Chapter 8.

Triangles∗ as a Networked, Computing Material

Overview of the *Triangles*

In many ways, the *Triangles,* (1997-1999)[1,2] may seem disconnected from the smart textile projects developed in this thesis. They are not soft, and do really rely on smart textiles, accept as part of their patented, flexible

∗ In collaboration with Matt Gorbet.

1 Much of the description in this chapter refers to:
Gorbet, M., Orth, M. and Ishii, H., *Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography*, Proceedings of Conference on Human Factors in Computing Systems, (CHI 1998), Los Angeles, ACM Press, (1998) pp. 49-56.

2 Gorbet, M., and Orth, M., *Triangles: Design of a Physical/Digital Construction Kit*, Proceedings of Designing Interactive Systems, (DIS '97), ACM Press, Amsterdam, (1997) pp. 125-128.

Figure 8.1 *Triangles* in relation to the Tree of Projects.

connector[3]. However, it is essential to this thesis that the *Triangles* themselves be seen as a large-scaled, networked, sculptural and active computing material. With the *Triangles* shapes and forms can be built and made, as with clay. The *Triangles* were originally created to explore the expressive possibilities of a large group of networked computational objects. Moreover, their triangular shape was always meant as a departure from the square and pictorial world of rectangular computers.

How They Work

Each *Triangle* contains a microprocessor programmed with a unique ID, and a patented hinge-like connector that allows the *Triangles* to simultaneously mechanically and electrically connect to one another. When the *Triangles* are physically connected together, they communicate their identities to one other across that physical/electrical connector, and then relay that connection information back to a master computer. This master computer keeps track of the physical configuration of the *Triangles*. Information from the *Triangles* about which side of which *Triangle* is connected to which side of another, allows the PC to infer the exact shape and relationship of a connected mass of *Triangles*. The computer can then use this information to control applications that range form HTML storybooks, to audio landscapes.

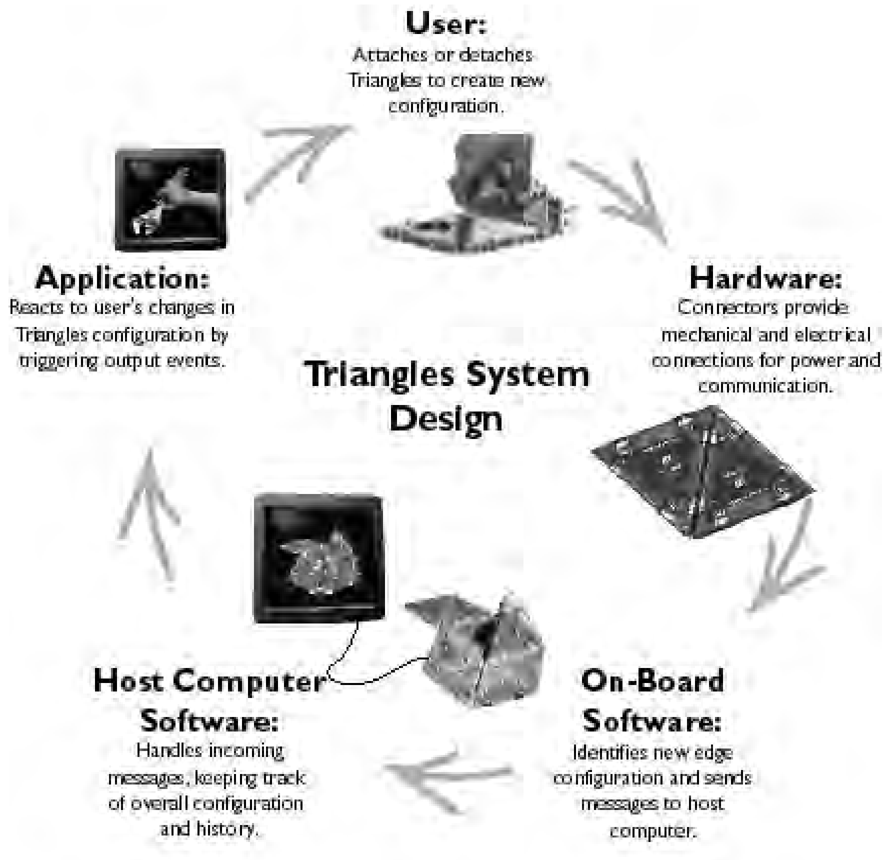


Figure 8.2 Overview of *Triangles* System, image from: Gorbet, M., Orth, M. and Ishii, H., *Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography*, Proceedings of Conference on Human Factors in Computing Systems, (CHI 1998), Los Angeles, ACM Press, (1998) pp. 49-5.

