Masters Project Proposal: WorldBlox

Student: James Truesdell
Email: jtrue@gatech.edu
Committee Chair: Celia Pearce
Advisors: Jay Bolter, Ali Mazalek, Michael Nitsche
Semester Due: Spring 2010

Table of Contents

Introduction to the Problem

Immersion and intuitive interaction design are something that many computing researchers are looking to today as the next big jump in modern computing. Some of these people concentrate on the potential for immersive games and interactive experiences that would allow the user to interact in a much more literal fashion. Surface computing, in-air motion controls, virtual reality, and augmented reality are four of the major approaches to solving the problem of creating an intuitive and immersive experience. The largest hurdle most of these face, or faced, is the fact that their technology tends to be related to getting the system into the hands of the consumer to begin with. Issues such as cost and technical knowledge often get in the way of any of these approaches being adopted in their purest forms.

Like these researchers and commercial entities, I see great potential in the widespread distribution of such systems. I have taken it upon myself to propose a system in which a level of the immersive interactions, afforded by the traditional systems, would be brought to the consumer in a way that they could make use of via things that many already have in their homes, like webcams and printers. I propose that this can be done using nothing more than a Windows PC that is connected to a webcam and a printer. I intend to create my game system using the freely available Microsoft C# language and XNA Games Library. By providing distribution of the software over the internet and using only standard PC peripherals, I argue that it will be entirely possible to create an immersive tactile interface for the average home.

An example that I intend to keep in mind while designing my system is the AR system designed by Cheok, Edmund, and Eng. [1] When they were designing their system to model curves and other geometry, they shunned the idea of using “expensive tracking systems and sensors” by treating the user’s fingers as “control points” and using a palette metaphor for their menu via the user’s open palm.

Although their system requires a head mounted display, it illustrates that the high-tech approach is not the one and only way. That stepping toward a more low-tech solution could not only make a system cheaper to design and implement but also be cheaper for a potential user to acquire and use. It is in this spirit that I move forward with my project and am determined to create an immersive game system that can be cheaply implemented and procured by others.

Previous Approaches to Immersion and Non-Standard Interactivity

Before moving onto a more in-depth discussion of *how* I will implement this system, it is important to first examine the four approaches to creating an immersive space that were mentioned previously. By doing so it will bring to light the benefits and complexities afforded by the other approaches and hopefully illustrate why the system that I am proposing is a viable solution to the problem of bringing immersive and intuitive computing into the home.

Surface Computing

Today there are a number of commercial products that attempt to mix things up by moving away from the traditional input model for the personal computer and moving toward more intuitive and innovative input models. In the realm of surface computing there have been a number of moves toward creating a commercially viable solution that would be designed for the family room. The camp that has been making the most noise over this input model is Microsoft, via the Microsoft Surface. At current it is only available to corporate partners and the cost is high enough that even if it were available to the consumer it would cost far too much for most people to afford. Additionally, as it is still mostly experimental technology it requires a level of technical expertise that is likely beyond what the average consumer would have.

In-Air Motion Controls

A lower-cost implementation of a non-traditional interface emerged with the release of the Nintendo Wii. The Wii remote, or Wiimote, is the culmination of years of formerly experimental research done at various companies as well as universities, into a small elegant package that works fairly well for a, comparatively, modest price. The problem that dawns upon the user when they play a Wii game is that more often than not there is not a 1:1 relationship with what is happening on screen and what they are doing with their hands. That is to say, it is a moderately frequent occurrence in Wii games, especially when they are ports from other consoles, that the swinging of the remote, rather than have any particularly meaningful relationship with the game world, is more or less the same as pressing the 'A' button. It is not that this is a bad thing necessarily but it feels like Nintendo fell somewhat short of their goal. Although, what they did do right was they brought their console to a whole new category of consumer and as a result went from being non-competitive to being the market leader in their industry.

