Chapter 1

Introduction

This initial module provides an orientation to the Snap*!* programming language. If you are already familiar with Snap!, you can skip ahead to the next chapter.

Snap*!* was developed at the University of California, Berkeley, and is used in computer science (CS) for non-CS majors at that university. Because of its widespread use, a user community has developed around this programming language. The capabilities of this language make it well-suited to explorations in art and music.

Topic 1.1 Securing a Snap! Account

A free Snap*!* account can be obtained from the University of California, Berkeley web site:

https://snap.berkeley.edu/

Note: At the time this is written, the blocks and music extensions described are available on the Snap! development site:

https://snap.berkeley.edu/versions/dev/snap.html

They will be incorporated into the main Snap! build after its is next update (from Ver 8.0 to 8.1).

Secure a Snap*!* account before continuing. The Snap*!* reference manual is available here:

https://Snap.berkeley.edu/Snap/help/SnapManual.pdf

Section II of the reference manual provides information about securing a Snap*!* account and saving projects. Review this section before continuing to the explorations that follow below.

Snap*!* help forums maintained at the University of California, Berkeley, are available here:

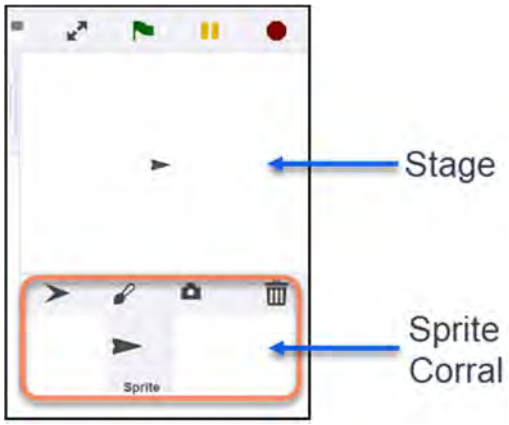
https://forum.Snap.berkeley.edu/

Assistance with questions that are not addressed in the reference manual can be obtained through the Snap*!* forum. In order to gain trusted status on the forum, you must read at least five different posts and spend at least fifteen minutes on the forum. This is to prevent spam posts by bots. For the same reason, if you are extremely fast typist, the forum software may assume that you are a bot.

Topic 1.2 The Snap! Workspace

In a typical Snap*!* session, blocks of code are dragged from the *Code Block Palette* on the left-hand side of the screen to a *Script Area* in the middle of the screen. Blocks of code are snapped together to create scripts. Clicking a group of code blocks causes the script to run, peforming an action. Often these actions involve movement of sprites on a *Stage* at the right-hand side of the screen.

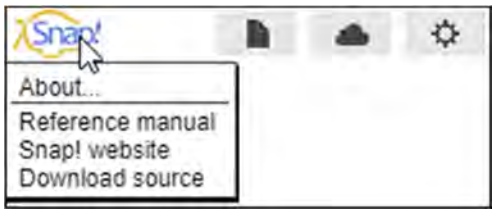
A *sprite corral* beneath the stage indicates which sprite is currently selected.



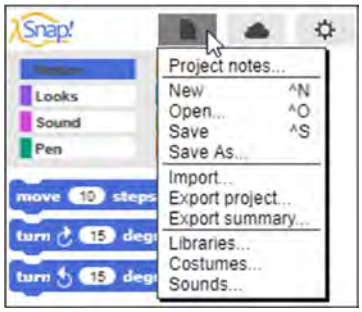
Each sprite has its own separate script space.

Topic 1.3 Snap*!* Menus

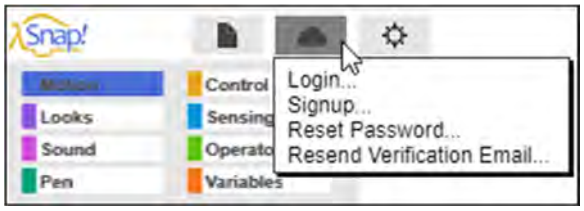
There are several menus that can be accessed in the top left-hand corner of the Snap*!* screen. The Snap*!* icon in the top left-hand corner can be used to access the Snap*!* reference manual.



