What Makes Things Fun to Learn? Heuristics for Designing Instructional Computer Games

Thomas W. Malone
Xerox Palo Alto Research Center
Palo Alto, California

In this paper, I will describe my intuitions about what makes computer games fun. More detailed descriptions of the experiments and the theory on which this paper is based are given by Malone (1980a, 1980b). My primary goal here is to provide a set of heuristics or guidelines for designers of instructional computer games. I have articulated and organized common sense principles to spark the creativity of instructional designers (see Banet, 1979, for an unstructured list of similar principles). To demonstrate the usefulness of these principles, I have included several applications to actual or proposed instructional games. Throughout the paper I emphasize games with educational uses, but I focus on what makes the games fun, not on what makes them educational.

Though I will not emphasize the point in this paper, these same ideas can be applied to other educational environments and life situations. In a sense, the categories I will describe constitute a general taxonomy of *intrinsic motivation*--of what makes an activity fun or rewarding for its own sake rather than for the sake of some external reward (See Lepper and Greene, 1979).

I think the essential characteristics of good computer games and other intrinsically enjoyable situations can be organized into three categories: *challenge, fantasy,* and *curiosity.*

Challenge

In order for a computer game to be challenging, it must provide a goal whose attainment is uncertain. A number of important consequences follow from this simple principle.

Goal. There are several reasons for believing that goals are important to good computer games. In the first study of my dissertation, I surveyed grade school children with up to two years of computer game experience about their preferences in computer games. The single feature of the games that correlated most strongly with preference was whether or not the game had a goal. In a sense, the very notion of "game" implies that there is an "object of the game."

Not all goals are equally good, however. A study by Morozova (1955) gives some tantalizing suggestions about what good goals for instructional games should be like. In this study, children read several variants of a text passage about latitude and longitude. The version in which a child hero was faced with the practical problem of finding his location was much more interesting (and understandable) for the children than the other versions. The goal in this version had several intriguing qualities: (1) Using the skill being taught was a means to achieving the goal, but it was not the goal in itself. (2) The goal was part of an intrinsic fantasy as discussed below. (3) Because of the child hero, the goal was presumably one with which the child readers could identify.

The implication of these ideas for computer games is that even rich, responsive environments may be unappealing unless they provide an appropriate goal. More specifically:

1. Simple games should provide an obvious goal. Usually the more obvious and compelling the goal is, the better. Goals can be made obvious or compelling by the use of visual effects (Breakout) or fantasy (Hangman). (Note 1.)

2. A complex environment without built-in goals should be structured so that users will be able to easily generate goals of appropriate difficulty. For example, one of the nice things about a programming system like Logo is that it is often easy to think of things for a moving "turtle" to do. But unless beginners have some suggested projects of the right difficulty level, they might easily pick tasks that are discouragingly difficult.

3. The best goals are often practical or fantasy goals (like reaching the moon in a rocket) rather than simply goals of using a skill (like doing arithmetic problems).

4. The players must be able to tell whether they are getting closer to the goal. This *performance feedback* may or may not be the same as the *informative feedback* discussed below.

Uncertain outcome. A game is usually boring if the player is either certain to win or certain to lose. There are four ways to make the outcome of a game uncertain for players over a wide range of ability levels (or for the same player at different times.):

1. *Variable difficulty level.* Good computer games should be playable at different difficulty levels. The choice of difficulty level can be either:

(a) determined automatically by the program according to how well the player does (Breakout, drill-and-practice),

(b) chosen by the player (perhaps with ego-involving labels like "Cadet", "Captain", "Commander" in Star Wars), or

(c) determined by the opponent's skill (chess, Chase, etc.). I think one of the important reasons why competition is motivating is simply because it provides a challenge at an appropriate difficulty level. (Note that if the computer is the opponent, this case can be equivalent to either (a) or (b).)

2. *Multiple level goals.* Good computer games often have several different levels of goals. With this feature, players whose outcome is certain at one level of goal may still be challenged by another level of goal. In general, two levels of goals can be created by having a basic goal of accomplishing something in the game world (like popping balloons or destroying bricks) and then a metagoal of reaching the basic goal efficiently. At least two game features create this kind of metagoal:

(a) *Score-keeping.* With this feature, the metagoal is to get as high (or as low) a score as possible. The score can reflect the number of tries, the number of successes, the difficulty of the successes, the resources expended, etc.

