

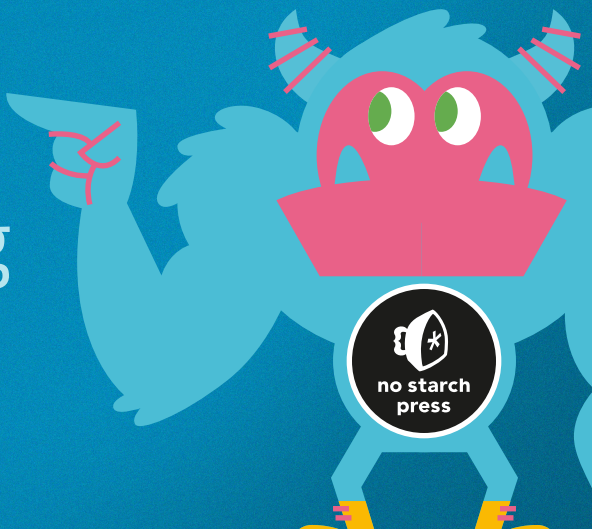
SCRATCH 3

GAMES

FOR KIDS

A Playful Guide to Coding

MAX WAINWRIGHT



25 SCRATCH 3 GAMES FOR KIDS



SCRATCH 3 GAMES FOR KIDS

MAX WAINWRIGHT



25 SCRATCH 3 GAMES FOR KIDS. Copyright © 2019 by Max Wainewright

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-10: 1-59327-990-6

ISBN-13: 978-1-59327-990-5

Publisher: William Pollock

Production Editor: Laurel Chun

Cover Design: Derek Yee

Interior Design: Derek Yee

Illustrator: Garry Booth

Developmental Editor: Annie Choi

Copyeditor: Sharon Wilkey

Compositor: Kim Scott, Bumpy Design

Proofreader: Emelie Burnette

For information on distribution, translations, or bulk sales, please contact No Starch Press, Inc. directly:

No Starch Press, Inc.

245 8th Street, San Francisco, CA 94103

phone: 1.415.863.9900; info@nostarch.com

www.nostarch.com

A catalog record of this book is available from the Library of Congress

No Starch Press and the No Starch Press logo are registered trademarks of No Starch Press, Inc. Other product and company names mentioned herein may be the trademarks of their respective owners. Rather than use a trademark symbol with every occurrence of a trademarked name, we are using the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The information in this book is distributed on an "As Is" basis, without warranty. While every precaution has been taken in the preparation of this work, neither the author nor No Starch Press, Inc. shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in it.

DEDICATION

To past and present pupils of New End Primary School

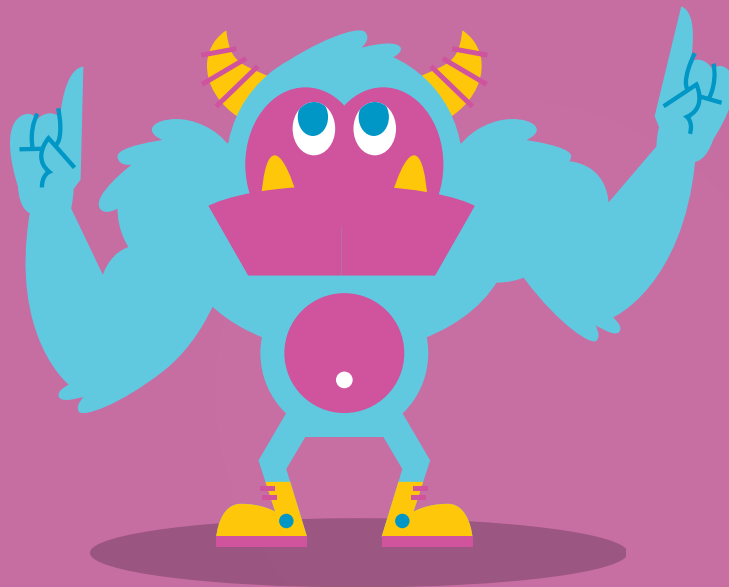


TABLE OF CONTENTS

ACKNOWLEDGEMENTS INTRODUCING SCRATCH 3

VII
VIII

1: GAME ON!

Cat on the Move	4
Cat and Mouse	6
Fish in the Sea	8
Bat in a Flap	10
Kiran and the Magic Crystals	12
Challenges	14

2: DRAWING BACKGROUNDS AND SPRITES

Down the River	18
Monster Maze	20
Penguin Bobsled Cross	22
Color Bounce	24
Racetrack	28
Challenges	32



3: WHAT'S THE SCORE?

Penguin Ski School	36
Shark Diver	40
Fruit Catcher	44
Brick Tricks	48
Colored Walls	52
Challenges	56

4: MATH MATTERS

Super Snake	60
Cactus Jumper	64
Tower Builder	68
Robot Power Up	72
Invasion of the Donuts	76
Challenges	82

5: MAKING YOUR OWN BLOCKS

Cave of Stars	86
Space Tacos	90
Two-Player Soccer	94
Rocket Racers	100
Catch the Snowflakes	106
Challenges	112

ACKNOWLEDGEMENTS

A big thanks to all my family for their support—Rachel, Linus, Elsa, and everyone else.

Thanks to all at MIT/Scratch for creating such a fantastic website.

Thanks to everyone, especially the pupils, at New End Primary School for inspiring me.

Thanks to everyone at No Starch Press for their encouragement, support, and patience.

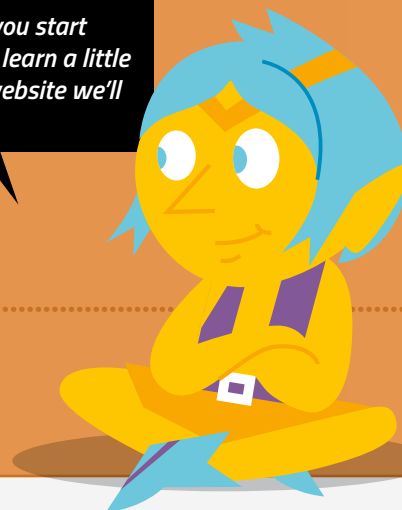
Thanks to previous colleagues and friends who have supported and championed my ideas, including Daniel, Anthony, Nigel, Lewis, Rob, Ben, Rich, Samson, Karyn, Dan, Donna, Lorraine, Jyoti, Charlie, Gillian, Maxime, and Natalie. Thanks to Anne for the advice and Bob for listening



INTRODUCING SCRATCH 3

In this book, you'll learn how to code your own games with Scratch 3. You'll use different blocks to create code. You'll also learn how to add objects called *sprites* and use code to tell them what to do. At the end of each chapter, you'll find challenges to help you take your skills to the next level!

