Distributed information resources and embodied cognition in software application training: Interaction patterns in online environments and digital games

Juan Carlos Sánchez Lozano

A Thesis

In the Department

of

Education

Presented in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy at

Concordia University

Montreal, Quebec, Canada

April 2010

Your file Votre référence
ISBN: 978-0-494-67335-5
Our file Notre référence
ISBN: 978-0-494-67335-5

ABSTRACT

Distributed information resources and embodied cognition in software application training: Interaction patterns in online environments and digital games

Juan Carlos Sánchez Lozano, Ph.D.
Concordia University, 2010

Research on software application training has been predominantly based on the premise that the user builds an internal functional model of the system. This view of cognition as purely internal has been challenged by studies showing that experts do not remember the exact commands required to complete a task if they are not in front of the computer. Display-based competence suggests that a flow of information takes place between the display and the user, who only holds the necessary information to understand the visual cues and act accordingly.

Distributed cognition, however, does not address the perceptual and motor actions necessary to acquire the information from the world and act on it. Embodied cognition attends to this gap. The Soft Constraints Hypothesis suggests that at the 'embodiment level', users will choose between cognitive, perceptual, and motor operators based on a cost-benefit analysis, the cost of the resources measured in milliseconds.

It was hypothesized that resources in different environments with various access costs would influence use patterns, and learning strategies and performance. For the purpose of this study, participants completed a 5-lesson course in Flash animation concepts. 50 participants were then assigned to two practice environments that contained a software simulator built based on visual cues and salient task features displayed by the real software. The two practice spaces manipulated the nature of, and access to, information resources, with one featuring game-like task completion. Every action taken during the study was tracked and time-stamped, producing a log file containing around 20,000 records for subsequent analysis.

The results showed that information access cost in online environments for software training has a clear impact on the strategies employed, the learning processes engaged, and learning outcomes. Traditional statistical analysis showed significant differences in declarative knowledge of rules, efficiency, and accuracy between groups. These results were complemented with data mining techniques to analyze user sequences as departures from an optimum path. The idea of a dual-learning process taking place when users are learning interactive behaviors, for example command sequences, was supported. A declarative-to-procedural process takes place in low-cost single-task environments whereas in the specific case of educational games, information access cost and concurrent game tasks can trigger a non-attentional cue-based behavior that results in higher task efficiency and accuracy, but has a negative impact on rule verbalization. A design framework for instructional environments mapping instruction-specific resources and their access cost to specific learning processes and outcomes is presented based on findings.

ACKNOWLEDGEMENTS

The path to get here has not been only intellectual, but deeply personal. Throughout it all my parents have been there for me, giving me their support and having faith in me, even when things didn't look very promising. This work is theirs as much as mine. Words cannot describe how much I love them and how grateful I am for everything they have done for me.

I would like to specially thank my supervisor, Dr. Richard Schmid for his guidance and patience during this process. I have thoroughly enjoyed our 'power meetings'. I was always motivated to improve my work and take it further. The results of this thesis are interesting because of the many stimulating and challenging ideas he proposed during our discussions.

My supervisory committee, Dr. Gary Boyd and Dr. Saul Carliner, provided me with great insights and feedback, making this a better research study. I am very grateful for their continuous support.

I would also like to thank Mr. Stef Rucco and Mr. Peter Blyszczak for always accommodating my late-minute requests for computers and lab time, making sure that I always had all the necessary equipment at my disposal.

I cannot forget to thank my cousin Patricia, who not only listened to me whenever I got into trouble but also made me laugh and helped me to be irresponsible from time to time, with wonderful results.

I would have been very happy to share this work with my dear friend Shulamit Lecthman. I hope she knows that completing my doctoral work would not have been possible without her wisdom and help, which I sorely miss.

Finally, a big thank you to my friends for putting up with my worries, for giving me unconditional support, and for dragging me out to relax and laugh.

I hope this is the beginning of a life dedicated to contributing in meaningful ways to those who might benefit from the few things I have learned so far. May God help me reach out to them and do my job well, whatever and wherever it may be.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

The purpose of this research was to determine if different patterns of use emerged in online environments with equivalent tasks and resources but differing contexts and information access costs. This section presents an overview of the rationale behind the study by first introducing the problem and then discussing how different views on cognition address the issue, and how they apply to digital games. The chapter concludes with the research questions and a short summary.

Problem Statement

Online environments have a variety of resources on the screen that provide information to accomplish a certain task. For example, web pages offer content, graphics, and links that will lead to other resources on a given topic. Software applications have a variety of panels, commands, and menus that help the user create a document, design a logo, modify a picture, etc. Game interfaces show the player a set of resources, score, time available, among others, which have to be taken into account to achieve the game's goal. With the appearance and popularization of new computers and mobile devices, new forms of interacting with these systems are emerging. Whereas before the mouse would be the device used to interact with the screen, now devices such as the I-Phone have popularized the manipulation of items on the screen with one's finger. Traditionally, graphic design principles have dictated the location of resources on the screen and the domain of human-computer interaction (HCI) has been concerned with the usability of these systems.

Why is this relevant from the point of view of education? If the purpose of the system is to teach the user a certain skill, then an important issue is to determine how the resources on the screen are used and if they play a role in how people learn from these online systems. This is particularly relevant in e-learning and digital game design where display resources are abundant. What resources are more used to complete a certain task? Are they used in the same way regardless of other characteristics of the environment? Do these resources shape in any way the learning process?

This research study attempted to answer some of these questions in the specific context of software training. Learning to use a new software application can be a frustrating experience for novices. Apart from feelings of confusion and trying to figure out which of the many commands will actually allow them to achieve a specific goal, they must learn essential processes before they can actually do something meaningful with the program. Although these basic procedures are extremely important, using drills is not a frequently used alternative to practice because of the novice's need for meaningful action. As a result, elementary operations are learned as a by-product of worked examples or simplified authentic tasks. It has been argued that novices frequently experience increased cognitive load in discovery or exploratory environments because of their weak knowledge of foundation concepts, thus reducing the effectiveness of this type of training (Kirschner, Sweller, & Clark, 2006).

Consequently, this research focused on practice environments designed to learn fundamental operations of software applications that are the building blocks of more complex procedures, and that the learner needs to master to complete authentic tasks in a professional environment. A simplified replica of a real software application interface was used in order to simulate as close as possible the interaction with the real program. The application learned was Adobe Flash ™, a software application widely used to create animations for the web and Rich Internet Applications (RIAs). It was initially believed that because Flash is such a popular application it would be easier to find participants for the study because interest in learning the application would be high.

Cognition Views in Software Training

What do we know about the interaction that takes place between the user and a software application? Formal models of software users suggest that the learner's goal is to internally encode procedures to reach an objective (Howes & Payne, 1990). In other words, working with software is considered to be a purely cognitive activity, the internal functional model of the user entirely guiding the series of commands that will lead to a specific goal.

This knowledge-in-the-head view has been challenged by research showing that experts do not seem to have internal functional representations of the software application but rely heavily on display cues to take appropriate action (Mayes, Draper, McGregor, & Oatley, 1988; Payne, 1991). Rather than storing the program commands and procedures in long term memory, experts seem to store enough information to decode the cues provided by the display in order to select appropriate commands (Kitajima & Polson, 1995). A perceptual cycle takes place between the display and the user. As the display presents information to the user, he/she decodes the relevant cues and internally determines the next step. Once the user takes action, the display is updated and new information is made available, starting the cycle again.

Display cues can provide a variety of information such as a label, an icon, tool tips, etc. Users appear to pay particular attention to the location of cues on the screen, and use this information to improve performance (Ehret, 1999). It is important to note that the capacity to decode this visual information is achieved through extensive practice, which justifies the goal of this study to achieve automaticity of fundamental operations.

If information is distributed and visual cues have to be decoded, it has been suggested that certain information resources can be made available to the user in order to reduce cognitive load and minimize errors (Hutchins, 1995). The Distributed Information Resources Model (Wright, Fields, & Harrison, 2000) provides a general non-exhaustive list of resources and interaction strategies with systems were information is distributed. Some of these resources were included in the practice environments used in this research. The objective was to examine what kind of use patterns emerged in the different instructional spaces. The analysis of these patterns based on the Distributed Information Resources Model only would have been insufficient because the model says nothing about access procedures. We look at the concept of embodied cognition next to extend this analysis framework.

An important common assumption in display-based interaction is that the cycle between display and individual involves mainly cognitive processing of visual cues, and considers perceptual and motor skills as a mere input/output mechanism. However, recent studies show that physical aspects of the interface and the motor skills required will have an impact on the type of strategy, cognitive or perceptual-motor, that individuals will use (Gray & Boehm-Davis, 2000; Gray & Sabnani, 1994; Gray, Sims, Fu, & Schoelles, 2006). In other words, cognition is not only distributed between the individual and the world, but is also influenced by the interaction of the body with the physical characteristics of the environment. Users of interactive environments assess the cost of accessing information from memory or the environment and decide which strategy to use depending on the retrieval cost, measured in milliseconds (Gray & Boehm-Davis, 2000). Individuals will even choose to rely on imperfect information if the cost of other strategies is higher (Gray & Fu, 2004). These findings have significant relevance when designing interactive environments.

Of particular importance for instructional interventions is that even though individuals in an interactive space tend to favor low cost strategies, performance will suffer as a result (Gray & Fu, 2004; Waldron, Patrick, Morgan, & King, 2007). The same studies showed that a cognitive approach was more effective than a perceptual-motor approach in spite of the higher cost. These findings influence the choice of resources that should be made available to the learner as well as the cost of accessing them. In consequence, another objective of this research was to confirm previous studies on embodied cognition and show that individuals using interactive environments choose between cognitive and perceptual-motor resources based on a cost-benefit analysis.

Embodied cognition is not a clear-cut construct. As Clark (Clark, 1999) mentions, research on embodied cognition ranges between simple and radical embodiment:

> "In addition to asking how far the embodied approach can go, we should also ask to what extent it is a genuinely radical alternative to more traditional views. To focus this concern, I would like to distinguish two different ways to appeal to facts about embodiment and environmental embedding. The first, which I will call 'simple embodiment', treats such facts as, primarily, *constraints upon a theory of inner organization and processing*. The second, which I will call 'radical embodiment' goes much further and treats such facts as *profoundly altering the subject matter and theoretical framework of cognitive science*. The distinction between the simple and radical forms is, however, not absolute, and many (perhaps most) good research programs end up containing elements of both." (p. 348)

This research adhered to the simple embodiment view where the constraints imposed by the physical environment determine the kind of resources selected by the user, optimizing computational efficiency.

Digital Games

This apparent emphasis on efficiency can also be seen in a very popular type of interactive environment: video games. Tetris is a well-known game that requires the player to rotate shapes on the screen so that they fall in a specific manner and fit the already existing blocks at the bottom. Maglio and Kirsch (Maglio & Kirsh, 1996) have shown that players use 'epistemic actions' or apparently redundant "external actions which simplify an agent's computation" (p.391). Rotating the pieces on the screen several times with the arrow keys allows the user to retrieve information faster and with less effort than if the same rotations were performed mentally. Thus perceptual-motor strategies are preferred in this case to cognitive ones because of computational efficiency and reduced cognitive load, and also because of the physical characteristics of the environment as well as the dynamics of the game. This agrees with Norman's (1983) observation that mental models are parsimonious.

It is possible to take this idea further and argue then that video games are virtual spaces with defined characteristics that create sets of constraints on the player, influencing higher-level strategies. These characteristics can include the physical ones such as context and the type of control used –a topic that will be discussed in more detail later-, as well as the resources available among others. As such, video games can be seen as environments where cognition is both distributed and embodied.

Is it possible to exploit the strengths of games and use them in computer-based software training? If so, will computational efficiency determine the strategies favored by players, as in any other interactive environment, or will there be a different pattern of use? How will the player handle the resources as compared to a purely instructional space? And will there be any difference in performance? This study also addressed these issues.

Research Questions

This research's purpose was to extend the previous analysis of simple interaction tasks to interaction with a software application interface while learning basic procedures that require multiple steps to be completed and that should become automatic. It specifically considered information resources available on the screen as well as the cost of accessing them. As a further contribution, the study also included other variables that have not been previously considered such as motivational beliefs, preferred learning strategy, personality type, and computer self-efficacy, as well as several performance measures.

The most important contribution, in the researcher's opinion, was the extension of the analysis to a game environment that encapsulated an identical instructional environment used in the other conditions. This addressed a common criticism that instructional games are usually compared to non-equivalent alternatives. Also, although measures in the order of milliseconds were taken in all conditions, this level of granularity at the embodiment level has not been incorporated in previous instructional game research as far as the author knows.

The main general research questions addressed here were:

1. Does the access cost of resources in an online environment impact learning and performance?

2. Do different patterns of use emerge depending on the characteristics of the online environment? In particular, does the process differ if the user is asked to complete exercises as simple drills as opposed to completing them within a game?

3. Are individual characteristics in any way related to differences in the learning process, performance, how the user interacts with the online environment?

Summary

Cognition can be seen as knowledge-in-the-head as well as knowledge-in-the-world. Additional information about the task that the individual has to carry out in the application software and that is not usually available can be added to the instructional interface as distributed information resources. Cognitive strategies such as retrieving information from memory will compete with obtaining information from the resources present on the display using perceptual-motor skills. Which strategy will prevail will depend on the cost, as measured by time required to obtain the relevant information in milliseconds. Depending on the physical characteristics of the interface, that is the resources present, their availability and the control devices, different patterns of use may emerge.

In this study it was expected that diverse display configurations would result in different strategy patterns and times to complete operations and tasks. Each would have a cognitive load associated with it, and a resulting level of performance. Video games were seen as special cases of interactive environments and one of the objectives was to determine whether the patterns of use differed due to the context.

The next section summarizes research on software training methods as well as views of cognition in interactive environments, which will serve as the basis for the design of the environments used here.

CHAPTER 2. LITERATURE REVIEW

Introduction: Computer use, facing the challenge

Computers have become ubiquitous in society and have impacted and shaped almost every aspect of our lives. From new forms of communication to the way we work, and the manner we spend our free time, computers have consolidated possibilities that were only imagined in the sixties when Vannebar Bush conceived the memex (Bush, 2003), Doug Engelbart first presented the mouse and videoconferencing (Engelbart, 2003), and Ivan Sutherland introduced Sketchpad (Sutherland, 2003). The digital space represents a new domain that we are eager to comprehend and harness for our benefit.

For education in particular, the computer was the long awaited crystallization of the instructional designer's dream: a machine that would enable the individualization of instruction. Each student would have a tutor that would allow for personalized instruction and pace. Skinner made use of early machines to create Programmed Instruction (Driscoll, 1994), and since then others have followed. However, in spite of the many expectations, we are not yet at the point where we can claim that education has been revolutionized by computers. In spite of the many advances, and the many specific successes, the educational processes continue to be similar to those used centuries ago. This is not to say that the computer has played no role in education, but that we are only beginning to see the possibilities offered by a new medium through initiatives such as computer mediated communication and e-learning.

It is encouraging to know that it took fifty years from the time Guttenberg invented the press, to the point where we had conventions for books the way we know them today (Murray, 1997). This could mean that we are still exploring but are as yet unaware of the possibilities the new medium offers. It could also be that the computer, like the radio and the television, will not drastically change the way we go about the business of learning. This, however, is a statement that is not intuitively very convincing because of the continuous innovation in software and hardware. The radio and the television have not changed that much from their original versions, whereas the computer has drastically evolved since the inception of earlier models.

Whether or not we believe technology will change the way we learn, it is undeniable that people have to use computers. It is evident that the way we work has been strongly impacted by the arrival of computers, from writing a simple e-mail to using tools that increase our productivity. Many jobs require a minimum of computer skills, and more specialized occupations such as creating visual effects for films can't simply exist without software and hardware. It is expected that in the next decade the United States will not have enough people with an acceptable level of computer skills to meet the demand (Hayes, 2006).

Contrary to the popular image of the teenager that frantically plays video games all day long and is a computer wizard, many young kids find computers intimidating and difficult to use (Hayes, 2006). It is even more difficult if we consider older adults who struggle more than younger individuals to learn new technology (Hawthorn, 2005). Add to the problem the fact that many countries can hardly provide for basic food and health needs, least computer instruction, and we have a digital divide that is likely to expand as the software and hardware continue to be upgraded at blazing speed (Cullen, 2001).

Apart from socioeconomic issues, which are outside of the scope of this research, it is fair to say that we have not remained totally idle. The fields of instructional technology and human-computer interaction have stepped in to try and make the new systems as user friendly as possible. Studies have been carried out to determine the characteristics of efficient interfaces, how to better design interaction devices, and how to create better training programs to learn software applications. Examples of instructional efforts are Carroll's minimalism (Carroll, 1997; Carroll & Carrithers, 1984; Carroll, Smith-Kerker, Ford, & Mazur-Rimetz, 1987; Lazonder, 1994), and Sweller's Cognitive Load theory (Chandler & Sweller, 1996; Sweller & Chandler, 1991, 1994; Sweller, Chandler, Tierney, & Cooper, 1990; Van Merrienboer & Ayres, 2005; van Merrienboer & Sweller, 2005). In spite of the progress made, many people still struggle with ever expanding software and steeper learning curves.

Approaches to Software Application Training

Extensive work has been done in the area of Human-Computer Interaction (HCI) in order to make interfaces as intuitive as possible. Training interventions have looked at the same problem from a different angle, which is to provide instructions on how to use an already existing interface. The goal is to reduce confusion and cognitive load, and allow the user to be able to use the software as quickly as possible. The following sections aim to provide a brief overview of this research.

Training Manuals

Considerable research has been carried out on instruction manuals, particularly those that allow self-paced individual learning. Many commercially available manuals for self-instruction in software applications follow some or all of the research guidelines presented here. Probably the most well known work on manuals is Carroll's minimal manual (Carroll, 1997; Carroll et al., 1987). Rather than boring the user with drills, this type of manual attempts to provide meaningful authentic tasks. This addresses the novice's desire for meaningful action (Carroll, 1997), which is normally frustrated by the lack of knowledge of the system. Text is supposed to be reduced to a minimum, and material should be presented as briefly as possible. Error recognition and recovery is an important skill, and this can be taught by using system hints and checkpoints to synchronize the training with the system's state. An important premise is that the user should be involved extensively in the learning process. Minimal manuals have been found to reduce training time by 25% to 50% (Lazonder, 1994). In line with research on situated cognition, Carroll emphasized that instruction had to take place in context, and unrelated tasks would not be so effective. The problem with this exploratory approach is that novices might not have enough knowledge to adequately select meaningful action, in which case guided instruction would prove more effective (Kirschner et al., 2006; van der Meij, 2000).

The way instructional information is presented has also been extensively studied. A significant amount of work in this area has been guided by Sweller's Cognitive Load theory (Chandler & Sweller, 1991, 1996; Sweller, 1988; Sweller & Chandler, 1991, 1994; Sweller et al., 1990; van Merrienboer & Sweller, 2005). This approach differentiates between the effort made by the individual to learn the content (germane cognitive load) and the way the content is represented (extrinsic cognitive load). Instructional material that splits information sources, say text and related diagrams in different pages, will place a higher cognitive load on the learner because the learner has to keep a significant amount of information in memory while switching from one page to the other. Chandler and Sweller (1996) have shown that a self-contained manual that does not require the use of a computer is far more effective that instructional material that involves using the manual and computer simultaneously. This is explained entirely in terms of cognitive load, which is reduced in the first case because the learner needs only to concentrate on the manual, whereas the second option involves frequent shifts of attention between the manual and the computer. Nevertheless, instructional manuals commercially available frequently assume simultaneous use of the computer when completing the instructional tasks.

Manuals that include screenshots of the different system states in the task give better results in performance and time measures than purely text-based material (Martin-Michiellot & Mendelsohn, 2000). Additionally it is interesting to note that extensive information in graphic format is not necessary. Martin-Michiellot et al. (2000) created two variations of graphic manuals: juxtaposed and integrated. The juxtaposed manual presented screenshots with instructions to complete a task, whereas integrated manuals included further text information superimposed on the screenshots. Although both manuals gave better performance and time results than the purely text-based version, the two graphic ones did not differ significantly from each other. The graphic manuals were used before actually working with the computer but allowed learners to become comfortable with the software twice as fast than those who used a conventional manual with a computer. Background knowledge of similar functions in other software applications played a significant role. Additionally, Martin-Michiellot et al. suggest that information should vary depending on the activity category: "command (transformation of a set of objects that are part of the task); workspace area (moving and selecting objects over the document); reading information (reading the screen without taking any action); and environment configuration (not directly related to the work)" (p.292).

Van der Meij (van der Meij, 2000) suggests that full-screen images –rather than partial shots– in a two-column format with corresponding instructions on the left is a more effective strategy reducing training time by 25% and increasing retention by 60%. It is important to note that there was a strong correlation between age and completion time, which might suggest that there are other intervening factors. Gellevij et al. (Gellevij, van der Meij, de Jong, & Pieters, 1999) also found that partial screen captures were less successful than full screen images. Van der Meij (2000) propose four important dimensions for software documentation: coverage, positioning, size, and cueing. Coverage refers to the portion of the screen that is captured and displayed in the material. Positioning considers the location of text relative to the corresponding image. Size specifies the percentage by which the actual screen is reduced to fit on paper, which is recommended to be between 50% and 75%. Finally cueing indicates all visual aids that direct attention to specific areas or elements of the screen. A distinction is made between manuals for learners and for frequent users, noting that the first might not succeed at creating meaningful action as Carroll recommended (Carroll, 1997).

Training Wheels

Another important training intervention in software applications is the concept of training wheels, first proposed by Carroll (Carroll & Carrithers, 1984). Only a core set of commands is available in this interface, whereas commands that lead to frequent errors when the user is learning the system are non-functional. The mouse also changes when the user rolls over a non-relevant function. The interface in Carroll's study was created iteratively, based on an empirically determined set of frequent errors made by novices. By blocking some of the functionality of the software, errors are minimized and the novice user can focus on learning the core functionality of the software.

Other studies have also tested the training wheels interface (Bannert, 2000; Hawthorn, 2005). Bannert compared a training wheels interface (TWINS) with a standard one (STAND), and a self-paced course (SELF) with a tutor-based one (TUTOR). Both TWINS and SELF outperformed STAND and TUTOR in learning time but not in performance. SELF had a higher acceptance among users than TUTOR. This may be because users can go at their own pace and repeat topics, which are some of the reasons why users seem to prefer self-paced material than tutor-based (Lippert & Granger, 1998). Good results have also been obtained with a training wheels interface designed for older adults that were unable to learn file management with other methods (Hawthorn, 2005).

Leutner (Leutner, 2000) took this idea further by applying the training wheels concept to CAD software. Leutner wanted to see if this type of instructional intervention was applicable to software different from word processors. The concept of training wheels was extended using a Double-Fading Support (DFS) approach, which means that the software had partial functionality and detailed guidance at the beginning, and functionality was increased and guidance reduced until the user ended up with a fully functional version of the software with no guidance. 50 commands were initially available and gradually expanded to 600 commands by the end of the study. Furthermore, Leutner compared text-based and icon-based versions of the CAD software. Users in the DFS group performed better, but only in the text-based version of the software but not in the icon-based version. Leutner hypothesized that icons reduce cognitive load, so when using graphic interfaces either guideline or functionality fading should be used but not both simultaneously.

Online Training

A recent approach between training manuals and digital media are guided simulations which are normally deployed online or as stand-alone applications. Software such as Adobe Captivate allows the instructional designer to 'capture' procedures that can later be shown to the learner as an animation. Alternatively, the learner can attempt to reproduce the sequence by clicking on the exact same spots the instructor did when creating the simulation. Immediate feedback tells the user whether the selection is correct or incorrect. This type of instructional material is known as guided simulation, and also falls into what is considered rapid e-learning development (De Vries & Bersin, 2004).

Although this type of material is interesting because it involves the learner rather than treating her as a passive recipient, studies involving web-based software instruction for call centers failed to show a difference in performance when compared to simple text-based instructions (Sukhai, 2005). Text instructions rated highest in efficiency against animations and guided simulations. Sukhai attributes these results to the background of users. If there is enough previous knowledge of computers, Sukhai suggests that the utility of animations and guided simulations may be limited.

There might be other factors affecting performance in online environments for software instruction. Tsai found that computer self-efficacy predicted better performance and satisfaction with different approaches in online training (Tsai, 2004). Tsai used different strategies to present the content in an online course designed to teach how to use a program (Netscape Composer) to create web pages. Four forms of information presentation were used: deductive-expository, deductive-inquisitory, inductive-inquisitory, and inductive-expository. No significant differences in performance were found. Users with high computer self-efficacy performed better and were more satisfied regardless of the condition.

An important issue in online training is the anxiety and confusion novices seem to experiment when encountering new digital environments. Due to the similarities of finding oneself in an unfamiliar place, some authors have suggested that the digital space can be compared to physical space, an idea that some have considered an early adoption of a poor metaphor (Farris, Jones, & Elgin, 2002). In a physical environment we find our way by means of landmarks and cues that allow us to create paths that later become automatic. In the same way, we seek to enable new users to find their way in the virtual space.

Farris et al. (2002) have strongly questioned the space metaphor for hyperspace. They created different sites that users had to browse. Users then drew the site's structure according to their own perception of the hierarchy of the space. Results showed that drawings of the hierarchy of the sites did not match the actual site's hierarchy. Boechler (Boechler, 2001) has analyzed the different arguments in favor and against the use of a spatial metaphor to describe hyperspace. Although the metaphor is intuitively attractive some difficulties still remain. For example, what is the concept of distance in digital space? Does it refer to the semantic distance between items or how "far" some menu items are from others? How does a spatial layout maps to conceptual spaces? Boechler and Dawson (Boechler & Dawson, 2005) have suggested that "the appropriate question is not whether a spatial versus conceptual navigation tool is better, but rather which combinations of conceptual and spatial cues provide the best support for a particular set of goals" (p. 42).

In Boechler and Dawson's (2005) study, groups were presented with four different navigational tools and their effect on performance, path patterns, and mental representations was measured. The four navigational tools were: alphabetic (items organized alphabetically), hierarchical (a single list with color codes that showed their hierarchy, which is the way many menus work), spatial (items spatially located and clustered), and hierarchical/spatial (a hierarchy graph). All groups built similar mental representations, although the way users navigated information with spatial cues was a predictor of the structure of their mental models. The number of steps necessary to go from one page or menu item to another was used to create a correlation matrix, which was analyzed using multidimensional scaling (MDS). These 'distances' were later compared to a distance rating between items given by users, suggesting some kind of semantic separation in their mental models. Each navigation tool substantially determined, although not entirely, where users would go next, and this in turn influenced how 'far' or 'close' items were perceived to be. It was hypothesized that as users interact with the document they build a mental model, which also affects the way the user navigates the space. This is consistent with different distributed task representations research (Zhang, 1987, 2000; Zhang & Norman, 1994). Regarding performance, users of the alphabetical navigation tool performed poorly as compared to the other navigation layouts. Those users interacting with spatial cues showed higher recall. This study also suggests that there should be a significant relationship between the representation and the conceptual organization of the content.

Summarizing, some of the existing research on software training approaches has been considered, as well as the way different presentation methods seem to impact learning. The next section will deal with how different views on cognition see the interaction between users and a computer system.

Cognition and Computer Use

Human-Computer Interaction and Mental Models

Research on computer skills training started in the late 80's and early 90's, when personal computers became more available and widely used in work environments. Many initiatives were based on previous work on problem-solving and mental models. Work carried out in the seventies (Newell & Simon, 1972) strongly influenced models of human-computer interaction in the following decades. Newell and Simon viewed humans as information processing systems, and problem solving was conceived as an inner process that involved the use of list-like structures or production rules. Once a goal was determined, the user would select the necessary operators to get the current system state as close as possible to the end state. Damper (Damper, 1997) notes that Newell and Simon acknowledged the importance of the task environment and the structure of the problem space, and their influence in problem solving, as well as the relevance of perceptual and motor skills.

The GOMS model (Goals, Operators, Methods, and Selection) and its different variants are based on Newell and Simon's approach (John, 1995). These symbolic operations models are able to accurately predict different outcomes including time to carry out a task and time to learn a new procedure. However, they also assume that the user already has knowledge of the different interface elements and their functions. This is not very informative from the point of view of instruction, as the challenge for novices is precisely learning the commands and functions available in software applications. The model also assumes that information is in the user's head, although the CPM-GOMS

(Cognitive-Perceptual-Motor GOMS) model includes operators that are, as the name indicates, not just cognitive but also perceptual and motor (Bonnie & Kieras, 1996).

Research on mental models, memory, learning, thinking, and problem-solving was more specifically applied by Manktelow and Jones (Manktelow & Jones, 1987) to interface use and design, which according to Sasse (1997) is not only a unique piece of work in the field but also unfortunately disregarded by later studies. However, their work is more in line with distributed cognition and will be considered in more detail later.

It seems to be generally agreed that the representation of the system does not have to be exact, but with just enough information and functionality to operate it. Sasse (Sasse, 1997) suggests that existing models will be used when no model is in place and background knowledge will play an important role on how the new model is built. Novices tend to use computer systems in a trial-and-error fashion and will find difficult to learn the functionality of a software application if no similar previous models are in place. Elements of the interface play an important role in building user models of the system. Sasse also proposes that there appear to be two ways of building a user's model of the system: either by cueing an existing one, or building it from scratch, each method requiring a different approach.

In the first case, analogies can be used but their structural integrity and correct match with the target system is important in order to avoid later breakdown (Spiro, Feltovich, Coulson, & Anderson, 1989).

When building the model from scratch, the prepositional representations have to match the heuristic processes and be in the appropriate format (Sasse, 1997). The user can build a surrogate model or a task-action mapping model (Young, 1983). According to Young, a surrogate model is a working model of the system mostly used to predict the system's behavior. It does not however consider the task or the user. In the task-action mapping model, the objective is to create a link between the task to be performed in the system and the actions the user needs to perform to carry them out. A smaller core set of tasks is normally initially selected to create the mapping, and other tasks are presented as variants of the initial core set. The surrogate model does not say anything about learning or performance. The task-action mapping model allows for the inclusion of performance, but not learning. Young points out that these models are more concerned with the device itself and as such are classified as accommodatory. In contrast, assimilatory models attempt to use models already familiar to the user. Again, in this last case breakdown of metaphor or analogies used can hamper the creation of an effective model.

Although studies suggest that users who are initially presented with a model of the system perform better than those who receive procedural instructions only (Borgman, 1986; Kieras & Bovair, 1984), the fact that users were unable to recall the initial model suggests that either they were unable to articulate it or they chose to create and use their own (Sasse, 1997). Other studies seem to confirm that an initial conceptual model of the system can be particularly helpful in the software domain since in many cases there is no background knowledge on which to rely (Ben-Ari & Yeshno, 2006).

They way in which a user encodes the system has also stirred some debate. Some studies suggest that information is actually encoded in procedural form, or following the command sequence as opposed to functional encoding, which uses the real task functionality to act as the criteria to organize information. Pennington (Pennington, 1987) suggests that in programming, novices create a mental model that is procedural and later when becoming experts their model includes functional information in the language of the real world. However, Navarro-Prieto et al. (Navarro-Prieto & Cañas, 1999, 2001) proposed that visual programming languages help create a user's model based on data flow (functional) rather than procedural, as images would allow the user to access semantic information faster.

Regardless of how the model is created, information in graphic form seems to play an important role in the creation of mental models of programming procedures. The graphic information in this case allows for better retrieval (George, 2000). Research seems to show that experts use visual imagery to guide the design and programming process (Petre & Blackwell, 1999). Although not in the software domain but adding to the importance of visual information, some studies suggest that highly dynamic visual representations create more images in learner's mental models, and motion cues increase this level of performance (Wu, 2003). Motion cues are graphic elements that suggest dynamic features that could otherwise be represented with animations. In both cases however, the different changes of the system state are evident and not left to the user's imagination. Spatio-temporal interfaces, representing the changes of state in space and time simultaneously, appear to be more effective when considering transformational processes (Sedig, Rowhani, & Liang, 2005).

System states are important in computer systems providing feedback to the user when an action is carried out. Sometimes however computer applications will not show any noticeable change on the interface except after a series of commands has been selected. In this case, planning is necessary. Nevertheless, Reimann and Neubert (Reimann & Neubert, 2000) note that new computer users rarely have plans, that is, they do not decompose the task into goals and subgoals, and they do not have detailed knowledge of primitive actions. Reimann et al. suggest that the environment will trigger some subgoals, whereas the user will know some of the necessary actions. Acquiring the knowledge of the effects of different actions is not an easy process particularly when not every single action has an evident effect on the system state. As Reimann et al. state: "Conversely, a single action may have no visible (or an unnoticeable) effect and it is only the combination of two or more actions which lead to significant changes in the [graphical] environment" (p. 318).

It can be seen that in spite of extensive work on mental models and human computer interaction, we are not clear about the mechanisms involved in building mental representations, particularly the accurate ones required for successful interaction with software artifacts (Sasse, 1997). Even if we would like to believe that mental models are clear and organized structures that follow logic, Norman's (1983) observations regarding characteristics of users' models are probably closer to the truth:

1. Users' models are incomplete.
2. People's abilities to "run" their models are severely limited.
3. Mental models are unstable: People forget the details of the system they are using, especially when those details (or the whole system) have not been used for some period.
4. Mental models do not have firm boundaries: similar devices and operations get confused with one another.
5. Mental models are "unscientific": People maintain "superstitious" behavior patterns even when they know they are unneeded because they cost little in physical effort and save mental effort.
6. Mental models are parsimonious: Often people do extra physical operations rather than the mental planning that would allow them to avoid those actions; they are willing to trade-off extra physical action for reduced mental complexity." (Norman, 1983, p.8)

Beyond the challenges presented by mental models, another problem with the functional model view is that it is unable to account for the multiple action slips experts make when using computer systems, which has been estimated between 5% and 20% (Kitajima & Polson, 1995). Even more puzzling is the fact that contrary to what would be expected, experts are unable to recall from memory the position and labels of many of the commands used to perform a task, even though they are able to verbally articulate the steps the task requires (Mayes et al., 1988). Mayes et al. found that experts' recall is slightly higher than that of occasional users, but this difference was not statistically significant. They suggested that these results challenged the view of an expert user with an accurate mental model of the system, and indicated the possibility of an information flow model, where the different display elements prompt the necessary steps to change the system state until a goal is achieved. Notice however, that experts do know the overall necessary steps required in a known task, a key issue that is not frequently mentioned when quoting Mayes et al. work to argue in favor of a more prominent role for the elements of visual interfaces. Mayes et al. correctly suggest that expertise may be equivalent to knowing which cues are the key ones for a given task, and what their function is. An alternative approach to the internal functional model is considered next.

