INTERFACE DESIGN
William Bricken
July 1989

The interface we are currently using developed during a prototyping phase of the project, a phase which emphasized core capabilities rather than ease of use. Our interface served its purpose excellently, providing a solid format for demoing to over 200 people. Now we have the responses of these people, we have many hours of experience, and we intend to take our demo to the professional level of SIGGRAPH.

In a demo, the medium is most certainly the message. This is especially so within a compelling environment, one which encourages suspension of disbelief. The content of the Cyberspace demo is the visual image, the feel of the headset, the connecting wires, the person guiding your experience, and your position in the real room. The environment itself, both real and virtual, determines the quality of the experience. As McLuhan has said: it is the method or process of an experience that determines its content. Our process includes restraining a patron to a local physical space, transferring sight to another realm, and establishing a trusting relation between the patron and the guide. Within the virtual reality, the processes of viewing, movement and manipulation determine the impression of the patron, not the objects and content with which the patron interacts. This is supported by the fact that enjoyment does not vary over the two demo experiences.

Although the concept and the potential of Cyberspace were overwhelmingly supported, the mechanics of bringing folks to Cyberspace were substantially criticized by our sample. Some hardware related concerns are within our control. Frame rate, for instance, has a known solution (more hardware computing power). Some hardware, like the headset, poses a problem not within our direct control. The important observation, though, is that we can and should address all software problems. So I'll assume that a comprehendible software interface is more important than frame rate.

From the perspective of the questionnaire data, software criticisms are not common. The headset and the frame rate dominated explicit concerns. From the perspective of the demo videos, the demo experience is quite awkward. The glove is a particular problem, as is the wheel. The sources of difficulty are spread across the hardware, our software, and our demo style.

The observations are of novices learning Cyberspace for the first time. Naturally, the awkwardness might go away as a patron becomes comfortable and experienced. The objective of this memo is to improve our SIGGRAPH *demo*, to configure the interface so that novices will not embarrass us at a professional forum. The suggestions are partial, since we should not attempt to change too much in the short time remaining.

SUMMARY

We should design interface tools to maximize freedom. Too little control and too much control both degrade the experience of freedom.

CONTENT

Folks love intrinsic perspective. Head-tracking and wide field-of-view combine to create the feeling of *inclusion*. "The sense of being there." Inclusion is intimately linked with the ability to move in Cyberspace, and to the views provided. Folks loved the feeling of freedom to move about, "the ability to move", the ability to fly. They loved "being able to view any perspective". These positive comments were not related to particular input devices such as the glove or the wheel or the orb.

The single advantage of intrinsic perspective is *freedom*, the feeling of interacting with the computed image as-if in a physical space. We should design interface tools to maximize freedom.

Freedom does not imply the desire to get lost. Automated orientation provides a sense of freedom by taking care of the bookkeeping of position. Freedom has a curvilinear relation to cognitive processing load (the main point of my first graduate thesis). Too little control and too much control both degrade the experience of freedom.

DESIGN PRINCIPLE

The structure of the interface must mimic the structure of the body.

Full spatial freedom in a 3D Cartesian model provides three independent vectors for translation and three (not independent) planes of rotation. The six parameters (x, y, z, turn, tilt, lean) are artifacts of the Cartesian model.

The body, in contrast, is not constructed as an abstract model of space. We have strong biases relative to Cartesian coordinates.

Translational bias: Our eyes are frontal. Translating forward is more natural than translating backward (frontal/dorsal bias, x-axis). Both are more natural than translating left and right (bilateral bias y-axis). Translating up and down (gravitational bias, z-axis) is unnatural, requiring an instrument such as an elevator or airplane.

Rotational bias: The gravitational bias makes lean (rotation around the x-axis, roll) unnatural. Turn (around the z-axis, yaw) is the only natural rotation, because it is the only symmetric axis. Tilt (around the y-axis, pitch) is partially natural, since we are hinged at the waist for forward tilt.

