

2008

Cognition and Technology: Effectiveness of intelligent tutoring systems for software training

Claver P. Hategekimana
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>

 Part of the [Business Commons](#)

Recommended Citation

Hategekimana, Claver P., "Cognition and Technology: Effectiveness of intelligent tutoring systems for software training" (2008).
Graduate Theses and Dissertations. 11409.
<https://lib.dr.iastate.edu/etd/11409>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

**Cognition and Technology:
Effectiveness of intelligent tutoring systems for software training**

by

Claver Pierre Hategekimana

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Human Computer Interaction

Program of Study Committee:
Anthony Townsend, Co-major Professor
Stephen Gilbert, Co-major Professor
Ana-Paula Correia
Brian Mennecke
James Oliver
Kevin Scheibe

Iowa State University

Ames, Iowa

2008

Copyright © Claver Pierre Hategekimana, 2008. All rights reserved.

TABLE OF CONTENTS

LIST OF FIGURES	v
LIST OF TABLES	vi
ACKNOWLEDGEMENTS	vii
ABSTRACT	viii
CHAPTER 1. INTRODUCTION	1
1.1 Context of the proposed ITS study.....	2
1.2 Overview of training approaches	4
1.3 Indicators of training approach success	6
1.4 Learner's ability to succeed indicators.....	7
1.5 Statement of the Problem	8
1.6 Purpose of study	9
1.7 Methodology	10
CHAPTER 2. REVIEW OF LITERATURE	12
2.1 Foundation of self-directed instruction	12
2.1.1 Book-based instruction	15
2.1.2 Computer-based instruction	16
2.1.2.1 Software simulation-based instruction	17
2.1.2.2 Intelligent tutoring systems	18
2.2 Intelligent tutoring system architecture.....	20
2.3 Characteristics of training approaches	24
2.4 Training outcomes and assessment measures	27
2.4.1 Training success indicators	27
2.4.2 Learner's ability to succeed indicators.....	29

2.5 Summary and the way forward	32
CHAPTER 3. METHODS AND PROCEDURES	34
3.1 Participants	34
3.2 Design of training curriculum	35
3.3 Materials.....	36
3.4 Measures	39
3.5 Setting and procedure.....	43
3.6 Data analysis	48
CHAPTER 4. RESULTS AND DISCUSSION.....	51
4.1 Handling missing data.....	51
4.2 Impact of training mode on training outcomes	53
4.3 Impact of learner's ability to succeed indicators and training mode on learning outcomes	54
4.3.1 Effect of spatial ability and training mode on training outcomes	55
4.3.2 Effect of general self-efficacy and training mode on training outcomes	57
4.3.3 Effect of computer self-efficacy and training mode on training outcomes	59
4.3.4 Effect of state self-efficacy and training mode on training outcomes.....	61
4.4 Results of qualitative measures.....	65
CHAPTER 5. CONCLUSION.....	69
5.1 Summary	69
5.2 Lesson learned.....	71
5.3 Limitations and direction for future research	72
REFERENCES	74

APPENDIX 1. Opening script: Effectiveness of Paint.NET training program	83
APPENDIX 2. Informed consent document	84
APPENDIX 3. Spatial ability survey: Paper folding	87
APPENDIX 4. Demographic questionnaire	91
APPENDIX 5. General self-efficacy survey	92
APPENDIX 6. Computer self-efficacy survey	99
APPENDIX 7. State self-efficacy: Anticipated confidence in the use of Paint.NET	94
APPENDIX 8. Paint.NET training system usability survey	95
APPENDIX 9. Paint.NET concept acquisition test	97
APPENDIX 10. Paint.NET task performance test.....	99
APPENDIX 11. Paint.NET training manual	101

LIST OF FIGURES

Figure 1. Snapshot view of ClearSighted's system to offer training through linking target application and intelligent tutor	3
Figure 2. Cognitive Tutor Algebra I and student interaction.....	23
Figure 3. Paint.NET interface enhanced with ITS-tutor	45
Figure 4. Paint.NET interface enhanced with ITS-tutor	46
Figure 5. Paint.NET interface enhanced with ITS-tutor	47
Figure 6. Research Model – Factors that impact training success.....	50
Figure 7. Effect of spatial ability (SA) on task performance.....	57
Figure 8. Effect of general self-efficacy (GSE) on system usability perception	59
Figure 9. Effect of computer self-efficacy (CSE) on task performance score.....	61
Figure 10. Effect of state self-efficacy (SSE) on task performance score	63

LIST OF TABLES

Table 1. Benefits of computer based training (example: software simulation and intelligent tutoring system) versus traditional instructor led training	14
Table 2. Comparison of strengths and limitations of book, simulation and intelligent tutoring system.....	24
Table 3. Dynamic training module design: worksheet example.....	36
Table 4. Example of training module in Paint.NET training.....	38
Table 5. Sequence of training and testing activities with estimated times	48
Table 6. Comparison of participant scores on measures of training mode success.....	53
Table 7. Comparison of participant training outcomes based on training mode and Learner's spatial ability	55
Table 8. Comparison of participant training outcomes based on training mode and Learner's general self-efficacy	58
Table 9. Comparison of participant training outcomes based on training mode and Learner's computer self-efficacy	60
Table 10. Comparison of participant training outcomes based on training mode and Learner's Paint.NET state self-efficacy.....	62
Table 11. Parametric and non-parametric test results are compared	64
Table 12. Comparison of parametric and non-parametric test results	65
BIOGRAPHICAL SKETCH	122

ACKNOWLEDGEMENTS

The work presented in this dissertation holds for more than the culmination of years of study. It also reflects the relationships with many generous and inspiring people I have met and worked with since the beginning of my doctoral work at Iowa State University. I express the deepest appreciation to my advisory committee members for giving me freedom to pursue independent work. Without their guidance and persistent support this dissertation would not have been possible. Two co-major professors, Anthony Townsend and Stephen Gilbert, demonstrated rigorous scholarship and have been instrumental in ensuring my academic success and financial support. Their work and encouragements demonstrated to me that modern technology should always transcend academia and provide a quest for our times. To Brian Mennecke, Keven Scheibe, James Oliver, and Ana Paula Correia for their thoughtfully criticism, encouraging words, and attention during busy semesters.

Too many people to mention individually have assisted in many ways and have my sincere gratitude. In particular, I thank John Mayfield and Joseph Munyaneza for their encouragement to pursue graduate studies. To Pam Shill and Lynette Sherer in the HCI program for assisting me with the necessary administrative tasks, and to Bridget Egan for helping with editing this document.

In addition to my mother Julienne Mukabiseruka, brothers, and sisters, my final and most heartfelt acknowledgement must go to my wife Claudine Musabyemariya and son Kevin S. Hategekimana. Their encouragement and companionship have turned my journey through doctoral studies into a pleasure. For all that, and for being everything I am not, they have my everlasting love.

ABSTRACT

This study addresses the potential of using an intelligent tutoring system (ITS) to tutor on off-the-shelf (OTS) software. ITSs have been successfully used to tutor on a variety of learning domains, but there has been little research comparing ITS-based training on an OTS application with traditional software training approaches such as books or interactive software simulations. The work presented here includes procedures and results for Paint.NET training and evaluation using three methods: book-based, interactive simulation, and an ITS. It is reported that there were some associations between the training method and training experiences. Book-based training exhibited higher scores on both task performance and system usability perception, while better times were recorded for the simulation approach. Concept acquisition score was not found to significantly correlate with training method, however. Additionally, it was found that interactions between training mode and spatial ability or general self-efficacy (GSE) significantly affected system usability perception. It was also learned that within ITS high computer self-efficacy (CSE) learners outperformed these with low CSE on task performance measure. Similar findings were reported for simulation group where high-spatial learners recorded better training times than low-spatial learners. Overall, results indicated that four individual characteristics to succeed indicators explored in this study significantly correlated with total training time and system usability measures. It is concluded that if an ITS is to be a tutor on OTS application then further refinements are needed.

CHAPTER 1. INTRODUCTION

Knowledge transfer and cost effective training programs are among the most critical issues faced by colleges and the field of human resources development today (see Kirkpatrick, 1998; Seels & Glasgow, 1998; Baldwin, 1992). Educational institutions are not only interested in the use of technology to effectively train personnel, but they are also exploring new ways they can provide students with self-directed instruction for software training that meets educational goals. At the same time, corporations have realized a great need for human resources development to cope with the gap between the skills of their personnel and the increasing technological demands of jobs. Perhaps the biggest challenge within this issue is to find the most effective training programs. Following the work of Carr (2004) and McCain (2002), the use of computer applications for training is a stepping stone to competitive advantage when used effectively.

This study seeks to evaluate the effectiveness of a newly developed intelligent tutoring system (ITS) application that was designed by ClearSighted, Inc. to tutor on a COTS (commercial off-the-shelf) or third-party software application (see Gilbert, et. al., 2007). ITS is a broad term used to describe any computer program that contains some intelligence and is useful for learning. ITS is an outgrowth of the earlier computer-based instruction (Freedman et al., 2000; Beck, et al., 2004). The ITS of ClearSighted Inc. guides learners through the process of learning how to use an image manipulation software called Paint.NET. Examples of combining tutors and existing software applications include a tool-tutor for geometric construction using Geometer's Sketchpad (Jackiw & Finser, 1993) and a tool-tutor for algebra problem solving using Microsoft Excel (Ritter & Blessing, 1998). The two research

questions are: 1) how does the ITS-based software training system compare with traditional software training methods, such as book-based training and simulation-based training? As noted by Corbett and Anderson (1997), one of common errors in technology assessment is to underestimate its potential consequences and effects on learning. And 2) how do learner's ability to succeed indicators affect the relationships between training modality and training outcomes?

1.1 Context of the proposed ITS study

This study is a small component of a larger project at ClearSighted Inc., a consultancy focusing on education and technology services in Ames, Iowa. One of the company core competencies is to leverage the strengths of ITSs to provide a cost effective training program for off-the-shelf-software. In 2006, ClearSighted received a grant from National Science Foundation to undertake a project titled: *SBIR/STTR Phase II: Developing a Cost-Effective Method for Creating Models for Cognitive Tutors*. The project had two clear objectives: The first objective was to develop an application that would simplify cognitive model creation. This would allow people from different areas of expertise to easily create ITS educational modules. The second objective was to link an ITS to an off-the-shelf software with little or no modification to the existing software interface (see Figure 1). Achieving these objectives would result in lower cost and less time when creating ITS modules, and possibly provide higher learning experiences than currently available software training methods. My contribution to this project relates to the second objective.

As an intern graduate student at ClearSighted for approximately 18 months, I was involved in the design process of the training modules for three training approaches: book-

based instruction, software simulation, and ITS. My other contribution was to design the pilot study, full experiment, and evaluation instruments. I am thankful for the flexibility of ClearSighted that allowed me to explore the effect that the interaction between training mode and learner's ability to succeed indicators (spatial ability and self-efficacy) has on training outcomes. This was in my academic interest more than it benefited ClearSighted.

The primary interest of ClearSighted was to prove that their ITS had potential to yield a higher return on investment (shorter training time and higher training outcomes) than currently available training methods, and to identify where ITS improvement opportunities can be realized. Book and simulation served as the baseline for evaluating the effectiveness of the intelligent tutoring system. Book-based instruction was chosen because it is a widely used self-directed method for learning computer applications. Similarly, interactive simulations built with Adobe Captivate 2 were chosen because they are a cutting edge technology that simplifies the task of creating training demos.

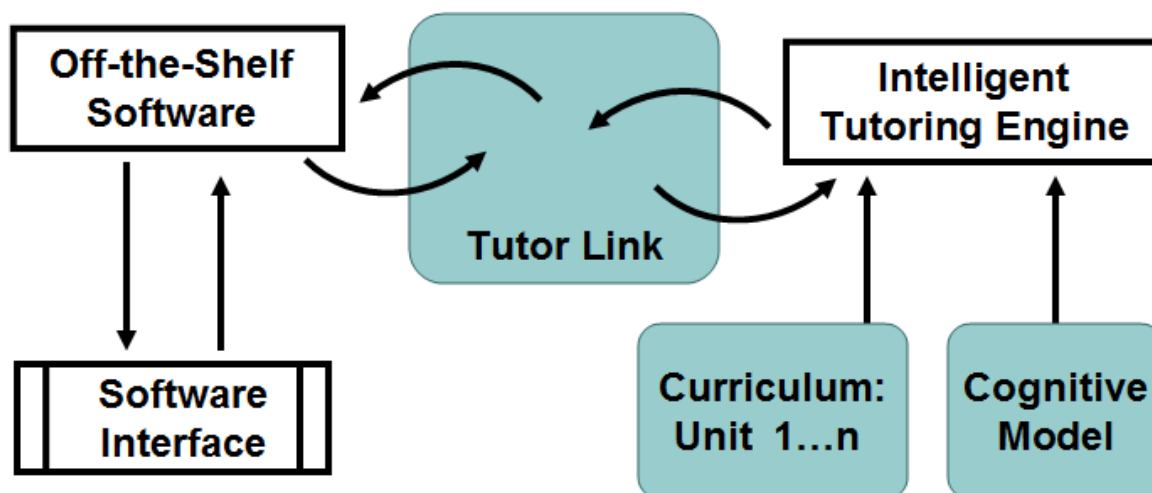


Figure 1: Snapshot view of ClearSighted's system to offer training through linking target application and intelligent tutor.

1.2 Overview of training approaches

Many forms of training materials exist, but for the purpose of this study the focus was on three approaches, which include printed book-based instruction (training manuals) and two forms of computer-based training (software simulations and intelligent tutoring systems). These three training modes use self-directed and learn-by-doing approaches, but they differ in how they present the steps for reaching the goal.

Book-based instruction: or a printed training manual is a document designed to teach users how to do something, such as using software. Training manuals are widely used in businesses, schools, and homes. According to Shneiderman (1998), training manuals can be effective and convenient if prepared properly. The benefits of well-designed manuals include shorter learning times, better user performance, increased user satisfaction and fewer calls for support. The book-based instruction is the oldest technology used to support self-directed instruction. For example, as early as 1873, Anna Ticknor created a volunteer program that provided paper-based correspondence instruction for the purpose of educational opportunities for women (Watkins, 1991). With the evolution of educational technology, self-directed instruction evolved through developments in computer-based instruction (CBI).

Computer-based training (Simulation and ITS): In this study, the terms computer-based-based training (CBT) and computer-based instruction (CBI) are used interchangeably. Rational evidence shows that CBT can be a viable alternative to a training manual. For example, in 2006, CBT counted for 31% of total corporate training, while printed material only counted for about 4% (Rivera & Paradise, 2006). Even though researchers argue that CBT is highly desirable, the ability of learners to transfer learning back to the job is frequently questioned (Baldwin, 1992). Marshall (1998), who saw CBT as a viable training medium made the same point, but the problem is how to use this delivery system to the best advantage.

a) Simulation: Over time, CBT advancement introduced computer simulations (one of the CBT approaches used in this study) as a way of exposing users to real-life experience. A computer simulation is a representation of a real-life system, which allows learners to experience some of the elements of the real world. With simulation, processes are represented in an abstracted or scaled down form. A good simulation can present facts and rules of a situation in a highly realistic manner and adjust these factors to respond to interaction by the learner (see Forcier & Descy, 2005; Lunce, 2004; Marshall, 1988). However, Lunce (2004) criticized that learners gain little from discovery when studying from computer simulations. Furthermore, simulations tend to oversimplify the complexity of real-life giving the learner a false understanding of a real life problem.

b) Intelligent tutoring system: By the 1980s, researchers studying CBT began exploring intelligent tutoring systems (ITSs) to offer personalized, adaptive feedback to learners. ITS is the second type of CBT used in this study. Sleeman and Brown (1982) coined the term intelligent tutoring system to describe and distinguish this approach from the previous CBT. An intelligent tutoring system is differentiated from traditional CBT program by its artificial intelligence component.

The theoretical basis for the current computer-based intelligent tutors is the cognition theory of Anderson (1983, 1995) called “adaptive control of thought: ACT*,” more recently renamed ACT-R (Anderson & Schunn, 2000). According to ACT-R, acquisition of cognitive skills involves the formulation of rules that relate to goals, tasks, and consequences. This means that human problem solving is enabled by a set of production rules manifested in “if, then” processes. Because printed training materials are inherently static, it is not feasible to integrate production rules within book-based training modules. On the other hand, software simulations rely on production rules to some extent.

The strengths of an intelligent tutoring system come from its ability to track a learner’s work and provide personalized feedback and appropriate guidance on a real-world

task. An ITS has an expert model component that knows all the information needed to solve problems within the domain of the tutoring material. The role of the expert model component is to check the responses of the learner to ascertain whether they are part of an acceptable path to the learner's goal. The process of checking student actions using expert model is known as "model tracing."

Intelligent tutoring systems have been used successfully to tutor in a variety of educational learning domains including, but not limited to, mathematics, geometry, and economics (e.g: Alevan et al., 2006; Bransford et al., 2000; Koedinger, 2004; Arruarte et al., 1997). Despite notable success stories, the ITSs are not widely used in education and training programs. There are two main reasons for this slow penetration. First, an ITS is expensive to develop; second, an ITS is generally evaluated according to its artificial intelligence criteria rather than with respect to educational effectiveness (Corbett & Anderson, 1997).

1.3 Indicators of training approach success

In this study, four indicators of training success, knowledge, skills, reaction, and total training time (TTT), were used to evaluate the effectiveness of training approaches (Harrison, 2006; Kirkpatrick, 1998). Knowledge (concept acquisition) can be defined as a broad range of what someone knows (ideas, information) about a particular subject. On the other hand, a skill (task performance) can be defined as the ability to do something well (Gheorghe, 2006; French Ministry of Education, 2006). Task performance measures demonstrate the ability of the learner to apply new knowledge or skills on the job. Reaction relates to learners' satisfaction as they interact with the system (perception of system usability, usefulness of the content, and whether they enjoyed it or not). Shorter TTT is an indication of an efficient

training approach. Reduced time spent on tasks may be categorized as the most accurate measureable result that has direct effect on productivity and cost.

The training approach that exhibits higher results in the areas listed above is desirable to potential adopters of the system. These adopters can make a decision based on how each method meets the needs of their organization. For example, two approaches may achieve comparable results on task performance, but the results on concept acquisition and TTT on tasks may be significantly different. In a scenario like this, an organization would make a decision based on how they value the concept acquisition versus TTT (with assumption that other evaluation parameters have the same score for both systems).

1.4 Learner's ability to succeed indicators

There are two factors considered in the present study that may predict training success: self-efficacy and spatial ability.

Self-efficacy: Researchers like McQuiggan, et al. (2008) and Bandura (1986, 1997) hold that self-efficacy is a person's beliefs about his/her judgment or ability to perform well in a given situation. This means that the concern is not on an individual's skills, but the judgments of what a person can do with whatever skills she/he possesses (Mulkey & O'Noel, 1999; Bandura 1989). In the ITS context, self-efficacy can be described as a judgment of one's capability to understand, use, and benefit from the ITS-based training. Three types of self-efficacy were considered: 1) general self-efficacy (GSE) which captures one's confidence to do well in different domains, 2) computer self-efficacy (CSE) which is one's confidence in using a computer, and 3) state self-efficacy (SSE) or task specific self-efficacy,

which is one's confidence in completing a specific task in a particular domain (Downey & McMurtrey, 2006).

Spatial ability: Learner's spatial ability is another potential predictor of training success. Some research interpretations indicate that high-spatial learners improve better through self-guided practice and feedback than low-spatial learners (Kyllonen et al., 1984). The implication here is that students may learn best with different training strategies, depending on their spatial ability. This study explored whether the combination of training method and learner's ability to succeed indicators (LASI) significantly impacted training outcomes. The hypothesis was that higher LASI levels would be correlated with training success across training approaches.

1.5 Statement of the Problem

Globally, many training manuals and computer-based training applications are developed each year. Both companies and universities have also integrated CBT into curricula to facilitate training on software applications. Current approaches for software training are problematic due to the passivity of reading and video, simulations that are maladapted to learner's preferences, and the lack real world relevance to the task the trainee will do with the software. For example, with printed instruction, the trainee does not find out what he/she is doing wrong until a self-evaluation is performed. Frequently, self-evaluation is not synchronized with the performance of a wrong action. Similarly, simulations may not keep pace with learners' speed. Simulations may not provide the option to speed up or slow down the transition time between slides and typically do not offer flexibility to refer back to previously presented material.

However, a model-tracing intelligent tutoring system addresses these issues by relying on artificial intelligence (production rules: if... then...) and cognitive models embodied in Anderson's (1983) ACT-R theory. The ITS described here is a tutoring system that aims to be more effective than traditional training approaches. It aims to offer a timely and appropriate guidance to the trainee through different levels of help (just-in-time and on-demand hints). In addition, learners interact with the actual software on which they are being trained, not a simulation or some other facsimile. Despite these apparent benefits, there has been little research comparing ITS-based training on an off-the-shelf application with traditional software training approaches. Similarly, there is very little research on how interaction between training mode and LASIs affect indicators of training approach success. The proposed research would study how effectively an ITS meets specific training goals when compared with two more traditional approaches to training.

1.6 Purpose of study

The purpose of this study is twofold: the first objective is to determine whether the Paint.NET tutor is a viable alternative to book and software simulation tutoring for off-the-shelf software. The second objective is to determine whether interaction between the training mode and learner's ability to succeed indicators (LASI) correlate to training outcomes. The LASI parameters include participants' spatial ability and self-efficacy (general self-efficacy, computer self-efficacy, and state self-efficacy). Specifically, the research design of this empirical study is structured by the following research questions:

1. *Is there a relationship between:*
 - a. Task performance and the training condition?

- b. Concept acquisition and the training condition?
 - c. Total training time and the training condition?
 - d. System usability perception and the training condition?
2. *Is there a correlation between:*
- a. Task performance and a combination of training condition and self-efficacy?
 - b. Concept acquisition and a combination of training condition and self-efficacy?
 - c. Total time on task and combination of training condition and self-efficacy?
 - d. System usability perception and combination of training condition and self-efficacy?
 - e. Task performance and a combination of training condition and spatial ability?
 - f. Concept acquisition and a combination of training condition and spatial ability?
 - g. Total time on task and combination of training condition and spatial ability?
 - h. System usability and combination of training condition and spatial ability?

1.7 Methodology

This empirical study describes the Paint.NET training sessions, assessment, and findings. The training sessions and assessments were administered at Iowa State University during the fall semester of 2007. Participants were undergraduate students without previous Paint.NET and Adobe Photoshop experience (a software with similar features to Paint.NET). Also, the study excluded subjects from computer science and graphic arts majors. Participants were randomly assigned to different training modes (book, simulation, or ITS-based training). While this study attempted to evaluate the impact that training mode and LASI have on the training success, there are limitations that are worth noting:

1. For convenience, the participants in the study were undergraduate college students. This population is not representative of corporate personnel, who are potential users of the ITS for software training.
2. Almost all participants were in the 18-24 age range, a young audience who was acquainted with technology at younger age (digital natives). Older adult learners (digital immigrant) in the work place may react differently to the Paint.NET training. “Prensky (2005) has coined the term ‘digital natives’ to refer to the generation who has grown up with digital technologies so that they are a part of their world view. Conversely, these generations who have not grown up in these technology-rich worlds are ‘digital immigrants’ – they can learn the new technologies but considerably more effort is required than for the digital natives.” See Prensky (2001) and Zevenbergen (2007).

CHAPTER 2. REVIEW OF LITERATURE

Educators, researchers in the area of human computer interaction, corporations, self-employed business people, trainers, and psychologists strive to find training programs that meet learners' preferred training environment and to maximize learning experiences (see Bartley & Golek, 2004). These professionals explore self-directed instruction (SDI) as a path to achieving effective and efficient training programs. In this literature review, self-directed instruction is described from both the training mode and learner's ability to succeed indicators prospective. The attempt is to answer two general questions: "Which training method is better?" And "What learner characteristics facilitate success within a particular modality?" The first part of this literature provides an overview of the foundation of SDI. Then, a description of a typical ITS architecture is provided. Next, success of self-directed training programs and assessment measures are explored. The chapter ends with an overview of expected training outcomes of this study.

2.1. Foundation of self-directed instruction

Self-directed instruction (SDI) is defined as a process by which learners have the ability to activate and sustain cognitions and behaviors, which are systematically oriented toward attainment of learning goals and control of learning strategies and process while involved with learning tasks (Song, 2007 and Abdullah, 2001). With SDI, learning activities are designed to encourage students to actively participate in their own learning process. SDI activities have many benefits: They are less expensive to support and offer opportunities in situations where traditional training is difficult to operate. Often institutions see self-directed instruction as a cost saving model or an additional source of revenues, as they can reach a

greater number of learners at a minimal cost. As indicated by McCain (2007), large and small companies are already taking advantage of technology to provide training opportunities to employees anytime and anywhere. The success of these programs depends on the learner's ability to learn new skills of the workforce, technological adaptability, increased productivity, and cost and efficiency. SDI applications include, but are not limited to, print media (example: training manual) and computer-based training (example: software simulations and intelligent tutoring systems). As indicated in the *Handbook of Technology-Based Training* by Tucker (1997), computer-based training (CBT) has proven potential to achieve the same or better training outcomes than traditional face-to-face instruction. See benefits of computer-based training presented in Table 1. The next section provides a general description of two popular SDI approaches: book or printed material and CBT. The reasons mentioned above merit an investigation to whether a newly developed Paint.NET tutor is comparable to or better than traditional training methods. It is also important to know more about factors that lead to and sustain learners to become successful, which can help to provide a framework for acquiring desired training success.

Table 1: Benefits of computer-based training (example: software simulation and intelligent tutoring systems) versus traditional instructor led training.

