Where Do You Want to Fly Today?
A User Interface Travel Genre Based on Flight Simulators

Raymond R. Panko	David K. Panko
University of Hawaii	Iolani School

Abstract

Document genres help readers orient themselves when they approach a new document. Once they know a document's genre, they understand what the document is likely to contain and how they probably can navigate the document. User interfaces also fall into genres that allow users to orient themselves. For instance, Xerox introduced the powerful desktop genre, which facilitated user insight when they worked at isolated PCs. This powerful genre helps users move between Macintosh, Windows, and other systems whose interfaces conform to the basic rules of the genre. Today, however, users often travel away from their desktops, to distant corporate and Internet computers. Under these conditions, we need new user interface "travel genres" based on movement metaphors. This paper suggests one such genre, based on a the metaphor of flight simulator games.

Introduction

In the 1970s, Xerox created a genre of graphical user interfaces (GUIs) based on the virtual desktop metaphor. The user interfaces of Star, Lisa, Macintosh, Solaris, and Windows are all members of this genre. Although a GUI places a heavy burden on machine processing power, the GUI desktop genre allows people to learn to use their systems quickly and probably to work more productively. In addition, this interface is attractive. People who viewed DOS as ugly were often dissuaded from using computers. When many of these people first saw a desktop GUI, in contrast, they wanted to use the system.

Among documents, genres help orient a reader approaching a new document. For instance, if you pick up a text book, you know that it will be divided into chapters and will probably contain a table of contents and an index. You can probably also find a glossary, review questions, and other material common to that genre. At a subtle level, text books usually have chapter numbers somewhere at the top or bottom of each pair of facing pages. User interface genres have a similar value in orienting users, but only as long as the genre's structure (usually based on a metaphor such as a desktop) is consistent with the work to be done.

Today, our computers have a great deal more processing power than their 1980s counterparts, which were barely able to support the flat desktop metaphor. In addition, 3D processing hardware and software APIs are becoming increasingly common. This means that we now have the processing to develop three-dimensional user interfaces or at least simulated 3D interfaces as we have in such computer games as SimCity and various flight simulators.

Why do we need 3D user interfaces? The most basic answer is that we intellectually no longer just sit at our desks. We constantly leave our desktop PC's hard disk to "surf" the Internet, access distant corporate hosts. and use PC network file servers. Instead of a *desktop genre* for user interfaces we need *travel genres* to represent what people do today. Instead of helping people point to things on their desks, we can let them travel through an information space containing their own computer, their corporate neighborhood, the Internet, and other "locations."

Instead of a *desktop genre* for user interfaces we need *travel genres* to represent what people do today

1060-3425/98 $10.00 (C) 1998 IEEE

Computer gaming has already brought us one possible travel metaphor to use in future user interface genres. This is the flight simulator metaphor. In flight simulator games, you travel from place to place, often flying over realistic 3D-like terrain. The navigation, pointing, and other techniques used in flight simulators generally appear to be applicable to a broader range of travel metaphor issues. In addition, the control stick hardware used in flight simulators is widely available and is already present on many home PCs. Finally, aircraft control embodies experience gained from experiments and long experience dealing with proper control operation and information revelation.

This paper is based on our reflections on flight simulator metaphors, based on our analysis with flight simulator games. It represents preliminary thinking rather than a coherent framework.

In flight simulator games, there is a strong bias toward realistic operation for the aircraft being simulated. The simulated aircraft should respond to actions just as a real aircraft would in terms of gravity, lift, and other physical considerations. In addition, the user should see only what a pilot of the aircraft would see. In contrast, we have chosen a *relaxed adaptation* stance, which argues that strict fidelity to a flight system metaphor should be sacrificed when desirable to reflect the fact that this is a computer interface rather than a simulation of aircraft flight. For instance, in computer analogs of an aircraft control panel, when you move your mouse or control stick reticle over a specific control, a small explanatory note may appear, just as it does in Windows when you move your mouse over an icon on a toolbar. In simulating a real aircraft, this would be inappropriate. As a computer user interface aid, however, it is completely appropriate.

