Interactive Learning Objects
Toolkit for Teachers & Learners
Copyright

The Commonwealth of Learning (COL) is an intergovernmental organisation created by Commonwealth Heads of Government to promote the development and sharing of open learning and distance education knowledge, resources and technologies.

© 2017 Commonwealth of Learning

Except where otherwise noted, Interactive Learning Objects Toolkit for Teachers & Learners is made available under a Creative Commons Attribution 4.0 International (CC BY 4.0) License: https://creativecommons.org/licenses/by/4.0. By using the content of this publication, the users accept to be bound by the terms of use of the Commonwealth of Learning's Open Access repository (http://oasis.col.org).

Disclaimer: The ideas and opinions expressed in this publication are those of the author; they are not necessarily those of Commonwealth of Learning and do not commit the organisation.

Caveat: This course material was published to support the learning of teachers. The Commonwealth of Learning does not grant any degree, certification or credits based solely on your completion of this course material.

Acknowledgement: The development of this course material was supported by the Commonwealth of Learning, Burnaby, Canada as part of Grant #2015-2585, generously made by The William and Flora Hewlett Foundation, USA.

Author: John Taiwo Okewole (dbeloved@gmail.com)
Editor: Ishan Abeywardena, PhD - Adviser: Open Educational Resources, Commonwealth of Learning (abeywardena@col.org)
Copy Editor: Dania Sheldon, DPhil
Administrative Support: Patricia Schlicht, Commonwealth of Learning
Publications Coordination: Sparrow McGowan, Commonwealth of Learning
Cover Design: Ania Grygorczuk

Interactive Learning Objects Toolkit for Teachers & Learners
Table of Contents

Copyright...................................................................................................................................................... i
Table of Contents....................................................................................................................................... ii
Table of Figures......................................................................................................................................... iv
About the Toolkit....................................................................................................................................... 1
   How to Use the Toolkit ........................................................................................................................ 2
   Using the Toolkit: Learners .................................................................................................................. 2
   Using the Toolkit: Teacher ................................................................................................................... 2
1.0 MIT’s Scratch Platform ................................................................................................................ 3
   1.1 Introduction to MIT’s Scratch ................................................................................................. 3
   1.2 Getting Started ........................................................................................................................... 4
   1.3 The Scratch Development Environment ............................................................................... 5
   1.4 Developing and Debugging in Scratch ................................................................................. 26
   1.5 Additional Tools ...................................................................................................................... 31
2.0 Hello Scratch ................................................................................................................................ 32
   2.1 Explore ...................................................................................................................................... 33
   2.2 Animation ................................................................................................................................. 42
   2.3 Stories ........................................................................................................................................ 48
   2.4 Game ......................................................................................................................................... 60
   2.5 Extensions and Simulations ................................................................................................... 71
3.0 Scratch Tutorials .......................................................................................................................... 79
   3.1 Book Information Session ..................................................................................................... 79
   3.2 Hair Salon .............................................................................................................................. 104
   3.3 Building Project ....................................................................................................................... 123
   3.4 Animated Music Videos (AMV) .............................................................................................. 127
   3.5 Electrical Circuit .................................................................................................................... 133
   3.6 Flashing Light with Movement Using a Raspberry Pi .................................................... 144
4.0 Packaging and Distributing Scratch ....................................................................................... 147
   4.1 Share a Project ....................................................................................................................... 147
   4.2 Scratch Studio ....................................................................................................................... 153
   4.3 Exporting and Importing Scratch Files .............................................................................. 159
   4.4 Supported Files ..................................................................................................................... 161
5.0 The Scratch Community .......................................................................................................... 162
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>About the Community</td>
<td>162</td>
</tr>
<tr>
<td>5.2</td>
<td>Community Concept</td>
<td>165</td>
</tr>
<tr>
<td>5.3</td>
<td>Scratch User Ranks/Accounts</td>
<td>170</td>
</tr>
<tr>
<td>6.0</td>
<td>Workshop for Scratch</td>
<td>196</td>
</tr>
<tr>
<td>6.1</td>
<td>Workshop Training Schedule</td>
<td>196</td>
</tr>
<tr>
<td>6.2</td>
<td>Workshop Evaluation</td>
<td>198</td>
</tr>
<tr>
<td>7.0</td>
<td>Useful Scratch Resources</td>
<td>202</td>
</tr>
<tr>
<td>7.1</td>
<td>Scratch Website</td>
<td>202</td>
</tr>
<tr>
<td>7.2</td>
<td>Scratch-Related Sites</td>
<td>202</td>
</tr>
<tr>
<td>7.3</td>
<td>Other Useful Sites</td>
<td>202</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1.1</td>
<td>A screenshot of the Scratch website home page</td>
<td>4</td>
</tr>
<tr>
<td>1.2</td>
<td>The Scratch program development environment (editor)</td>
<td>6</td>
</tr>
<tr>
<td>1.2a</td>
<td>The Scripts area</td>
<td>7</td>
</tr>
<tr>
<td>1.2b</td>
<td>The Blocks palette</td>
<td>8</td>
</tr>
<tr>
<td>1.2c</td>
<td>The Stage (display area)</td>
<td>9</td>
</tr>
<tr>
<td>1.2d</td>
<td>The Sprite list area</td>
<td>9</td>
</tr>
<tr>
<td>1.2e</td>
<td>The backdrop modification area</td>
<td>10</td>
</tr>
<tr>
<td>1.2f</td>
<td>The Backpack</td>
<td>10</td>
</tr>
<tr>
<td>1.3</td>
<td>The menu bar</td>
<td>10</td>
</tr>
<tr>
<td>1.4</td>
<td>A series of connected blocks</td>
<td>11</td>
</tr>
<tr>
<td>1.5</td>
<td>New sprite bar</td>
<td>16</td>
</tr>
<tr>
<td>1.6</td>
<td>Creating a new sprite</td>
<td>17</td>
</tr>
<tr>
<td>1.7</td>
<td>A new sprite is placed in a default position on the Stage</td>
<td>17</td>
</tr>
<tr>
<td>1.8</td>
<td>A new sprite’s position details and settings</td>
<td>18</td>
</tr>
<tr>
<td>1.9</td>
<td>Changing a sprite look using different costumes</td>
<td>19</td>
</tr>
<tr>
<td>1.10</td>
<td>The sound file of a sprite</td>
<td>20</td>
</tr>
<tr>
<td>1.11</td>
<td>Saving a sprite to the local disk</td>
<td>20</td>
</tr>
<tr>
<td>1.12</td>
<td>Small Stage layout</td>
<td>21</td>
</tr>
<tr>
<td>1.13</td>
<td>Regular Stage layout</td>
<td>21</td>
</tr>
<tr>
<td>1.14</td>
<td>Full-screen Stage layout</td>
<td>22</td>
</tr>
<tr>
<td>1.15</td>
<td>The costumes of a sprite</td>
<td>23</td>
</tr>
<tr>
<td>1.16</td>
<td>The two edit modes of a costume</td>
<td>24</td>
</tr>
<tr>
<td>1.17</td>
<td>The activity tabs (tools) of the vector mode</td>
<td>24</td>
</tr>
<tr>
<td>1.18</td>
<td>The activity tabs (tools) of the bitmap mode</td>
<td>25</td>
</tr>
<tr>
<td>1.19</td>
<td>Backdrop activity area</td>
<td>26</td>
</tr>
<tr>
<td>2.1</td>
<td>Click on the Create button to start the editor</td>
<td>32</td>
</tr>
<tr>
<td>2.2</td>
<td>Click on the Create button to start the editor</td>
<td>34</td>
</tr>
<tr>
<td>2.3</td>
<td>Name and save your project by clicking the File menu</td>
<td>35</td>
</tr>
<tr>
<td>2.4</td>
<td>The Blocks palette with Scripts and Events block active</td>
<td>36</td>
</tr>
<tr>
<td>2.5</td>
<td>The Blocks palette and the Scripts area, showing the activity carried out</td>
<td>37</td>
</tr>
<tr>
<td>2.6</td>
<td>The Scripts, showing (a) inactive blocks and (b) active blocks</td>
<td>37</td>
</tr>
<tr>
<td>2.7</td>
<td>The Stage, showing the action performed when the Script is live tested</td>
<td>38</td>
</tr>
<tr>
<td>2.8</td>
<td>The Blocks palette with the active costume menu</td>
<td>38</td>
</tr>
<tr>
<td>2.9</td>
<td>Selecting a new backdrop</td>
<td>39</td>
</tr>
<tr>
<td>2.10</td>
<td>The Backdrop Library</td>
<td>39</td>
</tr>
<tr>
<td>2.11</td>
<td>The xy-grid backdrop</td>
<td>40</td>
</tr>
<tr>
<td>2.12</td>
<td>Naming a backdrop</td>
<td>40</td>
</tr>
<tr>
<td>2.13</td>
<td>The block of code to say Hello Scratch! with two backdrops and costumes</td>
<td>41</td>
</tr>
<tr>
<td>2.14</td>
<td>The block of code to introduce sound for each backdrop</td>
<td>41</td>
</tr>
<tr>
<td>2.15</td>
<td>The various sprite costumes</td>
<td>45</td>
</tr>
<tr>
<td>2.16</td>
<td>The use of a broadcast block in place of an execution block</td>
<td>46</td>
</tr>
<tr>
<td>2.17</td>
<td>Editing sound in the Sound menu on the Blocks palette</td>
<td>46</td>
</tr>
<tr>
<td>2.18</td>
<td>The loop event block</td>
<td>47</td>
</tr>
<tr>
<td>2.19</td>
<td>Selecting and naming the backdrop</td>
<td>56</td>
</tr>
</tbody>
</table>
Interactive Learning Objects Toolkit for Teachers & Learners
Figure 5.40 Adding a new student to the Class................................................................. 188
Figure 5.41 The Students menu automatically shows the list of students added to a Class ..... 188
Figure 5.42 Upload CSV to add multiple students to a Class ............................................. 189
Figure 5.43 Error message when adding by uploading using CSV ...................................... 189
Figure 5.44 The Sign-up Link option to add students to a class ......................................... 190
Figure 5.45 A sample of what you will see when you click Generate.................................... 190
Figure 5.46 The Studios menu in a Class ............................................................................ 190
Figure 5.47 Add a studio to a Class .................................................................................. 191
Figure 5.48 The Activity menu on the Class page .............................................................. 191
Figure 5.49 A single view page concurrently showing the four menu in a class .................. 192
Figure 5.50 The page view of a student in a class............................................................. 193
Figure 5.51 The ScratchEd online community home page ................................................ 194
Figure 5.52 The ScratchJr website ..................................................................................... 195
About the Toolkit

Massachusetts Institute of Technology’s (MIT) Scratch is in its tenth year in 2017, and the project has grown tremendously all over the world. It has over 22 million projects shared and has over 19 million accounts from across the globe. All of these are a testament to the central focus of the program in delivering on the essential targets of the digital economy. In the 21st century, communication, collaboration, critical thinking and creativity have been identified as the cornerstones of success for the average learner. These are the expected/demanded skill sets in every field of endeavour in this century. A discipline that makes this possible is computing, which has permeated virtually every other field as a basis for and an enabler in learning, research, development, application, and beyond. It has also engendered learning support for all forms and levels of education — general and technical education; formal and non-formal learning; K-12 and Higher Education. Culturally, computer literacy is grouped with basic literacy and numeracy as essential skills.

It is therefore not surprising that a recent World Economic Forum 20171 statistic revealed software to be the largest venture capital investment sector in the United States by a huge margin. Software programming is at the core of computing, and it has delivered on the long-standing promise to shape different entry points for computing. Programming skills are needed at each stage of application in computing. In the 21st century, the stereotype that programming is only for “geeks” is rapidly giving way to a progressive approach that wants to make programming accessible for everyone.

Invariably, visual programming meets the need of a wide audience and is gaining ground in becoming the threshold at which people embrace programming and computer learning. Since the start of “what you see is what you get” (WYSIWYG) in web development, we are seeing the same trend: a fluent approach to core programming through visual programming. Learners’ baseline for programming has been effectively lowered, and even young people can now learn how to program. MIT Scratch, a free programming language and an online community, stands poised to introduce non-geeks to programming. With a target age of about eight to sixteen, it has established that programming can be effectively learned by anyone, dispensing with the idea that a subject-matter expert (SME) is necessarily required.

However, despite the remarkable opportunity that Scratch offers, millions across the world are still being left out. This is the central problem being addressed and solved by this toolkit — how to initiate the beginner into programming through Scratch. This toolkit is designed to empower the people who are meant to lead this change: teachers. Since most teachers were not trained in this system and can’t give what they don’t have, they need an initiation into this new framework of learning. As noted above, computing is no longer optional; it has become a core component in practically every field, including teaching. This toolkit has therefore been developed using a vocabulary and learning approach with no prerequisites, so you can learn programming and ultimately teach programming after you conclude the tutorials it contains.

1 https://www.weforum.org/agenda/2017/02/these-are-the-industries-attracting-the-most-venture-capital?utm_content=buffer30fd2&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer
How to Use the Toolkit

This toolkit introduces Scratch to an audience with little or no prior understanding in software programming. Scratch by design accepts every level of proficiency into the programming field, and this toolkit enables the learner to discover their point of entry. The toolkit is broken down into sections and quickly puts the user into action with the Hello Scratch section. There is a walkthrough portion for all activities in this section, and progress is simpler if you follow the activities in a step-wise manner. Next is the tutorials section, which contains activities to try out with complete programs. There is no walkthrough for the tutorials. However, if you have completed all the activities prior to the tutorials, you should have minimal difficulty in developing the programs. Once you have completed all the sections in this toolkit, you will be able to develop complete programs with the Scratch language.

Using the Toolkit: Learners

- Step 1: Follow the instructions given in Section 1 to set up your account on the Scratch website or download the offline version on your PC.
- Step 2: Get acquainted with the Scratch IDE (integrated development environment) used in development.
- Step 3: Do all the activities in Section 2. Ensure that you build your apps sequentially from Activity 1 to Activity 4. Each activity is designed to build on the previous activities, so do not skip any. Only Activity 5 requires an external resource.
- Step 4: Work through all six tutorials in Section 3. They are based on the knowledge you gained from doing the activities. Tutorials 1 to 6 are designed to take you from simple applications to intermediate and complicated application development.
- Step 5: Section 4 presents the various ways to package and distribute your projects.
- Step 6: Scratch enables the creation of a community and collaborative learning, and Section 5 details how you can bring your class alive with Scratch.
- Learning and mastering the information in this toolkit will empower you to teach Scratch programming.

Using the Toolkit: Teacher

- Sections 1 to 5 are designed to be printed and distributed as individual handouts.
- All activities and tutorials are designed to be independent of each other. They can be printed and distributed to your students during the workshop.
- A workshop training schedule is provided in Section 6.1. All the workshop sessions are included in this toolkit. Each handout can be used as reference material when you conduct your workshop.
- A workshop evaluation form is included in Section 6.2. This can be used to gather feedback on your workshop for further improvement.
- The workshop must be conducted in a computer lab. All sessions are fully hands-on.
1.0 MIT’s Scratch Platform

1.1 Introduction to MIT’s Scratch

The Massachusetts Institute of Technology (MIT) Media Lab and Google began a collaboration to develop a new generation of graphical programming blocks, called Scratch Blocks. Scratch Blocks builds on Google’s Blockly technology and the Scratch team’s expertise in designing creative interfaces for young learners. “Scratch” is a programming language and online community where you can create your own interactive stories, games, and animations — and share your creations with others around the world. In the process of designing and programming Scratch projects, young people learn to think creatively, reason systematically and work collaboratively.

Scratch is a project of the Lifelong Kindergarten group at the MIT Media Lab. It is available for free at http://scratch.mit.edu. Scratch introduces basic computational concepts such as iteration, conditionals, variables, data types, events and processes. Through learning Scratch, students develop many critical 21st-century learning skills: thinking creatively, communicating clearly, analysing systematically, collaborating effectively, designing iteratively and learning continuously.

Scratch was developed especially for young people eight to sixteen years old, so it is used most often in elementary schools and middle schools, but people of all ages create and share Scratch projects. Scratch is even used in some introductory computer-science courses in colleges. Younger children may want to try ScratchJr, http://www.scratchjr.org, a simplified version of Scratch designed for ages five to seven.

The Scratch team has a guiding philosophy in their Learning and Designing Principles,2 which form the fundamentals of the Scratch programming design and development.

The Learning Principles are:

- Projects
- Passion
- Peers
- Play

The Design Principles are:

- Low Floor and Wide Walls
- Make it as Simple as Possible — And Maybe Even Simpler
- Many Paths, Many Styles
- Design for Tinkerability

The Scratch program is provided free of charge and runs on a Creative Commons licence. There are several modes for getting introduced to Scratch and starting to use it. With or without an account, a user can start to program on the website. However, to save work in the cloud, a user account needs to be created. The three categories of user accounts are: general user, teacher and developer.

2 The Learning and Designing Principles are available here https://scratch.mit.edu/developers.
Program development can be initiated in the cloud through the Scratch website or by installing Scratch locally on a PC. Since Scratch was proposed and released, there have been two known stable versions. The earlier version is Scratch 1.4, which was taken down from the website in May 2014. There is also the current version, known as Scratch 2.0. At present, all development taking place on the Scratch website is on the Scratch 2.0 codebase. Projects created in version Scratch 1.4 runs on version Scratch 2.0, but projects created in Scratch 2.0 cannot be opened in Scratch 1.4.

1.2 Getting Started

The Scratch program was designed for absolute ease and flexibility. This makes using it either online or offline easy for the primary audience. To access and program with Scratch online, a user will do the following:

1. Make sure the system is connected to the Internet.
2. Go to https://scratch.mit.edu/ from your web browser address bar. Any updated browser will open the link. The page opened will be similar to Figure 1.1.

