Reliability of Interactive Computer Exhibits
or, Why Doesn't This @#!!#& Thing Work?

Jim Oker
New England Technology Group
One Kendall Square, Bldg. 700, Cambridge, MA 02139

It is important to know how to analyze the components that make up an interactive system. The components that suffer the worst abuse are the "visitor input devices." The safest approach is to use parts from devices that are known to stand up to unsupervised public abuse. Unfortunately, incorporating these devices into exhibits is not always easy. Standard mice, keyboards, and trackballs can be as troublesome as they are easy to use. Touchscreens are excellent interfaces for many exhibits. There are a few types of touchscreen, each with advantages and disadvantages. Proper program design and attention to overall system reliability can save touchscreens much abuse.

Broken software can be harder to resolve than broken hardware. You can call on a range of people to write your software, each with advantages and disadvantages. Off-the-shelf packages may not perform as advertised, and may exhibit bugs which will never be repaired. There are strategies which avoid most of these problems.

Each device has environmental constraints (temperature, humidity, dust control, etc.) which must be taken into account very early in the planning process, as the constraints effect fabrication, electrical work, air conditioning, and the general use of space. Locating all the equipment at the exhibit kiosk has advantages and disadvantages, as does placing some equipment in a remote control room.

No matter how reliable a system is, it may go down at some point. If no one has the job of dealing with problems (and adequate time available), the system will probably stay down for a very long time. The larger an investment a museum makes in interactive exhibits, the more important this issue becomes. Due to the pace of change, a museum may find that a few years after installing a system, it can no longer purchase a replacement for a system element. The system can be designed to minimize this problem.

A frustrating visitor interface may leave visitors thinking that the program is "broken." Slow response time may encourage more aggressive use of the input device. Constantly accessing one sector of a floppy disk can cause problems. In short, poor program design can contribute to downtime. Though there are a few guidelines, the best way to develop a good design is to involve the right parties in the design process, and to conduct formative evaluations.

Introduction

"This thing is broken." Usually a plaintive cry uttered by a child telling you that one of your more expensive (per square foot) exhibits is, for the moment, worse than empty floor space. Trouble. How often you have trouble and what happens after it starts are important determinants of the worth of any exhibit. Interactive computer exhibits are especially distressing for many museums. Creating and maintaining these exhibits is a technical specialty. Yet it is possible for any museum to anticipate and manage the inherent risks. After years of dealing with a wide variety of museums, I have met some bogeymen that await you.

"Interactive computer exhibits" is a very general term, ranging from a touch screen driven videodisc presentation to an station which records pulse, flexibility, and grip strength for a larger health profile. Because the exhibits are interactive, they are subject to the same abuse as your flip-panels, water screws, puzzles, and other interactive mechanical exhibits. Because they involve computers, they may even break when no one is using them.

Reliability of Individual Components

The most obvious contributor to a system's reliability is the reliability of each of its components. One broken element usually means that the entire exhibit is down. I could list my current picks and pans for hardware. Products, however, have short life cycles. The unique minds of exhibit designers call for using ever newer devices. It is better to know what to look for rather than have a list of good products.

The components that suffer the worst abuse are the "visitor input devices." Most museum professionals don't need to be told to be wary of buttons, joysticks, and the like. Standard mice, keyboards, and trackballs found in computer magazines usually come with software toolkits. Because these devices are easy to integrate into a range of off-the-shelf software (e.g., Hypercard), they are seductive.

A more prudent approach is to use parts from devices that are known to stand up to unsupervised public abuse (parts from arcade games, vending machines, automated teller machines, payphones, etc.). These devices can be found in catalogs and magazines that cater to the appropriate trades. Unfortunately, incorporating these robust industrial devices into exhibits takes more effort than using consumer oriented items. It takes quite a bit of research to find the parts and learn what they really do. You will probably have to write your own software drivers as well.

