Computer-Supported Cooperative Work Group
Faculty of Media
Bauhaus-University Weimar

Technical Report #: BUW-CSCW-2007-02 November 2007

The cueTable Cooperative Multi-Touch Interactive Tabletop: Implementation and User Feedback

Tom Gross, Mirko Fetter, Sascha Liebsch
Faculty of Media, Bauhaus-University Weimar, Germany
({tom.gross, mirko.fetter, sascha.liebsch}(at)medien.uni-weimar.de)

Contact:
Prof. Dr. Tom Gross
Faculty of Media
Bauhaus-University Weimar
Bauhausstr. 11, Room 113
99423 Weimar, Germany

E: tom.gross@medien.uni-weimar.de
T.: (++49-3643) 58-3733
F: (++49-3643) 58-3709
W: http://www.uni-weimar.de/medien/cscw/

Gross, T., Fetter, M. and Liebsch, S. The cueTable Cooperative Multi-Touch Interactive Tabletop: Implementation and User Feedback. Technical Report # BUW-CSCW-2007-02, Computer-Supported Cooperative Work Group, Faculty of Media, Bauhaus-University Weimar, Bauhausstr. 11, 99423 Weimar, Germany, Nov. 2007.

The cueTable Cooperative Multi-Touch Interactive Tabletop: Implementation and User Feedback

Tom Gross, Mirko Fetter, Sascha Liebsch
Faculty of Media, Bauhaus-University Weimar, Germany
(<firstname.lastname>(at)medien.uni-weimar.de)

Abstract. We developed a multi-touch interactive tabletop as a base technology to explore new interaction concepts for cooperative multi-touch applications. In this paper we explain how to build a cooperative multi-touch interactive tabletop with standard and low-budget hardware and little implementation effort. We present a software application we developed. And we report on user feedback to the tabletop and the applications.

Author Keywords. Interactive Tabletop, Cooperative Multi-Touch, Implementation, User Feedback.
ACM Classification Keywords. H.5.2 [Information Interfaces and Presentation]: User Interfaces—GUI, User-Centred Design; H.5.3 [Information Interfaces and Presentation]: Group and Organisation Interfaces – Computer-Supported Cooperative Work.

1 Introduction

Interactive tabletops provide horizontal large-screen surfaces that allow small groups of users to interact with software applications via touch. A number of interactive tabletops and a range of applications for interactive tabletops have been developed. For instance, Mitsubishi Electronics Research Laboratories have developed the DiamondTouch table for more than ten years and sell it commercially [MERL 2007]; and Jefferson Y. Han has recently presented an approach for a low-cost touch-based solution for interactive walls. Among the special-purpose applications that have been developed typically for same-time same-place scenarios are particularly applications for sharing photographs [Frohlich *et al.* 2002], and for sharing maps and urban planning [Sugimoto *et al.* 2004].

Despite this increasing wide-spreading of interactive tabletops, developing a cooperative multi-touch interactive tabletop is not a straight-forward task and the actual use in cooperative multi-touch scenarios is under-researched.

In this paper we present cueTable—a cooperative multi-touch interactive tabletop we developed as a base technology to explore new interaction concepts for cooperative multi-touch applications (Figure 1 shows the cueTable with the front side open to give an impression of the inside). We explain how to build a cooperative multi-touch interactive tabletop with standard and low-budget hardware and little implementation effort. We share technical information on the hardware and software setup as well as on a game application for the tabletop. And we report on user feedback to the tabletop and the applications. Finally, we report on related work, and draw conclusions.

Figure 1. cueTable from inside and outside.

2 Implementation

The cueTable aims to support small groups of up to seven users in same-time same-place settings. The cueTable furthermore aims to allow cooperative multi-touch—that is, users should be able to interact with the table in parallel with each other, and with a single touch of one finger or multi-touch of two or more fingers of the same or both hands. And, the cueTable should be low-cost. For this purpose the cueTable combines a collection of mechanics and hardware, algorithms and software.