The *File* menu to the right of the Snap*!* icon can be used to save projects and open projects that have been previously saved. This menu can also be used to access costumes for sprites, sounds, and libraries of additional Snap*!* code blocks.



The *Login* menu is to the right of the *File* menu.



A user must be logged into Snap*!* in order to save projects. Therefore it is a good idea to log in to Snap*!* at the beginning of every session.

A *Settings* menu is to the right of the *Login* menu. This menu provides a number of options for customizing Snap*!* For example, the *Zoom Blocks* option can be used to increase the size of the code blocks so that they can be more easily viewed in presentations.

Exploration 1.3

If you have not already, log into Snap*!* Explore some of the options available under the different menu settings.

Topic 1.4 The Code Block Palette

Scripts are created in Snap*!* by snapping blocks of code together (such as the **Move 10 Steps** code block in the illustration). The types of *code blocks* available are displayed in a *Code Block Palette* at the top left-hand side of the screen. For example, the *Motion* code blocks are currently highlighted in the palette below. Other categories of code blocks include *Looks*, *Sound*, *Pen*, *Control*, *Sensing*, *Operators*, and *Variables*. Each category is a different color (e.g.. *Motion* code blocks are blue). Click on the different categories (Motion, Looks, etc.) to access the code blocks associated with that category.

The *Motion* code blocks direct the movement of sprites (actors that can move about the stage on the right-hand side of the screen.) The *Looks* code blocks control the appearance of sprites. The *Sound* code blocks are used to play sounds. The *Pen* code blocks control the color and thickness of a pen that the sprite uses to draw lines on the stage. The *Control* code blocks provide control structures such as the **Repeat** command. The *Sensing* code blocks are used to sense the status of Snap*!* objects and monitor external inputs such as the keyboard and the microphone. The *Operators* code blocks provide mathematical and logical functions. The *Variables* palette is used to create and modify variables.

Exploration 1.4 The Code Block Palette

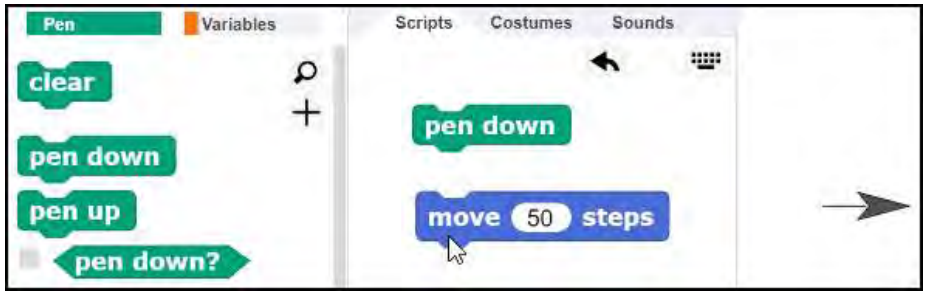
Click on each of the categories in the *Code Block Palette* to get a sense of the types of commands that are found under each category.

Topic 1.5 Motion and Pen Palettes

The different types of code blocks are arranged in palettes on the left-hand side. Each category of code block is a different color. The motion commands are blue code blocks.

If a motion block is moved into the script area in the middle of the screen and clicked, it moves a sprite (the black triangle on the right).

A group of commands in the *Pen Palette* control a pen that can be raised or lowered as the sprite moves.



Topic 1.6: Creating an Animated Biography

Snap*!* can be used as a scripted storytelling tool. In the following example, this capability will be used to create a short, animated biography. The example that follows, one of the authors describes her background as an orchestral percussionist and her interest in the game *Magic, the Gathering*.