But before you start coding, let's learn a little about the website we'll be using.



THE SCRATCH EDITOR

The Scratch Editor is where you create projects in Scratch. Here are its main parts:

Menus Use the File menu to save or load your projects.

Coding Area Drag and drop the blocks here and snap them together to create your program.

Start and Stop Click the green flag to start your code, or the stop sign to stop it.

Stage Size Use these buttons to make the stage smaller, set it to normal size, or fill the screen.

Main Tabs Switch from coding, drawing, or adding sound.

Block Categories Code blocks in the Block Palette are sorted into colored groups. Click a category to scroll to the blocks in that category.

Code Blocks The code blocks in the current group are shown here. Scroll down if you can't find the block you want.

Tutorials Some help videos are shown here. Click the white X to close this panel.

Sprite List All the sprites in your project are shown here. The selected sprite has a blue border around it.

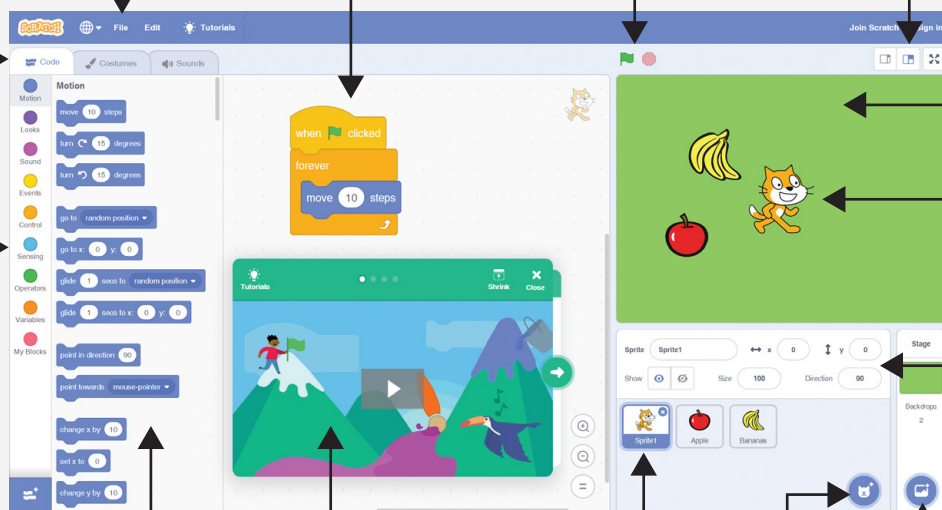
Choose a Sprite Add a new sprite to your project.

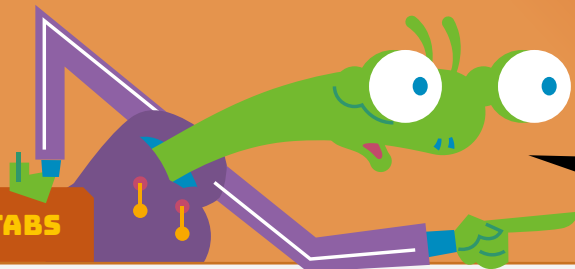
Choose a Backdrop Add a new background to your project.

The Stage Your program will run here.

Sprites Objects that move around on the stage.

Sprite Info Displays information about each sprite, such as how big it is.





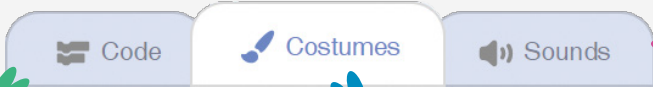
Scratch has so many code blocks, sprites, and sounds to learn about.

Don't panic—you'll learn how to use them in no time!

THE MAIN TABS

You'll need to switch between the three main areas of the Scratch Editor while you're creating your games. These areas let you code, change how your sprites look, or add sound to a sprite.

The Code tab lets you drag code blocks into the coding area.



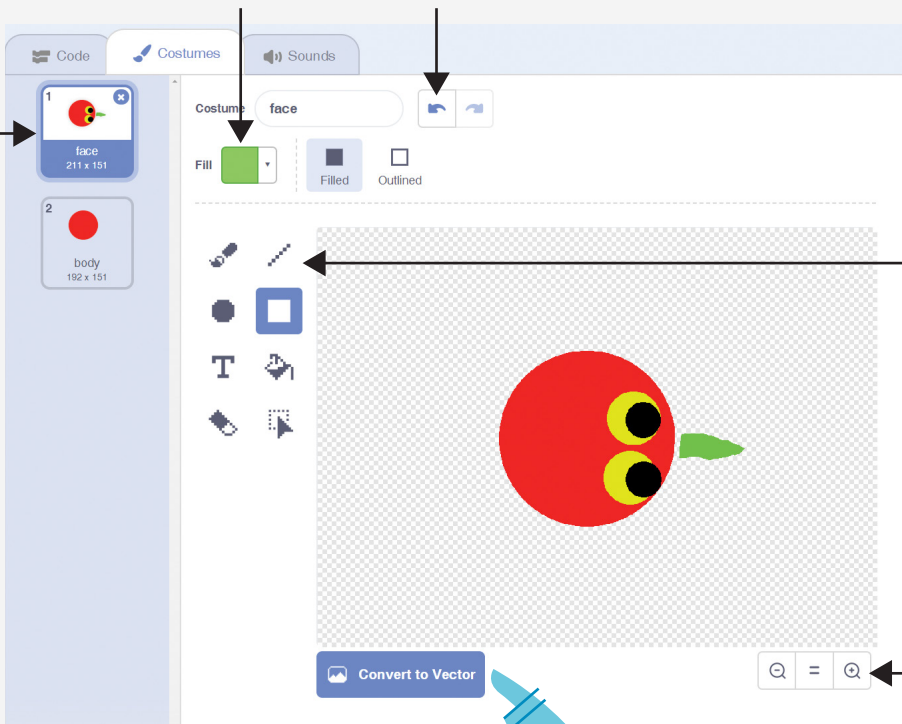
Click the Costumes tab to change how your sprite looks.

The Sounds tab allows you add a new sound to your project.

COSTUMES TAB

Fill Use this to set the color you're drawing with.

Undo Click this if you make a mistake while you're drawing.



