About this Guide

This facilitator’s guide is designed to be a tool that further explains activities introduced in *Discovering Computer Science & Programming through Scratch: Level One*. This guide suggests many activities which are “unplugged”: off-computer activities designed to help learners understand computer science concepts and augment the on-computer programming experience, much in the spirit of *Computer Science Unplugged* (http://csunplugged.org/). These unplugged activities are to be used as an introduction before students begin a coding activity. This guide also includes discussion questions, skills introduced, blocks introduced, and hints or suggestions to help youth better understand each activity and concept. Youth and facilitators are encouraged to explore, learn, and talk about Scratch together.

Acknowledgments:

Authors:
Lenny Pitt, Professor and Director of Undergraduate Programs, Computer Science, University of Illinois
Judy Rocke, Curriculum Development Specialist, Office for Mathematics, Science, and Technology Education (MSTE), University of Illinois
Jana Sebestik, Assistant Director STEM Curriculum Design, (MSTE), University of Illinois

This guide was created as part of the “4-H Computing Connections” grant funded by the University of Illinois Extension and Outreach Initiative.
See [http://web.extension.illinois.edu/initiative/](http://web.extension.illinois.edu/initiative/)

Principal Investigator: Lenny Pitt, University of Illinois

Co-Principal Investigators:
Maya Israel, Assistant Professor, College of Education, University of Illinois
George Reese, Director of the Office for Mathematics, Science, and Technology Education, University of Illinois

Key Extension/Outreach Contact:
Robert Smith, 4-H Statewide Robotics Educator

CS4H Team:
Lenny Pitt, Maya Israel, George Reese, Jana Sebestik, Judy Rocke, Travis Faust, Saadeddine Shehab, Robert Smith, Joseph Muskin

Layout and Design:
Khishigdelger Ganbold, University of Illinois, College of Media, Department of Advertising
Christina Tran, University of Illinois, College of LAS, Department of Mathematics, MSTE

Scratch is a project of the Lifelong Kindergarten Group at the MIT Media Lab (http://scratch.mit.edu). Images of the Scratch cat are used with permission. All other screenshots and images used in this guide are licensed under the Creative Commons Attribution-ShareAlike license.

Copyright © 2016 The Board of Trustees of the University of Illinois. All rights reserved.
For ordering information contact National 4-H Council supply Services at: (301) 961-2934
# Table of Contents

## Getting Started with Scratch

<table>
<thead>
<tr>
<th>Activity</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence</td>
<td>5</td>
</tr>
<tr>
<td>Unplugged Activity</td>
<td>5</td>
</tr>
<tr>
<td>Calendar Activity</td>
<td>5</td>
</tr>
</tbody>
</table>

## Iteration

<table>
<thead>
<tr>
<th>Activity</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unplugged Activity</td>
<td>7</td>
</tr>
<tr>
<td>Box It In</td>
<td>7</td>
</tr>
<tr>
<td>Calendar Cross Out</td>
<td>8</td>
</tr>
<tr>
<td>Bugs, Glitches, and Problems in Scripts</td>
<td>8</td>
</tr>
<tr>
<td>Ballerina</td>
<td>9</td>
</tr>
<tr>
<td>Cat Aerobics</td>
<td>10</td>
</tr>
</tbody>
</table>

## Conditionals

<table>
<thead>
<tr>
<th>Activity</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unplugged Activity 1</td>
<td>11</td>
</tr>
<tr>
<td>The Beetle</td>
<td>14</td>
</tr>
<tr>
<td>Unplugged Activity 2</td>
<td>15</td>
</tr>
<tr>
<td>Birthday Cake Game</td>
<td>16</td>
</tr>
<tr>
<td>Guess the Date</td>
<td>18</td>
</tr>
</tbody>
</table>

## Variables

<table>
<thead>
<tr>
<th>Activity</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unplugged Activity</td>
<td>20</td>
</tr>
<tr>
<td>How Much Money?</td>
<td>21</td>
</tr>
<tr>
<td>Regular Polygons</td>
<td>23</td>
</tr>
<tr>
<td>Create Any Regular Polygon Using Variables</td>
<td>23</td>
</tr>
<tr>
<td>Spiral Polygons</td>
<td>24</td>
</tr>
</tbody>
</table>

