From CAVE2[TM] to Mobile:
Adaptation of *Hearts and Minds* Virtual Reality Project Interaction

Arthur Nishimoto[1], Daria Tsoupikova[2], Scott Rettberg[3], Roderick Coover[4]

[1]Computer Science, University of Illinois at Chicago, IL, USA
`anishi2@uic.edu`
[2]School of Design, University of Illinois at Chicago, IL, USA
`datsoupi@gmail.com`
[3]University of Bergen, Bergen, Norway
`scottrettberg@gmail.com`
[4]Temple University, Philadelphia, PA, USA
`roderickcoover@gmail.com`

Abstract. Hearts and Minds: The Interrogations Project is an interactive performance made for the CAVE2[TM] [1] large-scale 320-degree panoramic virtual reality environment that describes veterans' testimonies about military interrogations in Iraq during the American counter-insurgency campaign. The project is based on interviews of American soldiers and on their actual testimonies [2]. The project was achieved through technical innovation, cross-disciplinary and international collaboration. It was developed using a novel method for direct output of the Unity-based virtual reality projects into the CAVE2 environment. Other portable versions of the work were developed to reach new audiences across educational, arts and public arenas which include 1) personal computer version navigable using Xbox 360 controller; 2) web-based version available for free download; 3) Oculus Rift immersive virtual reality HMD version; 4) mobile version of the project for Apple iPad (in progress). This paper describes the development and compares the interaction experiences across platforms.

Keywords: Virtual reality, navigation, interaction, immersive environments, 3D user interface, CAVE2[TM].

1   Introduction

The use of Virtual Reality (VR) in sciences, healthcare, arts and design is becoming more and more ubiquitous. The development of virtual art has been influenced by advancements of VR technology and media art movements that examined the concepts of interactivity, installation, immersion, interface design, responsiveness and storytelling. The history of virtual art has been defined by pioneering projects such as "World Skin" by Maurice Benayoun (1997), "Osmose" by Charlotte Davies (1995),

"The Legible City" by Jeffrey Shaw (1989), "Placeholder" by Brenda Laurel (1994), "Be Now Here" by Michael Naimark (1995) and many other important works [3]. Relying on major technological and artistic achievements, virtual art established new ways of making, viewing and understanding art through immersion, interaction and presentation inside the virtual space. With the advancement of technology, virtual art gradually moved out of research laboratories and scientific centers into galleries, museums and public exhibitions. The Oculus Rift, Leap motion, Google Glass, Kinect, HoloLens, Unity, and other virtual technologies additionally changed the way that contemporary VR art projects are planned and realized.

VR art has however been consistently limited by the fact that there are no cross-platform standards allowing seamless portability of Virtual Reality Environments (VREs) and interfaces between various domains and technologies. Preddy and Nance [4] argued for the need to establish a standardized API that would provide the ability of working on multiple levels of abstraction to support the portability of virtual environment interfaces. The development of the portable versions of the VRE for different technologies requires significant investments of time and resources. In addition, the authors often need to redesign complicated navigation interfaces from scratch and adjust interaction techniques for each platform to reach out particular target audiences.

Fig. 1. Presentation of the project using the Xbox controller at Litteraturhuset Oslo, Norway.

One of the goals of our project was to create a virtual environment with sufficient interaction complexity to show on several different VR platforms. To enable a more natural and user-friendly way of interacting with the virtual environment in the CAVE2, on the Xbox controller system running on personal computer, using the web-based Unity3D web player and on mobile device (iPad), we have adapted project's interactive interface to employ different interaction techniques. The efficacies of these techniques and interfaces were evaluated with participants during the several exhibitions of this project using informal qualitative methods (informal qualitative interviews and direct observations) [5, 6] (Fig. 1).

