# Bricks: Laying the Foundations for Grasp-able User Interfaces

*George W. Fitzmaurice (1), Hiroshi Ishii (2)* and *William Buxton (1, 3)*

(1) Dynamic Graphics Project
CSRI, University of Toronto
Toronto, Ontario,
Canada M5S 1A4
Tel: +1 (416) 978-6619
E-mail: gf@dgp.toronto.edu
E-mail: buxton@dgp.toronto.edu

(2) NTT Human Interface Lab
1-2356 Take,
Yokosuka-Shi, Kanagawa,
238-03 JAPAN
Tel. 81-468-59-3522
E-mail: ishii.chi@xerox.com

(3) Alias Research Inc.
110 Richmond Street East
Toronto, Ontario,
Canada M5C 1P1
Tel. +1 (416) 362-9181

## Abstract

We introduce the concept of Graspable User Interfaces that allow direct control of electronic or virtual objects through physical handles for control. These physical artifacts, which we call "bricks," are essentially new input devices that can be tightly coupled or "attached" to virtual objects for manipulation or for expressing action (e.g., to set parameters or for initiating processes). Our bricks operate on top of a large horizontal display surface known as the "ActiveDesk." We present four stages in the development of Graspable UIs: (1) a series of exploratory studies on hand gestures and grasping; (2) interaction simulations using mock-ups and rapid prototyping tools; (3) a working prototype and sample application called GraspDraw; and (4) the initial integrating of the Graspable UI concepts into a commercial application. Finally, we conclude by presenting a design space for Bricks which lay the foundation for further exploring and developing Graspable User Interfaces.

## Keywords:

input devices, graphical user interfaces, graspable user interfaces, haptic input, two-handed interaction, prototyping, computer augmented environments, ubiquitous computing

## Introduction

We propose a new paradigm, Graspable User Interfaces, which argues for having some of the virtual user interface elements take on physical forms. Traditional graphical user interfaces (GUIs) define a set of graphical interface elements (e.g., windows, icons, menus) that reside in a purely electronic or virtual form. Generic haptic input devices (e.g., mouse and keyboard) are primarily used to manipulate these virtual interface elements.

The Graspable UIs allow direct control of electronic or virtual objects through physical artifacts which act as handles for control (see Figure 1). These physical artifacts are essentially new input devices which can be tightly coupled or "attached" to virtual objects for manipulation or for expressing action (e.g., to set parameters or to initiate a process). In essence, Graspable UIs are a blend of virtual and physical artifacts, each offering affordances in their respective instantiation. In many cases, we wish to offer a seamless blend between the physical and virtual worlds.

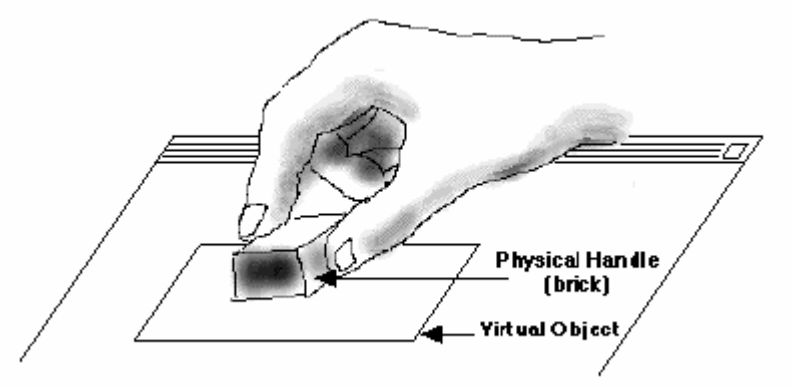


**FIGURE 1. A graspable object.**

The basic premise is that the affordances of the physical handles are inherently richer than what virtual handles afford through conventional direct manipulation techniques. These physical affordances, which we will discuss in more detail later, include facilitating two handed interactions, spatial caching, and parallel position and orientation control.

The Graspable UI design offers a concurrence between space-multiplexed input and output. Input devices can be classified as being *space-multiplexed* or *time-multiplexed*. With space-multiplexed input, each function to be controlled has a dedicated transducer, each occupying its own space. For example, an automobile has a brake, clutch, throttle, steering wheel, and gear shift which are distinct, dedicated transducers controlling a single specific task. In contrast, time-multiplexing input uses one device to control different functions at different points in time. For instance, the mouse uses time multiplexing as it controls functions as diverse as menu selection, navigation using the scroll widgets, pointing, and activating "buttons." Traditional GUIs have an inherent dissonance in that the display output is often space-multiplexed (icons or control widgets occupy their own space and must be made visible to use) while the input is time-multiplexed (i.e., most of our actions are channeled through a single device, a mouse, over time). Only one task, therefore, can be performed at a time, as they all use the same transducer. The resulting interaction techniques are often sequential in nature and mutually exclusive. Graspable UIs attempt to overcome this.

