Pīkau Name: Programming with Sequence and Output: CT PO2

Video Name: Inputs and outputs (EMP06-6)

Presenter: Tim Bell

When I was at school I remember in a woodworking class that we made mallets. To make a mallet you cut out the head and you put a hole in it, then you cut out a handle and you shape it so it will fit inside the hole. Then you put them together, but you have to bang them together. How do you bang them together? With another mallet. So here I was banging one mallet together with another one thinking 'Who made this mallet?' and 'Did they use another mallet for it?' and 'Is it mallets all the way down?' and 'How do you even start making mallets?'

Maybe I was overthinking my woodwork classes, but the same thing happens in computing. Even with a really simple system like the little robots that we have been programming, they have a programming language with forward and left and right and so on, but when you are pressing the buttons for that there is a computer inside it that's going to be reading in your program, and running your program. But the computer inside, of course, is running a program to run your program. Someone wrote that program inside the robot. When they wrote that program they used a programming language, and someone had to write that programming language, and, is it programming languages all the way down? We don't need to get into that amount of detail yet but what I want to point out is even these really simple devices give us an awful lot to think about in terms of computing and how this thing called programming actually happens.

The language of Progress Outcome 2 talks about sequences and outputs. Progress Outcome 3 introduces the idea of inputs. The simple devices we have been using look like they have got inputs but we are going to look at that in a little bit more detail, because that's going to help us think about exactly what programming is about. Actually, nearly all software has inputs and outputs. It's hard to imagine a program or an app that doesn't receive some kind of input from the user and give some kind of output that they [the user] receive from it.

Inputs are any way of getting information from the outside world into a computer program. It could be as simple as a key on a computer keyboard, or a button in an app. It could be a camera that's providing video input to a computer program. It could even be shaking a device. It could be a sensor that is detecting if there is a car parked in a particular place. The input doesn't have to come from a human - it could measure the GPS location of the device, or it could sense a wall, or it could measure the temperature outside.

Outputs are any way that a computer can communicate with a human. It could be on a screen, but it could also be music coming through headphones or speakers. It could be an alarm. It could be a physical output like vibration on a device. The output of a computer could also be things like switching on a heater, or switching on a motor. Being able to show a path and end up at a point on a grid is a surprisingly rich form of output given the simplicity of the program. You could have numbers, letters, even sentences as output. If the robot pauses or spins on a square you could use that to tell the viewer that they should note that as an output. You could spell out a word by navigating a grid of letters.

But do these bots have input? Well, the buttons on top look like inputs, but of course they are just writing the program. Once you press the go button you have no control over what the robot is going to do, it's not going to receive any input from the outside world. This makes more sense when you separate the roll of the programmer and the person who is going to use the program. The programmer writes a website or an app that someone downloads - hopefully many people are going to use their program, but the person who is actually using it has no control over the program: they are stuck with the program that the programmer wrote.

If these bots were used commercially, it would look like this:

User: Hi, I'd like my bot to go round in a square please.

Programmer: No problem, I'll use my programming skills to do that.

User: Thanks!

Programmer: Yes! Okay, I've written and tested a program and loaded it into your bot.

User: Great!

Programmer: That will be $20 thanks.

So you can see that the person running the program can't provide any input once the program is written. If they wanted to control the size of the square that it draws they would have to go back to the programmer and ask them to write a new program.

But the buttons on here still look like input, so what's with that? Well, when you are writing a computer program with them, it's running on a computer program! Wow! How are we going to understand that? Well, if you look inside one of these devices (now, don't try this at home or show this to your kids at school), but inside here is a small computer and it's running a program. The program in here is the one that is receiving the input from the buttons that you're pressing. Then, when you press the go button, the program running on this chip thats inside the device starts following the commands that you've typed in.
So that raises the question: who wrote the first computer program? It's the same problem that I had with the mallets. The first computer programs were written by flicking electrical

switches. After a while people realised that that wasn't very effective and they started writing languages that were based on text so that people could type in their programs. But, of course, to make those new programs for receiving text programs they had to write a program by flicking the switches.

These simple bots with just a few commands actually raise some really interesting questions about what programming is. We don't expect you to explore those kind of questions with your students but we did want you to be aware that you are laying the foundation to engage your students with some very deep concepts.