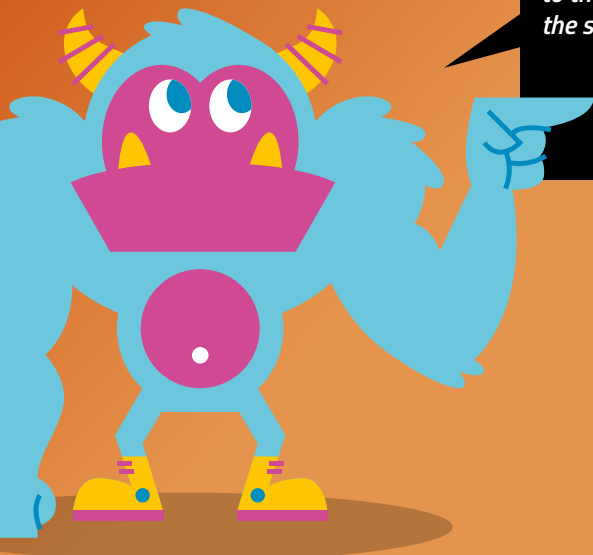
Costumes Just as you have more than one set of clothes, a sprite can have more than one look. Scratch calls these looks *costumes*. You'll use different costumes to make simple animations.

Drawing Tools Use these to draw on your sprite.

Zoom Tools Use these to zoom in and add detail to your sprite costume.

New Costume Add a new costume to your sprite.

Click **Convert to Vector/Bitmap** to store images as vectors or bitmaps. Vectors will look sharper, but bitmaps are much simpler to create.



When you're drawing on the backdrop, the Costume tab changes to the Backdrops tab. The drawing tools for backdrops work in just the same way.

Code

Backdrops

Sounds

DRAWING TOOLS (BITMAP)



Brush Use the mouse to draw on your sprite.



Line Hold the mouse and drag to make a straight line.



Circle Drag to make an ellipse. (Hold down Shift to make a circle.)



Rectangle Drag to make a rectangle shape.



Text Type on top of your sprite.



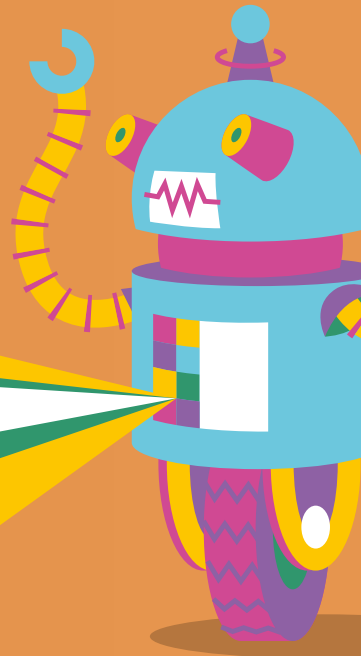
Fill Color in areas within your sprite.



Eraser Rub out part of your sprite. (Undo is better if you made a mistake!)



Select Select a part of your sprite to flip, copy, or rotate.



COLOR SLIDERS

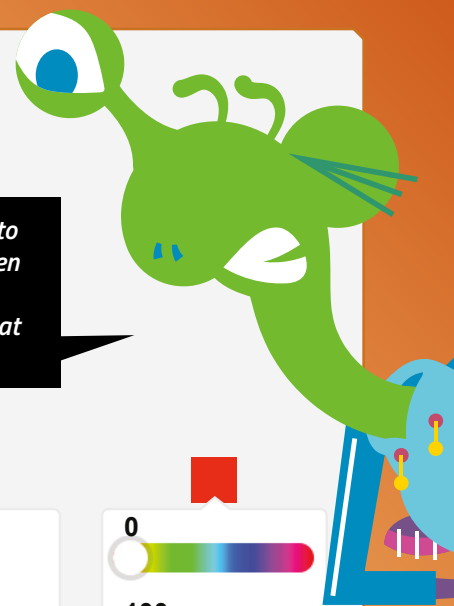
Color 16

Saturation 100

Brightness 66

You can move the sliders around to create more than 1 million colors! But finding the exact shade you want can be tricky.

Use the values below to make a basic color; then adjust the sliders and experiment to find what you need.



0

100

0

67

1

57

0

0

84

0

0

100

0

100

100

11

100

100

16

100

100

40

100

98

40

100

65

62

34

100

88

32

100

61

100

100

74

81

67

88

78

100

11

100

67

GAME ON!

In this chapter, you'll learn the basics of making a game with Scratch. Here are some blocks you'll be using.

INPUT BLOCK

An *input* is an action that tells a computer to do something. You can use *input blocks* to make code run when the player does something, like pressing a key or moving the mouse.

when right arrow key pressed

This input block makes code run when you press a key.

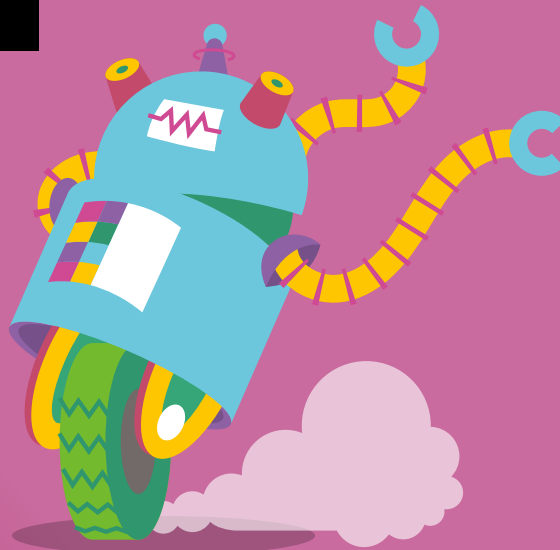


Look out! I'm on the move.

MOTION BLOCK

This motion block makes sprites move.

move 10 steps



FOREVER BLOCK

Any code you put inside this C-shaped loop block will keep repeating forever!



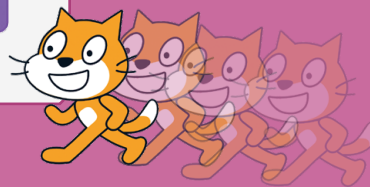
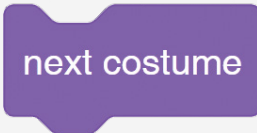
forever

Help, I'm stuck in this loop, and I'm getting dizzy!



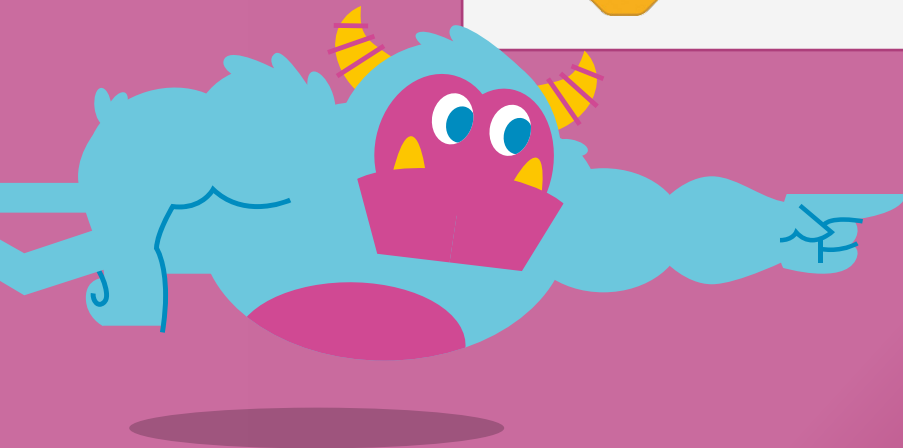
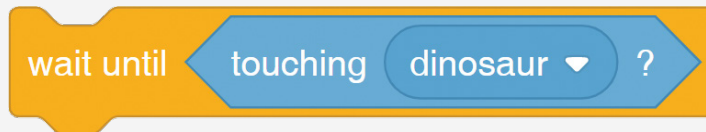
NEXT COSTUME BLOCK

This block tells a sprite to change into a different costume, making it look slightly different. You can use it to create simple animations.



