SESSION 10 HYPERCARD

Moderator—Drake R. Bradley, Bates College

Computers for cognitive research: A HyperCard adventure game

CLARK N. QUINN

University of Pittsburgh, Pittsburgh, Pennsylvania

For researchers interested in exploring the cognitive and metacognitive processes involved in problem-solving, one obstacle has been the difficulty of creating engaging and ecologically valid situations in which to observe these skills in practice. Computer games can provide a rich environment for research on such skills. I present a HyperCard example of an adventure game that serves as an environment for research on analogy in problem-solving. The design of the game and interface is detailed, and initial results of and suggestions for extensions to this work are presented.

Research on cognitive and metacognitive skills has often relied on artificial problem situations, ranging from variations on a simple puzzle (Hayes & Simon, 1976) to paper-and-pencil exercises with proposed problem situations (Gick & Holyoak, 1980, 1983), as contexts in which to gather data on subject performance. Here, I shall argue that computer adventure games are an effective environment for problem-solving, and I will consider analogical problem-solving as a particular strategy upon which to focus as an example.

One problem with currently existing problem-solving environments is that although participants may earnestly try their best, the tasks lack the "intrinsic motivation" (Malone, 1981) of activities that subjects voluntarily choose. Subjects are probably more likely to display their repertoire of reasoning skills in an engaging environment. Research on problem-solving also suggests that subjects cannot produce solutions for situations that are too unfamiliar (Wason & Johnson-Laird, 1972) or be situated in domains that the subjects either are unlikely to find interesting or may find aversive, such as mathematics or physics (see, e.g., Chi, Bassok, Lewis, Reimann, & Glaser, 1989). Furthermore, recent research on the situation-specific nature of learning (Brown, Collins, &

Duguid, 1989) and the importance of natural settings for the exhibition of problem-solving skills (Nisbett, Fong, Lehman, & Cheng, 1987) suggests that cognitive skills are most appropriately assessed with familiar tasks.

It is not trivial, however, to identify activities that are both natural and motivating and that incorporate problem-solving situations requiring the application of cognitive skills. It is also challenging to develop tasks that lend themselves to experimental manipulation without violating the principles of engagement and familiarity. Rarer still are tasks with these characteristics that lend themselves easily to data collection for the purposes of psychological analysis. Finally, tasks should also appeal across age groups, to facilitate developmental research in which different problem-solving strategies can be compared. Ideally, we should like to find a problem-solving environment that is motivating across a breadth of age groups, that possesses some degree of familiarity, that can be structured to contain the specific cognitive characteristics that we desire, and that can facilitate the collection of data. Computer adventure games meet the desired constraints: They are a part of many subjects' experience, they are motivating, and they are enjoyed across age groups.

The desire to simplify the collection of data suggests that computer activities are a reasonable starting point in the search for an environment within which to embed problem-solving situations. One activity that many, particularly young, people find engrossing consists of computer games (Malone, 1981). Greenfield (1984) describes two types of available computer games: arcade games and fantasy role-playing games. Arcade games are the quick-

I gratefully acknowledge the support of the faculty, staff, postdoctoral fellows, and graduate students at the Learning Research and Development Center, without whose support and tolerance this work would never have been completed. Requests for reprints should be addressed to Clark N. Quinn, School of Computer Science and Engineering, University of New South Wales, P.O. Box 1, Kensington, NSW 2033, Australia.

reaction games in video arcades; they typify the games available on most home-television game systems and home computers. Arcade games generally require a brute-force search through possible motor skill options, as well as the memorization and automatization of a required sequence of moves. Cognitive complexities are involved (Greenfield, 1984), but the time scale required for learning across multiple attempts precludes arcade games from being a reasonable source of data. Fantasy role-playing games are computerized versions of the famous Dungeons and Dragons game, in which the player searches through an area, encountering monsters that must be defeated, accumulating experience that increases the player's capabilities, and finding treasure that increases the player's wealth or confers an added combat capability. Problems to be solved are often embedded in such games, but other activities are included as well.¹ Since there is yet another category of computer game that includes embedded problems without the additional activities, I will not consider fantasy role-playing games any further.

This third type of computer game, which is mentioned but not characterized by Greenfield (1984), is the computer adventure game. Adventure games typically consist of movement through an artificial world, in which objects must be found and actions with these objects must be performed so that exploration can continue. The problem that players must solve involves the discovery of the specific actions, with the particular object, that must be performed at a particular location if the player is to explore any further. These characteristics—a motivating situation with embedded problems—provide the requisite research environment.

Adventure Games as Problem-Solving

In adventure games, the player takes on the role of a character in an imaginary world. The player directs the character, including the character's movement and actions in the environment. The character can hold and use some number of objects. Characters are subject to the physical characteristics of the world that they are in, so that while they can act in the world, the world can act on them.

