



redware

Scratch Lesson Plan

Author: Stamati Crook
Email: stamati.crook@redware.com
Telephone: +44 (0) 203 179 9444
Version: 0.7
Date: 6 September 2009
Doc: scratch-lesson-plan-07.doc

Introduction

This introductory course is designed for children aged 7-12 who are beginning to program with Scratch. The course comprises a series of six lessons with the final two lessons devoted to completing a Scratch project to become part of the class showcase. Each lesson should take about an hour, with children working in pairs and exchanging information with each other. Each lesson consists of a ten minute tutorial session that introduces aspects of Scratch programming and can be delivered by watching a video or by teacher demonstration (preferably with a whiteboard). Students then spend ten minutes exploring the tutorial themes before working on their own projects and ideas. A selection of sample projects complements the video tutorials alongside further videos illustrating how to build each type of project. The course can be delivered as a weekly ICT session or within a computer club. We provide notes and references to the national curriculum if you are delivering within the classroom in the UK. Parents and self-learners (maybe older than 7) can also use the videos directly to learn Scratch over a couple of hours.

The schedule of learning is as follows:

- Draw a Sprite and Background
- Turtle Graphics
- Moving and Sensing
- Sounds and Graphics
- Variables
- Broadcast and Receive
- Make a Game
- Build a multimedia presentation

The children should be introduced to Scratch as an exciting area of ICT which lets them create their own computer programs which incorporate their own sounds and graphics. Show them the introductory video from MIT and explain that they are going to learn how to be computer programmers. You can demonstrate a variety of sample applications from our collection or download examples from the Scratch website. Try and show a variety of applications to generate interest from everyone in the class. Now you can begin the lesson plan which introduces most of the important aspects of Scratch. The children need some structured guidance at the beginning of each session but should be exploring Scratch or building their own projects for at least half of each session.

The resources and videos that accompany this lesson plan are available on the internet at <http://www.redware.com/scratch>. Please register if you use this lesson plan at your school or computer club.

Draw a Sprite and Background

This lesson is suitable for very young children especially if they are already familiar with graphics packages. They first learn to draw a sprite and a background and then create several sprites and move them around with the mouse to compose a picture. They can add text if there is time and save and print the final picture.

Scratch is useful as a drawing package but is limited to a particular size. The main learning element of this lesson is to understand that a sprite is a graphic element that can move around the screen independently of the other graphics elements. The background appears behind the sprite and adds context to the drawing.

Children make a sprite and experiment with different drawing tools, many of which may already be familiar to them. They can resize the finished sprite and duplicate it and move it around the screen. More sprites can be added and located on the screen before finishing off the drawing with a background. Sprites and backgrounds can be imported from existing graphics files located on the drive.

Sample pictures might be an aquarium with various fish and plants or a house with several family members and pets located on the screen. Pupils can print and save their completed pictures and make several compositions by moving the sprites to different positions on the screen before printing for a second time.

Turtle Graphics

Turtle Graphics comprise Move and Turn commands which allow the child to move the sprite around the screen with simple control commands. These commands are an important way of introducing children to geometry when combined with the pen commands that leave a trail as the sprite moves. The idea is that the children can experiment themselves and discover how to draw a triangle, square, pentagon, and other shapes without necessarily knowing the angle required to draw each shape. This is an important principle of 'constructivist' learning that forms part of the power of the LOGO language used in education over the past thirty years. Scratch provides an easy way of getting children to use turtle graphics without the need for a physical robot.

Scratch provides the Move and Turn motion command blocks to move and turn the sprite. The lesson involves the children moving a sprite (which they might have drawn themselves) around the screen by selecting and clicking on the appropriate command blocks once they have dragged them over to the script area in Scratch. They can join blocks together and use the pen commands to draw shapes on the screen. By the end of the session they should be able to draw a triangle and a square and put sequences of commands together to draw more complex shapes.

The video shows many examples of turtle graphics including squirrels, and stars, and flowers, as well as the standard polygon drawing techniques. The children learn for themselves during the session that a 90

degree turn is a corner that can be used to draw a square. For younger children it is best to let them always use the same angle for turning (say 30 degrees) and discover that three turns is required for a square and four for a triangle. If they have experienced one key logo before this session, turtle graphics will be easy for them.

Moving and Sensing

The children should already have a good sense of how to move a sprite around the screen from the session on turtle graphics. Note that turtle graphics move the sprite relative to its current position and is a much simpler system to introduce to children than placing an object on a grid with X and Y co-ordinates. Scratch supports both approaches but we recommend that younger children stick to the relative movements and repeat turns of 30 degrees rather than learn the coordinates on a grid and angles. By all means use these features in Scratch if the children are already familiar with these concepts.

One exception is the initial commands which are executed when you press the green start flag in Scratch to start (or restart) the project. We introduce the GOTO command here to initialise the sprite in a start position to reset the application.

Sensing allows the sprite to follow the mouse or another sprite and recognise when a collision occurs. The forever loop is introduced to get a sprite to follow the mouse around the screen and keyboard events are introduced, along with a second sprite, to move the sprite in reaction to keyboard events such as pressing the arrow keys. This forms the basis of a two player game written in Scratch and children can substitute their own sprites, related to their favourite TV characters or to a topic covered in class, making an excellent base for a final project.

Sounds and Graphics

The Lifelong Learning Group at MIT used their experience of running computer clubs for young people to design Scratch as a multimedia programming environment because they knew that children love to incorporate graphics and sound into their applications (unfortunately Scratch does not support video).

Scratch makes it easy to import sounds and graphics into your application to use as effects or to create a complete multimedia presentation on your computer. This lesson shows children how to use the sounds and images provided with Scratch to create an interactive application which responds to the user clicking around the environment.