This is an ongoing project at the Electronic Visualization Lab (EVL) in Chicago that is being achieved through technical innovation, and cross-disciplinary international collaboration between artists, scientists, and researchers from five different universities. *Hearts and Minds: The Interrogations Project* was developed using a novel method for direct output of Unity-based virtual reality projects into the CAVE2 environment. The project premiered as the first virtual performance utilizing the Unity game engine in the CAVE2 environment. We are currently releasing the mobile version made for personal interaction on the iPad to enhance the project's accessibility for educational use.

2   Development and Technology

2.1   Project Concept

Hearts and Minds addresses a complex contemporary problem: as American soldiers are returning from the wars in Iraq and Afghanistan, it is becoming increasingly apparent that some of them participated in interrogation practices and acts of abusive violence with detainees for which they were not properly trained or psychologically prepared. The mental health impact of deployment during these wars is still being researched, as many veterans are at risk for developing chronic PTSD. At this point, American soldiers and citizens are left with many unresolved questions about the moral calculus of using torture as an interrogation strategy in American military operations. The project raises awareness of these issues and provides a platform for discussion of military interrogation methods and their effects on detainees, soldiers, and society.

2.2   VRE Architecture

The structure of the project consists of nine Virtual Reality Environments (VREs) linked together. The temple panorama, which is the entry point to the project, is positioned in the center. As the audience enters this space, they become acquainted with the four soldier characters who will be the focus of work, through monologues describing their reasons for enlisting in the military. Four open doors allow participants to peak into domestic environments connected to this central panorama: children room, kitchen, living room and the backyard. Participants see the rooms from a first-person perspective. Each connected room contains four interactive objects, the memory triggers, which serve as portals to the linked panorama environments. Users can click the objects using virtual laser pointer to transport themselves into a surreal panorama connected to it, which is intended to represent a subconscious space of interiority, and to provide the audience with a sense of initimate communication with the voices they will hear. The room fades out and participants hear short monologues about the solders' wartime experiences. Once the story is complete, the war panorama

fades out and users are transported back into the room. Once all four objects in that room are explored, the viewer is teleported back to the central temple panorama and the door to that room is closed. Once all four rooms are fully explored, the user is returned to the temple scene, which fades out to red accompanied by a heartbeat sound.

2.3   CAVE2

This project was developed at the Electronic Visualization Laboratory (EVL) at the University of Illinois at Chicago, the birthplace of the CAVE2. The CAVE2 is powered by a 37-node high-performance computer cluster connected to high-speed networks to enable users to better cope with information-intensive tasks. It is approximately 24 feet in diameter and 8 feet tall and consists of 72 near-seamless passive stereo off-axis-optimized 3D LCD panels, a 36-node high-performance computer cluster plus a master node, a 20-speaker surround audio system, a 14-camera optical tracking system, and a 100-Gigabit/second connection to the outside world [1]. The CAVE2 provides participants with the ability to see three-dimensional objects at a resolution matching the human visual acuity, explore the environment, and hear surrounding spatial sounds around them, similar to how they hear in real life.

Fig. 2. CAVE2 input devices; the head and the wand controllers with retro-reflective markers for Vicon infrared optical tracking system.

CAVE2 uses a Vicon infrared optical tracking system to track two objects, the wand and the head tracker (Fig. 2). Each object consists of a unique arrangement of retro-reflective markers. These markers allow the tracking system to determine the position and orientation of the object within CAVE2. The head tracker mounted on the participants' tracking glasses is used to calculate the viewpoint according to the participants' body and head movements in CAVE2 allowing for an immersive virtual reality experience. Objects within the virtual space will be drawn at 1:1 scale and displayed based on the position of the tracked user.

The wand enables hand motion interaction in the immersive VRE, and also has buttons and analog controls for user input. Pointing and pressing a button on the wand can be used for grabbing virtual objects or specifying a direction to navigate toward. CAVE2 uses the Omicron Input Abstraction Library [7] to combine mocap and controller data into a single 'wand' event. The project scene is continuously updated according to the orientation and position of the navigator's head, and the virtual laser pointer is moved in accordance with the participant's actual hand movements. When a participant moves inside the CAVE, rotates his or her head or pushes a button on the wand, the computer system controlling these devices will receive the input signals and provide feedback accordingly to achieve a seamless interaction experience.