Adventure games usually incorporate a theme that motivates the exploration of the environment and provides the contextual clues that guide the problem-solving. For instance, a game set in medieval times might have a quest for a religious artifact serve as the motivation for exploring the available world, and the obstacles could be either real hindrances such as brigands and foreign warriors or fantastic opponents such as dragons or trolls. The environment to be explored usually has a finite size and maintains a constant scale, but within these constraints it can range from a single house or dungeon to an entire continent or world. Typically, parts of the environment are not initially available for exploration and must be revealed by the correct use of an appropriate object at an appropriate location. For example, a portion of the world, including the goal, might lie across a chasm that cannot be crossed until a wand is found, brought to the chasm, and

waved to create a bridge. Objects to be found and used can be anything that can be logically or mythologically construed to have an effect on or in the world. The correct use of an object can usually be inferred from the properties of the object, as in the case of a weapon; the role that the object plays in the environment, as in the case of a magic scroll; or information provided during the course of the game. However, the use of the incorrect object or the use of an object in the wrong place can often lead to undesirable consequences, such as death for the character.

Adventure games are made more difficult by the use of a variety of methods for obscuring the solution. These form miniature problems within the overall structure of resolving the game. Such methods include making the locations in one area difficult to distinguish, requiring the player to map the region, or making the discovery of the objects and their purpose less than obvious. Hints about the correct use of objects can be inserted into the course of the game by means of messages or clues. A further method is to allow more than one solution to a problem, so that if the wrong solution is used at one place, the solution object will not be available later when it is appropriate. In standard problem-solving parlance, the different locations in the environment correspond to the states in a state-space search, and movement between the locations comprises the operators that move between states. However, certain spaces require another form of operator, that of using objects in combination with movement, to make further states of the space available for search. Of course, systematic trial and error can always, eventually, lead to the correct solution.

A cognitive problem-solving environment was implemented as an adventure game with these characteristics. Other complexities in adventure games will not be considered here.

Strategies in Problem-Solving

Adventure games can serve as a source of information on the question of what strategies subjects bring to bear in attempting to solve problems. In an adventure game, when a player has explored all available locations, and when it is obvious that further areas to be explored are blocked by an obstacle, the problem-solving situation is focused on the use of the available objects and commands for overcoming the obstacle. As mentioned previously, one way to ascertain the correct action or operator consists of a systematic trial-and-error search through all possible choices. There are reasons why this is not the most prudent choice. Because there can be a large number of possible actions to be performed, a linear search through the space can be quite lengthy. Another possibility, often incorporated into adventure games, is that the wrong action can have dire consequences. Some consequences are merely ineffectual, but in other circumstances, the incorrect action can lead to the death of the character. The consequences of death range from having to start again from some recent move to having to start over again from the

beginning. Depending on the depth of the game and the progress of the player, starting again can be quite time-consuming and consequently aversive.

One way to shorten the possible search and to minimize the potential for undesirable consequences is to use heuristic knowledge to guide the search. The combination of object and action to overcome the obstacle can be controlled by the designer to range in difficulty from the obvious to the obscure. Cause-and-effect relationships embedded in the means to overcome the obstacle can be based on physical laws, on established myth, on constructions of the designer that have been made available to the player (a feature that lends itself to educational and cognitive applications), or on totally arbitrary factors. To the extent that the knowledge is based on shared culture, fact or fiction, the solution will be controlled by the player's knowledge. To the extent that the information to solve the problem is available in the game itself, the success of the player will depend on the form that the information takes and on the ability of the subject to utilize strategies to obtain and infer the correct application of the knowledge.

Analogy in Problem-Solving

One issue in the practical implementation of an adventure game is the particular cognitive skill to be emphasized in the problem-solving situation. There are a variety of solution strategies around which a problem-solving scenario could be constructed. The information to solve the problem could be presented in a form that is amenable to a variety of strategies, or to a combination of strategies, to solve. For example, the information might be coded, or it might require induction. Ideally, the optimal solution strategy would be somewhat difficult, allowing for the expression and observation of alternate strategies. One solution strategy that is ineffectually utilized is analogy (Gentner, 1989; Gick & Holyoak, 1980, 1983; Quinn, 1990a). If information for the problem solution is presented analogically, data can be collected not only on the use of analogy, but on other strategies applied as a result of an inability to obtain an analogical solution.

An important issue in analogy is the role that other characteristics of the problem may play in triggering other solutions (Gentner & Landers, 1985; Ratterman & Gentner, 1987). Subjects in analogical problem-solving often focus on the "surface" features or physical appearances between possible solutions and the particular problem. To account for surface appearance strategies, alternate possibilities that represent incorrect solutions, ones that would appear to be useful if a surface feature approach was being taken, should be available to subjects. These alternate solutions discriminate between incorrect applications of an analogical solution and strategies other than analogy. Ensuring that solution options span the possibilities of correct application of an analogy, incorrect analogical solution, and other methods of solution facilitates the categorization of the strategy used by the player.

