Virtual Coding & Robotics

Michael Frankfort @mfrank_76

York Region District School Board

Michael.Frankfort@yrdsb.ca

Michael Minchella

York Region District School Board

Michael.Minchella@yrdsb.ca

#vrcodingrobotics

bit.ly/3w7SGIv

SESSION OVERVIEW

- Minds-On Activity
- Coding In The Curriculum / Curriculum Progression
- Coding In The Classroom
- Coding Examples
- VEX VR - Meet The VEXCode VR Robot
- VEX VR - Activities
- Assessment & Evaluation
- Supporting Resources
- Q & A

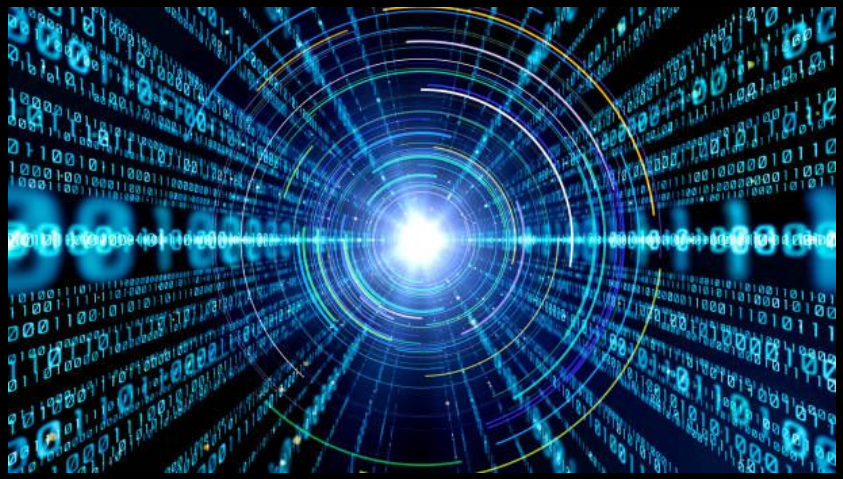


Photo Credit www.pexels.com

MINDS-ON

- If you had to provide instructions for how to brush your teeth, what steps would you include?

- What step or action would be first?
- What steps or actions must happen in a certain order?
- Are there any steps or actions that are repeated?
- Which steps or actions do you need to include and/or explain?
- What happens if you run out of toothpaste?

CODING IN THE CURRICULUM

code

An instruction or set of instructions that can be executed by a computer or other device. *See also* **execute**.

coding

The process of writing computer programming instructions.

As students progress through the grades, their coding experiences also progress, from representing movements on a grid, to solving problems involving optimization, to manipulating models to find which one best fits the data they are working with in order to make predictions. Coding can be incorporated across all strands and provides students with opportunities to apply and extend their math thinking, reasoning, and communicating.

Coding in Elementary

- Code is the language that a computer understands.
- Telling a computer to do what you want it to do.
- Breaking a task down into logically sequenced step-by-step commands.
- Can be incorporated into learning for all curriculum areas.
- Fundamental computational thinking skill of the modern world.
- Employable skill that teaches students to solve problems, take risks, think critically and logically while being empowered to fail in a safe environment.
- Introducing coding in the classroom can be a challenge, especially for educators who are not familiar with computer science. Fortunately, a number of apps, software and guides have been produced that make the subject matter of coding easy to grapes for young learners.

https://www.edugains.ca/resourcesTELO/CE/CodingInterface/interface.html

Ontario

Français

Curriculum and Resources

Curriculum | Assessment and Evaluation | Resources | Parents

- Home
- Curriculum context
- Grades
- Glossary
- Resources
- Downloads

ELEMENTARY

Mathematics (2020)

Curriculum overview videos

An Overview

Learning Areas Overview

Teaching Supports Overview

#vrcodingrobotics

Overall Expectation

Contents

C3. Coding

Specific Expectations

By the end of Grade 4, students will:

Coding Skills

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

nested events (coding)

Control structures that are placed inside other control structures; for example, loops occurring inside other loops, or a conditional statement being evaluated inside a loop. *See also* **control structure (coding)**.

Glossary of Terms

TERM	DEFINITION
Algorithm	A list of steps to finish a task. A set of instructions that can be performed with or without a computer. For example, the collection of steps to make a peanut butter and jelly sandwich is an algorithm.[1]
Coding	Instructions given to a computer to perform a specific task.
Condition	Actions that occur if a statement is true or false. (example: If it is raining then get umbrella. If it is not raining then do nothing.)
Debug	Finding and fixing errors in programs.[2]
Function	A piece of code that can be called over and over.[3]
If Statement	A line that determines whether or not something will happen in a program or not. (example: If it is raining...then get umbrella)
Input	Data provided to a computer system, such as via a keyboard, mouse, microphone, camera or physical sensors.[4]
Loop	The action of doing something over and over again.[5]
Output	The information produced by a computer system for its user, typically on a screen, through speakers or on a printer, but possibly though the control of lights, motors, etc. in physical systems.[6]
Parameters	Extra bits of information that you can pass into a piece of code to customize it.[7]
Procedure	A section of a program that performs a specific task.
Process	A series of actions or steps.
Program	An algorithm that has been coded into something that can be run by a machine.[8]
Repetition	Repetition is a way to get the computer to following the same set of instructions forever, a given number of times or until a condition is met.[9]
Selection	Part of a computer program that is only executed if a certain condition is met.[10]
Sequencing	Presenting are your steps in an algorithm in the correct order.[11]
Then	What happens if a condition has been met in a statement[12](example: If it is raining...then get umbrella)
Variable	A placeholder for a piece of information that can change.[13]

https://www.edugains.ca/resourcesTELO/CE/CodingInterface/Resources/Extra%20Resources/_content.html

CURRICULUM OVERVIEW

C3. Coding

solve problems and create computational representations of mathematical situations using coding concepts and skills

C3.1 solve problems and create computational representations of mathematical situations by writing and executing code, including code that…

C3.2 read and alter existing code involving…, and describe how changes to the code affect the outcomes [and the efficiency] of the code

CURRICULUM PROGRESSION

OVERALL EXPECTATION C3. solve problems and create computational representations of mathematical situations using coding concepts and skills

SPECIFIC EXPECTATIONS

Grade 1	Grade 2	Grade 3	Grade 4	Grade 5	Grade 6	Grade 7	Grade 8
Coding Skills							
C3.1 solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves sequential events	**C3.1** solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves sequential and concurrent events	**C3.1** solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves sequential, concurrent, and repeating events	**C3.1** solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves sequential, concurrent, repeating, and nested events	**C3.1** solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves conditional statements and other control structures	**C3.1** solve problems and create computational representations of mathematical situations by writing and executing efficient code, including code that involves conditional statements and other control structures	**C3.1** solve problems and create computational representations of mathematical situations by writing and executing efficient code, including code that involves events influenced by a defined count and/or sub-program and other control structures	**C3.1** solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves the analysis of data in order to inform and communicate decisions
C3.2 read and alter existing code, including code that involves sequential events, and describe how changes to the code affect the outcomes	**C3.2** read and alter existing code, including code that involves sequential and concurrent events, and describe how changes to the code affect the outcomes	**C3.2** read and alter existing code, including code that involves sequential, concurrent, and repeating events, and describe how changes to the code affect the outcomes	**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 the outcomes	**C3.2** read and alter existing code, including code that involves conditional statements and other control structures, and describe how changes to the code affect the outcomes	**C3.2** read and alter existing code, including code that involves conditional statements and other control structures, and describe how changes to the code affect the outcomes and the efficiency of the 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	**C3.2** read and alter existing code involving the analysis of data in order to inform and communicate decisions, and describe how changes to the code affect the outcomes and the efficiency of the code

