SCIENCE NORTH SCIENCE NORD
SUDBURY, ONTARIO, CANADA

Transformations	Grade 3-4

Lesson Plan

Coding Tool	Scratch
Time Required	Two periods

Math Curriculum Connections - Grade 3

Algebra - Coding

C3.1 solve problems and create computational representations of mathematical situations by writing and executing efficient code, including code that involves sequential, concurrent, and repeating events

C3.2 read and alter existing code, including code that involves sequential, concurrent, and repeating events, and describe how changes to the code affect outcomes

Geometric and Spatial Reasoning – Location and Movement

E1.4 give and follow multistep instructions involving movement from one location to another, including distances and half- and quarter-turns

Math Curriculum Connections – Grade 4

Algebra - Coding

C3.1 solve problems and create computational representations of mathematical situations by writing and executing efficient code, including code that involves sequential, concurrent, repeating, and nested events

C3.2 read and alter existing code, including code that involves sequential, concurrent, repeating, and nested events, and describe how changes to the code affect outcomes

Geometric and Spatial Reasoning – Location and Movement

E1.2 plot and read coordinates in the first quadrant of a Cartesian plane and describe the translations that move a point from one coordinate to another

E1.5 describe and perform translations and reflections on a grid, and predict the results of these transformations

Description

Geometry and coding go together like bread and butter! In this lesson, students will practice the fundamentals of transformation, as well as describing positions in terms of x-y coordinates with both hands-on unplugged and on-screen activities. Students will apply transformations using Scratch to build algorithms that translate and reflect shapes.

Success Criteria

- 3rd and 4th grade students will be able to describe and perform translations of basic geometric shapes on a grid
- 4th grade students will be able to plot and read x-y coordinates on a Cartesian plane
- 3rd and 4th grade students will be able to construct and execute block-based code that includes sequential, concurrent, repeated and nested events.

Materials and Media

- Computer or iPad with access to Scratch
- Long rope or string (50 – 100 ft)
- Pencils
- Rulers (not required, but helpful for drawing transformations on paper)
- Transformations Handout
- Transformations Coding Guide

Computational Thinking Skills

Students will practice their skills in thinking algorithmically to describe the instructions that a program must complete in sequential order. In the unplugged activity, students will practice breaking down the movements required to perform transformations on a shape into a series of steps, using x-y coordinates on a grid. This learning will prepare them for building computer algorithms to communicate instructions in coding languages.

During the online coding activity, students will use Scratch to build and execute programs that will draw geometric shapes on a grid and perform three types of transformations (translations, rotations, and reflections) on those shapes. They will also explore the use of loops, a coding element that makes code more efficient by allowing us to repeat selected steps, to redraw geometric shapes after a series of transformations have been executed.

The **Transformations Coding Guide** document for this lesson includes a detailed step-by-step procedure for using Scratch.

Introduction

Transformation means change. In geometry, **transformations** are changes that we can apply to shapes. In this lesson, we are going to investigate three types of transformations: translations, rotations, and reflections. These types of transformations are also known as rigid transformations because the shape does not experience a change in size, area, angles or line lengths. The post-transformation shape is congruent with the initial shape.

Translations involve moving our shapes up and down or left and right. The shape's direction, size, or arrangement do not change as we move the shape from one location to another.

A **Rotation** is a type of transformation that takes a shape and rotates it around a given point. The shape does not change size and is not distorted in any way (e.g., an equilateral triangle that is rotated will still be an equilateral triangle of the same size) — it will just be pointed in a new direction. We describe how far a shape has rotated around a given point in terms of degrees, where 90˚ is a quarter-turn, 180˚ is a half-turn, and 360˚ is a full rotation (or a full circle).

A **reflection** is a transformation that acts like a mirror. The shape that is being reflected does not change size and is not distorted in any way. Instead, the shape and all its points are flipped to the opposite side of a defined line of reflection — like how when you stand in front of a mirror and lift your right hand, your reflection will show a hand lifted on the same side as your lifted hand (but if it were a real person and not a reflected image, this would be the person's left hand).

Coordinates allow us to communicate precisely about the transformations that we apply to our shapes. In the case of a translation, for example, we can describe a change the vertical position of our shape (movement up or down) as a change in a y-value, and we can describe a change the horizontal position of the shape (movement left or right) as a change in an x-value. Coordinates are always written as (x, y), where x is the horizontal position of the point, and y is the vertical position of the point.

During this lesson, we will be practicing the use of coordinates on a grid system and using coordinates to describe positions of points of geometric shapes drawn on a grid. We will then be applying this knowledge by investigating transformations on paper, using physical manipulatives, and using code.

Action

Unplugged Activity

To reinforce the use of coordinates to perform transformations, have students practice tasks that require them to use coordinates to draw a shape.

Transformations Demonstration

As a class, we will be creating geometric shapes with string and performing transformations. The twist is that the students will be acting as points on a grid system.

This activity can be performed in the classroom (ideal if desks are arranged in rows or groupings that can be easily converted to a grid system. Otherwise, it can be performed outside or in a clear space with students sitting or standing along rows and columns marked out using tape or pylons. The first row and column should be labelled row/column 0, then continue sequential with rows/columns, 1, 2, 3 …

Once the grid is established, explain the two axes of the grid (horizontal, or x-axis; and vertical or y-axis) to students and how they can be used to describe coordinates (or points) on a grid. For a simplified introduction, you can also relabel one axis so that it used letters instead of numbers (e.g., a coordinate would be A6, instead of (0,6)).