Display-Based Competence, Situated Cognition, and Distributed Cognition

Some authors have proposed a rather different approach to problem solving from that suggested by Newell and Simon. Tversky and Kahneman (Tversky & Kahneman, 1973) explored the effect of availability when evaluating the frequency of a class or the probability of an event happening. The easier it is to bring a class or an event to mind, or the more readily available it is, the higher the estimated probability. Pollard (Pollard, 1982) extended this idea to reasoning tasks. According to Pollard, subject responses are cued by availability from the individual's own experience, and from the available or salient features of a task. In Pollard's opinion, available cues actually help people in the decision making process, although he did not define what these 'salient task features' were. Logical validity is not the user's main concern, which would explain why they frequently use a trial-and-error technique rather than logically analyze the goal and necessary steps to reach it. This approach might work well for experienced users who can transfer previous knowledge to the new environment but not so much for novices (Charney, Reder, & Kusbit, 1990). Highly familiar cues will trigger existing schemata or direct memories (automatic routines), and highly unfamiliar cues will activate heuristic procedures (Manktelow & Jones, 1987).

Studies in psychology have also dealt with the issue that cues can actually trigger unconscious and conscious automatic processes. In particular, unconscious stereotypical judgments have been shown to be an automatic unconscious response to a particular stereotype-related stimulus that the subject is not aware of. In some cases the individual will be aware of the external cue but not of its conscious effect. Finally, if a user has a specific goal in mind, cues that are conducive to achieving this goal will automatically trigger actions that will help complete the task (Gollwitzer & Schaal, 1998).

Larkin (Larkin, 1989) continued work on display-based interaction, disputing the idea that users follow sequential algorithms in their heads. In her work on display-based problem solving, she suggested that display-based elements cue internal operators and help carry out a perceptual process rather than a logical one. Errors appear when the user sees the display just as a string of symbols, failing to see the information represented. In accordance with Mayes et al. (1988), Larkin considered information captured by the display about the current state of the system as essential for the perceptual process to take place. This information is conveyed in the form of data structures that reflect the system's state and suggests the next action to take. This idea has been used by Smith et al. (Smith, Duke, & Wright, 1999) to guide users in virtual 3D environments.

Kitajima and Polson (1995) suggested that display elements hint correct action sequences rather than retrieval from schemata. Thus, an information flow from the device to the user is thought to be essential for problem solving in computer applications (Howes & Payne, 1990; Payne, 1991). The user selects from a variety of alternatives on the interface –which act as cues– and travels through system states, achieving the desired goal if the correct sequence is followed. Novices have to learn the different sequences by practicing them extensively. Kitajima and Polson (1995) agree with this last point noting that experts are able to decode the different cues and link them successfully to produce the desired result, but this information has to be retrieved from long-term memory. Novices will find initially hard to acquire this knowledge, and even experts will not be able to perform the tasks without errors, an interesting finding not consistent with the GOMS model as mentioned earlier.

The view that problem solving does not take place in the individual's mind exclusively, but that it is a result of a flow of information between the device and the user is congruent with learning theories such as situated and distributed cognition. A seminal article on situated cognition, Situated Cognition and the Culture of Learning (Brown, Collins, & Duguid, 1989) appeared around the same time Larkin (1989) was working on display-based problem-solving. Situated cognition is based on evidence that rather than processing information exclusively in their heads, individuals use external artifacts present in the situation at hand to build a solution. These artifacts can be cultural, social, or specific to a certain profession or activity (Driscoll, 1994; Henning, 2003). Distributed cognition similarly argues that knowledge is distributed among the individual and external representations, and different representations will cue different operators and processes, and will pose different cognitive loads (Zhang, 1987, 2000; Zhang & Norman, 1994; Zhang & Patel, 2006).

Situated cognition is particularly interested in the cultural artifacts and lived practices shared by different professions. The introduction of the novice into the circle of expertise is done through apprenticeship or legitimal peripheral participation in communities of practice (Driscoll, 1994). Practice with experts will lead the novice to use the artifacts and language of the specific domain. Such process of familiarization and appropriation is evidenced by the use of language and interaction. For this reason, situated cognition borrows heavily on ethnographic methods to analyze the conversations that take place and how knowledge is shared and acquired. This view naturally applies to software applications as well. Graphic designers, for example, use certain artifacts or tools such as PhotoShop and Illustrator, and handle a vocabulary that is typical of the profession. Novices gradually acquire both the language and the use of the tools. Even though the learning process could be approached from the point of view of cultural assimilation it is believed that in this instance the question of how cues can trigger perceptual problem solving is better analyzed by distributed cognition.

Distributed cognition assumes that information is distributed and flows across different representations. This has an impact on the unit of analysis selected and the processes that we are concerned with (Hutchins, 2000). For example, Hutchins suggested that a cockpit acts as a cognitive system where information flow is distributed across representations such as the pilot, speech, and cockpit artifacts (Hutchins & Klausen, 1996). The representational properties of the medium can support or hinder the flow of information. As an example, he showed that the simultaneous visual representation of the target and current speed in an aircraft cockpit allows for fast perceptual comparison of the two, as opposed to having to memorize the target speed and compare it with a digital display (Hutchins, 1995; Wright et al., 2000).

Zhang and Norman (1994) proposed that different representations make different information available to the user and hint different operations or problem solving strategies. By creating isomorphic representations of the same problem, in this case the Tower of Hanoi, they suggested the following about external representations:

1. External representations can provide memory aids
2. External representations can provide information that can be directly perceived and used without being interpreted and formulated explicitly
3. External representations can anchor and structure cognitive behavior
4. External representations change the nature of a task
5. External representations are an indispensable part of the representational system of any distributed cognitive task (Zhang and Normal, 1994, p. 29)

Zhang (1997) carried out an analysis of representations for the popular game Tic-Tac-Toe. Four isomorphic representations were created using lines, shapes, colors, and numbers. A set of experiments showed that different representations prompted different strategies, even though the underlying mechanics of the game and the winning strategy was identical for all four representations. They also identified what they called biases, which are based on specific properties of the representations guiding action in a particular direction. In spite of the fact that both experiments used well-defined problems, they confirmed the impact of representations on problem solving strategies.

The studies conducted by Zhang (Zhang, 1987, 1996, 1997, 2000), Zhang and Norman (Zhang & Norman, 1994), and Zhang and Patel (Zhang & Patel, 2006), provide a framework to analyze distributed cognitive tasks and external representations. Even though early articles outlined the general components of this framework, later ones have extended it and made it more precise. These studies were more concerned however with different existing representations and their effect on the strategies used. They do not say anything about what information should be made explicit and where it should be located.

The Resources Model proposed by Wright et al. (Wright et al., 2000) seeks to provide guidelines to analyze distributed environments in terms of data structures, how they are represented, and what strategies are needed to interact with these resources. An Extended Resources Model has been proposed by Wang et al (Wang, Deng, Ma, Hua, & Dai, 2005).

The distributed information resources model. Wright et al. (2000) proposed the Distributed Information Resources Model, or Resources Model for short to analyze distributed problems. Although there has been considerable research on distributed systems, and a framework proposed by Zhang (1996, 1997, 2000) and Zhang and Norman (1994) to analyze isomorphic representations, no framework has been proposed to specify what components need to be considered in distributed problems. The Resources Model tries to fill that gap. It is important to note that most of these research efforts have been directed at building better, more user-friendly interfaces, and not at developing training material. However, the Resources Model is a valuable tool for instructional efforts as well.

The Resources Model first considers the abstract information structures present in the problem, which can be represented and used as resources. These abstract information structures are plans, goals, possibilities, history, action-effect relations, and system states. How to represent each of them depends on the problem at hand. Processing in the Resources Model supports the cyclic flow of information and the model assumes the exchange of information between resource configuration and decisions about action. That is, resources are configured in a certain way at a given moment in time. Together they form what Larkin would call a data structure that can inform action. This action in turn modifies and updates the resources configuration and a new cycle starts. Strategies to interact with the resources are proposed, although the list is not exhaustive: plan following, plan construction, goal matching, and history-based selection and elimination. Each strategy can be defined in terms of necessary resources, as follows:

Plan following = *fn* (plan, history, state, goal)

Plan construction = *fn* (goal, state, possibilities, action-effect)

Goal matching = *fn* (goal, possibilities, action-effect)

History-based choice = *fn* (goal, possibilities, history)

For example, in plan following the plan must be evident in some way, the steps that have been or will be taken should be available to the user, state changes as the plan is followed have to be attended to, and the goal has to be specified.

How each resource is to be represented is something that is not specified by the model. Current interfaces do not offer many alternatives apart from visual, and less predominantly, auditory representations. Regarding visual representations, which have been studied extensively, Larking and Simon (Larkin & Simon, 1987) provide the following reasons why diagrams can be superior to verbal descriptions:

1. All relevant information can be gathered in one place simultaneously
2. Location can act as grouping criteria minimizing the need for labeling
3. The allow faster perceptual processing

The last item is particularly important in this discussion as it suggests the possibility that the user will "detect cues there [in the diagram] and use them to retrieve problem-relevant inference operators from memory" (p.97). Although consistent with Paivio's findings regarding higher recall for pictures, information is sometimes presented in both visual and verbal format in line with dual-coding theory (Paivio & Csapo, 1973).

The ideas proposed by distributed cognition are intuitively appealing. Nevertheless, in spite of the fact that the Distributed Information Resources Model has attempted to describe a set of possible resources that can be made available on the computer display, and a partial list of strategies used by individuals to make use of these resources, this information is insufficient to design interactive spaces because it does not say much about how these possible strategies are constrained by the characteristics of the digital environment. The lack of specific metrics and information about how people coordinate resources depending on the features of the environment they are acting on is one of the main challenges of distributed cognition (Kirsh, 2006).

Kirsh reminds us that individuals and the world are closely coupled systems, and that people do behave in certain ways depending on the affordances of the environment. The chosen tactics and actions will have some sort of cost that is not apparent unless a more detailed analysis is carried out. He points out that in saying that one environment is better than another for a certain purpose, we are not addressing the key issue which is why. If we know the combination of elements that make a digital space superior in some way, then we are able to incorporate this information in our designs. Embodied cognition addresses this concern.

Embodied Cognition

Just how important are the actions we take in interactive environments and how are they related to cognitive and perceptual mechanisms? Ballard et al. (Ballard, Hayhoe, Pook, & Rao, 1997) argue that at time scales of 1/3 of a second, physical action will have an impact and connect perceptual and cognitive processes: "At this 'embodiment level', the constraints of the physical system determine the nature of cognitive operations" (p. 723). Although their study refers to visual systems, the idea is that the separation between cognitive and perceptual processes is not so clearly defined and both "are interlocked for reasons of computational economy" (p. 724). Damper (1997) agrees with this position mentioning that it was somewhat interesting that

Newell, in spite of his emphasis on formal operations, felt the separation between cognition, perception, and motor actions would lead to an incomplete account of what happens between a stimulus and its corresponding response.

Different authors have partially criticized Ballard et al.'s work, in spite of appreciating their contribution. One of the claims is that the study overlooks the importance of learning and the progressively lower task completion times as a result (Epelboim, 1997; Juttner, 1997). It has also been suggested that the very narrow 'embodiment level' could give the impression that at larger or shorter time scales, cognition is not embodied (Feldman, 1997; R. Wilson, 1997). Additionally, the computational cost of different strategies is not addressed (Maglio, 1997).

The issue of computational efficiency and strategy cost is an important one. Kirsh and Maglio (Kirsh & Maglio, 1994) found that people playing Tetris will choose to rotate the pieces physically even if this implies a higher motor effort because it will reduce cognitive load and also because it is computationally more efficient. This agrees with observations by Norman (1983) reported before. Hitting the arrow keys several times is faster than rotating the piece in one's own mind. They do distinguish however between motor actions that get one closer to a goal –pragmatic– and those that reduce mental effort –epistemic–. These epistemic operations are thought to decrease memory required, number of steps, and probability of error of mental computations. The external game environment becomes an "interactive visuo-spatial sketchpad" (Maglio & Kirsh, 1996) p. 396 where users offload information.

Operations such as hitting a key, moving the mouse, verifying the mouse target position and so on combine to form what is known as microstrategies, which are influenced by physical characteristics of the interactive environment, and are largely chosen by the individual based on differences in efficiency as small as 150ms (Gray & Boehm-Davis, 2000). Whereas the 'embodiment level' refers to operators at a time scale of 1/3 of a second, the time scale for microstrategies range between 1/3 of a second and 3 seconds. Gray and Boehm-Davis created predictive models using the GOMS-CPM notation for two studies involving move-and-click and click-and-move tasks, finding that users tend to choose the fastest microstrategy, resulting from interface features.

Gray and Fu (Gray & Fu, 2004) carried this research further and proposed the soft constraints hypothesis. In their view, whereas hard constraints determine what the user can or cannot do in the interactive environment, at the 'embodiment level' soft constraints influence which of the possible microstrategies are most likely to be used. In interactive environments the selection of microstrategies is not thought to be planned as this type of processing would take a much longer time than the microstrategy itself. Simple observation of a person using a computer application will show that these actions are performed quite quickly.

Gray and Fu (2004) assumed that memory, perceptual, and motor operators were all given the same weight and the different possible combinations would create microstrategies of different durations. Three different conditions were analyzed using a simulated VCR environment where participants had to program different TV shows. In the first one the information necessary to program the shows was freely available. This scenario assumed that the user would need only to move his/her eyes to get the exact information at all times. Obtaining information by a simple perceptual-motor operation (eye saccade) was thought to have a lower cost that trying to commit it to memory. The second scenario put a cost on accessing the information by locating a grey box over it and forcing the user to roll the mouse over the box and click it to see the content. The increased cost for perceptual information was expected to create a mix of strategies. The third scenario asked participants to learn the show information before programming the unit. The information was still available on the screen as in the second scenario but selecting it obscured the VCR and left only the show information visible on the screen. This condition had the highest cost for perceptual-motor operations, and the lowest for memory operations. It was expected that memory operations would be favored over perceptual-motor ones. In all scenarios the cost of microstrategies only differed by milliseconds.

An analysis of the number of errors made while programming the shows provided interesting insights. People in the third condition where memory operators were encouraged made the least amount of errors of all three conditions. Participants in the second scenario where information had a somewhat lower cost but was not totally free had the worst performance. Individuals in the free access condition were in-between. In a somewhat counterintuitive manner, rather than making the necessary perceptual-motor effort to access the available information, in general users preferred to rely on imperfect memories because of the lower cost. From an instructional point of view, although it is very significant that memory, perceptual and motor operators at the embodiment level are intimately related and are affected by soft constraints, the most important conclusion here is that the choice of operators will actually have a major effect on performance.

The soft constraints hypothesis goes against assumptions that individuals try to conserve cognitive resources and favor perceptual-motor ones (M. Wilson, 2002). More recent research compares the minimum memory hypothesis versus the soft constraints hypothesis (Gray et al., 2006). While in the first approach perceptual-motor operators have a higher weight in order to preserve a very limited resource (memory), the second view assumes an equal weight for memory, perceptual, and motor operators and sees time as the limited resource. The Blocks World Task and a model built using the ACT-R model were used to compare both hypotheses. Both the task and the model supported the soft constraints hypothesis.

Freely available information does not necessarily mean better performance. It is commonly believed that the more information we make available to the user, the better. Waldron et al. (Waldron et al., 2007) have shown this is not necessarily the case. In particular, they argued that information that can be accessed freely may result in "impoverished encoding and poor retention of visual-spatial information" (p.696). The ideal amount of information will depend on the interactive environment's objective. The Blocks World Task was used in this instance to show "that reducing information accessibility or availability can be of benefit when criteria of task performance include the use of memory and learning" (p. 694). The Blocks World Task consists of a display that has three windows: the resources window, the workspace window, and the target window. The target window shows a pattern of color squares that has to be reproduced by dragging squares from the resources window to the workspace window. In this study, Waldron et al. (2007) created three conditions: the High Information Access Cost (H-IAC), the Medium IAC (M-IAC) and the Low IAC (L-IAC). Access cost was defined as the time it took to access the resource in milliseconds. Following Gray and Fu's (2004) technique, they used gray masks to cover some of the windows in some conditions. In the H-IAC the target window was covered and the user had to roll the mouse over the mask and wait one second to see the window. In the M-IAC access to the window was obtained by simply rolling the mouse over the mask. In the L-IAC all windows were visible at all times. Recall was better for the H-IAC condition, where users took longer times to complete tasks but uncovered the target window only a few times. This is attributed to more intensive memory operations.

Waldron et al. also showed that low IAC can have negative effects on visual-spatial encoding. Using an aircraft display where participants had to navigate to, and estimate the location of certain targets, the location of targets on terrain maps improved when onscreen information availability was reduced suggesting that in the absence of information memory encoding had taken place.

In another study Morgan et al. (Morgan, Waldron, King, & Patrick, 2007) tested again the idea that under certain circumstances increasing the IAC can be desirable. The objective in this case was to determine if different IAC would lead to better recall of visual information. Additionally, the effect of IAC on planning was also considered. The Blocks World Task was used for the first part, adding eye tracking to the measures taken. Results confirmed that in spite of the fact that participants in the low IAC condition took less time to complete the tasks, post-test recall was significantly better for medium and high IAC. In the high IAC condition the target window was accessed less than in the medium IAC one, which in turn was lower than the low IAC environment.

In the second part an interesting approach was the evaluation of two different planning strategies based on interface IAC. Morgan et al. (2007) assumed that an individual that does not manipulate the interface while thinking about a multistep task is using a memory-based planning approach. In contrast, if the individual manipulates in any way the resources available on the interface, then a display-based planning approach is being used. To determine planning style the Blocks Problem Solving Task, a modified version of a popular puzzle, was implemented. A 3x3 grid with 8 color squares in their final position showed the target pattern. The work window had the same 8 color squares but in a different order. By moving one square at a time using the remaining adjacent space in the grid, participants had to recreate the final state. The type of planning strategy, display-based or memory-based, was determined based on the number of moves. Whereas in the high and medium IAC environments participants slowly moved from a display-based to a memory-based approach as they solved the tasks, participants in the low IAC condition did not show this pattern. This was seen as evidence of learning in the high and medium IAC conditions.

Research reported here has mostly considered tasks that involve a single step with immediate feedback like moving a color square to its target position. Fu and Anderson (Fu & Anderson, 2008) have focused on typical tasks normally carried out in software applications, which are processes that require several steps without any immediate feedback from the interface. The use of internal and external cues, mediated by interactive processes, is important for solving this type of problem.

Fu and Anderson define interactive skills as "learning of action sequences in situations that depend critically on the utilization of external cues". In this type of tasks they differentiate between declarative memory encoding and reinforcement learning. The first one is assumed to happen as a result of specific repeated stimuli and requires memory and attentional resources, whereas the second appears to be the result of a habit mechanism slowly formed through interaction with the probabilistic environment and apparently independent of declarative memory. Reinforcement learning appears to depend on external cues and take place at the same time as the slow declarative-to-procedural process happens. It is proposed that reinforcement learning may "reflect a primitive form of learning that is sufficient to learn the associations between external cues and actions without verbal mediation" (p.190). Furthermore, Fu and Anderson state that "existing models of skill acquisition may need to be extended to include some forms of reinforcement learning that does not require the gradual declarative-to-procedural progression" (p.190). The suggestion is made that even in the presence of high cognitive load reinforcement learning may still take place.

To summarize the previous discussion, the important point here is that low-level cognitive, perceptual, and motor operations cannot be seen as separate as is frequently the case. Memory-based and display-based strategies act together to influence the way the individual uses the interface. The idea that cognition encompasses not only formal operations 'in-the-head' but also the situation the individual is in as well as the physical properties of the environment he/she is acting on is what is known as 'embodied cognition'.

According to Wilson (M. Wilson, 2002), the main contentions of embodied cognition are:

1. Cognition is situated
2. Cognition is time-pressured
3. We off-load cognitive work onto the environment
4. The environment is part of the cognitive system
5. Cognition is for action
6. Off-line cognition is body based (p. 626)

Instructional Framework

As mentioned before, distributed cognition and situated cognition share many elements, particularly the use of external artifacts in a distributed environment or contextual setting. By making resources available to the learner, it can be said that the environment is reducing the cognitive load required from the individual, as the user will not have to memorize all the information. The Soft Constraints Hypothesis informs the way these resources will be used based on the cost required to access their content.

As instructional designers we do not have the option of altering the software application itself. We start from the premise that software designers do their best to design user-friendly environments, even if the result is not always completely successful. Our task as instructional designers is to teach the novice how to use the application as efficiently and effectively as possible. The Resources Model helps us realize that in this learning process we need to 'perceptually enhance' resources that will be available in the application interface, and also create new resources that will make hidden information obvious to the learner thereby increasing the chances that he or she will engage in the information flow that leads to correct decision-making or perceptual heuristics.

As a reminder, this research focuses on basic operations that need to be automated before more complex tasks can be attempted. Nothing has been said regarding the instructional approach to teach these simple procedures. Nevertheless, the issue of cognitive load has been addressed and it is this theory that informs the structure and sequence of the exercises included in all environments used in this study.

This section reviews the Four Components Instructional Design System and the implications for content presentation and sequencing. Additionally, since one of the environments is a game, a brief overview of research on games as instructional tools is also presented.

The Four Components Instructional Design System

Many of the instructional initiatives that were covered in the Software Training section use tasks found in real settings to gradually introduce the novice to the application. Increased cognitive load may result from attempting complex tasks that are normally part of discovery learning and problem-based learning for which the novice is ill-prepared (Kirschner et al., 2006; van Merrienboer, Clark, & de Croock, 2002; van Merrienboer, Kirschner, & Kester, 2003; van Merrienboer & Sweller, 2005). In the case of a problem-solving approach, Sweller(Sweller, 1988) suggests the cognitive processes at play are incompatible with schema acquisition so carrying out both activities (schema acquisition and problem solving) at the same time increases cognitive load.

Novices normally start with step-by-step instructions that are later integrated into larger units until the process becomes automatic. Exploratory learning does not allow for the kind of repetitive drills that lead to automatic processes (Wiedenbeck, Zavala, & Nawyn, 2000). Without the constituent skills, more complex tasks will be likely overwhelming for novices (van Merrienboer et al., 2003). Without basic task-action mapping, an authentic task can create unnecessary anxiety. In terms of Landa (Ragan & Smith, 2003), it is important to acquire the necessary knowledge before being able to learn a skill.

Instructional approaches as well as human-computer models agree that novices require extensive practice to master the necessary skills to use a software application (Kitajima & Polson, 1995; Sukhai, 2005). Van Merrienboer et al. (2002) have proposed a four component instructional design system (4C/ID) for complex tasks. The system presents the following essential components for complex task learning:

1. Learning tasks: authentic tasks that help the user build relevant schemata
2. Supportive information: information supporting non-recurrent aspects of the task helping the learner use previous knowledge
3. Just-in-time information: relevant to recurrent aspects of the task, particularly procedural rules which also apply to the next component

4. Part-task practice: rule automation for procedures frequently repeated in the complex task

As Van Merrienboer et al. suggest, just-in-time information and part-task practice are essential to automate procedures, which is what we are looking for when we consider certain core repetitive tasks in software applications. A central tenet of this design system is that since the whole task is initially too much for the novice, the following methods for scaffolding the authentic task practice are necessary (van Merrienboer et al., 2003): (a) simple to complex sequencing of task, (b) worked-out examples, and (c) completion tasks.

In summary, the instructional content in all environments was informed by the 4C/ID system proposed by Van Merrienboer, Clark, and de Croock (2002) and extended by Van Merrienboer, Kirschner, and Kester (2003). While the 4C/ID system informed sequencing of the tasks and the general structure, the Resources Model guided the set of resources available to the user, and the Soft Constraints Hypothesis helped analyze the strategies used in different environments.

Before proceeding, it is important to note that recurrent procedural tasks are not in general complicated. Although there may be quite a few of them in current software applications, what makes them generally unappealing is the fact that even though they are key constituents of more complex tasks, they are not particularly interesting to work on. If working on small repetitive tasks is a prescription to shrink motivation, the fact that they do not make much sense in isolation helps them become the perfect recipe for boredom. Skinner was able to witness this undesirable effect in Programmed Instruction (Driscoll, 1994).

One solution to this dilemma has been to use games in the same way a cherry flavor is used to disguise a medicine's bad taste. By creating meaning around seemingly unrelated tasks, a game can increase learner engagement. The next section presents a small review of games as instructional tools and proposes that games will be useful instructional strategies only if the learning task is seamlessly embedded into the gameplay and reinforces the sense of agency of the learner/player in the game world.

Games as Instructional Tools

With the advances in computer graphics and the success of video games, there has recently been a renewed interest on games as instructional tools. In spite of this interest and almost sixty years of research, evidence-based generalizability of their effectiveness in achieving learning objectives is still debated. In 1981 Bredemeier and Greenblat (Bredemeier & Greenblat, 1981) summarized the lack of empirical results stating that:

> We do not have yet (1) a theoretically based taxonomy of games with (2) clear theories about (a) what aspects of them are expected to have (b) what sorts of distinct effects (c) on what sorts of students (d) for what reasons. Until these tasks are addressed, we shall probably continue to see results of investigations about "effectiveness" that are inconsistent, ambiguous, and nondefinitive in support or revision of widespread 'impressions'. (p. 327)

This statement still rings true in spite of an extensive body of literature that has contributed to our understanding of games. A brief summary is presented next.

Research on Games

Research on games has been rather inconclusive. Studies have unfortunately suffered from the same problems of more general research in Educational Technology. In a review of Ed Tech research carried out by Driscoll and Dick (Driscoll & Dick, 1999), results showed that many articles described software or tools, but "data to verify the tool or software effectiveness or to support recommendations made for design practice were noticeably absent from these articles" (p. 12). Additionally, of the 22 experimental articles that somewhat fitted the Culture Four criteria, none indicated the use of a systematic design process to develop the instruction.

A similar situation can be found in game design. One article reported that students were unable to replicate some games even with the help of the designers because no systematic methodology had been used and no records existed (Meadows, 2001). Descriptive articles are valuable contributions, but empirical studies are required to create a body of knowledge that will eventually lead to a theory.

Researchers have been criticized for concentrating their efforts on comparing apples (traditional classroom instruction) and oranges (video games) rather than trying to figure out the uses and possibilities of these tools (Allen, Otto, & Hoffman, 2003; Bredemeier & Greenblat, 1981). Regrettably many state that there is a high number of flawed or poorly designed studies (Dillon & Gabbard, 1998; Gosen & Washbush, 2004;

Remus, 1981) which undermine the knowledge that can be obtained from them (Butler, Markulis, & Strang, 1988).

There have been several attempts at establishing the strengths of games from a learning perspective, elucidating relationships between learning variables and games, and consolidating the results obtained in several studies –certainly not an easy task! – (Bredemeier & Greenblat, 1981; Dempsey, 1994; DeNike, 1976; Randel, Morris, Wetzel, & Whitehill, 1992; Williams, 1980; Wolfe, 1985).

Greenblat (1973) suggested that games are better suited to learn concepts, procedures and principles rather than facts. However, some games in the organizational domain have actually succeeded at teaching plain facts (Prensky, 2004), although this is not supported by experimental data.

Dempsey et al. (1994) reviewed 91 articles (1982 to 1994), and Randel et al (1992) reviewed 67 articles (1984 to 1991). High levels of motivation, usually measured through self-report, and higher retention of the material learned seem to be supported by many studies. The rest of the outcomes are mixed. Dempsey et al. (1994) offer a list of lessons that can be used by instructional game designers. A recent meta-analysis found positive gains for games and interactive spaces but with moderating factors such as age and learner control (Vogel et al., 2006).

There have also been a significant number of attempts to deal with the evaluation of games and simulations, as well as with issues of validity, both internal and external (Braskamp & Hodgetts, 1971; Dickinson & Faria, 1997; Dukes & Waller, 1976; Feinstein & Cannon, 2002; Gosen & Washbush, 2004; Miller & Leroux-Demers, 1992; D.

Norris, 1986; S. Norris & Snyder, 1982; Peters, Vissers, & Heijne, 1998; Rolfe, 1990, 1992; Ruben & Lederman, 1982; Wolfe & Roberts, 1986). The various divergent approaches led Feinstein and Cannon (2002) to state that "the literature is so cluttered with terms and concepts that it is hard to build a coherent program of validation research" (p.437).

Bredemeier and Greenblat (1981) called attention on two important issues: 1) game ability and academic ability may not go hand in hand, and 2) even if students consistently report much higher level of motivation in gaming environments, it is important to further analyze the reasons behind. As they clearly stated, interest depends on several factors such as learning style and personality fit, which should be taken into account in research studies. Also, even if games succeed at increasing intrinsic motivation, cognitive results or the achievement of instructional objectives do not necessarily follow (Feinstein & Cannon, 2002; Martens, Gulikers, & Bastiaens, 2004).

Regardless of the difficulties, it is commonly believed that games do have strengths that are worth exploring in instructional and training interventions:

1. Games are seen as fun as opposed to the serious nature of traditional interventions. This seems to help people see challenges with a more favorable and playful attitude (Coulson, 1991).

2. Although games can be abstract, they can engage the power of myth and narrative in a way that no other computer tool can. In particular, the gaming environment can provide a cohesive framework by virtue of the narrative and spatial landscapes it creates (Jenkins, 2004).

3. The game's 'magic circle', which is the world inhabited by the players and completely separated from the 'real' world, is one of safe unpredictability (Apter, 1991; Salen & Zimmerman, 2004). This can be seen as an opportunity for experimentation without the serious, long-term consequences normally associated with real life. It may also help reduce the anxiety learners experience at the beginning. Between the inhabitants of this magic circle, strong communities can be built (Gee, Squire, & Steinkuehler, 2005; Steinkuehler, 2006).

4. Computers enable a level of richness, interactivity, networking, scaffolding, feedback and recapitulation never seen before in games. They offer the possibility of apparently non-linear environments in spite of the difficulty of achieving a truly flexible interactive experience (Crawford, 2005; Murray, 1997; Spiro, Feltovich, Jacobson, & Coulson, 1992).

Elements of Games

Creating a game that skillfully harness these benefits, is fun and also helps participants achieve specific learning objectives is not only an art but also a considerable challenge. Instructional games are generally regarded as 'not-fun' and 'uncool'. That is one of the main challenges of game design at any level: the intangible element of 'fun'. Salen and Zimmerman (2004) consider the experience of fun or play from different perspectives but there isn't a single formula to create a satisfying interactive experience. From an instructional perspective, Koster's (Koster, 2004) definition of fun might prove more enlightening: the feeling that is present while we learn and master the pattern present in the game, which is exactly what is wanted for automatic operations commonly used in software applications.

Garris and Ahlers (2002) propose a set of elements that must be present in an instructional game to be successful: fantasy, rules/goals, sensory stimuli, challenge, mystery and control. This taxonomy is similar to the typology of pleasure proposed by LeBlanc (as quoted in Salen & Zimmerman, 2004) from the gaming perspective, which includes sensation, fantasy, narrative, challenge, fellowship, discovery, expression and submission. Some of these categories are present in traditional instructional interventions (challenge, fellowship, etc). Some others, however, are not so frequently found: sensation, fantasy, submission and narrative seem to be the ones missing at first glance. Submission is defined as the sense of surrendering to a rule-based system that stylizes actions and behaviors of players in particular ways and maintains the player 'safe'. In spite of their significance, the elements of fantasy and metaphor are largely optional in formal instruction and their use depends on the style of the instructor or designer. The positive impact of fantasy on motivation and learning is supported by previous research (Malone, 1981; Malone & Lepper, 1987; Parker & Lepper, 1992; Cordova & Lepper, 1996).

Malone's (1981) research is probably the most frequently study quoted in favor of instructional games. For Malone, successful instructional games require three components: challenge, fantasy, and curiosity. Fantasy can be either intrinsic if it is related to the use of the skill, or extrinsic if it is not. Malone suggests that the effectiveness of the fantasy used will depend primarily on the target audience, but he also believes that intrinsic fantasies are more interesting and instructional than extrinsic ones. Recently Nintendo has created a successful game, Brain Age, which does not use fantasy. With only a few gaming resources such as a time limit and a score, Nintendo has managed to make arithmetic operations and other exercises to develop the brain highly addictive.

There is no fixed recipe to create a good game, either commercial or instructional. From the simplicity of Tetris to the visually stunning environments of the most recent games, the mix of elements results from the skills of the designer. A strong sense of agency for the player and a careful balance of resources are probably the most important guidelines in game design.

Games as Distributed Information Environments

In modern video games, particularly in role-playing games (RPGs), there is a large amount of information that is made available to the player on the screen and that has to be managed to achieve the different goals. For example, in any of the releases of Age of Empires there are several panels that show the type of character that the player is controlling, the type of constructions that are available, the technologies of the current age, etc. This information is not all presented simultaneously but can be either retrieved depending on what object is selected or accessed with hot keys. There is significantly more information at any given time than what is normally found in instructional environments. In spite of the higher cognitive load that is required to keep track of the changes and evolution of the virtual world, if the game is properly designed and balanced, players cope with the increased cognitive load, deriving excitement out of the challenge (Ang, Zaphiris, & Mahmood, 2007).

Games as Embodied Experiences

Elements of embodied cognition are clearly present in games, as the previously described analysis of Tetris players has clearly shown (Kirsh & Maglio, 1994; Maglio, 1997; Maglio & Kirsh, 1996). Take for example the new Nintendo Wii game console. Using a highly innovative control that senses motion, the player is able to replicate certain actions normally carried out in the real world. You can play a tennis match in the comfort of your living room. The console is capable of matching the arm swing and transfers the action to a game character on the screen. Rather than a disconnected experience between the player and the avatar, an assortment of visual and aural cues creates a very immersive experience. The sound of the racket hitting the ball, which is actually made by the control itself, the visual cues present on the display, and most importantly the fact that the individual uses a very similar arm motion to that required in a tennis court gives an uncanny sense of playing a real tennis match. As technology develops and new forms of interaction appear, games will become even more embodied experiences (Grodal, 2003), the cognitive, perceptual and motor skills required carefully intertwined; understanding how they interact can inform the design of better games that will achieve their objectives, be it fun or learning, or both.

Summary

Inconclusive research on the instructional benefits of games has not prevented instructional designers from keeping them in their toolboxes. Gamers and learners alike have criticized instructional games as many fail to live to the expectations both from the point of view of gameplay and instructional objectives. Perhaps our understanding of games can improve as we are able to increase the granularity of our analysis.

Skipping the analysis and design phase by using existing games is not likely to work. By embedding instructional tasks in well-known games, the original successful flow of the game is often disrupted. Additionally, the disconnection between the learning task and the gaming environment disrupts the sense of agency of the player.

The feeling that an action in the gaming world determines the course of the game and impacts future developments is essential to create an engaging experience (Murray, 1997; Salen & Zimmerman, 2004). It is proposed here that the instructional value of a game will be higher whenever the learning tasks are part of the gameplay and failure to complete them will make it impossible to achieve the goal. Such impossibility should not arise from artificially stopping the game, but from the fact that those tasks are the gamers' tools for action and vehicle of agency. Integrating the learning scenario, in this case the interface, with the game environment was the main directive for the design used in this research. By seeing games as spaces where information is distributed and operations take place at the 'embodiment level', we may gain a better understanding as to what happens when the player/learner is interacting with the game. This in turn can inform what elements will be beneficial and how we can shape the interaction to achieve the instructional goal.

