



**redware**

## **Embedding Scratch in the Classroom**

Abstract.....	2
History of Scratch.....	2
Scratch .....	3
Scratch and the Curriculum .....	4
Scratch for Kids .....	5
Scratch in a real Classroom.....	6
Embedding Scratch in the Classroom .....	7
Conclusion.....	7
References .....	8

Author: Stamati Crook  
Email: [stamati.crook@redware.com](mailto:stamati.crook@redware.com)  
Telephone: +44 (0) 203 179 9444  
Date: 30 November 2009  
Document: scratch in the classroom-10.doc

## **Abstract**

The Scratch programming environment is a computer programming language designed for 8-12 year old children to learn how to program and create multimedia applications and games with ease. This paper documents one example of using Scratch in the classroom as part of the ICT curriculum for children aged 7-8 years in a primary school in the UK. A case is made for Scratch becoming a core element used to teach digital literacy in schools to extend the ICT curriculum beyond the teaching of basic office skills and to offer a creative environment to empower the child to program the computer. The case for Scratch is extended to a possible role in embedding the use of computers in the classroom and throughout the curriculum as part of everyday class activities.

## **History of Scratch**

MIT has a long history of contributions to computing for children beginning with the creation of the LOGO programming language by Seymour Papert in the 1970s. Papert is a mathematician as well as a child psychologist and turtle graphics forms a core part of LOGO, allowing a child to control the movement of a robot with Move and Turn commands. These commands prove useful to a child's understanding of geometry when used in combination with a pen to draw shapes on the screen or on the floor when using a physical turtle attached to the computer. Papert has identified 'constructivist' learning (Papert 1980) where a child learns by experimenting and observing and participating in a process. In this case, the child learns principles of geometry by playing with turtle graphics and creating different shapes rather than using a formula to calculate the angle required to draw a polygon. LOGO is easy to learn and Papert documents its use in playground games where LOGO commands are issued verbally during play as children move around the playground in a variation of the children's game 'Simon Says'.

LOGO has been used in many schools over the past three decades but has failed to fire the imagination of either the children or the teachers to the extent originally envisaged by Papert and his colleagues. Turtle graphics is ubiquitous now in all schools in the UK, as it forms part of the curriculum, but is often confined to the use of simple commands to move a simple robot and has lost the trappings of a proper computer programming language and the power of LOGO.

The MIT Lifelong Learning Group have continued the development of LOGO most notably by introducing block programming (in StarLogo) which uses graphical blocks to represent program commands eliminating typing and syntax errors. This greatly improves programmer productivity, and removes a common source of frustration for novices, allowing programmers as young as seven years old to begin programming. Block programming is also used commercially in the LEGO Mindstorms Robotics Kits used by children and educational establishments all over the world to build and program toy robots made with LEGO bricks.

The MIT group has also been involved in running youth computer clubs for many years and has identified that multimedia applications, incorporating pictures and sounds, are a great favourite for children who can impress each other with the 'cool' creations that they make. It is the combination of this experience, together with thirty years of involvement with computer programming in education, that has made Scratch a great contender for radically changing ICT in schools.

## **Scratch**

Scratch is a programming environment, based upon LOGO, which allows young programmers to easily create multimedia applications and simple games (Monroy-Hernández and Resnick 2008). Block programming eliminates the frustrations of syntax errors which afflict novices learning traditional computer programming languages. The Scratch environment allows the creation and control of graphical objects, known as sprites, which move around the screen and integration of graphics or photos and sound files to create multimedia applications.

Scratch is easily learnt by an adult in a couple of hours and children from the age of seven can be productive from the very first lesson. A typical lesson plan (Crook 2009) might require six sessions for a young programmer to learn the main programming concepts and begin creating their own games or multimedia projects.

Scratch applications are designed to be shared and the Scratch web site has thousands of applications which can be downloaded and remixed into new applications. Some applications that can be created by novice Scratch programmers are detailed below:

- A simple game where the player controls a sprite chasing another sprite around the screen.
- An animation with sprites representing the letters of the programmer's name with each letter spinning or otherwise moving around the screen and playing a sound when clicked.
- A quiz style application showing several pictures and a question which responds to the user clicking on the selected answer.
- A multimedia presentation incorporating photographs with sound recordings providing explanations as the user clicks on each item.

Scratch programming involves several IT skills and creation of a typical application might involve:

- Drawing a sprite within the Scratch application.
- Importing a background picture from a photograph or a file created in another graphics application.
- Recording a sound file or incorporating an MP3 file into the application.
- Adding text with different fonts to an application.
- Moving sprites around the screen with turtle graphics commands.
- Manipulation and digital effects on graphics objects.

Scratch programming also involves an understanding of some of the principles of computer programming:

- Creating a sequence of commands.
- Using an IF conditional statement to determine a course of action.
- Repeating a sequence of commands.
- Moving a sprite around the screen and responding to the environment.
- Responding to keyboard and other events.
- Using variables in an application.
- Using messages and broadcasts to control the application.