[3] Orth, Margaret, Gorbet, Matt, Digital Communication, Programmable Functioning and Data Transfer Using Modular, Hinged Processor Elements, US Patent # US5941714, (1999).

Why Three Sides?

The three-sided triangular shape of these pieces is both an aesthetic departure from square circuit boards, and a reflection of the non-linear, complex and branching structure of digital information. Three sides means that a *Triangle* can have two inputs and one output, or two outputs and one input. Three sides (vs. four) is also the least number of sides that are required to create a physical object that is non-linear. This keeps the physical complexity of the system as small as possible; two, three sided objects have nine unique combinations, and two, four sided objects have sixteen. In fact, the possible unique combinations of any number of *Triangles* is factorial. If each side of each *Triangle* has unique ID, the possible number of unique combinations is described by $x \geq 3^n(n+1)!/6$, where n= # of triangles and x = # of configurations. If the *Triangles* were four sided, this would both increase the complexity of the system, and also require more complex images, (possibly four, with one referring to each side), on the surface of each *Triangle.*

As A Macroscopic Material

As a group of networked *prefabricated* individuals, the *Triangles* are a networked computing material that allows people to create shapes and objects, which can influence events in software. While each *Triangle* cannot itself be broken apart, a group of twelve *Triangles* can be broken into two smaller groups of six, each group with properties similar to the larger group. Because the connectors are flexible and immediate, the *Triangles* can quickly create both two and three-dimensional objects whose exact shape is known to the

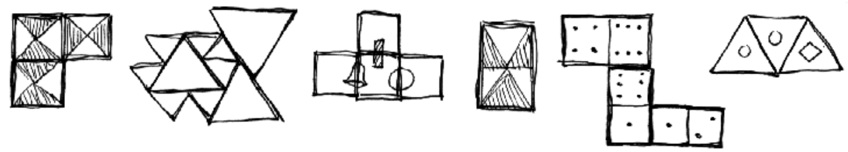


Figure 8.3 Early sketches for alternative shapes of pieces.

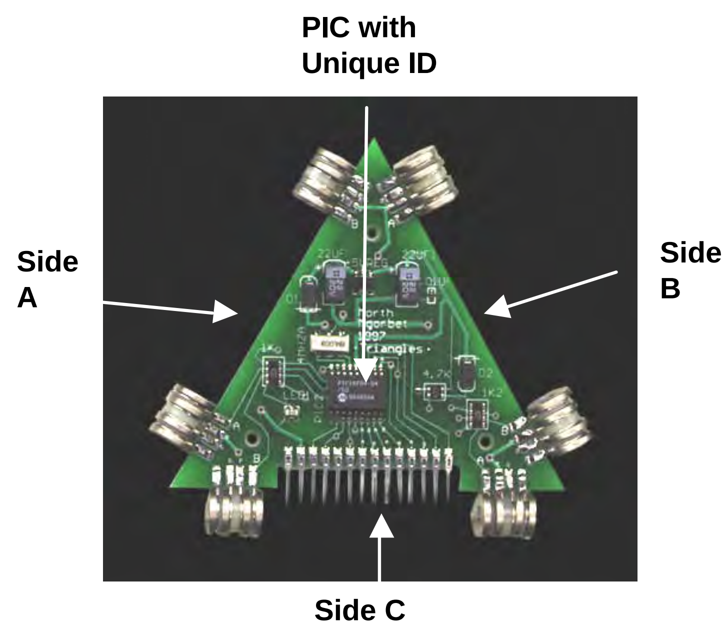


Figure 8.4 *Triangle* circuit board, hardware version 1, with early, sliding magnetic connectors.

computer. In this way, the *Triangles* suggest a sort of macroscopic, computational clay. If they were to shrink to microscopic scale, they would become a truly claylike input device that could play music or paint pictures. If they were to have display elements built into them they could suggest a way to shape a personalized visual display. In summary, what makes the *Triangles* truly material-like is both the repetitive nature of the individual elements, (i.e. they are all identical *Triangles*), and the flexible and immediate connections between them.