In general, the Graspable UI design philosophy has several advantages:

- It encourages two handed interactions [3, 7];
- shifts to more specialized, context sensitive input devices;
- allows for more parallel input specification by the user, thereby improving the expressiveness or the communication capacity with the computer;
- leverages off of our well developed, everyday skills of prehensile behaviors [8] for physical object manipulations;
- externalizes traditionally internal computer representations;
- facilitates interactions by making interface elements more "direct" and more "manipulable" by using physical artifacts;
- takes advantage of our keen spatial reasoning [2] skills;
- offers a space multiplex design with a one to one mapping between control and controller; and finally,
- affords multi-person, collaborative use.

# BASIC CONCEPTS

Graspable UIs allow direct control of electronic objects through physical artifacts which we call *bricks*. The bricks, approximately the size of LEGO(TM) bricks, sit and operate on a large, horizontal computer display

surface (the Active Desk, described later). A *graspable* object is an object composed of both a physical handle (i.e., one or more bricks attached) and a virtual object (see Figure 1).

The bricks act as specialized input devices and are tracked by the host computer. From the computer's perspective, the brick devices are tightly coupled to the host computer -- capable of constantly receiving brick related information (e.g., position, orientation and selection information) which can be relayed to application programs and the operating system. From the user's perspective, the bricks act as physical handles to electronic objects and offer a rich blend of physical and electronic affordances.

## One Handle

In the simplest case, we can think of the bricks as handles similar to that of graphical handles in computer drawing programs such as MacDraw(TM) (see Figure 2a). A physical handle (i.e., a brick) can be attached to an object. Placing a brick on the display surface causes the virtual object beneath it to become attached (see Figure 2b). Raising the brick above the surface releases the virtual object. To move or rotate a virtual object, the user moves or rotates the attached brick (see Figure 3). Note that the virtual object's center of rotation is at the center of the brick.

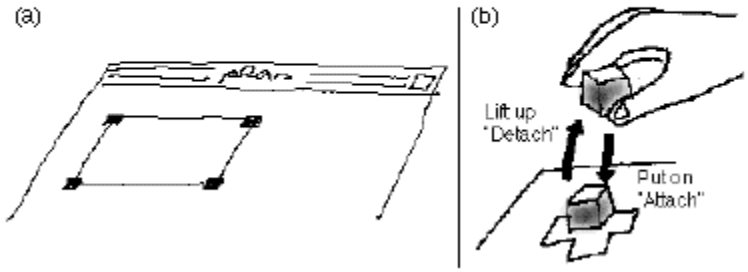


**FIGURE 2. (a) Traditional MacDraw-like application which uses electronic handles to indicate a selection. (b) Selecting using a brick.**

**FIGURE 3. Move and rotate virtual object by manipulating physical brick which acts as a handle.**

A simple example application may be a floor planner (see Figure 5a). Each piece of furniture has a physical brick attached and the user can arrange the pieces, most likely in a rapid trial-and-error fashion. This design lends itself to two handed interaction and the forming of highly transient groupings by touching and moving multiple bricks at the same time.

## Two Handles

More sophisticated interaction techniques can be developed if we allow more than one handle (or brick) to be attached to a virtual object. For example, to stretch an electronic square, two physical bricks can be placed on an object. One brick acts like an anchor while the second brick is moved (see Figure 4).

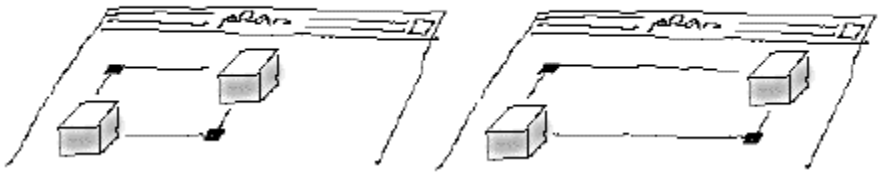

**FIGURE 4. Two bricks can stretch the square. One brick acts like an anchor while the second brick is moved.**

Placing more than one brick on an electronic object gives the user multiple control points to manipulate an object. For example, a spline-curve can have bricks placed on its control points (see Figure 5b). A more compelling example is using the position and orientation information of the bricks to deform the shape of an object. In Figure 6, the user starts off with a rectangle shaped object. By placing a brick at both ends and rotating them at the same time, the user specifies a bending transformation similar to what would happen in the real world if the object were made out of a malleable material such as clay. It is difficult to imagine how this action or transformation could be expressed easily using a mouse.

One key idea that the examples illustrates is that the bricks can offer a significantly rich vocabulary of expression for input devices. Compared to most pointing devices (e.g., the mouse) which only offers an x-y location, the bricks offer multiple x-y locations and orientation information at the same instances of time.

