A Low-Cost Infrastructure for Tabletop Games

Christopher Wolfe, J. David Smith and T.C. Nicholas Graham
School of Computing
Queen's University
Kingston, Ontario, Canada K7L 3N6
{graham,smith,wolfe}@cs.queensu.ca

ABSTRACT
Tabletop games provide an intimate gaming experience where groups of friends can interact in a shared space using shared physical props. Digital tabletop games show great promise in bringing this experience to video game players. However the cost of developing tabletop games is high due to the need for expensive hardware and complex software. In this paper, we introduce *EquisFTIR*, a low-cost hardware and software infrastructure for digital tabletop gaming. We illustrate the infrastructure through *Asterocks*, a novel tabletop game.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Input devices and strategies*

General Terms
Human Factors

Keywords
Computer games, tabletop games, FTIR, frustrated total internal reflection

Figure 1: Queen Nefertari playing Senet

1. INTRODUCTION
Electronic tabletop surfaces (or just "tabletops") have started to move from the research lab into commercial availability. Tabletops allow small groups of people to cooperatively interact with digital media within the familiar setting of grouping around a table.

Tabletops have enormous promise for supporting group gameplay. The lasting popularity of games such as chess and poker, or of newer games such as Carcasonne and the Settlers of Catan, shows that people enjoy the experience of gaming with friends while seated at a table. Tables provide an intimate setting, allowing people to interact simultaneously with shared game boards and pieces that everyone can see and touch. Indeed, early research has shown tabletops to be a promising medium for innovative digital games [10, 25, 11].

A significant barrier to exploring tabletop gameplay is the cost of the necessary hardware and software. Commercial products such as Microsoft's Surface or Mitsubishi Electronics' DiamondTouch cost tens of thousands of dollars, well beyond the reach of consumers, and overly expensive for many research labs. This expense makes it hard for the research community to explore design issues for tabletop games, and makes it difficult for game development studios to make a business case for entering the tabletop gaming market.

To address this problem, we have developed *EquisFTIR*, a low-cost infrastructure for tabletop games. *EquisFTIR* consists of a tabletop that can be built for about $2,000, and a free software library allowing the development of tabletop games using Microsoft's XNA Studio. To illustrate our approach, we present *Asterocks*, a game we have built with this infrastructure.

The paper is organized as follows. We first discuss the state of the art in tabletop gaming, motivating our interest in digital tabletop games. We then describe existing infrastructures for tabletop gaming. Following this, we introduce

Figure 2: "Cocktail Lounge" version of Ms. Pac-man arcade game [24]

the hardware and software components of the *EquisFTIR* infrastructure, and conclude with a presentation of our *Asterocks* game.

2. TABLETOP GAMING

Tabletop games have existed for at least 5,000 years [18]; for example, figure 1 shows Senet, a game played in ancient Egypt. Tabletop board games remain enormously popular, with US sales exceeding $800 million in 2006 [19]. Much of the enjoyment of games played on tabletops is due to the intimate social interaction they support, where players can see what others are doing and tangibly interact with the shared gaming surface.

Digital tabletop games made their first appearance in the 1980s. "Cocktail style" arcade games flipped traditional arcade units into a tabletop form, allowing seated play (figure 2 [24]). Players interacted with the games indirectly, via the same joystick and button controls as used in standard arcade game presentation. Competing players sat on opposite sides of the table and the orientation of the display was flipped back and forth to facilitate turn-taking.

More recently, augmented reality tabletop games have been used to combine physical tables and game pieces with virtual game play. For example, *STARS* allows a combination of physical pieces and a virtual game board projected on a physical table [8]. Similarly, *Tankwar* demonstrates how traditional war games can be played with a virtual board on a physical table [17]. Augmented reality tabletop games provide much of the the pleasurable aspects of traditional tabletop board games, allowing players to directly interact with the board and its pieces while in close proximity to their fellow players. They require, however, complex hardware that may in extreme cases hinder this interaction; for example, Tankwar requires a virtual reality helmet, and suffers from the technical problems of real time head tracking.

