Learning from games: seven principles of effective design

Authors: Rob Houser and Scott Deloach

Date: Aug. 1998
From: Technical Communication(Vol. 45, Issue 3)
Publisher: Society for Technical Communication
Document Type: Article
Length: 4,910 words

Abstract:

Seven principles of design that are usually found in game applications are presented. Technical communicators and designers of business applications can learn from these principles to create effective interface designs.

Full Text:

Why do players of computer games seem to approach those applications without fear, eagerly exploring and learning as they go, whereas users of business applications will go out of their way to keep from using the tools? Why do business applications require volumes of documentation when the most complex games come with a brief tutorial and a strategy guide for exploration? Why can games teach pilots to fly multi-million-dollar jets better than books and classroom training? These questions have led us to ask another question: Why can't business applications be more like games? In this article, we attempt to lay the ground work for future research by defining seven design principles found in games that we believe contribute to the creation of more usable applications.

INTRODUCTION: WHY STUDY GAMES?

When people play a game in an arcade or on their computers, they are transported to another world where they may get lost or encounter surprises. They usually find this exploration exciting. In contrast, when users get lost in a business application, they become frustrated (Carroll 1982). Games have a way of drawing users into them and keeping them engaged.

Perhaps we should not be surprised that the market for game applications is thriving. Its size and success rival any other product line in the world. Yet in the business environment, games - and business applications that look like games - are often viewed with suspicion. Traditionally, people have viewed fun activities such as games as a separate activity from work; however, activities we do for fun are not necessarily the opposite of work (Blanchard and Cheska 1985). In other words, work involving business applications does not have to be difficult to approach, confusing to learn, or frustrating to use.

In Things that make us smart, Donald Norman (1993) identifies seven basic requirements of a learning environment:

1. Provide a high intensity of interaction and feedback.

2, Have specific goals and established procedures.

3. Motivate.

4. Provide a continual feeling of challenge that is neither so difficult as to create a sense of hopelessness and frustration, nor so easy as to produce boredom.

5. Provide a sense of direct engagement, producing the feeling of directly experiencing the environment, directly working on the task.

6. Provide appropriate tools that fit the user and task so well that they aid and do not distract.

7. Avoid distractions and disruptions that intervene and destroy the subjective experience.

If we accept Norman's requirements for a learning environment, then we must acknowledge that learning is integrally related to games. In fact, games make learning look so much like fun that they mask the large amount of learning required to play them successfully. Casual observers of games may mistakenly see all games as less complex than business applications. However, on closer inspection they would discover that the difference is not the complexity of the tasks but the complexity of the design. Although game and business applications seem to be distant cousins on the surface, they share the same design goals: to be interactive, include guided discovery, be relevant and meaningful, and provide appropriate feedback (Hedberg and Harper 1996; Csikszentmihalyi 1990; Norman 1993).

In this first stage of our research, we demonstrate that Norman's seven principles of effective design are exemplified in game applications. The primary purpose of this paper is to get technical communicators thinking about design, perhaps in a new way, and to lay the groundwork for future research in this area. We believe that by analyzing the design of game applications, designers of business applications can gain insight into how to help their users learn while they complete their tasks. Even if you are not the actual designer of the application, you may be an important part of the design team, especially because of your experience helping users accomplish their tasks and teaching them how to use the product.

In the next stage of our research, we intend to determine how much each principle impacts the users' motivation, learning, and performance when using business applications. We also plan to establish practical "how to" guidance for applying these principles successfully. At this later stage, we will concentrate on the design of business applications. We will design and test prototypes that use the seven principles of effective design described here.

For this article, our primary focus is on how game applications are designed.

ATTRACT MODE

Principle #1: Users should be able to learn how to begin using the application quickly and with minimal effort.

Attract mode is the series of graphics (still images or video) that display on a game when it is not being played. Some games' attract modes start on a timed basis or run continuously like a screen saver; others' start when a potential player touches a control or moves in front of the game. The purpose of the attract mode is to get the attention of a potential user. However, several valuable instructional goals are accomplished. The attract mode demonstrates the basic controls required to get started and reveals strategies for achieving optimal performance. Attract mode also introduces the game's storyline, if one exists, which helps immerse the user in the scenario of the game.