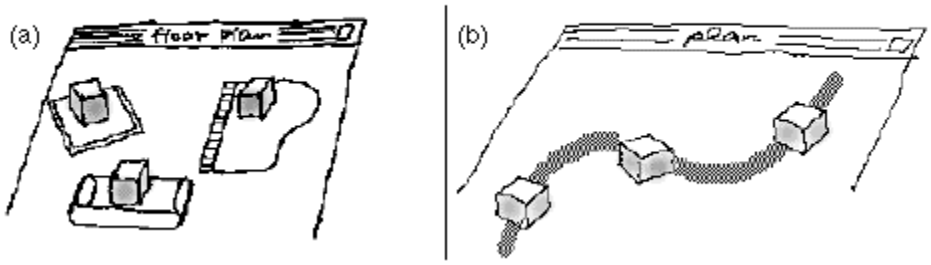


**FIGURE 5. (a) Proposed simple floor planner application. (b) Many physical bricks are used for specifying multiple control points for creating a spline curve.**

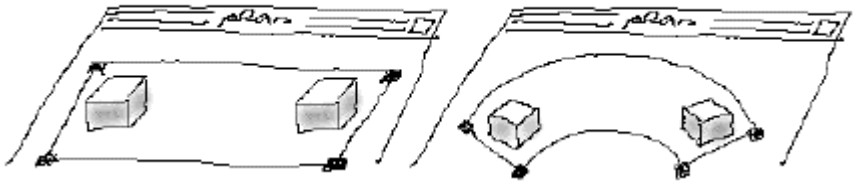

**FIGURE 6. Moving and rotating both bricks at the same time causes the electronic object to be transformed.**

# RELATED RESEARCH

Some research and commercial systems have been developed with a similar graspable theme. In some sense, many of these emerging systems exhibit the property of ubiquitous computing [14] in which computation is embedded in many physical artifacts and spread throughout our everyday environment. The following systems

illustrate the push towards ubiquitous computing, physical manipulation interfaces and merging physical and virtual artifacts.

The LegoWall prototype (developed by A/S Modulex, Billund Denmark in conjunction with the Jutland Institute of Technology in 1988) consists of specially designed LEGO blocks that fasten to a wall mounted pegboard panel composed of a grid of connectors. The connectors supply power and a means of communication from the blocks to a central processing unit. This central processing unit runs an expert system to help track where the blocks are and what actions are valid.

The behavior construction kits [9] consist of computerized LEGO pieces with electronic sensors (such as light, temperature, pressure) which can be programmed by a computer (using LEGO/Logo) and assembled by users. These LEGO machines can be spread throughout the environment to capture or interact with behaviors of people, animals or other physical objects. The "programmable brick," a small battery powered computer containing a microprocessor, non-volatile ROM and I/O ports is also being developed to spread computation.

The AlgoBlock system [13] is a set of physical blocks that can be connected to each other to form a program. Each block corresponds to a single Logo-like command in the programming language. Once again, the emphasis is on manipulating physical blocks each with a designated atomic function which can be linked together to compose a more complex program. The system facilitates collaboration by providing simultaneous access and mutual monitoring of each block.

Based on a similar philosophy of the 3-Draw computer-aided design tool [11], Hinckley et al. has developed passive real-world interface props [5]. Here users are given physical props as a mechanism to manipulate 3D models. They are striving for interfaces in which the computer passively observes a natural user dialog in the real world (manipulating physical objects), rather than forcing a user to engage in a contrived dialog in the computer generated world.

Finally, the DigitalDesk [15] merges our everyday physical desktop with paper documents and electronic documents. A computer display is projected down onto a real physical desk and video cameras pointed at the desk use image analysis techniques to sense what the user is doing. The DigitalDesk is a great example of how well we can merge physical and electronic artifacts, taking advantage of the strengths of both mediums.

# STAGE 1: EARLY BRICK EXPLORATIONS

A series of quick studies was conducted to motivate and investigate some of the concepts behind Graspable UIs. Having decided on bricks, we wanted to gain insights into the motor-action vocabulary for manipulating them.

## LEGO separation task

The first exploratory study asked subjects to perform a simple sorting task as quickly as possible. The basic idea was to get a sense of the performance characteristics and a range of behavior people exhibit while performing a task that warrants rapid hand movements and agile finger control for object manipulation. Subjects were presented with a large pile of colored LEGO bricks on a table and were asked to separate them into piles by color as quickly as possible.

We observed rapid hand movements and a high degree of parallelism in terms of the use of two hands throughout the task. A very rich gestural vocabulary was exhibited. For instance, a subject's hands and arms would cross during the task. Subjects would sometimes slide instead of pick-up and drop the bricks. Multiple bricks were moved at the same time. Occasionally a hand was used as a "bulldozer" to form groups or to move a set of bricks at the same time. The task allowed subjects to perform *imprecise* actions and interactions. That is, they could use mostly ballistic actions throughout the task and the system allowed for imprecise and incomplete specifications (e.g., "put this brick in that pile," which does not require a precise (x, y) position specification). Finally, we noticed that users would enlarge their workspace to be roughly the range of their arms' reach.

## Domino sorting task

The second exploratory study asked subjects to place dominos on a sheet of paper in descending sorted order. Initially, the dominos were randomly placed on a tabletop and subjects could use the entire work surface. A second condition was run which had the dominos start in a bag. In addition, their tabletop workspace was restricted to the size of a piece of paper.

Once again this sorting task also revealed interesting interaction properties. Tactile feedback was often used to grab dominos while visually attending to other tasks. The non-dominant hand was often used to reposition and align the dominos into their final resting place while, in parallel, the dominant hand was used to retrieve new dominos. The most interesting observation was that subjects seemed to inherently know the geometric properties of the bricks and made use of this everyday knowledge in their interactions without prompting. For example, if 5 bricks are side-by-side in a row, subjects knew that applying simultaneous pressure to the left-most and right-most end bricks will cause the entire row of bricks to be moved. Finally, in the restricted workspace domino condition we observed one subject taking advantage of the "stackability" of the dominos and occasionally piled similar dominos on top of others to conserve space. Also, sometimes a subject would use their non-dominant hand as a "clipboard" or temporary buffer while they plan or manipulate other dominos.