The recent development of electronic tabletop surfaces has opened new possibilities for digital tabletop gaming. The game board is displayed on the tabletop surface; game pieces can be virtual, or (e.g., using the Microsoft Surface's object tracking facility) physical. Players can interact with the game board by touching, dragging and gesturing. Since electronic tabletops have a similar shape to real tables, they provide a familar and natural form of interaction.

While this technology is still young, a few electronic tabletop games have been created that show its promise. Tse has adapted Warcraft 3 and The Sims to run on a Diamond Touch electronic tabletop [25]. Two players can cooperate to play the game, in which gestures and voice commands replace the traditional mouse and keyboard interface. Mattar's *PinguTouch* provides a wealth of practical experience about how to design tabletop games [10]: PinguTouch's table is octagonal in shape, allowing easy access for all players. The sizes of objects appearing on the display must be large enough to allow easy selection and manipulation with fingers, which are considerably less precise than a mouse. *TViews* is an example of a role playing game adapting the tabletop gameplay of Dungeons and Dragons [11]. A study of TViews players showed that they found the tabletop style enhanced teamwork and group interaction.

3. EXISTING INFRASTRUCTURES

Developers of tabletop games require a tabletop device as well as software libraries to control it. Numerous competing technologies are emerging that can be used to construct interactive tabletop surfaces. Today, state-of-the-art tabletop systems include a computer display and a sensor used to detect touches and movement of physical objects across the table surface. Displays are commonly provided through front or rear projection, or by using an LCD screen as a table surface.

A simple approach for tabletop touch sensitivity is to deploy an array of discrete sensors. These sensors can operate entirely independently [23], through a connected set of independent active elements [7, 28], or through a matrix of purely passive sensors [20, 16]. However these approaches are complex to construct and often suffer from poor resolution.

DiamondTouch is a unique system that uses electrical current to detect touch [1]. The table that has an built-in array of antennas. When a user touches the surface, she completes a circuit between an antenna in the table and a receiver contained in the chair. While this technique is robust and easy to deploy, the cost of the table is prohibitive for many commercial applications.

Microsoft Surface [13] is an emerging system that uses computer vision for touch sensitivity. An infrared (IR) lamp is placed under the table surface and illuminates objects as they approach the table surface. An array of IR cameras is placed under the table near the lamp that detect and identify objects as they contact the surface. Touch is determined through pixel intensities in the camera image. This technique is sufficiently high resolution to enable objects to be identified on the surface through spatially marked tags. However, similar to the DiamondTouch, current versions of Surface are expensive (greater than $10,000) and its availability is limited to strategic commercial partners.

