Spring Hill Tech Club Volume 2, # 14 17 Jan 2008

Tech News

Doodle

One of the demo programs that comes with v1.2.1 Scratch was written by a Tech Club member (Graham, who was in Tech Club last year). That program is Doodle (called “6 Doodle” in the v1.2.1 release). Several of the Tech Club kids have played Doodle, so today’s column will be about how the sprites work.

Most of the scripts are for the “guy” sprite, though each level of the game also has an “owie” sprite. There is also the “pen” sprite for the player’s input, and the “door” sprite to be the goal for each level.

The “door” is always in the same place and just shows at the beginning and hides when the game is won.

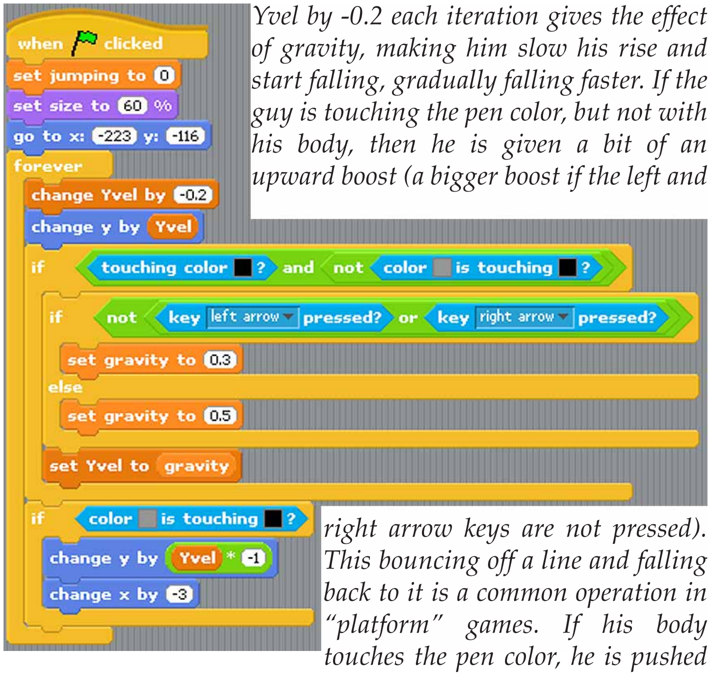
The “owies” are also simple. Initially they are hidden, but when the “redo” message is broadcast, they check the level and show themselves if the current level is theirs (the scripts for “owie 5” do this most cleanly). Several of the owies also have a movement forever loop, started by the green flag. It might have been better to put these movement loops as “repeat until” loops in the script that checks the level, so that the movement is only done for the right level.

The “pen” sprite is a bit more interesting. It runs a forever loop, going to the mouse pointer and either putting the pen down (if the mouse is pressed) or putting the pen up (if the mouse is not pressed).

The “guy” does most of the interesting stuff in the game. His stick figure costume has three different colors: body, front leg, rear leg. One forever loop detects whether the front leg is touching the pen’s color (in which case he turns counterclockwise, lifting his front leg a bit) or his back leg is touching the pen’s color (in which case he turns clockwise, raising his back leg). In another script, if the guy is not touching the pen color, he is gradually rotated back to his normal, upright position. (It might have been clearer to combine all these rotations into one script.)

There is a forever loop detecting whether the guy is touching an owie (or too low on the screen), and restarting the level by sending a “redo” message. There are also key-press scripts to move left and right.

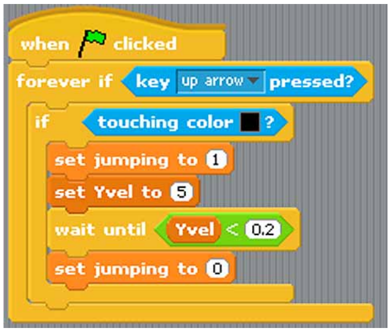
The most complicated scripts do the gravity simulation and jumping.



```
when ⚑ clicked
set jumping to 0
set size to 60 %
go to x: -223 y: -116
forever
    change Yvel by -0.2
    change y by Yvel
    if touching color ■ ? and not color ▨ is touching ■ ?
        if not key left arrow pressed? or key right arrow pressed?
            set gravity to 0.3
        else
            set gravity to 0.5
        set Yvel to gravity
    if color ▨ is touching ■ ?
        change y by Yvel * -1
        change x by -3
```

Gravity script: *Yvel is the upward velocity—his y-position is changed by the Yvel on each iteration of the loop. The change in Yvel by -0.2 each iteration gives the effect of gravity, making him slow his rise and start falling, gradually falling faster. If the guy is touching the pen color, but not with his body, then he is given a bit of an upward boost (a bigger boost if the left and right arrow keys are not pressed). This bouncing off a line and falling back to it is a common operation in “platform” games. If his body touches the pen color, he is pushed down and back. The variable gravity is badly named, as it is how fast the guy is being kicked upward by being on a line (or when jumping)—it could be eliminated, just changing the Yvel.*

Jumping script: *If the up arrow is pressed and the guy is on the line, set the upward velocity high and wait until the guy has started falling again. The jumping variable never seems to be used, and could be eliminated from the program.*



```
when ⚑ clicked
forever if key up arrow pressed?
    if touching color ■ ?
        set jumping to 1
        set Yvel to 5
        wait until Yvel < 0.2
        set jumping to 0
```

Zeum

During winter break, our family went to San Francisco. While there, we visited a kids’ museum we had not previously visited: Zeum at Fourth and Howard in Yerba Buena Gardens. Zeum is an interactive museum for creating art (animation, TV with green-screen technology to add backgrounds, making masks, using Photoshop Elements to manipulate images, making music, …). It is not as big and exciting as the Exploratorium, but well worth a visit. (Don’t go during winter break for SF schools—it was so crowded that the wait time for the stop-action animation studio was over 2 hours.)