Research Design Rationale

Research on distributed cognition has shown that rather than fully memorizing the system's functionality, users in interactive environments seem to follow cues from the display in order to perform an appropriate action with the system. These actions are non-deliberate operations at the embodiment level (Ballard et al., 1997; Gray & Boehm-Davis, 2000; Gray & Fu, 2004; Gray et al., 2006) and can be cognitive, perceptual or motor. Which operation will be selected is determined by a cost-benefit analysis of the available resources; this cost of accessing information is measured in milliseconds (Gray & Boehm-Davis, 2000; Gray & Fu, 2004; Gray et al., 2006). The studies reviewed here used simple tasks that participants had either encountered before, such as programming a VCR (Gray & Sabnani, 1994), or could learn quickly like puzzles such as the Blocks World Task (Ballard et al., 1997). Rather counter intuitively, high information cost led to better performance and retention (Gray & Fu, 2004; Morgan et al., 2007; Waldron et al., 2007).

This research extended these studies and considered interaction at the embodiment level in a simulated Flash interface while learning basic animation tasks with different resources available. In addition, interaction in an instructional game using the same interface was analyzed. Other factors were believed to play a role in how users interacted with the environments. These factors are considered next.

Additional Factors

Aptitude variables such as self-efficacy and anxiety have been shown to influence learning and performance, but efforts to pair learners with instructional interventions based on individual characteristics have had limited success (Park & Lee, 2003). Nevertheless, it is possible that factors different from the characteristics of the digital environment can have an impact on how a learner behaves in a digital space and how much he or she benefits from this particular instructional strategy. An initial survey was used to determine some aptitude variables, more specifically learning strategies and motivational beliefs, computer self-efficacy, personality traits and player type of the participants. This was of special interest when looking at games, as previous research has been criticized for not considering these factors (Bredemeier et al., 1981).

Modified Motivated Strategies for Learning Questionnaire (MSLQ). Individual's motivation and learning strategies may play a role in the way a person behaves and profits from online training environments. This was an issue of interest in this study. The Motivated Strategies for Learning Questionnaire (Pintrich & De Groot, 1990) is a reliable, adaptable tool that has been successfully used across different contexts such as cooperative learning, multimedia design and computer-based instruction (Garcia, 2005).

The questionnaire has two main sections, Motivational Beliefs and Self-Regulated Learning Strategies (Pintrich & De Groot, 1990). The Motivational Beliefs section measures self-efficacy, intrinsic value, and test anxiety. The Learning Strategies section assesses cognitive strategy use and self-regulation. The instrument has evolved since to a total of 81 items (Garcia, 2005) but a simplified version was used here with a total of 30 items. The modified instrument referred to software applications rather than the original focus on 'the class'.

The MSLQ has been used in several studies to determine the relationship between its scales and the use of technology (Hargis, 2001; Liu, 2003; McManus, 2000; Miltiadou, 2001; Polleys, 2000). Hargis (2001) found no relationship between any of the instrument's scales and the use of the Internet either within an objectivist or constructivist framework. McManus (2000) suggested that individuals with high self-regulation skills preferred non-linear environments, and those with lower self-regulation skills preferred linear environments. Miltiadou (2001) found that task value was a predictor of user satisfaction in online courses. Finally, several relationships were found between personality (using the Myers-Briggs instrument) and self-regulation (Polleys, 2000).

A high correlation was expected between self-regulation and computer self-efficacy (see below). In line with McManus (2000), high self-regulation was expected to negatively correlate with reported overall satisfaction in all environments because exercises in all conditions progressed in a linear manner. Relationships between personality and self-regulation were also expected, in line with Polleys (2000).

Computer self-efficacy. A factor that may partially influence how easy or difficult it is for a learner to deal with a new software application is computer self-efficacy. Computer self-efficacy "represents an individual's perception of his or her ability to use computers in the accomplishment of a task (...), rather than reflecting simple component skills" (Compeau & Higgins, 1995). Some learners with higher computer self-efficacy might be able to cope with the cognitive load imposed by learning tasks, but those with lower computer self-efficacy will most likely have difficulties.

In a study evaluating online instruction, Tsai (2004) suggested that people with high computer self-efficacy perform better and are more satisfied regardless of the way information is presented. Tsai however used Compeau and Higgins' scale which only contains 10 items, and although construct validity is supported more studies are needed. The most frequently mentioned scale to measure computer self-efficacy was proposed by Murphy et al (Murphy, Coover, & Owen, 1989), which includes 32 items and three main scales. Later work has validated the tool but with four scales instead of 3 (Torkzadeh & Koufteros, 1994; Torkzadeh, Koufteros, & Pflughoeft, 2003). This instrument has its limitations, too, as some of its items refer to outdated technologies such as floppy disks, and includes questions about mainframes, which Torkzadeh et al. (2003) acknowledge as currently irrelevant.

A third instrument has been proposed which includes 30 items plus a set of preliminary questions (Cassidy & Eachus, 2002) which is claimed to have higher internal and external reliability than the other instruments mentioned above, as well as construct validity but has only one study to support it. Because of the outdated nature of instrument described by Torkzadeh et al. (2003), the Cassidy and Eachus' instrument was used here with no modifications.

Personality traits. Some support has been found for the influence of 'offline' personality on online behavior in games. Van Meurs (Van Meurs, 2007) found that personality traits, as measured by Saucier's (Saucier, 1994) shorter version of Goldberg's (Goldberg, 1992) Big Five markers, supported Bartle's (Bartle, 1996) player types in

Multi-User Dungeons (MUDs). A more detailed explanation is provided in the next section.

The Big Five factors are extraversion, agreeableness, conscientiousness, emotional stability, and intellect, openness or imagination. As mentioned earlier, personality traits seem to be related to self-regulation as well (Polleys, 2000). The instrument was used here without modifications.

Player type. The player type was only applicable to the gaming environment. There are no standard instruments to determine what player style people have, if any, although there are several typologies. However, for the specific case of MUDs (Multi-User Dungeons) four types of player are widely accepted: Socializers, Killers (or Griefers), Achievers, and Explorers (Bartle, 1996). The names are pretty much self-explanatory. Socializers like virtual worlds for the opportunities to connect with other people. Killers or Griefers value the possibilities of attacking or inflicting damage to other players. Achievers spend most of their time achieving goals such as victory in campaigns or accumulation of wealth. Explorers enjoy discovering the virtual world, even going further to understand its inner workings and possibilities. Van Meurs (2007) has added the Role-Player, who naturally enjoys the chance of expression through different characters.

Van Meurs (Van Meurs, 2007) used this classification and extended it to include the Role-Player in a validated instrument. He found not only that player types seem to match those proposed by Bartle (1996), but also that Saucier's (1994) personality markers were able to explain a great deal of the player groups. Player type was also related to the type of MUDs people play.

Van Meurs (2007) found that conscientiousness and extraversion were important factors defining the achiever, whereas Intellect was extremely significant in role-players. Rudeness and moodiness defined the Killer or Griefer, agreeableness and extraversion were the defining characteristics of Socializers, and intellect and conscientiousness were for the Explorer. It can be seen that the results make quite a lot of sense and support Bartle's (1996) classification.

Another typology is proposed by Robin Laws (Laws, 2002) includes the following types: Power Gamer, Butt-Kicker, Tactitian, Specialist, Method Actor, and Casual Gamer. There is a tool available to determine what type of player a person is based on Laws' description of each type, but it has not been validated or used in any study.

Van Meurs'(2007) adaptation of Bartle's (1996) MUD player types was used, understanding that there is no established and accepted player typology or instrument to determine it. MUDs also differ from the digital environments considered here. Despite the differences, it was thought that the relationship between player type, strategies selected in interactive environments, and performance could provide interesting insights into resource use patterns.

Hypotheses

In line with the literature reviewed here, the following hypotheses were proposed, grouped by the main variable being evaluated.

Hypotheses Regarding the Training Environment

H1: *Online courses that do not offer practice of basic drills will not be instructionally effective, even if the nature of the task is very simple. This will be evidenced by a lower number of tasks completed, higher time to completion, and lower visio-spatial recall.*

H2: *Participants in the low-cost resources environment will finish the tasks faster but will evidence the lowest level of performance, as predicted by the Soft Constraints Hypothesis (Gray & Fu, 2004; Gray et al., 2006).*

H3: *Participants in environments where no resources are available during the exercises will take longer to complete the tasks but will evidence the highest level of performance (Gray & Sabnani, 1994; Morgan et al., 2007; Waldron et al., 2007).*

H4: *Participants in environments where resources are available but have a high access cost will exhibit intermediate completion times and intermediate performance levels (Gray & Sabnani, 1994; Morgan et al., 2007; Waldron et al., 2007).*

H5: *When embedded in the unifying context of a simple abstract game, the instructional effectiveness of software application training environments based on the training-wheels approach and distributed information resources will be at least equal to the high information access cost environment, but will result in higher satisfaction levels. Increased use of resources will not only imply a higher retrieval cost in milliseconds, but will also negatively affect the game score as time is also used as a valuable game resource. Assessment results are expected to be higher than those for the environment* *where the resources will have the same cost but no game context, and equal or higher to the No Resources environment.*

H6: *The game environment will result in a higher number of repetitions per task than any other environment.*

Hypotheses Regarding Computer Self-Efficacy

H7: *Individuals with higher computer self-efficacy will report higher satisfaction levels regardless of the instructional intervention used, as reported by Tsai (2004).*

H8: *Individuals with higher computer self-efficacy will perform better regardless of the instructional intervention used, as reported by Tsai (2004).*

Hypotheses Regarding Player Type

H9: *There will be variation in strategies in the gaming environment which will be influenced by the player type. Achievers are expected to obtain the highest scores in the least amount of time, in line with their desire for accomplishment in the game environment. Explorers are also expected to find their gaming style supported and report high game scores, with less emphasis on time. In consequence, if the Soft Constraints Hypothesis (Gray & Fu, 2004; Gray et al., 2006) applies here as well, resource use is expected to be the lowest for achievers, followed by explorers. Socializers, Killers, and Role-Players will have the highest use of resources.*

H10: *Differences in satisfaction levels between different player types will be observed in the gaming environment. Socializers, Killers, and Role-Players are expected to report low levels of satisfaction because their preferred gaming style is not supported.*

Explorers are expected to report moderate levels of satisfaction, and Achievers the highest satisfaction levels of all player types.

Hypotheses Regarding Self-Regulation

H11: *High self-regulating individuals will report lower satisfaction and low self-regulating individuals will report higher satisfaction, regardless of the environment due to linearity, as found by McManus (2000).*

CHAPTER 3. METHOD

The purpose of this chapter is to describe the design process that guided the creation of the computer training environments, the instruments and techniques used for data collection, the participants, and the methodology used in this study. Because the entire study took place online, extensive prototyping was necessary to make sure no problems were encountered during the final sessions. For the purpose of clarity, the description of the study has been divided into four main sections: general introduction, prototyping, phase I, and phase II. The general introduction section gives an overview of the entire study. The Design and Prototyping section will describe the computer components specifically created for this study, and how the design process prompted some modifications to the study and the hypotheses. The following section, Phase I, describes the method and instruments in the order used with the first group of participants. Due to difficulties encountered in this first stage, a second phase was conducted and final modifications were made. These are detailed in the final section Phase II.

Overview

This section presents a brief overview of the study. The first step was to select the instructional objective of the research. Rather than choosing a task unrelated to a real educational context, basic animation concepts in Adobe Flash were selected. Adobe Flash is a software application for the web which has transitioned from a simple animation tool into a very large multimedia authoring environment, widely used in entertainment and education. There were several reasons supporting this decision:

1. Flash is a real software application so the study results reflect a real learning process and not a hypothetical situation.

2. Flash is widely used so it was assumed that people would be motivated to participate in the study to learn the basics of the software.

3. A free full-day introductory Flash course was created to be offered to those participating in the study, relating the participation incentive to the research's objective.

As a broad introduction, the proposed study initially consisted of the following stages:

1. Initial Data Gathering: In this phase an online survey gathered information including demographics, motivational beliefs and learning strategies, personality traits, and player type. Commercially available tools were used for this purpose. The instruments will be described more in detail in Phase I.

2. Instruction: Each learner completed a short online course on animation basics in Flash. The course followed the same format as frequently found in online software training. The lessons were followed by two simple exercises. The instructional material was initially built using Captivate.

3. Practice Conditions and Assessment: Learners were going to be randomly assigned to one of five conditions (Control, No Resources, Low-Cost Resources, High-Cost Resources, and Game). The last four gave people the chance of practicing what they learned during the Instruction stage. All participants were going to be assessed to determine their final level of competence and satisfaction. Both the practice and assessment modules were custom-built in Flash 8, and the satisfaction survey was

created with the commercial tools used in the initial data gathering. Figure 1 shows a flow chart of the preliminary design.

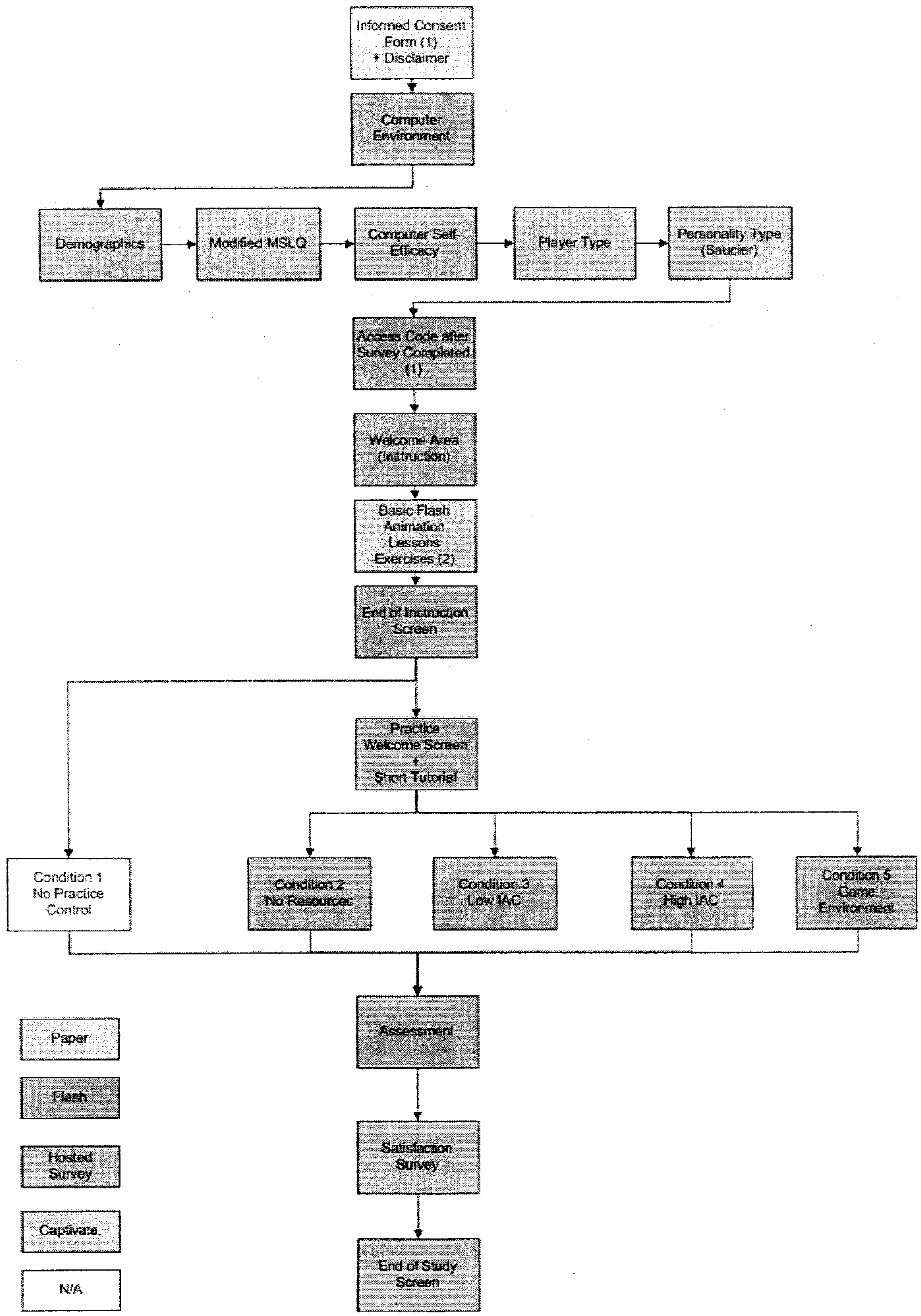

Figure 1. Preliminary Study Design

Design and Prototyping

Since all elements used for instruction, practice, and assessment were specifically built in Flash 8 for this research, it was necessary to build prototypes of all components and test them before using them in the study. This was not only necessary from the usability point of view. An iterative collaborative design process is also at the center of design-based research in education (Barab & Squire, 2004), giving validity to the tools used. Rather than simply assuming that the computer tools will work, they are extensively tested to make sure design errors will not contaminate the results. There were three main computer modules in this particular study: instructional component, practice component, and assessment component. The following subsections present the design rationale behind each component as well as the implementation. The final subsection describes the prototyping process and the effect on the components and the research design.

Instruction

A set of online lessons were created covering the following topics: (a) flash Interface, (b) drawing in Flash, (c) animation concepts, (d) how to animate the fill color of a simple object, and (e) how to animate one shape into another shape

Simple Flash animations refer to changes of an object on the screen. For example, a user can choose to draw a red rectangle, perform a set of operations and end up with a green circle. When the animation is played the user will see a red rectangle transforming into a green circle in a given time. The transformation takes place using the same metaphor as film: a frame-by-frame animation that when played at high speed gives the illusion of motion.

The practice environment was entirely built in Adobe Captivate within a Flash shell. Adobe Captivate produces a simulation by taking snapshots of the screen at key points in time while an expert uses the system. Figure 2 shows a screenshot of one of the demos. A message box with an explanation of the current step can be seen over the software interface. Notice that all commands are present as the software captures what an experienced user is doing.

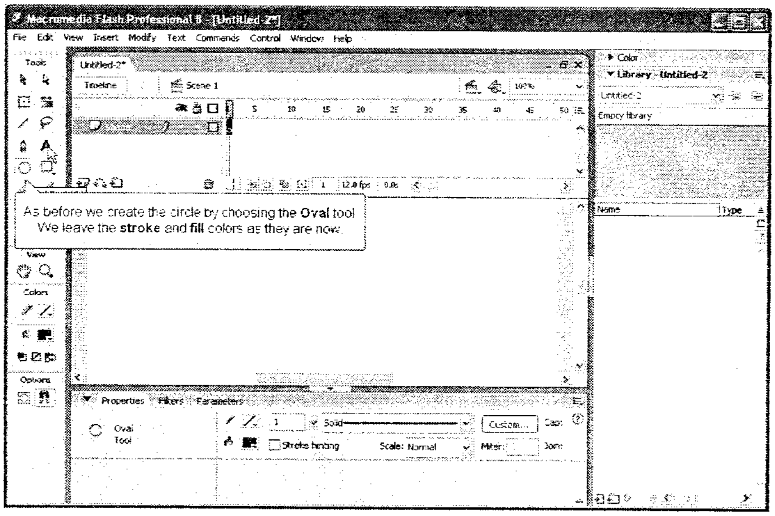

Figure 2. Flash Demo built in Captivate

Two simple exercises were planned at the end of the sessions to review the material. In Captivate's assessment mode trainees have to reproduce the sequence. When the user clicks on the wrong area, he/she gets a warning. If the user's selection is correct, then the sequence keeps playing until the entire exercise has been completed. Figure 3 shows a screenshot of one of the tutorials, where a message box tells the user what to do next.

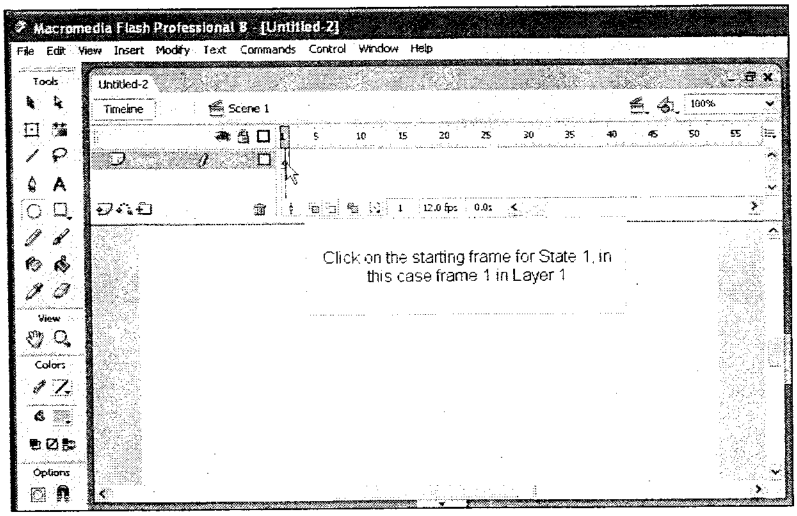

Figure 3. Flash tutorial built in Captivate

For each person participating in the study, the role of the Flash shell was to determine how long the participant took to complete each lesson and each exercise.

Practice Component

Twenty exercises were created to allow some of the participants to practice what they learned during instruction. There were 5 groups of 4 exercises, as follows:

1. Exercise 1 to 4: Complete a color animation
2. Exercise 5 to 8: Complete a shape animation
3. Exercise 9 to 12: Full color animation
4. Exercise 13 to 16: Full shape animation
5. Exercise 17 to 20: Full color and shape animation

The main assumption in this study was that users who did not practice would have more difficulties completing the animation sequences as opposed to those who were given the chance to practice. Those who completed practice exercises would benefit in a varying degree depending on the cost of the resources present on the screen: the higher the cost, the better knowledge of the sequences and retention the user would have. In order to test the more specific hypotheses enumerated in the previous chapter, five conditions were initially suggested: Control Group, No Resources Group, Free Resources group, Resources with Cost group, and Game group. In all conditions cost referred to the time it took to access a resource on the screen.

1. Condition 1 – Control group: Participants were not going to have any additional practice exercises and would proceed directly to assessment.

2. Condition 2 – No Resources: No resources were going to be available while executing the practice exercises.

3. Condition 3 – Low Cost: Resources were going to be made available at low cost while completing the exercises.

4. Condition 4 – High Cost: The same resources as condition 3 were going to be made available while practicing but at an intermediate cost, between condition 2 (very high) and condition 3 (low).

5. Condition 5 – Game: The resources and exercises in conditions 3 and 4 were going to be embedded in a game environment. Cost in this environment was affected by the time limit imposed by the game.

Note that conditions 2, 3, 4, and 5 all had the same exercises. The only difference was the environment, the resources made available to the user, and the cost of accessing them. Exercises not only included complete tasks but also part-tasks in order to help the user build partial plans once certain steps had been completed.

Important measures included: (a) time to complete each exercise, (b) number of times the exercise was repeated, (c) number and type of errors made, (d) where resources were available and had a cost, resources used and number of times each resource was used, and (e) each event was time-stamped.

Additionally, for the Game environment (condition 5) only: (f) number of times each level was tried, (g) time to successfully complete each level, and (h) game score.

Errors were coded according to the following codes: (a) E1 - Wrong icon, does not match the instruction (if any), (b) E2 - Attempting to create a Keyframe in the wrong frame, (c) E3 - Creating a Keyframe instead of a Blank Keyframe, (d) E4 - Creating a Blank Keyframe instead of a Keyframe, and (e) E5 - Applying the wrong tween

Treatment conditions: Design rationale. The practice environment was designed based on distributed cognition and the idea that salient task features (Pollard, 1982) guide the flow of information between the display and the user (Manktelow & Jones, 1987). The software simulated the Flash interface but only the necessary commands required for the task were available, in line with the training-wheels concept reviewed in the previous chapter (Carroll & Carrithers, 1984). This design automatically reduces the number of errors that the learner can make. The interface occupies roughly 75% of the environment real estate as recommended by Gellevij et al. (Gellevij et al., 1999).

As noted in preceding sections, research suggests that the location of the cues on the interface is also relevant (Ehret, 1999). Blackmon et al. (Blackmon, Kitajima, & Polson, 2005) propose that the user first parses the screen into smaller areas and then selects the one that he/she thinks is likely to complete the task at hand. Visual-spatial elements are of particular importance for speed and accuracy when compared to text elements (Hinesley, 2005). For this reason the original icons and menus were preserved and the screen was parsed in the same way as the original software. Figure 4 shows the real Flash interface and the simulated Flash interface. As can be seen from the figure, detail was reduced to a minimum in the simulated interface, leaving only the necessary icons and menu items visible.

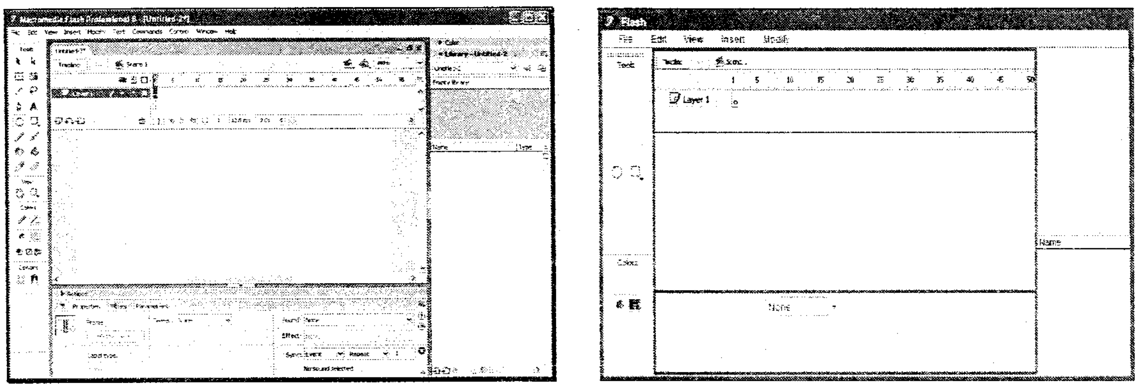

Figure 4. Complete vs. Simulated Flash interface

The design of this environment was also based on the flow of information premise (Howes & Payne, 1990; Payne, 1991). Instead of trying to replicate how the software actually works, the system reproduced only the changes on the interface as the task was carried out. As in the original software, not all actions had an immediate effect on the display. In order to provide some sort of feedback, all environments displayed a visual cue indicating a correct (green) or incorrect (red) action.

Each environment was preceded by a very basic tutorial. Initially a more extensive set of instructions was going to be presented to the learner, but it was decided instead that in order to minimize the differences between the four conditions, the tasks that were going to be targeted in the tutorial would become very simple exercises at the beginning of the practice session.

Resources. In all conditions some distributed information resources and interaction strategies outlined in the Distributed Information Resources Model (Wright et al., 2000) were made explicit. Resources initially included:

1. The task goal: Unlike a VCR or the Blocks World Task, the final goal was in the form of an animation, which did not provide enough information to allow the user to complete the task. It only gave information about the initial and final states of the animation. For example, the user saw a green circle that transformed into a yellow circle, but had no way of figuring out how long the animation took, a key piece of information required to complete the task.

2. Task Description or Semantic Goal: This panel explained in plain English the task that had to be accomplished. Continuing with the previous example, it read something like: "Create an animation where a green circle transforms into a yellow circle in 2 seconds". Although the information contained here was enough to complete the entire task, it did not say anything about the necessary steps.

3. History/Plan panel: This panel provided the information necessary to complete the task step by step. The user was able to see the entire task plan (plan following) and make decisions based on previous actions (history-based choice). Again, in the previous example, the resources panel had the following instructions: (a) select frame 1, (b) select the Oval tool, (c) choose a color (green), (d) draw the green circle, (e)

select frame 24, (f) create a Keyframe, (g) choose the new color (yellow), and (h) apply Shape Tween.

The following subsections describe more in detail each condition.

Condition 1: Control group. Online courses assume that after a short explanation and a couple of exercises people are ready to use the software and recreate what has been demonstrated. In a certain way, this is possible if the user follows the lesson with his/her own copy of the software and practices many times. It was hypothesized that since users in this scenario were not going to have the opportunity to practice, their performance in the assessment test was going to be the lowest from all groups.

Condition 2: No resources environment. Figure 5 shows a screenshot of the suggested prototype of the first environment. No resources were going to be made available in this case. The user was going to see an initial screen that detailed the steps that were to be followed to complete the task. When the user hit the <START> button, the interface appeared and the participant had to complete the task. Two buttons were available: <REPEAT TASK> or <CANCEL>. If the learner selected <REPEAT TASK>, the instructions panel appeared again and the user had the opportunity to repeat the exercise. If the user hit <CANCEL> then the next task was presented. Notice that in figure 5 the interface is shown simultaneously with the instructions panel for clarity purposes, but once the participant started the exercise the instructions were no longer available. This environment was assumed to be cognitive intensive, as predicted by the

Soft Constraints Hypothesis (Gray & Fu, 2004; Gray et al., 2006) and Information Access Cost (Morgan et al., 2007; Waldron et al., 2007).

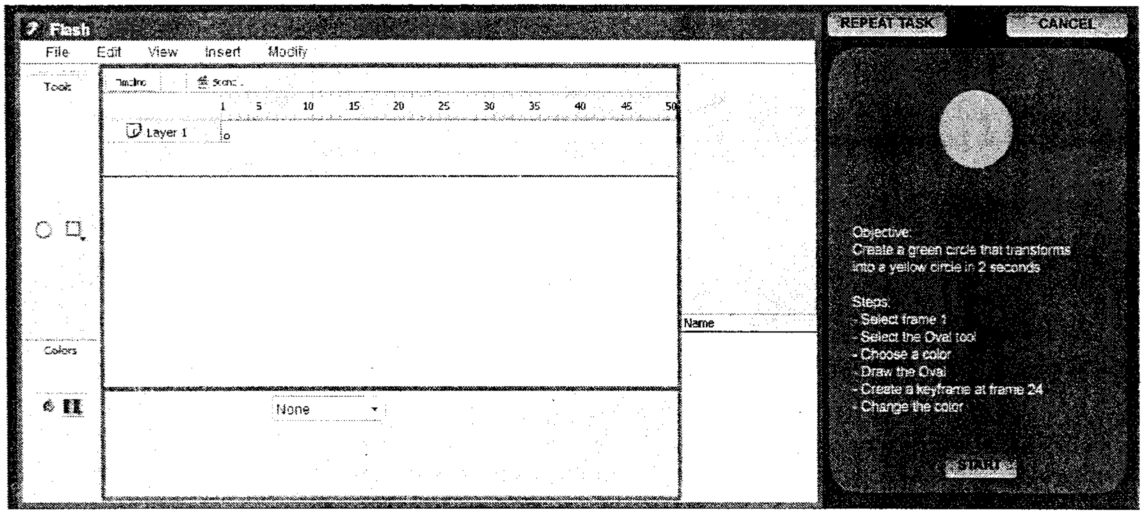

Figure 5. Condition 2: No resources practice environment

Condition 3: Low-cost resources environment. The next two environments, Low-Cost and High-Cost, had identical interfaces (see Figure 6). The information in the sliding panels stayed visible during each exercise once they had been opened. Again, the <REPEAT TASK> and <CANCEL> buttons were available. Notice also in figure 6 that as the user completed each step, the corresponding instruction was checked in the panel. This environment was assumed to be perceptual intensive, in agreement with the Soft Constraints Hypothesis (Gray & Fu, 2004; Gray et al., 2006) and Information Access Cost (Morgan et al., 2007; Waldron et al., 2007). Once the panel was open, the only necessary operation to access information was an eye saccade.

Condition 4: High-cost resources environment. A screenshot of this environment is shown in figure 6. In this case, the sliding panels are had a delay of 1 second before they opened, well within the 1/3 to 3 seconds microstrategy range suggested by Gray and Boehm-Davis (Gray & Boehm-Davis, 2000). This environment was expected to encourage a mix of cognitive and perceptual-motor strategies.

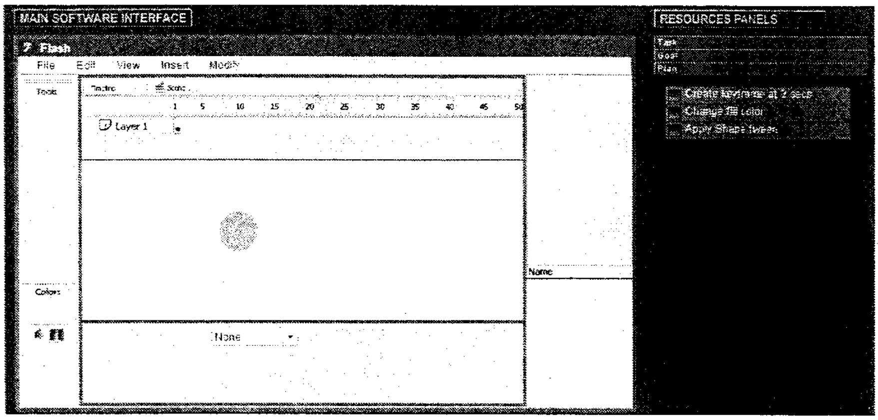

Figure 6. Conditions 3 and 4: Resources with Cost practice environment

Condition 5: Game environment. This environment embedded the interface presented in the previous conditions in an abstract game. The exercises and resources were identical to those in the other environments. In order to keep the game simple enough, each level was made of only five exercises that had a similar goal. For example, level 1 involved only a series of basic operations necessary to complete other tasks. They were simple enough to get a novice started, but the player only had a certain amount of time to complete the 5 exercises. Achieving balance between the relative pressure that the game had to put on the player and the fact that he/she was learning to complete a task was the main design challenge here. Exercises became slowly more complex as the game progressed, like in previous environments. Figure 7 shows the game environment.

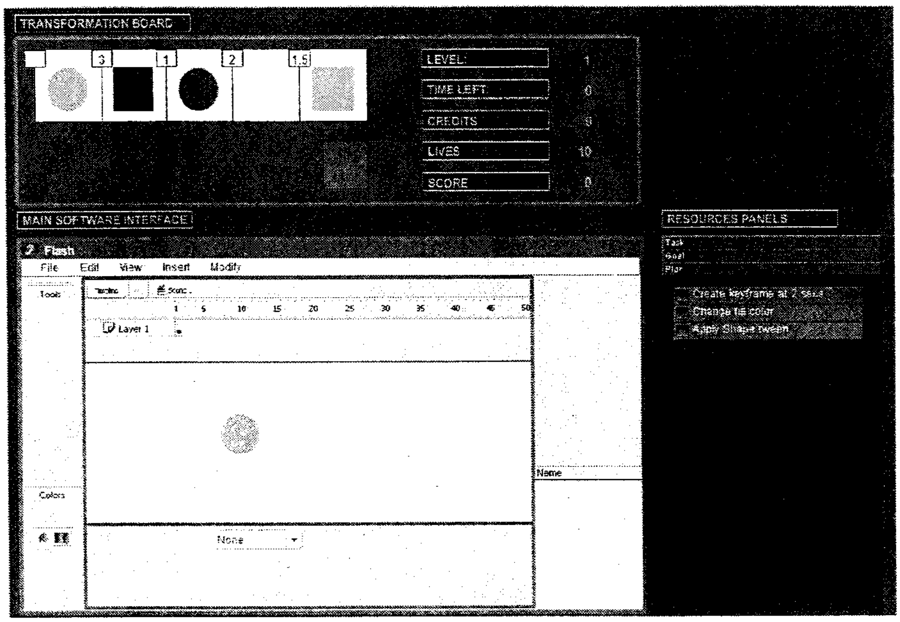

Figure 7. Condition 5. The Game practice environment

Each level had a time limit to give the game some degree of excitement. Time as a game resource was used based on Nintendo's success with Brain Age, where simple mathematical operations are solved within a certain time period. The visual cues that told the user that a correct or incorrect action had been taken were also present. The resources panels were available as soon as the user clicked on them.