CURRICULUM PROGRESSION

Grade 1 - Sequential events

Grade 2 - Concurrent events

Grade 3 - Repeating events

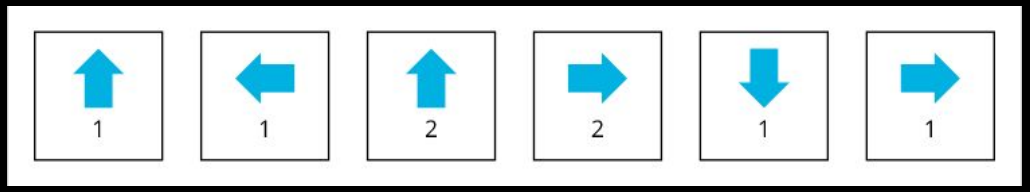
Grade 4 - Nested events

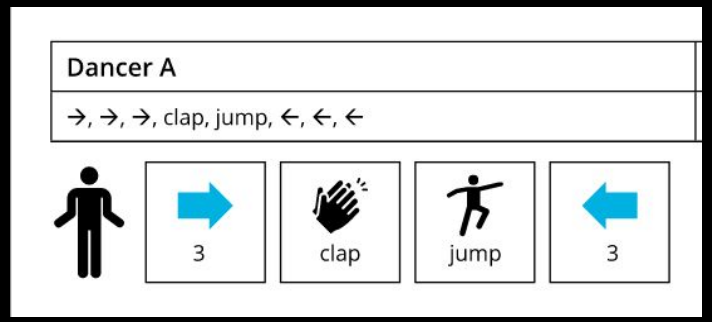
Grade 5 - Conditional statements and other control structures

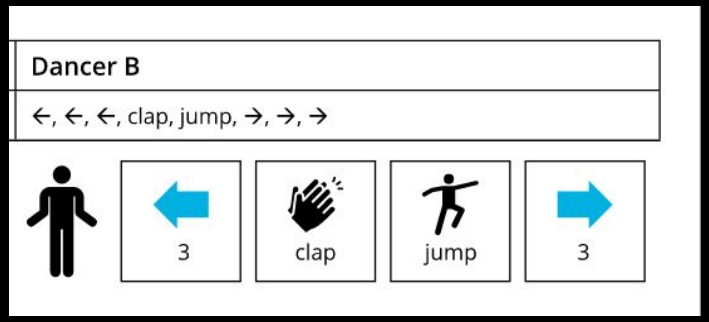
Grade 6 - Efficient code

Grade 7 - Events influenced by a defined count and/or sub-program

Grade 8 - Analysis of data in order to inform and communicate decisions







CURRICULUM PROGRESSION

Grade 1 - Sequential events

Grade 2 - Concurrent events

Grade 3 - Repeating events

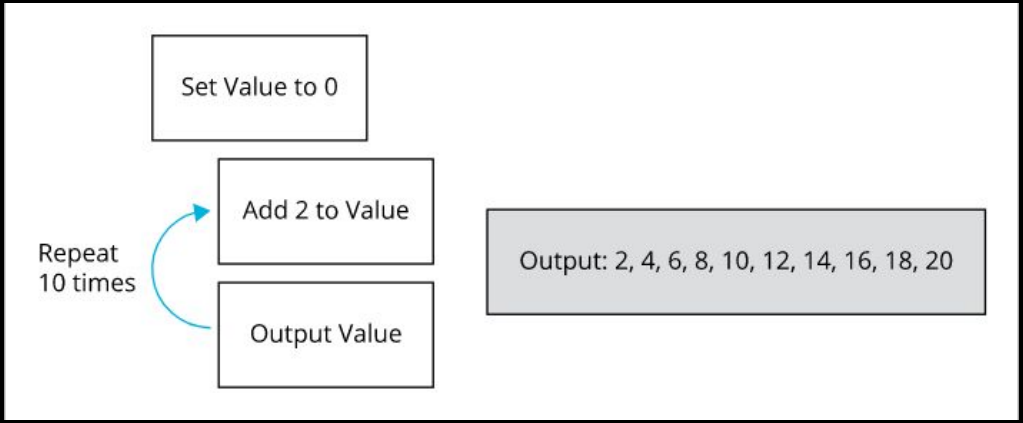
Grade 4 - Nested events

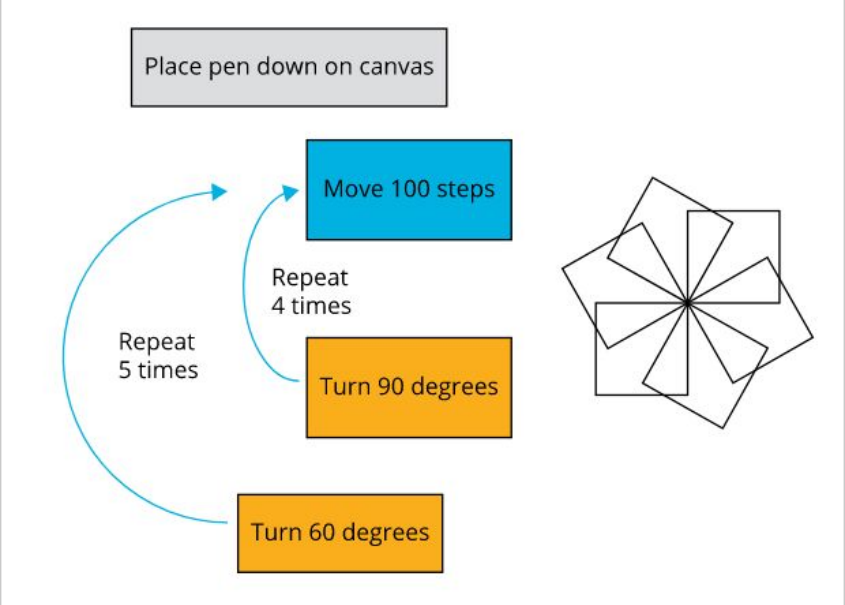
Grade 5 - Conditional statements and other control structures

Grade 6 - Efficient code

Grade 7 - Events influenced by a defined count and/or sub-program

Grade 8 - Analysis of data in order to inform and communicate decisions





CURRICULUM PROGRESSION

```
If guess > actualNumber, then print "Sorry, your
guess was too high."
    else if guess < actualNumber, then print
    "Sorry, your guess was too low."
    else print "Yes! Your guess was correct."
```