## Physical manipulation of a stretchable square

To get a better sense of the issues for manipulating physical versus virtual objects, we designed a "stretchable square" constructed out of foam core. This square looks like a tray with a one inch rim around each side. Users could expand or collapse the width of the square (see Figure 7). We displayed an end position, orientation and scale factor for the physical square and asked subjects to manipulate the square to match the final target as quickly as possible. A variety of cases were tested involving one, two or all three transformation operations (translate, scale, and rotate).

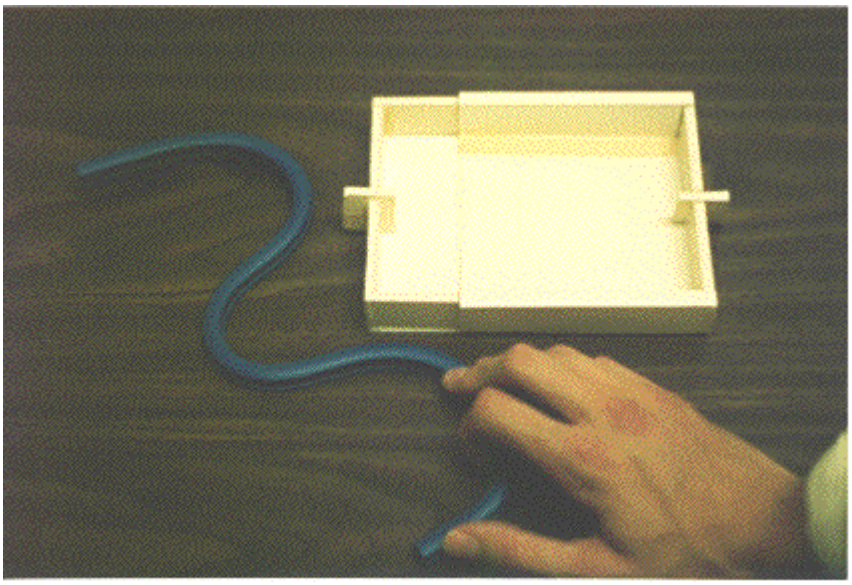

**FIGURE 7. Flexible curve and stretchable square.**

We found that each subject had a different style of grasping the stretchable square for position and orientation tasks. This served to remind us that physical objects often have a wide variety of ways to grasp and to manipulate them even given natural grasp points. In addition, subjects did not hesitate and were not confounded by trying to plan a grasp strategy. One subject used his dominant hand to perform the primary manipulation and the non-dominant hand as a breaking mechanism and for finer control.

Perhaps the most salient observation is that users performed the three operations (translation, rotation and scaling) in parallel. That is, as the subjects were translating the square towards its final position, they would also rotate and scale the square at the same time. These atomic operations are combined and chunked together [1].

### Comparison Using MacDraw Application

The same matching tasks were then done using virtual objects and a stylus on a large, horizontal drafting table with a computer display projected on the writing surface. Using the MacDraw II(TM) program, subjects were asked to move a virtual object on top of a target virtual object matching position, orientation and scale factors.

We observed that even when we factor out the time needed to switch in and out of rotation mode in MacDraw, task completion time was about an order of magnitude longer than the physical manipulation using the stretch-able square. We noticed a "zoom-in" effect to reach the desired end target goal. For example, subjects would first move the object on top of the target. Then they would rotate the object, but often be unable to plan ahead and realize that the center of rotation will cause the object to be displaced. Thus, they often had to perform another translation operation. They would repeat this process until satisfied with a final match.

The MacDraw user interface, and many other interfaces, forces the subject to perform the operations in a strictly sequential manner. While we can become very adept at performing a series of atomic operations in sequence, the interface constrains user interaction behavior. In effect, the interface forces users to remain novices by not allowing them to exhibit more natural and efficient expressions of specifying atomic operations in parallel.

### Curve Matching

Continuing to explore our skills at physical manipulations, we asked subjects to use a flexible curve (see Figure 7) to match a target shape. The flexible curve, often used in graphic design, consists of a malleable metal surrounded by soft plastic in the shape of a long (18 inch) rod. The inner metal allows the curve to hold its shape once deformed.

We found that users quickly learned and explored the physical properties of the flexible curve and exhibited very expert performance in under a minute. All ten fingers were often used to impart forces and counterforces onto the curve. The palm of the hand was also used to preserve portions of the shape during the curve matching task. We observed that some subjects would "semantically load" their hands and arms before making contact with the flexible curve in anticipation of their interactions. The semantic loading is a preconceived grasp and manipulation strategy by the user which, in order to execute properly, the arms, hands and fingers must start in a specific, sometime uncomfortable, loaded position. This process often allowed the subject to reach the final target curve shape in one gestural action.