Two videos describing the way in which this animated biography was created can be accessed here:

Part 1: Animated Biography – Video 1

Part 2: Animated Biography – Video 2

Once you have watched the video overview, open the Snap! biography program that is available here:

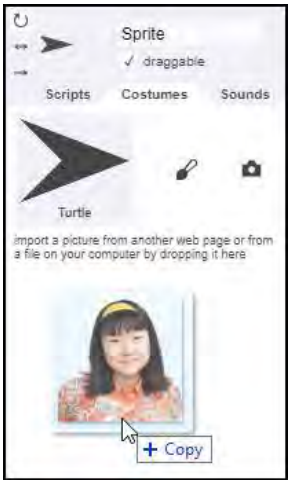
Snap! Biography Program

Note: Log into Snap! before accessing the link above. The Snap*!* file will enable you to inspect the code blocks in the program to see how the blocks were used to create the animated biography. Once you understand how everything works, create a similar animated biography for yourself.

The remainder of this chapter describes the process in greater detail. This will enable you to refer to a specific action or method if you need more information about the way in which a result was obtained.

Topic 1.7: Creating an Animated Biography

Begin by dragging a picture of yourself from the desktop into the *Costumes* tab.



The costume of the sprite will change from its default, a black triangle, to a thumbnail image with your picture. Change the name in the textbox beside the thumbnail image from the default name of "sprite" to your name.

The image of the sprite will also appear on the stage. Attributes such as size can be adjusted using commands in the *Looks* palette.

Topic 1.8: Controlling the Position of a Sprite

The default size of the stage is 480 steps wide by 360 steps wide, with the coordinate of [0 0] representing the center of the stage. The Go To block can be used to send the sprite to any part of the stage. The coordinates of [– 100 - 60], for example, would send the sprite to the lower left-hand corner of the stage.

Blocks such as Right of Stage and Top of Stage (found in the Sensing palette) can be used to send the sprite to a specific corner of the stage.

For example, this command would send the sprite to the top, right corner of the stage.

The command below would send the sprite beyond the boundaries of the stage.

Since several different sprites will be asked to go offstage, this block will be used frequently. For that reason, creation of a custom block named Go Offstage will document the reason that these coordinates are used.

Once the Go Offstage block is created, the following sequence would ask the sprite to start off stage and then glide to the lower left-hand corner of the stage.

The method described provides the foundation of a method for creating objects in Snap! and animating them to tell a story.

Topic 1.8: Creating Dialog

The **Say** command (found under the *Looks* palette) causes a speech bubble to appear beside the object.

The *Sounds* tab above the script area can be used to record speech using the microphone of the computer. (Note: the *Record* function does not work in the Safari web browser.)

Once speech has been recorded, the caption of the sound can be changed in the text box below the sound.

Putting it all together, the following sequence of blocks would cause the sprite to go offstage, glide to the lower left-hand corner, play the recording, "I'm Rachel" and place the same phrase in a speech bubble.

In this manner, a character can be created using a costume of a sprite and used to develop an animated biography.

Topic 1.9: Initiating the Sequence

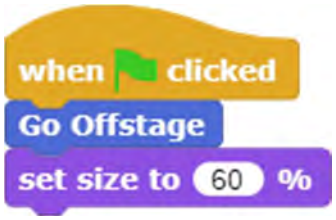
A *Green Hat* block (founded in the *Control* palette) can be used to initiate the sequence.

When the Green Hat block is added, clicking the green flag in the upper right-hand corner of the Snap! window will initiate the sequence of commands found under this block.

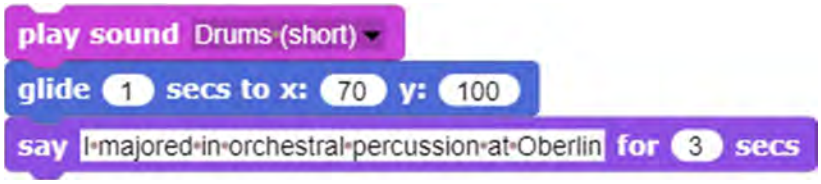
Additional sprites can be created to add other objects to illustrate the story. An additional sprite can be added by clicking the black triangle in the sprite corral (found just below the stage).

