AGenT

AGen**T** stands for Automated Genetics Tutor. It is a computer program written by Bill Sofer of the Waksman Institute at Rutgers University and Alan Gerstein of Pharmacia/LKB Biotechnology. It has been designed to supplement a "module" – a set of laboratory exercises – entitled "*Molecular Aspects of Gene Activity*" that was developed under the auspices of an NSF/Merck-sponsored grant to Bill Sofer and George Pallrand of Rutgers. AGenT is SchoolWare. See the note below for information on ordering AGenT and the complete module.

AGenT (like the module that it supports) is intended for high school students, although there is no reason that younger pupils – even grade school students – could not make use of it. Although a single individual can use it, we envision that most schools will encourage groups of 4-6 students at a time to gather around the computer with one student at the keyboard/mouse and the others directing him/her (there is little premium on manual dexterity - this is not a video game). The program runs on the Apple Macintosh II series of computers and requires at least 8 bit color (256 colors) capability.

Programming

AGenT is written in the SuperTalk language (a part of the SuperCard environment). Supertalk is like HyperTalk (the programming language of HyperCard) – it is a so-called scripting language – but is much more powerful. For example, it is possible to produce a true stand-alone program with its own icon in SuperCard, so that students do not have to buy either SuperCard or HyperCard. In addition, the SuperCard environment makes use of draw graphics and color to a greater extent than HyperCard. Finally, SuperTalk has many additional commands for animation than HyperCard.

We have decided to program in SuperTalk rather than Pascal, BASIC or C – more traditional programming languages – because of the rapidity with which programs can be fashioned. On the other hand, SuperTalk is interpreted, and therefore it runs more slowly than compiled languages.

General characteristics

AGenT is part simulation of a laboratory exercise, and part adventure game. In addition, it contains elements of a tutorial. Through it all, the student (or groups of students working together) try to solve a problem in biochemical genetics by exploration, trial and error, and by the use of simulated scientific tools and abstract reasoning. The program tries to strike a balance. On the one hand, we try to prevent students from being frustrated by the difficulty of finding a solution to the problem posed. On the other, we want to provide a sufficient challenge so that they have a reason to explore the subject matter, and learn to use the tools provided, and thereby solve the problem on their own.

The program – a brief description for teachers

Upon opening the program, the student sees a corridor and an entrance to a suite of genetics laboratories. He/she must type in their name, pick up an unknown vial of flies, and open the door. The door will not open without signing in or taking the vial. The method for carrying out an action is to place the arrow cursor on the object (for example the vial, or the door handle) and click the mouse. The disappearance of the vial, or the opening of the door provides visual feedback, and students very quickly adapt to this simulated environment.

The problem that the program provides is to determine whether the vial presented at the onset contains flies with alcohol dehydrogenase enzyme activity or not. In the module that complements the program, there are multiple ways of making this determination, and when the door to the genetics laboratories is open, the student is presented with several laboratory each of which corresponds to a specific technique. We have only implemented one of these techniques in the computer program: the ethanol selection laboratory.

Once "in" the laboratory, the students are presented with a laboratory bench with drawers and cabinets, a blackboard and a bulletin board. On top of the bench is their vial of unknown flies. The blackboard welcomes them to the lab (by name) and tells them that their task is to determine the state of alcohol dehydrogenase activity of the flies in their vial . There are no other explicit directions. The only menu item (the Macintosh paradigm makes use of "pull-down" menus for guidance) is one that allows the student to leave the program. Students are thereby forced to explore. Exploration is accomplished by moving the cursor around the screen and clicking the mouse button when the cursor "points" to specific objects. Students quickly find that the drawers and cabinets open (accompanied by appropriate sounds), that materials and tools in the cabinets can be brought on to the desk top, and that these materials can be made to do appropriate things. In this particular exercise, the flies are emptied in to a vial containing ethanol, and then tested to see if they survive. Appropriate controls must be carried out if the student hopes to come to a correct conclusion.