## STAGE 2: MOCK-UP AND SIMULATIONS

Next, we mocked-up some sample brick interactions using a prototyping tool (Macromind Director) and acted them out on the Active Desk. By using a few LEGO bricks as props and creating some basic animations using the prototyping tool, we could quickly visualize what the interactions would look like and begin to get a sense of how they will feel. These sample interactions were video taped and edited. We were able to mock-up many of the primary ideas such as: attaching and detaching bricks from virtual objects; translation and rotation operations using one brick; using two bricks each attached to separate virtual objects, and finally two bricks attached to a single virtual object to specify stretching and simple deformations.

All of these exploratory studies and mock-ups aided us to quickly explore some of the core concepts with minimum set-up effort. Finally, the video tapes that we create often serves as inspirational material.

## STAGE 3: PROTOTYPE

After the mock-ups, we built the bricks prototype to further investigate the Graspable UI concepts. The prototype consists of the Active Desk, a SGI Indigo2 and two Ascension Bird receivers (see Figure 8).

## Active Desk

The Active Desk is a large horizontal desktop surface which has a rear projected computer screen underneath the writing surface (see Figure 8). Modeled after a drafting table, the dimensions of the surface are roughly 4.5' by 3.0' on a slight 30 degree angle. The projected computer screen inset has a dimension roughly 3' by 2'. A Scriptel transparent digitizing tablet lays on top of the surface and a stylus device may be used for input. The LCD projection display only has a 640x480 resolution so the SGI screen is down converted to an NTSC signal and sent to the LCD display.

## Bricks

To prototype the graspable objects (bricks), we use the Ascension Flock of Birds(TM) 6D input devices to simulate the graspable objects. That is, each receiver is a small 1 inch cube that constantly sends positional (x, y, and z) and orientation information to the SGI workstation. We currently have a two receiver system, which simulates two active bricks that operate on top of the Active Desk. More receivers can be added to the system but the wires attached to the receivers hinder interactions. Nevertheless, the two receivers offer us an initial means of exploring the design space in a more formal manner.

## GraspDraw -- A simple drawing application

A simple drawing application, GraspDraw, was developed to test out some of the interaction techniques. The application lets users create objects such as lines, circles, rectangles and triangles (see Figure 8). Once created, the objects can be moved, rotated and scaled. GraspDraw is written in C using the GL library on an SGI Indigo2.

The two Bird receivers act like bricks and can be used simultaneously to perform operations in parallel. One of the bricks has a push button attached to it to register additional user input. This button is primarily used for creating new objects. Grasps (i.e., attaching the brick to a virtual object) are registered when a brick is near or on the desktop surface. To release a grasp, the user lifts the brick off of the desktop (about 2 cm).

To select the current tool (select, delete, rectangle, triangle, line, circle) and current draw color, we use a physical tray and an ink-well metaphor. Users dunk a brick in a compartment in the tray to select a particular tool. A soft audio beep is heard to act as feedback for switching tools. Once a tool is selected, a prototype shape or tool icon is attached to the brick. The shape or icon is drawn in a semi-transparent layer so that users may see through the tool.

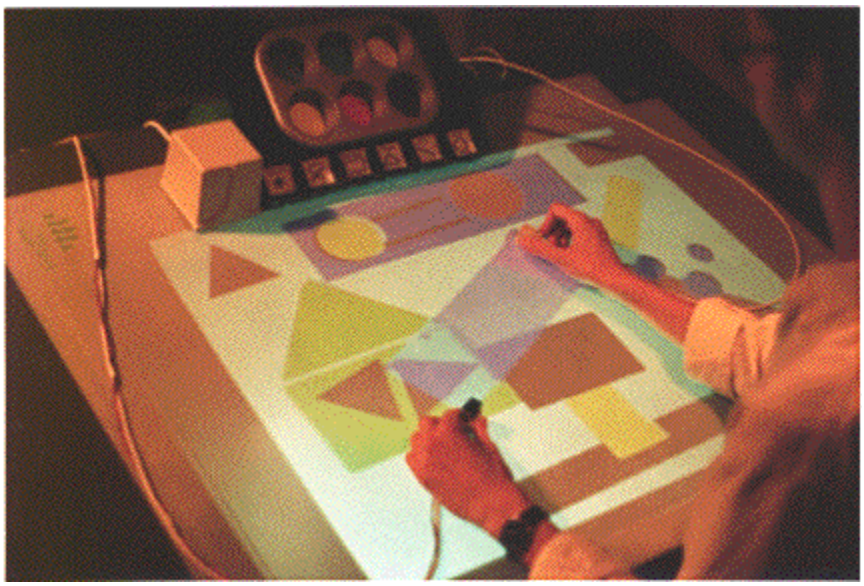

**FIGURE 8. GraspDraw application and ActiveDesk.**

The concept of an *anchor* and *actuator* have been defined in interactions that involve two or more bricks. An anchor serves as the origin of an interaction operation. Anchors often specify an orientation value as well as a positional value. Actuators only specify positional values and operate within a frame of reference defined by an anchor. For example, performing a stretching operation on a virtual object involves using two bricks one as an anchor and the other as an actuator. The *first* brick attached to the virtual object acts as an anchor. The object can be moved or rotated. When the *second* brick is attached, it serves as an actuator. Position information is registered relative to the anchor brick. If the first anchor brick is released, the actuator brick is promoted to the role of an anchor.

