2002

The development, design, and theory of educational interactive multimedia software

William James Chapman

THE DEVELOPMENT, DESIGN, AND THEORY OF EDUCATIONAL INTERACTIVE MULTIMEDIA SOFTWARE

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Arts
in
Education:
Instructional Technology

by
William James Chapman
June 2002

THE DEVELOPMENT, DESIGN, AND THEORY OF EDUCATIONAL INTERACTIVE MULTIMEDIA SOFTWARE

A Project
Presented to the
Faculty of
California State University,
San Bernardino

by
William James Chapman
June 2002

Approved by:

[redacted]

James Monaghan, First Reader

Date

[redacted]

Todd Jennings, Second Reader

ABSTRACT

This thesis outlines major concepts in the development of interactive educational multimedia software. The development issues contained within this thesis include mental models, educational theories, and basic design issues as applied to the development of educational software. Other issues of importance to software design discussed include the need for effective navigation and effective use of graphics and videos. These developmental issues are demonstrated through human rights educational software which was developed to accompany this thesis.

ACKNOWLEDGEMENTS

I wish to acknowledge both advisors on this project, James Monaghan, Ed.D. and Todd Jennings, Ph.D.for their help, guidance and input on this thesis and project. I also wish to acknowledge Lyne Chapman, for her support. Special thanks go to Clay and Beth Chapman. I also would like to acknowledge Ron and Pat Chapman, my parents, for their support all the way through college. I would like to thank Amnesty International for the use of their resources in the completion of this thesis. I also wish to acknowledge everyone who answered questions or gave me ideas, thoughts or encouragement throughout the completion of my thesis.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE

Introduction

In the past 20 years of desktop computing much as changed. It was once that computers were mainly business tools with business applications. During this time a person needed to learn how to "program" the computer to create software. As computers became more affordable they made their way into classrooms as educational tools. Now with advancements in both software and hardware teachers and students no longer need to rely upon "programmers" to create educational software, they can easily create the software themselves.

Need for Design Theory

Now that multimedia software exists which makes it easy for teachers and students to create their own educational software there also needs to be education about software design. In creating educational software designers need to be able to create educational software which will support education. Improperly designed software can lead to confusion, frustration and the inability to accomplish the educational objectives of the software. Properly designed software can provide exceptional support in the classroom for the teacher and can lead to a positive view of technology when students create educational software.

Need for Human Rights Educational Material

It is very common today to open a newspaper or listen on the news and hear about human rights violations. Many times these issues exist in the news without even a mention of the words "human rights". The frequency and effects of human rights violations make human rights topics pertinent and timely to be addressed in education . It is through education that society can become informed about human rights violations and take a stand to not tolerate abuses against other people. Computer software is an effective method to get human rights information into the classrooms. Multimedia software is engaging for students, it is comprehensive and allows a teacher, even without having much education about human rights issues, to effectively teach about this important topic.

Purpose of this Project

This project provides teachers and students with background information regarding software design. Specifically it will focus on design theory including content, interface, graphics, animation, navigation, and audio issues which may arise during the design and construction of educational multimedia software. The design issues represented here are demonstrated in the software "You can make a difference … Human Rights", which was designed to accompany this research.

Definition of Terms

Authoring Tool - software which can be used to create or "author" multimedia computer software. Many of these tools, such as Hyperstudio, are simple enough for students to use in a classroom setting.

CD-ROM - Compact Disc - Read Only Memory - a CD-ROM is a method of media distribution which is very effective with multimedia content. CD-ROM's can hold up to 680 megabytes of information thereby allowing software to include a large amount of media.

GUI - graphical user interface

Hypermedia - this term is typically used synonymously with Multimedia, however hypermedia tends to denote multimedia with interactivity, i.e. computer software. Hypermedia is characterized by the ability of the user to jump to any point within the software easily.

Multimedia - the use of many different media to present information. Media such as sound, music, graphics, video, etc.

Multimedia Authoring Tool - see authoring tool.

Quicktime - this is Apple computers multimedia presentation layer which is available for both Macintosh and IBM compatible computers. This software allows the computer to be able to playback / create video, audio and static graphics.

Xerox Palo Alto Research Center (PARC) - an early (1970's) research center which was a pioneer in the development of GUI's for computer software and operating systems.

CHAPTER TWO

Basic Design Issues

Early design of the computer GUI was done at the Xerox PARC research center and as Lynch (1994) states, involved concepts which were firmly grounded in the theories of educational psychologists. The research at Xerox PARC focused on the works of Jean Piaget and Jerome Bruner, which demonstrated "that our understanding of the world is fundamentally linked to visual stimulation and the tactile experience of manipulating objects in our environment" (Lynch, 1994 paragraph 7). This research is of key importance to the development of the GUI for computer users, and eventually became the basis of most operating systems GUI's that are in use today including Apple Mac OS, Microsoft Windows, and the GUI's in use on various Unix workstations (Lynch, 1994). According to Lynch (1994) there are two issues which are faced in hypermedia interface design. The first of these is "informing and guiding the computer user through a complex body of information" (Lynch, 1994 paragraph 3). The second issue which must be dealt with is "the creation of a visual design rhetoric appropriate for interactive computer displays" (Lynch, 1994 paragraph 3). The design of a hypermedia application "involves capturing and organizing the structure of a complex domain and making it clear and accessible to users" (Isakowitz, et al., 1995 paragraph 1).

In summarizing the major points of Alan Marcus's User Interface Standards Manual, Rettig (1992) denotes that it is a theory of interface design is very closely based on the theory of Semiotics (the study of symbols). There are four parts which make up the computer interface:

1. Lexical structure - what actual symbols (graphics) are on the screen.

2. Syntactic structure - how do theses symbols (graphics) relate to one another.

3. Semantics - how are the symbols (graphics) related to what they are supposed to represent.

4. Pragmatics - what do they represent to the user.

According to Rettig (1992), these 4 aspects of interface design listed above will enable the creation of software that is easier to user, less confusing and more educational.

Thuring, et al. (1995) show support for the above theory basing the idea that software should be a product which does not create too much cognitive overhead. They stated that cognitive overhead occurs during a task in which the user is presented with too many decisions, choices, or things to remember at one time. According to Thuring et al. (1995) software that is too cognitively overloading for the user can lead to a lack of comprehension of informational content of the software. Their recommendations to minimize cognitive overhead would be to provide methods for the user to:

1. easily locate current position in software.
2. show how they got where they are.
3. make it clear how to navigate the software to get to new areas and provide an indicator of current direction user is moving in.

Thuring, et al. (1995) also place a focus on software coherence. Their recommendations to develop software that is designed with coherence in mind would be to:

1. have information presented in a non-fragmented manner so that users can see the relationships between the information.
2. provide an overview of the document.

To facilitate coherence, the designer would want to present the information in a manner which would provide for the semantic relationship (as mentioned above in Rettig (1992)) between information, allowing the user to know what contents of other sections are and how the sections are related Thuring, et al. (1995). By maximizing coherence and minimizing cognitive overhead Thuring, et al. (1995) state that we can help the user to more easily comprehend the content contained in the software.

Mental Models

Norman (1986) defines learner interactions with software as taking place within mental model. According to Reiber (1994) mental models can be defined as "an individual's conceptualization, or theory, of a specific domain or system" (Reiber, 1994, p.222). Norman (1986) defines mental models as model that a person mentally creates of the world to make sense of it. According to Norman, users, while interacting with software will create mental models about the software and how to interact with it. In Norman's (1986) theory of software interaction there are three sections, the Design Model, the User's Model, and the System Image. He describes the Designer Model as the way the software designer has visualized the system working. He describes the User's Model as the way in which the learner visualizes the system working. Finally he describes the System Image, which is the only physical part of the design, as comprised of the actual images and interactions which take place within the computer.

He states that to promote a high level of usability in software, the software would have a user's model which is exactly the same as the design model. Norman (1987) sums this up by stating "Ideally, the user's model and the design model are equivalent" (Norman, 1997, pg. 189). According to Norman (1986),

> "The designer should want the User's Model to be compatible with the underlying conceptual model, the Design Model. And this can only happen through interaction with the System Image. … If one hopes for the user to understand a system, to use it properly, and to enjoy using it, then it is up to the designer to make the System Image explicit, intelligible, consistent" (p.47).

Norman claims that the problem is that the Design Model and the User Model may vary considerable depending upon the knowledge, experiences, etc. of both the designer and the user. It is imperative that the system image is accurate, that is "consistent with and exemplifies the operation of the proper conceptual model" (Norman, 1988, p.190). He states that,

> " it is up to the designer to start with a design model that is functional, learnable and usable. … Only then can the user acquire the proper user's model, and find support for translation of intention into actions and system state into interpretations. Remember, the user acquires all knowledge of the system from that system image." (Norman, 1988 p.190).

Norman believes that to be most effective the software developer must be capable of creating an effective model, for the user to interact within, one which is capable of being understood and learned by the user.

According to Hollingworth (1995), while developing software the designer should want to make sure that all user actions are small steps toward a more complex action. If an action does too many things, or is too complex for the level of the user, the user may have a difficult time attempting to understand and use the action properly in the future.

According to Watson (1997), software developers can use the knowledge of how a user develops a mental model to make inferences about learning time, errors and the overall usability of the software. He focuses this by stating that the designer can create software which develops appropriate mental which are easier to use and more effective at transferring learning from the software to the user.

Watson (1997) explains that when using metaphors the designer could describe to a user that the computer word processor is similar to a typewriter, this description being a verbal metaphor. He describes the virtual metaphor as a metaphor in which the designer would compare two concepts, one that is unfamiliar to the user and another one that is familiar to the user. According to Watson, a virtual metaphor can be a very powerful teaching tool within software. He describe some of the virtual metaphors have appeared within the computer's basic operating system, such as folders, trash can, recycling bin, cut and paste, etc. Watson states that by using the software to develop an effective mental model the developer can create metaphors which can be used to teach about abstract thoughts.

Expanding upon user's interactions with software, Mark (1986) points out that the User's Model is constantly changing as the user interacts longer with the software. As Mark points out, users will continually construct and reconstruct their mental models regarding the software. Montague and Knirk (1992) state that it is during this constructing and reconstructing of ideas the users are organizing them and forming new relations between. Therefore they believe that in developing effective software, the software must be able to effectively communicate how to access both the software and the content in a manner which is appropriate for the user. Accordingly, they state that this can be accomplished through facilitating the development of effective mental models. A problem with mental models, according to Wittman (1997), is that user's mental models may be "incomplete, self-contradictory, and/or inconsistent with he world around us". Richmond (1996) points out that problems can also arise if the design does not meet the users expectations. He uses the example of a clock that is numbered counterclockwise. The clock would still be usable, however in a rush the user may make an error getting the time due to mixing up the numbers. Montague and Knirk (1992) also point out that often times mental models which have been developed by students are incorrect, and can quite often lead to errors in their learning.