Benefits type	Observation
<i>People learn more through CBT than through traditional training programs.</i>	Traditionally, the trainer and learner do not find out how much learning took place until post-training test, which is often too late for meeting expected level of knowledge. With CBT, the learner is constantly tested to ensure that acceptable knowledge level has been reached. If not, the learner is directed back to the topic where their knowledge is not up to the required level.
<i>People learn faster using CBT than using traditional training programs.</i>	Trainer-centered approach is tailored to the group, rather than the individual, which has potential to slow down leaning process. With CBT, learners only study in depth those parts of the course where they do not have prior knowledge or where an improvement is needed, and move rapidly through these modules where they have some understanding (work on their own pace). This has proved to reduce training time to by as much as a half.
<i>Retention of knowledge is better through CBT than through traditional training programs.</i>	Studies in knowledge retention proved that learners learn and retain knowledge longer when using CBT than through instructor led training. Learners were immediately given post-training test and again after 3 months. Results were compared with favorable outcomes for TBT training condition.
<i>Flexibility: CBT makes training possible where traditional training is impractical.</i>	Traditional training programs, whether run in-house or externally, are inflexible. They can only be run on a predetermined schedule and the number of attendees may be set to a certain number. Program changes or cancellations have unnecessary opportunity costs. In contrast, CBT is very flexible. Once learning material are purchased, they are available all the time.
<i>CBT can be used as a reference material when a problem arises.</i>	CBT is generally available within the work place, and it only takes a few moments to search back to the appropriate part of the course and find the answer. This is not possible with instructor led training.
<i>CBT is often more cost effective than traditional training.</i>	There is a number of areas where savings can be made by using CBT rather than traditional training programs. CBT reduces training time, it takes place in the most convenient location and time which eliminates travel time, trainer's compensation, etc.
<i>Offering just-in-time training.</i>	With CBT, training more often takes place when the learner is about to carry out a task. That is when motivation is highest and when most knowledge from the learning translates into the work.
<i>People learn at their own pace.</i>	With traditional training programs, the pace does not meet every learner's need in the group. The pace is either too slow or too fast, making learning difficult, which has a detriment effect on learning. Even on one-on-one tutoring, the learner may not be prepared to ask the tutor to slow down or speed up. In contrast, with CBT the learner is in total control of his/her learning pace.
<i>Learner makes mistakes in private.</i>	Unlike traditional training environment where learner has fear of looking foolish by giving a wrong answer or asking a question, CBT doesn't pass judgment--giving the trainee more confidence. This is one of the benefits that is highly rated by trainees.

This list is a summary of CBT benefits described in the *Handbook of Technology-Based Training* of Tucker , B. (1997, p. 23-32.)

2.1.1. Book-based instruction

The terms “book-based instruction” or “printed instruction” or “training manuals” are used interchangeably in this study. A training manual is a document designed to teach the user how to do something, such as how to use software. They are widely used in business, schools, and homes. Training manuals are the oldest technology used to support SDI. For example, in 1873, Anna Ticknor created the educational program to encourage at home studies for women. Her program provided correspondence instruction (printed self-directed instruction) to 10,000 members over a 24-year period (Watkins, 1991). By 1883, the state of New York officially recognized collegiate instruction by correspondence and degrees offered by Chautauqua College of Liberal Arts (Moore, 2003). SDI that uses training manuals has benefits: people learn at their own pace and the book can be used as a reference when a problem arises. Training manual users, however, suffer from lack of help at the moment of need, which breaks the learning momentum. The trainer and learner do not find out how much learning took place until post-test, which is often too late for meeting expected level of knowledge (Tucker, 1997). Additionally, when the learner faces a problem, he/she must wait for help from a trainer, who, depending on the instructor-to-student ratio, may take awhile to respond (Ong & Ramachandran, 2000). Currently, SDI has evolved to include computer-based training (CBT). The CBT is also referred as computer-based instruction (CBI), computer-assisted instruction (CAI), computer-based learning (CBL), and technology-based training (TBT).

2.1.2 Computer-based instruction

Using CBI or CBT to further educational goals is not a new concept. The history of CBT can be linked to the first educational machine called ‘Teaching Machine’ that was described by Sydney Pressey in 1926. His machine offered a sequence of multiple-choice questions, and students selected answers by pressing a button. The machine indicated whether the given answer was correct or not, and was sometimes referred to as computer-aided instruction (Marshall, 1988). Some of the early CBT applications are described in the next paragraph.

The actual first use of computers for teaching and learning can be traced back to 1960 when International Business Machine (IBM) introduced Coursewriter I (Suppes & Macken, 1978). This application was an authoring tool intended to help people lacking programming skills to develop computer-based teaching materials. In the same year (1960), Donald L. Bitzer, who had a dream for teaching with computers, originated PLATO (Programmed Logic for Automated Teaching Operation) at the University of Illinois (VanMeer, 2003). The PLATO system favored drill-and-practice methodology by incorporating course material into larger interrelated conceptual packages. That is, if a student found the material easy or familiar, he/she could “leapfrog” through a course in a minimum of lessons. At the same time, students who needed more time and explanation found themselves directed back and forth through the total sequence of lessons until the concept was mastered (VanMeer, 2003). By the 1970s, the use of CBI had reached public schools. In 1971, the TICCIT (Time-shared, Interactive, Computer Controlled Information Television) project was initiated at Brigham Young University. The system combined television technology with the computer to deliver instruction to learners (Levy, 1977). In 1975, Computer Curriculum Corporation (CCC)

developed a CBT system that involved a computer that provided individualized lessons to as many as 96 teletype terminals simultaneously. Terminals located at different school sites were linked to the main computer through telephone lines (Suppes & Mackens, 1978). Throughout 1975, CBT continually advanced and introduced interactive computer simulations as a way of exposing users to real-life experience. Similarly, in the 1980s, the representation of a learner's knowledge through an intelligent tutoring system (ITS) was seen as opportunity for CBT improvement. In addition to the training manual, computer simulations and ITS are of interest to this study.

2.1.2.1 Software simulation-based instruction

An interactive computer simulation is a program that allows instant feedback and the possibility of an ongoing review of the two-way communication process (Casey, 1999). Simulations offer many benefits, including allowing learners to experiment with problems that would be too dangerous or expensive to explore in reality (Luce, 2004). Some researchers believe that simulation may be best suited to teaching problem-solving or decision-making skills (Lunce, 2004). Adobe Captivate 2 is one of current innovative applications that make the development of software simulation possible for learning purposes, including software training. Simulations have also proven to attain higher learning goals than traditional methods (Hooper & Thomas, 1991).

As early as the 1960's, researchers had moved beyond the systems that presented instruction in a pre-determined pattern based on the history of responses of the learner. That is, researchers in the area of computer-aided instruction (CAI) employed technology to achieve educational goals, including automated testing and routine drill-and-practice tasks

(Corbett & Anderson, 1997). These researchers had to anticipate different types of responses and decide which information the system would then present to the learner in a drill-and-practice fashion. CAI has been successful in education and training markets because it is very effective in improving skills and factual recall. Early CAI researchers were the first to model behavior, but they have been criticized for not modeling learner's knowledge states (Corbett & Anderson, 1997).

2.1.2.2 Intelligent tutoring systems

In 1982, Sleeman and Brown (1982) envisioned an advanced CAI system they called intelligent tutoring system (ITS). Their system's emphasis was learning by doing and representing a learner's knowledge. Sleeman and Brown are credited for coining the term "intelligent tutoring system" and they were the first to use the term "student model," used to describe an abstract representation of the learner within the computer program. ITSs can be understood as a type of CBT technology called cognitive tutor. Presently, developing an ITS structure is a daunting task because its work relies heavily on artificial intelligence (AI) and research from outside the computer science discipline.

In 1982, Anderson (1983 and 1995) developed a theory of cognition called "adoptive control of thought: ACT*" that has been the basis for current computer-based intelligent tutors. The underlying theory of ACT* states that all knowledge begins as declarative information and procedural knowledge that is learned by making references to already existing factual knowledge (Anderson & Schunn, 2000). The theory holds that cognitive skill acquisition involves the formulation of rules that relate to goals, task states, actions, and consequences. That is, human problem solving is enabled by a set of production rules 'If...

Then...’ The ACT* theory was successfully implemented and tested in many computer applications including the LISP Tutor, Geometry Tutor (Koedinger & Anderson, 1993), and the Algebra Tutor developed for the Pittsburgh School District (Corbett & Anderson, 1997).

In the 1990s, Pittsburgh Advanced Cognitive Tutor (PACT) center created Cognitive Tutor Algebra. Currently, this intelligent tutoring system for teaching algebra is believed to be the most successful ITS application for use in K-12 classrooms (see Carnegie Learning, inc., 2007 and Koedinger et al., 1997). By 1998-1999, the tutor was implemented in 75 schools, a number that increased to 1,400 schools by the year 2003. Despite notable success stories, ITSs are not widely used in education and training. There are two main reasons for this slow penetration:

First, ITSs are expensive to develop. To produce one hour of an ITS-based instructional material will take a team of experts about 300-1,000 hours of work effort (Ainsworth & Fleming, 2006). Creating ITS educational modules is a task that is still beyond an average computer user. The team needed to create an ITS includes a software developer, domain experts, a teaching expert, and usability engineers. So far, a number of authoring tools have been developed to address this issue. See 13 systems described in the book of Murray and colleagues (2003).

Second, ITSs are generally evaluated according to the artificial intelligence criteria rather than with respect to educational effectiveness (Corbett & Anderson, 1997). Workplace professionals and educators worry that ITSs may not reflect their own pedagogic approaches. A solution is to include teachers in the design and evaluation of ITS application. In this way, ITS training modules will be developed based on a sound plan that insure learning goals, learning objectives, and assessment measures meet desired training outcomes (e.g: Harrison,

2006). The reasons mentioned above merit an investigation to whether a newly developed Paint.NET tutor is comparable to or better than traditional training methods. It is also important to know more about factors that lead to and sustain learners to become successful, which can help to provide a framework for acquiring desired training success.

2.2. Intelligent tutoring system architecture

Many traditional instructor led lessons present learners with information and concepts followed by assessment questions. These methods are effective in exposing learners to information and testing their recall. These approaches, however, stimulate a learner's ability to recall but may not effectively apply their knowledge and skills in a real-world operational setting. On the other hand, the ITS system uses a highly interactive learning environment that requires learners to apply their knowledge and skills the same way they would in an operational setting. (see Ong & Ramachandran, 2000). Similarly, individual features of training manual, simulations, and ITS lead to different characteristics in the training experience. All training methods use a learning-by-doing approach which has potential to excel because they represent experience as opposed to information communicated as facts (Hays, 2005). However, these approaches primarily differ in how they present the steps for reaching the goal. For example, unlike the ITS, book-based instruction does not support multiple paths to the solution and does not provide help at the point of need (see strengths and limitations of training approaches in Table 2).

Like any typical ITS architecture, the Paint.NET tutor described has four main components: domain knowledge (expert model), pedagogical model (instructor model), learner's model, and interface model.

Domain knowledge:

The domain knowledge module (also called expert model) presents subject matter expertise and provides the ITS with knowledge of what it is teaching as well as problem solving ability. The domain knowledge enables the ITS to compare the action of the learner against the ITS software reasoning (embedded rules in the subject matter: if...then...). Basically, course developers create templates that specify an allowable sequence of correct actions and what to do in case an incorrect action is performed. Domain knowledge is a repository of production rules that is accessible to other components of the tutor. This component is the most important because without it there would be nothing to teach.

Pedagogical model:

The pedagogical model encodes instructional strategies (curricula) that are appropriate for the target domain and the learner. This module selects both the most appropriate representation of a learning task and the most effective instructional intervention. Besides presenting learning tasks, the pedagogical model generates three types of feedback: on-path action (correct answer), off-path action, and on-demand help.

The on-path action feedback is experienced when the learner performs a correct action. From the learner's perspective, a correct action is recognized if the tutor does not comment, but rather the tutor allows the learner to progress with the solution. The off-path action is experienced when the learner received a just-in-time (JIT), a hint-like message indicating that a mistake was made. A JIT message may suggest what the learner needs to do to get back on the right path. A JIT message may also simply indicate that a mistake was made, but the tutor lets the learner figure out by him/herself how to correct the mistake. On-

demand help can be requested at any time during the problem solving process. The tutor presents different levels of requested help, which not only provide guidance leading to the solution, but also offer enrichment with conceptual explanation. The level of on-demand help increases with each request and generates a more specific message until an acceptable answer is reached.

Learner model:

The learner model component monitors and evaluates a learner's sequence of action at any given moment throughout the problem solving process. The feedback described in a pedagogical model is carried out as a response to the learner's actions, giving personalized feedback. The purpose of the learner model is to provide data for the pedagogical module. Some complex tutors apply pattern-matching rules to detect sequences of actions that indicate how well a learner is performing on the material being taught.

Interface model:

The interface model is a graphic representation that enables the learners to interact with the content they are learning and to receive feedback from the tutor. For example, while participants in the ITS-based Paint.NET interact with a typical Paint.NET interface, they also interact with an ITS window that presents the concept ideas and feedback. The interface deals with how to present the material in the most effective way (see Figure 2, screenshot of Cognitive Tutor Algebra I and student interaction). As mentioned by Jacko and Sears (2003, p. 928), "...users interpret what they see in an interface and draw their own conclusions about how it works (and) that may be different from the designers' intentions—and then they act on their own conclusions not on the designers' attentions." That is, designers must solicit

input from potential users to ensure they develop an acceptable, successful interface. For example, one of the techniques used to ensure the quality of a system's interface is to test a prototype of the system by using think-aloud protocol (having user talk out loud as they work). The assumption is that a well-designed interface has the potential to increase learning, while a negatively perceived designed system would break learning momentum.

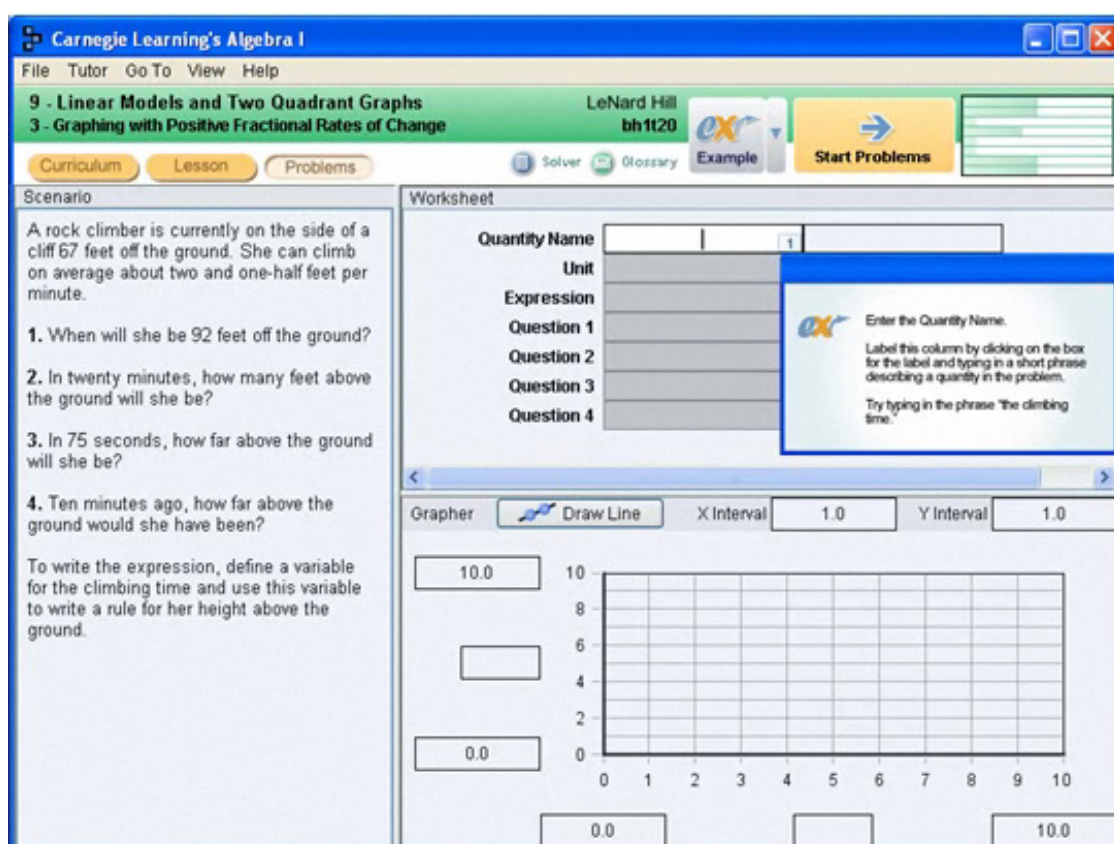


Figure 2: Cognitive Tutor Algebra I of Carnegie Learning, Inc.. This screenshot shows the screen as it looks to a student in the middle of Algebra I interactive example. If a student enters incorrect input, he/she will be notified accordingly and have a chance to correct the error. After a few incorrect entries, the cognitive tutor will complete the field correctly for him/her and move to the next step.

2.3 Characteristics of training approaches

The distinction between three training modes described in this study is more of a continuum than strict distinctions. The modes have similarities in how they present instruction, but the natural affordances of each medium have potential characteristics leading to differences in the training experience.

Table 2: Comparison of strengths and limitations of book, simulation, and intelligent tutoring system. This data is based on system architecture, observation, and comments from a December 2006 pilot study at a mid-western university in the United States (Hategekimana, Gilbert & Blessing, 2008).

Opportunity for strength or weakness	Book	Simulation	ITS
Learn-by-doing (hands-on experience)	High	Medium ¹	High
Self paced instruction (adjust to learner's speed)	High	Medium ²	High
Detailed step-by-step instruction on how to complete the task	High	None	None
Demonstration by using example (vicarious learning)	None	High ³	None
Timely feedback (ex. just-in-time help, on-demand help)	None	High ⁴	High
Multiple paths to the solution	None	None	High
Revisit previously viewed material	High	Medium	Low

1. Learner interacts with screen shots of Paint.NET rather than the actual Paint.NET software.
2. Learner can not speed up or slow down the transition time between slides.
3. Learner watches a demo that introduces new skills needed to complete the training tasks.
4. People solve problems in different ways. A learner who solves a problem one way gets a different hint than a student who solves in a different way (Hafner, 2004).

Learn by doing approach:

This hands-on experience has enormous potential to increase learning and understanding of content knowledge. The majority of the hands-on experience for software training is the actual practice tasks the learners have to do using the target application they are trying to learn. This is the exact learning experience both the book and model tracing ITS approaches provide. However, simulation approach mimics these learning experiences rather than interacting with the actual software the learner is being taught. This approach somewhat

misses a complete touch and feel experience proved by the other two training methods.

Self-paced instruction:

All three training approaches described in this study allow learners to work on their own pace. Book and ITS give the learner total control of the pace, while a typical simulation developed using a demo application like Adobe Captivate or Camtasia have some built-in timing functions that have potential to prevent learners from achieving the maximum pace control. The transition between slides (how long a message is displayed on the screen for example) can be too fast or too slow, which can break the learning momentum. Book users decide when to flip to the next page and the ITS users have total control on when to open or close each window message for example.

Step-by-step instruction:

Printed training manuals provide step-by-step instructions on how to solve a problem. This key practice allows learners to achieve incremental steps in knowledge, making the attainment of skills more achievable and less stressful. With this learning environment, however, learners are often completely unaware of their mistakes while completing a task. In contrast, learners in simulation and ITS conditions have to use judgment and guesses to arrive at the solution (exploratory learning). At the same time, ITS provides timely feedback to discover misconceptions and provides guidance leading to the right path to the solution.

Demonstration by using example:

A typical simulation training module can easily include a demo on how to complete a task that is similar to the exercise the learner is required to perform (vicarious learning). This is not possible with self-directed instruction provided by book or ITS.

Timely feedback:

A good computer-based training program has a similar role to a human tutor-teacher. In this role, computers present lectures, give feedback and remedial explanation to the learners as they go through the program offered. Little is known about the mechanism of

effective feedback-hinting strategy in computer based training, but there is research evidence showing that feedback at point of need positively correlates with learner's success (see Matsuda & VanLehn, 2003; Corbett & Anderson, 2001, and Bransford, et al. 2000). It is technically feasible for the ITS and simulation to accommodate immediate feedback or just-in-time (JIT) help, and feedback on demand. Similarly, the learner is allowed to proceed to the next step throughout the problem solving only if a correct solution to the current problem state is provided. In contrast, learners in the book-based mode receive no feedback on how to complete the training tasks. Printed training manuals are, by nature, static and give no opportunity for learners to ask questions. Furthermore, a learner does not find out how well he/she is doing until a self-evaluation is performed or a test score is received. This is an important contrast with the other two training conditions (simulation and ITS) in which learners are required to achieve a correct solution.

Multiple paths to the solution:

Usually there is an ideal solution to any given problem, but different learners solve problems differently, which implies that multiple paths to the solution are worth exploring. With book training, a step-by-step instruction to the solution is provide. However, it is unlikely that if a learner misses a step, the problem solving process will actually reach a desired solution. Similarly, simulation requires learners to linearly solve their problems without a chance to explore their own path. In contrast, the ITS allows learners to explore different ways of solving a problem (exploratory learning). That is, a learner receives feedback-help not only for the ideal path the designer-developer has envisioned, but for other paths as well.

Revisit previously reviewed material:

It is typically easier to flip back and forth among pages in a book, for example, than to refer back to previously viewed video footage. Users in the ITS can only restart and repeat the entire task as they do not have flexibility to revisit a previously performed steps.

2.4 Training outcomes and assessment measures

As mentioned earlier, self-directed learning continues to be a mainstream in many organizations, but the effectiveness of a new self-directed-based system should be comparable or better to these of traditional training programs (Bartley & Golek, 2004). The focus must include the evaluation of whether or not the new system is meeting the goals specified in the planning stage. Without a comprehensive evaluation, it would be difficult to gain the program acceptance of educators and corporate managers because they are concerned with net benefits (costs versus benefits -- opportunity cost) or the answer to the question “Which training approach is better?” Also, in order to help determine the future of self-directed training programs, this study focuses on both training modality and comparing learner characteristics that facilitate favorable outcomes within a particular modality. The emphasis is not only on the training approach, but also the way to increase tutor sensitivity to the individual learner characteristics by preparing them to facilitate training. The next two sections describe training success and learner’s ability to succeed indicators.

2.4.1 Training success indicators

a) Conceptual knowledge: Indicates theoretical knowledge gained as a result of training, including understanding of facts presented in the training material (Harrison, 2006). For example, teaching the theoretical aspects of the software the learner is being trained helps to understand how the system works from a high level. This way, learners can adapt easily to changes in technology because they understand the fundamentals. This is an important aspect because advancements in technology seem to dictate the technology of the workplace of tomorrow. Furthermore, learners who have deep acquisition of conceptual knowledge have potential to utilize their knowledge to stimulate new discoveries (Bransford, et al., 2000).

b) Task performance: Indicates the practical application of conceptual knowledge applied in the real world environment to accomplish tasks similar to these presented in training (Harrison, 2006). That is, task performance is the efficiency with which learners can solve problems on their own outside the training environment. A good training program produces participants who understand not only how the system works from a high level, but it also produces participants who have good grasp of how it actually works and who are capable of using it efficiently to meet their job demands (learning transfer). Bransford, et al. (2000) emphasize that one of common principles of learning transfer is about learning and then assessing the learner's ability to apply it to another situation. One of specific focuses of this empirical study is to whether a training condition makes a difference in learner's ability to transfer what they have learned to everyday settings of home, community, and workplace.

c) Total training time: An efficiency measure that indicates the amount of time used for completion of curriculum. An efficient training system results in shorter learning time with more material covered. Training time required by an employer is always considered work time, implying that shorter training time reduces training costs and produces more time for other projects (McMacain, 2007). The time the learner spends on each of the training tasks can be recorded, and total training times for individual learners must be computed and compared to see which system speeds up the training process.

d) System usability: The term usability can be equated to the five-dimensional definition of learnability, efficiency, memorability, errors, and satisfaction (Nielson 1993). Usability is a major factor in whether learners will be able to make use of a new technology. It addresses questions such as how quickly and easily a system can be learned and whether the system meets the user's learning goals (see Jacko & Sears, 2003; Brooke, 1996).

According to Six Sigma, a statistical-based quality improvement methodology, the most commonly used usability metrics are task completion, time on task, error counts, and satisfaction scores (Sauro & Kinklund, 2005).

Given that the focus of this study is primarily to compare the effectiveness of three training approaches, the attention here is to measure how each system is relative to the other two products. The assumption is the design and inherent characteristics of training systems observed in this study will differently impact the overall satisfaction of participants, how much learning takes place, and how much time it takes to complete training tasks (training outcomes). This conjecture creates the following hypothesis:

Hypothesis 1: *There is a significant relationship between training condition and training outcomes.*

2.4.2 Learner's ability to succeed indicators

For many years, comparative studies have been devoted to finding the answer to the basic research question: “*Is instruction delivery mode A as good as, or better than, traditional instruction delivery mode B?*” (Where A=ITS, and B=Book or Simulation). At the present time, the adult learning theory paradigm has shifted to include the constructivist learning perspective that the learner plays a role constructing new knowledge by relating new information to prior knowledge and experience. In this view, the tutor (delivery mode) is regarded as a guide rather than a dispenser of knowledge (Abdal-Haqq, 1998). That is, the mode of delivery is only important to the extent that it facilitates communication. An additional emphasis must be on facilitating learning through matching learning activities to student learning preferences (ex: learning style, self-esteem, self-efficacy, spatial ability etc.).

This study will observe how two indicators of abilities to succeed, self-efficacy and spatial ability, affect the relationship between training approach and training outcomes.

a) Self-efficacy: Self-efficacy precepts suggest that learners may perceive new educational technology, ITS included, as either an opportunity or a threat. Mulkey and O’Noel (1999) held that learners with higher levels of computer self-efficacy perform better than those with lower levels of self-efficacy in computer training situations (see Baldwin, 1990; Gist et al., 1989). In this study, three types of self-efficacy will be explored.

