Drexel-SDP GK-12 ACTIVITY

Introduction To Scratch

Subject Area(s) **Science & Technology**

Associated Unit **Scratch Computer Programming**

Associated Lesson

Activity Title **Introduction To Scratch**

Grade Level 5,6

Time Required **30 minutes**

Group Size **1-2 (depending upon the number of computers available)**

Summary

This activity is designed to introduce students to the Scratch programming environment. Students will learn about the different user interface components. They will create several short programs to become familiar with changing backgrounds and sprites.

Engineering Connection

Engineers often use computers in a variety of activities such as modeling or analyzing data. There are many software packages that allow engineers to write simple programs to complete their tasks. Mechanical and Civil engineers use drawing software to illustrate objects or layouts. Electrical engineers use software to design circuits or execute simulations. No matter the type of engineer he or she has to learn how to the software.

This lesson is to be used to introduce students to the Scratch programming environment. In approximately 20 minutes students who have never written software are able to write several programs and see them execute.

Keywords

Computer Science, Software, Programming, Technology, Scratch

PA Educational Standards

- Science and Technology: 3.7 D – Apply computer software to solve specific problems

Pre-Requisite Knowledge

None

Learning Objectives

After this lesson, students should be able to:

- Understand how to create their own programs in Scratch consisting of:
    - Changing backgrounds
    - Changing sprites
    - Writing short programs to animate sprites
    - Creating dialogue with multiple sprites

Materials List

Each student needs:

- A computer with the Scratch environment installed.

To share with the entire class:

- A projector that can display images to a screen

Introduction / Motivation

Engineers use computers as tools to make their jobs easier and make certain tasks faster. For example, instead of calculating twenty-five data points by hand and then graphing them on paper an engineer may use a spreadsheet program. Using this program will not change their analysis but will enable them to do it faster than using a pencil and paper. If an Electrical Engineer wanted to see if part of an electrical system would be compatible with another part she might make a model of the system and then run a simulation on the model.

No matter what the goal of the computer use each engineer is able to use the computer to make their job a bit easier, not doing calculations by hand, or see things before they build it, like an electrical system. Each engineer must learn to use the computer program that enables him or her to do their job.

Scratch, a children's programming environment created at MIT, was created to allow middle school students to develop their own software. There are a lot of programs that you can create using Scratch to make your own digital stories, video games, and animations. If you have never written a computer program before you will learn how simple and easy it can be. In this activity you will learn about the different features of Scratch and how to put them together to make a simple program. Once you learn the basics you can write more complex software.

Vocabulary / Definitions

Word	Definition
Scratch	Programming environment developed at MIT, www.scratch.mit.edu
Software	A program, set of instructions, used to operate a computer or other device
Computer Science	The study of the foundation of computation and information implemented and applied in computer systems.

Procedure

Background

Students do not require any background of computer programming. It is assumed that they are familiar and comfortable using computer components, i.e. keyboard, mouse.

Before the Activity

- Discuss the use of computers in student's daily lives. Explain how benefit from computer programs. Share examples of how various science and engineering fields make use of computing.
- Ask students if they could give you walking directions from one corner of the room to another, turn around, and then get you back where you started. Be sure their instructions include phrases like "turn left", "walk 30 steps", etc. Ask them to be more specific if they use phrases like "go forward" or "turn around". To guide them towards shifting their thinking from "go forward" to "walk 30 steps", ask them how much forward to go. "Turn around" can be changed to "turn 90 degrees". In front of the class down student instructions. Inform the students that this process is similar to writing computer programs.