VOODOO ADVENTURE AN ANALOGY RESEARCH ENVIRONMENT

Voodoo Adventure is an initial exploration of the applicability of a computer adventure game as an environment for problem-solving research. It has a simple interface, is predicated on an artificially structured reality, provides four novel problems, structures information thematically so that analogical reasoning is the most effective problem-solving method, and saves a trace of the subject's moves. A separate training game serves as an introduction to the interface and as practice in the auxiliary requirement of talking aloud.

Artificial Reality

One requirement for creating a motivating environment is to create a coherent theme within which to embed the problems so that they are intrinsic to the activity. A constraint on the problems is that they must be structured to reflect the desired cognitive property without violating the theme of the story. It is difficult to do this with either existing situations or established myths as themes. It is difficult to find problems with the requisite characteristics in the obstacles faced by characters in existing fable, and it is unacceptable to creatively alter existing myth. It is also difficult to construct problems that contain the desired structure and are also believable situations, given the player's knowledge of the world.

I addressed these problems in Voodoo Adventure by taking advantage of "the power of myth" (Campbell, 1988) to design an artificial culture incorporating the specific elements needed to assess analogical reasoning while not losing a thematic thread. The voodoo culture of Haiti (Davis, 1985) provided the basis on which to build such a reality. One benefit of this choice is that the voodoo culture is relatively new, arising in a synthesis of ancient African rituals and the imposition of the Christian religion. This newness suggests a flux within which different regions could have the same beliefs, but with different particulars. Also, while specific elements of the voodoo culture may be familiar, the underlying structure of the beliefs is not overly familiar to anyone but the participants. A second benefit of the culture is its location in a region of many small islands, which permits the possibility that there could easily be one more location with a domestic variation on the prevalent culture. By taking an established example of a culture, one that has a relatively recent origin and a synthetic nature, and relocating it in a fictional but easily accommodated location, one can manipulate the culture to have the desired structure without any gross violation of believability.

In the constructed culture of Voodoo Adventure, the major deity is a *loa*. There are different loas for different regions of the island. The loas have three subservient entities: *protectors*, *demons*, and *zombies*. Protectors are typically used to do good, whereas demons are invoked

by the loa as forces for evil. Offending the local loa can be grounds for the punishment of being made a zombie. Zombies are created by taking the spirit of the offender, and the spiritless creature is used as another vehicle of the loa. All four of these entities can be involved in a particular situation.

In addition, certain objects play a role in the culture. Small stuffed replicas of a denizen of the island may be used to house a stolen spirit, which has been removed in the process of creating a zombie. In addition, each loa has a particular religious symbol, an ikon, which indicates the affiliation of the bearer. Some of the familiar elements of voodoo and other cultures are also included: dead chickens, bones and skulls, The Book of the Dead, crosses, and knives are all involved in the notes and the game.

Although an artificial culture allows the manipulation of the problem structure while a current theme is maintained, players might not have any idea of how to conduct themselves as participants in such a culture. They are, instead, new observers of the culture. To let the role of the player reflect the player's knowledge, the character is cast as an official observer, an anthropologist who is studying the culture. Thus the situation presented in Voodoo Adventure is derived from and taps into existing knowledge, but it provides a departure point from which the problem-solving tasks can be constructed without violating the unity of the game theme. (See Figure 1.)

Figure 1. The introduction screen.

Figure 2. A sample notes page.

Base Knowledge: The Notebook

The game environment is an extension of reality. The contextual information needed to support the player's transition is placed in the introduction to the game, a mechanism common to adventure games. The game situation is presented in a preamble that defines the current situation and lays out the goal to be achieved. The location of information for solving the problems is more flexible, however. The form of conveyance of the information to solve the problems controls the type of problem to be solved. In this example, the desired relation between the problem and the solution is analogical. This requires that the information be obtained prior to problem solution. The role that the player takes in this game provides a natural mechanism with which to present the information. Since the character is an observer of the culture, the information about the culture can be cast as the notes that the observer has acquired. Thus, the player can have access to the notes before the problems must be faced. (See Figure 2.)

Within this structure, four problems were created and information pertaining to these problems was placed in the notes. Each problem involves one of the four postulated supernatural entities that act in the mythology of the culture: protectors, demons, zombies, and loas. For each entity, abstract information about the role of the entity is established. For example, the notes tell about how each loa is associated with a part of the body and that the local loa is associated with the head. In addition, for each entity, a concrete example is presented that is analogically related to the situation that the character will face in the course of the game. In the case of the loa, the story tells how a man was under the scrutiny of a loa associated with the limbs; his solution was to show a leg bone. The analogy in this situation is that for a loa of the head, the character would not want to use a leg bone, but rather a head bone or a skull. The object mentioned as the solution to the story in the notes, in this case the leg bone, is not the object that will work in the game, but rather a superficial distractor.

