MIT/LCS/TR-298

VIDEO GAMES

AND

COMPUTER AIDED INSTRUCTION

Ken Krugler

VIDEO GAMES AND COMPUTER AIDED INSTRUCTION

June 16th, 1983

Ken Krugler

ABSTRACT

This document will briefly outline the evolution of video games, discuss current video game theory, and describe a program to teach typing on the IBM personal computer.

KEYWORDS: CAI, video games, IBM PC

CONTENTS

1.0 INTRODUCTION

This paper is the result of my research during the fall of 1982 under the 6.100 program. The project was the writing of this research paper, which investigates the possible link between computer aided instruction (CAI) and video games. This project was supervised by Mr. Nathaniel Rochester, a visiting scientist from IBM.

1.1 PROJECT STATEMENT

This has been a research project, with most of the effort going into 1) Finding out about other CAI research efforts, 2) Analyzing their results and 3) Drawing conclusions about how to link CAI and video games together, with the emphasis being on applications to teaching typing.

1.2 JUSTIFICATION

Education and computers seem destined for each other. As the amount of information increases, the need for effective, self-paced learning in a flexible instructional environment becomes clear. Our school system even now is bogged down by high student/teacher ratios, limited flexibility and an inability to help the highly motivated student. Computers offer an apparent remedy for this situation.

Several CAI systems exist now which allow the student to study a particular topic independently. These are usually AFO (ad-hoc frame oriented), ie. the material is arranged as independent blocks of ideas or information to be learned. This provides a self-paced instructional environment, albeit not nearly as flexible as a true student/teacher situation. Other CAI systems have come up with solutions to this lack of true dialogue between instructor and pupil (eg. information structure oriented), but the required programs have been large, complex and slow.

With the advent and growing ubiquity of personal computers (PC's) the use of CAI in small schools and the home has become a possibility. Personal computers provide an intimate yet powerful educational tool, if the instructional software can capture the attention of the students. The strengths of PC's aren't in their ability to handle large data-bases and programs, however, but in their immediate availability to the user, high level graphics and real-time response.

In an attempt to use these strengths the fusion of video games and CAI becomes an obvious goal. Video games have the power to entrance us for hours, while motivating us to concentrate on an often frustrating and difficult goal. If the same concentration was applied during school to difficult subjects, the learning rate of students would show a dramatic increase. The basis of this fascination with video games and whether it comes only from motor-sensory stimulation or some additional combination of aggression and goal achievement is only now being studied in detail.

1.3 GOALS

As mentioned previously, the first step towards investigating video games and education will be to outline a CAI program which will teach typing. The goals of the project were as follows.

1. Identify the characteristics of videos games which make them fun to play

2. Propose general frameworks for games which could be useful for the different types of learning applicable to video games.

3. Specifically decide how to teach typing on a PC using video game theory.

4. Investigate the possibility of a more general framework for CAI programs, such as a game model which could act as a user interface between a class of CAI programs and the student.

The main focus of this project is to outline a workable solution to item 4 above, then implement the program next semester. Items 1 and 2 are primarily researching published papers on the subject, while drawing conclusions from these studies leads to item 3.

Implementing a CAI program on the IBM PC, however, must draw on several other sources. Teaching people how to type in the most efficient manner is a topic with a vast pool of theory and research. A concrete technique which can be used by the video game program must be decided on. In addition, the theories of student modelling in CAI must be used if the proposed implementation is to be as effective as possible.

Item 4, the generalization of concepts from the previous four goals, will be unreachable in a single-semester project. The idea here is to allow a 'game filter' to be placed between a CAI program and the student, in order to make the information more exciting and improve the student's concentration and learning curve. The CAI programs which can be used by this 'filter' may be a special group, but this could accelerate the acceptance and use of CAI.

As stated previously, the main goal of this project is to propose an outline for a CAI video game to be used on the IBM PC. The program will hopefully improve the learning curve for students studying typing on the IBM single-hand chord keyboard. While working towards this goal, however, I should acquire useful information about CAI in general and the use of synthetic student models.

2.0 VIDEO GAMES

2.1 HISTORICAL PERSPECTIVE

Games have been around for a long time, but the recent wave of video games is fairly new. Fueled by advances in electronic hardware, game halls previously filled with pin-ball machines now flicker and beep with sophisticated mazes, war games, space battles and monsters.

The original games were fairly simple, starting with PONG. Sound, scoring and a few simple figures on the screen were enough, however, to draw impressive numbers of quarters out of pockets and into the slots. Variations on PONG yielded hockey, tank battles, biplane fighters and car racing.

As micro-processors and video display technology became more sophisticated, the level of detail and complexity of the games increased. The next major advance came with the still popular Asteroids and Space Invaders games. These were the first of the true 'computer as an enemy' games, where the player attempted, through acquired knowledge of the game and superior reflexes, to defeat the computer, which used its advantages of more pieces and superior reaction speed to test the ability of the player.

The gimmick here, the one most important feature of all current games, is that the player never really 'wins'. Every game ends with the eventual destruction of the player, through missiles, meteors or some other dastardly weapon. The vital key needed to continue the player's interest is the presence of a goal, through the use of score-keeping. A clever ploy is to allow the player to permanently record his score (with his initials) on the machine for everyone else to see. Ego and competitive drive now work to extract more money from eager young gun-slingers who compete with friends and local 'video heroes'.

