Scratch programming and Numeracy in Senior Primary Classes

Module 2

lero THE IRISH SOFTWARE ENGINEERING RESEARCH CENTRE

National Centre for Technology in Education
Ionad Náisiúnta don Teicneolaíocht san Oideachas

Module 2

Scratch programming and Numeracy in Senior Primary Classes (NCTE/Lero)

Course Notes

Suggested Duration of Module

4 Hours

Objectives

Module 2 will cover:

1. The use of Scratch operators to round numbers to a certain place value
2. The exploration of writing and designing algorithms on paper
3. The use of Scratch to draw 2D shapes such as squares, triangles and circles using programming skills in Scratch.
4. The use of Scratch to support problem solving
5. Discussion on how Scratch can support Numeracy in the classroom

Participants are asked to bring a digital camera for Module 3 of this course.

1. The use of Scratch operators to round numbers to a certain place value

1.1 Curriculum Requirements

The child should be enabled to:

- round whole numbers and round decimals

1.2 Activity

Click on stage in the Sprite List, and then click on Backgrounds.

Choose a background from the Scratch folders.

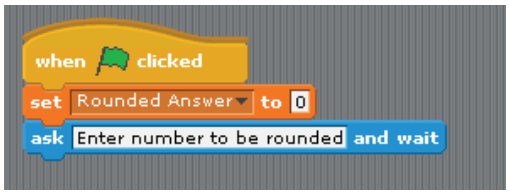
Select a sprite. You may delete the cat if you use another sprite.

This sprite will ask the user to input a decimal and will give the rounded answer.

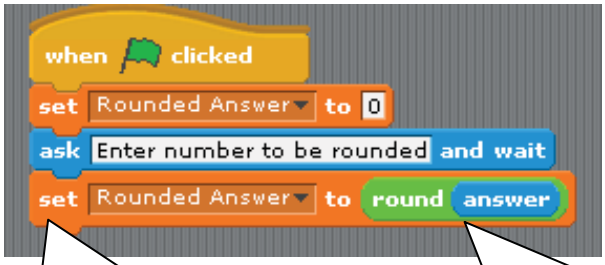
This script will use a variable and operators.

This will delete any previously entered numbers.

The sprite must ask the user to input a decimal.



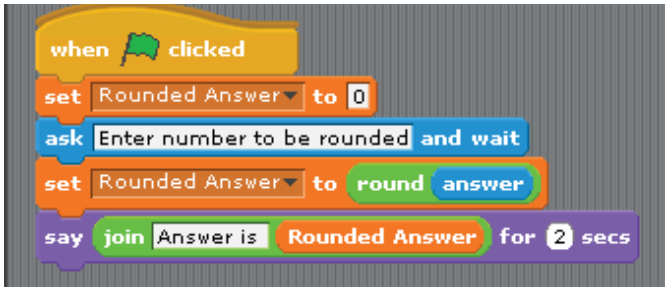
The blue block asks a question of the user and the response is saved as "answer". This answer can then be used in various calculations. In this case we will round it to the nearest whole number and output the solution as the variable "rounded number".



This block will change the variable from 0 to the solution..

This operator will calculate the closest whole number to the inputted decimal.

We must now display the solution on the stage.



1.3 Participant Activity

Discuss how you could use these features in your Maths lesson. Remember that the participants should be actively working with Scratch.

Task 1. Challenge your class to create a program which tests their friends' knowledge of rounding decimals.

2. The exploration of writing and designing algorithms on paper

When writing programs, we must provide the computer with very specific instructions, we cannot omit any details. Discuss whether it would be good if people followed instructions exactly. e.g. what would happen if you pointed to a closed door and said "Go through that door?" Explain that computers work by following lists of instructions, and that they do exactly what the instructions say, even if they are incorrect (or nonsensical). This lesson explores algorithms, or instructions for tasks. Try the following activity.