![A screenshot of the Scratch website home page](image)

Figure 1.1 A screenshot of the Scratch website home page

It is good practice to know a bit about the website before trying things out. The initial information can be found on About Scratch or at https://scratch.mit.edu/about/. To explore example projects on the website, see https://scratch.mit.edu/starter_projects/.
After learning this basic information and understanding the look and feel of the website, a new user can try out an example project on the editor at https://scratch.mit.edu/projects/editor/?tip_bar=getStarted. Unlike when you are using the information sites, trying out the Scratch program takes you to the Scratch Development Environment (more about this shortly). A new user account can be created by clicking the Join Scratch icon or going to the upper right-hand corner of the home page.

Apart from the online version, the Scratch editor, like any other program, can be installed offline by downloading the installer. Two stable versions of the Scratch editor can be downloaded from the Scratch website.

- Scratch 1.4 download is available for Mac, Windows and some Linux OS at https://scratch.mit.edu/scratch_1.4/.
- Scratch 2.0 can be downloaded from https://scratch.mit.edu/scratch2download/. It can run on Mac, Windows and some Linux OS.

Note that Scratch 2.0 is used throughout this toolkit.

**System Requirements**

<table>
<thead>
<tr>
<th>Table 1.1 System Requirements for Scratch 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Display</strong></td>
</tr>
<tr>
<td><strong>OS</strong></td>
</tr>
<tr>
<td><strong>Disk size</strong></td>
</tr>
<tr>
<td><strong>CPU and memory</strong></td>
</tr>
<tr>
<td><strong>Sound/Video</strong></td>
</tr>
<tr>
<td><strong>Project file size</strong></td>
</tr>
<tr>
<td><strong>Plug-ins</strong></td>
</tr>
<tr>
<td><strong>Update</strong></td>
</tr>
</tbody>
</table>

1.3 **The Scratch Development Environment**

The Scratch development environment or project editor (Figure 1.2) is the interface where all programming activities are carried out in Scratch. It has six sections, labelled A to F in Figure 1.2:

- A – Scripts area
- B – Blocks palette

3 Source: https://scratch.mit.edu/info/faq/
- C – Stage
- D – Sprite List Area
- E – Backdrop area
- F – Backpack

**NOTE** – At the time of writing, the backpack is not yet included in the offline editor. It also doesn’t show on some browsers.

Section A is the Scripts area of the editor (see Figure 1.2a). Here, program blocks are arranged (snapped together) in the sequence that the user wants. It also serves as the editing area for images and graphics. All activity that needs to be carried out on the elements of the program is done in Section A.
Section B (Figure 1.2b) is the Blocks palette, comprising Scripts, Costumes and Sounds. The Scripts consist of the blocks for program development, which are: Motion, Looks, Sound, Pen, Data, Events, Control, Sensing, Operators and More Blocks. Costume is the panel for character graphic selection and modification. The Sounds panel is used for sound selection and modifications in the program editor.
IMPORTANT – “Script” means more than one thing in Scratch and is used interchangeably.

Section C (Figure 1.2c) is the Stage (display area) for viewing executed Scripts. The sprites and backdrop of a project are also shown in the Stage area. The execution (start and stop) buttons and the file name panel are on the Stage.
Section D is the Sprite List area (Figure 1.2d). All selected sprites in a project are shown in this section.

Section E is the backdrop modification area (Figure 1.2e). Backdrop is the term for the various background elements that can be used on the Scratch editor.
NOTE – Only one active backdrop is in use at a time, but several backdrops can be interchanged in a single project.

Section F is the Backpack that is used to store scripts and costumes for later use in a project.

It is also worth mentioning the top menu, shown in Figure 1.3.

Table 1.2 Description of the Scratch Editor Menu Bar Items

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scratch logo</td>
<td>When clicked, it redirects to the Scratch home page.</td>
</tr>
<tr>
<td>Globe Icon</td>
<td>This indicates languages that are available for program development in Scratch. Presently, over 40 languages can be used to program on Scratch.</td>
</tr>
<tr>
<td>File Tab</td>
<td>For starting new projects, uploading and downloading saved projects from the user’s computer and reverting to what has been previously done.</td>
</tr>
<tr>
<td>Edit Tab</td>
<td>Has the undelete, small stage layout and turbo mode sub-menu.</td>
</tr>
<tr>
<td>Tips tab</td>
<td>Opens a panel on the right of the editor that displays useful tips for programming on Scratch. Step-by-step information, various how-tos and details on blocks can be found on the tips panel.</td>
</tr>
</tbody>
</table>
About tab
Redirects the user to the About Scratch page (https://scratch.mit.edu/about/).

**Cursor Tools**

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicate</td>
<td>When selected, any block or sprite clicked will be duplicated in the Script area and the Stage.</td>
</tr>
<tr>
<td>Delete</td>
<td>When selected, any block or sprite clicked will be deleted in the Script area and the Stage.</td>
</tr>
<tr>
<td>Grow</td>
<td>When selected, any sprite clicked will increase in size in the Script area and the Stage.</td>
</tr>
<tr>
<td>Shrink</td>
<td>When selected, any sprite clicked will decrease in size in the Script area and the Stage.</td>
</tr>
<tr>
<td>Block help</td>
<td>When selected, it will show tips for a clicked block in the Script area and the Stage.</td>
</tr>
</tbody>
</table>

In addition, some tools will be used consistently during the lifetime of a project in Scratch. These tools are outlined below.

**Programming Blocks**

Blocks are puzzle-piece shapes that represent the codes in Scratch. The blocks are snapped to each other like a jigsaw puzzle, where each data type (event, command, reported value, reported Boolean or script end) has its own shape and a specially shaped slot for it to be inserted into; this prevents syntax errors. Series of connected blocks are called scripts.

![Figure 1.4 A series of connected blocks](image)

There are 12 categories of blocks: Motion, Looks, Sound, Pen, Variables, List, Event, Control, Sensing, Operators, PicoBoard and LEGO WeDo (although there are some other categories and blocks in some Scratch modifications). Only ten are shown in the Blocks palette.

Examples of the blocks categories are shown in Table 1.3.
<table>
<thead>
<tr>
<th>Block Category</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion</td>
<td><img src="image" alt="Motion Examples" /></td>
</tr>
<tr>
<td>Looks</td>
<td><img src="image" alt="Looks Examples" /></td>
</tr>
<tr>
<td>Sound</td>
<td><img src="image" alt="Sound Examples" /></td>
</tr>
<tr>
<td>Pen</td>
<td><img src="image" alt="Pen Examples" /></td>
</tr>
<tr>
<td>Variables (Data)</td>
<td><img src="image" alt="Variables Examples" /></td>
</tr>
<tr>
<td>List (Data)</td>
<td><code>add thing to t</code></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td><code>insert t contains thing ? at item 1 of t of t</code></td>
</tr>
<tr>
<td></td>
<td><code>show list t</code></td>
</tr>
<tr>
<td>Event</td>
<td><code>when clicked</code></td>
</tr>
<tr>
<td></td>
<td><code>when loudness &gt; 10</code></td>
</tr>
<tr>
<td></td>
<td><code>broadcast message1 and wait</code></td>
</tr>
<tr>
<td>Control</td>
<td><code>when I start as a clone</code></td>
</tr>
<tr>
<td></td>
<td><code>wait 1 secs</code></td>
</tr>
<tr>
<td></td>
<td><code>if then</code></td>
</tr>
<tr>
<td></td>
<td><code>else</code></td>
</tr>
<tr>
<td>Sensing</td>
<td><code>ask touching mouse-pointer ? and wait</code></td>
</tr>
<tr>
<td></td>
<td><code>set video transparency to x position of Sprite1 %</code></td>
</tr>
<tr>
<td>Operators</td>
<td><code>pick random + to =</code></td>
</tr>
<tr>
<td></td>
<td><code>join hello mod</code></td>
</tr>
</tbody>
</table>
There are six different block shapes: Hat, Stack, Boolean, Reporter, C and Cap. These are broken down into 11 hat blocks, five C blocks, 37 reporter blocks, 13 Boolean blocks, two cap blocks and 77 stack blocks.

a. Hat blocks start every script. They are shaped with a rounded top and a bump at the bottom; this is so you can only place blocks below them. There are 11 Hat blocks, six of which are in the Events category, one in the Control category, and four more.
b. Stack blocks perform the main commands. They are shaped with a notch at the top and a bump on the bottom; this is so blocks can be placed above and below them. There are 77 Stack blocks, as it's the most common block shape.
c. Boolean blocks are the conditions — so, they are either true or false. A condition is like asking your friend, “Does 2 + 2 = 4?” and they tell you either “yes” or “no.” With a hexagonal shape, there are 13 of these blocks.
d. Reporter blocks are the values. Reporter blocks can hold numbers and strings. This is like asking a friend, “What is 2 + 2?” and them answering “four.” These blocks are not just equations, though; they can report a variable. For example, if you ask your friend, “What is your age?” they may answer “15.” Shaped with rounded edges, there are 37 of these blocks — not counting the theoretically infinite number of Reporter blocks that can be made for each variable and list.
e. C blocks have the shape of Cs and are also known as “Wrap blocks.” They loop the blocks within the Cs or check whether a condition is true. There are five C blocks, and they can be found in the Control category. C blocks can be bumped at the bottom or capped.
f. Cap blocks end scripts. They are shaped with a notch at the top and a flat bottom; this is so you cannot place any blocks below them. There are two Cap blocks, which can both be found in the Control category.
<table>
<thead>
<tr>
<th>Block Shape</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hat</td>
<td><img src="image" alt="Hat Example" /></td>
</tr>
<tr>
<td>Stack</td>
<td><img src="image" alt="Stack Example" /></td>
</tr>
<tr>
<td>Boolean</td>
<td><img src="image" alt="Boolean Example" /></td>
</tr>
<tr>
<td>Reporter</td>
<td><img src="image" alt="Reporter Example" /></td>
</tr>
<tr>
<td>C</td>
<td><img src="image" alt="C Example" /></td>
</tr>
</tbody>
</table>
In the Scratch 2.0 editor online, shift-clicking the Edit menu makes an option called “Edit block colors” appear. Selecting this makes a menu appear with a few colour sliders and tools for modifying the block colours of a specific block category. Personalised colours can be saved to one’s computer, but loading the colours currently does not work properly.

**Sprites**

Sprites (Figure 1.2d) — which can be user-created, uploaded, or found in the sprites library — are the objects that perform actions in a project. Whilst the Stage can also be programmed in a project, most projects have at least one sprite as well because only sprites can move.

The bar above the Sprite List (Figure 1.5) has four buttons for creating sprites:

- The giga button allows you to choose a sprite from the library.
- The paintbrush button creates a blank sprite with an empty costume.
- The folder button allows you to upload a sprite from your computer.
- The camera button allows you to take a picture and uses that image as the costume.

![New sprite bar](image)

*Figure 1.5 New sprite bar*

When a sprite is created (Figure 1.6), the program will place that sprite at a random location on the Stage, usually around the centre, and open the sprite in the tab you’re currently viewing.

Each sprite in a Scratch project has an area for scripts, called the scripts area. Users can give instructions to a sprite (such as telling the sprite to move) by snapping blocks together in the scripts area. Clicking on the block(s) in the scripts area will cause the sprite to react based on the function of the block(s) clicked. Clicking on a sprite’s thumbnail in the sprite pane will bring up the script area of that sprite.
Figure 1.6 Creating a new sprite

Figure 1.7 A new sprite is placed in a default position on the Stage
The look of a sprite can also be changed by using costumes (Figure 1.9). The current costume of a sprite can be changed by clicking on the “Costumes” tab and clicking on the desired costume of choice, or by using blocks to select the sprite’s costume. New costumes for the sprite can be imported, created and edited in the Scratch Paint editor.
Some sprites also have at least one sound. Unlike costumes, sounds are an optional field, so you can have a sprite with no sounds. The sounds tab allows you to add, delete and edit sounds. Sounds can be played in the sound editor or with blocks that play a specific sound. Sprites (with all of their scripts, costumes and sounds) can be exported and then imported into another project if desired. This is achieved by right-clicking on a sprite’s thumbnail in the sprite pane and then selecting “Save to local file” in the pop-up menu. A sprite can also be dragged into the backpack and dragged out into another project for transporting. However, this will not save the sprite to your computer.
Stage

The Stage (Figure 1.2c) is the term for the background of the project, but similar to a sprite, it can have scripts, backdrops (costumes) and sounds. It has some restrictions regarding sprite functionality, such as motion and size blocks.

No sprites can move behind the Stage — it is always at the back layer.

The Stage can be three different sizes (Figures 1.12–1.14):
- Small – The stage is half the size, with a resolution of 240x180 pixels; this is useful for having more room in the scripts area.
- Regular – Normal mode; the Stage is 480x360 pixels.
- Full-screen mode – Fits the player to the computer’s current resolution. This is done by clicking the full screen icon in the upper left corner. When in non-full-screen mode, it looks like this; after clicking and in full screen, it looks like this.

![Small Stage layout](image1)

**Figure 1.12** Small Stage layout

![Regular Stage layout](image2)

**Figure 1.13** Regular Stage layout
Costume
A costume is one out of possibly many “frames” or alternate appearances of a sprite. A sprite’s
look can be changed to any of its costumes. They can be named, edited, created and deleted, but
every sprite must have at least one costume. One of the most common uses of costumes is to
make an animation for a game or other project.

What backdrops are to the Stage, costumes are to sprites. They can be used in the same way.
There are two different costume types: bitmap and vector. Bitmaps uses pixels to create an
image, whereas vectors are graphics stored and drawn with instructions rather than with grids of
pixels. There are ten tools you can use to edit a costume in bitmap. There are nine tools for the
vector editor, but only the reshape and the select tools are different from those for the costume
editor.
Figure 1.15 The costumes of a sprite
Figure 1.16 The two edit modes of a costume

Figure 1.17 The activity tabs (tools) of the vector mode
Backdrop
A backdrop (Figure 1.2e) is one out of many frames, or backgrounds, of the Stage. It is located in the backdrops library. The Stage can change its look to any of its backdrops. They can be named, edited, created and deleted, but the Stage must always have at least one backdrop, otherwise it will not have an image to resemble.

Backdrops are edited in the same way as costumes. They are found in the Backdrops tab on the Stage (Figure 1.19).
1.4 Developing and Debugging in Scratch

Scratch is a visual programming language. The implication of this is that you can start to develop a project, then run it and view the results simultaneously on the same editor. This attribute is known as **Live Testing**. The Scratch editor was constructed to make programming highly simplified.

Let’s review the steps involved in developing a project on Scratch.

### 1.4.1 Developing your program

**Step 1** – Open the editor. This can be done either online by clicking the **Create** menu or by starting your installed editor on the local computer. Depending on your operating system, you can simply navigate through your programs to start the Scratch editor.

**Step 2** – Choose a backdrop. There is always a default plain backdrop on Scratch. It is up to you to decide whether you will use that or change it. However, a project will always have a backdrop.

**Step 3** – Select a sprite. To write a script, at least one sprite must be active. That is when all the blocks can be used. A default Scratch mouse is always present on the editor.

**Step 4** – Write a script. This involves snapping two or more blocks together to form an executable script.

**Step 5** – Execute the script. After you have written your script, you run it to view what you are making. This is done by clicking the green flag on the Stage. The program you have written comes alive on the Stage.

---

**Table 1.5** Steps to Develop a Program
<table>
<thead>
<tr>
<th>Step</th>
<th>Screenshots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 – Open the editor</td>
<td><img src="https://scratch.mit.edu" alt="Scratch Editor" /></td>
</tr>
<tr>
<td>a. Online</td>
<td><img src="image1.png" alt="Online Editor" /></td>
</tr>
<tr>
<td>b. Offline</td>
<td><img src="image2.png" alt="Offline Editor" /></td>
</tr>
<tr>
<td>Step 2 – Choose a backdrop</td>
<td><img src="image3.png" alt="Select Backdrop" /></td>
</tr>
</tbody>
</table>
**Step 3 – Select a sprite**

![Sprite Selection](image)

**Step 4 – Write a script**

![Script Writing](image)

**Step 5 – Execute the script**

![Script Execution](image)

**TROUBLESHOOTING**

- If your editor does not start online, wait to see the prompt it is showing. The most likely problem is a plug-in issue. Confirm that your browser is up to date. Some browsers, such as Chrome, have the plug-in already installed. Others, such as Opera, will request that you install Adobe Flash from a prompt. Follow the instructions to update the browser or install the plug-in.
For installation on a local computer, the Scratch 2.0 installer will not run without Adobe Air. Its installation precedes Scratch. When it is set to automatically update, it may require you to run an update after you start Scratch.

Whilst running Scratch, activities may be slow on your computer. Try checking your Internet connection to see whether it is still on. You can stop other applications from running whilst you develop in Scratch.

### 1.4.2 Debugging your program

Developing a program is the first step in programming. Debugging is the next step. Sometimes, a program won’t execute what you have in mind or have scripted together into it. Debugging helps you find out why the program is not running and resolve the problem to achieve your project objective. In Scratch, the single visual interface makes it simpler to debug. To debug your project, you may consider the following steps.

- Inspect your scripts
- Check your values
- Test on the go

### Table 1.6 Script Debugging Steps with Examples

<table>
<thead>
<tr>
<th>Debugging step</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inspect your scripts</td>
<td><img src="image" alt="Example" /></td>
</tr>
</tbody>
</table>

A starting point is needed for the Sprite

This block is not responding to the script. The degree of turning was not indicated.
Common causes of error include:

a. Invalid operations – When the logic of a program is wrong, this will halt it from performing the required task. Most times, it is good practice to read the sequence on paper and see whether it integrates properly. A storyboard helps in resolving this kind of problem.

b. Wrong order of commands – This is the most common problem for Scratchers. Since blocks can be lumped together, doing this incorrectly can prevent the script from executing properly. You can try to resolve this by changing the order of the blocks and testing until the program delivers the right task.

c. Timing – The computer runs so quickly that some scripts may not be obvious when they perform their task. Solving this involves using different techniques to show what is been done each time.

d. Incompatibility – If a set of commands does not work together, a script will be highlighted in red. In this case, you can simply remove the problem block and replace it with something that represents what you intend.
NOTE

- The best way to prevent silly mistakes is to have neat, organized code. Ordering scripts by their functions or order of evaluation will help find and prevent mistakes.
- Leave comments wherever you had to spend time to get something to work right; this way, if the script breaks again, you can refer back to what worked before. Anyone with enough experience should be able to understand your script at first glance.
- Do not blindly copy scripts from other projects, unless that was the intention of the project. Other than moral copyright issues, chances are the script will not work in the context of your program.