- **General self-efficacy (GSE):** It captures the perceptions of one’s ability to transfer his/her confidence among domains. GSE forms over time from a range of activities, it changes slowly, and it persists in different domains (Downey & McMurtrey, 2006). See an example of GSE scale of Bosscher and Smit (1998), who modified the original self-efficacy scale of Sherer et al. (1992).
- **Computer self-efficacy (CSE):** It is described as individual judgment of one’s capability to use a computer (Downey & McMurtrey, 2006). Higher computer self-efficacy offers organizations a proven way to boost technological productivity and efficiency, enhance the role of technology assets, and gain competitive advantage (Carr, 2000). See example of computer self-efficacy scale of Compeau and Higgins’ (1995) questionnaire that asked participants about their ability to use an unfamiliar piece of software.
- **State self-efficacy (SSE) or task-specific self-efficacy:** SSE is believed to be an effective predictor of performance (Yi & Im, 2004; Awang-Hashim, et al. 2002; Blair, et al. 1999). At the completion of training, participants are asked to complete a

self-efficacy survey that asks them how they feel about completing a future task similar to the ones they were trained on.

b) Spatial ability: Some research interpretations indicate that high-spatial learners improve more through self-guided practice and feedback than low-spatial learners do (Kyllonen, Lohman, & Snow, 1984). Some spatial visualization tests have shown to be reliable predictors of achievement in technical training such as aviation, architecture, and engineering courses. Outside of those fields, however, spatial ability tests have shown mixed results (Rowe, 1991). Paint.NET training tasks involve altering visual images (example: changing orientation of an image, colors, size, and layers), which requires some spatial visualization skills. The implication here is that trainees may learn best with different training strategies, depending on their spatial ability. This triggers interest to find correlations between a combination of spatial ability, training conditions, and training outcomes. An example of spatial ability scale using paper-folding test is presented in Appendix 3, see Ekstrom et al. (1976). It will be interesting to find out whether there is any significant relationship between interaction of training mode and LASI, and training outcomes.

Based on this literature review, it is assumed that the relationship between the two focal variables, training mode and training outcomes, is stronger when self-efficacy and spatial ability levels are high. This assumption creates the following two hypotheses:

Hypothesis 2: *There is a significant relationship between learner's ability to succeed indicators and training outcomes.*

Hypothesis 3: *Learner's ability to succeed indicators significantly moderates the relationship between training condition and training outcomes*

2.5 Summary and the way forward

The above review has described some of the trends in the use of training manuals and computer-based training. There are some popular self-directed training opportunities that have not emerged in this account that should be noted in concluding this review. One of these trends is the use of computer games and the proliferation of web-based training manuals. The other trend is how to set up the optimum teaching and counseling arrangements in an environment marked by a highly self-directed instruction. Due to time limitations and cost, this study chose to use simulation and training manual as a baseline to test ITS effectiveness.

This review highlighted potential benefits of using ITS to tutor in different learning domains; the main lesson learned is that ITS has proven potential to achieve the same or better training outcomes than traditional instructor-led courses. However, there is very little research 1) comparing the effectiveness of ITS and other self-directed training modes, and 2) describing the effectiveness of ITS for tutoring on off-the-shelf software. Similarly, different measures of self-efficacy and spatial ability have proven to be successful predictors of learner's success. However, there is very little data describing the effect of the interaction between training mode and LASI has on training outcomes. The next chapter describes an empirical study with a two-fold goal: 1) comparing the effectiveness of book, simulation and ITS, and 2) indentifying relationships between LASI and training outcomes based on training mode.

Findings of this research will make a three-pronged contribution to the ITS studies and the field of self-directed learning with suggestions from formative and summative evaluation and new knowledge. Since the ITS for software training is new, formative evaluations will provide feedback to improve its design, teaching, and learning experiences.

Summative evaluation measures will indicate what participants have learned at the end of training activities, which will lead to sound judgment about adoption, continuation or expansion. Similarly, the primary purpose of this research is to add to the body of knowledge in the field by contributing to the growth of theories associated with self-directed learning approaches. This can facilitate learning through matching learning activities to student learning preferences.

CHAPTER 3. METHODS AND PROCEDURES

The experiment involved three training conditions: 1) book-based, 2) interactive simulation, and 3) model-tracing ITS. The book-based instruction was chosen because it is a widely-used method for learning computer applications. Similarly, interactive simulations build with Adobe Captivate were chosen not only because they are a cutting edge technology that simplifies the task of creating computer-based training demos (interactive simulations), but also because simulations permit the attainment of learning goals that are beyond traditional methods (Hooper & Thomas, 1991). As noted in the literature review, this experiment setup is driven by the goal to test three research hypotheses stated in null form:

Hypothesis 1: *There is no statistically significant relationship between training condition and training outcomes.*

Hypothesis 2: *There is no statistically significant relationship between learner's ability to succeed indicators and training outcomes.*

Hypothesis 3: *There is not statistical significance that learner's ability to succeed indicators moderates the relationship between training condition and training outcomes.*

What follows in this chapter describes how participants were recruited, what training materials were used, the types of survey and testing instruments, the setting and procedures, and the data analysis procedure.

3.1 Participants

Participants in the experiment were 75 undergraduate students at a mid-western university in the United States. An invitation to participate was emailed to students at the same university. Flyers were posted at different locations on campus, and an announcement

was posted on the web site facebook.com. Prospective participants were invited to register online (there were a designated web page for registration). Thirty-five dollars and a chance to win one of three iPod Nanos were used as incentives for participation. To ensure fairness, a computer application randomly picked a winning number among 4-digit student IDs provided. The study required a commitment to attend both a 90-minute training session and a 60-minute testing session. Participants were divided into three groups (book, simulation and ITS condition).

3.2 Design of training curriculum

Training modules were designed based on course design guidelines of Harrison (2006) with emphasis on training outcomes, learning goals, learning objectives, and assessment measures (see course design worksheet in Table 3 and training manual in Appendix 11) outlined below:

- Expected training outcomes are set in terms of knowledge (what we want learners to know), skills (what we want learners to be able to do), and attitude (changed behavior) as a result of received training.
- Learning goals are broad and general as they are described in terms of the overall intended training outcomes.
- Learning objectives are specific and concrete; they are specific things the learner was able to do because of learning.
- Assessment measures are metrics of how much learning and satisfaction took place (quantitative and/or qualitative data) in relation to specified learning objectives.

Table 3: Dynamic training module design: worksheet example

Step 0: Results of training	Knowledge: What learners should know as a result of training	Skills: What learners should be able to do as a result of training	Attitudes: Changed behavior as a result of training
Step 1: Training outcomes	<ul style="list-style-type: none"> Understand basic elements of size and orientation of the image. 	<ul style="list-style-type: none"> Learn how to change the size and orientation of the image. 	<ul style="list-style-type: none"> Value and accept image with varying orientations and formats.
Step 2: Performance measures	Learner will be able to: <ul style="list-style-type: none"> Identify and describe the effect of orientation and size of the image. 	Learner will be able to: <ul style="list-style-type: none"> Resize the image Rotate the image Flip the image 	Learner will be able to: <ul style="list-style-type: none"> Justify (compare and contrast) the orientation and size parameters of the image based on its intended use.
Step 3: Means of assessment	Demonstrate theoretical knowledge through: <ul style="list-style-type: none"> Conceptual test questions on the image size and orientation parameters. 	Demonstrate practical Skills through: <ul style="list-style-type: none"> Task performance test on the image orientation and size manipulation. 	Demonstrate fluency and efficiency through: <ul style="list-style-type: none"> Measure of total training time spent on task (shorter time being desirable)

Because the research question focused on the effectiveness of three training approaches, identical content was used in all three training media as much as possible. At the same time, the individual features of each medium led to different characteristics in the training experience. It is typically easier to flip back and forth among pages in a book, for example, than to refer back to previously viewed video footage.

3.3 Materials

Materials for the study included web-based software simulation, ITS, and printed texts on how to use basic features of Paint.NET. All three training approaches used the same training curriculum that included three units. In Unit One, participants learned how to edit the image (resize, rotate, flip and adjust colors of the image). In Unit Two, they learned how to

enhance the image by selecting, cropping, and adding text to the image. In Unit Three, they learned how to build the image by working with layers. Each unit had several tasks to complete, and each of the tasks had three main components. Two of these components, *What's the Point?* and *Concept*, were identical regardless of the training mode. The first component was the *What's the Point?* section that presented a real world scenario of what needed to be done. The second component was the *Concept* section that provided learners with an explanation of important elements of the lesson to be learned. The third component was the *Steps*. Step-by-step instructions were presented differently based on the training mode. Participants in the book group followed step-by-step printed instructions, while participants in the simulation and ITS based training received guidance from the system they interacted with (See training manual, Appendix 11). Regardless of the training mode, all participants were shown what the target image should look like if the task is completed successfully (see example of training module in Table 4).

To ensure the quality of training, the curriculum used in this study was tested and refined based on the results of a December 2006 pilot study (see Table 2). This study, with eight participants, was performed to compare book and simulation training conditions (formative evaluation). The intent was not only to identify potential weaknesses in training materials and potential areas where improvement opportunities can be realized, but also to guide the development of ITS.

Table 4: An example of a training module in the Paint.NET training used as a guide to complete an image processing task.

What's the Point?

Your aunt sent you a photo of one of her sheep. You decide to make a funny e-card using this image and then share it with your sister. By mistake, the picture of a sheep is on its side, so you will have to rotate the image to look correct. Also, to better fit the layout of the card you are imagining, flip it horizontally so that the sheep faces left, instead of right. Your sister will probably post it on a web site, so a resolution of 72 dpi (pixels/dots per inch) would be appropriate for this image. In general, an image may need rotating if it was taken with the camera on its side. You might flip an image for aesthetic reasons.

Concepts:

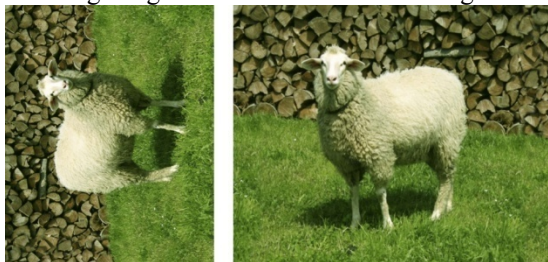
Two-dimensional digital images have three different sizes to consider: the pixel size (ex: 350x500 on the screen), print size (ex: 4x6 inches on the paper), and the image file size (ex: 962 KB on your hard drive). A large image file has pros and cons. A large image file leads to higher quality prints. However, a large image requires more storage space and takes longer to download. When you resize an image, you can specify the pixel size or the print size that you desire. If you decrease the number of pixels (dots of color in the image), then you effectively throw away quality. Sometimes that's ok, if you don't mind making the image smaller or using it only on the web. Web images are lower resolution than printed images. Resolution is the number of pixels that print within one inch of space. Also, Paint.NET has a zooming feature that works like a magnifying glass on your screen, but it does not change the size of the image.

Your task is to produce two images:

- 1) You need to rotate and flip the image Task1-1-sheep.jpg so that the sheep looks to the left. Also, set the resolution to 72 dpi for the web and resize the image to 625x500 pixels.
- 2) You need to produce a good quality photo print to mail to your aunt. This image needs a resolution of 300 dpi and the print size of 5x7 inches.

Images:

Starting image <-----> Desired image



Step-by-step instruction:

Learners in book group will receive printed step-by-step instruction on how to complete the task described above, while learners in simulation and ITS groups will receive guidance and help from the system they use.

3.4 Measures

Participants were required to complete the following six measures:

- a 10-item demographic questionnaire (see Appendix 4);
- a 10-item spatial ability: paper folding test (see Appendix 3);
- a 31-item self-efficacy survey:
 - a 12-item general self-efficacy (see Appendix 5);
 - an 11-item computer self-efficacy (see Appendix 6); and
 - an 8-item intermediate Paint.NET state self-efficacy (see Appendix 7).
- a 10-item system usability perception survey with an additional 3 open ended questions (see Appendix 8);
- a 10-item Paint.NET concept acquisition test (see Appendix 9);
- a 3-tem Paint.NET task performance project (see Appendix 10)

Demographic questionnaire:

All participants filled out a demographic questionnaire consisting of 10 items. They were asked to report their gender, age, major, and year in school. The participants also provided information on their computer proficiency and their experience with image manipulation software and ITSs (see Appendix 4).

Spatial ability:

Spatial ability level were assessed using the 10 item paper folding test of Ekstrom et al. (1976). This instrument, found in the ETS “Manual for Kit of Factor-Referenced Cognitive Tests,” has been used extensively and has demonstrated validity for cognitive tasks

involving visualization, manipulation, or transformation of images. The test involved showing participants a sequence of folds in a piece of paper. The last image in the sequence showed where a hole had been punched through the entire thickness of the paper. Next, they were shown another set of figures of completely unfolded papers with some small circles drawn on them. Participants had to choose the drawing with holes that correspond to the ones of the paper with holes punched through. They had three minutes to complete the test (see spatial ability test, Appendix 3).

Self-efficacy:

Three subscales of self-efficacy were administered in this study (see Appendix 5, 6, and 8). For the construct of general self-efficacy, the subscale that Bosscher and Smit adapted from an earlier scale was used (1998). An example of a general self-efficacy subscale item is “if I can’t do a job the first time, I keep trying until I can.” Participants responded to each of the items using a five-point Likert scale with values ranging from “strongly disagree” to “strongly agree.”

Another instrument was the 11 item computer self-efficacy scale adopted from Compeau & Higgins (1995). This instrument is commonly used to gauge a person’s ability to use an unfamiliar piece of software, since workers and students often encounter new software that is to make jobs or tasks easier to complete. An example of a computer self-efficacy question is “I can use a Paint.NET training system successfully if there is someone to give me step by step instructions.” Subjects responded to each item using a five-point Likert scale ranging from “not confident” to “very confident.”

For the construct of state self-efficacy, an eight question subscale was adopted from Blair et al. (1999). This state self-efficacy instrument is often used to measure user confidence over time; therefore, the results of this scale are dynamic and change according to the situation. An example of a state self-efficacy item is “After the Paint.NET training I received, I feel that during the test I will do well on navigating the menus in Paint.NET.” Participants responded to each item using a five-point Likert scale ranging from “unlikely” to “very likely.”

System usability perception:

Participants were asked to complete the 10-item system usability questionnaire developed based on the scale of Brooke (1996). His system usability scale (SUS) is widely used to measure user’s subjective reactions to navigating the system. His scale can be used as a stand-alone evaluation or as part of a user test and it can be applied to any product (Jacko & Sears, 2003). It was created by a group of professionals then working at Digital Equipment Corporation. It is generally used after the respondent has had an opportunity to use the system being evaluated, but before any debriefing or discussion takes place. Participants were asked to record their immediate response to each item, rather than thinking about items for a long time. In addition to 10-item SUS questions that have a Likert scale format, participants completed three attitudinal questions about things they like and dislike, and whether the training they receive help them learn better than they would have learned without it (see Appendix 8).

Concept acquisition:

One week after training, participants returned to complete both a conceptual test and task performance projects. Participants' conceptual knowledge was assessed using ten mastery questions drawn from the *Concept* and *What's the Point?* sections that were presented throughout the training (see training manual, Appendix 11 and concept acquisition test, Appendix 9). For example, participants could be asked a multiple-choice question:

Why would you reduce the size of an image (decrease the width and height)?

- A) Smaller images are better for archival purposes.
- B) Smaller images are better for printing.
- C) Smaller images are higher quality.
- D) Smaller images are easier to download from the Internet.

Paint.NET task performance projects:

Participants' task performance were assessed based on their ability to complete three image manipulation projects using Paint.NET without the assistance of ITS tutor, simulation, or book manual. This test showed image editing skills and the ability of participants to transfer what they learn into the real world scenario.

All testing instruments that were used in the fuller study were previously used in a December 2006 pilot study. The goal was not only to ensure the quality of training material and identify where system improvement opportunities can be realized, but also it was necessary to ensure that survey and testing instruments measured what they were designed for.

3.5 Setting and procedure

Paint.NET training and testing experiment sessions were administered in a spacious, modern campus classroom equipped with 40-networked computers distributed in 8 x 5 rows. Computers were alternatively tagged with three different colors and letters (letter A on the blue tag represented book, letter B on the yellow tag represented simulation, and letter C on the pink tag represented ITS condition). Three training sessions took place on the first day, while three testing sessions were conducted on the second day. All sessions were run in the afternoon to accommodate the schedules of the experiment participants.

Upon the arrival for training on the first day, participants were randomly assigned to different training condition. Each participant was handed a paper card that match the tag on the computer. Cards were handed out in ABC—ABC order to ensure that participants are evenly assigned to different training conditions. Four facilitators monitored activities by helping to set up the room, providing instructions, and assisting in case technical issues arose. All clicking behaviors and images from participants' work were monitored and recorded for evaluation purposes. In order to keep their responses anonymous and confidential, they were asked to use the last four digits of their student ID, which were not linked to any of the information they provided when they voluntarily sign up to participate in the study.

Once participants were seated, the session started with a welcome message (see Appendix 1). One of the facilitators gave them an overview of what the experiment is about with an emphasis on the sequence of the events during the training session and testing session (see sequence of activities in Table 5). Other information the facilitator provided included incentives, confidentiality, and dealing with technical issues (e.g: what to do if the

application freezes). Participants were asked to read and sign an informed consent form; signed copies were available for participants to take home (see informed consent form, Appendix 2). They were advised that they can answer or skip any question they feel uncomfortable with or withdraw from the study at any time without penalty.

Next, participants completed a series of online surveys (spatial ability test, general self-efficacy test, and computer self-efficacy) and watched a video clip that introduced them to the basic features of Paint.NET (headphones were provided). Then, they started the actual Paint.NET training. Participants in the book group interacted with Paint.NET, while trainees in the simulation interacted with Adobe Captivate simulation, rather than the actual Paint.NET software they were learning. Similarly, participants in the ITS condition interacted with Paint.NET enhanced by an embedded tutor (see Figure 3, Figure 4, Figure 5 for examples of the interface learners see in the middle of a problem). At the completion of training, participants took two surveys: Paint.NET state self-efficacy and system usability survey. On the second day, participants were briefed on the sequence of the events, took a concept acquisition test, and completed task performance projects.

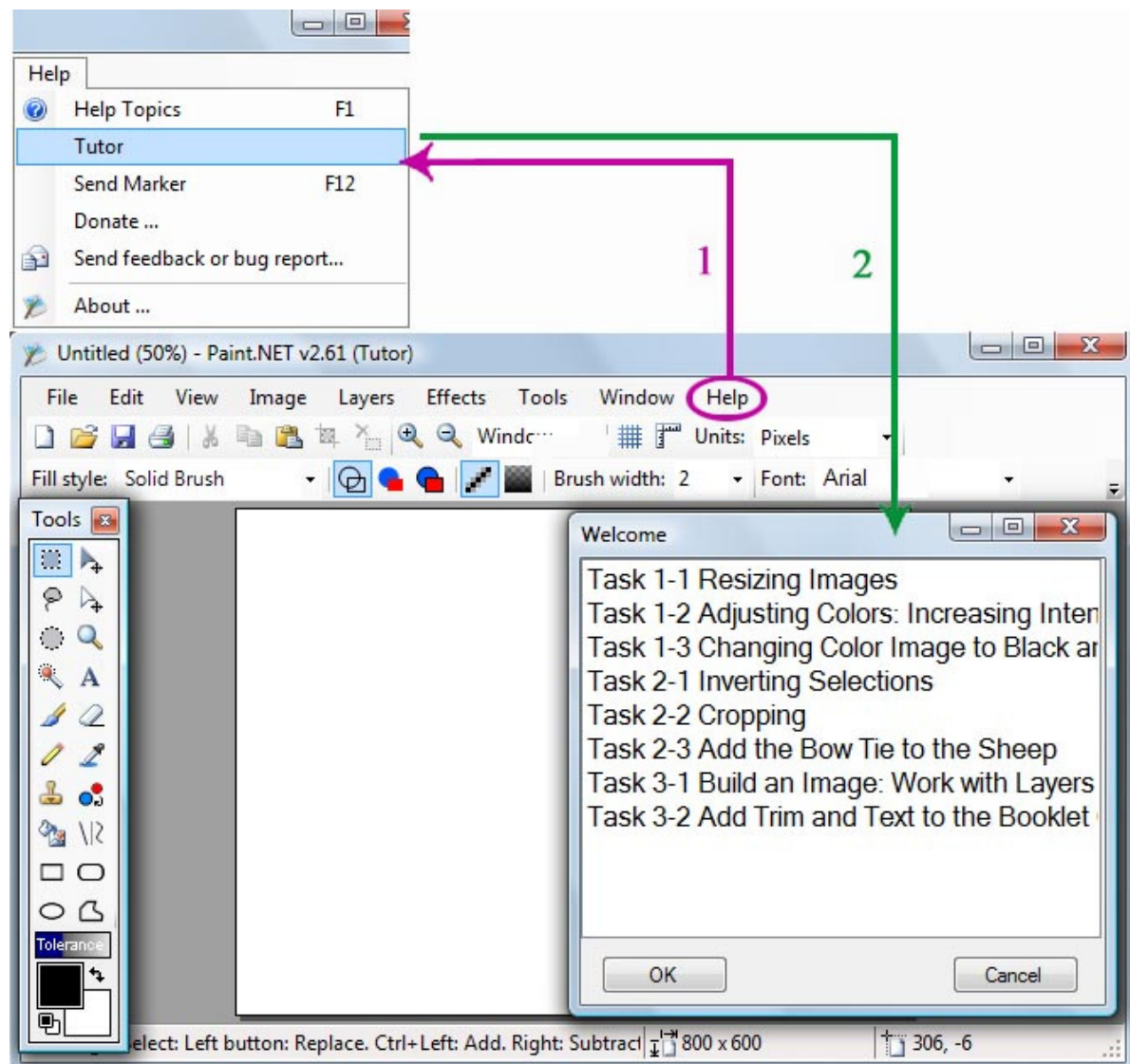


Figure 3: Paint.NET interface enhanced with ITS-tutor. A student activates the ITS-tutor by 1) accessing the Help menu and 2) choosing the Tutor to activate the ITS. When the ITS is started, a student is presented with a list of tasks to choose from. Student can choose a task to practice on a random order, but a sequential order is preferred. The remaining parts of the image show a typical window of the Paint.NET application.

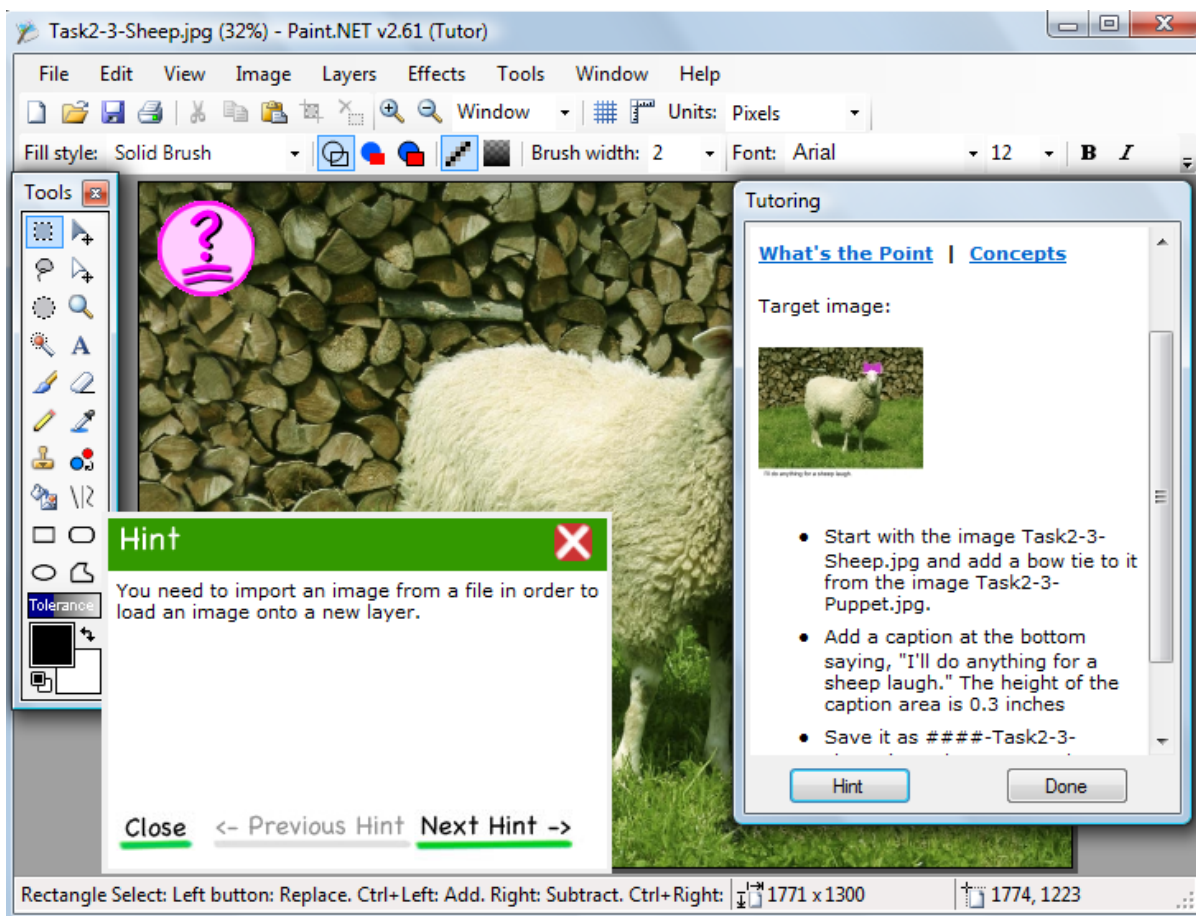


Figure 4: Paint.NET interface enhanced with ITS-tutor.