Once the important hooks were in place, development consisted of trying to find the right combination of fantasy, surprise and challenge which, when used with increasingly sophisticated graphics and sound, would attract the game room crowd this year. Games change so fast that currently the average game is only good for a few weeks at most before the regulars have mastered it to their satisfaction and begin looking at other offerings.

Exceptions exist, such as Asteroids or most notably Pacman, a game which departs from traditional offerings. Pacman, where the user alternates between being a master escape artist and a voracious great white shark, has continued its popularity over the course of several years. It apparently derives its longivity from a combination of factors; It offers a challenge besides the 'shoot until you drop' variety of most games, it offers the beginner a reasonable game while still challenging the expert with ever-increasing rewards and challenges, and the game is visually attractive to boot.

The most current trend in video games is to mix the basics of a good game with a story, ala Dungeons and Dragons. This trend started with Donkey Kong, a tragi-comedy involving a helpless maiden, a giant gorilla and you as the fearless hero. The game goes through various stages, ranging from simply reaching the maiden while dodging drums to destroying the gorilla by removing all the rivets from a building.

2.2 DESCRIPTION

In describing current video games, it is important to realize that a successful game is actually a system, a working combination of the game design, implementation, software and hardware/video components. The best games use a solid design, one which is based on valid video game design theory, avoids the physical limitations of the system and exploits the positive features of hardware/software/video components.

2.2.1 IMPORTANT ASPECTS

Video games in general employ combinations of four actions.

1. Destruction: The elimination of objects on the screen though a variety of means, including shells, hammers and neutron bombs.

2. Evasion: The dodging of all sorts of objects directed at the individual, usually with harmful intent, by the computer.

3. Progression: Deliberate stages of difficulty and presentation shown to the player during the course of a gaming session.

4. Goal attainment: The reaching of definite skill levels in the game, which range from surviving the current wave of rockets to crossing the last lane of rush-hour traffic.

Almost all games mix these four actions, though the mechanisms are widely diverse and often subtle. Satisfaction during a gaming session is achieved by success at all the actions, so it makes sense, in the interest of variety and flexibility of game design, to use all of these techniques.

2.2.2 AUDIO/VISUAL EFFECTS

A good game design makes use of and depends on the graphics/sound effects provided by the physical implementation. Fast, high resolution monochrome systems use detailed animation of moving objects to grab the viewer's attention, while a color system with good sound effects can be equally effective, given that the implemented game has been design with these limitations and advantages in mind. In general, as technology has advanced it has become vital to make use of the latest developments in these areas.

2.2.3 HARDWARE ISSUES

In order to achieve the special effects required by most of the games, extremely fast hardware coupled with high-resolution displays and sophisticated sound generation hardware must be used. In general these games are programmed in assembly (machine) language in order to achieve the speeds necessary.

2.3 AFFECTED GROUPS

It appears that the distribution of people who use video games follows a bell shaped curve. There are obviously those who enjoy playing for hours at a time, while other people avoid arcade games like the plague. The center of the curve is probably somewhere around the age of 14-15, with the majority of players being male.

The fact that so many video game users are male can lead to two conclusions. The first is that video games are centered around aggression and destruction, with lots of masculine characteristics (the military, male heros, pilots, etc), thus the players tend to be male, as sex differences in our culture cause males to identify with these attributes, while discouraging female participation. Another angle would be that video arcades are a natural gathering place for young males, thus games are tailored to their desires.

Pacman has proven surprising popular with women, however, far surpassing every other game. Reasons for this seem to be the more 'artistic' appearance of the game and the lower emphasis on aggression and destruction. This seems to give some indication that video games can be designed to attract both sexes, especially if presented in an environment that both males and females are comfortable with.

2.4 THEORIES OF ENJOYMENT

Video game manufacturers and software writers spend enormous amounts of time trying to create a winner, the game that grabs the attention (and quarters) of a large percentage of video game players, the game that makes the players happy and the author rich.

2.4.1 RESEARCH

The reasons why people enjoy games have been investigated by several groups. Obvious answers are that games stimulate curiousity, challenge the individual, and give satisfaction through goal attainment. The aspects of games which attract people are derived from an understanding of the basic motivating principles behind game participation. Game motivation in general has had a fairly respectable investigation done by various groups, but the effects of video games haven't been studied extensively.

2.4.2 REQUIREMENTS

One study which has been done on video game aspects was undertaken by Thomas Malone as his PhD dissertation for the Stanford University Department of Psychology (reference 1). Part of his efforts resulted in an outline of requirements for a successful video game design. The list of his requirements is as follows:

- Challenge: The most basic motivating factor, and the most important. Challenge can be provided by a variety of techniques, such as:

  - Goals: A game must have a set of clear goals in order to offer a meaningful challenge to the player. Some of the aspects of goals which are important are:

    - Personally meaningful: This relates to the problem of attracting a wide variety of users. The goals presented must provide a focal point for the player's efforts, thus they must be meaningful to the player. Goals can range from surviving the current attack to learning the meaning of a new foreign word.
    - Multiple levels: To continue to hold the player's attention, the goals should have a dynamic representation, ie. the goal should change to match the game situation and the player's current ability. Some standard techniques for implementing variable goal levels are:
    - Score-keeping: Constant feedback as to how the player is doing provides a continuous incentive for higher scores. A comman embellishment is to separate the scoring spectrum into classes of abilities, from novice to expert.
    - Timed response: By operating in real time it becomes possible to give the player a goal based on how long it takes him to do something. A time limit can be imposed, which when varied gives the diffent goal levels required.