### **Scratch and the Curriculum**

Scratch programming can be taught with a series of sessions in a computer club or during school time to satisfy large parts of the ICT curriculum. The curriculum here in the UK has many areas relating to ICT which can be satisfied with Scratch including:

- Understanding fonts and graphics to produce a greetings card.
- Control of a turtle with turtle graphics.
- Developing ideas with sounds and graphics to present in a multimedia application.
- Sharing information with animation and music.
- Controlling events by planning, testing, and modifying sequences of instructions.

Many classrooms in the UK have one or more computers in the classroom, as well as the teacher's computer connected to an interactive whiteboard, allowing computers to be used outside of the computer lab. The whiteboard can be used to access internet and multimedia applications to illustrate classroom topics unrelated to ICT. There is an opportunity here for Scratch to be used to create applications to illustrate topics and bring together cross-curricular and topic-based activities, either with a sample application demonstrated by the teacher on a whiteboard, or as a multimedia application created by the children. This moves us on from using computers just for ICT related activities and aspires to teach children 'digital literacy' (Buckingham 2007) where they can combine all of their computing skills to illustrate a classroom topic in the same way that general literacy is used to write up a topic.

Some examples where Scratch can be used in the UK curriculum are illustrated below:

- **Simulation.** Scratch applications can be created to illustrate a simulation such as a car racing around a track and running out of fuel at a different rate depending on the average speed around the track. Pupils can use and alter an existing application or create their own from Scratch.
- **Data Gathering.** Scratch can be connected up directly to a device called a ScratchBoard which has sensors to measure light, sound, and other factors from the environment, which can be used to collect data for a science or data gathering activity.
- **Robotics.** Scratch can be connected up directly to a Lego WeDo device connected to the computer which has a motor and various sensors allowing a simple robot to be controlled with the Scratch programming language.

- ICT. Scratch is a programming language and can be used to create applications involving a high degree of digital literacy and the integration of text, pictures, graphics, and sound, in an interactive application.
- Geometry. Turtle graphics are an established means of teaching children the principles of geometry in a constructivist manner where they learn core principles by experimenting with drawing different shapes using an onscreen turtle.
- Epistemic Games (Shaffer 2008). These games allow children to mimic practises from the professional world in a role playing context whilst building an application. Pupils can role play at being computer programmers, game designers, or multimedia producers whilst creating a Scratch project.
- Art. Children can scan in their drawings and create animations easily accompanied with sound from their own recordings or their favourite MP3 track.
- Music. Simple applications to control activities for a music class are easy to develop in Scratch where the teacher rearranges sprites representing notes on the screen to prompt the pupils to play notes in the correct order for example.

### **Scratch for Kids**

A lot of money has been spent on the infrastructure for ICT in education and most schools in the UK have a computer lab and a computer and whiteboard in each classroom. Much of the current practise and ICT curriculum is based upon introducing primary school children to computing skills that they might require a decade later when they start work. There seems to be too much focus on basic ICT skills such as word processing, spreadsheet and database use, sending emails, using search engines and so forth. Although turtle graphics, robotics, and simulations are in the curriculum, they are often taught with simplified and specialised applications that match only the minimum requirements for the curriculum.

Scratch empowers children to control the computer and create their own applications rather than use the computer in a passive manner. Instead of running a simple application and pressing a button on the screen, the children learn to program and create their own applications. Turtle graphics can be taught as it was meant to be taught, with pupils using their own creativity and imagination to create much more complex shapes that can be done with a simple robot used for a couple of sessions. The experience of creating their own shapes will construct their own understanding of geometry which will help reinforce learning as they are introduced more formally to the subject during mathematics lessons.

Creativity and originality is core to the process of designing and creating a computer application. Students can be given the same application to create but each will employ different techniques to create the final application. Simply allowing children to originate the graphics for a project ensures originality and ownership of the project even if the programming is constant.

Pupils often work in small teams when creating applications and communicate with each other and the teacher when they have discovered a novel technique that helps them build the desired application. The

process of communicating programming issues with their team members and classmates can build up confidence and communication skills that apply to all aspects of their schooling.

Creating a finished application involves many skills used by professional computer programmers, games designers, and multimedia producers, and the children learn the process of moving from the requirements of a desired application, through a design phase, to the engineering and testing of the finished application. The completed application can be uploaded to a web server to create a showcase for parents and other children to view, again building up the confidence of the programmers.

Scratch projects are designed to be shared and programmers can easily take an existing application and modify and improve on it. The understanding and communication skills involved building on the work originated by another child is another great communication skill.

### **Scratch in a real Classroom**

A Year 3 class at a UK primary school in Brighton was taught Scratch as part of the normal ICT program for an hour once a week in a computer lab backed up with an optional after school computer club. The lab was conveniently situated next to the classroom so ad hoc Scratch sessions were also held periodically. There were 23 children between 7 and 8 years old in the class and all of them enjoyed and looked forward to the sessions each Wednesday morning.