2.1 cueTable Hardware and Mechanics

The cueTable is composed of the following standard hardware: a self-made table with a surface of 136x112 cm (53.4x44 in) and a height of 100 cm (39.5 in) covered by tracing paper and with an acrylic glass sheet with a surface of 120x90 cm (47x35.5 in) and a thickness of 0.5 cm (0.2 in), equipped with 32 Osram SFH485 (880nm) LEDs, a Philips SPC900NC camera with a wide-angle lens and an IR filter, a Toshiba TLP-T60M projector with two mirrors, and a standard Macintosh PowerPC G4 1.8 Gigahertz or a standard PC Intel Centrino Dual 1.6 Gigahertz with a dual-core processor.

The mechanics of the cueTable are based on the optical phenomenon of Frustrated Total Internal Reflection (FTIR). Total Internal Reflection (TIR) describes the behaviour of light at the border between two media with different indices of refraction. Light inside the medium with the higher index is reflected at the medium's surface as long as the angle of incidence is not going below a certain critical angle with respect to the surface normal. While this condition is fulfilled, all light is kept within the medium. When a third medium of a higher refractive index is placed on the medium, the light is no longer reflected in the area, but frustrated (thus FTIR occurs)—that is, the light escapes to the outside.

In our setup, 32 LEDs are placed at two opposite sides of the acrylic sheet in order to cast IR light into it. The edges of the sheet are covered with reflective tape to prevent the light from escaping at the opposite side. Upon the touch of a finger, the light is frustrated and creates a light spot at the finger's area. The camera with the wide-angle lens captures the light spots created by the fingers on the acrylic sheet from below. The wide-angle lens allows us to cover a large area without the need to place the camera in great distance from the table surface, thereby circumventing problems concerning the mirrors and projector. The IR filter takes out the visible light from the captured image and only lets the IR light pass through. The projector, mounted on the back side of the table, points downwards. Its image is deflected by two mirrors, enabling us to project the image from a short distance onto the tracing paper that acts as a screen affixed under the transparent acrylic sheet.

2.2 cueTable Algorithm

In order to detect the places on the cueTable tabletop surface where touch occurs, a vision-based blob tracking algorithm is used. This algorithm takes as input an image from the camera, detects the IR light blobs in this image, and generates events from these blobs. Moreover, the blobs' relation is kept throughout successive frames in order to detect sequences of blobs (e.g., a drag gesture with a finger).

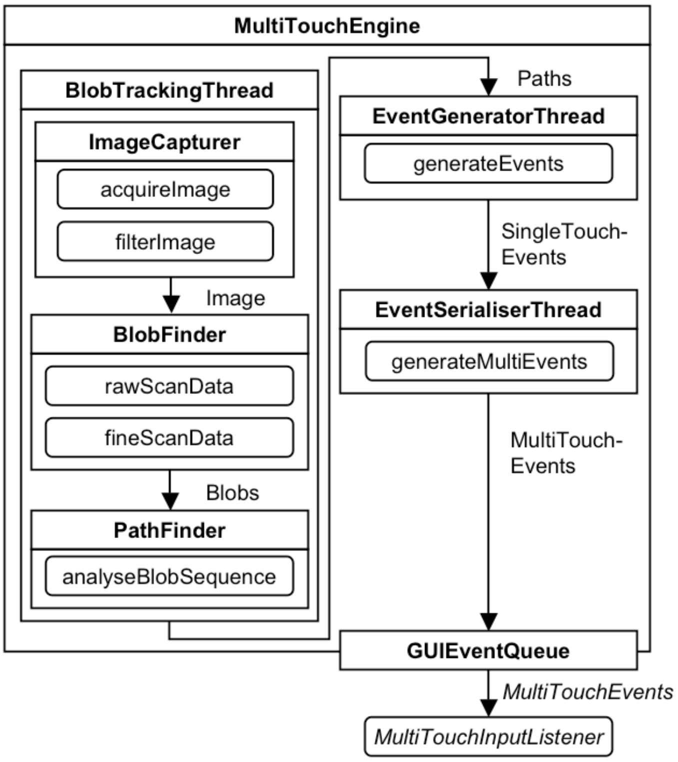

Figure 2. cueTable information flow in MultiTouchEngine.