# STAGE 4: COMMERCIAL APPLICATION

In following the goals of user centered design and user testing, there are some real problems when working with new interaction techniques such as the Graspable UI. First, in order to conduct formal experiments, one must generally work in a restricted controlled environment. The demands of experimental control are often at odds with human performance in the more complex context of the real world. Secondly, University researchers typically do not have access to the source code of anything but toy applications. Therefore, testing and demonstrations of innovative techniques like the Graspable UI are subject to criticisms that "it's fine in the simple test environment, but it won't work in the real world."

The first point can be dealt with by careful experimental design and differentiating between controlled experiments and user testing. Our approach to the second is to partner with a commercial software company that has a real application with real users. In so doing, we were able to access both a real application and a highly trained user community.

Hence, we have implemented a critical mass of the Graspable UI into a modified version of Alias Studio(TM), a high-end 3D modeling and animation program for SGI machines. Specifically, we are exploring how multiple bricks can be used to aid curve editing tasks. Although we have just begun this stage of research, we currently have two bricks integrated into the Studio program. The bricks can be used to simultaneously edit the position, orientation and scale factor for points along a curve. Future investigations may use bricks to clamp or freeze portions of the curve. This integration process and evaluation will further help us to refine the Graspable UI concepts.

# DISCUSSION

We have conducted some preliminary user testing of the bricks concept using the GraspDraw application. All of the approximately 20 users who have tried the interface perform parallel operations (e.g., translate and rotate) at a very early stage of using the application. Within a few minutes of using the application, users become very adept at making drawings and manipulating virtual objects. Some users commented on the fact that the bricks were tethered, which hindered some of their interactions.

One could argue that all Graphical UI interactions, except perhaps touch (e.g., touchscreens) are already graspable interfaces if they use a mouse or stylus. However, this claim misses a few important distinctions. First, Graspable UIs make a distinction between "attachment" and "selection." In traditional Graphical UIs, the selection paradigm dictates that there is typically only one active selection; Selection N implicitly causes Selection N-1 to be unselected. In contrast, when bricks are attached to virtual objects the association persists across multiple interactions. Selections are then made by making physical contact with the bricks. Therefore, with Graspable UIs we can possibly eliminate many of the redundant selection actions and make selections easier by replacing the act of precisely positioning a cursor over a small target with the act of grabbing a brick. Secondly, Graspable UIs advocate using multiple devices (e.g., bricks) instead of channeling all interactions through one device (e.g., mouse). Consequently, not only are selections persistent, there can be one persistent selection per brick. Thirdly, the bricks are inherently spatial. For example, we can temporarily arrange bricks to form spatial caches or use them as spatial landmarks for storage. By having more spatial persistence, we can

use more of our spatial reasoning skills and muscle memory. This was exhibited during the LEGO and Domino exploratory studies. Clearly, the bricks are handled differently than a mouse.

One may suggest to eliminate using bricks and instead use only our hands as the physical input devices. While this may be useful for some applications, in general using a physical intermediary (i.e., brick) may be more desirable. First, tactile feedback is essential; it provides a way of safeguarding user intent. The bricks supply tactile confirmation and serve as a visual interaction residue. Secondly, hand gestures lack very natural delimiters for starting and stopping points. This makes it difficult to segment commands and introduces lexical pragmatics. In contrast, the affordances of touching and releasing a brick serve as very natural start and stop points.

There are many open design issues and interaction pragmatics to research. For example, should we vary the attributes of a brick (shape, size, color, weight) to indicate its function? Should all the bricks have symmetrical behavior? How many bricks can a user operate with at the same time? Do the bricks take up too much space and cause screen clutter (perhaps we can stack the bricks and they can be made out of translucent material)? For fine, precise pointing, do bricks have a natural hot spot (perhaps a corner or edge)? Sometimes it is more advantageous to have a big "cursor" to acquire a small target [6].

### Inter-Brick behaviors

Much of the power behind the Bricks is the ability to operate and interact with more than one brick at the same time. Our interaction techniques need to be sensitive to this issue and define consistent inter-brick behaviors for one-handed (unimanual) or two-handed (bimanual) interactions. Moreover, we will need to develop a new class of techniques that use combinations of unimanual and bimanual interactions during the life span of a single technique. For instance, a technique may be initiated using one hand, transfer to using both hands and then terminate back to using one hand. The key point is that we need to provide for seamless transitions within a single interaction technique that switches between unimanual and bimanual interactions. As we noted earlier, the Anchor/Actuator behavior serves as one example.

Our goal has been to quickly explore the new design space and identify major landmarks and issues rather than quantify any specific subset of the terrain. The next phase of our evaluation will include a more detailed evaluation at places in the design space that have the most potential.

# DESIGN SPACE

We have developed an initial design space for bricks which serves to lay the foundation for exploring Grasp-able UIs. Table 1 summarizes the design space. The shaded region in the table represents where our current Bricks prototype fits in the design space. Each of the rows in the table represent dimensions of the design space which are described below.