WAIT UNTIL BLOCK

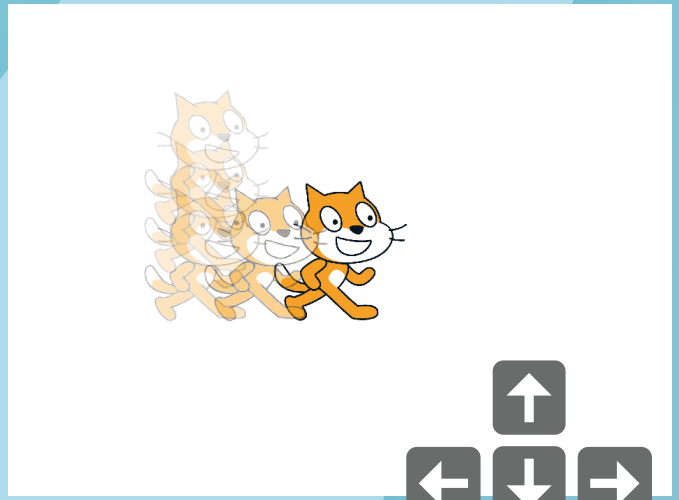
You can combine two blocks to make code that runs when one sprite bumps into another.



The best way to learn more about these code blocks is to start coding. Let's go!

CAT ON THE MOVE

For our first game, you'll learn how to make objects move and change direction. You'll use input blocks to let players interact with the computer by using the keyboard. Let's get started!



1. START SCRATCH

scratch.mit.edu

To start using Scratch, open your web browser, type **scratch.mit.edu** in the address bar, and then press **Enter** on the keyboard. Click **Create** on the main page to get started.

2. CLICK EVENTS

Motion

Looks

Sound

Events

Find the Code tab near the top-left corner of the screen.

Click the **Events** button to show the Events code blocks.

3. START CODING

Code

Backdrops

Sounds

Motion

Looks

Sound

Events

Control

Sensing

Events

when clicked

when space key pressed

when stage clicked

when backdrop switches to backdrop1

when space key pressed

Find the **when space key pressed** code block.

Press the mouse button and drag the block into the Coding Area.

4. SET THE KEY



You want to make some code run when the right arrow key is pressed.

Click the drop-down menu and select **right arrow**.

In games, we switch between slightly different pictures to make it look like sprites are walking. In movies, these images are called frames.



costume1

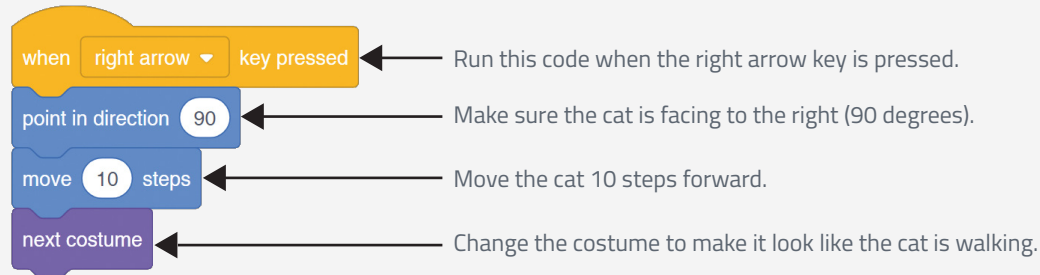


costume2

In Scratch, they're called costumes.

5. GET MOVING!

Drag the following code blocks into the Coding Area. Use their colors to help you find the blocks.



Run this code when the right arrow key is pressed.

Make sure the cat is facing to the right (90 degrees).

Move the cat 10 steps forward.

Change the costume to make it look like the cat is walking.

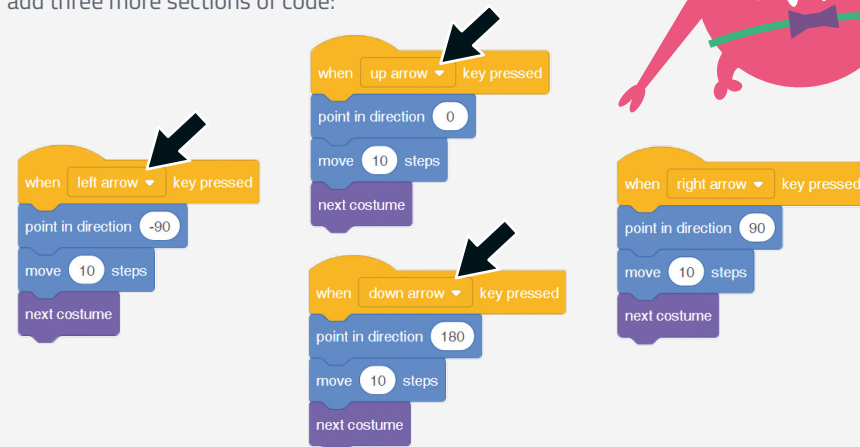


Click the **green flag** at the top of the screen to test your code. Press the **right arrow** key and watch the cat walk!

Choose the key that will make each section of code run. (See step 4.)

5. ALL DIRECTIONS

To let the cat move in all directions, add three more sections of code:



Click the **green flag** to test your code. Use the arrow keys to move the cat around! For more help visit maxw.com.

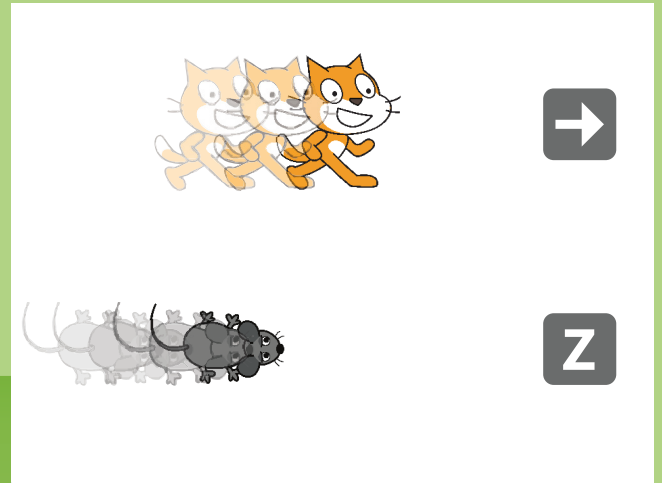
Be sure to change the direction for each code section. Enter the degrees, or use the white arrow to select the direction to move in.