Joysticks provide an excellent example. We have seen good consumer quality joysticks break in under a month of use. Arcade quality joysticks are known for spraining wrists without breaking ("PAC Man Wrist" made it into the New England Journal of Medicine). However, a consumer joystick will plug into a "game port," a standard item on many computers. The consumer item will also work with a wide range of software. The arcade joystick works a bit differently from the consumer version. To make it robust, the designers simplified it. A consumer joystick will send out numbers that represent a range of x,y positions to the computer. Such sticks can be used, for instance, to guide a cursor across a screen with a fair amount of control. The arcade sticks simply trigger one of four (or eight in a slightly fancier version) switches as they are pulled or pushed. The simplicity of the design allows for very sturdy construction at a cost that is lower than the cost of the consumer item. The problem, however, is that the exhibit designer must take into account the different characteristics when planning a visitor interface. The PAC Man arcade game is an example of a good design for an arcade stick. The player guides a PAC Man around a maze by pushing the stick in one of four directions. Using consumer software as a starting point for program design may lead the designer down an unworkable path.

There are very high quality joysticks which behave like consumer joysticks. They are made for the military, and carry price tags well over one thousand dollars a piece (versus well under one hundred for the arcade style).

Touchscreens are excellent interfaces for many exhibits. Out of over five hundred touchscreen systems that we have put in the field, we have seen very few problems other than gradual wear. Though a touchscreen may be more expensive than buttons or trackballs, they require no more fabrication expense than would a monitor alone.

Touchscreens come in a few flavors, and each has its strong and weak points. Each also has its advocates. NETG has favored the "resistive" style of screen. The touch sensor is a plastic and glass sandwich that is curved to fit the shape of the monitor tube. By taking a standard monitor, removing the tube, taping a curved touch panel to it, and reassembling the monitor, you have a touchmonitor. A drawback to these screens is that they present a plastic membrane surface to the public. We have seen some of our screens come back with scratches that disabled the screen. We have also seen a few come back with the glass backing broken, evidence that a visitor had punched or pounded the screen (perhaps in frustration, see below under "Design of the Program"). The screens also reduce the brightness of the screen somewhat, and if purchased with an anti-glare coating, fuzz the image a little (we use anti-glare coating for the obvious advantages). On the positive side, these screens offer fine resolution and quick response, allowing more software flexibility than other styles of screen.

Another common type of screen uses infra-red sensors around the edge of the screen. Visitors' fingers interrupt both a vertical and a horizontal beam as they get near the tube surface. These touchscreens offer the advantage that visitors are touching the sturdy surface of the monitor's tube. However, they have relatively low resolution, and can be defeated if visitors put a piece of gum on the edge of the monitor tube. They also provide significant parallax error, where visitors of different heights will interrupt a different beam though they are all aiming at the same point on the monitor screen. Without extensive experience, I can't offer an opinion as to their overall reliability.

There are a few other styles of screen. Before using any style of screen, it is wise to speak with other museums who have had the same type of screen installed for enough time to spot any problems.

Other common system elements include computers, expansion cards, monitors, and disc players. These can be evaluated through the experience of others, mean time between failure ratings, and inspection and testing of samples. A few general principles include: lots of moving parts can mean trouble (printers, video tape players, etc.); items that require care and feeding are tough for most museums to keep up with (e.g., printers that require ribbons, paper, and un-jamming); and industrial equipment is often more reliable than consumer equipment (but sometimes it's just more expensive).

The imagination of designers can bring in a host of other devices. It's tough to be doing "new" things with the same old reliable equipment. When possible, it's a good idea to try out individual elements, subjecting them to some abuse in a simple prototype system, before incorporating them into a system. Get kids to come in to play with your prototype. They will do things you would never imagine.

Reliability of the Software

Broken software can be harder to resolve than broken hardware. While it is easy to swap in a spare piece of hardware, if the software has a bug or doesn't perform to expectations, the choice of action is much less clear. The possible problems vary in nature depending on whether a museum is creating it own program or hiring a consultant or company to create custom software.