Most casual observers of games have noticed that games attract attention through flashy images and sounds. What they may not know is that an arcade always looks and sounds exciting even if nobody is playing because of the attract mode. Because of its eye-catching nature, attract mode can be seen solely as an attention-getting gimmick; however, the attract mode also has an instructional value. Through watching the attract mode, users interact with a game even before they start to play.

Consequently, they also begin learning how to play before they ever consciously begin. Their risk is nonexistent and their effort minimal, although they have already started exploring the game application simply by watching the attract mode. Instructional designers have known the importance of observation to learning: "Observers learn by watching and imitating others; they tend to behave as they have seen others behave" (Mager 1984).

Figure 1 shows some of the cycling screens from an attract mode for the arcade version of Tetris. Potential Tetris players can walk up to a Tetris game and find our how to play without ever inserting a token. Through the attract mode, the user can learn how to use the basic controls (a joystick and button) and different playing strategies (for example, pulling the joystick to drop pieces) to achieve their goal (filling an entire row). Essentially, the attract mode for this game is a brief demo of the application given from the perspective of looking over another user's shoulder as he or she points out the basics of the game. Because users are invited to play by the application and are presented with the basic information required to start, they are more motivated to play.

Business applications rarely take advantage of the instructional value of an attract mode. Many recent Microsoft products provide an optional "Tip of the Day," which provides users with hints about available controls and strategies. To some extent, the "Tip of the Day" feature helps instruct users in a risk-free manner, but the tip is so random that it provides "hit and miss" instruction.

Another example of how business applications can use some aspect of an attract mode can be found in the programs that provide informative splash screens during the installation process. The timed images point out potential uses of the product (like new features) and some try to help the users get started. However, the majority of business applications do not proactively help the user get started or try to motivate users to use the application.

CLEARLY STATED GOALS

Principle #2: Users should always know what they are supposed to do to succeed.

Clearly stated goals are an essential part of effective instruction. "Learners need to know what they are trying to accomplish and how they are doing" (Zemke 1995). Games are extremely goal oriented. The goal is demonstrated in the attract mode, it is often stated on the game casing, it appears in the initial instructions (tutorial), and it is often repeated throughout the game. Goal identification is a basic tenet of instructional design: users need to know where they are going before they can get there (Cranton 1989).

In Figure 2, a screen shot of a tank battle game called Heavy Gear shows the goal is briefly stated at the top center of the screen (Go to Way Point WHISKY). The goals change throughout the game, but the users know when the goals change, and they are always aware of what goal they are working toward. Games also use audio reinforcement of goals such as a jet fighter game that begins a new round by announcing dramatically: "Shoot 9 enemies!" Another common technique games use to relay goals is to have an advisor explain the background of the game's story when the game starts.

Business applications always have inherent goals but do not always state them overtly. Business applications rarely indicate the goals possible (or best suited) to the user's situation because most of them are not "aware" of what the user wants to do or has tried to do so far. Most business applications leave the user at the main screen - full of possibilities but without direction. Tools are hidden behind menus and mysterious icons on toolbars. Goals are nowhere to be seen. Without "knowledge" of the user's goals or actions, an application cannot help the user achieve a specific goal.

Some examples of how business applications use clearly stated goals are the prompts given by tools such as Powerpoint or Robohelp when the tools are opened. The prompt, a form with radio buttons, suggests possible goals that the user may want to pursue. Unfortunately, even these applications seem to "lose their awareness" of the users' choice once the proper template is applied. To their credit, these applications also anticipate users who have different goals in mind (for example, to create blank document), allowing the user to bypass the predefined goals of the application to substitute new ones.

On the surface, it may seem that games can anticipate the users' goals more clearly than business applications because they are not as complex. However, complex business applications should be designed to help users achieve specific, definable goals, which should make the goals identifiable even if they are complex. If we cannot anticipate users' goals in our business applications, then we must investigate the usability of the applications. Anyone who has tried to play games like Simcity, where thousands of microsimulators are running to simulate real-world city management, would not underrate the complexity of games. Many games push logic and motor skills to the limits, but they still manage to focus on specific goals at any one time.