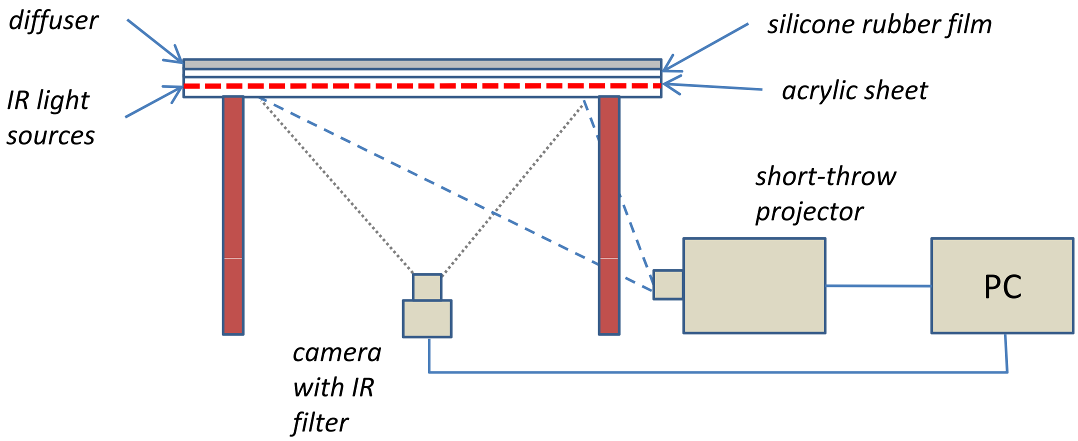

Figure 3: Components of the *EquisFTIR* tabletop

Other vision-based systems provide lower cost solutions to multi-touch sensitivity. These systems either approximate the 3D position of the user's hand through pixel intensity [21], stereoscopy [9, 29], or through markers attached to a deformable material [6, 22, 26, 27]. Using a deformable material has the added advantage of providing passive haptic feedback and adds an element of depth to the interaction surface. Additionally, these surfaces typically report pressure as a vector, meaning touch pressure can be interpreted in directions not necessarily perpendicular to the interaction surface. However the markers on the deformable material must be opaque, meaning the system must be top-projected.

Han proposed a low-cost, simple FTIR-based sensor [3]. The system introduces infrared (IR) light into a medium (typically acrylic) with an index of refraction significantly different than the air around it. When the user touches the acrylic, the area of the touch appears as an IR glowing spot on the reverse side of the acrylic.

Developers require a software library to interact with tabletop surfaces. The Microsoft Surface SDK is currently the most robust multi-touch library available [13]. However, the SDK is tied to the Surface platform and cannot be used with other hardware systems.

The Surface SDK is presented as two libraries. The "Core" library presents low-level coordinate data similar to open-source libraries such as TouchLib [2]. The library is optimized to work with Microsoft XNA [14], a game development platform targeted for Microsoft Windows and XBox 360 platforms. In each frame of the game loop, a "ContactTarget" object is polled for the current state of the interactive surface. Data is reported as a collection of "Contact" objects, consisting of finger touches, tagged objects, and unrecognized blobs.

The second library is a set of touch-enabled widgets implemented with the "Core" that integrate with the Windows Presentation Foundation (WPF) [15], an emerging Microsoft UI development toolkit. These widgets include buttons, lists, and menus, as well as a "ScatterView" container, which enhances its elements with physical properties suitable for multi-touch. For example, a collection of "Image" objects contained in a ScatterView would behave like a stack of photos tossed onto a physical tabletop with gesture support for scaling, flicking, dragging, and rotating. Through integration with WPF, user experiences can be developed using a WYSIWYG interface in Microsoft Expression Blend [12].

4. LOW COST TABLETOP

While a number of tabletop devices are now commercially available, it is clear that their cost is prohibitive for experimentation in research labs or for eventual deployment in consumers' houses. In contrast, our *EquisFTIR* surface provides multi-touch, pressure-sensitive tabletop interaction, and, depending on the components used, can be built for a few hundred to a few thousand dollars. This table follows from Han's basic design [3], but has been refined for performance and cost. Figure 3 shows the components from which the table is built, and figure 4 shows the table itself.

The table is based around a custom-built cast iron frame. This provides a stable and rigid platform, allowing small groups of people to interact with the table without fear of it moving. In earlier iterations, we have used a simple $99 Ikea table, providing improved portability and lower price at the cost of stability. The rigid cast iron table requires minimal cross-bracing, making it easy to mount equipment underneath.

The tabletop surface consists of three layers.

- A 1/4" acrylic sheet forms the main layer. We have found a 4'× 3' square to provide a good balance between size and stability. At larger dimensions, the acrylic may sag, leading to poor performance. Beside the acrylic surface, a set of 13.5v infrared light emitting diodes (LEDs) are mounted. We have placed 60 LEDs along two of the four sides of the table, powered by a 12 volt transformer. The LEDs are soldered together into 12 parallel circuits with enough resistence to operate at approximately 75% power. The reduced power setting is used to prolong the life of the LEDs.
- A thin film of water-clear silicone rubber (purchased