The number of objects available for use in one particular problem controls the strategies available. To compare problems, equivalent numbers of objects available for each problem solution are needed. Eight objects are important in this game, one correct object and one distractor element for each problem. The greatest number of objects that can be available for any one position is five (by the fourth problem, three of the objects have already been used). The more objects available, the better the ability of the subject to exhibit strategies. Objects that are related to a particular problem, however, can serve as nonrelational distractors with respect to other problems, minimizing the total number of objects necessary. Of course, if the subject has not picked up all the objects available, fewer than five are available. For this reason, the number of objects maximally available is five. To make sure that the five maximally possible objects are available, and no more, the subject has to find some objects after encountering previous problems. If the subject does not take these objects, the subject could have fewer than five objects avail-

able. So that this control can be maintained, each object that is successfully used also has to be used up by the action so that new objects can be found while the maximum is maintained at five.

A further control on the strategies used by subjects involves whether the successful object is found or is possessed in the beginning. So that subjects cannot successfully use a strategy that specifies using only objects that were originally possessed or only objects that are found, the successful objects are equally divided between original objects and found objects.

One particular feature of the design is the quantity of information in the notes. There is a deliberate effort to provide more material than is likely to be recalled. This permits the examination of strategies other than the optimal. When players are unable to recall the information in the notes, they have to rely on other strategies. Since the other strategies are of interest, this design feature ensures that alternative strategies are likely to be observed.

Of course, there must be a goal to be accomplished, and that goal must thematically include the use of the information to be presented. Thus, the goal must require the use of information about the culture. In this case, the goal is to encounter problems that are structured to be analogically related to the information about the culture. Since the problems of the voodoo culture center on appeasing the various deities and entities that populate the spirit world, the problems in the game can utilize this structure. The trick is to get in trouble in the voodoo culture. Rather than predicate the problem on some error of the character, to increase the confidence of the player, there must be an outside influence. Typically, the governments of countries or islands with voodoo cultures do not take kindly to the autonomy and power of the cults, so the problem is situated in the form of a potential connection between the government and the player that, as a threat to the culture, must be examined. (See Figure 3.)

Acting In The World: The Interface

One design issue in the construction of a computer-based game environment is the interface through which the player interacts with the program. The interface should

Before you can go, however, a traitor to the cult is caught attempting to pass information to the secret police. In a futile effort to save his life, the informant names you as a secret infiltrator. Despite inclinations to ignore the claims of an informant, the cult has traditions for handling such charges. You are to be placed in a situation where your knowledge of their traditions (and, maybe, a little luck) will determine whether you escape and survive to return home.

You are given a chance to review your notes...

Review Notes

Figure 3. The goal statement.

support the activity without interfering in the execution. It also should reflect the way the user thinks about the task (Norman, 1986). A variety of interfaces have been used with computer adventure games, spanning the possibilities from the linguistic, with examples from simple two-word commands to sophisticated language parsers, to the graphic, using the latest in input devices. For the purposes of assessing a cognitive skill, the interface has to be simple enough to be learned quickly, comprehensive enough to permit the full range of reasonable actions, and generic enough to forestall any hint of the solution to the problems.

To understand how to make a program easy to learn, an analogy can be drawn here between the interface to the game and interfaces to disk operating systems. Two major interfaces to operating systems have been developed, the "command line interface" and the "graphic user interface" (Quinn, 1990b). The first operating systems were command line interfaces; users typed the commands that they wanted executed, but these systems were difficult to use because the commands were hard to remember, the specification of the desired action required typing skills, and often there was little feedback. As an alternative, graphic user interfaces were developed that presented the available options on the screen, often using a graphic representation to facilitate understanding, and allowing the user to point, with an input device, to the desired option and select it. Both operating systems and applications with such graphic interfaces are easier for novices to use. This holds for adventure games, so the text-based adventure game interface was discarded. Also, the graphic appeal of computer and video games would have been ignored, minimizing the potential attraction and consequent motivation for the players.

Once a graphic interface has been specified, the next task is to design the appearance of the screen. To do this, however, the semantics of the interface must be designed. Two aspects of the interface need to be determined: the actions that the player can instruct the character to perform and the feedback that the system gives to the player about the results of that action.

The actions that the player can choose must be limited to a representable set. A minimum set of actions was determined by the actions that a player must have available. A character must be able to pick up, or "take," an object at the character's current location, adding that object to the inventory of the objects that the character possesses. A character must also be able to "drop" an object at the current location. Due to potential ambiguities in the representation of an object, the player also needs to be able to have the character "examine" the object to determine its exact specification. Finally, the character needs to be able to utilize an object according to its nature. Rather than allow a variety of different uses, a generic "use" option was made available. Although this limits the creative uses of an object, it also simplifies the interface and minimizes confusions for the player. A character can also move in any of the available directions.

Figure 4. The interface (the help screen).

The complete interface must represent the current location in the game, the objects that the player has already obtained, any objects available at the current location, the available actions that can be performed, the directions in which the user can move to get to other locations, and feedback concerning the results of actions. To implement this graphically, there must be a picture of the room, pictures of the currently held objects (the inventory), pictures of the available objects, a representation of the possible directions in which to move, and a place to represent relevant feedback. The available screen is divided up into regions to accommodate these needs.