BRIEF INSTRUCTIONS

Principle #3: Users should be provided only enough instruction to get started and keep going.

Users approach an application to perform a task. They often have very little time or interest in learning about an application and want a quick and efficient way to identify and achieve their goals. If they read the instructional or reference information, they are reading to learn to do (Redish 1988; Sticht 1985). They are interested in reading only the bare minimum that is needed to achieve their goal. The user who is "reading to learn to do" is best supported by minimalist instructions, an introduction to basic concepts, and just-in-time instructions at the point of use (Carroll 1990). Minimalism does not simply mean that instructions are brief; rather, it means that instructions are integrated closely with the application through extensive usability testing to keep the users focused on their tasks, not on the instructions.

Although most game applications have many features, they do not try to tell the user about them all at one time. Instead, games explain the basic goals and techniques on the game itself (that is, on the console, cartridge, CD, or diskette) and in the attract mode. Games then build on this information with brief instructions that appear at relevant moments throughout the game. By presenting the instructions at the point of use, games allow for a gradual and timely assimilation of information. When game applications do provide more information at once such as through a book, the information is usually about developing successful strategies for optimal performance rather than following step-by-step instructions.

Figure 3 shows the brief instructions for a puzzle game called Marble drop. This introductory screen introduces the interface elements of the game (vertical diverters, warning lights, and the exit tube), provides an explanation of the game's feedback, and explains the goal of the game ("Match colors to solve puzzle."). Armed with this information, the user is ready to begin playing the game.

Business applications provide the user with information about each feature, but they make the user look for help in online help files and print documents. Because the instructional information is separated from the interface, users must leave the application and interrupt their tasks. Users have to find the necessary information, remember it, and apply it to their specific situations.

If used effectively, cue card and context-sensitive help can provide effective brief instructions for business applications. The challenge, however, is to provide the right amount of information to make the information useful and not too overwhelming with detail.

TRANSPARENCY

Principle #4: Users should not be distracted physically or psychologically from the task at hand.

Successful learning occurs when learners are able to suspend disbelief and interact with an application on a physical and psychological level without distraction (Laurel 1991; Low 1994). Ideally, the interface should be transparent, allowing the users to focus completely on accomplishing their goal (Heidegger 1986). Games create a transparent interface by emphasizing the story and metaphor, and de-emphasizing game options. The options are either removed from the main interface and presented on an easily accessible options screen, or a "smart" toolbar is used, and unavailable or irrelevant options are disabled.

Figure 4 represents a screenshot from Rockett's new school, a role-playing game for pre-teen girls. In the game, the options are represented by and accessed through Rockett's backpack and locker. For example, when the player needs to use Rockett's camera, she opens her backpack and takes it out. Because all the options are not constantly presented on the screen, the game allows players to stay focused on their task (Norman 1993; Bodker 1990). In the case of Rockett's new school, the player is able to focus on exploring the relationship with the game's characters rather than manipulating a distracting interface that shows all possible (grayed-out) options. The player can also benefit by seeing a task as a whole, rather than as a series of seemingly unrelated steps (Betz 1995; Zemke 1994).

In a typical business application, most of the features are always accessible, although many can lead to errors or user confusion. In attempting to increase user freedom, business applications often become more confusing and discouraging to use. However, a good example of transparency in a business application is Kai's Photosoap, a graphics application that allows the user to touch-up and add effects to digital photographs. Photosoap uses a "photo lab" metaphor, where different rooms are used to group similar tasks. For example, in the "Touch-up" room, the user can select the "Red eye" command to remove red eye reflections. Only photo touch-up actions can be performed in the touch-up room, so the user is not overwhelmed with an overabundance of choices. Photosoap even uses realistic erasers, brushes, and pencils as tools to make using the program as transparent as possible.

PERFORMANCE COACHING

Principle #5: Users should receive helpful suggestions related to the task they are performing, which they can adopt or ignore.