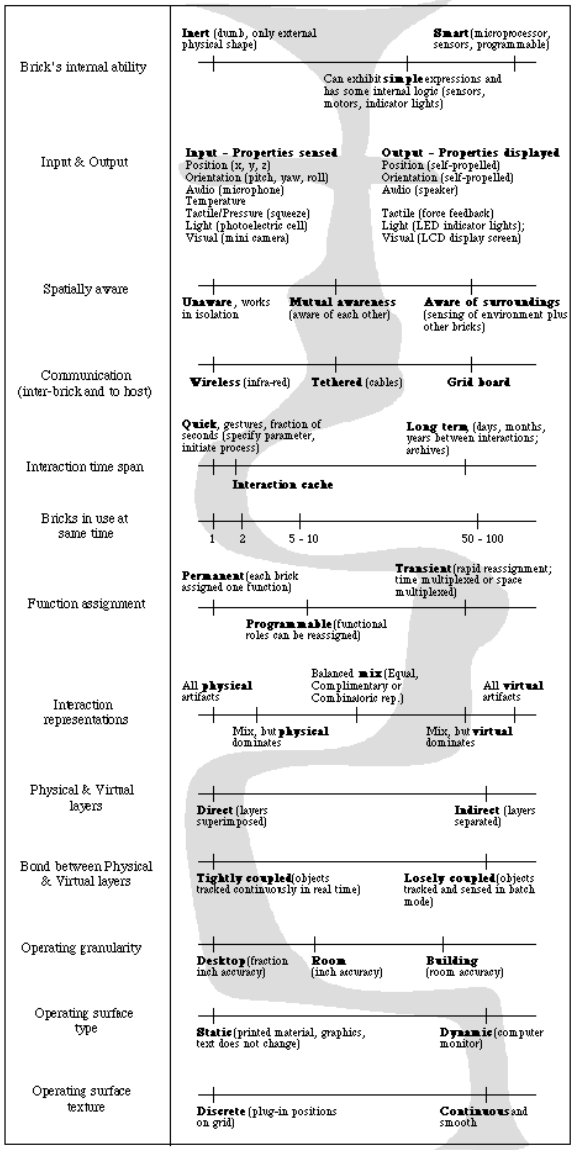


**TABLE 1. Design space of Bricks for Graspable User Interfaces. Gray region shows where current Brick prototype fits into design space.**

*Brick's internal ability* -- Does the brick have any internal mechanisms (physical or electronics) that generates additional information or intelligence? Inert objects have no internal mechanisms, only external features (color, shape, weight). Smart objects often have embedded sensors and microprocessors.

*Input & Output* -- What properties can be sensed and displayed back to the user (or system)?

*Spatially aware* -- Can the brick sense the surroundings and other bricks? Bricks can be unaware (work in isolation); mutually aware (aware only of each other); or be aware of their surroundings (primitive sensing of its environment and other bricks) [4].

*Communication* -- How do the bricks communicate among themselves and to host computers? The mechanisms range from wireless (such as infra-red), tethered (requiring wires or cables) and grid board (a specialized operating surface with pluggable connecting parts).

*Interaction time span* -- Given a task, are users manipulating the bricks in quick bursty interactions (sometimes gesturing in fractions of a second); using a set of bricks, accessing them within seconds or minutes (an interac-

tion cache); or are the interactions long term running days, months, years between interactions (e.g., an archive)?

*Bricks in use at same time* -- Do users manipulate one brick at a time (one handed interactions), two at a time (two handed interactions), or more than two? Users could manipulate 5 to 10 bricks at a time (e.g., bulldozer) or perhaps even 50 to 100 at a time.

*Function assignment* -- How frequently and by what mechanism do the bricks get assigned their functions? Permanent assignment means that each brick has one function or role for its lifetime. With some effort, programmable assignment allows bricks to have their function reassigned. Transient assignment allows for users to rapidly reassign the brick's function.

*Interaction representations* -- Is the system designed to have a blend of physical and virtual artifacts? When there is a mix, are the dual representations equal (i.e., functions can be performed using either physical or virtual artifacts), complimentary (i.e., one medium can perform a subset of the functionality that the other medium cannot) or combinatoric (together both offer functionality that either one could not provide alone).

*Physical & Virtual layers* -- Are the layers direct (superimposed) or indirect (separated)?

*Bond Between Physical & Virtual layers* -- Tightly coupled systems have the physical and virtual representations perfectly synchronized, the physical objects are tracked continuously in real time. Loosely coupled systems allow for the representations to become out of synchronization for long stretches of time (i.e., minutes, days) and updated in more of a batch mode.

*Operating Granularity* -- What is the range of space that the bricks operate in and at what sensing resolution? For example, the devices may operate at a *building* level (e.g., capable of determining what room they are currently in), *room* level (e.g., capable of determining, within an inch accuracy, position and orientation information inside a room), and *desktop* level (e.g., micro accuracy within 0.1 in of position and orientation information on a desktop).

*Operating surface texture* -- What granularity or texture do the bricks operate on? A discrete texture requires that the bricks be plugged into special receptors (e.g., a grid board) while a continuous texture allows for smooth movement or dragging (e.g., tabletop).