1.5 Additional Tools

Increasingly, Scratch is being used for many educational activities, and several developments are underway to bring Scratch to teachers, learners and classrooms. The following are some important developments.

- Student and teacher tools: Various student and teacher tools have become prominent in the Scratch community. The companion site ScratchEd (http://scratched.gse.harvard.edu/) is the Harvard Graduate School of Education site that sets up discussions about Scratch for educators. There are several resources about coding with Scratch available on that site. ScratchJr (https://www.scratchjr.org/) is a dedicated app for young children (ages five to seven) to program their own interactive stories and games.

The following resources should be consulted for their direct relevance to writing codes and using Scratch in the classroom:

- Scratch Rubrics: This was designed as a companion resource for assessing the use of Scratch in class. It is available at this link: http://scratched.gse.harvard.edu/resources/rubric-assessing-scratch-projects-draft-0
- Lesson Plan for Scratch: A quick search online will produce several useful tools for lesson plans about Scratch in the classroom.

IMPORTANT This toolkit is a very valuable resource whose activities and tutorials can be used to start a class in Scratch.

- Storyboarding: This is an effective concept in programming, and toolkits have been created for it. A storyboard helps with laying out program tasks in sequential order for ease of developing and executing. It also helps with debugging code issues. Storyboarding is valuable in a team project, as every team member can design their own part independently with a storyboard.

NOTE A storyboard4 performs tasks such as pre-visualising a motion picture, animation, motion graphic or interactive media sequence, making it a suitable graphic organiser for the sequential display of images and illustrations.

2.0 Hello Scratch

The look and feel of Scratch have been presented in the previous section, and you now can start making codes and tinkering with the environment. Hello Scratch and the accompanying tutorials are meant to get you into writing your own codes and executing them. Hello Scratch will use a systematic approach that is simple and can always help you or your class through problems and design. Each of the activities is in sequential progression, with new concepts added to each activity to enrich your knowledge and give you practical view of the various blocks and what you can do with them. Go through the walkthrough with representations in Figure 2.1.

Walkthrough

Start a new project for each activity/tutorial and name the project accordingly. Click on a sprite or select a new sprite from the Sprite List area. You may leave the backdrop as is or select a new backdrop from the Backdrop modification area. Afterwards, go to the Blocks palette and select the required block, costume and/or sound for the sprite. Drag the selected block to the Scripts area, where you can snap them together. If you want to edit the costume, then select the costume, and the Scripts area will change so that you can do so. You can test the blocks snapped together by clicking on them in the Scripts area and viewing their execution on the Stage. When you are okay with your script, you can run it by clicking the Green flag on the Stage. To reuse your script, drag it to the Backpack, where the snapped blocks will be kept. You can also add costumes or recorded sounds to the Backpack. Your actions are saved automatically when you are logged in online.

Figure 2.1 Click on the Create button to start the editor

The Scripts area, where blocks are snapped together

The Blocks palette for selecting the scripting block, costume and sound

The Stage for displaying executed blocks

The Backpack that stores used blocks and costumes for reuse

The Sprite list area for editing

The backdrop modification area

Green flag is clicked to execute the scripts
2.1 Explore

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Hello_Scratch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Description</td>
<td>This activity is to make the sprite say, “Hello Scratch.” It also shows how to switch from one sprite or backdrop to another.</td>
</tr>
<tr>
<td>Resources</td>
<td><a href="https://scratch.mit.edu/projects/163283041/">https://scratch.mit.edu/projects/163283041/</a></td>
</tr>
<tr>
<td>Screenshot</td>
<td><img src="image_url" alt="Hello Scratch Screenshot" /></td>
</tr>
</tbody>
</table>
| Expected Learning Outcomes | At the end of the activity, you should be able to:  
  - Use the various parts of the editor  
  - Write and execute a simple script  
  - Use the Scripts and Costumes blocks  
  - Change and switch between sprites and backdrops  
  - Produce sound from the script |
| Components and Attributes | The **Event**, **Motion** and **Sound** blocks are used. |
Blocks

Walkthrough:
1. Log in to https://scratch.mit.edu/ with your account details.
2. Click on the Create button (Figure 2.2) to open the editor.
3. Name and save your project by clicking File and Save (Figure 2.3). It is good programming practice to give a name to the project once you start.

Figure 2.2 Click on the Create button to start the editor

Interactive Learning Object Development Toolkit for Teachers
4. In this activity, you will start with the default backdrop and sprite provided in the editor. Head to the Blocks palette (Figure 2.4). Click on Scripts, then Events to drag the green flag block to the Scripts area. In doing this, make sure the sprite is selected in the Sprite List area. There should be one sprite in the editor, named “Sprite1.”
5. Whilst in the Blocks palette, click on the Looks menu to make its block appear. Then drag this block, as shown in Figure 2.5.
6. Now live test what has been done by clicking on the snapped block called Scripts. Look at the difference between (a) inactive blocks and (b) active blocks in Figure 2.6. A halo-like yellow outline appears around the block. As shown in the Figure 2.6a, the text area “Hello Scratch” is highlighted, indicating editing mode. The text was actually changed from “Hello” to “Hello Scratch!”

![Figure 2.5 The Blocks palette and the Scripts area, showing the activity carried out](image)

![Figure 2.6 The Scripts, showing (a) inactive blocks and (b) active blocks](image)
7. When the Script is live tested — i.e., when the blocks are active in the Scripts area — a corresponding action is displayed on the Stage. This depends on what the Script is written to do or what the sprites are meant to perform. Figure 2.7 shows the action that took place.

![Figure 2.7 The Stage, showing the action performed when the Script is live tested](image)

8. The action should be tested with the green flag button, which is the execution button to run all the Scripts. Live testing is good to see instantly what a selected Script is doing, but the green flag runs the whole program from start to finish.

9. Congratulations! Your program ran. With that completed, you are now going to choose a different costume for the sprite and a different backdrop for the Stage, then run it again.

10. To select a different costume for the sprite, go to the Blocks palette and select the Costume menu. Ensure the sprite is active in the Sprite List area. As shown in Figure 2.8, there are two costumes in the Blocks palette (costume1 and costume2).

![Figure 2.8 The Blocks palette with the active costume menu](image)
NOTE: When the costume menu is active in the Blocks palette, it reveals several icons for editing and creating a new costume. You can place your mouse on each to see what each icon does.

11. You can also change the backdrop by selecting the backdrop in the **Backdrop area** and the **Blocks palette** (Figure 2.9), respectively, and going into the Backdrop Library (Figure 2.10) to make a selection.

![Figure 2.9 Selecting a new backdrop](image)

![Figure 2.10 The Backdrop Library](image)

12. In the Backdrop Library, the xy-grid backdrop is selected for emphasis. As shown in Figure 2.11, the xy-grid shows the length and breadth of the Scratch stage. On the x-axis, which is the horizontal expanse of the stage, x starts from -240 on the left-hand side to +240 on the right-hand side, which equals 480 in total size. On the y-axis, which is the vertical direction, it starts from -180 at the top and goes to +180 at the bottom, making a total of 360 in height. This is important for several programming
considerations. For instance, the locus of any object (sprite) on the stage is indicated by the xy point of the object. Additionally, you can site a sprite in the scripting blocks simply by quoting the xy point. Directions of movement depend on the xy point as well.

![Figure 2.11](image1.png)

**Figure 2.11** The xy-grid backdrop

13. The new selected backdrop is named (Figure 2.12) in the text area. Whatever it is named will be what shows in the scripting blocks.

![Figure 2.12](image2.png)

**Figure 2.12** Naming a backdrop

14. Some additional blocks will have to be introduced to reflect the additional choices. Since we have two costumes for the sprite and two backdrops, we will tell the code to switch between the two. The switch backdrop and switch costume blocks (Figure 2.13) will be introduced and will be coded on the sprite for now. The program will recognise the sequence and display how we have arranged it, showing the scene1 (backdrop) and
costume1 (sprite) together to say “Hello Scratch!” Afterwards, it will switch to scene2 (backdrop) and costume2 (sprite) and display the dialogue “Hello Scratch!”.

![Code block](image)

**Figure 2.13** The block of code to say Hello Scratch! with two backdrops and costumes

15. To show some scripting on the backdrop, we will input blocks to play sound when the backdrops are switched (Figure 2.14). With that set, our code blocks will run with a sound for each backdrop.

![Code block](image)

**Figure 2.14** The block of code to introduce sound for each backdrop

**CHALLENGE**: At the end of completing this activity, you are challenged to change the sprite and two or more costumes. Also change the backdrop or add a third backdrop. Locate another sound file and replace the two sound files in the backdrop. Add your own statement to Hello Scratch! and test the result.
2.2 Animation

<table>
<thead>
<tr>
<th>Project Name</th>
<th>MeetnGreet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Description</td>
<td>In this activity, we animate the character (Bovi) by having him come down the stairs, welcome the audience and walk off the screen.</td>
</tr>
<tr>
<td>Resources</td>
<td><a href="https://scratch.mit.edu/projects/163283275/">https://scratch.mit.edu/projects/163283275/</a></td>
</tr>
<tr>
<td>Screenshot</td>
<td><img src="image" alt="Screenshot" /></td>
</tr>
</tbody>
</table>

<p>| Expected Learning Outcomes | At the end of this activity, you will have learned how to: • set up your own sprite and backdrop; • set the stage for animation; • break down your scripts into scenes for ease, clarity and manageability; • use additional blocks; and • add sound. |</p>
<table>
<thead>
<tr>
<th>Components &amp; Attributes</th>
<th>Event, Control, Sound, Motion, Sensing, Looks, Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocks</td>
<td></td>
</tr>
</tbody>
</table>

- **when clicked**
- **broadcast** Scene 1
- **play sound** Andy_G_Cohen___11___Space_Outro.mp3
Walkthrough:

1. In this activity, there are three things the character (Bovi) is required to do:
   • come down the stairs;
   • welcome the audience; and
   • walk off the screen.
2. Log in to Scratch and open the editor. After opening, look for the backdrop that shows a staircase. Through the backdrop area, we can find a corresponding backdrop that fits the target from the corresponding backdrop library. Having selected the backdrop, let’s name it Welcome1 and then delete the default backdrop.

3. Creating a sprite that walk, talks and walks away is not daunting but requires logic. We will have to improvise using the various positions and postures to achieve our goal. Let’s select Bovi by clicking on the new sprite folder from the Sprite List area (Figure 2.15). By default, Bovi has 13 costumes that we can use to select the appropriate posture we need at different intervals of the animation.

4. Additional work needs to be done to set the moves for Bovi. Recall the xy-dimension of the stage, as this will help you to position Bovi correctly.

**IMPORTANT:** For both the sprite and the backdrop, there is always a centre point that represents the location. Each time, the centre point is the xy location that the script will recognise.

5. In setting Bovi for action, six costumes are configured: two for steps, three for speech and one for walk. This can be done by selecting the demeanours that best mimic walking and speaking. After you have picked a disposition, you can do this optional activity. Using pencil and paper, imagine and draw how a person will walk down the stairs from the right of the screen, stand for a period to welcome the audience facing the camera, and walk away to the left of the screen. Put a point at each location where the character will be in time, and determine the locations’ position in terms of the xy-coordinates. Trying this will help you begin to solve a major part of the animation activity.
6. Now you can code by putting the execution block on the backdrop. We are introducing the use of scenes in programming, just like in theatre (Figure 2.16). This is an important practice for sequencing tasks in programming.

![The broadcast block](image)

**Figure 2.16** The use of a broadcast block in place of an execution block

NOTE: There will be only one green flag execution block, and it will be on the backdrop. This is a practice we have adopted throughout the activity and tutorial sections and encourage you to keep up with. As you will see, this saves you a lot of hassle and makes your code easy to read and debug.

7. Sound is added on the backdrop. In this case, the selected sound was from the local computer. It was in .mp3 format and small for use in the program. You can add sound by clicking the Sound menu in the Blocks palette (Figure 2.17) and retrieving the sound from the folder icon. When the sound is loaded, you can edit it and even reduce it to the size you want.

![Editing sound in the Sound menu on the Blocks palette](image)

**Figure 2.17** Editing sound in the Sound menu on the Blocks palette

IMPORTANT: Make sure the sound format is .mp3, which runs generically on Scratch and is smaller than most other sound file formats. You have a limited file size for your project and don’t want a sound to take a big chunk of the space.

8. Now, let’s click on the sprite where our main blocks of activity will be. Start by using the Event block to refer to the Scene1 broadcast in the backdrop. There are six costumes,
so you have to switch to the first costume you want to use. Then select where the sprite will start from.

9. Follow the remaining blocks for the sprite. As you arrange the blocks, make sure to test by clicking on the Script.

**NOTE:** You will need to manoeuver the sprite’s costumes at each turning to align them with the appropriate posture.

10. An important block introduced in this activity is the loop event block (Figure 2.18). At the stage where the sprite is walking off the stage, you have to specify to the script that the sprite should keep walking until it is off the stage. A simple logic is to introduce a less-than operator first so that the sprite moves to a position less than \(-240\), which is the extreme position of the x-axis; \(-360\) is selected. Then inside the loop, x changes at a rate of \(-5\) so we can see that it’s walking.

![Figure 2.18 The loop event block](image)

**CHALLENGE:** At the end of completing this activity, you are challenged to change the type of sound played in the project and change the orientation of the sprite so that it moves from left to right rather than right to left. Bear in mind that the backdrop will need to be changed and so too will the positioning of the sprite.
## 2.3 Stories

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Project Intro</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Description</strong></td>
<td>In this activity, a story is completed. Sprite (Bovi) walks down the stairs, welcomes the audience and walks offstage. The next scene shows a sprite (Alex) dancing. Bovi enters and a conversation ensues. At the end, they show a list of projects to be practised.</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td><a href="https://scratch.mit.edu/projects/163283606/">https://scratch.mit.edu/projects/163283606/</a></td>
</tr>
<tr>
<td><strong>Screenshot</strong></td>
<td><img src="image" alt="Screenshot" /></td>
</tr>
</tbody>
</table>
| **Expected Learning Outcomes** | At the end of this activity, you will have learned how to:  
- remix a project;  
- prepare a complete story;  
- edit a costume;  
- use the different graphic types in Scratch;  
- use more than one sprite concurrently; and  
- create a new sprite. |
Interactive Learning Object Development Toolkit for Teachers

Components & Attributes

Event, Control, Sound, Motion, Sensing, Looks, Operators

Blocks

Stage

when clicked

broadcast Scene 1

switch backdrop to Welcome 1

play sound Andy_G_Cohen___31___Space_Outro.mp3 until done

when I receive Scene 2

switch backdrop to Introduce 2
when I receive
show
switch costume to Bovi step 1

go to x: 53 y: 60
move -20 steps
wait 1 secs
switch costume to Bovi step 2

go to x: 20 y: 10
move -15 steps
wait 1 secs
switch costume to Bovi Speak 1

go to x: 20 y: -20
say Hello Scratchers! for 2 secs

say Welcome to Project Learn Scratch for 4 secs

Part 1: Snap with Part2 below
Part 2: Snap with Part1 above

wait 1 secs

switch costume to Bovi Speak 2

go to x: 20 y: -20

say Presenting with me is Alex for 3 secs

say She will introduce some cool projects for 4 secs

wait 1 secs

switch costume to Bovi Speak 3

go to x: 20 y: -20

say Let's join her for 3 secs

wait 1 secs

switch costume to Bovi walks

go to x: 20 y: -20

repeat until x position of Bovi < -250

change x by -5

broadcast Scene 2

hide
when I receive Scene 2.1

go to x: 360 y: -20

show

repeat until distance to Alex < 200

change x by -5

say You are having fun Alex! for 4 secs

say Can we get rolling? for 3 secs

broadcast Scene 2.2

when I receive Scene 2.3

hide

Alex

when I receive Scene 1

hide
when I receive Scene 2

show

switch costume to Alex sing

go to x: -90 y: -10

play sound dance around

wait 1 secs

Part 1: Snap with Part 2 below
Part 2: Snap with Part 1 above

1. *repeat 3*
2. *switch costume to Alex sing2*
3. *wait 1 secs*
4. *switch costume to Alex sing3*
5. *wait 1 secs*
6. *switch costume to Alex sing4*
7. *wait 1 secs*
8. *switch costume to Alex sing5*
9. *wait 1 secs*
10. *switch costume to Alex sing4*
11. *wait 1 secs*
12. *switch costume to Alex sing3*
13. *wait 1 secs*
14. *switch costume to Alex sing2*

*switch costume to Alex watch*
*go to x: -90 y: -10*
*broadcast Scene 2.1*

*when I receive Scene 2.2*
1. *say Welcome Bovi, We are ready. for 3 secs*
2. *say Our outline projects are: for 3 secs*
3. *broadcast Scene 2.3*
4. *hide*
Walkthrough:

1. If you have completed the previous activity (MeetnGreet), then you can simply open the project, then save with a new project name (Project_Intro), and you will have the entire asset in one place. Check the link for MeetnGreet to reuse the code. This is a continuation of that project and is meant for you to learn how to reuse and remix projects.

**NOTE:** You can start to reuse and remix the code in Scratch simply by saving the project as a new project and editing the code.

2. You have to set up the requirements of the project before putting the additional blocks together. Let’s consider a storyboard for this specific project to identify the assets we will use. Sprite (Bovi) comes down the stairs, welcomes the audience and walks off the screen, completing a scene. A new scene comes on afterwards, showing the playground (backdrop) with a sprite (Alex) taking some dance steps to music. Bovi enters the playground and starts a conversation with Alex. Alex introduces the project to the audience, and a display board (sprite) shows. The implication of the storyline is that there will be two backdrops and three sprites used for this project.

