Sonic Pi

Lesson 1

The Art of Programming: Sequencing

Designed by
Sam Aaron & Carrie Anne Philbin

Introduction

The Raspberry Pi is a tiny computer less than the size of a pack of cards which can transform the way we perceive and approach computation. In this lesson we will introduce the basic components of the Raspberry Pi and how they relate to a traditional computer. We will discuss the generic nature of computation and how the same computer can be programmed to simultaneously do many different things - from word processing to music synthesis. Finally, we will introduce the most basic principle of programming - a program as a sequence of instructions.

Learning Objectives	**4.4 Computers**
	Computers are general-purpose devices
	Basic architecture: CPU, storage, input/output
	The idea of a program as a sequence of statements written in a programming language
	Changing the order of the statements changes the behaviour of the program

Teaching Progression

For the majority of the lesson, it is suggested that work is carried out in pairs. Each pair should have access to the standard equipment described below. In addition, is suggested that you have your own setup connected with a speaker for the demonstration sections.

Sonic Pi Equipment

- A Raspberry Pi with the Sonic Pi software installed per pair;
- A keyboard and mouse connected to the RPi per pair;
- A monitor connected to the RPi per pair;
- A headphone splitter connected to the RPi audio jack per pair;
- A pair of headphones connected to the splitter per pupil.

Equipment

- Computer program cards

Lesson Summary

- An introduction to the basic physical parts of a RPi
- A demonstration that the RPi can behave like a *traditional* computer
- A group exercise to act out a simple program.
- Starting the Sonic Pi application
- The first music program

Starter

Have a demonstration RPi already connected and the Sonic Pi software running. Hold up an RPi board and ask the students what they think it is. Explain that it's actually a computer and that in the coming lessons we're going to do something special with it. Instead of running apps and games other people have created for us, we're going to learn to write our own software - to make music. Start the demo code, play it for a moment or two and explain that in a few weeks they'll be able to make computers do this for themselves. Emphasise that they'll be free to do what they want with it and have a lot of fun in the process - programming is about getting the computer to do exactly what you want it to do. It's not important for the students to see the application or any code at this stage, just for them to hear the sounds coming from the computer and being modified by moving the mouse around.

Main/development

1. Start with all the parts of the RPi on a table: keyboard, mouse, speaker, memory card, power supply, monitor, monitor cable and RPi itself. Ask the class to name and describe each component as you connect it to the RPi in front of the class. Finally, plug in the power and watch it boot up. An alternative to this demonstration is to leave out the memory card and try and boot it to no avail. It should then be possible to describe the memory card as something that contains instructions to tell the RPi how to start. The RPis should all be booted and sitting on the login prompt waiting for authentication.
2. Split the class into groups again and give each group a deck of the *computer program* cards. Ask each group to take out the *statement* cards and the *control* card from the deck. Then ask each group to form a line and to give each member of the group a *statement* card after shuffling them. The person at the start of the line should be given the *control* card. Explain that the person holding the *control* card should carry out the instructions on the *statement* card and then pass the *control* card to the next person in the line like a relay baton. When the *control* card has reached the end of the line, they should stop. This should be repeated for a number of random orderings after which the groups could be invited to create their own orderings. A helpful analogy might be one of cooking - where collections of statements are recipes and the control flow is which stage of the recipe you're in.
3. Start the Sonic Pi software. First, invite the students to log into their RPi and start the windowing environment. It might help to display instructions on how to achieve this on a projector for all to see.
4. Explain to them that they can use the same statements on the cards in the computer program: `play` and `sleep`. Invite them to spend the remaining time writing their own programs and listening to the results.

Plenary

Groups should be invited to choose card orderings for other groups to act out. Following this, a discussion should be held about how this relates to a computer. A computer works by executing *statements* one after another in a specific order. A given order of statements is called a *program*. Each program executes with a given *control flow* which describes which statement we are executing and what the next statement will be.

Homework

Students should be asked to invent statements of their own which they could act out with their family.

All students are able to...	Know how to plug the components of a RPi together. Act out some basic statements and understand that a sequence of statements is a program Write a simple program.
Most students are able to...	Create their own programs for others to act out and have an idea of what might happen when it is acted out.
Some students are able to...	Invent their own statements. Start to explore the limitation of a program with only statements and control flow. Understand the consequences of their program before they run it and therefore design a musical program that's interesting to them.