Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography

Matthew G. Gorbet
Tangible Media Group
MIT Media Laboratory
20 Ames Street
Cambridge, MA 02139 USA
+1 617 253 0128
mgorbet@media.mit.edu

Maggie Orth
Hyperinstruments Group
MIT Media Laboratory
20 Ames Street
Cambridge, MA 02139 USA
+1 617 253 0804
morth@media.mit.edu

Hiroshi Ishii
Tangible Media Group
MIT Media Laboratory
20 Ames Street
Cambridge, MA 02139 USA
+1 617 253 7514
ishii@media.mit.edu

ABSTRACT

This paper presents a system for interacting with digital information, called Triangles. The Triangles system is a physical/digital construction kit, which allows users to use two hands to grasp and manipulate complex digital information. The kit consists of a set of identical flat, plastic triangles, each with a microprocessor inside and magnetic edge connectors. The connectors enable the Triangles to be physically connected to each other and provide tactile feedback of these connections. The connectors also pass electricity, allowing the Triangles to communicate digital information to each other and to a desktop computer. When the pieces contact one another, specific connection information is sent back to a computer that keeps track of the configuration of the system. Specific two and three-dimensional configurations of the pieces can trigger application events.

The Triangles system provides a physical embodiment of digital information topography. The individual tiles have a simple geometric form which does not inherit the semantics of everyday physical objects. Their shape, size, and connectors encourage rapid rearrangement and exploration of groups of Triangles. The infinitely reconfigurable 2D and 3D topographies of the Triangles system create a new language for tangible interface.

Keywords

Interface design, tangible interface, physical interface, graspable interface, digital connector, physical connector, magnetic connector, tangible bits.

INTRODUCTION

Physical building blocks are powerful tools for thought and play. They allow quick construction and manipulation of structures through two-handed tactile interaction. This enables an easy exploration and understanding of the blocks' interactions with one another, and also of physical properties like balance, gravity, composition, and form (Fig. 1).

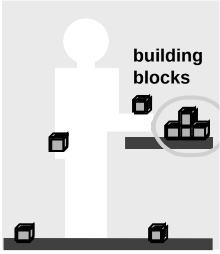


Fig. 1: physical world

However, these blocks are usually passive objects, and computers can not recognize or interpret the composite structures of blocks.

In comparison, personal computers allow users to create, edit, and store complicated digital information structures. Manipulating these digital objects is heavily graphical-display-oriented, with the information relationships represented visually on a screen. Although computers support malleable information manipulation, current interactions are largely constrained by the mouse and keyboard (Fig. 2). Shifting focus between the 'controls' of the computer (keyboard and mouse) and the 'data' (as represented by visual images on the screen) can be a source of confusion and delay in working with digital information [7]. The affordances of physical building blocks are inherently richer than what current GUI affords through conventional "direct manipulation" techniques. How can we get the best of both the digital and physical worlds?

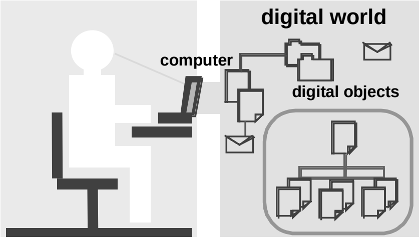


Fig. 2: looking into the digital world

Increasingly, we have seen an interest in physical systems for interacting with information [2, 3, 6, 11, 12, 13]. The Triangles system was designed to address this issue by physically embodying the *topography* of digital information – the relationships and connections between data elements – so that users can directly feel and manipulate the digital information with their two hands (Fig. 3). In the same way that building blocks help explore physical properties like balance and composition, Triangles were designed to provide easier interaction with and understanding of data space.

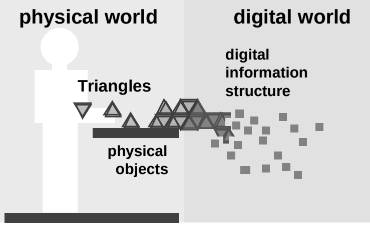


Fig. 3: manipulating the digital world

Unique contributions of the Triangles system are the following:

- The introduction of a concept of joint digital/physical objects which, by design, embody information and provide a general physical means to make a computer aware of their topography
- A user interface that
  - takes advantage of the human ability to easily manipulate physical objects with both hands,
  - provides a persistent physical indicator of state, and
  - allows many people to use it simultaneously.
- A new design for an electrical and magnetic connection mechanism that provides
  - a low-bandwidth, computationally inexpensive means for sensing the manipulation of complex physical structures, and
  - tactile feedback to users in attach and detach operations.

In this paper, we introduce the Triangles system, along with four Triangles applications. We discuss salient features of these applications and lessons learned from developing them. We also describe the iterative design and development of the Triangles system itself, including hardware, software and conceptual design. We begin with a brief scenario from one of our current Triangles applications.