NETG regularly uses off-the-shelf packages as building blocks for our custom software. This practice saves our time, our customers' money, and allows us to incorporate good ideas from a wide variety of sources. We never commit to using a package for a project based on published information. Magazines and company literature often inaccurately depict the capabilities of off-the-shelf packages, embellishing without pointing out the limitations of features. Even the best magazine reviewers won't be using the software in the same way that we intend to. Many "interactive software" packages don't provide the sort of "bomb-proofing" that we feel is required in an unsupervised public access setting. After years of disappointing experiences, we have developed an intuition that allows us to read between the lines. We can anticipate some of the unpublished drawbacks. Even with our experience, however, we continue to be surprised.

We try to work with a limited set of packages. Each has its own quirks which we have learned from extensive trial, debugging, and spending countless hours on the phone with telephone support staff. In some cases the telephone support is relatively useless, leaving us to solve the problems on our own, devising "work-arounds" for observed problems.

We have efficiently incorporated off-the-shelf software into our work by amortizing the effort involved in learning each package over many projects. Museums that are creating interactive computer exhibits on their own should allow adequate time to surmount the steep learning curve before they get the first program running. Nevertheless, creating your own interactive computer programs can be rewarding and cost-effective in some cases.

One way to avoid some of these problems is to hire a consultant to work with you on your program. Most cities have many contract programmers who are experienced with some sort of interactive software. The wisdom of this approach varies widely depending on your ability to manage the consultant, the complexity of the exhibit, and the actual experience of the consultant. Most programmers are very confident of their abilities, even when they are approaching unfamiliar territory. Ask about relevant experience and check references.

One of the greatest problems with contract programmers is their transience. Will they be around in eight months when you begin to have problems or would like changes made? This is where your management of the programmer is important. Your consultant may leave town. Another programmer should be able to make sense of your consultant's work. Hold contract programmers to established standards of programming style and documentation (a topic too involved to even gloss over here). You should either get the source code from them or, if they are unwilling to provide it at a reasonable cost, arrange to have it placed in escrow where you can get at it if they move on.

I will grant that I a biased by my job. I feel that the most cost-effective approach for most museums is to hire an established interactive computer production firm. Companies like NETG have the experience to guide your project toward meeting your goals. We have design expertise that only comes from extensive practice of our craft. We will be around for years to come. We can provide the stability and support that contract programmers lack.

Interactive production companies, however, are not perfect. Our software may have bugs that are tough to sort out. It may be expensive to have us make what seem like minor changes to your program. We may be hesitant to give you our source code (allowing you to make changes), as it represents our second most valuable capital asset (after our staff). Again, check references. Make sure the company you are considering hiring is really listening to your concerns rather than trying to force one of their handy solutions down your throat. Consider whether they are committed to the exhibit business, or are just trying to fill in a slow period with your project.

I can suggest a few general principles that you can apply regardless of who creates your software. Plan. Are you going to want to update the exhibit on a regular basis, or is the information stable? Before designing specifics, establish your goals for the exhibit (it's surprising how often our customers have not clearly defined their goals). With a simple goal statement in hand you can change the program design as you learn about unforeseen constraints in the software or hardware.

The final principle of software reliability is to make the interface work for your visitors. A poor interface will be perceived as a broken program. It will invite your visitors to bang on your touchscreens, to try to wrench your joysticks out of the countertop. And it will not achieve your goals. There are many books written on the topic of what makes a good interface. Be aware that a museum has its own constraints which many of these books do not deal with. You have to keep traffic moving, so you may not want people to spend more than 3-5 minutes at a one-on-one exhibit. Visitors have no training on the interface, so it must be immediately understandable. While involved hypertext systems can be great for exploring information, they may not be the best way to achieve simple interpretive goals. Five seconds of frustration is enough to ruin an exhibit for most visitors.

