SCIENCE NORTH SCIENCE NORD
SUDBURY, ONTARIO, CANADA

	Grade 7-8

Lesson Plan

Coding Tool	Scratch
Time Required	2 periods

Math Curriculum Connections

Algebra: Coding
C3. Solve problems and create computational representations of mathematical situations using coding concepts and skills

Specific Expectations
C3.1 solve problems and create computational representations of mathematical situations by writing and executing code
C3.2 read and alter existing code, including code that involves events influenced by a defined count and/or sub-program and other control structures, and describe how changes to the code affect the outcomes and the efficiency of the code

Data: Probability
D2. Describe the likelihood that events will happen, and use that information to make predictions

Specific Expectations
D2.1 describe the difference between independent and dependent events, and explain how their probabilities differ, providing examples
D2.2 determine and compare the theoretical and experimental probabilities of multiple independent events happening and of multiple dependent events happening

Science Curriculum Connections

Understanding Life Systems

Interactions in the Environment
Ecosystems are made up of biotic (living) and abiotic (non-living) elements which depend on each other to survive

Overall Expectations
- Demonstrate an understanding of interactions between and among biotic and abiotic elements in the environment

Specific Expectations
2.1 demonstrate an understanding of an ecosystem as a system of interactions between living organisms and their environment
3.2 identify biotic and abiotic elements in an ecosystem, and describe the interactions between them
3.7 explain why an ecosystem is limited in the number of living things that it can support

Description
Through hands-on and coding activities, students will work with data to investigate independent and dependent events and correlations. They will use these concepts to code a program in Scratch that models interactions in the environment using data inputs. Students will practice using loops, conditional statements, and variables to build and modify their code.

Success Criteria	Materials and Media
By the end of this lesson, students will be able to identify independent and dependent events. They will be able to code programs that depend on other programs to be executed, using such control functions as conditional statements, loops, and variables.	• Computer or Tablet with access to Scratch (either browser version or app) • Ball of twine • Pieces of Paper or Index Cards • Eco Data Coding Guide • Eco Data Handout Activities

Computational Thinking Skills

This lesson makes heavy use of variables. Variables store data — or information — that can be used in a program. Students will practice creating variables, manipulating the values that they represent, and inserting them into programs to modify program outcomes.

Because we will be modelling an ecosystem using code, students will gain experience using control structures such as loops, to repeat tasks until an event occurs, and conditional statements, to signal different outcomes in accordance to changes that occur while the simulation is running.

The unplugged classroom activity will also have students practice describing dependent events using conditional statements.

This lesson also makes use of the “broadcast” function in Scratch, which is a control function that allows a program to call on and execute a sub-program at controlled points while the main program is running. This ties well into the concept of dependent and independent events as the relationship between sub-programs and the programs that broadcast the cue that executes the is a dependent one (i.e., the functioning of a sub-program relies on the outcome of another program executing first).

Students will also see examples of independent programs that will function no matter the activity of other programs running at the same time.

Introduction

An event is the outcome of a specific incident or situation being tracked in an experiment. In this lesson, we are going to explore two different types of events, independent events and dependent events, as they apply to interactions in the environment.

Independent events are two or more events that can happen simultaneously without one affecting the outcome of the other, that is, the probability of one does not affect the probability of the other. Two events are **dependent** if the outcome of the first affects the outcome of the second.

In code, dependent events can often be described using **conditional statements**. A conditional statement can be thought of as an outcome based on the answer to a question. Think of a situation that describes a predator-prey interaction in a habitat.

The question that you're answering might be:

"Are there insects present for bats to eat?"

This question can be answered with a yes (there are insects available in the habitat) or no (there are no insects available).

The outcomes are:

- **if** there are insects available, **then** the bats will eat the insects.
- **if** there are no insects available, **then** the bats have no food and cannot thrive.

In code, this decision can be represented as a conditional statement, known as an if/then/else statement.

In this case the **if** part of the statement is the condition being met (insects available) and **then** is the program that is executed as a result of the condition being met. We can even code for the outcome that will occur when the condition is not being met (insects are not available) as an **else** statement. It might be helpful to think of **Else** as "otherwise".

- **if** there are insects available, **then** the bats will eat the insects; **else** the bats have no food and cannot thrive.

In this lesson, we will be exploring interactions in the environment as independent and dependent events. An ecosystem is made up of **biotic** (living) elements, like plants and animals, and **abiotic** (non-living) elements, like air and water, which interact with each other and depend on each other to survive. These elements can sometimes be so inter-dependent that changes to one element, whether biotic or abiotic, might have a big impact on one or more other elements. Biotic elements of an ecosystem can be further classified as **producers**, capable of creating their own energy from abiotic elements like sunlight, air, and water (e.g., plants, some bacteria), **consumers,** which must eat producers or other consumers to gain energy (e.g., humans, rabbits, fish, bears...), and **decomposers** which gain energy by consuming decaying matter (e.g., earthworms, mushrooms). These three types of biotic elements interact in ways that keep an ecosystem, an environment, in balance. We will practice describing these types of events and interactions using code.

Action

Unplugged Classroom Activity

For this activity, students will be exploring the relationships between biotic and abiotic elements of an ecosystem. Students may sit at their desks or in a large circle to do this activity.