Grade 1 - Sequential events

Grade 2 - Concurrent events

Grade 3 - Repeating events

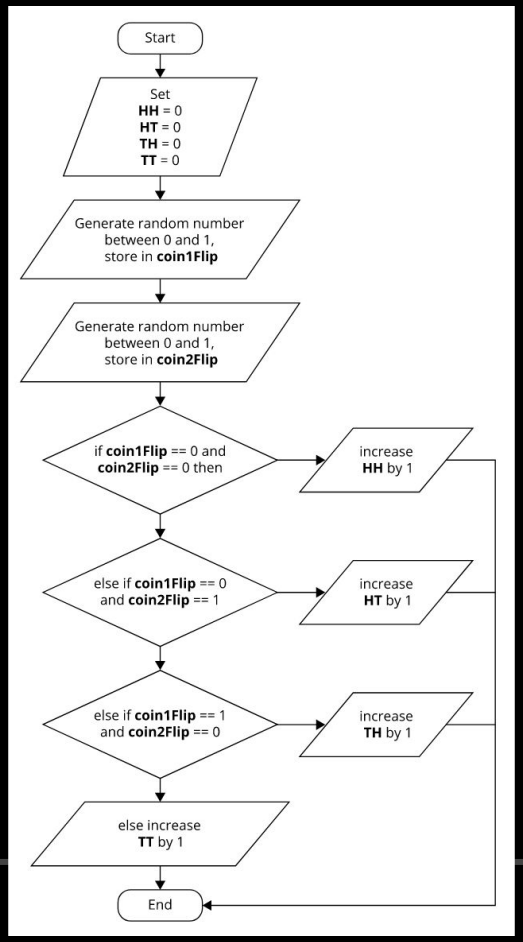
Grade 4 - Nested events

Grade 5 - Conditional statements and other control structures

Grade 6 - Efficient code

Grade 7 - Events influenced by a defined count and/or sub-program

Grade 8 - Analysis of data in order to inform and communicate decisions



CURRICULUM PROGRESSION

Grade 1 - Sequential events

Grade 2 - Concurrent events

Grade 3 - Repeating events

Grade 4 - Nested events

Grade 5 - Conditional statements and other control structures

Grade 6 - Efficient code

Grade 7 - Events influenced by a defined count and/or sub-program

Grade 8 - Analysis of data in order to inform and communicate decisions

- **mathematical situations:**
  - creating, enlarging, and shrinking shapes (e.g., fractals)
  - performing transformations
  - determining the cost of a purchase
  - determining the surface area and volume of prisms
  - solving optimization problems
  - performing probability simulations

- **possible mathematical situations:**
  - making financial choices and other decisions involving purchases, loyalty programs, and currency exchange rates
  - applying proportional reasoning (e.g., use a sample of a schoolyard cleanup and apply an appropriate scale to make predictions about the entire cleanup)
  - simulating a race around a track that is a composite shape, and comparing race times and distances
  - applying geometric properties, including angle measures, and analysing data to strategize moves in sports (e.g., the best serve locations in volleyball and the best shot locations in basketball)
  - analysing real-life patterns (e.g., heart rate, sleep patterns, daily steps)
  - solving optimization problems involving surface area and volume

CURRICULUM PROGRESSION

GRADE 7 (2020)	GRADE 8 (2020)	GRADE 9 (2021)
STRAND C: Algebra *By the end of this grade, students will:*	**STRAND C: Algebra** *By the end of this grade, students will:*	**STRAND C: Algebra** *By the end of this course, students will:*
C3. solve problems and create computational representations of mathematical situations using coding concepts and skills	**C3. solve problems and create computational representations of mathematical situations using coding concepts and skills**	**C2. apply coding skills to represent mathematical concepts and relationships dynamically, and to solve problems, in algebra and across the other strands**
C2.4 solve inequalities that involve multiple terms and whole numbers, and verify and graph the solutions	**C2.4** solve inequalities that involve integers, and verify and graph the solutions	**C2.1** use coding to demonstrate an understanding of algebraic concepts including variables, parameters, equations, and inequalities
C3.1 solve problems and create computational representations of mathematical situations by writing and executing efficient code, including code that involves events influenced by a defined count and/or sub-program and other control structures	**C3.1** solve problems and create computational representations of mathematical situations by writing and executing code, including code that involves the analysis of data in order to inform and communicate decisions	**C2.2** create code by decomposing situations into computational steps in order to represent mathematical concepts and relationships, and to solve problems
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	**C3.2** read and alter existing code involving the analysis of data in order to inform and communicate decisions, and describe how changes to the code affect the outcomes and the efficiency of the code	**C2.3** read code to predict its outcome, and alter code to adjust constraints, parameters, and outcomes to represent a similar or new mathematical situation

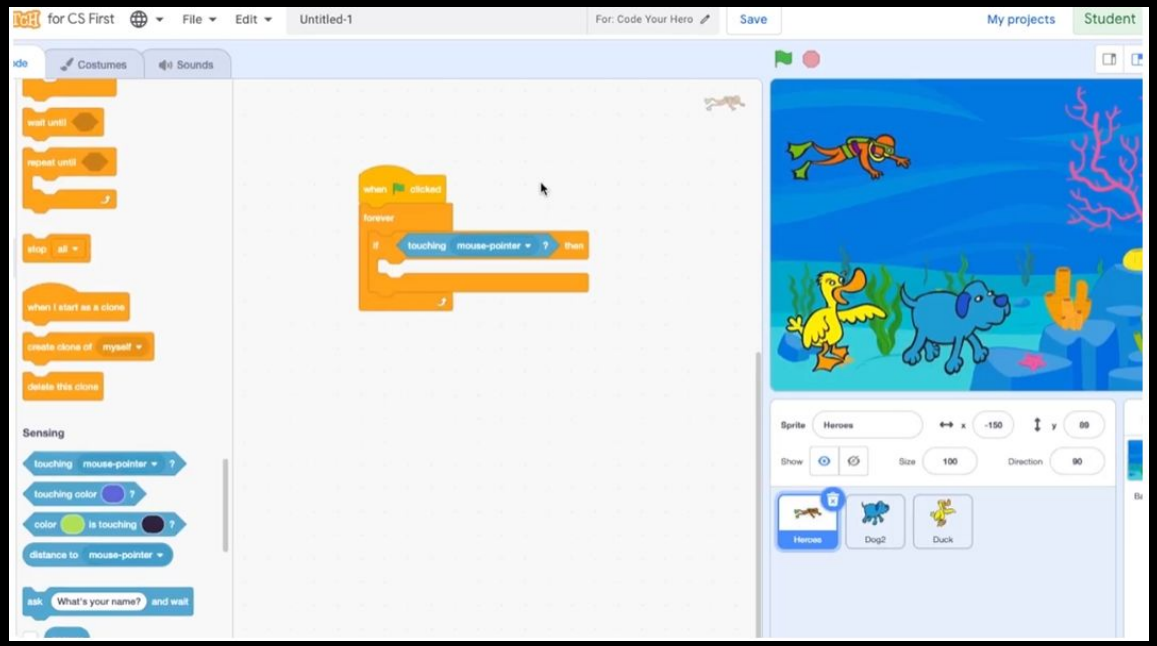
CODING EXAMPLES

#vrcodingrobotics

Photo Credit Typorama App

CS First

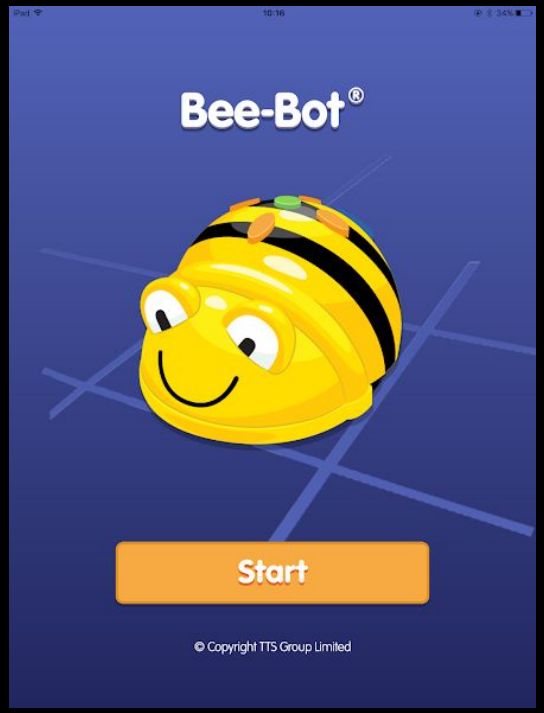
- "A computer science curriculum that makes coding easy to teach and fun to learn."
- Free of charge
- Learning through video tutorials and block-based coding in Scratch