  - Variable difficulty levels: An other way to provide challenge is to alter the difficulty level presented to the player. This difficulty level can be set at the beginning of the game ("Please enter your skill level") and/or dynamically varied throughout the game. Variable difficulty levels tie in closely with dynamic goals, as a goal is often to overcome the obstacle after the difficulty has been increased. Thus common methods of varying the level are increased game speeds, decreased time, increased complexity and increased enemy (computer) power.

- Uncertain outcome: The challenge of a game is increased if the outcome is in doubt. If the player is certain to win, or get a certain number of points, the motivation and enjoyment is much less than if the issue is in the air. In current games the player is reasonably certain to lose (not beat the computer), but the uncertainty comes from the level of proficiency displayed in the current gaming session. Uncertainty can be implemented by:

  — Hidden information: Not revealing information until the game is in a required state or the correct action is made by the player is definitely one way to render the outcome uncertain, so long as there is always more information waiting to be found. This is to say that the outcome will remain uncertain so long as the player can expect new twists to appear for as long as his ability increases.
  — Randomness: The most popular way to create uncertainty is to randomize aspects of the game. Unfortunately, as games get more complex the ability to randomize the actions of the computer becomes more difficult. In general, however, if the best action for the player in a given situation depends directly upon the randomized action of the game, uncertainty will be maintained.

• Fantasy: In addition to challenge, fantasy is an important aspect of video game design. A truely fascinating game allows the player to immerse himself in an artificial environment 'created' by the game. The actual attainment of this goal is perhaps the most difficult aspect of game design, as it relies on the right presentation of information, video/audio effects and situation/response handling. There are two general classes of fantasies:

  - Extrinsic fantasy: An extrinsic fantasy has no feedback from the game to the user. This means that the fantasy depends on the skill, but not vise versa. An example of this would be the Hangman game, where the game situation depends on the ability to spell, but the skills required don't change with the game situation.

- Instrinsic fantasy: An intrinsic fantasy has feedback going both ways. This means that the fantasy depends on the skill and the skill depends on the fantasy. An example of this would be a Dungeons and Dragons game, where the skill involved (remembering a spell, fighting a demon) depends on the game situation. In general intrinsic fantasies are more useful, they are generally more entertaining and offer a more complex interaction between player and game.

• Curiosity: The final requirement for a good game design is that it inspires curiosity in the player. This interacts closely with both the fantasy presented and the challenges offered the player. Curiosity can be inspired by a variety of means:

- Optimal information complexity: The complexity at which information is presented by the game should be appropriate for both the current game state and the current ability of the player. For example, if at the beginning of an Adventure game a player was presented with a difficult riddle to solve before he could advance, the odds are that many potential users would be stymied and ultimately discouraged from using the game.
- Novel and surprising information: It's fairly obvious that the more interesting the information which is presented to player, the greater will be his curiosity. Information which is novel and/or suprising in general will interest and intrigue the player, excersing his analytic or reflexive abilities.
- Audio/visual effects: Perhaps the most important technique for maintaning a player's curiosity (and interest) is to attract him with eye-catching graphics and dazzle him with wild sounds. Any arcade worth its weight these days is filled with zapps and beeps and clicks, synchronized to the flashing graphics displays.
- Constructive feedback: Giving the player useful (and eye-catching) information about his current situation is yet another way to keep his attention and curiosity high. Nothing causes a game to lose its appeal faster than lack of feedback to the player. The information can be in

the form of sounds, numbers or symbols, but the important thing is to let the player know how he's doing in an interesting, constructive and even suprising manner.

2.4.3 CASE ANALYSIS

A good way to clarify what has been said is to analyize a popular arcade game, in this case PACMAN. By showing how it follows good game design principles the previous outline gains clarity and additional meaning.

Pacman, as stated before, is a highly successful arcade game. The game environment is a colorful maze, containing numerous regularly spaced small dots (munchies), occasional large dots (wheaties), and four brightly colored monsters with names like 'Blinky'. The player is represented as a yellow circle with a pie-shaped mouth, which can be moved through the maze under the control of the player's joystick. The colors are visually attractive, and the game has satisfying 'swallowing' and 'eating' sounds.

The immediate goal of the player is to eat all of the munchies and wheaties in the maze without being caught by the monsters. Once the maze is cleared the game moves to a faster, more difficult level with a new set of orange dots. The twist is that the pacman can turn into a preditor once it is energized, by eating a wheatie. This allows the monsters to be pursued and eaten, increasing points and giving the pacman breathing room as it moves around the maze. Monsters are re-created once eaten, thus the game never finishes.

The important points to remember are that:

- The game has a definite, highly visible goal (clearing the maze).

- The graphics and sound effects are well done.

- The game depends on quick reflexes and the ability to plan ahead a sequence of moves in real time, offering significant challenges.

- The game has an apparent of randomness to it, as the monsters' paths are initially difficult to predict.

- As the game progresses the difficulty level increases, with the speed of the monsters increasing with every new maze.

- The game creates a vivid chase fantasy, with clever tactics allowing the player to turn on the computer adversary and achieve revenge.

- The information presented is sufficient to make immediate decisions yet complex enough to prevent a completely accurate analysis.

3.0 VIDEO GAMES AND LEARNING

3.1 TYPES OF LEARNING

The various types of learning and the underlying processes have been studied extensively for many years. In this paper only a specific class of learning will be examined, from the viewpoint of learning typing. The two cases of learning we will be applying to CAI are:

- Muscle memory: The learning of muscle response patterns to given stimuli.