- Assign each student a role as either a biotic (bear, a maple tree, salmon, beetle, mould...) or abiotic component (water, air, sunlight, wind) of an ecosystem (see below for a list of roles. Have the students create a name tag for their desk using paper or an index card so that their component is visible to other students. Pass each student a piece of string that is at least 2 m long (alternatively: you can do this activity with one ball of string or twine that gets passed from student to student).
    - At this point, students should also fill out their role on their **Eco Data handout** and answer Section 1 questions on their handout sheet about their role (which ask students to characterize their role as biotic or abiotic, and to further say whether their role (if biotic) is a producer, consumer, or decomposer).
- Begin with the student who was assigned the role "Air". Ask them to describe the relationship between their role and one other student's role (for example, a deer will need air to breathe). Students may correctly state that many other students' components would have a relationship with theirs; however, direct them to choose and describe only one. They will pass the string to the student whose role participates in the relationship described (e.g., the student who is assigned the "deer" role) but will continue to hold the end of the string.
- Ask the next student ("deer") to look around and find what relationship a deer might have with another student's role and to describe that relationship (e.g., "deer eats birch bark") and have the student pass the string to "birch tree" student while continuing to hold the string.
- Continue in this fashion for up to ten minutes, or until students feel that they have exhausted potential connections. Note that repeating connections (e.g., squirrel, moose, mouse, etc. also connecting to air because they also need air to breathe) is encouraged. You might have to prompt some directions for more challenging interactions (e.g., mushrooms, bacteria as decomposers).
- Ask the students to describe what they observe about the relationships represented by the strings. Some possible observations:
    - Each role is connected to at least one other role
    - All living organisms (except some bacteria) need both air and water
    - Consumers will have fewer connections
    - Producers and decomposers will have more connections
- Choose two relationships from the ecosystem web that the students have created and ask students to describe whether the relationships are independent or dependent.

- For example: "Bumble Bee pollinates Apple Tree" and "Deer eats apples from Apple Tree" are **dependent** events because if the tree is not pollinated, then the outcome is that the apple tree will not produce apples for the deer to consume.
- Another example: "Bear eats Salmon" and "Rabbit needs air to breathe" are **independent** events because the outcome of whether a bear eats salmon or not will not affect whether the rabbit can breathe air or not, and vice versa.
- Ask students to identify one or two more examples.

Students can now complete section two of the **Eco Data handout** to reinforce their learning from this activity and practice describing dependent events as **conditional expressions**.

Here is a list of 35 biotic and abiotic factors that you can assign to your students for this activity:

1) Air
2) Water
3) Sunlight
4) Soil
5) Wind
6) Bear
7) Salmon
8) Maple Tree
9) Bumble Bee
10) Beetle
11) Mushroom
12) Bacteria
13) Deer
14) Grass
15) Dandelion
16) Wolf
17) Frog
18) Birch Tree
19) Squirrel
20) Butterfly
21) Minnow
22) Spider
23) Rain
24) Moose
25) Raccoon
26) Earthworm
27) Mouse
28) Hawk
29) Robin
30) Pollution
31) Crow
32) Porcupine
33) Trout
34) Apple Tree
35) Snake

<u>Coding Math Activity</u>

We are going to use Scratch to create a model of interactions between a consumer (rabbits) and a producer (plants). The program will demonstrate a simplified relationship between the number of consumers in a habitat and the number of plants available for consumption.
A complete example of the base Eco Data program can be viewed and played here:
https://scratch.mit.edu/projects/472504058

Let's break down the basics of what we want our program to do:

- Set the number of plants that exist in our ecosystem and generate these plants in random positions;
- A number counter reflects the number of plants on the screen;

- Set the number of rabbits that exist in our ecosystem and generate these rabbits in random positions, pointing in random directions;
- A number counter reflects the number of plants on the screen;
- Have the rabbits move continuously;
- When a rabbit sprite touches a plant sprite, the plant sprite disappears (is consumed by the rabbit);
- When a plant sprite is removed, the plant counter value decreases by 1;
- A visible timer counts the number of seconds that pass from the start of the program until all of the plant sprites are consumed by rabbit sprites.

Once we have our basic ecosystem, we can introduce other factors to increase complexity of relationships and interactions. These factors can include:

- Plant growth — have plant sprites that haven't been consumed produce more plant sprites over time
- Introducing a continuously moving predator sprite that consumes rabbit sprites;

A detailed step-by-step guide to building these programs in Scratch, including iterations for each of the added factors described above, is described in the **Eco Data Coding Guide.**

Closure and Assessment

By the end of this lesson, students should be able to describe relationships between biotic and abiotic factors within an ecosystem and identify whether those relationships involve independent or dependent events. They should be able to describe dependent relationships using conditional statements. Students should be able to modify code change the output of a digitally created program, playtest their code, and iterate.

For assessment, collect the "Eco Data" handouts from the students. Review their work to ensure that they understood the concepts of independent and dependent events and conditional statements by correctly completing the activities.

Adaptations	Extensions
• The unplugged classroom activity can be conducted with students standing or seated.	• Students who finish early can experiment with their virtual ecosystems by adding complexity (some step-by-step examples are outlined in detail in the **coding guide**) and making modifications to their code for the interactions or relationships between different elements of their ecosystem.

	• Students may also choose to modify the type of ecosystem that they are representing, for example, changing the sprites and code accordingly to represent an aquatic ecosystem.

Additional Resources

- Scratch.mit.edu – Scratch is a free resource and no account is required to create a program; however, an account is required to save your work.