

Implementing Augmented Reality into the Gaming Industry on a Desktop Display Interface

Travis Christian, Sam Levine, Matt Scheinerman

Virtual Reality Applications Center and Human Computer Interaction at Iowa State University

Keywords: Augmented Reality, Gaming, Desktop Display

ABSTRACT:

This project focuses on the use of desktop display interfaces for augmented reality (AR) gaming. With this type of interface, the user is able to look at a screen and see the computer-generated graphics displayed above markers. The desktop display interface alleviates most of the problems associated with head mounted displays (HMDs). The software designed to test the system's usability is a space station construction game created with ARToolkit, OpenSceneGraph, Maya, and 3ds Max. Using an AR-enabled pointing device, the user adds various parts to the station, limited by resources and part requirements.

I. INTRODUCTION

One of the main goals of this project is to determine the effectiveness of augmented reality on a desktop display rather than a HMD for gaming purposes. The desktop interface is a rebuild of an older, less ergonomic model. It is constructed with 80/20 aluminum alloy and a 42" high definition television with a mirror attached to the back (Figure 1). A webcam faces the center of the mirror, where the markers are reflected. The computer is able to identify the markers and display the corresponding AR images.



Figure 1: The desktop display in use.

To manipulate the AR environment, the user is given a wand pointing device with two back-to-back markers, ensuring that the selected image is always displayed on the screen. The user can reach under the TV and see his or her hands displayed on that screen. The parts library and building area are also placed under the TV so that the user can manage the wand through button clicks to grab objects from the part library and move them to the building area.

II. BACKGROUND RESEARCH

A. Usability

One of the main problems with current augmented reality systems is the lack of usability. The old version of the desktop display did not have ample room for the user to manipulate the markers underneath the display screen. If the program is noticeably lagging behind real-time speed, the user has to slow his or her pace to wait for the computer to catch up. Also, low frame rate can frustrate the user and lead to nausea. To create a user-friendly system, observability (perceptual consistency) and honesty (cognitive consistency) are vital [4]. Observability is essentially defined as the need for real-time imaging. For a desktop display in which the user is able to see his or her hands, they must appear to move at the same rate as the user is actually moving them. Honesty is the degree to which the display conforms to the user's idea of what the environment should look like, including level of detail and spatial representation. In addition to the two aspects of consistency, the user needs an interactive task that draws his/her attention [3]. Regardless of the technical usability of the device, the software itself needs to be engaging in order to increase interaction time and satisfaction.

B. AR Interfaces

The variety of interfaces available for augmented reality is a problem itself. The chosen interface for this project was a desktop display rather than the increasingly popular HMD. Table 1 gives a summary of the differences between HMDs and desktop displays. HMDs can be bulky and cause visual perception issues [2]. They can also reduce a user's field of view and make perceptual and motor

manipulations more difficult [12]. Since this application is designed to be played without a time limit, and HMDs cause both fatigue and motion sickness to increase at a quicker rate due to increased inertia, it was apparent that the desktop display would be the best choice for an interface [11]. These displays allow for a longer use time since the user is looking at a monitor without any part of the interface attached to the body [6]. Also, the desktop display allows the user to manipulate more than one marker at a time, unlike a handheld accompaniment to HMDs. The nature of the application calls for the user to move different markers and the desktop display gives an “interface range in which the user can manipulate objects and display the state” which means that on a desktop display, it is easy to incorporate a head’s-up display (HUD) onto the screen to assist the user [9]. The desktop display uses the AR markers to display the images. Figure 2 displays two of the markers both unaugmented and augmented.

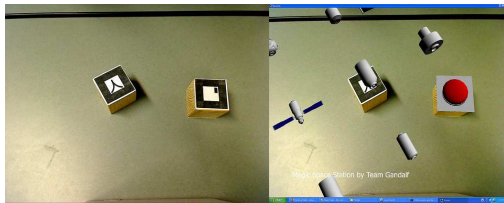


Figure 2: Left: The blocks without augmentation. Right: The blocks with AR enabled, showing cylindrical space station parts affiliated with them.

Table 1.

Differences between HMDs and Desktop Displays.

Device	Head-Mounted Display	Desktop Display
Portability	Sometimes portable	Not portable
Marker Manipulations	Single Marker	Multiple Markers
Health Issues	Increased fatigue and motion sickness	No known issues
Time allowed in use	Limited due to difficulty of perception issues and bulkiness of device	Virtually unlimited
Collaboration	Only by other HMD users	By anyone who can see the screen

C. AR Gaming

The use of augmented reality in the entertainment field is a growing trend. It provides a more interactive style of game-play than the traditional game-play interface by allowing the user to physically control most objects. One of the more well-known examples of an AR game is ARQuake in which the user wears a HMD to play the classic 1996 Quake game in a real environment [10] (Figure 3). This game takes advantage of augmented reality by reducing the graphics needed since some objects are not computer generated [7].