(b) *Speeded responses.* With this feature, the meta-goal is to do something as fast as possible, or faster than a deadline, or faster than one's opponent.

Sometimes these features provide subgoals rather than metagoals. For example, in Breakout, one can play with a goal of simply getting a higher score for a long time before the goal of destroying all the bricks becomes reasonable. Some games, like Adventure, provide not just two, but a whole series of levels of challenge.

3. *Hidden information.* Many games, especially guessing games, make the outcome of a game uncertain by hiding information from the player or players and selectively revealing it. This feature seems to provoke curiosity and well as contributing to the challenge of the game.

4. *Randomness.* A final way of making the outcome of a game uncertain is to introduce randomness. Many gambling games seem to succeed almost entirely on the basis of this principle, and randomness can be used to heighten interest in many other kinds of games (Hammurabi, Adventure).

Self-esteem. Goals and challenges are captivating because they engage a person's self-esteem. Success in a computer game, like success in any challenging activity, can make people feel better about themselves. In fact, I think that people's self-esteem is often connected to their success even in activities like gambling that depend entirely on chance. The opposite side of this principle, is of course, that failure in a challenging activity, like a computer game, can lower a person's self-esteem and--if it is severe enough--decrease the person's desire to play the game again. One implication of this principle is simply that games should have a variable difficulty level so players can play at an appropriate level for their ability. Another implication might be that performance feedback should be presented in a way that minimizes the possibility of self-esteem damage. Note that there is a tension here between the need to provide clear performance feedback to enhance challenge and the need to not reduce self-esteem to the point where the challenge becomes discouraging rather than inviting.

Fantasy

Fantasies often make computer games more interesting. In general, games that include fantasy show or evoke images of physical objects or social situations not actually present. These objects or situations may involve varying degrees of social or physical impossibility from completely possible (running a lemonade stand in Lemonade) to completely impossible (moving instantly from one barrier to another in Chase). Non-fantasy games involve only abstract symbols.

Intrinsic and extrinsic fantasies. One relatively easy way to try to increase the fun of learning is to take an existing curriculum and overlay it with a game in which the player progresses toward some fantasy goal, or avoids some fantasy catastrophe, depending only on whether the player's answers are right or wrong. To a great degree, the fantasies used in this kind of game are interchangeable across subject matters. For example, Baseball and Hangman fantasies could be used just as well for arithmetic problems as for spelling problems, with people being hung or advancing around a baseball diamond depending on whether the arithmetic problems were worked correctly. In both of these examples, the fantasy depends on the use of the skill, but not vice versa. I would like to call this kind of fantasy *extrinsic fantasy* and contrast it with *intrinsic fantasy* where the skill also depends on the fantasy (See Figure 1).

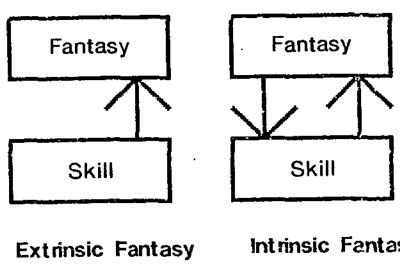

Figure 1. Logical dependencies in extrinsic and intrinsic fantasies

Most extrinsic fantasies depend only on whether or not the skill is used correctly (i.e., whether the answer is right or wrong), but other factors such as how quickly the answer is given or how close the answer is to being correct can also affect extrinsic fantasies. For example, a computer "race car" game where students' cars move along a race track depending on how fast they answer arithmetic problems is an extrinsic fantasy.

In *intrinsic fantasies*, not only does the fantasy depend on the skill, but the skill also depends on the fantasy. This usually means that problems are presented in terms of the elements of the fantasy world. For example, in Darts, the skill of estimating distances is applied to the fantasy world of balloons on a number line. The use of the skill then affects the fantasy world by shooting arrows and popping balloons. In intrinsic fantasies, the events in the fantasy world usually depend not just on whether the skill is used correctly, but on how its use is different from the correct use. For example, players of the Darts game can see graphically whether their answers are too high or too low and if so by how much.

Other intrinsic fantasies in math games include the search for a hidden animal in the Hurkle game and Snoopy shooting at the Red Baron in the Snoopy game. The Adventure game where a vast underground cavern system is explored in response to the players' commands can be considered an intrinsic fantasy for the skills of reading (the cave descriptions) and writing (the commands). Hammurabi and Lemonade are both simulations where a fantasy is intrinsic to the social science concepts being taught.