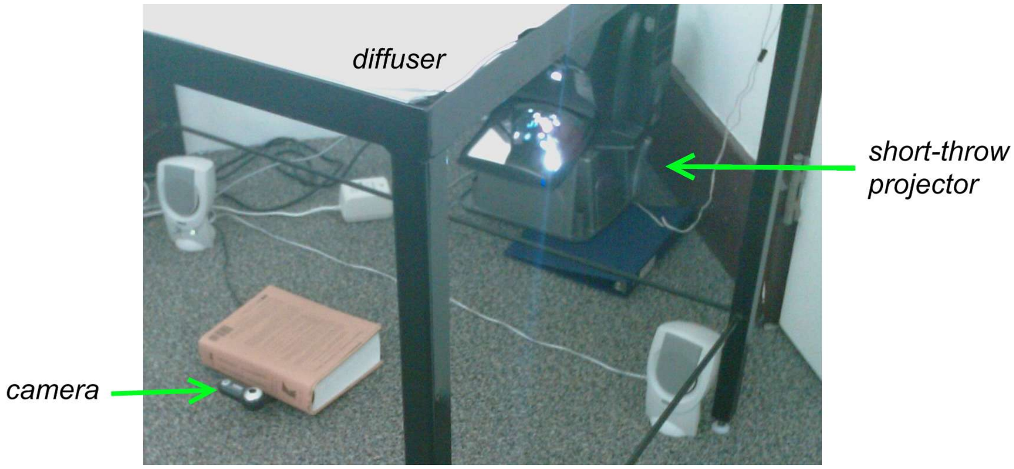

Figure 4: Photograph of FTIR table

from a hardware store) is painted on top of the acrylic. It is important that this film be uniformly applied without bubbles. When the table is in use, depression of the silicone rubber causes frustration of the infrared light flooding the acrylic surface. If the silicone rubber layer is thicker, it is easier to detect pressure; however, after it is depressed, the surface will be slower to return to flat state, leading to a latency between releasing a touch and having that action communicated to the application. We used a 1/8" thickness to provide a balance between sensitivity and latency. To acquire this thickness, we used a thinning agent in the silicone rubber when we applied it to the surface.

- A diffuser layer is then used to provide an opaque surface against which the projector can display, and to protect the silicone rubber. We have found large sheets of newsprint to work effectively.

A Logitech QuickCam Fusion camera with an infrared filter is mounted below the table. This provides input in the form of a 1.3 MB greyscale bitmap, showing the presence of IR light on the tabletop surface. White areas on the bitmap show high IR intensity (e.g., as produced by someone touching the tabletop surface.)

Finally, an NEC short throw data projector (at resolution 1024x768 pixels) is used to project images onto the tabletop surface. This projector is convenient because from a flat position next to the table, it is capable of projecting onto the surface. Less expensive projectors can also be used, but must be positioned at greater distance, and require a mirror under the table to reflect the projected image.

The total cost of the components required to build the *EquisFTIR* table is approximately $2,000, far less than commercial tables. If cheaper components are used, the cost can be reduced to approximately $1,000. These costs make tabletops accessible to research labs, and, pending commoditization, are within plausible price range for home use.

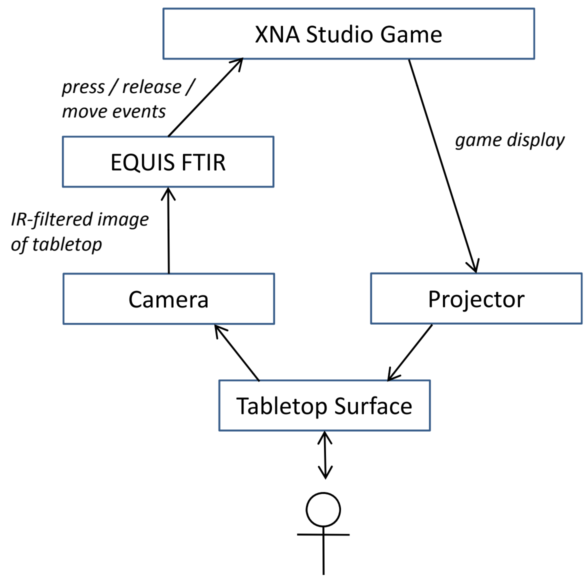

Figure 5: *EquisFTIR* software stack

5. SOFTWARE INFRASTRUCTURE

One of the challenges of creating tabletop games is the difficulty of processing input. As we saw in the previous section, input from the table is a sequence of greyscale bitmaps showing the presence of infrared light on the tabletop surface. This low-level input must be interpreted to determine game actions such as selecting or dragging game pieces, or performing zooming or panning gestures. The raw luminosity input is noisy, making it hard to interpret accurately, and the demands of processing the bitmaps can lead to noticeable input latency.