Incorrect actions reduced the score, and correct actions increased the score. Each level had a maximum number of incorrect actions (or lives). Completing a level under a certain amount of time gave the user a bonus. A mix of cognitive and perceptual-motor strategies was expected, as well as a different pattern of use to account for the presence of game resources and constraints.

In line with gaming environments, the user had to repeat the tasks as many times as necessary to achieve the goal.

Preliminary interactivity test. Mockups with little graphic detail were built in Flash 8 to test interactivity of the different environments, a common procedure in interactive software development and game design and development (Rollings & Morris, 2004). Condition 2, the No Resources Environment, was found to be too complex to be a realistic training scenario. It is likely that this scenario would have forced participants to memorize more information that any other condition. However, a valid comparison between environments would not have been possible since all resources were displayed simultaneously. Condition 2 was eliminated and more effort was directed to the other environments and the game. The following hypothesis was directly related to condition 2 and was no longer included in the study.

H3: *Participants in environments where no resources are available during the exercises will take longer to complete the tasks but will evidence the highest level of performance (Gray & Sabnani, 1994; Morgan et al., 2007; Waldron et al., 2007).*

The Game environment was modified as the gameplay in the design initially proposed was not particularly interesting. A Tetris-like game was created, with a 4 by 4 board that required the user to create one animation in order to clear each line. Figure 8 shows the new game elements in this environment.

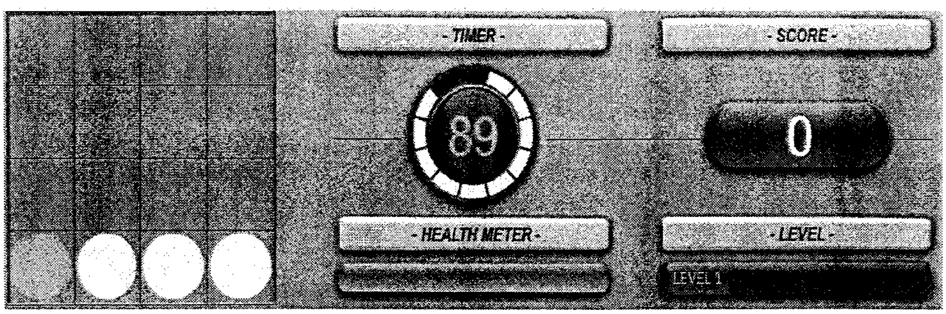

Figure 8. Game resources: board, timer, score, health, level

The goal of the game was to complete all transformations in a level before the four lines were full. If the board filled up, then the user had to start all over again. The player was given a limited amount of time to complete the transformations. The board also provided an additional source of information about the task. In figure 8 it can be seen that the green circle (unique) has to transform into a yellow circle (the other 3) in order to remove the line. Other resources were maintained, namely a health meter (previously 'lives'), a level indicator, and the score.

Prototype test. The final prototypes of the environments for conditions 3, 4, and 5 were tested in the computer lab where final data collection was conducted. Four users volunteered to test the software. One user was assigned to the Free Resources Environment, one user to the Resources with Cost Environment, and two users to the Game Environment. The latter was considered more complex and thus it was decided that two users should test the environment to make sure no problems would appear later in the data collection phase. A think-aloud protocol was used (C. Lewis & Reiman, 1994), along with four cameras to track user behavior. The tests took place on G5 machines, the same computers used later to conduct the study. The researcher was with the users during the lessons and initial exercises of the practice session, but allowed them to work on their own to complete the practice exercises. The researcher remained in the room, however, in case any problems appeared.

This initial test uncovered several problems with the training environments. Although users liked the Captivate lessons, they were confused by the controls and found the pace too slow. Two of the users opened the real Flash application to try to understand the content. Had this happened in the final study, the results would have made no sense at all because an additional factor (the real software) would have been introduced in some cases.

As a result of the previous observation it was decided that the computers used in the final study would not run the Flash application. Additionally, the Captivate animation lessons were replaced by five Camtasia videos that allowed for narration as well as visual aids. A Camtasia video is like a class on video. This format is widely used in software training, as can be exemplified by the courses found on Linda.com, an official training provider for Adobe products.

As mentioned before, the two exercises following the lessons were built with Captivate. Captivate creates software simulations that allow the user to replicate a sequence of commands on the screen. However, users quickly discovered that they could click on different areas on the screen, get hints and finally find out where they had to click next. All four users adopted a 'discovery' approach by clicking anywhere to get hints and proceed accordingly. There was no way to determine if the user had completed the exercise because of skill or simply because the hints guided him/her. Additionally, there was no way of determining the sequence of commands executed by the user, or the errors made. Furthermore, the final exercises in the assessment section, which measured the time required to find a command on the screen, did not accurately reflect the user's capacity to complete the exercise.

To eliminate these issues the Captivate exercises after the initial lessons were eliminated, leaving the Control Group with no practice exercises at all. In the assessment section, the Captivate exercises were replaced by exercises directly on the simulator. This ensured a fair comparison of skills (practice AND assessment were conducted in the same environment). It also allowed the researcher to track the exact sequences of each user, and most importantly, the errors made.

Common to all three environments was the lack of a feedback panel. Each environment had a feedback bar to tell the user whether the action was correct (it turned green) or incorrect (it turned red). However, no information was available as to what kind of error was made and users did not find that very helpful. A feedback panel was included below the Task, Target Animation, and History/Plan panels that explained the error made in addition to the visual indication (green/red).

Users also had difficulties understanding how to complete the initial task in the simulator, and how to use the system. Horsky (Horsky, 2005) analyzed distributed cognitive resources and suggested a distinction between internal and external resources, as well as domain- and system- related. There is a clear difference between experiencing technical difficulties with the environment itself, and having difficulties to complete the Flash animation task. The four panels (Task, Target Animation, History/Plan, and Feedback) were all related to the task at hand but there was no information about the environment.

As a first step, a thorough tutorial was created in Camtasia for all environments. The low-cost and high-cost environments had the same tutorial, and the game environment had identical content but included additional information about the game elements and gameplay.

Secondly, three help buttons were added to the screen: Back to Tutorial, Practice Environment Help, and Exercise Demo. The two first buttons were related to the system itself as the tutorial focused on the training environment and not on animation concepts. The exercise demo showed a Camtasia video that explained how to complete the first exercise, in case a participant was too confused to know how to proceed. The Game environment had an additional fourth button providing information about game elements. Figure 9 shows a sample of the system help provided in the low-cost and high-cost environments.

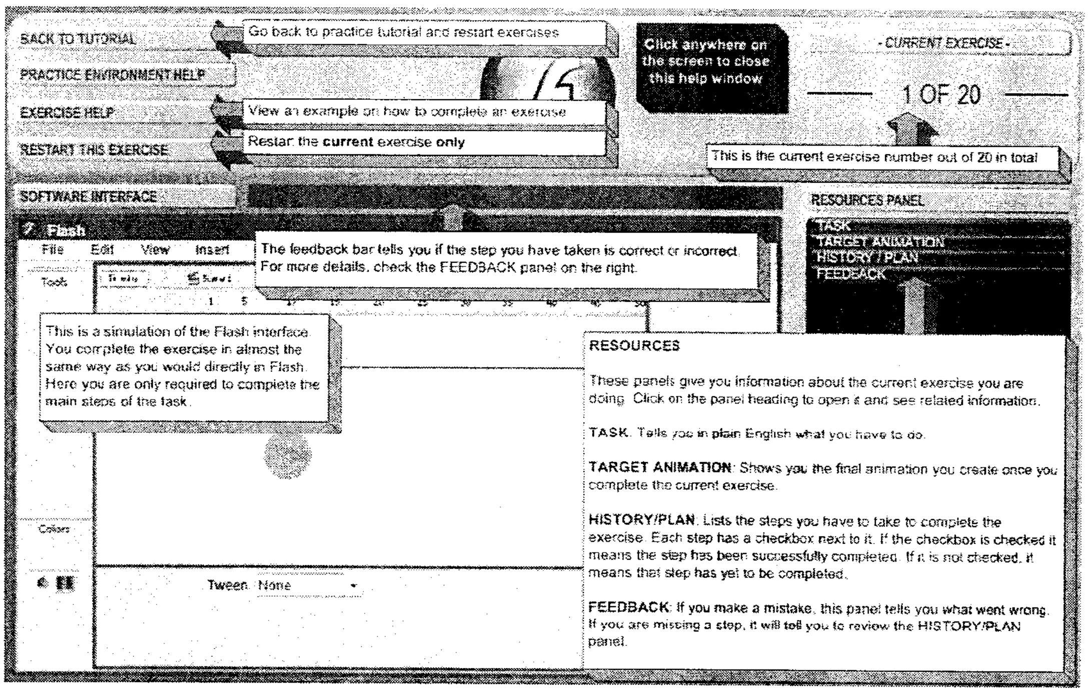

Figure 9. Low-Cost and High-Cost system help

Summary

Initial Captivate lessons were replaced with Camtasia videos, and the two preliminary exercises were removed. A thorough tutorial was included at the beginning of each practice session. Condition 2, No Resources, was eliminated because it did not reflect a realistic training environment, was too cumbersome to use and did not allow for direct comparison with other environments. A feedback panel was added to all remaining environments, as well as help buttons for the system and the task. A counter showing how many exercises had been completed of the total 20 was included in the low-cost and high-cost resources environments. The game was modified to provide better gameplay. In the final assessment section, the Captivate exercises and the command finding tasks were replaced for exercises on the Flash simulator to fully track the command sequences and errors made. Figure 10 shows a screenshot of the low-cost and high-cost environments, which were visually identical. Figure 11 shows a screenshot of the game environment. Notice that the low-cost and high-cost environments, and the game environment shared the Flash simulator and the Target Animation, Task Description, History/Plan, and Feedback panels, with the upper game panel being the only difference between them.

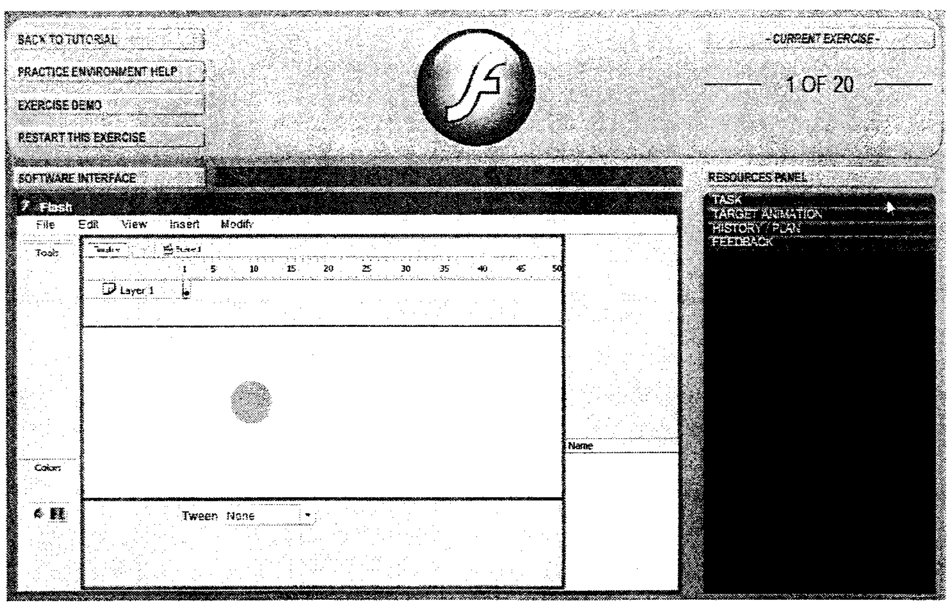

Figure 10. Low-Cost/High-Cost environment

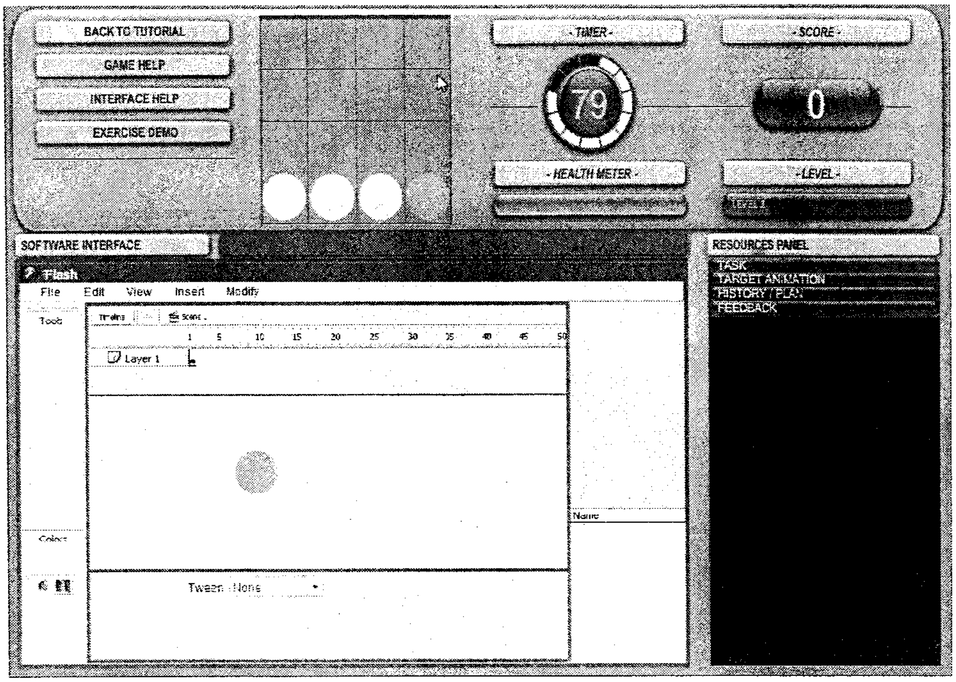

Figure 11. Game environment

Revised Hypotheses

Since a few elements in the study changed, the hypotheses were revised as follows:

Hypotheses Regarding Training Environment

H1: Slightly modified to reflect change in assessment – *Online courses that do not offer practice of basic drills will not be instructionally effective, even if the nature of the task is very simple. This will be evidenced by a lower number of tasks completed and higher time to completion.*

H2: Same – *Participants in the low-cost resources environment will finish the tasks faster but will evidence the lowest level of performance, as predicted by the Soft Constraints Hypothesis (Gray & Fu, 2004; Gray et al., 2006).*

H3: Eliminated – *Participants in environments where no resources are available during the exercises will take longer to complete the tasks but will evidence the highest level of performance (Gray & Sabnani, 1994; Morgan et al., 2007; Waldron et al., 2007).*

H4: Same – *Participants in environments where resources are available but have a high access cost will exhibit intermediate completion times and intermediate performance levels (Gray & Sabnani, 1994; Morgan et al., 2007; Waldron et al., 2007).*

H5: Reworded to reflect the elimination of the No Resources condition – *Assessment results are expected to be higher for the game than those for the environment where the resources will have the same cost but no game context. When embedded in the unifying context of a simple abstract game, the instructional effectiveness of software application training environments based on the training-wheels approach and distributed information resources will be the highest of all environments (and will result in higher satisfaction levels) because of the high cost of the information resources: increased use of resources will not only imply a higher retrieval cost in milliseconds, but will also negatively affect the game score as time is also used as a valuable game resource.*

H6: Same – *The game environment will result in a higher number of repetitions per task than any other environment.*

Hypotheses Regarding Computer Self-Efficacy

H7: Same – *Individuals with higher computer self-efficacy will report higher satisfaction levels regardless of the instructional intervention used, as reported by Tsai (2004).*

H8: Same – *Individuals with higher computer self-efficacy will perform better regardless of the instructional intervention used, as reported by Tsai (2004).*

Hypotheses Regarding Player Type

H9: Same – *There will be variation in strategies in the gaming environment which will be influenced by the player type. Achievers are expected to obtain the highest scores in the least amount of time, in line with their desire for accomplishment in the game environment. Explorers are also expected to find their gaming style supported and report high game scores, with less emphasis on time. In consequence, if the Soft Constraints Hypothesis (Gray & Fu, 2004; Gray et al., 2006) applies here as well, resource use is expected to be the lowest for achievers, followed by explorers. Socializers, Killers, and Role-Players will have the highest use of resources.*

H10: Same – *Differences in satisfaction levels between different player types will be observed in the gaming environment. Socializers, Killers, and Role-Players are expected to report low levels of satisfaction because their preferred gaming style is not supported. Explorers are expected to report moderate levels of satisfaction, and Achievers the highest satisfaction levels of all player types.*

Hypothesis Regarding Self-Regulation

H11: Same – *High self-regulating individuals will report lower satisfaction and low self-regulating individuals will report higher satisfaction, regardless of the environment due to linearity, as found by McManus (2000).*

Table 1 shows the hypothesis and how the study was to support each of them.

Table 1

Hypotheses and Supporting Data

Hypothesis	Support
H1: *Online courses that do not offer practice of basic drills will not be instructionally effective, even if the nature of the task is very simple. This will be evidenced by a lower number of tasks completed, higher time to completion.*	This hypothesis is supported if all treatment conditions perform better in the assessment than the control group, across all performance measures.
H2: *Participants in the low-cost resources environment will finish the tasks faster but will evidence the lowest level of performance, as predicted by the Soft Constraints Hypothesis (Gray & Fu, 2004; Gray et al., 2006).*	The total time to complete the exercises in Condition 3 (Low-Cost Resources) will be the lowest of all treatment environments, but this group will evidence lower task completion and higher time-to-completion than other treatment groups.
H3: *Participants in environments where no resources are available will take longer to complete the tasks but will evidence the highest level of performance (Gray & Sabnani, 1994; Morgan et al., 2007; Waldron et al., 2007).*	ELIMINATED

H4: *Participants in environments where resources are available but have a high access cost will exhibit intermediate completion times and intermediate performance levels (Gray & Sabnani, 1994; Morgan et al., 2007; Waldron et al., 2007).*	The total time to complete the exercises in Condition 4 (High-Cost Resources) will be higher than Condition 3 (Low-Cost Resources) but lower than Condition 5 (Game). Assessment results will also be between the results obtained by individuals in Condition 3 (Low-Cost Resources) and 5 (Game).
H5: *Assessment results are expected to be higher for the game than those for the environment where the resources will have the same cost but no game context. When embedded in the unifying context of a simple abstract game, the instructional effectiveness of software application training environments based on the training-wheels approach and distributed information resources will be the highest of all environments (and will result in higher satisfaction levels) because of the high cost of the information resources: increased use of resources will not only imply a higher retrieval cost in milliseconds, but will also negatively affect the game score as time is also used as a valuable game resource.*	The assessment results for Condition 5 (Game) will be higher than Condition 3 (Low IAC). Reported satisfaction will be higher.
H6: *The game environment will result in a higher number of repetitions per task than any other environment.*	The average number of exercises completed in Condition 5 (Game) is higher than in any other environment.

H7: *Individuals with higher computer self-efficacy will report higher motivation levels regardless of the instructional intervention used, as reported by Tsai (2004).*	Individuals that report higher computer self-efficacy as measured by the CUSE (Cassidy & Eachus, 2002) will report higher satisfaction levels than other participants regardless of condition.
H8: *Individuals with higher computer self-efficacy will perform better regardless of the instructional intervention used, as reported by Tsai (2004).*	Individuals reporting higher computer self-efficacy as measured by the CUSE (Cassidy & Eachus, 2002) will have better performance than other participants regardless of condition.
H9: *There will be variation in strategies in the gaming environment which will be influenced by the player type. Achievers are expected to obtain the highest scores in the least amount of time, in line with their desire for accomplishment in the game environment. Explorers are also expected to find their gaming style supported and report high game scores, with less emphasis on time. In consequence, if the Soft Constraints Hypothesis (Gray & Fu, 2004; Gray et al., 2006) applies here as well, resource use is expected to be the lowest for achievers, followed by explorers. Socializers, Killers, and Role-Players will have the highest use of resources.*	If participants in Condition 5 (Game) are grouped by player type, different patterns of resources access (number of times accessed, type of resource) will be evidenced. Achievers are expected to use fewer resources, followed by Explorers. Socializers, Killers, and Role-Players are all expected to display a high use of resources.

H10: *Differences in satisfaction levels between different player types will be observed in the gaming environment. Socializers, Killers, and Role-Players are expected to report low levels of satisfaction because their preferred gaming style is not supported. Explorers are expected to report moderate levels of satisfaction, and Achievers the highest satisfaction levels of all player types.*	If participants in Condition 5 (Game) are grouped by player type, significant differences in reported satisfaction level will be evidenced. Achievers are expected to report higher satisfaction levels, followed by explorers, and again by Socializers, Killers, and Role-Players, according to their specific goals in gaming environments.
H11: *High self-regulating individuals will report lower satisfaction and low self-regulating individuals will report higher satisfaction, regardless of the environment due to linearity.*	Self-regulation (modified MSLQ) will negatively correlate with reported satisfaction (Satisfaction Survey).

Study – Phase I

Participants

Participants were adults coming from any background, with or without experience with graphic software applications. They were required to have a minimum knowledge of basic computer operations, namely how to use a mouse to click on buttons and open panels. Nobody was expected to have experience playing video games even though some of them were assigned to the Game condition.

Calls for participation were posted at major universities in Montreal, on electronic news boards in two companies, and on a Facebook account with a minimum reach of 2,500 people. The success of Facebook was evidenced by the fact that a few people from around the world thought the study was going to be conducted entirely online and wanted to participate. At this stage a full-day introductory Flash course was offered as incentive to all those who participated. Four data-collection sessions were offered during two consecutive weekends, and the course was offered one week later. Prospective participants emailed the researcher who replied with information about the study and the conditions to participate. Once accepted, the researcher replied with a confirmed time to participate, and instructions to reach the computer lab. None of the people who applied were rejected, but some chose not to participate after they received more details about the study.

A total of 33 people applied and completed the study at this point. They were randomized to one of the four conditions. Nevertheless, due to a bug in the Game environment many people did not complete it and more participants had to be assigned to that group. The assignment breakdown per condition was as follows: (a) control group = 6, (b) Low-Cost Resources environment = 6, (c) High-Cost Resources environment = 8, and (d) Game environment = 13.

Procedure

The general flowchart of the study is presented in figure 12.

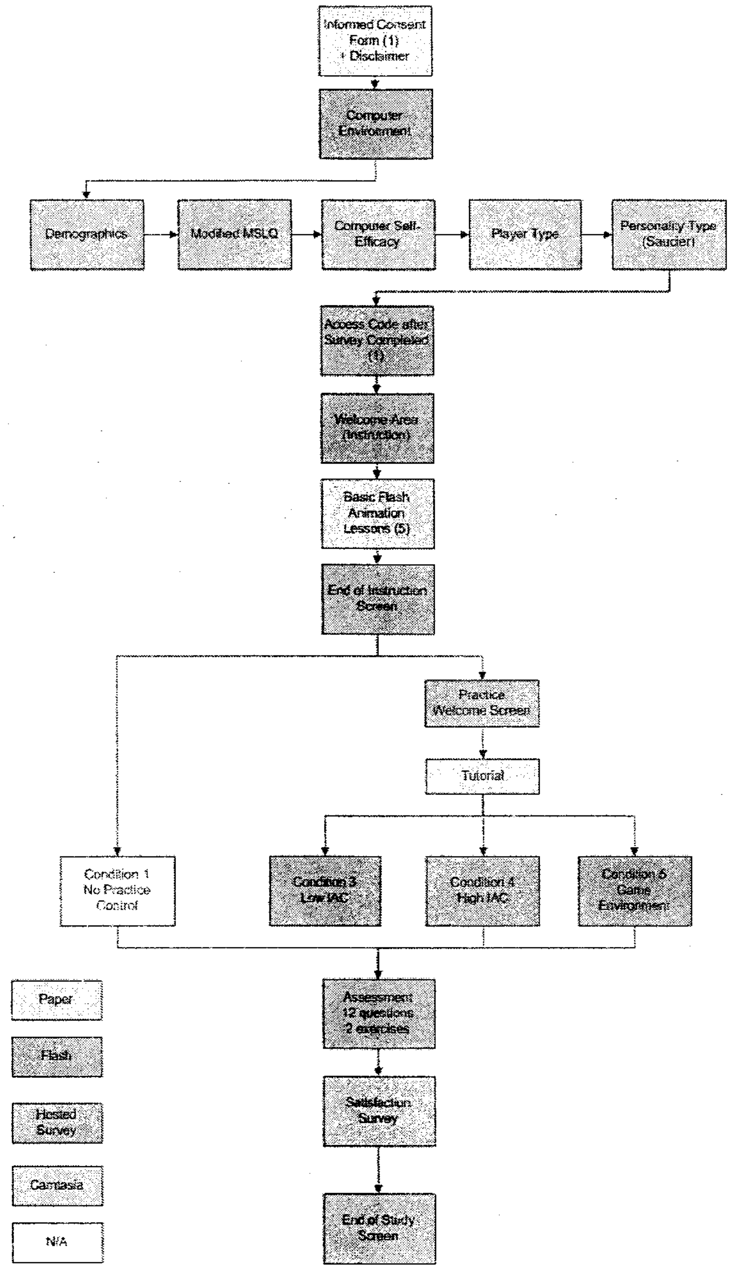

Figure 12. Study flowchart

Welcoming participants. Participants arrived at the agreed time to the computer lab. The researcher confirmed that the participant was in the list and gave him/her a code to use during the entire study. The initial letter in the code allowed the computer program to determine what environment the person had been randomly assigned to, and it was also used to assign the collected data to the corresponding database table.

The researcher then assigned the participant a computer. All machines used in the study did not run the Flash application. The computers had Firefox already open with the consent form on the screen. The researcher explained to each participant that the study consisted of an initial survey, followed by 5 short lessons on Flash animation concepts. If the participant had been assigned to the Control group, he/she was told that an assessment section would follow. All other participants were told that a practice session followed the short Flash lessons. Participants were also told that each computer had different training programs, in case they saw different interfaces in other computers or overheard instructions given for other environments. They were then given time to read the consent form and if they had no questions and agreed to proceed they were allowed to work on their own. Headphones were provided for the lessons. The researcher remained in the computer lab at all times to help with any problems. As soon as they finished the study, participants signed up for the Flash course. Location and time details were sent by e-mail.

Preliminary data collection. At the beginning of the study, all participants completed a set of computer-assisted self-administration questionnaires: demographics, a modified MSLQ (Pintrich & De Groot, 1990), CUSE instrument for computer self-efficacy (Cassidy & Eachus, 2002), Saucier's (1994) personality mini-markers, and Van Meurs' (2007) player type inventory. The surveys were created on HostedSurvey which is a free online survey tool. Except for the demographics data, all questions had a 1 to 5

Likert scale. There were no problems with the data collection process. Survey completion took between 15 to 20 minutes. The data was stored in a text file in tabular form that was downloaded once the study was completed.

Flash environment: Instruction, practice and assessment. After preliminary data collection, participants logged into the Flash application. From the moment the user logged in, the computer program recorded every mouse click and time stamped every action. The information was sent to a database which recorded the following information in a table specific to each user:

1. Date and time of login

2. Time elapsed from beginning of practice, in milliseconds.

3. Exercise number: 0 for lessons, the corresponding number for the exercises in the practice section, or the corresponding question number in the assessment section.

4. Action taken: Beginning and end of a section (lessons, practice, or assessment), beginning of a lesson, click on an icon, select a command, submit a quiz answer, or hit any button.

5. Relative time stamp: time elapsed from beginning of section (lessons, practice, or assessment).

6. Code: A code assigned to specific actions. For the lessons the code had no specific meaning, so a 'null' value was stored. Specific conventions during the practice exercises will be further discussed in that section. In the assessment section the code corresponded to the answer to the multiple-choice questions.

7. Expected action: During the practice exercises, the system recorded the expected action for every step taken by the user, so that it was possible to compare differences between what the user was doing and what the task required. This also applied to the exercises in the assessment section.

Lessons. Participants started the training by viewing 5 video lessons on Flash animation concepts created with Camtasia. The videos run for approximately 20 minutes in total but users were able to go back, review the videos, or watch the lessons twice. The topics covered were:

1. Flash Interface
2. Drawing in Flash
3. Animation concepts
4. How to animate the fill color of a simple object
5. How to animate one shape into another shape

Practice tutorial. Participants in the Control group went directly to the assessment section. All other participants were shown a tutorial corresponding to the particular environment in which they were going to do the practice exercises. The Low-Cost and High-Cost conditions had both the same interface so the tutorial was the same. The Game condition added to that information the necessary details about the game and how it was played. The tutorial lasted about 5 minutes.

Practice exercises. The practice environment was the treatment condition in this study. In order to practice what they had learned in the lessons, all participants with the exception of those in the Control group had to complete a total of 20 exercises. These exercises were identical in all treatment conditions. The Flash simulator reproduced the commands and visual changes a user would normally see in the real Flash interface. The entire process was identical to the real software with the sole exception of drawing the shapes which was done automatically by the simulator in all exercises to reduce code complexity.

Depending on their code, participants were assigned to the corresponding practice session. There were 3 practice environments:

1. Low-Cost Resources: included resources, according to the Distributed Information Resources Model (Wright et al., 2000) were available by clicking on them.

2. High-Cost Resources: Same as the Low-Cost environment except that there was a delay of 1 second before the panel would open after the panel bar was clicked.

3. Game Environment: The same functionality as in the Low-Cost Resources environment, but embedded in a game. The user had a board that started filling up with rows. The objective was to clear up the rows by completing the corresponding transformation in the Flash simulator.

Participants in the Low-Cost and High-Cost environments had no limit of time. They completed the exercises at their own pace. Those assigned to the Game condition had initially 25 seconds per exercise, and the time accumulated as more exercises and more levels were completed. With full knowledge a short transformation took about 8 seconds to complete and a long transformation around 13 seconds. The sequences built on the previous ones and the pattern was similar in all of them and quickly became evident.

All conditions had the following common elements:

1. Task Description panel: It stated the objective in plain English, for example "Complete the animation of a green circle into a yellow circle in 1 second".

2. Target Animation panel: This panel showed the final completed animation.

3. History/Plan panel: A list of each of the steps that had to be taken to complete the task. As the user took a step, the corresponding item on the list was checked.

4. Feedback panel: If the user made a mistake, this panel described the error in detail.

5. Feedback bar that turned green when the step taken was correct, and red when a mistake was made.

6. System help button, which gave the user an overview of the panels on the screen.

7. Return to Tutorial button, allowing the user to go back to review the tutorial.

8. Exercise Demo, showing a Camtasia video describing how to complete the first exercise.

In addition, the Low-Cost and High-Cost environments had: (a) a counter telling the user how many exercises had been completed, for example 5 of 20., and (b) a Restart button, allowing the user to restart the current exercise.

The Game environment had: (a) a game board with four rows that had to be cleared before they filled up the board in a given time, (b) a timer showing the user how much was left to complete the exercises, (c) a health meter, which decreased with each mistake made, and (d) a level indicator, showing how many levels had been completed and the current level.

All exercises were representative of the type of tasks learners would attempt when first learning animation concepts in an instructor-based course. Correct execution required a mix of commands located in the top menu, and icons on the screen. The exercises were divided into groups of 4 but this division was not apparent for the Low-Cost and High-Cost environments. In the Game environment each group represented a level, for a total of 5 levels.

Group 1: Color animation, completion exercises, 1 to 4. These exercises had a shape already on the stage and the user had to complete the sequence to create an animation of a color transformation in a given amount of time. The user required the animation time, which could be obtained from the Task Description panel or from the History/Plan panel. Once this piece of information was obtained, the user had to enter the following sequence:

Select correct frame – Create Keyframe – Change color by clicking on the color swatch – Select a middle frame between the first and last keyframes – Apply Shape tween.

Group 2: Shape animation, completion exercises, 5 to 8. In a similar manner, the participant had to complete the sequence to animate a shape transformation. The number of steps was the same, but the type of Keyframe changed to a Blank Keyframe, and instead of changing color a new shape was created. The full sequence was:

Select correct frame – Create Blank Keyframe – Select new shape (oval or rectangle) – Select a middle frame between the first and last keyframes – Apply Shape tween.

Group 3: Full exercises, color animation, exercises 9 to 12. In this set a full color animation had to be created. Once the animation time was known, the entire sequence was:

Select default first Keyframe – Select shape (oval or rectangle) – Select color swatch – Select frame corresponding to animation time – Create Keyframe – Change color by clicking on color swatch – Select frame between first and last Keyframes – Apply Shape tween.

Group 4: Full exercises, shape animation, exercises 13 to 16. This group required the creation of a full shape animation, the color being constant. The sequence had the same number of steps as the previous one, but with a different type of Keyframe and a shape transformation:

Select default first Keyframe – Select shape (oval or rectangle) – Select color swatch – Select frame corresponding to animation time – Create a Blank Keyframe – Change shape by clicking on the new shape icon (oval or rectangle) – Select frame between first and last keyframes – Apply Shape tween.

Group 5: Full exercises, color and shape animation, exercises 17 to 20. These combined all possible changes in color and animation. The final sequence had an additional step:

Select default first Keyframe – Select original shape (oval or rectangle) – Select color swatch – Select frame corresponding to animation time – Create a Blank Keyframe – Change shape by clicking on the new shape icon (oval or rectangle) – Select new color by clicking on the color swatch – Select frame between the first and last keyframes - Apply shape tween.

Data recording. During the practice session all actions taken by the participants were time stamped and recorded in a database, as previously described. The code used to classify the actions was:

1. CM - For a command including oval, rectangle, color swatch, Keyframe creation, and tween.
2. FS - For a frame selection.
3. TD: The Task Descripton panel was opened.
4. TA: The Target Animation panel was opened.
5. H: The History/Plan panel was opened.
6. FB: The Feedback panel was opened.
7. IH: Interface help button clicked.
8. GH: Game help button clicked.
9. ED: Exercise Demo button clicked.
10. BT: Back to Tutorial button clicked.
11. RS: Restart Exercise button clicked.

Additionally errors in the sequence were coded according to the following codes:

1. E1 - Wrong icon, does not match the instruction (if any)

2. E2 - Attempting to create a Keyframe in the wrong frame
3. E3 - Creating a Keyframe instead of a Blank Keyframe
4. E4 - Creating a Blank Keyframe instead of a Keyframe
5. E5 - Applying the wrong tween

Assessment. The final stage was the assessment section. All groups completed a test that included a set of 12 questions that evaluated sequence knowledge, and two exercises on the simulator without any help panels. The test aimed to determine not only if the users knew the steps to complete the animation tasks but also if they were able to complete exercises without any help.

