

SUPER SCRATCH

PROGRAMMING ADVENTURE!

COVERS
VERSION 2



LEARN TO
PROGRAM
BY MAKING
COOL
GAMES!

THE  PROJECT



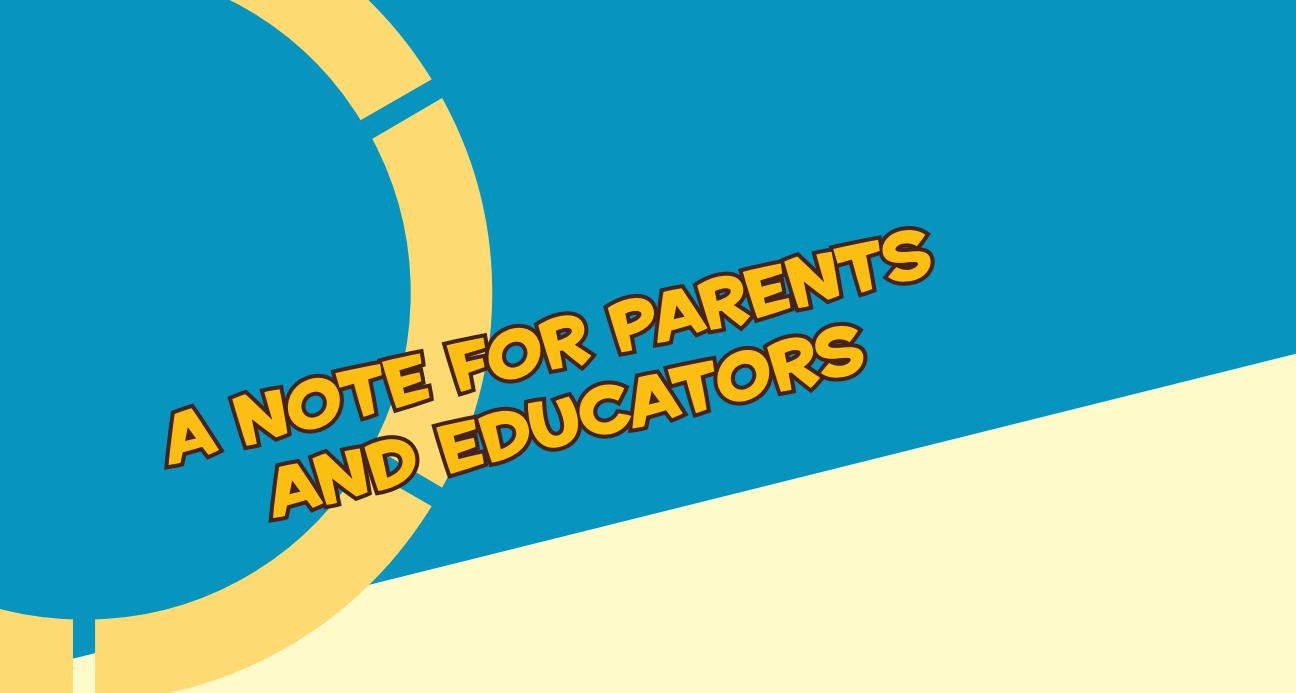
This is an excerpt of

SUPER **SCRATCH** PROGRAMMING ADVENTURE!

LEARN TO
PROGRAM BY
MAKING COOL
GAMES!

THE  PROJECT





A NOTE FOR PARENTS AND EDUCATORS

Scratch opens up an exciting world of computer programming for kids and other beginning programmers. To follow along with this book and use Scratch 2, you'll need:

- A computer with a recent Web browser (Chrome 7 or later, Firefox 4 or later, or Internet Explorer 7 or later) with Adobe Flash Player version 10.2 or later installed
- A display that's 1024×768 or larger
- A reliable Internet connection
- A microphone and speakers (or headphones) to record and listen to music

Once you have a browser and Adobe Flash Player installed, just point your browser at <http://scratch.mit.edu/>. You can create a new Scratch project without logging in by clicking the **Create** button. You'll want to eventually **Join Scratch** to create your own account and save your projects (see how in “Join the Community!” on page 15).

You should download the projects used in this book from <http://nostarch.com/scratch/>. This online resource includes complete working projects, custom sprites, and a short *Getting Started with Scratch* guide produced by the Scratch team.

NOTE

The Resources file includes two versions of each game in the book. One version is a completely finished and playable game, perfect for young learners and anyone who wants to build on the games in the book. The second set of projects has no programming added, so that students can follow along with the programming instructions in this book. Remember, there's no wrong way to play with Scratch!

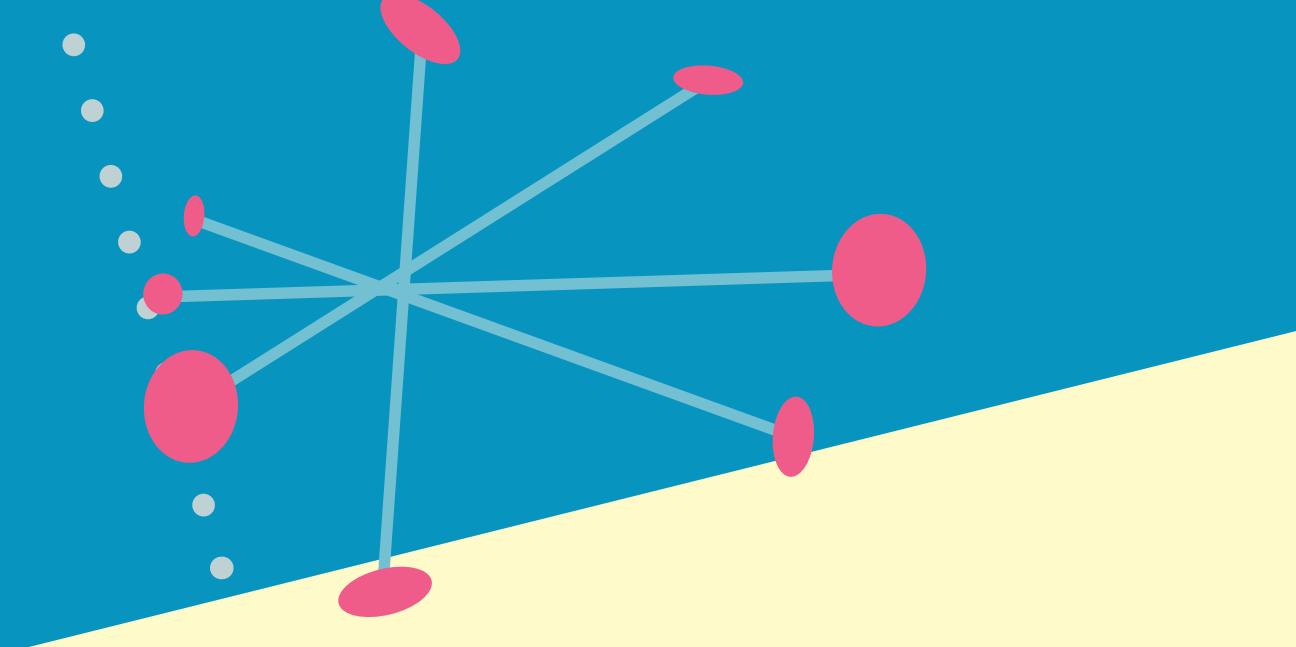
BUT WHAT IS SCRATCH, ANYWAY?

Scratch is a graphical programming language that you can use for free. By simply dragging and dropping colored blocks, you can create interactive stories, games, animation, music, art, and presentations. You can even upload your creations to the Internet to share them with Scratch programmers from around the world. Scratch is designed for play, self-directed learning, and design.



WHERE DID THE NAME SCRATCH COME FROM?

Scratch is named for the way that hip-hop disc jockeys (DJs) creatively combine pieces of music, using a technique called *scratching*. In the same way, Scratch programmers join different media (images, photos, sound effects, and so on) in exciting ways to create something entirely new.

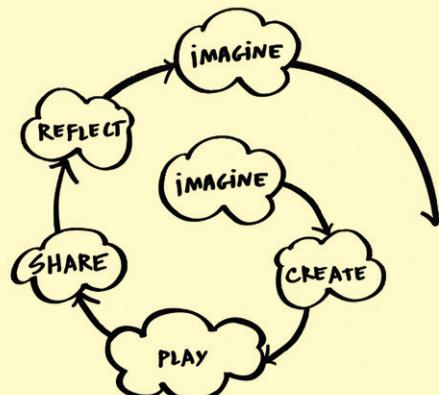


WHO CREATED SCRATCH?

Scratch is a project funded by the US National Science Foundation (NSF). It was developed by the Massachusetts Institute of Technology (MIT) Media Lab's Lifelong Kindergarten Group.

WHO IS SCRATCH FOR?

Scratch was developed for young people aged 8 and up to help them develop creative learning skills for the 21st century. When kids create programs, they learn important mathematical and computer concepts that improve their creative thinking, logical reasoning, problem solving, and collaboration skills.



This creative thinking spiral is from Professor Resnick's article, "Sowing the Seeds of a More Creative Society," published in *ISTE* (International Society for Technology in Education).

Designing Scratch projects challenges kids to think creatively, and learning how to overcome obstacles and solve problems builds confidence. This gives learners an advantage later in life.

IS IT EASY TO USE SCRATCH?

Scratch was designed to prevent the common beginner pitfalls in traditional programming languages, like misspelling and errors in consistency. Instead of typing commands, programming in Scratch is performed by dragging and joining programming blocks. This graphical interface allows users to easily control the way in which different types of commands react to each other. Additionally, each block can fit with another only if it makes computational sense. Colorized categories help organize and group different sets of related commands based on their particular functions.

