1109/M.R.A.2004.1337826

Murphy, R. R. (2019). *Introduction to AI Robotics*. MIT Press.

Nourbakhsh, I., Sycara, K., Koes, M., Yong, M., Lewis, M., & Burion, S. (2005). Human-Robot Teaming for Search and Rescue. IEEE Pervasive Computing, 4(1), 72–78. 10.1109/mprv.2005.13

Philips, B. H., & Morton, T. (2015). Making Driving Simulators More Useful for Behavioral Research - Simulator Characteristics Comparison and Model-Based Transformation: Summary Report.

Riek, L. (2012). Wizard of Oz Studies in HRI: A Systematic Review and New Reporting Guidelines. Journal of Human-Robot Interaction, 119–136. https://doi.org/10.5898/jhri.1.1.riek

Roblox. (2022, February 22). *Home*. Retrieved February 22, 2022, from https://corp.roblox.com

Schaff, E. (2021, June 27). Rescuers continued their search for victims of the Champlain Towers South condo in Surfside, Fla., on Sunday [Photograph]. New York Times. https://static01.nyt.com/images/2021/06/27/us/27miami-building-collapse-briefing-rescue-explainer3/merlin_189987675_5478574a-27b0-4414-a9ac-882ffa9d169c-superJumbo.jpg?quality=75&auto=webp

Schnell, T. (2019, March 14). *Outcome-Oriented Thinking for Teams*. Tim Schnell. Retrieved February 23, 2022, from https://tims.io/outcome-oriented-thinking-for-teams/