3. We already have one sprite and one backdrop from the former project. We will now add the other backdrop and two sprites. There is a playground in the Backdrop Library, so we add it (Figure 2.19).
4. For the sprites, Alex will be selected in the sprite library whilst the project display board will be added from the computer to the editor. Use a graphic application package to design a .jpeg file (Figure 2.21) and add it as a new sprite in the editor (Figure 2.20).
5. Your code begins by broadcasting Scene1. Switch the backdrop to Welcome1 and play a sound to start together with Scene1 (Figure 2.22).

![Figure 2.22 Scene1 broadcast from the backdrop](image)

6. Go to the sprite (Bovi) and bring it on Scene1 (Figure 2.23). The show block is used since we have more than one sprite now. The block makes sure the sprite come on and does not get hidden. Follow the details in the previous project for the breakdown of the actions of Bovi in Scene1.

7. The last thing Bovi is required to do in Scene1 is broadcast Scene2 and hide.

![Figure 2.23 Sprite Bovi in edit mode (a bitmap graphic)](image)

8. Scene2 broadcasts another backdrop (Introduce1). This is done on the backdrop Scripts. According to the storyline established initially, sprite (Alex) comes onstage at Scene2. Note Alex was hidden for Scene1, so the first thing is to show Alex in Scene2.

9. Alex’s activity in Scene2 is dancing, and she has to be configured for that. Now, Alex has four costumes by default. So, we look at the ones that reflect dancing or can be edited for dancing. Alex is a vector graphic, and just clicking the costume sets it in editing mode (Figure 2.24). Duplicate a costume to edit and keep the original costume. You can view the move adjustment on the stage.
NOTE: There are two types of image file in Scratch — bitmap and vector — and editing is slightly different for each (Figures 2.23 and 2.24).

10. Recall that Alex was hidden for Scene1, so in Scene2, Alex should be set to show. Complete the coding with the costumes you made. Afterwards, broadcast Scene2.1

11. Go back to Bovi to introduce him to Scene2.1. When Scene2 was broadcast, Bovi was set to hide. In Scene2.1, he should be set to show. However, mind the location Bovi ended with in Scene1 —240 on the x-axis (the left side of the stage). In Scene2.1, he will
have to walk in from the right +240 x-axis. You can maintain the same point on the y-axis.

![Figure 2.26 Location where Bovi ended the movement in Scene2.1](image)

12. One more important thing in Scene2.1 is the distance Bovi will walk before stopping close to Alex. The less than operator block is used, and 200 is set as the distance from Alex where Bovi will stop. Afterwards, Scene2.2 is set for Alex’s response.

![Figure 2.27 Distance between Bovi and Alex](image)

13. After Alex responds to Bovi, Scene2.3 is broadcast to usher in the display sprite (board). Note that the board should be set to hide at Scene1. Then at Scene2.3, it should be set to show. Bovi and Alex should also be set to hide at this point for Scene2.3, which completes the story.

14. Make sure you test as you go to confirm the additions work appropriately.

**CHALLENGE:** At the end of completing this activity, you are challenged to design your own sprite and insert it into the project, record your sound for the new sprite and extend the conversation by making the sprite think the words instead of saying them.
### 2.4 Game

<table>
<thead>
<tr>
<th>Project Name</th>
<th>DiceMaths</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Description</strong></td>
<td>This project is a game that uses dice as a counter for arithmetic operations. On pressing the spacebar, two dice are selected, then a dialogue box shows an arithmetic operation. A space for a user response is displayed, and depending on the answer supplied, it indicates whether the answer is right or wrong. The game ends when the score reaches 20.</td>
</tr>
<tr>
<td>Screenshot</td>
<td><img src="image" alt="Screenshot" /></td>
</tr>
</tbody>
</table>
| **Expected Learning Outcomes** | At the end of this activity, you will have learned how to:  
  - call a variable;  
  - store a value on a variable;  
  - call for user input; and  
  - use keyboard input in Scratch. |
| **Components & Attributes** | **Event, Control, Sound, Motion, Sensing, Looks, Operators, Data, Keyboard Input and Mouse Action** |
Blocks

Backdrop
Walkthrough:

1. The project is a game application using dice. The assets for the project include two dice faces showing one to six and a board for playing the dice. Three sprites are used with one backdrop.

2. Start a new project in the editor and save as required. Set up the assets. For this project, a die set with six faces was downloaded, whilst the other die set was created in Scratch (Figure 2.28). To make the die, create a new sprite. Go to the Costumes menu on the Blocks palette and click on the paintbrush to paint a new costume. You can draw a square by selecting the rectangle tool and pressing shift down to draw. A circle can also be drawn by selecting the ellipse tool and pressing shift down to draw. Note that the drawn shape is in bitmap mode. This can be done for all the required six faces of the die.

![Figure 2.28 Drawing an object using the tools on the Costume menu](image)

3. After creating the assets, start coding from the backdrop, then make a new variable. You make a new variable by clicking on the data block and giving it a name. Only a global variable can be made from the backdrop (Figure 2.29). However, either a global or a private variable can be made from an active sprite (Figure 2.30).

**NOTE**: Global variables are usable across the whole project. They can be called (referenced) and changed by any sprite or stage. Private variables are local to the sprite that they were created for. A stage cannot have a private variable. It cannot be changed by another sprite but can be called by another sprite using the “(of)” block.
4. In this project, we will create three global variables (Figure 2.31), and you will call all the variables at the beginning of the project to make the program aware of their existence (Figure 2.32). The variables are for Score reporting using the first die and second die.

Figure 2.31 The three variables created (Score, jdice, ndice)
5. The next block of code requests the user’s input to start the game (Figure 2.33). Obviously, in most games, the user is required to click a start button. In this case, we want the user to make use of the keyboard to get started. The program does not proceed until a response is given back. This capacity is important in programming for user input.

6. Having completed those initial steps, a broadcast of the next scene is made. All the sprites are set in position for the scene. The importance of this is that both dice can appear on the board in the right way.

7. In this project, the game is played from the board, so a scene is broadcast from the board to the two dice for action. A wait event is placed for the dice to set their initial random numbers and store these as variables whilst waiting for the user’s action.

**IMPORTANT:** It can be tricky to store values in a variable, as sending the entire request at the same time may elicit mixed responses. Ensure the program follows an appropriate sequence to initially store the value before the user asks.

8. By pressing the space bar, the stored values are called out in a dialogue and it will be the exact values showing on the two dice.

9. The arithmetic operation is set on the board. To make this happen, use parentheses according to the BODMAS order of mathematical operations (brackets, order, division, multiplication, addition and subtraction). The join operator more or less does the work.
of a parenthesis in Scratch, and you can have layers of brackets with the innermost parenthesis performing the first operation and going on till the outermost. Using the join operator block (Figure 2.34), you can infuse letters, numbers and special characters.

Figure 2.34 Using the Join block is valuable for arithmetic operations

**IMPORTANT**: This is a potential source of error in the program, so you can start by writing what you intend to do on paper and fixing the equation in brackets. Then you will know where to start and what to expect.

10. Look closely at the loop that holds the answer, and check this against the correct answer. This is important to ensure that the game displays accurate figures. After you have completed the loop, a cycle is completed. A broadcast message goes out again for the user to act by pressing the space bar.

**IMPORTANT**: It is important that you get the first loop working before adding other loops, to avoid confusion and errors in the program.

11. After getting the operation right on the board, you can program the dice. They have similar code but pass different variables. The simple activity here is this: **wait to receive a broadcast, pick a number at random and store it on the variable**. This will also be required to show the die for the number that was selected, so a loop is created to activate it with the different costumes of the die.

12. Back to the board. Recall that the program was set for forever. To stop the program, a new call has to be made with a different set of blocks that terminates the program and displays the score. Figure 2.35 shows the call that sets the program to end when the score reaches 20. In a game, this is valuable, as various levels of difficulty can be created with this type of setting. A timer can be set as well, such that when a particular period has elapsed, the program terminates.

Figure 2.35 Block of code to terminate the game when the score reaches 20
**CHALLENGE:** At the end of completing this activity, you are challenged to use a timer to terminate the program instead of the score.

**In the Resources:**
The list of images below is used to represent the dice in this tutorial.
“nd1.png” is based on Dice, by Alex Fuller, from the Noun Project.
https://thenounproject.com/term/dice/10541/, used under a CC BY 3.0 licence
“nd2.png” is based on Dice, by Alex Fuller, from the Noun Project.
https://thenounproject.com/term/dice/10541/, used under a CC BY 3.0 licence
“nd3.png” is based on Dice, by Alex Fuller, from the Noun Project.
https://thenounproject.com/term/dice/10541/, used under a CC BY 3.0 licence
“nd4.png” is based on Dice, by Rohith M S, from the Noun Project.
https://thenounproject.com/term/dice/10541/, used under a CC BY 3.0 licence
“nd5.png” is based on Dice, by Alex Fuller, from the Noun Project.
https://thenounproject.com/term/dice/10541/, used under a CC BY 3.0 licence
“nd6.png” is based on Dice, by Mike Valstar, from the Noun Project.
https://thenounproject.com/term/dice/10541/, used under a CC BY 3.0 licence
### 2.5 Extensions and Simulations

<table>
<thead>
<tr>
<th>Project Name</th>
<th>LOC_LED Robot</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Description</strong></td>
<td>This project makes a robot which beeps and has a flashing antenna using a Raspberry Pi. This is a case of coding a hardware extension with Scratch software. Whilst Scratch is embedded in Raspberry Pi OS, a script is written to trigger a sound for the conjoined LED-wire. NOTE: A Raspberry Pi complete package with the GPIO connection is required for this project.</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td><a href="https://scratch.mit.edu/projects/163283963/">https://scratch.mit.edu/projects/163283963/</a></td>
</tr>
</tbody>
</table>
| **Expected Learning Outcomes** | At the end of this activity, you will have learned how to:  
  - connect a hardware extension with Scratch;  
  - listen to a trigger from hardware through a set of blocks; and  
  - create a sound with hardware. |
Components & Attributes

Scripts were written for one sprite. **Event, Control** and **Sound** scripts were used.

**Blocks**

![Image of blocks]

**Walkthrough**

**About the Raspberry Pi**

The Raspberry Pi provides low-cost and flexible hardware for experimenting with programming and electronics. It is a computer capable of doing most of the things a laptop, desktop, tablet or phone can do when coupled with the peripherals (monitor, keyboard and mouse), and it is available for $25–35 dollars, depending on the model. Because of the incredibly low price but relatively strong computing power, you can use the Pi to do all kinds of interesting things that you would not consider doing with your desktop or laptop machine, such as running home
automation, media servers, dedicated graphics workstations, an email terminal, a streaming media interface for your TV, and much more.5

The first generation (Pi 1) was released in February 2012 in a basic model A and a higher specification model B. A+ and B+ models were released a year later. Raspberry Pi 2 model B (Figure 2.36) was released in February 2015 and Raspberry Pi 3 model B in February 2016. You can purchase the Raspberry Pi in most online stores, but you can get the most up-to-date only through the Raspberry Pi Foundation, https://www.raspberrypi.org/products/.

As an introduction to Raspberry Pi, watch these simple but useful videos:

- What is a Raspberry Pi? https://youtu.be/uXUjwk2-qx4
- Setting up your Raspberry Pi. https://youtu.be/ZYE_ob_rx1Y

Making an antenna for your robot with an LED

You will program a small light called an LED (light-emitting diode) to flash. First you will need to make a circuit.

- The LED has a short leg and a long leg. Slot a jumper wire onto the long leg.
- Slot the resistor into the other end of the same jumper wire.
- Add another jumper wire to the other end of the resistor.
- Take another jumper wire and slot one end onto the short leg of the LED. You should end up with something that looks like Figure 2.37.

5 See http://lemasney.com/consulting/2013/08/05/an-introduction-to-arduino-raspberry-pi-and-open-source-hardware/
1. Find the first 3v3 pin and a GND pin on your Raspberry Pi using the diagram below:

![Figure 2.38 LED wire pinned on a Raspberry Pi](image)

The general purpose input output (GPIO) pins on the Raspberry Pi speak and listen to the outside world and can be controlled or programmed. Each pin has a specific role. To make life easier, the pins are numbered for reference. A 3v3 pin is for power and a GND pin is for grounding.

2. Plug the resistor jumper wire into the 3v3 pin on your Raspberry Pi and the other jumper wire into the GND pin.

3. Plug in the micro USB power supply. You should see some text appear on your screen.

**NOTE**: How the light antenna works

- Now you have a circuit, and the LED should be on. If it is not, make sure that you have plugged the jumper wires into the correct pins, by checking the diagram above.
- So, why does the LED shine?
- When the circuit is plugged into the Raspberry Pi GPIO pins, electricity can flow through it. This flow is called the current. The LED only lights up when electric current flows from the long leg through the bulb to the short leg.
- The resistor reduces the amount of electric current passing through the circuit. This protects the LED from breaking, as a high current would make the light shine more brightly and then stop working.

Make the antenna flash with code

Now that you have an antenna that lights up, you can write a program to tell the LED when you want it to be on.

For this section, you will need to use pin 17 rather than 3v3 to power your LED. Pin 17 is special as it can switch power on and off – if you tell it to! Follow the instructions below to learn how to switch pins.

4. Shut your Raspberry Pi down and remove the power cable. Move your jumper wire that is connected to a resistor from the 3v3 pin to the GPIO pin 17. See Figure 2.39 to make sure that your circuit is correct.

5. Connect the power cable to the Raspberry Pi and wait for it to boot.
6. Open Scratch by clicking on Menu and Programming, followed by Scratch.
7. Click on Edit and Start GPIO server if it has not been started already.

8. Right-click on the Scratch cat and choose delete from the menu.
9. Then click on the button for a new sprite and choose robot3 from the fantasy folder.

![Figure 2.41 Selecting a new sprite icon](image1)

10. Click on control. Drag the “when green flag clicked” block onto the scripts area. Then connect a broadcast block underneath. Click on the dropdown menu on the broadcast block and select new.

In the message name box, type config17output. This instruction will tell the Raspberry Pi that pin 17 will be an output. This is because you are telling the pin to turn on and off an LED, which is an output component.

![Figure 2.42 A broadcast script for config17output](image2)

11. Drag the “when space key pressed” block onto the scripts area. Then click on Sound, drag the “play sound” block onto the scripts area and connect it to the control block.

![Figure 2.43 Making a sound script](image3)

12. Click on the Sound tab above the scripts area and then click on Import. Choose Electronic and then ComputerBeeps2. This will add it to the sounds tab.

13. Now go back to the scripts area by clicking on the scripts tab. Click on the dropdown box next to “play sound.” Choose the sound you just imported from the menu.

![Figure 2.44 Adding sound to the script](image4)

14. Test that your program so far is working by pressing the space key. It should beep!

15. Save your work so far by clicking on File and Save As. Name your file Robot and click OK.

16. Click on the control in the blocks palette, drag a broadcast block to your scripts area and attach it to the “play sound” block. Click on the dropdown menu on the broadcast block and select new.
In the message name box, type gpio17on. This instruction will tell the Raspberry Pi to light the LED.

![Figure 2.45 Adding a broadcast block for GPIO pin 17](image)

17. Drag a “wait 1 second” block onto the scripts area and connect it to the broadcast block.
18. Test your program by clicking on the robot sprite. You should see the LED shine and stay on.
19. Drag another broadcast block onto your scripts area and connect it to the “wait 1 second” block. Click on the dropdown menu on the broadcast block and select new.
20. In the message name box, type gpio17off. This will switch off the LED.
21. Now add another “wait 1 second” block to the script.
22. Test your program again by clicking on the robot sprite. You should see the LED turn on for one second and turn off for one second.