This screenshot shows the screen as it looks to a student in the middle of a problem. The question mark error icon, the Tutoring, and Hint windows are a part of the ITS tutor.

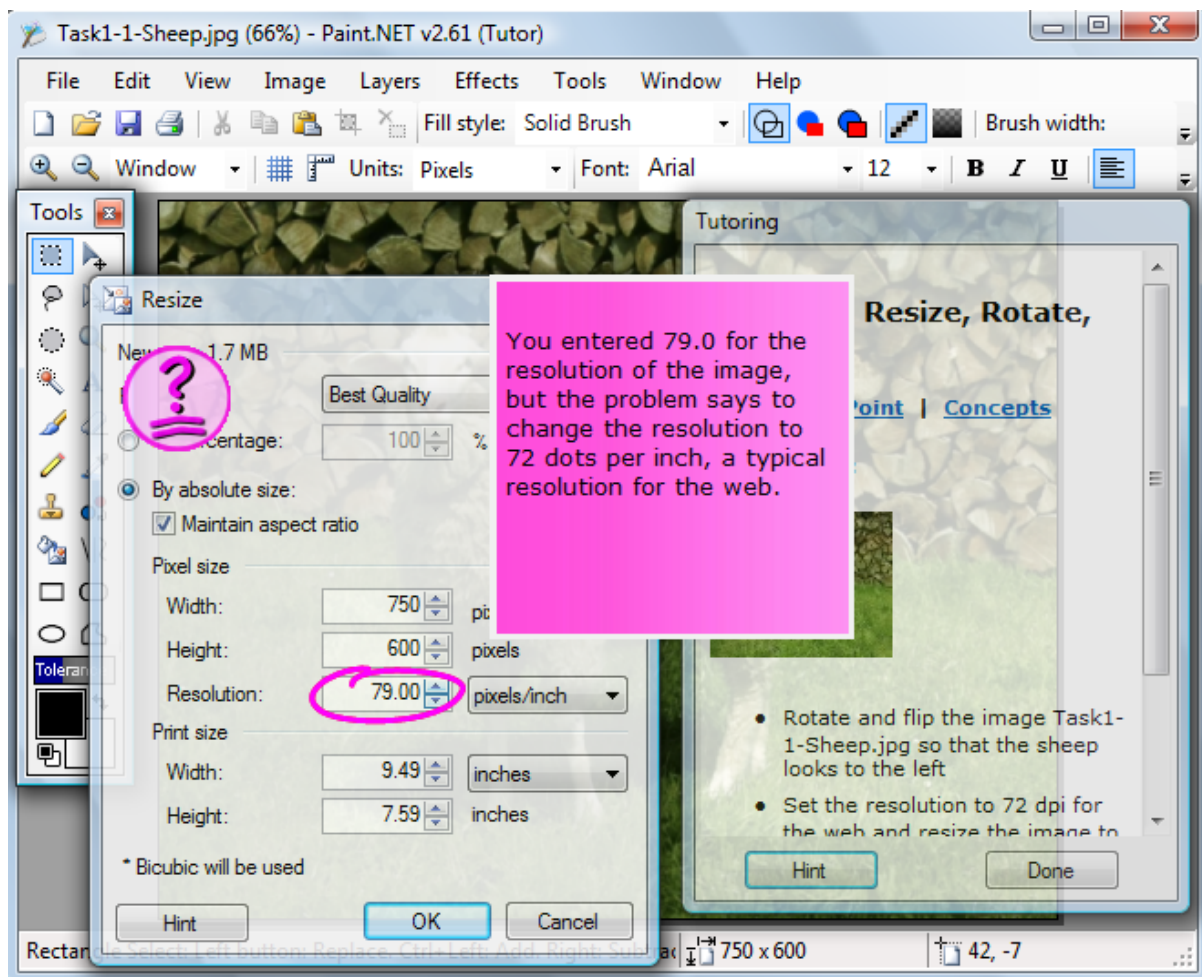


Figure 5: Paint.NET interface enhanced with ITS-tutor. The Resize and Tutoring windows are part of the ITS tutor. The message in the center of the screen (a window without a title) is a just-in-time message that provides guidance to help a student get back to the right path to the solution.

Table 5: Sequence of training and testing activities with estimated times

Sessions	Activities	Estimated Time	References
<i>Day one, training session</i>	1. Welcome message	3 minutes	Appendix 1
	2. Signing informed consent form	7 minutes	Appendix 2
	3. Spatial ability test: Paper folding (timed)	3 minutes	Appendix 3
	4. Introductory video (overview of Paint.NET features)	2 minutes	-
	5. Demographic questionnaire	4 minutes	Appendix 4
	6. General self-efficacy (GSE)	4 minutes	Appendix 5
	7. Computer self-efficacy (CSE)	4 minutes	Appendix 6
	8. Paint.NET state self-efficacy	3 minutes	Appendix 7
	9. Paint.NET training consisting of 8 training tasks	55 minutes	Appendix 11
	10. System usability survey	5 minutes	Appendix 8
	<i>Duration of entire training session</i>	90 minutes	
<i>Day two*, testing session</i>	1. Welcome message	5 minutes	-
	2. Concept acquisition test	10 minutes	Appendix 9
	3. Paint.NET task performance project (3 projects)	30 minutes	Appendix 10
	4. Compensation and iPod sweepstake	15 minutes	-
	<i>Duration of entire testing session</i>	60 minutes	-

* One week after initial training, all participants returned for testing session.

3.6 Data analysis:

This empirical study, in which participants used only one product, is between-subjects design. The research model included dependant, independent, and moderating variables (see Figure 6). The dependent variable was learner's performance measured by score received on training success indicators. The independent variables were the different training modes. The moderating variables were the learner's spatial ability and self-efficacy. Clouse and Evans (2003) used similar variables to focus research on individual learner characteristics that make them successful in different delivery modalities.

Data were collected to identify system differences. The learning outcome data (dependant variables), the interaction method (independent variables), and learner's ability to succeed covariates were analyzed using SAS general linear models (GLM) procedures (SAS Institute, 2003) and SPSS V12.0 (SPSS Inc., Chicago IL). Time spent on each training task

was computed throughout the final images participants created, and add the times to ascertain total training time. Next, the ten multiple choice questions for conceptual test were graded and each participant was assigned a total grade. For test performance, three image manipulation projects were completed without assistance and against the project rubric. Students were asked to complete a system usability survey, which were included open-ended questions to give insight on participants' perception of the system. Then, the means were calculated and compared based on the training mode to ascertain the best method (ITS, training manual, or simulation.) Additionally, the relationships between training mode and training success indicators (TSI) were assessed. Training mode were the independent variable, while TSIs were dependent variables. Similarly, correlations between TSI and interaction between training condition and LASI were determined (see research model Figure 6). The LASI were ranked either high or low based on the rating received (High: if score \geq median; low: if score $<$ median). These descriptive values were cross-linked to the training condition to establish the significance of their effect on the training outcomes. The hypothesis is that regardless of the training condition participants with high spatial ability would outperform participants with low spatial ability; likewise, the same effect was expected for self-efficacy measures.

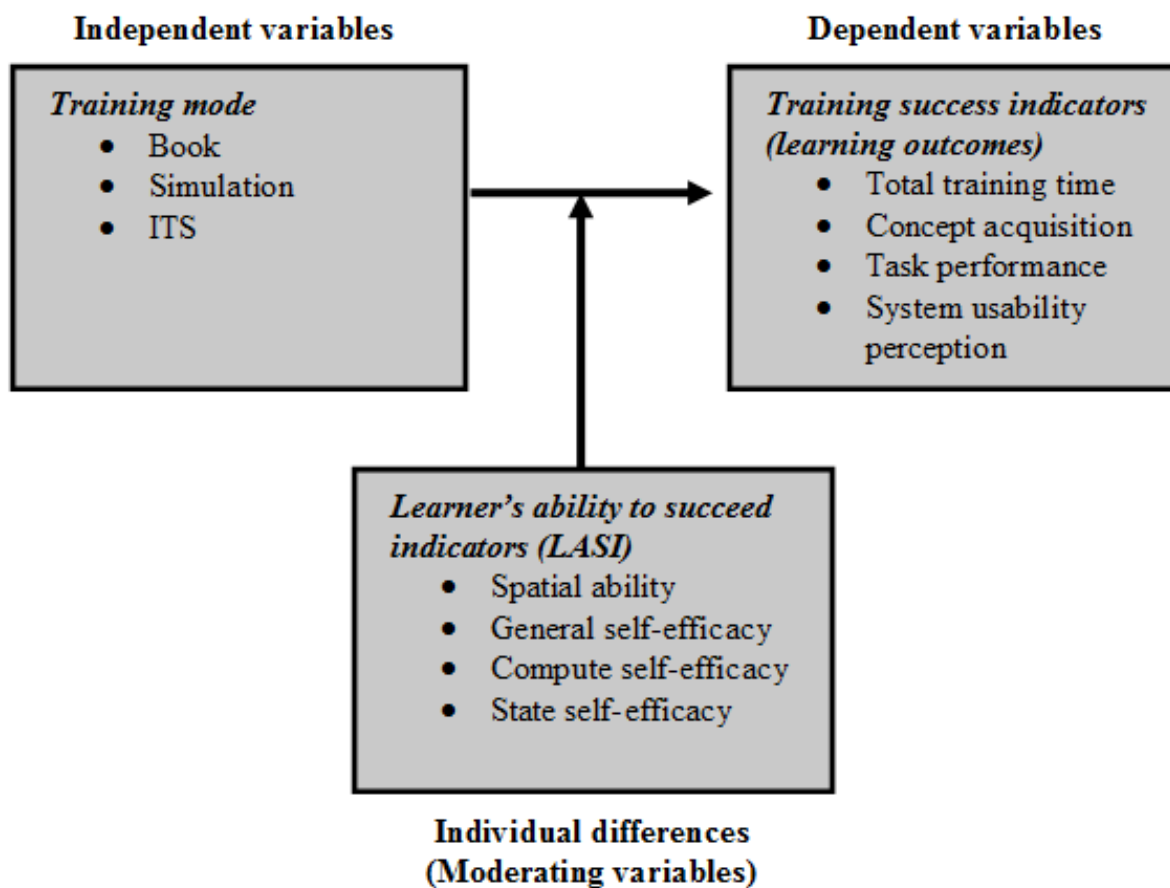


Figure 6: Research Model - Factors that impact training success

CHAPTER 4. RESULTS AND DISCUSSION

The empirical study presented in this chapter summarizes a Paint.NET tutoring experiment run in December 2007. Data were analyzed to assess the impact of training mode and learner's ability to succeed indicators (LASI) on learning success. The effect of a combination of training mode and LASI (interaction effect) was also explored. The sample of this study consisted of 75 participants of whom five did not disclose their demographic information. Thirty nine male and thirty one female participants were identified. Sixty nine were in the 18-24-age group and one participant was in the 25-35-age group. All participants reported that they use computers everyday. Most of them ($n=50$) described themselves as average computer users, though there were 19 advanced users and one novice computer user. None of them had experience with Paint.NET or was familiar with an intelligent tutoring system. Their levels of education ranged from freshman (16), sophomore (21), junior (20), and senior (11). Participants were from a wide variety of majors including business (18), engineering (15), education, and psychology (5). Statistical analyses were carried out using SAS (SAS Institute, 2003) and SPSS V12.0 (SPSS Inc., Chicago IL); $p < 0.05$ was considered significant. One-way ANOVA was used to compare training outcomes across training modes. Two-way ANOVA were also used to explore the impact that learner's ability to succeed indicators have on the relationship between training mode and training outcomes.

4.1 Handling missing data

All training activities were tracked around the clock, recorded, and care was taken to minimize discrepancies associated with missing data. One participant in the simulation group completed one task out of eight training tasks. Training and testing data of this participant

were excluded from the analysis because his/her total training time was an extreme outlier. It is believed that this training experience was too limited to make a positive impact on the test score or overall system usability rating. In addition, three cases (one person in the book group and two people in the ITS group) of incomplete data due to software crashing were identified. The participant in the book group did not complete three training tasks while each of the two participants in the ITS group did not complete one task. An example of a technical issue included unexpected shutdown of the program, forcing the participant to restart a task, or errors logging a participant's actions.

It was observed that the majority of participants complete training tasks in a sequential order. Most of the missing records were the ones near the end of the list of activities to be completed. This pattern suggests that to eliminate all incomplete data (20 cases out of 75 records) would have been inappropriate because some participants just may be slower while others may be faster. Multiple imputation procedure (expectation-maximization) was used to predict missing values in total training time data set. An analysis with exclusion of incomplete data and another with imputed missing values were run and compared. The data sets for spatial ability, GSE, CSE SSE had few missing values and records with missing values in those data sets were eliminated from the analysis because they only marginally reduced the sample sizes.

All participants (n=75) completed a concept acquisition test. Data for nine participants were excluded from task performance measures because they did not finish at least 1/4 of task performance project. Their scores were outliers. One participant did not complete the usability perception survey. A similar approach was used to deal with missing data for spatial ability and self-efficacy measures.

4.2 Impact of training mode on training outcomes

The first research question was to determine the effect of training condition on four learning success indicators. The expectation was that the ITS group would outperform the other groups, in terms of total training time, post-test measures, and overall system perception. What follows are results showing that some learning outcome scores significantly correlated with training mode. See descriptive statistics in Table 6.

On the task performance score the book group ($M=25.00$) and the ITS ($M=22.56$) were significantly different than the score of simulation group ($M=17.70$). The scores of book and ITS however, were not significantly different. This finding was consistent with the results of a December 2006 pilot study with seven participants where the means of the book and the ITS groups on both concept and task performance scores were not significantly different at the outset.

Likewise, the overall system usability perception rating for the book group ($M=60.28$) was significantly better than the rating for the ITS ($M=50.91$); but the ratings of book and simulation were not statistically significantly different. This was not the case during the pilot study where simulation group significantly reported their system as more usable than the book group. Due to the small sample size ($n=7$); however, it was hard to draw strong conclusions from the pilot study.

Despite this, the present imputed data set ($n=74$) showed that the simulation group ($M=37.06$) used less total training time (less time being desirable) than both the book ($M=52.56$) and the ITS ($M=69.18$), a significant difference. Total training time for book and ITS were also significantly different. Similarly, a second analysis without missing values

(n=55) showed better total training time for simulation than both book and the ITS. However, the total training times for book and ITS were not significantly different.

Table 6: Comparison of participant scores on measures of training mode success

	n	Min	Max	Mean
<i>Total training time in minutes (F(2,71)=3.12, p=.00)</i>				
Book	24	31.33	94.79	52.96a
Simulation	24	26.07	48.78	37.06b
ITS	26	48.60	86.80	69.18c
<i>Concept acquisition score (F=(2,72)=1.46, p=0.24)</i>				
Book	24	4.00	9.00	6.92a
Simulation	25	1.00	10.00	6.12a
ITS	26	2.00	9.00	6.12a
<i>Task performance score (F(2,64)=3.86, p=0.03)</i>				
Book	18	11.00	40.00	25.00a
Simulation	25	4.50	35.00	17.70b
ITS	24	6.00	34.50	22.56a
<i>Usability perception score (F(2,71)=7.61, p=0.00)</i>				
Book	24	45.00	76.50	60.28a
Simulation	25	40.50	72.00	59.89a
ITS	25	33.75	69.75	50.91b

Means followed by the same letter within column are not significantly different (p>0.05)

4.3 Impact of learner's ability to succeed indicators and training mode on training outcomes

The second research question focused on identifying significant relationships between learner's ability to succeed indicators (LASI) and learning success indicators (training outcomes), while the third question asserted the impact of a combination of training condition and LASI on training outcomes by exploring the significance of interaction effect. Results from single-factor ANOVA suggested that some of the correlations between LASI and LSI were significant. Ratings received on LASI measures were classified either as high

or low (High: if rating score \geq median; Low: if rating score $<$ median). A two-way between-groups analysis of variance was conducted to assess the impact of a combination of training mode and LASI on measures of training outcomes. Two statistically significant interaction effects on usability perception were identified. Those two LASIs with significant interaction effects are spatial ability ($F(2, 64)=5.71, p=0.00$) and general self-efficacy ($F(2, 61)=3.58, p=0.04$). All other interaction effects were not significant (see Table 7 and Table 8).

4.3.1 Effect of spatial ability (SA) and training mode on training outcomes

In addition to statically significant main effects for high SA ($F(2, 34)=3.29, p=0.00$) and for low SA ($F(2, 29)=3.35, p=0.01$), the interaction effect on system usability measure ($F(2, 64)=5.71, p=0.00$) reached statistical significance. Independent of the high or low levels of SA, participants perceived book and simulation to be superior to the ITS. Furthermore, participants in simulation recorded better total training time; and those with high SA ($M=34.61$) significantly outperformed their counterpart with low SA ($M=40.73$) on training time. A similar SA effect was found within the book group, a finding that was not observed before imputation of missing values. See Table 7. This finding was consistent with the initial prediction that high-spatial learners would have an increased likelihood of success than low-spatial learners do (Kyllonen, et al., 1984). However, results failed to validate the assumption that higher spatial ability levels would increase the likelihood of success independent of the training approach. All other correlations with spatial ability were not significant. Despite the failure to reject null hypothesis for SA effect on task performance score, there is an indication that high-spatial learners performed somewhat better than low-spatial learners

independent of the training mode (see Figure 7). Thus, there is a chance that further research with a larger sample would seemingly support this finding.

Table 7: Comparison of participant training outcomes based on training mode and learner's spatial ability

Spatial ability (SA) measure	Book	ITS	Simulation
Total training time in minutes			
High SA (F(2, 31)=3.30, p=.00)	n=13, M=45.77a	n=9, M=66.92a	n=12, M=34.61a
Low SA (F(2, 29)=3.33, p=.00)	n=9, M=55.17b	n=14, M=69.03a	n=9, M=40.73b
Interaction effect: training mode x SA (F(2, 63)=0.22, p=0.81)			
Concept acquisition score			
High SA (F(2,34)=3.30, p=0.60)	(n=11, M=7.10)a	(n=11, M=6.36)a	(n=13, M=6.46)a
Low SA (F(2,26)=3.40, p=0.25)	(n=7, M=7.14)a	(n=10, M=6.20)a	(n=10, M=5.60)a
Interaction effect: training mode x SA ((F(2, 61)=0.35, p=0.71)			
Task performance score			
High SA (F(2,32)=3.29, p=0.23)	(n=12, M=26.42)a	(n=10, M=23.00)a	(n=13, M=19.88)a
Low SA (F(2,24)=3.44, p=0.33)	(n=5, M=20.20)a	(n=11, M=22.23)a	(n=9, M=16.11)a
Interaction effect: training mode x SA (F(2, 59)=1.23, p=0.31)			
Usability perception score			
High SA (F(2,34)=3.29, p=0.00)	(n=12, M=63.38)a	(n=10, M=49.73)a	(n=13, M=60.75)a
Low SA (F(2,29)=3.35, p=0.01)	(n=8, M=61.88)a	(n=13, M=50.02)a	(n=9, M=62.25)a
Interaction effect: training mode x SA (F(2, 64)=5.71, p=0.00)			

Means followed by the same letter within column are not significantly different (p>0.05)

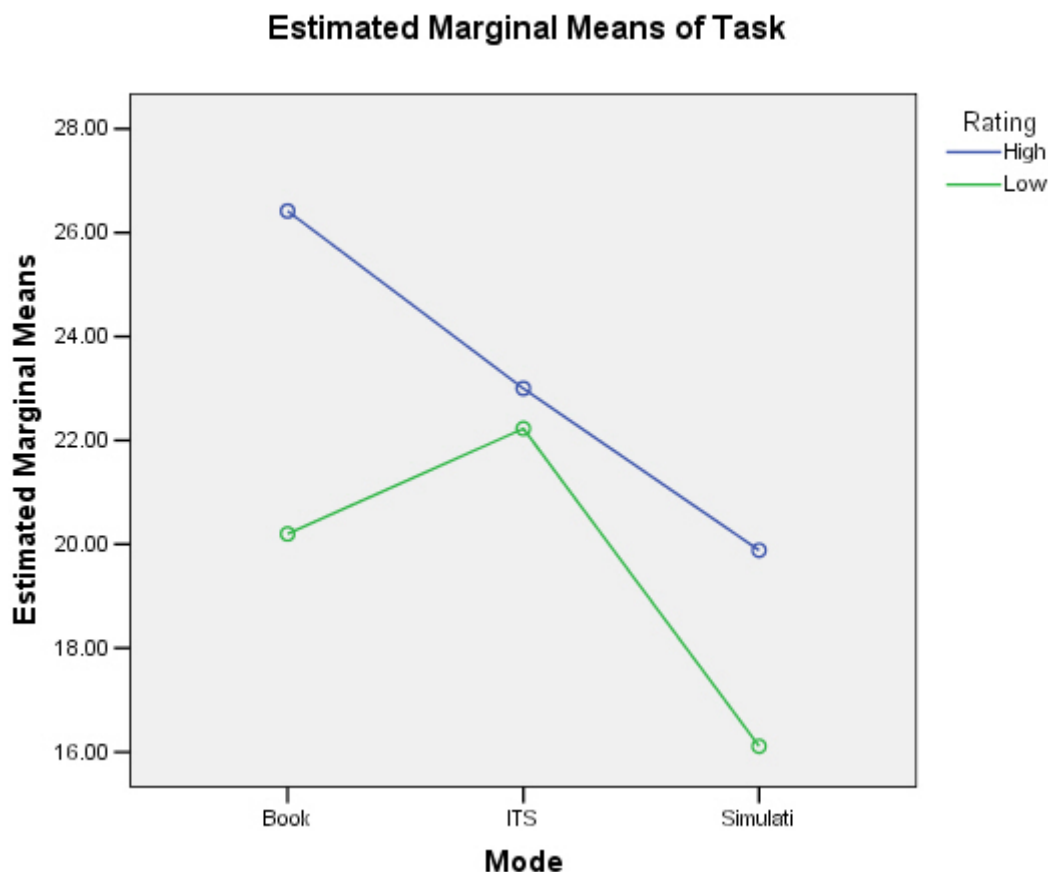


Figure 7: Effect of spatial ability (SA) on task performance.

4.3.2 Effect of general self-efficacy and training mode on training outcomes

Results indicated that the interaction effect of GSE and training mode reached statistical significance on system usability measure ($F(2, 61)=3.58, p=0.04$). However, GSE alone did not have significant effect on system usability rating. Main effect for high GSE ($F(2, 33)=3.30, p=0.00$) and low GSE ($F(2, 27)=3.39, p=0.00$) were also statistically significant. See Table 8 and Figure 8. Despite high or low levels of GSE, participants in the book group reported significantly better system usability experience than ITS; and

participants in simulation recorded significantly shorter total training times than book and simulation. Within the simulation group, participants with high GSE ($M=63.00$) recorded significantly more positive perception ratings on system usability than those with low GSE (55.43), a finding that was constant with the initial prediction. However, all other correlations with GSE were not significant. Like the findings for SA, it is suggested that participants with high GSE somewhat outperformed these with low GSE on task performance measure.

Table 8: Comparison of participant training outcomes based on training mode and learner's general self-efficacy

General self-efficacy (GSE) measure	Book	ITS	Simulation
Total training time in minutes			
High GSE ($F(2, 34)=3.28, p=.00$)	$n=13, M=54.92a$	$n=11, M=67.97a$	$n=13, M=36.53a$
Low GSE ($F(2, 33)=3.30, p=.00$)	$n=9, M=46.81b$	$n=14, M=70.32a$	$n=11, M=37.95a$
Interaction effect: training mode x GSE ($F(2, 68)=2.16, p=0.12$)			
Concept acquisition score			
High GSE ($F(2,31)=3.33, p=0.62$)	$(n=9, M=7.33)a$	$(n=10, M=7.30)a$	$(n=13, M=6.69)a$
Low GSE ($F(2,21)=3.37, p=0.84$)	$(n=7, M=6.86)a$	$(n=7, M=5.43)a$	$(n=8, M=5.38)a$
Interaction effect: training mode x GSE ($F(2, 53)=0.66, p=0.52$)			
Task performance score			
High GSE ($F(2,30)=3.34, p=0.45$)	$(n=8, M=25.19)a$	$(n=10, M=25.60)a$	$(n=13, M=20.54)a$
Low GSE ($F(2,25)=3.42, p=0.05$)	$(n=8, M=22.81)a$	$(n=9, M=21.61)a$	$(n=9, M=15.17)a$
Interaction effect: training mode x GSE ($F(2, 56)=0.13, p=0.88$)			
Usability perception score			
High GSE ($F(2,33)=3.30, p=0.00$)	$(n=9, M=59.25)a$	$(n=11, M=51.55)a$	$(n=14, M=63.00)a$
Low GSE ($F(2,27)=3.39, p=0.00$)	$(n=8, M=65.53)a$	$(n=9, M=49.50)a$	$(n=11, M=55.43)b$
Interaction effect: training mode x GSE ($F(2, 61)=3.58, p=0.04$)			

Means followed by the same letter within column are not significantly different ($p>0.05$)

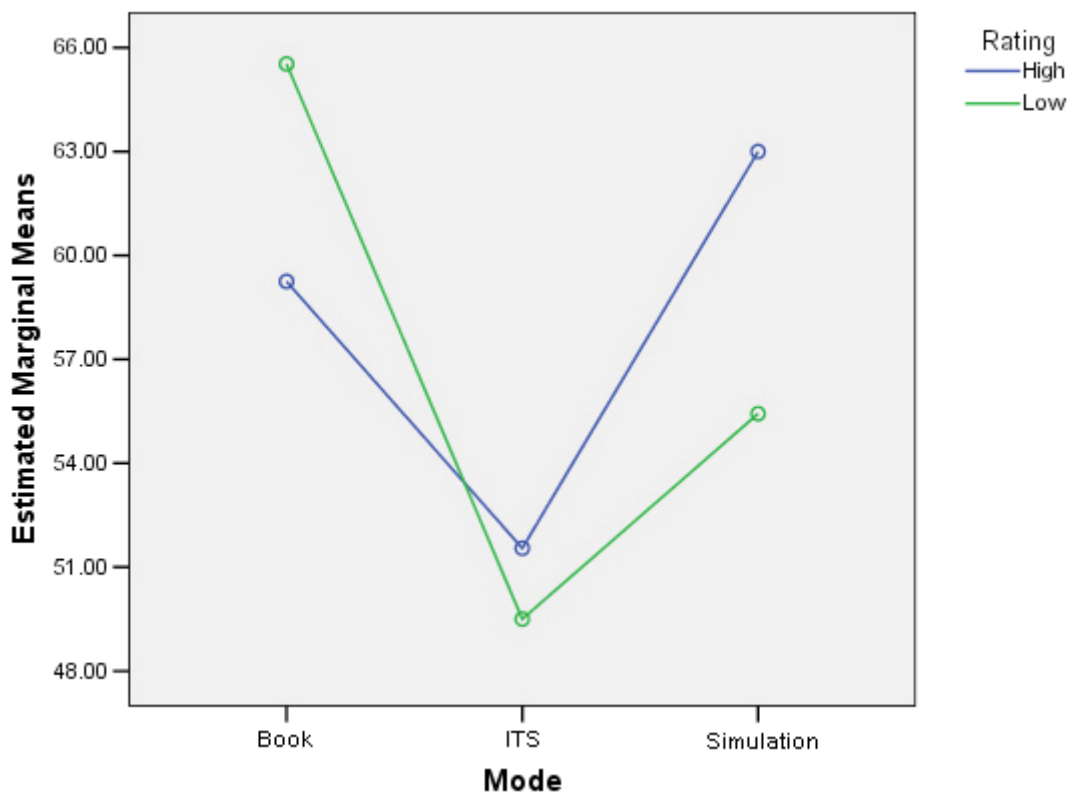


Figure 8: Effect of general self-efficacy (GSE) on system usability perception.