Educational Theory

Basic design models may vary based upon the pedagogical viewpoint of the individuals creating the software. This project will focus primarily on the constructivist viewpoint, as it is the design theory chosen by the author of this paper for the development of the accompanying software. The decision for the use of the constructivist design theory will developed in more detail later in the paper during the discussion of the software.

According to Reiber (1994) constructivist theory states the each individual defines and constructs understanding of the world. When applied to education he expands on the theory be stating that because each person sees the world in a different way, each person needs to construct knowledge, it cannot simply be handed down to them. According to Reiber (1994, p.248), "Knowledge cannot be imposed on an individual; rather, knowledge is itself constructed by each person." Reiber describes that a constructivist viewpoint would believe an individuals knowledge base is constantly being updated and revised to take into account new learning. Following from this he states that in the constructivist view of education a teacher would provide students with a variety of experiences from which to learn, or construct meaning. As Reiber (1994) states;

> "In constructivism, quality of knowledge structures, not their quantity, is the issue. In other words, learning is not about acquiring new knowledge, but about the constant reconstruction of what someone already knows. As a person's knowledge structures are continually "revised," there is the occasion where a new structure is formed because new information just no longer matches the available structures" (Reiber, 1994, p. 240)

Reiber (1994) presents certain guidelines which should be followed when developing interactive software. He states that the software should "Provide a meaningful learning context that supports intrinsically motivating and self-regulated learning" (Reiber, 1994, pg 241). To expand on this he believes that learning is most effective when it is presented in a context which is relevant or interesting for the learner, and this context needs to be constantly changing, such as in level of difficulty or tasks, to continue to hold attention and for continued appeal to the students.

According to Reiber (1994), software should "establish a pattern where the learner goes from the 'known to the unknown'"(pg 241). Reiber points out that it has been shown that one of the most important concepts in learning is the ability of students to apply what they are learning to what they already know.

According to Reiber (1994), the software must support both an equal amount of deductive and inductive learning. He describes software which is based on inductive principles as being much like a workbook for the students to complete, often this can lack imagination and each section tends to start looking like the previous section. Reiber decribes this type of design as typically best for creating software which is focusing on lower level learning or simply learning facts. Reiber describes software which is designed using deductive principles can often become confusing for the user if they do not know how to use the software. He believes that this design strategy can cause students to be disinterested or bored if they do not understand what to do. Therefore to accomplish this he recommends that software should provide both direct instruction and also allow time for the users to explore the software.

Reiber states that while designing using a constructivist design theory, errors made by users should be explained in a useful non-negative manner. According to Reiber software should "emphasize the usefulness of errors" (Reiber, 1994, p.243). He states that when explaining errors it is important that the error message does not imply failure. Reiber points out that within the constructivist design theory errors can actually be seen as an important portion of learning when engaged in problem solving. He concludes that for information from errors to be effectively processed, the user must first have a goal of what it is that is to be accomplished, and this goal must be well understood. Secondly he states that the information from errors should be easily transferred to the goal the user is attempting to accomplish (Reiber, 1994).

As an extension of constructivism Soloway et al. (1996) presents the design pedagogy; Learner-Centered Design. He describes Learner-Centered Design as focusing on three important aspects; Growth, Diversity and Motivation. Within Learner-Centered design he expands on the concept of growth as keeping the user focused and the goal of the educational software to develop the user's level of expertise (growth) within the software academic focus. Within Soloway's description of Learner-Centered Design, the software must also take into account the diversity of the users. He substantiates the reasoning behind this as being the fact that different users have different cultural, gender, etc. beliefs which must be taken into account to create software suitable for each user. According to Soloway, the software must be motivating for the user to continue using. As Soloway et al. (1996) states "the students initial interest and continuing engagement cannot be taken for granted."

Soloway (1996) describes scaffolding as a method of providing support for the user to allow them to more easily accomplish more difficult tasks. Soloway states that providing scaffolding support directly built into software is extremely important aspect of software design.

As described by Soloway (1996) Learner-Centered design focuses on the theories of constructivism and socioculturalism. As was described earlier by Reiber (1994), constructivism focuses on the users learning as being a process of assimilating ideas, restructuring ideas, and creating new constructs for ideas. Socioculturalism, as an educational process, can best be defined by Soloway et al. (1996) when he states; "learning is enculturation, the process by which learners become collaborative meaning-makers among a group defined by common practices, language, use of tools, values, beliefs, and so on." (Soloway, 1996, paragraph 3) In Learner-Centered Design, with a focus on constructivism and socioculturalism, Soloway states that scaffolding is provided for the users in different methods. Having interactions between teacher/student, student/student, and even student/material interactions can be methods used to provide scaffolding. He states that scaffolding can also be provided, in a socioculturalism design, by "the contextual constraints and affordances that support understanding, action and meaning making through language use, practices, tools, social interactions, and so on of a culture (Soloway, et al., 1996, paragraph 3). Quintana et al. (1996) expand on this that to further provide scaffolding for users the designer should reduce the complexity of tasks the user will have to undertake, and provide built in support for low-level tasks. Therefore they recommend that software should have all of the necessary tools built into the current software. They would also recommend that the software's tools should support the different learning styles of the users and also the different levels of expertise the users may have. Accordingly they would have the interface of the software developed to help the user by providing a visual structure of the task and to encourage articulation by the user (Quintana, Abotel, and Soloway, 1996).

It was described in Soloway (1996) and Quintana et al. (1996) that scaffolding has typically been provided by other people, teachers, students, tutors, etc. Accordingly they would now promote the development of software which utilizes a Learner-Centered Design. They describe this software as being able to provide scaffolding for the user during interactions with the software.

One such software that demonstrates scaffolding is Digital Chisel (1997). Digital Chisel is a web / java authoring tool. Figure 1 shows the action menu of Digital Chisel when set at its easiest user mode. When

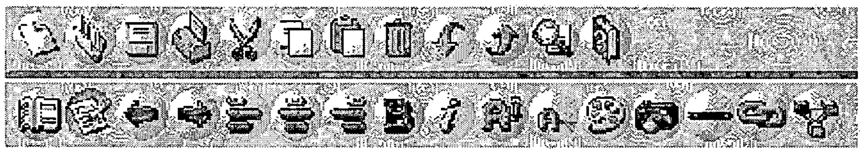


Figure 1. Digital Chisel Menu Bar as Shown in the Easy Mode

using Digital Chisel, the user has fewer choices when the software is set at the easy user mode, than when set at the medium user mode which is shown in Figure 2. Finally the advanced user is presented with the options shown in Figure 3. The more advanced levels of Digital Chisel offer the most power and ability for constructing more elaborate projects, however the

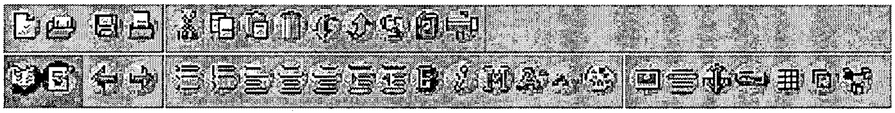


Figure 2. Digital Chisel Menu Bar as Shown in the Medium User Mode

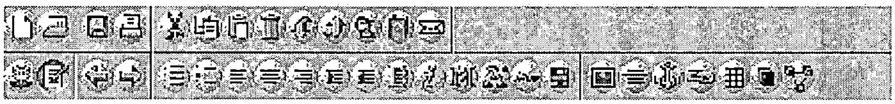


Figure 3. Digital Chisel Menu Bar as Shown in the Difficult User Mode

software makes it easy for the new user who is just learning how to use the software.

As an outgrowth of Constructivism, Papert (1993), describes learning in the theory known as Constructionism. According to Papert (1993), the basic idea of constructionism focuses on the "construction" of knowledge, it is based upon the theory that people mentally construct knowledge of their own. He states that within the theories of constructionism there is an importance in the construction of physical items within the world as being necessary to facilitate constructions within the mind (Papert, 1993). He promotes within the theory of constructionism that the learner develops mentally as they interact within the world. Therefore he believes that education should take place within the real world, by completing realistic tasks. Papert mentions kitchen math as being a constructionist method of learning. He presents this example of how math can be accomplished in a real life setting within a kitchen. In kitchen math the learner learns about math by the formulas within a cookbook. As the user works with the cookbook they develop mental tools for dealing with the problems they might encounter. This can particularly be shown by the learning that would occur while doubling a recipe and creating a problem for the user having to multiply times two. According to Papert (1993) constructionist pedagogy allows for the fact that different students will learn in different manners and will also approach different problems in different manners. Therefore Papert states that within a constructionist viewpoint, the focus of software should be on the learner creating something. As Papert (1996) explains, in the instructivist pedagogy, you have a game that teaches, whereas in the constructionist pedagogy; you make the game yourself. According to Papert (1996) by having the learner create the game themselves they will:

> "learn some technical things, for example how to program computers. They will learn some knowledge traditionally incorporated in the school curriculum, for example in order to make shapes and program movements they will have to think about geometry and about numbers. They will develop some psychological, social and moral kinds of thinking" (Papert, 1996 p. 47).

Papert (1996) believes that it is important that the learner through the constructionist theory of teaching and learning will be able to start taking a sense of control over their own learning. He recommends that software provide something for the user to share, or there should be something to create which can be collaborated on. According to Papert, this type of activity is structured around the constructionist theory of learning, that they learner is making their own meaning, but also gives the user a sense of pride and ownership over their creation.

As was stated by Papert (1993, 1996), constructionism as an educational theory, has it's focus on power of the construction of things as a method of learning. According to Evard (1994) this is especially true if the user is able to share these items with others. Evard believes that learning through software can be enhanced if the learners are presented with a chance to discuss authentic questions in regards to the software. Given the chance to construct conversation around relevant questions will enhance the learning within software, particularly if the questions are shared in a common space available to other users to access and respond to (Evard, 1994).

According to Norman (1993), there are two main modes of cognition which are relevant to learning. These two modes are experiential and reflective cognition. The mode of cognition which Norman calls the experiential mode of cognition is the mode that is commonly used when performing activities which are familiar and practiced. He states that users within this mode of cognition are typically completing actions are commonly reactions or knowledge which can be called forth without much thought. Norman describes other types of experiential cognition as being an expert in a specific field who has practiced these actions and can perform them without much effort. The other mode of cognition which Norman (1993) discusses in relation to education is the reflective mode. He states that reflective cognition is used to describe actions which must be thought about before being able to perform them. As Norman (1993) states:

> "The reflective mode is that of concepts, of planning and reconsideration. It is slow and laborious. Reflective cognition tends to require both the aid of external support - writing, books, computational tools - and the aid of other people" (Norman, 1993, p.25)

According to Norman, these two modes of thinking are very similar to the modes of software. He describes software which has a focus on the experiential mode use multiple methods of providing sensory information to the user and not have much focus on the use or development of skills of logical deduction. Whereas, he states that software designed in the reflective cognition mode would provide the user with the means to easily explore the software.