![Figure 2.46 A complete script for the activity](image)

**In This Resource**

Physical resources required include:

- Complete Raspberry Pi package
- LED
- Resistor
- Jumper wires
To set up the Raspberry on your laptop/PC, the following pages provide additional helpful details:

https://diyhacking.com/connect-raspberry-pi-to-laptop-display/
## 3.0 Scratch Tutorials

### 3.1 Book Information Session

<table>
<thead>
<tr>
<th>Project Name</th>
<th>BookInfoSession</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Description</strong></td>
<td>This project storifies and animates this book in an information session. When it is initialised, an introductory page with instructions shows on screen, and then it enters into the class. When any one icon is clicked, a recorded sound is played. When any other icon is clicked, the previously sounding one stops. The Exit button can be clicked to close the session.</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td><a href="https://scratch.mit.edu/projects/163285782/">https://scratch.mit.edu/projects/163285782/</a></td>
</tr>
</tbody>
</table>
| **Expected Learning Outcomes** | At the end of this activity, you will have learned how to:  
  - loop events;  
  - automate a text using duplication and angle of sight;  
  - add multiple sounds to a script;  
  - control script actions from other scripts; and |
• use the mouse action to implement a script.

| Components & Attributes | Scripts were written for 20 sprites and the backdrop. **Motion**, **Event**, **Looks**, **Sound**, and **Control** scripts were used. |

**Blocks**
**Backdrop Script**

```
when [flag] clicked
switch backdrop to info_back
play sound Andy_G_Cohen___10___Worky_Worky.mp3
broadcast InfoPage

when I receive StartPage
switch backdrop to school1
wait 8 secs
switch backdrop to school2
wait 9 secs
stop all sounds
play sound DoorClose
broadcast ClassPage
switch backdrop to blackboard

when I receive ClosePage
wait 6 secs
switch backdrop to BISClose
play sound Ovation-Mike_Koenig-1061486511.mp3 until done
```
W Scripts

- When I receive InfoPage
  - Hide

- When I receive StartPage
  - Repeat 2
    - Show
    - Set y to -120
    - Point in direction 90
    - Wait 1 secs
    - Turn (15 degrees)
    - Wait 1 secs
    - Point in direction 90
    - Wait 1 secs
    - Turn (15 degrees)
    - Wait 1 secs
  - Hide
Part 2: Snap with Part 1 above

- Repeat 2
- Switch costume to e1
- Wait 1 secs
- Turn 15 degrees
- Wait 1 secs
- Point in direction 90
- Wait 1 secs
- Turn 15 degrees
- Wait 1 secs
- Point in direction 90
- Hide
Part 1: Snap with Part 2 below

when I receive InfoPage
hide

when I receive StartPage
repeat 2
show
set y to -120
switch costume to
point in direction 90
wait 1 secs
turn 15 degrees
wait 1 secs
point in direction 90
wait 1 secs
turn 15 degrees
wait 1 secs
point in direction 90
set y to 132
Part 2: Snap with Part 1 above

- switch costume to n
- wait 1 secs
- turn 15 degrees
- wait 1 secs
- point in direction 90
- wait 1 secs
- turn 15 degrees
- wait 1 secs
- point in direction 90
- hide

Part 1: Snap with Part 2 below

- when I receive InfoPage
  - hide
- when I receive StartPage
  - repeat 2
    - show
    - set y to -120
    - switch costume to c
    - point in direction 90
    - wait 1 secs
    - turn 15 degrees
    - wait 1 secs
    - point in direction 90
    - wait 1 secs
    - turn 15 degrees
    - wait 1 secs
    - point in direction 90
    - set y to 130
repeat 2

switch costume to t

wait 1 secs

turn (15) degrees

wait 1 secs

point in direction 90

wait 1 secs

turn (15) degrees

wait 1 secs

point in direction 90

hide
o Scripts

Part 1: Snap with Part 2 below

- set y to -120
- switch costume to o
- point in direction 90
- wait 1 secs
- turn 15 degrees
- wait 1 secs
- point in direction 90
- wait 1 secs
- turn 15 degrees
- wait 1 secs
- point in direction 90
- set y to 130

Part 2: Snap with Part 1 above

- switch costume to e1
- wait 1 secs
- turn 15 degrees
- wait 1 secs
- point in direction 90
- wait 1 secs
- turn 15 degrees
- wait 1 secs
- point in direction 90
- hide
when I receive InfoPage
hide

when I receive StartPage
repeat 2
show
set y to -120
switch costume to m
point in direction 90
wait 1 secs
turn (15 degrees
wait 1 secs
point in direction 90
wait 1 secs
turn (-15 degrees
wait 1 secs
point in direction 90

set y to 130
Part 1: Snap with Part 2 below

repeat 2
switch costume to r
wait 1 secs
turn (15 degrees
wait 1 secs
point in direction 90
wait 1 secs
turn (-15 degrees
wait 1 secs
point in direction 90

hide
Part 2: Snap with Part 1 above
E2 Scripts

```
when I receive InfoPage
hide

when I receive StartPage
repeat 2
  show
  set y to -120
  point in direction 90
  wait 1 secs
  turn ( 15 degrees
  wait 1 secs
  wait 1 secs
  point in direction 90
  wait 1 secs
  turn ( 15 degrees
  wait 1 secs
  hide
```
**Tutor Scripts**

```
when I receive InfoPage
hide

when I receive ClassPage

go to x: 204 y: -107
show
wait (1) secs
say You are welcome for 2.5 secs
say to the Book Info Session for 3 secs
say Click on the icons on the board for more details for 4 secs

when I receive Closepage
say Thanks for attending the Book Info Session for 3 secs
say See you back soon for 2 secs
hide
```

**Intro Scripts**

```
when I receive InfoPage

go to x: 0 y: 20
show
glide 1 secs to x: 0 y: -20
wait 6 secs
broadcast StartPage
hide
```
Exitbutton Scripts

- when I receive InfoPage
  - hide

- when I receive ClassPage
  - show
  - go to x: -200 y: -65

- when this sprite clicked
  - stop all sounds
  - broadcast Closepage
  - wait 6 secs
  - hide
OutlineBox Scripts

```
when I receive [InfoPage]
hide

when I receive [ClassPage]
hide

when I receive [COL]
go back 1 layers
x: -8 y: 112
show
repeat 4
switch costume to [SB4]
wait 3 secs
switch costume to [SB5]

when I receive [OER]
go back 1 layers
x: -8 y: 112
show
switch costume to [SB3]
```
when I receive Scratch
go back 1 layers
go to x: -8 y: 112
show
switch costume to SB2

when I receive Toolkit
go back 1 layers
go to x: -8 y: 112
show
switch costume to SB1

when I receive Closepage
hide
TextBox Scripts

- `when I receive InfoPage`
  - `hide`

- `when I receive ClosePage`
  - `hide`

- `when I receive ClassPage`
  - `hide`

- `when I receive COL`
  - `go to x: -20 y: 40`
  - `go to front`
  - `wait 1 secs`
  - `show`
  - `repeat 3`
    - `switch costume to S4`
    - `wait 5 secs`
    - `switch costume to S5`
    - `wait 5 secs`
Scratch2 Scripts

when I receive InfoPage
hide

when I receive ClassPage
wait 10.5 secs
show
go to x: 128 y: 115

when I receive Closepage
wait 6 secs
hide

when I receive Toolkit
hide

when I receive COL
hide

when I receive OER
hide

when this sprite clicked
broadcast Scratch
hide
Toolkit Scripts

When I receive InfoPage:
- Hide

When I receive ClassPage:
- Wait 10.5 secs
- Show
- Go to x: -43 y: 167

When I receive Closepage:
- Wait 6 secs
- Hide

When I receive COL:
- Hide

When I receive Scratch:
- Hide

When I receive OER:
- Hide
OER2 Scripts

when I receive InfoPage
hide

when I receive ClassPage
wait 10.5 secs
show
go to x: -55 y: 81

when I receive Closepage
wait 6 secs
hide

when I receive Toolkit
hide

when I receive Scratch
hide

when I receive COL
hide
Col2 Scripts

when I receive InfoPage
hide

when I receive ClassPage
wait 10.5 secs
show
go to x: 145 y: 135

when I receive Closepage
wait 6 secs
hide

when I receive Toolkit
hide

when I receive Scratch
hide

when I receive OER
hide

when this sprite clicked
broadcast COL
hide
Interactive Learning Object Development Toolkit for Teachers

OER Scripts
tool Scripts

```
when this sprite clicked
  broadcast Toolkit
  show

when I receive Toolkit
  when I receive InfoPage
  hide

when I receive Toolkit
  when I receive ClassPage
  hide

when I receive Toolkit
  when I receive Closepage
  wait 6 secs
  hide

when I receive COL
  when I receive OER
  go to x: -140 y: 183
  show
  go to front

when I receive Scratch
  go to x: -140 y: 183
  show
  go to front
```
In the Resources:
- The “Worky Worky” sound from an Andy G. Cohen album is adapted under a Creative Commons (CC BY) Licence.
- The “Ovation” sound from Mike Koenig is adapted under a Creative Commons Attribution 3.0 (CC BY) Licence.
### 3.2 Hair Salon

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Ab Hair Clinic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Description</strong></td>
<td>This project is an animation and simulation about the functions of a hairdresser. When the green flag is clicked, an intro page is shown to welcome and give instructions, after which you are directed to watch a demo or try out the design yourself. When trying out, a user is required to select the equipment on the right shelf to perform the operations. There is instant action when the tool crosses the hair.</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td><a href="https://scratch.mit.edu/projects/163287813/">https://scratch.mit.edu/projects/163287813/</a></td>
</tr>
</tbody>
</table>
| **Expected Learning Outcomes** | At the end of this activity, you will have learned how to:  
  - design a simulation for an activity on Scratch;  
  - use the mouse option for movement; and  
  - implement multiple actions with an icon. |
| **Components & Attributes** | |

---

Interactive Learning Object Development Toolkit for Teachers 104
Scripts were written for 14 sprites and the backdrop. **Event, Control, Sound, Motion, Sensing, Looks, Operators and Mouse Action** were used.

**Blocks**

**Backdrop Scripts**

```
when Green flag clicked
broadcast Salon Load

switch backdrop to back2
play sound pop
wait 4 secs
broadcast Go to Menu

when I receive Go to Menu
switch backdrop to back1
```

**DemoButton Scripts**

```
when I receive Salon Load
hide

when I receive Go to Menu

go to x: -80 y: -15
switch costume to but1
show

when I receive Trying
hide

when this sprite clicked
broadcast DemoMode
hide
```
Button2 Scripts

when I receive Salon Load ↓ hide

when I receive Go to Menu ↓
go to x: 50 y: -15
show

when I receive DemoMode ↓ hide

when this sprite clicked
broadcast Trying ↓ hide
Hair Scripts

when I receive Go to Menu
  hide
  stop all

when I receive DemoMode
  go to x: -5 y: -10
  switch costume to Hair1
  show

when I receive Trying
  go to x: -5 y: -10
  switch costume to Hair1
  show

when I receive try_shampoo
  switch costume to Hair2
  go to x: -5 y: -10
  show

when I receive CombedHair
  switch costume to Hair4
  go to x: -5 y: -10
  show

when I receive Salon Load
  hide

when I receive shampooped
  switch costume to Hair2
  go to x: -5 y: -10
  show

when I receive CombingMode
  switch costume to Hair3
  go to x: -5 y: -10
  show

when I receive RinsedHair
  switch costume to Hair3
  go to x: -5 y: -10
  show
Locker Scripts

when I receive Salon Load
hide

when I receive DemoMode
go to x: 190 y: 35
show

when I receive Trying
go to x: 190 y: 35
show

when I receive Go to Menu
hide
Shower Scripts

- When I receive: Salon Load
  - Hide

- When I receive: Go to Menu
  - Hide
  - Stop all

- When I receive: DemoMode
  - Go to x: 180 y: 55
  - Switch costume to: TShower
  - Show
  - Go to front

- When I receive: Trying
  - Hide

- When I receive: shampoosed
  - Wait 2 secs
  - Glide 1 secs to x: 180 y: 55
  - Glide 1 secs to x: 120 y: 40
  - Glide 1 secs to x: 80 y: 25
  - Glide 1 secs to x: 55 y: -5
  - Glide 1 secs to x: 48 y: -1
  - Glide 1 secs to x: 29 y: 40
  - Glide 1 secs to x: -68 y: 40

- Switch costume to: TShower2
  - Glide 1 secs to x: -93 y: 40
  - Glide 1 secs to x: -98 y: -10
  - Glide 1 secs to x: -98 y: -10
  - Glide 1 secs to x: -93 y: 40
  - Glide 1 secs to x: -60 y: 40

- Switch costume to: TShower
  - Glide 1 secs to x: 29 y: 40
  - Glide 1 secs to x: 48 y: -1
  - Glide 1 secs to x: 80 y: 25
  - Glide 1 secs to x: 120 y: 40
  - Glide 1 secs to x: 180 y: 55
  - Go to x: 180 y: 55

- Broadcast: CombingMode
DeComb Scripts
DComb Scripts

when I receive Salon Load
hide

when I receive Go to Menu
hide
stop all

when I receive Trying
hide

when I receive DemoMode
go to x: 187 y: 79
switch costume to TComb1
show
go to front
wait 2 secs
Part 1: Snap with Part 2 below
shampoo Scripts

when I receive Salon Load
hide

when I receive Go to Menu
hide
stop all

when I receive Trying
hide

when I receive DemoMode

go to x: 187 y: -10
show
go to front
wait 21 secs
glide 1 secs to x: 154 y: -12
glide 1 secs to x: 100 y: -15
glide 1 secs to x: 47 y: -52
glide 1 secs to x: 48 y: -1
glide 1 secs to x: 29 y: 40
glide 1 secs to x: -48 y: 40
glide 1 secs to x: -58 y: 7
glide 1 secs to x: -65 y: -40
glide 1 secs to x: -2 y: -30
glide 1 secs to x: 65 y: -20
glide 1 secs to x: 187 y: -10
go to x: 187 y: -10
broadcast shampooed
ExitButton Scripts

when I receive Salon Load
hide

when I receive Go to Menu
hide

when I receive Trying
  go to x: -20 y: -160
  show

when I receive DemoMode
  go to x: -20 y: -160
  show

when this sprite clicked
  broadcast Go to Menu
  hide
TComb Scripts

when I receive Salon Load
hide

when I receive Go to Menu
hide
stop all

when I receive DemoMode
hide

when I receive Trying

go to x: 187 y: 79
switch costume to TComb1
show
go to front

when this sprite clicked

glide 1 secs to x: 146 y: 77

glide 1 secs to x: 105 y: 72

glide 1 secs to x: 47 y: -52

glide 1 secs to x: 48 y: -1

glide 1 secs to x: 29 y: 40

switch costume to TComb2

glide 1 secs to x: -48 y: 40

glide 1 secs to x: -58 y: 7

glide 1 secs to x: -65 y: -40

glide 1 secs to x: -65 y: -40

glide 1 secs to x: -55 y: 7

glide 1 secs to x: -48 y: 40

switch costume to TComb1

glide 1 secs to x: 29 y: 40

glide 1 secs to x: 48 y: -1

glide 1 secs to x: 47 y: -52

glide 1 secs to x: 105 y: 72

glide 1 secs to x: 146 y: 77

glide 1 secs to x: 167 y: 78

glide 1 secs to x: 187 y: 79

go to x: 187 y: 79

switch costume to TComb3
TrComb Scripts

when I receive Salon Load
hide

when I receive Go to Menu
hide
stop all

when I receive DemoMode
hide

when I receive Trying
go to x: 188 y: 10
switch costume to TComb
show
go to front
when this sprite clicked

if costume ≠ "TShower" = 3 then
glide 1 secs to x: 188 y: 10
glide 1 secs to x: 120 y: 2
glide 1 secs to x: 80 y: 0
glide 1 secs to x: 55 y: -5
glide 1 secs to x: 48 y: -1
glide 1 secs to x: 29 y: 40
glide 1 secs to x: -48 y: 40
switch costume to "TComb2"
glide 1 secs to x: -93 y: 40
glide 1 secs to x: -98 y: -10
glide 1 secs to x: -98 y: -10
glide 1 secs to x: -93 y: 40
glide 1 secs to x: -60 y: 40
switch costume to "TComb"
glide 1 secs to x: 29 y: 40
glide 1 secs to x: 48 y: -1
glide 1 secs to x: 80 y: 0
glide 1 secs to x: 120 y: 2
glide 1 secs to x: 188 y: 10
go to x: 188 y: 10
broadcast "CombedHair"
say Congratulations! You can do HairWashing. for 3 secs
else
say Oops! You need to rinse the hair. for 2 secs
Tshampoo Scripts

when I receive Salon Load
hide

when I receive Go to Menu
hide
stop all

when I receive DemoMode
hide

when I receive Trying
go to x: 187 y: -10
switch costume to shampoo
show
go to front

when this sprite clicked
if costume # of TComb = 3 then
    glide 1 secs to x: 154 y: -12
    glide 1 secs to x: 100 y: -15
    glide 1 secs to x: 47 y: -52
    glide 1 secs to x: 48 y: -1
    glide 1 secs to x: 29 y: 40
    glide 1 secs to x: -48 y: 40
    glide 1 secs to x: -53 y: 7
    glide 1 secs to x: -65 y: -40
    glide 1 secs to x: -2 y: -30
    glide 1 secs to x: 65 y: -20
    glide 1 secs to x: 187 y: -10
    go to x: 187 y: -10
    switch costume to shampo02
    broadcast try_shampoo
    say Ooops! You are meant to brush first. for 2 secs
else

Interactive Learning Object Development Toolkit for Teachers
TShower Scripts

when I receive Salon Load
hide

when I receive Go to Menu
hide
stop all

when I receive DemoMode
hide

when I receive Trying
go to x: 180 y: 55
switch costume to TShower
show
go to front
when this sprite clicked

if costume # of Tshampoo = 2 then

  glide 1 secs to x: 180 y: 55
  glide 1 secs to x: 120 y: 40
  glide 1 secs to x: 80 y: 25
  glide 1 secs to x: 55 y: -5
  glide 1 secs to x: 48 y: -1
  glide 1 secs to x: 29 y: 40
  glide 1 secs to x: -48 y: 40

  switch costume to TShower2

  glide 1 secs to x: -93 y: 40
  glide 1 secs to x: -98 y: -10
  glide 1 secs to x: -98 y: -10
  glide 1 secs to x: -93 y: 40
  glide 1 secs to x: -60 y: 40

  switch costume to TShower

  glide 1 secs to x: 29 y: 40
  glide 1 secs to x: 48 y: -1
  glide 1 secs to x: 80 y: 25
  glide 1 secs to x: 120 y: 40
  glide 1 secs to x: 180 y: 55
  go to x: 180 y: 55

  switch costume to TShower3

  broadcast RinsedHair

else

  say Oops! Apply Shampoo first for 2 secs
D-story Scripts

```plaintext
when I receive Salon Load
hide

when I receive Go to Menu
hide

when I receive DemoMode
switch costume to Title
go to x: -82 y: 143
show

when I receive Trying
switch costume to Title
go to x: -82 y: 143
show
```
### 3.3 Building Project

<table>
<thead>
<tr>
<th><strong>Project Name</strong></th>
<th>LOC Building Project</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Description</strong></td>
<td>This project makes available resources or building elements for the user to construct a building from start to finish. When the project is begun, the arrow down key is used to activate the elements whilst the arrow up key moves to different element choices. When an element is loaded and the space bar is pressed, the element is released on the stage and another one is picked up. To exit the project, just press the A key, which deactivates the element from the cursor. You can then press the Exit button. Element colours can be changed with the right and left arrow keys.</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td><a href="https://scratch.mit.edu/projects/162738659/">https://scratch.mit.edu/projects/162738659/</a></td>
</tr>
<tr>
<td><strong>Screenshot</strong></td>
<td><img src="image" alt="LOC Building Project" /></td>
</tr>
</tbody>
</table>
| **Expected Learning Outcomes** | At the end of this activity, you will have learned how to:  
  - call a variable;  
  - store a value on a variable; |
Interactive Learning Object Development Toolkit for Teachers

- call for user input; and
- use keyboard input in Scratch.

**Components & Attributes**

Scripts were written for four sprites and a backdrop. **Motion, Event, Sensing, Pen, Control** and **Looks** scripts were used.