I think that: *In general, intrinsic fantasies are both more interesting and more instructional than extrinsic fantasies.* This seems to be true for several reasons. One advantage of intrinsic fantasies is that they often indicate how the skill could be used to accomplish some real world goal. Simulations, like the Lemonade stand simulation, are obvious examples of this. More importantly, when the fantasy in a game is intimately related to the material being learned, the players are able to exploit analogies between their existing knowledge about the fantasy world and the unfamiliar things they are learning. For example, players in the Darts game already know about physical objects (like arrows and balloons) being higher or lower than other objects. If they make the crucial connection between number size and position on the number line, then they are able use this old knowledge in the new domain to make inferences about the relative sizes of unfamiliar fractions.

Because extrinsic fantasies are domain-independent, it is possible to say more about them in general. To spark the imagination of designers of extrinsic fantasy games, Table 1 presents a list of extrinsic fantasies that have been or could be used in instructional computer games. One potential problem with extrinsic fantasies involving a fantasy catastrophe (like a person being hung) is that the catastrophe may be so interesting that players try to get *wrong* answers so they can see it.

Table 1

Extrinsic fantasies in which a fantasy goal is approached:

A train on a track is approaching a city.

A rocket is passing the other planets of the solar system on its way to earth.

A complicated building is being built, piece by piece

A fleet of space invaders is being destroyed, one by one

Extrinsic fantasies in which a fantasy catastrophe is avoided:

A man is hung, one body part at a time

A person advances toward the edge of a cliff, one step at a time

A time bomb is ticking toward an explosion

Emotional aspects of fantasy. Fantasies in computer games almost certainly derive some of their appeal from the emotional needs they help to satisfy in the people who play them. It is very difficult to know what emotional needs people have and how these needs might be partially met by computer games. It seems fair to say, however, that computer games that embody emotionally-involving fantasies like war, destruction, and competition are likely to be more popular than those with less emotional fantasies. In fact, a Freudian view that the two fundamental human drives are sex and aggression suggests that the current set of computer games based on aggressive fantasies may soon be joined by a whole new set of games based on sexual fantasies.

One obvious consequence of the importance of emotional aspects of fantasies, is that different people will find different fantasies appealing. For example, in my experiment with the Darts game, the boys seemed to like the fantasy of arrows popping balloons and the girls seemed to dislike it. If computer game designers can create many different kinds of fantasies for different kinds of people, their games are likely to have much broader appeal. For example, one can easily envision a math game, where different students see the same problems, but can choose which fantasy they want to see. Clearly the differences in fantasy choices may depend on many factors other than the simple sex difference found in the Darts study.

Curiosity

Curiosity is the motivation to learn, independent of any goal-seeking or fantasy-fulfillment. Computer games can evoke a learner's curiosity by providing environments that have an *optimal level of informational complexity* (Berlyne, 1965; Piaget, 1952). In other words, the environments should be neither too complicated nor too simple with respect to the learner's existing knowledge. They should be *novel* and *surprising*, but not completely incomprehensible. In general, an optimally complex environment will be one where the learner knows enough to have expectations about what will happen, but where these expectations are sometimes unmet. (Note that the optimal complexity I am talking about here is different than the optimal level of challenge discussed above. Challenge refers to what a player can do; complexity refers to what a player can understand.)

It is useful, in the following discussion, to distinguish between *sensory curiosity* and *cognitive curiosity*.

Sensory curiosity

Sensory curiosity involves the attention attracting value of changes or patterns in the light, sound, or other sensory stimuli of an environment. Mander (1978) proposes an extremely interesting hypothesis about how television artificially manipulates the sensory curiosity of its viewers to give television a kind of "counterfeit" interest value. He defines a "technical event" as a change of camera angle, a zoom, or some other similar technical change in the image being presented. He claims that these technical events artificially capture our attention independent of the content of the television program. According to his observations, television commercials average 20 to 30 technical events per minute, regular programming averages 8 to 10 technical events per minute, and public television averages only 2 to 3 technical events per minute.

Computer games can appeal to sensory curiosity by the use of *audio and visual effects*. There are several different ways of using these effects, some of which also appeal to other motivations:

(1) *As decoration.* When sound and graphics are used in a program regardless of what the player does, they can be considered "decorative" (e.g. circus music at the beginning of Darts). My conjecture is that this kind of effect will enhance the initial interest of a game, but will quickly become boring.