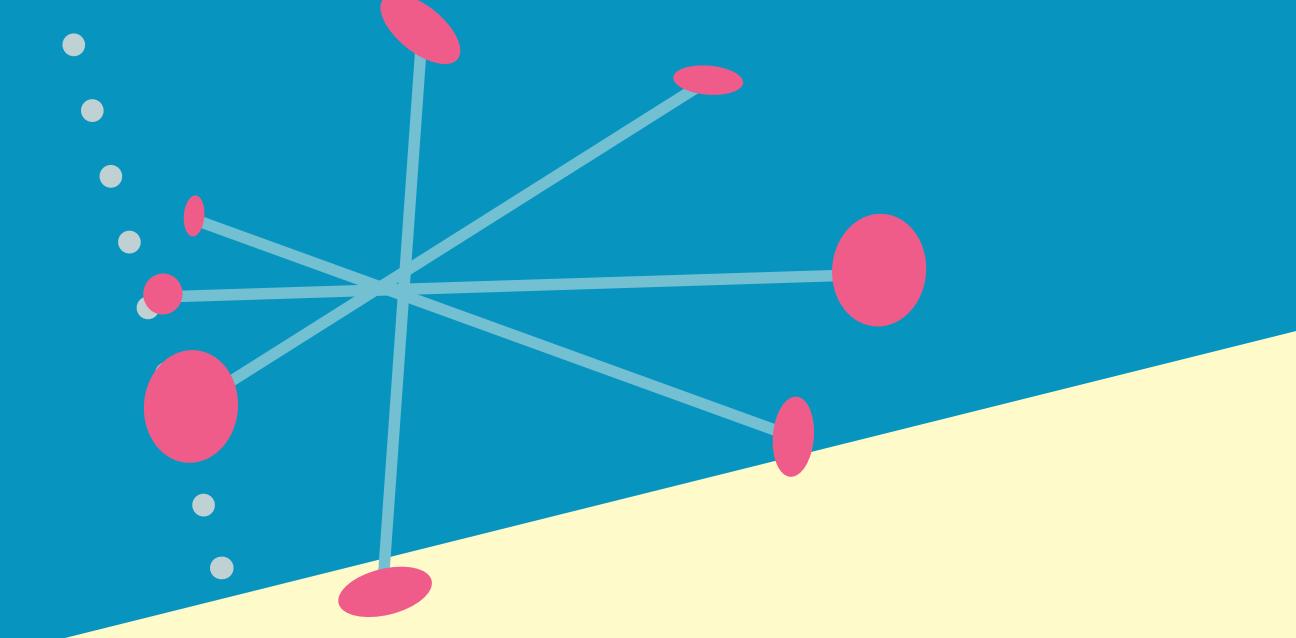
Since programs in Scratch run in real time, they can be edited and tested at any given moment, even while the program is running. This allows users to easily experiment with new ideas or to repeatedly test their improvements!

HOW MANY LANGUAGES DOES SCRATCH SUPPORT?

Scratch can be used in 50 different languages. Choose your language from the pull-down menu at the bottom of the Scratch website.

WHERE CAN YOU USE SCRATCH?

You can use Scratch at schools, libraries, community centers, and home. Even though Scratch is designed for young people aged 8 and up, younger children can also learn to design and create alongside their parents or siblings.



Scratch is used around the world in elementary, middle, and high schools. Computer science professors also use Scratch as a means of introducing programming concepts to college students.

HOW CAN SCRATCH BE USED TO EDUCATE IN SCHOOLS?

Schools can use Scratch to aid teachers in subjects like mathematics, English, music, art, design, and information technology. Scratch is designed for exploration and experimentation, so it supports many different learning styles.

No matter what they use Scratch for—creative storytelling, unique video games, or simple demonstrations of programming concepts—Scratch will provide a space for students to explore and imagine. By engaging in design-based activity individually or in groups, students will be motivated to learn.

Here are just a few of the things that students have used Scratch to do:

- A school in New York City used Scratch to build simulations of the spread of infectious diseases.
- A group of teenagers in India used Scratch to make an animated map of their village, illustrating environmental concerns where they live.

- Students at a university in Istanbul used Scratch to examine video game culture by rapidly prototyping their own games and testing the games with the public.
- English students in a middle school in California used Scratch to build a random story generator.
- Students in an elementary school in Russia used Scratch to build their own personalized tutorials for learning about the coordinate system and trigonometry.
- High school students in Michigan used Scratch to build a physics simulator.

The possibilities are endless. It is our sincere hope that this book inspires students to create their own games, stories, and more.

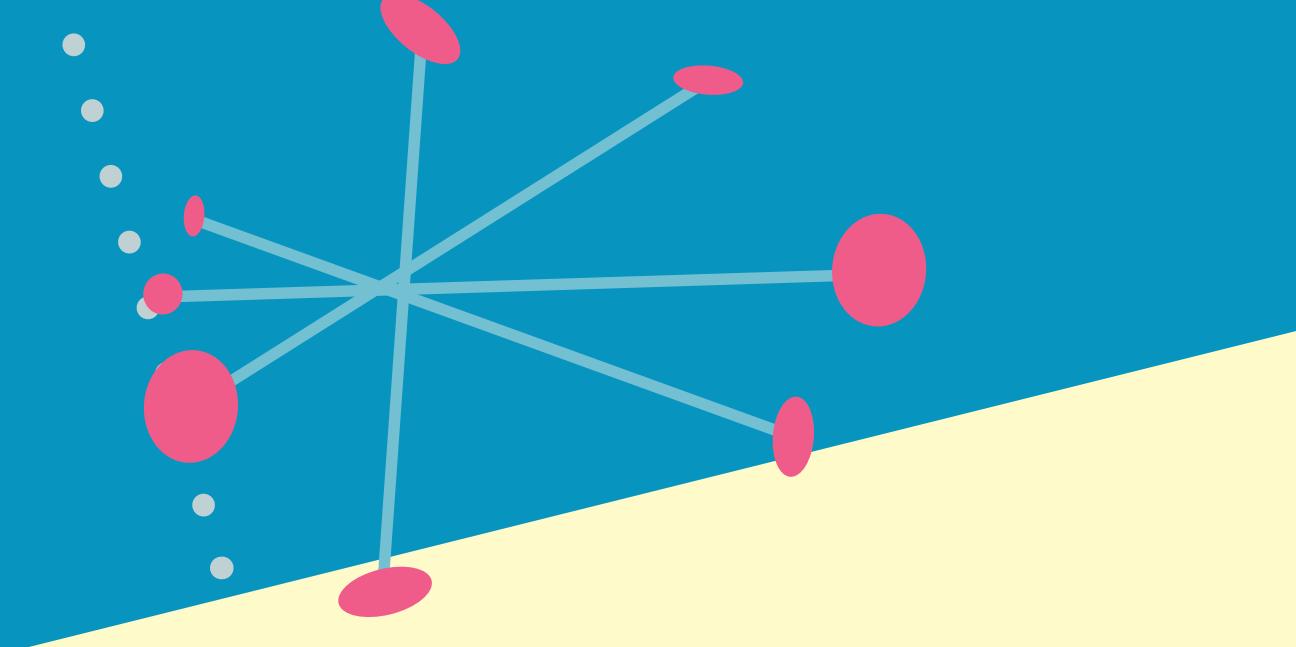
JOIN THE COMMUNITY!

Because Scratch is online, kids can easily share their own Scratch projects with their friends, family, and teachers. Once someone shares their work publicly on the Scratch website, other Scratch programmers can remix their projects, give them feedback, and more.

Follow these steps to join Scratch:

1. Visit the Scratch home page (<http://scratch.mit.edu/>) and click **Join Scratch** to register (you only need to register once).
2. Choose a username (don't use your real name), and then fill out the rest of the information. If the person registering is under 13, Scratch will ask for the email address of a parent or guardian.

NOTE *Once you share a project, everyone in the whole world can see what you've made! Make sure that your kids*



or students know to keep their personal information private.

As long as they have the username and password at hand, kids can find games to play through the project gallery, remix them, and share their thoughts with others from around the world! To see how someone else's game was built, just click the **See Inside** button (). To add to the program, click the big orange **Remix** button ().

To share your own projects with the rest of the world, click the big **Share** button () in the Scratch editor. To make a project private again, click the **Unshare** button in the **My Stuff** listing.

Just remember that as a member of the Scratch community, you'll be sharing projects and ideas with people of all ages, all levels of experience, and all parts of the world. So be sure to:

- Be respectful of other players
- Be constructive when commenting
- Help keep the site friendly and fun
- Keep personal information private

For more ideas and information about sharing and remixing projects, visit <http://wiki.scratch.mit.edu/wiki/Remix>.

MY COMPUTER CAN'T RUN SCRATCH 2!

If your computer doesn't meet the requirements listed on page 10, you can still download and install Scratch 1.4. (http://scratch.mit.edu/scratch_1.4/). Scratch 1.4 projects are compatible with the Web-based Scratch 2, and you can still share your projects on the Scratch website using Scratch 1.4. (Unfortunately, Scratch 1.4 cannot read programs created in the Scratch 2 software.)

You can download free PDF versions of Chapters 1 and 2, which explain how the older 1.4 interface works, by visiting <http://nostarch.com/scratch/>. You can also find versions of the book's games that are compatible with 1.4 on that page.

I'M AN EDUCATOR USING SCRATCH

Awesome! This book is great place to start for classes and after-school programs. You'll want to download the free Educator's Guide at <http://nostarch.com/scratch/>. Visit the official Scratch educator's forum at <http://scratched.media.mit.edu/> to exchange resources, share success stories, and ask questions of other educators already using Scratch as an educational tool.

I STILL HAVE OTHER QUESTIONS...

You can find more information on the Scratch website:

- Visit the Scratch FAQ at http://info.scratch.mit.edu/Support/Scratch_FAQ/.
- Visit the Scratch Help at <http://scratch.mit.edu/help/>.

“Online Resources” on page 156 has other helpful links. For updates to this book, visit <http://nostarch.com/scratch/>.

MEET THE CAST

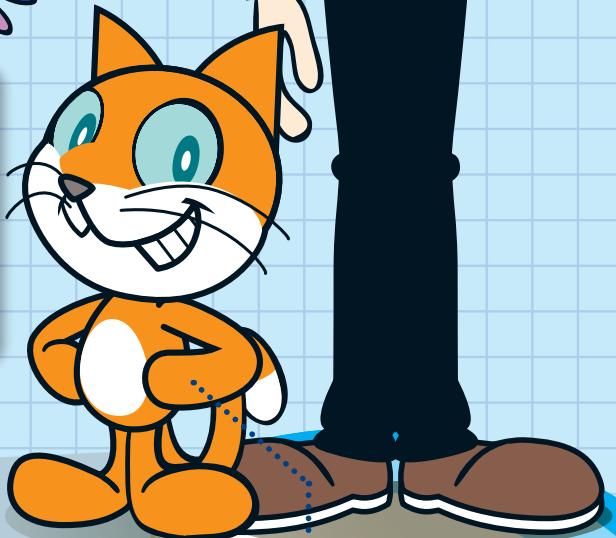


The Cosmic Defenders: Gobo, Fabu, and Pele

The Cosmic Defenders are trans-dimensional space aliens who can travel through space and time. Formally deputized by the Galactic Council, the Cosmic Defender's duty is to maintain the balance of the universe.

Mitch

A computer science student who loves to make cool programs, he's passionate about movies and art, too! Mitch is an all-around good guy.