Norman (1993) also describes three modes of learning; accretion tuning, and restructuring. He describes the first of these three modes, accretion, is the acquisition of knowledge. Learning in this style, according to Norman, requires that the user already have a framework to build on, or the learning is much more difficult. He states that when there is not the appropriate framework then the learner will need to use learning tools, such as mnemonic strategies, repetition, etc. The second mode of learning described by Norman is tuning. Norman describes users in this mode as practicing using skills or knowledge they have acquired during the accretion stage. Tuning, according to Norman, is a long process by which a learner takes knowledge and slowly becomes an expert at using it. Within these first two modes, he describes the user being in the reflective mode of cognition. The third mode of cognition described by Norman, the restructuring mode, places the focus within the reflective mode of cognition. As described by Norman, during the stage the user develops conceptual skills related to the learning. He describes the difficulty with this stage of learning is that it is typically characterized by hard work, getting students interest and keeping them motivated to stay with the learning.

To keep students motivated software must have an "optimal flow". Optimal flow can be described as "a continual flow of focused concentration: absolute absorption in an activity" (Norman, 1993, p.31). While in an optimal flow the users attention would be focused completely on the software and the task they are trying to accomplish. Quoting Csikszentmihalyi, Norman (1993) states that to promote an optimal flow the developer should focus on having "built-in goals, feedback, rules and challenges, all of which encourages one to become involved in one's work, to concentrate and lose oneself in it (Norman, 1993, p.33). To create an optimal flow software should follow some basic design ideas. The software should have a high level of interaction and feedback. The goals and procedures should be well defined. The software needs to be capable of motivating the learner. It needs to provide a continuous challenge to the learner, but not a challenge which is discouraging or too easy. The learner should have the impression that they are working within the environment on the task. The software's tools should be appropriate for the task and not be distracting (Norman, 1993).

Computer Interfaces

The computer interface, according to Berenguer (1994), is one of the most important portion of the software. The computer interface is highly responsible for the success of the software. If the user of the software does not understand the software, it is typically the interface that they do not understand. Computer interface can be summed up as follows:

> "Interface design encompasses everything a user sees, touches, hears, and interacts with … its scope extends to include all visual components, as well as audio elements and, most importantly, interaction and navigation." (Haykin, 1994, p.131).

Lynch (1994) describes the interface and in particular its relations to design and development of the interface as follows,

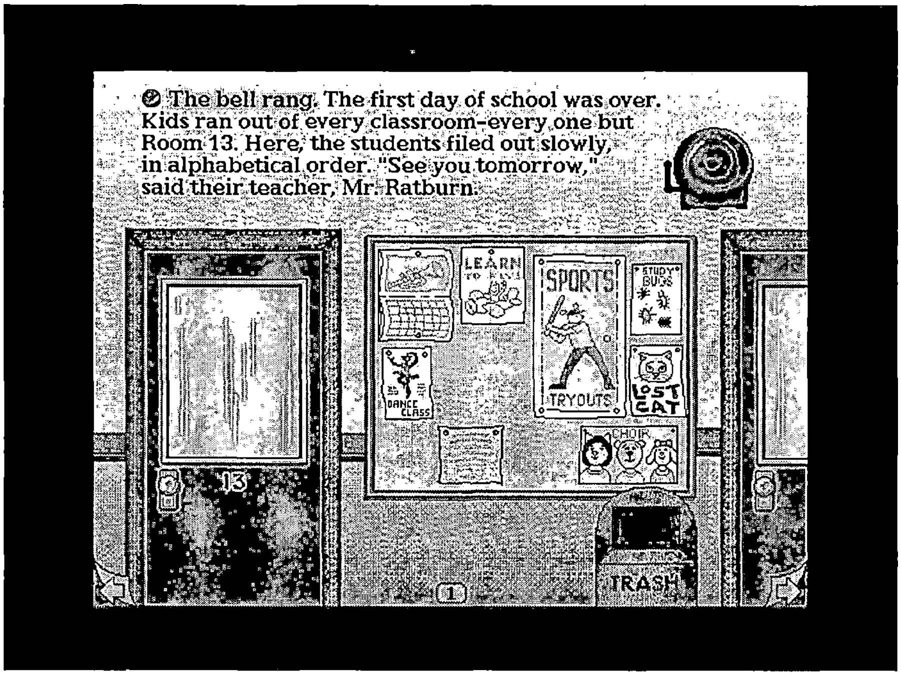
> "Information-oriented graphic design in user interfaces seeks a balance between the practical necessities of information management and the esthetics of presenting text and graphics to the reader. The graphics design of a program or multimedia document is built up through the systematic use of standardized interface elements such as symbols or icons, and interactive screen elements such as windows and buttons." (Lynch, 1994, paragraph 7).

It is these topics listed, visual, sound, etc. that the developer is accountable for implementing when developing software. Aesthetic content, having "well organized and consistent with principals of visual design" (Apple, 1998), should be what the software designer should strive for.

One basic design issue regarding software design is to keep the presentation (not necessarily the content) clear and simple. If a user is presented with too many diverse actions which can be undertaken at one time they will become confused. Too many colors, images, will eventually be ignored, or not used by the user. An easy rule to remember in multimedia design is that in many instances "less is often more" (Haykin, 1994, p.128).

The software design should be consistent. Consistent design leads to software which is easy to use and builds trust in the software with the user. Interactions on all screens should be similarly implemented if possible. Design that is not consistent will lead to confusion, whereas "simple and consistent design paves the way for understanding and interaction" (Haykin, 1994, p.128). Consistency also allows users to transfer knowledge and skills they may have between applications. This in turn creates software which is easy to use because the user is already prepared with background knowledge about how it works. The software should also be consistent within itself, from screen to screen things should work similarly. If this software is an upgrade from a previous version it should retain consistency from the original version. The software should also remain consistent to traditional operating system interfaces (Apple, 1998). In striving for consistency software should have a theme which runs through the software. There should be a definite style to the graphics. The theme and style should be linked to what the users will find as being similar to the ideas being presented in the software (Kristof and Satran, 1995).

A demonstration of consistency could best be demonstrated by Broderbund's Living Books series of software. One such title, Arthur's Teacher Trouble is presented in a story format where the user moves from page to page listening to and playing within a story. This product demonstrates both consistency within itself and consistency within other versions of the same software. When starting Arthur's Teacher Trouble (Broderbund, 1992) the user is presented with a menu screen allowing them to choose between playing within the story or having the story read to them. From this main screen the user can also change the language or quit. As is shown in figure 4 within Arthur's Birthday Party (Broderbund, 1992) on each screen in the lower right hand corner is an arrow pointing to the next card, or part of the story. In the lower left hand corner of the screen is an arrow pointing back to the previous card, or most recently previous part of

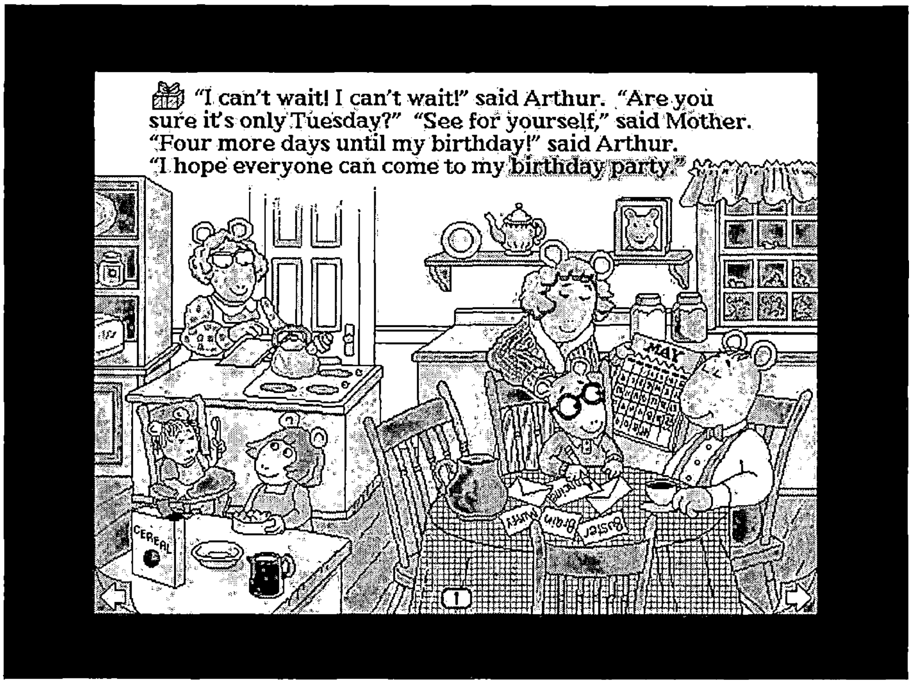


The bell rang. The first day of school was over. Kids ran out of every classroom–every one but Room 13. Here, the students filed out slowly, in alphabetical order. "See you tomorrow," said their teacher, Mr. Ratburn.

Figure 4. Screen from Page One of Arthurs Teacher Trouble (Broderbund, 1992)

the story. In the very center of the page is the page number, which also functions as a button back to the main menu. These buttons are located in the same place within each card.

As was mentioned above the Broderbund Living Books provide consistency with other products within the same line of software. When viewing another Living Book, Arthur's Teacher Trouble (Broderbund, 1992), shown in Figure 5, the user will find that the programs operate similarly. The main menu looks exactly the same and each button on the main menu performing the same



"I can't wait! I can't wait!" said Arthur. "Are you sure it's only Tuesday?" "See for yourself," said Mother. "Four more days until my birthday!" said Arthur. "I hope everyone can come to my birthday party."

Figure 5. Screen from Page One Arthurs Birthday Party. (Broderbund, 1994)

operation as on the previous software. Each page has the same arrows in the same location to navigate to next and previous cards. The page number button in the middle bottom of the page operates the same way by taking the user back to the main menu.

Software should be engaging for the user. It should hold their interest and create interactivity between the user and the software (Haykin, 1994). Software should be fun for the user to use. If it is not fun they most people will not continue to use the software. The developer needs to evaluate who the end user is and find activities which those users will find fun and engaging, yet still meet the educational criteria of the software under development (Haykin, 1994, p.40).

Content

The content of software includes all text, graphics, video, activities, and etc. that is to be included within the software. It is this content that comprises the educational focus of the software.