*Operating surface type* -- Do the bricks operate on a static surface (e.g., a tabletop) or a dynamic surface which can be changing constantly (e.g., Active Desk)?

It should be noted that this is not an exhaustive parsing of the design space. Robinett [10], however, proposes a more formal taxonomy for technologically mediated experiences which may aid our investigation. Yet, the many dimensions of our design space exhibit its richness and provides a more structured mechanism to explore the concepts behind Graspable UIs.

# FUTURE WORK

There are many future directions we would like to explore. First, we will conduct more formal evaluation measures on the GraspDraw program. Next we will investigate other regions of the design space including developing techniques in 3-D as well as to operate on 3-D virtual objects. In addition, we hope to develop multiple, untethered bricks. Two promising areas are computer vision techniques [12] and electric-field sensing [16].

# CONCLUSIONS

In this paper we have introduced a new technique, the Graspable User Interface, as a means of augmenting the power of conventional Graphical User Interfaces. In so doing, we have attempted to go beyond a simple "show and tell" exercise. Through the methodology described, we have attempted to both explore the overall design

space effectively, and tease out the underlying human skills on which we could build our interaction techniques.

The Graspable User Interface is an example of "radical evolution." It is evolutionary in the sense that it builds upon the conventions of the GUI. Hence, both existing technology and human skill will transfer to the new technique. However, it is radical in that the incremental change that it introduces takes us into a radically new design space. Assuming that this new space is an improvement on what preceded it, this combination gives us the best of both worlds: the new and the status quo.

From the experience gained in the work described, we believe these new techniques to be highly potent and worthy of deeper study. What we have attempted is a proof of concept and exposition of our ideas. Hopefully this work will lead to a more detailed exploration of the technique and its potential.

# Acknowledgments

This research was undertaken under the auspices of the Input Research Group at the University of Toronto and we thank the members for their input. We especially would like to thank Mark Chignell, Marilyn Mantei, Michiel van de Panne, Gordon Kurtenbach, Beverly Harrison, William Hunt and Kim Vicente for their input. Thanks also to Ferdie Poblete who helped design and build the stretchable square prop. The Active Desk was designed and built by the Ontario Telepresence Project and the Arnott Design Group. Our research has been supported by the Information Technology Research Centre of Ontario, the Natural Sciences and Engineering Research Council of Canada, Xerox PARC and Alias Research.

More material including dynamic figures can be found on the CHI'95 Electronic Proceedings CD-ROM and at URL: http: //www.dgp.utoronto.ca/people/GeorgeFitzmaurice/home.html

DYNAMIC FIGURE 1 (QuickTime Movie, about 10 mb) **No caption given.**

# References

1. Buxton, W. (1986). Chunking and phrasing and the design of human-computer dialogues. *Proceedings of the IFIP World Computer Congress*. pp. 475-480.
2. Eilan, N., McCarthy, R. and Brewer, B. (1993). Spatial Representation. Oxford, UK: Blackwell.
3. Guiard, Y. (1987). Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain Model. In *Journal of Motor Behavior*, 19(4), pp. 486-517.
4. Fitzmaurice, G.W. (1993). Situated Information Spaces and Spatially Aware Palmtop Computers, *Communications of the ACM*. 36(7), pp. 38-49.
5. Hickley, K., Pausch, R., Goble, J. C. and Kassell, N. F. (1994). Passive Real-World Interface Props for Neurosurgical Visualization. *Proc. of CHI'94*, pp. 452-458.
6. Kabbash, P. and Buxton, W. (1995). The 'Prince' Technique: Fitts' Law and Selection Using Area Cursors, To appear in *Proc. of CHI'95*.
7. Kabbash, P., Buxton, W. and Sellen, A. (1994). Two-Handed Input in a Compound Task. *Proc. of CHI94*, pp. 417-423.
8. MacKenzie, C. L. and Iberall, T. (1994). The Grasping Hand. Amsterdam: North-Holland, Elsevier Science.
9. Resnick, M. (1993). Behavior Construction Kits. In *Communications of the ACM.* 36(7), pp. 64-71.
10. Robinett, W. (1992). Synthetic Experience: A Proposed Taxonomy. *Presence*, 1(2), pp. 229-247.
11. Sachs, E., Roberts, A. and Stoops, D. (1990). 3-Draw: A tool for the conceptual design of three-dimensional shapes. CHI'90 Technical Video Program, ACM SIGGRAPH Video Review, Issue 55, No. 2.

12. Schneider, S.A. (1990). Experiments in the dynamic and strategic control of cooperating manipulators. Ph.D. Thesis, Dept. of Elec. Eng., Stanford Univ.
13. 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, pp. 297-303.

14. Weiser, M. (1991). The computer for the 21st Century. In *Scientific America*, 265(3), pp. 94-104.
15. Wellner, P. (1993). Interacting with paper on the DigitalDesk. In *Com. of the ACM.* 36(7), pp. 86-96.

16. Zimmerman, T., Smith, J.R., Paradiso, J.A., Allport, D. and Gershenfeld, N. (1995). Applying Electric Field Sensing to Human-Computer Interfaces. To Appear in *Proceedings of CHI'95*.