The Control Panel

Figure 1 shows that nearly every flight simulator game has a *control panel* at the bottom of the screen. (Some do not. We discuss the benefits of such omissions below.) This control panel takes up valuable screen real estate, and it is usually possible to turn it off, giving a clear view of the outside world. However few game players use this option.

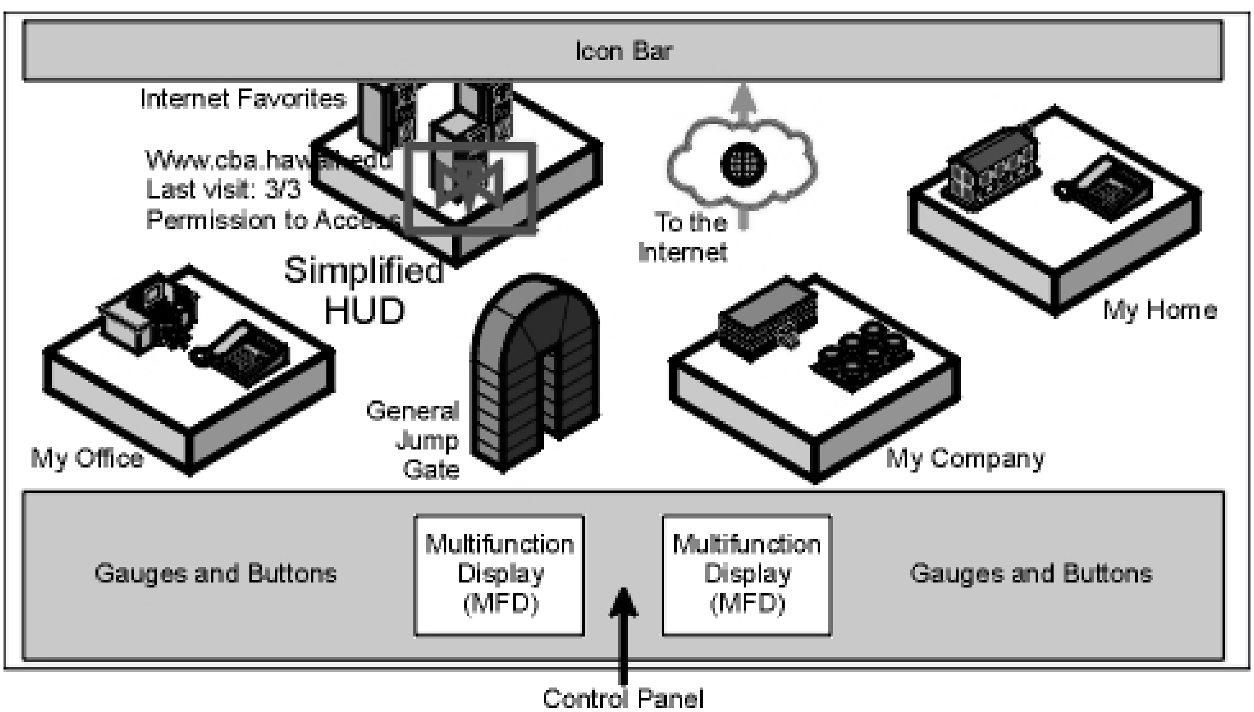


Figure 1: Control Panel, Heads Up Display, Reticle, and Objects

The control panel, while intrusive, is extremely valuable. In addition, as we move toward 17” screens at home and 21” screens in business, the space penalty for the control panel will be reduced.

Consistent "Flight" Information

The key advantage of control panels is that they give consistent "flight" information that you may need in all situations, such as speed, altitude, available resources, and so forth. For instance, if you are surfing the Internet, one gauge may keep track of your online time. Always having the same information available reduces mental workload. Even if there is a keyboard button that toggles the control panel on or off, merely knowing that it is available is helpful.