One minor note: you will save yourself frustration if you ensure that you can easily get at the floppy disc drive in your exhibit's computer.

Locating the Hardware

Each device has environmental constraints (temperature, humidity, dust control, etc.). Take these constraints into account very early in the planning process, as they effect fabrication, electrical work, air conditioning, and the general use of space. You can usually learn the specific requirements of each device through product literature or from phone conversations with the manufacturer. What most of the requirements tend to boil down to, however, is that interactive computer systems work well in a range of temperatures and humidity that is comfortable to people. Remember, however, that the equipment itself is a source of heat, and tends to be located in very tight spaces. Therefore, the spaces that house the equipment must usually be ventilated or cooled.

When ventilating a cabinet, follow nature's rules by blowing the cool air in the bottom, allowing the hot air to escape out the top. Allow adequate air space around each piece of equipment, especially around any ventilation holes.

If your space is at all dusty, the dust will build up inside the equipment, and will eventually cause problems that require your attention. If your equipment is inside a cabinet, you can slow down the accumulation of dust by filtering the air that you blow in to the kiosk. By putting the fan at the air entrance rather than the air exit, you will also be blowing out air through all those tiny cracks in the cabinet. If you put the fan at the air exit, you will be sucking in filtered air through the entrance, but will also be getting dusty air through the cracks.

It is important that the equipment have a "clean" source of power. Significant power fluctuations can cause very strange intermittent problems that will force you to chase in many wrong directions. Do not use a circuit that is shared by any device that occasionally draws heavy loads of electricity, such as elevators, heaters, air conditioners, or Xerox machines.

Also take care not to locate either your equipment or any cable runs near electrical devices that are sources of electromagnetic disturbance. Watch for things like heaters, transformers, pumps, and so on. Some of our most novel sources of trouble included a Van De Graff generator and a pair of subway lines that crossed below the corner of a steel frame building. It can be very tough discover the source of these sorts of problems, and it can be equally difficult to relocate the equipment once the source is found. You should try to anticipate problems during the planning stage of a project.

Some museums have located all of an exhibit's hardware right by the public interface (monitor, leds, speakers, buttons, etc.), while others place as much as possible in a remote location such as an equipment closet. Though I have heard many justifications for the idea of locating equipment in a remote area, the only one that I give much credence to is for the sake of space. Some exhibit designs don't allow for placing bulky computers and videodisc players near the public interface.

Putting the non-interface portion of an exhibit in a remote area can cause some problems. Running lengths of cable can get expensive. You may bring in some environmental problems due to the cable run. If you need to troubleshoot a problem, you practically need two people with walkie talkies, one on the exhibit floor, the other by the remote equipment. Though you can monitor the apparent status of many exhibits from one remote room (if you have status monitors and the software will give you some sense of status), a good status reading may be deceptive. There is no way other than playing with the interface to be sure that there are no problems.

The Importance of Planning for Maintenance

From time to time we hear from a customer who has had a broken exhibit for months, and has done nothing about it. More often than not we are able to give them advice over the phone that allows them to correct a minor problem. Until the call, their visitors saw only a black screen. Though they were eager to resolve their problem for months, other pressing matters took up their time. No one person was really responsible, or the responsible person also had to write a few grant proposals, stage a temporary exhibit, and collect samples in the field.

No matter how reliable a system is, it may go down at some point. If no one has the job of dealing with problems (and adequate time available), the system will probably stay down for a very long time. The larger an investment a museum makes in interactive exhibits, the more important this issue becomes. Someone at the museum needs to own the role of dealing with the interactive computer exhibits. This "AV tech" should perform some very quick and infrequent maintenance on each system. Hardware suppliers should provide maintenance instructions.