The initial 12 questions were presented one after the other, but there were in fact three specific groups:

1. Questions 1 to 4: A sequence with a missing step was presented and the user had to select the correct step from the list. No visual information was provided.

2. Questions 5 to 8: The user had to select the correct full sequence to complete an animation task in the software. No visual information was provided.

3. Questions 10 to 12: A snapshot of the Flash interface was presented and the user had to select the correct sequence that completed the animation task.

Participants also had to complete two exercises in the simulator. These tasks were representative of the typical basic exercises a learner would attempt in an instructor-based course. This part of the test aimed to determine if the participant was able to complete the task without help. It tested for procedural knowledge of the sequences but did not test for transfer. In the first exercise the users had to complete an animation and in the second one they had to create an entire sequence from scratch. In both the multiple-choice questions and exercises users had a 'DON'T KNOW' button in case they didn't know the answer or the procedure. Again both the answers and the procedures were recorded on the database.

Satisfaction Survey. After the test, a final survey on HostedSurvey measured the level of self-reported motivation and satisfaction with the training session and the computer system. The survey was based on usability research at IBM. Consistency with other tools provided support for the reliability and validity of the results (J. R. Lewis, 1995). The word 'system' was replaced for 'practice environment' to suit the specific needs of this study.

Issues in Phase I

Phase I of the study did not attract enough people to support statistical significance of the results. The incentive of a free introductory Flash course failed to attract enough attention.

No issues were reported in the Low-Cost or High-Cost environments. Contrary to what was expected, there were no significant differences between the Low-Cost and High-Cost groups. In view of the small number of participants, it was decided that the two conditions were going to be consolidated. The following hypothesis was directly related to the difference between them and thus was eliminated:

H4: *Participants in environments where resources are available but have a high access cost will exhibit intermediate completion times and intermediate performance levels (Gray & Sabnani, 1994; Morgan et al., 2007; Waldron et al., 2007).*

The Game condition was the most problematic one. The game was affected by a speed issue with the Flash Player that only affect Mac computers. As the game progressed and the interactions of several people with the database increased, the system performance slowed. Several seconds would pass after a user clicked on the panels before they opened. Actions were sometimes not acknowledged by the system, generating false errors. The Game environment was the only one to experience this problem because if the user ran out of time, he/she was forced to start the game from the beginning, compounding the memory problem. Forcing people to restart from the beginning was necessary to compare the full completion time for all 20 exercises in all environments. Since time was a key variable in this study, all 13 results gathered in the Game environment were discarded. It is important to note that despite the difficulties, some people managed to finish the game. The bug was fixed prior to starting phase II of the study by upgrading the Flash Player from version 8.0 to version 9.0.

The participants' final breakdown by condition at the end of phase I was: (a) control group = 6, (b) Low-Cost Resources environment (merged with High-Cost) = 14, (c) Game environment = 0.

Study - Phase II

Overview

In view of the difficulties encountered during phase I, a larger group of people was needed in this phase. Because the study had only 3 final treatment conditions, a target number of 20 participants per condition in total was set. Several unsuccessful attempts were made to find enough participants, including courses at school and university level. It was then decided that financial compensation would be offered instead of the Flash course.

Participants

Calls for participation were posted on universities' news boards, Facebook, and Craigslist offering $25 dollars to those completing the study. A total of 52 people applied and participated in this second stage of the study. Data collection was carried out during one weekend. The second stage had 3 final conditions: Control Group, Low-Cost Resources Environment, and Game Environment. The final breakdown of participants in phase II was: (a) control group = 14, (b) Low-Cost Resources environment = 6, and (c) Game environment = 32. Although participants were randomized to each condition, assignment was influenced by the required target number in each environment.

The large number of participants in the Game environment allowed for people to withdraw from the study without impacting the final target number of 20 participants per condition. The information associated with two users in the Low-Cost Resources Environment was discarded as it was evident from the computer logs that the two participants had randomly played with the controls and the test. At the end of phase two the total number of participants per condition was: a) control group = 20, b) low-cost group = 18, and c) game group = 32. Two participants in the game group had missing values for their tests but one of them completed the game, so the corresponding log data was incorporated in the data mining analysis.

Procedure

The procedure and all instruments used were the same as in phase I, with the exception of the High-Cost Resources practice environment which was eliminated. Whereas nobody in the Control Group or the Low-Cost Resources Environment withdrew from the study, several people did not finish the game but completed the test and satisfaction survey. This is an important factor as will be shown in the results section.

CHAPTER 4. RESULTS

Introduction

This chapter presents the results of the study. It will include hypotheses tests as well as additional analysis carried out with the data.

Data

Survey data collected through HostedSurvey was downloaded as a text file and later imported into Excel. Data was organized per condition and processed to obtain final scores for the following:

1. Modified MSLQ: Self Efficacy, Intrinsic Value, Anxiety, and Self Regulation
2. Computer Self-Efficacy
3. Player Type: Achiever, Socializer, Griefer, Explorer, and Role Play
4. Personality Markers: Extraversion, Agreeableness, Conscientiousness, Emotional Stability, and Intellectual Openness
5. Overall Satisfaction

Data collected through the Flash application was transferred from the online database to Excel, checked, and cleaned up for further processing. Two main spreadsheets were initially created.

The first spreadsheet compiled the time spent per lesson, time spent in all lessons, time spent in the practice tutorial, time spent in the practice environment, time spent in each test question, answers to each multiple choice question, time spent per test exercise in the simulator, number of total clicks, and type and number of errors made in each exercise.

The second spreadsheet compiled all the logs created by each participant during the practice exercises. Data was checked to ensure completeness and look for any system errors. This data was not analyzed using standard statistical methods. Instead data mining techniques were used to determine variations in sequences.

Demographic Data

A total of 70 participants were included in the final analysis, 47 female, 23 male. Of all participants 51% had a university degree, 30% had finished secondary school, 15% had a Master's degree, and 3% had a PhD. 43% were students, 24% worked full-time, 17% part-time, and 16% were unemployed. 49% had an advanced command of the English language, 37% were native speakers, and 14% had an intermediate command of the language. Only 3% considered themselves average Flash users, while the rest of the participants had never used Flash before. The same percentage applied to the use of other type of animation software. 17% had previous knowledge of keyframes, and 19% knew the concept of tweening. These two last percentages were used to determine if some participants had previous knowledge that could provide an advantage while doing the exercises or the tests. Of 70 participants, two did not complete the test but finished the study. The Control group had 20 participants, the Low-Cost group 18, and the Game group 32, this last group including the two that did not complete the test.

Modified MSLQ Data

Table 2 show summaries obtained, per condition, for the modified MSLQ.

Table 2

Modified MSLQ Results

Condition	Mean	Min	Max	Std. Dev.
		Self-Efficacy		
Control	3.4	2.7	4.4	0.50
Low-Cost	3.5	2.6	4.2	0.50
Game	3.5	2.2	4.8	0.66
		Intrinsic Value		
Control	4.0	2.9	5.0	0.58
Low-Cost	4.2	3.0	4.8	0.46
Game	3.9	2.5	5.0	0.67
		Anxiety		
Control	2.1	1.0	3.3	0.78
Low-Cost	1.8	1.0	3.8	0.74
Game	2.1	1.0	4.8	1.05
		Self-Regulation		
Control	3.1	1.7	4.3	0.67
Low-Cost	3.1	2.2	4.3	0.58
Game	3.3	2.0	4.3	0.58

The only statistically significant result found between any of these constructs and assessment results was a low correlation of $r(66) = 0.301$, $p = 0.13$, between Anxiety and total number of errors made in the assessment.

Computer Self-Efficacy

The summary for computer self-efficacy is shown in Table 3.

Table 3

Computer Self-Efficacy

Condition	n	Mean	Min	Max	Std. Dev.
Control	20	120	82	144	15.6
Low-Cost	18	124	105	149	13.9
Game	32	124	82	150	18.0

Only a very weak correlation ($r(66) = 0.239$, $p = 0.05$) between computer self-efficacy and total assessment score was found. No correlation was found with any other assessment result or satisfaction score.

Player Type

Table 4 shows the summary for player types across conditions.

Table 4

Player Type

Condition	Mean	Min	Max	Std. Dev.
		Achiever		
Control	3.3	1.3	4.7	0.9
Low-Cost	3.6	2.2	5.0	0.8
Game	3.4	1.3	5.0	1.0
		Socializer		
Control	2.6	1.0	3.6	0.8
Low-Cost	2.3	1.0	4.0	1.0
Game	2.6	1.0	3.8	0.9
		Griefer		
Control	2.1	1.2	4.8	0.9
Low-Cost	2.4	1.0	4.6	1.2
Game	1.9	1.0	4.2	0.8

		Explorer		
Control	3.0	1.7	4.0	0.7
Low-Cost	2.9	2.2	3.8	0.6
Game	3.2	1.7	4.5	0.8
		Role-Player		
Control	3.2	2.3	4.3	0.7
Low-Cost	3.3	1.7	4.4	0.8
Game	3.1	1.3	5.0	1.0

A small correlation ($r(68) = 0.238$, $p = 0.047$) was found between the Achiever player type and overall satisfaction score. A low correlation ($r(68) = 0.305$, $p = 0.010$) also appeared between the Explorer player type and the time it took to complete the lessons. A negative, low correlation ($r(66) = -0.240$, $p = 0.049$) was found between the Explorer type and the time it took to complete the assessment section.

Personality Traits

Table 5 summarizes results for personality markers according to Saucier (1994).

Table 5

Personality Traits

Condition	Mean	Min	Max	Std.Dev.
		Extraversion		
Control	3.3	1.7	4.6	0.64
Low-Cost	3.6	2.4	4.6	0.66
Game	3.3	1.3	4.6	0.82
		Agreeableness		
Control	4.1	3.1	5.0	0.52
Low-Cost	4.0	3.0	5.0	0.56
Game	4.1	3.0	5.0	0.57

Conscientiousness				
Control	3.9	3.0	4.9	0.57
Low-Cost	3.9	2.0	4.9	0.81
Game	3.6	1.6	5.0	0.94
Emotional Stability				
Control	3.3	1.3	4.6	0.76
Low-Cost	3.3	1.8	4.8	0.69
Game	3.3	1.5	4.5	0.69
Intellectual Openness				
Control	4.2	2.4	5.0	0.67
Low-Cost	4.4	3.1	5.0	0.53
Game	4.2	2.5	5.0	0.57

A low correlation ($r(66) = 0.280$, $p = 0.021$) was found between Emotional Stability and the total assessment score value. Another small negative correlation ($r(68) = -0.298$, $p = 0.012$) appeared between Conscientiousness and the time it took to complete the practice exercises.

Practice Session

Table 6 summarizes information for time used for the practice tutorial, time used for all lessons, and total number of actions for each condition with a practice environment (control group n=20, low-cost group n=18, game group n=32).

Table 6

Practice Environment Summary

Condition	Mean	Min	Max	Std.Dev.
Lessons Time (ms)				
Control	971	733	1489	193
Low-Cost	924	0	1553	338
Game	1018	0	2104	352

Practice Time (ms)				
Low-Cost	801	521	1142	168
Game	1929	453	4330	1004
Clicks				
Low-Cost	183	159	240	21
Game	513	172	1104	226

The value of 0 reflect the fact that some participants in the Low-Cost and the Game environments chose to skip the lessons.

Assessment

The assessment section was divided into two sections. The first section comprised 12 multiple-choice questions. The second section included two exercises. The 12 multiple choice questions were further subdivided into three groups. Each group was marked separately and also a combined score was produced for all 3 groups.

The partial and total assessment scores for the multiple choice questions is evaluated first, where a) control group n = 20, b) low-cost group n = 18, and c) game group n = 30. Remember that two participants in the Game group did not have test results. The summary is shown in Table 7.

Table 7

Assessment Results - Multiple Choice Questions

Condition	Mean	Min	Max	Std. Dev.
Question Group 1				
Control	3.75	2	4	0.55
Low-Cost	3.83	3	4	0.38
Game	3.83	0	4	0.75

Question Group 2				
Control	3.30	1	4	0.98
Low-Cost	3.56	2	4	0.71
Game	2.97	0	4	1.07
Question Group 3				
Control	3.20	0	4	1.11
Low-Cost	3.61	2	4	0.61
Game	2.80	1	4	0.93
Total Score				
Control	10.25	4	12	2.12
Low-Cost	11.00	10	12	0.97
Game	9.60	3	12	1.94

As can be seen, the Low-Cost group had the highest score, followed by the Control group. The Game group had the lowest multiple choice questions score. A one-way ANOVA was conducted to determine if there were differences in assessment scores due to treatment condition. A significant effect was found for the third group of questions, $F(2,65) = 4.509$, $p = 0.015$, and for the overall assessment score, $F(2,65) = 3.431$, $p = 0.038$, as shown in Table 8. In both cases, a Scheffé post hoc analysis showed a significant difference between the Low-Cost and the Game group.

Table 8

Multiple Choice Third Question and Total Score ANOVA

	Sum of Squares	df	Mean Square	*F*	Sig.
	Third Question				
Between Groups	7.531	2	3.766	4.509	0.015
Within Groups	54.278	65	0.835		
Total	61.809	67			

	Total Score				
Between Groups	22.271	2	11.135	3.431	0.038
Within Groups	210.95	65	3.245		
Total	233.221	67			

The multiple choice questions reflected better knowledge of the animation sequences. However, the test exercises in the simulator evaluated if participants were really able to complete animations tasks without help. Table 9 shows the summary results for the time to completion for the test exercises.

Table 9

Test Exercises Summary

Condition	*n*	*M*	Min	Max	*SD*
		Text Exercise 1 Time (s)			
Control	20	108	39	228	54
Low-Cost	18	40	19	77	13
Game	30	41	24	76	15
		Text Exercise 2 Time (s)			
Control	16	71	33	201	39
Low-Cost	18	33	20	78	13
Game	28	35	20	77	13

Both the Low-Cost and the Game groups spent considerable less time to complete the test exercises. It is also important to note that not all participants finished the exercises. Table 10 shows the completion rate per condition.

Table 10

Test Exercises Completion Rate

Condition	Finished		Did Not Finish	
	n	%	*n*	%
		Exercise 1		
Control	17	85%	3	15%
Low-Cost	18	100%	0	0%
Game	30	100%	0	0%
		Exercise 2		
Control	16	80%	4	20%
Low-Cost	17	94%	1	6%
Game	28	93%	2	7%

A one-way ANOVA showed an important effect of treatment condition on time used to complete each assessment exercise, as can be seen in Tables 11 and 12.

Table 11

Assessment Exercise 1 ANOVA

	Sum of Squares	df	Mean Square	*F*	Sig.
Between Groups	7.08E+10	2	3.54E+10	33.323	0
Within Groups	6.10E+10	65	1.06E+09		
Total	1.40E+11	67			

Table 12

Assessment Exercise 2 ANOVA

	Sum of Squares	df	Mean Square	*F*	Sig.
Between Groups	1.65E+10	2	8.26E+09	15.799	0
Within Groups	3.09E+10	59	5.23E+08		
Total	4.74E+10	61			

Post hoc comparisons using the Scheffé test indicated that the control group mean was significantly different from both the Low-Cost and the Game groups, in both exercises. Finishing the exercises was important, but another key consideration is how many mistakes users made before completing the correct sequences of commands. Table 13 summarizes this information.

Table 13

Average Number of Errors per Exercise

Condition	*n*	*M*	Min	Max	*SD*
		Test Exercise 1			
Control	20	3.15	0	17	4.00
Low-Cost	18	0.50	0	5	1.25
Game	30	0.43	0	5	1.07
		Test Exercise 2			
Control	16	2.25	0	13	3.39
Low-Cost	18	1.00	0	9	2.22
Game	28	0.77	0	4	1.07

A significant effect was found for number of errors made in both exercises, as Table 14 shows.

Table 14

Errors in Exercises 1 and 2 ANOVA

	Sum of Squares	df	Mean Square	*F*	Sig.
			Exercise 1		
Between Groups	102.333	2	51.167	9.126	0
Within Groups	364.417	65	5.606		
Total	466.75	67			

	Exercise 2				
Between Groups	44.165	2	22.083	4.223	0.019
Within Groups	308.545	59	5.23		
Total	352.71	61			

Post hoc analyses using Scheffé showed significant differences between the Control group and both the Low-Cost group and the Game group in exercise 1, and between the Control group and the Game group in exercise 2.

Separation in the Game Group

The previous results show that the Low-Cost group did considerably well in all assessment items. However, an important issue in the Game group was the completion rate of the game. 8 of the participants did not finish the game and requested to stop the practice session and continue directly to the assessment test. Of these eight participants, one did not complete the test. The Game group was the only one to have this problem, as nobody in the Low-Cost group had any issues with the practice session. Interesting results appear if the Game group is further divided into those who finished and those who did not finish. It is important to keep in mind that the small number of those who did not finish is a limiting factor in the following analysis.

The first thing to notice is the amount of time it took those who finished the game to complete all 20 exercises in the practice environment once they mastered the game, when compared to the Low-Cost group. A t-test for equality of means showed a significant difference between the two groups , $t(40) = 4.604$, $p = 0.038$. Table 15 shows the means, minimum and maximum value, and standard deviation for the Low-Cost and the Game (Finished) groups.

Table 15

Time to Completion 20 Practice Exercises

Condition	*n*	*M*	Min	Max	*SD*
Low-Cost	18	801	520	1142	168.3
Game	24	535	417	1036	131.25

In the assessment section the results for the multiple-choice questions are shown in Table 16, where a) control group n=20, b) low-cost group n = 18, c) game group (Finished) n = 23, and d) game group (Did not Finish) n = 7.

Table 16

Multiple Choice Questions Results Summary - 4 Groups

Condition	*M*	Min	Max	*SD*
Question Group 1				
Control	3.75	2	4	0.55
Low-Cost	3.83	3	4	0.38
Game (Finished)	3.96	3	4	0.21
Game (Did not Finish)	3.43	0	4	1.51
Question Group 2				
Control	3.30	1	4	0.98
Low-Cost	3.56	2	4	0.71
Game (Finished)	3.13	1	4	0.87
Game (Did not Finish)	2.43	0	4	1.51
Question Group 3				
Control	3.20	0	4	1.11
Low-Cost	3.61	2	4	0.61
Game (Finished)	3.04	1	4	0.83
Game (Did not Finish)	2.00	1	3	0.82

	Total Score			
Control	10.25	4	12	2.12
Low-Cost	11.00	10	12	0.97
Game (Finished)	10.13	7	12	1.46
Game (Did not Finish)	7.86	3	10	2.41

Participants who did not finish the game scored well below the other groups, including the Control group. A significant difference was found again for the third group of questions, $F(3,64) = 5.891$, $p = 0.01$, and the total score, $F(3,64) = 5.822$, $p = 0.01$. A post hoc comparison using Scheffé showed that in the third group of questions, the mean of the Game (Did not Finish) group was significantly different from the Control and Low-Cost group. In the total score, the mean of the Game (Did not Finish) group was significantly different from all the other groups.

The Game (Did not Finish) group also did poorly on the test exercises time on the simulator, as summarized in Table 17. Notice that participants who finished the game had the shortest times of all groups. There was a significant difference in time in exercise 1, $F(3,64) = 22.358$, $p < 0.01$, and exercise 2, $F(3,58) = 11.221$, $p < 0.01$. Again, a post hoc analysis using Scheffé showed a significant difference between the mean of the Control group and the rest of conditions in exercise 1, and between the Control group and the Low-Cost and the Game Finished group in exercise 2.

Table 17

Test Exercises Time Summary - 4 Groups

Condition	*n*	*M*	Min	Max	*SD*
		Test Exercise 1 Time (s)			
Control	20	108	39	228	54
Low-Cost	18	40	19	77	13
Game	23	38	24	76	14
Game Did not Finish	7	50	37	65	13
		Text Exercise 2 Time (s)			
Control	16	71	33	201	39
Low-Cost	18	33	20	77	13
Game	23	32	20	77	12
Game Did not Finish	5	47	28	64	13

Regarding the number of errors, the results for the group that did not finish the game are also above the Control group but below the Low-Cost group and the Game group that finished the entire game. There was a significant differences in both exercise 1, $F(3,64) = 6.340$, $p < 0.01$, and exercise 2, $F(3,58) = 3.628$, $p = 0.018$. A post hoc analysis showed a significant difference between the mean of the Control group and the Low-Cost and Game Finished groups in exercise 1. In exercise two the only significant differences between means was between the Control group and the Game Finished group. Table 18 shows the summary.

Table 18

Number of Errors in Exercises - 4 Groups

Condition	*n*	*M*	Min	Max	*SD*
		Test Exercise 1			
Control	20	3.15	0	17	4
Low-Cost	18	0.50	0	5	1.25
Game	23	0.22	0	3	0.67
Game Did not Finish	7	1.14	0	5	1.77
		Text Exercise 2			
Control	16	2.25	0	13	3.39
Low-Cost	18	1.00	0	9	2.22
Game	23	0.52	0	2	0.73
Game Did not Finish	5	1.57	0	4	1.62

Overall Satisfaction

A final section evaluated the self-reported satisfaction shown in Table 19.

Table 19

Overall Satisfaction

Condition	*n*	*M*	Min	Max	*SD*
Control	20	4.4	3.2	5.0	0.66
Low-Cost	18	4.8	4.1	5.0	0.25
Game	32	4.2	2.6	5.0	0.64

The satisfaction value was further explored by dividing the Game group into those who finished and those who didn't, as can be seen in Table 20. Not surprisingly, those who did not finish the game reported the lowest levels of satisfaction.

Table 20

Overall Satisfaction - 4 Groups

Condition	*n*	*M*	Min	Max	*SD*
Control	20	4.4	3.2	5.0	0.66
Low-Cost	18	4.8	4.1	5.0	0.25
Game	24	4.3	3.0	5.0	0.47
Game Did not Finish	8	3.9	2.6	4.9	0.95

Open questions at the end of the survey requested feedback from the participant. They asked what they liked and did not like about the entire course. Most people liked the Camtasia lessons. The Control group was happy with the training in spite of the fact that they did not have any practice, a point some of them mentioned in their negative comments. People in the Low-Cost group were enthusiastic about the software, and most of the people who played the game liked the experience. A few really disliked the game and found it very stressful. In the Game group, the fact that people had to start from the beginning was a major complain. Notice that even though people who did not finish the game reported the lowest level of satisfaction, the mean was still 3.86 (SD = 0.95).

Practice Logs

The statistical results presented provide an interesting picture of the learning effectiveness and efficiency of the Low-Cost and Game training conditions on simple Flash animation processes. However, these figures do not offer enough information to find the reasons behind these differences.

The Flash environment captured all actions taken by users during the practice session. In particular, the use of resources in both environments was of interest to determine if there were any differences prompted by the different contexts. Data mining techniques were used to explore the patterns.

Data Mining

Data mining can be defined as "the efficient discovery of valuable, non-obvious information from a large collection of data" (Sumathi & Sivanandam, 2006, p.9). Data mining is also associated with knowledge discovery in databases, which is the "nontrivial process of identifying valid novel potentially useful and ultimately understandable patterns in the data" (Sumathi & Sivanandam, 2006, p.9). Data mining has been widely used for a variety of purposes, including finance, medicine, genetics, marketing, and sports (Larose, 2005).

With the popularization of the web and online applications, data mining has been more recently used on data collected on the Internet. There is a differentiation between analyzing web content –which refers to the content itself that can be found on the pages, including links, pictures, and text– and web usage –which refers to the behavior of the people using the web– (Markov & Larose, 2007). Normally this information is analyzed from the logs created by the servers and the data require extensive processing to filter visits made by bots used by search engines such as Google and also to determine visits made by the same user.

Since online training initiatives are becoming more popular, data mining techniques have been incorporated in educational research (Hung & Crooks, 2009;

Romero & Ventura, 2007; Shen, Han, Yang, Yang, & Huang, 2003; Su et al., 2006). In these environments students can be classified depending on the actions taken and suggestions can be made depending on previous choices made or paths followed. However, as opposed to the Flash environment created in this study, web pages as well as content management systems have several resources to choose from. Their use can generate several possible paths leading to different levels of learning effectiveness.

The tasks in this research study were fairly specific and the resources were limited. Once the user knew how to do the exercise, there was only one possible sequence of commands that would be correct. It is possible then to think about the paths followed as departures from an ideal path, the correct solution. This "departure" included the use of the resources available on the screen as well as errors made. The focus of this analysis was the sequences of use followed by each participant, and whether or not they were evidence of specific strategies.

There are several software options for data mining analysis. For this study an open source package, TraMineR, was used. TraMineR is "an R-package for mining and visualizing sequences of categorical data" (Gabadinho, Ritschard, Studer, & Muller, 2008, p.9.) developed at the Department of Econometrics and the Laboratory of Demography at the University of Geneva, and founded by the Swiss National Foundation for Scientific Research. R is a language and environment for statistical computing and graphics.

Data preparation. TraMineR makes a difference between sequences of states and sequences of events, and the researcher has to prepare the data in a suitable format for analysis. Basically an event triggers a change of state (Gabadinho et al., 2008). For example, if the user clicks the color swatch (event), the shape changes color (state).

Each user in the study clicked on different items to complete the exercises. Although the solution sequence had a fixed number of steps, each user took a different number of actions at specific times. One user could have opened one panel and then complete the task, whereas another could have required several help panels or made mistakes in the process. Each task was time stamped, so the format used in this case was the vertical 'time-stamped' event (TSE) format. It basically included the user ID, the time stamp, and the action taken. The action was stored as a code.

Events in this study were classified as previously mentioned:

1. CM - For a command including oval, rectangle, color swatch, Keyframe creation, and tween.
2. FS - For a frame selection.
3. TD: The Task Descripton panel was opened.
4. TA: The Target Animation panel was opened.
5. H: The History/Plan panel was opened.
6. FB: The Feedback panel was opened.
7. IH: Interface help button clicked.
8. GH: Game help button clicked.
9. ED: Exercise Demo button clicked.
10. BT: Back to Tutorial button clicked.

11. RS: Restart Exercise button clicked.

Additionally errors in the sequence were coded as shown next:

1. E1 - Wrong command, incorrect sequence step
2. E2 - Attempting to create a Keyframe in the wrong frame
3. E3 - Creating a Keyframe instead of a Blank Keyframe
4. E4 - Creating a Blank Keyframe instead of a Keyframe
5. E5 - Applying the wrong tween

The 20 exercises were divided into 5 groups. Each group had a correct sequence solution to perform the task, as follows:

1. Group 1 (exercises 1 to 4) FS-CM (keyframe)-CM-FS-CM
2. Group 2 (exercises 5 to 8) FS-CM (blank keyframe)-CM-FS-CM
3. Group 3 (exercises 9 to 12) FS-CM -CM-FS-CM (keyframe)-CM-FS-CM
4. Group 4 (exercises 13 to 16) FS-CM -CM-FS-CM (blank keyframe)-CM-FS-CM
5. Group 5 (exercises 17 to 20) FS-CM-CM-FS-CM-CM-CM-FS-CM

For example, the sequence for group 1 was:

Select correct frame – Create Keyframe – Change color by clicking on the color swatch – Select a middle frame between the first and last keyframes – Apply Shape tween.

Once coded, the sequence is expressed as:

FS-CM (keyframe)-CM-FS-CM

The same equivalence applied to the other groups. These minimal sequences were taken by every user who completed the exercise, plus other individual choices made by the user (for example help panels or errors). Notice that even though two groups could have the same sequence, this did not imply that the exercises were identical. They looked the same because commands were all coded as CM. This was done to simplify the analysis. Otherwise the sequences became too long for the software to handle.

The software detected the number of common sequences across users. Interpretation of the data had to be done carefully as the software produced all sequences and subsequences in the data set, with a certain degree of support specified by the researcher. A support level simply meant that at least a specified number of users selected that sequence. Since subsequences were also added, finding for example CM-FS did not provide much information as this was a subsequence of CM-FS-CM-CM or CM-CM-FS-CM. The focus of attention then was the longest sequences present across users, and unique sequences specifying panel use. These were easily identifiable as peeks or valleys in the data.

It did not make much sense to analyze the entire set of logs, plus it was also not viable due to the fact that people in the Game group would restart the game and go over an exercise several times whereas those in the Low-Cost group went over each exercise only once. Instead, an analysis of the first time the participant encountered the first exercise of each group was conducted, along with an analysis of the last exercise. In other words, the following exercises were analyzed: 1, 5, 9, 13, 17, and 20.

Two files per exercise were created, one for the Low-Cost group and another for the Game group. The information for each exercise was extracted from the complete logs, saved in separate Excel files and then loaded into R. The TraMineR package was loaded and a sequence of events was created for each file. For each exercise and each group the software produced the set of subsequences observed. All files associated with the Low-Cost group were given the prefix 'swd' and all files associated with the Game group were given the prefix 'swg'. The Low-Cost group had 18 participants and the Game group 24, as only those who finished the game were included in the analysis.

Exercise 1. The first exercise is of interest because it was the first time the participant faced both the practice environment and the exercises. The files analyzed were swdexe01.csv and swgexe01.csv. Table 21 shows the summary for both groups.

Table 21

Exercise 1 Frequencies Summary

Low-Cost Group swdexe01.csv		Game Group swgexe01.csv	
Event	Frequency	Event	Frequency
CM	56	FS	68
FS	48	CM	67
TD	24	TD	22
H	10	E1	11
TA	10	E2	7
E2	5	TA	7
Other	13	Other	13

In order to analyze the sequences, a minimum support level of 4 was selected because a lower level produces a very large amount of sequences for later exercises. For example, a support level of 2 for exercise 1 produced 1939 sequences for the Low-Cost environment, and 1037 sequences for the Game environment. A support level of 4 produced 181 sequences for the Low-Cost environment and 115 sequences for the Game environment. It did not make much sense either to explore the sequences at an individual level but instead to obtain the most frequently used sequences.

The most significant sequences for the Low-Cost group are shown in Table 22.

Table 22

Low-Cost Group Sequences Exercise 1

Sequence	Support	Count
(FS)-(CM)-(CM)-(FS)-(CM)	1	18
(TD)-(FS)-(CM)-(CM)-(FS)-(CM)	0.83	15
(H)	0.39	7
(TA)	0.33	6
(TD)-(H)	0.28	5
(TA)-(H)	0.22	4
(FB)	0.22	4

For the Game environment, the significant sequences are presented in Table 23.

Table 23

Game Group Sequences Exercise 1

Sequence	Support	Count
(FS)-(CM)-(CM)-(FS)-(CM)	0.92	22
(TD)-(FS)-(CM)-(CM)-(FS)-(CM)	0.63	15
(FS)-(E1)-(FS)	0.33	8
(H)	0.21	5
(TA)	0.21	5
(E2)	0.21	5

These sequences implied that everybody completed the first exercise in their first attempt in the Low-Cost environment (FS-CM-CM-FS-CM) whereas two people in the Game environment were unable to complete it and run out of time. 15 people in the Low-Cost environment used the Task Description panel to complete the task (TD-FS-CM-CM-FS-CM). The History panel was used by 7 people, the Task Animation panel by 6, and the Feedback panel by 4. A combination of panels (TD and TA used with H) was also found in 4 participants. Notice that no errors appear meaning that fewer than 4 people actually account for the mistakes detailed in the summary in this environment.

In the Game environment on the other hand, also 15 people relied on the Task Description panel to complete the task. Notice that 8 people made E1 errors, meaning they hit the wrong command. 5 people used the other panels in the exercises. 5 people made E2 errors, that is, selecting the wrong frame.

Exercise 5. Table 24 shows the summary for exercise 5 for both groups.

Table 24

Exercise 5 Frequencies Summary

Low-Cost Group swdexe05.csv		Game Group swgexe05.csv	
Event	Frequency	Event	Frequency
CM	54	CM	68
FS	49	FS	63
TD	20	TD	23
E3	10	E1	9
H	5	E3	9
E2	4	E2	7
Other	9	Other	9

At a minimum support of 4, TraMineR produced 165 sequences for the Low-Cost group and 139 sequences for the Game group. Table 25 shows the most frequent sequences for the Low-Cost group.

Table 25

Low-Cost Group Sequences Exercise 5

Sequence	Support	Count
(FS)-(CM)-(CM)-(FS)-(CM)	1	18
(TD)-(FS)-(CM)-(CM)-(FS)-(CM)	0.89	16
(E3)	0.28	5
(H)	0.22	4
(TA)	0.22	4
(TD)-(FS)-(E3)-(TA)-(CM)-(CM)-(FS)-(CM)	0.22	4

Table 26 shows the frequent sequences for the Game group.

Table 26

Game Group Sequences Exercise 5

Sequence	Support	Count
(FS)-(CM)-(CM)-(FS)-(CM)	0.88	21
(TD)-(FS)-(CM)-(CM)-(FS)-(CM)	0.71	17
(E3)	0.33	8
(E1)	0.17	4
(E2)	0.17	4
(H)	0.17	4

In the Low-Cost group everybody finished the exercise, whereas 3 people did not complete exercise 5 in the Game group the first time they attempted it. Most people in both groups relied on the Task Description panel to complete the task, and the most common error was E3 (the wrong type of Keyframe) although a smaller percentage in the Low-Cost group made the mistake. It was easy to expect that some people would be confused by the type of Keyframe as this was the fundamental difference between the first and second groups of exercises.

Exercise 9. Table 27 presents the summary information for exercise 9.

Table 27

Exercise 9 Summary

Low-Cost Group swdexe09.csv		Game Group swgexe09.csv	
Event	Frequency	Event	Frequency
CM	90	FS	96
FS	59	CM	92
TD	22	E1	56
E1	20	TD	24
E2	12	E2	23
H	7	H	9
Other	10	Other	10

The search for sequences in the Low-Cost environment produced 1,276 sequences with minimum support of 4. In contrast, the Game environment produced 4,013 sequences with the same level of support. Table 28 shows the main sequences for the Low-Cost environment and Table 29 for the Game environment.

Table 28

Low-Cost Group Sequences Exercise 9

Sequence	Support	Count
(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(FS)-(CM)	1	18
(TD)-(CM)-(CM)-(FS)-(CM)-(CM)-(FS)-(CM)	0.89	16
(E2)	0.61	11
(E1)	0.56	10
(H)	0.39	7
(E1)-(E1)	0.33	6
(E1)-(H)	0.28	5
(TD)-(H)	0.28	5
(E2)-(E1)	0.22	4
(TD)-(FB)	0.22	4

Table 29

Game Group Sequences Exercise 9

Sequence	Support	Count
(TD)	0.86	21
(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(FS)-(CM)	0.67	16
(E2)	0.63	15
(TD)-(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(FS)-(CM)	0.54	13
(E1)-(E1)	0.54	13
(E1)-(E1)-(E1)	0.5	12
(E1)-(E2)	0.33	8
(E1)-(E1)-(E1)-(E2)	0.29	7
(H)-(CM)	0.29	7
(E2)-(E1)	0.25	6
(E2)-(E2)	0.25	6
(H)	0.25	6
(E1)-(E1)-(E1)-(H)	0.21	5
(TD)-(E1)-(H)	0.21	5
(E1)-(E2)-(E1)	0.17	4
(E2)-(E1)-(E1)-(E1)	0.17	4

The rate of errors increased for both groups with task complexity. However, all participants in the Low-Cost group finished, and 89% relied on the Task Description panel to complete the task, which was similar to previous exercises. The History panel was used by 7 people, 5 users relied on the Task Description and the History panel together, and 4 people used the Task Description and Feedback panels together.