As discussed earlier, following the principle of relaxed adaptation, when you pass your mouse cursor or control stick reticle over a control button or object visible through the "windshield," a little note could pop up explaining its function. The note may even have a small icon that would lead to a tutorial on the use of that button or display.

Multifunction Displays

One characteristic of modern aircraft control panels is the presence of one or two *multifunction displays (MFDs)* on the control panel. As Figure 1 shows, these are computer screens. For each MFD, there is a small number of fixed-format pages that the pilot can call up. Each page gives a carefully crafted view of a coherent set of information.

Following the principle of relaxed adaptation, one can envision that these MFD windows will have standard Windows control features. For instance, you may be able to expand the MFD to a larger size or even full screen size. While this would make no sense in a real aircraft, most people should find it sensible in the computer domain.

In addition, in real aircraft, the pilot has no ability to create custom multifunction displays. An incorrect MFD page could lead to a fatal crash. There is no need for this restriction in computer navigation. While having a well-conceived repertoire of pre-set MFD pages is essential, there is no reason to restrict customization.

Heads Up Display

In aircraft, the pilot looks out at the world through a glass panel called the *heads up display* or *HUD*. The HUD shows basic flight information in a (relatively) unobtrusive way. The HUD allows the pilot to get immediate access to critical information without looking down at the control panel.

Usually, the HUD is a fixed plane of glass sitting just before the windshield. When the pilot looks forward, he or she sees critical flight information without having to look down at the control panel.

However some aircraft helmets have HUDs linked to the pilot's helmet. The pilot has a monocle in one eye, onto which HUD information is projected. As the pilot moves his or her head, the eye mounted display moves with them. In a computer interface, the reticle that shows where the control stick is pointing could have its own associated HUD for local information relevant to the spot to which the control stick is pointing. For instance, if you point to a webserver image on your screen, information about that webserver would appear in the movable HUD.

In current user interfaces, when you move a cursor over an icon, a small box usually appears to tell you about the icon. However this opaque box prevents you from seeing behind it. In contrast, HUD information is shown with a transparent background, so that you can see the scene behind it.

One useful feature of aircraft HUDs is that they often have a small arrow that shows the direction in which you should fly to get to a desired destination. For instance, if your destination is to the left of your viewable screen, there will be an arrow at the top of the HUD pointing left. This would also be useful in computer interfaces.

Control Stick Operation (and Beyond)

For most flight simulators, it is almost mandatory to use a control stick. Although you can use a mouse, it will be like writing while a brick is attached to your pencil.

Basic Controls

The most obvious controls in the control stick is the movement of the entire stick forward and backward or side to side.

The second most obvious control on the control stick is the trigger button. In games, this allows you to shoot at things. In a general computer interface, it would allow you to select or open an item. In a HUD, there is a visible change when you have a “lock” on an object. In a computer display, you could do this with the HUD, but you could also have the object change visibly as well. Pressing the trigger would open the item.

Modern control sticks also have additional buttons. Although these buttons are preassigned by default, they can be user-programmed in most flight simulator environments. This allows the user to do certain common chores with little delay. One button on the control panel, by the way, should give immediate access to a list of current button assignments and other “properties” and “options” information, because these are easily forgotten.

There are two types of buttons on most control sticks. One is the *hat*, which is like a miniature control stick. It can be pushed up or down, right or left. Other buttons are of the conventional press-and release variety. For instance, in flight simulators, one button might switch you to the next logical target. For navigation, in turn, a button might switch you to the next object on the screen, allowing you quick selection without pointing.

Finally, most control sticks have one or more rotating wheels that can increase or decrease some function (usually speed) in small increments throughout some range.

Using a Control Stick to Move Around

To move around on screen, there are two basic operations—translation and rotation. In *translation*, you move along an axis without turning. For instance, translating up and down is like being in an elevator. Helicopters and jump jets can also translate up and down. Translating forward and backward is like walking forward or backward in a straight line. Translating left and right is like a crab shuffling sideways.