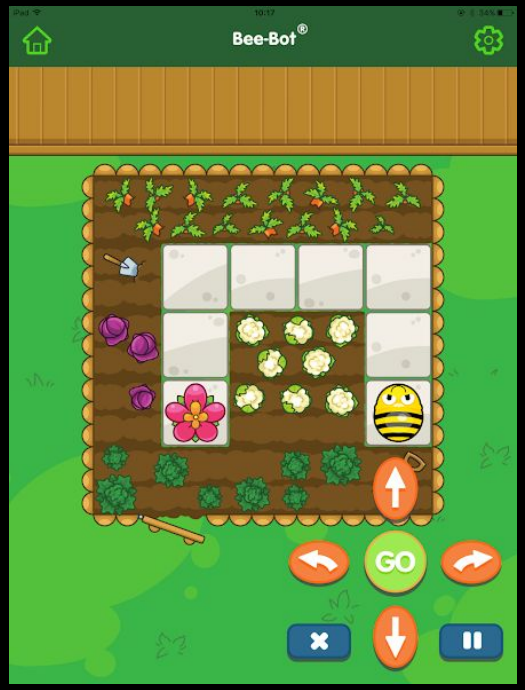


https://csfirst.withgoogle.com/s/en/home#

Bee-Bot App

- Based on the floor Robot.
- Directional language, programming sequences of forwards, backwards, left and right 90 degree turns.
- Available on the Apple and Google Play stores.



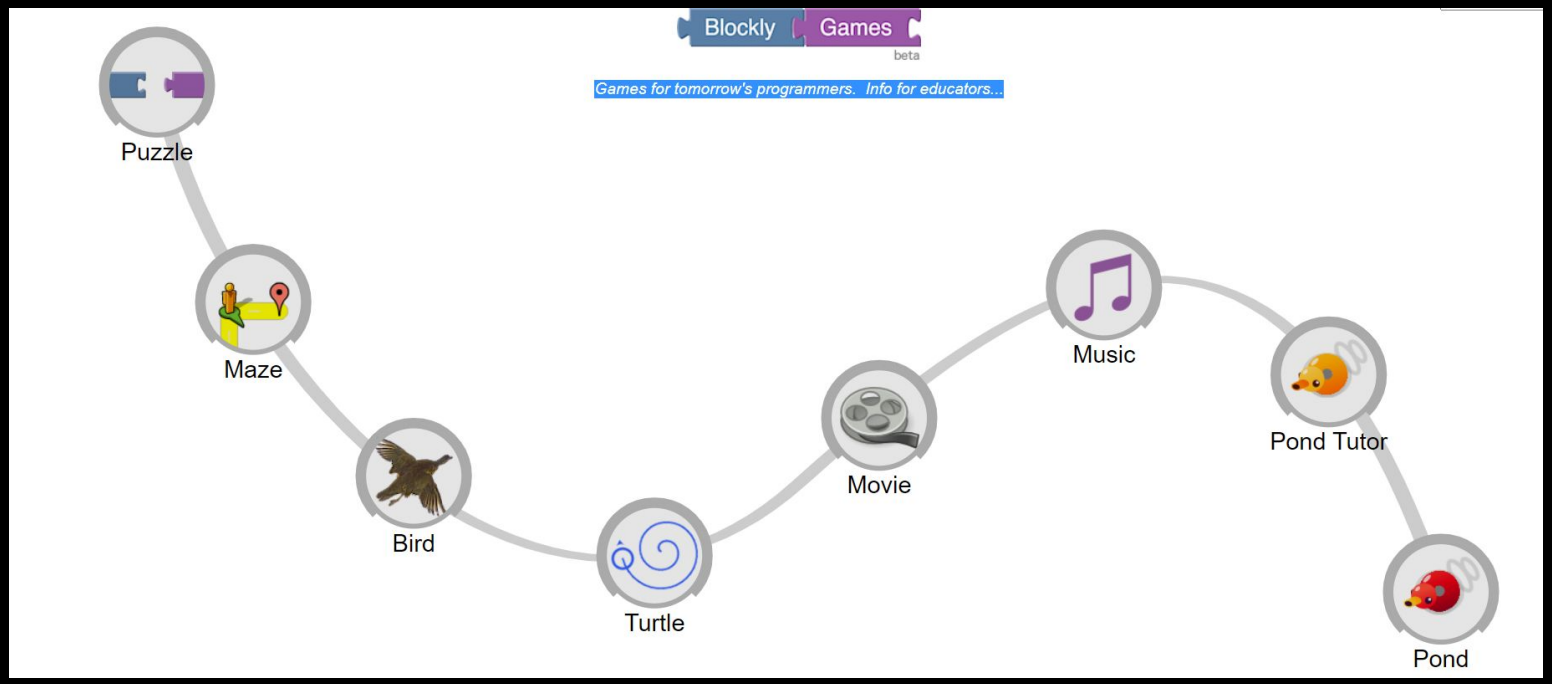


MakeCode Arcade

- Web-based coding platform using Blocks and JavaScript
- Creating and modifying retro-style arcade games
- Focus on connection between coding and gaming may be more engaging for some students