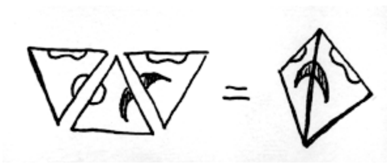


Figure 8.5 Three *Triangles* acting as a material, shaped and/or connected to form a pyramid

Integrated Connector Design

The unique connectors of the *Triangles* demonstrate how artists can use new materials to create an appropriate and integrated form for what is normally a bulky, prefabricated precursor element, (like an electrical connector), that is usually hodgepodged into an object. The final version of these connectors uses a combination of magnets and conductive female Velcro to create an instantaneous and flexible mechanical and electrical connection between two *Triangles*. This new connector allows power, ground, and both local and serial data to be transmitted between the *Triangles*, while guaranteeing a quick and immediate connection. Imagine if the *Triangles* had connected together with a standard serial or DB9 connector. Not only would the *Triangles* have been ugly; the process of physically interacting with them would have been awkward and time consuming, and the effect of the physical object as a means for immediately controlling information far less convincing.

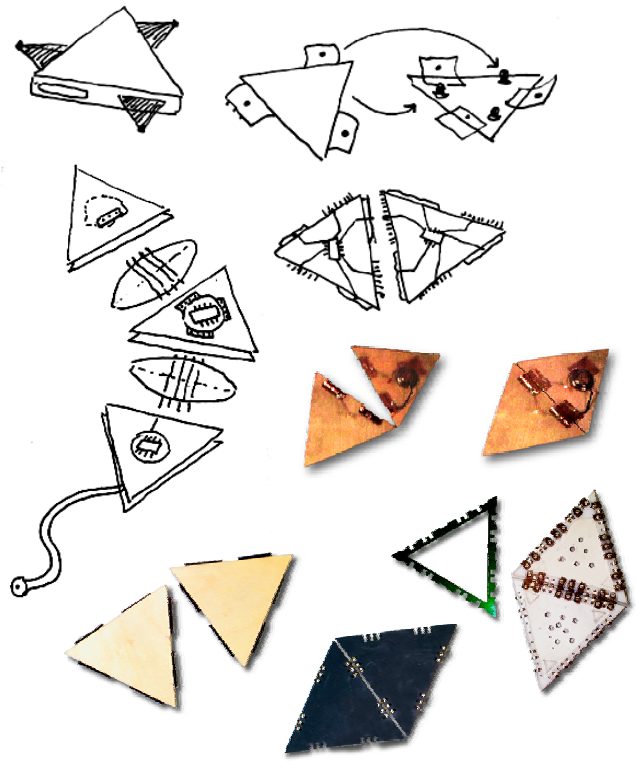


Figure 8.6 Early sketches and prototypes for various integrated connector designs.

First Generation *Triangles*

The first generation of *Triangles* used sliding magnets to create a mechanically stable multi-pin connection. Because magnets are fired, they have low physical tolerances. Consequently, the sliding mechanism was not as consistent as required, and the communication and electrical connection between *Triangles* would sometimes fail.

This generation of *Triangles* also used a unique *Triangle*-to-*Triangle* routing scheme to pass messages back to the host computer. "The message-passing algorithm was based on a 'gradient-descent' algorithm, through which messages were passed along a gradient, established between the host computer and each of the triangles. A good analogy for this is that of altitude: if one thinks of the connection to the host computer as being at the lowest altitude, then our gradient is established by slightly augmenting the 'height' of each new tile as it is added. This ability of the *Triangles* network to 'self-organize'[4] ensures a sloping, direct path from every tile back to the host connection. When any *Triangle* generates or receives a message, it simply seeks its *next lowest* neighbor, and passes the message along in the 'downward-sloping' direction. This system guarantees that each message will eventually reach the host computer, and requires each tile to store very little topographical information. It also avoids redundant and unnecessary passing of messages."

[4] Smith, J.R., *Distributing Identity*, IEEE Robotics and Automation Magazine, Vol.6, No.1, (March, 1999).

Figure 8.7 *Triangles*, first generation hardware, detail of connectors with sliding magnets, and image of *Triangles* connected.