Learning requires more than access to information. It requires that the information be presented at "the right time and the right place." It requires some kind of guidance that the learners cannot always provide themselves. Information or motivation provided too early or too late or even all at once may not result in the most effective learning. In fact, undirected learning could result in bad habits or the creation of improper mental models that must be unlearned before the desired performance can be achieved.

Traditionally, formal guidance for learners comes from teachers. Informally, it comes from asking the person in the next office for help or from observing the behavior of another user. An effective guide

* Helps the learner identify what tasks need to be accomplished

* Assists the learner in locating the type, level, and amount of information required

* Encourages the learner, providing hints and suggestions when necessary

* Ensures that the learner adopts the optimal performance behavior

Ideally, the guide would be aware of the user's actions as well as previous user behavior so it could respond in context whenever the user needs help. Guides need not be animated or real characters. However, the act of guidance must be unobtrusive, timely, and appropriate.

Instructional designers call this type of guidance "coaching" because the system focuses on aiding optimal performance rather than just providing information (Raybould 1995). The coach is a vital component of an electronic performance support system (EPSS), "an electronic system that provides integrated, on-demand access to information, advice, learning experiences, and tools to enable a high level of job performance with a minimum of support from other people" (Leighton 1996).

EPSSs can take many forms, but they include aids such as wizards and cue cards as well as just-in-time computer-based training. The coach takes a proactive approach to help the user do "whatever is necessary to generate performance and learning at the moment of need" (Gery 1991).

One study showed that learner control with coaching is more effective than total learner control or browsing (Hannafin 1992). Another study showed that although learners may complete a task faster with learner control, they do not always have superior retention or recall (Murphy and Davidson 1991). Other studies have shown that coaching seems to be most effective when the system makes suggestions that the user can ignore (Wynn 1996; Ross and Morrison 1988). In other words, a coach should help the user set goals and ensure that those goals are achieved effectively; however, the coach should not restrict the user's interaction with the application, remove the user from the application, or override the user's ability to deviate from the norm.

Figure 5 shows an example of performance coaching in a game called Sid Meyer's Gettysburg, a strategy game in which the user fights Civil War battles against the computer. Strategy games are often complex simulations of real-world situations, requiring the user take into account multiple variables (terrain, equipment, weather, number of troops, even morale). Because such games are so complex, users sometimes have difficulty learning optimal strategies quickly and determining what effects their actions have on the goal of the game. Many games such as Gettysburg evaluate the users' actions and provide feedback to reinforce desirable behavior.

Other performance coaching is designed for error recovery. In NBA jam, a basketball game, the computer provides coaching to keep the action going. If a user does not use certain controls when appropriate, the coach displays a brief message explaining the purpose of the control and the best time to use it. As soon as the user moves the controls again, the message disappears and the game continues. In Figure 6, the user has moved off of the screen and is no longer "in the game." The coach ensures that the user achieves some success in the game.

An excellent example of effective performance coaching in a business application is Microsoft Publisher. The opening screen for Publisher offers numerous wizards for creating the types of communications users most commonly create with Publisher. Publishers wizards not only create a default document for the users, but they show and explain to the users what is happening as the application creates the defaults. Users can control how fast the explanations display, and expert users can bypass the instruction altogether. Optional cue card help is integrated into the main window and can be used for step-by-step guidance. As users encounter new screens, Publisher displays bubble help that points out special features that may be helpful to them. If users drag an object so it straddles a margin, Publisher alerts them that the portion of the object outside the margin will not print.

Publisher illustrates several important points about performance coaching:

1. The advice must relate to the user's goals and specific context.

2. The user must be able to easily dismiss or turn off the advice.

3. The advice must be integrated with the application.

TRAINING WHEELS

Principle #6: Users should be immediately successful with the application and be gradually introduced to its more complex aspects through staged interfaces.

John Carroll's work with "training wheels" interfaces has shown that new users are more successful and are more likely to learn from their mistakes in a simplified environment (Carroll 1990). In fact, Carroll's training wheels research led him to conclude that "incorporating such training wheels into computer systems and applications would produce staged interfaces in which the full complexity of a system could be gradually revealed to users over a course of time and individual experience" (Carroll 1990).