Blockly

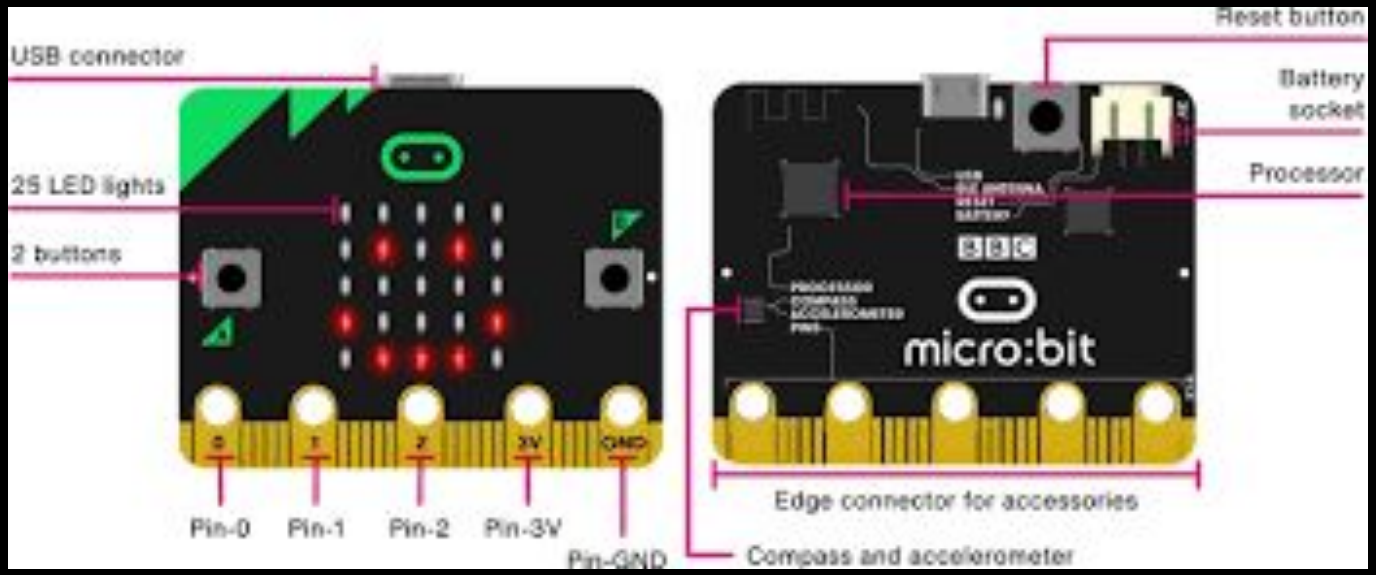
- Teaches kids to code through a series of block programming puzzles.
- Drag and drop blocks of code to solve specific challenges.
- Great tool for learners who have not had any prior experience with computer programming.



https://blockly.games/

micro:bit

- Block-based coding similar to Scratch/CS First
- Platform can be used with and without physical micro:bits
- Physical micro:bits include sensors for movement/motion (accelerometers), temperature, light, and touch



Kodable App

- Play and create games that transition from learning how to think like a programmer all the way to writing real code using a custom, built-for-kids coding interface.
- Sequence/Order of operations, algorithmic operations, conditional logic statements, syntax, classes & subclasses, properties, methods, and variables including strings, integers, & arrays.
- Available on the Apple and Google Play stores (in-app purchases and subscriptions available)

Run Marco!

- An adventure game through learning to code.
- Uses visual instructions to guide Marco or Sophia through a series of levels as they try to discover themselves.
- Learn how to think as a software developer one step at a time.
- Instructions in the form of the standard visual programming language “Blockly”, which is the same used by the official Hour of Code tutorials.
- Run Marco! is recommended on the official website for Hour of Code.
- Offered in multiple languages.
- Available in the Apple Store.

Code To Learn

- TakingITGlobal Project that is funded by the CanCode Initiative of the Canadian Government until March 31, 2024.
- No cost for Canadian schools or educators.
- Free LYNX coding software, micro:bits, Climate Action Kits, and more.
- Support in diverse subject areas.
- Offer workshops.
- NOTE: LYNX is available at NO Cost to Canadian schools in Canadian English, French, Ojibway, Oji-Cree, Mohawk, and Mi'Kmaw (with more Indigenous languages in development.

CODE to LEARN
PROGRAMMER pour APPRENDRE

https://sites.google.com/view/lynxcoding-org/cancodetolearn/about?authuser=0

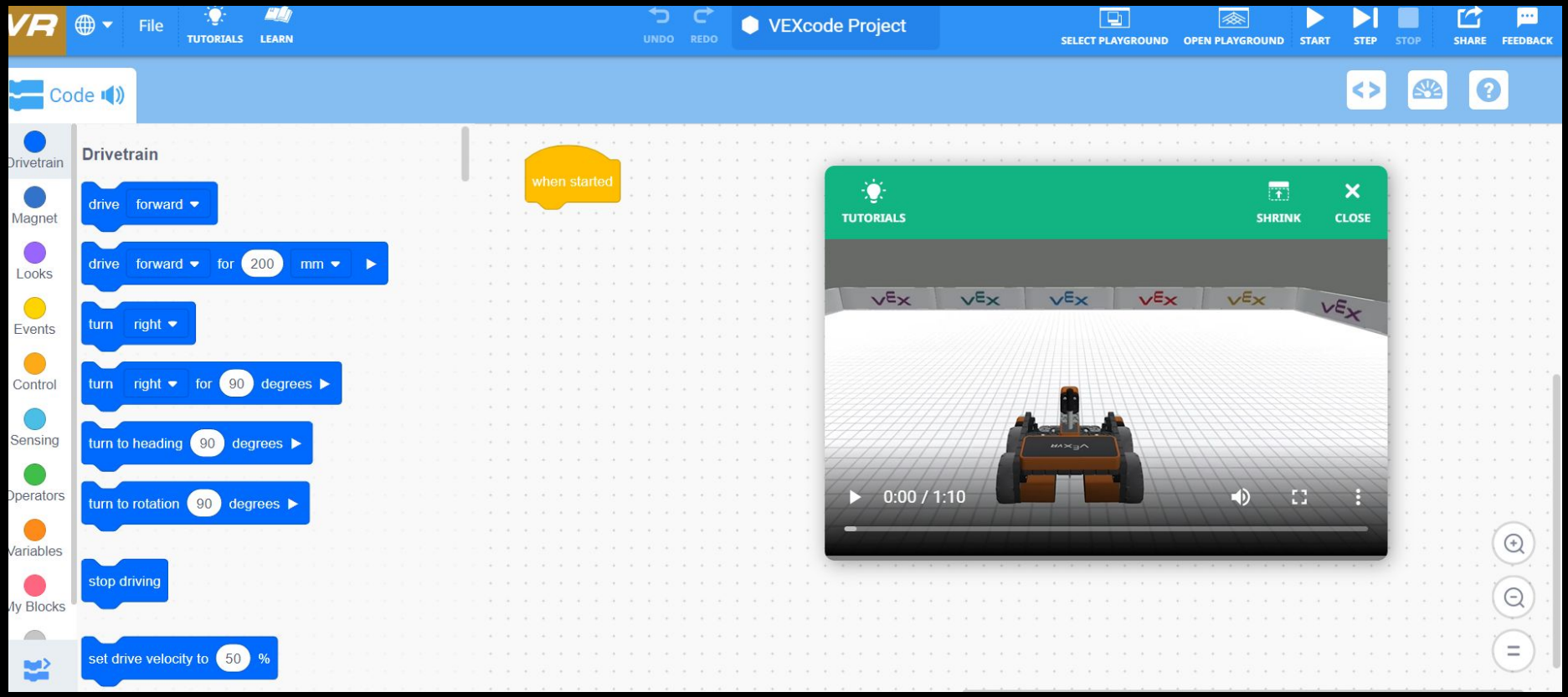
VEX VR

What is VEX VR?