The second motion is *rotation*. In aircraft terminology, *pitch* is rotating the nose up or down. *Yaw* is rotating the aircraft right or left, which is like panning while taking pictures with a camcorder. Finally, *rolling* is rotating around the aircraft’s front-back axis; it is like turning a screw driver.

In this environment with *six degrees of freedom*, there is also a time element. For each dimension, we can have position, velocity (speed in a particular direction), and acceleration. For translation dimension, position, speed, and acceleration are obvious. For rotation, the correlates are number of degrees rotated from the resting position, angular velocity (spin rate), and acceleration in angular velocity.

To complicate matters, aircraft movements usually couple movement in different dimensions. For instance, if you rotate the control stick right, you roll (rotation around the central axis) the aircraft. Differential air pressure over the wing will yaw you (rotation around the vertical axis) in the direction of the roll. In addition, you lose altitude (vertical translation velocity).

To give another example of coupling, when you pull the control stick back, you pitch up (rotate around the left-right axis). You also gain altitude (translate upward), but you to lose speed (change the forward translation velocity).

Flight simulator operation, of course, are patterned after aircraft controls. Pulling the control stick forward or backward or right or left produced coupled motion. Most control sticks do not emulate the rudder, which gives (almost) pure yaw, although you can program your hat for rudder operation. The Microsoft Sidewinder control stick does allow rudder operation because the control stick itself can rotate around the vertical axis.

Coupled motion leads to a difficult design choice. One option is to continue aircraft control principles with their coupled motions. This will be natural for people who have played flight control games. However control stick flight control was designed for aircraft flying forward. Continuous forward motion can be undesirable in computer interfaces. (Imagine that when you open a file in Windows, you automatically scroll through files without stopping unless you took a special act to stop. This problem, by the way, exists in helicopter flight simulations.) In addition, for people who have not used flight simulation, linkages between roll and other actions and other linkages can create a learning burden.

Another set of design issues stems from the fact that an aircraft simulation, usually allows you to use more than a control stick. For instance, foot pedals for yawing (rudder control) are desirable. In helicopter simulations, there must be a way to translate without rotating. Such things typically require using devices other than pure control stick motion. The most common tools are a hat control on the control stick and a rotating wheel on the control stick base.

Another option is to use a different device designed for 3D operation. For instance, there can be a ball control that can rotate in three axes slightly. The ball can also go forward, backward, left, right, and even up and down. The problem with this approach, of course, is that it requires special hardware.

In practice, it would probably be necessary to develop control processes for 3D control units, traditional control sticks, and even mice. Otherwise, the usefulness of the interface would be severely limited.

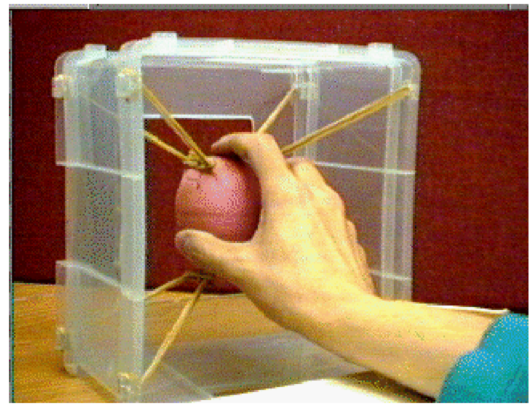
Pointing On Screen

Another issue is aircraft control versus *gaze* among the objects within the view through the windscreen. Suppose you are flying a helicopter in a combat simulation. Sometimes, you want to move your entire aircraft front or back or side to side. Turning the control stick does this. The entire scene in front of you rolls or yaws. In other cases, you might want to point to something on the current screen. For instance, in a combat simulation you may wish to point to a particular target without shifting the entire aircraft. If you wish to do this with your control stick, then the control stick must have two modes—one for turning the entire aircraft, another for moving a pointing reticle around on the current screen without turning the aircraft. Of course using one tool in two modes always causes problems.