VR Systems

At SIGGRAPH 01 a panel presented an extensive analysis on large-scale VR installations such as CAVEs, Domes, etc. These systems are generally found in places like museums and art galleries and they "deliver unique artistic, entertainment, and educational experiences" to their users. [2] These systems are very immersive experiences for their users but do have a number of drawbacks, the most obvious being cost and accessibility. These systems have to be literally built into and for the spaces in which they are going to exist and that takes a considerable amount of money. Cost alone makes such systems basically out of the reach of the normal consumer. According to Josephine Anstey another issue with large-scale VR installations is that due to the cost and being put in a public location many people tend to use them at the same time. The issue here is that in order "for the experience to have the most impact, the user must feel comfortable and confident enough to engage with the piece physically and emotionally" and for this to happen it "often means being alone with the piece." [3] This leads to a

situation where we have a system that works quite well but rarely operates under what we might consider the optimal conditions for user immersion.

AR Systems

Augmented reality is similar to virtual reality but rather than occlude the user's vision and present them with an entirely virtual world, it allows the player to see the real world but superimposes the virtual upon it. Examples of its use range in variety significantly and two such projects are *Four Angry Men* and an AR version of the classic game GO. In *Four Angry Men* [10] the player wears a head mounted display and sits in one of four chairs at a table. Upon sitting down the player sees three virtual men sitting in the other chairs and can listen to them deliberate on the guilt of a person on trial. Depending on where the player sits, they literally become that man and the dialogue of the other characters change to reflect the stereotypes and biases of the player's current character. The problem with games like this is that there tends to be a floatiness to the virtual elements that makes them appear not quite in the environment. This can be rather jarring and break the immersion of the simulation somewhat. In AR GO the first player manipulates real go pieces on a virtual GO board, which is displayed via a head mounted display, and the second player has a traditional PC interface. The second players' pieces are placed in the correct place on the first players' virtual board and first players' real pieces are placed on the second players' PC based board properly via a camera interpreting the positions of the first players' real pieces. [5] The issue here is that we have what is designed as and played as a two player game, however, removing the players physically from one another might in some way impact the richness of the experience of playing a game with someone. Additionally, much like VR, AR often requires a wearable computer with a head mounted display as well as a room that is set up with markers, usually fiduciary markers, which the AR system can interpret in order to display the desired world to the user. This equipment requires a level of technical knowledge as well as money that the average consumer would not likely have.

Conceptual Design

Proposal

The project I am proposing is to create a pc-based gameworld that is controlled by the player(s) via simple manipulation of fiducially tagged objects/blocks/pieces of paper under the eye of a webcam. I wish to create, perhaps literally, a sandbox environment on the player's desk/floor/coffee table within which they can exploit the affordances that the simple objects allow. This allows me, or other developers, to create games *for* the device rather than just create any one game. The project can be thought of as a sort of experimental pseudo-console that is operated primarily via the manipulation of the objects. However, since the software lives on a computer, the keyboard/mouse and gamepad interfaces could also be easily supported for other types of games.

As mentioned before, a constraint that I am striving to work within is to use components that nearly everyone should have so that the game could be easily distributed over the internet and cause as few headaches as possible. My system will be designed to address and minimize the issues of cost, technical complexity, and feasibility of installation that plague the established methods. If a comparison were to be drawn to any of the previously mentioned artifacts it would seem that AR GO might be the most natural comparison. My system will alter the virtual on-screen game space via the manipulation of the tangible blocks. This is not dissimilar to how AR GO alters the second player's on-screen game world via the manipulations of the first player's physical pieces. An advantage, however, is that my system will not be operated over a long range network like AR GO was and so will keep both players in the same place both physically and virtually interacting with one another.

Run-through

The following is a potential run-through of a one-player game. The player sits down at their desk with their laptop and sets up their webcam which is viewing the surface at a perpendicular angle. They load up the game they wish to play, in this case it is a puzzle game, and after calibrating itself with the camera the game begins. The player moves about the game world by sliding a player piece around under the webcam. This directly manipulates the player character on the screen and moves them in whatever direction it was moved. After moving about the game world the player sees that they will need to add a span, a tower, and a ladder to the environment to solve the puzzle. They grab the block that has a tower marker on it and slide it into place, then the ladder, and then the span. There doesn't have to be a necessary order to this (but there could be) this is just how they chose to add them. The player then moves their manipulation block over the other blocks and to the goal. They win the game.