Scratchy

An energetic cat living in cyberspace, Scratchy is exactly what you'd expect from a cat on the Internet. He's quite curious and impulsive.



The Dark Wizard

He is a shapeless yet powerful and vengeful spirit, whose origins are unknown. Nothing can stop his ambition of destroying the order of space and time.

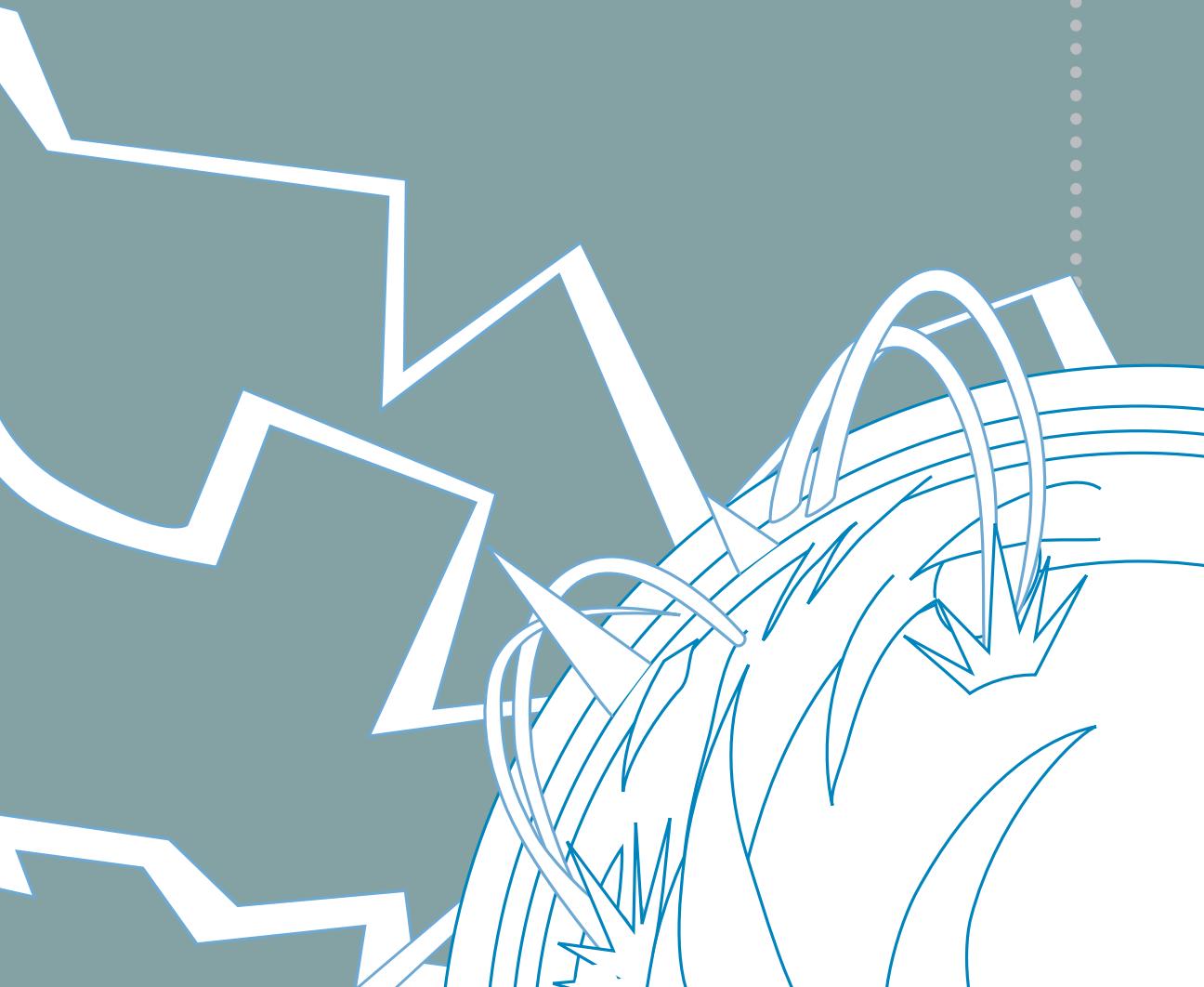


The Dark Minions

These pesky foes are Cosmic Defenders who have fallen to the dark side. They work for the Dark Wizard now.

RIDING A FLARE FROM THE SUN

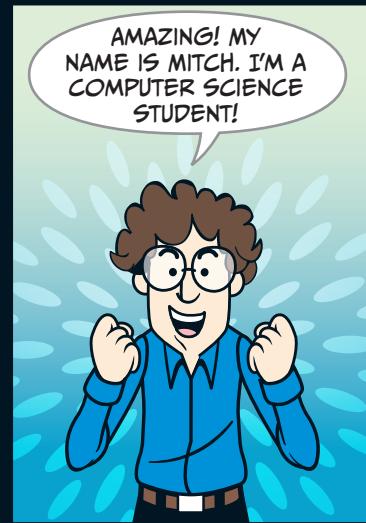
1
STAGE



STAGE

1





1 STAGE

BREAKING THE SPELL!

Chapter Focus

Let's get to know Scratch! We'll also learn about *sprites* and *coordinates*.

The Game

We need to get Scratchy the cat moving again. We'll make him dance across the Stage.



To follow along with the Secret Manual, you first need to open Scratch. Once you **Create** a new project, you'll see Scratchy the cat on a white backdrop. The cat doesn't do anything yet because he doesn't have any programs. Scratch calls Scratchy the cat—and all the other characters and objects we add to a project—a *sprite*. Soon, we'll start giving him directions to move by using the blue blocks in the middle of the screen.

The command blocks you can give a sprite are here. We'll stack these commands together to break the magic spell and get Scratchy back on his feet. The blocks here are all blue, as they're from the **Motion** palette.

You'll need to give each sprite its own instructions. In other games we play, we'll have more than one character to control, so we'll have more than one sprite listed here, in the Sprite List. To give a particular sprite instructions, click a sprite in the Sprite List first and then drag blocks into the Scripts Area.

To move a block, just click and drag it over here. This is called the Scripts Area. It's where we write our programs.

A Guided Tour of the Scratch Interface!

Play the game full screen.

Give your project a new name.

Sprite Toolbar

Contains the Duplicate, Delete, Grow, Shrink, and Block Help tools

Palette

Each of these ten buttons lets you choose functions (called *blocks*) for programming your sprites. You can combine these command blocks in stacks to create programs that control objects on the screen.

Stage

Displays your creation



The green flag starts the game and the red flag stops the game.

Sprite List

Here are the characters and objects you've created, including the Stage itself. Click the icons to edit each sprite individually.

Scripts Area

Here's where you build your programs. Stacking blocks together here lets you control the sprites in your project. Click one of the three tabs at the top to change to other functions:

Scripts: Allows you to drag command blocks from the Palette and put them together to write a program

Costumes: Allows you to draw, import, or edit images for a sprite

Sounds: Allows you to record or import sound files for a sprite to use

New Sprite Buttons

There are four ways to add a sprite:

- Pick one from Scratch's built-in library
- Draw a new one
- Upload an image you already have
- Take a photo with your computer's webcam

1 STAGE

Sprite Information

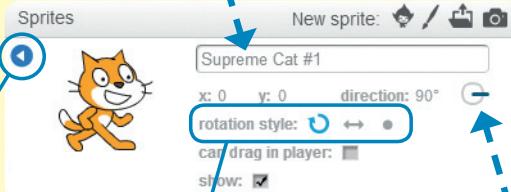
You might have noticed a little blue **i** in the corner of the box around Scratchy when you select his sprite in the Sprite List. Try clicking the **i**, and you'll get information about that sprite.

This section shows the sprite's name, position, and direction it is facing (the little blue line).

CLICK HERE



This is how you can rename the Scratchy sprite. Right now it's *Sprite1*. Don't you think that's a little boring? Try renaming this sprite.



Click this arrow when you're done with the Sprite Settings pane. We'll play with these other settings later.

Rotation Settings

You can control how a sprite rotates in three ways:

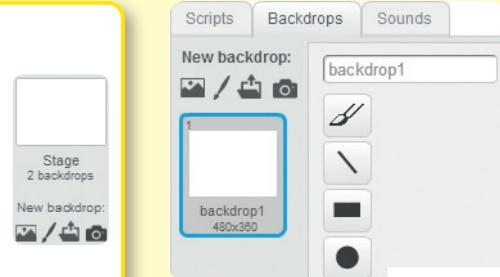
- Can rotate freely
- Can face only left or right
- No rotating allowed

Try clicking and dragging the little blue line—see what happens to Scratchy's orientation.

Now, onto the fun stuff. To use Scratch to program movements, you first have to understand how Scratch positions things.

Click the **Stage** icon in the Sprite List. Switch to the **Backdrops** tab in the Scripts Area and choose **Choose backdrop from library**.

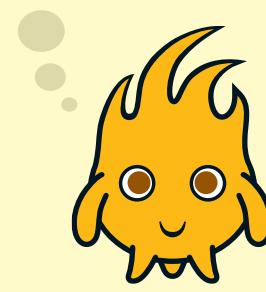
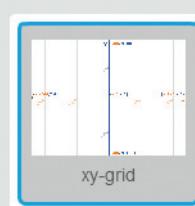
Note: Sprites have *costumes* while the Stage has *backdrops*.



Choose the *xy-grid* backdrop and click **OK** to use it. It's in the "Other" category.

Backdrop Library

Category
All
Indoors
Outdoors
Other
Theme



Now you can see exactly how Scratch positions objects. Everything is on a grid with two axes:

y-axis: A vertical line that marks up and down positions; ranges from -180 (lowest) to +180 (highest)

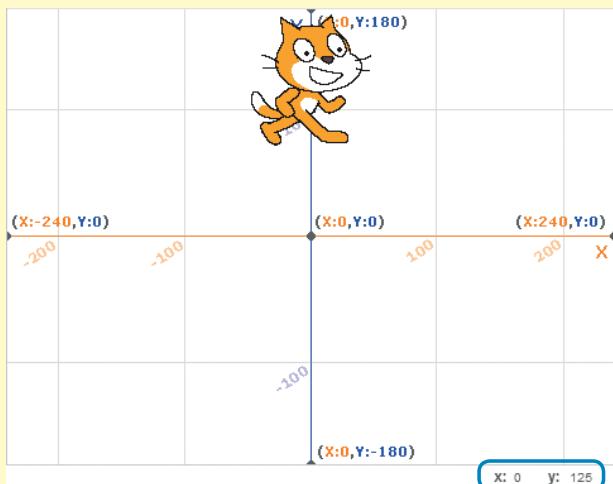
x-axis: A horizontal line that marks left and right positions; ranges from -240 (farthest left) to +240 (farthest right)

Scratchy's default position is at the point where the x-axis and y-axis meet. His coordinates are (X: 0, Y: 0).

