

Scratch—A Free Computer Programming Language Just for Kids

by Paul Giganti Jr., CMC Math Festival Program



Our K–12 students often forget that every program or game that they use on the computer started as an idea in someone’s head. Programmers that had the idea then had to make the computer do what they wanted, because a computer really can not do anything unless someone first teaches it how. We call this computer programming.

Although using computer programs or games allows students to do amazing things, there is nothing quite so powerful as creating something on the computer from scratch. Now for students of all ages, there is a free program called *Scratch* that does just that.

Scratch is a programming language that makes it easy to create your own interactive stories, animations, mathematics simulations, games, music, and art, and to share your creations through the Internet. By creating and sharing *Scratch* projects, young people learn important mathematical and computational ideas, while also learning to think creatively, reason systematically, and even work collaboratively. Students can use *Scratch* to experience the logical thinking, planning, problem solving, and trouble-shooting that goes into even the simplest computer program—and they can do it by themselves. The motto of *Scratch* is:

Imagine • Program • Share.

Scratch was developed by the Lifelong Kindergarten Group at the MIT Media Lab, with financial support from the National Science Foundation, Microsoft, Intel Foundation, Nokia, Iomega, and MIT Media Lab research consortia. It can be used in many different settings: schools, museums, community centers, and homes. Although *Scratch* is intended especially for ages 8–16, younger children can also work on *Scratch* projects with the help of their parents or older siblings. And college students can use *Scratch* in some introductory computer science classes. Even the author of this article enjoyed programming in *Scratch*—I wanted to program a bouncing ball, complete with

bouncing sounds. You can view my first *Scratch* program at:

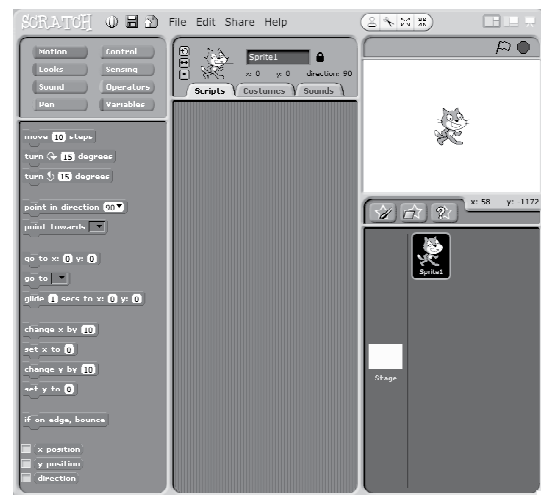
Scratch.mit.edu/projects/pgiganti/850828.

How to Use Scratch

Downloading, using, and sharing your *Scratch* creations online is absolutely free. Go to Scratch.mit.edu/ to view the *Scratch* home page. Before you download your free copy of *Scratch* and try your first program, it is a good idea to first browse the *Scratch* web site. On the home page you'll find many examples of students' work in *Scratch*. By clicking on them you will see the incredible diversity of projects that students of all ages have imagined, programmed, and shared with the world, and hopefully provide ideas of new creations.

A child will need to create an online account in order to share his or her *Scratch* programs. Younger students may need their parents' permission and help to do this.

Below is the first screen that will appear after downloading and opening the *Scratch*:



On the right-hand side is the “Stage,” the place to view and test a creation. The little cat shown is one of hundreds of pre-programmed “sprites,” moveable objects available to use in programs, or you can also create your own sprites. On the left-hand side of the window is

a selection of tools for assembling a program. These tools are in the form of building blocks that can combine with other building blocks to accomplish the desired actions. Above the building blocks are eight palettes that will change the contents of the building blocks and allow you to change what the sprite can do.



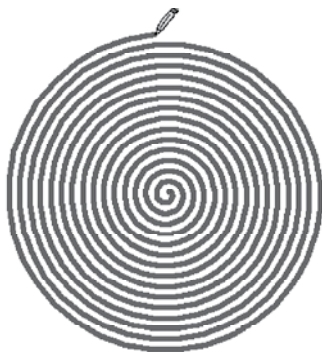
The middle of the window is where you assemble your program by simply dragging and dropping each building block you wish to use. *Scratch* will even help you assemble them in the middle so they fit together to create the action desired.

This article is far too short to teach the process to create a program in *Scratch*, but there are many online Scratch tutorials that will do just that—or you can just play with the tools and see what they do.

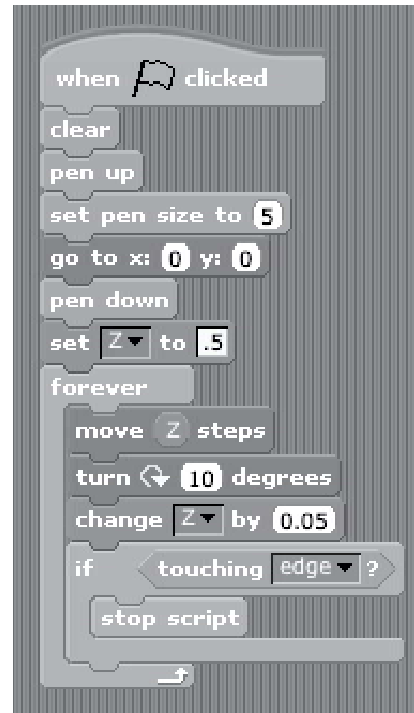
Most of the hundreds of tools or building blocks are intuitive. You can learn them by trying them out. Some tools are harder to master, but the more you program, the more these complicated tools will also make sense.

An Example of a Scratch Project

For my second *Scratch* project I wanted to get the computer to draw a nice spiral, better than any spiral I could draw by hand. Here is a picture of the spiral that Scratch and I taught the computer to draw:



And here are the building blocks I put together to make my spiral program:



You can probably figure out the function of each of the building blocks I used when you look at them in my program. You can watch my spiral draw itself online at Scratch.mit.edu/projects/pgiganti/859430.

Online Help

The best part of the learning process is that *Scratch* is also a community of learners. If you want to know how to do something, you can return to the web site and find help, including other student programmers who can answer your questions.

Conclusion

There is no more powerful activity with computers than making a computer do what you want—by teaching it how to do it. Share this wonderful resource with your children. Programming in *Scratch* will seem like play, but they will be using important mathematical and computational ideas, problem solving, and using logic and reason—all while being creative and having a good time. Computer programming is good for young minds! 📦

Permission is granted to reproduce and share this article for instructional use by parents, guardians, teachers, and families—provided it is duplicated with full credit given to the author, the California Mathematics Council, and its Journal, the ComMuniCator.