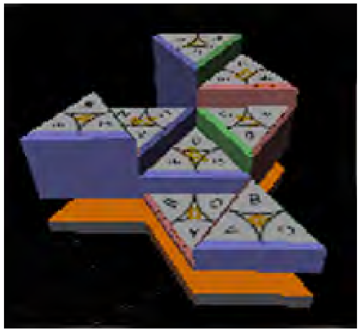


Figure 8.8 Illustration of gradient algorithm used for communication in the first generation *Triangles*.

“This method for acquiring and relaying topographical information minimizes demands on each microprocessor in terms of memory and functionality. Each PIC chip only has to store five items of information: its unique ID, the most recent IDs of its three neighbors, and its ‘height’ value. Functionally, each chip needs to be able to perform the following functions: set its height value, poll its neighbors (detecting changes in IDs and heights), and generate and pass messages. The circuitry required for this system is also fairly simple in that each *Triangle* needs only a direct local connection to each of its neighbors, plus power and ground”[5]

Second Generation *Triangles*

The second generation of *Triangles* overcame the unreliability of the first generation by using a new, and more mechanically stable connector. A combination of conducting female Velcro and magnets created a reliable and immediate multi-pin connector on each side of the *Triangles*. In this connector, The Velcro was used to overcome the low physical tolerances of the magnets, by making each pin “hairy” and guaranteeing that it would electrically connect to its mate on the other side. The polarity of the magnets was used to ensure that the connectors lined up correctly every time two *Triangles* were connected. .

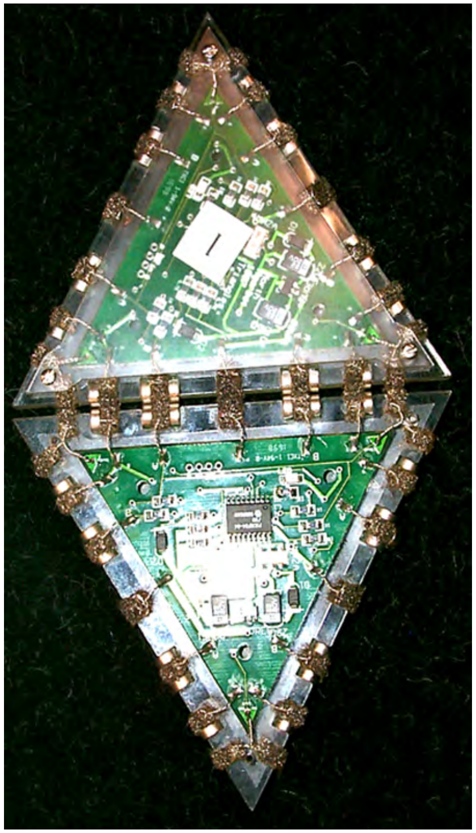


Figure 8.9 Second generation with magnetic and conductive female VELCRO connectors.

[5] Gorbet, M., and Orth, M., *Triangles: Design of a Physical/Digital Construction Kit*, Proceedings of Designing Interactive Systems, (DIS '97), ACM Press, Amsterdam, (1997) pp. 125-128.

The second generation of *Triangles* registered connections much faster by using a speedy serial bus to pass messages back to the host computer. In the first generation, each *Triangle* acted like a router, passing the message along and looking for new connections simultaneously. This required a lot of time. Adding an isolated serial bus, meant that the *Triangles* could communicate much quicker with the host computer. It also required an additional pin. Each *Triangle* now needed a pin for power, ground, *Triangle*-to-*Triangle* communication, (to determine who on what side was connected to whom), and serial communication. Because equilateral triangles are radially symmetrical, the connectors also needed to be arranged so that 'male' would always meet 'female' and vice-versa. To guarantee that each transmit pin meets a receive pin, and that shared pins, such as power and ground, would always find the correct mate when two *Triangles* are connected, there had to be redundant pins.[6] Practically, this meant that each side of a Second Generation *Triangle*, needed at least seven pins.