2.1 Activity

1. Ask everyone to take out some blank sheets of paper each and a pen or pencil.
2. Ask them to write a set of instructions to describe how to make a cup of tea. Remember that a computer would require every step. This is what you must write.
3. Once this has been attempted, ask participants to swap their instructions and note any differences or omissions. Reflect on the problems involved in creating a set of instructions

2.2 Participant Activity

Discuss the importance of algorithms in writing a program.

Task 1. Participants are paired for the next activity. Ask participants to write instructions for each other about how to get to a mystery location by starting at the front door.

Participants then test their partner's algorithm by trying to follow their instructions to the mystery location. If necessary the algorithm can be refined until the most clear and efficient set of instructions is produced.

Clear, precise instructions are very important in Scratch, as we will see in Section 3.

3. The use of Scratch to draw 2D shapes such as squares, triangles and circles using programming skills in Scratch.

3.1 Curriculum requirements

The child should be enabled to:

- sort, describe and name 2-D shapes
- construct and draw 2-D shapes
- solve problems involving shape and space
- give simple moving and turning directions
- identify half and quarter of shapes
- identify and discuss the use of 2-D shapes in the environment
- explore, describe and compare the properties (sides, angles, parallel and non-parallel lines) of 2-D shapes
- combine, tessellate and make patterns with 2-D shapes
- use angle and line properties to classify and describe triangles and quadrilaterals
- classify 2-D shapes according to their lines of symmetry
- plot simple co-ordinates and apply where appropriate

3.2 Participant Activity

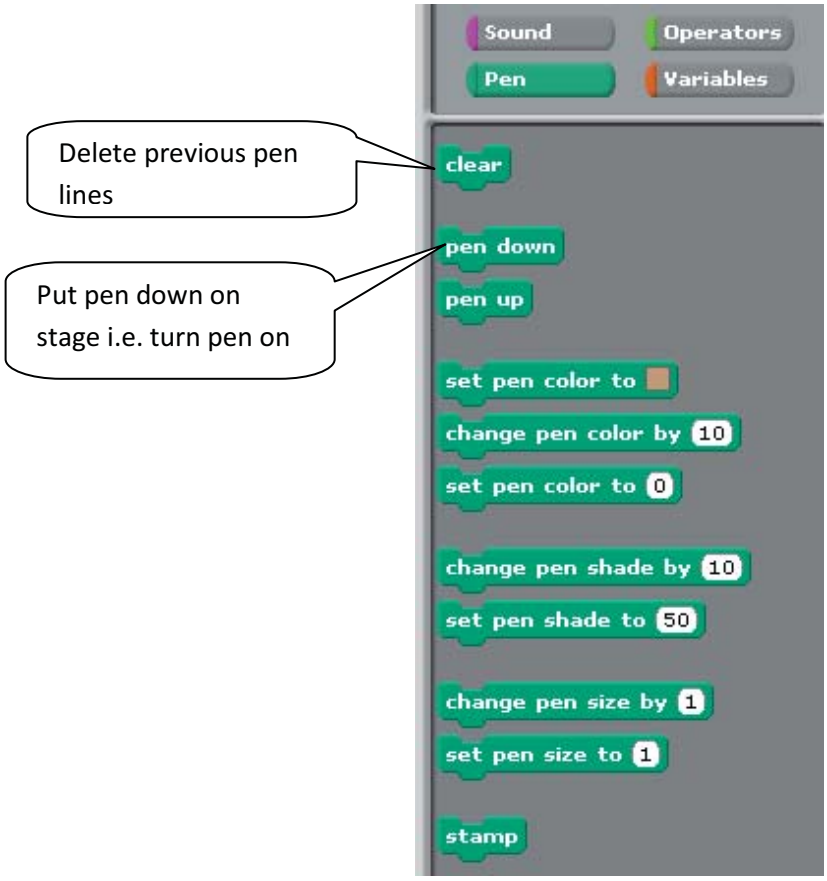
Drawing in Scratch relies on very specific instructions, like the algorithms we just worked on. Scratch drawing works in a similar way to the programming software, Logo.