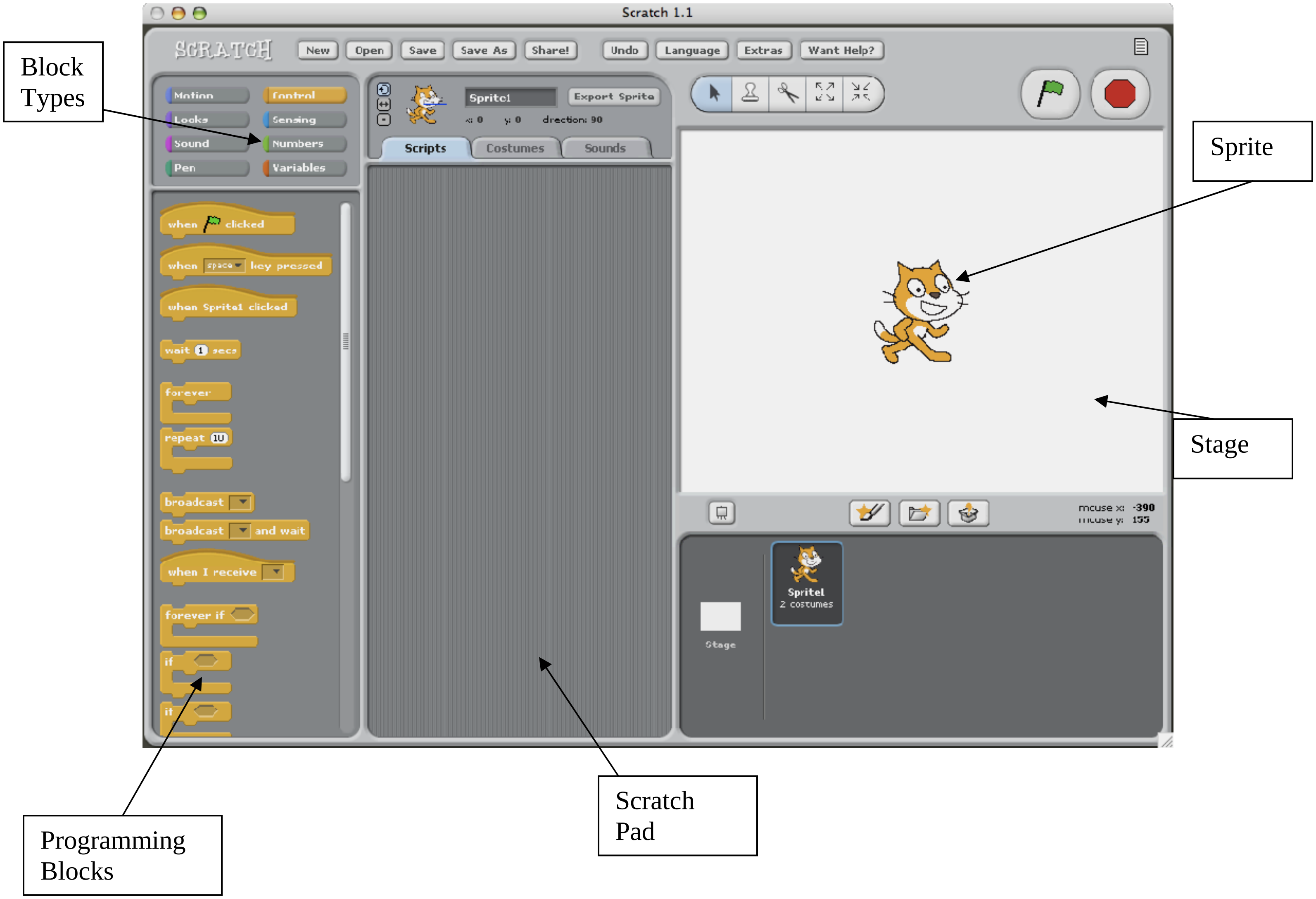
With the Students

1. To begin this exercise with a new Scratch file select "New" on the main Scratch menu.
2. Point out the various parts of the interface:
   a. Point out the large white **Stage** on the right most side.
   b. Explain that the cat is called a **Sprite** and that is similar to a character in a video game
   c. Point out the middle grey area known as the **Scratch Pad.**
   d. Point out the code blocks on the left most portion of the interface. Describe the various categories and allow students to select each one. Point out that the colors change and that each category contains different types of programming blocks.
3. Changing The Background
   a. Select the white stage icon in the lower right.
   b. Select the middle tab, Background, in the Scratch Pad area.
   c. Press the Import button and allow students to look through the various folders and select a background they like.
4. Changing The Sprite
   a. Press the 'Chose New Sprite' button below the white stage.
   b. Allow students to select one sprite they would like to use with their background.
   c. If necessary, use the shrink sprite and grow sprite buttons to adjust the sprite to the appropriate size.

5. Delete The Cat Sprite
    a. Select the scissors icon
    b. Select the cat to remove it from the stage.
6. Create The First Script
    a. Select the sprite icon in the window below the stage.
    b. Select the Scripts tab in the Scratch Pad
    c. Drag the "move 10 steps" block to the center of the Scratch Pad.
    d. Ask the student to predict what will happen when they run the program, by pressing the large green flag on the upper right.
    e. Nothing will happen. Explain that here needs to be a control so the sprite knows *when* to move the 10 steps.
    f. Select the **Control** group of blocks from the left of the interface.
    g. Drag the "When flag clicked" block to the Scratch Pad and demonstrate how the blocks snap together.
    h. Ask the student to predict what will happen when the green flag is pressed.
    i. Allow them to change the value 10 to 100 and run the program again.
    j. Select the **Looks** group of blocks and drag a think block below the blue move block. Replay the program.
7. Create The Second Script
    a. Allow the student to select a second sprite.
    b. Select that sprite icon in the area below the stage.
    c. Add motion and looks blocks to have the second sprint move and talk.
    d. Use a different control block such as "when space key pressed" or "when Sprite clicked" for the second sprite.
    e. Run the program using the different controls and allows the student to observe how their code affects the sprites.