(2) *To enhance fantasy.* A special kind of "decorative" use of sound or graphics is to enhance the fantasy involved in a game (like the circus music mentioned above). In this case, the special effects are captivating not only in their own right, but also because of the fantasy associations they evoke.

(3) *As reward.* When sound or graphics displays are used to reward good performance, they can increase the salience of the goal and thus add to the challenge of the game. The challenge of reaching a goal, however, can sometimes distract people from exercising their curiosity, and so might decrease certain kinds of learning. (See Lepper & Greene, 1979, for an extensive review of research showing how rewards can sometimes undermine people's interest in a task and degrade their performance of it.)

(4) *As a representation system.* Perhaps the best use of sound and graphics in computer games is to represent and convey information more effectively than with words or numbers. For instance, the Darts game uses a graphic representation system for fractions and the Breakout game signals bounces and misses of the ball with different tones.

Cognitive curiosity

Cognitive curiosity can be thought of as a desire to bring better "form" to one's knowledge structures. In particular, I claim that people are motivated to bring to all their cognitive structures three of the characteristics of well-formed scientific theories: *completeness, consistency,* and *parsimony.* According to this theory, the way to engage learners' curiosity is to present just enough information to make their existing knowledge seem incomplete, inconsistent, or unparsimonious. The learners are then motivated to learn more in order to make their cognitive structures better-formed. For example, if you have just read all but the last chapter of a murder mystery, you have a strong cognitive motivation to bring completeness to your knowledge structure by finding out who the murderer was. As another example, Morozova (1955) suggests that a topic can be made more interesting by creating inconsistencies in a learner's knowledge. For instance, students may be told that plants require sunlight for the photochemical processes on which they depend, but that some plants, namely fungi, can live in the dark.

Informative feedback. Several more specific principles for designing games follow from these general ideas. One way of making environments interestingly complex is to make them responsive (see Moore and Anderson, 1969). In particular:

(a) *To engage a learner's curiosity, feedback should be surprising.* The "easy" way to do this is by using randomness. A deeper way to do this is to have environments whose underlying consistency is revealed by things that seem surprising at first. For instance, players of the Hammurabi simulation may be surprised at how many people starve at first. But the underlying relationships between amount of grain and number of people are consistent and can be discovered by the players.

(b) *To be educational, feedback should be constructive.* In other words, the feedback should not just reveal to learners that their knowledge is incomplete, inconsistent, or unparsimonious, but should help them see how to change their knowledge to become more complete, consistent, or parsimonious.

One extremely powerful tool for tailoring feedback to specific learners and thus maximizing their curiosity is to maintain on-line cognitive models of the users (Burton & Brown, 1979).

Combinations of features

One use of the above list of features is simply to suggest additions to existing or planned computer games. Almost no computer games include all the features mentioned above, and it is usually possible to think of ways that any given game could incorporate more of these features. For example, many popular computer games do not have any way of varying the difficulty level and could probably be improved by adding this.

Some computer games are successful because they are particularly strong on one of the above features, but most good computer games seem to have clever combinations of several of the above features. For example, Breakout has a visually compelling graphic display that both presents a goal and serves as a score-keeping device.

One particularly interesting class of games uses a compelling visual goal to get players started and then keeps them interested with curiosity-arousing unexpected consequences. For example, in Darts, players are presumably first attracted by the goal of popping the balloons. But if they have a misconception about fractions, say that increasing the denominator makes the fraction larger, then they will be surprised when an arrow they thought would be higher on the line turns out to be lower. This surprising feedback seems likely to both arouse curiosity and provide constructive information for correcting the misconception.

Applications

In order to illustrate the usefulness of the above principles, I will now apply them to several examples:

An arithmetic drill-and-practice program

In this example, I will suggest several relatively minor changes to an existing arithmetic drill-and-practice program that I think would make it more fun for students. I will stop short of suggesting major changes in the way the curriculum for this program is conceived.

In this program, the difficulty level of arithmetic exercises is automatically adjusted according to how well the student does, and the percent of problems correct is printed at the end of each lesson. At first glance, this automatic difficulty level adjustment appears to be a good way of maintaining the challenge of the program. But according to the principles described above, the first necessary element of a challenging environment is a goal. The only thing like a goal provided by this program is the percent correct printed at the end of each lesson, and some students do try to get "hundred percents". But this goal is not made particularly obvious or compelling, and given the automatic difficulty adjustment, it is actually fairly rare for students to get all their problems correct. In fact, since the difficulty adjustment is hidden from the students, the goal of getting all the problems correct may seem to students to be inexplicably receding as they approach it.

