Tech News

Sound Crew

Last week we learned how the school's PA system works and practiced operating it. The students had fun playing with the different digital effects and undoubtedly annoyed several of the people looking at the science fair projects in Fellowship Hall. (Actually, the digital effects weren't annoying, but the occasional feedback squeals were.)

I had one mistake in last week's *Tech News:* the microphones that the school has are not condenser microphones but dynamic microphones.

Running scratch or Alice at home

There are pointers to the version of scratch we are using (as well as some example programs) at

http://www.soe.ucsc.edu/~karplus/scratch_programs

I've also put PDF files with the back copies of *Tech News* there, so parents of kids who join Tech Club late can still read all the issues.

There are also pointers to Alice, a somewhat more sophisticated language for beginning programmers based on 3D animation.

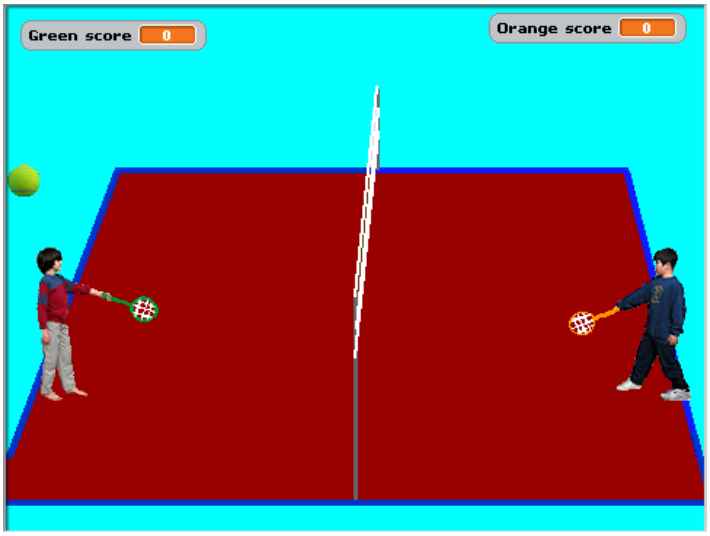
Programming tip

This week we'll talk a bit about game design and game programming. There are several things that distinguish a game from other animations:

- user interaction. There may be one player or multiple players, but they must be able to affect what happens on the screen.
- an objective. Just having things interact isn't enough to be a game—the player must have some desired goal.
- scoring system. Although some games have just a single win/lose result (like solitaire), many have scoring systems that allow the player to keep track of progress.

Having a program respond to player inputs is fairly straightforward in scratch, relying on the control blocks "when <key> is pressed" and "when <sprite> is clicked" or the sensor values "mouse x", "mouse y", "mouse down?", "<key> pressed?", "touching mouse pointer?", "distance to mouse-pointer", "loudness", and "loud?. (The loudness sensors only work if the computer has a microphone.)

For beginning programmers, it is often easier to work with keyboard commands than with the mouse. The arrow keys are often used for moving left, right, up, or down on the screen. Two-player games can be set up by using keys on the left or right side of the keyboard to control actions. For example, in the Tennis game that Abe and I wrote together, the left-hand player uses "r" to move up, "f" to move down, and "space" to raise the racket and serve. The right-hand player uses the up and down arrows to move and the left arrow to raise the racket and serve.



The player sprites just handle the keyboard input and movement of the players. All the action and scoring is handled by the tennisball sprite. We edited the photos used for costumes, so that the rackets would have a single distinctive color, to make it easier for scratch to detect when the ball touches the rackets.

Scoreboards are easily made by creating variables, which can be displayed on the screen by clicking the checkbox next to the variable. There are 3 different display options: displaying with the variable name (good for score boards and for debugging), displaying the value alone, and displaying with a slider for inputting values. The Tennis game has the scores as global variables that are initialized by the stage and incremented by the tennisball sprite.

One tricky part of many games is making sure that the program stops responding to most user inputs when the game is over. The Tennis game uses as "Game Over" variable to communicate this information to the sprites.