Now we can program movements for Scratchy the cat! But first, try dragging him to the top of the Stage, as shown on the right.

Note: The bottom-right corner displays the coordinates of your mouse. This will be really helpful when we start setting the positions of sprites!

The current coordinates of a sprite are shown in the upper-right corner of the Scripts Area, too.



To make sure we're giving Scratchy the cat instructions, click him in the Sprite List (the box at the bottom left of the screen). Switch to the **Scripts** tab in the Scripts Area and then click the **Motion** palette button. Click and drag out the command block `go to x: 0 y: 0` to the Scripts Area.



1 STAGE

Click the number of a coordinate to change it. Set x to 0 and set y to 125. Now click the block to run it! Scratch goes right to that position. We've just written our first program! It's really that simple.

go to x: 0 y: 125

We want Scrathy to move around, but at the moment, he moves too fast for us to see! To make him move more slowly, click the **Control** palette and drag out the command `wait 1 secs` to the Scripts Area. Make sure to drag it under your blue command block. Wait for a white line to appear and then release the mouse.

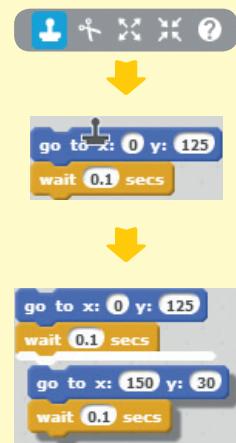
go to x: 0 y: 125
wait 1 secs

The two commands are joined together! Now change the time to 0.1 secs.

Tip: If you want to separate the commands, simply drag away the block. If you want to delete a block, simply drag it back to the palette. Give it a try. To move a big stack of blocks, click and drag the topmost block in the stack.



Next, select the **Duplicate** button on the Sprite Toolbar and stamp it on the commands to make five copies.



Type these coordinates in your own program, so it matches this picture. When you're finished, click the whole command block to make Scratchy jump around in a pentagon shape!

```
go to x: 150 y: 30
wait 0.1 secs
go to x: 100 y: -120
wait 0.1 secs
go to x: -100 y: -120
wait 0.1 secs
go to x: -150 y: 30
wait 0.1 secs
```

To make him move in a loop continuously, drag out the command block **forever** from the **Control** palette and place it at the top of the code. Click the block, and it will actually run! Click to stop Scratchy from moving around. You can test any program in this way—just click it with your mouse.

Tip: Whenever you're writing scripts, you'll want to test them every now and then to see if they work the way you expect.

Now let's make Scratchy glide around instead of jumping from point to point. To do this, click the **Motion** palette, drag out five **glide** commands, and join them together. Follow the picture on the right, and copy the seconds and coordinates. Once you're finished, click the script to see the results!

```
glide 0.1 secs to x: 150 y: 30
glide 0.1 secs to x: -100 y: -120
glide 0.1 secs to x: 0 y: 125
glide 0.1 secs to x: 100 y: -120
glide 0.1 secs to x: -150 y: 30
```

Now we can join these two programs together! From the **Events** palette, drag out the **When green flag clicked** command and put it at the top of your two scripts.

Tip: We'll often need multiple scripts to start at the same time, and using the **When green flag clicked** command will help us do that.

```
forever
  go to x: 0 y: 125
  wait 0.1 secs
  go to x: 150 y: 30
  wait 0.1 secs
  go to x: 100 y: -120
  wait 0.1 secs
  go to x: -100 y: -120
  wait 0.1 secs
  go to x: -150 y: 30
  wait 0.1 secs
```

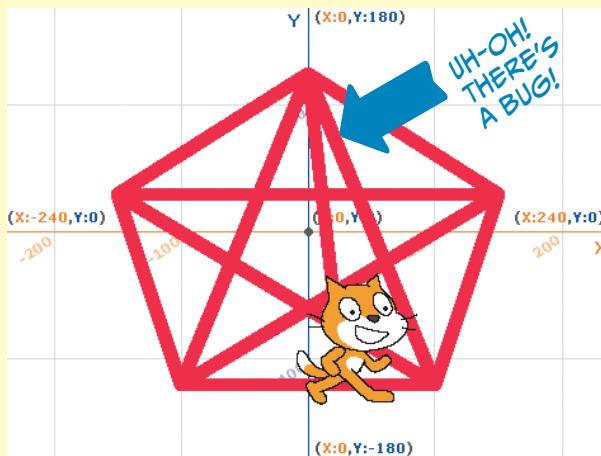
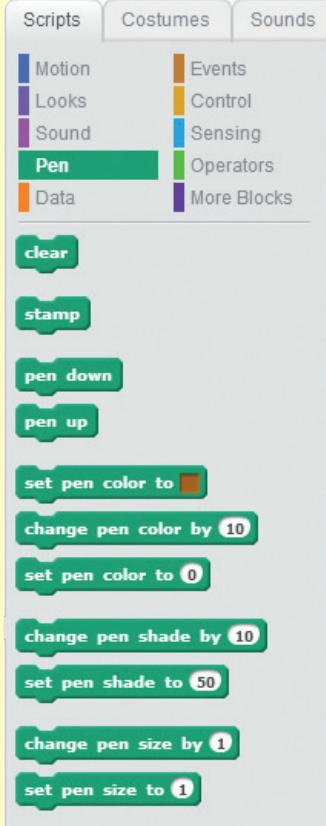
```
when green flag clicked
forever
  go to x: 0 y: 125
  wait 0.1 secs
  go to x: 150 y: 30
  wait 0.1 secs
  go to x: 100 y: -120
  wait 0.1 secs
  go to x: -100 y: -120
  wait 0.1 secs
  go to x: -150 y: 30
  wait 0.1 secs
  glide 0.1 secs to x: 150 y: 30
  glide 0.1 secs to x: -100 y: -120
  glide 0.1 secs to x: 0 y: 125
  glide 0.1 secs to x: 100 y: -120
  glide 0.1 secs to x: -150 y: 30
```

1 STAGE

Because we used the `When green flag clicked` command, we can use these buttons above the Stage to start (green flag) and stop (red circle) the game.



Next, click the **Pen** palette and drag out the four green Pen blocks shown on the right. Now when Scratchy moves, he'll draw a *magic star web*!

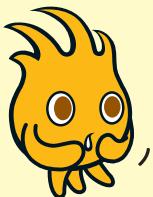


Occasionally, when you run your program, there is a *software bug*. This is the most exciting part of computer programming: discovering an error in something you have made and then solving the problem. In this case, sometimes Scratchy will draw an odd line at the beginning of the program.

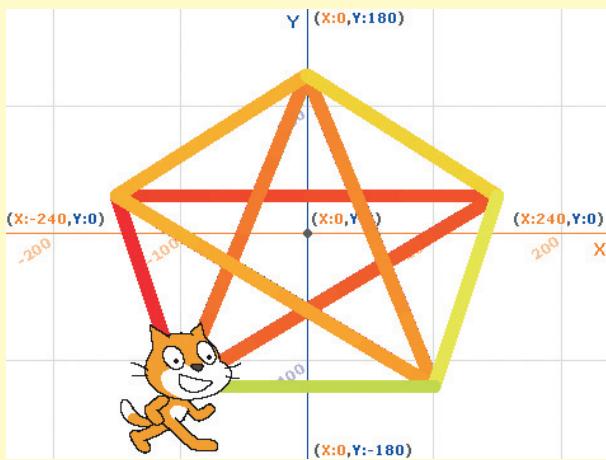
If we drag Scratchy anywhere else on the Stage and then press `green flag`, he draws an extra line because he starts in the wrong place. Try doing this multiple times to see if you can spot the bug.

This software bug can be fixed by adding some more code—that is, new blocks—to your program. In this case, simply place a new `go to` block (from the blue **Motion** palette) above the green Pen blocks and below the `when green flag clicked` block.

With this little correction, Scratchy will always begin drawing from the correct position in the grid. The bug is gone!



```
when green flag clicked
  clear
  set pen color to red
  set pen size to 10
  pen down
  forever
    go to x: 0 y: 125
    wait 0.1 secs
    go to x: 150 y: 30
    wait 0.1 secs
    go to x: 100 y: -120
    wait 0.1 secs
    go to x: -100 y: -120
    wait 0.1 secs
    go to x: -150 y: 30
    wait 0.1 secs
    glide 0.1 secs to x: 150 y: 30
    glide 0.1 secs to x: -100 y: -120
    glide 0.1 secs to x: 0 y: 125
    glide 0.1 secs to x: 100 y: -120
    glide 0.1 secs to x: -150 y: 30
```

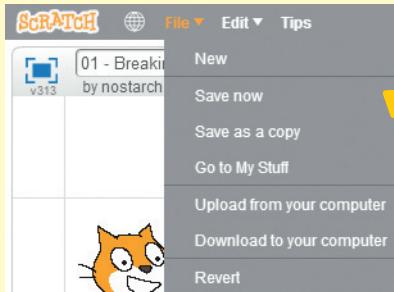


Let's add a whole new program to make a magic star web that changes colors. Build a second stack of blocks that uses the `change pen color by` command and see what happens.

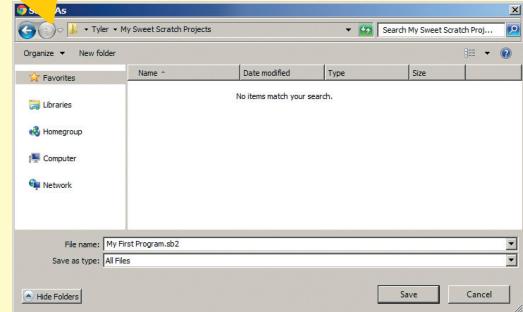
Isn't that cool? You can give a single sprite more than one set of blocks! Scratchy now has two programs. This tiny second program sure makes a big difference in how the game looks.

```
when green flag clicked
  forever
    change pen color by 1
```

1 STAGE



Remember to save this file so you can play with it later!



If you are logged into Scratch, the website stores all of your projects into **My Stuff** so you can easily find them. The website saves your progress every so often, but you can save manually too: **File ▶ Save Now**. You can also save different versions of your programs to make sure you don't lose older versions of your games and can safely experiment—**File ▶ Save Copy** creates a new version of your project in My Stuff. If you want to download a version for yourself, try **File ▶ Download to your computer**. Then save it in a safe spot!