Children can express their creativity and personality by including their own images and sounds in an application. They might simply add sound recordings to each sprite or include a photograph which they can annotate with sound. They might import a photograph of the class and add recordings from their classmates that are played when the user clicks on the individual in the photograph.

These techniques are used to create interactive multimedia applications (with sound) which can become quite complex when combined with sprites and changing backgrounds. Children can create very sophisticated applications using their favourite music downloaded from an MP3 player and their own photographs. This type of application can be used to illustrate a topic from the classroom or to create an interactive quiz or storyboard and is another favourite for some children.

Variables

Variables were briefly introduced in the session on turtle graphics and are a fundamental part of Scratch and computer programming in general. Variables are introduced as simple numbers which can be made available to the whole application or kept private for a particular sprite. The speed of a sprite might be a private variable whereas the score in a game might be available to all sprites.

The video tutorial introduces a private variable to keep track of the speed of a sprite which can be increased and decreased by pressing keys. The score is maintained in a global variable which is increased each time the sprite collides with a target and the background changes depending on the level reached. Control blocks that use programming constructs such as 'increase the speed only if the speed is less than 10' or 'show a different background if the score is greater than 3' are introduced.

Random numbers are introduced as a useful means for starting off sprites in different positions each time a game starts or for creating unpredictable behaviour.

Broadcast and Receive

Broadcasting messages is crucially important as you begin to build sophisticated Scratch applications and is covered formally as part of the lesson plan to ensure that every Scratch programmer understands the principle of messaging even if they do not use it in their applications.

The children should already understand that events such as pressing the green flag to start the project or pressing a key can be used to control the application and perform a sequence of commands. Broadcast is a way to get a sprite (or the background) to send a message to all the other sprites in the application which can then react to the message by performing a series of actions. Moving to a different level in a game, for example, requires that the background changes and the sprites show and hide themselves and move around the screen. This is extremely difficult to coordinate without using broadcast and receive.

The concept could be explained to the children as one object shouting out a message such as 'I have crashed into another sprite' and the other sprites (and the background) listening and reacting to the message. It is also a great way to get several sprites to perform an action at the same time and might be used to synchronise a dance routine so that all sprites move to the beat together for example. This is the example used in the tutorial video.

Make a Game

Many of the children should by now be fairly competent Scratch programmers and will have explored and used many of the programming blocks not covered directly in the previous lessons. Some will have already been working on their projects for a few sessions and may be creating a game or a multimedia application. This lesson covers the basics of creating a game and should be explored even if the pupil has decided not to create a game as their final project. Several sample games of differing complexity can be used during this session with alternative video tutorials available.

The simplest game is 'dog and crab' and has two sprites created from the sample Scratch images with the dog moving around the screen with mouse-following behaviour and bumping into the crab which makes a sound and moves away. Movement is performed using sensing to follow the mouse pointer and a message is broadcast each time the dog sprite touches the crab sprite to control the required behaviour.

Fish! is a more complex game that can be built by more advanced pupils or maybe those that have already been working on a game type application. A shark moves around in an underwater background eating yellow fish until he gets indigestion from eating the wrong sort of fish. This game involves duplicating sprites to populate the ocean and using broadcasts to coordinate behaviour. A variable is used to keep score if there is time.

Build a Multimedia Presentation

The final session brings together all the concepts learnt in Scratch so far to create a sophisticated multimedia presentation which can be viewed by parents over the internet as part of a class showcase. It is particularly effective when used to illustrate a topic that the class have been studying or when the child's own photos and sounds are incorporated.

Children who are developing multimedia applications might want to view the video tutorial before the final session to help them as they are building their project. A more advanced video tutorial is available for these pupils concentrating on 'digital literacy' skills so they can learn to incorporate images and sounds that they have created with other software into their Scratch applications.

The video shows students how to make a quiz type application, using images and sounds, allowing the user to select from various sprites on the screen to answer a question. The application has both sound and text and keeps a score of the correct answers, changing the background and showing different sprites on the screen for each question. The project incorporates advanced programming using variables and broadcast messages to control changes to the background as the user answers each question.

Conclusion

Pupils will have directly explored many of the features of Scratch in the structured part of the eight sessions and should have used most of the remaining control blocks in Scratch during their own

explorations. The structured part of the course is designed to take up the first third of each session and pupils need complete only the first six sessions to gain a good understanding of Scratch. The more enthusiastic and able students will have used the time in the later lessons to begin developing their own applications whilst other students will need encouragement by starting off modifying and extending one of the sample projects.

The final two lessons are optional and designed to get pupils to understand how to build applications that might be different to the one they have selected for their own project. If boys are developing games and girls multimedia applications, they should at least understand how to build a range of applications.

Children will have been given the opportunity to develop a wide range of skills in addition to Scratch programming. Working in pairs and communicating with other children in the class and watching the videos and experimenting with, and modifying applications, should have extended their communication and comprehension skills. The teacher can facilitate this by showing individual projects to the class and encouraging pupils to 'show and tell' and explain how they solved particular problems.

Children with poor English or special needs often thrive when using Scratch and may gain in confidence and enthusiasm during the Scratch sessions. Geometry and mathematical concepts of angles and coordinates can be expanded upon for older children but turtle graphics and the process of discovering shapes and polygons through exploration should improve the children's understanding of the subject.

Scratch excels in improving digital literacy by providing a framework for children to bring together images and sounds they may have created in other applications together into a single application. Creating a class showcase which can be uploaded to the internet for parents to see can be a great way of bringing a classroom topic to life and creates pride in the children for their creative and engineering skills whilst developing them as computer programmers.