CINDERELLA

> *Two children are sitting on the floor, with a set of ten triangular tiles in between them. On the tiles are images from the popular fairy-tale Cinderella. One child spreads the tiles out on the floor, looking through them. She picks one up, with an image of Cinderella's evil stepmother on it. The other child hands her a tile with Cinderella's house on it, and she brings them together. They snap together magnetically with an audible click, and instantly the step-mother's voice is heard (emanating from a set of nearby speakers) echoing through the halls of the house: "Cinderella! Cinderella?! Oh, where* is *that girl?…"*
>
> *The stepmother's voice continues, calling out for Cinderella, until one of the children picks up the image of Cinderella sweeping, and snaps it onto another edge of the stepmother triangle.*
>
> *"Ah,* there *you are! I need you to clean up this messy room. And when you're done with that, there's laundry to fold downstairs," says the stepmother. Cinderella obeys, and sounds of sweeping and humming are heard. Next, the children remove the stepmother. As soon as she is gone, Cinderella explains how she longs to be free from her stepmother's tyrannical rule. Just then, the children attach an image of the stepsisters, and we hear them burst into the room, teasing Cinderella…*

This scenario is an example of an interaction with one of the storytelling applications that was developed for the Triangles system. These applications will be discussed in detail later in this paper.

THE TRIANGLES SYSTEM

Overview

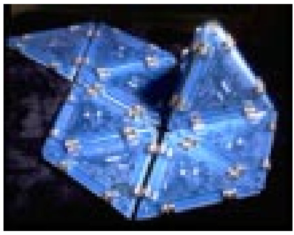


Photo ©Webb Chapell

Fig. 4: The Triangles

The Triangles system consists of a set of identical flat, plastic equilateral triangles, each with a microprocessor inside and a unique digital ID. The Triangles each have different images or markings on them, which can be changed depending on the intended application. They have magnetic connectors on their edges which allow easy physical interconnection, as shown in Figure 4. The connectors also pass electricity, and the Triangles use them to communicate digital information to each other and to a desktop computer. Thus, when the pieces contact one another, specific information about that connection is sent back to the computer, which keeps track of the history of connections and current topography of the system.

The triangles can be used to make two- and three-dimensional objects whose exact configuration is known to the computer. Changes in this configuration can trigger specific computer-controlled events. For example, connecting or disconnecting two specific Triangles could provide access to a specific web page, or cause a digitized audio sample to play. Events can also be associated with specific groupings of Triangles, rather than simple connection or disconnection of individual triangles. The actual output event that results from a given interaction depends on the application being used, but can be practically anything that a computer can control. The roles played by each part of the Triangles system are illustrated in Figure 5.

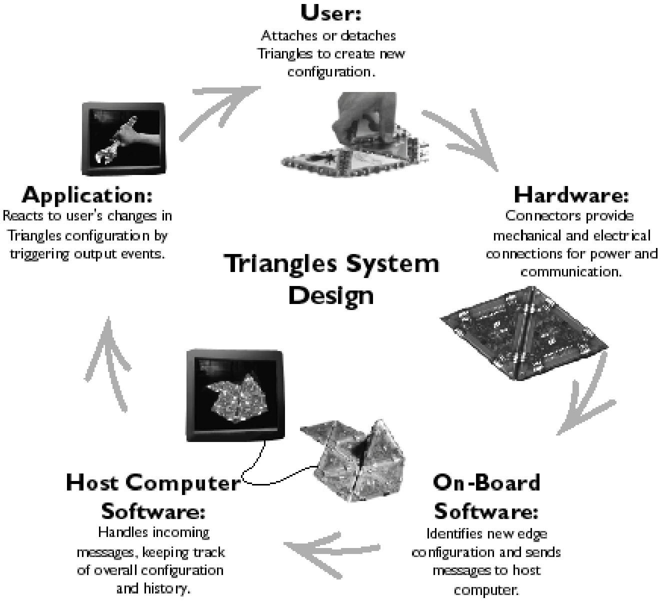

User:
Attaches or detaches Triangles to create new configuration.
Hardware:
Connectors provide mechanical and electrical connections for power and communication.
Triangles System Design
On-Board Software:
Identifies new edge configuration and sends messages to host computer.
Host Computer Software:
Handles incoming messages, keeping track of overall configuration and history.
Application:
Reacts to user's changes in Triangles configuration by triggering output events.

Fig. 5: Elements of the Triangles system and their roles.

The triangular tiles that make up the Triangles system have many affordances that help them to create a new language for interacting with digital information:

- They are easily handled and sorted with two hands.
- They can tile or make branching structures in two dimensions or create three-dimensional polyhedra.
- They can be manipulated by more than one person at a time.
- Their connections remain physically persistent, as a physical reminder of the state of the system.
- They provide tactile feedback when connections are made or broken, through their magnetic connectors.
- Their very general shape allows them to represent any type of digital information without carrying the semantic weight often associated with everyday physical objects.

Why Triangles?