One of the standard interfaces used in graphic adventure games presents all the possible actions as a set of choices for the player. If the objects to be used are also present, the player can select the action and object, completing the execution of actions without needing any keyboard interaction. To simplify the specification, the choice was made to have the player specify the action and then specify the object. Thus, a player can "take" or "drop" an object, but the player cannot specify an object and then specify what to do to it.

To represent the movement information, a symbolic mapping to a common representation of directions of movement is used. The available directions are mapped onto a compass rose. For each of the four directions, a visible arrow represents the option of moving in that direction. Directions that are not available have no corresponding arrows. If obstacles are removed to allow new directions, the corresponding arrow is presented.

The representation of the character's location is a picture of the room that the character is currently in. Associated with that room is a region that contains pictorial representations of the objects that are available in that location. A separate region contains representations of the objects that the character currently possesses. Another area of the screen provides textual feedback about the results of actions. To facilitate use of the interface, a help screen is available that details the function of the different regions of the screen. (See Figure 4.)

Implementation

Once the interface has been designed, the next issue is how to implement the game. To completely implement an interface for a game is a daunting task; placing the burden of task management on the computer requires a sophisticated program. Various levels of support for interface construction exist, with a tradeoff between the difficulty of implementing the program and the flexibility of the resulting application. The possibilities range from the use of a programming language to the use of a canned application that generates adventure games. To use a programming language may be the only effective long-term solution, but for a preliminary investigation, the overhead was too high. The available adventure game implementation packages had a number of problems, ranging from a text interface to lack of control of the output. Other solutions suffered from a variety of problems, including difficulty of implementation, freedom of distribution, or flexibility in use.

One solution, HyperCard, presents a simple yet robust environment for creating interfaces and provides a small but reasonable set of interaction possibilities. HyperCard is an implementation application that starts with the metaphor of a stack of cards and expands it with buttons that can be pressed to cause a variety of things to happen. A simple programming language, HyperTalk, allows fairly sophisticated semantics to be assigned to the buttons. HyperCard is readily available on the Apple Macintosh computer. HyperCard also allows the implementation of automated data collection. Program-generated text can be stored in a text file. Through the recording of important actions in a text file, traces of the players' actions can be saved and analyzed. This option has been implemented so that the program keeps track of each of the player's moves and of any uses of objects the player makes.

Voodoo Adventure is implemented in HyperCard. The program includes an opening screen, an introduction, the notes, a screen asking for a name with which to label the associated text file, and the game itself. (See Figure 5).

Figure 5. A puzzle screen.

Table 1
Total Attempts per Problem, Use's, Deaths, and Go's, by Subject

Category	Subject										
	1	2	3	4	5	6	7	8	9	10	11
Attempts: Problem 1	2	2	1	1	2	2	1	1	1	4	2
Attempts: Problem 2	2	4	3	4	1	5	5	1	3	4	4
Attempts: Problem 3	1	4	4	5	2	1	4	1	3	2	5
Attempts: Problem 4	1	2	3	1	4	3	1	1	2	3	3
Use's	5	16	15	18	9	11	13	5	17	13	21
Deaths	1	3	2	2	1	1	1	0	4	2	3
Go's	17	46	57	49	13	27	20	11	41	19	43

AN INITIAL TRIAL

As a preliminary exploration of the utility of Voodoo Adventure as a research environment, an experiment was conducted in which data were collected on subject problem-solving in the environment. In this investigation, I had two goals: to determine whether the environment was understandable and usable by the subjects, and to find out whether the data collected were of use in the investigation of cognitive strategies.

Method

Eleven college undergraduates from the psychology subject pool served as subjects. The subjects were seated at a table, where they completed a questionnaire concerning their experience with computers and computer games, and their preferences with respect to computer games, if any. They were then given basic instruction in using a graphic user interface. Pilot experiments indicated that unfamiliarity with the format of an adventure game interfered with the completion of the problems. This motivated the creation of a separate, smaller adventure game, which served as an introduction to the adventure game and the particular interface. This game, Graduation Adventure, consists of three problems that are solved by exploration and appropriate uses of objects. The information for determining which object to use is based on common knowledge. The game has the same interface as Voodoo Adventure. The subjects played this small adventure game prior to playing Voodoo Adventure. They were encouraged to practice talking aloud, and they had a chance to learn about how to keep track of location, how to specify the desired action, how to move around, and how to complete a game.

The subjects then played the research game. First they read the introductory material, and then they had a chance to read the notes. They could move back and forth through the material contained in the notes, but they could not review the material in the notes once the game began. After reading the notes, they began the game. During the entire game, the subject's verbalizations were recorded to serve as a second source of information. After the game, the subjects were debriefed.

Results

This experiment yielded two forms of data: transcriptions of the verbal protocols and the computer traces.