So the rough hierarchy is:

    frontal translation
    turn
    lateral translation
    tilt
    lean
    gravitational translation

The flying model makes gravitational translation natural while minimizing tilt and lean. The dynamics of flight mixes turn with lean, but the dynamics of perception cancels the lean. We feel as though we are level as the plane banks. (So the unusual dynamics of the wheel, turning without leaning, did not bother people.)

The headset responds to head movements. We use our head for rotations, predominately turning, secondarily tilting, rarely leaning. We are built so that head turning and body turning are redundant. We rarely both turn and tilt our heads or our bodies in different directions.

Our demo is dominated by these restraints: folks can physically turn while standing, they can tilt their heads while sitting (but they rarely do). They can also lean their heads, however this is considered disorienting.

Experiment: Try turning off lean, holding all views horizontal.

Movement in an intrinsic perspective is limited to frontal translation. Lateral translation is achieved by turning and heading forward. Flight provides gravitational translation. Perceptually, tilt is constrained to head movement, lean is eliminated.

An input device for flying needs to provide only

    forward/backward translation,
    up/down translation, and
    right/left rotation

One way to do this is with two joysticks. The original intrinsic perspective game Battlezone used

    both forward = forward, velocity in proportion to deflection
    both back = backward, velocity in proportion to deflection
    right back and left forward = turn right
    left back and right forward = turn right.

Vertical movement can be controlled by a mode button:

both buttons down, up/down rather than forward/back
one button down, up from that joystick, forward from other joystick.

Another method is to configure the 6df orb by removing the signals for lateral translation, tilt and lean.

Forward/backward = forward/backward translation
Twist left/right = turn left/right
Up/down = up/down translation

Another configuration: the wheel and a mode button.

wheel turn = turn
wheel push/pull = forward/backward translation
push/pull plus one button = up/down translation
two buttons = accelerate

THE SPECIAL CASE OF THE GLOVE

The glove has undeniable appeal, six folks liked "the ability to manipulate graphical objects with the glove". The glove provides a separate visual indicator of inclusion, and a separate channel of motor-visual coordination mediated by software.

The glove is also a major modeling problem. Folks often loose the image of their hand. It is difficult to calibrate well. Folks try to move by holding their glove in awkward positions, they hold their arm awkwardly throughout glove demos. Eighteen folks singled out the glove, its lag, and its perceptual ambiguity as dislikes.

The glove is used to indicate a direction of translation and to grasp objects. Although the glove is lifelike, its cost is high. It introduces computational lag while tracking finger movements, requires calibration, is inaccurate, is difficult to align, and requires awkward movements to use. It forces physical locality in order to grab, accentuating qualities that Cyberspace is currently weak in (distance perception and controlled movement). To pick up a book, most people bang into the shelves, or move their perspective two inches from the object. The entering hand is often accompanied with a question ("Have I got it?") and always needs reinforcement from the guide ("You've got it now"). Letting go was equally clumsy, people needed to be told repetitively to stop grabbing, to fly while holding, to open palm to let go.

The basic problem with using a glove is that *not gesturing* is unnatural.
When I don't want to gesture, my hand is usually by my side. Few of our

patrons relaxed their arm when not gesturing. To stop flying by stop pointing is also unnatural. Another gesture, such as flat palm, is better since positive gestures are associated with change in the real world.

If we stay with a gestural language, it now consists of three commands and a void concept:

NOTHING RECOGNIZED: stop flying
POINT: to fly
GRASP: to attach to the hand
FLAT: to release from the hand

It's easy to use just the forefinger to provide the information for all these commands. Forefinger coding would eliminate all the bending and rebending we do when the glove does not seem to respond to a person's hand (I'd say about 10% of the time). It might also be possible to cut the thumb calibration out of the calibration sequence.

POINT is "straight forefinger": this displays a vector from the virtual finger along the path of flight.

GRASP is "fully bent forefinger": to pick up.

The new POINT and GRASP have the same hand gesture as the old one, it's just that we don't care about the rest of the fingers.