4.3.3 Effect of computer self-efficacy and training mode on training outcomes

The interaction effect of training condition and computer self-efficacy (CSE) on task performance did not reach statistical significance. However, both training mode (as shown above) and CSE when considered individually do have a significant impact on task performance score. Unlike the main effect for high CSE ($F(2, 32)=3.32, p=0.18$), it was realized that participants with low levels of CSE in the book group ($M=25.81$) achieved significantly higher task performance success than participants in simulation ($M=15.7$) and ITS ($M=16.85$) groups. CSE was the only indicator that exhibited interesting outcomes on measure of task performance. Within the ITS group, participants with high CSE ($M=27.82$)

did significantly better than participants with low CSE (M=16.85). This result was consistent with the prediction of Mulkey and O’Noel (1999) that learners with higher levels of computer self-efficacy perform better than those with lower levels of self-efficacy in computer training situations. In addition, the means of task performance measure for those with low levels of CSE significantly differed across the groups while those with high levels of CSE were relatively stable. Findings of this study suggest that learners with low CSE may perform better in book-based training and those with high CSE may perform better in ITS group (see Table 9 and Figure 9), but further research is warranted. This is an interesting finding particularly for organizations that concerned with ways to increase technology productivity and efficiency (see Carr, 2000).

Table 9: Comparison of participant training outcomes based on training mode and learner’s Computer self-efficacy

Computer self-efficacy (CSE) measure	Book	ITS	Simulation
Total training time in minutes			
High CSE (F(2, 33)=3.28, p=.00)	n=9, M=49.98a	n=14, M=66.03a	n=13, M=36.68a
Low CSE (F(2, 30)=3.32, p=.00)	n=11, M=50.11a	n=12, M=72.85a	n=10, M=37.70a
Interaction effect: training mode x CSE (F(2, 66)=0.91, p=0.41)			
Concept acquisition score			
High CSE (F(2,32)=3.32, p=0.95)	(n=10, M=7.00)a	(n=13, M=6.77)a	(n=10, M=6.8)a
Low GSE (F(2,27)=3.39, p=0.66)	(n=7, M=5.86)a	(n=8, M=6.38)a	(n=13, M=5.62)a
Interaction effect: training mode x CSE (F(2, 60)=0.30, p=0.74)			
Task performance score			
High CSE (F(2,32)=3.32, p=0.18)	(n=9, M=23.50)a	(n=11, M=27.82)a	(n=13, M=20.54)a
Low CSE (F(2,26)=3.40, p=0.02)	(n=8, M=25.81)a	(n=10, M=16.85)b	(n=9, M=15.17)a
Interaction effect: training mode x CSE (F(2, 59)=2.84, p=0.07)			
Usability perception score			
High CSE (F(2,34)=3.29, p=0.01)	(n=9, M=61.75)a	(n=13, M=52.27)a	(n=13, M=61.96)a
Low CSE (F(2,27)=3.39, p=0.00)	(n=10, M=65.48)a	(n=10, M=46.80)a	(n=8, M=59.34)a
Interaction effect: training mode x CSE (F(2, 62)=1.81, p=0.17)			

Means followed by the same letter within column are not significantly different ($p > 0.05$)

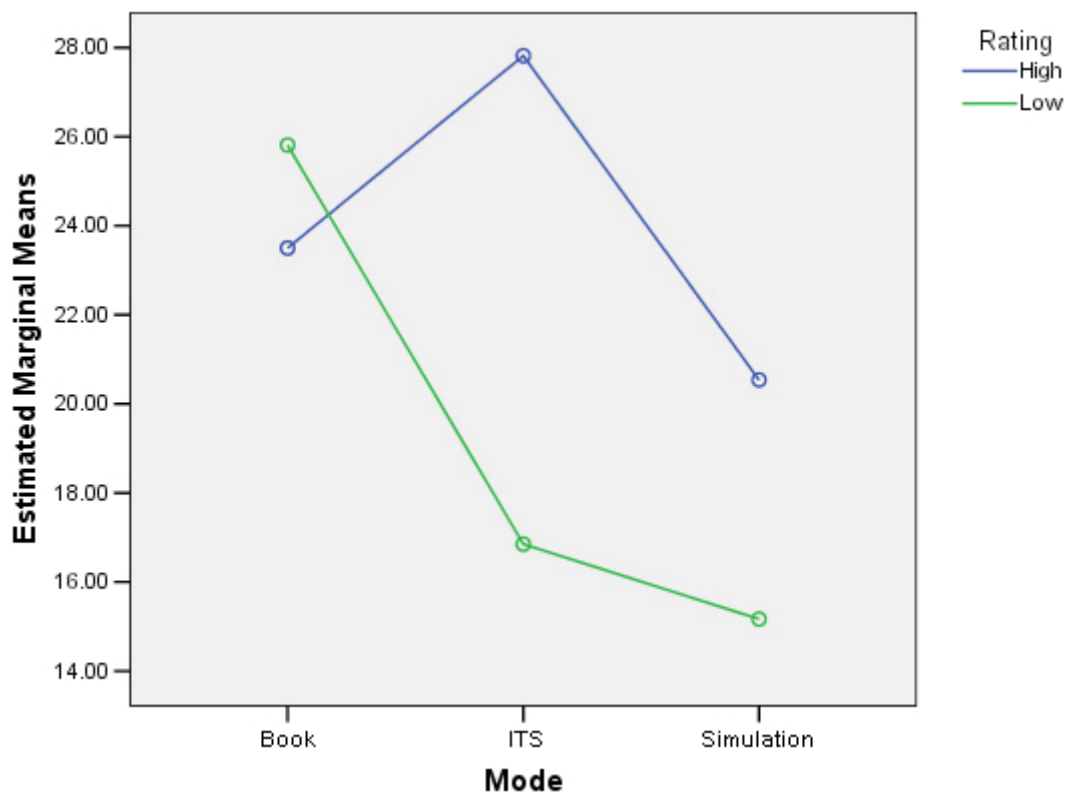


Figure 9: Effect of computer self-efficacy (CSE) on task performance score.

4.3.4 Effect of state self-efficacy and training mode on training outcomes

There was no statistically significant interaction effect of Paint.NET SSE and training mode on LSI. But, main effects on SSE on total training time and system usability were significant regardless of the level of SSE (see Table 10). Even though that SSE did not statistically correlate with task performance across training modes, there is an indication that SSE analysis showed a tendency of lower performance within simulation group (see Figure 10). Thus, there is a chance that further research with a larger sample would seemingly support this finding. All other main effects on LSI were not statistically significant.

Table 10: Comparison of participant training outcomes based on training mode and learner's Paint.NET state self-efficacy

Paint.NET state self-efficacy (SSE) measure	Book	ITS	Simulation
Total training time in minutes			
High SSE (F(2, 36)=3.26, p=.00)	n=11, M=54.92a	n=16, M=67.03a	n=12, M=36.78a
Low SSE (F(2, 30)=3.32, p=.00)	n=12, M=50.66a	n=9, M=73.29a	n=12, M=38.27a
Interaction effect: training mode x CSE (F(2, 69)=1.40, p=0.25)			
Concept acquisition score			
High SSE (F(2,31)=3.33, p=0.23)	(n=8, M=7.00)a	(n=12, M=6.33)a	(n=12, M=5.58)a
Low SSE (F(2,28)=3.37, p=0.84)	(n=10, M=6.90)a	(n=6, M=6.50)a	(n=13, M=6.46)a
Interaction effect: training mode x CSE (F(2, 60)=0.45, p=0.64)			
Task performance score			
High SSE (F(2,34)=3.29, p=0.32)	(n=8, M=23.56)a	(n=14, M=23.82)a	(n=13, M=18.62)a
Low SSE (F(2,24)=3.44, p=0.31)	(n=8, M=24.44)a	(n=5, M=23.40)a	(n=12, M=18.29)a
Interaction effect: training mode x CSE (F(2, 62)=1.81, p=0.17)			
Usability perception score			
High SSE (F(2,35)=3.28, p=0.01)	(n=8, M=61.31)a	(n=15, M=51.30)a	(n=13, M=60.75)a
Low SSE (F(2,25)=3.42, p=0.01)	(n=9, M=63.00)a	(n=5, M=48.60)a	(n=12, M=58.50)a
Interaction effect: training mode x CSE (F(2, 61)=0.34, p=0.72)			

Means followed by the same letter within column are not significantly different ($p > 0.05$)

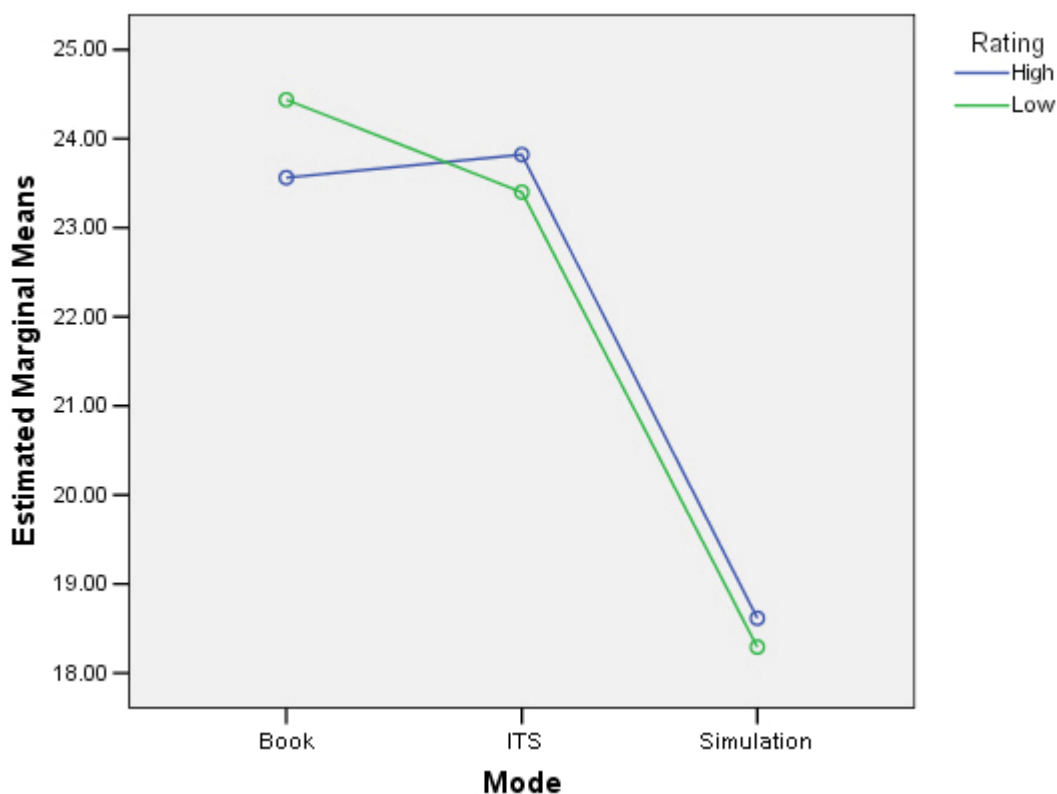


Figure 10: Effect of state self-efficacy (SSE) on task performance score.

Generally, results indicated that all four learner's ability to succeed indicators (SA, GSE, CSE, SSE) significantly correlated with total training time and system usability measures independently of high versus low levels of LASI. These findings suggest that there are other factors to consider beyond the simple assumption that a system that is easy to learn and use has the potential to minimize total time it takes to complete a task. Such a system also results in increased positive attitude toward the system. Results of this study show that a system that is easy to use and takes shorter time to complete the task (in this case: simulation) is not always the one given favorable usability rating from the user. This confirms that usability goes beyond the easy to learn and time it takes to complete training.

Other factors to consider beyond easy to learn and time include meeting specific learner goals, such as concept acquisition, task performance, and appeal, to name a few. It is also suggested that users who are primarily concerned with how much time it takes to complete training are better off in the simulation training mode. Another interesting finding is that users with low CSE who are mainly interested in task performance are better suited for the book condition than they would be in the ITS or Simulation modes.

Molich (2004) credibly asserted that a usability study with five users will find 85% of the usability problems. Due to small sample sizes and limited information on whether variables were normally distributed, a non-parametric test was performed to validate parametric tests with 95% confidence level used to analyze data presented above. A Kruskal-Wallis H Test was performed to compare scores on continuous variables. Kruskal-Wallis H Test is a non-parametric alternative to a one-way between-groups analysis of variance. The interesting observation was that non-parametric tests validated results from parametric analysis. Examples of parametric and non-parametric test results are shown in Table 11 and Table 12.

Table 11: Parametric and non-parametric test results are compared to validate findings from small sample size data. Training outcomes (continuous dependant variable) of three training modes (categorical independent variables) are observed. Data show similar results on both statistical analyses.

Training outcomes of training modes: Book, simulation, ITS	Results of Parametric tests*	Results of Non-parametric tests
Total training time in minutes	F(2,71)=3.12, p=0.00	Chi-Square=46.32, df=2, p=0.00
Concept acquisition score	F(2,72)=1.46, p=0.24	Chi-Square=2.71, df=2, p=0.26
Task performance score	F(2,64)=3.86, p=0.03	Chi-Square=7.19, df=2, p=0.03
Usability perception score	F(2,71)=7.61, p=0.00	Chi-Square=13.11, df=2, p=0.00

* Numbers shown in results of parametric tests column were extracted from Table 6

Table 12: Comparison of parametric and non-parametric test results from small sample size data used. The level of computer self-efficacy (CSE) within three training modes was categorical independent variable and training outcomes were continuous dependant variables.

Data showed similar results on both statistical analyses.

Training outcomes of training modes: Book, simulation, ITS	Results of Parametric tests*	Results of Non-parametric tests
Total training time in minutes		
High CSE	F(2, 33)=3.28, p=0.00	Chi-Square=23.54, df=2, p=0.00
Low CSE	F(2, 30)=3.32, p=0.00	Chi-Square=29.30, df=2, p=0.00
Concept acquisition score		
High CSE	F(2,32)=3.32, p=0.95	Chi-Square=0.09, df=2, p=0.95
Low CSE	F(2,27)=3.39, p=0.66	Chi-Square=1.95, df=2, p=0.38
Task performance score		
High CSE	F(2,32)=3.32, p=0.18	Chi-Square=2.74, df=2, p=0.26
Low CSE	F(2,26)=3.40, p=0.02	Chi-Square=6.15, df=2, p=0.04
Usability perception score		
High CSE	F(2,34)=3.29, p=0.01	Chi-Square=8.74, df=2, p=0.01
Low CSE	F(2,27)=3.39, p=0.00	Chi-Square=14.34, df=2, p=0.00

* Numbers shown in results of parametric tests column were extracted from Table 9

4.4 Results of qualitative measures

On the open-ended questionnaire (see Appendix 8), almost all participants in all conditions expressed that the training they received help them use Paint.NET better than they would have without it. The systems showed them where to find the features of the software, introduced them to new concepts and provided hands-on experience that taught something new. Only two out of 24 participants in the book condition and two out of 26 participants in the simulation indicated that the training modes they received were not helpful. For example, unsatisfied book participants felt that they would have had less frustration if there were offered a different type of training, otherwise they would have preferred to experiment on their own rather than following every step outlined in the book. At the same time frustrated simulation participants expressed that they would have preferred freedom to struggle while

discovering how Paint.NET works rather than being forced to one way of doing things when there could be other alternatives. Probably this was a matter of learning preferences that was not well matched with the assigned training conditions.

While participants with book-based instruction were told exactly what to do and enjoyed skimming and sometimes skipping background instructions, they also reported frustration completing tasks. There is some research evidence that learners who have higher control tend to over-estimate their ability. As result they may view less material and skip important instructional components. Positive findings show that learners with prior specific knowledge are better able to gauge their training needs and benefit from high degrees of learner control (Piccoli, Ahmad, & Ives, 2001). The book was criticized because it was not interactive, did not provide sufficient help, and was much like a long and arduous lab assignment. Participants ended up skipping some steps, which somewhat hurt their task performance even though they performed equally well as the ITS group and better than the simulation. Alternating back and forth between physical pages and the computer screen was also seen as a downside. However, none of these criticisms was repeated at an alarming rate.

The majority of simulation participants indicated that their system was clear in giving instructions. Unlike book-based instructions, clicking through the simulations and seeing the steps that participants would need to do to succeed (vicarious learning) was an advantage. They also liked just-in-time help, a pop-up window that provided instruction in case of error. Their system controlled the learning environment rather than the learners because the option to proceed to the next step depended on the success of current step. This guaranteed successful completion of the task, but it minimized the amount of creativity. About 95% of the participants felt like simulations: 1) did most of the work, 2) did not teach the concepts

well enough, 3) did not support multiple paths through the software, and 4) presented tasks and steps in a fixed linearly fashion. About a half of the participants indicated that simulations: 1) did not offer the same level of flexibility to refer back to previously presented materials as the printed material allowed users to skim instructions, and 2) did not provide option to speed up or slow down the transition time between slides to accommodate speed of faster or slower readers.

Similarly, the ITS group enjoyed a high level of creativity because their system supported multiple paths to the solution. The most appreciated feature of ITS was the hint function. Unlike the book and the simulation, ITS participants could ask for a hint which presented different levels of guidance. There was also just-in-time (JIT) help that displayed a message when a mistake was made. Participants could alternate the sequence of tasks, but a sequential order was encouraged. However, almost all participants expressed that the ITS did not offer an exhaustive list of paths to the solution. It forced users to successfully complete a given sub-goal in order to proceed to the next sub-goal. Some reported that it was too sensitive in displaying error messages. Other complaints were technical issues including frequent application freeze.

The interesting results come from the ITS's ability to track learners' actions. It was observed that participants visited all additional content (What's the Point and Concepts information) for the first few tasks. As they learned how the system worked, different participants selected different paths. Some participants relied only on What's the Point direction that described the goal of the exercise and ignored the Concepts link. This reaffirms the evidence that learners who have higher control tend to over-estimate their ability and skip important instructional components (Piccoli, et al., 2001). Also, many participants frequently

asked for hints as they did not have step-by-step instructions to guide them. Fortunately, the different levels of requested help not only provided guidance leading to the solution, but also offered enrichment with conceptual explanation. Arguably, the help provided by ITS was comparable to the step-by-step instructions supported by book. A higher number of paths to the solution would have potentially resulted in a more favorable result on task performance than the book and simulation. It would also have minimized the frequency of requesting help which had direct effect on the time spent on the training tasks. Minimizing technical issues (e.g., application freeze) and better ITS curriculum presentation (e.g., unconditional transition to the next task and offering the option to follow step-by-step instruction) could have minimized frustration, which had direct effect on system usability perception and concept acquisition scores. Findings presented above were reported in the 2008-proceedings of Society for Information Technology and Teacher Education (Hategekimana, et al. 2008).

CHAPTER 5. CONCLUSION

5.1 Summary

The results of this empirical study complement other findings in the literature on the use of self-paced training approaches. Earlier studies supported a positive relationship between learner's ability to succeed indicators and training outcomes. This study examined different aspects of self-paced training using three different training approaches: Book-based, software simulation, and intelligent tutoring systems (ITS). The main research questions focused on how the ITS-based Paint.NET training compares with traditional training methods: book-based and computer simulation-based training. The other research question was about how individual characteristics of the learner (spatial ability and self-efficacy) moderate the relationships between training modality and training outcomes.

Participants were randomly assigned to different training modes, received similar training, and completed the same survey questions and test performance measures. Statistical analysis was conducted to challenge three research hypotheses: 1) there is a significant relationship between training condition and training outcomes, 2) there is a significant relationship between learner's ability to succeed indicators and training outcomes, and 2) learner's ability to succeed indicators significantly moderates the relationship between training condition and training outcomes.

Data analyses showed evidence that training condition significantly correlated with all training outcomes (total training time, task-performance scores, system usability rating) except conceptual scores. When considering total time it takes to complete training tasks simulation group performed better than participants in book-based and ITS group.

Participants in book and simulation groups performed equally well on task performance measures and system usability rating. At the same time they performed better than those in the ITS group. In addition, the interaction between training mode and spatial ability significantly affected the system usability rating. The same interaction effect was identified for general self-efficacy. Within the software simulation mode high-spatial learners take significantly less total training time than low-spatial learners. Similarly, within the ITS mode learners with higher computer self-efficacy (CSE) do better on task performance measures than learners with low CSE. It is also reported that within simulation mode learners with high general self-efficacy (GSE) have better perception of their system usability than these with low general self-efficacy. Participants in book-based training condition showed higher scores on both task performance and system usability perception, while better times to complete training were recorded for the simulation approach. However, there was no significant indication that concept acquisition scores correlate with training mode. Two significant interaction effects on usability perception were identified. Moderating variables, spatial ability, and general self-efficacy, along with training mode significantly influenced usability perception rating. All other interaction effects were not significant.

Qualitative results showed that participants appreciated training that offer freedom to explore on their own while discovering rather than following every step outlined in the book. However, it was observed that participants who skimmed and/or skipped background information experienced frustration which had a direct effect on system usability perception and concept acquisition scores. This finding was consistent with research evidence of Piccoli et al. (2001) that learners who have higher control of their learning environment tend to over-estimate their ability. All participants, independently of training mode, expressed a need to

support an exhaustive list of paths to the solution rather than being forced to one way of doing things when there could be other alternatives. The biggest advantage of the intelligent tutoring system over the other systems observed in this study is a high level of interactivity (supporting multiple paths) and different levels of timely guidance. The evidence presented in this study show that ITS used for Paint.NET training needs system usability improvement.

5.2 Lessons learned

Overall, results indicated that all four learner's ability to succeed indicators (SA, GSE, CSE, SSE) significantly correlated with total training time and system usability measures independently of high versus low levels of LASI. These findings suggest that there are other factors to consider beyond the simple assumption that a system that is easy to learn and use has the potential to minimize total time it takes to complete a task. It was observed that participants in the simulation group used shorter time to complete tasks, but their system was not always the one given favorable usability rating by the user. Other factors to consider beyond the easy to learn and time include meeting specific learner goals, such as concept acquisition, task performance, and appeal, to name a few. The other lessons learned include:

- Learners who are primarily interested in the total time it takes to complete training are better off in simulation mode.
- Training mode is a good predictor of task performance score. Learners who are primarily concerned with practical application succeed better in the ITS mode.
- Level of CSE is an indicator of which training mode will help a learner to succeed on task performance measure. Like learners with low CSE are better off in book-based training; learners with high CSE succeed better in the ITS group.

- Within simulation, group level of spatial ability is a good predictor of how much time it takes to complete image manipulation task. High-spatial learners do better than low-spatial learners.

It is expected that the information from this study will make it possible for ITS designers to improve the technical component of the system, presentation of the curriculum, and system usability. Refinements should lead to improved measures on this educational system's effectiveness; otherwise alternatively examined systems would be the best option.

5.3 Limitations and direction for future research

For convenience, the participants in the study were undergraduate college students. Almost all participants were in the 18-24 age range, a young audience that was acquainted with technology at younger age. Older adult learners in the workplace may react differently to the Paint.NET training with ITS enhancement. This sample was not representative of corporate personnel, who are among potential users of the ITS for software training. Future research should be extended to include participants from the workplace.

A major limitation of this study was the sample sizes. There was a total of 75 participants equally distributed between three groups. Their training and testing records were verified and data with known anomalies were discarded. This considerably reduced sample sizes, particularly within groups where high versus low individual characteristic to succeed indicators were compared. Even though non-parametric tests agreed with findings of parametric analysis used in this study, future research with larger sample sizes would exhibit more conclusive findings.