The aim is to create a generic object that can represent the topographical relationships of information elements. Keeping the form of the objects generic enables them to be all about connections and relationships -- a simple geometric shape has less semantic loading [1] than familiar objects such as a digitally augmented set of dolls or books. Pieces that can be tiled are appropriate to physically embody the idea of connection or association, and the flat faces of tiles enable them to carry a pictorial representation of the information which they represent.

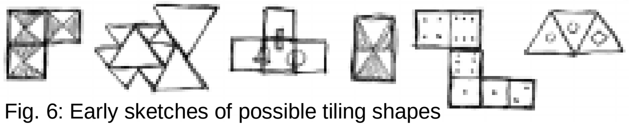


Fig. 6: Early sketches of possible tiling shapes

With these concerns in mind, many shapes were considered for the individual tiles (Fig. 6). Equilateral triangles are the simplest shape that can tile in two dimensions, so triangular tiles have the minimum number of sides required to physically reflect the complex possibilities of branching structures and relations in computer information. The three sides of a Triangle suggest junctions with one input and two outputs, or vice-versa. In contrast, objects with only two points of connection, like stacking blocks, suggest structurally linear data relationships, with only one input and one output. Another advantage to Triangles over other shapes like squares is that the triangular pieces allow robust three-dimensional (polyhedral) forms to be built.

Even though the interaction with Triangles is fairly simple, allowing users to quickly create and manipulate associations between the elements, the configurations that can be created are quite complex, due to the combinatorics of equilateral triangles. Since any edge of each tile can create a uniquely identifiable physical connection with any edge of another triangle, the number of unique configuration possibilities for a set of tiles is given by:

$$x \geq \frac{3^n (n+1)!}{6}$$

where n is the number of triangles and x is the number of possible unique configurations.

This means that for a set of just four triangles there are 1,620 possible 2D configurations. As the number of triangles increases, this number of configurations quickly grows to millions. Such a potentially vast world of information combined with the simplicity of interaction on the part of the user makes Triangles flexible and powerful, while creating interesting design issues for applications.

APPLICATIONS

The Triangles system presents a very specific means for interaction, lending itself to certain types of applications over others. For example, while it might be very well suited to exploring a non-linear narrative or configuring an audio/visual system, it would not be appropriate for applications which require a rich input vocabulary, like text editing or technical illustration.

The Triangles system has been seen and used by hundreds of people during the first year of its development. These include researchers from a wide variety of fields, artists, children visiting our laboratory, and representatives of many diverse industries. Our observations of these users' interactions with the system have contributed a great deal to its development. Following, we discuss four applications that we developed for the Triangles system, and what we learned from each.

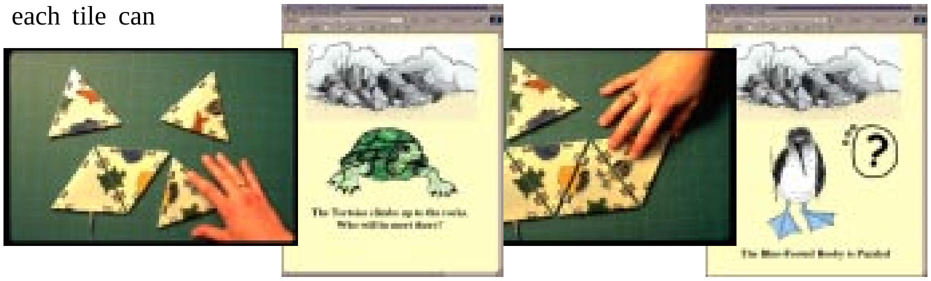
Non-Linear Storytelling

The first applications that were developed for the Triangles system (March/April 1997) were storytelling systems.

Galapagos! – A World-Wide Web Story

In *Galapagos!*, partial illustrations of characters, places and events are placed on the faces of the Triangles in such a way that one or more users connecting two edges together can complete these images. As the two halves of a character or event are connected, web pages containing the content of the story appear on the user's screen. Which triangles are chosen and the order in which they are connected to one another determine aspects of the progression of the story. The result is a non-linear narrative that is told partially by a comic book-like arrangement of physical tiles, and partially by animated images and text on a computer screen (Fig. 7).

One problem with *Galapagos!* is that its content is entirely visual, requiring the user to split their visual focus between the images and text on the computer screen and the Triangles themselves. Children who played with the application did not always know where they should be looking, and expected audio feedback. This issue was addressed in the next storytelling application that we created, *Cinderella 2000*.



The Tortoise climbs up to the rocks. Who will he meet there?

The Blue-Footed Booby is Puzzled

Fig. 7: Interacting with *Galapagos!*, a user connects Triangles to access web content.

Cinderella 2000 -- An Audio Comic Book

Cinderella 2000 presents a modern version of the Cinderella fairy tale. Interactively arranging seven triangles that depict various aspects of the story, user can trigger audio samples stored on a desktop computer, creating a soundtrack of sound effects, narration and dialogue in the voices of the characters. These sounds are synchronized with the progression of the story, because they are triggered by specific connection events and Triangle configurations. Using audio for the output avoided the split-focus experienced with *Galapagos!*, creating a more compelling and complete storytelling experience. One interaction scenario for Cinderella 2000 is described at the beginning of this paper.