Now you know how to make a simple interactive game that moves a sprite in different directions. Try using what you've learned to make the Dog on the Move game on page 14.

CAT AND MOUSE

Now that you've built a simple game, let's make a two-player game! You'll add a second animal sprite to race against the cat. By adding code, you can make both animals move when different keys are pressed.



1. START SCRATCH

scratch.mit.edu

Open your browser and go to scratch.mit.edu. Click **Create** to get started.



2. START CODING

Drag the following code blocks into the Coding Area. Use their colors to help you find the blocks.

when right arrow key pressed

Run this code when the right arrow key is pressed.

move 10 steps

Move 10 steps forward.

next costume

Use costumes to make it look as though the cat is really walking.



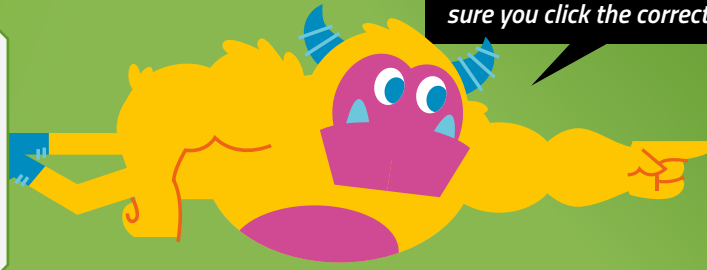
Click the **green flag** to test your code. Each time you press the **right arrow** key, the cat should take a step forward.

3. ADD ANOTHER SPRITE

Look near the bottom right of the screen to find the button. Make sure you click the correct one!

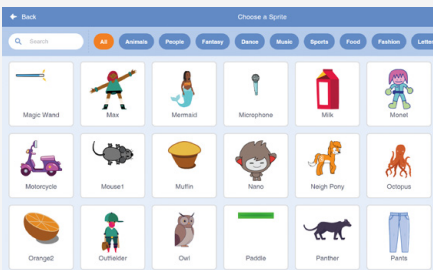


Click the **New Sprite** button.

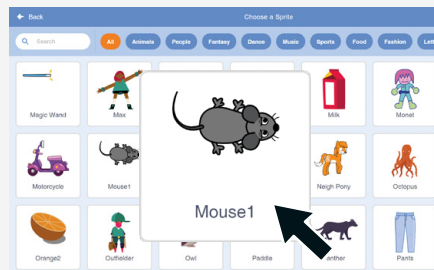


4. CHOOSE THE MOUSE IMAGE

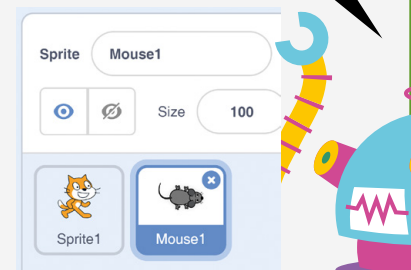
The sprite you are adding code to will have a blue line around it to show it is selected.



Scroll through the sprites until you find a picture of a mouse.



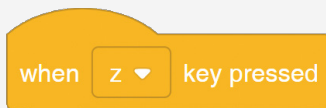
Click it to choose the image for your new sprite.



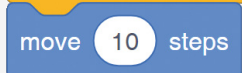
Your new sprite will show up in the Sprite List.

5. ADD CODE

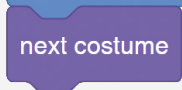
Add this code to make the mouse move when the Z key is pressed:



Run this code when the Z key is pressed.



Move the mouse sprite 10 steps forward.



Switch costumes to make it look as though the mouse is walking.

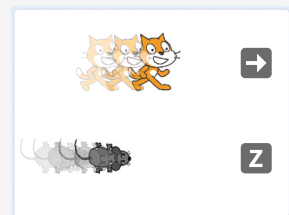


Click the **green flag** to test your code. Press the **Z** key to make the mouse move.

6. PLAY!

Find a friend to race against. Drag your sprites to the left side of the Stage before you start. Decide who will be the cat and the mouse. Get your fingers ready on the right arrow or Z key, and then go!

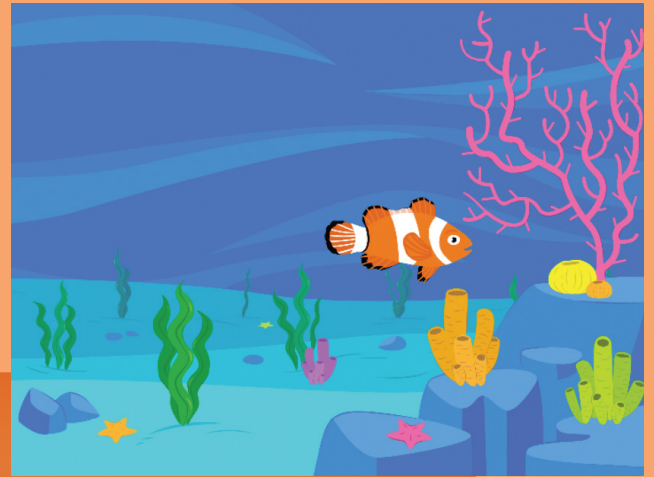
Press your key as quickly as you can to see who wins!



Here, you learned how to make a basic two-player game. When you're ready for a challenge, turn to page 14 and try making the Animal Olympics game.

FISH IN THE SEA

In this game, you'll learn a different way of controlling a sprite as it moves around. To keep it moving, you'll use a *loop*, which makes code in your game repeat over and over again. Pressing keys will be the input to steer the fish. By adding a background, you'll make the game look even better!



1. START SCRATCH

scratch.mit.edu



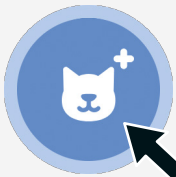
Open your browser and go to scratch.mit.edu. Click **Create** to get started.

2. NO CATS



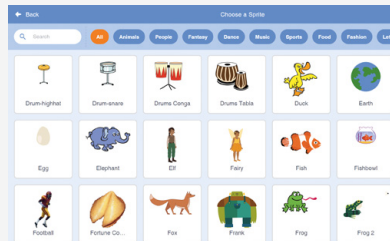
Cats aren't great swimmers. Click the blue X in the corner of the sprite icon to delete it.