In contrast, only 67% in the Game environment completed the task, and only 54% relied exclusively on the Task Description panel to complete the exercise. The Game environment also evidenced a higher number and frequency of combinations of errors. In spite of a higher number of mistakes, the Feedback panel was not found in the resulting sequences in the Game group.

Exercise 13. Table 30 shows the summary for exercise 13.

Table 30

Exercise 13 Summary

Low-Cost Group swdexe13.csv		Game Group swgexe13.csv	
Event	Frequency	Event	Frequency
CM	90	CM	120
FS	56	FS	78
TD	18	TD	21
E1	10	E1	10
E2	1	H	6
E3	1	FB	3
Other	3	E3	1

TraMineR produced 339 sequences for the Low-Cost group and 387 sequences for the Game group. Table 31 shows the selected sequences for the Low-Cost group. In this group sequences were found at a minimum support level of 8 only, even though the minimum support established was 4. In this group, 94% (17 people) used the Task Description panel to complete the exercise.

Table 31

Low-Cost Group Sequences Exercise 13

Sequence	Support	Count
(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(FS)-(CM)	1	18
(TD)-(CM)-(CM)-(FS)-(CM)-(CM)-(FS)-(CM)	0.94	17
(E1)	0.5	9

Table 32 shows the sequences for the Game group.

Table 32

Game Group Sequences Exercise 13

Sequence	Support	Count
(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(FS)-(CM)	1	24
(TD)	0.86	21
(TD)-(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(FS)-(CM)	0.83	20
(E1)	0.33	8
(H)	0.21	5
(H)-(FS)-(CM)-(FS)-(CM)-(CM)-(FS)-(CM)	0.17	4

A lower error rate can be seen in the Game environment (33%) as opposed to the Low-Cost environment (50%). In this case, all 24 people finished the exercise, 21 people relied on the Task Description panel to complete the task, and 4 used the History panel. It is not possible to know from this information how many people concurrently used both panels, but it was below the minimum support of 4.

Exercise 17. Table 33 consolidates the information for exercise 17, which is the first exercise in the group that requires the longest sequence to complete the task.

Table 33

Exercise 17 Summary

Low-Cost Group swdexe17.csv		Game Group swgexe17.csv	
Event	Frequency	Event	Frequency
CM	108	CM	142
FS	57	FS	86
TD	20	TD	25
TA	4	E1	20
E2	3	H	9
H	2	E2	5
Other	2	FB	3

A total of 183 sequences with minimum support of 4 resulted for the Low-Cost environment and 1,141 for the Game environment. Table 34 presents the main sequences for the Low-Cost environment and Table 35 for the Game environment.

Table 34

Low-Cost Group Sequences Exercise 17

Sequence	Support	Count
(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(CM)-(FS)-(CM)	1	18
(TD)-(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(CM)-(FS)-(CM)	0.94	17
(TA)	0.22	4

Table 35

Game Group Sequences Exercise 17

Sequence	Support	Count
(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(CM)-(FS)-(CM)	0.96	23
(TD)	0.92	22
(TD)-(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(CM)-(FS)-(CM)	0.71	17
(E1)	0.38	9
(E1)-(E1)	0.21	5
(H)	0.21	5
(H)-(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(CM)-(FS)-(CM)	0.21	5
(E2)	0.17	4
(FS)-(CM)-(CM)-(TD)-(FS)-(CM)-(CM)-(CM)-(FS)-(CM)	0.17	4

For the Low-Cost group only two sequences were at the minimum support level of 4, and they both included TA (Task Animation). The rest of the sequences had a minimum support of 17, adding evidence to a more unified approach to task solving. Almost every person (17) in this group used the Task Description panel. In the Game group there was more variation with participants using the Task Description panel or the History panel to complete the exercise. Whereas fewer than 4 people (22%) accounted for the errors in the Low-Cost environment, 9 of the participants (37.5%) in the Game group accounted for E1 errors (wrong step), 20% for double E1 errors, and 16% for E2 errors (wrong frame selected). A very interesting pattern emerged in the Game environment which is the last sequence on Table 35. 4 people performed the first three steps and then used the Task Description panel. Although further exploration revealed that 3 participants had used the History panel before, a careful analysis of the Game environment logs showed that in final exercises people had discovered the pattern and instead of consulting any of the panels at the beginning, they would perform the predicted initial steps and then check the task. This approach suggests a more mechanical and trial-and-error strategy as opposed to the streamlined progression to a unified tactic of the Low-Cost group.

Exercise 20. The summary for this exercise is presented in Table 36.

Table 36

Exercise 20 Summary

Low-Cost Group swdexe20.csv		Game Group swgexe20.csv	
Event	Frequency	Event	Frequency
CM	108	CM	128
FS	59	FS	69
TD	21	TD	26
E1	9	H	6
H	3	E1	5
TA	3	E2	2
E2	2	TA	1

The number of total sequences at minimum support level of 4 for the Low-Cost group was 209, and for the Game group 190. Table 37 shows the selected sequences for the Low-Cost group and Table 38 shows the sequences for the Game group.

Table 37

Low-Cost Group Sequences Exercise 20

Sequence	Support	Count
(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(CM)-(FS)-(CM)	1	18
(TD)-(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(CM)-(FS)-(CM)	1	18
(E1)	0.22	4

Table 38

Game Group Sequences Exercise 20

Sequence	Support	Count
(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(CM)-(FS)-(CM)	0.83	20
(TD)-(FS)-(CM)-(CM)-(FS)-(CM)-(CM)-(CM)-(FS)-(CM)	0.75	18
(FS)-(CM)-(CM)-(TD)	0.17	4
(H)	0.17	4

The Low-Cost group had full completion rate and all participants used the Task Description panel to complete the task. E1 errors were due to 4 people. No sequences were found with a support level between 18 and 4. In the game environment a small subsequence was found at level of support 5 only. However only 20 people completed the exercise, 18 relied on the Task Description panel but the History panel was present in 4 sequences. Participants in the Game environment who reached this point but did not complete exercise 20 were not required by the system to repeat the entire game. A final lower number of errors in the Game group agrees with the lower error rate also found in the test exercises, suggesting a higher level of accuracy for the Game environment.

Error rate and panel use. From the previous data it can be observed that exercises 9 and 17 created the most variation in sequences. This was expected as exercise 9 was the first full exercise and exercise 17 was the first exercise that combines both shape and color transformations. The increase in non-unique sequence variation means that there were more departures from the ideal path and these departures were higher for the Game group than for the Low-Cost group. It is useful to examine if this variation is due to the learner making more mistakes or consulting more information resources. Table 39 shows error support and frequency of each error for both groups. The level of support means that at least one person had this type of error in a given exercise. For example, if a user sequence had E1, this user would contribute to the level of support and the frequency would be 1. However, if he or she had (E1)-(E1), the user would still only contribute once to the level of support, but the frequency in would be 2.

Table 39

Error Support Level and Frequency

	Low Cost Group					Game Group				
	Error Type					Error Type				
	E1	E2	E3	E4	E5	E1	E2	E3	E4	E5
	Exercise 1					Exercise 1				
Support	11.11	11.11	0	5.56	5.56	33.33	20.83	0	12.5	0
Freq.	4	5	0	1	1	11	7	0	3	0
	Exercise 5					Exercise 5				
Support	11.11	5.56	27.78	0	0	16.67	16.67	33.33	0	4.16
Freq.	3	4	10	0	0	9	7	9	0	1
	Exercise 9					Exercise 9				
Support	55.56	61.11	0	16.67	0	54.17	62.5	0	4.16	0
Freq.	20	12	0	4	0	56	23	0	1	0
	Exercise 13					Exercise 13				
Support	50	5.56	5.56	0	0	33.33	0	4.16	0	0
Freq.	10	1	1	0	0	10	0	1	0	0
	Exercise 17					Exercise 17				
Support	5.56	11.11	0	0	0	37.5	16.67	0	0	0
Freq.	1	3	0	0	0	20	5	0	0	0
	Exercise 20					Exercise 20				
Support	22.22	5.56	0	0	0	8.33	4.16	0	0	0
Freq.	9	2	0	0	0	5	2	0	0	0

Notice that we see a higher number of errors in the Game group. Exercise 17 shows a much higher error support level for the Game group.

Was the variation in number of sequences also related to the use of panels? Table 40 shows the support level and frequency for the panels.

Table 40

Panel Use Support and Frequency

	Low Cost Group Panel				Game Group Panel			
	TD	TA	H	FB	TD	TA	H	FB
	Exercise 1				Exercise 1			
Support	88.89	33.33	38.89	22.22	70.83	20.83	20.83	12.5
Freq.	24	10	10	5	22	7	6	3
	Exercise 5				Exercise 5			
Support	88.89	22.22	22.22	5.56	83.33	8.33	16.67	4.16
Freq.	20	4	5	2	23	3	4	1
	Exercise 9				Exercise 9			
Support	94.44	11.11	38.89	22.22	87.5	12.5	33.33	4.16
Freq.	22	2	7	4	24	4	9	1
	Exercise 13				Exercise 13			
Support	94.44	5.56	5.56	5.56	87.5	0	20.83	8.33
Freq.	18	1	1	1	21	0	6	3
	Exercise 17				Exercise 17			
Support	100	22.22	5.56	5.56	91.67	0	20.83	12.5
Freq.	20	4	2	1	25	0	9	3
	Exercise 20				Exercise 20			
Support	100	16.67	16.67	0	95.83	4.16	16.67	0
Freq.	21	3	3	0	26	1	6	0
	Total				Total			
Average	94.44	18.52	21.3	10.19	86.11	7.64	21.53	6.95

Note. TD = Task Description; TA = Task Animation; H = History / Plan; FB = Feedback

It can be seen on Table 40 that panel use remains similar across exercises, showing no evidence of peaks at exercises 9 and 17. The task description panel follows an upward trend in both groups and the Game group seems to have relied more on the History panel than the Low-Cost group. Overall, panel use is lower in the Game group than in the Low-Cost group.

CHAPTER 5. DISCUSSION AND IMPLICATIONS

Introduction

This chapter first presents the hypotheses and discusses whether or not the results supported them. The outcomes are considered within the theoretical framework established in the literature review as well as additional work relevant to this study. It looks at the assumptions made and how they were or were not supported by the study, as well as insights obtained from the analysis. Limitations and contributions are reviewed and recommendations for further research are presented.

Testing Hypotheses

There were several hypotheses formulated at the beginning of this study. The following sections use the results obtained to evaluate if they were supported or not.

Training Environment

H1: *Online courses that do not offer practice of basic drills will not be instructionally effective, even if the nature of the task is very simple. This will be evidenced by a lower number of tasks completed, higher time to completion.*

The Control group scored between the Low-Cost group and the Game group in the multiple choice questions ($M = 10.25$, $SD = 2.12$; $M = 11$, $SD = 0.97$; and $M = 9.60$, $SD = 1.94$ respectively). However, the Control group took much longer to complete the test exercises in the simulator. In exercise 1 the mean completion time was 108 seconds ($SD = 54$) as opposed to 40 seconds ($SD = 13$) for the Low-Cost group and 41 seconds ($SD = 15$) for the Game group. In exercise 2, the mean for the Control group was 71 seconds ($SD = 39$), compared to 33 seconds ($SD = 13$) for the Low-Cost group and 35 seconds (SD

= 13) for the Game group. The standard deviations for the Control group were much higher. Completion rate was lower in the Control group. Regarding declarative knowledge of rules (multiple-choice questions) this hypothesis was not supported by the results but at a practical level (simulator exercises) it can be seen that the Control group had the lowest performance and completion rate.

H2: *Participants in the low-cost resources environment will finish the tasks faster but will evidence the lowest level of performance, as predicted by the Soft Constraints Hypothesis (Gray & Fu, 2004; Gray et al., 2006).*

Participants in the Low-Cost environment did take much less time to complete the exercises (801 seconds (*SD* = 168) versus 1929 seconds (*SD* = 1004) for the Game group). However, they had the highest level of performance across all groups. This hypothesis was not supported.

H3: *Eliminated during the pilot test*

H4: *Eliminated during Phase I*

H5: *Assessment results are expected to be higher for the game than those for the environment where the resources will have the same cost but no game context. When embedded in the unifying context of a simple abstract game, the instructional effectiveness of software application training environments based on the training-wheels approach and distributed information resources will be the highest of all environments and will result in higher satisfaction levels because of the high cost of the information resources: increased use of resources will not only imply a higher retrieval* *cost in milliseconds, but will also negatively affect the game score as time is also used as a valuable game resource.*

The Game group score was the lowest of all groups in the multiple-choice part of the test, particularly for those who did not finish the game (Control: 10.25 (*SD* = 2.12), Low-Cost: 11 (*SD* = 0.97), Game – Finished: 10.13 (*SD* = 1.46), Game – Did not Finish: 7.86 (*SD* = 2.41)). However, those who finished had the lowest completion time for exercise 1 (Control: 108 s (*SD* = 54), Low-Cost: 40 s (*SD* = 13), Game – Finished: 38 s (*SD* = 14), Game – Did not Finish: 50 s (*SD* = 13)) and exercise 2 (Control: 71 s (*SD* = 39), Low-Cost: 33 s (*SD* = 13), Game – Finished: 32 s (*SD* = 12), Game – Did not Finish: 47 s (*SD* = 13)) in the simulator. They also had the lowest rate of errors for exercise 1 (Control: 3.15 (*SD* = 4.0), Low-Cost: 0.5 (*SD* = 1.25), Game – Finished: 0.22 (*SD* = 0.67), Game – Did not Finish: 1.14 (*SD* = 1.77)) and exercise 2 (Control: 2.25 (*SD* = 3.39), Low-Cost: 1.0 (*SD* = 2.22), Game – Finished: 0.52 (*SD* = 0.73), Game – Did not Finish: 1.57 (*SD* = 1.62)). Satisfaction levels were also lowest for the Game group (Control: 4.35 (*SD* = 0.66), Low-Cost: 4.8 (*SD* = 0.25), Game – Finished: 4.3 (*SD* = 0.47), Game – Did not Finish: 3.86 (*SD* = 0.95)).

This hypothesis was partially supported. Those in the Game group had the lowest performance in the multiple-choice test and the lowest level of satisfaction but had the fastest completion times in the simulator exercises and highest accuracy. Also, a higher access cost meant that panels were used less often as shown in Table 40.

H6: *The game environment will result in a higher number of repetitions per task than any other environment.*

This hypothesis was supported. In average, people in the Game environment completed 57 exercises by repeating the levels until they finished. The minimum value was 20, that is, there were participants who completed the game in one single go, and the maximum value was 122 repetitions.

Computer Self-Efficacy

H7: *Individuals with higher computer self-efficacy will report higher motivation levels regardless of the instructional intervention used, as reported by Tsai (2004).*

This hypothesis was not supported. Computer Self-Efficacy had no correlation with motivation levels.

H8: *Individuals with higher computer self-efficacy will perform better regardless of the instructional intervention used, as reported by Tsai (2004).*

This hypothesis was not supported either. Computer Self-Efficacy had no correlation with final score.

Player Type

H9: *There will be variation in strategies in the gaming environment which will be influenced by the player type. Achievers are expected to obtain the highest scores in the least amount of time, in line with their desire for accomplishment in the game environment. Explorers are also expected to find their gaming style supported and report high game scores, with less emphasis on time. In consequence, if the Soft Constraints Hypothesis (Gray & Fu, 2004; Gray et al., 2006) applies here as well, resource use is expected to be the lowest for achievers, followed by explorers. Socializers, Killers, and Role-Players will have the highest use of resources.*

As shown before, resource use was fairly similar across users and lower overall of the Game group regardless of player type. There was no correlation between player type and scores. In consequence, this hypothesis was not supported.

H10: *Differences in satisfaction levels between different player types will be observed in the gaming environment. Socializers, Killers, and Role-Players are expected to report low levels of satisfaction because their preferred gaming style is not supported. Explorers are expected to report moderate levels of satisfaction, and Achievers the highest satisfaction levels of all player types.*

This hypothesis was not supported. No relationship was found between player type and reported satisfaction.

Self-Regulation

H11: *High self-regulating individuals will report lower satisfaction and low self-regulating individuals will report higher satisfaction, regardless of the environment due to linearity, as found by McManus (2000).*

No support was found for this hypothesis as no correlation was obtained between self-regulation and reported satisfaction.

Discussion

Individual Characteristics

Several instruments were used to determine individual characteristics of participants, including previous knowledge, learning style, computer self-efficacy, personality traits and player style. None of these were found to be related to any of the results or satisfaction levels.

Other studies have used individual characteristics to tailor the training to specific profiles (Shen et al., 2003) but others have found no relation between personality traits and performance (Bishop-Clark, Dietz-Uhler, & Fisher, 2007) or learning styles and performance (Lu, Yu, & Liu, 2003).

Even though these tests are useful, with the increase in computer power it is now possible to analyze user data and create models that allow to determine user clusters depending on behaviors (Dalkir, 1997). Such models are more dynamic, avoid the need for previous testing and adapt with new users and behavior trends.

Distributed Cognition and Resources Use

This study adhered to the idea that individuals rely heavily on visual cues to take action in a computer environment (Mayes et al., 1988) (Payne, 1991). It was also based on a theoretical framework in human-computer interaction, the Distributed Information Resources model (Wright et al., 2000), which presents a set of resources and strategies that can help a user complete a task and reduce errors. A software simulator was designed and built based on these assumptions. The software of choice was Adobe Flash, a multimedia authoring tool, and the tasks selected were simple color and shape animations that required specific sequences of commands.

From the point of view of resources, it is clear that the dominant ones were the Task Description and the History panels. Other help panels were used by only a few people, below 4 in each environment. This can be attributed to the simplicity of the tasks, as evidenced by the good performance of the Control group, or because the environment was easy to understand. Also, the open comments in the Control group made it clear that the tasks were repetitive and that people would have been able to handle more complexity, e.g.: "I wanted more!" or "Too repetitive, it should be expected that people know how to multiply and divide".

There is also one important consideration regarding the external resources and that is the relevance and quality/quantity of information they provide for the task at hand. Only the Task Description and History panels provided enough information to complete the animations which in retrospective seems fairly obvious. Because of this, extraordinary variations on the use of the external resources could not be really expected as it is a one-solution type of exercise, not an exploratory task.

However, it was expected that the Feedback panel would be used quite a lot because it would indicate what type of mistake was made. This was not the case and participants, particularly those in the Game group, generated long sequences of errors instead of trying to figure out from the Feedback panel what was wrong. Again, this may be related to the pressure of the timer.

Those who used the feedback panel were enthusiastic about it: "The tutorials were clear and the feedback was very helpful to correct my mistakes. The game was fun." It also appears that the History panel was sometimes used for feedback: "The part that I liked about this practice software is that every time you make a mistake it wouldn't just tell you that you're wrong but help(ed) you by telling you where to go like the history for example, and see what you forgot to do".

Embodied Cognition and the Soft constraints Hypothesis

In addition, this research also considered studies showing that the choice between cognitive and perceptual-motor approaches would depend on their access cost. Higher access costs were also thought to lead to better performance because it was assumed that more memory encoding is taking place (Gray & Fu, 2004; Waldron et al., 2007).

Using any of the resources in the Game environment had a higher cost than in the Low-Cost environment because the participant not only had to roll the mouse over the panel top bar and click on it to open it, but he/she also had the pressure of the timer. The motor effort required to access the panel was the same in both environments, but the time limit imposed an additional cost on the task for those playing the game. This was supported by the fact that overall use of panels was lower in the Game group than in the Low-Cost group.

In line with previous research (Morgan et al., 2007; Waldron et al., 2007) it was expected that people in the Game environment would have the best performance in the assessment. In fact, those who finished the game were able to complete the test exercises on the simulator faster and with fewer errors. However, performance in the multiple choice questions was the worst of all groups, including the Control group that had no opportunity to practice. Even though panel use was lower, this did not imply that there was some type of encoding of panel information taking place.

The practice logs provided useful information to explain why the multiple-choice and practical scores of the test were so different for the Game group. The number of sequences produced by TraMineR had to be evaluated carefully as the total includes sequences and subsequences, not unique sequences. However, a large number indicates more variety of actions and variation across users. Figure 13 shows the number of sequences for each environment for each analyzed exercise in the practice environment.

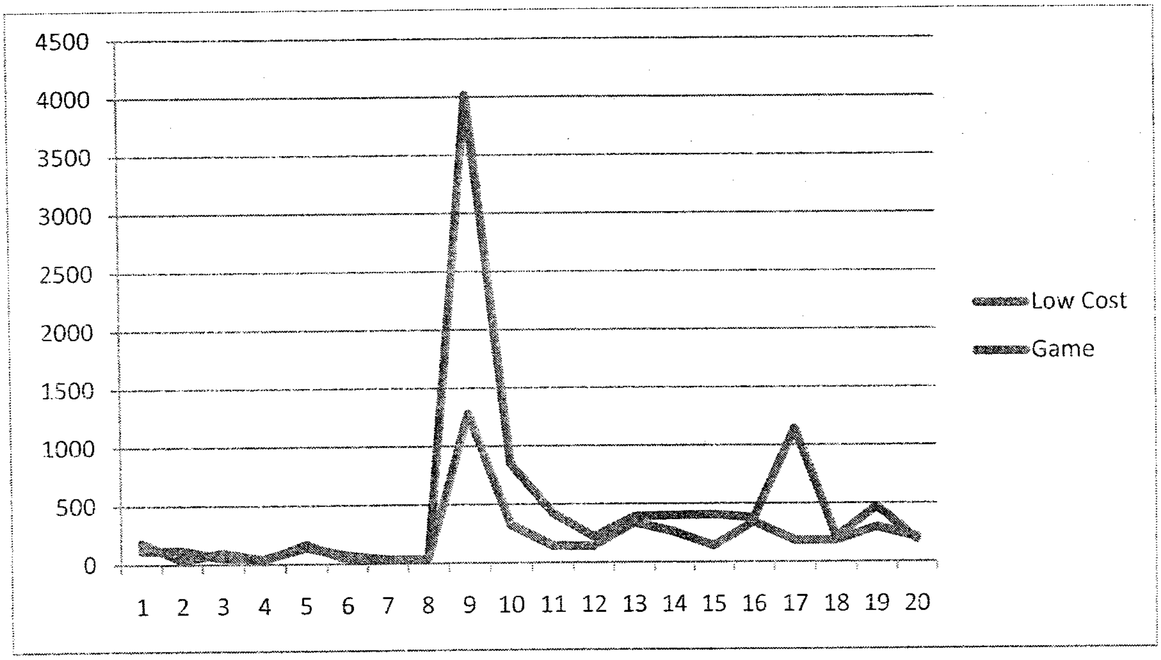

Figure 13. Number of generated non-unique sequences for the Low-Cost and Game environments per practice exercise

Exercises 1 and 5 were very similar, with a small variation in one step required to complete the task, the type of keyframe. Exercise 9 was the first significant change in the exercises, triggering an increase in action variations in both environments but more noticeable in the Game environment. Exercise 13 sees both groups at the same level, whereas exercise 17, the first full long sequence exercise, triggers again an increase in action variation which is once more larger for the Game group. Both groups end up at a similar level at exercise 20.

In the initial exercises the Game group made more mistakes. By exercise 13, participants in the Low-Cost group 'streamlined' significantly their approach as evidenced by the fact that the actual support level (number of people that share a sequence) was higher. That is, more people performed the same actions in this group whereas the Game group still had considerable variation. Errors were still quite prevalent in exercise 17 in the Game group, while less than 22% (4 people) accounted for the errors generated in the Low-Cost environment. By exercise 20 the Game group has reached a lower number of errors than the Low-Cost group, and even though it reached a more unified strategy (less variation in sequences) the Low-Cost group shows full unification of tactic (all 18 people shared the same sequences). These observations were supported by the analysis of the error support level and frequency that showed a significant peak in the Game group for exercises 9 and 17, coupled with standard panel use.

This is an interesting result suggesting that the Game group might have followed a trial-and-error, cue-based approach to learning the sequences because the timer put considerable pressure on speed and accuracy. On the contrary, the Low-Cost group seemed to have taken a more discovery oriented approach because they had as much time as they wanted to explore each of the exercises and the environment. This observation seems to be supported by the higher use of panels in the Low-Cost group.

Contrary to the popular belief that only a slow declarative-to-procedural process takes place while learning sequences, research on interactive skills (Fu & Anderson, 2008) suggests that in learning action sequences that depend on cues on the interface, reinforcement learning may be responsible for the correct use of external cues and takes place at the same time as the slower declarative memory encoding procedure. Furthermore, reinforcement learning can take place even in the presence of a demanding secondary task. Fu and Anderson also point out that the procedural knowledge acquired through reinforcement learning will be of a tacit kind and difficult to verbalize.

Even though Fu and Anderson considered reinforcement learning present in probabilistic sequences with no immediate feedback, the results in this study suggest that this dual learning process could have taken place here: a simultaneous memory encoding and reinforcement learning process in the Low-Cost environment, and a predominantly reinforcement learning process in the Game environment. This would explain why the Game group had the lowest scores in the multiple-choice questions but the highest speed and accuracy in the simulator exercises (those who finished the game). They were able to react quickly to cues but were not so good at verbalizing their knowledge. It is likely that the timer played the role of the demanding secondary task in the Game group, modifying the learning strategy. As one user put it: "The time-limit feature renders the exercise rather frantic. Instead of allowing the information to be acquired and understood, it becomes a race to reach the next element in the exercise. It is effective, and I can see how the game situation would have one learn differently than a more relaxed exercise."

The higher number and type of errors in the Game group when a new task was encountered in the practice environment suggests a more mechanical strategy that was optimized with the following exercises of the same kind. This agrees with Gray and Fu (2006) who noted that in interactive behavior where information access cost is high people do not fully explore the task space and stabilize at a suboptimal level of performance. This is not to say that there was no improvement in strategy. In fact, the improvement in the Game group between exercise 17 and 20, which are all of the same kind, was much greater than the Low-Cost group. In the last exercise Game participants showed no errors above the minimum support level specified, whereas in the Low-Cost group, 22% of people still made an E1 error (wrong command in the sequence).

When encountering a new task however, the Game group seemed to prefer a trial-and-error approach rather than looking for helpful information in the environment. In a similar manner to the situation found by Maglio and Kirsh (1996) where Tetris players chose motor actions rather than rotating the pieces in their heads because it was easier, players here seemed to rely on lower-cost memory or perceptual skills, whether accurate or inaccurate, than incur in the cost of opening the panels. As stated by Gray and Fu (2006):

> In other words, information-seeking costs may be affecting how likely people will be to use more efficient procedures in computer applications or other interactive environments. Our results therefore suggest an important warning

> for designers of interactive applications –even small differences in information-seeking costs may affect long-term learning and performance (p. 234).

It has also been suggested that in sequences that are predictable and follow a specific order, in other words non-probabilistic, two independent forms of learning, attentional and non-attentional, take place (Curran & Keele, 1993). Non-attentional refers to a reduced level of attention in the presence of another task. If the attentional form is disabled because of distraction, the non-attentional form remains unaffected. Furthermore, Curran et al. proposed that attentional learning would be required to parse similar chunks in the sequences that would lead to hierarchical representations. On the contrary, non-attentional learning would use simple associations.

When items in the sequences are not always followed by the same element (pairwise variation), it is hypothesized that hierarchical representations or plans would be necessary to determine the correct order (Cohen, Ivry, & Keele, 1990). This would explain the larger increase in sequences when new exercises were encountered in the Game group, where the timer would act as a distraction, triggering a non-attentional form of learning. No clear rules would also explain the lower level of performance in the multiple-choice test, where it was necessary to reconstruct the sequences without the simulator.

Task Representation

The goal of the game board was to provide an alternative simultaneous representation of the tasks by showing the first and last states of the animation. The other objective was to provide a relevant unifying context for the tasks done in the simulator. It was expected that this would impact the strategies used as well, as shown in previous studies (Zhang, 1997).

Nevertheless this attempt seems to have been rather unsuccessful. Some participants found it confusing or useless: "I'm not sure I really understood the purpose of the row of circle and squares at the top as the timer was going. It actually gets in the way", and "To my view the training area designed for the trainee to exercise after the lessons had many unnecessary elements in it (like moving circles and squares on the top)". Apart from the information provided in the tutorial, the game did not allow much time for the participants to familiarize themselves with the meaning of the board. It seems this part of the environment was mostly ignored by the majority of people and they chose to focus on the tasks, which would explain why the game help buttons were accessed only a couple of times.

It is possible that a few users understood the board as there were sequences in the Game group where the Task Description panel was accessed only after a couple of correct commands were executed. Without accessing information in the panels this was not possible. However, most people missed the purpose of the board.

Reported Satisfaction

Satisfaction levels were quite high for all environments but significantly high for the Low-Cost environment (M = 4.8, SD = 0.25). The fact that exercises in this environment looked more like drills was not an issue for users. Even though the Game environment was the most problematic one, it did score an overall 4.19 with all participants included which is not bad taking into consideration that a few people were unable to finish it.

High satisfaction does not necessarily mean better performance or effective use of the environment (Dillon, 2000). In spite of the variations in performance and use, all environments were reasonably well accepted by users.

Implications

The study informs various areas of design of online instructional training environments. The main conclusion is that information access cost in online spaces has an impact on the learning process of interactive behaviors. Depending on the learning objective and the process that wants to be encouraged a lower or higher cost can be selected for resources present on the screen. The cost is not the only consideration but also the type of information provided and how all resources contribute to the solution of a task. Balancing the information available and its cost across the space is a key design issue.

In the specific case of games, resources have to be carefully included in the analysis, and it is important to consider how their cost and content will contribute to the overall process of information acquisition and resulting learning strategy. The presence of concurrent tasks in the game, which in this case was a simple timer, can foster memory-intensive or perceptual-motor intensive strategies, with specific learning outcomes. The simplicity of the game used in this study suggests that coordination of learning task and gameplay is not a trivial issue and require an analysis grounded in learning, human-computer interaction and information theories.

Educational games might not be a solution for everybody. As opposed to traditional drills, some learners might find the dynamics of a game confusing and stressing. In contrast to commercial games, a more progressive introduction to the game objective might ease the process for users who encounter difficulties. Another consideration is whether the performance and learning outcomes are worth the development time and cost incurred in designing a complex game. This is a question that has to be answered in each individual case depending on specific needs.

Finally, individual characteristics may help determine what the best environment is for a specific person but unfortunately this study cannot shed any light in how these differences are related to successful use and performance.

Limitations

The research presented here applies to simple sequence learning in software applications. It generalizes to most software applications where simple command sequences are found. There are however limitations in this research in different areas, as will be discussed next.

Learning Processes

It is believed that regardless of the type of software, the learning processes evidenced here, attentional and non-attentional, will be indeed present in interactive environments. Previous research and this study seem to point in that direction. However this applies to simple sequences only. Other, more complex tasks may evidence different patterns of use and/or learning dynamics. It was possible to add more difficult tasks to the simulator, but it is likely that if they had been included less people would have finished the game. The type of exploratory learning that many users choose in the real world is important and widely used, but it is not addressed here either. Another unresolved issue here is that the tasks are software-specific. It is not clear whether the capacity to formulate rules for these sequences can be transferred or even be useful in other software applications such as those that are text-based.

Software

The software selected for the simulator was Adobe Flash. Even though this is a popular software application, each computer program has its own design philosophy behind it. In this case it is the frame-by-frame animation technique. Whereas it is true that simple sequences will be present across applications, design rationale behind the application may make other processes and resources more important, thus generating different patterns.

Game Design

Some features in the Game environment would have been desirable but were not included in order to allow direct comparisons with other environments. In particular it would have been possible to foster competition between players. However, it was thought that some users would have become very anxious if they compared their score and time with other, more skilled players. In view of the fact that quite a few people withdrew from the game, this would have been a likely scenario. Nevertheless, competition and collaboration are very important in learning and it would be interesting to see how the learning patterns are affected, if at all.

Resources and Access Cost

Resources were limited to those proposed in the DIRM (Wright et al., 2000). Other types of resources can be made available to the user and may produce different patterns of use. Regarding how these resources were accessed, this analysis was restricted to mouse clicks and although it seems that the importance of information access cost can be generalized to other scenarios, it is not certain that different forms of interaction will lead to identical results.

Data Mining Techniques

Although an innovative point in this study is the data mining analysis, which made possible to understand results that would have otherwise remained without a readily apparent source, the techniques used were relatively simple and conservative as the researcher is not a data mining expert.

Resources/Information Access Cost Design Framework

The results obtained shed some light into the processes that take place when learning command sequences in interactive environments. Instructional designers can benefit from these results, but it is useful to specifically articulate guidelines that can be used for other projects. In addition, many design decisions taken were based on the need to answer specific research questions. Many of the features would have to be modified in a real-world learning intervention. The purpose of this section is to present a design framework that can be used when creating interactive environments for training, particularly games.

Task to Resources Mapping

Several resources can be made available on the screen. In this study the selection of resources was informed by the Distributed Information Resources Model (Wright et al., 2000). This set of resources included: goal, task, and history/plan. The feedback panel was added to help the learner when mistakes were made but was not used frequently by participants.

Even though the DIRM was useful to analyze interactive behavior, the resources in this model are more suited to help someone complete a task but not necessarily acquire knowledge. The DIRM does not provide any guides to map the resources to the learning task. For example, the history/plan panel offers a step-by-step solution to the task at hand, but it does not structure the information in any particular way.

In the case of command sequences, the information to learn the tasks (not just complete them) can be organized in a more optimal way. In general, tasks have recurrent and non-recurrent aspects (van Merrienboer et al., 2003). Rules can be created out of these aspects so that they can be used in new situations. Noncurrent elements are the pieces of information that vary from task to task, and may or may not be necessary to learn. In the particular case of this study, the length of the animation is a noncurrent aspect, one that only needs to be understood but not memorized. The same applies to the initial and final states of the animation. Taking these elements into consideration, tasks can be broken down into their recurrent and nonrecurrent aspects, and the type of resources and cost can be determined.

Let S_i be a task, a general sequence of elements or commands:

$$S_i = \{e_1, e_2, e_3, \dots, e_i\}$$

where each element e_i may be part of a recurrent or nonrecurrent aspect of the general task. For example, the task "complete the animation of a red circle into a green circle in 1 second, with a frame rate of 12 fps" has the following steps that can be broken down into recurrent and nonrecurrent aspects as shown in Table 41.