Triangles Applications

As a physical interface that relies on a desktop computer for output, the *Triangles* are an excellent example of the positives and negatives of creating controllers. Because the *Triangles* are physically free

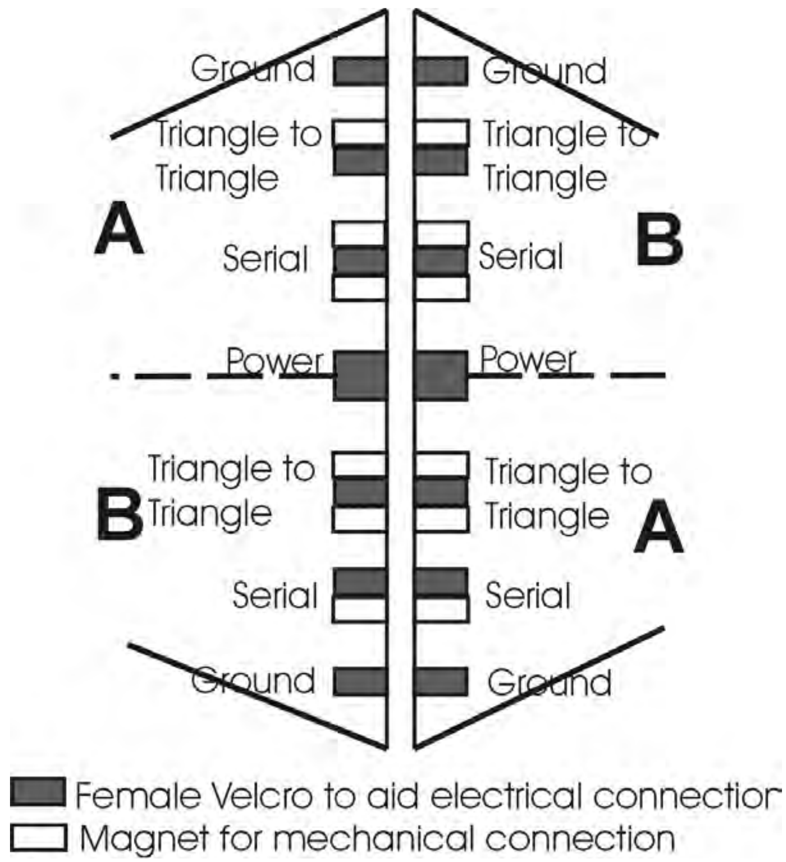


Figure 8.10 Diagram of second generation connectors. Symmetry means that connector half *A* must meet half *B*, so redundancy of pins is required. Seven total pins are used.

[6]Gorbet, M., Orth, M. and Ishii, H., *Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography*, Proceedings of Conference on Human Factors in Computing Systems, (CHI 1998), Los Angeles, ACM Press, (1998) pp. 49-56.

from any visual or audio display device, they are light and have little power requirements. Relying on a PC to create content also allows for experimentation with applications and their relationship to the physical form to the *Triangles*. But being physically separated from their output media also presents many drawbacks when designing applications for the *Triangles*. In general, these drawbacks revolve around the split between the physical objects themselves and their output media, especially screen-based images.

Triangles software includes both a C++ API* and content applications that were written on top of this API. The API is documented and available on the *Triangles* CD. The higher-level content applications were written in Visual Basic, HTML and C++. The applications I created for the *Triangles* include *Galapagos** (a simple non-linear, web-based, children's story, *Cinderella 2000*, (a feminist retelling of Cinderella), *Toy Search*, and the *Digital Veil* for Ars Electronica*.

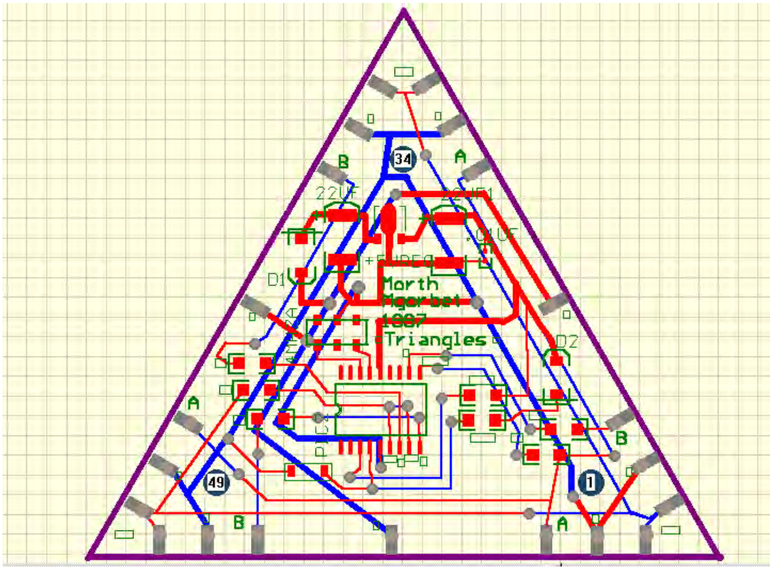


Figure 8.11 Circuit diagram of generation 1.

