

Hello, I'm Neil Rickus and welcome back to another part of using tablets to teach programming in the primary classroom. During this session, we'll finish off looking at apps to enable us to teach the new computing curriculum, then examine how to use the apps in the classroom, along with appropriate pedagogy and finally assessment. Before we move on, just a reminder all the apps discussed during this session can be found on this grid at computingchampions.co.uk/mooc. So, to the iPad...

We'll first look at Hopscotch, which allows us to cover all aspects of the 7-11 computing curriculum. I start by creating a project and choosing a sprite and, at a basic level, I can sequence instructions linked to an events. For example, I'm going to execute two of the built in instructions, "wiggle" and "dance", when I press the play button. I drag the instructions into the middle area, press play and off she goes! However, as pupils' programming ability should now be progressing, they're expected to begin using features such as repetition and variables, so I'm going to use Hopscotch to produce an Octagon and to count the number of sides drawn. As the program to achieve this involves a number of sections, I also have to begin decomposing the problem into smaller parts. Firstly, I drag a repeat block across, which will execute 8 times, then leave a trail for 100 pixels (little dots on the screen), in addition to a turn of 45 degrees. I'll press the play button to show you the result so far. I'll now include a variable, which is a data store in the computer's memory, to count how many sides have been drawn. In Hopscotch, I initially set the value of my variable, which I'll name Sides, to zero, then increase its value by one after every turn. I'll also make sure I display the value to the screen, so you can see the value of the variable increasing. Just by looking at the instructions here, it's evident our programs are becoming longer and more complex. I'll press play so you can see the final result. I could then improve my program further by adding other sprites and producing more detailed images, such as in this example.

The final app we're going to look at is Tynker, which is available for both iPad and Android. Tynker uses an interface similar to the popular, desktop based, Scratch, and has both "Play" and "Create" modes. Within create mode, our sprites have a number of characteristics, as seen here, which can be altered to customise our programs. With regards to the instructions I've entered for this sprite, I've used repetition in the form of a forever loop, which means the instructions will continue for as long as the program is running. I'll press the play button so you can see how this program in action before I add further complexity. Within this program, I can now add in selection, or choices, through the use of "if...then" statements. I'll add a nice background first, then an additional sprite, such as the Pink Monster. Next, I'll add an additional command to say, "I'm waiting", then change the text to, "I'm bored!" after five seconds. So, within a forever loop, I'll create a variable named "time", initially set it to zero, then increase its value every second. I'll then use selection and say, if the time is equal to 5 seconds, say, "I'm bored", else, say, "I'm waiting". Let's press the play button to see it in action. You may have also noticed the Tynker interface can respond to touch, so I also had some code to change the sprite's costume when I tap him, which you can see here. As pupils progress beyond primary school, Tynker can also be used for parts of the computing curriculum at secondary school level. For example, it allows pupils to enter functions, which is a set of instructions used repeatedly throughout a program, and Boolean logic, through statements such as AND, OR or NOT.

So, when using these apps within the classroom, it's great if you can demonstrate the app to pupils before they begin work. The content of an iPad can be displayed on an Interactive Whiteboard using an Apple TV or software, such as Air Server, while Android tablets can use a Chromecast. Alternatively, a VGA or HDMI cable with an appropriate adaptor can be used.

It's worth bearing in mind the app doesn't have to be used for the whole lesson, thus it could be used for a short period of time to introduce or reinforce a concept. It is therefore vital you're clear on your lesson's objectives and you're not simply, "using Hopscotch today". Where possible, differentiate your session content, such as through a pupil's starting point in the app, or perhaps with a different app entirely. Using a scheme of work can assist with differentiation and help to identify suitable progression paths for pupils.

A scheme of work can also provide a range of pedagogies to use during your sessions. For example, the recently published Quickstart Computing primary.quickstartcomputing.org identifies six techniques for use in computing lessons and emphasises the importance of revisiting and practicing concepts over a period of time, in addition to sharing experiences with others.

One of the problems with implementing a new curriculum, as we are currently doing in the UK with Computing, is that assessment methods are often not fully defined. The UK government funded Computing at Schools (CAS) computingatschool.org.uk provides a set of Progression Pathways, which help identify pupils' understanding of different strands of computational thinking. For example, the screenshot here shows the initial pupil progression for algorithms. Alternatively, Makewaves makewav.es has a collection of online badges pupils can earn as their understanding develops, while Quickstart Computing again has guidance for primary teachers. However, one of the benefits of using tablets is screenshots can easily be recorded to provide evidence of learning. On most devices, this can be achieved by pressing the home and lock buttons at the same time.

So, that concludes our sessions on using tablets to teach programming in the primary classroom. I hope you've found the sessions useful and informative. Please don't hesitate to contact me if you have any questions about the content, or would like me to work further with you, such as working with your organisation to implement programming across the curriculum.