Children shared a computer and collaborated on projects. Initially pairs would complain that other children were copying their work until the teacher explained that sharing of knowledge and programming techniques was part of the process. All the projects were stored on the same computer drive and it was easy for the teacher to show particular solutions on the interactive whiteboard and to encourage children to share sprites and project solutions.

Both boys and girls liked Scratch although the boys were more drawn to creating games. Some of the pupils preferred the creative process of drawing and creating sprites and had to be encouraged to get on with the task of programming, but all completed their tasks. One pupil from Romania did not speak English well but was able to excel in the Scratch lessons and gained in confidence even though she was working on her own. Another pupil changed the language to Polish and was able to keep up with the rest of the class without problems. A special needs child with a permanent adult helper was able to work with the helper to create his projects and enjoyed the lessons so much that he attended extra computer club sessions after school with his mother.

The teacher made available a series of photographs of the local area taken as part of a field trip for a geography lesson and these were incorporated into Scratch projects by many of the pupils. The teacher is committed to using Scratch in the future and plans to personalise small applications using pictures and sounds created by the class. One example is a timer application that runs on the whiteboard and is personalised to show the photo of a random member of the class whilst counting down the minutes left for the class to complete a task.

All the children loved the Scratch sessions and the teacher found them useful. The teacher had to spend considerable effort preparing lesson plans for the ICT sessions and ensuring that she had the programming knowledge to keep up with some of the more advanced young programmers who were working on projects at home and downloading projects from the internet. Interest from the other teachers in the School is not high however and it seems unlikely that the other teachers in the school will want to adopt Scratch in the immediate future.

### **Embedding Scratch in the Classroom**

There are many barriers to overcome before Scratch can be embedded into everyday classroom use but the technical infrastructure of a computer and an interactive whiteboard is already present in many classrooms in the UK.

One major issue is that many teachers want to adopt an incremental approach to help them change their attitudes and teaching practice. Scratch allows these teachers to teach the standard the ICT curriculum whilst offering an opportunity to radically change the teaching of ICT by empowering children with high levels of digital literacy and creative control of the computer through computer programming. There should be few barriers to this provided that teachers are given the time and the training to become familiar with the Scratch environment and the resources and lesson plans to be able to run ICT lessons and computer clubs successfully.

The creative curriculum being introduced in the UK is giving teachers more scope to cross-reference different parts of the curriculum and create cross-subject activities. Scratch is a great mechanism for bringing different activities together and presenting topic knowledge as a set of applications created by the class. A second stage, once teachers and pupils are familiar with Scratch, is to incorporate Scratch throughout the curriculum and embed its use within the classroom as a means of exploring digital literacy to create class showcase applications to illustrate the topics being studied in class. The best of these applications can also be used by the teacher as teaching aids for the next generation of students.

### **Conclusion**

ICT in many schools is treated as an independent part of the curriculum with pupils trained in a computer lab to use computer applications as an office worker might use computer software as part of their daily routine. Many topics that form part of the curriculum are suitable for teaching as part of a computer programming course but are usually taught with simplified applications that stick to the minimum required by the curriculum and leave little scope for creativity or stretching a child's knowledge.

Scratch offers an opportunity to teach children as young as seven years old how to program a computer and create powerful multimedia applications and games. The thought processes involved in collaborating in the planning, design, creation, and fixing, of a computer application can bring out and improve communication skills in many children. Using Scratch in a variety of ways to improve digital

literacy as well as computer programming skills, truly empowers pupils to control the computer and break free from the limitations of the simplified applications currently used to satisfy the minimum requirements of the curriculum.

Scratch offers the teacher the opportunity to embed the computer into everyday school activities by getting the class to develop skills in digital literacy related to a variety of topics in the curriculum. Sounds, pictures, photos, and data can be combined in a Scratch application for many topics in the curriculum and used to create a class showcase for parents and other children in the school to see.

Teachers and educational establishments need access to the resources and training required to give teachers the experience and confidence to incorporate Scratch into their lesson plans for ICT as well as other topics on the curriculum.

Scratch was released in 2007 and has made little impact in schools in the UK over the past two years. The task now is to motivate local authorities to provide the budget and resources to expose teachers to Scratch and give them the opportunity to incorporate computer programming and digital literacy into the curriculum. The adoption might be incremental at first and geared around satisfying the ICT curriculum, but it is hoped that Scratch has the potential to help begin the process of embedding the computer into everyday classroom activities as part of an overall approach to teaching digital literacy.

## References

- Buckingham, D. (2007). *Beyond Technology: Children's Learning in the Age of Digital Culture*. Polity Press.
- Crook, S.J. (2009). Scratch Lesson Plan. <http://scratch.redware.com>. Redware Research Limited.
- Monroy-Hernández, A. and Resnick, M. (2008). Empowering kids to create and share programmable media. [Interactions](#), March-April 2008.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books.
- Shaffer, D. (2008). *How Computer Games Help Children Learn*. Palgrave Macmillan.