The *EquisFTIR* software library provides accurate and high-performance processing of FTIR tabletop inputs. The library has a simple application programming interface; it is open-source and freely available; and the library's design is optimized to work with the Microsoft XNA game development environment [14].

5.1 Other Software Libraries

EquisFTIR is an example of an emerging class of software libraries supporting FTIR tabletop surfaces. Other mature examples include *reacTIVision* [4] and *TouchLib* [2].

Both libraries provide touch information via TUIO, an event-based protocol that report when fingers press, release or drag on the table surface [5]. Events are keyed with the "session id" of the object causing the event (allowing, e.g., press and release events to be matched), the event's position on the table, and the mass of the event; this last parameter specifies the size of the light pool on the image, giving an indication of finger pressure. The primary advantage of TUIO is that it allows the game client to be located on a different computer from the image processing server. This is important for processor-intensive image processing libraries.

In addition to touch events, reacTIVision supports object tracking. Tagged objects can be identified in the input image, allowing their position and orientation to be reported.

In FTIR tables, particularly low-cost tables, noise in the input image is a problem. Both libraries address this by capturing a "background image" before the table is used. The background image is subtracted from the images provided by the camera to reduce the risk of spurious identification of events. In our experience, background subtraction is insufficient for low-cost tabletops. There are many changing sources of infrared light, for example due to the projector, background light in the room, or shadows as users move their hands and arms over the table. A single background image does not account for all of these possibly changing sources.

5.2 The EquisFTIR Library

Although its feature set is smaller, *EquisFTIR* provides several advances over reacTIVision and TouchLib. *EquisFTIR* requires low processing power; it can run on the same computer as the game, reducing infrastructure and configuration costs. *EquisFTIR*'s algorithm is based on *frame subtraction* rather than background subtraction, providing better handling of the noisy environments typical of low-cost tabletops. And finally, *EquisFTIR*'s architecture supports easy integration with XNA Studio applications.

Figure 5 shows how *EquisFTIR* interacts with the tabletop surface and a game application. Grayscale bitmaps representing IR intensity are retrieved from from the camera. The library tracks intensity changes from frame to frame, and uses them to identify the movement of "blobs" around the table. Blobs are simply pools of light, generated by users adding, removing or dragging objects around the table. These "objects" are typically fingers, but we have successfully captured the use of other objects such as cookie cutters or even a paintbrush on the tabletop.

EquisFTIR compares adjacent frames looking for changes in brightness. Changes that are above a threshold value are interpreted as user inputs (press/release/drag). This approach deals well with noisy input, as areas that are statically bright due to other IR light sources will not generate inputs. Frame subtraction performs better than background subtraction as it is robust to changes in background noise.

EquisFTIR has very modest hardware requirements. On 1.6 GHz dual core laptop, with the QuickCam camera operating at maximum data rates (30 Hz for an image at 640 $\times$ 480 pixels), the library consumes approximately 15% of available CPU. This makes it practical to run all but the most demanding of games on the same computer as the image image processing library.

The libary generates three types of event: the user has pressed an object onto the table; the user has removed an object from the table; and the user has dragged an object along the table. Events are queued, and accessed via polling. This design is compatible with game architectures: games poll *EquisFTIR* in their main frame loop to determine what events have occurred since the last iteration. The provided events allow easy programming of basic functions such as selecting, dragging and activating game elements. In addition, these events provide the necessary hooks to implement basic gestures such as pan, zoom or rotate. Session id tracking is scheduled for an upcoming release.