- Memorization: The long-term retention of information.

Other levels of learning, such as the assimilation of concepts, are also applicable to CAI, but they involve a more complex analysis and are not required for teaching typing.

3.2 INTRODUCTION TO CAI

CAI has been discussed and experimented with for as long as computers have been around. The computer appears to be the ideal teacher in many respects: able to tirelessly drill students, grade tests flawlessly, have on-hand a vast amount of information. Unfortunately the promise of CAI has been far from realized (reference 2). Some of the obvious reasons for its failure, at least previous to the advent of personal computers, were limited public access and high costs for systems. This paper will concentrate on the exploitation of personal computers, which helps in solving these two particular problems (reference 3).

3.3 TYPES OF CAI

There are several type of CAI instruction systems commercially available, as well as several research projects which are exploring new techniques.

3.3.1 DRILL

Drilling students on material which must be memorized was the first real use of CAI. This consisted mostly of presenting appropriate questions to the student and sometimes grading the results (reference 4). One of the problems with this approach is that it makes no use of the strong points of computers, such as large data-bases and quick response rates, in addition to offering only minor advantages over books, teacher-prepared questions and quizzes.

3.3.2 AFO

AFO, or ad-hoc frame oriented, is the most popular and wide-spread system out today. It combines the idea of 'information frames' with drills and ability dependent presentation paths. The idea is to present conceptual 'blocks' of information as frames, then at strategic points quiz the students on the previous information. Depending on the results, the program can present new, related material or review material which the test revealed a deficiency in. PLATO, an AFO system, is the most wide-spread CAI system in use currently (reference 5). The major disadvantage of AFO systems is that to provide sufficient flexibility to the program for handling student mistakes requires a vast amount of information to be entered, and this information must be carefully thought out first by someone.

3.3.3 ISO

Currently some research is being done on ISO systems, or Information Structure Oriented CAI programs. ISO systems rely on the fact that most information can be presented as a relational structure, allowing the information to be extracted by a reasonably intelligent program. One example is the BRAZIL system, which teaches students about South America (reference 6). The system includes a question parser and response generator, allowing the student to enter standard English questions, which the program analyzes to find out which information to extract from the data structure. The program then generates a reasonable English response with this information. In a similar manner test questions can be generated from the database.

This system allows flexible student/program dialogues without redundant information. The major problem with the program is that there is currently no structure to the system, ie. no set path in which information is presented in a logical fashion, and more importantly the questions and answers generated by the program are consistantly boring. An additional difficulty is that the information must be carefully and logically entered into the information structure by someone who understands both the material and the way the program operates.

3.3.4 WURSOR II

Previously it was stated that the learning of concepts was not considered in CAI at this time. An experimental attempt, however, has been made in the form of WURSOR II, a program developed from the theories of Ira Goldstein, a former MIT professor (reference 7). WURSOR teaches the concepts required to be successful at playing WUMPUS, a logical game centered around avoiding dangers and locating a monster called a wumpus. Success requires the application of a set of logic and probability rules to the game. The important aspects of the game are the use of a knowledge network and an expert player program. The knowledge network is a representation of the rule acquisition paths taken by every normal player. Thus the game assumes that rules A and B must be known before rule C can be learned.

This network is then used by the expert player program, the wumpus advisor. Based on the current position of the student in the network the advisor decides which rule should be learned next. The game is then altered to provide a situation where the rule should be applied, and the advisor gives the student some hints about the rule.

3.4 JUSTIFICATION

The goal of tying video game theory and CAI systems together is to increase the learning rate of students using the instructional system. As stated previously, video games may provide the ideal vehicle for this goal, as the characteristics associated with popular arcade games are desirable for a successful CAI system.

3.4.1 LEARNING RATE

In general, an increase in learning rate can be achieved by several different means:

- Increase student concentration: increase the intensity with which a student focuses on the material.
- Increase average session length: increase the amount of time a student can and will spend with the CAI system, ie. decrease difficulty of focusing on material while increasing the amount of time a student wants to work with the system.
- Present material more effectively: improve the communication of learning material to the student by presenting the material more clearly.
- Present better material: chose effective material to present to the student during sessions.

Note: The first and second items deal with motivating the student, while the last two points are more concerned with the quality of material and its delivery to the student.

Items one and two are clear goals, with fairly obvious solutions. The key is in motivation of the student, creating a situation in which the student's drive to learn is high. Positive and negative feedback are the two basic motivational factors in all endevours. The fear of failure or threat of punishment is a powerful motivator, as is the pleasure in success and the excitement of a challenge. In CAI systems positive feedback is stressed, as not only is it generally more effective, but positive feedback is also much easier to provide effectively to the student.

In joining video games with CAI, the major goal is to make the instructional system fun to use, ie. to increase the positive feedback to the student in a way which closely resembles those techniques used by popular video games to capture the attention of players. If a CAI system is fun and challenging to use, it is fairly obvious that the student will concentrate harder and longer in the instructional sessions.

3.4.2 PERSONAL COMPUTER EXPANSION

With the rapid growth of personal computers, it appears that the home computer may become, after television, the single most important source of information and attention in the average American home. Having personal computers in a high percentage of homes creates the ideal situation for the wide-spread use of CAI, given that the programs can be effective in such an environment.