- VEXcode VR (Virtual Robot) created for those that don't have access to a physical robot (VEX Robot).
- Web-based tool providing computer lessons.
- No software installations required.
- Program currently has 15 FREE playgrounds. Schools/school districts can opt into a subscription model to unlock other aspects.
- Functions on all major desktops and tablets.
- Lessons can be found at https://education.vex.com/vr/
- Virtual Robots.
- Virtual Playgrounds.
- Challenges and activities (some in Google Docs for editing or modifications). **https://vr.vex.com/**

https://www.techlearning.com/news/vex-robotics-launches-free-virtual-platform

Meet VEX VR



Meet The VEXCode VR Robot

#vrcodingrobotics

https://vr.vex.com/

Block Shape	Description	Block Examples	
Hat blocks	Start a stack of blocks and are shaped to attach blocks below them.	when started	
Stack blocks	Perform main commands. They are shaped to attach above or below other stack blocks.	move robot pen down	drive forward for 200 mm turn right for 90 degrees
Boolean blocks	Return a condition as either true or false and fits inside any blocks with hexagonal (six-sided) inputs for other blocks.	LeftBumper pressed?	wait until LeftBumper pressed?
Reporter blocks	Report values in the form of numbers and fits inside any blocks with oval inputs for other blocks.	distance in mm	round distance in mm
C blocks	Loop the block(s) within them or check if a condition is true or false. They are shaped to attach stack blocks above, below, or inside them.	repeat 10	if LeftBumper pressed? then turn left for 90 degrees

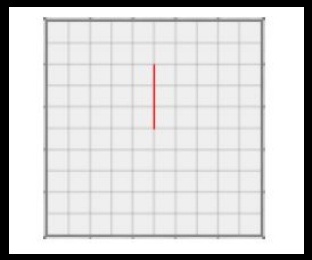
VEX VR Activities

VEX VR - ACTIVITIES

#vrcodingrobotics

https://vr.vex.com/

Intro + Fundamentals
- Distance Drive Challenge -



Playground: Grid Map

Challenges:

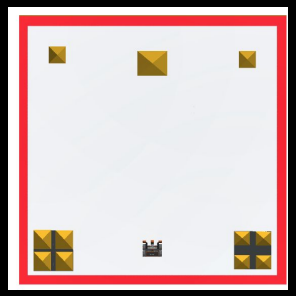
Level 1:
Program VR Robot to drive forward 3 grid squares.
How far should you program the VR Robot to move if you want it to travel 3 grid squares?

Level 2:
Program VR Robot to drive forward 6 grid squares, turn around, & drive back 6 grid squares.
How many degrees did you have to turn to face the way you came?
What is the total distance traveled?

Level 3:
Program VR Robot to drive in a square. Make each side of the square 3 grid squares long.
What is the area of your square in mm? Use a calculator and/or pencil and paper to find out.

Helpful Hints:
- Each square in the Grid Map measures 200mm by 200mm.
- Try changing the distance to program the VR Robot to move shorter or farther distances on the Drive for block from the Drivetrain category.

Developing Algorithms
- Crash The Castle -



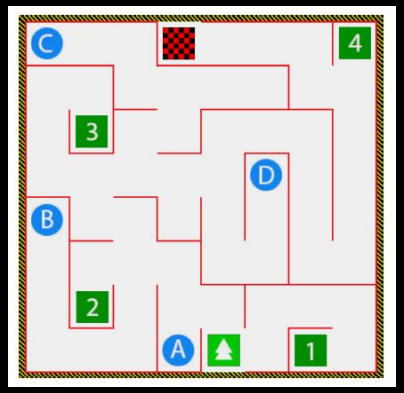
Playground: Castle Crasher

Challenges:
Level 1:
Create an algorithm to knock over all five buildings in one Castle layout using the VR Robot.

Level 2:
Improve your algorithm to knock over all buildings in three different Castle layouts without any project modifications.

Level 3:
Improve your algorithm to knock over all buildings in five Castle Layouts in eight minutes or less without any project modifications.

Detecting Walls From A Distance
- Number Maze -



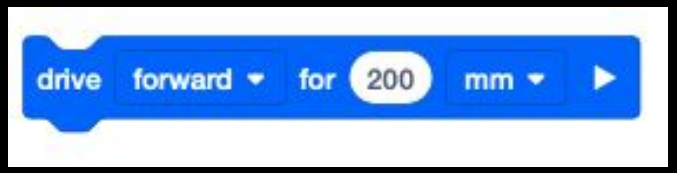
Playground: Dynamic Wall Maze

Challenges:
Level 1: Program the VR Robot to move through the maze from Start (green square) to Finish (checkered square).

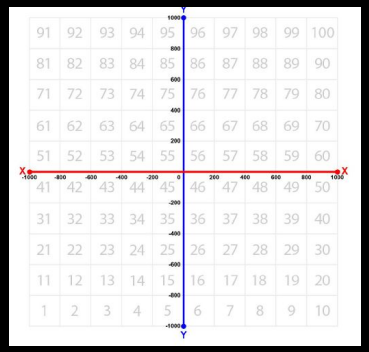
Level 2: Program the VR Robot to move through the maze stopping on each of the numbers in order (1, 2, 3, 4) and then traveling to Finish. The VR Robot should stop on each number for 2 seconds.

Level 3: Try it in reverse! Program the VR Robot to move from Start, to 4, 3, 2, 1, then Finish. The VR Robot should stop on each number for 2 seconds.

Helpful Hint:
- **Try using the Drive for block from the Drivetrain category. This block moves the Drivetrain for a given distance.**



Knowing Your Location
- Coordinate Numbers -



Playground: Number Grid Map

Challenges:

Level 1: Program the VR Robot to move to the coordinates (-500mm, 700mm). On which number does the VR Robot stop?

Level 2: Look at the following pairs of coordinates. Estimate which number the VR Robot will touch after traveling to these coordinates in sequence. Use the position of block to move the VR Robot to check your answer.

- ➔ (-300 mm, -900 mm)
- ➔ (700 nm, 700 nm)
- ➔ (-100 nm, 900 nm)
- ➔ (500 nm, -300 nm)

Level 3: Estimate the coordinates of the following numbers. Then, use the position of block to move the VR Robot to check your answer.

- ➔ 22, 38, 64, 85

Helpful Hints:

- Each square on the Number Grid Map measures 200mm by 200mm.
- Having a problem finding your position? Check the VEXcode VR Dashboard.
- Use the position of block from the Sensing category to program specific locations

ASSESSMENT & EVALUATION

- In Ontario, assessment, evaluation, and reporting policies can be found in Growing Success and in the Mathematics Curriculum Grades 1-8 (2020).
- The primary purpose is to improve student learning.
- Assessment is the process of gathering information that accurately reflects how well a student is achieving the curriculum expectations in a grade or course.
- Assessment for the purpose of improving student learning is seen as both “assessment *for* learning” and “assessment *as* learning’.

Photo Credit www.pexels.com

ASSESSMENT *FOR* / *AS* LEARNING

Assessment For Learning

- Provide students with descriptive feedback
- Provide students with coaching for improvement

Assessment As Learning

- Helping students develop capacity to be independent, autonomous learners who are able to set individual goals, monitor their own progress, determine next steps, and reflect on their thinking and learning.

Photo Credit www.pexels.com

EVALUATION
- ASSESSMENT *OF* LEARNING -

- Process of judging the quality of student learning on the basis of established performance standards, and assigning a value to represent that quality.
- Evaluation is based on assessment *of* learning that provides evidence of student achievement at strategic times throughout the course, often at the end of a period of learning.
- Determining a report card grade involves the interpretation of evidence collected through observations, conversations, and student products, combined with the teacher's professional judgement and consideration of factors.