This description of staged interfaces perfectly describes standard game designs. Games typically begin with simplified levels to allow players to learn how to perform basic tasks. These simplified levels enable the player to be immediately successful, a fact that increases their confidence and encourages them to continue playing. Many games also provide a training mode that disables more advanced, complex features until the user masters the basic tasks. Figure 7 shows a screenshot from a flight simulation game called G-LOC. In G-LOC, the player can select from beginner, intermediate, and expert levels. The beginner level provides a simplified, almost literal training wheels environment, where the user does not have to control the throttle or thrust of the plane.

In contrast to G-LOC's staged training wheels environment, business applications often provide the same interface for both new and expert users. New users quickly become lost in a confusing and overwhelming environment without guidance. Without initial success, the new user quickly becomes frustrated and unmotivated.

A good example of training wheels in a business application is the Autocontent wizard in Powerpoint. This wizard uses multiple choice questions to format and outline the user's presentation. When users complete the wizard, they can change the sample text to create a simple but very usable presentation. The simplified environment provided by the Autocontent wizard encourages learning and exploration by allowing the user to be immediately successful.

CONSISTENT FEEDBACK

Principle #7: Users should be aware of how effectively their actions are helping them achieve their goal at all times.

Consistent feedback consists of status information that is always presented on the screen. The most prevalent example of consistent feedback in a game is the player's score. By learning which actions generate the most points, players can earn a high score and achieve their goal. In Figure 8, Manx, a racing game, illustrates multiple modes of dynamic feedback. Users can monitor feedback about the elapsed time, their position relative to other cars in the race, and their location on the track. This feedback helps players assess how well they are progressing toward their goal. Consistent and continual feedback motivates the user to learn and explore new features (Cooper 1995).

Business applications rarely offer any assessment feedback. They do not help the user get started, introduce the application, offer suggestions, or provide any feedback. Instead, they provide cryptic error messages that usually require users to leave the task at hand to solve a problem. Users of business applications have no indication that they are not moving toward their goal until they either confront an error message or recognize that the application is not doing what they expect it to do (Norman 1988). The problem is that business applications are unaware of users' goals. Therefore, users must determine whether their performance is optimal or even correct based on a small amount of feedback and an unclear mental model of the system.

Business applications must become more proactive, keeping track of users' tasks and goals, and offering consistent feedback about how well they are reaching those goals. For example, an application could tell you the time elapsed since your last save, your current projects' position relative to today's list of "to do" items, and the amount of time remaining before your next meeting.

CONCLUSION

If you're still not convinced that game design is a valid basis for the study of effective interface design, go to your local arcade and watch. Observe how subtly the games present instruction to the users. Try to determine where the game stops and the instruction begins. Maybe even try a few of the games mentioned in this paper yourself to see how they are designed.

Although we have mentioned the fun aspect of games in this paper, we are not advocating that all business applications employ the highly fantastical imagery of games nor that they should all attempt to be fun or challenging. Some users and tasks may not benefit from those aspects of game design. However, all business applications can follow the seven principles of effective design detailed in this paper, even those for complex tasks or "serious" business, because all business applications must be concerned to some extent with learning.

In 1990, Theodor Holm Nelson wrote, "To see tomorrow's computer systems, go to the video game parlors! Go to the military flight simulators! Look there to see true responsiveness, true interaction. Compare these with the dreary, pedestrian office software we see everywhere, the heavy manuals and Help Screens and Telephone Support" (Nelson 1990).

We have come a long way in the 8 years since Nelson's statement, but games still set the direction for effective interface design. We are encouraged by business applications that are attempting to create effective learning environments such as the ones mentioned in this article. To be sure, there are other good examples of well-designed business applications. However, there are many more poorly designed business applications that continue to create ineffective learning environments.

We started our research into the design of game applications by asking: "Why can't business applications be more like games?" Our answer is, "they can."