2.4   Development Platform

The VRE was developed using the Unity game development platform based on C# scripting language (Unity Technologies Inc., CA), that is typically used by video game developers (Fig. 3). 3D objects, spaces, textures, and materials were developed in Maya (Autodesk Inc., CA), which supports all stages of the 3D modeling, including surface creation and manipulation, texturing, lighting and export to Unity. We used freely available 3D models of the utilitarian objects and interiors from royalty-free websites, in which we modified, triangulated and collaged the geometry, reassigned new textures and materials in order to make the scene more realistic. We imported the objects, environments, textures, and animations from Maya into Unity to design our VRE. Animations and special effects were also incorporated into the scene by using advanced Unity techniques in order to further encourage engagement.

One of the goals of our project was to create a convincing environment with sufficient immersion to focus the participant's sense of presence throughout the narrative. The engaging graphics and diverse special effects were employed to enhance immersion in the environment to facilitate involvement. We used a variety of special effects and advanced Unity features to recreate unique atmospheric and geospatial characteristics in the war landscape such as sandstorms, desert grass, smoke, trees, and fires with accompanying visuals and sounds. We added special effects simulating smoke in the living room fireplace and the connected panorama. The smoke particle is configured to the desired settings and the emitter is positioned at the center of the log stack.
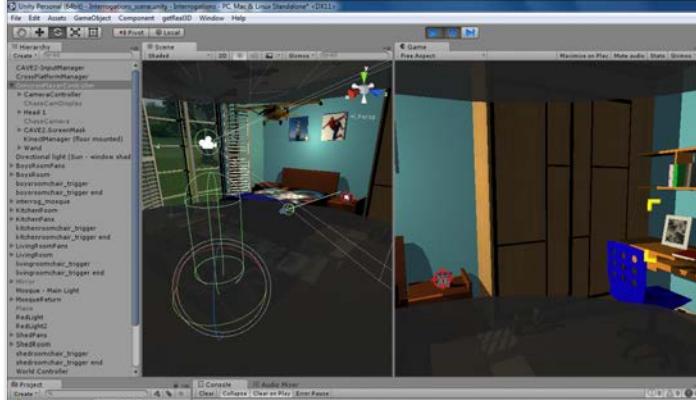
Fig. 3. Development of the VRE in Unity; the CAVE simulator.

The voice recordings performed by professional actors were integrated with interactive media elements and panoramic photographic backgrounds to bring story elements together into an interactive 3D environment. The visual, auditory and narrative elements were brought together using Unity. We used real-time shadows and powerful lighting effects to enhance the illusion.

The getReal3D plugin for Unity developed by Mechdyne Corporation was used to run Unity across the CAVE2 cluster [8]. Scripts from the getReal3D plugin handle the user-centered perspective and synchronize 37 instances of Unity across the cluster creating a seamless 320-degree environment across CAVE2. This includes user inputs, moving objects, anything that uses a random function, and physics collision detection. The getReal3D plugin handles most synchronizations automatically.

User interaction was scripted using the Omicron [7] input abstraction library developed by EVL. Omicron also provides tools to simulate the CAVE2 interaction and display environment for development (Fig. 3). The OmicronManager script handles connection with an Omicron input server, parses events and then broadcast those events to registered Omicron clients (OmicronEventClient.cs). The Omicron Manager also works with the CAVE2Manager to help simplify event handling for head and wand inputs. The CAVE2Manager also provides some basic keyboard emulation of tracking and wand inputs for development systems. Both OmicronManager and CAVE2Manager are packaged into the CAVE2-Manager prefab for easy integration into a CAVE2 Unity project.