Aside from major curriculum revisions involving intrinsic fantasies and curiosity-driven learning, I think there are still a number of ways that extrinsic fantasies can be combined with goals and performance feedback to make this program more interesting. One of the simplest things to do is to select an extrinsic fantasy like those in Table 1--or better yet let the students themselves pick fantasies from a list. Ideally, this fantasy can be represented graphically and remain on the screen throughout a lesson as correct and incorrect answers affect a student's progress in the fantasy world. It would be nice to use sound effects for right and wrong answers. Reaching the final goal or catastrophe in the fantasy world should be accompanied by more elaborate sound and graphics. In addition to the first two levels of goals within a lesson (getting individual right answers and reaching the fantasy goal), the automatic difficulty adjustment can provide a higher level goal of making progress in the curriculum. If the extrinsic fantasy includes multiple levels of goals, the movement of a student to a higher difficulty level can be accompanied by even more fanfare in the fantasy world. Obviously, the details of these changes still have to be worked out, but perhaps this short description shows how the principles above can be used to suggest changes to existing programs.

A simple program to teach children how to tell time

In this example, I will suggest how to increase the interest of a proposed computer game for teaching the relationship between three different notational systems for time: clock face, digital display, and English words (Note 2). The original proposal for this system was to have the three different representational systems simultaneously displayed on the screen so that when the student changed any one representation, the other two changed accordingly. One insight afforded by the above list of principles is that this game has no apparent goal. A nice way to provide a goal would be to use an analogy of the Darts game. In this new game, a time is represented in one system, and the student tries to guess the time in one of the other systems. After each incorrect guess, either the incorrect guess in displayed in the target system or the student is simply told that the guess is too early or too late. The game might be even more interesting if it included an intrinsic fantasy about setting alarm clocks and being early or late for school.

Other educational applications

More generally, a game prototype can suggest analogous games in domains very different from the original ones. For example, the guessing game structure can be used to invent games to teach many different kinds of knowledge:

(1) *To teach an ordered list, use a guessing game that gives high/low feedback.* For example, to teach the list of U. S. Presidents in order, use a game where the players try to guess a secret President. After each guess, they are told whether their guess is before or after the secret President and perhaps how close it is. This kind of game can be used to teach either the contents of a list (U. S. Presidents, steps in a procedure, etc.) or the ordering relationship ("less than" and "greater than" in a number-guessing game).

(2) *To teach the correspondence between two representation systems, use a guessing game that gives hints in one system and asks players to guess in the other*

system. For example, the Darts game is designed to teach the relationship between numbers represented on a number line and in mixed number format. In the previous section, I just described a new game to help teach children how to tell time. This kind of game can be used to teach other correspondences like foreign language vocabulary, Cartesian coordinates for points on a plane (Hurkle), or spelling of words (Hangman).

(3) *To teach the properties of items in a set, use a guessing game where players guess properties of the target item (like Twenty Questions).* For example, Hollan, McCandless, Prince, Putz, Sharp, & Williams (1980) describe a Twenty Questions game to teach Navy crew members about the equipment on enemy ships. In this game the players try to guess the secret ship by asking questions about its equipment. Similar games could be used in almost any domain. For example, they could be used to teach the characteristics of different diseases to medical students or the climates and economies of various countries to geography students.

This technique of using structural analogies with old games seems to be a powerful way of inventing educational games in new domains.

Computer programming

In some senses, computer programming itself is one of the best computer games of all. In the "computer programming game", there are obvious goals and it is easy to generate more. The "player" gets frequent performance feedback (that is, in fact, often tantalizingly misleading about the nearness of the goal) The game can be played at many different difficulty levels, and there are many levels of goals available, both in terms of the finished product (whether it works, how fast it works, how much space it requires, etc.) and in terms of the process of reaching it (how long it takes to program, etc.). Self-esteem is crucially involved in this game, and there are probably occasional emotional or fantasy aspects involved in controlling so completely, yet often so ineffectively, the behavior of this responsive entity. Finally, the process of debugging a program is perhaps unmatched in its ability to raise expectations about how the program will work, only to have the expectations surprisingly disappointed in ways that reveal the true underlying structure of the program.