The MultiTouchEngine (cf. Figure 2) is responsible for blob detection and event generation. It basically consists of three threads which can partly run in parallel, allowing to benefit from multi-(core) processor systems. The first of the three threads, the BlobTrackingThread, uses the ImageCapturer to acquire the camera images in the acquireImage and filter them using algorithms for background subtraction, contrast enhancing, and undistortion in filterImage. Then, the image gets analysed by the BlobFinder, which detects blobs through two scans, the rawScanData to identify possible blobs and the fineScanData to verify and detail. Next, the found blobs are processed by the PathFinder, which converts blobs into path information by comparing the current set of blobs with the set from the previous frame in analyseBlobSequence. After this, generateEvents in the second thread, named EventGeneratorThread, converts the paths into SingleTouchEvents, that is, touch events for single fingers. The EventSerialiserThread finally aggregates SingleTouchEvents on the same target object into MultiTouchEvents in generateMultiEvents. Moreover, the EventSerialiserThread acts as a buffer for events; it ensures that events originating from different frames are not confused. Finally, the MultiTouchEvents are inserted into the system's GUIEventQueue, which is the interface of the MultiTouchEngine to the applications that want to use multi-touch. These applications just have to implement a MultiTouchInputListener in order to receive MultiTouchEvents from the engine (in italics because they are not part of the MultiTouchEngine).

2.3 cueTable Implementation

The MultiTouchEngine was developed on the Java 2 Standard Edition 5.0 platform. For image acquisition on Mac OS X version 10.4.9 Quicktime version 7.2.0, and for Windows XP SP2 the Java Media Framework (JMF) version 2.1.1e are used. Additionally, the Macam drivers version 0.9.1 are required to use all cameras under Mac OS X. With Quicktime and JMF, all common USB and Firewire cameras can be accessed.

The camera image, after being filtered, is analysed with a very quick, yet simple algorithm, which firstly uses a coarse and afterwards a finer raster instead of processing every single pixel. Likewise, the filters are only applied to the pixels covered by the raster. The threaded MultiTouchEngine architecture described above allows for quick processing of the images in parallel to event generation and dispatching. In ideal cases, the next image is already being analysed while the latest events are being processed, so delay can be minimized.

The MultiTouchEvents generated by the cueTable software are compatible with the existing Java concepts, that is, AWT and Swing. The MultiTouchEvent class is derived from the Java MouseEvent class; MultiTouchEvents are inserted into the Java AWT Event Queue, and treated like mouse events there. Therefore, every existing Swing application may be controlled by touch events, even though without multi-touch features. For real multi-touch applications, the events received from the AWT Event Queue can simply be cast back to MultiTouchEvent and be processed by implementing the MultiTouchInputListener interface.

3 The cueTable Puh Game Application

In order to test the cueTable and to get user feedback in a cooperative multi-touch scenario we developed a Puh game application that is very similar to Atari's Pong game. In the Puh game two teams consisting of two players each play against each other. They use a paddle to shoot a ball from their own side of the table to the other side of the table. If the team succeeds in moving the paddle to the place where the ball comes to their side of the table, the ball bounces back to the other side and the game continues. If the team fails to do so and the ball hits the edge of the own side of the table, the other team scores a point and a new ball comes in from the centre.

Figure 3. cueTable, Puh game, and four users.

In the Puh game for the cueTable (cf. Figure 3) the ball is a dot on the screen moving between the two fields of the teams. The paddle is not visible per default; when a player touches the table with two fingers, the paddle is drawn as a line between the two fingers. If more than two fingers are detected on a field, the Puh game draws the paddle as a line on the shortest possible distance of two fingers (e.g., assume we have finger a, b, and c; the distance from a to b is 2, from b to c is 3, and from c to a is 3.6; then the paddle is drawn between a and b). Furthermore, a threshold limits the maximal size of the paddle to prevent cheating by spanning a paddle over the full table width.

4 Initial User Feedback

The cueTable and the Puh game were tested in informal settings with about 100 different users (forming about 25 settings of two teams with two players) at our Cooperative Media Lab Open house from 13 to 15 July 2007. Half of them were students of our university from diverse study programmes, with an age ranging from 19 to 27 years; and half of them were visitors, between 27 and 50 years. The game was played between 10 and 15 minutes. The duration depended on the score: if the result was even after 10 minutes, many teams insisted on coming to a clear result. We asked the users to think-aloud while playing, and we made unstructured interviews after the games.