The *EquisFTIR* library is used within Microsoft's XNA Studio, a freely available environment supporting the development of 2D and 3D games. XNA Studio is compatible with PCs, the Xbox 360, the Zune, and now, FTIR tabletops.

Output is sent to the projector. A callibration step (built in to *EquisFTIR*) is required when the table is set up to establish the correspondence between positions in the camera image and the projected display. Events are then expressed in terms of display positions, consistently with how mouse events are reported by standard PC input libraries.

6. LIMITATIONS

While the *EquisFTIR* hardware and software library provide an excellent basis for research into tabletop games, the approach does have limitations (some of which are in fact shared by tables costing tens of thousands of dollars!)

Players interacting with games via a mouse or game controller expect pixel-level accuracy in pointing tasks. We experience accuracy typically within the range of 0.5 cm to 1 cm. As pointed out by Mattar, this means that games should be designed to tolerate this level of potential error [10]. Error comes from a combination of the pointing "device" (it is hard to position a finger at pixel-accuracy), the low input resolution of the camera, and errors in callibration. Callibration errors can be reduced by careful positioning of the camera and projector, reducing the work that the software has to do in mapping bitmap positions to inputs.

As shown in figure 4, we have used books and binders to carefully prop up and position the camera and projector. Because of this, the table becomes hard to transport.

The *EquisFTIR* library must choose appropriate thresholds to distinguish between real events and background noise. Too low a threshold leads to false positives; too high a threshold leads to unnatural interaction where users must conciously press hard to activate a touch event. *EquisFTIR* has been tuned to find acceptable an acceptable tradeoff between these problems. These problems can be mitigated with more expensive cameras where noise is less of an issue

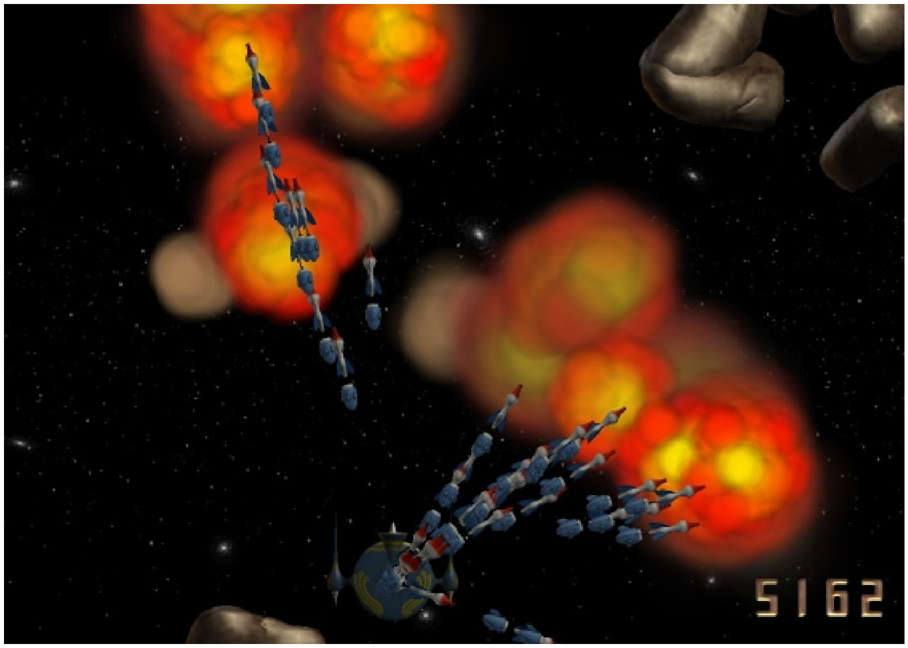

Figure 6: The *Asterocks* game

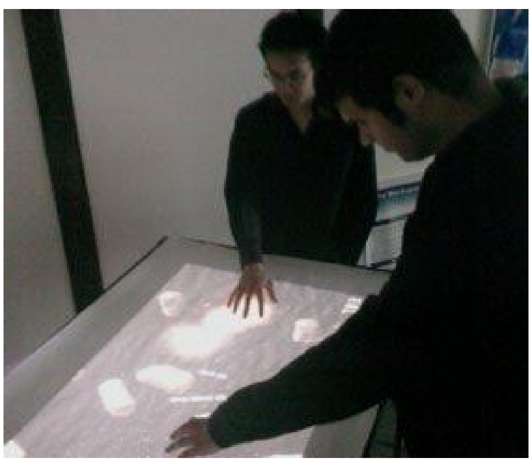

Figure 7: Cooperative play of *Asterocks*

and proper shielding to reduce sources of IR noise.