Quantitative data that could be extracted included the number of movements that each subject made, the number of attempts that each subject made to use an object to get rid of an obstacle (attempts per problem), the total number of "use's" that a subject made, the number of times that each character died, and the number of "go's" that a subject made. (See Table 1.) One caveat needs to be made: This is a preliminary investigation, and the exploratory nature of this experiment precludes any confidence in the results obtained. These data are being presented only to illustrate the types of analysis that the data accommodate, and to suggest further research. For this reason, no statistical tests are reported.

From the raw data of the subjects' actions, further quantitative results can be calculated. First, the average number of attempts for each problem, average number of "use's," deaths, and "go's" can be extracted. (See Table 2.) The average number of attempts can be compared to the total number of attempts possible, five.² (See Figure 6.) The results suggest that the subjects had the most difficulty with the second and third problems. Since the order of the information in the notes was reflected in the order of the problems, and the order was not balanced, this effect could be due to a serial position effect (Klatzky, 1980). Further research is needed to determine the locus of this difference.

The transcripts were examined for evidence of the subjects' strategies. Four categories of strategic behavior emerged from an analysis of the protocols. The first category, *recall*, consisted of recalling and using information from the notes. This information could be accurate or could be incorrectly recalled. A second category involves the situations in which the subjects were clearly trying to determine cause and effect relationships that could be used to predict what object ought to work—a

Table 2
Means and Standard Deviations for Attempts per Problem, Use's, Deaths, and Go's

Action or Outcome	<i>M</i>	<i>SD</i>
Attempts: Problem 1	1.727	0.905
Attempts: Problem 2	3.364	1.286
Attempts: Problem 3	2.909	1.578
Attempts: Problem 4	2.182	1.079
Use's	13.0	5.158
Deaths	1.818	1.168
Go's	31.182	16.339

Figure 6. Average attempts per problem.

cause category. A third strategy consisted of *trial and error*. Finally, a category was created for actions that could not be explained, *other*. These categories were scored for instances of correct and total use. Overall, the performance was superior for the *recall* category and the *other* category. (See Table 3.)

These results give merely an indication of the type of data and results that can be derived from this environment. It is hard to evaluate whether these data, albeit different, are equivalent, superior, or inferior to data from other problem-solving research methods. I surmise that the data will prove equally useful as a source of information for theorizing.

Problems

Several problems were discovered in the course of this preliminary investigation of the game, problems not in the game but with the game itself. These problems include one that is a result of the HyperCard implementation, some that are a result of the particular implementation of the design, and some that result from the game design itself.

One problem results from the HyperCard implementation. The particular features of the program limit the sophistication of the graphic manipulations possible, and

it is therefore hard to make the rooms distinguishable. Subjects often lost track of the rooms they had visited, because the room descriptions are sparse enough to allow ambiguity, and the pictures are not distinguishable from one another. Unfortunately, the HyperCard solution to this problem requires more complex graphics code, which would slow the program significantly. This tradeoff is an unfortunate side effect of the ease of use of HyperCard.

A second problem arose from the particular implementation of the design. For simplicity's sake, the syntax for executing an action requires one to specify the action first and then the object. The subjects were not sensitive to the order constraint in the syntax and quite often specified the object and then the desired action. Although it was easy for the subjects to remedy this problem, they repeatedly ignored the order in their attempts to execute their objectives. Ideally, the interface would allow either form of specification. This problem can easily be remedied with careful programming.

Another problem that arose from the particular implementation involved the feedback for attempted actions. Feedback in the form of text informs the subject when there is no response to the action. In several cases, the same message is used for several different null actions. Similarly, when the same action is tried twice, the message is the same. When these messages are displayed again, there is no obvious change in the screen, since the messages appear in the same physical location on the screen. This often confused subjects; it may have contributed to the difficulty they had in keeping track of which actions they had performed. But this problem could easily be remedied with more obvious effects of new messages and different wording for different actions.

The complexity of the wording in the notes was another problem that resulted from this particular implementation. Pilot subjects were educationally advanced (graduate students and postdoctoral fellows) and had no problem with the notes. Undergraduates, however, had considerably more difficulty with some of the vocabulary. This effect may be confounded with the ability of subjects to remem-

Table 3
Successes and Attempts per Category by Subject, Total, and Percentage

Subject	Category			
	Recall	Cause	Trial and Error	Other
1	2/3			2/3
2		0/4	0/2	4/6
3	2/7	0/2	2/2	
4	1/1	1/4	2/5	0/1
5	0/1	1/4	1/1	2/3
6	1/1	2/5	1/4	0/1
7	2/3	2/4	0/3	0/1
8	2/2	2/2		0/1
9		3/8		1/1
10	1/1	2/6	1/5	0/1
11	1/3	1/1	2/8	0/1
Correct/Total	11/22	14/39	9/30	9/19
Percentage	50%	36%	30%	47%

ber the material in the notes. The material in the notes needs to be rewritten to ensure comprehensibility.

A final problem with the particular implementation was that the subjects did not always report the reasons for specific actions. This problem undermines the effectiveness of having converging evidence from the trace. Although this may be a problem with protocol research in general, in the present implementation there may be a solution. Since the computer records subjects' actions, at the specification of an action, the computer could prompt for a verbal justification. An alternative would be for the subject to choose between several presented justifications, removing the necessity for a protocol transcription at the cost of potentially limiting subject explanations.