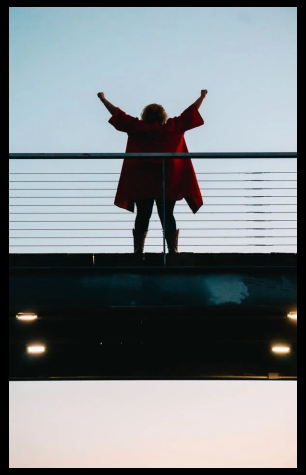
Photo Credit www.pexels.com

EVALUATION
- ASSESSMENT *OF* LEARNING -

- All overall and specific expectations begin with: "By the end of grade …, students will…"
- Students should have multiple opportunities throughout the year to engage in math learning with and about coding
- Students' achievement should not be penalized for early not knowing

Photo Credit www.pexels.com

ACHIEVEMENT CHART



Knowledge and Understanding

- Subject-specific content acquired in each grade or course (knowledge), and the comprehension of its meaning and significance (understanding)

Thinking

- The use of critical and creative thinking skills and/or processes

Communication

- The conveying of meaning and expression through various forms

Application

- The use of knowledge and skills to make connections within and between various contexts

Photo Credit www.pexels.com

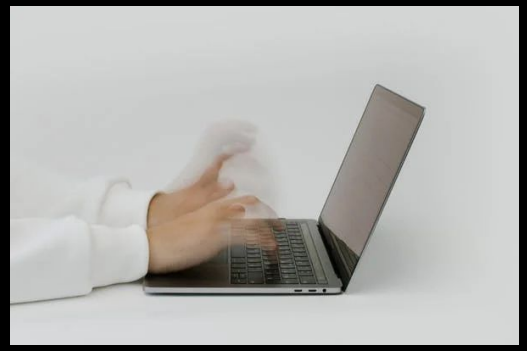
STUDENT ASSESSMENT WITH CODING

With reference to [Teaching Coding Tips: Student Assessments]

- Temptation to assess a student's code line-by-line, which can be time consuming and not an effective way to monitor students' progress.
- Recommendation to assess coding that "go beyond traditional tests and assignments and take in the bigger picture of your students' learning".
- "Coding isn't just about the code itself – it's also the process, and the skills your students develop such as problem-solving, resilience, and creativity."

Photo Credit www.pexels.com

STUDENT ASSESSMENT WITH CODING - EXAMPLES -



- Assessment Criteria Charts
- Peer Review
- Self-Reflection
- Student(s)-Teacher Mini Conferences
- Formative Assessments
  - (questionnaires, multiple choice questions, writing a few lines of code, writing the next line of code, labelling diagrams, etc.)
- Online Resources

Photos Credit www.pexels.com

SUPPORTING RESOURCES

Overall Expectation

Elementary curriculum > Mathematics > Grade 7

C3. Coding

Specific Expectations

By the end of Grade 7, students will:

Coding Skills

C3.1 solve problems and create computational representations of mathematical situations by writing and executing efficient code, including code that involves events influenced by a defined count and/or subprogram and other control structures

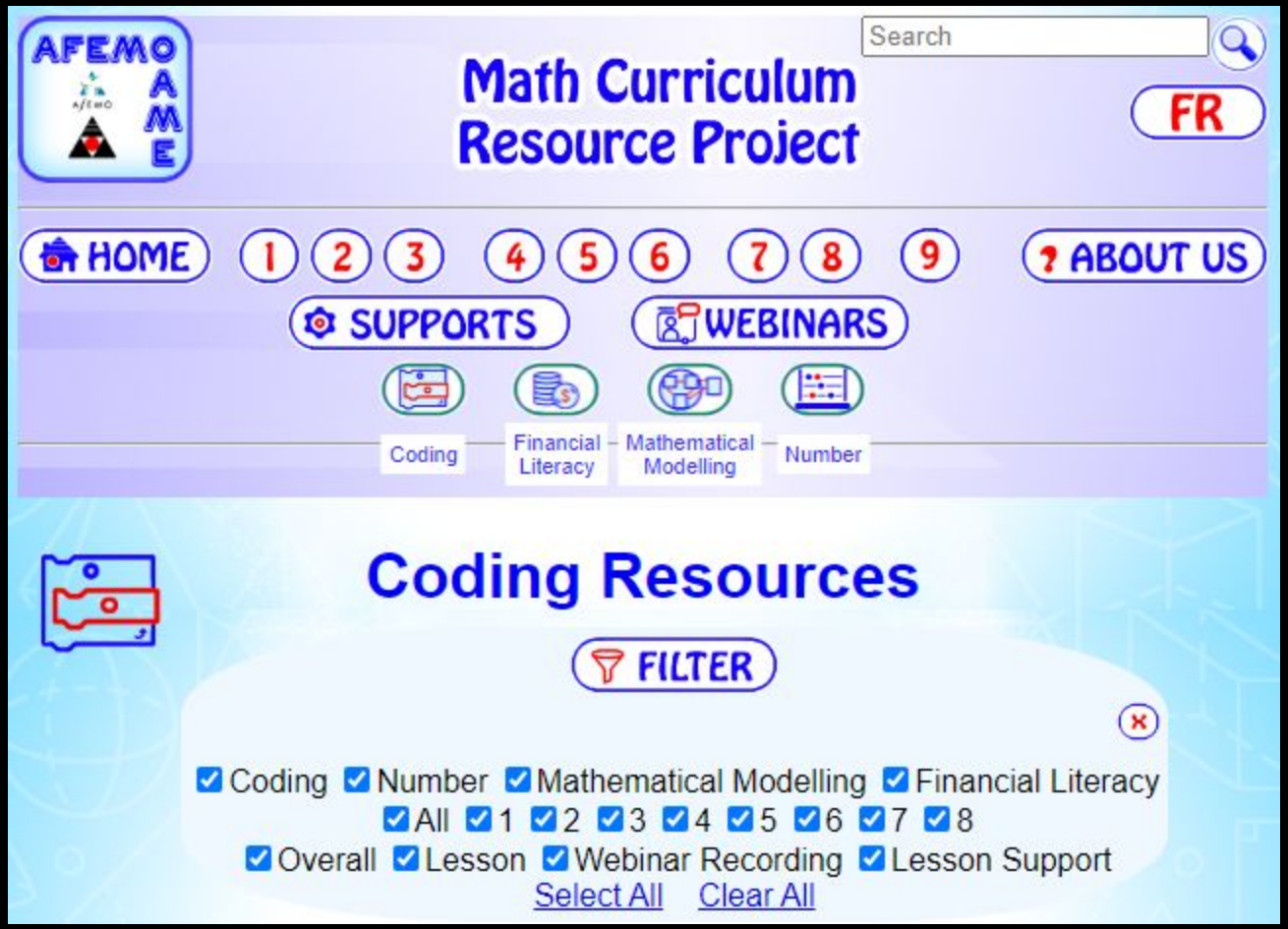
Teacher supports

Examples

Key concepts

Sample tasks

SUPPORTING RESOURCES



SUPPORTING RESOURCES

tvoLearn

NEW Kindergarten Grades 1-8 Grades 9-12

Learning Activities

Learning Activities provide opportunity for deeper exploration of a subject. Organized by grade and topic (or strand), students should be guided through each Learning Activity by an adult. Before clicking on a topic to prepare for or begin this guided instruction, be sure to read these helpful tips about how to get the most out of TVO Learn.

Number | Algebra | Data | Spatial Sense | Financial Literacy

Students continue to develop their understanding of patterns, including those that involve integers. They use Algebraic notation, such as, s = d/t, to represent the relationship between speed, distance and time. They solve Algebraic equations involving multiple terms, integers and decimal numbers. Students will write code to create a line or curve that falls between the greatest number of data points. They also use modelling for real-life situations, such as making predictions about future fundraisers based on the funds raised from past fundraisers.

