PROGRAMMING CONCEPTS AND SKILLS SUPPORTED IN SCRATCH

In the process of creating interactive stories, games, and animations with Scratch, young people can learn important computational skills and concepts.

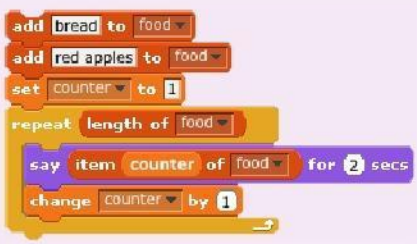
PROBLEM-SOLVINGANDPROJECT-DESIGNSKILLS

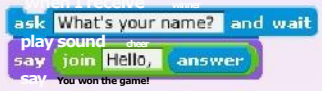
- logical reasoning
- debugging problems
- developing ideas from initial conception to completed project
- sustained focus and perseverance

FUNDAMENTALIDEASABOUTCOMPUTERSANDPROGRAMMING

- Computer programs tell the computer precisely what to do, step-by-step
- Writing computer programs doesn't require special expertise, just clear and careful thinking

SPECIFICPROGRAMMINGCONCEPTS

Concept	Explanation	Example
sequence	To create a program in Scratch, you need to think systematically about the order of steps.	
iteration (looping)	**forever** and **repeat** can be used for iteration (repeating a series of instructions)	
conditional statements	**if** and **if-else** check for a condition.	
variables	The **variable** blocks allow you to create variables and use them in a program. The variables can store numbers or strings. Scratch supports both global and object-specific variables.	
lists (arrays)	The **list** blocks allow for storing and accessing a list of numbers and strings. This kind of data structure can be considered a "dynamic array."	

http://scratch.mit.edu ncept	Explanation	E
event handling	**when key pressed** and **when sprite clicked** are examples of event handling - responding to events triggered by the user or another part of the program.	
threads (parallel execution)	Launching two stacks at the same time creates two independent threads that execute in parallel.	
coordination and synchronization	**broadcast** and **when I receive** can coordinate the actions of multiple sprites. Using **broadcast and wait** allows synchronization.	
		
keyboard input	**ask and wait** prompts users to type. **answer** stores the keyboard input.	
random numbers	**pick random** selects random integers within a given range.	
boolean logic	**and, or, not** are examples of boolean logic.	
dynamic interaction	**mouse_x, mouse_y,** and **loudness** can be used as dynamic input for real-time interaction	
user interface design	You can design interactive user interfaces in Scratch - for example, using clickable sprites to create buttons.	

Creating Your Own Video Games from SCRATCH

Free Download for Mac and PC (http://scratch.mit.edu/download)

KSDE Partners in Excellence Objectives:

(1) Familiarize yourself with the buttons in the upper left hand corner

(2) Learn the difference between a sprite, background, costumes, and scripts

(3) Go to learn scratch.org and "mod," or customize 1 game; then share your game using the tab at the top

a. Etch-a Sketch….Go to learnscratch.org, click on Scratch 3, Unit 3, Lesson 11, go to the bottom and click "download the project"

b. Once you have "modded" the game click on "share

(4) Build a simple game from "SCRATCH"

(5) Explore the other teacher resources at the Teacher Resource Page (http://www.scratch-ed.org/) and brainstorm for ideas about this software could be used in your school/class.

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See http://scratch.mit.edu

Presented by: Curtis Chandler (USD 320) chandlerc@usd320.com

1. Get your questions together in this format…

   Question*answer

   a. For example..

      What is 5 x 5 * 25

      What is 4 x 5 * 20

      What is 3 x 5 * 15

      What is 2 x 5 * 10

      What is 1 x 5 * 5

2. Go to **“classtools.net”**
3. On the right hand side of the site, click on **“arcade game generator”**
4. Copy and paste your questions and answers list into the box
5. Click on **“play”**
6. Watch the kids blow stuff up, etc.