It was concluded that ITS needs improvement in order to match or exceed the experiences offered by book and simulation. Future empirical research is warranted to evaluate the effectiveness of the improved ITS for off-the-shelf software training. The ITS approach used in this study is similar to an approach ClearSighted Inc. used with the VaNTH web-based authoring tool for constructing online assignments. Preliminary workshop results with eight participants showed that VaNTH with ITS enhancement is a promising approach for self-guided learning, but there are some areas where improvement still need to be realized (Rosalli, et al. 2008). Future ITS research may be extended to include other applications such as music.

Despite no statistical significance of the effect of spatial ability on task performance score, data showed that high-spatial learners performed somewhat better than low-spatial learners independent of the training mode. Similarly, this study showed an opportunity for future investigation to challenge the assumption that high computer self-efficacy (CSE) learners will more likely perform better in ITS group while those with low CSE may perform better in book-based training. It is urged to run a follow-up spatial ability and CSE studies to investigate whether the general trends that appeared in this study could be replicated.

In addition, it was reported that book participants sometimes skipped background instruction and ITS participants sometime ignored the concepts links. This level of learner control of instruction somewhat contributed to their frustration throughout task completion. Future research may explore patterns of learners who skip instructional components and how to help them best.

REFERENCES

- Abdal-Haqq, I. (1998). Constructivism in teacher education: Considerations for those who would link practice to theory [Electronic version]. *ERIC Clearinghouse on Teaching and Teacher Education*, Washington DC.
- Abdullah, M.H. (2001). Self-directed learning [Electronic version]. *ERIC Clearinghouse on Reading, English & Communication, Digest #169*.
- Ainsworth, S. & Fleming, P. (2006). Evaluating authoring tools for teachers as instructional designers. *Computers in Human Behavior*, 22, 131-148.
- Aleven, V. et Al. (2006). Toward meta-cognitive Tutoring: A Model of Help-Seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence in Education*, 16, 101-130.
- Anderson, J.R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.
- Anderson, J.R. et al. (1995). Cognitive Tutors: Lessons Learned. *The Journal of the Learning Science*, 4(2), 167-207.
- Anderson, J. R. & Schunn, C. D. (2000). Implications of the ACT-R learning theory: No magic bullets. In R. Glaser, (Ed.), *Advances in instructional psychology: Educational design and cognitive science* (Volume 5), pp. 1-34. Mahwah, NJ: Lawrence Erlbaum Associates.
- Anderson, J.R. & Gluck, K. (2001). *What role do cognitive architectures play in intelligent tutoring systems? Cognition & Instruction: Twenty-five years of progress*. D. K. S. M. Carver, Erlbaum: 227-262.

- Arruarte, A. et al. (1997). The IRIS Shell:”How to Build ITSs from Pedagogical and Design Requisites. *International Journal of Artificial Intelligence in Education*, 8, 341-381.
- Awang-Hashim, R., O’Neil, H.F. & Hocevar, D. (2002). Ethnicity, Effort, Self-Efficacy, Worry, and Statistics Achievement in Malaysia: A Construct Validation of the State-Trait Motivation Model. *Educational Assessment*, 8(4), 341-364.
- Baldwin, T.T. (1992). “Effects of alternative modeling strategies on outcomes of interpersonal skills training. *Journal of Applied Psychology*, 77, 147-154.
- Bandura, A. (1986). *Social foundations of thought and action: Social cognitive theory*. Englewood Cliffs, NJ: Prentice-Hall.
- Bandura, A. (1989). Human agency in social cognitive theory. *American Psychologist*, 44, 1175-1184.
- Bartley, S.J. & Golek, J.H. (2004). Evaluating the Cost Effectiveness of Online and Face-to-Face Instruction. *Educational Technology and Society*, 7(4), 165-175.
- Beck, J., Stern, M. & Haugsjaa, E. (2004). *Application of AI in Education*. Retrieved on 19th Feb. 2008, from <http://www1.acm.org/crossroads/xrds3-1/aied.html>
- Blair, D.V., O’Noel, H.F & Price, D.J. (1999). "Effects of expertise on state self-efficacy and state worry during a computer-based certification test." *Computers in Human Behavior*, 15(3-4): 511-530.
- Bransford, J.D., Brown, L.A. & Cocking, R.R. (2000). *How People Learn: Brain, Mind, Experience, and School*. Washington, D.C National Academy Press.
- Brooke, J. (1996). A “quick and dirty” usability scale. In P.W. Jordan, B. Thomas, B.A. Weerdmeester & A . L. McClelland (Eds.). *Usability Evaluation in Industry*. London: Taylor and Francis.

- Bosscher, R.J. & Smit, J.H. (1998). Confirmatory factor analysis of the general self-efficacy scale. *Behavior Research and Therapy*, 36, 339-343.
- Carnegie Learning, Inc. (2007). Executive Summary. Retrieved on December 10, 2007, from http://www.carnegielearning.com/web_docs/ExecutiveSummary-062007.pdf
- Carr, N.G. (2000). *Does IT matter?: Information technology and the corrosion of competitive advantage*. Boston, MA; Harvard Business School Press.
- Clouse, S.F. & Evans, G.E. (2003). Graduate Business Students Performance with Synchronous and Asynchronous Interaction e-Learning Methods. *Decision Sciences Journal of Innovative Education*, 1(2), 181-202.
- Compeau, D. & Higgins, C. (1995). Computer self-efficacy: Development of measure and initial test. *MIS Quarterly*, 19(2), 189-211.
- Corbett, A.T. & Anderson, J.R. (2001). *Proceedings of ACM CHI 2001 Conference on Human Factors in Computing Systems Locus of Feedback Control in Computer-Based Tutoring: Impact on Learning Rate, Achievement and Attitudes*. In Jacko, J., Sears, A., Beaudouin-Lafon, M. and Jacob, R. (Eds.), 245-252. New York: ACM Press.
- Corbett, A.T. & Anderson, J.R. (1997). Intelligent Tutoring Systems: Handbook of Human Computer Interaction. Second Ed. In M. Helander, T.K. Landauer, P. Prabhu (Eds), *Elsevier Science B.V*, Chapter 37.
- Downey, J.P & McMurtrey, M. (2006). Introducing task-based general computer self-efficacy: An empirical comparison of three general self-efficacy instruments. *Interacting with Computers*, 19(3), 382-396.

Dede, C. J. (1987). Empowering environments, hypermedia and microworlds. *Computing Teacher*, 15(3), 20-24.

Ekstrom, R.J., French, & Harman, H. (1976). *Kit of Factor-Referenced Cognitive Tests*. Princeton: NJ, Educational Testing Service.

Feedman, R., Ali, S.S. & McRoy, S.W. (2000). What is an intelligent tutoring system? *Intelligence*, 11(3), 15-16.

French Ministry of Education (2006): The common base of knowledge and skills. Decree date 11 July 2006, presented by Gilles de Robien, CNDP, Paris.

Forcier, R. & Descy, D. (2005). *The computer as an educational tool: Productivity and problem solving*. Upper Saddle River: NJ, Pearson Education, Inc..

Gheorghe, M. (2006). Professional Skill Development in Higher Education for Knowledge-Based Regions. Presented in the EUA 2006 Autumn Conference BRNO, 19-21 October 2006.

Gilbertr, S., Blessing, S., Ourada, S. & Ritter, S. (2007). *Proceedings of the 13th International Conference on Artificial Intelligent in Education: Lowering the Bar for Creating Model-Tracing Intelligent Tutoring Systems*.

Gist, M.E., Schwoerer, C.E. & Rosen, B. (1989). Effects of alternative methods on self-efficacy and performance in computer software training. *Journal of Applied Psychology*, 74, 884-891.

Gustafson, K.L. & Maribe, B.R. (1997). Revisioning Models of Instructional Development. *Educational Technology Research and Development*, 45(3), 73-89.

Hafner, K. (2004). Software Tutors Offer Help and Customized Hints. *The New York Times*.

Retrieved on 20th Feb. 2008, from

<http://www.nytimes.com/2004/09/16/technology/circuits/16tuto.html>

Harrison, E.G. (2006). Working with faculty toward universally designed instruction: The process of dynamic course design. *Journal of Postsecondary Education and Disability*, 19(2), 152-162.

Harrison, N. (1999). *How to design self-directed and distance learning: a guide for creators of web-based training, computer-based training, and self-study materials*. New York: NY, McGraw-Hill.

Hategekimana, P.C., Gilbert, S. Blessing, S. (2008). *Proceedings of Society for Information Technology and Teacher Education (SITE) Effectiveness of Using an Intelligent Tutoring System to Train Users on Off-the-Shelf Software*, Mar. 4-7, 2008, p. 414.

Hays, R.T. (2005). The effectiveness of instructional games: A literature review and discussion. Technical Report, Naval Air Warfare Center Training Systems Division. Orland, FL.

Hooper, E.J. & R.A. Thomas, R.A. (1991). Simulations: An opportunity we are missing. *Journal of Research on Computing Education*, 23(4), 495-513.

Jackiw, R.N. & Finzer, W.F. (1993). *The geometer's sketchpad: Programming by geometry*. In a. Cypher (Ed.) *Watch what I do: Programming by demonstration* (pp. 292-307), Cambridge: MA, MIT Press.

Jacko, J.A. & Sears, A. (2003). *The Human-computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*. Mahwah: NJ, Lawrence Erlbaum Associates.

- Kemp, J.E., Morrison, G.R. & Ross, S.M. (2004). *Designing effective instruction*. Fourth Ed., Hoboken: NJ, J. Wiley & Sons.
- Kirkpatrick, D. (1998). *Evaluation Training Programs: The Four Levels*. San Francisco: CA, Berrett-Koehler.
- Koedinger, K.R. et al. (2004). *Proceedings of the Seventh International Conference of Intelligent Tutoring Systems, Opening the Door to Non-Programmers: Authoring Intelligent Tutor Behavior by Demonstration*. Maceio, Brazil.
- Koedinger, K.R., Anderson, J.R., Hadley, W.H., & Mark, M.A (1997). Intelligent Tutoring goes to school in the big city. *Journal of Artificial Intelligence*, 8,30-43.
- Kyllonen, P.C., Lohman, D.F., & Snow, R. E. (1984). Effects of aptitudes, strategy training, and task facets on spatial task performance. *Journal of Educational Psychology*, 76(1), 130-145.
- Lunce, L.M. (2004). Computer Simulation in Distance Education. *International Journal of Instructional Technology and Distance Learning*, 1(10), 29-40.
- Mansour, B. E. (2006). Challenges and Solutions in Offering Distance Education Programs: A case study of an HRD program. *International Journal of Instructional Technology and Distance Learning*, 3(11), 33-39.
- Marshall, D.V. (1988). *Proceedings of the 9th International Conference on User Modeling CAL/CBR: the great debate*. Lund: Sweden, Charlwell-Bratt Ltd.
- Matsuda, N. & VanLehn, K. (2003). Modeling Hinting Strategies for Geometry Theorem Proving., Pittsburg, PA, 373-377.
- McCain, M. (2007). *E-Learning: Are We in Transition or Are we Stuck? An update Five Years on*. Center for Workforce Success and National Association of Manufacturers,

retrieved on 3rd Apr. 2008, from

<http://www.techvision21.com/team/mmcainarticles.html>

- McQuiggan, S.W., Mott, B.W. & Lester, J.C. (2008). Modeling self-efficacy in intelligent tutoring systems: An inductive approach. *User Modeling and User-Adapted Interaction*, 18(1-2).
- Molich, R., Ede, M., Kaasgaard, K., & Karyukin, B. (2004). Comparative usability evaluation. *Behaviour & Information Technology*, 23(1), 65-74.
- Mulkey, J.R. & O'Noel, H.F. (1999). The effects of test item format on self-efficacy and worry during a high-stakes computer-based certification examination. *Computers in Human Behavior*, 15(3-5), 495-509.
- Murray, T. S., Blessing, S. & Ainsworth, S. (2003). *Authoring Tools for Advanced Technology Learning Environments: Toward Cost-Effective Adaptive, Interactive and Intelligent Educational Software*. Dordrecht, The Netherlands Kluwer Academic Publishers.
- Nielson, J. (1993). *Usability Engineering*. London: Academic Press.
- Ong, J. & Ramachandran, S. (2000). Intelligent Tutoring System: The What and the How. ASTD's Source for E-Learning, Retrieved on 19th Feb. 2008, from <http://www.learningcircuits.org/2000/feb2000/ong.htm>
- Piccoli, G, Ahmad, R. & Ives, B. (2001). Web-based virtual learning environment: A research framework and preliminary assessment of effectiveness in basic IT skills training. *MIS Quarterly*, 25(4), 401-426.
- Prensky, M. (2001). Digital natives, digital immigrants. *ON the Horizon*, 9(5), 1-9.

- Ritter, S & Blessing, S.B. (1998). Authoring Tools for Component-Based Learning Environments. *The Journal of the Learning Sciences*, 7(1), 107-132.
- Rivera, R.J. & Paradise, A. (2006). *ASTD's Annual Review of Trends in Workplace Learning and Performance*. Alexandria: VA, American Society for Training and Development.
- Roger, E. (2003). *Diffusion of Innovations*. 5th Ed., New York: NY, The Free Press.
- Rosalli, R., Gilbert, S. Howard, L., Blessing, S.B., Raut, A. & Pandian, P. (2008). *Proceedings of the American Society for Engineering Education (ASEE) Annual Conference: Integration of an intelligent tutoring system with a web-based authoring system to develop online homework assignments with formative feedback.*, Pittsburgh, PA.
- Rowe, H.A.H. (1991). *Intelligence: reconceptualization and measurement*. Hillsdale: NJ, Lawrence Erlbaum Associates.
- Sauro, J. & Kindlund, E. (2005). *In proceedings of the 14th Annual Conference of Usability Professional Association Making sense of usability metrics: Usability and Six Sigma*. (UPA), Montreal, Canada.
- Sherer, M., et al. (1982). *The self-efficacy scale: Construction and Validation Psychological Reports*, 51, 663-671.
- Seels, B. Glasgow, Z. (1998). *Making instructional design decisions*. Upper Saddle River: NJ, Prentice Hall.
- Simon, J.S., & Werner, M.J. (1996). Computer Training Through Modeling, Self-Pace, and Instructional Approaches: A Field Experiment. *Journal of Applied Psychology*, 81(6): 648-659.

- Sleeman, D.H. & Brown, J.S. (1982). *Intelligent Tutoring Systems*. New York:NY Academic Press.
- Smith, P.L & Ragan, T.J. (1999). *Instructional Design. Second Ed.* New York:NY, John Wiley & Sons, Inc. (pp.1-12).
- Song, L. & Hill, J.R. (2007). A Conceptual Model for Understanding Self-Directed Learning in Online Environments. *Journal of Interactive Online Learning*, 6(1), 27-42.
- Suppes, P., & Macken, E. (1978). The historical path from research and development to operation use of CAI. *Educational Technology*, 18(4), 9-11.
- Tucker, B. (1997). *Handbook of Technology-Based Training: Forum for Technology in Training*. Hampshire: England, Gower Publishing Limited.
- VanMeer, E. (2003). PLATO: From Computer-Based Education to Corporate Social Responsibility. *The Charles Babbage Institute for the History of Information Technology, (Iterations: An Interdisciplinary Journal of Software History)*. Retrieved on Oct. 26th, 2007, from: <http://www.cbi.umn.edu/iterations/vanmeer.pdf>
- Yi, M.Y. & Im, K.S. (2004). Predicting computer task performance: Personal goal and self-efficacy. *Journal of Organizational and End-User Computing*, 16(2), 20-37.
- Zevenbergen, R. (2007). Digital natives come to preschool: implications for early childhood practice. *Contemporary Issues in Early Childhood*, 8(1), 19-29.

APPENDICES

Appendix 1: Opening script: Effectiveness of Paint.NET training program (pre-training, day one)

Clearsighted, Inc. is a startup company in the ISU Research Park. We aim to train people to use software more effectively than current software training methods by using Intelligent Tutoring Systems. As a prototype, we have developed a system for training people how to use software called Paint.NET. It's an image-editing tool.

The research in which you're participating will help us evaluate the effectiveness of this training system. The study has 4 components:

1. First, the study begins with a short questionnaire that will give us a sense of your proficiency with computers and image-editing software as well as your skill to visualize, analyze and comprehend imaginative objects, and your demographic information. It should take you about 5 minutes to complete the questionnaire. Your responses will remain anonymous and confidential. Also, because this study is approved by the ISU Human Subjects Committee of the Institutional Review Board, you'll have the opportunity to sign an informed consent form acknowledging that you know there are minimal risks to you if you participate.
2. Second, we will give you training on using Paint.NET application to edit images. You will work through a series of image editing tasks with guidance on what to do. If needed, a help document is available as hard copy and within Paint.NET. You will not receive any extra help except in technical issues such as the computer freezing, images being missing, etc. This training on tasks will take you about 1 hour. As you work, do the tasks as fast as you can while being careful to follow directions. After training, you will be asked to complete a survey that will give feedback about what you think about the our training program.
3. Third, when you return for the second session, you will be given an evaluation to measure the effectiveness of the training you received. The evaluation will include:
 - ⇒ A series of multiple-choice questions to measure your conceptual understanding of concepts from training. There is text in the training materials called "What's the Point?" Reading this text will help you on these questions. (10 min)
 - ⇒ Paint.NET tasks requiring the skills taught during training, but without the help that you had during training. (30 min)
4. If you do all the above steps, you will be entered in a drawing to win one of several iPod Nanos and will receive it at the end of the second session. In addition, you will get a \$30 gift card (15min)
5. Finally, in order to keep your responses anonymous and confidential, we ask you to use the last 4 digits of your ISU ID card as an ID number in our system. The iPod sweepstakes will also rely on this number, so please use it consistently throughout the sessions.

Appendix 2: Informed consent document (pre-training, day one)

INFORMED CONSENT DOCUMENT

Title of Study: “Effectiveness of Using an Intelligent Tutoring System to Train Users on Paint.NET”

Investigators:

Stephen Gilbert, Ph.D.
515-296-7800,
stephen@clearsighted.net

Steve Blessing, Ph.D.
(813) 253-3333 x3461,
sblessing@ut.edu

Claver Hategekimana
515-231-2173
claver@iastate.edu

This is a research study. Please take your time in deciding if you would like to participate. Please feel free to ask questions at any time.

INTRODUCTION

The purpose of this study is to evaluate the effectiveness of using an intelligent tutoring system (ITS) to train users on the application Paint.NET as opposed to using tradition software training techniques. You are being invited to participate in this study because you have at least basic skills to operate and learn how to use a new computer application.

DESCRIPTION OF PROCEDURES

If you agree to participate in this study, your participation will last for two days: one sessions of 90 minutes and a later second session of 60 minutes. During the first session, you will learn how to manipulate images using Paint.NET software. During the second session, you will be asked to perform image manipulation tasks, answer conceptual questions about image editing, and complete a survey about key elements of this learning environment.

Note: On the survey, you may skip any question that you do not wish to answer or that makes you feel uncomfortable. We will record your actions on the computer screen while you work and may record your voice as you answer questions.

RISKS

There is no known potential for physical or social harm while participating in this study.

BENEFITS

If you decide to participate in this study you may benefit by learning new skills with image manipulation. It is hoped that the information gained in this study will help developers of Intelligent Tutoring Systems to create effective systems that meet personal preferences of users.

COSTS AND COMPENSATION

You will not have any costs from participating in this study, aside from your time. At the end of the study, you will be entered in a drawing to win one of several iPods.

PARTICIPANT RIGHTS

Your participation in this study is completely voluntary and you may refuse to participate or leave the study at any time. If you decide to not participate in the study or leave the study early, it will not result in any penalty or loss of benefits to which you are otherwise entitled. The principal investigator may terminate participation of a subject or the project entirely without regard to the subject's consent. In the event of question or difficulties of any kind during or following participation, the subject may contact the Principal Investigator as indicated in the Questions or Problems section.

CONFIDENTIALITY

Records identifying participants will be kept confidential to the extent permitted by applicable laws and regulations and will not be made publicly available. However, federal government regulatory agencies and the Institutional Review Board (a committee that reviews and approves human subject research studies) may inspect and/or copy your records for quality assurance and data analysis. These records may contain private information.

To ensure confidentiality to the extent permitted by law, the following measures will be taken: Only the primary investigator(s) will have access to the data files, and participants will be identified only by numbers in these files. If the results are published, your identity will remain anonymous.

QUESTIONS OR PROBLEMS

You are encouraged to ask questions at any time during this study.

- For further information about the study contact:
Stephen Gilbert, Ph.D. Phone: (515) 296-7800, Email: stephen@clearsighted.net
- If you have any questions about the rights of research subjects or research-related injury, please contact the IRB Administrator, (515) 294-4566, jcs1959@iastate.edu, or Diane Ament, Director, Office of Research Assurances (515) 294-3115, dament@iastate.edu.

PARTICIPANT SIGNATURE

Your signature indicates that you voluntarily agree to participate in this study, that the study has been explained to you, that you have been given the time to read the document and that your questions have been satisfactorily answered. You will receive a copy of the written informed consent prior to your participation in the study.

Participant's Name (printed) _____

(Participant's Signature)

(Date)

INVESTIGATOR STATEMENT

I certify that the participant has been given adequate time to read and learn about the study and all of their questions have been answered. It is my opinion that the participant understands the purpose, risks, benefits and the procedures that will be followed in this study and has voluntarily agreed to participate.

(Signature of Person Obtaining
Informed Consent)

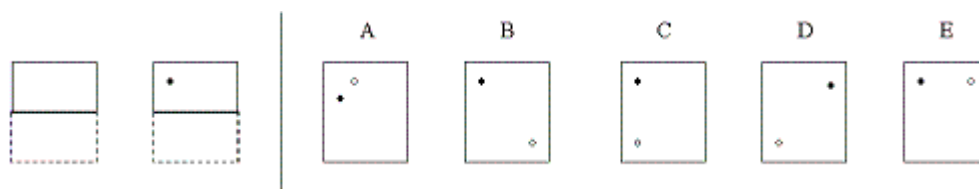
(Date)

Appendix 3: Spatial ability survey: Paper folding

(pre-training, on day one)

In this survey you are to image the folding and unfolding of pieces of a paper. In each problem in the survey there are some figures drawn at the left of a vertical line and there are others drawn at the right of the line.

Sample scenario: As shown below, the figures at the left represent a square piece of paper being folded and the last of these figures has one or two small circles drawn on it to show where the paper has been punched. Each hole is punched through all the thickness of paper at that point. One of the five figures at the right of the vertical line shows where the holes will be when the paper is completely unfolded. You are to decide which one of these figures is correct.



Question: Which of the 5 results above is the correct result of the folding and hole-punch? Indicate your answer by circling the appropriate letter associated with the correct figure above. (Note: In this problem only one hole was punched in the folded paper.)

Answer: The correct answer to the sample problem above is C. The figures below show how the paper was folded and why C is the correct answer.



In these problems all of the folds that are made are shown in the figures at the left of the line, and the paper is not turned or moved in any way except to make the folds shown in the figures. Remember, the answer is the figure that shows the positions of the holes when the paper is completely unfolded.

This paper folding survey measures one type of spatial ability that may or may not be related to your success in Paint.NET. Your score on this survey will be the number marked correctly minus a fraction of the number marked incorrectly. Therefore, it will not be useful to guess about the answers.

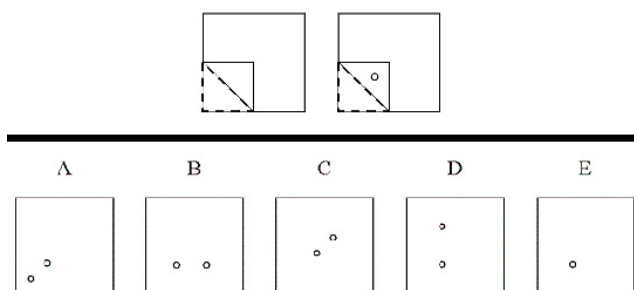
(Please don't move to the next page until you are told to so.)

For each of the next 10 questions, there are some figures drawn first, and there are five others drawn below them. The figures on top represent a square piece of paper being folded, and the last of these figures has one or two small circles drawn on it to show where the paper has been punched. Each hole is punched through all the thickness of paper at that point.

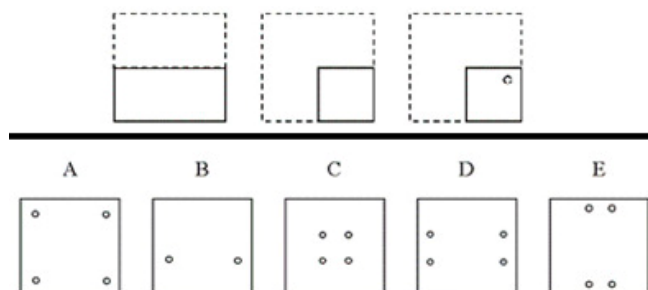
One of the five labeled figures below shows where the holes will be when the paper is completely unfolded. You are to decide which one of these figures is correct and circle it. *You have 3 minutes to complete it.*

What is your four-digit ID number?

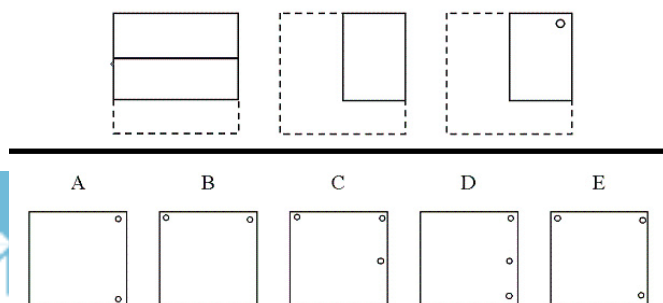
Q1: Below is a figure of folded and punched paper, circle the correct unfolded paper presented below the line.