## Modularization

<table>
<thead>
<tr>
<th>Activity</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suggested Unplugged Activity for Modularization</td>
<td>26</td>
</tr>
<tr>
<td>Blocks and More Blocks</td>
<td>26</td>
</tr>
<tr>
<td>Spider Web</td>
<td>28</td>
</tr>
<tr>
<td>Modularization - Talk About</td>
<td>28</td>
</tr>
</tbody>
</table>

## Balloon Pop

<table>
<thead>
<tr>
<th>Activity</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balloon Pop</td>
<td>29</td>
</tr>
</tbody>
</table>

## Show What You Know

<table>
<thead>
<tr>
<th>Activity</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extensions, Connections, and Resource Guide</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>32</td>
</tr>
</tbody>
</table>
Your Role as Facilitator

This computer science curriculum introduces sequence, iteration, conditionals, variables, and modularization – the most fundamental principles in any programming language. It uses Scratch, an MIT Media Lab project, to explore these concepts while creating a foundation for coding in any language.

This curriculum provides youth with a series of tutorials and challenges within the Scratch environment. Young people can work on the activities individually, with partners, or in a guided instructional setting. This facilitator guide further explains the activities introduced in the youth guide. It suggests many “unplugged” activities. It also includes discussion questions, previews of skills and blocks introduced, and hints or suggestions to help youth better understand each activity and concept.

As a facilitator of this project, encourage youth to talk about what they learn as they try new scripts and find new blocks. Youth will learn faster and more, when they discuss their projects with others. The Scratch community encourages users to share their projects on the Scratch website and to remix other’s projects. Just be sure to give credit to the original project creator. There are nearly 11,000,000 registered Scratch users sharing projects. Join the fun!

This curriculum was written for youth in Grades 5-12, but may be used and adapted for younger and older audiences, based on experience.

Scratch is a project of the Lifelong Kindergarten Group at the MIT Media Lab. Each youth in a group should have his/her own guidebook.
Experiential Learning

The Experiential Learning Model of Instruction provides learners an opportunity to become familiar with the content (Experience), explore a deeper meaning of the content (Share and Process), connect the learning to other examples or opportunities (Generalize), and apply it in real world situations.

The facilitator will guide youth through this process by helping them to focus on the activities, provide support and feedback for the learning, and debrief with them about their learning experience: what went well, what they could have done differently, what they could do next. This debriefing process fits hand-in-glove with the engineering design process used throughout the curriculum.

Youth do before being told or shown how.
Youth share how they will use the project and life skill practiced in other parts of their lives.
Youth describe the experience and their reaction.
Youth relate the project and life skill practiced to their own everyday experiences.
Youth discuss what was most important about what they did.

Scratch is a project of the Lifelong Kindergarten Group at the MIT Media Lab.

People need time to explore Scratch! They will discover many things on their own. Encourage them to share and problem solve together. Encourage them to listen to each other and make suggestions as they share the successes and problems they encounter as they complete the activities.

Discovering Computer Science & Programming through Scratch: Level One explores these fundamental computer science concepts: Sequence, Iteration, Conditionals, Variables, and Modularization. Suggested unplugged, or off computer, activities are included for each concept to help develop a deeper understanding of the concept before beginning the computer activities.

The facilitator notes include:

- Unplugged activities
- Skill lists
- Lists of the blocks introduced in each activity
- Suggestions to help with group discussions
- Hints for helping youth understand a concept

When you put your project on the Scratch website it is private unless you click on the share button. If you share your project, other Scratchers can see it. The Scratch Team encourages Scratchers to remix projects within guidelines that respect the work of the creator. Note the following FAQ’s from https://scratch.mit.edu/info/faq/#remix

What is a remix?
When a Scratcher makes a copy of someone else’s project and modifies it to add their own ideas (for example, by changing scripts or costumes), the resulting project is called a “remix.” Every project shared to the Scratch website can be remixed. We consider even a minor change to be a valid remix, as long as credit is given to the original project creator and others who made significant contributions to the remix.

What if I don’t want others to remix my projects?
By publishing your project on the Scratch website, you agree to license it under a Creative Commons Share Alike license. If you don’t want others to view and remix your creations, don’t share them on the Scratch website.