A new costume can be added to the second sprite by dragging an image from the desktop into the costume tab of the second sprite. In this instance, an image of percussion instruments has been added to the stage.

The following script in the script area of the second sprite (which has been named “Percussion” to reflect the image of percussion instruments) positions the sprite and adjusts its size when the green flag is clicked.



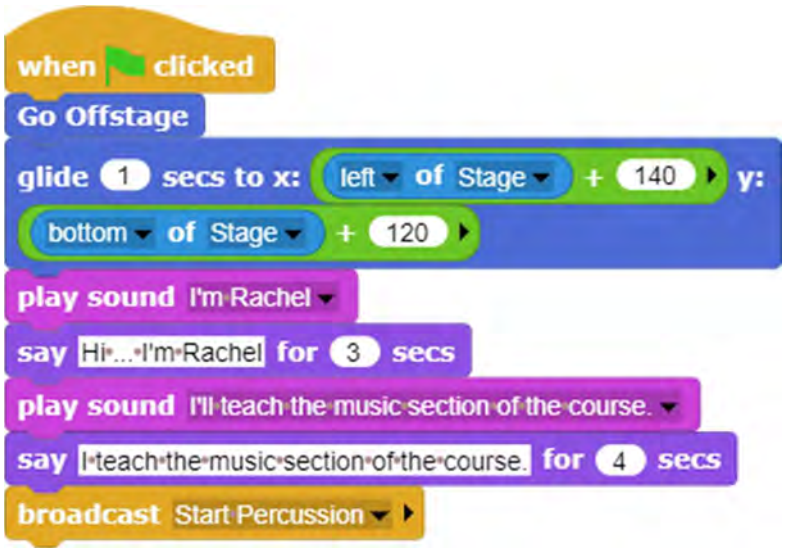
One of advantage of using the *Green Flag* to begin a program is that the scripts of several sprites can be initiated simultaneously, taking place in parallel. The script for the *Percussion* sprite employs a format similar to that used for the first sprite. It plays the sound of drums as the sprite glides to a point in the upper right-hand corner of the stage. A speech bubble then appears that says, “I majored in orchestral percussion at Oberlin.”



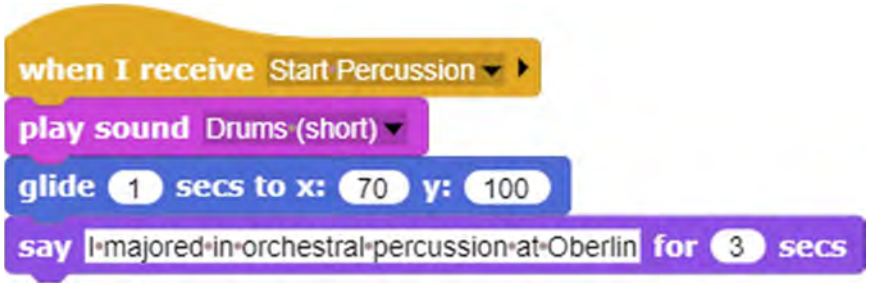
Topic 1.10. Completing the Animated Biography

The **Broadcast** block (found in the Control palette) enables the first sprite to send a message to the second sprite.

This enables the first sprite to broadcast the message "Start Percussion" to initiate the percussion sequence.



The **When I Receive [message]** block (found under the *Control* palette) can then be used to enable the second sprite to begin its sequence of commands when it receives the message "Start Percussion."



In this manner, sprites can send messages to one another to coordinate the order in which events occur.

Conclusion

This introduction to scripts and sprites in Snap*!* provides an orientation to a programming and development tool that will be used in the subsequent chapters. If you would like to know more about this tool, the Joy and Beauty of Computing is an introduction to general computing concepts developed at the University of California, Berkeley:

Beauty and Joy of Computing

In the next chapter, this tool will be used to introduce digital sound and sound sampling.