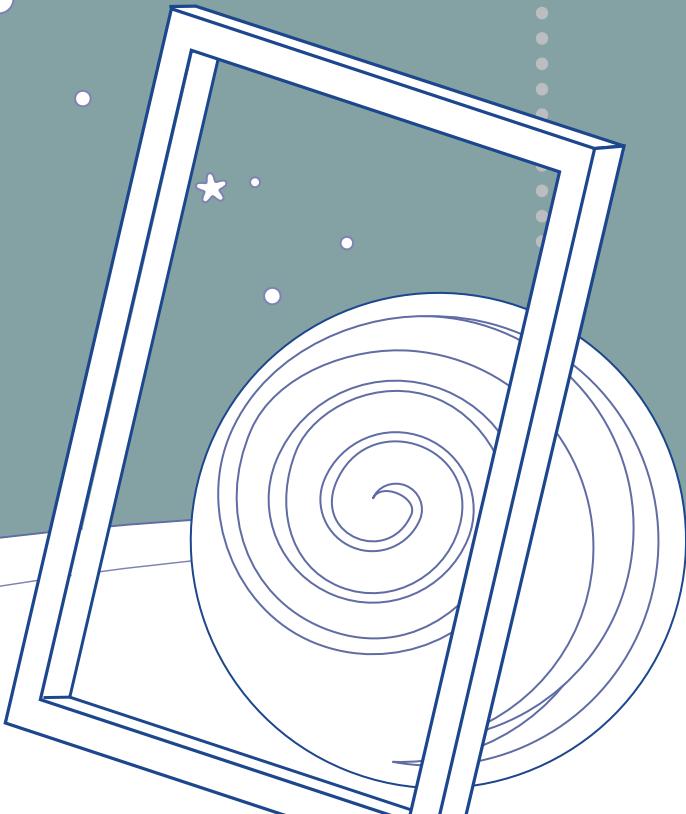
Scratchy's Challenge!!

Can you edit this program to make Scratchy draw different kinds of shapes? Give it a try!



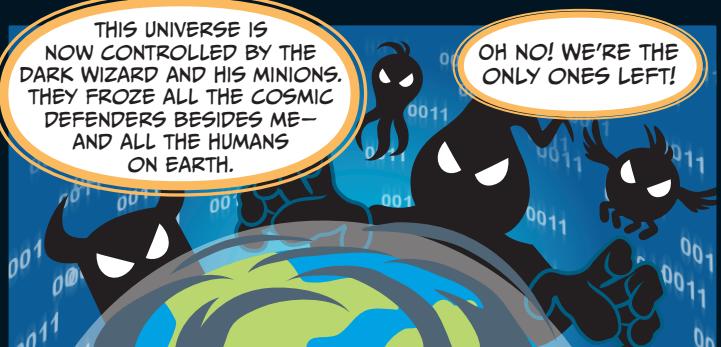
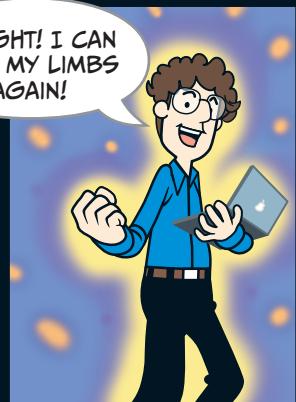
**ENTERING
SPACE**

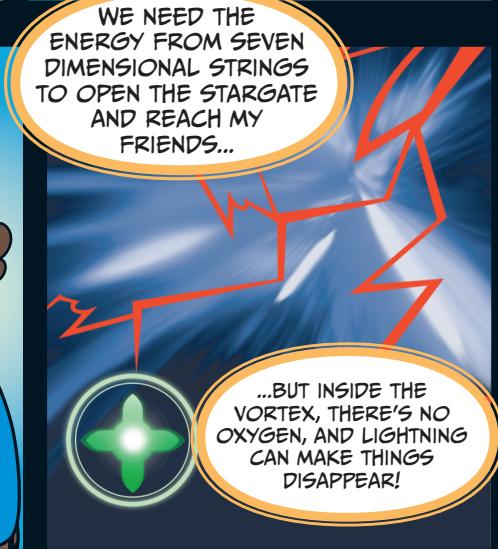
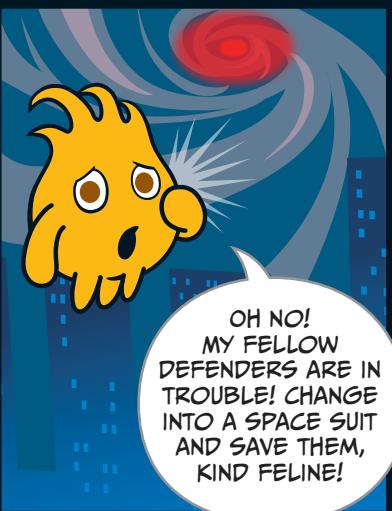
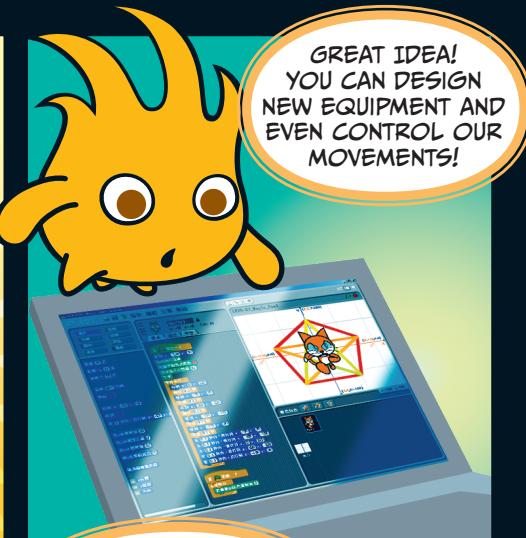
2
STAGE



STAGE

2





2 STAGE

A SPACE ODYSSEY!

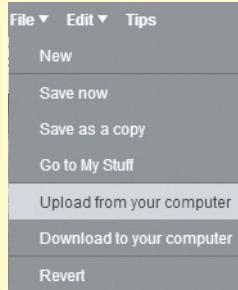
Chapter Focus

Learn to design new costumes and program a sprite's movements, reactions, and sound effects.

The Game

Avoid the lightning bolts and collect seven dimensional strings. Once you've got them all, the Monolith will appear!

To make things really easy, let's start by opening a blank project called **02 - A Space Odyssey.sb2**. This project has all the sprites you'll need, but none of the programming yet. To open a file, click **File > Upload from your computer**.

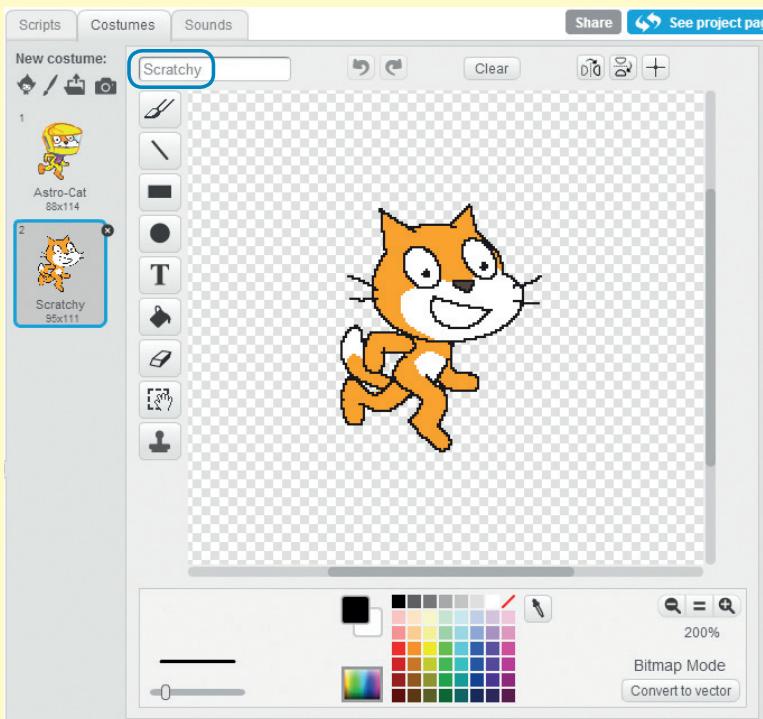


But let's try making some sprites of our own, so you can make changes to this game's characters and invent your own games, too! Click Scratchy's sprite icon in the Sprite List, and then click the **Costumes** tab. You'll see the Paint Editor—just be sure to click the costume you want to change.

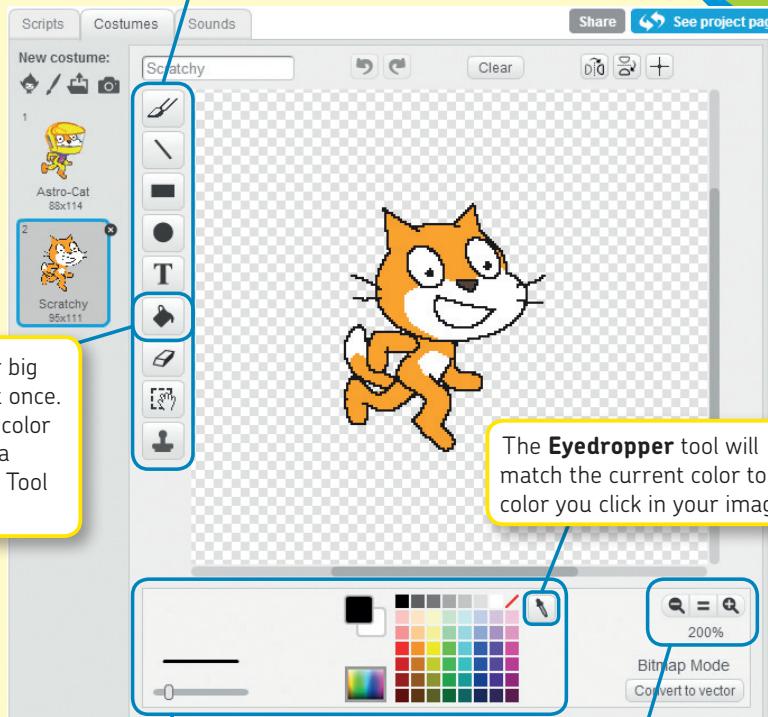
At the top of the Paint Editor, you can give your Costume a name. We can then reference the costume names in our programming.

If your Paint Editor looks different, it could be because you haven't opened the blank project file (**02 - A Space Odyssey.sb2**) that has Scratchy's astronaut costume.

Scratch has two modes for editing graphics—on the right is **Bitmap** mode. See page 38 to learn more about editing in Vector mode.