Figure 3: A screenshot of ARQuake’s University level as seen from the HMD. The poles, street, and building are all real whereas the wall and crate are virtual add-ons.

However, ARQuake is only functional in an environment that has been previously mapped. One application in development that uses both a HMD and a desktop display at the time this paper was written is a multiplayer AR racing game [8]. This game also incorporates the direct collaboration aspect of AR. One user “drives the car” while others place objects for the driver to avoid. Figure 4 demonstrates the setup where the “driver” wears a HMD while the other users look at a screen and place components. If the driver were using a desktop display, the game could most likely last much longer [6].

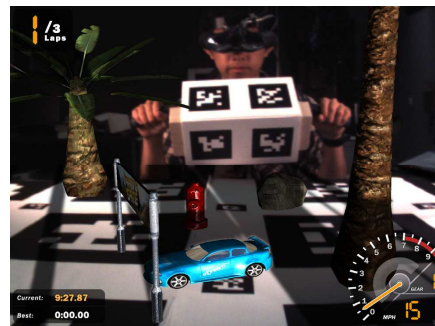


Figure 4: The set up of the AR racing game [8]. The driver is the only one with a HMD while others are viewing the image from a computer screen.

One of the cornerstones of the video gaming industry is the user's visual sense of his or her surrounding environment. As evidenced by Liarokapis, augmented reality provides an increased visual perception of the gaming environment because the user is directly manipulating components rather than using a mouse or keyboard interface for movement [7].

III. GAME MECHANICS AND DESIGN

A. Design Goals

In the preliminary planning stages of the project, four main components were discussed. It was necessary to construct a game that would not have any associated time limit since a main advantage of the desktop display is the long period of time it can be used. Since the webcam resolution inherently limits the quality of the image displayed, it was crucial to draw attention to the AR elements rather than the environment. To keep the user engaged in the application, another necessary aspect was to veer away from a simple construction application and incorporate gaming aspects such as victory conditions when a certain combination of parts are placed in the build area. The final goal was to make the interface usable for a variety of users by making the hardware adjustable. This was accomplished by adding two pivots at the top of the frame to let the TV swing up from vertical to horizontal and any angle in between.

B. Software Overview

The entire game was coded in C++ using OSGART, a combination of OpenSceneGraph and ARToolkit. All of the modeling was drafted in Maya and then converted to 3ds Max. Each set of models correspond to a unique marker. Black and white markers are recognized by ARToolkit, and used as anchors for graphic elements. The markers are attached to a thick paper for durability. These AR markers each have a distinctive set of AR images that appear when the program is in use. The user can move the markers with his or her hands to get various orientations of the AR images. The parts library is made up of a triple marker multimarker. This means that as long as one of the three markers is visible to the camera, the AR images that are tied to the parts library will be displayed. The building area is also tied to a different triple marker multimarker. Figure 5 shows the parts library on its corresponding multimarker. The AR images that the user places in the build area will appear where they are placed relative to the build area marker. The AR wand has two AR markers on opposite sides. This is the tool that the user manipulates to select the various pieces

from the parts library and transfer them to the build area.



Figure 5: The parts library on its multimarker. One of the images is partially covered, but the AR images are still displayed.

The game was developed in two phases: the basic application and the advanced application. The basic application uses the AR wand to copy parts from a part library onto the building area. Each piece snaps to the closest point on an invisible grid in the nearest perpendicular orientation to the wand. In this basic version, the user may place any piece at any position. Figure 6 shows a screen capture of an early version of the basic application in use. After some user tests, it was apparent that more points needed to be added to the build grid and a larger building region was needed. After these changes were made, the user was able to build in 3D space, rather than on one 2D layer. A reset button was also added that clears all objects in the build area. The parts library takes precedence over the build area. When within a specified radius of the library marker the wand interacts with the library, otherwise it interacts with the build area. The advanced application implements a tracker for energy and money, since each part of the station has an associated "cost" with it. Solar panels provide energy, while an automated timer increases resources at a rate dependent on parts present. The required energy and cost numbers are shown when the user selects a piece from the parts library. The total energy and resources available are always displayed in the corner of the screen. Victory is achieved when one of each basic component is placed in the building sphere and the user has a positive amount of energy.