"Real life"

There are even times in "real life" when some of these principles can be applied to make a boring situation more interesting. For example, when someone "makes a game out of" something, I think it means that they select a goal--usually a higher order one ("Let's see how fast we can wash all these dishes")--and perhaps a fantasy ("Let's pretend that all the dandelions are Klingons and we are trying to destroy them.").

Conclusion

With the dramatically decreasing costs of computer technology, the spread of microcromputers in homes and classrooms appears almost inevitable. But it is up to us to see that these new educational applications use the unique capabilities of computers to make learning more efficient, more interesting, and more enjoyable. I hope the ideas in this paper will point the way toward that goal.

Notes

I would like to thank John Seely Brown, Mark Lepper, Patrick Suppes, and Stuart Cooney for helpful comments. The research on which this paper is based was supported by the Xerox Palo Alto Research Center and a National Science Foundation Graduate Fellowship.

Note 1: For those readers who are not familiar with the computer games I use as examples, the appendix includes brief descriptions of all the games mentioned.

Note 2: The original version of this system was suggested by Laura Gould.

References

Banet, B. Computers and early learning: A new direction for High/Scope Foundation. *Calculators/Computers*, 1979, *3*, 17.

Berlyne, D. E. *Structure and direction in thinking.* New York: Wiley, 1965.

Burton, R.R. & Brown, J.S. An investigation of computer coaching for informal learning activities. *International Journal of Man-Machine Studies*, 1979, *11*, 5-24.

Hollan, J., McCandless, T., Prince. R., Putz, S. Sharp, J., & Williams, M. *Tactical memorization software.* Technical Note, Navy Personnel Research and Development Center, San Diego, Calif., forthcoming (1980).

Lepper, M. R., & Greene, D. *The hidden costs of reward.* Morristown, N. J.: Lawrence Erlbaum, 1979.

Malone, T. W. *What makes things fun to learn? A study of intrinsically motivating computer games.* Unpublished Ph.D. dissertation, Department of Psychology, Stanford University, Stanford, Calif., 1980a.

Malone, T. W. *What makes things fun to learn? A study of intrinsically motivating computer games.* Technical report, Xerox Palo Alto Research Center, Palo Alto, Calif., forthcoming, (1980b)

Moore, O. K. & Anderson, A. R. Some principles for the design of clarifying educational environments. In Goslin, D. (Ed.) *Handbook of socialization theory and research,* New York: Rand McNally, 1969.

Morozova, N.G. [The psychological conditions for the arousal and modification of interest in children in the process of reading popular scientific literature.] *Izvestiia Akad. Pedag. Nauk,* 1955, *73,* 100-149. (cited and summarized by Berlyne, 1965.)

Piaget, J. *The origins of intelligence in children.* N.Y.: International University Press, 1952.

Appendix

Descriptions of games mentioned in text

Game	***Description***
Adventure	The player explores a vast underground systems of caves with dragons, etc. trying to find treasures
Breakout	The player controls a paddle to hit a ball that breaks bricks out of a wall
Chase	Two players chase each other across an obstacle course.
Darts	A game designed to teach elementary students about fractions. Three balloons appear at random places on a number line and students try to guess where the balloons are. They guess by typing in mixed numbers, and each time they guess an arrow shoots across the screen to the position specified. If the guess was right, the arrow pops the balloon. If wrong, the arrow remains on the screen and the player gets to keep shooting until all the balloons are popped. Circus music is played at the beginning of the game and if all three balloons in a round are popped in four or fewer tries, a short song is played.
Hangman	The player tries to guess a word, letter by letter. After each incorrect letter guessed, one more body part of a man being hung is drawn. The player loses if the whole body is drawn.
Hammurabi	Player acts as king of ancient Babylonia and decides each year how much wheat to plant, how much to store, and how much to save. There are occasional plagues, rat infestations, etc. The number of people who are born, starve, etc. each year is reported.
Hurkle	The player tries to guess where an animal called a "Hurkle" is hiding in a Cartesian coordinate grid.
Lemonade	The player runs a lemonade stand buying supplies, advertising, etc.
Logo	A sophisticated programming environment where children control the motion of a "turtle" on the screen or on the floor.
Snoopy	Snoopy and the Red Baron appear at different positions on a signed number line. Player says how far Snoopy should shoot to hit the Red Baron (as a signed integer).
Star Wars	Player tries to shoot down enemy space ships as they appear in a graphic display.