Download Scratch at: https://scratch.mit.edu/
A computer does exactly what you tell it to do, and it follows the steps in the code in exactly the order that you give it. Order matters.

Unplugged Activity

Three people come to the front of the room. Person 1 gives instructions and the other persons follow the instructions.

- Person 1 tells Person 2 to go forward 5 steps, turn 90 degrees, go forward 3 steps.
- Next, Person 1 reorders the instructions and tells Person 3 to go forward 3 steps, turn 90 degrees, go forward 5 steps.

- Where do Persons 2 and 3 stop?
- Talk about the results. Emphasize that even though the instruction components are the same, arranging them in different sequences results in different outcomes.
- What is the result when these same instructions are executed in this order? go forward 3 steps, turn 90 degrees, go forward 5 steps.
- When Scratch blocks are arranged in an incorrect sequence the result may be incorrect or unexpected. Sequence matters in coding.

Sequence and Scratch

Calendar Activity 1

A calendar is used as a backdrop for these activities.

Open the project at https://scratch.mit.edu/projects/52455886/. Click See inside and then click Remix.

Skills

- Make a sprite travel to a specific location on the stage
- Draw the path a sprite makes as it moves around the stage
- Use Event blocks to start a script
- Create a reset script to return a sprite to its starting location
- Use (x,y) coordinates to position a sprite on the stage
Blocks introduced in this activity are in the **MOTION**, **PEN**, **EVENTS**, and **CONTROL** categories.

Challenge Activities offer opportunities to apply new skills. Experiment and try new blocks in the **Looks** category.

Notes
- One step is a very short distance on the stage. The stage is 480 steps across and 360 steps from top to bottom.
- When Scratch opens the cat appears in the center of the stage at point (0,0) on a (x,y) coordinate graph.
- Blocks connect to each other to create a script. A script is a series of instructions for a sprite.

Talk About
1. What are the advantages of having a reset script?
2. What are some of the things you’ve learned in these activities?
3. What is the difference between a block and a script?
4. What ideas would you share with someone who has never used Scratch before?
5. How would you explain sequence?
Iteration

A computer can not only do exactly what you tell it, it can also do the same task over and over again. It never gets tired, and it never makes a mistake. We use iterations to ask a computer to do something repeatedly.

Unplugged Activity

Each learner needs a pen and a blank sheet of paper. Ask each to follow these steps:
• Move your pen to the left middle of the paper
• Repeat six times:
  • Put your pen tip down on the paper
  • Draw a circle with approximate radius of one inch
  • Lift your pen up
  • Move your pen one inch to the right

You are using iteration to draw this series of six circles. In the same way you can use a repeat block to tell the computer how many times you want a specific task performed.

Iteration and Scratch

A calendar is used as a backdrop for some of these activities.
Open the project at https://scratch.mit.edu/projects/52455886/. Click See inside and then click Remix.

Box It In

Skills
• Use the repeat block and blocks from the Pen and Motion categories to draw a square at a specific location on the stage
• Place blocks inside a repeat block

Blocks introduced in this activity are in the Control and Pen categories.

Challenge Activities offer time to create. They encourage linking new scripts to previously coded scripts and applying skills in new situations.
Notes
Scratch uses three kinds of repeat blocks. The first block tells the computer to repeat a task a certain number of times. The second tells the computer to repeat the task forever. The third tells the computer to check the until condition and if it is not true, repeat the entire task inside the block. Then check the until condition again and if still not true repeat the entire task inside the block again. The computer continues to check and repeat until the condition is true.

Calendar Cross Out
Skills
• Combine blocks in a correct sequence to recreate a picture
• Use iteration to accomplish a task
• Use a negative number in a move block to make a sprite move left across the stage

Notes
• Use the repeat block whenever the same series of blocks are used over and over, Think about how a repeat block can eliminate this repetition.

Bugs, Glitches, and Problems in Scripts
Skills
• Recreate a script
• Debug or fix the script so that it creates a specified picture
• Right click a script to make a duplicate copy of the script
• Pull a script apart one block at a time to test the script step by step while searching for errors

Notes
• Clicking and holding the script allows users to detach all the blocks from the cursor location and below.
• Right clicking a script allows users to duplicate the script.
• One possible “debugged” script, for this activity, is shown on the right.