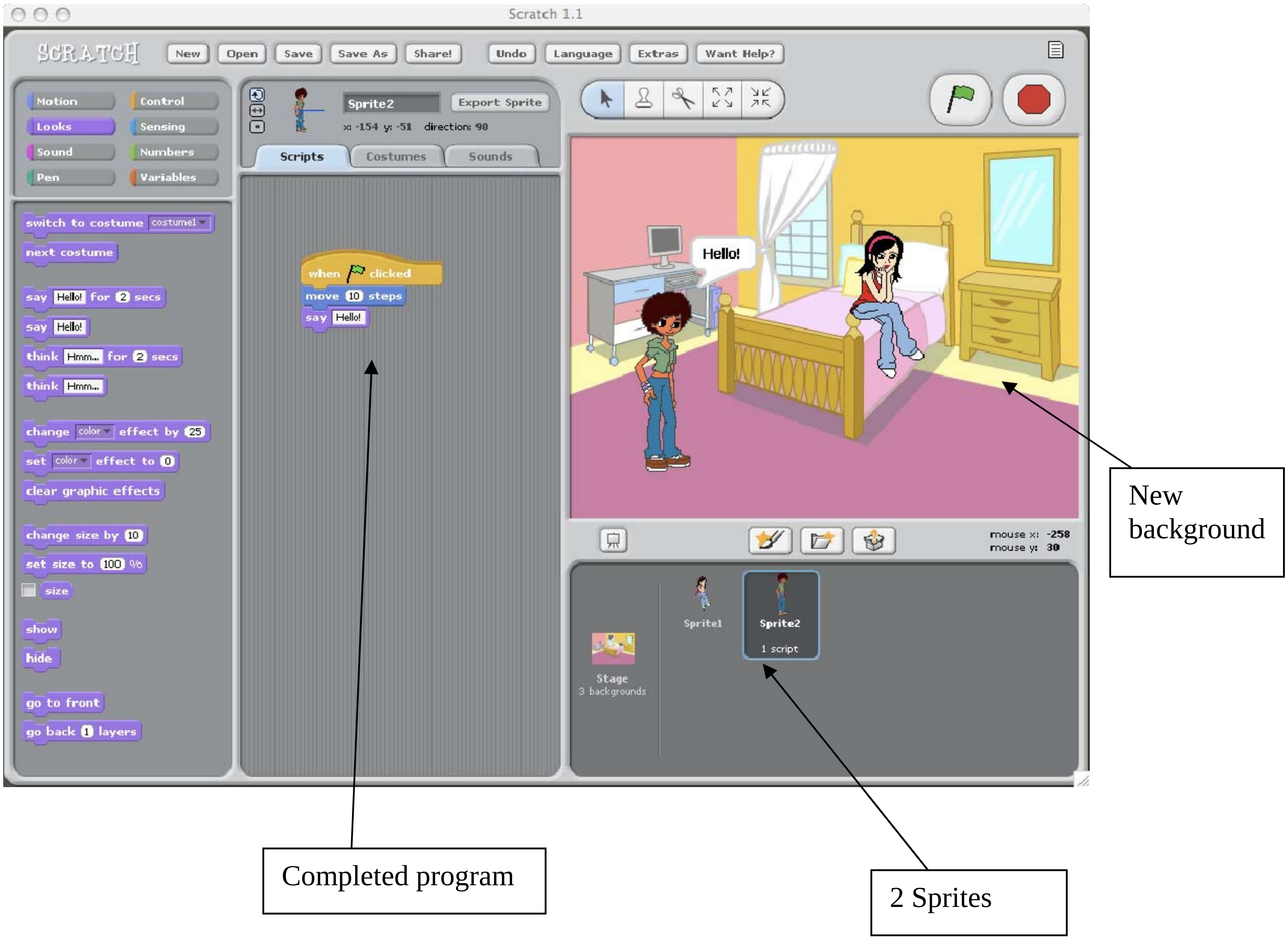
Image

Before: Initial Screen



After:

Program with two sprites, control block, motion block, looks block



```
when clicked
move 10 steps
say Hello!
```

New background

Completed program

2 Sprites

Safety Issues

- None

Troubleshooting Tips

If the programs do not work allow the students to delete the blocks:

- Select the scissors over the stage and select the desired block

Or

- Drag the block off the pad on to the left area

Investigating Questions

How does changing the code blocks affect the sprite? For example, if the student changes the number of steps in a move block, the sprite should move more or less depending upon the change. If the student includes a “say Hello” block the text should appear near the sprite.
How can you create dialog between multiple sprites? There are at least two ways to do this: 1) include dialog in a sequential manner in the code blocks or 2) have one sprite activated by the ‘green flag’ and another activated by the space bar (for example). Include ‘say’ blocks in the code for each sprite and execute each independently.

Assessments

Pre-Lesson Assessment
Ask students if they have experience writing computer programs. Discuss prior experience with students.

Activity Embedded Assessment
Monitor students periodically to be sure they each are following along.

Lesson Summary Assessment
Students should not be expected to become proficient in Scratch at the completion of this lesson. However, they should have familiarity and be able to locate sprites, change backgrounds, and write simple programs. Although there are others, below are 2 tasks that you can ask students to complete to demonstrate their ability to navigate in Scratch:

- Ask students to have the sprite move fewer or more steps and see if they are able to.
- Ask students to add another sprite to the stage.

Activity Extensions

Allow students to create stories using multiple sprites and backgrounds.
Introduce additional block categories – Pen and Sound to other projects.
Ask students to create stories related to other subjects they are studying.

Activity Scaling

- The activity can be scaled up or down according to the age of the students. The programs can contain degrees of complexity.

References

Scratch – www.scratch.mit.edu

Owner

Drexel University GK-12 Program

Contributors

Quincy Brown, Bill Mongen, Dara Kusic, Elaine Gabarine

Copyright

Version: Mar 2007