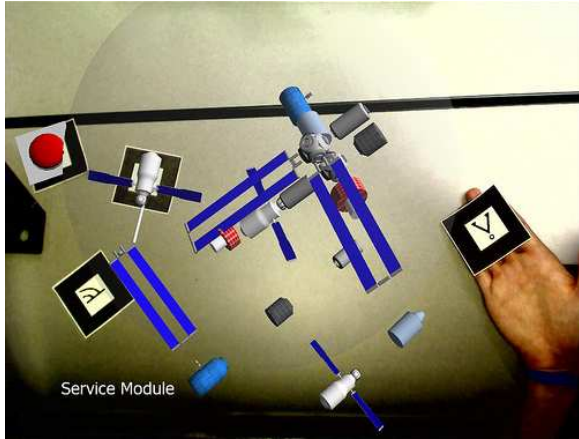


Figure 6: The basic space station application in use.

The more advanced version also incorporates additional parts but restricts the user's ability to place certain components until other prerequisite components have been placed. After another round of usability testing, helpful text displays were added, including a title screen with instructions, a timer, and colored text to serve as a warning that resources and income were getting low or have been depleted (Figure 7). In this version, two paths to victory are possible—a military victory and a commercial victory—since the advanced AR components are geared toward one or the other. Another aspect of the advanced version was the inclusion of certain parts (such as the space hotel) that generate income.

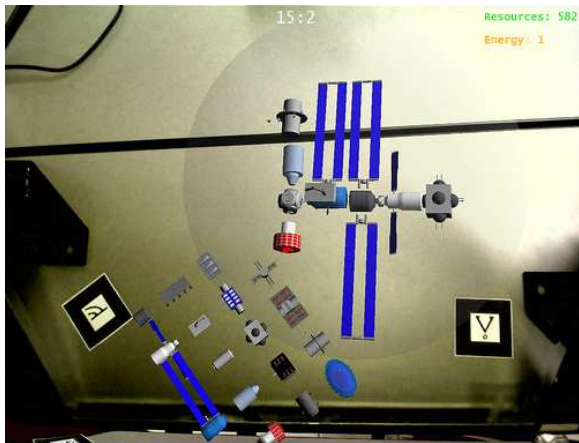


Figure 7: The resource and energy count on the top right of the screen. Notice that the energy counter is in yellow because the player is running low on energy. The timer is in the top middle (reading 15 minutes and 2 seconds).

C. Hardware Overview

A new hardware interface was required after an analysis of the current interface. As previously stated, the old interface was not sufficiently useable. For the new design, it was required that it incorporate

features to make it appealing to a variety of users. Some features that were integrated into the final design were the adjustable pivots, high definition television, and a wall mount system to keep the front of the TV relatively free of building material. Figure 8 shows the SolidWorks concept drawing of the hardware interface. One of the first items to be determined was the type of material to use. After extensive research on different types of metals, the choice came down to either UniStrut or 80/20. It was determined that 80/20 was the only viable option because UniStrut did not have the necessary joint brackets. Also, 80/20 is more appealing to the eye due to its erector set quality, making the fittings nearly seamless in most cases. The black models were chosen instead of the silver because the TV is black, which again gives the interface a more appealing look. The television used is a Samsung 42" LCD HD flatscreen. The flat screen model was chosen so the TV would have a minimum depth, thus allowing for more room between it and the tabletop.



Figure 8: The left, front, and isometric SolidWorks views of the final design. The side bars are clamped to the user's desired height and can be adjusted via the top pivots.

The wall mount serves the purpose of keeping the front of the TV untouched by material. Parallel to the mount is a mirror. The mirror faces the table and reflects the AR markers and the webcam reads them from the mirror. Before feeding the video to the application, the camera reverses the image to negate the effects of the mirror.

There were some minor problems that occurred during construction. The holes on the wall mount were of different sizes, so some of the 80/20 fasteners needed to be screwed into the mount before they could be attached to the 80/20 bar. The webcam must be behind the TV structure and elevated. The old system used a box with the light behind the camera to add direct lighting. In this system, a tripod was used for the webcam and two lights were attached to the sides of the structure.

In addition to the TV interface, a wand was also constructed. This wand is the tool that the user controls to move AR components around the workspace. The wand is a wireless presentation pointer with two AR markers extended from the

front. The user can move the wand around and select pieces with the left click button. The one problem that occurs with the wand is that it becomes somewhat difficult to place the components in the exact position desired. Figure 9 shows the wand with its AR markers attached.

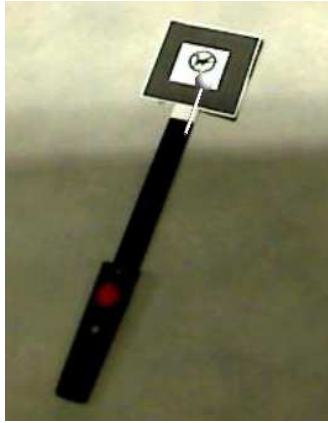


Figure 9: The wand with one side of the AR marker showing. The extension bar gives the user more The other side of the wand contains the other half to the wand multimarker.