Here's where all the tools are. The **Brush** and **Eraser** tools make it easy to draw.



Use the **Fill** tool to color big parts of your drawing at once. You can choose a single color from the palette or use a gradient effect from the Tool Options.

The **Eyedropper** tool will match the current color to any color you click in your image.

Tool Options

Whatever tool you are using, the options for the tool will appear at the bottom of the Paint Editor. For example, the size of the Brush or Eraser can be adjusted if you want to make a big drawing or add fine detail. Just click and drag here to pick the right size.

You pick a color for your tool here, too.

Click the **Zoom** buttons (the magnifying glasses on the bottom right) to zoom in or out on your creations. This will make it easier to draw! Clicking the equals sign (=) shows you exactly how your sprite will appear on the Stage.

If you're not sure what a button does, simply hover your cursor over it, and a description will appear!

2

STAGE

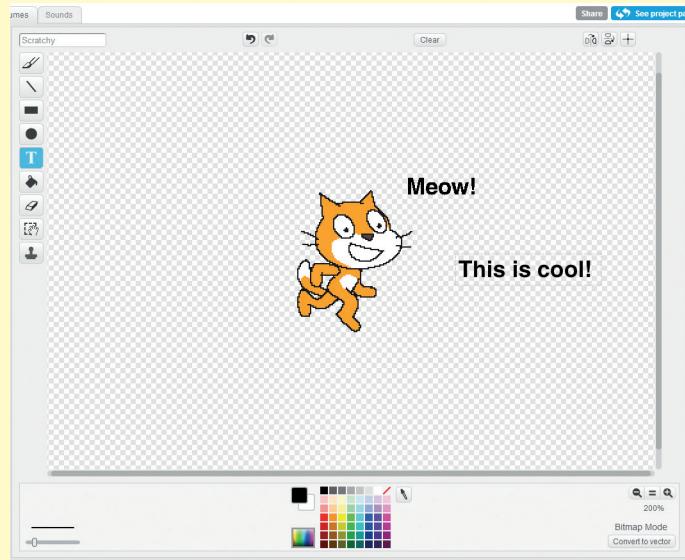


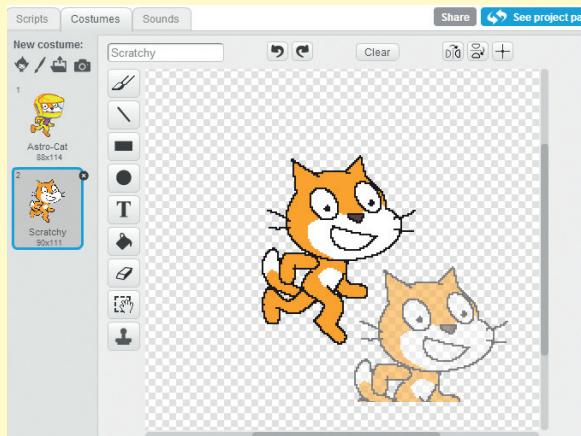
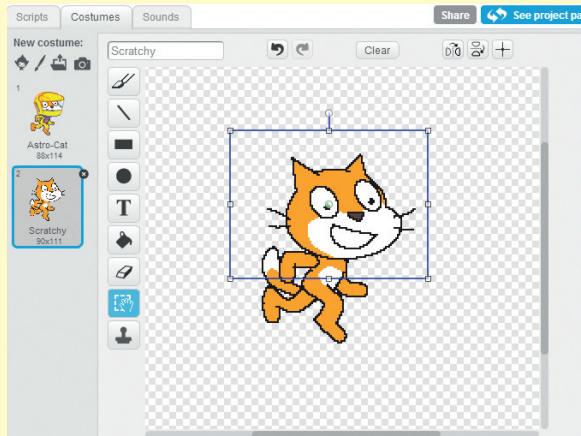
You also have tools to draw rectangles and ellipses. Can you give Scratchy a stovepipe hat like Abe Lincoln using the **Rectangle** and **Ellipse** tools?

These shapes can be empty inside or filled in. Try experimenting with different colors for the inside and outside. If you press the SHIFT key when you start to draw, you'll have a perfect circle or square! (You can also use this SHIFT trick when using the **Line** tool to draw a straight line.) Try rotating your shapes using the handle on the top of the box.



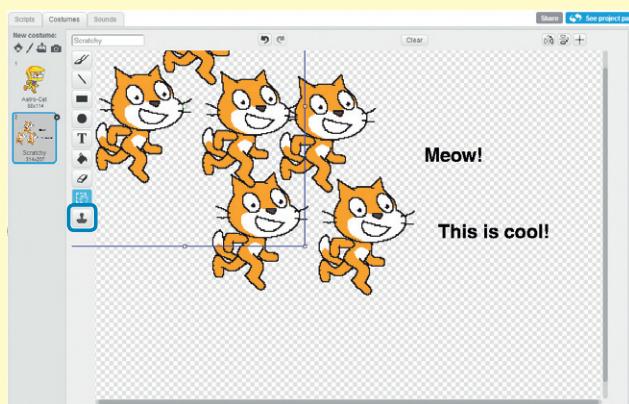
The **Text** tool lets you add writing to your sprite. We'll use this tool when we need to give the player instructions for our games. If you want to move the text, simply click and drag the black box that surrounds your text.





To use the **Select** tool, use your mouse to create a frame around a certain area. Then you can do all sorts of things to the selected part of your costume:

- Click and drag the selection to move it to a new location.
- Resize, smush, or stretch the image using the handles on the sides of the box
- Rotate the selection by clicking and dragging the handle at the top center of the box
- Press and hold the **CTRL** key and **C** key at the same time to copy the image area (Mac users can use **⌘-C** instead). Then press **CTRL-V** to paste your selection, as many times as you like.
- Press the **DELETE** key to erase the selection.



The **Set costume center** button marks the center of your sprite. This helps to make sure your sprite doesn't end up in the wrong place when it spins or rotates!



By using the **Duplicate** tool, you can copy and stamp a selected area as many times as you want! Just draw a frame around the area you want to copy and then click wherever you want to paste.

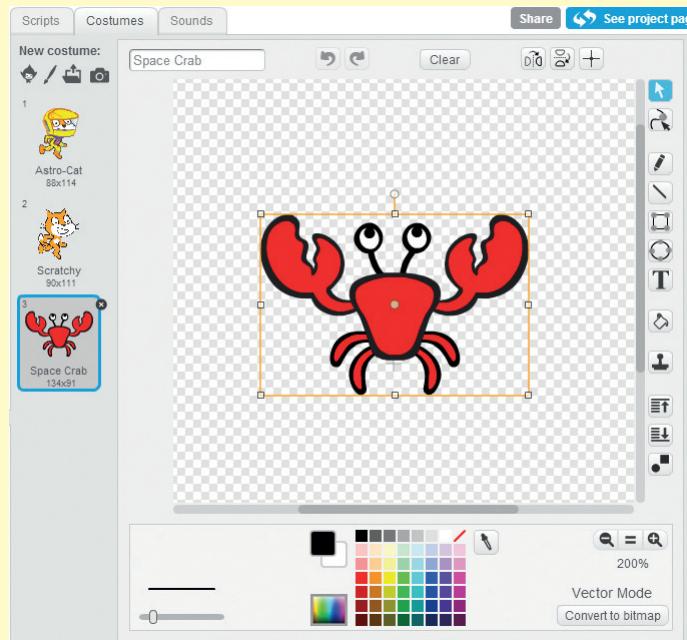
2

STAGE

Vector Mode

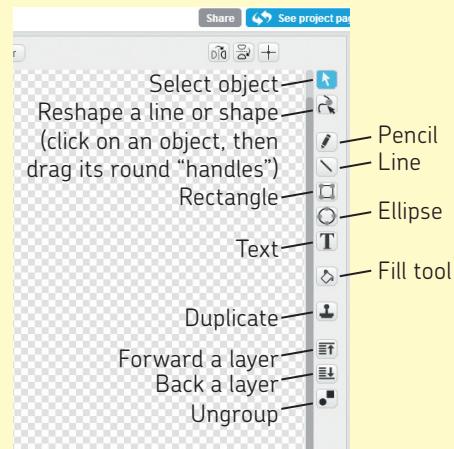
You may have noticed that when you edit other sprites in Scratch, you don't see the same Paint Editor tools. Some newer sprites are *vector* art—that's just a fancy way to say they're made of shapes, instead of pixels. Vector art have small filesizes, but they are great quality—and they can be resized without losing quality.

Note: For simplicity's sake, all of the graphics in this book use Bitmap mode. But your custom projects can use a mix of vector and bitmap graphics.

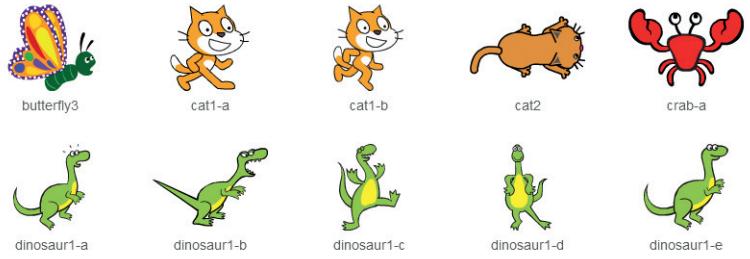


You can switch from Scratch's **Bitmap** mode (the one seen earlier) to **Vector** mode by clicking the **Convert to Vector** button at the bottom right of the Paint Editor. The difference between using these two tools in Scratch is like the difference between Adobe Photoshop and Illustrator—or GIMP and Inkscape. Use whichever Paint Editor mode you like the most!

You can import SVG files into Scratch's vector editor. In Vector mode, you can squeeze and shape lines, reshape, and ungroup. Here's how the Vector mode works.

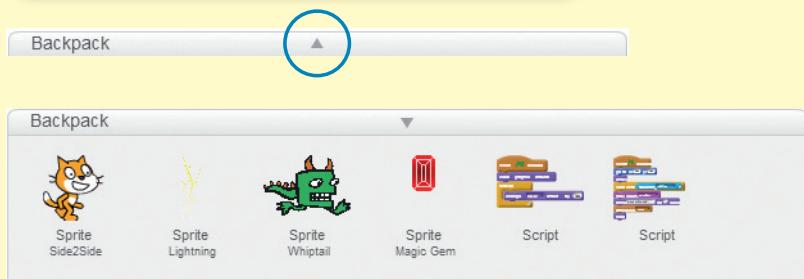


Try opening a vector graphic from Scratch's library, and give editing one a try.



The Backpack

Here's a cool new feature. If you're logged into the Scratch website, you'll see something called the **Backpack** at the very bottom of the screen. Click it, and it'll open up. Yours will be empty until you throw some sprites in it.