It is as if your sprite is holding a pencil. Whenever the sprite moves it leaves a line behind it. To draw a shape we must "walk" that shape. Ask a volunteer to stand up. Ask a second person to instruct them in "walking" a square.

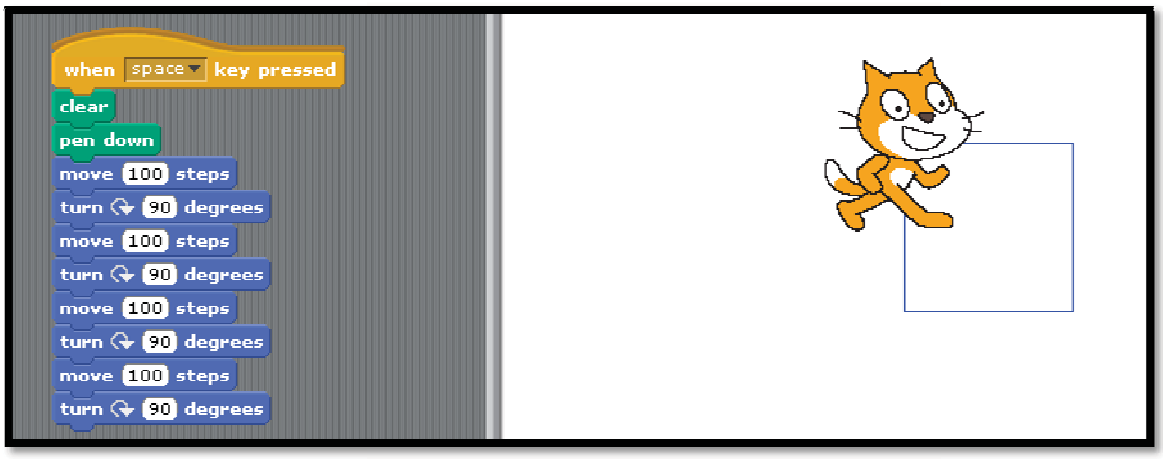
For example:

Walk forward 2 steps, turn 90 degrees to the right.
Walk forward 2 steps, turn 90 degrees to the right.
Walk forward 2 steps, turn 90 degrees to the right.
Walk forward 2 steps, turn 90 degrees to the right.

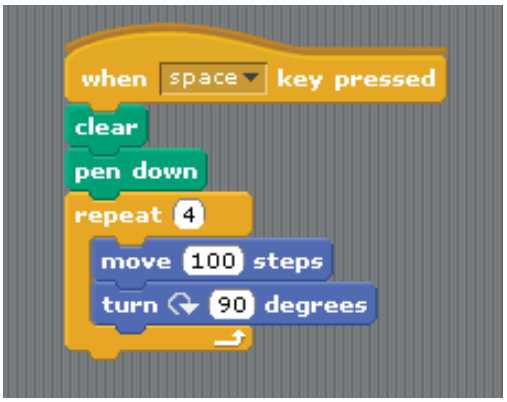
We work in the same way with Scratch.



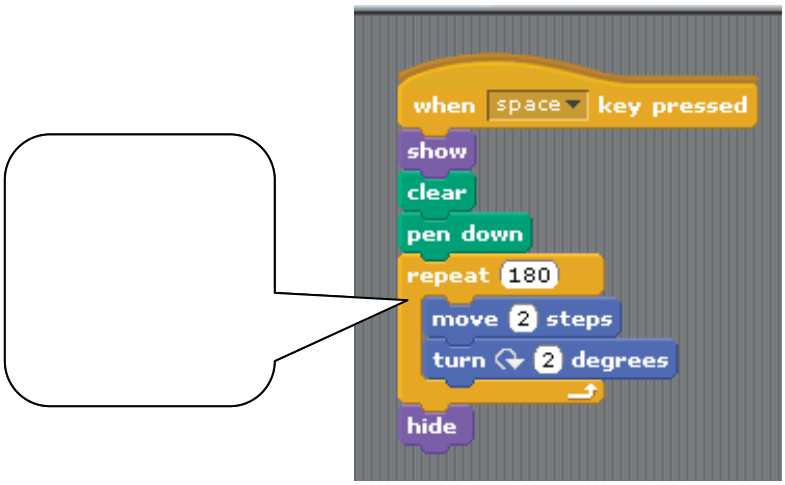
Draw a square



Did you notice this code is repetitive (Because the movements and turns are all equal)? In Scratch we can use a repeat loop to create a shorter piece of code.