Stop flying is stop pointing, just like we do now. Stop holding (release) is stop the grasp gesture. To fly while grasping, the patron must change the current grasp to point directly, say within a half second. Anything else is release.

If we add a third gesture, we need to read another finger. FLAT means the middle finger is straight, as well as the forefinger. Then we have a comfortable positive language:

POINT to fly
GRASP to attach
FLAT to stop flying and to stop grasping

The sequence GRASP then POINT, without an intervening FLAT means fly while attached.

Stopping pointing without FLAT is ambiguous. I rather like a gesture to stay in force until I explicitly cancel it. This means the gesture gets read once and doesn't go away. But this is getting extended, since the following better plan is to get rid of the glove altogether.

Options to the glove must combine indication of direction with identification of objects. A Polhemus and spoken words are certainly adequate to replace the glove. Cyberspace is ideal for spoken word commands, the Polhemus furnishes directional information.

THE WAND

The Wand is simply a Polhemus mounted on a holding stick. Software would convert the positional signal to a vector emanating from the end of the stick. The display would be of the pointing device, the wand, possibly with a static hand wrapped around it. The Polhemus would still provide positional information. The hand-and-wand would remain at anthropomorphic distances, and provide the same illusion of connectivity as a glove.

The wand could be placed in a stand in a standard location for initialization, making WandWorlds as easy to enter as CyberCity. All hand computation is eliminated. Instead the wand would send out a beam from its end, stopping at the first polygon it hits.

The wand vector forms the basis for all operations, the required verbal instructions are:

> WAND ON/OFF: toggles the wand's functionality. On or Off.
> VECTOR ON/OFF: (optional)
> FORWARD/STOP/REVERSE: Fly along the wand vector
> FASTER/SLOWER: Change velocity along the wand vector. Only if flying.

REVERSE could be eliminated, since folks can turn their bodies around.

JACK: constructs a Jack (oriented by default in the same orientation as the user) on the surface of the object.

A jack has many purposes. It identifies particular objects. It locates a perspective (point-of-view) on another object. It is a handle and a toggle point.

To grasp an object, point the wand at it and say CONNECT. This attaches the end of the wand vector to the object. When the wand is moved, the attached object moves with it, on the end of a skewer. To let go, say RELEASE.

> CONNECT/RELEASE: use the wand vector to move objects.

When the wand is placed inside an object, we degenerate to the current glove grabbing technique. Note that we can translate in two dimensions (normal to the wand vector) and rotate in one (around the wand vector). This is also the same functionality as the glove. The Polhemus on the glove is limited to the flexibility of our wrists, the Polhemus on a wand is not.

New features offered by the wand:

> SWITCH: move perspective to the jack at the end of the wand.
> DISTANCE: report the length of the wand vector.

SWITCH lets us change perspectives naturally, by pointing.

So we have a single intuitive device that is simple, computationally elegant, general (provides many functions), and direct. It is also easy to make into a paddle for RacquetWorld.

SHORT CUTS

We can get nearly the same functionality out of the glove by turning finger tracking off, and using the glove-mounted Polhemus as the wand. Without voice input, we are back to defining a set of gestures to use as signals. Regardless, the wand/glove vector concept is needed.

We're exploring a set of buttons attached to a wand to replace voice control. A possible button configuration would be:

> WAND ON/OFF: a single click button
> FORWARD/STOP: one button
> FASTER/SLOWER: should be omitted
> CONNECT/RELEASE: a button
> JACK: omit

So with three buttons we can get adequate functionality. Three buttons is the outside maximum we can expect people to learn and remember in a short demo. Two would be much better. In any case, buttons will require cognitive processing to locate and remember, decreasing freedom.

OTHER THINGS

No void spaces: We must provide visual cues from all locations, except very high off the ground.

With our current setup, we should initialize OpenPlan with keyboard input rather than making a fist, which tilts the hand and disorients the Polhemus alignment. It would be a great improvement to just smoothly hold out the hand and get typed into OpenPlan.

We should make the grasp-beep a once only signal, or eliminate it entirely. Just say no to beep-beep-beep-beep-beep-....