A second game uses the system in a reverse way. The players print off a map, in this case one of the states of the United States, and place it under the eye of the camera. On the screen the system asks the first player to put a marker on Tennessee. The player puts a marker on Tennessee and the system responds that they got the right answer. The second player is then asked to put a marker on the sunshine state and puts a marker with their color on Florida. The game continues as such until an endpoint is reached and the players' scores are compared. This game illustrates the fact that the system is not limited to just having one player at a time. Additionally, the two player games that are created for the system could be turn-based or real-time as well as co-operative and counter-operative. I feel this is important because it can bring two people together in a way that a networked game or a multi-player video game cannot. Additionally, there is no reason the game needs to be limited to just two players. It is reasonable that games could be made to allow as many people to play as can fit around the playing area; it's just a matter of creating games for such scenarios.

Required Resources

The materials that I will need to generate for the project will be software based and hardware based. On the hardware side I will need a webcam to attach to my/any computer, I will need to create the fiducial markers necessary to do the tracking, and will need to create/procure the necessary blocks/shapes to attach them to. On the software end I will need to create a program that is able to interpret the XY coordinates of the markers and another side of the software that takes this information and inserts it into the game world that the player operates within. Other questions such as should the player control the PC with a gamepad, keyboard and mouse, or with a special super-fiducial and whether or not a printable map of some sort to drop the blocks on is desirable/necessary will need to be answered at a later date via iterative playtesting.

There are three aspects of software that need to be created in order for the project to be a success. First, the actual software for capturing the camera data needs to be created or appropriated from elsewhere. As I have experience using the TUIO system from a prior project in the Synlab at Georgia Tech where we worked on a surface computer, [5] it has been discussed that trying to reverse engineer or repurpose the TUIO software so that it will work from above instead of from underneath the surface of a table would be the best way to go. This is also a desirable course of action because the TUIO system has already been written to work with C# and the XNA Game Library. If this does not work I will need to find some sort of other fiducial interpreter because it is unrealistic for me to develop such a system from scratch. After this it will be important to create the general code that will allow the data taken in by the camera system to be interpreted in a way that the games in the *WorldBlox* system will understand. Finally, at least one game needs to be created to demonstrate the system for the end of spring defense week and demo day. What will be better is if I can create two original games to help illustrate the breadth of games that the system can cover. These games will be made, as said before,

using C# and the XNA Game Library as I am familiar with them and the TUIO system already can interact with it.

My Inspirations for a Low-Tech Solution

Blocks

My initial inspiration for the project came out of a desire to recreate the simple pleasure and openness of a set of children's building blocks in a computational space. Conceptually I was thinking of building blocks as a sort of level editor but one that children use to create imaginary worlds. This was to be an extension of this experience except it would be a software level editor controlled in a more simple way than the traditional FPS level editor. After a few meetings and ruminations, however, I stopped thinking of it as a simple building block set and began considering the possibilities for other sorts of games that could work in such an environment. The benefit of this approach is that I can still create a simple building block style game but can use this as a simple test bed application and make more complex software based on my findings.