Ask students to determine what their personal coordinate is on the grid. If you are using an x-y coordinate system, the x-value (the column that the student is sitting in if you think in terms of

Science North is an agency of the Government of Ontario and
a registered charity #10796 2979 RR0001

classroom arrangement) should always come first and then the y-value (the row that the student is sitting in) should come second. These values are written as (x, y). For example, the student sitting in the corner at the intersection of the first row and first column would have the coordinate (0, 0), and the student sitting next to them would be (1, 0), while the student behind them would be (0, 1). For additional visual reinforcement, a similar grid could be shown on the board for students to use as a reference.

Ask students to determine how to create a square using string and coordinates. Note that this square should be small enough to leave room for translations. Recall that the square should have four equal sides and the same angle at all for corners. Together, they should determine which four students (and their coordinates) would be the corners of the square. This square can be drawn on the board using the coordinates as well as using the string among the students.

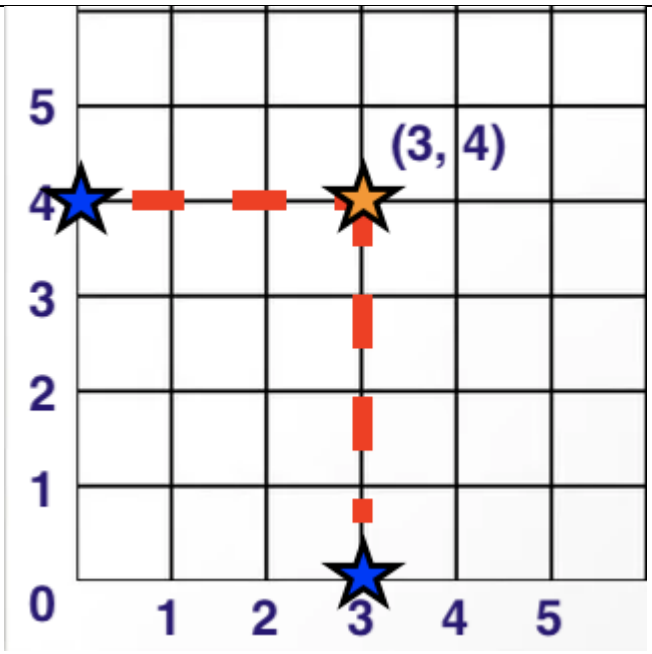
Once the students have created a square, ask the students to describe what steps are needed to translate the square so that a corner is now held by a specific student. Establish what the new coordinates would be and what movements are needed (e.g., down one row and to the left by two columns) to accomplish the translation. Draw the translation on the board and perform the translation with string.

Ask the students to describe what steps they would have to perform to rotate the square one quarter-turn. Establish together where the coordinate points should be after this rotation and perform the rotation with string.

You can try different examples with different geometric shapes (e.g., rectangle, triangle).

The **Transformations handout** provides additional activities for students to practice locating coordinates on a grid and describing transformations as a series of steps.

Tip: a helpful way to find a coordinate's point on a grid is to put one finger on the x-value (on the horizontal or x-axis) and put one finger on the y-value (on the vertical or y-axis) and to slide your fingers in straight lines across the grid until they meet at the coordinate point. Below is an example for the coordinate (3, 4) — the orange star shows the location of this coordinate on the grid:



Coding Math Activity

We are going to use Scratch to create programs that draw geometric shapes on a grid using Scratch's **Pen** extension and perform three types of transformations (translation, rotation, and reflection) on these shapes.

We will create a base shape-drawing program that we will iterate upon to perform, translations, rotations, and reflections. This lesson is easily adaptable into segments so that you can do all transformations in Scratch as one lesson, or you can separate each transformation into its own coding lesson as you introduce different transformation types to your classroom.

A detailed step-by-step guide to building these programs in Scratch is described in the **Transformations Coding Guide.**

Closure and Assessment

By the end of this lesson, students should be able to use (x, y) coordinates to place points on a grid. They should also be able to describe the movement steps required to perform translations, reflections, and rotations of simple geometric shapes on a grid. They should be able to build, test, and iterate simple sequential programs in Scratch.

For assessment, collect the Transformations handouts from the students. Review their work to ensure that they understood the concept of coordinate systems by appropriately "decoding" the drawing in the first question and successfully described a drawing of their

own creation using coordinates in question 2. Ensure that students understood how to describe the steps to perform translations and rotations on a grid by evaluating their responses to questions 3 & 4 using the answer key.

Adaptations

- The transformations classroom demonstration can be done individually on a paper grid, or as a class on a blackboard/SMART board/screen with students contributing answers that the teacher places.
- The transformations classroom activity can be done with students seated at desks or on the floor or standing (outside or in a space clear of obstacles).
- Students may describe movement using written sentences or by drawing arrows.

Extensions

- Many tabletop games use coordinate systems and/or require players to break down movement as steps. Other games can easily have coordinates applied to them (e.g., Tic Tac Toe, Connect Four). This is a great opportunity to introduce games such as Battleship or chess into the classroom for students to investigate how coordinate systems are used in these games.
- Students who finish the coding activity early can be challenged to try and combine transformations into one program (e.g., Translate a rectangle up 20 px and to the left 60 px, and rotate it one half-turn).

Additional Resources

- Scratch.mit.edu (free – no account required)