Eventually, we may wear eye tracking lenses that allow us to point to something on screen as helicopter pilots can now point to specific targets. This would allow the control stick to have only a single mode. It would allow simultaneous scene movement and pointing within a scene.

Going back to the 3D control device, if the ball that can be swiveled up or down and left and right, this can give pointing capability. Figure 2 shows such a device.

Figure 2: Pointing Device with Six Degrees of Freedom



One possibility is to have pointing mean different things when you are within a screen than when you are at its edge. For instance, in a word processor, you can point anywhere on a screen with your cursor. But in some word processors, if you move your cursor to the top or bottom of the screen, you will automatically scroll up or down. In a flight simulator metaphor, when you joystick reticule is within the screen, it can mean pointing; but when you move it to the edge of a screen, you may rotate your "aircraft" in that direction. Again, this is relaxed adherence to the flight simulator metaphor.

Speed

One issue is how fast to move when the user moves through the 3D space in front of him or her. In many flight simulator games, keeping pressure on the control stick can adjust forward/backward speed and side/side speed.

Beyond the Single Hand

While the control stick is in some ways an improvement on the mouse, real pilots also have separate opposite-hand controls and foot controls. The opposite-hand control typically is a throttle, but it can have other buttons as well. The foot control is for yawing left and right through rudder control. Not even games can be counted on to have opposite-hand controls, foot controls, or both. Still, these should be integrated into the interface.

More generally, when Engelbart built the NLS system in the 1960s, there was an opposite-hand control, namely a five-key keyboard whose keys could be pressed in any of 32 possible combination.

Although people vary widely in their ambidexterity, adding an opposite-hand control of some sort seems to make a great deal of sense. Foot pedals were also used in NLS to jump between the graphic and text screens. As noted earlier, we may eventually use eye-tracking helmets as well. Certainly, we should not create a user interface that assumes only a control stick or worse yet only a mouse.

Navigation

Having looked at controls, we can now move to the higher issue of navigation—how the user finds his or her way around through 3D space.

Rooms

Figure 1 shows the basic flight simulator view that you get when you are sitting in the cockpit. But in most games, this is not your only situation or view. For instance, there is usually a briefing room before the flight, a debriefing room afterward, a view of the base, and so on.

Each of these "rooms" is loaded with objects you can click on for information, to set parameters, and so forth. In part, the meaning of the objects in indicated by their shape. In part, it is indicated by their location in the room.

With a 3D view instead of a 2D view, you could "fly" over these rooms as well, perhaps looking down through open ceilings on the objects below. This would probably give you a better view than the traditional 2D view or the Doom-like view that requires you to rotate your view within the room. If you keep the same angular perspective, it would be less jarring going from a level of overview down to a view of a lower level structure.

You might, by the way, have a house with a separate "bedroom" for each person using the computer or an office suite with a separate room for each person using the office computer. By going to "their room," each person would get a customized view. Continuing this trend, each person would have their own aircraft parked outside (which they could customize in terms of color scheme and features, of course) as well as access to a multimedia training room and other facilities.

The Local View

Figure 1 shows the local view outside the cockpit. The local view is what you see before you. Whatever the view would be, it would be a 3D map of something. For instance, these might be 3D representations of World Wide Web hosts. Or, they might be pictures of jump gates (which are described below). You should be able to see the object from multiple sides as well. Finally, color or even texture can have categorical meaning. For instance, in a search, items that contain both search terms might be red, while those that contain only one may be blue.

Gradual Revelation

One trick of gaming is to have farther objects appear not just smaller than closer objects but also dimmer and less defined, as if partially obscured by haze. Such hazing could help the user focus on nearby objects in the cockpit viewscreen.

Emphasis

In accurate flight simulations, building sizes represent their physical sizes. However there is no need to continue this physical analogy with a computer interface. Host icon sizes, at least to a degree, could represent the importance of the host. Brightness could also reflect the importance of various host computers and other resources. Even blinking could be used.