It is tempting for museums to believe that interactive computer exhibits will simply run themselves for years without any troubles. They ask for an opinion from a producer like us, or hardware manufacturers, and they get assurances of the reliability of the equipment. We assure them that many systems do run for years without any trouble, and most problems that occur are easy to resolve. We're optimistic. Having a limited budget, and quite a bit of excitement about the various exhibits that they are in the midst of creating, the museum catches the optimism. They make no plans for maintenance of the exhibits other than assigning the AV/Tech job to some unfortunate individual who is already neck-deep in work.

In the case of a museum that is getting only one or two interactive exhibits, this approach is probably the best use of limited resources. On the other hand, when a museum is under-going a major expansion they need to plan for an increase in maintenance effort. How much depends very much on the nature of the exhibits that are included in the expansion. During the design stage, the plans inevitably are shrunk to fit the budget. This is a painful process. The temptation can be to put all the available money into the exhibits rather than setting aside some funding for ongoing maintenance. I would rather, however, see one less exhibit so that the remaining exhibits will continue to work for years to come.

Due to the pace of change, a museum may find that a few years after installing a system it can no longer purchase a replacement for a system element. The system can be designed to minimize this problem. Monitors have been a particular problem. With new monitors phased out after less than twelve months in many cases, kiosks should be designed to accommodate a potentially larger replacement monitor.

Design of the Program

A frustrating visitor interface may leave visitors thinking that the program is "broken." Slow response time may encourage more aggressive use of the input device. Constantly accessing one sector of a floppy disk can cause problems. In short, poor program design can contribute to downtime. Though there are a few guidelines, the best way to develop a good design is to involve the right parties in the design process, and to conduct formative evaluations.

Formative evaluations conducted on poor designs will give few clues toward the development of an excellent program. In order to get a design off to a good start, take advantage of the experience of people who have already created many programs. Include them on your design team at an early stage in the process.

Look at as many interactive computer exhibits as you can find. Play with the exhibit for a while, then step back and watch the visitors. As museum professionals, we tend to find even poorly presented information to be interesting. Visitors will show you problems you never expected. Take note of what you see, but resist the urge to create a set of rules or a formula. Each program sets unique demands on the designer.

Burn-in, or The Honeymoon's Over

When we are finished with the production work on an exhibit, we like to put it on a rack for a week or so to let it run. We encourage each other and our visitors to play with it. We call this "burning-in" the exhibit. After a week on the rack, we feel comfortable shipping the exhibit to the museum for installation. We often find a few subtle problems that are easy to resolve in our shop, but tough to resolve once we've installed the exhibit and gone home.

Despite our care, problems may still slip by. Conditions unique to the museum may cause new problems to crop up. Whether intended or not, the museum undergoes a second burn-in period. Though it is our responsibility to deal with the problems, the museum is responsible for noticing them and describing them to us. Unless we charge an exorbitant amount for an on-site service contract, they are also responsible for either running on-site tests under our direction or packing the equipment up to ship back to us.

This on-site burn-in period will put a greater strain on the museum's resources than will general maintenance of the exhibits. Just as a museum should allocate resources for ongoing maintenance, they should plan for a burn-in period. The time period will vary depending on the amount and complexity of the exhibits, along with the experience of the people who are producing the exhibits. The best way to get a sense of the effort involved is to speak with other museums who have installed comparable exhibits.

The Recipe

People are of of lists and commandments, so I've summarized the preceding information with the following list, or "ten commandments of reliability" if you will:

1. Begin by setting your goals for the exhibit(s). Keep them simple and clear

2. Generate a design specification. Involve an experienced interactive designer.

3. Analyze the reliability of the specified hardware elements.

4. Analyze the "user friendliness" of the design.

5. Coordinate the design with the overall space, including the enclosures for the hardware and any required conduits.

6. Choose a reliable software solution.

7. Plan for ongoing maintenance.

8. Repeat steps 2-7 until you are satisfied.

9. Produce the exhibits.

10. Allow for a "burn-in" period.

Good luck!