The images for *Cinderella 2000* were more varied in their arrangement and design than those in *Galapagos!* The design of the visuals was greatly influenced by the techniques and visual language of comics [9], making use of framing, scale, implied action and composition to create a narrative progression through still frames (Fig 8).

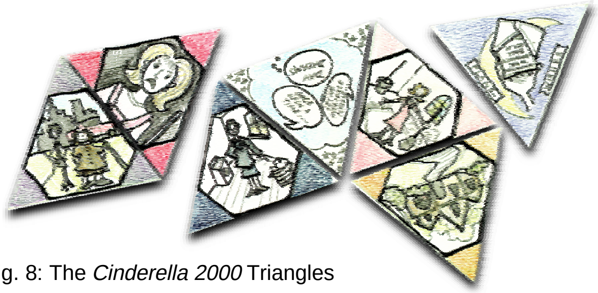


Fig. 8: The *Cinderella 2000* Triangles

Also, two specific Triangles were created as interaction devices:

- *Event Triangles*, symbolizing specific events in the story, for example the arrival of the invitation to the ball. Attaching an event triangle changes the context of the story, and thus the behavior of the characters.
- An *Info Triangle*, depicting three comic-book 'voice bubbles'. Attaching a specific edge of this triangle to a character would cause that character to reveal certain information about themselves.

Lessons Learned from Storytelling Applications

With *Galapagos!* and *Cinderella 2000*, we showed the potential of the Triangles system as a general interface for non-linear storytelling. In creating these applications, we also explored techniques and developed general authoring tools that would be useful for others creating nonlinear Triangles content.

However, one thing that became clear from implementing these two applications was the difficulty inherent in authoring unique content for the astonishing number of configurations possible with the Triangles system. The *Galapagos!* and *Cinderella 2000* applications each used seven triangles, offering literally millions of possible unique configurations. It was clear that limiting the number of 'appropriate' connections was necessary, in order to avoid having to create a huge number of unique content events. This was addressed in *Galapagos!* by using the partial illustrations to suggest which connections would be appropriate. Still, interaction issues arose around what would happen if 'incorrect' connections were made. For example, connecting half of a turtle to half of a bird might seem reasonable in a fantasy story about mythical animals.

If the application could respond appropriately to any of hundreds of thousands of possible connections, some extremely compelling and interesting storytelling applications might be possible. Advanced artificial intelligence and emergent behavior research [8] suggests that such applications could actually be written, generating or modifying content on the fly and thus making full use of the Triangles system's potential. In the future, we hope to collaborate with experts in this field to further investigate this possibility.

Another critical lesson learned from the storytelling applications was the importance of providing a single focus for the user's attention. The use of audio feedback was much more effective with children than pure visual content, as discussed above.

Triangles provide a very simple means for interacting with a potentially very complicated set of character relationships and storytelling situations. This ease of manipulation can also be applied to other sorts of information relationships. The next applications we developed use Triangles as an interface for configuring complicated media systems.

Media Configuration

TriMediaManager

Triangles' potential as a control system for information was further explored in *TriMediaManager*, an application in which the Triangles system is used to select and configure various media during a broadcast lecture or presentation. Triangles are given markings representing 'content' – audio and video clips, 3D datasets, images and other documents to be used during the session. The triangle that is directly attached to the computer is labeled as the *display*. During the session, the presenter can interactively decide which media is being shown by physically rearranging the positions of the triangles in relation to each other and to the *display* Triangle.

For example, if a presenter has access to a live video feed of herself, an audio clip, two video clips and a variety of images (presentation slides, for example), these can each be selected at any time by simply attaching the appropriate tile to the *display* triangle. If the presenter wishes to present several of these media in parallel, this can be achieved by joining the content triangles with one another. When this occurs, *TriMediaManager* attempts to simultaneously display as much of the total content as possible, giving precedence according to the proximity of each tile to the *display* triangle. In our example, if the tile representing the live feed of the presenter were directly connected to the *display* triangle, it would be broadcast as a full-screen image with audio. If an *image* tile were then connected to another edge of the *live video* triangle, it would appear as a smaller image, inset into the video feed (Fig. 9). Connecting a *video clip* triangle on the remaining edge of the *live video* tile would inset the appropriate video clip in another corner of the screen (the audio streams from the live feed and the clip would be mixed).

