CPSC 301: Computing in the Life Sciences
Lecture Notes 2: Scratch (v2)

Jessica Dawson
jqdawson@cs.ubc.ca
http://www.cs.ubc.ca/~jqdawson

University of British Columbia
Department of Computer Science

2016 W2
Learning Goals

At the end of this section, you will be able to:

• List the components of computing
• Identify the components of computing for some example processes
• State that the choice of computing platform (including programming language) may affect the efficiency with which an algorithm executes, but not what it can compute
• Explain the distinction between Scratch sprites, scripts and blocks and how they are related to one another
• List some types of statements / commands and data in Scratch
Welcome to Racquel (our lecture TA)

New lecture hall – different sort of set up from our old one

Instructor office hours: Changed to Thurs 11:30am – 12:30pm (or by appointment)

Waitlisted students
  – section still can’t increase in size (TA and lab constraints)
  – sign in again today after class
How will note posting / solutions work?

• You’re expected to fill in the blank sections of your notes yourselves (for questions we do in class)
• I will only fill in sections for extra questions/examples that we don’t cover in lecture

• **Clickers:** I will posted the clicker questions only after lecture
  – your job to make notes about solution(s)
  – marked 50% participation + 50% correctness

• **In-class exercises:** You’ll hand in the exercises during class, and then I will post solutions after class (start next week)
  – mark will be based on participating and completing question
  – marks will be posted to Connect
The Components of Computing

- A general purpose computational process has three conceptual components:
  - Procedure (or recipe) or program or script
  - Data
  - Platform or control or computational engine which executes the procedure
- The platform defines the primitive operations and types of data that the procedure can use
- Successful completion
  - Depends on the data satisfying pre conditions
  - Results in the data satisfying post conditions
The Components of Computing: Hand Washing

- Consider the process of washing your hands
- Procedure:
  - Platform:
  - Data:
  - Preconditions:
  - Postconditions:
Programming Languages

• A programming language provides
  – a variety of types of data values, i.e. numbers, character strings, etc.
  – a set of operations on data,
  – rules on how to put together operations to create statements or commands

• A procedure, called a program or script, is a sequence of statements

• Different languages offer different operations and statements depending on the application area the language is made for

• A language interpreter or compiler is a software program that translates the language statements into operations the computer operating system knows how to execute
What is the right Platform / Programming Language?

• Consider:
  - Hello
  - Hello
  - Hello
  - Bonjour
  - Hola
  - Aloha
  - Guten Tag

• To a large extent, it doesn’t matter

• *Universal Turing Machine* can do the same thing as any other computer (although it may take longer, require more memory and/or be harder for a person to understand)
Scratch (from the Programmer’s Point of View)

- Scratch is a visual programming language for simple animations, games, music and arts
- In Scratch, we have a number of *sprites* (story characters who can do things)
- Each *sprite* has *scripts, costumes, sounds* and other data
  - Data may be implicit (position, orientation, etc) or explicit
- *Costumes* and *Sounds* also store data, but cannot perform any actions by themselves
- Each script is made up of blocks and operators and are executed sequentially
- The Scratch environment knows how to execute each block
A Brief Look at Scratch

- Some example Scratch projects
  - In “see examples”: Dance Party, Pong Starter, Starfish Choir
Scratch (from the Scratch Interpreter’s Point of View)

- The Scratch interpreter is a piece of software that runs on the Scratch site.
- It reads the Scratch project that you create and translates it into instructions which are sent to your web browser which executes them and displays the result.
Scratch Blocks

• Within the “scripts” tab, there are a number of palettes with different types of operations and data:

• Each palette contains blocks representing the specific operations or data; for example

Motion:

Operators:

Control Statements:
More on Scratch Blocks

• Some blocks set, modify or examine data
  – Movement, pen, operators, data
    ![Image of Scratch blocks](image1.png)

• Some provide input / output
  – Looks, sounds, sensing
    ![Image of Scratch blocks](image2.png)

• Some control whether other blocks are executed
  – Control, events, “more blocks”
    ![Image of Scratch blocks](image3.png)

• Not every block in a palette does the same thing
    ![Image of Scratch blocks](image4.png)
Data in Scratch

• Scratch has different types of data
  – Sprites: the cat, Cassy
  – Numbers: 10, 4.5, etc.
  – Strings of characters: Hello, this is a longer string, etc.
  – Costumes
  – Sounds

• Different types have different operations

• Some values can be stored in **variables**, so they can be modified and/or used again
  – Some variables are predefined: time, sprites’s x position
Operators and Expressions

- Arithmetic expressions return numerical values:
  - +, -, *, /
  - pick random 1 to 10
  - sqrt of 10

- String related expressions:
  - join hello world
  - letter 1 of world
  - length of world

- Comparison operators create **boolean** expressions that return **true** or **false**:
  - <, =, >

- Can combine boolean expressions with **and, or, not**:
  - and
  - or
  - not

- Note the different shapes of the different expressions:
  - What kind of expressions can we use with: +, -, *, /?
  - What kind of expressions can we use with: and, or, not?
  - Which expressions can we put in the top box of if, if-else, forever-if, wait-until, repeat-until blocks?
Scratch Variables

• Variables are names that hold values we want to keep and use later
• Using the Variables palette, can define two types of variables:
  – for a single sprite or for all sprites to share
• Can set and change the variable's value any time:
  – its value can be set (or changed ) to be a number or a string
• A variable can then be used anywhere we need a value
  – depending on its value, the variable can be used where we need that type of value, for instance
  – variables with numerical values can be used with +, -, *, /
  – any variable can be used with <, >, =, say command, think command, etc.
• Note how Scratch treats wrong values:
  – if a number is used when a string is needed, the number is treated as a string of its digits
  – if a string is used when a number is needed, the string is treated as 0
Summary

• The main components of a computational process are:
  – program (procedure), data, and a platform that runs the procedure

• A programming language provides
  – data types for defining values for the data
  – variables for storing values
  – operations on data
  – statements that control how the statements will be executed

• Scratch is a graphical programming language for simple animations, games, music and arts

• Scratch data types include: sprites, numbers, strings costumes, sounds, etc.

• Scratch statements include: choice and repetition statements

• Scratch scripts operate on sprites
For next time

• Review slides and readings for next section and complete the pre-class quiz (will be posted by end of day today)
change log

• added: admin slide, note-taking expectations slide,