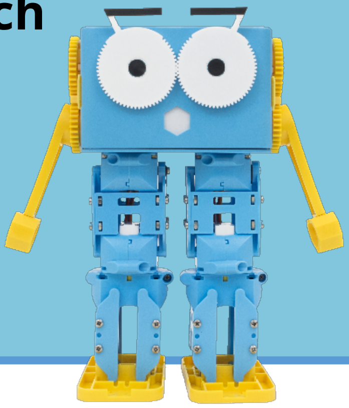
Lesson 1.3 – Getting Comfortable with Scratch



Education Level: Second Level (Ages 7-11)
Lesson Duration: 45 minutes

Prerequisite Knowledge: Lessons 1.1, 1.2
Device Compatibility: Laptop, PC or Tablet

Lesson Overview

Students have now had their first taste of programming using the Scratch editor. In this lesson, they will begin to become more comfortable with the interface being used and start to build up an understanding of where certain blocks can be found. As part of this lesson, we will be introducing a little bit of debugging and trying to get students to recognise when something isn't happening correctly and why it is important that we test out our programs frequently.

Learning Objectives

- Relate coding blocks and coding to real life situations
- Predict what outcomes different blocks of code will have on the world
- Recognise when the program has a bug or error
- Understand why the error is happening and how to fix it

Key Vocabulary

- Programming
- Block Coding
- Instructions/Recipe
- Testing
- Debugging

Resources & Equipment

- Marty Workbook (Lesson 3)
- Marty the Robot
- Access to computers/laptops/tablets
- Scratch editor linked to Marty the Robot

Additional Reading

- Educators Guide
- Introduction to Programming with Marty using Scratch

Learning Plan & Activities

1. Remind students of the last lesson where we introduced programming Marty with Scratch by showing some examples of blocks on the screen and asking students what they think would happen
2. Given a few examples, ask students to run a couple of small, pre-written, programs in Scratch and observe whether they do what they should be doing
   a. Encourage students to think about what the aim of the program is and how they would program Marty to meet it
   b. Ask students to highlight what the small problems are with the program(s)
3. Introduction to debugging and why it is important that we test our scripts out with emphasis from real-world situations/events
4. Ask students to go and fix the problem with the program
5. Repeat exercise with a couple of different small programs of debugging and remixing

Additional Challenges

- Encourage students to think about how they would improve these small programs and them to try implementing the ideas

Curriculum Benchmarks

Curriculum for Excellence – Technologies Benchmark Guide

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 1.3
Digital Literacy	TCH 0-01a	●
	TCH 1-01a	●
	TCH 2-01a	○
Technological Developments in Society and Business	TCH 0-05a	●
	TCH 1-05a	○
Craft, Design, Engineering and Graphics	TCH 0-12a	●
Computing Science	TCH 0-13a	●
	TCH 1-13a	○
	TCH 2-13a	●
	TCH 3-13a	○
	TCH 4-13b	○
	TCH 0-14a	●
	TCH 0-14b	○
	TCH 1-14a	●
	TCH 1-14b	○
	TCH 0-15a	●
	TCH 1-15a	●
	TCH 2-15a	○

National Curriculum – Computing, Design & Technology

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 1.3
Computing	1-a	●
	1-b	●
	1-c	●
	1-e	○
	2-a	○
	2-b	●
	3-a	○
Design & Technology	1.1-b	○
	1.3-b	○
	2.3-c	○

Australian F-10 Curriculum – Digital Technologies, Design & Technologies

● = Fully Addresses Benchmark ○ = Partially Addresses Benchmark

Curriculum Organiser	Benchmark Covered	Lesson 1.3
Digital Technologies	ACTDIK001	●
	ACTDIK002	●

	ACTDIK003	○
	ACTDIK004	●
	ACTDIK008	○
	ACTDIP009	○
	ACTDIP010	●
	ACTDIP011	○
	ACTDIP013	●
	ACTDIP019	○
	ACTDIP020	○
	ACTDIP027	○
	ACTDIP029	○
Design & Technologies	ACTDEK001	○
	ACTDEK004	○
	ACTDEP006	●
	ACTDEP009	●