Your Backpack lets you share sprites and scripts between projects. If you play a really cool game on the Scratch website and want to use the character in an entirely new project, just click and drag the sprite right into your Backpack.

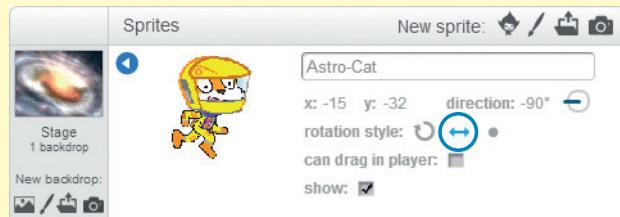
When you create a new project of your own, just open the Backpack again and drag the sprite out. You can write all new programs, or use the ones that were already with the sprite. You can even use your Backpack to store programs you want to reuse!

2

STAGE

Once you know how to use the Paint Editor's tools, Scratchy can put on his space suit! Go ahead and draw your own, or use the costume that's already in the project.

Because we've selected the horizontal rotation style (circled below) Scratchy will face only left and right.



Now we have the main character for our game: Scratchy the astronaut!



Next, let's take a look at the other sprites in the game. You can use the art that's already in the game, or draw new artwork yourself! Click  to draw a new sprite.

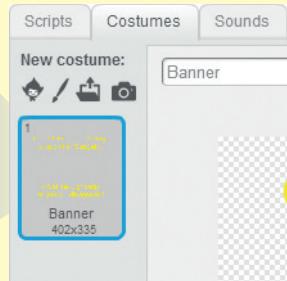
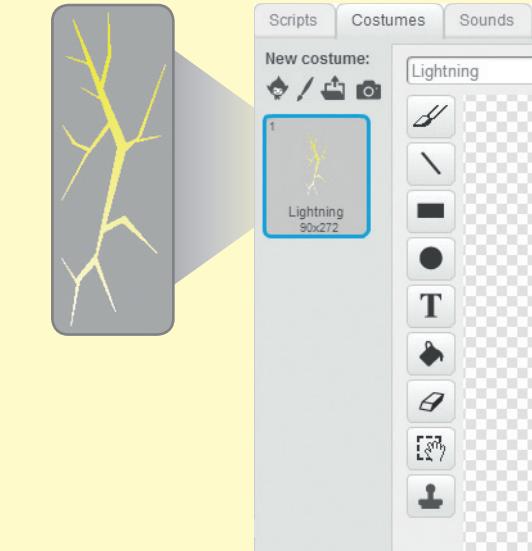
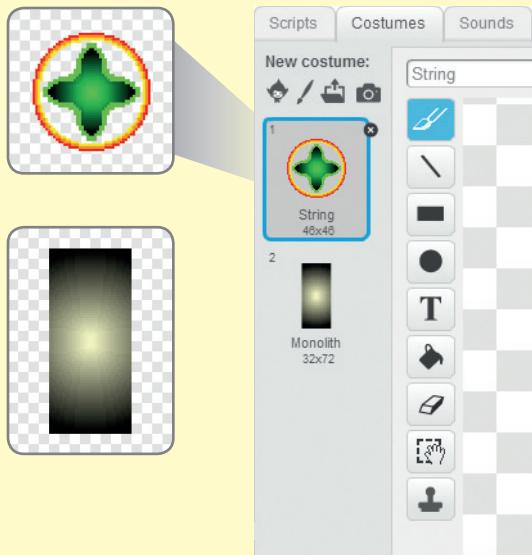
First, take a look at the String and the Monolith. They are two costumes for the same sprite, **String**. If they were two separate sprites, we'd have to write two programs. But now we can make this sprite switch costumes and write only one program.

Now for our third new sprite, some scary **Lightning**! The player will need to avoid the lightning.

We also need some instructions to appear at the start of the game. We'll call this sprite **Banner**.

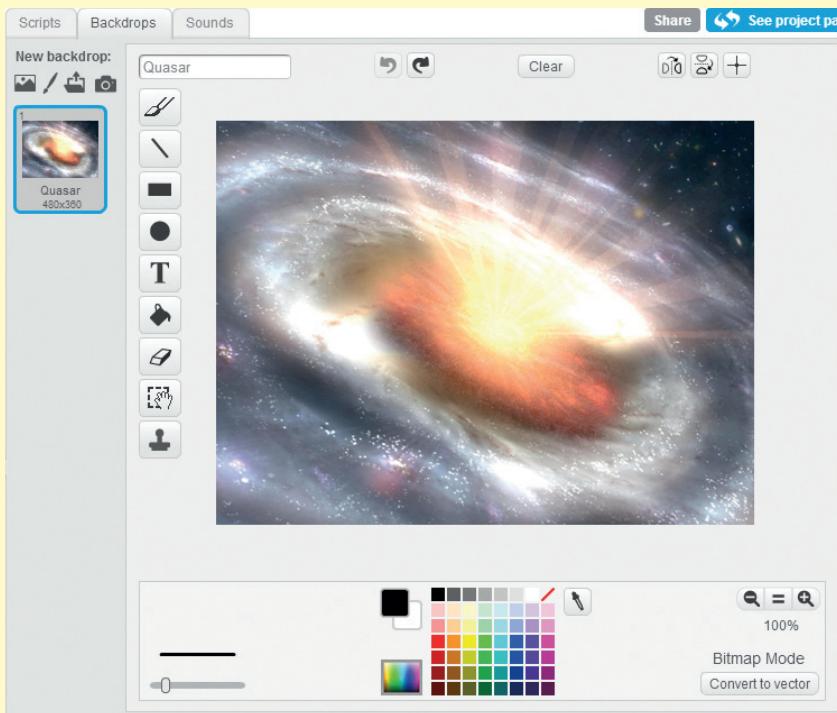
Get 7 Dimension Strings
to open the Stargate!!

Avoid the Lightning
or you will disappear!!



2 STAGE

Next, let's look at the Stage. I used artwork of a black hole from NASA! You can draw a new backdrop if you like. Click the Stage in the Sprite List, and then click the **Backdrops** tab.



Now that we have a bunch of sprites for the game, you can see how everything appears in the Sprite List. To give a sprite new instructions or costumes, you'll first have to click it in the Sprite List. Let's start by giving Scratchy the astronaut his programming.

Let's write our first program ① for Scratchy! Make sure he's selected in the Sprite List and you've clicked the **Scripts** tab. His first program is a short one that makes him bounce up and down a little. This makes him look like he's floating in zero gravity!



2

```

when green flag clicked
  point in direction 90
  go to x: 0 y: 0
  wait (1) secs
forever
  if [key up arrow v pressed? then
    change y by (15)
  ]
  if [key down arrow v pressed? then
    change y by (-15)
  ]
  if [key left arrow v pressed? then
    point in direction (-90)
    change x by (-15)
  ]
  if [key right arrow v pressed? then
    point in direction 90
    change x by (15)
  ]

```

For program 2, we'll make a *conditional*—if something is true, then something else will happen. In the **Control** palette, drag out an **if** block. Then for the diamond shape, drag the **Sensing** block **key [up arrow] pressed?**. Right below the **if**, put what you want to happen when the statement is true. Drag out the rest of these commands to form the complete program. Now you can move Scrathy up, down, left, and right by using the keyboard!

Now we'll give Scrathy two more programs. We'll need to program them individually, and then use **When green flag clicked** to make them all run at the same time.

3

```

when green flag clicked
  switch costume to [Astro-Cat v]
forever
  go to front

```

Let's write programs 3 and 4. Click the **Control** and **Looks** palettes and drag out these commands.

Program 3 controls which costume Scrathy wears, and program 4 makes Scrathy become invisible like a ghost each time he gets struck by lightning.

When you've finished all of this, Scrathy's programming is complete!

4

```

when green flag clicked
  clear graphic effects
forever
  if [touching [Lightning v] ? then
    repeat (10)
      change ghost effect by (1)
    end
  ]

```

Next, let's click the **Banner** sprite. We just need a simple program to make these instructions appear at the start of the game. The **repeat 2** loop using the **show** and **hide** blocks makes our instructions flash, so the game is even more exciting.

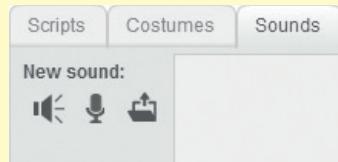
```

when green flag clicked
  hide
  go to x: 0 y: 0
  go to front
  repeat (2)
    show
    wait (0.4) secs
    hide
    wait (0.1) secs

```

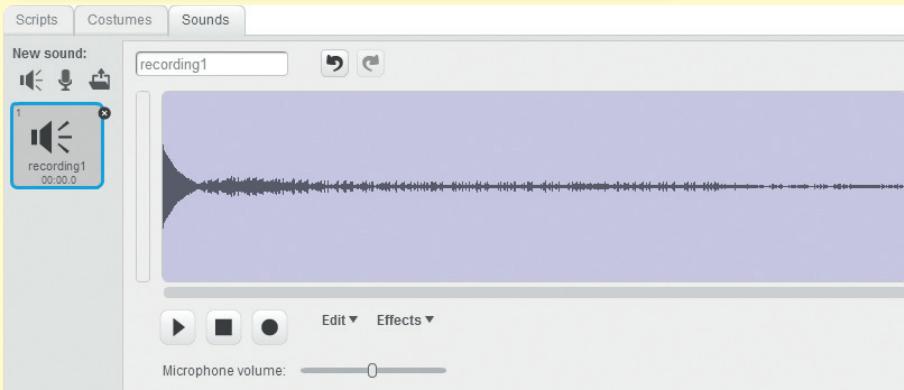
2 STAGE

Now we can add sound effects to the game! I've already added a few, but you can change things up. First, click the **Stage** in the Sprite List. Then click its **Sounds** tab. You can create whatever kind of sounds effects or music you like for your Scratch projects. You can even record your own sounds right in the Scratch program.

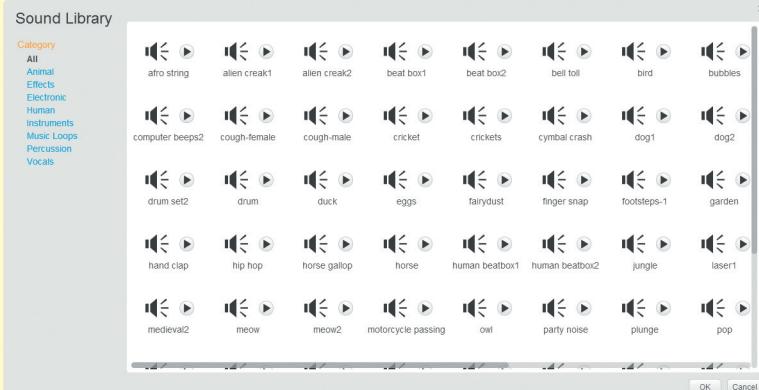


If you click the **Record** button, a sound recorder will pop up. You can click the round button to record speech or sound effects through a microphone. When you're finished, click **OK**.