Even though the content of the project is important and most of the times thought of as a starting point for creating software it is not always the best starting point. The design process should be started with the goals that the learner should accomplish while interacting with the software (Kristof and Satran, 1995). After having determined the goals of the software project then this will determine which content, including text, graphics, activities, and etc. will be most effective to include in the software. The goal of the software may determine how the software is structures and the layout of the screens. By determining the goal ahead of time a lot of time can be saved by creating the software right the first time instead of redoing it. Each part of the software should have an objective and contribute to the overall goals of the software in some manner. If you cannot determine the goal of any section of the software and how it functions to promote the goal them it should be removed or redesigned (Apple, 1998)

There are many places that content can be acquired from. Sometimes it is necessary and preferable to have new content designed for a project. In creating new content the designer then has ownership over this material which often makes it easier to retain creative control over software. Having ownership of the content is often a benefit when it comes to software distribution. Often times though there is already existing material which my fit the content requirements of a project currently under development. If this is the case a designer my wish to repurpose the content. Repurposing content may be simply using content which was used in a previous version of software, using clip art libraries, etc. Although repurposing content alleviates the costs of creating new content there are often additional time and costs involved in licensing the necessary content (Haykin, 1994).

The designer of the content needs to verify that the content that has been chosen or created is of good quality. It is also the designers responsibility to check that the content is accurate and current (Haykin, 1994).

Navigation

Navigation is an extremely important aspect which needs to be well thought out during the design of the interface. If flawed, the basic navigation of the software as was shown above by Thuring, et al.(1995), can lead to cognitive overhead, thereby creating software which is more difficult and detracting from the educational content and goals of the software.

To reduce cognitive overhead, navigation tools should allow the user to know where they are at while interacting with the software (Haykin, 1994). In the Living Books series by Broderbund this is done by putting a page number on the bottom of each page. The page number allows the user to have a sense of where they are at in the story.

The user should never have more than three jumps to get to the information they are looking for (Haykin, 1994). This is also mentioned by Apple (1998) that information should not be buried within the program. Apple (1998) also recommends no more than three jumps to get to the information they are looking for. Effective navigation will create the shortest path between where the user is and where they want to go. The navigation of the software should not be to complex and allow to many ways to accomplish the same task.

This will lead the user to be confused in how to access the software (Kristof and Satran, 1995). Navigational buttons or graphics should be created appropriately to enhance the users interactions with the software. The user should easily be able to navigate to the next section of the software. Navigational buttons and graphics should be consistently located on the screen. This creates software which is easier to use and provides a sense of stability. (Apple 1998). By keeping consistent with navigation the designer can lower the cognitive overhead in the software being developed.

This is navigational style is represented in Figure 6 which is a map demonstrating the structure of the software Amnesty Interactive. As is shown in the map once the user is at the topics screen, which is also the main menu, they are only one mouse click away from any screen. Once the user selects a section all interaction takes place on the new screen. As an example the section rights portrays the 30 articles of the Universal Declaration of Human Rights. Once in this section the screen never changes, only the video and graphical text changes to reflect the users choice. Software such as Thinking Things 2 (Edmark, 1997) also employ a method by which the user is only one jump away

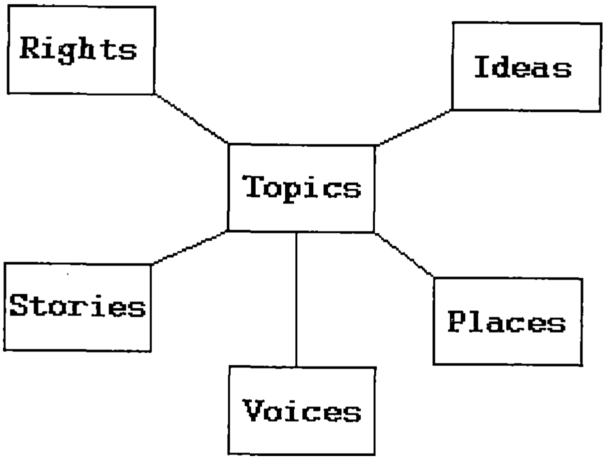


Figure 6. A Design Map of Amnesty Interactive Software

Figure 7. Main Menu from Thinking Things 2 Software

from each activity within Thinkin' Things 2. Each one of these images will take the user to a different activity. Fig 7. shows the main menu of Thinkin' Things 2. This software is presented in only six different screens, one for the menu and one for each section of the software.

However this is not the only model for software design. Actual structure will vary depending upon the needs of the software. Berenguer (1994) presents many different models of software design. He also presents software which follows a time sequence path would move the user in a linear motion from one screen to the next screen, almost in a style like a storybook. Software like this is most commonly seen in the current market of animated storybooks. Such example as Broderbund's (1992) Arthur's Teacher Trouble are demonstrations of software which follows a time sequence path. The user starts on screen one and either after a predetermined time or upon a mouse click will be moved to the next screen. This would also be very representative of the circular model design. In this model the user simply moves through the software in a circle, with no beginning or ending (Berenguer, 1994).

In the User Constructive model each user is provided the option of adding to the software, thereby creating new information / sections within the software for other users to explore. Other models mentioned include branching software in which the user follows a path of branches down through the software. Sometimes these models are mixed together within software depending upon the needs of that certain part of the software (Berenguer, 1994).

The software should be forgiving and allow the user to undo any action. It should be easy to return to the previous point they just left (Haykin, 1994). This provides a feeling of comfort while working with the software, thereby encouraging exploration. Software should not cause any irretrievable data loss without first prompting the user (Apple, 1998).

Effective use of navigation should motivate and encourage users to explore the software. What the user is presented with should be representational and clear. It should make interacting with the software easy and intuitive for the user. The navigational controls should allow the user to have the ability to access the portions of the software they want to go to (Kristof and Satran, 1995).

Graphics

Every designer of hypermedia software is going to use graphics for many different tasks during the creation of software. Graphics can serve many purposes if used properly, but can also create problems with completing the goals of the software if used improperly.

Representational graphics would be used to present an accurate picture or drawing of an item. Representational graphics are most powerful if the user already has preexisting background knowledge about the item. In the case that the user has not knowledge about the item it is best to use analogical graphics, which show similarity between two or more concepts (Reiber, 1994). An example of an analogical graphic would be used in teaching a student who had never seen an airplane about airplanes and starting by showing pictures of birds (analogical graphics) and then moving to pictures of planes (representational graphics). A third type of graphic would be an arbitrary graphics, such as a chart or diagram. This type of graphic is not similar at all to what it is representing, but is typically used to illustrate relationships between items.

When graphics are used there are five predominant uses for these graphics in hypermedia design. These five being cosmetic, motivation, attention getting, presentation, and practice (Reiber, 1994).

Cosmetic graphics appear simply to make the software appear complete or finished. Sometimes these graphics can also fall into the motivational category by getting attention drawn back to the program. However they can occasionally cause errors in the learning process if the graphics are not truly representational, or even worse, misrepresentational, of something on the screen (Reiber, 1994).

Motivational graphics provide motivation for the student to continue on with the software. These could be graphics which appear when they get the correct answer, finish a task, etc. However it is necessary to remember that these types of graphics effects will eventually wear off as the novelty of the graphic wears off. Motivational graphics can also be curiosity raising and provoking to entice the user to continue through the software (Reiber, 1994).

Graphics can also be used for getting or maintaining a users attention while interacting with a software program. An attention getting graphic could be a graphic which is related to the material, but does not add to the understanding of the content. One important point to remember is to make sure that the graphic does not take attention away from the expected goals of the software (Reiber, 1994).

Presentation graphics are used to more adequately present information to the user. These graphics can either be used by themselves or in coordination with text or narration. These types of graphics most often appear in software as representational graphics which are presented with text accompaniment. The use of good graphics help a user to create a mental model of the content contained within the software (Reiber, 1994).

Graphics with are closer to each other on the screen will be perceived by the user as being put together for a reason. Proximity will imply that the graphics have some meaning to each other and should be viewed as a group. Graphic which are similar is the same theory except that in this case the graphics actually look similar. This can be a method that a designer can use to provide cues for users as to what certain graphics and buttons might do (Reiber, 1994).

Graphics can often be used as metaphors for both learning and navigational purposes. When used as graphics are used as metaphors to express navigational options the designer must make sure that the graphic helps the user understand what actions can be undertaken. It must also be appropriate for the content of the software. A graphic must also be careful not to create unreasonable expectations for the user to expect of the software (Kristof and Satran, 1995, p.41).

"Metaphors serve as natural models; they allow us to take our knowledge of familiar objects and events and use it to give structure to abstract, less well understood concepts" (Erickson, 1990 pg. 73). Metaphors allow the user to "understand and experience one kind of interaction in terms of another more familiar kind" (Haykin, 1995, p.131). Furthermore the designer should base metaphors used on the real-world experience that the users may have. These metaphors should be clear and convey expectations that the user can use in the context of the software. Metaphors used should be consistent throughout the software. Metaphors should be able to allow the user to not have to remember many rules about the software. The interactions with the metaphor should be self-evident to the user (Lynch, 1994).

Graphics should be carefully chosen to avoid cultural misrepresentations. Often times graphics can have a negative meaning to people from other cultures. Apple (1996) recommends specifically avoiding graphics dealing with seasons and holidays, unless the software is specifically addressing one of these topics. It is important to know who the software is being designed for to avoid the cultural misrepresentations of graphics.

Animation / Video

Typically a developer will add animation within software for a couple of different reasons. Many times the animation is simply to get attention or for cosmetic purposes. Other times though the animation can be used as the focus of content presentation, or as a feedback in practice or simulations. According to Reiber (1994) early research into the use of animation / video sequences in software did not yield any additional learning for the user. Other early research although demonstrated a completely opposite conclusion. It was found that when viewing video sequences of procedural tasks the users viewing the video were able to demonstrate an increase in learning. However the same study did show that if static graphics were used in place of the video the same learning occurred. Further research, although showed that by using video / animation learners were capable of demonstrating greater recognition memory. The most recent research has shown that students viewing animation / video demonstrated more learning than those with static graphics or no graphics at all. For the animation / video to be effective though it needed to be accompanied by an audio narration (Reiber, 1994). The research on video use has been one which as been very inconclusive, however newer research is leading to conclusions that the use of animation / video does increase student learning.

Kristof and Satran (1995) mention many good reasons to use video within a multimedia production. Video is perfect to be used to provide a message which is best if the message comes from a specific person. Video works good from showing live action or demonstrating the functionality of items. Video provides a way to show multiple views of an object which might lack from being presented using a static graphic.