4.1 Findings

From the study we got three different types of findings: general findings about the tabletop per se; findings about the users and their learning progress; and findings about the tabletop related to the Puh game.

Tabletop. All users understood the *basic interaction paradigm* of touching the screen. However, users tended to assume that it was possible to touch everywhere, and expect reactions. When those users touched the surface with a single finger, they wondered about the non-appearance of a reaction. This often led to diffidence for the next tries. These users were a little disappointed as their expectations have not been met, and they assumed that they may not have understood the interface and usage right. Still, in order to explore multi-touch, this reduction to multi-touch—that is, no support for single-touch—was intended.

Many users held their hands in *exhausting positions*: although it was possible to place the fingers flatly on the surface, many users chose a perpendicular position for their hands in order to press harder. The main reasons were twofold: users assumed they had to press, because they were used to from other systems (e.g., some ATMs and ticket vending machines in Germany require strong pressing); and some users assumed they had to press hard, because they had the impression that the cueTable was just a research prototype and not (yet) sensitive to gentle touch.

Users. Looking at the learning effects, we could identify three different types of users: *fast-learners* who understood the concepts and possibilities upon our first explanation (they sometimes even helped out their game team colleagues if they did not play and react fast enough), and who were very active in the game from the very beginning; *learners* who diffidently tried out the cueTable after the first explanation, and who did not clearly understand all concepts and causalities behind their discoveries, but who understood the concepts after repeated explanations, and who then get quite active in the team; and *slow-learners* who stayed insecure and diffident even after multiple explanations and who stay rather passive during the game. These types of users could be identified independently of their age.

Tabletop and Puh. The specific concepts of the Puh game and the *affordances of the paddle* were not clear to everybody. Some users did not accept the restriction of only using two fingers; they tried to use more fingers in order to create multiple paddles; and even despite our explications, many kept on trying. Users often tried to create the paddle with one or both fingers outside of their personal playing zone (i.e., the half circles in front of users). Some users took the tangibility too literal: they tried to push to ball by moving the paddle towards it or asked if this was possible. Also, many tried to touch the ball itself at the beginning.

Another learning effect was discovered concerning the *latency* of the system—the Puh game in the current implementation has an average latency of about 70 ms (with the system running on 15 fps) to detect the multi-touch and create the paddle, so users have to react to the ball in advance. At the beginning most users touched the cueTable upon arrival of the ball and missed the ball, whereas later they anticipated the trajectory and touched early enough. Yet, some users never learned this anticipation.

There was a clear wish for *increasing the ball speed* for all users: at the beginning of a game the new teams scored many goals, because the players did not create the paddle fast enough, while later there were considerably less goals, because of the faster reactions.

Several users even asked for a higher ball speed, which we provided. Overall the game has four categories of ball speed (with a speed factor as follows: slow is 1, medium is 1.5, fast is 2, fastest is 2.5).

4.2 Discussion

From the three-day testing of the cueTable and its Puh game and the findings above we can derive the following lessons.

The design of any interactive system always start with an early focus on users, besides tasks and technology. This also holds true for the design and implementation of tangible and embedded systems; as Hornecker & Buur [2006] write: '[t]angible interaction is not restricted to controlling digital data'. The design for tangible and embedded interaction has to take into account the digital realm as well as the physical realm, in which the users are in. Consequently, it needs to be rooted in a thorough understanding of the properties and affordances of the digital artefacts and the physical artefacts involved—and particularly about the users' background and experiences from both areas when they interact with the tangible and embedded system at hand. Furthermore, the novel and emerging properties can be difficult to anticipate for both—designers, and users.

Several of our findings provide evidence for that. For instance, the fact that users were confused that they could construct paddles with their multi-touch, but that they could not touch the ball. So, obviously the interaction with the paddle was (besides the little latency) realistic enough to make them assume that they can touch and manipulate other items as well.

Another important issue is the cooperative interaction with the cueTable tabletop. The interaction among the team players could be monitored, but the reasons for the behaviour were not always clear. Scott et al. [2004] did a study on cooperation via tabletops and found similar distinctions of individual and group territories and patterns of crossing their boundaries.