Finally, the light emitting diodes that are used in the table have a limited lifetime, requiring them to be occasionally replaced. As described above, using a large amount of LEDs allows them to be run at lower power, increasing their life span.

In consumer versions of such tables, many of these problems could be addressed through creation of better housing and more robust electronics.

7. EXPERIENCE: ASTEROCKS

To demonstrate the effectiveness of *EquisFTIR* in creating tabletop games, we have developed *Asterocks*, a cooperative multiplayer game reminiscent of Atari's 1980 Missile Command arcade game. Figure 6 shows a close-up of the game, while figure 7 shows the game as played by two people.

In *Asterocks*, players stand together at the table. Asteroids "fall" toward the players. If an asteroid falls off the edge of the table, the players lose points. The players can fire missiles at the asteroids by touching a location on the table. A missile is fired from the players' gun, and when it arrives at the point, it explodes, destroying any nearby asteroids. Players gain points for destroying asteroids, and spend points for the missiles they use.

The game illustrates several of the benefits of tabletop play. The touch interaction feels natural: in our experience, new players watch someone else for a few moments, and immediately understand how to interact with the game, even when they have never used a tabletop surface before. The gameplay is cooperative, where two or more players stand beside each other, typically each responsible for a part of the display. We find there to be much laughter and joking when two people play, as the proximity of players around the tabletop and the shared input device provides intimacy. The multi-touch input afforded by FTIR tables allows players to fire missiles at the same time without turn-taking.

The *EquisFTIR* library provides instantaneous gameplay. Placement of missiles is sufficiently accurate that errors are unnoticeable. Players have no sense of interacting with an input device whose behaviour must be learned; instead, they simply touch the table at the desired location, and the missile fires immediately.

While we have not performed controlled studies to measure *Asterocks*'s entertainment value, we have considerable anecdotal evidence that it is fun and engaging to play. Hundreds of people have played *Asterocks* in our lab. During lab tours of high school students, we experience long lineups while people eagerly wait to get their turn. The *EquisFTIR* table has proven robust and stable under such heavy use. We believe that *Asterocks* is an example of how effectively a low-cost tabletop gaming infrastructure can provide a social and engaging gaming experience.

8. CONCLUSIONS

In this paper, we have presented *EquisFTIR*, a low-cost infrastructure for developing tabletop games. We have argued that tabletop gaming provides a novel experience, where physical colocation and tangible sharing of game artifacts provides similar intimacy to friends' experience when playing a board game. *EquisFTIR*'s contribution is that it provides a complete infrastructure for creating and playing tabletop games using Microsoft's XNA Studio, and that this infrastructure can be built considerably cheaper than commercial electronic tabletop surfaces. It is our hope that this will make tabletop technology available to a wider community, allowing further experimentation with this novel style of digital game.

Acknowledgments

We gratefully acknowledge the support of the Natural Science and Engineering Research Council and the NECTAR research network in carrying out this work.

9. REFERENCES

[1] Dietz, P., and Leigh, D. DiamondTouch: a multi-user touch technology. In *Proceedings of the 14th annual ACM symposium on User Interface Software and Technology* (2001), ACM, pp. 219–226.

[2] Group, N. Touchlib: A multi-touch development kit. http://nuigroup.com/touchlib/.

[3] Han, J. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User Interface Software and Technology* (2005), ACM, pp. 115–118.