* Matt Gorbet.
* In collaboration with Matt Gorbet.
* In collaboration with Matt Gorbet.

Galapagos: An Interactive Web-based Narrative*

Galapagos (1997) was an interactive web-based, storytelling program that let players connect two halves of characters (like the turtles and blue-footed boobies), or places (like beaches and rocks), to guide a story. When two halves of a character or place were connected, web pages containing the appropriate animated content of the story appeared on the computer screen. The result was a non-linear narrative told partially by a puzzle-like arrangement of physical tiles, and partially by animated images and text on a computer screen. The design of the characters on the *Triangles* reflected this simple branching story. The first *Triangle* was always a turtle half. Once the other half was connected, the player then had two choices; go to the beach or the sea. After choosing either the beach or the sea, the player then had another set of choices: meet a booby or another turtle. If the player met another turtle, they would eventually have eggs and babies. In this application, the design of the images on the *Triangles* made only certain connections valid (for instance egg to egg is valid, but not egg to turtle), limiting the possible valid connections and therefore the content that had to be created. In other, words, nothing happened if you connected an egg to a turtle half, so we did no need to create a story page for it.

A major problem with *Galapagos* was the separation of its media or dynamic output, which was in the form of the animated images on the computer screen, from the

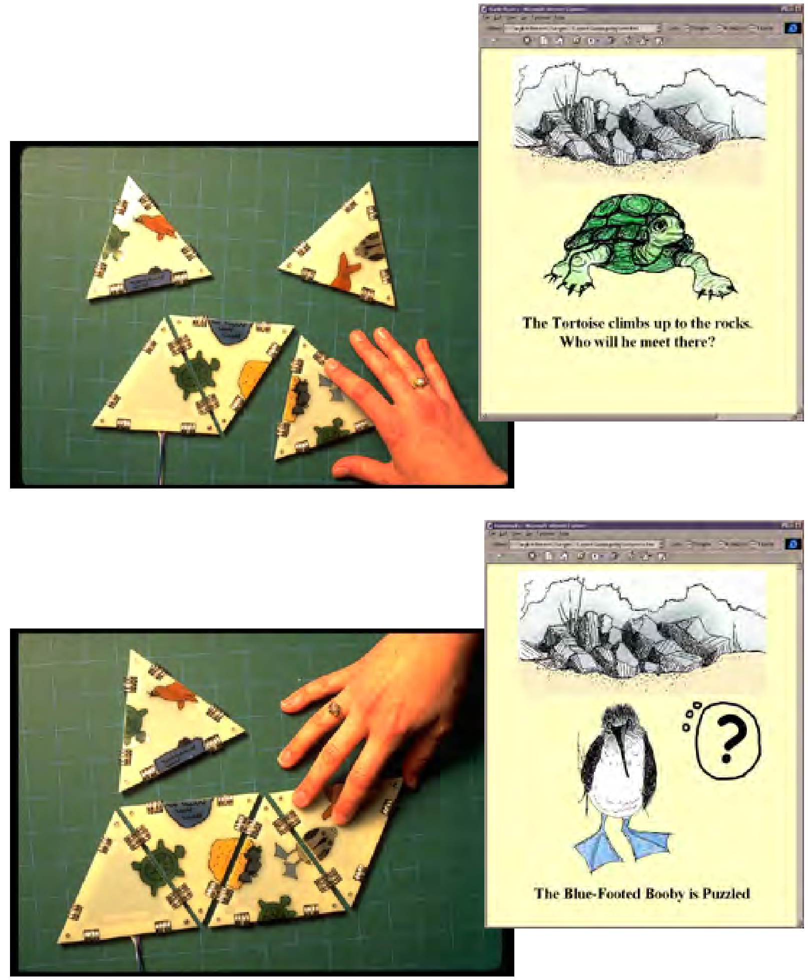
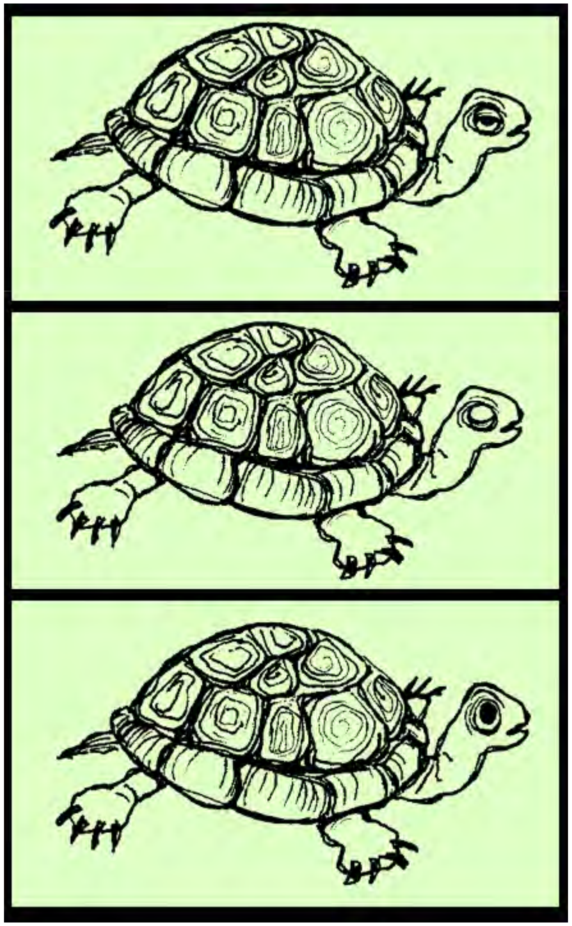