3   Interaction

3.1   Interaction inside the CAVE2

The methods of interaction between the user and the CAVE have been studied by different researcher groups [9, 10]. Several studies described evaluation processes and

frameworks to assess the effectiveness of VE interactive technologies [11, 12, 13]. Research has shown that navigation in sparsely populated VREs leaves users disoriented without landmarks [14]. Guidelines of VE design and navigation encourage providing orientation and landmark cues [15, 14]. The superiority of pointing and ray-casting techniques for many interactive tasks has been described in the experimental study of interactive devices used performance times as a dependent variable [12]. It has been shown that simple walking can significantly improve the engagement and immersion in the CAVE-based applications and potentially enhance the sense of presence in VRE [16].

CAVE-based applications typically use direct manipulation and navigation, which are considered the core styles of interaction inside the CAVE [14, 17, 18]. Ben Schneiderman described direct manipulation as an interaction style that allows the user to use the use graphical representations to interact with the operating system [19]. The user can select an object and then an action to be performed on that object. A continuous visual representation of virtual objects and related actions as well as immediate feedback are found to be the main characteristics of direct manipulation [20], which are especially important in the CAVE environment.

In order to navigate through our environment in the CAVE2, a participant needs to point the wand to move toward a specific direction and press a button to activate the transition into one of the soldier's stories. The fly mode has been disabled to ensure close proximity of the user to the interactive objects. We also implemented collision control to prevent navigating beyond the project visuals and getting lost in infinite virtual space.

Fig. 4. Interaction in the CAVE2. The performer walks to one of the invisible interactive colliders holding a chair.

CAVE2 provides a large walking space (20 feet) for interaction in comparison to other CAVEs and VR environments. Our project takes advantage of this larger stage and merges the performance elements with the navigation style of interaction. In our CAVE2 performance the performer had to walk to one of two steel folding chairs situated in the physical environment and turn his head facing the direction of the desired entry in order to enter each room (Fig. 4). The chair was positioned in the first collision area prior to the beginning of the performance. In order to enter the second room, the performer had to physically move the chair from one interactive area to a second interactive area inside the CAVE2. The use of a foldable metal chair was inspired by the narrative describing the memory of one of the solders who participated in the interrogation of the detainee exploiting a metal chair.

Fig. 5. Direct manipulation interaction in the CAVE2. By pushing a button on the Wand, the performer can point the virtual laser pointer and click on the watering can trigger object.

Direct manipulation was used to interact with trigger objects in each room. By pushing a specific button on the wand, the performer could point the virtual laser pointer and click on the trigger objects in the room (Fig. 5). The C# code example of wand pointing and triggering an object on multiple platforms is shown in Fig 6. All input and event processing is done on the master node which then sends the final event trigger across the cluster.

3.2   The computer version

In the standalone computer version of the project, navigation is performed using typical first-person shooter interaction using an Xbox controller, which replaces the wand. Instead of physically navigating through the CAVE2 space, movement is

controlled by the joystick (Fig. 7). Looking around using the second analog stick can be substituted using the Oculus Rift headset. Pointing on the objects and clicking on the triggers using the Xbox controller was very similar to interacting with the objects using Wand in the CAVE2. We added a target image to simplify the selection of objects using a laser pointer, and decided to discard the direction of the navigator's point of view (the angle) as a required parameter to enter the room.