In addition, you might wish to classify things by color. For instance, all websites that deal with entertainment might be in blue. This should be set up as a style, so that the user can declare all entertainment sites read at a later time.

Pop-Up Map

One feature of almost all flight sims is a *pop-up map*. This should be brought up directly from the control panel. Usually, this map can be seen at different levels of zoom, with more detail added at higher levels of zooming.

There could also be multiple maps, with their names listed on the bottom of the pop-up map window. One map, for instance, may have all your most recently sites. Another might have your "favorite" websites. This would make your map window act like a multifunction display. In fact, one of the *MFDs* on the control panel could be the pop-up map.

One map might be your *waypoints map*. As you travel, you may want to mark certain sites so that you can return to them. Current browsers have bookmarks, but they are permanent. In the waypoints map, you can mark particular sites as waypoints *only for that flight*. You should also be able to mark waypoints ahead of time, to help you plan your "trip." You can also add future waypoints during a flight.

Jump Gates to Starting Points

If you are traveling over a small landscape, travel time is not important. However if you are going some distance, you might need a travel aid. Given the flight metaphor, you might have a *jump gate* button on your control panel that would show you a wormholes that would allow you to *hyperjump* instantly to any of several preset distant locations.

Objects on the Cockpit Viewscreen

We would envision the cockpit viewscreen *vista*—what you see outside the windscreen—to be built automatically but to be modifiable easily by users. SimCity 2000 comes to mind as a possible metaphor for the ability to modify and build the vista. Your vista might contain several different *plateaus* (building sites) for different purposes (entertainment, news, technical material), add hosts and other resources to these plateaus. Figure 1 shows the use of plateaus.

One or more plateaus might be frozen in place in the distance until your clicked on them. These might be "favorites" plateaus, plateaus holding jump gates specialized for common hyperjumps, and so forth. This is not shown in Figure 1.

Search

Each plateau might also have an *information center* icon (not shown in Figure 1), say it its lower right hand corner. Clicking on this icon would allow you to search for a specific host in your current vista or out in the flat world of World Wide Web searches. A jump gate could allow you to jump there immediately.

In Context

In general, what we have discussed is a 3D interface borrowing from flight simulator games and certain other games, such as SimCity. This very derivation, however, raises some questions.

Too Macho?

One issue is whether the flight simulator metaphor is too male-oriented. While it might help people with game experience learn to navigate the world of resources beyond the desk more easily, flight simulator games tend to be used more by men than by women. Perhaps if the travel metaphor were expressed primarily in automobile imagery, instead of flight simulator imagery, it might be more broadly palatable. Perhaps the imagery of moving through a shopping mall would be appropriate. This is already used in some virtual malls.

KISS or Augmentation?

In the 1960s, Doug Engelbart developed NLS, which introduced the mouse and hypertext. His goal was to augment the human intellect by allowing people to approach, comprehend, and solve larger problems than they could have handled without technology. He did not build a system that was easy to use. Rather, he built a system that required extensive training to maser.

Computer flight simulations are also like this. In part, this is because gamers really do need to do complex things when they play. In part, however, it merely reflects the desire to simulate a complex real aircraft and also a desire to keep the game challenging.

When Xerox hired away many of Engelbart's team members, it build systems that introduced the desktop imagery but that also focused radically on simplicity. They discarded hypertext documents and the three-button NLS mouse. They also discarded the idea of an opposite hand control. While these KISS (keep it simple, stupid) steps may have made acceptance easier, they also set us back many years in functionality that we are just now recovering. The World Wide Web, for instance, could have appeared in the early 1980s instead of the early 1990s.