Sound

Sound is a very important component of multimedia production. Careful use of sound can enhance the users experience and provide for easier to use software. Effective use of sound can add to the motivation and help to keep learners engaged with the software. Sound can be used as feedback for users to know that something has happened, either positive or negative. Often time sounds used within multimedia productions are music. Music is most often used as an enhancement to the software. Music can be used to provide action while something else is happening which the user cannot see, i.e. loading other parts of program, etc. Music can be used in certain areas of software to enhance the content by producing an emotional response to coincide with the content. (Haykin, 1994). However the sound should not be the only focus because the user may have sound turned off or not have sound on the computers (Jimmieson, 1994).

CHAPTER THREE

Review of Human Rights Educational Literature

Human rights, according to Amnesty International (1998), can be defined as the basic rights which people are "…entitled to simply because they are human beings, irrespective of their citizenship, nationality, race, ethnicity, language, gender, sexuality, or abilities…" (p. 134). These rights are divided into three different areas. They define one section of human rights as civil and political rights. Within this group of rights they have described the rights of individuals to the freedoms of religion, speech, political activity, etc. The second set of rights they describe is economic, social and cultural Rights. Within these rights they have listed the rights to a have a cultural identity, economic security, etc. The third set of rights they describe are environmental, cultural, and developmental rights. These rights are described as giving all individuals the right to a safe living space, the ability to participate in politics, etc. The colloquial version of the Universal Declaration of Human Rights is presented in Appendix B.

It is imperative, according to Starkey (1991), that students acquire knowledge in the area of human rights. He states that "the understanding of human rights is an important element of the preparation of all young people for life in a democratic and pluralistic society" (Starkey, 1991, p. 22). At the most essential level, human rights education, is supported directly by the State of California in the History - Social Studies Framework for California Public Schools (1997). According to the State of California (1997), students should have knowledge of historic documents, including the United Nations Universal Declaration of Human Rights and also students should "Realize that concern for ethics and human rights is universal and represents the aspirations of men and women in every time and place" (State of California, 1997, p.14).

These thoughts about human rights education are further supported through current research by Human Rights U.S.A. (1997). In summarizing their research they stated that:

> "At the approach of the fiftieth anniversary of the Universal Declaration of Human Rights, this international document that sets forth the fundamental rights and freedom to which all human beings are entitled is unknown to the vast majority of Americans." (Human Rights U.S.A, 1997, paragraph 5).

From their research, Human Rights U.S.A. also concluded that most Americans agree that human rights issues should be dealt with in method which is "careful and deliberate, that calls upon both government and individuals to play a part and that includes public awareness and education" (Human Rights U.S.A., paragraph 2, 1997).

According to Amnesty International (1996) an effective human rights education program should have a focus on the Universal Declaration of Human Rights. The Universal Declaration of Human Rights has been defined as the document which "spells out the meaning of the fundamental rights and freedoms proclaimed within the Charter" (Amnesty International, pg 6. 1998). Starkey (1991) further reflects on the basic goals of human rights education when he states,

> "Texts remain dead letters unless and until they are known and acted upon. The challenge for human rights educators is to ensure that useful official texts are known and that they are accompanied by sufficient practical help for them to be acted upon. (Starkey, 1991, p. 20).

The ultimate goal of human rights education, as represented by many human rights educators and organizations, including Ferneke (1992), Action Professionals' Association for the People (1996) and Reardon (1994) is the empowerment of the learner; the ability of the learner to be capable of enacting social change. Amnesty International (1996) defines the goal of a human rights education program would "challenge students to ask what human rights means to them personally and encourages them to translate caring into informed, non-violent action" (pg 5).

According to Lyseith-Jones (1991) to promote human rights education in the classroom the teacher would want to provide an environment which would allow for free discussion within the classroom. She states that the teacher would also want to develop a classroom atmosphere which allow the students to develop a trust within the classroom and would also help them to develop skills for conflict resolutions. She describes this classroom as having a focus on problem solving techniques that are not teacher focused. Another important aspect she points out is that one of the goals of this classroom would be to help students to develop responsibilities to those outside of the classroom group and to help prepare the student to take action.

Starkey (1991) describes some of the components of an effective human rights education program should have according to the Council of Europe. He first states that there should be a focus on communication skills, particularly written and oral communication, with these skills helping the student to develop the ability to defend their opinions. The next skill he describes is the ability to gather information and be able to analyze the information and arrive at a conclusion, including being able to identify any forms of prejudice, bias, etc. contained within the information. He also believes that students need to be able to develop social skill, with a focus on the differences between individuals, with the students being able to understand these differences and accept them. He states that they need to be able to establish "positive and non-oppressive personal relationships" (Starkey, 1991, p. 27). Other interaction skills that he feels should be focused on are for the students to be able to take responsibility and make decisions, and to also learn non-violent methods of conflict resolution. Finally he focuses on the study of human rights, human rights documents, duties, obligations and responsibilities and how these affect the student, the country and the world in general.

Action Professionals' Association for the People (1996) present human rights education as being a process of education which enhances the students knowledge of human rights issues. They also want the student to develop a critical understanding of issues. They promote students working on issues dealing with values clarification and attitudinal changes. They present all of these working towards the final goal of empowerment of the student to be capable of working towards social change.

Human rights education, according to Reardon (1994) should be approached from a reconstructionist viewpoint, with the ultimate goal to be the development of the learner to be able to become involved in social change. This should be undertaken according to the objectives set forth within the People's Decade for Human Rights Education by having all people have knowledge of the rights set forth within the UNIVERSAL DECLARATION OF HUMAN RIGHTS, and having the ability to see the problems that prevent human rights and what can be done about them (Reardon, 1994).

In human rights education an educational design (which can be applied to software), would "facilitate students defining themselves as connected and interdependent with others" (Jennings, 1994). The students should be taught in an atmosphere which places focus on the students, has open dialogue, dialogue between students, and allows the students the ability to choose topics of interest (Jennings, 1994).

The basic overriding goal of any human rights education program can be summed up by Fernecke s (1992) when he states that "the development of an enlightened, participatory citizen requires that young people be capable of effecting change" (p. 24).

CHAPTER FOUR

Basic Design Issues

This section demonstrates the use of effective software development as applied to the development of a particular title of software. The software "You Can Make a Difference … Human Rights" was developed in conjunction with this paper to provide a demonstration of some of the concerns with software development. "You Can Make a Difference … Human Rights" is an educational software title developed to teach students about human rights.

Goals of the Software

The first step in developing software, as was pointed out by Kristof and Satran (1995) is to determine the goals of the software. In developing "You Can Make a Difference … Human Rights" it was first determined that the software would be used by students in grades four to eight (ages eight to thirteen). The primary goals of the software are:

1. to introduce the students to the Universal Declaration of Human Rights.
2. to help the students to develop an understanding of human rights and human rights issues.

3. to encourage the students to take action in human rights issues.

These goals are all consistent with the goals of developing an effective human rights education project as discussed by Amnesty International(1996), Fernekes (1991), Starkey(1991), Human Rights U.S.A.(1997), Jennings(1994), etc. When presenting information one goal of the software would be to present human rights in an engaging yet non-threatening manner to the students. Another goal of the software is that it should help the students to start to develop a sense of interconnectedness between themselves and other children around the world.

Content

After defining the goals of the software available content was evaluated in relation to creating the software which would meet the goals set forth for this project. Amnesty International granted use of their resources for the development of this software. This allowed access to a great wealth of material which could be repurposed to save both costs and development time. Some of the important content which would be repurposed for use within "You can Make a Difference … Human Rights" were:

1. Video - Universal Declaration of Human Rights: Colloquial Version. (Amnesty International, 1986)
2. Video - Think About Tomorrow… Amnesty International's Student Action. (Amnesty International, 1991)
3. Video - Colors of Hope. (Amnesty International, 1985)
4. Many pictures from Amnesty International books, magazines, website, etc which would be used to comprise graphics found throughout the software.
5. Amnesty International Urgent Action reports. (Amnesty International).

Overall Design Strategy

"You Can Make a Difference … Human Rights" uses a constructivist design strategy. The constructivist design strategy was chosen because there are many parallels between human rights education and constructivism. The software is designed to allow the student to work with each right along with stories about the right and in context of larger issues. This allows the user to work with each right in multiple settings, allowing for reconstruction of their knowledge about the right; see Reiber (1994). In keeping with the constructivist design strategy, the software is self-regulating and is designed to be intrinsically motivating for the user. All of the content is presented in an exploratory manner with no skill and drill (instructivist) content. All input from the student is never judged as being correct or incorrect, nothing has a predefined right answer. The software's main input from the user is a writing screen where the user responds about each right. This allows the software to value the students opinions and ideas and create additional content based upon the student's prior knowledge. The software follows the design strategies of Learner Centered Design (Soloway, 1996) by promoting the rights towards all regardless of race, gender, age, religion, etc and reinforcing and focusing on the diversity found within the classroom. "You Can Make a Difference … Human Rights" also utilizes the constructionist design strategy which has been described by Papert (1993, 1996). Within this software there is a section which allows the students to write about the right. This allows the student to construct something while using the software. Each section is saved and becomes part of a collaborative document of student created viewpoints about each specific right. This collaborative document is viewed from within the software while interacting with the right. As was mentioned by Evard (1994) allowing the students to create a shared document will help enhance student learning.

The overall design strategy of "You Can Make a Difference … Human Rights" is to present all of the content while using a minimal amount of different screens. This is similar in presentation to the software Thinking Things 2 (1995) and Amnesty Interactive (1994). By presenting the software using a minimal number of screens, cognitive overhead can be lessened. Keeping the design clear and consistent will make it easier for the to navigate without getting lost, see Haykim (1994). This will encourage exploration and helps to enhance learning. Figure 8 shows a map of the layout of "You Can Make a Difference

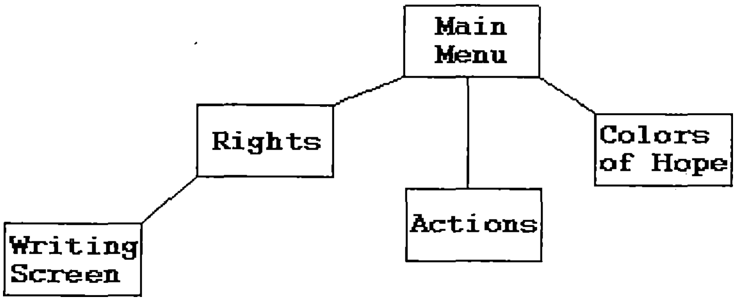


Figure 8. Map of "You Can Make a Difference … Human Rights".