The performance navigation in which a performer could use his physical position and orientation in the virtual environment by walking to interactive zones was impossible to adapt for the flat screen computer version. Consequently, the performance navigation was converted into a first-person interaction in which the participant had to navigate to each interactive entry using the joystick mode on the controller.

```
public class WandPointer : OmicronWandUpdater {
    …
    void Update() {
        // Shoot a ray from the wand
        Ray ray = new Ray(transform.position, transform.TransformDirection(Vector3.forward));
        RaycastHit hit;

        // Get the first collider that was hit by the ray
        wandHit = Physics.Raycast(ray, out hit, 100, wandLayerMask);
        if (wandHit) { // The wand is pointed at a collider
            // Send a message to the hit object telling it that the wand is hovering over it
            hit.collider.gameObject.SendMessage("OnWandOver", SendMessageOptions.DontRequireReceiver);

            // If the laser button has just been pressed, tell the hit object
            if (CAVE2Manager.GetButtonDown(wandID,CAVE2Manager.Button.Button3)) {
                hit.collider.gameObject.SendMessage("OnWandButtonDown", laserButton, SendMessageOptions.DontRequireReceiver);
            }
        }
        …
    }
    …
}

public class TriggerScript : MonoBehaviour {
    …
    void OnWandButtonDown(CAVE2Manager.Button button)
    {
        if( !triggered && panoramaHider.nextPanorama == -1 )
        {
#if USING_GETREAL3D
            // used to sync trigger across cluster
            // #if define used for cross platform purposes (getReal3D is a Windows only library)
            getReal3D.RpcManager.call("TriggerMemoryEvent");
#else
            // used for non-CAVE cases
            TriggerMemoryEvent ();
#endif
        }
    }
    …
}
```

Fig. 6. C# code example of Wand pointing and triggering an object on multiple platforms.

3.3   The web-based version

In the web-based version of the project, the navigation was adapted to be used with a game controller or set of standard navigation keys typically used for the web-based games (A/D – rotate left/right. W – Move forward. S – Move backward.   R – Menu. Mouse– Point on the objects. Click - Trigger objects). The participant had to navigate to each interactive entry point using navigation keys. The user explores the VRE using

the keyboard to move and the mouse for direct manipulation to point and click on objects.

3.3   Mobile version

For the mobile version of the project, we optimized all the media of elements to achieve decent frame-rate performance. 3D spatial sound effects were converted to 2 channel stereo. The textures were optimized from 4K resolution in the CAVE version to 1K resolution. Collision triggers were enlarged to ensure smooth collision detection and faster interaction. The navigation interaction was converted to use touch interface of the iPad. The user has to swipe the environment left and right to turn in the desired direction and move forward or backward by swiping up or down. The user can also rotate the camera by tilting the iPad up or down. Instead of wand or controller buttons, the iPad utilized double tap to click on the objects and single tap to activate laser pointer.

Fig. 7. Interaction with the VR environment running on the personal computer using the Xbox 360 controller. By manipulation a joystick on the controller, the participant can navigate and explore the virtual environment.

4   Discussion

We assessed the responses of participants and navigators to the interaction with VRE during project performances, exhibitions, panels and Q&A sessions following the events and demonstrations of the CAVE2, computer and web-based versions of the project. Participants were asked about any difficulties they experienced while

navigating through the environment and problems encountered during the interaction. Overall, participant's responses were largely positive regarding the environment, as well as the interactions performed. We also received positive feedback from the performers and the audience about accuracy and time required to learn a navigation system controls. The majority of participants described the project as immersive and provocative, and expressed an interest in exploring the project in more depth and on different platforms.

5 Acknowledgments

6 List of Abbreviations

3D	Three-dimensional
API	Application Programming İnterface
CAVE	CAVE Automatic Virtual Environment
GUI	Graphical User Interface
HMD	Head Mounted Display
VR	Virtual Reality
VRE	Virtual Reality Environment

References

1. Febretti A., Nishimoto A., Thigpen T., Talandis J., Long L., Pirtle JD., Peterka T., Verlo A., Brown M., Plepys D., Sandin D., Renambot L., Johnson A., Leigh J., CAVE2: A Hybrid Reality Environment for Immersive Simulation and Information Analysis. Proc. SPIE 8649, The Engineering Reality of Virtual Reality 2013, 864903 (2013)
2. Tsukayama J., By Any Means Necessary: An Interpretive Phenomenological Analysis Study Of Post 9/11 American Abu-sive Violence In Iraq. PhD diss., University of St Andrews, St Andrews, UK (2014)
3. Virtual Art, From Illusion to Immersion, Oliver Grau, Leonardo Book Series, MIT Press (2004)