Use the same basic controls to draw a rectangle, triangle, pentagon and other shapes. Can anyone draw a circle?



When the basic shapes are mastered, you can make patterns by using repetitions.

3.3 Participant Activity

Discuss how Scratch can support learning of 2D shapes.

Task 1. Create a program which will create a pentagon

Task 2. Create a program that will draw 4 different shapes on the stage.

The following are some projects which explicitly demonstrate number in Scratch.

Flower –

http://www.scratch.mit.edu/projects/ballns/1905084

2D Shape Maker

http://scratch.mit.edu/projects/Dastyruck/75089

4. The use of Scratch to support problem solving

As discussed earlier children will be problem solving from the moment they start using Scratch. The will plan, execute and debug their programs. You may also wish to use Scratch to support more formal mathematical problem solving.

4.1 Curriculum Requirements

Problem solving has a central role in the Primary Mathematics Curriculum, which states:

> "Developing the ability to solve problems is an important factor in the study of mathematics. Problem-solving also provides a context in which concepts and skills can be learned and in which discussion and co-operative working may be practised. Moreover, problem-solving is a major means of developing higher-order thinking skills. These include the ability to analyse mathematical situations; to plan, monitor and evaluate solutions; to apply strategies; and to demonstrate creativity and self-reliance in using mathematics.

Success helps the child to develop confidence in his/her mathematical ability and encourages curiosity and perseverance. Solving problems based on the environment of the child can highlight the uses of mathematics in a constructive and enjoyable way."

(http://www.curriculumonline.ie/en/Primary_School_Curriculum/Mathematics/Mathematics_Curriculum/)

4.2 Activity – Create an animation which accurately demonstrates the solution to a word problem.

Word problems feature frequently in Maths textbooks. They can be the most challenging aspect of any topic. However, by allowing children to use Scratch we can bring the problem to life. View the following examples.

Q1. There are 12 apples in one box. How many apples are there in 6 boxes?

A1. http://scratch.mit.edu/projects/ballns/2524268

Q2. The monkey has 16 bananas. He eats 7. How many bananas are left?

A2. http://scratch.mit.edu/projects/ballns/2553431

4.3 Participant Activity

In Microsoft Word or on a piece of paper, create 3 word problems which you want your class to solve using Scratch. Ask the pair beside you to solve one of your problems.

5. Discussion on how Scratch can support Numeracy in the classroom

5.1 Today's objective was to cover:

1. The use of Scratch operators to round numbers to a certain place value
2. The exploration of writing and designing algorithms on paper
3. The use of Scratch to draw 2D shapes such as squares, triangles and circles using programming skills in Scratch.
4. The use of Scratch t support problem solving
5. Discussion on how Scratch can support Numeracy in the classroom

5.2 Discuss:

In what way could you extend or adapt today's activities for your own classroom? What would work well? What might be difficult to implement or use?

5.3 Sample Projects

All Scratch projects use some element of number. Children will experience real world use of number when completing any task in Scratch. The following are some projects which explicitly demonstrate number in Scratch.

Addition tutor –

http://scratch.mit.edu/projects/jsnyders/39620

Subtraction Calculator -
http://scratch.mit.edu/projects/timmy555590/2486310

Addition of Fractions Quiz

http://scratch.mit.edu/projects/joelzehring/862192

These projects are examples of what you can build to incorporate mathematical skills. Browse these projects for ideas! Look at the code for a deeper understanding of Scratch.