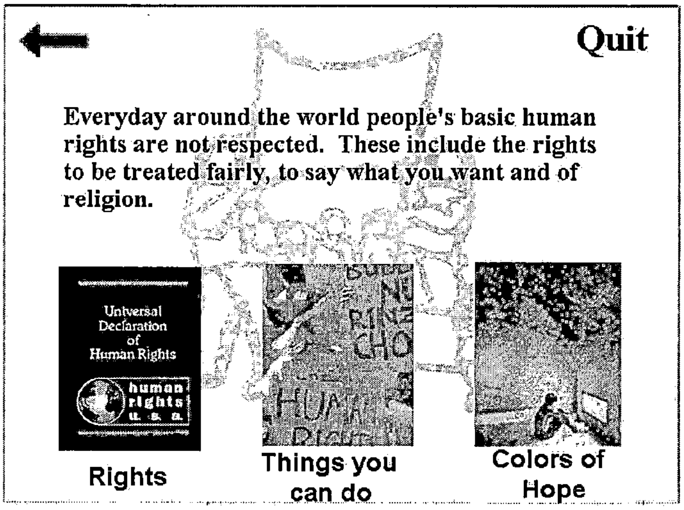
… Human Rights". As is shown by the software map, all screens are at the furthest two hops away from each of the other screens.

"You Can Make a Difference … Human Rights" starts with an introduction that has three pictures of children from around the world. These pictures are of a young girl from India, a boy from Thailand wading in a river with a fishing net, and a girl from Guatemala, her head wrapped in a green shawl. During the introduction while these three pictures are being show pan flute music is playing. These pictures and music were chosen for the introduction because they were stunning photos of children from around the world. The photos (graphics), are basically cosmetic in function, however in some instance may provide motivation and are also used to help set the tone and mood of the software, see Reiber (1994) and Kristof and Satran (1994).

After the last picture the screen fades to black and the words "You can Make a Difference…" appear on the screen. A few seconds later underneath these the words "Human Rights" appear. The title in it entirety being "You can Make a Difference … Human Rights" was chosen with certain goals in mind. The title was chosen with the primary focus of having a title with an empowering name. It needed to be catchy, yet still set the idea to the user that they can make a difference, even as young as they might be. The font for this was specifically chosen to look like a child's handwriting to further add to the tone of the software as being software for children.

Main Menu

The main menu is the menu where the user selects which area of the software they are going to explore. Figure 9 shows the main menu and the options



Everyday around the world people's basic human rights are not respected. These include the rights to be treated fairly, to say what you want and of religion.

Figure 9. Main Menu of "You Can Make a Difference … Human Rights"

available to the user. In the background there is a simple gray line drawing. This graphic is simply a cosmetic graphic, it's only purpose except to is to create a full looking screen. The three main graphics

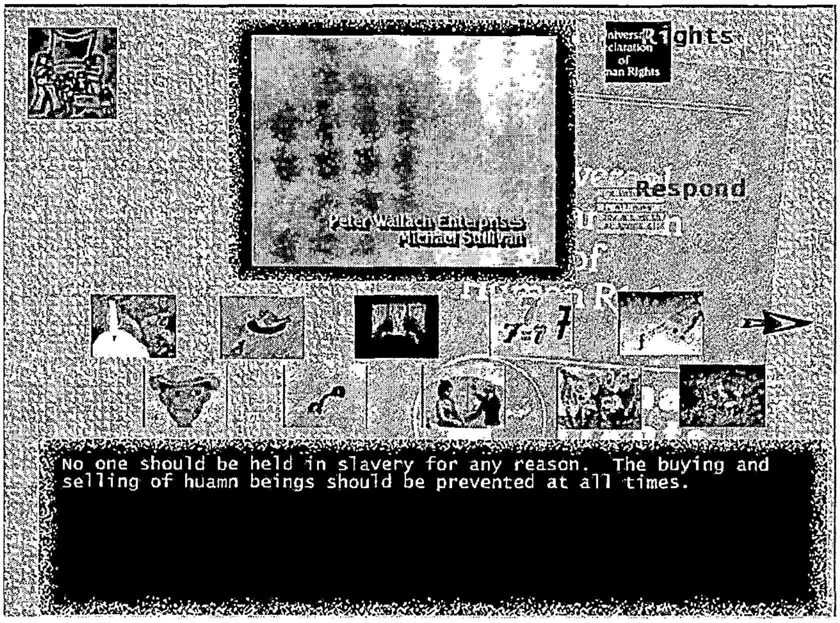
(Human Rights, Things you can do, and Colors of Hope) are set larger than the other icons (arrow and quit)to attract more attention to them. These graphics when rolled over change the text and narration in the middle of the screen to represent what that particular section is about. Appendix A lists the narrations for the main menu. This accomplishes many tasks, it first provides feedback to the user in a couple of different ways. The narration is provided to give the user an idea about the content of a particular section. By providing the user with navigational clues, cognitive overhead can be reduced, thereby encouraging exploration and enhancing student learning, see Thuring (1995). It also provides the user feedback that the software is still functioning properly and has not locked up. The graphics used in this section were chosen to be a representation of the area that by clicking on the user is taken to. The book is taken from the handbook of The Universal Declaration of Human Rights by Human Rights U.S.A. (1998). This booklet is also available for free (at the time of this writing), giving the user something to further learn from. The graphic of the protest (which is a link to the section "Things you can do") is taken from Think about Tomorrow … Amnesty Internationals Student Action (1991). The third section, Colors of Hope, uses a section of one of the video clips that are used in the story. The protest picture in the background is representational of types of actions that are frequently used in human rights work. This graphic is taken from Human Rights Here and Now (Amnesty International, 1998).

The small graphics at the top of the screen have different actions. The arrow in the corner takes the user back one section and replays the introduction. On all screens within "You Can Make a Difference … Human Rights" the graphic located in the corner is used to take the user back to the previous area they were last at. This provides a consistent navigational structure for the user. Using a common graphic located in a similar place on all screen will also help reduce cognitive overhead and create easier to use software as mentioned by Thuring et al. (1995) and Isakowitz (1995). By using this graphic the user can simply get back if they feel that they are lost, or if they clicked on the wrong icon thereby encouraging exploration within the software.

Universal Declaration of Human Rights Screen

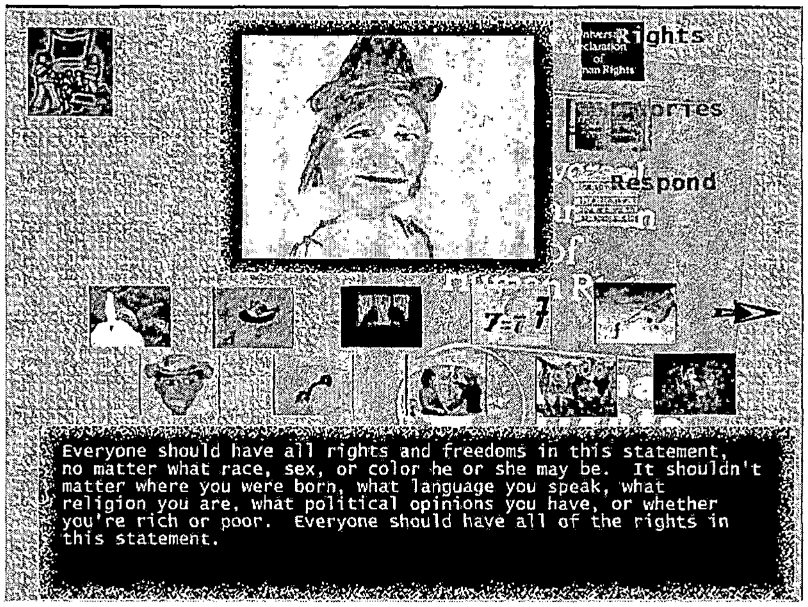
One of the main screens contained within the software is the Universal Declaration of Human Rights screen. This screen handles the display of all thirty rights and narrations. Also associated with these rights are the stories of human rights violations against children. Figures 10 and 11 show the rights screen at different stages of user interaction.

Figure 10 shows the rights screen as it may be viewed when a user is interacting with the software. In the upper left hand corner, as it is on all of the screens is the background graphic from the main menu.



No one should be held in slavery for any reason. The buying and selling of huamn beings should be prevented at all times.

Figure 10. Screen Shot from "You Can Make a Difference … Human Rights" Without Story Icon



Everyone should have all rights and freedoms in this statement, no matter what race, sex, or color he or she may be. It shouldn't matter where you were born, what language you speak, what religion you are, what political opinions you have, or whether you're rich or poor. Everyone should have all of the rights in this statement.

Figure 11. Screen Shot from "You Can Make a Difference … Human Rights" With Story Icon

This graphic when clicked on will take the user back to the main menu (the last section visited). As can be seen in Figures 10 and 11 the background of this section contains the picture of The Universal Declaration of Human Rights booklet. This provides a direct connection with the graphic contained within the main menu. Providing these links within the software makes it easier for the user to develop relations between what certain graphics do and where they will be taken when clicking on that particular graphic. By providing clues as to the basic location within the software cognitive overhead can be lessened (Thuring et al., 1995).

The screen is separated into distinct sections, with each section having a task. The first section (the top section of the screen) contains the videos and actions that can be undertaken while interacting with each right. The second section (the center section of the screen) contains the icons that represent the rights. It is these icons which when clicked on will change the video and actions contained in the top section of the screen. The third section (the bottom of the screen) contains the text that accompanies each video.