5 Related Work

Many interactive tabletops have been presented in the fields of HCI and UbiComp, where they are mostly called interactive tabletops; and in the field of CSCW, where they are often called single-display groupware. The work that is most closely related with the cueTable is Jefferson Y. Han's work on multi-touch sensing based on frustrated total internal reflection [Han 2005]. Some studies such as [Forlines *et al.* 2007] report on the usability of tabletops, and some studies such as [Hornecker 2005; 2004] also report on the interaction among users in cooperative settings surrounding tabletops. Finally, other users have emphasis the importance of play for the design and evaluation of ubiquitous systems [Block *et al.* 2004].

6 Summary and Conclusions

In this paper we presented the design and realisation of the hardware and software of the cueTable, a low-cost interactive tabletop supporting cooperative multi-touch interaction.

We have reported on initial findings from a user test of the cueTable with a cooperative multi-touch Puh game. This paper aims to contribute both knowledge and experience concerning the technology of the cueTable, and stress the fact that for a better understanding of tangible and embedded interaction the actual building and testing is crucial.

Acknowledgments

We thank all members of the Cooperative Media Lab—esp. Manuel Meyer, Clement Welsch, Thilo Paul-Stueve—for contributing to the cueTable concept and implementation.

References

Block, F., Schmidt, A., Villar, N. and Gellersen, H.W. Towards a Playful User Interface for Home Entertainment Systems. In *Proceedings of the European Symposium on Ambient Intelligence - EUSAI 2004* (Nov. 8-10, Eindhoven, NL). Springer-Verlag, Heidelberg, 2004. pp. 207-217.

Forlines, C., Wigdor, D., Shen, C. and Balakrishnan, R. Direct-Touch vs. Mouse Input for Tabletop Displays. In *Proceedings of the Conference on Human Factors in Computing Systems - CHI 2007* (Apr. 28-May 3, San Jose, CA). ACM, N.Y., 2007. pp. 647-656.

Frohlich, D., Kuchinsky, A., Pering, C., Don, A. and Ariss, S. Requirements for Photoware. In *Proceedings of the ACM 2002 Conference on Computer-Supported Cooperative Work - CSCW 2002* (Nov. 16-20, New Orleans, LO). ACM, N.Y., 2002. pp. 166-175.

Han, J.Y. Low-Cost Multi-Touch Sensing through Frustrated Total Internal Reflection. In *Proceedings of the ACM Symposium on User Interface Software and Technology - UIST 2005* (Oct. 23-27, Seattle, Washington). ACM, N.Y., 2005. pp. 115-118.

Hornecker, E. A Design Theme for Tangible Interaction: Embodied Facilitation. In *Proceedings of the Nineth European Conference on Computer-Supported Cooperative Work - ECSCW 2005* (Sept. 18-22, Paris, France). Springer-Verlag, Heidelberg, 2005. pp. 23-43.

Hornecker, E. and Buur, J. Getting a Grip on Tangible Interaction: A Framework on Physical Space and Social Interaction. In *Proceedings of the Conference on Human Factors in Computing Systems - CHI 2006* (Apr. 22-27, Montreal, Canada). ACM, N.Y., 2006. pp. 437-446.

MERL. *MERL - DiamondTouch*. Mitsubishi Electronic Research Laboratories, http://www.merl.com/projects/DiamondTouch/, 2007. (Accessed 2/10/2007).

Scott, S.D., Carpendale, S.T. and Inkpen, K.M. Territoriality in Collaborative Tabletop Workspaces. In *Proceedings of the ACM 2004 Conference on Computer-Supported Cooperative Work - CSCW 2004* (Nov. 6-10, Chicago, IL). ACM, N.Y., 2004. pp. 294-303.

Sugimoto, M., Hosoi, K. and Hashizume, H. Caretta: A System for Supporting Face-to-Face Collaboration by Integrating Personal and Shared Spaces. In *Proceedings of the Conference on Human Factors in Computing Systems - CHI 2004* (Apr. 24-29, Vienna, Austria). ACM, N.Y., 2004. pp. 41-48.