Zune Buggies

One of the earliest inspirations for the project was one of Andy Wilson's experimental games *Zune Buggies* [6]. For this he set up an overhead camera pointing downward at a table's surface. On a separate computer screen a 3D orthogonal view of the table's surface was displayed. Then he would place various folded pieces of paper and objects under the camera's eye and upon doing so the camera would interpret the data and add the form as a bump in the table's surface that was the same color as the object. Using an Xbox360 controller he was able to drive a small virtual dune buggy over the hills formed by the objects on the computer screen. Additionally a person could place their arm or anything else under the camera's eye and have a similar result. I found this virtual world building via real objects

to be enthralling and see a great deal of potential in such a system. I have differentiated my system from this approach by the use of the fiducially marked blocks. Rather than create a semi-literal representation of the surface of the desktop in the game world, I want the user to be able to drop in a block that represents a bell tower in one game and car in the next. I think this creates a much richer simulation environment for the user that engenders an element of surprise to every game. More importantly, however, the people creating games for the system have a set of standard inputs that they can plan on when creating their games. This allows them to create a game experience that doesn't have to plan for as many bizarre user interactions and also allows them to create vast and rich worlds as well as the simple representation of the desk. That is to say, in *WorldBlox* there is a much greater level of abstraction compared to *Zune Buggies*.

CYPHER

CYPHER: Cyber Photographer in Wonder Space is a project that was submitted to SIGGRAPH 00. *CYPHER* generates a fully three-dimensional world based on the locations and configurations of "crystal blocks" which represent things like buildings, trees, etc. that are placed onto an "exclusive table" which represents the space of the world. [7] What kind of tree, for example, that gets placed in by the system depends on what mode the system is running as. There are four themes that a user can create with: Edo era Japan, rabbit world, cosmic world, and a haunted world. After the user configures the layout of the world to their liking they then place a "human figure block" which represents where the picture that they took of themselves is going to be placed. The system then takes into account the position of all the blocks and the user and compares it to "a database of about 200 masterpieces" and generates the composition from this. This is actually quite similar to the idea behind the *WorldBlox* system even going as far as to use "the same building blocks that children play with," one of my earliest inspirations. It is quite different, however, because this is not a game in the traditional sense. Indeed, the *WorldBlox* system could include a camera mode like this but at this time it is not planned. A more important

distinction is the hardware used to achieve the *CYPHER* system. It is a tabletop interface that makes use of ultrasonic oscillators and sensors to get positional data for the blocks, something most people probably don't have in their homes.

Mightier

Finally, the game *Mightier* by Lucas Pope and Keiko Ishizaka is a game that formed a good deal of how my system interacts with the real world. [8] In the game the user gets to the beginning of their level, which is a platforming style puzzle, and is asked to print off a sheet that represents the layout of the level from above. Then, the player takes a marker or pen and draws lines on the print off map and when done places it under a webcam. The game then takes a snapshot of the sheet and interprets the lines that the player drew. After this a laser fires down from the sky and creates the platforms that the player drew on the page and then the player is given control of their character and can jump on the platforms and complete the puzzle. It is because of this game that I decided a good method for determining what game *WorldBlox* should load is to have the user print off their own maps and fiducial markers. The fiducials could be universal across the games but they could print off different page-sized map fiducials that the system takes a quick snapshot of before the game starts. Based on which fiducial it finds it would load that game. Additionally, it would be possible to have multiple sheets for a single game and lay the pages next to each other in any order the player chooses so they could create unique game worlds for each play through in a method similar to the board game classic *The Settlers of Catan* [9]. Where *Mightier* and *WorldBlox* differ is in the sorts of interactions they afford. Both allow the user to change the elements of the world around them but in *Mightier* they are given only one option on how to change the world, raising platforms, but they can do this as much as they like within the space. The *WorldBlox* player, on the other hand, has more options on how to affect the world but can only create the world using the pieces they are given. It is not dissimilar to comparing creating a sand castle

in a sandbox full of sand and a block fort using a toy chest full of blocks. Both allow for creative construction but in different ways.

Evaluation

Throughout the development timeline of the project a good amount of playtesting and evaluation will be needed. In the initial phases of the design I intend to make use of paper prototyping and roommates, friends, other DM students, and whoever else will play it to determine what does and does not work with the interaction and other elements of the system. In the initial phases of the programming I intend to develop small prototypes with very minimal art and test these with users. For example, it is important to determine what the best course of action for manipulating the player character and camera is at an early stage. Questions to be asked are: should the player have direct control of the player via a fiducial block or should it be via a controller like in *Zune Buggy*? If they are controlling the player via a block how then do they manipulate the camera? If they are controlling the player via a controller how do we account for the disjoint between manipulating the blocks and moving around with the controller? Each question brings more questions and all of them need to be answered, hopefully prototyping and user testing will do that.

