Soundtrack

Author: Keith Alcock 09/07/2008 07:07:00 PM MDT

VITAL INFORMATION

Subject(s): Careers, Computer Fundamentals 1-2

Topic or Unit of Study: Software Development

Grade/Level: 9-12

Objective: At the conclusion of this lesson students will be able to:

1. Declare, define, assign to, and read Scratch variables.

2. Use a repetition structure (repeat until) and control its condition.

3. Compare values for equality.

4. Implement an if/else decision statement.

Summary: Students add a sound track to their Scratch animations in order to learn and practice a number of programming constructs. The sound track plays only during the action portion of the animation. Despite the stage using a broadcast and wait block and the musician potentially looping forever, animation action is allowed to complete so that the credits can roll. Sound is muted and unmuted when the user presses the space bar.

IMPLEMENTATION

Learning Context: Numerous students are still working on their Scratch animations and have difficulty incorporating two of the requirements: variables and user interaction. They now have headphones and soundtracks on finished animations have been well received. This lesson provides a way to meet the animation requirements and teaches by transmission what other students have learned in a constructive manner working on their own animations or can recall from an earlier Scratch exercise. Those students can continue their other work while this extra lesson should help the struggling students catch up.

Procedure: 1. Find out which students have not yet incorporated variables or user interaction into their programs and gather them around the center table where they can either see that computer screen or the projection screen and at the same time hear me and the computer.

2. Begin with the project file from the splash and credit screen exercise, which they will probably use for their own animation. The code that needs to be written has been saved in the musician sprite and can be loaded straight into the splash and credit screen project to demo the result. After the demo, delete it and start building the code.

3. The initial construction steps are approximately as follows: create a new sprite, name it Musician, and hide it; have it listen for introduction, action, and conclusion messages; import one of the music loops and play it in a forever loop in the action phase and show how this will prevent the animation from completing; to allow completion, broadcast a play message but don't wait for it; move the forever loop to where play is received; in the conclusion stop all sounds; and finally test.

4. Students will probably notice that the sound plays through the credits because it is restarted in the forever loop. To prevent this, a variable can be used. Add one called concluded, set it to 0 in the introduction and 1 in the conclusion, then change the forever to repeat until conclusion=1. Now the sounds shouldn't run over into the credits.

5. For user interaction listen for when space is pressed. If the volume is 100%, make it 0%, and vice versa. This can be achieved with an if/then/else block. Set the volume to 100% in the introduction. All should be functional now.

6. Make a screen shot of the completed code so that they can enter it with their own hands at their own computers.

Differentiated Instruction: There is little differentiation in this assignment. Students may work at their own pace at their own computers and choose their own musicians and music loops. Not all students need the lesson as some have met the standard prior to it.

Sample Student Products: Student products should very closely resemble the screen shot that accompanies the lesson plan.

Collaboration: Students will work individually.

Time Allotment: 1 class period. 30 Min. per class.

Author's Comments & Reflections: Reflections will follow in a diary entry.

MATERIALS AND RESOURCES

Instructional Materials: Blank sound track project, screen shot of completed work (just in case the shot in class doesn't work).

Attachments

1. **Musician**

Resources:

- Technology resources:
  Scratch

STANDARDS & ASSESSMENT

Standards:

AZ- Career and Technical Education Programs

- **Level :** Career Preparation (Grades 10 - 12)
  - **Program :** Information Technology CIP No. 15.1200
    - **Option :** Software Development - Option C
      - **Competency :** 27.C DEMONSTRATE PROGRAM ANALYSIS AND DESIGN
        - **Indicator :** 27.6c Use stepwise refinement to improve design
      - **Competency :** 28.C USE SOFTWARE TO CREATE PROGRAMS
        - **Indicator :** 28.1c Enter and modify code using a program editor
      - **Competency :** 29.C TEST AND DEBUG TO VERIFY PROGRAM OPERATION
        - **Indicator :** 29.1c Test individual program modules
      - **Competency :** 32.C WRITE CODE USING CONDITIONAL STRUCTURES
        - **Indicator :** 32.1c Compare values using relational operators (=, >, <, >=, <=, not equal)
        - **Indicator :** 32.2c Evaluate Boolean expressions
        - **Indicator :** 32.4c Construct decision statements such as if/else, if, switch case
        - **Indicator :** 32.6c Implement multiple-choice decision statements such as if/else, if, switch case
      - **Competency :** 33.C UTILIZE REPETITION STRUCTURES
        - **Indicator :** 33.1c Identify various types of repetition structures
      - **Competency :** 34.C USE SIMPLE DATA TYPES AND STRINGS
        - **Indicator :** 34.1c Declare numeric, Boolean, character and string variables
        - **Indicator :** 34.4c Write assignment statements for initializing and modifying variables

Assessment/Rubrics: The sound track should show up in the final animation without much problem. Variables and user input are evaluated there already. If students followed this lesson perfectly, it is fair to give them 100% even though the sound track wasn't their idea. Many other students were helped substantially but informally on the variable and user input requirements. If there is time to evaluate independently of the animation, students can simply demonstrate the sound track.