

Marty Teacher Guide – Lesson 1.5

Second Level

PROGRAMMING SHAPES

Getting students to think about how they would program a robot to walk around in different shapes not only gets them to think about left and right turns but also the angles involved with different shapes.

- Start off with simple shapes like a square or rectangle
- Then move on towards more complex shapes by getting students to draw the shapes out and think about the angles before programming each other/Marty to walk around in that shape

LOOPS/REPEAT STATEMENTS

After programming Marty to walk around in different shapes, prompt students to look at their code or list of instructions to find patterns. Hopefully, they will notice a lot of repetition which is where we come in with the concept of loops to reduce repetition and get shorter and more efficient code.

- What patterns can you see in your code blocks/list of instructions?
- (After showing the code as a loop) Why do you think we would use something like this instead for these kind of code blocks/instructions?

Marty Teacher Guide – Lesson 1.6

Second Level

EVENTS

In this lesson, students will be asked to think about the events that their program waits for to happen before running code. They have already subconsciously been using events programming when they have been using Scratch.

- How does the computer know when to run your blocks of code?
- What kind of events have we used already with Scratch?
- What do you think would happen if we didn't have these events? What would happen if all of our code tried to run one after the other without waiting for an event? Would our program be doing what we want?

BUILDING A REMOTE CONTROL PROGRAM

Students will now have been tasked with creating a remote control program so that they can give Marty instructions by clicking on different button sprites in Scratch. Get students to think about the following things,

- What things would you like Marty to do? Basic movements like walk forwards?
- Don't forget to regularly test your programs to make sure each button works individually
- What other movements or actions could we add onto our remote?

Marty Teacher Guide – Lesson 1.7

Second Level

PARALLEL PROGRAMMING

So far, Marty has only done one thing at a time but in this lesson, we will explore programming Marty to do two things at once. For example, we will program him to lift both of his arms. Getting students to compare how they move to how Marty moves is really important to allow them to understand the limits of the things that we can do in unison.

- How do you think we would program a sprite to move forwards by 10 steps and say hello at the same time?
- How do you think we would program Marty to lift both of his arms up at the same time?
- Do you think we could program to do any two things at the same time?
- Are there any two moves that you couldn't do at the same time? Why?
- What other movements could we program Marty to do at the same time then without him falling over?

Marty Teacher Guide – Lesson 1.8

Second Level

VARIABLES

Students will be introduced to the concept of variables as a way of remembering a value or a message for us, so that we don't need to remember them. Students should be aware that variables can be updated and how this changes what is stored.

- Why do you think having variables is important?
- What do you think we can use variables for in our programs?
- If I have a variable holding the number 2 and then assign the number 6 later on and afterwards print out what that variable is holding, what will be printed?

GETTING INPUT FROM THE USER

Now that we have created a counting program with variables, we are going to get user input in our programs to help us make decisions.

- What do we mean by user input? Can you think of what kinds of user input we have had in our other programs? Like the remote control program?
- What blocks do you think we will have to use to ask our user questions and get them to respond?

Marty Teacher Guide – Lesson 1.9

Second Level

MARTY FEATURES

Students have been using Marty for quite a few lessons now and will have started to build up an idea of the different things that he can do.

- What do you think we use robots for?
- Can Marty do any of these? Do you know any names of robots that could do some of the things we spoke about?

IF STATEMENTS

We introduce if statements in this lesson and relate them to decisions that we make in real life every day. Getting students to relate programming concepts to everyday things is a really good way to make abstract concepts a lot easier to understand and get to grips with.

- [Given an example of an if statement like the crossing road example] What other decisions do you make that could be translated into an if statement?
- Why do you think we need if statements? What do they help us with?

MARTY MAGIC 8

Using if statements, students will now make a program that will select a random number which will let Marty know what moves to do – a robot take on a magic 8 ball.

- How can we generate and store a random number? What blocks? What do we need to use to store the number?
- How do we decide which action to do? Can we use if statements?

PROGRAMMING MARTY TO RESPOND TO USER INPUT

We have already discussed user input in the last lesson by getting students to take in a name from the user. A challenge extension of this lesson is to create a program that asks the user questions, they type in an answer and then Marty responds based on this. We need to highlight to students that this will need to be simple questions because we will need to check the answers using if statements.

- Think about the kind of questions to ask. Simple things are easier to use like, *Do you like cake?*
- Think about what happens if the user doesn't say yes or no? What feature of if statements can we use? [Hint: the **else** part of the if statement]