Our interface would seem to be difficult to master with all of its features. However just as games have skill levels, there could be skill levels in

1060-3425/98 $10.00 (C) 1998 IEEE

the user interface. Individuals could start at a fairly simple skill level that would probably even have a simplified control panel. For instance, in the simple game Raptor, you fly over a flat 2D map, seeing straight down. You can only translate forward and backward, left an right. Additional capabilities could be added individually or in groups. These additional skills would be linked to optional multimedia training room sessions. One could also envision the user choosing their personal aircraft from a list of several alternatives with different degrees of complexity in operation.

Virtual Reality

Although virtual reality helmets are still many years from becoming common, they add a new dimension of space. For most people, the jump from a 14" screen to a 21" screen is an amazing leap in their ability to view the computer world. VR can potentially increase the amount of viewable space ten-fold. With a head mounted display, you could see not merely out a forward viewscreen but also in perhaps 180 degrees around your front view. In flight simulators, you can do this only be turning your whole aircraft or hitting a function key that turns your "head" left or right with a discontinuous jump. This is very disorienting.

With virtual reality, eye tracking is only one tool. There are data gloves and other devices to record gestures. These gestures can easily give six degrees of freedom while adding the ability to "click" and take other actions, as shown in Figure 3.

Figure 3: Gesture

True 3D

In this paper, we have focused not on true 3D but on the types of simulated 3D that you get today in flight simulators. The reason was processing power. With today's computers, simulated 3D is attainable relatively easily. We will not have to go through the agonies of the early slow Macintoshes.

At the same time, we need to consider true 3D. For instance, if you are conducting a World Wide Web search with a search engine, relevance ratings could be organized in three dimensions. This would allow you to move in a true 3-space. At the same time, games such as DOOM have shown that many people have a difficult time with 3D visualization when they do not have the flexibility of a head-mounted display but only have a fixed screen in front of them. It is like walking around the world your head held fixed in the forward position by a neck brace.

Individual Differences

As we move to travel genres, we will need to conduct new research on individual differences involving visualization. For instance, in the game of *Doom*, some people take naturally to the turning motions and can visualize their surrounding space easily. Others become highly disoriented. We would like travel genres that are not highly disorienting to a substantial percentage of potential users.

Conclusion

As we move out to the Internet and corporate-wide facilities, we can visit literally have millions of computers. However the desktop imagery that worked quite well for individual desktop machines may be the wrong metaphor for a world in which we are essentially leaving our offices to go out into the broader corporate and Internet worlds. We need to move beyond interface genres that work best for a single desktop, toward genres based on travel metaphors that reflect the distant location of many of the resources we use.

In this paper, we have discussed how flight simulator games suggest several ways to allow people to travel in a 3D world. One key theme has been consistency. Our control panel is always visible in the flight mode unless turned off. Viewing things from above, using the same angle perspective, should make it easier to zoom in and out and move from scene to scene.

Another key theme has been relaxed adaptation, that is, not following the flight simulator metaphor slavishly. Mechanical aircraft have limitations that we do not need to repeat in the computer world. Just because a button on the control panel does not tell you about itself when you look at it, this is no reason for buttons on your computer control panel not to pop up a brief description when you pass your mouse or flight stick reticle over them.

One of the major issues in the design of travel genre interfaces based on flight simulation metaphors is the six degrees of freedom we have in 3D situations. It is difficult to map three translation and three rotation motions into a control stick or worse yet, into a mouse. While there are several ways to do the mapping with control sticks or mice, it is not easy to decide which would work the best. Control devices created specifically for 3D travel would solve this problem, but such devices are not widely available on computers today.

Finally, our experience suggests that viewing 3D objects from above, as in Figure 1 will be less disorienting than viewing objects all around you, as in the game Doom.

Authors

Dr. Raymond R. Panko is a professor of information systems at the University of Hawaii in Honolulu, Hawaii. His e-mail address is panko@hawaii.edu. His URL is http://www.cba.hawaii.edu/panko.

Mr. David K. Panko is a student at Iolani School in Honolulu, HI.

This paper grew primarily out of their reflections on the applicability of flight simulations they had used to the broader issue of user interface design.