Deliverables

In the spring semester I would like to have at least one, but preferably two or three small-scale sample games that illustrate the various interactions afforded by the system as well as the breadth of game types that the system is capable of supporting. The interface itself will need to be completed. The interface includes: a camera with software that supports the interpretation of fiducial markers written, all the printable documents necessary of map, character, and world interpretation to be completed and

a website set up where the system and games can be accessed. Furthermore, the written project documentation will need to be completed and ready to be turned in. Also a secondary version of this document needs to be written that is appropriate for submission to a number of conferences, SIGGRAPH etc. Finally, a demo video should be created for easy demonstrations of the system in the future without the need of carrying around the software and hardware everywhere it needs to be. If there is further interest after the end of spring semester I could continue to work on it and release a fuller version of the system or the code could be GPLed and left for others to develop further.

Timeline

September 30th

- Proposal in

October 15th

- Project request due

Mid Oct-Beginning of Nov

- Playtesting

Nov-Dec

-Prototyping
-Initial camera coding
-Game design, iterative design

Jan-Feb

-Finish camera code
-Initial game code
-Initial project design document writing

Feb-March

-Finish draft for project design document
-Playtesting
-Continue game code work

March-April

-Finish project
-Finish paper revisions

April-May

-Make any other necessary revisions
-Demo day

References

[1] Cheok, A.D., Neo Weng Chuen, E., Ang Wee Eng. “Inexpensive Non-Sensor Based Augmented Reality Modeling of Curves and Surfaces in Physical Space”. Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. 30 Sept.-1 Oct. 2002 Page(s):273 – 274.

[2] Anstey, Josephine et al. “The CAVE and Beyond: VR in Museums and Galleries”. SIGGRAPH 2001. ACM SIGGRAPH. New York, 2001. Page visited 10/01/09. <http://www2.ccr.buffalo.edu/anstey/RESUME/pres/sig01/>

[3] Anstey, Josephine et al. “The CAVE and Beyond: VR in Museums and Galleries”. SIGGRAPH 2001. ACM SIGGRAPH. New York, 2001. Page visited 10/01/09. < http://www2.ccr.buffalo.edu/anstey/RESUME/pres/sig01/anstey/index.html>

[4] Hua, Hong. “A Head-Mounted Projective Display and its Applications in Interactive Augmented Environments”. SIGGRAPH 2001. ACM SIGGRAPH. New York, 2001.

[5] Mazalek, Ali. "Projects: gamewell". TSRB: Synlab. Page visited 10/1/09 <http://synlab.gatech.edu/project.php?id=57>.

[6] Posted by: snackfight. Video of: Andy Wilson. TechFest. 2007. Youtube. Page visited 10/1/09 <http://www.youtube.com/watch?v=xujhFInvyxo>.

[7] Nakai, Takahiro et al. “CYPHER: Cyber Photographer in Wonder Space”. SIGGRAPH 2000. ACM SIGGRAPH. New York, 2000. Page visited 10/01/09. < http://www.cosignconference.org/downloads/papers/nakai_et_al_cosign_2001.pdf>

[8] Pope, Lucas and Ishizaka, Keiko. “Mightier”. Ratloop. 2009. Page visited 10/1/09 <http://www.ratloop.com/?games/mightier>

[9] "The Settlers of Catan". *Wikipedia*. 2009. Wikimedia Foundation, Inc.. Page visited: 10/1/09 <http://en.wikipedia.org/wiki/Settlers_of_catan>.

[10] Dow, Steven et al. “Four Angry Men”. *Augmented Environments Laboratory*. Georgia Tech. Page Visited: 10/1/09 < http://www.cc.gatech.edu/gvu/ael/projects/4am.html >