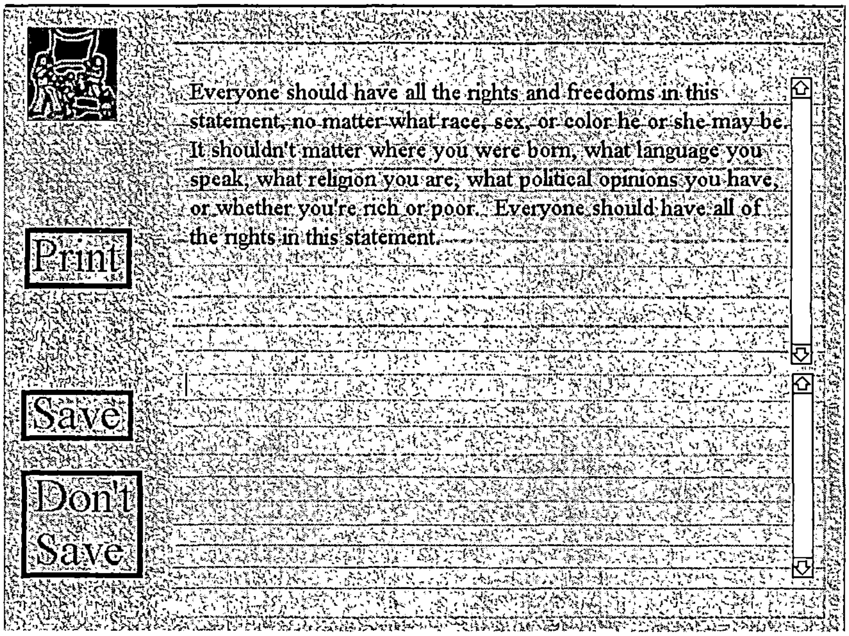
Within this section all of the videos play at the same location on the same screen as is shown in Figures 10 and 11. To the right of the video screen are three icons. These three icons are placed using the theories of graphical proximity, see Reiber (1994) and Rettig (1992), next to each other and next to the video. Placing these graphics close together (proximally, see Reiber, (1994) provides the user with a sense that these three icons work together. These three graphics are actions that allow the user to explore different content areas associated with the right they are learning about. The first graphic is a corner clipping of The Universal Declaration of Human Rights booklet with the words "Human Rights" written across it. This graphic replays the right so that the user can view it over and over if they choose. This graphic portion of the booklet is used as a metaphor to represent the right to the user. The second graphic is of a book and has the word "Stories" written across it. This graphic is also presented metaphorically to the user to represent the individual stories of human rights abuse that are associated with each right. This graphic, when clicked on, replaces the video of the human right with a video of a story of a human rights violation against a child. As can be noticed in the difference between Fig 10 and Fig 11 this graphic does not appear with every right. The third graphic is a section of a yellow piece of paper. This graphic is from a section of the graphic contained in the "response" section of the software. If the user clicks on this graphic they will be taken to the writing screen. Providing these as metaphors allows the user to more quickly develop skills about what each section of the software does, see Norman (1993), Erickson (1990) and Lynch (1994).

The second part of the screen (the center of the screen, contains the video icons that are clicked on by the user to select a video. Originally "You Can Make a Difference … Human Rights" had all thirty icons on one screen in an attempt to keep all interactions on one screen. In this state the software was very cluttered, confusing and difficult to use. The thirty icons were eventually replaced by ten on each one of three screens. Each icon is a section from the screen of the video. The choice to use small icons to represent each video was done to allow the user to more easily be able to remember what each right was. The user can change from screen to screen by clicking on the arrow/arrows pointing up and down. When the user clicks on an arrow the only change in the screen are the ten icons and the adding or removing of the arrows. This still allows the software to provide the image to the user that all interaction is taking place on one screen. The arrows allow the user to go side to side for a total of three screens. The arrows changing can be seen in Figures 10 and 11. As the user moves down through the rights only the appropriate navigational arrows remain visible on the screen. This provides a sense of beginning and end to the rights to limit the confusion of where the arrows may take the user to.

The third section of the screen contains a graphic which contains the text of the right which is being learned about.

Writing / Response Screen

This section is non - functional on the CD contained with the thesis. The writing response screen is divided into two basic section. The large section on the right simulates a piece of paper. This is used metaphorically to help build in a sense that the student will be writing within this section. The graphics are basically cosmetic and motivational in



Everyone should have all the rights and freedoms in this statement, no matter what race, sex, or color he or she may be. It shouldn't matter where you were born, what language you speak, what religion you are, what political opinions you have, or whether you're rich or poor. Everyone should have all of the rights in this statement.

Figure 12. Writing Screen from "You Can Make a Difference … Human Rights"

its use within the software. Within this section the student can type into the field on the bottom of the screen. Whereas the field on the top of the screen has the text of the right within it.

The second part of the screen is the small section to the right of the paper. This section has four actions which the user may undertake. In the upper left hand corner is the standard square picture to return back to the section which is right above this section (the rights section). Below this is a button with the word "print" on it. If the user prints from this area they will print the text in the upper field (the text of the right they are currently working with) and whatever they have typed in the bottom screen. Below this are two buttons, one with the word "save" on it and the other with the words "don't save" on it. These two buttons allow the user to save their writing or to trash their writing. When the user saves their writing it gets saved into a file specifically for that right. That way the next user, when interacting with that right, can read all of the writing of the previous user. This section utilizes the constructionist viewpoint of educational design and software, as have been described earlier by Papert (1993, 1996) and Evard (1994). This section allows the user to create a document within the software, and enables them to be able to share it with other students. This section creates an internal dialogue about human rights with the students. It also provides a method of assessment for the teacher.

Colors of Hope Screen

The colors of hope section is designed around an Amnesty International video entitled "Colors of Hope". This video presents the story of a human rights activist who was jailed in Argentina for the action of forming unions. The story contains references to many of the articles of the Universal Declaration of Human Rights. The difference between this section and the others is that this section is presented in a linear

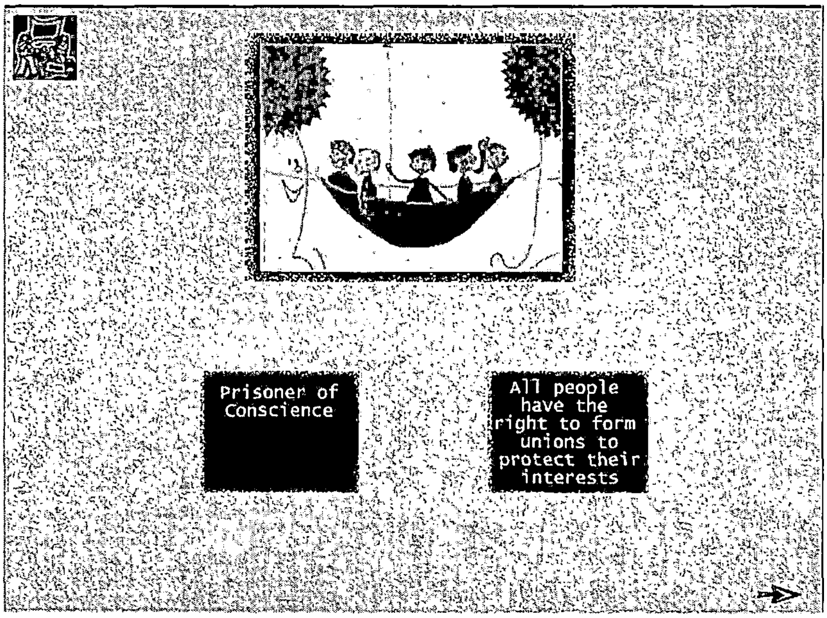

Prisoner of Conscience
All people have the right to form unions to protect their interests

Figure 13. Screen Shot from Colors of Hope Section of "You Can Make a Difference … Human Rights"

manner. The screens of this section are all similar to the one shown in figure 13. As can be noticed in the upper left corner of the screen is the square icon which links back to the previous page. The background graphic is a picture of the subject of the story in a cell in Argentina. This graphic has been faded out to provide a non-distracting background, and is purely cosmetic in function. The other navigational graphics appear in the bottom corner of the screen. These arrows allow the user to change pages either forward or backwards. In the center of each screen are videos that portray the story. Below the pictures are terms that are being mentioned during the current narration. These terms allow the user to get more information about these terms or concepts. When the user clicks on the information they will be taken to a section where they can explore certain concepts from within the story. Within these sections the users can interact with other videos about each section of the main video to help reinforce the learning.

Final Design Considerations

The software "You can make a difference … Human Rights is set to provide a method for students to learn about the rights and communicate with others about these concepts. It is set up so that the student can advance through levels of development, as mentioned by Norman (1993) to allow the students to become experts about the rights. Because on of the main sections of interaction is with the students writing in a collaborative journal they should become more confident in their writings and move through the stages of accretion tuning, and restructuring. As the students become "experts" in working with the human rights content and other stories their writings should begin to develop in the amount of critical thought and knowledge about the rights in general. The rights are presented with stories, animations, etc to help keep an optimal flow (Norman, 1993) and keep the students engaged in working with the software.

APPENDIX A

NARRATIONS FROM THE MAIN MENU

Human Rights are basic rights which are guaranteed to all people. These rights include freedom of religion, the right to gather as a group and of legal protection.

Universal declaration of human rights icon rollover narration

In 1948 the Universal Declaration of Human Rights was written. This important document guarantees these basic rights for all people.

Action icon rollover narration

Sometimes peoples basic rights are violated. When this happens other people will take actions to help out the people whose rights are being violated.

Colors of Hope icon rollover narration

Human rights violations affect men, women, and children around the world. The story of Juan Carlos Rodriquez demonstrates many of these human rights violations

APPENDIX B

UNIVERSAL DECLARATION OF HUMAN RIGHTS TEXT

Article 1. All human beings are born free and equal. You are worth the same and have the same rights as anyone else. You are born with the ability to think and to know right from wrong, and should act towards each other in a spirit of friendliness.

Article 2. Everyone should have all the rights and freedoms in this statement, no matter what race, sex, or color he or she may be. It shouldn't matter where you were born, what language you speak, what religion you are, what political opinions you have, or whether you're rich or poor. Everyone should have all of the rights in this statement.

Article 3. Everyone had the right to live, to be free, and to feel safe.

Article 4. No one should be held in slavery for any reason. The buying and selling of human beings should be prevented at all times.

Article 5. No one shall be put through torture, or any other treatment or punishment that is cruel, or makes him or her feel less than human.

Article 6. Everyone has the right to be accepted everywhere as a person, according to law.

Article 7. You have the right to be treated equally by the law, and to have the same protection under the law as anyone else. Everyone should have protection from being treated in ways that go against this document, and from having anyone cause others to go against the rights in this document.

Article 8. If your rights under the law are violated, you should have the right to fair and skillful judges who will see that justice is done.

Article 9. No one shall be arrested, held in jail, or thrown and kept out of her or his own country for no good reason.

Article 10. You have the same right as anyone else to a fair and public hearing by courts that will be open-minded and free to make their own decisions if you are ever accused of breaking the law, or if you have to go to court for some other reason.

Article 11. 1) If you are blamed for a crime, you have the right to be thought of as innocent until you are proven guilty, according to the law, in a fair and public trial where you have the basic things you need to defend yourself.

2) No one shall be punished for anything that was not illegal when it happened. Nor can anyone be given a greater punishment that the one that applied when the crime was committed.

Article 12. No one has the right to butt-in to your privacy, home, or mail, or attack your honesty and self-respect for no good reason. Everyone has the right to have the law protect him or her against all such meddling or attacks.

Article 13. 1) Within any country you have the right to go and live where you want.

2) You have the right to leave any country, including your own, and return to it hen you want.

Article 14. 1) Everyone has the right to see shelter from harassment in another country.

2) This right does not apply in cases where the person has done something against the law that has nothing to

do with politics, or when she or he has done something that is against what the United Nations is all about.

Article 15. 1) You have a right to a country where you are from.

2) No one should be able to take you away from, or stop you from changing your country for no good reason.

Article 16. 1) Grown men and women have the right to marry and start a family, without anyone trying to stop them or make it hard because of their race, country, or religion.