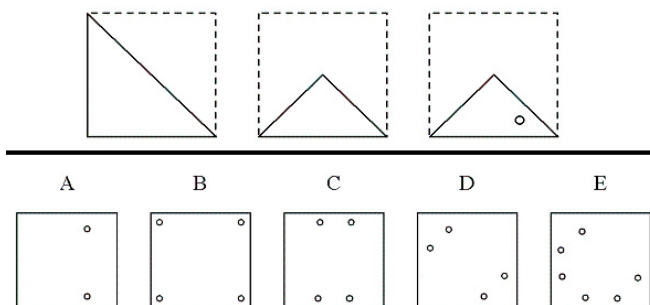
Q2: Folded and punched paper, circle current unfolded paper below the line



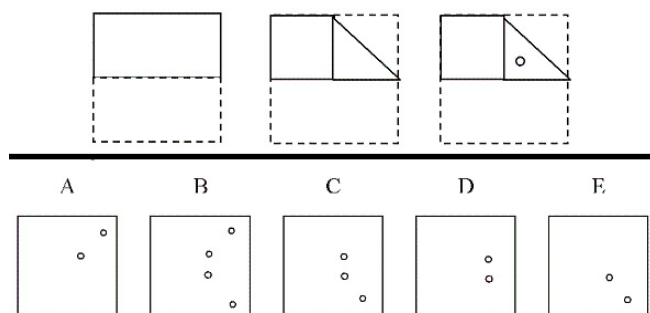
Q3: Folded and punched paper, circle current unfolded paper below the line



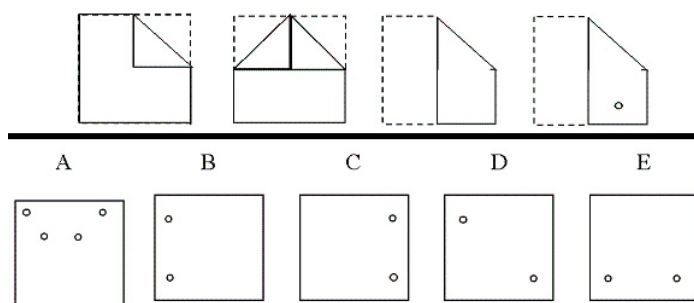
Q4: Folded and punched paper, circle current unfolded paper below the line



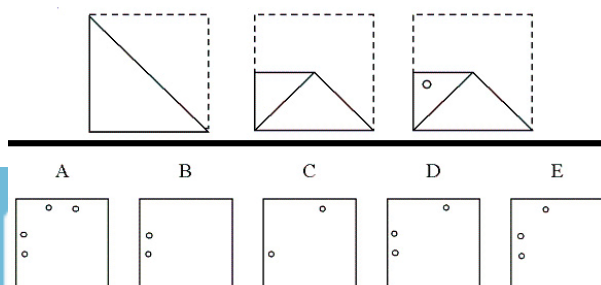
Q5: Folded and punched paper, circle current unfolded paper below the line



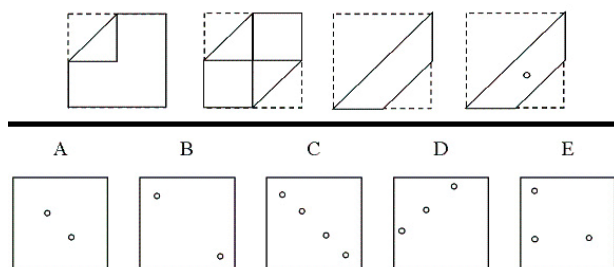
Q6: Folded and punched paper, circle current unfolded paper below the line



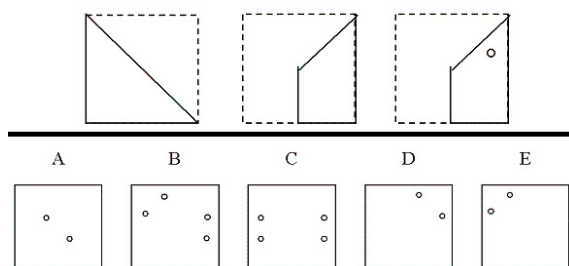
Q7: Folded and punched paper, circle current unfolded paper below the line



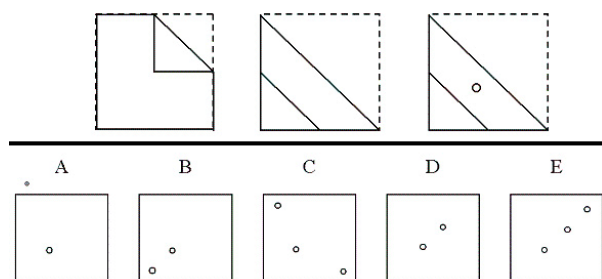
Q8: Folded and punched paper, circle current unfolded paper below the line



Q9: Folded and punched paper, circle current unfolded paper below the line



10: Folded and punched paper, circle current unfolded paper below the line



Appendix 4: Demographic questionnaire (pre-training, on day one)

What is your four-digit ID number?

- | | |
|---|--|
| <p>1. Please select your sex</p> <p>A) Female</p> <p>B) Male</p> <p>2. Which age range do you fit into?</p> <p>A) Less than 18</p> <p>B) 18-24</p> <p>C) 25-35</p> <p>D) 36 or more</p> <p>3. What is your highest level of education?</p> <p>A) 1st year undergraduate</p> <p>B) 2nd year undergraduate</p> <p>C) 3rd year undergraduate</p> <p>D) 4th year undergraduate</p> <p>E) Master's level</p> <p>F) Doctoral level</p> <p>4. What is your major?</p> <p>•</p> <p>5. How often do you use a computer?</p> <p>A) Every day</p> <p>B) Every other day</p> <p>C) Twice a week</p> <p>D) Once a week</p> | <p>6. How would you rate your level of computer proficiency?</p> <p>A) Novice user</p> <p>B) Average user</p> <p>C) Advanced user</p> <p>7. Do you have any basic experience with Photoshop?</p> <p>A) Yes</p> <p>B) No</p> <p>8. Do you have any basic experience with Paint.NET?</p> <p>A) Yes</p> <p>B) No</p> <p>9. Do you have any basic experience with Microsoft Paint?</p> <p>A) Yes</p> <p>B) No</p> <p>10. Have you ever used intelligent tutoring system before?</p> <p>A) Yes</p> <p>B) No</p> <p>If yes, which one?</p> |
|---|--|

Appendix 5: General self-efficacy survey (pre-training, on day one)

What is your four-digit ID number?

Please indicate the extent to which you agree or disagree with the following statements.

	Strongly Disagree <--		-->Strongly Agree		
1. If I can't do a job the first time, I keep trying until I can.					
	1	2	3	4	5
2. When I set important goals for myself, I rarely achieve them.					
	1	2	3	4	5
3. I often give up on things before completing them.					
	1	2	3	4	5
4. When I have something unpleasant to do, I stick to it until I finish it.					
	1	2	3	4	5
5. When I decide to do something, I go right to work on it.					
	1	2	3	4	5
6. When trying to learn something new, I soon give up if I am not initially successful.					
	1	2	3	4	5
7. When unexpected problems occur, I don't handle them well.					
	1	2	3	4	5
8. I avoid trying to learn new things when they look too difficult.					
	1	2	3	4	5
9. Failure just makes me try harder.					
	1	2	3	4	5
10. I feel insecure about my ability to do things.					
	1	2	3	4	5
11. I give up easily.					
	1	2	3	4	5
12. I do not feel capable of dealing with most problems that come up in life.					
	1	2	3	4	5

Appendix 6: Computer self-efficacy survey (pre-training, on day one)

This part of the questionnaire asks you about your ability to use an unfamiliar piece of software. For the following questions, imagine that you were given a new piece of training software called Paint.Net Training System that was intended to make it easier for you to learn something. You have never used it before.

You are asked to indicate whether you could use this unfamiliar software package under a variety of conditions. For each condition, please rate your confidence level on 5-point scale. Your responses are anonymous and confidential. There are no right or wrong answers.

Not Confident<----->Very Confident
1.....2.....3.....4.....5

What is your four-digit ID number?

I can use a Paint.NET Training System successfully...

1. If there is someone to give me step by step instructions.

	Not Confident <---		-->Very Confident	
	1	2	3	4 5

2. If there is no one around me to tell me what to do as I go.

1	2	3	4	5

3. If I had never used a package like it before.

1	2	3	4	5

4. If I had only the software manuals for reference.

1	2	3	4	5

5. If I had seen someone else using it before trying it myself.

1	2	3	4	5

6. If I could call someone for help if I got stuck.

1	2	3	4	5

7. If someone else helps me get started.

1	2	3	4	5

8. If I had a lot of time to complete the job for which the Paint.Net Training application was provided.

1	2	3	4	5

9. If I had just the built-in help facility for assistance.

1	2	3	4	5

10. If there is someone to show me how to use it first.

1	2	3	4	5

11. If I had used similar Paint.Net Training applications before this one to do the same job.

1	2	3	4	5

Appendix 7: State self-efficacy: Anticipated confidence in the use of Paint.NET (post-training, on day one)

For each of the next 8 statements, you are asked to indicate your anticipated level of likeliness that you will be able to accomplish a task using new software 'Paint.NET'. This anticipated confidence may or may not be related to your success in Paint.NET.

What is your four-digit ID number?

After Paint.NET training I received I feel that during the test I will do well on

Unlikely<--- --->Likely

1. Navigating the menus in Paint.NET

1	2	3	4	5

2. Resizing an image

1	2	3	4	5

3. Selecting object of an image

1	2	3	4	5

4. Changing colors of an image

1	2	3	4	5

5. Choosing a color to paint with

1	2	3	4	5

6. Manipulating layers

1	2	3	4	5

7. Rotating and flipping an image

1	2	3	4	5

8. Adding text to an image

1	2	3	4	5

Appendix 8: Paint.NET training system usability survey

(Post-training, on day one)

Thank you for giving us the opportunity to test a prototype of our training program. Please help us improve our training program by taking a couple of minutes to tell us about your impression of this training. For each of the following statement check which most closely matches your opinions of this Paint.Net training system. Please record your immediate response, rather than thinking about item for a long time.

What is your four-digit ID number?

	Strongly Disagree			Strongly Agree
1. I think that I would like to use this training system frequently.				
	1	2	3	4 5
2. I found this training system unnecessarily complex.				
	1	2	3	4 5
3. I thought this training system was easy to use.				
	1	2	3	4 5
4. I think that I would need the support of a technical person to be able to use this training system.				
	1	2	3	4 5
5. I found the various functions in this training system were well integrated.				
	1	2	3	4 5
6. I thought there was too much inconsistency in this training system.				
	1	2	3	4 5
7. I would imagine that most people would learn to use this training system very quickly.				
	1	2	3	4 5
8. I found this training system very cumbersome to use.				
	1	2	3	4 5
9. I felt very confident using the training system.				
	1	2	3	4 5
10. I need to learn a lot of things before I can get going with this training system.				
	1	2	3	4 5

Open-ended questions:

- A) What are three things you liked about the training you received? (free response)
- B) What are three things you did not like about the training you received? (free response)
- C) Did the training you received help you use Paint.NET better than you would have without it? If yes, How? Otherwise, what do you see to be its weaknesses? (free response)

Appendix 9: Paint.NET concept acquisition test

(Post-training, on day two)

Thank you for giving us the opportunity to test a prototype of our Paint.NET Training software. Please help us by answering the following questions to the best of your knowledge.

What is your four-digit ID number?

1. Why would you reduce the size of an image (decrease the width and height)?
 - E) Smaller images are better for archival purposes.
 - F) Smaller images are better for printing.
 - G) Smaller images are higher quality.
 - H) Smaller images are easier to download from the Internet.

2. Which one of these is true about flipping and rotating this car image?



- A) Flipping and rotating will have the same effect on this car image.
 - B) Rotating will change the size of this car image.
 - C) Rotating the image will make the car face to the left and still be right-sided-up.
 - D) Flipping image will make the car face to the left and still be right-side-up.
3. If I want to change image to grayscale so I can use it in a black and white newspaper, I will:
 - A) Change values of brightness and contrast to 0.
 - B) Set value of saturation of 0.
 - C) Use black and white command.
 - D) B and C are correct.
4. Choose the best statement about the Invert Selection command;
 - A) It selects the currently unselected region of an image and deselects currently selected pixels.
 - B) It allows selecting an entire image.
 - C) It affects the size of image.
 - D) It has the same effect as cropping function.
5. Which option describes the best way to delete the giraffe from an image of a giraffe in front of a solid blue sky?



- A) Use Magic Wand selection tool to select the giraffe and then delete selected region.
 - B) Use Magic Wand selection tool to select solid blue colored region and then delete selected region.
 - C) Use Magic Wand selection tool to select the giraffe, apply invert selection command, and then delete selected region.
 - D) Use Magic Wand selection tool to select the solid blue colored region, apply invert section command, and then delete the selected region.
6. Which one is NOT true about selection in Paint.NET?
- A) Selection defines the editable area of an image.
 - B) The lasso and Magic Wand are types of selection tools.
 - C) The Paint Bucket is a type of selection tool.
 - D) Selection affects the region on the active layer only.
7. You have a black and white image. Which of the following is true?
- A) You can adjust brightness and contrast to add green color.
 - B) Adjusting Hue color property will allow you to add green color.
 - C) Adjusting color properties will NOT allow you to add a green color.
 - D) Adjusting color properties will only allow you to add a yellow color.
8. Choose the best statement about the Color Display.



- A) The color displayed in the primary display icon will be the one used when you use the Paint Bucket tool.
 - B) The color displayed in the secondary display icon will be the one used when you use the Paint Bucket tool.
 - C) Using the Color Picket tool does not affect the color that shows in color display icons.
 - D) Adding color to the image using the Paint Bucket tool will alter color displayed in the primary display icon.
9. The file format with “.PDN” extension:
- A) Results in a smaller file size than “.JPG” file format
 - B) Has the same effect on the file as “.JPG” file format
 - C) Is an indication that there are flattened layers.
 - D) Is the format to use to save information about layers.
10. Which statement is NOT true about layers?
- A) You can rearrange layers so that the top one can be put underneath the others.
 - B) The basic use of layers is to keep objects separate from each other.
 - C) You increase the file size of an image when you flatten multiple layers into one.
 - D) Text that is added on a separate layer can be moved around as many times as you want.

Appendix 10: Paint.NET task performance test (knowledge transfer) (post- training, on day two)

Dear Participant:

Thank you for giving us the opportunity to explore different training methods for Paint.NET. Please help us evaluate how effective our training was by completing the three projects below as best you can.

If you get to a point where you just can't continue on a project, please use Word to make some notes about:

- ◆ Where you got stuck and
- ◆ What steps you would have done next.

Save those notes in the folder for the respective project, e.g. "Test Project 1."

Project 1:

Claudette M.H. came to Iowa for the first time and made a booklet about her experience. She has asked you to create a cover for the booklet, including certain words and colors. She sketched out a desired image that looks something like this below. You can see that she wants to show her American world, her home country, and her thoughts about the experience.

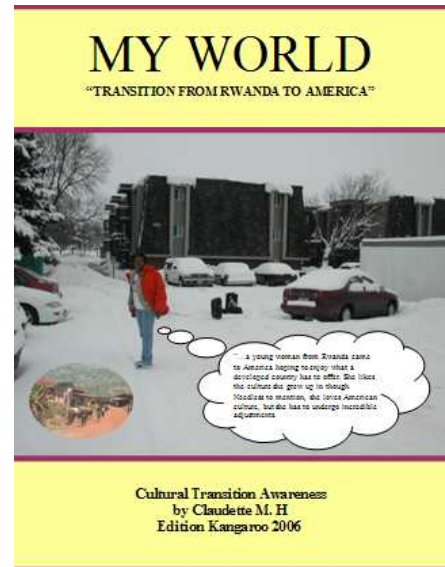
She gives you the image you need to manipulate to complete her project "myworld.jpg". It is in the TestProject01 folder on your computer. The image you will get after completing all necessary work has to resemble the shown target image "MY WORLD."

Your final image should have the following print size specifications: 8.5x11," 150 dpi. For all text use Times New Roman.

Text to use:

1. TRANSITION FROM RWANDA TO AMERICA
2. Cultural Transition Awareness
3. by Claudette M.H
4. Edition Kangaroo

Save your work in the Test Project folder as YourFourDigits-MyWorld.jpg (replace "YourFourDigits" with your numbers, e.g. "5555-MyWorld.jpg")



Project 2:

You decide to create an artistic piece showing a giraffe buffalo in front of a stormy background, but with one bright hope: a tasty sunflower and a buffalo. A botanist friend of yours has supplied you with some sunflowers, already selected out of a photo, and you have some storm clouds, a giraffe, and a buffalo. Combine them to look something like this:



Use giraffe.jpg, buffalo.jpg, storm_clouds.jpg, and sunflowers.tif together (from the Test Project 2 folder).

Specifications for final image:

- 72 dpi and 800 pixels wide for easy web viewing
- Save your image in the Test Project 2 folder as both .PDN and .JPG formats:
 - YourFourDigits-hungry-animals.pdn
 - YourFourDigits-hungry-animals.jpg

Project 3

Adam owns an electronic store BestBargain. He wants to create a promotional advertisement for cell phone toys and laptops selling at his store. To help promote his products he teams up with Knee High Corn, Inc. Supposing, you are one of his employees and he asked you to create a flyer that meets the following specifications.

- The flyer is meant for high quality printing: 5.6 x 7 inches at 300dpi resolution.
- The layout of the intended flyer must look like the target image below. You may use whichever font you like.
- You are given 4 images in Test Project 3. You can edit them as needed for the purpose of the flyer (background.jpg , kids-cellphone.jpg, laptop-accountant.jpg, and laptop-student.jpg).
- The flyer will be printed and displayed at the main entrance of Knee High Corn, Inc.. You need to save the flyer with caption: "Knee High Corn, Inc.." Make sure to save it as a .PDN file type (your file should contain more than 1 layer to showing different images and text).
- You will also have to create and save a black and white version of the flyer. This version is needed for an ad in a newspaper that requires a printout size of 3.5 x 5 inches at 200 dpi.
- In the end save 2 images in the Test Projects 3 folder:
 - YourFourDigits-Knee.pdn, and
 - YourFourDigits-BlackWhite.jpg



Appendix 11: Paint.NET training manual

(Training, on day one)

---See next page---...

Welcome to Training for Paint.NET

Paint.NET is a software application for editing images. It is based on the older Microsoft Paint software, and has features similar to Adobe Photoshop, though Photoshop has many more bells and whistles.

In this training course you will learn how to use the basic features of Paint.NET.

There are three units:

Unit 1: Editing the Whole Image: Resize, Rotate, Flip, Adjust Colors

Unit 2: Enhance the Image: Select, Crop, Add Text

Unit 3: Build an Image: Work with Layers

Each unit has several tasks to complete. After you complete these practice tasks using the guidance provided, you will return another day to do several test tasks to demonstrate what you have learned.

Remember! If you make a mistake in Paint.NET, you can always **undo** what you have done by doing **Ctrl+Z** or **Edit** menu → **Undo**.

Time to get started!

Unit 1: Editing the Whole Image: Resize, Rotate, Flip, Adjust Colors

Concepts to be learned

- Resize by percentage vs. absolute size
- Print vs. pixel resizing
- Resolution
- Zoom
- Rotate
- Flip
- Changing color
- Adding Saturation is like adding tint/color

Skills to be learned

- To resize the image.
- To change orientation of the image.
- To change color of the image.
- To locate and use the zooming tool.

Task 1-1: Resize, Rotate, and Flip the image.

What's the Point?

Your aunt is a farmer. She sent you a photo of one of her sheep. You decide to make a funny card using this image and then share it with your sister. Your sister will probably post it on a website, so a resolution of 72 pixels/dots per inch (dpi) would be appropriate for her image.

Concepts

How big is your image?


Computer images have three different sizes to consider:

- ◆ The **pixel size** in pixels (e.g. 400 by 600 px. on the screen)
- ◆ The **print size** in inches (e.g. 5 x 7 in. on paper)
- ◆ The image **file size** (e.g. 1.2 MB on your hard drive)

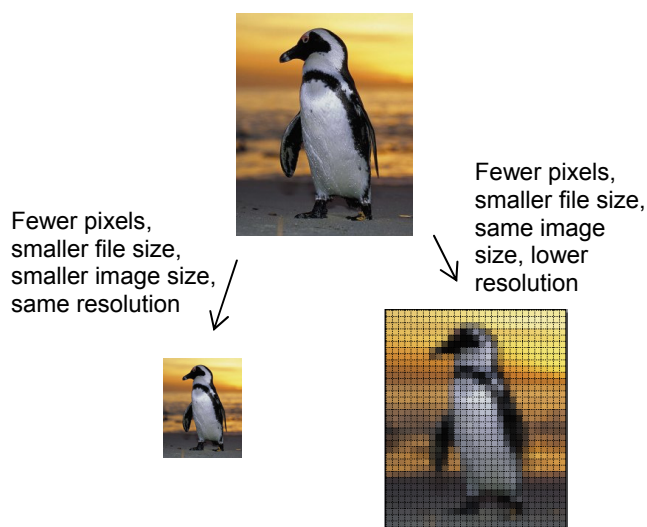
A large image file has pros and cons. A larger image file leads to higher quality prints. However, a large image requires more storage space and takes longer to download. What's often confusing at first is that you can change the print size of an image without changing the pixel size. The key to this idea is resolution. You may have seen images that lose quality as you enlarge them. That's because as you increase the print size, but keep the same number of pixels along the width and height, you're spreading those pixels thinner, and the image doesn't have enough detail in between them. You are decreasing resolution, which you measure in pixels per inch, sometimes called dots per inch (dpi). As an example, the figure illustrates the two things that can happen when you reduce the number of pixels (perhaps to reduce your file size).

When you resize an image, you can specify the **absolute pixels** or the **print size** that you desire. If you decrease the number of pixels (dots of color in the image), then you effectively throw away quality. Sometimes that's ok, if you don't mind making the image smaller or using it only on the web.

Web images don't need as many pixels to look good as printed images do. Another way to say this is that web images are lower **resolution** than printed images. **Resolution** is the number of pixels that print within one inch of space.

Make sure that you don't confuse zooming with resizing. Zooming, using buttons like these: , works like a magnifying glass on your screen; it doesn't change the size of the image.

Two Ways to Decrease File Size



Task 1-1: Rotate and Flip the Sheep; Resize for the Web

What's the Point?

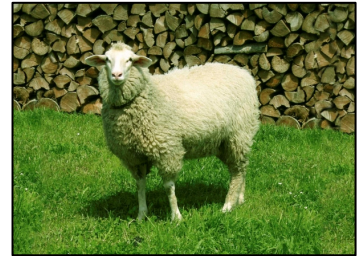
By mistake, the picture of a sheep is on its side, so you will have to rotate the image to look correct. Also, to better fit the layout of the card you're imagining, flip it horizontally so that the sheep faces left, instead of right. You also need to resize and save the image for your sister to put it on a website. The appropriate resolution is 72 dpi. In general, an image may need rotating if it was taken with the camera on its side. You might flip an image for aesthetic reasons—you just think it looks better that way.

Remember! If you make a mistake in Paint.NET, you can always **undo** what you have done by doing **Ctrl+Z** or **Edit** menu → **Undo**.



Your Task

Rotate and flip the image Task1-1-Sheep.jpg so that the sheep looks to the left. Also, set the resolution to 72 dpi for the web and resize the image to 625x500 pixels. Your desired image:



Save it as YourFourDigits-Task1-1-Sheep.jpg.

Steps

1. Use the *File* menu → *Open* to open the image Task1-1-Sheep.jpg from Unit 1 folder.
2. Rotate the image 90 CW (ClockWise) by using the *Image* menu → *Rotate* → 90 CW or Ctrl-H.
3. Flip the image horizontally using *Image* menu → *Flip* → *Horizontal*
4. Open the *Image* menu → *Resize* or click Ctrl-R.
5. Under the *Pixel size* heading set *Resolution* to 72 dpi by typing in 72. (*By absolute size* and *Maintain aspect ratio* must be checked.)
6. Set *Pixel size width* and *height* to 625x500 pixels for easy viewing on the web.
7. Open the *File* menu → *Save As* or click Ctrl-Shift-S.
 - i. Name it **YourFourDigits-Task1-1-Sheep.jpg** and save into the Unit 1 folder.
 - ii. Click OK.

Task 1-2: Adjusting Colors: Increase Intensity

What's the Point?

Your uncle is making an artistic collage of photos. He would like you to make the colors in an image of his house particularly bright and intense. He also wants a printed picture which is 5x7 inches. To produce a good quality photo print at your photo store, your image needs a resolution of 300 dpi. Then you need to resize the image by changing the print size to make it print at the correct size (5x7).



Your Task

Re-color the image Task1-2-House.jpg so that it's especially colorful. Change the resolution of the image to 300 and resize the image to 5x7 inches. Your desired image:

Save it as YourFourDigits-Task1-2-House.jpg.



Steps

1. Open the image Task1-2-House.jpg from the Unit 1 folder
2. Intensify the colors:
 - a. Open *Layers* menu → *Adjustments* → *Hue/Saturation* Ctrl+Shift+U.
 - b. Type 140 value in the *Saturation* box (a value between 120 and 170 is acceptable).
3. Open the *Image* menu → *Resize*.
4. Under the *Pixel size* heading set *Resolution* to 300. (*By absolute size* and *Maintain aspect ratio* must be checked.)
5. Set *print size height* and *width* to 5x7 inches.
6. Save as YourFourDigits-Task1-2-House.jpg.

Task 1-3: Adjusting Colors: Black and White

What's the Point?

Your uncle is still making that artistic collage of photos. This time he would like you to convert an image of his cousins to black and white for the collage.

Concepts

The simplest way to erase the colors in your picture is to convert an image to black and white, also called grayscale. Another way to convert a colored image to a black and white one is to set the **saturation** level to zero. Saturation is the intensity of the colors in the image.

However, the reverse is impossible. You can't turn a black and white image into a colored image by increasing the level of saturation; there's no color there to intensify. If there is color in the image to start with, increasing the level of saturation will make the colors brighter and more intense.