Writing successful CAI software for the home PC market requires a careful analysis of the problems associated with such a task. Currently the market has been penetrated by a few drill programs, some single-purpose games, and mos recently by a home version of the PLATO learning system. Of all the offerings now available, only PLATO gives the buyer a series of instructional programs which have a wide variety of appeal in terms of age group and interests. PLATO, however, makes little or no use of both video game theory and the strengths of personal computers.

3.4.3 CAI/VIDEO GAME INTERACTION

This fusion of disciplines requires several key principles for success.

- The structure of the program, while a game, should be flexible enough to attract a wide range of users, both male and female, aggressive and passive.

- The game structure chosen must match the instructional material being presented, such that the goals and actions of the game lead to the acquisition of the skills, knowledge and/or understanding associated with the lessons being taught.

- The game must abide by the basic principles of design used by all successful video games, ie. incremental goals, appropriate levels of difficulty, etc.

- The game must fit within the constraints of the system it is used on, making best use of superior features and avoiding reliance on game aspects difficult or impossible to achieve with the target equipment.

So far discussion has been on points 1 and 2 from the list of techniques to improve learning rates. Items 3 and 4, however, bear close examination. While more difficult to pin down and analyze, the quality and presentation style of the material also has a major effect on learning rate. Even if the information to be learned is presented within the framework of an interesting, challenging game, if the level of the lesson does not progress upward at a reasonable rate for the student, the gains achieved by using video games will be useless or even nullified. If the material presented is far above a student's level of comprehension, not even the most interesting and challenging game will be able to communicate the key ideas or even hold his attention for very long.

Thus it becomes obvious that the design of a flexible, challenging CAI system which pushes the student at a maximum learning rate is far from trivial. Merely placing the system in the framework of a game is not enough; rather, an integrated approach in creating a CAI system is required.

4.0 THE IBM PERSONAL COMPUTER AND VIDEO GAMES

As I said before, the main goal of the project was to create a video game for the IBM PC, one which would teach students how to type on the IBM chord keyboard in the shortest time possible. The current name of this project is VARGO; the design is outlined in this paper, while implementation will take place next semester.

4.1 AN INTRODUCTION TO THE IBM PC

The game must be designed around the limitations of the PC, while taking advantage of its strong points. The IBM PC is a powerful desk-top single-user computer, complete with keyboard, floppy disk drives and a CRT screen. Internally the computer uses an 8088 8/16 bit microprocessor, and offers a variety of options through the use of expansion slots on the bus (reference 8).

The PC offers several positive features.

- Real-time response to inputs, as it is a single user computer.
- Graphics capabilities through the addition of add-on boards.
- Sound generation through a software controllable internal speaker.

- Flexible data storage with the built-in floppy disk drives.
- Fast execution with the potential for machine language programs for the 8088 processor.

The PC does have some draw-backs, however.

- Slow disk access (at least with floppy disks).
- Limited built-in graphics.

The key is to design around the limitations, while making best use of the advantages of the PC. This means limiting disk access while using real-time animation and sound, such as what a video game would require.

4.2 THE PROBLEM

The focus of this research effort is to use the previous facts and reasoning to create a program which will teach people how to type on the IBM Chord keyboard. The IBM chord keyboard uses only one hand to select a full set of ASCII characters, thus learning to use it effectively is distinctively different from touch typing on a standard keyboard (reference 9).

The Chord keyboard offers several advantages over conventional keyboards. Not only does it offer the advantage of single-hand typing, but even more important is the space and money savings it provides. For portable computers the Chord keyboard seems to fill a vital need, by allowing full ASCII typing at near normal speeds, while complementing the small size and portability of these new computers.

Previous efforts have been made at the Cambridge Scientific Center to teach typing on the Chord keyboard. This has involved several versions of the TEACH system, a combination of a hard-copy manual with practice material and a computer scoring and analysis system. The TEACH system delves into the murky waters of typing instruction and its theory (reference 8).

4.3 TYPING THEORY

There appears to be almost as many different typing theories as there are typing instructors, with a considerable amount of typing 'lore' built up over the years. After considerable reading and testing the IBM group decided to use the book (and views) by Dr. Leonard West, a Professor at CUNY. West's ideas appeared to make the most sense, and his book backed by the greatest amount of test and investigation. In addition this offered the advantage of a consistant approach to the problem, with a clear statement of rules and concepts presented in West's book, "The Acquisition of Typing Skills" (reference 10).

4.3.1 WEST'S THEORIES

West has strong ideas on most aspects of teaching typing, and he states these in a series of rules. A listing of the most important ones, in terms of Chord keyboard training, follows:

- Alternate periods of stressing speed over accuracy with periods of recoverying accuracy while retaining part of the speed increase.
- Get the student to practicing normal passages as quickly as possible.
- Stay away from 'nonsense' practice material, especially repetitive practice with a few letters only.
- Give the student immediate feedback about errors.

4.3.2 STAGES OF LEARNING

In learning to type on the Chord keyboard, the typical user goes through three rather general stages.

- Chord memorization: The initial step is to learn the position and fingering of the more comman chords. As quickly as these are known by the student the second step is taken.
- Sentence drill: The known chords are used to create practice sentences of real words. The sentences remain somewhat incoherent until a good base of chords are learned in stage one. The transition between differently fingered chords becomes more fluid, leading to stage 3.
- Passage work: Textual passages of various lengths are typed by the student. These passages are in general extracted from normal literature, giving the student exposure to typical words and chord patterns.

The stages are approximations only, and are somewhat parallel in nature. After practicing sentence drill a student may review some less common chords which will be necessary for passage work. Stages one and three are fairly well defined, while stage two is more a transition period between the two.