Figure 8.12 Connections of *Triangles* and the websites they called in *Galapagos*. This sequence starts by connecting the turtle, and then the beach, taking the turtle to the beach. Finally, the player chose to meet a bobby rather than another turtle, by connecting the two-halves of the booby. Drawings by Maggie Orth.

* In collaboration with Matt Gorbet.

physical *Triangles* themselves. Because the storytelling output was entirely visual, the player was *required* to split his or her visual focus between looking up at the images and text on the computer screen, or down at the *Triangles* on the table. Children who played with the application were so busy looking at the *Triangles* in their hands, that they often did not look up to see what happened on the screen when the *Triangles* were connected. This issue was addressed in the next storytelling application that was created, the purely audio comic book, *Cinderella 2000*.

Figure 8.13 *Galapagos* characters and places. Connecting two halves of a character or place would call a new website in the story. (Note: these images are not in the order of the final application.)



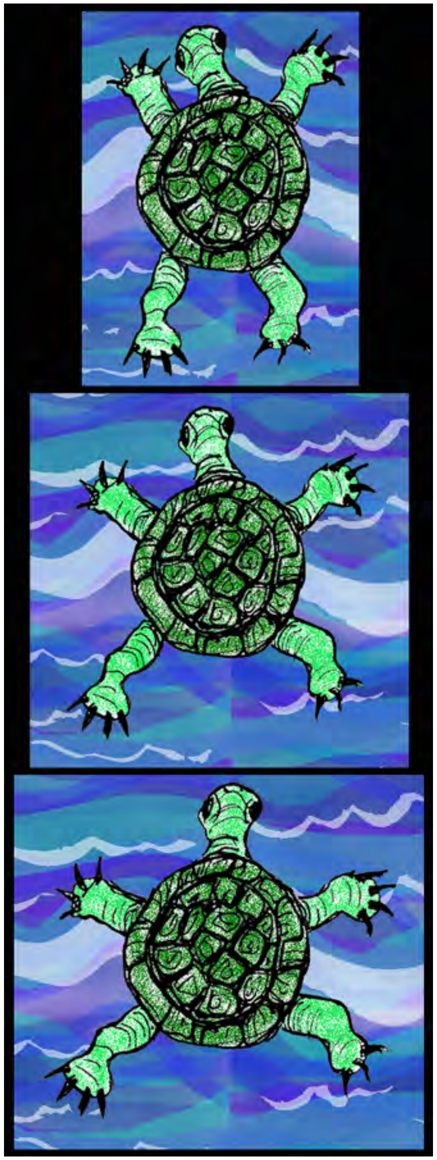


Figure 8.14 Two typical animation sequences from *Galapagos.* Drawings and animations by Maggie Orth.