**Blocks**

**Backdrop Scripts**

![Backdrop Scripts](image)

**IntroPage Script**

![IntroPage Script](image)
ClosePage Scripts

when I receive IntroPage
hide

when I receive ClosePage
show
go to x: 240 y: 7
glide 4 secs to x: 0 y: 7

ExitButton Scripts

when I receive IntroPage
hide

when I receive WorkPage
show
go to x: -65 y: 58

when this sprite clicked
broadcast ClosePage
hide

Interactive Learning Object Development Toolkit for Teachers 125
BuildingElement Scripts

when I receive IntroPage
hide

when I receive WorkPage
show

down arrow pressed
repeat

key pressed
until

glide 0.1 secs to x: mouse x y: mouse y

down arrow
when pressed

right arrow key pressed
change color effect by -25

left arrow key pressed
change color effect by 25

up arrow key pressed
next costume

when I receive ClosePage
clear
hide
### 3.4 Animated Music Videos (AMV)

<table>
<thead>
<tr>
<th>Project Name</th>
<th>LOC Music Studio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Description</td>
<td>This project is an animated music video (AMV). In it, a sound file is played and a voiceover is recorded in Scratch. Images are added to feature simultaneously whilst the music is being played. An artist (sprite) dances around whilst the music plays.</td>
</tr>
<tr>
<td>Resources</td>
<td><a href="https://scratch.mit.edu/projects/163263268/">https://scratch.mit.edu/projects/163263268/</a></td>
</tr>
</tbody>
</table>

| Screenshot | ![LOC Music Studio](image) |

<table>
<thead>
<tr>
<th>Expected Learning Outcomes</th>
<th>At the end of this activity, you will have learned how to:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• use a sound file through a project;</td>
</tr>
<tr>
<td></td>
<td>• interface images with a sound file to make a video;</td>
</tr>
<tr>
<td></td>
<td>• put effects on an image;</td>
</tr>
<tr>
<td></td>
<td>• add an image as an artist on the project, with movements and effects; and</td>
</tr>
<tr>
<td></td>
<td>• arrange different costumes for dancing steps.</td>
</tr>
</tbody>
</table>
Scripts were written for two sprites and the backdrop. **Motion**, **Event**, **Control**, **Looks**, **Sound** and **Operator** scripts were used.

---

**Blocks**

**Backdrop Script**

```
when [旗] clicked
broadcast [StartMusic]
set volume to 5%
play sound Andy_G_Cohen_-_09_-_Trophy_Endorphins.mp3
wait 75.9 secs
play sound VoiceOver
```

**Artist Script**
when I receive StartMusic

hide

wait 15 secs

show

go to front

switch costume to costume1

go to x: -250 y: 40

glide 10 secs to x: -50 y: 40

switch costume to costume3

go to x: -50 y: 40

show

Part 1: Snap with Part 2 below

Part 2: Snap with Part 1 above

glide 3 secs to x: -35 y: 75

glide 3 secs to x: -15 y: 105

glide 3 secs to x: 0 y: 125

glide 3 secs to x: 15 y: 105

glide 3 secs to x: 35 y: 75

glide 3 secs to x: 50 y: 40

glide 3 secs to x: 35 y: 10

glide 3 secs to x: 15 y: -15

glide 3 secs to x: 0 y: -40

glide 3 secs to x: -15 y: -15

glide 3 secs to x: -35 y: 10

glide 3 secs to x: -50 y: 40

switch costume to costume2

glide 3 secs to x: 50 y: 40
Part 3: Snap with Part 2 above

repeat 8

switch costume to costume8

wait 1 secs

switch costume to costume9

wait 1 secs

switch costume to costume10

wait 1 secs

switch costume to costume9

wait 1 secs

switch costume to costume8

wait 1 secs

switch costume to costume7

wait 1 secs

switch costume to costume6

wait 1 secs

switch costume to costume7

wait 1 secs

switch costume to costume5

wait 1 secs

switch costume to costume6

wait 1 secs

switch costume to costume7

wait 1 secs

switch costume to costume11

glide 10 secs to x: 400 y: 40
In this resource:
  o This project is inspired by kittymitty6868’s project at
  o The “Trophy Endorphins” sound from an Andy G. Cohen album is adapted under a
    Creative Commons (CC BY) Licence.
  o Mountains, by Jonas B (2006), is adapted under a Creative Commons BY 2.0 Licence,
    https://www.flickr.com/photos/jonasb/120921750 PhotoForClass.
  o Mountain, by TooFarNorth (2007), is adapted under a Creative Commons BY 2.0 Licence,
    https://www.flickr.com/photos/toofarnorth/1500448709 PhotoForClass.
  o Mountains, by DHPersonal (2006), is adapted under a Creative Commons BY 2.0 Licence,
    https://www.flickr.com/photos/toofarnorth/1500448709 PhotoForClass.
  o Mountains, by abdulrahman stock (2012), is adapted under a Creative Commons BY 2.0
  o Mountains, by pthread1981 (2012), is adapted under a Creative Commons BY 2.0 Licence,
    https://www.flickr.com/photos/pthread/7847253398 PhotoForClass.
  o Mountains, by aegidian (2013), is adapted under a Creative Commons BY 2.0 Licence,
    https://www.flickr.com/photos/aegidian/8500579154 PhotoForClass.
  o Mountains, by oatsy40 (2014), is adapted under a Creative Commons BY 2.0 Licence,
    https://www.flickr.com/photos/oatsy40/14291892698 PhotoForClass.
  o Mountain, by Kiwi Tom (2014), is adapted under a Creative Commons BY 2.0 Licence,
    https://www.flickr.com/photos/tom_hall_nz/15613249585 PhotoForClass.
  o Mountains, by Eu_go (2013), is adapted under a Creative Commons BY 2.0 Licence,
    https://www.flickr.com/photos/127534097@N04/15232292419 PhotoForClass.
### 3.5 Electrical Circuit

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Series and Parallel Circuits</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Description</strong></td>
<td>Using Ohm’s Law, the circuits are calculated in series and parallel. When the program starts, the instructor gives an introduction, enters the classroom and presents calculations in series and parallel. The user supplies the corresponding values, and the results appear. The current of a circuit is displayed when the resistance and voltage values are input. Presently, it can be used to calculate a circuit with at most three resistors.</td>
</tr>
<tr>
<td><strong>Resources</strong></td>
<td><a href="https://scratch.mit.edu/projects/163296701/">https://scratch.mit.edu/projects/163296701/</a></td>
</tr>
<tr>
<td><strong>Screenshot</strong></td>
<td><img src="image" alt="Screenshot" /></td>
</tr>
</tbody>
</table>

**Expected Learning Outcomes**

At the end of this activity, you will have learned how to:

- call a variable;
- store a value on a variable;
- call for user input;
- use loops effectively; and
- trigger an action using multiple variables.
Scripts were written for four sprites and the backdrop. **Event, Data, Control, Sensing, Motion, Looks** and **Operators** scripts were used.

**Blocks**

**Backdrop Scripts**

```
when I receive Dashboard
switch backdrop to chalkboard

when I receive Exit
wait 2 secs
switch backdrop to ExitClass
```
**SeriesButton Script**

```
when I receive Intro
hide

when I receive Dashboard
switch costume to button
wait 10 secs
go to x: 36 y: 70
show

when I receive Parallel
hide

when I receive Exit
wait 2 secs
hide
stop all
```
ask Input your value for resistor, R1: and wait
set R1 to answer
ask Input your value for resistor, R2: and wait
set R2 to answer
ask Input your value for resistor, R3: and wait
set R3 to answer
set Resistance to R1 + R2 + R3
ask Input your value for voltage, V: and wait
set Voltage to answer
say join Current (Voltage / Total Resistance) = Voltage / Resistance
wait 3 secs
broadcast Dashboard
stop all
else
say You have a limit of 3 resistors. for 4 secs
broadcast Dashboard
stop all
ParallelButton Scripts

when I receive Intro
hide

when I receive Dashboard
switch costume to button
wait 10 secs
go to x: 36 y: 25
show

when I receive Series
hide

when I receive Exit
wait 2 secs
hide
stop all
when this sprite clicked
broadcast Parallel
switch costume to text

go to x: 36 y: 27

forever

ask How many resistor? and wait

if answer = 1 then

ask Input your value for resistor, R1: and wait

set Resistance to answer

ask Input your value for voltage, V: and wait

set Voltage to answer

say join Current (Voltage / Resistance) = Voltage / Resistance

wait 3 secs

broadcast Dashboard

stop all
Instructor Scripts

```
when I receive Intro
  go to x: -56  y: -27
  switch costume to devin-c
  show
  say Welcome! for 2 secs
  say Today in class for 3 secs
  say we are treating for 3 secs
  say Series and Parallel calculations. for 4 secs
  say Let’s get started. for 3 secs
  broadcast Dashboard
  switch costume to devin-a
  go to x: -156  y: -27
  wait 1 secs
  say Using Ohm’s Law for 3 secs
  say you can calculate in for 3 secs
  say Series or Parallel for 3 secs
  say Select one
  hide
```
ExitButton Scripts

when I receive Intro
hide

when I receive Dashboard
  go to x: -82 y: -50
  show

when I receive Series
  show

when I receive Parallel
  show

when this sprite clicked
  broadcast Exit
  wait 2 secs
  hide

Interactive Learning Object Development Toolkit for Teachers
### 3.6 Flashing Light with Movement Using a Raspberry Pi

<table>
<thead>
<tr>
<th>Project Name</th>
<th>LOC_FlashMove</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Description</td>
<td>This project was inspired by children’s trainers (running shoes) that flash lights when they move. At the start of the project, the sprite moves and light flashes from the LED. The costume changes after some steps, turns in the opposite direction and moves back. Sound is produced with the steps. NOTE: A Raspberry Pi complete package with the GPIO connection is required for this project.</td>
</tr>
<tr>
<td>Resources</td>
<td><a href="https://scratch.mit.edu/projects/163296908/">https://scratch.mit.edu/projects/163296908/</a></td>
</tr>
<tr>
<td>Screenshot</td>
<td><img src="image" alt="Screenshot" /></td>
</tr>
</tbody>
</table>
| Expected Learning Outcomes | At the end of this activity, you will have learned how to:  
  - program a hardware extension with Scratch;  
  - combine movement with a sensor concurrently through a set of blocks; and  
  - add sound with hardware. |
| Components & Attributes | Scripts were written for one sprite. **Event, Control, Sound, Motion** and **Look** scripts were used. |
In This Resource:

Physical resources required include:

- Complete Raspberry Pi package
- LED
- Resistor
- Jumper wires
4.0 Packaging and Distributing Scratch

4.1 Share a Project

A Scratch project can be shared on two levels: online and offline. There are a few differences between the levels, the main one being that sharing online makes a project available to the general community, whilst sharing offline saves the project from the offline editor into the account of a user.

Online Sharing
To share a project on Scratch, follow these steps. A new project can be shared and a saved project can be shared. Anyone with an account on Scratch can share a project except a user whose account is yet to be confirmed (Figure 4.1). In this case, an exclamation mark will show on the sharing tab, indicating a requirement that needs to be fulfilled before sharing can be done.

An unshared project is shown with the name of the account under the project name (Figure 4.2).

IMPORTANT: You can only share a project with your confirmed Scratch account online.

![Figure 4.1 Note the exclamation mark on Share; the account has not been confirmed](image)

![Figure 4.2 How to note an unshared project in the editor](image)

Information is also available on the project page if you cannot share (Figure 4.3).

![Figure 4.3 Information to confirm mail in order to share](image)
So, to be able to share, make sure the account is confirmed by checking your email for the Scratch message. Alternatively, you can follow the instructions given when you click on the exclamation mark (Figure 4.4).

![Figure 4.4 Instructions to confirm your account](image)

After you confirm, sharing is enabled, but each project you have needs to be saved one by one. An unshared project will have a comment on top of the project page (Figure 4.5).

![Figure 4.5 A comment that appears on a page when it is not shared.](image)

**Sharing a new project**
On the project editor, a share button appears if you are logged in. So, after you have completed and tested your project, name it and click share. Sharing is automatically done. A message appears for you after sharing has been completed (Figure 4.6).

![Figure 4.6 Note to inform you that your project has been shared](image)

**NOTE:** Sharing makes your project available to the general community and public to view, comment on and remix your project.

**Sharing a saved project**
By default, every saved project goes into your account space called My Stuff (Figure 4.7). Two labels appear on the left side of the screen, showing Shared Projects and Unshared Projects. To share a project, click the unshared label. All unshared projects will appear in the middle of the
page. Click the project you want to share. This action will take you to the project page, where you can click on the share button.

**NOTE:** You can both share and unshare a project.

![Image of Unshared projects on My Stuff](image)

**Figure 4.7 Unshared projects on My Stuff**

**Offline Sharing**
For a project developed using the offline project editor, the process of adding it to your account online is also called sharing. This process is straightforward.

1. After you have completed your project in the offline editor, click on the File menu and select **Share to website** (Figure 4.8).

![Image of Scratch 2 Offline Editor](image)

**Figure 4.8 Share to website in the offline editor**

2. A form appears (Figure 4.9) with the title **Share to Scratch Website**, and three fields are available for you to fill: Project name, Your Scratch name and Password. If you have already saved your project, it selects that name automatically, but you can edit it to what you want. You should then supply your Scratch username and password, after which you should click the OK button.
3. If the login details supplied are accurate, the form vanishes. To verify your shared project, log into the account you used to share and click on My Stuff (Figure 4.10). You will find the project listed there.

How to Embed a Published Project in an LMS, Blog or Website
1. Go to the published project on the Scratch website. Below it you will see Embed in grey, as shown in Figure 4.11.
2. Click on it. This will reveal the embed link for the project (Figure 4.12). Every link is unique to that particular project. It can also be shared directly on Facebook, Twitter and Google Classroom.

3. Copy the link and paste it into your LMS, blog or website editor. The link should be pasted on the HTML view of your website editor, as it contains some markup code required for running.
4. The project will be available in viewer’s mode when it is published on the frontend of the site, but you may initially have an issue such as the one shown in Figure 4.14. It is a plug-in issue, and your Flash player needs to be updated.

5. After the Flash issue is rectified by updating your Flash software, the project appears ready to be run (Figure 4.15).
4.2 Scratch Studio

A studio (Figure 4.16) is a place where users can put multiple projects into an accessible group. Prior to Scratch 2.0, they were called galleries and had some different features.

A studio can be created by going to My Stuff and clicking “+ New Studio”. A new studio will be created, with all of the default settings: name is “Untitled Studio”; 0 projects, comments, followers, and a default thumbnail; a blank description; and the creator as the only manager. After this, these settings can be changed.

6 Source: https://wiki.scratch.mit.edu/wiki/Studio
The creator of the studio can edit it in many ways, and other users can make changes as well. The creator can add or remove curators, add or remove projects, choose the way projects are sorted, and change the description (Figure 4.17), thumbnail and title (Figure 4.18). Everyone can comment on or follow the studio. Curators can also add or remove projects (but not ones submitted by other curators). The creator may enable or disable comments, but regular curators cannot.

![Figure 4.17 Fill or edit the description of the studio](image)

![Figure 4.18 How to change the title of a studio](image)

You can change the studio image by pressing “change picture” in the top left corner of the current image and then selecting the image you want from your files. Scratch will automatically resize the picture so it will fit in the thumbnail.

You can also change a studio’s thumbnail to a .gif file, which is an animated picture. Unlike animated project thumbnails, animated studio thumbnails do not make the front page lag, since they do not use a user-generated hack.
The default thumbnail if no projects are in the studio will be a grey silhouette of Scratch Cat’s head in a grey circle with a grey background; otherwise, it will be a thumbnail of the most recent project added. Before 2.0, the default thumbnail was a cube resembling the Rubik’s Cube.

Curators
Studios trump galleries with the curators feature. This lets the creator decide exactly who can add and remove projects. The creator can invite people they are following and who are following them, or they can add curators by username. Curators (Figure 4.19) do not have all of the permissions that managers have, but they can still add and remove projects that they added themselves. The creator or manager of the studio can delete curators or other managers by clicking the x in the top right corner of the user in the curators tab. Managers cannot delete the creator of the studio, however. Galleries did have a “some of my friends” feature, but this meant that you had to friend the users whom you wanted to have curate. In Scratch 2.0, friending is renamed following.

Managers
Managers have greater permissions in studio capabilities. They can add or remove any project in the studio and can even remove other curators or managers. They are also able to edit the title, notes or icon of a studio and choose the way projects in the studio are sorted. Managers can also invite curators to the studio. To promote a curator to a manager, hover over their username and click the “Promote” button that appears. This can only be undone by removing the manager from the studio and then re-inviting them. The creator of the studio is the “original manager” (also referred to as the owner) and cannot be removed. He or she has the most control over the activity. It is recommended that you only invite and promote curators whom you trust, to avoid blatant damage to the studio (such as removing projects, changing information and title, and demoting managers).

Figure 4.19 View of the curators menu in a studio
Adding Projects
There are two ways to add projects to a studio. One method allows multiple projects in a single studio; the other allows multiple studios for a single project.

You are allowed to add as many projects as you like to a studio if the owner lets you, or if you are the owner.

From a Studio
When a user is a studio curator, they can add projects by clicking the “Add projects” button (Figure 4.20) or the bar on the bottom (Figure 4.21). Both open the bar. It can show My Shared Projects, My Favorites and My Recent Views. To add a project, click its thumbnail.

Clicking the link below it will open the project. Projects can also be added by URL (see the arrow in Figure 4.20). To do this, copy (ctrl + c) the link of the project in the search bar, then paste it (ctrl + v) in the bar beside the button that says “Add by url”. Once you’ve done that, click “Add by url.”

From a Project
From a project (Figure 4.22), you can click “Studios,” which will open a section allowing you to add the project to studios that you curate. It lists the studios, showing their names. Clicking adds a project to the studio.