4.3.3 TEACH SYSTEM

The IBM TEACH system implemented aspects of West's theories in an effort to teach Chord keyboard typing. The basic program involved a hard-copy manual, one which provided the student with a structured program. The program was set of passages presented to the students for practice, and the structure came from the careful selection of which passages would be presented at which time. The structure followed West's theories as closely as possible, within the limits of the system.

The other aspect of the program was that the manual was used in conjunction with a computer running the true TEACH program. This provided the student's with practice material (all passages were available on-line as well as printed in the manual), a test-bed to type the material in, a 'teacher' which corrected the typed practice passages and gave speed/accuracy feedback to the student and lastly a database which maintained a detailed record of all that the student had done with the TEACH system.

The TEACH system was a large step forward, but it has several problems. One of the most major drawbacks is that it is boring, especially when practicing on passages of text. Another difficulty is that it is relatively self-paced, with the student advancing in the manual at a rate which he decides on, which might not be as fast as is desirable. It would also be better to have the manual on-line, ie. make the TEACH system totally software driven, eliminating the need for a manual.

4.4 DESCRIPTION OF VARGO

VARGO is my implementation of a CAI system to teach chord keyboard typing on an IBM PC. VARGO will hopefully accomplish several goals:

- Increase the learning rate of students being taught chord typing.
- Provide experimental results on the fusion of video games and CAI.
- Explore the potential of finite-state structures in CAI.

4.4.1 PREMISES

The design of VARGO is based on seven major premises:

1. The game sections of VARGO will follow the video game design theories presented earlier in this paper.
2. The major goal of VARGO will be to create a game situation which allows students to practice standard text typing (stage 3 from sec. 4.3 .2).

3. The system used to teach typing will follow the rules presented by West (reference 11).

4. VARGO will have the potential to appeal to a wide group of users.

5. A functional and testable version of VARGO will be done in one semester of work.

6. The version of VARGO implemented will stress practice with standard prose.

7. VARGO will closely follow the instruction manual for the IBM TEACH system (reference 10).

4.4.2 REQUIREMENTS OF VARGO

In order to implement a system based on the above premises, the program must have the following characteristics:

- The game or games used to drill students must have several options, levels and variations. This is to attract the student for a reasonable amount of time, as the time required to drill students is of a sufficient magnitude that just one game, no matter how interesting, would soon become worthless due to student apathy.

- The material presented must follow a structured path, ie. the level of typing expertise must correspond to a known position in the program structure. Because VARGO will be following the TEACH outline, some way of knowing where the student is at all times is important, so that the level of performance expected can be determined.

- The presentation of material should allow for flexible formats. As VARGO should appeal to a wide spectrum of users, the manner in which the course is presented will have to vary as well. A structure or game which is suitable for a young male might not work well at all for an older woman.

- The design of VARGO needs to be fairly modular. While the entire description of VARGO should be laid out at some time (in greater detail then here), it will probably be necessary, due to time constraints, to implement only a piece of it, thus the advantage of a modular design.

- The structure of VARGO should reflect the emphasis on clear skill levels, definite goals and extensive user feedback.

4.4.3 THE SOLUTION

The description presented here is that of a program which offers a flexible presentation of practice text, allowing the user to advance in skill by practicing in game situations, during typing tests and with free typing periods. The advances in skill are clearly visible to the user, as his progress in the program is represented as the movement of a player through a 'maze', which is a loose approximation of a finite-state structure or tree. His position in the 'maze' represents only an approximate level of proficiency, but the goal is to move upward in difficult, where passage between 'levels' is regulated by tests.

4.4.3.1 State Structure

A state structure is used as a convenient representation of the TEACH manual. In the manual students progress through groups of exercizes which build on each other. A state structure involves a representation of the current situation (here the typing skills the student has) and connections to other 'nodes', where the nodes 'higher up' represent increased proficiency.

Given an action, such as passing a typing test with a given proficiency, the state changes as the student moves to one of these 'higher' nodes. Thus the connections between nodes are regulated by tests of some sort, whether presented as such or disguised as a game.

4.4.3.2 Dungeons & Dragons in VARGO

The problem with modeling typing in this manner is that skill levels are not clearly defined, and the amount of work needed to advance is highly variable, depending on the student and the current skill level. Thus a looser representation is needed, one which allows movement (for student feedback) yet does have some hierarchical arrangement.

My proposal uses a Dungeons and Dragons format. Dongeons and Dragons is an instrinsic fantasy game, where the goal is typically to wander through an underground castle, collecting objects as you go, and gradually trying to work your way down to lower (and more rewarding) levels. On each level are interconnected rooms with various menaces and valuable objects.

The mapping between an instruction system and D & D is clear. The castle is a form of finite-state structure, with definitive skill levels. To survive at lower levels requires greater skills and useful acquired objects, much as advancement in typing requires higher skill levels.

The menaces in D & D can be compared to tests of typing. Answering riddles or obtaining special objects corresponds to succeeding at various typing games. Movement between levels would be freely allowed, but failure at a test on a certain level is equivalent to losing a battle in D & D, with the result being a loss of level or even a resurrection back at the beginning. Thus the student is constantly being presented with challenges to pass in order to continue exploring what is hopefully a fascinating world.

The amount of incidental detail built into the world is highly variable. Actual objects to be collected could be present. Riddles might be posed to the player, where an incorrect response doesn't bar advancement, but rather requires the student to perhaps take a more difficult path. The format is loose enough to allow considerable experimentation with the various trade-offs between increased motivation/fascination and losses in typing concentration/practice time.