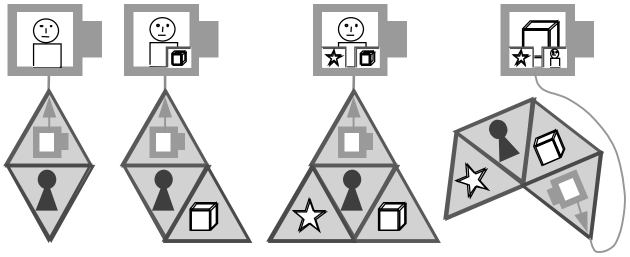


Fig. 9: *TriMediaManager*

At any point in the presentation, the presenter could easily change the display priorities of the various media: Moving the *display* triangle so that it was connected directly to the *video clip* tile would cause the hierarchy to shift, and the output to respond accordingly: the clip would take over the full screen, with the image and the live video feed each inset as smaller windows.

The presenter might attach still more triangles to the configuration, which might or might not be immediately displayed, depending on available output resources and proximity to the *display* triangle. The *TriMediaManager* application keeps track of what display resources are available (inset windows, audio channels, volume control, etc) and how each type of information is being displayed (i.e. moving video, still image, audio, or rendered object) and allocates resources for as much of the data as possible.

The human ability to intuitively manipulate, sort and arrange physical objects is exploited in *TriMediaManager*, creating a physical interface for accessing and arranging the display of a great deal of presentation information in realtime. Using the Triangles to represent high-level content selection like ‘vacation clip’ or ‘earnings slide’ as opposed to using traditional patching controls like ‘VCR1→MON2’ or ‘carousel advance’ allows the presenter to focus on content and dynamically rearrange their presentation if the need arises.

Artistic Expression

In creating *Galapagos!*, *Cinderella 2000*, and *TriMediaManager*, we explored many of the benefits of the Triangles system, including the exploratory nature of rearranging Triangles, the combinatorial potential of Triangles configurations, and the narrative potential of spatial tile arrangements.

One drawback common to all of these applications is their use of pre-defined mappings of information to Triangles. This requires extensive content authoring before each application can be used. *The Digital Veil*, the next Triangles project that was undertaken, allows users to control not only the output generated by specific Triangles interactions, but to assign and reassign meaning to groupings of Triangles during the course of an interaction. *The Digital Veil* was created as an art installation for the 1997 *Ars Electronica* festival in Linz, Austria.[4]

The piece consists of a table on which are laid out 35 Triangles. Each tile has a photograph, illustration, graphic symbol or physical texture applied to its surface(Fig. 10). These elements were designed to be beautiful, evocative and meaningful, both individually and in combination with each other. The public is invited to interact with the tiles in two locations on the table (Fig. 11), creating arrangements of Triangles and connecting them to the ‘input station’ or the ‘output station’ – two small boxes on the table with exposed triangle-edge connectors. The ‘input station’ has a light-up button and a microphone on it. When a user connects up to four triangles together and attaches this arrangement to the input station, the button lights up, and the user can push it and speak into the microphone. Their voice is sampled and linked with the specific arrangement of Triangles that they created. In this way, participants can ‘assign meaning’ to their configurations, creating illustrated phrases and small narratives that hold personal meaning.

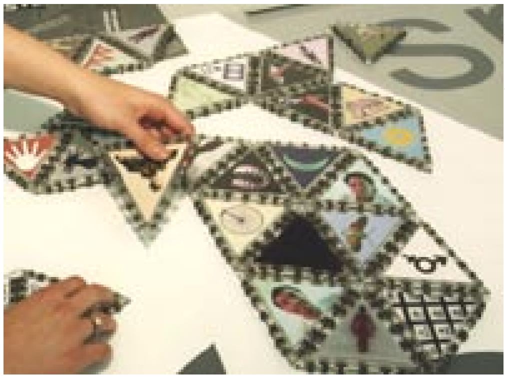


Photo: rubra

Fig. 10: Images used in *The Digital Veil*, designed to be beautiful and evocative.

At the ‘output station,’ participants can create large configurations of triangles, building a visual and tactile texture on the table in front of them. As they do so, each individual triangle that they add triggers its own evocative audio sample, building an aural texture to accompany the configuration that they create. In creating the large configuration, if the user arranges any of the tiles to form one of the ‘phrases’ that had been recorded by a previous participant, that audio recording is also played back. In this way, the piece grows and changes over the course of its presentation, keeping a memory of the meanings and associations that users have created.

Fig. 11: Members of the Public interacting with *The Digital Veil.*

The Digital Veil was created to address the reconfigurability of meaning that is inherent in digital information systems. Its use of Triangles allows participants to explore this in an interactive and creative way without needing to learn the particulars of a new user interface. The Triangles provide the feeling that users are actually holding and rearranging the information itself.

Audience members greatly enjoyed being able to quickly make new groupings of Triangles that they could personalize with their voice. They found their own uses for the application, sometimes leaving ‘secret messages’ in the system to be retrieved later, or even singing ‘rounds’ that could be controlled by adding new Triangles at the right time.