3. ADD A SPRITE



Click the **New Sprite** button.

4. CHOOSE THE FISH



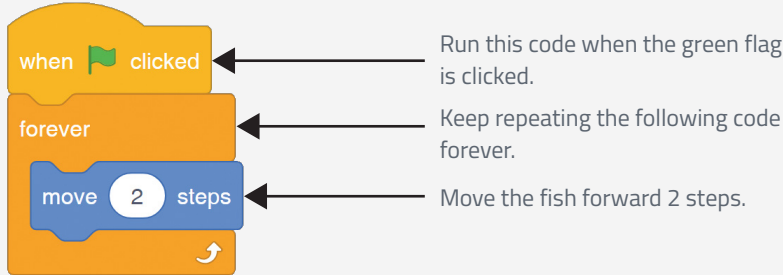
Scroll through the sprites until you find a picture of a fish.



Click it to choose the image for your new sprite.

5. ADD CODE

Drag the following code into the Coding Area:

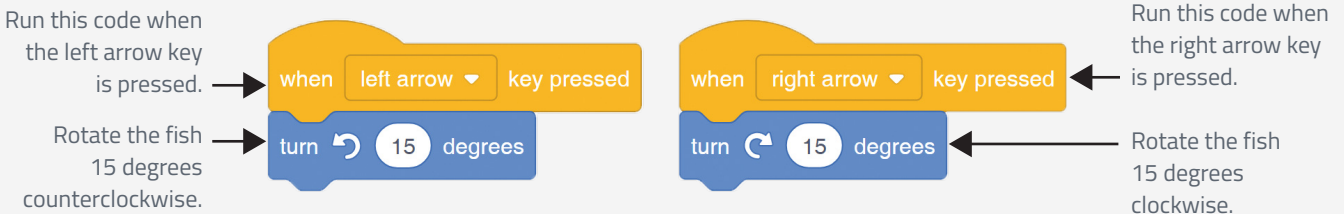


Normally, code blocks run only once, or when an event happens, like a key being pressed. A forever code block repeats all the code inside it over and over ... forever!

Click the **green flag** to test your code. The fish should start moving forward slowly. It keeps moving until it reaches the edge of the Stage because the move code block is inside the forever block.

6. CHANGE DIRECTION

You need to be able to steer the direction of the fish by pressing keys. Drag in two more sections of code, shown here:

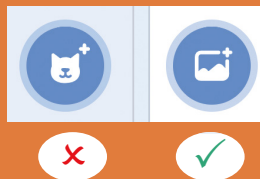


Drag the fish to the left-hand side of the Stage, and then click the **green flag** to test your code. Use the **left** and **right arrow** keys to steer the fish around.

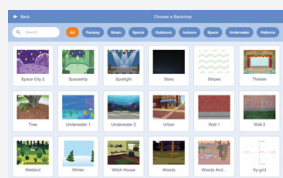
7. ADD A BACKDROP



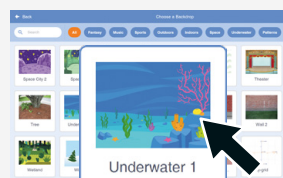
Look near the bottom right of the screen to find the button. Make sure you click the correct one!



8. CHOOSE UNDERWATER



Scroll through the backdrops until you find **underwater1**.



Click it to add the new backdrop for your game.

In this game, you learned how to use a forever loop. A loop makes code in your game repeat over and over again. When you're ready for a challenge, turn to page 15 and try the Bird in the Sky game.

BAT IN A FLAP

So far, our games have explored how sprites can move around. But most games have a more specific goal, like finding objects to collect. In Bat in a Flap, you'll learn how to code a bat that collects oranges to eat. You'll control how the bat moves by using the computer's mouse!



1. NO CATS



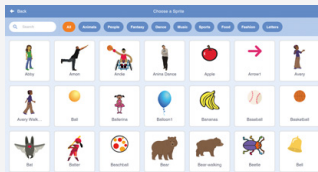
We need bats, not cats. Click the blue X in the corner of the cat sprite icon to delete it.

2. ADD A SPRITE



Click the **New Sprite** button.

3. CHOOSE THE BAT



Scroll through the sprites until you find a picture of a bat.



Click it to set the image for your new sprite.

4. ADD CODE TO THE BAT

Drag this code into the Coding Area to make the bat fly around the Stage:

when clicked — Run the following code when the green flag is clicked.

set rotation style — Stop the bat from rotating when it changes direction.

forever — Keep repeating the code inside this block forever.

point towards — Set the direction of the bat so it will move toward the mouse.

move 4 steps — Move the bat toward the mouse pointer.

next costume — Switch costumes so it looks as if the bat's wings are flapping.



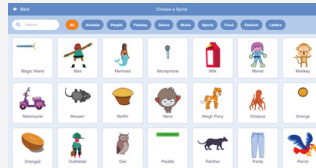
Click the **green flag** to test your code. The bat should fly toward the mouse pointer. Move your mouse to practice making the bat fly around the Stage.

5. ADD A SPRITE

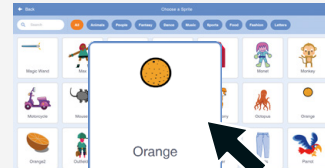


Click the **New Sprite** button.

6. CHOOSE THE ORANGE



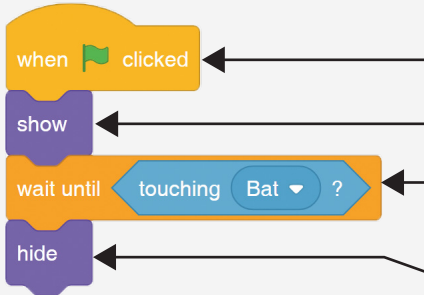
Scroll until you find a picture of an orange.



Click it to select the image.

7. ADD CODE TO THE ORANGE

Drag this code into the Coding Area to control the orange:

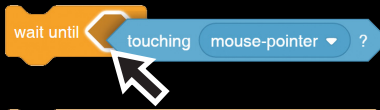


Run this code when the green flag is clicked.
Make sure the orange is visible.
Wait until the bat touches (eats) the orange.
Hide the orange—it's been eaten!

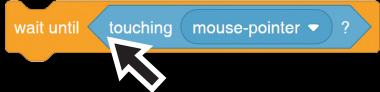
HOW TO COMBINE CODE BLOCKS



Start with a wait until code block.

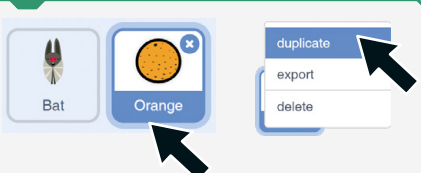


Drag a touching mouse pointer block into the wait until block. (Drag it from the left end.)