Note: To record your own sounds, you'll need a microphone attached to the computer. To listen to sound effects and music, you'll need speakers.



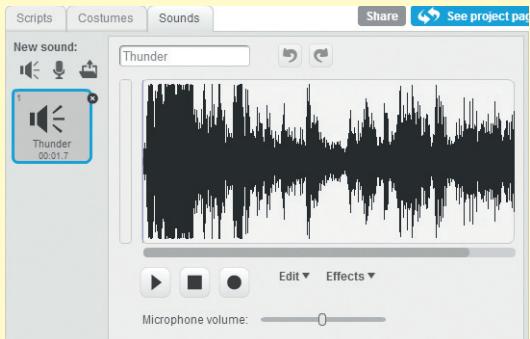
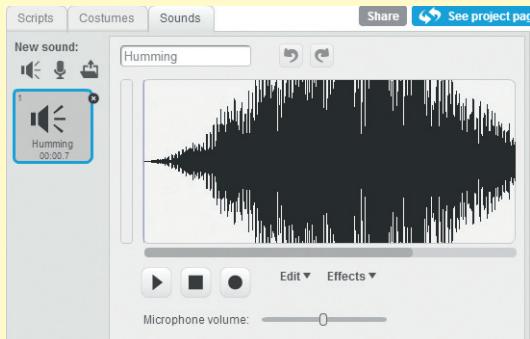
If you want to use sounds that are prerecorded, you can press to use Scratch's sound library, or to choose files from your own computer (MP3 and compressed WAV, AIF, and AU formats are supported).



Now we can add some simple programs to the Stage. Program 1 makes its backdrop change colors. In program 2, use the **Sound** palette to add a song to the Stage.



Next, we can add some sound effects to the String and Lightning sprites to make the game more exciting! Test how you like my sound effects, and make your own if you like.



You can record a sound yourself and then change it using the **Effects** menu. Try reversing what you record to make it sound really weird!

Click the **Lightning** sprite, and write a program so that whenever Scratchy touches a lightning bolt, a sound will play.



2

STAGE

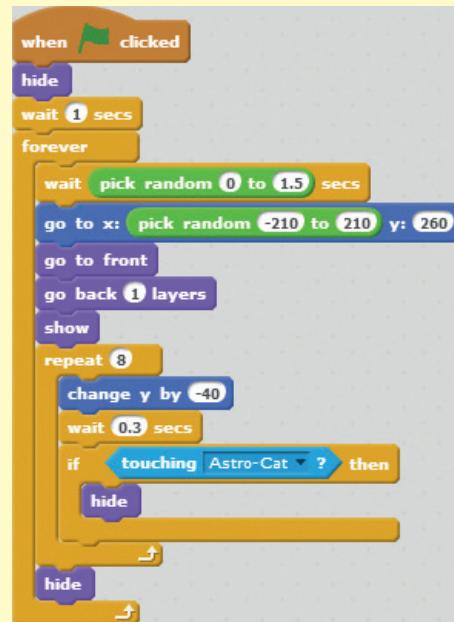
The **Lightning** sprite needs some more programs. Go to the **Control**, **Events**, **Looks**, and **Operators** palettes and program these commands to have the lightning bolt randomly grow bigger or smaller, making the game more magical.



Next, write this program to make the lightning disappear whenever Scratchy touches it and to control the way it moves.

The lightning's vertical position (y-axis) changes because we repeat eight times the subtraction of 40 steps (-40) from its original y-coordinate of 260. To make the lightning move differently, you can change and play with these numbers.

So that the lightning bolt makes Scratchy disappear, we must make sure that each time it moves—that is, the position of its y-axis changes—the program will check if it touches Scratchy.



Tip: Sometimes when you've used the **hide** and **show** blocks, a sprite can disappear while you're working on the program—running it, testing it, and checking for bugs. Simply click the **show** block in the **Looks** palette to make the sprite appear again. (You can also check the **show** box in the Sprite Information pane.)

Now it's time to program the **String** sprite. Make sure you click it in the Sprite List first! Program ① makes it change color, just like our Stage. Program ② will give it a simple animation, using the fisheye effect.

① when green flag clicked
clear graphic effects
forever
change color effect by 5

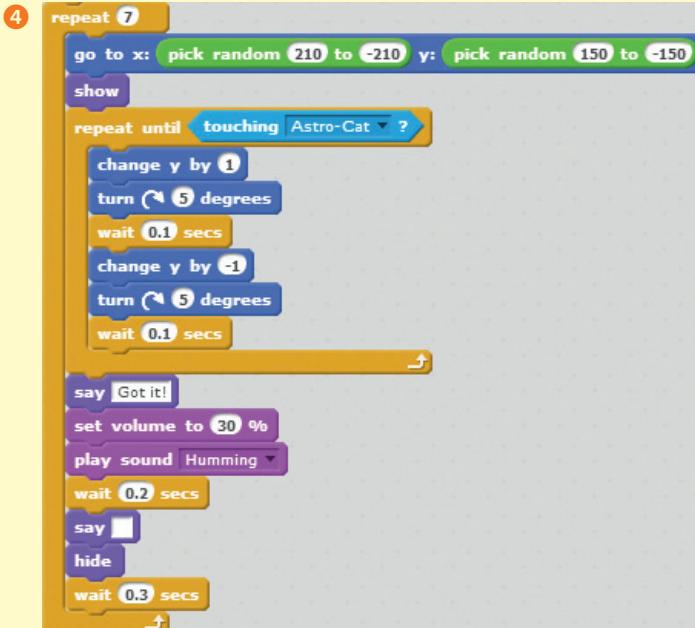
② when green flag clicked
forever
change fisheye effect by 30
wait 0.1 secs
change fisheye effect by 30
wait 0.1 secs
change fisheye effect by -30
wait 0.1 secs
change fisheye effect by -30
wait 0.1 secs

Now for a big program. Let's start by dragging out the blocks you can see in ③. These will control how the String costume spins and moves.

③ repeat until touching Astro-Cat ?
change y by 1
turn (5 degrees
wait 0.1 secs
change y by -1
turn (5 degrees
wait 0.1 secs



Then add to your program so that it looks like ④. This will make your dimensional string appear in a random place on the Stage seven different times. The `say` blocks and `play sound` blocks at the end of the program make sure the player knows he has grabbed a dimensional string.



We're not done yet!
This is a big script.



Add a `when green flag clicked` block at the top of our script and some instructions at the very bottom so that once Scratchy has collected seven dimensional strings, the String sprite will change to its Monolith costume. When that happens, the player wins the game. Make sure your finished program looks like 5.

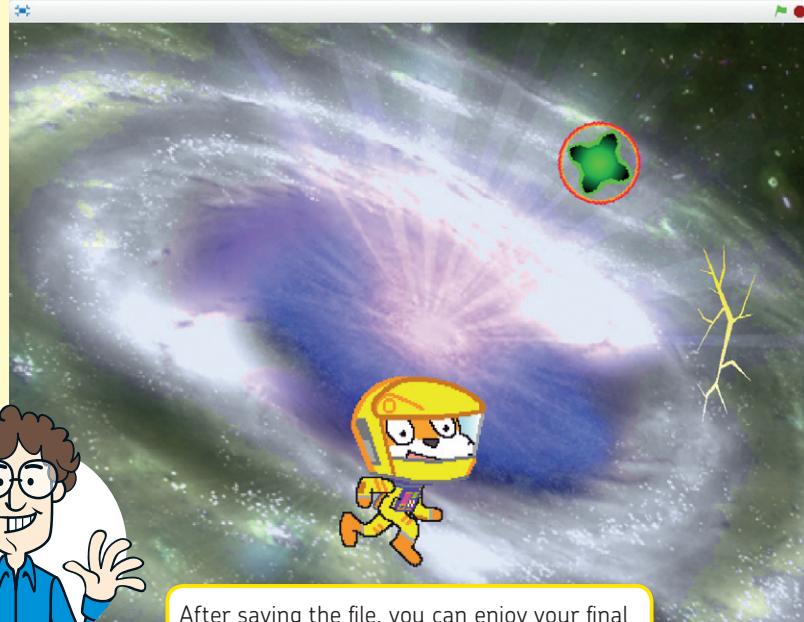
5

```
when green flag clicked
  switch costume to [String v]
  hide
  wait (1) secs
  repeat (7)
    go to x: (pick random (210) to (-210)) y: (pick random (150) to (-150))
    show
    repeat until [touching [Astro-Cat v] ?]
      change y by (1)
      turn (5) degrees
      wait (0.1) secs
      change y by (-1)
      turn ((-5) degrees)
      wait (0.1) secs
    end
    say [Got it!]
    set volume to (30) %
    play sound [Humming v]
    wait (0.2) secs
    say [ ]
    hide
    wait (0.3) secs
  end
  go to x: (0) y: (0)
  point in direction (90)
  switch costume to [Monolith v]
  go to front
  go back (2) layers
  show
  say [Stargate opened!] for (2) secs
  stop all
```

Now you're done!
Nice work!

2

STAGE



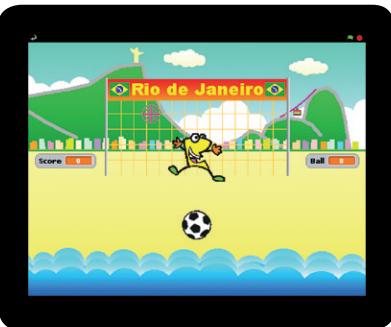
After saving the file, you can enjoy your final creation! Make the Stage full screen and click  to begin a new round.

Scratchy's Challenge!!

Add more lightning bolts to give yourself a challenge. Or you could replace the lightning bolt with a big, scary space monster you drew yourself! Give it a try!



THE ADVENTURE CONTINUES...



IT'S SUDDEN DEATH IN THIS PENALTY KICK SHOOTOUT!



DEFEND THE SERVER FROM SNEAKY HACKER ATTACKS!



GET THE GEM AND ESCAPE THE BOOBY-TRAPPED MAZE!



BATTLE THE DARK WIZARD IN AN AWESOME FIGHTING GAME!

Buy the book at <http://nostarch.com/scratch> and use **SCRATCH2** to get 40% off your purchase.