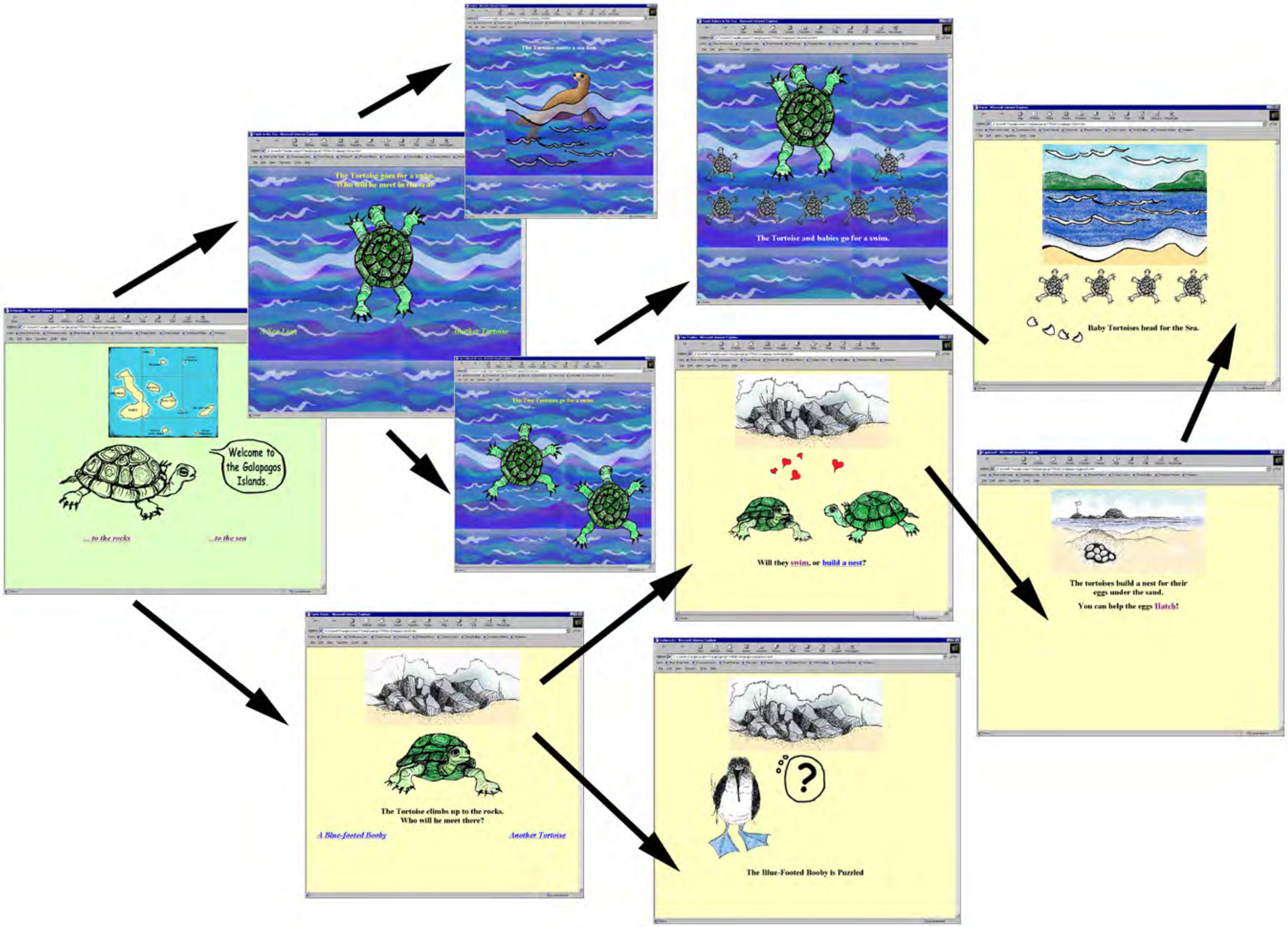


Figure 8.15 Animated websites and narrative structure from *Galapagos*. Drawings, animations and story by Maggie Orth.