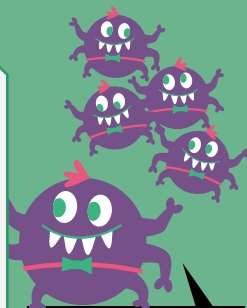
From the drop-down menu, choose Bat.

8. MORE ORANGES



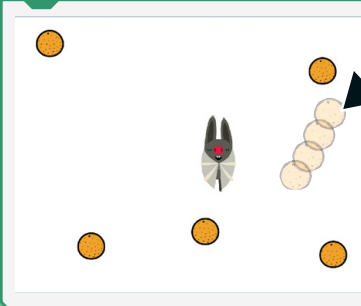
Right-click the **Orange** sprite in the Sprite List.

Click **duplicate**.



Repeat step 8 until you have five oranges!

9. ORANGE SPREAD



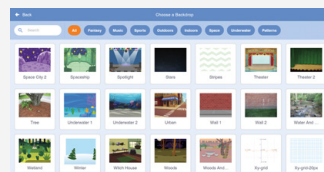
Spread the oranges out around the Stage.

(If you can't see the new oranges, click the green flag first, followed by the Stop button.)

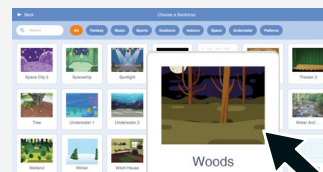
10. SET THE BACKDROP



Click the **New Backdrop** button.



Scroll until you find **Woods**.



Click it to set the backdrop.

Test your code. Use the mouse pointer to guide the bat around the Stage until the bat eats up all the oranges!

In this game, you used the hide and show blocks to make it look as if a sprite got eaten. You used the wait until and touching code blocks to tell when one sprite has touched another. Finally, you learned how to duplicate a sprite to create multiple targets to collect. When you're ready for a challenge, turn to page 15 and try creating the Dinosaur Diner game!

KIRAN AND THE MAGIC CRYSTALS

This is another collecting game. Our brave astronaut Kiran has to travel around space, collecting the magic crystals. You'll use what you learned from the previous game and duplicate multiple sprites. You'll also use a new kind of loop and a special effect to make the magic crystals glow!

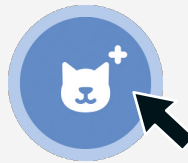


1. NO CATS IN SPACE!



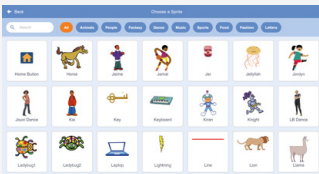
Space is bad for cats. Click the blue **X** to delete the cat.

2. ADD A SPRITE

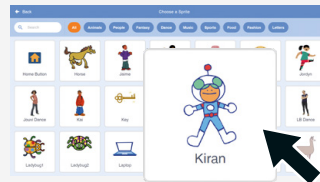


Click the **New Sprite** button.

3. CHOOSE KIRAN



Scroll through the sprites until you find Kiran.



Click **Kiran** to set the image for your new sprite.

4. ADD CODE TO KIRAN

Make Kiran fly by dragging this code into the Coding Area:

when clicked ← Run the following code when the green flag is clicked.

set size to 25 % ← Shrink Kiran to one-quarter of her normal size.

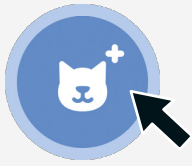
forever ← Keep repeating the code inside this block forever.

point towards mouse-pointer ← Point Kiran toward the mouse pointer.

move 3 steps ← Move Kiran toward the mouse pointer.

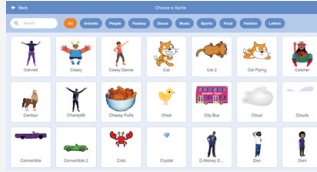
Click the **green flag** to test your code. The code should make Kiran fly toward the mouse pointer. Practice making her fly around the Stage by moving your mouse.

5. ADD A SPRITE

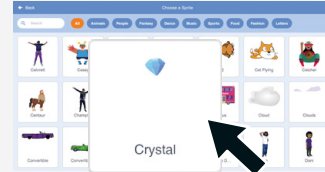


Click the **New Sprite** button.

6. FIND THE CRYSTAL



Scroll until you find the **Crystal** sprite.



Click it to select the image.

7. CODE THE CRYSTAL

Drag this code into the Coding Area to control the crystal:

when **clicked** ← Run this code when the green flag is clicked.

show ← Make sure the crystal is visible.

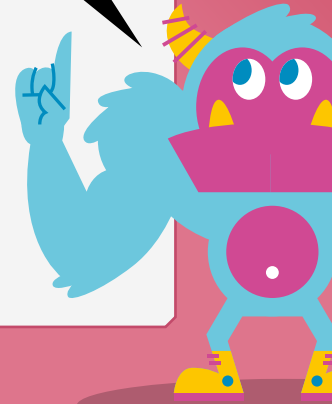
repeat until **touching Kiran** ? ← Repeat the next two code blocks until Kiran collects the crystal.

turn **15** degrees ← Rotate the crystal slowly.

change **color** effect by **25** ← Make the crystal glow by slowly changing its color.

hide ← Make the crystal disappear when Kiran touches it.

See page 11 for help combining code blocks.



8. MORE CRYSTALS

Right-click the **Crystal** sprite in the Sprites List.

Click **duplicate**.

Repeat step 8 until you have enough crystals spread around.



9. WE NEED STARS!

Click the **New Backdrop** button.

Scroll until you find **Stars**.

Click it to set the backdrop.

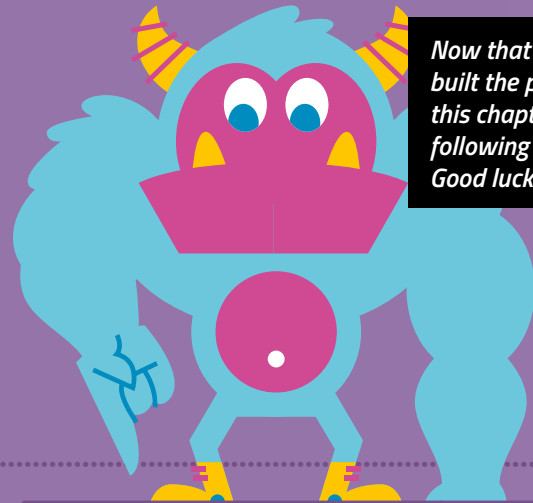
Test your code. Use the mouse pointer to guide Kiran around the Stage and collect the magic crystals!