Table 41

Recurrent and Nonrecurrent Aspects of the Sample Task

Step	Recurrent	Nonrecurrent
Select frame 12	Select final frame	1 second or frame 12
Select Insert > Keyframe	Same geometry, so insert a keyframe	N/A
Click on Color Swatch	Click Color Swatch	Green
Click on any frame between 1 and 12	Click on a frame between the first and last keyframes	N/A
Select Tween > Shape Tween	Change of color or shape require Shape Tween	N/A

The first column was the information displayed in this study in the History/Plan panel. It certainly allows the user to complete the task but makes no difference between the elements that repeat throughout all similar tasks and those that change. One user complained that the Task panel had to be opened every single time a new exercise was started only to get the minimum information to complete the task. This was required to actually see if the user consulted this panel. It would not have been necessary if an eye-tracker had been used. The user is correct to suggest that the nonrecurrent aspects of the task, in this case 1 second and Green, should be made readily available. This information changes for every task and although it is necessary to complete the animation, it is information that can be discarded after completion. The immediate availability of this information in the game would have saved significant time for players. Those unfamiliar with the task or having difficulties with completing it could access the information regarding the recurrent aspects to remember the general rule. As a last resource the History/Plan panel, as was used in the study, could guide the user step-by-step through the exercise. Thus, each tasked can be mapped to the following resources:

1. Task description (semantic)
2. General rule (recurrent aspects of task)
3. Non-recurrent aspects of task (in this case time, shape, or color). This could be represented in a graphic shortcut in this case as shown in figure 14, making perceptual recognition easier. This representation also shows the initial and goal states, as suggested in the DIRM.

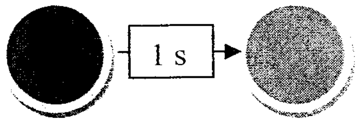

Figure 14. Graphical representation of task

Information Depth

Later tasks that include previous rules do not need to include already learned information, in agreement with the suggested approach in the 4C/ID framework of providing just the necessary information at each level (van Merrienboer et al., 2002). Information can be nested and made available only if the user requests it.

Take a complete sequence $S_{complete}$:

$$S_{complete} = \{e_1, e_2, \dots, e_i, e_j, \dots, e_m\}$$

Where part of the sequence has already been learned:

$$S_{learned} = \{e_1, e_2, \dots, e_i\}$$

And the rest is new:

$$S_{new} = \{e_j, \dots, e_m\}$$

Then

$S_{complete} = S_{learned} + S_{new}$

$S_{learned}$ can be simply summarized in one line, and the new part of the rule can be presented in a more detailed manner as in the previous section. Because of the interactive nature of digital media, with a click of a button the learned general procedure can be expanded to have the full information displayed. Other items can be nested producing a hierarchy of information. At a superficial level the user will find what is needed immediately, and more specific information will be provided at deeper levels which will naturally have a higher access cost and foster encoding. Figure 15 shows a sample of the information depth concept.

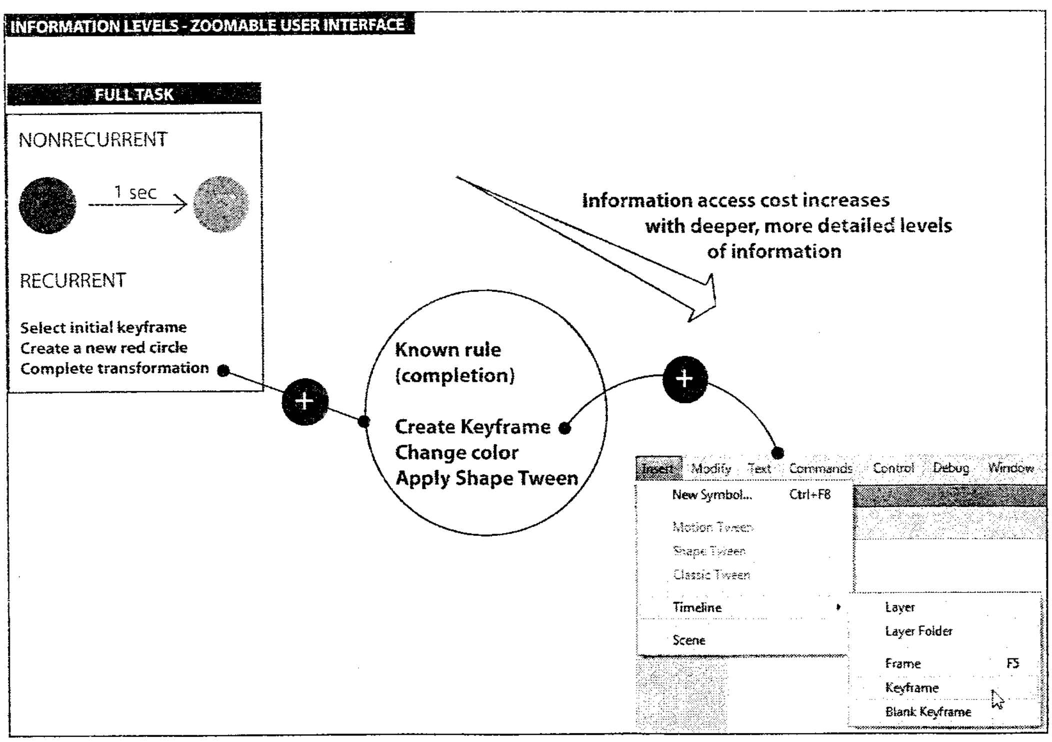

Figure 15. Information depth

The concept of layers of information is not new in computer systems. The screen has limited real estate and as the amount of information increases, it is also necessary to come up with ways to allow for different layers of complexity. One solution to this problem comes from information visualization in the form of zoomable interfaces, where detail is added as the user clicks on an item and the interface zooms in. Two important requirements of these interfaces from the point of view of instructional design are that they must support hierarchical grouping and multiple representations (Bederson, Meyer, & Good, 2000). Indeed, an instructional task can be represented at different levels of granularity and complexity, but remain equivalent, as proposed by Structural Analysis (Scandura, 2001).

Availability of information in layers forces the designer to carefully map the different resources to the task at hand, segmenting the information in a way that will suit different levels of expertise. The additional time required to perform this analysis will make the system more usable and address complaints from some more experienced learners who wanted to work faster. The panels will still offer full information for those who struggle and who are willing to pay the access cost to get more help. Notice that unlike user interface design, where the objective is to facilitate access to commands and speed up the task, the objective of an instructional analysis is to provide suitable and varied levels of scaffolding but not to ease the task so much that nothing will be learned. From the development point of view, this analysis also requires the designer to work closely with the programmer of the system. The use of reusable objects becomes essential if development time and cost are to be kept at reasonable levels.

Access Cost

As mentioned above, the organization of the information contained in panels, videos or any other type of resource is not the only step to carry out. Having nested layers of information also requires from the designer decisions regarding the access cost of items from the perceptual and motor point of view, as well as additional hidden costs imposed by the system (e.g. higher cost in the Game environment due to time lost).

Take for example the Feedback panel in the study. The information in the panel was quite important as it told the user exactly what was wrong in the step taken. However, it had to be opened by clicking on the upper bar, an identical cost to opening the History/Plan panel, which not only hinted what was wrong but also showed the detailed plan to complete the task. By performing the same motion, more information was obtained from the History/Plan panel than from the Feedback panel. Some users noted this and suggested that the Feedback panel should pop open as soon as a mistake was made. This could be a good idea, but then experienced users, who are likely to still make mistakes (Kitajima & Polson, 1995) would have been annoyed by information that was not necessary to them. One possible solution is to allow the panel to pop open the first time and then give options to the user. These can include: Don't show again, Minimize, etc.

Such options allow customization of the environment where the user has some saying regarding which resources he/she wants close and accessible, and which are not necessary. These options are normally available in software applications and games, and can also be useful in learning environments.

Access cost has to be carefully matched with the kind of learning that is required. Preliminary research confirmed by this study suggests two types of processes in sequence learning in interactive environments but more may appear as interfaces and controls evolve. For the time being, it seems clear that the presence of secondary tasks that increase access cost (as in the Game environment) will foster a non-attentional, cue-based, reinforced-based type of learning, whereas a low-cost environment that allows the user to focus on one task at a time will foster a procedural-to-declarative process. This is an important consideration at the time of design. It also agrees with a layered help structure. The more advanced the task, the more layers below and the more effort to reach those lower levels. The Soft constraints Hypothesis will predict learning of the material that is harder to reach, as long as no distractions are present.

Interactivity and Agency

It was mentioned before that the game board as an external representation of the task was not successful. In retrospective a key property of interactive environments was ignored: agency. The only control the player had in the game was to complete the task as fast as possible to beat the timer. Other game resources became irrelevant because they did not offer any possibility to contribute to the game goal, and they could not be controlled by the user. Since many users did not understand the role of the game board (a simplified representation of each task), the stated game goal (empty the board lines before the board is full) did not register in the user's mind. It was instead replaced by the user's own goal: complete the task before running out of time.

The issue of control and agency over assets is important. Observe in figure 16 the interactivity map of the Game environment as was used in this study. Notice the disconnection between the game assets on the top part of the screen and the Flash simulator on the lower area. Interaction was mostly restricted to the lower area of the screen. Only saccades were necessary to watch the timer.

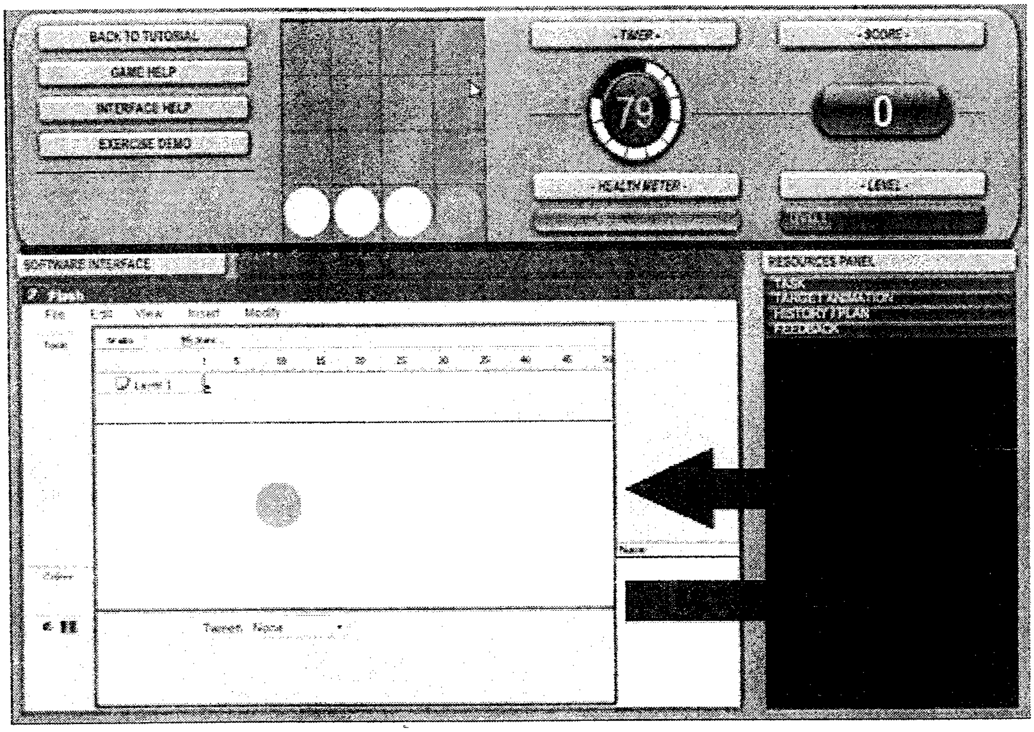

Figure 16. Interactivity map for the Game environment

The separation between the two areas was useful for the study because it allowed direct comparison between the Low-Cost and the Game environments, and it made evident that two different learning processes took place by “attaching” a game component. However, it made the game “sterile”, as a user called it.

An obvious solution to this problem would be to increase interaction in the upper levels of the screen. For example, the board could be a set of different geometric figures, as in the original board, but would also be a place where the user could select the initial and final states of the animation. At least the user would be forced to click on the board items and decide which animations he/she would try first. If the designer still wanted the timer, then other game resources could be used to add more time. For example, reaching a certain score would give the user 100 seconds more, or maintaining a certain health level would also add a bonus. Task resources could also be linked to game resources. For example, accessing only the general rule of the task would give the user a time bonus at the end of the task, but this bonus would not be present if the user opened the History/Plan panel. Careful manipulation of game points, time, and information access cost would foster focusing on certain resources and not others depending on the learning goal.

Game Difficulty and Progression

Another important design issue is the difficulty level of the game. In this study, some people did not benefit from the game at all. Those who did not finish had the lowest level of procedural performance. Participants did not appreciate that the entire game had to be started from the beginning if they ran out of time. The purpose of this feature was to compare final times between exercise environments, but would not be recommended in a real instructional environment.

It does not seem easy to determine the progression of the game based on an initial questionnaire to determine individual characteristics. This is not an approach used in commercial games, but then again in that scenario players choose to either play the game or not. Instructional games are expensive to develop and most of them do not have such a large potential target group. It would be expected that all those who need to learn a specific skill would complete the game. A withdrawal rate and performance difference as encountered in the study would be unacceptable in a real training program.

Rather than relying on individual characteristics, it seems more feasible to allow the game to adapt as the learner plays it. However, previous efforts based on artificial intelligence algorithms in learning adaptive systems have not been completely successful (Park et al., 2003). Instead of systems based on specific characteristics or aptitudes, Park et al. suggest a response-sensitive method. Traditionally, this has been attempted using branching, but with very low complexity the tree becomes very large and unmanageable (Crawford, 2005). An alternative solution now that computer power has increased is to use models created from user data using data mining techniques. Data mining can be used to create a model that learns as more users play with the system and offers options to different users according to their responses and selections. An open source application such as RapidMiner offers the possibility of creating such a model but being resource-intensive, it would be advisable to initially run the learning environment using a designer model. After enough data has been collected, the model can be refined iteratively until a satisfactory solution has been found.

Summary

Figure 17 summarizes the different elements of the suggested framework. Whereas all elements are important, this study has focused on the resources that can be made available, their access cost, and their impact on learning effectiveness, efficiency, and the resulting learning processes. A traditional environment will naturally omit the gaming components.

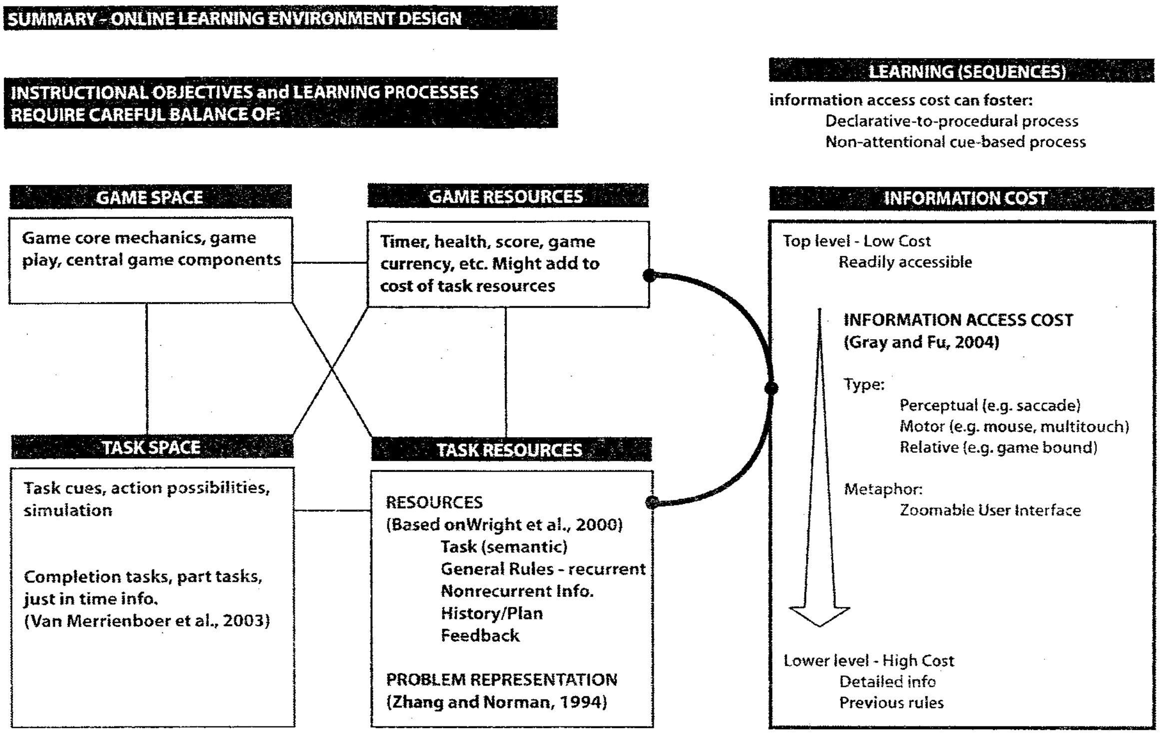

Figure 17. Design framework

Contribution to Knowledge

The results presented here support previous research on sequence learning in general (Curran et al., 1993; Gray et al., 2006), and more specifically in interactive environments. They also point out the need to tailor information resources more specifically to learning environments were scaffolding is needed rather than immediate effortless information to complete the task. An important difference is that participants used a detailed simulation of a software application as opposed to a simplified task, so the findings are based on an analysis of interaction that is very close to the real setting. Even though data mining has been previously used in educational research in an effort to tailor the environments to the learner (Shen et al., 2003), a strong point of this research is the use of sequence analysis to uncover learning strategies that can be linked to specific learning outcomes.

At a larger scale, the results will hopefully help instructional designers better design and analyze learning environments where information is distributed. Although both the distributed and embodied views of cognition are appealing, there are few specific guidelines for constructing these spaces. As noted by David Kirsh: "We need better theories of how people are embedded in the world as well as better theories of how the world and the larger systems we are part of coordinate action" p. 250 (Kirsh, 2006).

Finally, although educational games have become very popular, their effectiveness is still challenged by many. The findings here suggest that it may not be an issue of a game being better than traditional instruction or vice versa, but rather that different learning dynamics take place, entailing different learning outcomes. Since instructional games have in many instances various resources and are normally combined with other ludic tasks, the study has shown that secondary tasks, information resources and their access cost should be carefully chosen since they can have an impact on the strategies used and the learning outcomes.

Future Research

This study sets the ground for more research on learning in interactive environments. The following is a non-exhaustive list of possibilities:

1. It would be interesting to see how learning dynamics are affected by more complex tasks that are not necessarily sequences but involve other actions such as entering information, manipulating virtual objects, free exploration, or collaborating with others.

2. In addition, innovative forms of interaction such as touchscreens, Wii remotes or motion sensors might offer users new possibilities and thus encourage different strategies to access information.

3. Analysis of interactions in online environments where information resources are carefully matched with tasks and offer equivalence at different levels of granularity but variation in access cost may offer further insights that can help improve online task support that leads to learning.

4. In terms of research technique, more advanced data mining techniques may be used to better analyze learner preferences and tailor the environment in real time, and may be more compatible with the approach applied in this study to track user information.

5. Finally, the framework proposed here can be further extended to analyze the information provided in educational games, the cost/benefit structure, and the way other tasks affect learning.

Conclusions

Despite of the fact that few hypotheses as initially stated were supported, the study provided strong evidence not only for the Soft constraints Hypothesis in software learning tasks, but also for context-bound dual-learning processes in command sequence learning in online environments. Information access cost determined the strategy used and the learning process triggered, and this in consequence affected the skills learned. Reinforcement, non-attentional learning was the product of a secondary task (complete the exercises in time) that increased perceived access cost, resulted in faster, more accurate task performance and was dominant in the Game group. Ability to articulate rules was more evident in the Low-Cost environment, with good levels of efficiency and accuracy.

The work presented here also supports the fact that screen resources in online instructional interventions cannot be randomly chosen and located on the screen but must be carefully selected and balanced to obtain the desired learning outcome. There is an important difference between the areas of human-computer interaction and educational technology: whereas the first strives to make the information readily available to complete the task as smoothly as possible, the second has to consider the adequate cost that will lead to effective and efficient learning.

As digital environments develop and new forms of control emerge, more research will be needed to understand the effect new forms of technology have on learning processes, and instructional effectiveness and efficiency. As we learn more about how we acquire knowledge in these environments, we are more likely to propose strong design guidelines that will allow instructional designers to develop effective technology-based interventions with more clarity about the effects of their decisions on the entire process.

References

Allen, B., Otto, R., & Hoffman, B. (2003). media as lived environments: The Ecological psychology of educational technology. In D. Jonassen (Ed.), *Handbook of Research on Educational Communications and Technology* (pp. 169-177). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Ang, C. S., Zaphiris, P., & Mahmood, S. (2007). A model of cognitive loads in massively multiplayer online role playing games. *Interacting with Computers, 19*(2), 167-179.

Apter, M. (1991). A structural phenomenology of play. In J. Kerr & M. Apter (Eds.), *Adult Play: A Reversal Theory Approach* (pp. 13-30). Amsterdam: Swets & Zeitlinger.

Ballard, D., Hayhoe, M., Pook, P., & Rao, R. (1997). Deictic codes for the embodiment of cognition. *Behavioral and Brain Sciences, 20*, 723-742.

Bannert, M. (2000). The effects of training wheels and self-learning materials in software training. *Journal of Computer Assisted Learning, 16*, 336-346.

Barab, S., & Squire, K. (2004). Design-based research: putting a stake in the ground. *The Journal of the Learning Sciences, 13*(1), 1-14.

Bartle, R. (1996). *Hearts, clubs, diamonds, and spades: players who suit MUDs*. Retrieved from http://www.brandeis.edu/pubs/jove/HTML/v1/bartle.html

Bederson, B., Meyer, J., & Good, L. (2000). *Jazz: an extensible zoomable user interface graphics toolkit in Java.* Paper presented at the Symposium on User Interface Software and Technology: Proceedings of the 13th annual ACM symposium of user interface software and technology, San Diego, CA.

Ben-Ari, M., & Yeshno, T. (2006). Conceptual models of software artifacts. *Interacting with Computers, 18*, 1336-1350.

Bishop-Clark, C., Dietz-Uhler, B., & Fisher, A. (2007). The effects of personality type on web-based distance learning. *Journal of Educational Technology Systems, 35*(4), 491-506.

Blackmon, M., Kitajima, M., & Polson, P. (2005). *Tool for Accurately Predicting Website Navigation Problems, Non-Problems, Problem Severity, and Effectiveness of Repairs.* Paper presented at the Conference of Human Factors in Computing Systems CHI 2005, Portland, OR.

Boechler, P. (2001). How spatial is hyperspace? Interacting with hypertext documents: Cognitive processes and concepts. *CyberPsychology & Behavior, 4*(1), 23-46.

Boechler, P., & Dawson, R. (2005). The effects of spatial layout on relationships between performance, path patterns and mental representation in a hypermedia information search task. *Interactive Technology & Smart Education, 2*, 31-45.

Bonnie, J., & Kieras, D. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Human-Computer Interaction, 3*(4), 320-351.

Borgman, C. L. (1986). The user's mental model of an information retrieval system: An experiment on a prototype online catalog. *International Journal of Man-Machine Studies, 24*, 47-64.

Braskamp, L., & Hodgetts, R. (1971). The role of an objective evaluation model in simulation gaming. *Simulation & Games, 2(2),* 197-212.

Bredemeier, M., & Greenblat, C. (1981). The Educational effectiveness of simulation games: A synthesis of findings. *Simulation & Games, 12*(3), 307-332.

Brown, J., Collins, A., & Duguid, P. (1989). Situated cognition and the culture of learning. *Educational Researcher 18*(1), 32-42.

Bush, V. (2003). As We May Think. In N. Wardrip-Fruin & N. Montfort (Eds.), *The New Media Reader* (pp. 35-47). Cambridge, MA: The MIT Press.

Butler, R., Markulis, P., & Strang, D. (1988). Where are we? An analysis of the methods and focus of research on simulation gaming. *Simulation & Games, 19*(1), 3-26.

Carroll, J. (1997). *Minimalism: Beyond the Nurnberg Funnel.* Cambridge, MA: MIT Press.

Carroll, J., & Carrithers, C. (1984). Training wheels in a user interface. *Communications of the ACM, 27*, 800-806.

Carroll, J., Smith-Kerker, P., Ford, J., & Mazur-Rimetz, S. (1987). The minimal manual. *Human-Computer Interaction, 3*, 123-153.

Cassidy, S., & Eachus, P. (2002). Developing the Computer User Self-Efficacy (CUSE) scale: Investigating the relationship between computer self-efficacy, gender and experience with computers. *Journal of Educational Computing Research, 26*(2), 133-153.

Chandler, P., & Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and Instruction, 8*(4), 293-332.

Chandler, P., & Sweller, J. (1996). Cognitive load while learning to use a computer program. *Applied Cognitive Psychology, 10*, 151-170.

Charney, D., Reder, L., & Kusbit, G. (1990). Goal setting and procedure selection in acquiring computer skills: A comparison of tutorials, problem solving, and learner exploration. *Cognition and Instruction, 7*(4), 323-342.

Clark, A. (1999). An embodied cognitive science? *Trends in Cognitive Sciences, 3*(9), 345-351.

Cohen, A., Ivry, R., & Keele, S. (1990). Attention and structure in sequence learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 16*(1), 17-30.

Compeau, D., & Higgins, C. (1995). Computer self-efficacy: Development of a measure and initial test. *MIS Quarterly, 19*(2), 189-211.

Coulson, A. (1991). Cognitive synergy in televised entertainment. In J. Kerr & M. Apter (Eds.), *Adult Play: A Reversal Theory Approach* (pp. 71-85). Amsterdam: Swets & Zeitlinger.

Crawford, C. (2005). *Chris Crawford on Interactive Storytelling*. Berkeley, CA: New Riders.

Cullen, R. (2001). *Addressing the Digital Divide.* Paper presented at the Libraries and Librarians: Making a Difference in the Knowledge Age. 67th General Conference, Boston, MA.

Curran, T., & Keele, W. (1993). Attentional and nonattentional forms of sequence learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 19*(1), 189-202.

Dalkir, K. (1997). *Improving user modeling via the integration of learner characteristics and learner behaviors.* Unpublished Doctoral Thesis, Concordia University, Montreal.

Damper, R. (1997). Connecting perception to cognition. *Behavioral and Brain Sciences, 20*(4), 744-745.

De Vries, J., & Bersin, J. (2004). *Rapid E-Learning: What Works*. Retrieved from http://download.macromedia.com/pub/breeze/whitepapers/bersin_elearning_study.pdf

Dempsey, J. (1994). Since Malone's theory of intrinsically motivating instruction. What's the score in gaming literature? *Journal of Educational Technology Systems, 22*(2), 173-183.

DeNike, L. (1976). An exploratory study of the relationship of educational cognitive style to learning from simulation games. *Simulation & Games, 7*(1), 65-74.

Dickinson, J., & Faria, A. (1997). A random strategy criterion for validity of simulation game participation. *Simulation & Games, 28*(3), 263-275.

Dillon, A. (2000). Evaluating the user experience of information spaces, *Argus Center for Information Architecture Conference*. La Jolla, October 25-27th.

Dillon, A., & Gabbard, R. (1998). Hypermedia as an educational technology: A review of the quantitative research literature on learner comprehension, control, and style. *Review of Educational Research, 68*(3), 322-349.

Driscoll, M. (1994). *Psychology of learning for instruction*. Boston, MA: Allyn and Bacon.

Driscoll, M., & Dick, W. (1999). New research paradigms in instructional technology: An inquiry. *Educational Technology Research and Development, 47*(2), 7-18.

Dukes, R., & Waller, S. (1976). toward a general evaluation model for simulation games: Gem. *Simulation & Games*(March 1976).

Ehret, B. (1999). *Learning where to look: The acquisition of location knowledge in display-based interaction.* Unpublished Doctoral Dissertation, George Mason University.

Engelbart, D. (2003). Augmenting human intellect: A conceptual framework. In N. Wardrip-Fruin & N. Montfort (Eds.), *The New Media Reader* (pp. 93-108). Cambridge, MA: The MIT Press.

Epelboim, J. (1997). Deictic codes, embodiment of cognition, and the real world. *Behavioral and Brain Sciences, 20*(4), 746.

Farris, J., Jones, K., & Elgin, P. (2002). User's schemata of hypermedia: what is so 'spatial' about a website? *Interacting with Computers, 14*, 487-502.

Feinstein, A., & Cannon, H. (2002). Constructs of simulation evaluation. *Simulation & Gaming, 33*(4), 425-440.

Feldman, J. (1997). Embodiment is the foundation, not a Level. *Behavioral and Brain Sciences, 20*(4), 746-747.

Fu, W. T., & Anderson, J. (2008). Dual Learning processes in interactive skill acquisition. *Journal of Experimental Psychology, Applied, 14*(2), 179-191.

Gabadinho, A., Ritschard, G., Studer, M., & Muller, N. S. (2008). *Mining Sequence Data in R with the TraMineR Package: A User's Guide*. Retrieved from http://mephisto.unige,ch/traminer

Garcia, T. (2005). The making of the motivated strategies for learning questionnaire. *Educational Psychologist, 40*(2), 117-128.

Gee, J. P., Squire, K., & Steinkuehler, C. (2005). *Discover how games are reshaping business & learning.* Paper presented at the Accelerate Madison, Madison, WI.

Gellevij, M., van der Meij, H., de Jong, T., & Pieters, J. (1999). The effects of screen captures in manuals: a textual and two visual manuals compared. *IEEE Transactions on Professional Communication, 42*(2), 77-91.

George, C. (2000). Experience with novices: The importance of graphical representations in supporting mental models. In A. Blackwell & E. Bilotta (Eds.), *Proceedings of the 12th Workshop of the Psychology of Programming Interest Group* (pp. 33-44). Cozenza, Italy.

Goldberg, L. R. (1992). The development of markers for the big-five factor structure. *Psycholgical Assessment, 4*, 26-42.

Gollwitzer, P., & Schaal, B. (1998). Metacognition in action: The importance of implementation intentions. *Personality and Social Psychology Review, 2*(2), 124-136.

Gosen, J., & Washbush, J. (2004). A review of scholarship on assessing experiential learning effectiveness. *Simulation & Games, 35*(2), 270-293.

Gray, W., & Boehm-Davis, D. (2000). Milliseconds matter: an introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied, 6*(4), 322-335.

Gray, W., & Fu, W. T. (2004). Soft constraints in interactive behavior: the case of ignoring perfect knowledge in-the-world for imperfect knowledge in-the-head. *Cognitive Science, 28*(3), 359-382.

Gray, W., & Sabnani, H. (1994). *Why you can't program your vcr, or, predicting errors and performance with production system models of display-based interaction.* Paper presented at the CHI'94, Boston, MA.

Gray, W., Sims, C., Fu, W. T., & Schoelles, M. (2006). The soft constraints hypothesis: A rational analysis approach to resource allocation for interactive behavior. *Psychological Review, 113*(3), 461-482.

Grodal, T. (2003). Stories for eye, ear, and muscles: video games, media, and embodied experiences. In M. Wolf & B. Perron (Eds.), *The Video Game Theory Reader* (pp. 129-155). New York, NY: Routledge.

Hargis, J. (2001). Can students learn science using the internet? . *Journal of Research on Computing in Education, 33*(4), 475-488.

Hawthorn, D. (2005). *Training wheels for older users.* Paper presented at the Annual Conference Computer-Human Interaction Special Interest Group, Canberra, Australia.

Henning, P. (2003). Everyday cognition and situated learning. In D. Jonassen (Ed.), *Handbook of Research on Educational Communications and Technology* (pp.143-168). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Hinesley, G. (2005). *The impact of graphical conventions and layout location on search for webpage widgets.* Unpublished Doctoral Dissertation, University of Colorado, Boulder.

Horsky, J. (2005). *Distributed cognitive resources: The analysis of human interaction with a complex clinical information system.* Unpublished Doctoral Thesis, Columbia University, New York.

Howes, A., & Payne, S. (1990). Display-based competence: Towards user models for menu-driven interfaces. *International Journal of Man-Machine Studies, 33*(6), 637-655.

Hung, J.-L., & Crooks, S. (2009). Examining online learning patterns with data mining techniques in peer-moderated and teacher-moderated courses. *Journal of Educational Computing Research, 40*(2), 183-210.

Hutchins, E. (1995). How a cockpit remembers its speeds. *Cognitive Science, 19*(3), 265-288.

Hutchins, E. (2000). *Distributed Cognition.* Retrieved from http://eclectic.ss.uci.edu/~drwhite/Antrhro179a/DistributedCognition.pdf

Hutchins, E., & Klausen, T. (1996). Distributed cognition in an airline cockpit. In Y. Engeström and D. Middleton (Eds.) *Cognition and communication at work (*pp. 15-34). New York: Cambridge University Press.

Jenkins, H. (2004). Game design as narrative architecture. In Noah Wardrip-Fruin and Pat Harrigan (Eds.), *First person: New media as story, performance, game* (pp. 118-130). Cambridge: MIT Press.

John, B. (1995). Why GOMS? *Interactions, 2*(4), 80-89.

Juttner, M. (1997). On Learning and Shift (In)variance of pattern recognition across the visual field. *Behavioral and Brain Sciences, 20*(4), 751-752.

Kieras, D., & Bovair. (1984). The role of a mental model in learning to operate a device. *Cognitive Science, 8*, 255-273.

Kirschner, P., Sweller, J., & Clark, R. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist, 41*(2), 75-86.

Kirsh, D. (2006). Distributed cognition: A methodological note. *Pragmatics & Cognition, 14*(2), 249-262.

Kirsh, D., & Maglio, P. (1994). On distinguishing epistemic from pragmatic action. *Cognitive Science, 18*(4), 513-549.

Kitajima, M., & Polson, P. (1995). A comprehension-based model of correct performance and errors in skilled, display-based, human-computer interaction. *International Journal of Human-Computer Studies, 43*, 65-100.

Koster, R. (2004). *Theory of Fun for Game Design*: Paraglyph.

Larkin, J. (1989). Display-based problem solving. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A. Simon* (pp. 319-341). Hillsdale, NJ: Lawrence Erlbaum.

Larkin, J., & Simon, H. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science, 11*, 65-99.

Larose, D. (2005). *Discovering knowledge in data: an introduction to data mining.* Hoboken, NJ: John Wiley & Sons, Inc.

Laws, R. (2002). *Robin's laws of good game mastering*: Steve Jackson Games.

Lazonder, A. (1994). *Minimalist computer documentation: A study on constructive and corrective skills development.* Unpublished Doctoral Thesis, University of Twente, Enschede.

Leutner, D. (2000). Double-fading support -a training approach to complex software systems. *Journal of Computer Assisted Learning, 16*, 347-357.

Lewis, C., & Reiman, J. (1994). *Task-centered user interface design: A practical introduction.* Retrieved from http://www.hcibib.org/tcuid/tcuid.pdf

Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions of use. *International Journal of Human-Computer Interaction, 7*(1), 57-78.

Lippert, S., & Granger, M. (1998). *Tired of teaching software applications?* Paper presented at the 13th Annual Conference of the International Academy for Information Management, Helsinki, Finland.

Liu, M. (2003). Enhancing learner's cognitive skills through multimedia design. *Interactive Learning Environments, 11*(1), 23-39.