Your Task

Re-color the image Task1-3-Cousins.jpg so that it's black and white. Your desired image:

Save it as YourFourDigits-Task1-3-Cousins.jpg.



Steps

1. Open the image Task1-3-Cousins.jpg from the Unit 1 folder.
2. Convert this image to a black and white or grayscale image. Use one of these two choices:
 - a. *Use Menu Choice*
 - i. Open the *Layers* menu → *Adjustments* → *Black and White* or Ctrl+Shift+G
 - b. *Lower Saturation*
 - i. Open the *Layers* menu → *Adjustments* → *Hue/Saturation* or Ctrl+Shift+U
 - ii. Type a value of 0 and click OK in the saturation box or slide the Saturation slider to the far left.
3. Save as YourFourDigits-Task1-3-Cousins.jpg into Unit 1 folder.



Congratulations! You have finished Unit 1, and you are on your way to impressing your sister, mom, and uncle.

Unit 2: Enhance the Image: Select, Crop, Add Text

Concepts to be learned

- Actions apply only to selection (you must select in order to do something)
- Tolerance
- Using Ctrl key to extend selection
- Methods of selection
 - Rectangle, magic wand
 - Inverse selection
- Methods of deleting something
 - Magic Wand + delete, cropping,
- Resizing canvas vs. resizing image
- The color you want to add should appear on the Primary color display
- Putting an object on a separate layer will allow you to apply actions to it independently

Skills to be learned

- Select object(s) of the image.
- To crop the image to selection.
- To resize the canvas.
- To set and use primary and secondary color displays.
- To add text to the image.
- To import object(s) on the image from other files.
- To change the size of the selected object of the image.
- To delete the object from the image.

Task 2-1: Invert Selection

What's the point?

You decide to give some character to your picture of the sheep. You found an interesting picture with a bow tie. You want to put this bow tie on the sheep.

You need to learn some new skills to do it: selection, cropping, and changing the canvas size. To practice selection and changing the canvas size, you will remove two of the ducks in the image and resize it.


Concepts

Selection

Selection is an important skill to learn, because actions apply only to selections. You need to select the item in order to do something with it. Selection can be done with several different tools:

-  **Rectangle Select**
-  **Lasso Select**
-  **Ellipse Select**
-  **Magic Wand**

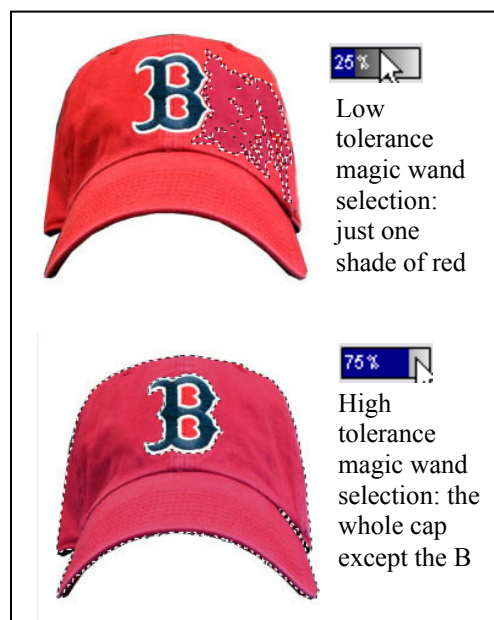
The Magic Wand selects pixels that are similar to each other, e.g. all the red pixels in someone's red baseball cap. This is very helpful for selecting odd shapes (like a baseball hat). But when

you use the Magic Wand, setting the right **Tolerance**  on the Tools panel is important.

The lower the tolerance value is, the more conservative the Magic Wand will be, e.g. selecting only pixels that perfectly match the one you picked. If the tolerance is higher, the Wand will select more liberally, including perhaps pink pixels along with red ones. With a very high tolerance, you will select just about everything.

To deselect the object, click on the Rectangle Select tool and then click anywhere on the image. The Paint.NET Help (under the *Help* menu → *Help Topics*) has more information on these tools.

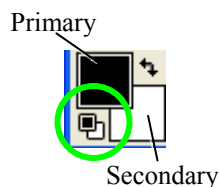
Holding **Ctrl** key while you select will allow you to select multiple objects or extend a selection you already have.



Invert Selection can be useful if what you want to delete everything *except* your selection. Choosing Invert Selection selects everything that you don't currently have selected, so you could then press the **Delete** key.

Canvas Size

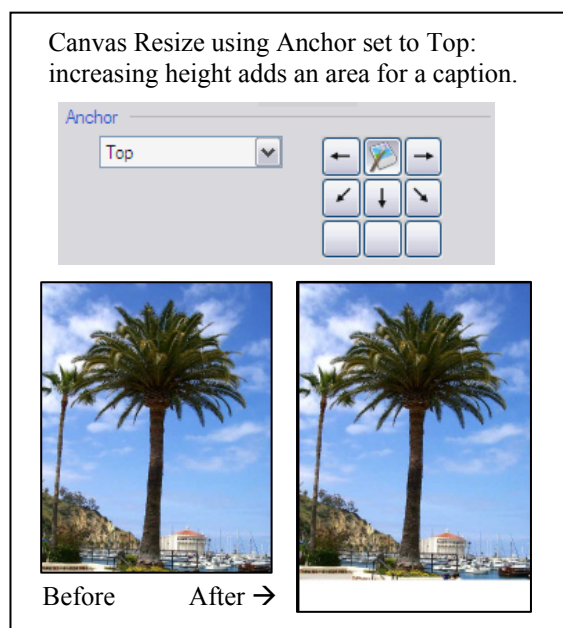
In some cases you want to change the size of the **Canvas**, rather than the size of the image. This means that you leave your picture the same size and resolution, but you either add or subtract pixels along the edges. Adding pixels will add the secondary color (white by default), and subtracting pixels will "eat into" the image. For example, if you want to make a collage you need a canvas bigger than an image. You may also want to reserve some room for a caption on the canvas. **Anchor** will allow you to position the image on the canvas. For example, if you set the anchor to **Middle**, it will add some extra canvas size to both sides of the image. If you set the anchor to **Top**, it will add some extra canvas size at the bottom of the image and so on.



Setting Colors to Black and White

If you ever want to set the colors to the default black and white (perhaps after experimenting with other colors), click on **Black and**

White icon under the **Primary** and **Secondary** color display icons.



What's Next?

Practice these skills on different images so that you can add the bow tie to the sheep.







Your Task

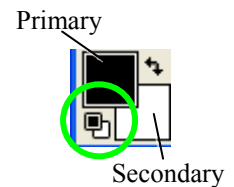
Use the Invert Selection command to delete everything except the smallest ducky from the image Task2-1-Duckies.jpg. Then increase the image height to 3.20 inches to make a stripe at the bottom of the image and add the caption, "I have a long way to go." Your desired image:



Save it as YourFourDigits-Task2-1-Ducky.jpg.

Steps

1. Open the image Task2-1-Duckies.jpg from Unit 2 folder.
2. Select the smallest duck using the magic wand:
 - a. Set the *Tolerance* on the *Tools* panel to about 50%. 
 - b. Click on the *Magic Wand* and then on the smallest duck. Note: hold the *Control (Ctrl)* key and click again until you have selected all the parts of the duck. Don't forget the eye!
3. Invert your selection using *Edit* menu → *Invert Selection* or Ctrl+I.
4. Delete the selection by pressing the *Delete* key.
5. Pick a blue color by doing the following:
 - a. Click on the *Primary* color display tool on the *Tools* menu (a color wheel panel will pop up if it is not already present) or choose *Window* menu → *Colors*.
 - b. Click on a blue color on the *Color* wheel and close the *Color* Dialogue.
 - c. Click on the *Paint Bucket* tool  and onto the background around your lone ducky to fill the color there.
6. Change the canvas size by doing the following:
 - a. Open the *Image* menu → *Canvas Size*.
 - b. Set the *Anchor* to *Top*.
 - c. Set *Print Height* to 3.20 inches.
 - d. Click OK.
7. Add text (caption):
 - a. Click on the *Black and White* icon under the *Primary* color display. It will set the *Primary* color to black. 
 - b. Click on the *Text* tool. 
 - c. Type "I have a long way to go..." in the caption area.
8. Save as YourFourDigits-Task2-1-Ducky.jpg into the Unit 2 folder.



Task 2-2: Cropping

What's the Point?

In this task you practice cropping, which will help you with the bow tie image later. Your goal is to remove one of the lions from the image provided.

Concepts

Cropping:

Yet another way of removing distracting background elements, cropping reduces the size of the image canvas and is the opposite of increasing the canvas size.



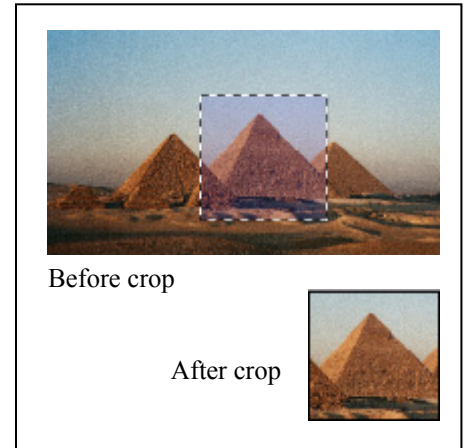
Your Task

Crop out the left-hand lion from Task2-2-Lions.jpg. Your desired image:

Save it as YourFourDigits-Task2-2-Lion.jpg.

Steps

1. Open the image Task2-2-Lions.jpg from Unit 2 folder.
2. Click on the *Rectangle Select* tool and holding the left mouse select the left-hand lion by drawing a rectangle around it.
3. Crop the selection using one of the following options:
 - a. Open the *Image* menu → *Crop to Selection* or
 - b. Use the *Crop to Selection* tool icon in the top toolbar.
4. Save as YourFourDigits-Task2-2-Lion.jpg into the Unit 2 folder.



Task 2-3: Add the Bow Tie to the Sheep

What's the Point?

Now that you have the skills to give some character to your picture, you can put a bow tie on the sheep. After that you will want to expand the canvas to make room for a caption.



Your Task

Add the bow tie from the image Task2-3-Puppet.jpg onto the image Task1-3-Sheep.jpg and then add a caption at the bottom. The height of the caption area is 0.3 inches. Your desired image:


Save your work as YourFourDigits-Task2-3-sheep.jpg.



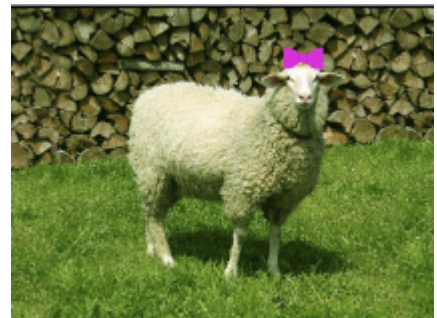
Part 1: Add a bow tie

1. Open the image Task2-3-Sheep.jpg from The Unit 2 folder.
2. Choose *Layers* menu → *Import From File*.
3. Select the Task2-3-Puppet.jpg image from the Unit 2 folder.
4. If you do not see the Layers panel in the lower right corner, Press **F7** key or choose *Windows* menu → *Layers*. You can see that the image you have imported is put on a separate layer on the Layers menu at the bottom right corner. You will be able to work with the imported image without affecting the image of the sheep.
5. Set the *Tolerance* on the *Tools* panel to about 36%, if necessary, and click on the Magic Wand and then on the bow tie. You may need to hold the Ctrl key and click again to select additional areas.
6. Under the *Edit* menu → *Invert Selection (Ctrl-I)*. You have now selected everything on the current layer except the bow tie.
7. Press the *Delete* key to delete everything except for the bow tie from that layer.
8. Select the bow tie with the *Rectangle Select* tool or any other selection tool so that you can edit it.

Increase the size of the bow tie to look appropriate on the sheep:

- a. Click on the *Move Selected Pixels* tool on the *Tools* menu.  Use the nub on the corner of the image to change the size of the selection. When you can see the mouse cursor icon, which looks like a black arrow, you can drag the entire selection.
- b. Position the tie on the sheep's head using the *Move Selected Pixels* tool.
- c. To disable the selection press Enter or click on the *Rectangle Select* tool and then anywhere on the image.

At this point your picture may look like this:



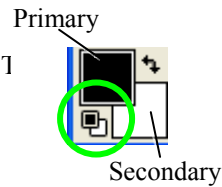
(continue on to Part 2, below)

Part 2: Add a Caption

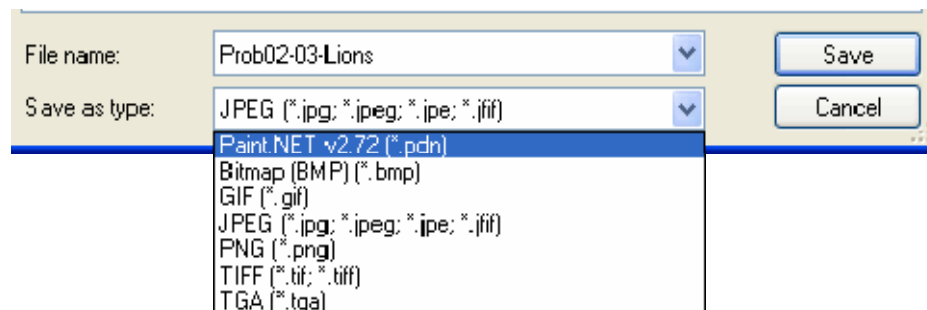
1. If needed Click on *Black* and *White* icon on the T white.

This is important because you want to set the color for the text in the caption to black.

2. Open the *Image* menu → *Canvas size*.
3. Check the radio button: *By absolute size*.
4. Uncheck the box: *Maintain aspect ratio*.
5. Under the *Print size* heading increase the value of *Height* by 0.3 inches.
6. Set *Anchor* to *Top*.
7. Click OK. You will add a white region at the bottom of your image for a caption. The white comes from the current secondary color tile. (If you wanted a different color caption area, you could change that secondary color before changing the canvas size.)
8. Add text (caption):
 - a. Click on *Text* tool.
 - b. Set the font size to 48 up in the toolbar.
 - c. Type in the caption area: "I'll do anything for a sheep laugh."
9. Save as **YourFourDigits-Task2-3-sheep.pdn** into the folder Unit 2. Make sure that you save it as a **pdn** file; otherwise, Paint.NET will flatten your layers.



urrent colors to black and



Congratulations! You have finished Unit 2, making sheep farmers everywhere complement your work.

Unit 3: Build an Image: Work with Layers

Concepts to be learned

- Layers vs. overwriting (single-layer), flattening
- Ordering layers
- Moving regions of an image
- Working with text
- Adding layers by importing them from files vs. adding new layers
- Flattening layers

Skills to be learned

- To add color to the image.
- To use Move Selection Pixels tool.
- To add layers.
- To use text.
- To name layers.
- To save the image with .PDN v. .JPG extension.

Task 3-1: Manipulating Layers

What's the point?

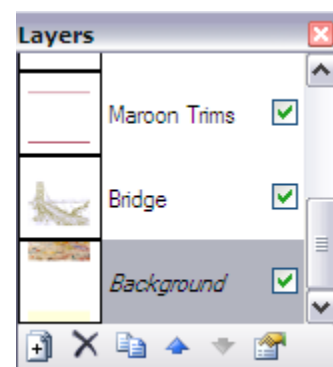
A nearby long-term care facility has asked you to design the cover image for a collection of stories that they've gathered from their elder residents. They have certain text and color requirements, and this task will require that you work well with layers.

Concepts

Layers are nothing more than independent images that you can set to interact with each other from the Layers panel. By putting image elements and effects on independent layers, you give yourself the flexibility to adjust your artwork in the future.

The most basic use of layers is to keep objects separated from each other so you can modify their horizontal and vertical position as well as their front-to-back arrangement. Paint.NET allows you to transform individual layers by scaling, rotating, or even distorting them. When you want to add something to a layer or edit one, make sure that that layer is selected within the Layers panel.

As you add layers, your composition gets bigger; this implies that Paint.NET requires more space in memory and on disk to manage the file. The **.pdn** file format is the only choice within Paint.NET to save images with more than one layer.



You can reduce the size of an image on disk by merging multiple layers into one by choosing *Image* menu → *Flatten*. Flattening the image is also important when we want to apply an action to an image with multiple layers. For example, flattening will be appropriate when you want to convert a colored multi-layered image to a black and white image.

What's Next?

To complete the booklet project you will need to learn how to manipulate layers and add text.

Task 3-1: Creating a Booklet Cover from Several Images

What's the Point?

The long-term care facility has given you several images to work with for the cover of their booklet of stories. They would like you to use some art work from their favorite artist, Josh Dorman. They would also like you to use a specific picture of a bridge. Before you add all the text and other trim, they'd like to see how those two images look together.



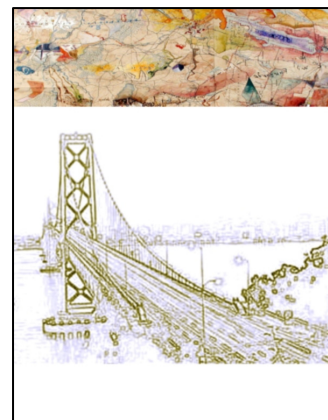
Your Task



Use the images Task3-1-Josh.jpg and Task3-2-Bridge.jpg to construct an initial mockup of the booklet cover. The cover will be 5.6" wide by 7.2" tall. Because you are creating a mockup, use lower resolution (72 dpi) rather than full print resolution. You've decided that the upper artistic area should be 1.7" tall. Your desired image:

Save your work as YourFourDigits-Task3-1-Bridge.pdn.

Steps

1. Open the image Task3-1-Josh.jpg.
2. Rotate the image 90 degrees clockwise (*Image* menu → *Rotate* → *90 CW*).
3. Resize the image to be as wide as the entire booklet cover. Open the *Image* menu → *Resize* or Ctrl+R.
4. To set Resolution to 72
 - a. Go to *Image* → *Resize*
 - b. Under the *Pixel size* heading set *Resolution* to 72.
5. Uncheck *Maintain aspect ratio* box so that you can squeeze the entire Josh image into the desired space.
6. Type into the cell print size width to 5.6 inches and height to 1.7 inches.
7. Click OK.
8. Add room on the canvas for the bridge and the subtitles to come later below.
 - a. Make sure that the *Secondary color* on the *Color display* is white
 - b. Open the *Image* menu → *Canvas size* or Ctrl+Shift+R.
 - c. Check the radio button: *By absolute size*
 - d. Uncheck the box: *Maintain aspect ratio*
 - e. Under the *Print size* heading change the *Height* to 7.2 inches.
 - f. Set the *Anchor* to Top.
- a. Click OK. (Note: If your new image area is not white, Undo, change the secondary color to white, and change the canvas size again.)
9. If you don't see a small panel with layers shown at the lower right, open the *Windows* menu → *Layers* or type F7 (a layer window will pop-up). You should have one layer right now with the Josh artwork on it.
10. Choose *Layers* menu → *Import from File*.
11. Choose the image Task3-1-Bridge.jpg from the Unit 3 folder. It will create a new layer for this image (notice that a new layer is added in the layer panel).



12. Rename the new layer using one these choices:
 - a. Double click the layer in the Layers panel. Type in the name Bridge.
 - b. Open the *Layers* menu → *Layer properties* or type F4. Type in the name: Bridge.
 - c. Click on the *Layer properties* icon at the bottom right corner of the Layers panel.  Type in the name: *Bridge* and click OK.
13. Use the *Move Selected Pixels (M)* tool  on the *Tools* panel to position the bridge so it looks like the target image. Press the *Enter* key to set the selection after moving it.
14. Save as YourFourDigits-Task3-1-Bridge.pdn into the Unit 3 folder. Make sure that the extension of your file is .pdn.

Task 3-2: Add Trim and Text to the Booklet Cover

What's the Point?

Your client at the long-term care facility liked your earlier mockup generally and gave the following suggestions:

- Lighten up the Josh artwork some; otherwise the text titles over it won't be readable.
- Shrink the artwork so that there's more white space above the bridge.

A colleague of yours has made these changes and suggested some additional design ideas, and now it's time to add the remaining trim and text.

Also, the client would like a black and white version of the final design to make sure that it will photocopy well.

This task continues your work with layers.



Your Task

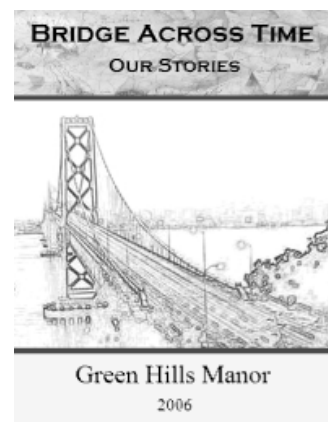
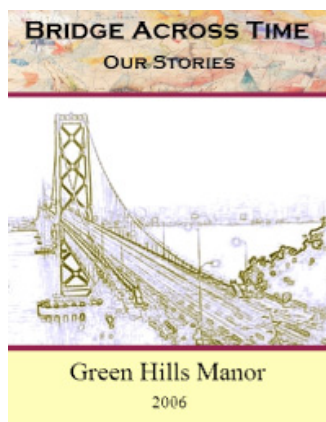
Use Task3-2-Booklet.jpg as the basis for your final booklet cover design. Use the Font *Times New Roman*, size 24. Your desired images:


Save your final color design as YourFourDigits-Task3-2-Booklet.pdn (note the **pdn** file format).

Save your black and white version as YourFourDigits-Task3-2-BookletBW.jpg.

Part 1: Design the Cover

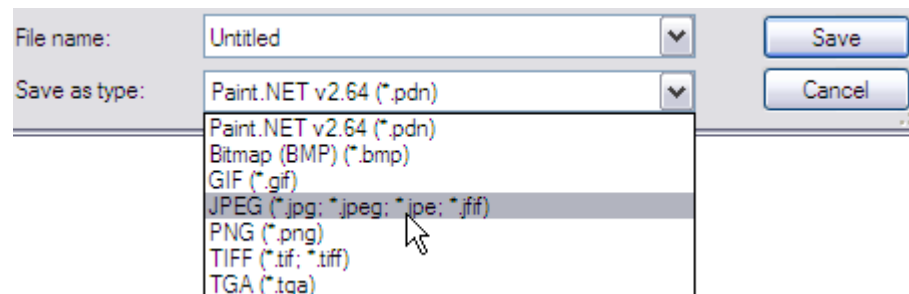
1. Open Task3-2-Booklet.jpg from Unit 3 folder.
2. Add a new layer for the maroon trim stripes. Open the *Layers* menu → *Add New Layer* or Ctrl+Shift+N.
3. Open the *Layers* menu → *Layer properties* or type F4 and rename the layer, "Maroon Trims."
4. Use the *Rectangle Select* tool to outline the areas for two maroon strips (see the target image). Hold the Ctrl key to outline both areas at the same time.
5. Pick a maroon color and fill in the selected areas with this color using the Paint Bucket.
6. Deselect the areas by choosing the *Rectangle Select* tool and clicking anywhere on the image.
7. Set the *Primary Color* to black (for the text).



8. Now add the text. Add a new layer by doing one of these two options:
 - a. Open *Layers* menu → *Add New Layer* (Ctrl+Shift+N)
 - b. Click on the icon *Add New Layer* from Layer panel 
9. Name the new layer: *Green Hills Manor*.
10. Type the "Green Hills Manor" title:
 - a. Use the Font: *Times New Roman* , size 24
 - b. Type in: *Green Hills Manor* (see target image)
 - c. If needed, move your typing by selecting it with the *Rectangle Select* tool and then using *Moving Selected Pixel* tool (M) on the *Tools* panel.
11. Add a new layer → name this layer: *2006*.
12. Add the text "2006" using size 16 Times New Roman font.
13. Click on the Background layer in the Layers Panel.
14. Use the *Rectangle Select* tool to select the background region behind the text "Green Hills Manor" and "2006." Because your text is on different layers, you don't have to select "around" the text.
15. Fill in selected background with creamy yellow color using the Paint Bucket.
16. Save as YourFourDigits-Task3-2-Booklet.pdn into Unit 3 folder. Make sure to use the extension **.pdn** (Don't close the file, instead do the next step.)

Part 2: Create a Black and White Version

17. Flatten the image by choosing *Image* menu → *Flatten* (click OK if a window pops up asking about discarding hidden layers).
18. Change the image to black and white using one of these approaches:
 - a. Choose the *Layers* menu → *Adjustments* → *Black and White*.
 - b. Choose the *Layers* menu → *Adjustments* → *Hue/Saturation* and change the Saturation to 0.
19. Save your work as YourFourDigits-Task3-2-BookletBW.jpg into Unit 3 folder. Make sure to use the extension **.jpg**. Use the "Save as type" drop down menu in the Save As window to choose "jpg."





Congratulations! You have finished Unit 3 and the entire training.

BIOGRAPHICAL SKETCH

Claver Pierre Hategekimana was born on July 21, 1971 in Gitarama, Rwanda where he received primary and secondary education. In December 1997, he embarked on an ambitious journey to attend college in the United States. In 2000, he received an Associate degree from Des Moines Area Community College in Ankeny, Iowa. In 2002, he received a Bachelors of Science in Management Information System from Wartburg College in Waverly, Iowa. In 2005, he earned a Masters of Science in Management Information Systems with an E-Commerce concentration from Dakota State University in Madison, South Dakota. In the fall of 2005, he started his doctoral studies at Iowa State University with a major in Human Computer Interaction and a concentration in Curriculum and Instructional Technology. Concurrently, he served as a Research Assistant in the Department of Logistics and Management Information Systems until the summer of 2007. In 2007-Present, he has been holding the position of Teaching and Learning Center Coordinator at Wenatchee Valley College in Wenatchee, Washington. Claver's interest in education and information technology stems from his growing interest in academia and empowering businesses with necessary applications for effective development.