DESIGN AND DEVELOPMENT OF THE SYSTEM

The development of the system brought together skills from a variety of fields to give the Triangles functionality in the physical and digital worlds. Software architecture, electrical engineering, mechanical and industrial design converged to make the Triangles balanced physical/digital objects. One example of this balance is the physical form of the tiles, which suggest association and rearrangement (key elements of data manipulation). Another is the use of the physical properties of magnets to ensure good mechanical and digital connections while reducing ambiguity and providing tactile feedback to the user.

The design process was an iterative undertaking by two researchers, each with extensive prior interaction-design experience. One was well-versed in the physical and mechanical design of interactive objects, and the other had experience in software systems and traditional (GUI) user interface design. Close collaboration and a good understanding of each other's field of expertise enabled us to find innovative solutions to design challenges at many levels. A tight design loop was established, with new hardware, software and conceptual design maturing simultaneously. Following is a discussion of key conceptual, mechanical and software design issues faced in the development of the Triangles system.

Entropy and Physical Delimiters

When designing physical interfaces, it is important to consider the number of degrees of freedom and possible scenarios that can occur. This is sometimes called the *entropy* of the system. It is impossible to anticipate all of the potential interactions that people can have with real-world objects; for example, a user might pick up the interface and wave it around in the air, juggle it, or otherwise use it in unpredictable ways. Some interface systems, such as speech recognition systems or vision systems, continuously track of all of the user's interactions. These systems must determine *action delimiters* to differentiate between continuous action events in order for those actions to be correctly interpreted. Although powerful in certain situations, this kind of continuous physical sensing can be complicated and computationally expensive [10].

In the design of the Triangles, we sought to address the issues of entropy and action delimiters through the physical design of the objects themselves. Limiting the types of interaction that are suggested by the form of the object reduces the entropy of the system. The equilateral triangle shapes suggest tiling and edge-to-edge connections as appropriate ways of manipulating the objects, and the magnetic edge connectors reinforce this with tactile feedback when appropriate connections are made. Action delimiters are provided by the physical connection mechanism itself, which reports only *significant* changes to the system (connections and disconnections), as discrete events. These physical design decisions helped create a low-bandwidth and computationally inexpensive means for accurately interpreting a user's interaction with the Triangles system.

Physical Connectors

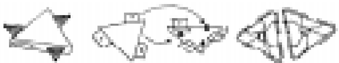


Fig. 12: Connector design sketches

The physical connectors had to be designed so as to make it easy to link any Triangle edge with any other edge in a robust, hinging configuration. Disconnecting the Triangles also had to be as simple as pulling them apart from one another. In addition, the connectors needed to provide a consistent supply of power and uninterrupted data connections between the microprocessors on the tiles.

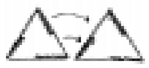


Fig. 13

Several designs for the physical connectors were examined, including slotted edges, snaps, zippers and conductive Velcro® fasteners (Fig. 12). Since equilateral triangles are radially symmetrical, connections such as magnets, snaps or Velcro, which have 'male' and 'female' components (or polarity, in the case of magnets) needed to be arranged so that 'male' would always meet 'female' and vice-versa (Fig. 13). Depending on the design of the electronic circuit, issues of symmetry arose in ensuring that each transmit pin would meet a receive pin, and that shared pins, such as power and ground, would always find the correct mate when connected.

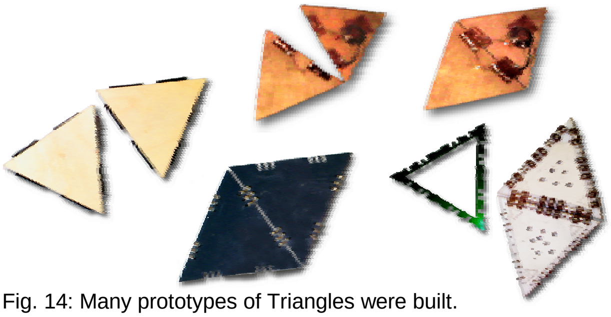


Fig. 14: Many prototypes of Triangles were built.

Many prototypes were built (Fig. 14), and eventually it was decided that the feel of the magnetic connectors was important to reinforcing the sensation of connection and disconnection. They make the triangles easy to attach and remove, and because they require no lateral motion, as do zippers or slots, connecting complex forms such as hexagons is easy. In addition, clever arrangement of the polarity of the magnets on the individual connectors can help force the edges into proper alignment with one another when making connections, and enables them to physically resist being wrongly connected. This avoids false connections and further reduces the entropy of the system.

Microprocessor Code

One of the Triangles in a given set acts as the *mother triangle*. It differs from the others in that it has a cable that provides power to itself and to the other triangles, as well as serial communication to a *host* computer. As soon as any Triangle is connected to this *mother triangle*, the new Triangle's microprocessor receives power through its edge connectors, and it can communicate with its neighbors and the host computer. The new Triangle then provides power and a communication pathway to subsequent Triangles as they are connected to it, and so on.