2) A marriage shall take place only with the agreement of the couple.

3) The family is the basic part of society, and should be protected by it.

Article 17. 1) Everyone has the right to have belongings that they can keep alone, or share with other people.

2) No one has the right to take your things away from you for no good reasons.

Article 18. You have the right to believe, to have ideas about right and wrong, and to believe in any religion you want. This includes the right to change your religion if you want, and to practice it without anybody interfering.

Article 19. You have the right to tell people how you feel about things without being told that you have to keep quiet. You have the right to read the newspaper or listen to the radio without someone trying to stop you, no matter where you live. Finally, you have the right to print your opinions in a newspaper or magazine, and send them anywhere without having someone try and stop you.

Article 20. 1) You have the right to gather peacefully with people, and to be with anyone you want.

2) No one can force you to join or to belong to any group.

Article 21. 1) You have the right to be part of your government by being in it, or choosing the people who are in fair elections.

2) Everyone has the right to serve her or his country in some way.

3) The first job of any government is to do what its people want it to do. This means you have the right to

have elections every so often, where each person's vote counts the same, and where everyone's vote is his or her own business.

Article 22. Everyone, as a person on this planet, has the right to have her or his basic needs met, and should have whatever it takes to live with pride, and become the person he or she wants to be. Every country or group of countries should do everything they possibly can to make this happen.

Article 23. 1) You have the right to work and to choose your job, to have fair and safe working conditions, and to be protected against not having to work.

2) You have the right to the same pay as anyone else who does the same work, without anyone playing favorites.

3) You have the right to decent pay so that you and your family can get by with pride. That means that if you don't get paid enough to do that, you should get other kinds of help.

4) You have the right to form or be part of a union that will serve and protect your interests.

Article 24. Everyone has the right to rest and relaxation, which includes limiting the number of hours he or she has to work, and allowing for holidays with pay once in a while.

Article 25. You have the right to have what you need to live a decent life, including food, clothes, a home and medical care for you and your family. You have the right to get help from society if you're sick or unable to work, if you're older or a widow, or if you're in any other kind of situation that keeps you from working through no fault of your own.

Article 26. 1) Everyone has the right to an education. It should be free of charge, and should be required for all, at least in the early years. Later education for jobs and college has to be there for anyone who wants it and is able to do it.

2) The idea of education is to help people become the best they can be. It should teach them to respect and understand each other, and to be kind to everyone, no matter who they are or where they are from. Education should help to promote the activities of the United Nations in an effort to create a peaceful world.

Article 27. 1) You have the right to join in and be part of the world of art, music, and books. You have the right to enjoy the arts, and to share in the advantages that come from new discoveries in the sciences.

2) You have the right to get the credit and any profit that comes from something that you have written, made or discovered.

Article 28. Everyone has the right to the kind of world where their rights and freedoms, such as the ones in this statement, are respected and made to happen.

Article 29. 1) You have a responsibility to the place you live and the people around you - we all do. Only by watching out for each other can we each become our individual best.

2) In order to be free, there have to be laws and limits that respect everyone's rights, meet our sense of right and wrong, and keep the peace in a world where we all play an active part.

3) Nobody should use her or his freedom to go against what the United Nations is all about.

Article 30. There is nothing in this statement that says that anybody has the right to do anything that would weaken or take away these rights.

APPENDIX C

COPYRIGHT

The following details copyrights and ownership of content contained within You Can Make a Difference … Human Rights.

Graphics - Most graphics courtesy of Amnesty International. Graphics were either scanned from Amnesty International literature or taken from Amnesty International websites. All text boxes created specifically for this project. Introductory Pictures purchased from PhotoDisc.

Video - All video clips courtesy of Amnesty International. Video clips were taken from the videos "Think About Tomorrow" (1991), Colors of Hope (1985), and Universal Declaration of Human Rights - Colloquial Version (1988). Universal Declaration of Human Rights Colloquial Version video produced and funded by Reebok Corporation.

Music - Background music is "Lives in the Balance" by Jackson Browne. Closing music is "Fragile" by Sting. Both tracks taken from Amnesty International video "Think About Tomorrow. Backing tracks to stories taken from Ethnic Journey CD (freely usable sounds). Introduction music (Pan Flute) was taken from Director 6 Demystified book CD.

Narration - Narrations for stories taken from Amnesty International Urgent Action Reports. All narrations were created by William Chapman.

Programming - All programming and design content created by William Chapman

REFERENCES

Amnesty Interactive CD-ROM [Computer Software]. (1994). Irvington, N.Y.: Voyager.

Amnesty International. (1996) Children can be human rights advocates. Minneapolis:Human Rights Educators Network

Amnesty International. (1998) Human Rights Here and Now. Colorado:Urgent Action Office

Amnesty International. (1985) Colors of Hope. Available from Amnesty International.

Amnesty International. (1991) Think about Tomorrow. Available from Amnesty International.

Amnesty International. (1986) UNIVERSAL DECLARATION OF HUMAN RIGHTS Colloquial version. Available from Amnesty International.

Apple Computers (1997). Human Interface Principles. [On-line].Available: http://applenet. apple.com/hi/resources/resources.html.

Apple Computers (1996). Apple Web Design Guide. [On-line]. Available: http://applenet.apple.com/hi/ web/web.html.

Berenguer, X. (1994). Writing Interactive Programmes. [On-line]. Available: http://www.iua.upf. es/formats/art/a01at.htm.

Digital Chisel [Computer Software]. (1998). Pierian Springs Software.

Erickson, T. D. (1990). Working with Interface Metaphors. In Laurel, B. (Eds.), The Art of Human-Computer Interface Design (pp. 65-74). Reading, Massachusetts: Addison-Wesley.

Evard, M. (1994). Articulation of Design Issues: Learning Through Exchanging Questions and Answers. [On-line]. Available: http://mevard.www.media.mit.edu/people/mevard /papers/necc94/necc.html.

Fernekes, W. Children's Rights in Social Studies Curriculum: A Critical Imperative. *Social Education V.56 No. 4* 203-205.

Haykin, R. (Eds.). (1994). *Multimedia Demystified.* New York: Random House.

History - Social Science Framework for California Public Schools. (1997). California Department of Education: Sacramento.

Hollingsworth, J., Weide, B., Edwards, E. & Latour, L. (1995). Micro-Architecture of Software Components and the Need for Good Mental Models of Software Subsystems. [On-line]. Available: http://www.umcs.maine.edu/~ftp/wisr/wisr7/mmwg/hreport.html.

Human Rights Plain Language. (1990). Madrid: Editorial Amnistia Internacional (EDAI).

Iskaowitz, T., Stohr E., & Balasubramanian, P. (1995). RMM: A Methodology for Structured Hypermeida Design. *Communications of the ACM V.28 No.8* 34-44.

Jennings, T.(1994). Self-in-Connection as a Component of Human Rights Advocacy and Education. [On-line] Available: Ebsco Host.

Jimmieson, P. (1994). A Multimedia Aide to Investigative Interviewing. [On-line] Available: http://www.csc.liv.ac.uk/~phil/thesis/front.html.

Kristof , R. & Satran, A. (1995). *Interactivity by Design.* Mountain View, CA: Adobe Press

Lynch, P. J. (1994). *Visual Design for the User Interface. Part 1: Design Fundamentals.* [On-Line] Available: http://info.med.yale.edu/caim/publications/papers/gui.p1.html.

Lynch, P. J. (1994). *Visual Design for the User Interface. Part 2: Graphics in the Interface.* [On-Line] Available: http://info.med.yale.edu/caim/publications/papers/gui.p2.html.

Lyseith-Jones (1991). Human Rights in Primary Education. In Starkey (Eds.), The Challenge of Human Rights Education. Pp. 73-89. Cassell:London

Mark, W (1986). Knowledge-Based Interface Design. In Norman, D. A. & Draper, S. W. (Eds.), User Centered System Design (pp.32-61). Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Montague, W. and Knirk, F. (1992). What Works in Adult Instruction: The Management, Design and Delivery of Instruction. [On-line] Available: http://www.nprdc.navy.mil/wworks/cover.htm.

Norman, D. A. (1986). Cognitive Engineering. In Norman, D. A. & Draper, S. W. (Eds.), User Centered System Design (pp.32-61). Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Norman, D. A. (1988). The Design of Everyday Things. New York: Currency Doubleday

Norman, D. A. (1993). Things That Make Us Smart. Publishing: Reading, Massachusetts: Addison Wesley.

Papert, S. (1993). The Children's Machine. Basic Books.

Papert, S. (1996). The Connected Family. Atlanta Georgia: Longstreet Press.

Peter D. Hart Research Associates, Inc. (Human Rights U.S.A. (1997). Report Card on Human Rights in the USA.

Quintana, C., Abotel, K, and Soloway, E. (1996). NoRIS: Supporting Computational Science Activities Through Learner-Centered Design. [On-line] Available:

Reardon, B. (1995). Education for Human Dignity. Philadelphia: University of Pennsylvania Press.

Reiber, L. (1994). Computers, Graphics, & Learning. Madison: Brown and Benchmark Publishers.

Rettig, M. (1992). Interface Design When You Don't Know How. Communications of the ACM V.35 No.1 29-34.

Richmond, M., Bodnick, S., Pageau K., Pereira, F., & Woof, B. Manual Revisited. [On-line] Available: http://watarts.uwaterloo.ca/ENGL/courses/engl210e/DevelopersDirectory/MikeRichmond/ManualRevisited/toc.htm

Soloway, E., Jackson, S. L., Klein, J., Quintana,. C., Reed, J., Spitulnick, J.. Stratford, S. J., & Studer, S. (1996). Learning Theory in Practice: Case Studies of Learner-Centered Design. [On-line] Available: http://www.acm.org/sigs/sigchi/chi96/proceedings/papers/Soloway/es_txt.htm

Soloway, E. & Pryor, A. (1996). The Next Geration in Human-Computer Interaction [On-line]. Available: No longer available.

Starkey, H (1991). The Council of Europe Recommendation on the Teaching and Learning of Human Rights in Schools. In Starkey (Eds.), The Challenge of Human Rights Education. Pp. 20-38. Cassell:London.

Thinking Things 2 [Computer Software]. (1995). Edmark.

Thuring, M., Hannemann, J.,& Haake, J. (1995). Hypermedia and Cognition: Designing for Comprehension. Communications of the ACM V.38 No.8 57-66

Watson, R. (1997). Mental Models [On-line]. Available: http://www.sci.usq.edu.au/units/66309/rwlect/ucd.html

Wittman, M.C. (1997). Student Difficulties With Wave Concepts. [On-line]. Available: http://www.physics.umd.edu/rgroups/ripe/mcw / seminartalk/