Laboratory manual

In addition to this simulation, one of the drawers contains a laboratory manual. This too can be taken out and examined. Students do not have to read the manual before carrying out the exercises (although they may), but if their interest has been piqued by the simulation or if they are just curious, they can explore the subject of biochemical genetics and Drosophila biology in greater depth at any time at any time during the program.

Conclusion

In the end, the students report on the status of their unknown. The program then tells them whether they are right or wrong (the computer randomly puts negative or positive flies in the unknown vial). If they are successful, they earn a diploma However, it is not so much the answer that is important, rather in the process of working the program the students become acquainted with the materials and methods, and the dimensions of the problem and solution. When they actually go into the laboratory, they should be ready for the real exercises, their interest primed.

Pitfalls

There are several criticisms that have been raised about the program.

Isn't the Macintosh computer (especially color versions) expensive? Won't many schools have others brands? Why not develop programs for a machine that is cheaper and more popular ?

There are several responses to these questions. First, this program (and the others that we intend to generate) are complex. It would be very difficult to develop them on computers with less power. Second, the computer industry is changing very rapidly. New machines are coming to market with increased power at ever decreasing prices. In this environment, old machines quickly become outmoded. Because we are shooting at a moving target , it is better, in our opinion, to direct our programming efforts at a more modern machine rather then aim at outmoded ones. In this regard, relatively inexpensive Macintosh computers have recently come on the market, and they promise to become quite popular. Of course we aren't wedded to the Macintosh line, and if new developmental platforms become widely available we will port our programs to them. Third, computers have often been purchased by schools in response to vague pressures such as a need for 'computer literacy'. It is a better principle to buy machines because they meet particular needs and run suites of specific programs. The presence of good software for teaching science has been lacking, and when such software arrives it will encourage teachers and administrators to buy appropriate computers.

Mightn't simulations replace real laboratory exercises?

We fell strongly that our program is a supplement to the laboratory. It is not intended to substitute for a genuine laboratory exercise. Rather it is designed to familiarize students with the tools and aims of exercises and to stimulate them to work in the laboratory.

Aren't some students frightened by computers? Might they not be put off by the programs?

There are several answers to these questions too. First, many students have been frustrated and frightened not by computers but rather by the quality of the programs that they were forced to work with. We have tried to design our program so that it is easy to use and stimulates the user. Second, while some students will not care to use computer technology to learn elements of biology, others may be turned on by the experience. It is important to provide a variety of tools that students can work with so that different students find the right ones for themselves.

Guiding principles in program development

In formulating AGenT (and other programs that will accompany the other modules), we have tried to follow certain principles. They are listed below.

1. The program should <u>not</u> replace the original authentic laboratory experience. It is important for students to handle real materials, to see real organisms, and to actually carry out hands-on exercises. Computer-based simulations of laboratory exercises are tools that can augment the real laboratory, but are not substitutes.

2. The program should stimulate curiosity by encouraging exploration. Unlike a laboratory manual, a computer program should present a non-linear route of inquiry, allowing students to easily investigate different paths towards determining a solution to a problem.

3. It should provide the students with a enhanced experience by allowing them to perform certain experiments that might not be carried out in the classroom because of a lack of sophisticated equipment or because some experiments might be too hazardous or time consuming.

4. It should be very easy to use. There should be little typing. The program should have little text and many illustrations. The use of equipment and apparatus should be similar to the real thing. Little instruction in the use of the program should be required, and when it is, it should be provided in as simple and easily accessed manner as possible.

5. It should allow self instruction at a pace that is set by the user or groups of users.

6. As in a genuine laboratory exercise, a real problem should be posed. Many different routes to correct solutions should be allowed.

Information

AGenT is being distributed as SchoolWare. That is, it may be copied freely and shared among teachers, students and other interested parties. However, if you are a teacher and you use it in your classes, we request that you send a check for $100 made out to **Rutgers University** to:

Dr. William Sofer
AGenT
Waksman Institute
Rutgers University
Piscataway, NJ 08854-0759

Any funds that are forthcoming will be used in support of the development of software for instructional purposes. Comments should be directed at Bill Sofer (sofer@mbcl.rutgers.edu) or by mail to the above address.