To add a project to a studio, the studio must already be in existence. When you click the panel on a project page, a field shows below and lists the studio to which you can add a project (Figure 4.23). If there is no studio, nothing will be in the studio field.
Removing Projects
If you’re an owner or manager of a studio, you can remove any project in the studio. Curators who are not the manager or owner can only remove projects they add themselves. Even if you’re not curating a studio, you can still remove your own projects from others’ studios.

Removing Your Own Project
To remove your own project from your or someone else’s studio, on the project statistics row, access the Studios tab. When clicked, a dropdown menu will appear, consisting of all the studios the project is associated with. On the right-hand side of the dropdown menu are check marks: green ones mean the project is in the corresponding studio, and grey checks mean the project has not yet been added to that studio. Grey check marks will only appear next to studios you curate, because otherwise almost all studios on the website would have to be listed. To remove the project from the desired studio, click the green check mark next to the corresponding studio; the check mark will then turn grey, signifying it has been removed.

Display in Projects
Underneath the right-hand side of a project (Figure 4.24) are displayed the studios it features in and, in brackets, the number of studios to which the project has been added. If there are remixes of the project, they will be displayed underneath the remixes.
Differences from Galleries
Possibly the biggest difference from galleries is the curator feature, which allows the owner or other curators to invite curators. There are also numerous small differences, such as splitting into tabs, and the absence of tags.

The name was changed because galleries were being used for many things other than simply showing projects. Examples include collaborations and RPGs. The Scratch Team decided a more general name would be better.

Places to Find Studios
Studios can be found all over the website. There are two Front Page sections: Featured Studios, and Projects in Studios I’m Following. Profile pages also list studios the user curates, in the fourth row from the top, and studios the user follows, in the third. If a project belongs to a studio, the studio name will show up in the Studios tab, which is located right under the Remixes tab.

Types of Studios
- Add Everything studios
- Themed studios
- Scratch Design studios

Figure 4.24 Display of studios for and remixes of a project
• Hour of Code studios
• RPGs studios
• Series studios
• Announcement studios
• Specific studios
• Save (username) studios

Following a Studio
Following a studio will make it appear on your profile page in the “Studios I’m Following” box. To do this you simply select the studio you wish to follow and click the “follow” button. You will also get notifications if something happens.

Deleting a Studio
To delete a studio, go to your My Stuff page (make sure that you’re logged in, or it won’t appear). Select the Studios tab and search for the studio that you want to delete. When you find it, click “Delete” next to the studio. You can delete only your own studios.

4.3 Exporting and Importing Scratch Files

Exporting and importing are the processes of extracting a particular file and implementing it back into a program. In Scratch, these processes deal with the transferring of projects, sprites, costumes, backdrops, sounds, scripts and lists.

Exporting Processes
A variety of objects within the Scratch program can be exported within Scratch. Often, you need to save the file to your computer.

Projects
To export a Scratch project, access File > Download on your computer. Downloading to the computer is the same as exporting the project. You will need to save it to your computer (or external hard drive) as a .sb2 file in Scratch 2.0 or a .sb file in Scratch 1.4.

Sprites
To export a sprite, right-click (or shift-click) on the sprite’s icon in the sprites panel and select the option “Save to local file.” As with the projects themselves, sprites will then be saved in .sprite2 format.

Costumes
To export a particular costume, right-click on the costume’s icon in the costumes pane and select “Save to local file.” If the costume is in the vector format, it will save as a .svg file, but bitmap costumes save in various formats.

Backdrops
To export a particular backdrop, right-click on its icon in the backdrops pane and select the “Save to local file,” which allows you to save the backdrop to your computer.

7 Source: https://wiki.scratch.mit.edu/wiki/Importing_and_Exporting
Sounds
To export a particular sound, right-click on its icon in the sounds pane and select the option “Save to local file,” which will save the sound in the same format to your computer.

Scripts
Without exporting the entire sprite file, scripts can only be exported using the backpack. To export a script, open the backpack and drag the script into it. The script is then saved on the Scratch server.

Lists
To export a list, right-click on the list monitor and select export, which allows the list to be saved as a .txt file.

Importing Processes
Importing is the process of transferring external files into a program.

Projects
To import a project into the project editor, access File > Upload on your computer. Once a project is selected (Scratch 2.0 reads both .sb and .sb2 files), it will be loaded into Scratch.

Sprites
To import a sprite, click on the Import button underneath the stage and select the sprite file (.sprite or .sprite2) or supported image file to import.

Costumes
To import a costume, in the paint editor select the Import button and choose the desired costume file to upload.

Backdrops
To import a backdrop, in the paint editor select the Import button and choose the desired backdrop file to upload.

Sounds
To import a sound, in the sound editor, click the Import button and select the sound to import. Scratch can only read .MP3 and .WAV files.

Scripts
To import a script, drag the desired script within the backpack into the scripts area of the project editor.

Lists
To import a list, right-click on an existing list and select the import option. Then, select any .txt file to import into the list. Each line in the .txt file represents each item in the list.
4.4 Supported Files

Scratch File Formats
There are two file formats in Scratch, based on the two major releases: Scratch 1.4 and Scratch 2.0. Each release has a specific file format; Scratch 1.4 saves with .sb whilst Scratch 2.0 saves with .sb2.

The Scratch 1.4 file format is the format with which a Scratch file .sb (Scratch Project) or .sprite (exported sprite) file is encoded. Scratch needs to save all the information and media — stage, sprites, scripts, sounds and images — in a Scratch project or script to a single file.

The Scratch 2.0 file format is the file format used to encode Scratch 2.0 projects when they are downloaded to a user’s computer. Unlike the binary Scratch 1.4 file format, the 2.0 format comprises a .ZIP archive containing project information encoded in a text-based format called .JSON; project media are in separate files. Projects conventionally have the extension .sb2 and sprites the extension .sprite2.

Other File Formats

Sound File Format
A sound file format is a file format for storing audio on a computer. Scratch can only read .MP3 and .WAV sound files.

Scratch is limited to only reading .MP3 and .WAV files because part of the data that makes up the Scratch program consists of the ability to read and convert to those files. If Scratch were to have conversions and readings for every type of file, the data size of the Scratch program would increase, and therefore the online editor would take slightly longer to load. Furthermore, .MP3 and .Wav files are compressed, keeping data space within the 50MB file size limit of a Scratch project.

---

8 Source: https://wiki.scratch.mit.edu/wiki/File_Format
5.0 The Scratch Community

5.1 About the Community

Community emerges when a group of people are joined in a mutual task or achievement of a goal. In the case of Scratch, which is a desktop application, the Scratch online community was conceived to allow anyone, especially young people between eight and 16, to share their animated stories, interactive art, and video games. Scratchers use the development environment to create or remix projects by putting together images, music and sounds with visual programming blocks that control the items’ behaviour. Scratch and its website enable young people to express themselves creatively, gain new media literacy skills, and learn core computational thinking concepts, all in a community of peers.

The Scratch website offers an alternate model for how children might use the Web as a platform for learning, enabling them to create and share personally meaningful projects rather than merely accessing information. Children create and share Scratch projects to express themselves creatively, much as they would paint a picture or build a castle with LEGO bricks. In the process, they not only learn important mathematical and computer-science concepts, but they also develop important learning skills: creative thinking, effective communication, critical analysis, systematic experimentation, iterative design and continual learning. The ability to produce (not merely interact with) interactive content is believed to be a key ingredient for achieving digital literacy and becoming a full participant in the interactive online world.

Over the years, the objective of the community has richly extended the use of Scratch across the globe. Live statistics at the end of 2016 of activities in the online community, shown with the data, charts and images below (Figures 5.1–5.5) express the ability of the community paradigm to empower learning.

![Scratch online community statistics 2016](https://scratch.mit.edu/statistics/)

<table>
<thead>
<tr>
<th>19,156,254 projects shared,</th>
<th>15,689,527 users registered,</th>
</tr>
</thead>
<tbody>
<tr>
<td>100,180,616 comments posted,</td>
<td>3,121,340 studios created</td>
</tr>
</tbody>
</table>

Figure 5.1 Scratch online community statistics 2016
(source: https://scratch.mit.edu/statistics/)

---

Figure 5.2 Scratch online monthly project shares  
(source: https://scratch.mit.edu/statistics/)

Figure 5.3 Scratch online age distribution of new Scratchers  
(source: https://scratch.mit.edu/statistics/)
Figure 5.4 Scratch online monthly active users
(source: https://scratch.mit.edu/statistics/)

Figure 5.5 Scratch online monthly activity trends
(source: https://scratch.mit.edu/statistics/)
The initial community has also expanded and extended further into other communities and projects. Examples of these are ScratchEd (http://scratched.gse.harvard.edu), a community of educators using Scratch, and ScratchJr (http://www.scratchjr.org) a mobile-based app for children aged five to seven.

We include such information in this toolkit to expand the reader’s view of the tremendous opportunity that exists in extending the Scratch community personally and/or with your group, class, etc. by hosting your projects on the community. The concepts of sharing, remixing, reusing, etc. are fundamental to learning because learning in a community is more convenient, as well as more rewarding and engaging. Also, the new addition of a class perspective with the creation of teacher accounts will foster community more quickly, with teachers supplying effective guidance for learners to use Scratch in tandem with their curriculum.

5.2 Community Concept

The simple introduction to Scratch reads: “Scratch is a programming language that makes it easy to create interactive art, stories, simulations, and games — and share those creations online.” Contained in the statement are two key elements of community: creating and sharing. Embedded in the two words are concepts such as co-operation, collaboration, remixing, reusing, etc. In previous chapters, we have talked about sharing projects on the Scratch website. This section will extend the sharing of information using just one of the fundamental concepts: remixing. The initiator of the Scratch online community had this primarily in mind and based a whole thesis on the remixing paradigm and how the Scratch website was designed using this community concept. At the end of this section, we will rehearse the process of remixing in Scratch.

Today, the term remixing refers to the act of creating something new based on existing materials. Remaking has generally been known to represent the concept of digital reuse. Even before digital technology started to be used for creative expression, remixing described the mechanical process of combining physical source materials and was often used in fields such as agriculture, chemistry and manufacturing. Creative activities, based on the idea of building new things by combining existing ones, are not new. For a long time, artists have engaged in similar practices through appropriation art, pastiche, collage, sampling and bricolage.

Monroy-Hernández proposed a taxonomy for categorising remixing, based on two dimensions: originality and generativity. Let us examine the dimensions and practices of remixing.

Originality

This can be defined as a way to indicate how different a remix is from its source without drawing conclusions about the value of the remix. Thinking of remixing along the originality axis helps us categorise derivative works by how transformative they are. It helps us distinguish remixes that are merely inspired by someone’s idea from those that did some tweaking of existing works and those that are replicas of previous creations. We outline three practices: inspirational, incremental and component-based.

a. Inspirational: It is often the case that people are simply inspired by the ideas in someone else’s work. In Scratch, this happens when people browse the website and find projects that motivate them to create similar ones.
b. Incremental: Incremental remixes consist of making tweaks or adding something extra to a project. This type of remixing often involves downloading someone’s project to customise it or to fix a bug.

c. Component-based: Remixing also occurs when people use pieces of others’ projects to produce something new, rather than building on top of existing work.

**Generativity**

Another way of categorising remixes is based on their intended or accomplished generativity — in other words, the number of derivative works that a particular source material engenders, or tries to engender.

a. Crowd-based: Crowd remixes are often initiated when a creator explicitly invites others to remix.

b. Group-based: Group-based remixing tends to involve several back-and-forth interactions through remixing. The process has been referred to as “ping pong” remixing.

c. Versioning: This is when remixes are used as some form of version control.

We can derive the following from the dimensions explained above:

- Everyone has remixed at one point or the other via doing a project.
- Remixing is an established practice that is highly beneficial for learning.
- It is essential for the life of a community to encourage remixing.
- The Scratch project keeps thriving on the capacity to remix.
- Remix is an encompassing concept that involves other community concepts such as creation, sharing, collaboration, co-operation, reuse and revision.

Now let’s rehearse the remix concept in Scratch.

1. To remix in Scratch, you must have an account, which is simply achieved by going to the Scratch website and clicking on **Join Scratch**. Fill out the form and confirm with an email account.

2. You have access to check various projects on Scratch. Go to a project, open it and view it. To see the Scripts that produced the project, click **See inside** (Figure 5.6).
3. The action takes you to the project editor view (Figure 5.7), where you have access to the script and you can start to make additions. You will see the Remix icon on the right-hand side of the page. When you place your mouse over it, a dialogue box is displayed, showing “Save a copy of this project and add your own ideas.” That is indicative of what can be done with the icon. On clicking Remix, the process begins.

4. Alternatively, a question tag appears over the Remix icon (Figure 5.8). On clicking the question starts the remix process.

5. A dialogue box opens (Figure 5.9), which gives further details about the remixing process. When you click OK, Got it, it goes into the remix stage automatically by saving in your account a copy of the project you want to remix.
6. Once you finish amending the project and click save or share, it goes into the front panel (Figure 5.10) of a project for additional information on the project. Instructions, credit and attribution can be added at this stage. One of the things that you automatically see is that it has given a credit to the original project you remixed.

Figure 5.9 Dialogue box showing additional details about remixing

Figure 5.10 The project front panel for instructions, credits and attributions
7. On the part for the remixed project is an indicator showing how many projects have taken off from it. This is called the remix tree. A tree icon is shown under the individual project page (Figure 5.11).

![Figure 5.11 The tree icon on the project page, showing the number of remixes](image)

**NOTE:** The number of remixes is the total of the original project and other remixes. So, when it is showing 3, this means 2 other projects have been remixed from it.

8. The remix tree page arranges itself in a branch based on the number of remixes. Figure 5.12 shows the arrangement of the Hair Salon tutorial, which has 45 remixes.

![Figure 5.12 A remix tree, with the original project in front and the remixes branched](image)

9. On the project page of a remixed project, the words “remixed by” are shown under the name of the remixed project, indicating the user who remixed the project (Figure 5.13).
10. A list of the projects that have remixed the original project appears on the original and remixed projects (Figure 5.14). If there are no remixes, 1 is on the tree and no project appears in the remixes area.

5.3  Scratch User Ranks/Accounts

The Scratch community has a couple of user ranks and accounts based on the user’s experience, activities and contributions to the community. The teacher account also has different registration requirements from all other types of users. So, the accounts can be based on ranks in the Scratch website or on registration mode. By rank, Scratchers can be: a New Scratcher; a Scratch Team
Member; or a Scratch Mentor. By registration, there are two types: the general (basic) registration mode and the teacher account registration mode.

The various ranks determine the level of access a Scratcher has on the Scratch website. A Scratcher can move from the basic user level to a higher level based on contribution and other criteria set out by the Scratch Team. The registration modes present the type of access a user has on the website. Below we will outline the two registration modes that are available at the time of writing this toolkit.

**Basic Registration**

1. On the website, you will see **Join Scratch** at several points (Figure 5.15). The tag is also available on the project editor when you are online (Figure 4.16). Clicking it takes you to the Scratch registration form.

![Figure 5.15 Click Join Scratch on the website](image)

**Figure 5.15 Click Join Scratch on the website**

![Figure 5.16 Click Join Scratch on the project editor](image)

**Figure 5.16 Click Join Scratch on the project editor**

2. Clicking Join Scratch directs you to the form (Figure 5.17), where you will fill in the Scratch username and password, which are user generated. After both have been completed, you can click Next to go to the second page of the form.

**IMPORTANT:** A user must supply a unique Scratch name for it to be accepted. The program auto-validates the name before you can move to the next page. You can use a unique nickname instead of your real name whilst setting up a Scratch account.
3. On the second page of the form (Figure 5.18), the following biodata are requested: birth month and year; gender; and country. These are needed to fill the user’s data store on the website and for statistics and research purposes. Click Next to move to the next page.

4. The third page (Figure 5.19) asks the registrant for an email address in order to confirm the user. A registrant under 13 years of age will be asked to supply the email of the parent/guardian, who will then confirm the user’s right to use it. An email is sent to the email address supplied to confirm the account. Afterwards, click Next to go to the next page.
5. The form is completed and a welcome message is shown on the next page (Figure 5.20). This sets the user to automatic login. All of a user’s activities can be carried out, including creating a project, writing scripts and saving the project, except sharing, until the message sent to the user’s email is dealt with — i.e., the account is confirmed.

**Figure 5.19** Request for a functional email address to confirm the account

**Figure 5.20** The welcome page after the form has been completed

**Becoming a Scratcher**

When a user joins scratch, he/she has almost full access to post, share and remix projects on the Scratch website. However, access is not granted for everything automatically. Being acknowledged
as a **Scratcher** (receiving this status) is one example. The attributes or additional benefits of a Scratcher include access to cloud variables on the project editor.

To become a Scratcher, you have to be active on the Scratch online community after registration, by doing things such as posting, sharing, commenting and remixing. When the activities have been acknowledged, a message is sent to the account to complete other steps in becoming a Scratcher. The steps are detailed below in Figures 5.21a–g.

![Figure 5.21a Becoming a Scratcher](image1)

![Figure 5.21b Becoming a Scratcher](image2)
Figure 5.21c Becoming a Scratcher

Figure 5.21d Becoming a Scratcher
Figure 5.21e Becoming a Scratcher

Figure 5.21f Becoming a Scratcher
Teacher Account Registration

1. The teacher account is for educators who are using the Scratch platform with their class. Several affordances are provided within the framework of teacher and student accounts. Some of the details will be available to you after the registration processes are completed.
2. To register and own a teacher account on Scratch, go to the dedicated page for educators here [https://scratch.mit.edu/educators/](https://scratch.mit.edu/educators/) (Figure 5.22) and click on Teacher Accounts or simply use this link [https://scratch.mit.edu/educators/#teacher-accounts](https://scratch.mit.edu/educators/#teacher-accounts) and click on Request Account (Figure 5.23).
3. When you click “Request Account,” a form (Figure 5.24) opens for you to create a username and password, which are user defined. After completing that, click Next Step to move to page 2.