4. Preddy, S. M. Nance, R. E., Key requirements for CAVE simulations, Proceedings of the 34th Conference on Winter Simulation: Exploring New Frontiers, San Diego, pp. 127–135 (2002)

5. Tsoupikova D., Coover R., Rettberg S., Nishimoto A. "Hearts and Minds: The Interrogations Project," In Proceedings of the IEEE VIS Arts Program (VISAP), Paris, France (2014)

6. Tsoupikova D., Rettberg S., Coover R., Nishimoto A., The battle for hearts and minds: interrogation and torture in the age of war. In Proceeding SIGGRAPH '15 ACM SIGGRAPH 2015 Posters, Article No. 12, ACM New York, NY, USA (2015)

7. Omicron, accessed February 8, 2016, http://github.com/uic-evl/omicron

8. Mechdyne Corporation, getReal3D, accessed February 8, 2016, http://www.mechdyne.com/getreal3d.aspx

9. Sutcliffe, A., Gault, B., Fernando, T., & Tan, K. Investigating interaction in CAVE virtual environments. ACM Transactions on Computer-Human Interaction, 13(2), 235–267 (2006)

10. Muhanna M. Virtual reality and the CAVE: Taxonomy, interaction challenges and research directions, Journal of King Saud University - Computer and Information Sciences, Volume 27, Issue 3, 344–361 (2015)

11. Sutclıffe, A. G. And Kaur, K. D. Evaluating the usability of virtual reality user interfaces. Behav. Info. Techn. 19, 6, 415–426 (2000)

12. Bowman, D. A., Johnson, D. B., And Hodges, L. F. Testbed evaluation of virtual environments interaction techniques. In Proceedings of Virtual Reality Software Technology (VRST) 99. ACM Press, New York, pp. 26–33 (1999)

13. Poupyrev, I. And Ichıkawa, T. Manipulating objects in virtual worlds: Categorization and empirical evaluation of interaction techniques. J. Vis. Lang. Comput. 10, 1, 19–35 (1999)

14. Darken, R. P. And Sıbert, J. L. Wayfinding strategies and behaviors in large virtual worlds. In Human Factors in Computing Systems CHI 96 Conference Proceedings (Vancouver BC), M. Tauber, V. Bellotti, R. Jeffries, J. D. MaCkinlay, and J. Nielsen, Eds. ACM Press, New York, pp.142–149 (1996)

15. Vınson, N. G. Design guidelines for landmarks to support navigation in virtual environments. In Human Factors in Computing Systems CHI 99 Conference Proceedings (Pittsburgh PA) (1999)

16. Usoh M., Arthur K., Whitton MC., Bastos R., Steed A., Slater M., et al. Walking>walking-in-place>flying, in virtual environments. In: Proceedings of the 26th annual conference on computer-graphics and interactive techniques. ACM Press, Addison-WesleyPublishingCo.; pp. 359–64 (1999)

17. Hauber J., Billinghurst M., and Regenbrecht H.. Tangible tele-conferencing. In Proceedings of the 6th Asia-Pacific Conference on Computer-Human Interaction (APCHI 2004), pp. 143–152, 29 June–2 July (2004)

18. Bourdot P. and Touraine D. Polyvalent display framework to control virtual navigations by 6DOF tracking. In Proceedings of the IEEE Virtual Reality 2002 (VR'02), pp. 277–278, 24–28 Mar (2002)

19. Schneiderman B. Direct manipulation: a step beyond programming languages. IEEE Comput., 16 (8) pp. 57–69 (1983)

20. Schneiderman B., Plaisant C., Cohen M., Jacobs S.. Designing the User Interface: Strategies for Effective Human–Computer Interaction, (Fifth ed.)Addison Wesley (2009)