A problem with the overall design of the research procedure involves the potential consequences of executing an inappropriate action. In some circumstances, inappropriate actions result in game termination and the subject must begin again. This clearly unpleasant result yields cautious behavior on the part of the subject. Unfortunately, this was not controlled in the pregame. Some subjects had their first experience with game termination in the pregame, whereas others did not encounter this situation until the research game (if at all). The subjects who had not encountered an unfortunate action would probably not be as hesitant to try an action as would those who had failed earlier. Controlling the number of previous failures would lead to a more balanced experiment. This is not as simple as it might seem. Since subjects choose their actions, choosing all the correct actions for whatever reason would lead to perfect performance, with no aversive consequences. The answer may lie in having the program decide on the correct solution path, depending on the performance of the subjects; there could be more than one correct solution, with only the second solution allowed to work.

A related problem involves the consequences of certain actions. For most of the problems, there is at least one way for the character to die. For one problem, however, no action has any unpleasant consequences. This may affect the subjects' exploratory behavior. Certain actions need to be consistently harmful to provide motivation for more effective strategies than trial and error, if they are available. The exposure that subjects receive to unpleasant consequences must be considered, to maximize the subjects' strategic behavior.

CONCLUSION

The results of this preliminary investigation suggest a certain need for revisions, lead to speculation about extension of this work, and imply the value of adventure games as a research environment. First, Voodoo Adventure clearly needs specific changes so that the game's playability and research viability can be improved. The problems with the wording need to be addressed, as does the feedback on actions; prompts for verbalization need to be added; the syntax problem needs to be addressed;

and the death rates need to be controlled. In addition, a major decision needs to be made about making the rooms more distinguishable, either by accepting slow game play or by reimplementation.

The status of the analogies in Voodoo Adventure also needs some investigation. Two different analyses of the analogies in the game are needed. First, the structures of the analogies between the notes and the problems must be carefully assessed to determine the degree to which they meet the criteria of matching on deep structure and not on surface structure. Second, the base rate of solution of the analogies must be examined. Ideally, these analogies would elicit the pattern of performance that has been observed in other forms of analogical research.

Reimplementation might be desirable for more reasons than just to resolve the tradeoff between speed and room discrimination. If this game is to be useful as a research environment, it should be available to other researchers as an environment within which to conduct research and across which to compare results. To facilitate portability, reimplementation in a language that is available on a number of platforms is desirable.

Either in HyperCard or in another implementation, additional manipulations to examine the details of subject reasoning are conceivable. One easy manipulation for examining the effect of recall on problem-solving ability is to give one group only one access to the notes while a second group has unlimited access. A manipulation could also be used to carefully examine where in the notes the subject refers, depending on the problem faced. The subjects' base rate of solution of the problems could be assessed through having a group without access to any notes at all.

Further manipulations would involve instructional applications. Levin and Kareev (1980) have suggested using adventure games for education by structuring the problems to incorporate domain material. This problem-solving environment could be used not only to investigate the cognitive skills involved but also as an environment within which to learn, use, discuss, and practice these skills. The Cognition and Technology Group at Vanderbilt (1990) has called for "anchored instruction," in which environments support extended exploration by students. This environment (or, ideally, a much extended version) could serve as the starting point for discussions of a broad variety of topics, including the role of knowledge, culture, and problem-solving. Similarly, the effects of instructional manipulations on subjects prior to the game could be compared to the effect of subjects' receiving no or neutral manipulations before the game. For example, training on analogical skills (Gick & Holyoak, 1983; Quinn, 1989) could be used as a basis for constructing a coaching component.

Another popular idea for computer applications in education is the intelligent discovery world (Shute & Glaser, 1990). This approach adds an artificially intelligent coaching component to an exploratory environment. Just as coaching has generally been applied in what Greenfield (1984) calls simulations, the same approach could con-

ceivably be applied in a computer adventure game, which is a simulation of another sort. I have argued previously (Quinn, 1988) that one area of research for computer instruction comprises cognitive skills. The empirical results on cognitive processing that come from further investigation with games of this sort should lead to a model robust enough to support such a component. A coaching component could even be introduced thematically as a coherent aspect of the situation. Imagine a mentor for an apprentice, a contact for an operative, or another form of available information that might occasionally be of use.

A similar approach that has been used in intelligent tutoring systems (Ohlsson, 1986) is to use a model of the students' capabilities to alter the difficulty of problems on the basis of an assessment of the current student user's ability. One problem with this approach might be the potential difficulty of creating problems with a flexible level of difficulty. Supporting a cognitively structured problem while providing a variable degree of challenge and maintaining a compelling story may be too difficult a task.