4.4.3.3 Flexibility

D & D also offers a surprisingly flexible approach to the problem. A game such as this is easily table driven, with an accompanying set of text files for object and room descriptions. Thus the flavor of the game can be changed with the substitution of different descriptions. The D & D style game can be changed into a corporate struggle between executives or a giant shopping store.

The initial implementation of the program could even eliminate the game aspect of the VARGO structure entirely by replacing 'rooms' and 'objects' with descriptions of the current skill level of the student and speed evaluations. This would create a 'base' program to judge the effectiveness of various embellishments (and might be necessary from a commercial marketing viewpoint).

4.4.3.4 Goal Levels

Goal levels are obviously a part of the game, in that the tests will become more stringent at lower levels. The student, however, is not forced to the lower levels, but given incentives to explore there on his own. The the idea of a game is extended to the learning path of the subject.

Rooms on the same level will have approximately equal skill levels required by tests and games. Menaces will correspond to required tests, while games can be considered as doors to open or riddles to solve in order to reach another area on the level, obtain useful objects or move vertically in the 'castle'.

Free typing is probably a desirable feature in the system, especially with 'barriers' that need practice in order to overcome. A means of implementing this would be through 'grafitti boards', ie. like writing on the walls of the rooms, or firing off a memo to someone else in the company. It is obvious that many such 'frills' can easily be added; it is important to consider their relative values before much time is spent thinking about and implementing possible distractions.

4.4.4 GAMES AND TESTS

Previously in this paper I stated that the nature of the game must fit the lesson being taught. This basic concept puts several constraints on the games and tests proposed here. For example, at higher (easier) levels the goal is to teach chords, thus an applicable game might be 'Asteroids' (detailed later), while at lower (more difficult) levels the typing of passages is important. Thus the games presented would look like 'Jaws III' or 'Fuse'.

The game must also fit the individual. It's quite possible that executives would feel uncomfortable or silly playing 'Asteroids'; the game would have a difficult time being an effective teaching tool. The game then must be flexible, given that we are looking for a general CAI program in VARGO. Different system structures can easily be created by altering the text files used by the state-structure, but the games and tests used in conjunction with this D & D approach must be in character.

The idea of tests presented as such, rather than disguised as games, has some implications which need careful thought. Putting a student under a test situation creates pressure, which may or may not be beneficial. At the present time I think it would be wise to plan for their inclusion in the system, but not making any binding decisions about their worth.

The games presented here are viewed as outlines for implementation. Thus for every game I will present a rough outline of its purpose and structure, followed by a list of comments, embellishments and related ideas.

4.4.4.1 Game Goals

Games can be divided into three broad instructional classes; drilling chords, drilling sentences, and drilling passages. In the following sections only games concerned with practicing chords and drilling passages will be considered, as the catagory of drilling sentences is still unclear in its scope and importance.

4.4.4.2 Game Styles

There are many different styles of games for drilling chords, but only a few for drilling passages. These games can be divided into three main classes; chasing, chased and beat the clock.

Chasing games involves having an object, controlled by the player's typing speed and/or accuracy, chasing a computer-controlled quarry. The quarry will probably move in close vicinity to the text, as will the representation of the player.

Chased games refer to the inverse of the above situation. Here the player is the target, with a primary goal being to keep some distance (or as much distance as possible for as long as possible) between his representation on the screen and the computer-controlled chaser.

Beat the clock refers to typing games organized around time limits. Thus a certain amount of time is allocated to a set typing task, with an initial goal being to finish inside the time frame alloted. The other two classes of games give an immediate feedback, while this reflects only over-all typing speed for the test passage.

4.4.4.3 Game ideas

Presented here are four main game ideas. These offer examples of how arcade-style video games could be modified to offer typing challenges to students.

Asteroids: This is a game for drilling individual chords. The student attempts to destroy incoming asteroids by 'selecting' them. The goal is to survive each wave of projectiles. This is possible since the asteroids have chords written on them; typing the chord explodes the corresponding asteroid.

Important related ideas:

- Speed and number of asteroids are variable for degrees of challenge.
- Asteroids are only one representation, thus could also be spiders or foreclosure agents.
- Feedback is through asteroids exploding and running score.
- Game stresses only individual chords and not transitions.

Jaws III: This is one of the chase games, where the computer follows the player. The basic idea is for the player to start copying a passage, and some time later a shark begins to eat the letters he has typed. The goal is to make it to the end of the passage without being eaten.

Important related ideas:

- Again, the shark is only a symbolic object, thus ugly men and aggressive young executives could also be used.
- Embellishments of the game would be adding an error penalty, where the pursuer gains ground on an error. Another idea is to give bonus point opportunities; eg. chances for the player to increase his score by typing optional words or phrases, but with the penalty of losing ground to the shark while doing so.
- Varying the initial head-start, shark swimming rate, length of the passage and error penalty can all be used to shape the game to a particular skill level for both speed, accuracy and endurance.

Hare and Hound: This is an example of the chaser style game, where the computer controls the game and the player pursues by means of fast/accurate typing. The exact inverse of the Jaws game could be used, with the quarry being a rabbit and the pursuer shown as a beagle.

- The representations are flexible.
- The error penalty could be having the rabbit disappear down a hole and appear somewhere else in the text.