REFERENCES

Betz, Joseph A. 1995. "Computer games: Increase learning and thinking in an interactive multidisciplinary environment." Proceedings of the Conference on Instructional Technology, pp. 14-17.

Blanchard, K., and A. Cheska. 1985. The anthropology of sport.' An introduction. Cambridge, MA: Bergin & Garvey Publisher, Inc.

Bodker, Susan. 1990. Through the interface.' A human activity approach to user interface design. Hillsdale, NJ: Lawrence Erlbaum.

Carroll, J. M. 1982.: "The adventure of getting to know a computer." IEEE computer 15, no. 11:49-58.

Carroll, J. M. 1990. The Numberg funnel: Designing minimalist instruction for practical computer skill. Cambridge, MA: MIT Press.

Cooper, Alan. 1995. About face: The essentials of user interface design. Foster City, CA: IDG Books Worldwide.

Cranton, Patricia A. 1989. Planning instruction for adult learners. Toronto, ON: Wall and Emerson.

Csikszentmihalyi, Mihaly. 1990. Flow: The psychology of optimal experience. New York, NY: HarperPerennial.

Gery, G. 1991. Electronic performance support systems: How and why to remake the workplace through the strategic application of technology. Boston, MA: Weingarten Publications.

Hannafin, M. J. 1992. "Emerging technologies, ISD, and learning environments: Critical perspectives." Educational technology research and development 40, no. 1:49-63.

Hedberg, J., and B. Harper. 1996. "interactive educational technologies: Effective design and application in the classroom." 3rd International Interactive Multimedia Symposium, pp. 160-168.

Heidegger, Martin, 1986. Being and time. San Francisco, CA: Harper Collins.

Laurel, Brenda. 1991. Computer as theatre: A dramatic theory of interactive experience. Reading, MA: Addison-Wesley.

Leighton, Chet. 1997. "What is an EPSS?" http://itech1.coe.uga.edu/EPSS/Whatis.html.

Low, M., S. Venkataraman, and V. Srivatsan. 1994. "Developing an entrepreneurship game for teaching and research." Simulation & gaming 25, no. 3:383-401.

Mager, Robert F. 1984. Developing attitude toward learning. Belmont, CA: Lake Publishing.

Murphy, M. A., and G. V. Davidson. 1991. "Computer-based adaptive instruction: Effects of learner control on concept learning," Journal of computer-based instruction 18, no. 2:5156.

Nelson, Theodor Holm. 1990. "The right way to think about software design." In Brenda Laurel, ed., The art of computer-human interface design. Reading, MA: Addison-Wesley, pp. 235-244.

Norman, Donald A. 1988. Psychology of everyday things. New York: Basic Books.

Norman, Donald A. 1993. Things that make us smart: Defending human attributes in the age of the machine. New York, NY: Addison-Wesley.

Raybould, B. "Solving human performance problems with computers," Performance improvement quarterly 8, no. 1:31-46.

Redish, Janice C. 1988. "Reading to learn to do," The technical writing teacher 15, no. 3:223-233.

Ross, S. M., and G. R. Morrison. 1988. "Adapting instruction to learner performance and background variables," In D. H. Jonassen, ed., Instructional designs for microcomputer courseware. Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 227-245.

Sticht, T. G. 19xx. "Understanding reader's and their uses of text," In T. M. Duffy and R. Waller eds., Designing usable texts. Orlando, FL: Academic Press, pp. 315-340.

Wynn, Sue. 1997. "Interactive multimedia: Ensuring motivation of the learner." http://www.bs.ac.cowan.edu.au/misforum/swynn.html.

Zemke, Ron, and Susan Zemke. 1995. "Adult learning: What do we know for sure?" Training (June):31-40.

Source Citation (MLA 9th Edition)
Houser, Rob, and Scott Deloach. "Learning from games: seven principles of effective design." *Technical Communication*, vol. 45, no. 3, Aug. 1998, pp. 319+. *Gale Academic OneFile*,

link.gale.com/apps/doc/A21148250/AONE?u=canyonuniv&sid=bookmark-AONE&xid=559a5e76. Accessed 16 June 2022.

Gale Document Number: GALE|A21148250