Lu, J., Yu, C.-S., & Liu, C. (2003). Learning style, learning patterns, and learning performance in a WebCT-based MIS course. *Information & Management, 40*(6), 497-507.

Maglio, P. (1997). Beyond embodiment: Cognition as interactive skill. *Behavioral and Brain Sciences, 20*(4), 753-754.

Maglio, P., & Kirsh, D. (1996). Epistemic action increases with skill. In *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society* (pp. 391-396). Mahwah, NJ: LEA.

Manktelow, K., & Jones, J. (1987). Principles from the psychology of thinking and mental models. In M. Gardiner & B. Christie (Eds.), *Applying Cognitive Psychology to User-interace Design* (pp. 83-117). Chichester: John Wiley & Sons.

Markov, Z., & Larose, D. (2007). *Data mining the web: Uncovering patterns in web content, structure and usage*. Hoboken, NJ.: John Wiley & Sons, Inc.

Martens, R., Gulikers, J., & Bastiaens, T. (2004). the impact of intrinsic motivation on e-learning in authentic computer tasks. *Journal of Computer Assisted Learning, 20*, 368-376.

Martin-Michiellot, S., & Mendelsohn, P. (2000). Cognitive load while learning with a graphical computer interface. *Journal of Computer Assisted Learning, 16*, 284-293.

Mayes, J., Draper, S., McGregor, A., & Oatley, K. (1988). Information flow in a user interface: the effect of experience and context on the recall of MacWrite screens. In D. M. Jones & R. Winder (Eds.), *People and Computers IV* (pp. 275-289). Cambridge, UK: Cambridge University Press.

McManus, T. F. (2000). Individualizing instruction in a web-based hypermedia learning environment. *Journal of Interactive Learning Research, 11*(2), 219-251.

Meadows, D. (2001). Tools for understanding the limits to growth: Comparing a simulation and a game. *Simulation & Gaming, 32*(4).

Miller, R., & Leroux-Demers, T. (1992). Business simulations: Validity and effectiveness. *Simulation & Games for Learning, 22*(4).

Miltiadou, M. (2001). *Motivational constructs as predictors of success in the online classroom.* Unpublished Doctoral Dissertation, Arizona State University.

Morgan, P., Waldron, S., King, S., & Patrick, J. (2007). Harder to access, better performance? The effects of information access cost on strategy and performance. In M. J. Smith & G. Salvendy (Eds.), *Human Interface, Part I, HCII 2007* (pp. 115-125). Berlin Heidelberg: Springer-Verlag.

Murphy, C., Coover, D., & Owen, S. (1989). Development and validation of the computer self-efficacy scale. *Educational and Psychological Measurement, 49*(4), 893-899.

Murray, J. (1997). *Hamlet on the holodeck: The future of narrative in cyberspace*. New York, NY: The Free Press.

Navarro-Prieto, R., & Cañas, J. (1999). *Mental representation and imagery in program comprehension.* Paper presented at the Psychology of Programming Interest Group, Granada, Spain.

Navarro-Prieto, R., & Cañas, J. (2001). Are visual programming languages better? The role of imagery in program comprehension. *International Journal of Human-Computer Studies, 54*, 799-829.

Newell, A., & Simon, H. (1972). *Human Problem Solving.* Englewood Cliffs, NJ: Prentice-Hall.

Norris, D. (1986). External validity of business games. *Simulation & Games, 17*(4), 447-459.

Norris, S., & Snyder, C. (1982). External validation of simulation games. *Simulation & Games, 13*(1), 73-85.

Paivio, A., & Csapo, K. (1973). Picture superiority in free recall: Imagery or dual coding? *Cognitive Psychology, 5*, 176-206.

Park, O., & Lee, J. (2003). Adaptive instructional systems. In D. Jonassen (Ed.), *Handbook of Research on Educational Communications and Technology* (pp. 651-684). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Payne, S. (1991). Display-based action at the user interface. *International Journal of Man-Machine Studies, 35*(3), 275-289.

Pennington, N. (1987). Stimulus structures and mental representations in expert comprehension of computer programs. *Cognitive Psychology, 19*, 295-341.

Peters, V., Vissers, G., & Heijne, G. (1998). The validity of games. *Simulation & Games, 29*(1), 20-30.

Petre, M., & Blackwell, A. (1999). Mental imagery in program design and visual programming. *International Journal of Human-Computer Studies, 51*, 7-30.

Pintrich, P., & De Groot, E. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psychology, 82*(1), 33-40.

Pollard, P. (1982). Human reasoning: Some possible effects of availability. *Cognition, 12*, 65-96.

Polleys, M. (2000). *A study of the relationships between self-regulated learning, personality, and achievement.* Unpublished Doctoral Dissertation, Auburn University.

Ragan, T., & Smith, P. (2003). Conditions theory and models for designing instruction. In D. Jonassen (Ed.), *Handbook of Research on Educational Communications and Technology* (pp. 623-650). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Randel, J., Morris, B., Wetzel, D., & Whitehill, B. (1992). The effectiveness of games for educational purposes: A review of the recent research. *Simulation & Gaming, 23*(3), 261-276.

Reimann, P., & Neubert, C. (2000). The role of self-explanation in learning to use a spreadsheet through examples. *Journal of Computer Assisted Learning, 16*, 316-325.

Remus, W. (1981). Experimental designs for analyzing data on games. *Simulation & Games, 12*(1), 3-14.

Rolfe, J. (1990). The proof of the pudding: The effectiveness of games and simulations. *Simulation & Games, 21*(2), 99-117.

Rolfe, J. (1992). Training transfer: The basis for validating effectiveness. *Simulation & Games, 22*(4), 249-259.

Rollings, A., & Morris, D. (2004). *Game architecture and design: A New edition*. Berkeley, CA: New Riders Publishing.

Romero, C., & Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert Systems with Applications, 33*, 135-146.

Ruben, B., & Lederman, L. (1982). Instructional simulation gaming. *Simulation & Games, 13*(2), 233-244.

Salen, K., & Zimmerman, E. (2004). *Rules of play: Game design fundamentals*. Cambridge, MA: The MIT Press.

Sasse, M. A. (1997). *Eliciting and describing user's models of computer systems.* Unpublished Doctoral Thesis, University of Birmingham, Birmingham.

Saucier, G. (1994). Mini-markers: A brief version of Goldberg's unipolar big-five markers. *Journal of Personality Assessment, 63*(3), 506-516.

Scandura, J. M. (2001). Structural (cognitive task) analysis: An integrated approach to software design and programming. *Journal of Structural Learning and Intelligent Systems, 14*, 433-458.

Sedig, K., Rowhani, S., & Liang, H. (2005). Designing interfaces that support formation of cognitive maps of transitional processes: An empirical study. *Interacting with Computers, 17*, 419-452.

Shen, R., Han, F., Yang, F., Yang, Q., & Huang, J. (2003). Data mining and case-based reasoning for distance learning. *Journal of Distance Education Technologies, 1*(3), 46-58.

Smith, S., Duke, D., & Wright, P. (1999). Using the resources model in virtual environment design. In M. Harrison & S. Smith (Eds.), *Workshop on User-Centered Design and Implementation of Virtual Environments* (pp. 57-72). York, England: University of York.

Spiro, R., Feltovich, P., Coulson, R., & Anderson, D. (1989). Multiple analogies for analogy-induced misconception in advanced knowledge acquisition. In S. Vosniadou & A. Ortony (Eds.), *Similarity and Analogical Reasoning* (pp. 498-531). Cambridge, England: Cambridge University Press.

Spiro, R., Feltovich, P., Jacobson, M., & Coulson, R. (1992). Cognitive flexibility, constructivism and hypertext: Random access instruction for advanced knowledge acquisition in ill-structured domains. In T. M. Duffy & D. H. Jonassen (Eds.), *Constructivism and the technology of instruction: A conversation* (pp. 57-76). Hillsdale, N.J.: Erlbaum.

Steinkuehler, C. (2006). Why game (culture) studies now? *Games & Culture, 1*(1), 1-6.

Su, J.-M., Tseng, S.-S., Wang, W., Weng, J.-F., Yang, J. T., & Tsai, W.-N. (2006). Learning portfolio analysis and mining for scorm compliant environment. *Educational Technology and Society, 9*(1), 262-275.

Sukhai, D. (2005). *The effects of web-based animated demonstration and guided simulation on time and performance in business software applications training.* Unpublished Doctoral Dissertation, University of Alabama.

Sumathi, S., & Sivanandam, S. N. (2006). *Introduction to data mining and its applications*. Berlin: Springer.

Sutherland, I. (2003). Sketchpad: A man-machine graphical communication system. In N. Wardrip-Fruin & N. Montfort (Eds.), *The New Media Reader* (pp. 109-127). Cambridge, MA: The MIT Press.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science, 12*, 257-285.

Sweller, J., & Chandler, P. (1991). Evidence for cognitive load theory. *Cognition and Instruction, 8*(4), 351-362.

Sweller, J., & Chandler, P. (1994). Why some material is difficult to learn. *Cognition and Instruction, 12*(3), 185-233.

Sweller, J., Chandler, P., Tierney, P., & Cooper, M. (1990). Cognitive load as a factor in the structuring of technical material. *Journal of Experimental Psychology: General, 119*(2), 176-192.

Torkzadeh, G., & Koufteros, X. (1994). Factorial validity of a computer self-efficacy scale and the impact of computer training. *Educational and Psychological Measurement, 54*, 813-821.

Torkzadeh, G., Koufteros, X., & Pflughoeft, K. (2003). Confirmatory analysis of computer self-efficacy. *Structural Equation Modeling, 10*(2), 263-275.

Tsai, M. H. (2004). *The effects of four different strategies of information presentation in software training.* Unpublished Doctoral Dissertation, University of Central Florida.

Tversky, A., & Kahneman, D. (1973). Availability: A heuristic for judging frequency and probability. *Cognitive Psychology, 5*, 207-232.

van der Meij, H. (2000). The Role and design of screen images in software documentation. *Journal of Computer Assisted Learning, 16*, 294-306.

Van Merrienboer, J., & Ayres, P. (2005). Research on cognitive load theory and its design implications for e-learning. *Educational Technology Research & Development, 53*(3), 5-13.

van Merrienboer, J., Clark, R., & de Croock, M. (2002). Blueprints for complex learning: The 4C/ID-model. *Educational Technology Research & Development, 50*(2), 39-64.

van Merrienboer, J., Kirschner, P., & Kester, L. (2003). Taking the load off a learner's mind: Instructional design for complex learning. *Educational Psychologist, 38*(1), 5-13.

van Merrienboer, J., & Sweller, J. (2005). Cognitive load theory and complex learning: Recent developments and future directions. *Educational Psychology Review, 17*(2), 147-177.

Van Meurs, R. (2007). *How to play the game: A study on mud player types and their real life personality traits.* Unpublished Master Thesis, Tilburg University, Tilburg.

Vogel, J., Vogel, D., Cannon-Bowers, J., Bowers, C., Muse, K., & Wright, M. (2006). Computer gaming and interactive simulations for learning: A meta-analysis. *Journal of Educational Computing Research, 34*(3), 229-243.

Waldron, S., Patrick, J., Morgan, P., & King, S. (2007). Influencing cognitive strategy by manipulating information access. *The Computer Journal, 50*(6), 694-702.

Wang, C., Deng, C., Ma, C., Hua, Q., & Dai, G. (2005). An extended resources model based on distributed cognition theory. *Journal of Software, 16*(10), 1717-1725.

Wiedenbeck, S., Zavala, J. A., & Nawyn, J. (2000). An activity-based analysis of hands-on practice methods. *Journal of Computer Assisted Learning, 16*, 358-365.

Williams, R. (1980). The ability of a simulation game to change attitudes when structured in accordance with either the cognitive dissonance or incentive models of attitude change. *Simulation & Games, 11*(2), 177-196.

Wilson, M. (2002). Six views of embodied cognition. *Psychonomic Bulletin & Review, 9*(4), 625-636.

Wilson, R. (1997). Pointers, codes, and embodiment. *Behavioral and Brain Sciences, 20*(4), 757-758.

Wolfe, J. (1985). The teaching effectiveness of games in collegiate business courses: A 1973-1983 update. *Simulation & Games, 16*(3), 251-288.

Wolfe, J., & Roberts, R. (1986). The external validity of a business management game: A five-year longitudinal study. *Simulation & Games, 17*(1), 45-59.

Wright, P., Fields, R., & Harrison, M. (2000). Analyzing human-computer interaction as distributed cognition: The resources model. *Human-Computer Interaction, 15*(1), 1-41.

Wu, C. (2003). *Visual representations and mental models: thinking about probability.* Unpublished Doctoral Dissertation, Columbia University, New York.

Young, R. M. (1983). Surrogates and mappings: Two kinds of conceptual models for interactive devices. In D. Gentner & A. Stevens (Eds.), *Mental Models* (pp. 35-52). Hillsdale, NJ: Erlbaum.

Zhang, J. (1987). Representations in problem solving. *Cognitive Science, 21*(2), 179-217.

Zhang, J. (1996). A representational analysis of relational information displays. *International Journal of Human-Computer Studies, 45*, 59-74.

Zhang, J. (1997). The Nature of external representations in problem solving. *Cognitive Science, 21*(2), 179-217.

Zhang, J. (2000). External representations in complex information processing tasks. In A. Kent (Ed.), *Encyclopedia of Library and Information Science* (Vol. 68, pp. 164-180). New York: Marcel Dekker.

Zhang, J., & Norman, D. (1994). Representations in distributed cognitive tasks. *Cognitive Science, 18*(1), 87-122.

Zhang, J., & Patel, V. (2006). Distributed cognition, representation, and affordance. *Pragmatics & Cognition, 14*(2), 333-341.

Appendix A. Demographics

The following questions will allow the researcher to know a little about you.

What is your gender? Male Female

What is your age in years? ____

What is your occupational status? Full-time
Part-time
Student
Home
Unemployed
Retired

What is your highest completed education?
Primary School
Secondary School
University
Graduate – Masters
Graduate – PhD

Rate the following set of statements according to how much you agree or disagree with them ranging from 1, meaning you strongly disagree, up to 5, meaning you strongly agree.

1. I consider myself an advanced user of animation software such as Lightwave 3D, Cinema 4D, Maya, 3D Max, etc.
strongly disagree 1 2 3 4 5 strongly agree

2. I consider myself an expert Flash user
strongly disagree 1 2 3 4 5 strongly agree

3. I know what a Keyframe is
strongly disagree 1 2 3 4 5 strongly agree

4. I know what frame-by-frame animation means
strongly disagree 1 2 3 4 5 strongly agree

5. I know what tweening means
strongly disagree 1 2 3 4 5 strongly agree

Please indicate what software applications you use at least once a week. Next to each application rank your level of competence using a scale from 1 to 5, 1 meaning you consider yourself to be a novice, up to 5, meaning you consider yourself an expert.

Appendix B. Modified MSLQ (Based on Pintrich & De Groot, 1990)

Please rank the following set of statements according to how much you feel they apply to you.

1. When learning software applications I prefer course work that is challenging so I can learn new things.	Not at all true of me 1 2 3 4 5 Very true of me
2. I expect to complete this course without difficulty.	Not at all true of me 1 2 3 4 5 Very true of me
3. I am so nervous when I am learning how to use a new software application that I cannot remember facts I have learned before.	Not at all true of me 1 2 3 4 5 Very true of me
4. In general, I ask myself questions to make sure I know the material I have been learning.	Not at all true of me 1 2 3 4 5 Very true of me
5. It is important for me to learn what is being taught in this course.	Not at all true of me 1 2 3 4 5 Very true of me
6. I like what I will be learning in this course.	Not at all true of me 1 2 3 4 5 Very true of me
7. I am certain I can master the skills taught in this course.	Not at all true of me 1 2 3 4 5 Very true of me
8. When work is hard I either give up or focus only on the easy parts.	Not at all true of me 1 2 3 4 5 Very true of me
9. I think I will be able to use what I learn in this course.	Not at all true of me 1 2 3 4 5 Very true of me
10. I expect to do very well in this course.	Not at all true of me 1 2 3 4 5 Very true of me
11. I am a proficient computer user.	Not at all true of me 1 2 3 4 5 Very true of me
12. I often choose software courses or tutorials I will learn something from, even if they require more work.	Not at all true of me 1 2 3 4 5 Very true of me
13. I am sure I can do a great job on the problems and tasks that are part of this course.	Not at all true of me 1 2 3 4 5 Very true of me
14. I have an uneasy, upset feeling when I am learning how to use a new software application.	Not at all true of me 1 2 3 4 5 Very true of me
15. I work on practice exercises when learning a new software application even when I don't have to.	Not at all true of me 1 2 3 4 5 Very true of me
16. Even when instructional material is dull and uninteresting, I keep working until I finish.	Not at all true of me 1 2 3 4 5 Very true of me
17. Even when I struggle to learn a software application, I try to learn from my mistakes.	Not at all true of me 1 2 3 4 5 Very true of me
18. I think that what I will learn in this course is useful for me to know.	Not at all true of me 1 2 3 4 5 Very true of me
19. Before I learn to use a new software application I think about the things I will need to do to learn.	Not at all true of me 1 2 3 4 5 Very true of me
20. My computer skills are excellent compared to other people.	Not at all true of me 1 2 3 4 5 Very true of me
21. I think that what we will be learning in this course is interesting.	Not at all true of me 1 2 3 4 5 Very true of me
22., I think I know a great deal about the subject taught in this course.	Not at all true of me 1 2 3 4 5 Very true of me
23. I know that I will be able to learn the material in this course.	Not at all true of me 1 2 3 4 5 Very true of me
24. I worry a great deal about learning any new software application.	Not at all true of me 1 2 3 4 5 Very true of me
25. Understanding what will be presented in this course is important to me.	Not at all true of me 1 2 3 4 5 Very true of me
26. When I am following software tutorials or manuals I stop once in a while to go over what I have learned.	Not at all true of me 1 2 3 4 5 Very true of me
27. When I am learning how to use a new software application I think about how poorly I am using it.	Not at all true of me 1 2 3 4 5 Very true of me
28. I want to understand what is taught in this course because it is important to show my ability to my family, friends, employer, or others.	Not at all true of me 1 2 3 4 5 Very true of me
29. I prefer course material that really challenges me so I can learn new things.	Not at all true of me 1 2 3 4 5 Very true of me
30. When learning a new software application I prefer to do it on my own, without help from anyone.	Not at all true of me 1 2 3 4 5 Very true of me

Appendix C. Computer User Self-Efficacy Scale

Computer User Self-Efficacy Scale, CUSE - Adapted from Cassidy & Eachus, 2002

Below you will find a number of statements concerning how you might feel about computers. Please indicate the strength of your agreement/disagreement with the statements using the 5-point scale shown below. Tick the number (i.e., between 1 and 5) that most closely represents how much you agree or disagree with the statement. There are no *correct* responses, it is your own views that are important.

1. Most difficulties I encounter when using computers, I can usually deal with.
   strongly disagree 1 2 3 4 5 strongly agree

2. I find working with computers very easy.
   strongly disagree 1 2 3 4 5 strongly agree

3. I am very unsure of my abilities to use computers.
   strongly disagree 1 2 3 4 5 strongly agree

4. I seem to have difficulties with most of the software applications I have tried to use
   strongly disagree 1 2 3 4 5 strongly agree

5. Computers frighten me.
   strongly disagree 1 2 3 4 5 strongly agree

6. I enjoy working with computers.
   strongly disagree 1 2 3 4 5 strongly agree

7. I find that computers get in the way of learning.
   strongly disagree 1 2 3 4 5 strongly agree

8. In general, I don't have trouble using computer applications.
   strongly disagree 1 2 3 4 5 strongly agree

9. Computers make me much more productive.
   strongly disagree 1 2 3 4 5 strongly agree

10. I often have difficulties when trying to learn how to use a new computer application.
strongly disagree 1 2 3 4 5 strongly agree

11. Most of the computer applications I have had experience with, have been easy to use.
strongly disagree 1 2 3 4 5 strongly agree

12. I am very confident in my abilities to make use of computers.
strongly disagree 1 2 3 4 5 strongly agree

13. I find it difficult to get computers to do what I want them to do.
strongly disagree 1 2 3 4 5 strongly agree

14. At times I find working with computers very confusing.
strongly disagree 1 2 3 4 5 strongly agree

15. I would rather that we did not have to learn how to use computers.
strongly disagree 1 2 3 4 5 strongly agree

16. I usually find it easy to learn a new computer software application.
strongly disagree 1 2 3 4 5 strongly agree

17. I seem to waste a lot of time struggling with computers.
strongly disagree 1 2 3 4 5 strongly agree

18. Using computers makes learning more interesting.
strongly disagree 1 2 3 4 5 strongly agree

19. I always seem to have problems when trying to use computers.
strongly disagree 1 2 3 4 5 strongly agree

20. Some computer packages definitely make learning easier.
strongly disagree 1 2 3 4 5 strongly agree

21. Computer jargon baffles me.
strongly disagree 1 2 3 4 5 strongly agree

22. Computers are far too complicated for me.
    strongly disagree 1 2 3 4 5 strongly agree

23. Using computers is something I rarely enjoy.
    strongly disagree 1 2 3 4 5 strongly agree

24. Computers are good aids to learning.
    strongly disagree 1 2 3 4 5 strongly agree

25. Sometimes, when using a computer, things seem to happen and I don't know why.
    strongly disagree 1 2 3 4 5 strongly agree

26. As far as computers go, I don't consider myself to be very competent.
    strongly disagree 1 2 3 4 5 strongly agree

27. Computers help me save a lot of time.
    strongly disagree 1 2 3 4 5 strongly agree

28. I find working with computers very frustrating.
    strongly disagree 1 2 3 4 5 strongly agree

29. I consider myself to be a skilled computer user.
    strongly disagree 1 2 3 4 5 strongly agree

30. When using computers I worry that I might press the wrong button and damage it.
    strongly disagree 1 2 3 4 5 strongly agree

Thank you

Appendix D. Player Type (Van Meurs, 2007)

The following questions are about computer games, how frequently you use them, and the things you value when you play them.

Do you play computer games? Yes No

If yes:

1. How many hours a week do you play computer games? ___
2. For how many years have you been playing computer games? ___
3. What kind of computer games do you like to play? (Select all that apply)
   - Role-Playing
   - Strategy
   - First Person Shooter
   - Adventure
   - Sports
   - Fighting
   - Racing/Driving
   - Flight simulators
   - Other (please specify)

The next part asks you about your playing style on the computer games that you play. Please indicate if you disagree or agree with the following statements, considering your usual playing style in mind.

1. I consider advancing in the game as the most important goal of playing	Strongly disagree 1 2 3 4 5 Strongly agree
2. I like to be a part of the storyline of the game	Strongly disagree 1 2 3 4 5 Strongly agree
3. I like to explore all the areas in the game	Strongly disagree 1 2 3 4 5 Strongly agree
4. I like to group up with other players	Strongly disagree 1 2 3 4 5 Strongly agree
5. I like to annoy other players	Strongly disagree 1 2 3 4 5 Strongly agree
6. I like to role-play in the game	Strongly disagree 1 2 3 4 5 Strongly agree
7. I want to be noted for my achievements	Strongly disagree 1 2 3 4 5 Strongly agree
8. I like getting to know my fellow players	Strongly disagree 1 2 3 4 5 Strongly agree
9. It is unimportant to fit my character to the story	Strongly disagree 1 2 3 4 5 Strongly agree
10. I want to be known for my knowledge of the game	Strongly disagree 1 2 3 4 5 Strongly agree
11. I like to know where to find things	Strongly disagree 1 2 3 4 5 Strongly agree
12. I want to score high on rating lists	Strongly disagree 1 2 3 4 5 Strongly agree
13. I consider causing distress to other players to be rewarding	Strongly disagree 1 2 3 4 5 Strongly agree
14. I often have meaningful conversations with fellow players	Strongly disagree 1 2 3 4 5 Strongly agree
15. I hate making people angry	Strongly disagree 1 2 3 4 5 Strongly agree
16. I like to be immersed in the world	Strongly disagree 1 2 3 4 5 Strongly agree
17. I consider the storyline of the game unimportant	Strongly disagree 1 2 3 4 5 Strongly agree

18. I try to find out all there is about game mechanics	Strongly disagree 1 2 3 4 5 Strongly agree
19. I like to compete with other players	Strongly disagree 1 2 3 4 5 Strongly agree
20. I always try to find bugs in the gameplay	Strongly disagree 1 2 3 4 5 Strongly agree
21. I think that realism in the game is important	Strongly disagree 1 2 3 4 5 Strongly agree
22. I often find myself disclosing personal information to other players	Strongly disagree 1 2 3 4 5 Strongly agree
23. Rare and powerful items that have special benefits are unimportant to me	Strongly disagree 1 2 3 4 5 Strongly agree
24. I never play out-of-character	Strongly disagree 1 2 3 4 5 Strongly agree
25. I like to show everyone my knowledge of the game	Strongly disagree 1 2 3 4 5 Strongly agree
26. I like to impose myself upon others	Strongly disagree 1 2 3 4 5 Strongly agree
27. Becoming powerful is very important to me	Strongly disagree 1 2 3 4 5 Strongly agree
28. I like to try out new roles in the game	Strongly disagree 1 2 3 4 5 Strongly agree
29. I like maps, charts and tables with information about the game	Strongly disagree 1 2 3 4 5 Strongly agree
30. I hardly use the communciation channels available in the game	Strongly disagree 1 2 3 4 5 Strongly agree
31. I always try to find as many secrets as possible	Strongly disagree 1 2 3 4 5 Strongly agree
32. I like dominating and killing other players	Strongly disagree 1 2 3 4 5 Strongly agree
33. I want to accumulate as many valuable objects as possible	Strongly disagree 1 2 3 4 5 Strongly agree
34. I always tend to help other players	Strongly disagree 1 2 3 4 5 Strongly agree
35. I like to make up storylines for my character(s)	Strongly disagree 1 2 3 4 5 Strongly agree

Appendix E. Personality Traits (Saucier, 1994)

The list below describes common human traits. Use this list to describe yourself as accurately as possible. Please describe yourself as you see yourself in the present time, not as you wish to be in the future. Describe yourself as you are generally or typically, as compared with other persons you know of the same sex and roughly the same age. Please indicate how accurate or inaccurate each trait is using a 1 to 5 scale, 1 being very inaccurate and 5 being very accurate.

Bashful	Very inaccurate 1 2 3 4 5 Very accurate
Bold	Very inaccurate 1 2 3 4 5 Very accurate
Careless	Very inaccurate 1 2 3 4 5 Very accurate
Cold	Very inaccurate 1 2 3 4 5 Very accurate
Complex	Very inaccurate 1 2 3 4 5 Very accurate
Cooperative	Very inaccurate 1 2 3 4 5 Very accurate
Creative	Very inaccurate 1 2 3 4 5 Very accurate
Deep	Very inaccurate 1 2 3 4 5 Very accurate
Disorganized	Very inaccurate 1 2 3 4 5 Very accurate
Efficient	Very inaccurate 1 2 3 4 5 Very accurate
Energetic	Very inaccurate 1 2 3 4 5 Very accurate
Envious	Very inaccurate 1 2 3 4 5 Very accurate
Extraverted	Very inaccurate 1 2 3 4 5 Very accurate
Fretful	Very inaccurate 1 2 3 4 5 Very accurate
Harsh	Very inaccurate 1 2 3 4 5 Very accurate
Imaginative	Very inaccurate 1 2 3 4 5 Very accurate
Inefficient	Very inaccurate 1 2 3 4 5 Very accurate
Intellectual	Very inaccurate 1 2 3 4 5 Very accurate
Jealous	Very inaccurate 1 2 3 4 5 Very accurate
Kind	Very inaccurate 1 2 3 4 5 Very accurate
Moody	Very inaccurate 1 2 3 4 5 Very accurate
Organized	Very inaccurate 1 2 3 4 5 Very accurate
Philosophical	Very inaccurate 1 2 3 4 5 Very accurate
Practical	Very inaccurate 1 2 3 4 5 Very accurate
Quiet	Very inaccurate 1 2 3 4 5 Very accurate
Relaxed	Very inaccurate 1 2 3 4 5 Very accurate
Rude	Very inaccurate 1 2 3 4 5 Very accurate
Sly	Very inaccurate 1 2 3 4 5 Very accurate
Sloppy	Very inaccurate 1 2 3 4 5 Very accurate
Symphatetic	Very inaccurate 1 2 3 4 5 Very accurate
Systematic	Very inaccurate 1 2 3 4 5 Very accurate
Talkative	Very inaccurate 1 2 3 4 5 Very accurate
Temperamental	Very inaccurate 1 2 3 4 5 Very accurate
Touchy	Very inaccurate 1 2 3 4 5 Very accurate

Uncreative	Very inaccurate	1 2 3 4 5	Very accurate
Unenvious	Very inaccurate	1 2 3 4 5	Very accurate
Unintellectual	Very inaccurate	1 2 3 4 5	Very accurate
Unsymphatetic	Very inaccurate	1 2 3 4 5	Very accurate
Warm	Very inaccurate	1 2 3 4 5	Very accurate
Withdrawn	Very inaccurate	1 2 3 4 5	Very accurate

Appendix F: Satisfaction Survey (Adapted from Lewis (J. R. Lewis, 1995))

This questionnaire gives you an opportunity to express your satisfaction with the usability of the software practice training system. Your responses will help us understand what aspects of the system you are particularly concerned about and the aspects that satisfy you.

To as great a degree as possible, think about all the exercises that you have completed with the system while you answer these questions. **Please consider only the software you have just used to complete the exercises. The questions below refer to this software as the 'practice environment'.**

Please read each statement and indicate how strongly you agree or disagree with the statement by selecting a number on the scale.

1. Overall, I am satisfied with how easy it is to use the practice environment.	Strongly disagree 1 2 3 4 5 Strongly agree
2. It is simple to use the practice environment.	Strongly disagree 1 2 3 4 5 Strongly agree
3. I can effectively complete Flash exercises in this practice environment.	Strongly disagree 1 2 3 4 5 Strongly agree
4. I am able to complete all exercises quickly in this practice environment.	Strongly disagree 1 2 3 4 5 Strongly agree
5. I feel comfortable using this practice environment.	Strongly disagree 1 2 3 4 5 Strongly agree
6. It was easy to learn how to use this practice environment.	Strongly disagree 1 2 3 4 5 Strongly agree
7. I believe I quickly learned to do the exercises in this practice environment.	Strongly disagree 1 2 3 4 5 Strongly agree
8. This practice environment gives me enough feedback to complete the exercises.	Strongly disagree 1 2 3 4 5 Strongly agree
9. Whenever I make a mistake using the practice environment, I recover easily and quickly.	Strongly disagree 1 2 3 4 5 Strongly agree
10. It is easy to find the information I need to complete the exercises in the practice environment.	Strongly disagree 1 2 3 4 5 Strongly agree
11. The instructions provided in the practice environment are easy to understand.	Strongly disagree 1 2 3 4 5 Strongly agree
12. The information provided is effective in helping me complete the exercises.	Strongly disagree 1 2 3 4 5 Strongly agree
13. The organization of information on the screen is clear.	Strongly disagree 1 2 3 4 5 Strongly agree
14. The interface (graphics, panels, colors, and labels) of the practice environment is pleasant.	Strongly disagree 1 2 3 4 5 Strongly agree
15. I like using the interface of the practice environment.	Strongly disagree 1 2 3 4 5 Strongly agree
16. The practice environment has all the functions and capabilities I expect it to have.	Strongly disagree 1 2 3 4 5 Strongly agree
17. Overall, I am satisfied with the practice environment.	Strongly disagree 1 2 3 4 5 Strongly agree

APPENDIX G. CONSENT TO PARTICIPATE IN DISTRIBUTED INFORMATION RESOURCES AND EMBODIED COGNITION IN SOFTWARE APPLICATION TRAINING: INTERACTION PATTERNS IN ONLINE ENVIRONMENTS

This is to state that I agree to participate in a program of research being conducted by Juan Carlos Sanchez-Lozano of the Department of Education of Concordia University, who can be reached at 514-499-9603, or by e-mail at jua_sanc@education.concordia.ca.

A. PURPOSE

I have been informed that the purpose of the research is as follows:

To inform the design and creation of instructional interactive environments, specifically in the area of software training. This study focuses on practice environments for the automation of fundamental operations in software applications that are the building blocks of more complex procedures, and that the user needs in order to complete authentic tasks in a professional environment.

B. PROCEDURE

You have been previously assigned a code which was sent to you via e-mail when you signed up to participate in this study. Once you consent to participate, the program will ask you to enter this code. You will be required to complete a short survey about your motivational beliefs and learning strategies, how comfortable you feel using computers, some of your personality traits, and your preferred playing style in case you play computer games. A short online course in basic Flash animation concepts will follow, after which you will have the opportunity to practice what you have learned. The study concludes with a short assessment and a satisfaction survey. The entire process will take approximately 3 hours. A screen will tell you once you complete the study.

Important: If you close your browser window you will lose all your data and you will not be able to restart. Please complete the study in one single session.

C. RISKS AND BENEFITS

By participating in this study you will have the opportunity to learn basic animation concepts in Flash. If you complete the study, you will also be able to sign up for a four hour instructor-based course in Flash. You will also contribute to understand how people learn to use software applications which will help improve computer instruction.

Except for the normal discomfort you can experience while working with a computer, your participation does not entail any risks.

D. CONDITIONS OF PARTICIPATION

- I understand that I am free to withdraw my consent and discontinue my participation at anytime without negative consequences, except that I will not be able to sign up for the free instructor-based Flash course.

- I understand that my participation in this study is:

  CONFIDENTIAL (i.e., the researcher will know, but will not disclose my identity)

- I understand that the data from this study may be published.

E. RESULTS

If you wish to be informed when the final consolidated results of this research become available, please type your e-mail: ______________________________

BY CLICKING ON THE 'I ACCEPT' BUTTON, I DECLARE THAT I HAVE CAREFULLY STUDIED THE ABOVE AND UNDERSTAND THIS AGREEMENT. I FREELY CONSENT AND VOLUNTARILY AGREE TO PARTICIPATE IN THIS STUDY.

If at any time you have questions about your rights as a research participant, please contact Adela Reid, Research Ethics and Compliance Officer, Concordia University, at (514) 848-2424 x7481 or by email at areid@alcor.concordia.ca.

[I ACCEPT]   [I DO NOT ACCEPT]

Appendix H. Call for Participation

Would you like to start learning Flash? From simple sites to web applications, Flash is the software of choice to create advanced animations and deliver rich web content.

A current research project at Concordia University's Department of Education is looking for participants for a study on online training environments. Participants will be asked to complete a few questionnaires and take a short online course in Flash. The entire session will take approximately 3 hours of your time. Only basic computer knowledge is required, i.e. you should be comfortable navigating in an online environment, use a mouse to interact with the computer, and select different menu items.

Participants that finish the study will be able to sign up for a free 4-hour instructor-based course in Flash. The instructor has 7 years of experience in teaching the software. The online course will cover basic animation concepts that will be extended further during the instructor-based session.

Please contact Juan Carlos Sanchez-Lozano at jua_sanc@education.concordia.ca for more information.