Learning Activity 1
Extending Patterns

Learning Activity 2
Representing Patterns

Learning Activity 3
Algebraic Expressions and Patterns

Learning Activity 4
Making Predictions

Learning Activity 5
Algebraic Expressions

Learning Activity 6
Solving Algebraic Equations

Learning Activity 7
Monomials and Binomials

Learning Activity 8
Interpreting Graphs

Learning Activity 9
Solving Equations with Decimals

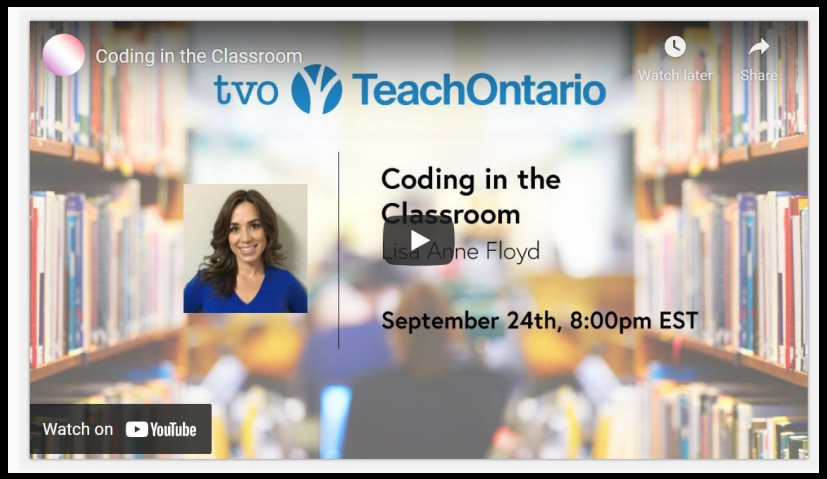
Learning Activity 10
Inequalities

Learning Activity 11
Graphing Inequalities

Learning Activity 12
Coding and Analyzing Data

TVO Digital Learning Outreach - Coding in the Classroom -

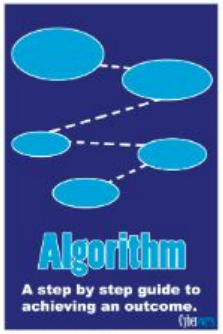
- Website with supports for educators to incorporate coding into their classroom.
- Main focus on Grades 1-8 and Grade 9 Mathematics.
- Pedagogical articles.
- Resources that focus on coding in Primary (Gr.1-3), Junior (Gr.4-6), and Intermediate (Gr.7-8 & 9).
- Great webinars that have passed that support coding in the classroom.



#vrcodingrobotics

https://outreach.tvolearn.com/codingintheclassroom/

Some Posters



Download image

Download image

Download image

Download image

Download image

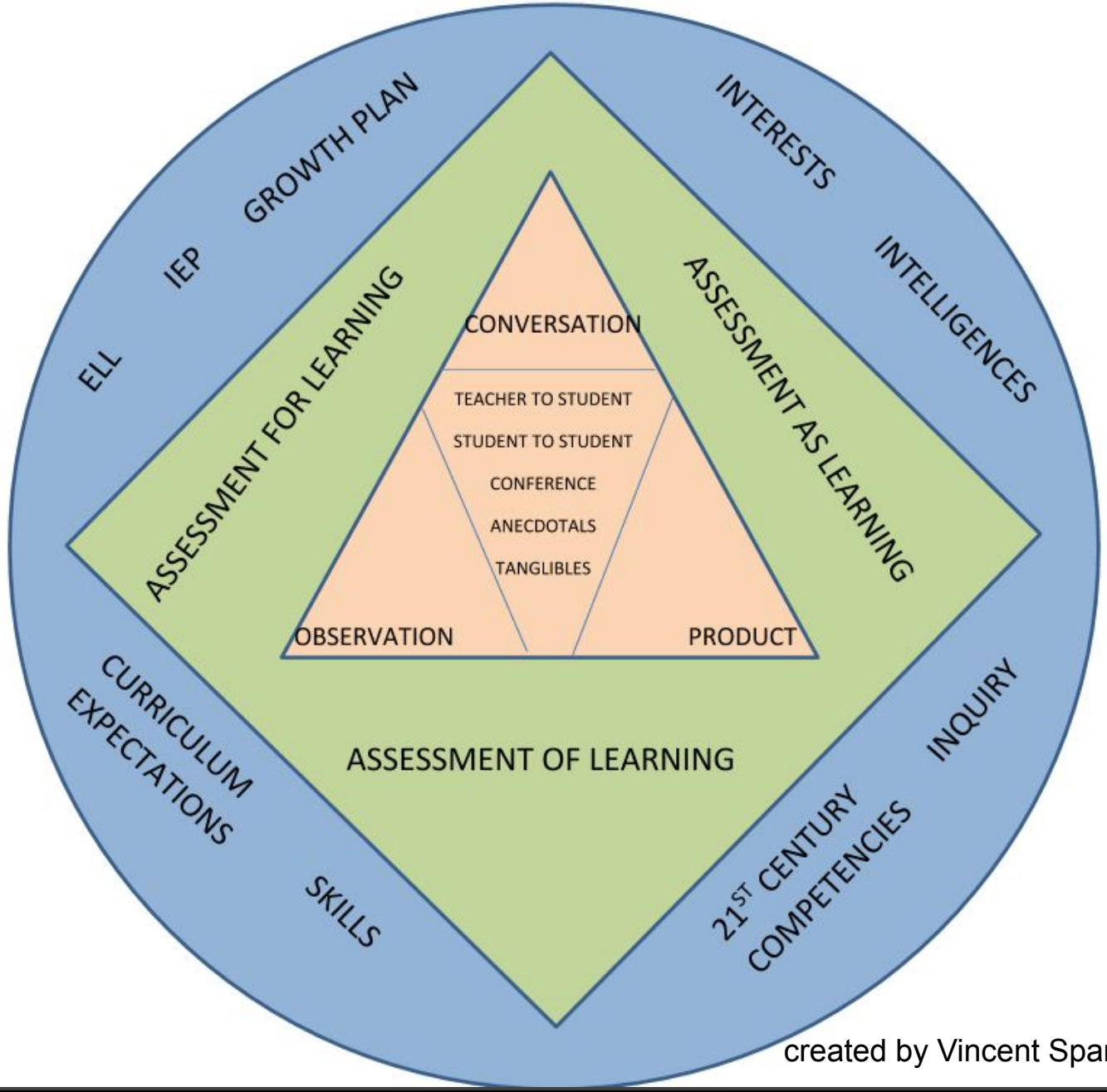
Download image

Download image

Download image

Download image

https://www.edugains.ca/resourcesTELO/CE/CodingInterface/Resources/Extra%20Resources/_content.html



created by Vincent Spano

Thank You!

Virtual Coding & Robotics

Michael Frankfort @mfrank_76

York Region District School Board

Michael.Frankfort@yrdsb.ca

Michael Minchella

York Region District School Board

Michael.Minchella@yrdsb.ca

#vrcodingrobotics

https://bit.ly/3https://bit.ly/3w7SGlvw7SGlv