The microprocessor inside each Triangle is responsible for managing that Triangle's identity and information about its connections. When a Triangle receives power, its microprocessor begins alternately polling and transmitting its ID to each of its edges. If any other Triangles are connected to it, they will be able to identify it and also the edge to which they are connected. Whenever such an event occurs, each Triangle involved generates a message containing the new configuration information, which it passes to the host computer. Disconnections between Triangles result in a similar message being passed by the Triangle that is still connected to the system.

The host computer receives these messages, and the Triangles system software layer interprets them. It reconciles connection messages that were generated simultaneously by two Triangles into *connection events* and keeps track of *disconnection events*. The system keeps a time-stamped history of all such events, and provides functionality, through an Application Programming Interface (API), for independent software applications to access and react to specific interactions with the Triangles.

Communication Structure

In order for the host computer (to which the Triangles are attached) to know the exact configuration of the system, a distributed network routing system was implemented. Each Triangle is responsible for determining when local events occur (connections and disconnections) and relaying information about these events back to the host computer, as discrete messages. The host computer then consolidates and reconciles these messages with one another in order to determine the configuration of the entire system.

To help develop and test the host software and various message-passing schemes in parallel with the development of the hardware, computer simulations (Fig. 15) of several routing algorithms were built, including:

- Probabilistic ‘broadcasting’ and regeneration of messages to every triangle with the expectation that the host would eventually receive the message;
- ‘Maze-walking’ which attempts to find the host connection by exploring all the possible connections;
- ‘Topography maps’ stored within each microprocessor to route the message at each step of its journey;
- ‘Perimeter hugging’ which attempts to find the edge of the configuration, and from there to trace a path back to the host;
- ‘Gradient descent’ which establishes gradients between each triangle and the host, and pass messages ‘downhill’ to ensure that they reach their goal.

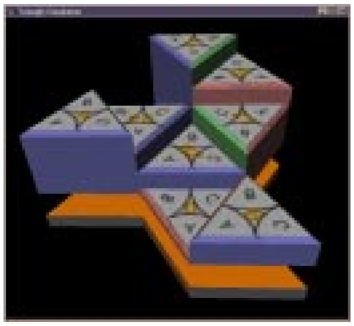


Fig. 15: Software Simulation of ‘Gradient Descent’

Each of these algorithms had flaws, especially when implemented in a distributed system of low-memory microprocessors. For the initial implementation of the Triangles system, the ‘gradient descent’ algorithm [5] was used. This ensured that connection and disconnection messages would always be reported, but was unforgiving of errors caused by noisy connections, and introduced an undesirable amount of latency into the system.

Eventually, message-passing was simplified by making a hardware modification to the Triangles: A common serial bus was added to the architecture of the system, so that in addition to communicating with one another locally, any Triangle could pass messages directly to the host computer or to any other Triangle. This greatly improved the speed of the system and simplified the software design.

This illustrates a recurring theme in the design and development of the Triangles system. Almost every design challenge that was encountered could be solved in microprocessor software, electronics, mechanical hardware, or host software. Understanding the relationships between these elements and the trade-offs and balances between them was crucial to developing this system.

The Triangles API

A comprehensive API layer was developed for creating applications which use the Triangles system. The API layer consists of a C++ library of function calls and data structures that manage information about a user’s interactions with the system.

The simplest way for developers to use the API is to specify certain Triangle *events* which the system should watch for, and provide functions in their application that should be called when these events occur. An *event* is either a connection or a disconnection between two Triangles. The event structure contains the following information:

- *Event type:* was this a connection or a disconnection?
- *Event time*: at what time did the event occur?
- *Triangle IDs:* what were the IDs of each triangle?
- *Triangle edges:* which edge of each triangle was involved in the event?

The event registration functionality is flexible, so that an application may specify some, all or none of these fields to monitor, allowing very general or very specific interactions to be specified. For example, an application might specify an action to be triggered when Triangle 4, side 2 is connected to Triangle 12, side 0, or simply any time Triangle 3 is connected or disconnected from anything else. (If none of the fields are specified, the application will be notified whenever *any* event occurs.) When a registered event does occur, the specified application function is called, and the details of the event that occurred are passed as parameters to the function. This enables the application to know exactly how the user interacted with the system.

The API also provides more comprehensive functionality. Using a simple command structure and a set of linked lists, an application can retrieve a complete description of which Triangles are present in the system and their connections. Once this list is provided, the application can query the API for information about any individual Triangles, including a complete history of that Triangles’ interactions with its neighbors. In this way, an application can act not only on events that are occurring in real-time, but also on past changes in the system. For example, if a Triangle is connected to another Triangle for the first time, it might trigger a different event than if the same Triangle is connected again later. If two Triangles are connected and another, specific Triangle is present in the system, this might have a different meaning than if the two Triangles were connected alone.

Using the API, developers need not be concerned with the message-passing protocols or internal functionality of the Triangles themselves. This creates a useful abstraction that allows Triangles applications to be developed by competent programmers from a variety of backgrounds. One of the aims of the Triangles project is to allow researchers in other fields such as computer science, design, education or physics to experiment with this tangible interface system.