[4] Kaltenbrunner, M., and Bencina, R. reacTIVision: a computer-vision framework for table-based tangible interaction. In *Tangible and Embedded Interaction* (2007), pp. 69–74.

[5] Kaltenbrunner, M., Bovermann, T., Bencina, R., and Costanza. TUIO - a protocol for table based tangible user interfaces. In *Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation* (2005).

[6] Kamiyama, K., Vlack, K., Mizota, T., Kajimoto, H., Kawakami, N., and Tachi, S. Vision-based sensor for real-time measuring of surface traction fields. *IEEE Comput. Graph. Appl. 25*, 1 (2005), 68–75.

[7] Lee, S., Buxton, W., and Smith, K. A multi-touch three dimensional touch-sensitive tablet. *SIGCHI Bull. 16*, 4 (1985), 21–25.

[8] Magerkurth, C., Memisoglu, M., Engelke, T., and Streitz, N. Towards the next generation of tabletop gaming experiences. In *Proceedings of Graphics Interface 2004* (2004), ACM, pp. 73–80.

[9] Malik, S., and Laszlo, J. Visual touchpad: a two-handed gestural input device. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces* (2004), ACM, pp. 289–296.

[10] Mattar, C. PinguTouch: Investigating multi-touch technology for collaborative casual gaming. Master's thesis, RWTH Aachen University, December 2007.

[11] Mazalek, A., Mironer, B., O'Rear, E., and Devender, D. V. The TViews table role-playing game. In *Proc. 4th International Symposium on Pervasive Gaming Applications* (2007), Shaker Verlag, pp. 127–134.

[12] Microsoft, Inc. Microsoft Expression Blend, 2008.

[13] Microsoft, Inc. Microsoft Surface, 2008.

[14] Microsoft, Inc. Microsoft XNA Framework, 2008.

[15] Microsoft, Inc. Windows Presentation Foundation, 2008.

[16] Nicol, K., and Hennig, E. Apparatus for the Time-Dependant Measurement of Physical Quantities. *U.S. Patent 4,134,063* (1979).

[17] Nilsen, T., and Looser, J. Tankwar – tabletop war gaming in augmented reality. In *Proceedings of the 2nd International Workshop on Pervasive Gaming Applications (PerGames)* (2005).

[18] Piccione, P. In search of the meaning of Senet. *Archaeology* (July/August 1980), 55–58.

[19] Plumb, T. Low-tech gamers remain loyal to the board. In *Boston Globe* (November 29 2007).

[20] Rekimoto, J. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (2002), ACM, pp. 113–120.

[21] Rekimoto, J., Oka, M., Matsushita, N., and Koike, H. HoloWall: interactive digital surfaces. In *ACM SIGGRAPH 98 Conference abstracts and applications* (1998), ACM, p. 108.

[22] Smith, J., Graham, T., Holman, D., and Borchers, J. Low-Cost Malleable Surfaces with Multi-Touch Pressure Sensitivity. *TABLETOP'07* (2007), 205–208.

[23] Tactex. Smart Fabric Technology.

[24] The International Museum of Video Games. Ms. Pacman, 2008. http://tinyurl.com/6wdo4.

[25] Tse, E., Greenberg, S., Shen, C., and Forlines, C. Multimodal multiplayer tabletop gaming. *Comput. Entertain. 5*, 2 (2007), 12.

[26] Vlack, K., Mizota, T., Kawakami, N., Kamiyama, K., Kajimoto, H., and Tachi, S. GelForce: a vision-based traction field computer interface. In *CHI '05 extended abstracts on Human Factors in Computing Systems* (2005), ACM, pp. 1154–1155.

[27] Vogt, F., Chen, T., Hoskinson, R., and Fels, S. A malleable surface touch interface. In *ACM SIGGRAPH 2004 Sketches* (2004), ACM, p. 36.

[28] Westerman, W., and Elias, J. Method and Apparatus for Integrating Manual Input. *U.S. Patent 6,323,846* (2001).

[29] Wilson, A. TouchLight: an imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th international conference on Multimodal interfaces* (2004), ACM, pp. 69–76.