Before any further manipulations can be considered, an assessment needs to be made of the applicability of adventure games as a research environment. The current results only suggest the direction such an assessment might take; a complete evaluation awaits the results of the deeper analyses indicated above, from subsequent tests with an improved version. Despite the technical problems that have been encountered, the potential for effective remedies leaves the promise that these can be overcome. Anecdotal evidence and the results of a research subject pool survey suggest that the subjects found the environment motivating and felt situated enough so that they had few qualms about engaging in the task. Also, the collected data seem capable of supporting cognitive research. The categorizability of subject strategies is an encouraging sign. On the other hand, the environment is not directly comparable to other problem-solving research environments. For one thing, the actions of the subjects are more circumscribed by the environment. In addition, the style of problem to be solved is also different.

I believe, however, that the very characteristics that differentiate the game environment from previous research settings also indicate the need for research on the sorts of problems that are embedded in this setting. The exploration of problem-solving strategies in a motivating environment is a valid alternative area of research that can provide a new perspective on questions about what makes us intelligent. It also supports the potential for a new instructional setting within which to capitalize on such research. In some ways, it is a new adventure.

REFERENCES

- BROWN, J. S., COLLINS, A., & DUGUID, P. (1989). Situated cognition and the culture of learning. *Educational Researcher*, 18(1), 32-41.
- CAMPBELL, J. (1988). *The power of myth*. New York: Doubleday.
- CHI, M. T. H., BASSOK, M., LEWIS, M. W., REIMANN, P., & GLASER, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13, 145-182.
- COGNITION AND TECHNOLOGY GROUP AT VANDERBILT (1990). Anchored instruction and its relationship to situated cognition. *Educational Researcher*, 19(6), 2-10.
- DAVIS, W. (1985). *The serpent and the rainbow*. New York: Simon & Schuster.
- GENTNER, D. (1989). The mechanisms of analogical learning. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 199-241). New York: Cambridge University Press.
- GENTNER, D., & LANDERS, R. (1985). Analogical reminding: A good match is hard to find. In *Proceedings of the International Conference on Systems, Man, and Cybernetics* (pp. 607-613). Tucson, AZ.
- GICK, M. L., & HOLYOAK, K. J. (1980). Analogical problem solving. *Cognitive Psychology*, 12, 306-355.
- GICK, M. L., & HOLYOAK, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology*, 15, 1-38.
- GREENFIELD, P. M. (1984). *Mind and media: The effects of television, video games, and computers*. Cambridge, MA: Harvard University Press.
- HAYES, J. R., & SIMON, H. A. (1976). The understanding process: Problem isomorphs. *Cognitive Psychology*, 8, 165-190.
- KLATZKY, R. L. (1980). *Human memory: Structures and processes*. San Francisco: W. H. Freeman.
- LEVIN, J. A., & KAREEV, Y. (1980). *Personal computers and education: The challenge to schools* (Tech. Rep. No. 98). San Diego: University of California, Center for Human Information Processing.
- MALONE, T. W. (1981). Towards a theory of intrinsically motivating instruction. *Cognitive Science*, 5, 333-370.
- NISBETT, R. E., FONG, G. T., LEHMAN, D. R., & CHENG, P. W. (1987). Teaching reasoning. *Science*, 248, 625-631.
- NORMAN, D. A. (1986). Cognitive engineering. In D. A. Norman & S. W. Draper (Eds.), *User centered system design* (pp. 31-61). Hillsdale, NJ: Erlbaum.
- OHLSSON, S. (1986). Some principles of intelligent tutoring. *Instructional Science*, 14, 293-326.
- QUINN, C. N. (1988). Pedagogical framing and cognitive skills. *Technology & Learning*, 2, 1.
- QUINN, C. N. (1989). Analogical process performance and training. *Dissertation Abstracts International, Section B*, 50, 8, 3735B. (University Microfilms No. DA8926247)
- QUINN, C. N. (1990a). Analogical process performance. In *Proceedings of the Twelfth Annual Meeting of the Cognitive Science Society* (pp. 16-21). Boston, MA.
- QUINN, C. N. (1990b). *Screen design*. Manuscript submitted for publication.
- RATTERMAN, M. J., & GENTNER, D. (1987). Analogy and similarity: Determinants of accessibility and inferential soundness. In *Proceedings of the Ninth Annual Meeting of the Cognitive Science Society* (pp. 23-35). Hillsdale, NJ: Erlbaum.
- SHUTE, V. J., & GLASER, R. (1990). A large-scale evaluation of an intelligent discovery world: Smithtown. *Interactive Learning Environments*, 1, 51-77.
- WASON, P. C., & JOHNSON-LAIRD, P. N. (1972). *The psychology of deductive reasoning: Structure and content*. Cambridge, MA: Harvard University Press.

NOTES

1. The problems referred to here are problems for the subject to solve, not problems with the game. Until the discussion of problems with the particular implementation, following presentation of the results of an experiment that was carried out to examine the feasibility of the research, the use of the word *problem* will refer to the particular obstacles that the player must attempt to overcome.
2. However, 1 subject did repeat previously attempted actions. The apparent reason is that the subject lost track of the actions that had already been attempted.