V. FUTURE WORK

The manipulation of components was done solely with a wand in this application, but it is possible to use a glove with AR markers, reducing the number of objects the user is forced to use and allowing a more natural interface. It can also extend to include two handed operations if two gloves or other devices were used. Azuma mentions the possibility of including haptics in a glove approach [1]. If the user was able to “feel” components locking into place or force feedback from an attempted invalid action, it would increase the human interaction in the game by adding another sense that he or she experiences during game-play. The incorporation of audio would give the game more of a “video game” feel regardless of the inaccuracy of sound in space.

This particular game can also be improved to incorporate advanced game-play rules. A time limit or a specific required space station model could make the game more challenging. Random events can also be added, thus adding an element of surprise for the user. An example of such a scenario would be if a meteor crashed into the station and destroyed components.

It is also possible to extend augmented reality games into massive multiplayer online games (MMOs) [5]. In this case, the user would be able to build ships to combat “enemy” ships (other players). It may be possible for some users to play on a computer screen while others use a combination of AR and computer systems in order to increase the number of users.

The construction and disaster elements of the game can be applied to simulations. With the incorporation of a physics engine the space program could potentially run tests to determine the potential damage of various incidents (e.g solar flare) on parts of a space station in 3D.

The hardware component can also be modified to fix minor flaws. The mirror is about eight inches away from the back of the TV. If a custom mount was built, the mirror could be closer to the back of the TV, thus reducing the space the user has to reach around.

VI. CONCLUSION

The application demonstrates that AR is a viable step forward in the gaming industry, especially with desktop display interfaces. Current technology is a gateway to explore and expand the capabilities of AR in entertainment. The project accomplished all four of its design goals. With the incorporation of some proposed future work, augmented reality has the potential to revolutionize the gaming industry.

VII. ACKNOWLEDGEMENTS

This summer’s research opportunity was made possible due to the support of the Program for Women in Science and Engineering, the National Science Foundation (IIS-0552522), and Iowa State University’s Human Computer Interaction Graduate Program. The authors would like to thank Dr. Eliot Winer for his supervision on this endeavor and Dr. Stephen Gilbert for his assistance on the research paper. The project would not have been completed without the guidance and support of graduate students Andrew Koehring and Marisol Martinez. Thank you to the Iowa State Engineering Department for the use of its machine shop required for cutting the material before assembly could begin.

VIII. REFERENCES

- [1] Azuma, R. “A Survey of Augmented Reality.” Presence: Teleoperators and Virtual Environments, 6.4 (1997):355-385.
- [2] Bimber, O.; Frohlich, B.; Schmalstieg, D.; Encarnacao, L.M. “The Virtual Showcase.” Computer Graphics and Applications, IEEE, 21.6 (2001):48-55.
- [3] Bogen, M.; Wind, J.; Giuliano, A. “ARiSE-Augmented Reality in School Environments.” Innovative Approaches for Learning and Knowledge Sharing, Proceedings, 4227 (2006):709-714.
- [4] Dubois, E.; Nigay, L.; Troccaz, J. “Consistency in Augmented Reality Systems.” Engineering for Human-Computer Interaction, 2254 (2001):111-122.

- [5] Lang, T.; MacIntyre, B.; Zugaza, I.J., "Massively Multiplayer Online Worlds as a Platform for Augmented Reality Experiences." Virtual Reality Conference, IEEE. (2008):67-70.
- [6] Lee, J.W.; Kim, B.C. "AR Pueblo Board Game." Entertainment Computing-ICEC. 3711 (2005): 523-526.
- [7] Liarakapis, F. "An Exploration From Virtual to Augmented Reality Gaming." Simulation and Gaming. 37.4 (2006): 507-533.
- [8] Ohan, O; Lister, L; White, S; & Feiner, S. "Developing an Augmented Reality Racing Game." Proceedings of the 2nd International Conference on INtelligent TEchnologies for Interactive Entertainment, Games Session. (2008): Article 2.
- [9] Park, Y.; Woo, W. "The ARTable: An AR-Based Tangible User Interface System." Technologies for E-Learning and Digital Entertainment, Proceedings. 3942 (2006):1198-1207.
- [10] Piekarski, W.; Thomas, B. "ARQuake: The Outdoor Augmented Reality Gaming System." Communication of the ACM. 45.1 (2002): 36-38.
- [11] Riva, Giuseppe. "Virtual Reality." Wiley Encyclopedia of Biomedical Engineering. International Ed. 2006.
- [12] Sanderson, P.M.; Watson, M.O.; Russell, W.J.; Jenkins, S.; Liu, D.; Green, N.; Llewelyn, K.; Cole, P.; Shek, V.; Krupenia, S.S. "Advanced by the Distracted Anesthesiologist." Anesthesia and Analgesia. 106.6 (2008): 1787-1797.