**IMPORTANT:** A footnote states that the approval process may take up to 24 hours. That means you should not expect to use the account immediately after it was created.
4. On the second page of the form (Figure 5.25), you will be asked to supply your personal information. Again, as it states, this information won’t be made public. It is meant for verification and research purposes. When you have given the appropriate information, go to the next page by clicking Next Step.

5. On the next page of the form (Figure 5.26), you are asked for your first and last names. These won’t be made public.
6. This page (Figure 5.27) requests the educator’s phone number. This might be verified by the Scratch Team, so you must provide a phone number.
7. After clicking Next Step, you will be required to supply the name of your organisation, your role, and the category of educational or education-related organisation it is (Figure 5.28). These are meant to verify the validity of the registrant. A field for the website of your organisation is available but optional.

Figure 5.28 Teacher account form page 5, your organisation information

8. On the next page (Figure 5.29), you are requested to provide your address details. Since the country field has been supplied initially, it gets auto-filled. You can then provide the other address components to complete the page before clicking Next Step.
9. The next page (Figure 5.30) requires you to write an essay about how you intend to use Scratch. The maximum character count is 300, so it is meant to be concise and clear on your anticipated goal after obtaining the account.
10. Getting to the final page of the form (Figure 5.31), you are requested to supply your email address twice for validation and confirmation of the account. This will facilitate your receipt of further information to access the account.
11. A thank you page and additional instructions about the account you just created is shown (Figure 5.32) after you click Next Step.
12. You can still log in with the user details you provided after confirmation via the email sent to the email account you provided during registration. However, you can’t have access to the features of the teacher account until it is approved and upgraded.

Features of the Teacher Account
We will now consider some of the specific features of the teacher account.

1. After the teacher account has been approved, you will have the dashboard shown in Figure 5.33. Consider exploring the very important menus — My Classes, Educator Resources and Teacher Account FAQ. The Educator Resources and Teacher Account FAQ are generally available without a teacher account. My Classes is the only teacher account-specific menu.

![Figure 5.33 Front view of a teacher account](image)

2. The menu tab available beside the username in the top-right-hand corner of the page (Figure 5.34) gives a quick selection of tabs and activities that are unique to a teacher account on Scratch. You have the profile, My Stuff, My Classes, Account settings and Sign out tabs.

**NOTE:** A teacher account is an upgrade, and that means that it has all the right and privileges of a basic scratch account. My Stuff is generally available for all Scratch accounts, and the account holders can make contributions as members of the community.

![Figure 5.34 A menu tab beside the username has a quick list of actions](image)

3. Clicking My Classes takes you to a page (Figure 5.35) where you have access to various toolkits available for the teacher account. On the top is the title and a tab for creating a
new class. Every class created appears in the middle of the page. On the left-hand side is a list that includes All Classes (active classes on the account), Ended Classes (closed classes on the account) and All Class Alerts.

![My Classes dashboard, which shows all classes created on a teacher account](image)

**Figure 5.35** My Classes dashboard, which shows all classes created on a teacher account

4. To add a new class, click on the My Classes tab, and you’ll see a dialogue box (Figure 5.36). Put in the Class Name and Class Description for the class you are creating.

![Create a new class](image)

**Figure 5.36** Create a new class

5. When the details have been supplied and added, a page appears (Figure 5.37) showing a series of actions that take place in a class. There are four menus on the page: Settings, Students, Studios and Activity. By default, the Settings menu is active and shows: the title of the course; **About this Class**, from the Class description; and **What the Class is working on**, which can be filled in directly from the Settings menu.
6. There is a picture thumbnail (Figure 5.38), which you can change to a customised picture for the class. This will enable your students to differentiate between your classes at a glance.

![Figure 5.38 The picture thumbnail](image)

7. The next menu is Students (Figure 5.39), which is used to add students to the Class. Effectively, there are three ways a student can be added to a Class: the New Student tab (to add a single student per time); and two ways of adding multiple students to a class — the Student Sign-up Link and CSV Upload. Let’s see how these work.

![Figure 5.39 The Students menu for adding students to your Class](image)
8. When you click New Student under Add one student, you receive a pop-up (Figure 5.40), which gives two fields to select a class: Add to Class and Username. After you input the required information, click on Add Student to save the request.

![Figure 5.40 Adding a new student to the Class](image)

**IMPORTANT:** Consider the information under the fields — these accounts will be publicly visible. Please do not choose usernames that contain personal identifiers, such as real names or schools. A chosen name will be verified against all of the identities already stored on Scratch, so consider establishing a unique name or naming style for your class.

9. After you have completed the process for adding a name, the new user appears under the Students menu in the Class (Figure 5.41).

![Figure 5.41 The Students menu automatically shows the list of students added to a Class](image)
10. If you already have a list of students in your class, you can add in bulk by clicking on the CSV Upload (Figure 5.42) and uploading a prepared file through the pop-up menu.

![Figure 5.42 Upload CSV to add multiple students to a Class](image)

**NOTE:** Adhere to the rules stated or the file will not upload (Figure 5.43). You can use the example link to prepare your own CSV file.

![Figure 5.43 Error message when adding by uploading using CSV](image)
11. The third way to add students to a Class in a teacher account is the Sign-up Link (Figure 5.44). When you click the Generate tab, a link is generated (Figure 5.45), which you can share with your class for them to independently add themselves to the class. Anyone who has the link can automatically add themselves to the class.

![Figure 5.44](image1.png)
**Figure 5.44** The Sign-up Link option to add students to a class

![Figure 5.45](image2.png)
**Figure 5.45** A sample of what you will see when you click Generate

12. The Studios menu is next on the Class page (Figure 5.46). The studio acts as folder for projects in the class. Each student can have their own studio; each assignment can have a studio and more, depending on the decision of the teacher. Click on New Class Studio to add studios to a class.

![Figure 5.46](image3.png)
**Figure 5.46** The Studios menu in a Class
13. A pop-up appears, with the fields to create a studio (Figure 5.47). Input the studio name and description and save with the Add Class Studio tab.

![Figure 5.47 Add a studio to a Class](Image)

14. The fourth menu is the Activity menu for the Class (Figure 5.48). This is by design an information menu, and you view information at a glance with the two tabs Activity for and Sort by. Each is a menu list that in turn provides a dropdown list you can select from and then view the activity therein. This may be a selected student’s activity in a class.

![Figure 5.48 The Activity menu on the Class page](Image)

15. You can have a single view page (Figure 5.49) if you want all four menus in a Class to appear concurrently, with their various details.
16. Clicking a student thumbnail opens a page (Figure 5.50) that is similar to a Scratch's page view, with some additions, such as the students in a class who are set to follow you by default. Other information about each student is also shown, including About me, what the student is working on, shared projects and favourite project. The student can change the thumbnail and input his/her own.
The ScratchEd Community

ScratchEd (Figure 5.51) is an online community where Scratch educators share stories, exchange resources, ask questions, find people and discover events. Established in 2009, the community was a response to the growing need for educators to share information on the use of Scratch. According to the website, this was necessitated because of the wide variety of educators supporting Scratch creators, in both formal and informal learning environments: a teacher who wants to share stories about Scratch and cross-curricular integration; a researcher who wants feedback on materials developed to explore Scratch for participatory literacy; a parent who wants advice on how to introduce Scratch at a local all-girls high school; a museum programme director who wants to connect with other museums that have introduced Scratch. The list is endless.

This online community can be accessed at http://scratched.gse.harvard.edu. A user can join by registering at http://scratched.gse.harvard.edu/user/register. This is a highly moderated site.

---

10 Source: http://scratched.gse.harvard.edu/help

Figure 5.50 The page view of a student in a class
Registrants should share the goals of the community, and you must be an educator/teacher to be admitted as a member.

The information on the website, however, is freely available to the public.

![The ScratchEd online community home page](image)

**Figure 5.51** The ScratchEd online community home page

**Scratch Jr**

Whilst the main Scratch concept covers the age range of eight to 16, other community members have found various means to expand the capacity of coding to reach learners younger than eight. One such project is ScratchJr (Figure 5.52). With ScratchJr, young children (ages five to seven) can program their own interactive stories and games. In the process, they learn to solve problems, design projects and express themselves creatively on the computer. More information is available on the ScratchJr website: [http://www.scratchjr.org](http://www.scratchjr.org).

---

11 Source: [http://www.scratchjr.org/about.html#info](http://www.scratchjr.org/about.html#info)
Figure 5.52 The ScratchJr website
6.0 Workshop for Scratch

6.1 Workshop Training Schedule

The following is the workshop training schedule to be used when conducting a training workshop. Each session corresponds to the activities and tutorials in this toolkit.

<table>
<thead>
<tr>
<th>TITLE</th>
<th>WORKSHOP ON REUSABLE LEARNING OBJECT DEVELOPMENT TOOLKIT FOR TEACHERS USING SCRATCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATES</td>
<td>dd to dd month year</td>
</tr>
<tr>
<td>VENUE</td>
<td></td>
</tr>
<tr>
<td>TRAINER</td>
<td></td>
</tr>
<tr>
<td>PREREQUISITES</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME</th>
<th>GETTING STARTED WITH MIT SCRATCH Day 1 (dd/mm/yyyy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.30 - 8.45am</td>
<td>Registration</td>
</tr>
<tr>
<td>9.15 - 9.30am</td>
<td>Icebreaking</td>
</tr>
<tr>
<td>9.30 - 9.45am</td>
<td>Objective and expected outcome of the workshop</td>
</tr>
<tr>
<td>9.45 - 10.00am</td>
<td>Learning outcomes of Day 1</td>
</tr>
<tr>
<td>10.00 - 10.15am</td>
<td>Introduction to Visual Programming and MIT Scratch</td>
</tr>
<tr>
<td>10.15 - 10.30am</td>
<td>Tea break</td>
</tr>
<tr>
<td>10.30 - 11.30</td>
<td>Becoming a Scratcher (setting up an account in Scratch)</td>
</tr>
<tr>
<td>11.45 - 12.15pm</td>
<td>Developing and Debugging in Scratch</td>
</tr>
<tr>
<td>12.15 - 1.00pm</td>
<td>Introduction to the Scratch Development Environment</td>
</tr>
<tr>
<td>1.00 - 2.00pm</td>
<td>Hello Scratch</td>
</tr>
<tr>
<td>2.00 - 3.00pm</td>
<td>Lunch Break</td>
</tr>
<tr>
<td>3.00 - 3.15pm</td>
<td>Hello Scratch</td>
</tr>
<tr>
<td>3.15 - 3.45pm</td>
<td>Tea break</td>
</tr>
<tr>
<td>3.45 - 4.00pm</td>
<td>Recap and wrap-up</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME</th>
<th>INTERMEDIATE CONCEPTS Day 2 (dd/mm/yyyy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.00 - 9.15am</td>
<td>Learning outcomes of Day 2</td>
</tr>
<tr>
<td>9.15 - 10.00am</td>
<td>Tutorial 1</td>
</tr>
<tr>
<td>10.00 - 10.15am</td>
<td>Tea break</td>
</tr>
<tr>
<td>TIME</td>
<td>ADVANCED CONCEPTS</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td></td>
<td><strong>Day 3 (dd/mm/yyyy)</strong></td>
</tr>
<tr>
<td>10.15 - 11.15am</td>
<td>Tutorial 2</td>
</tr>
<tr>
<td>11.15 - 12.45pm</td>
<td>Tutorial 3</td>
</tr>
<tr>
<td>12.45 - 1.00pm</td>
<td>Q&amp;A Session</td>
</tr>
<tr>
<td>1.00 - 2.00pm</td>
<td>Lunch break</td>
</tr>
<tr>
<td>2.00 - 3.00pm</td>
<td>Tutorial 4</td>
</tr>
<tr>
<td>3.00 - 3.15pm</td>
<td>Tea break</td>
</tr>
<tr>
<td>3.15 - 3.45pm</td>
<td>Tutorial 4</td>
</tr>
<tr>
<td>3.45 - 4.00pm</td>
<td>Recap and wrap-up</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME</th>
<th>ADVANCED CONCEPTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Day 3 (dd/mm/yyyy)</strong></td>
</tr>
<tr>
<td>9.00 - 9.15am</td>
<td>Learning outcomes of Day 2</td>
</tr>
<tr>
<td>9.15 - 10.00am</td>
<td>Creating a Teacher Account</td>
</tr>
<tr>
<td>10.00 - 10.15am</td>
<td>Tea break</td>
</tr>
<tr>
<td>10.15 - 11.15am</td>
<td>Creating a Teacher Account</td>
</tr>
<tr>
<td>11.15 - 12.45pm</td>
<td>Tutorial 5</td>
</tr>
<tr>
<td>12.45 - 1.00pm</td>
<td>Packaging and Distribution</td>
</tr>
<tr>
<td>1.00 - 2.00pm</td>
<td>Lunch Break</td>
</tr>
<tr>
<td>2.00 - 3.00pm</td>
<td>Tutorial 6</td>
</tr>
<tr>
<td>3.00 - 3.15pm</td>
<td>Tea break</td>
</tr>
<tr>
<td>3.15 - 3.45pm</td>
<td>Sharing Your Project and More</td>
</tr>
<tr>
<td>3.45 - 4.00pm</td>
<td>Reflections and feedback</td>
</tr>
</tbody>
</table>
6.2 Workshop Evaluation

The following evaluation form can be used to gather qualitative and quantitative feedback on multiple aspects of the workshop. The evaluation form is to be administered at the last session of the workshop training schedule (see section 6.1). The form has been validated and field tested. It can be administered in hard copy or online format, depending on the circumstances.

REUSABLE LEARNING OBJECT DEVELOPMENT TOOLKIT FOR TEACHERS USING SCRATCH
Workshop Evaluation Form

As a participant, you are invited to evaluate this workshop. Please use this form to provide your feedback. Your personal information will be kept confidential at all times. However, your feedback might be disseminated as research findings.

<table>
<thead>
<tr>
<th>PERSONAL INFORMATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Age</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Gender</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Occupation</td>
</tr>
<tr>
<td>Institution/Organisation</td>
</tr>
<tr>
<td>Work Address</td>
</tr>
<tr>
<td>Email Address</td>
</tr>
</tbody>
</table>
### BACKGROUND INFORMATION

<table>
<thead>
<tr>
<th>Question</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Are you a teacher or a student? (choose only one)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Teacher</td>
<td>![Teacher]</td>
<td>![Student]</td>
</tr>
<tr>
<td>What’s your IT background? (choose only one)</td>
<td>![Teacher/Student (IT-related subject)]</td>
<td>![Teacher/Student (non-IT-related subject)]</td>
</tr>
<tr>
<td>What’s your area of specialisation? E.g., Education, Social Science, Mathematics, Computer Science, etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do you have any prior experience in computer programming?</td>
<td>![No]</td>
<td>![Yes]</td>
</tr>
<tr>
<td>If yes, what’s your level of expertise in computer programming?</td>
<td><img src="https://example.com" alt="I have learned programming in school, university, etc." /></td>
<td><img src="https://example.com" alt="I am a programmer by profession." /></td>
</tr>
<tr>
<td>Have you worked with Visual Programming before this workshop?</td>
<td>![No]</td>
<td>![Yes]</td>
</tr>
<tr>
<td>If yes, please describe your experience (e.g. platform, programming language, etc.).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Have you used Scratch before this workshop?</td>
<td>![No]</td>
<td>![Yes]</td>
</tr>
<tr>
<td>If yes, please describe your experience (e.g., what type of projects you developed, did you share on the Scratch online community, etc.).</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Feedback on the Workshop

The objectives of the workshop were to:

- Familiarise yourself with the Scratch platform
- Use the Scratch development environment
- Adapt sprites and backdrops to develop scripts
- Produce animations, stories and games with Scratch
- Share, remix and reuse objects in Scratch

What is your overall rating of this workshop? □ Poor □ 2 □ 3 □ 4 □ Excellent

How did you feel about the length of the workshop? □ Too short □ Just right □ Too long

How did you feel about the pacing of the workshop? □ Too slow □ Just right □ Too fast

Please indicate your agreement or disagreement with the following statements, using a scale of 1 to 5 where 1 = strongly disagree and 5 = strongly agree. Tick the appropriate box.

<table>
<thead>
<tr>
<th>Strongly Disagree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Strongly Agree</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>The objectives of the workshop have been achieved.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Your personal objectives for attending the workshop have been achieved.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Statement</td>
<td>Agree</td>
<td>Strongly Agree</td>
<td>Disagree</td>
<td>Strongly Disagree</td>
<td>Neutral</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------------------------------</td>
<td>-------</td>
<td>----------------</td>
<td>----------</td>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>Your understanding of the Scratch program has improved or increased as a result of the workshop.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Your skills in the Scratch program development have improved or increased as a result of the workshop.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>You would recommend to others that they attend this workshop in the future.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The material provided was necessary for the workshop.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>There was a good balance between lecture sessions, activities, tutorials and discussions.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The workshop was logically sequenced from basic to intermediate concepts.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The practical activities and tutorials were effective in teaching the concepts.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The amount of time given for the activities and tutorials was sufficient.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The amount of time given for the follow-up discussions was sufficient.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The handouts given were useful and of good quality.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Are there any other comments or feedback that you would like to share?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7.0 Useful Scratch Resources

7.1 Scratch Website

About Scratch
https://scratch.mit.edu/about
Discuss Scratch
https://scratch.mit.edu/discuss/
Scratch Help
https://scratch.mit.edu/help/
Scratch Teacher Account FAQ
https://scratch.mit.edu/educators/faq
Scratch Wiki
https://wiki.scratch.mit.edu

7.2 Scratch-Related Sites

ScratchEd
http://scratched.gse.harvard.edu
ScratchJr
http://www.scratchjr.org
ScratchX
http://scratchx.org/
Scratch3d
https://wiki.scratch.mit.edu/wiki/Three_Dimensional_Projects

7.3 Other Useful Sites

Icons
http://www.iconarchive.com/
http://makeappicon.com/
http://www.cartoonize.net/
http://cartoon.pho.to/

Images
https://thenounproject.com
http://www.youtodesign.com

Sounds
http://freemusicarchive.org/tag/instrumental/
http://soundbible.com