- Game options are many, such as being able to 'disappear' down a hole at the end of a line of text, then appearing at a random place in the text, etc.

Burn-out: This is an example of a time game, where a burning fuse to a bomb is the time limit for the player typing text. The feedback is through whether the player finishes the passage in time or not. This is a good candidate for the 'test' situations in VARGO, where the player must pass it in order to move to a lower level.

- The representation could be changed to typing out a sell instruction before a falling Dow Jones average reaches zero, etc.

- Options such as slowing down the burn by typing more difficult lines are possible.

- The feedback here is not as good as with the other ideas, being a pass/fail measure for an entire section of text rather than an immediate feedback situation.

4.5 VARGO IMPLEMENTATION

Implementing VARGO on the IBM PC has a major constraint imposed on it, in that a workable system must be finished before the end of the term. Thus VARGO must be brought up in a modular fashion, with usable pieces becoming quickly available.

4.5.1 DESCRIPTION

The modules of VARGO which need work, in no order of importance, are:

- The shell: The D & D style framework which allows the player to move in the outline of a game. This is primarily a finite state machine with the appropriate text files.
- Jaws III: A main game idea, involving the player being chased by a shark. This is important to test the actual uses of a game in drilling students on muscle-memory tasks.
- Burn-out: A tester of typing ability, which can be used as the menace in a D & D situation.
- Asteroids: The game to teach chords initially. This isn't all that necessary to the success and evaluation of VARGO, but it would be nice to implement.

This is a very broad picture of VARGO, which will be refined during the actual implementation process.

4.5.2 IMPLEMENTATION PATH

As stated before, the most important aspect of the project is to get a working version of one of the games up and running, thus allowing tests on the benefits of video game theory and CAI. Thus the implementation path is:

- Get Jaws III up and running alone.
- Get Burn-out going.
- Link the games and implement skill levels with the shell.
- Add Asteroids and other embellishments.

Most likely it will be possible in one semester to complete item 1, with partially working versions of items 2 and 3.

5.0 CONCLUSIONS

This project was intended to investigate the link between video games and CAI, with emphasis on the IBM PC. The four main goals were:

1. Identify important characteristics of video games which are keys for enjoyment.

2. Propose general game outlines which will be useful for teaching typing.

3. Specifically propose an implementation of a teaching system for the IBM Chord keyboard.

4. Investigate a general means for implementing CAI systems with a finite-state network, particularly in the format of an 'Adventure' or 'D & D' framework.

5.1 ACCOMPLISHMENTS

In general the goals have been achieved. Specifically:

- Research on video game theory was investigated and important points isolated.

- Based on the guidelines from this research several game formats were suggested.

- After evaluating the TEACH program, as well as various papers on AFO and ISO systems, a system (VARGO) for teaching IBM Chord keyboard typing was outlined.

- During this process several ideas for future research arose concerning the potential of finite-state structures in CAI.

5.2 FUTURE WORK

Next semester will see the implementation of VARGO on an IBM PC. After the first few pieces are up and running the verification of my design decisions will begin. I anticipate many changes, as implementing a program which deals extensively with human ideosyncracies is very difficult; I can see many problems cropping up as I continue work on VARGO.

The goal of a working TEACH-style system is a reasonable one, I believe. After Jaws III or a look-alike are going it will be possible to evaluate the benefit of having the drilling being presented as a game.

My hope is that I can get to the stage of implementing the shell, thus giving myself the opportunity to investigate the possiblility of using such a structure as a framework for a variety of instructional programs.

5.3 ACKNOWLEDGEMENTS

Several people provided valuable aid and discussion during this research project. I'd especially like to thank Mr. Nat Rochester and Mr. Frank Bequaert, both members of the IBM Cambridge Science Center. Their practical knowledege of the TEACH system and the problems in CAI kept my feet on the ground, even as they offered encouragement and useful feedback on my ideas.

5.4 SUMMARY

I enjoyed the opportunity to perform academic research of this nature; it was a new experience for me. Tracking down leads and picking out important points from lengthy papers were valuable skills I acquired. This paper was rather open-ended; the amount of detail present and the breadth of coverage were entirely dependent upon how much time I was willing to spend on it. I've got a much better feeling for the problems of CAI now, and I'm looking forward to real progress on VARGO next semester.

1) Carbonell, J. (1970) "Scholar : Mixed Initiative Man-Computer Dialogue," Massachusetts Institute of Technology Thesis.

2) Carr, B. (1977) "Wursor II : A CAI Program with Student Modelling Capabilities," MIT AIM-417.

3) Davis, A. (1973) "The Development, Implementation and Use of a Computer-assisted Instructional System," UIUCDCS-R-73-563.

4) Frenzel, L. (1980) "The Personal Computer - Last Chance for CAI?," Byte (5, 7) pp. 86-94.

5) Lemmons, P. (1981) "The IBM Personal Computer : First Impressions," Byte (6, 9) pp. 26-34.

6) Malone, T. (1981) "What Makes Computer Games Fun?", Byte (6, 12) pp. 258-270.

7) Rochester, N. & Bequaert, F. (1978) "The Chord Key-board", Computer (4, 12) pp. 57-63.

8) Rochester, N. & Bequaert, F. (1977) "Teaching Typing on a Chord Keyboard," IBM TR 00.2918.

9) Sammons, J. (1973) "A PLATO Lesson to Teach Begin Blocks," UIUCDCS-R-73-599.

10) West, L. (1969) "Acquisition of Typewriting Skills," Pitman Publishing Co.