FUTURE WORK

The Triangles system was conceived of as a foundation for exploring tangible interface in a broad variety of fields. Future work on this project will focus on providing the complete Triangles system to other researchers who are interested in developing interactive applications using the tangible interfaces.

Research areas that we have targeted at present include:

- Nonlinear narrative

  Further development of the storytelling potential of ‘comic-book’ Triangles with audio feedback.

- Groupwork Coordination

  Exploration of scheduling, group dynamics modeling and workflow systems using Triangles.

- Emergent behavior/AI models for content generation

  Using semantic webs and intelligent agent technology to interpret the meanings of various Triangles associations.

Continued refinement of the API is planned, along with the development of GUI and tangible authoring tools to support Triangles application development. Software ports of the API from its current Windows® platform to Silicon Graphics™, MacOS™, and general UNIX platforms will enable a broader variety of applications to be accessed with the Triangles.

CONCLUSIONS

We have presented a new tangible interface for the manipulation and exploration of digital information topography. Bridging the gap between the physical and digital worlds is not trivial. Rather than attempting to give computers a complete understanding of human language (voice, gesture) and the real world, our approach provides a new language which both people and computers can understand. Synchronization of what users have in their hands with the digital connections that computers can perceive occurs when the Triangle pieces are attached and detached from each other. Although this language (connect/disconnect) is quite simple, the power of expression enabled by a topography of multiple triangles is tremendous and very rich.

The concept of Triangles was born in October 1996. Many physical mockups of the Triangles and scenarios were created along with software models and simulations. This iterative development process led to the first working prototype in March 1997, which was demonstrated internally, with the sample application *Galapagos!* The benefits and limitations of Triangles became apparent through the development of several other applications and refinement of the system itself. Although we have not yet undertaken extensive usability testing with the Triangles system, our internal development has led us to be confident about this new language for tangible interface.

Acknowledgments

We would like to express our sincere thanks to the following people and organizations for their contributions to this project: Wendy Chien, Emily Cooper, James Hsiao, Chris Lopes, Betty Lou McClanahan, Gong Ke Shen, Paul Yarin, and the MIT Media Laboratory’s Things That Think consortium.

REFERENCES

1. Csikszentmihaly, M. and Rochberg-Halton, E. (1981*).* “The Meaning of Things: Domestic Symbols and the Self”, Cambridge University Press, 1981.
2. Feiner, S., MacIntyre, B. and Seligmann, D. (1993). Knowledge-based augmented reality. *Communications of the ACM*. 36, 7 (1993), 52-62.
3. Fitzmaurice, G. W., Ishii, H. and Buxton, W. (1995). Bricks: Laying the Foundations for Graspable User Interfaces. *Proceedings of Conference on Human Factors in Computing Systems (CHI ’95)*, Denver, Colorado, ACM Press, 442-449.
4. Gorbet, M, and Orth, M. (1997). Triangles and The Digital Veil, *FleshFactor: Informationsmacschine mensch (1997 Ars Electronica Festival Catalog)*, Linz, Austria, SpringerWein-NewYork, 280-283.
5. Gorbet, M. and Orth, M. (1997). Triangles: Design of a Physical/Digital Construction Kit. *Proceedings of Designing Interactive Systems: Processes, Practices, Methods, and Techniques (ACM DIS ’97)*, Amsterdam, ACM, 125-128.
6. Ishii, H. and Ullmer, B. (1997). Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. *Proceedings of Conference on Human Factors in Computing Systems (CHI '97)*, Atlanta, Georgia, ACM Press, 234-241.
7. Kurtenback, G., Fitzmaurice, G., Baudel, T., and Buxton, W. (1997). The Design of a GUI Paradigm based on Tablets, Two-hands, and Transparency. *Proceedings of the ACM CHI ’97 Conference on Human Factors in Computing Systems (CHI ’97)*, Atlanta, Georgia, ACM Press, 35-42.
8. Maes, P. and Brooks, R. (1990). Learning to Coordinate Behaviors. *Proceedings of AAAI-90: The American Conference on Artificial Intelligence*, Boston, MA, AAAI Press/MIT Press, 796-802.
9. McCloud, Scott. “Understanding Comics.” Kitchen Sink Press, 1994.
10. Pentland, A. P. (1996). Smart Rooms. *Scientific American*. 274, 4 (1996), 54-62.
11. Resnick, M. (1993). Behavior Construction Kits. *Communications of the ACM*. 36, 7 (1993), 64-71.
12. Suzuki, H., Kato, H. (1993). AlgoBlock: a Tangible Programming Language, a Tool for Collaborative Learning. *Proceedings of 4th European Logo Conference,* Aug. 1993, Athens Greece, 297-303.
13. Wellner, P. (1993). Interacting with Paper on the DigitalDesk. *Communications of the ACM*. 36, 7 (1993), 87-96.