In this game, you made the crystals disappear when Kiran touched them. You used a repeat until touching loop to keep running some code until each crystal was found. This code created an animation makes the game look more advanced and fun to play. Turn to page 15 and try creating the Ripley and the Aliens game when you're ready for a challenge!

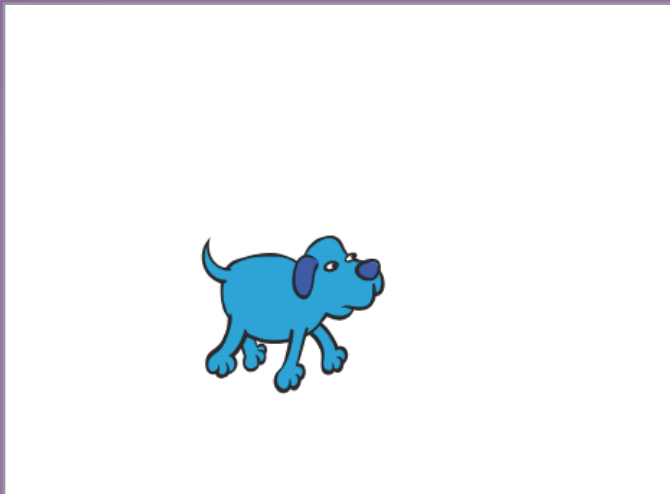
CHAPTER 1

CHALLENGES

In this chapter, you learned the basics of making a Scratch game. You used loops to repeat code, and input blocks to respond to the keyboard and mouse. Move and rotate blocks allowed you to move sprites around. You also used hide, show, next costume, and change color effect blocks to alter how sprites look.

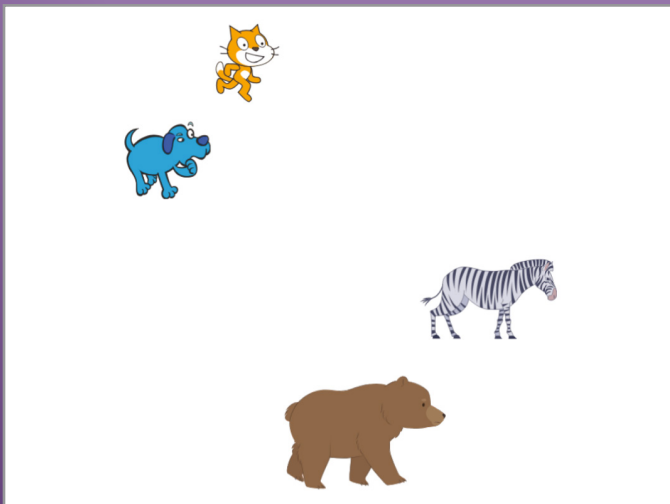


Now that you've built the programs in this chapter, try the following challenges. Good luck!



CHALLENGE 1 DOG ON THE MOVE

Look at how the code works in the Cat on the Move game. Use the same idea to make a program that moves a dog around the Stage. Start by deleting the Cat sprite and adding a dog. Using the code from page 4 to help, make the dog move by pressing the arrow keys.



CHALLENGE 2 ANIMAL OLYMPICS

Cat and Mouse showed you how to make a game with two moving sprites. How about making a four-player game using the same idea? Look through the animal sprites and choose four to add. Pick different keys you can press to make each one move. Use the code on page 6 to make each animal move when a different key is pressed.



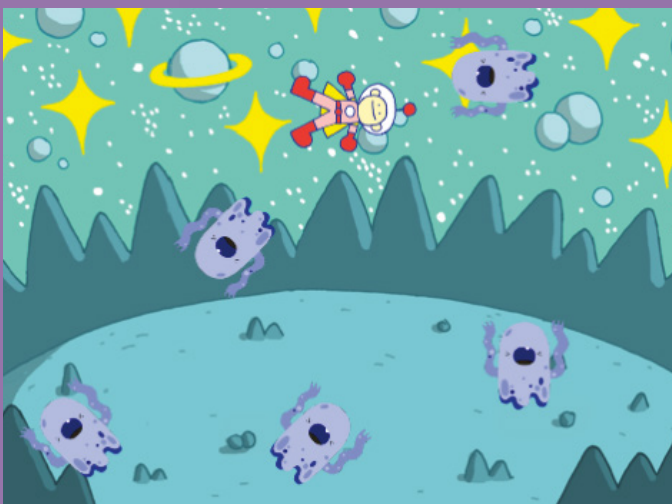
CHALLENGE 3 BIRD IN THE SKY

Start a new project in Scratch and choose a backdrop. Delete the Cat sprite and add the Parrot. Make it fly around when you press the left and right arrow keys by using what you learned in Fish in the Sea on page 8. Make the Parrot flap its wings by using the next costume code block.



CHALLENGE 4 DINOSAUR DINER

In Bat in a Flap, you built a more complex game with several sprites. Start a new project with a different background. Instead of a bat, add a dinosaur sprite. Use similar code to the bat sprite to move the dinosaur around. What will the dinosaur eat? Eggs? Apples? Other dinosaurs? Look at how the oranges were coded on page 10 and use those ideas to make the food disappear when the dinosaur eats it.



CHALLENGE 5 RIPLEY AND THE ALIENS

Remember how the Kiran and the Magic Crystals game worked? Build a new game where the Ripley sprite is flying around the Stage. Add some aliens instead of crystals. Use the ideas from the crystal code to make the aliens spin and change color. Experiment to make new effects. When Ripley reaches an alien, it must disappear. Use code similar to that on page 12 to make this game work.

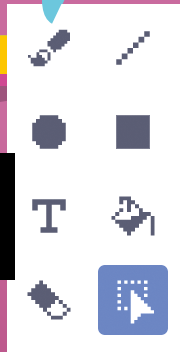
DRAWING BACKGROUNDS AND SPRITES

In this chapter, you'll learn how to create your own backdrops and draw your own sprites. You'll use code to detect when one sprite hits another one, and you'll use coordinates to position sprites and move them around.

To pick the colors you want, use the new Scratch color-mixing tool.



You'll be using these drawing tools in this chapter.

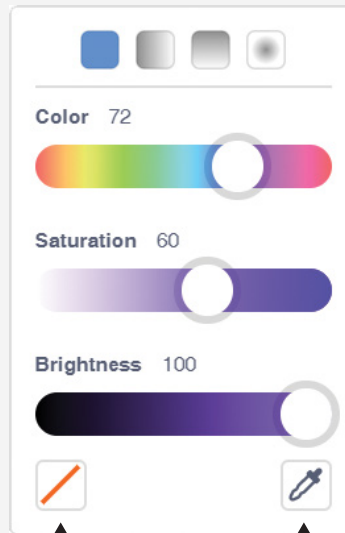


PICK A COLOR

In Scratch, you need to mix your own colors by adjusting some sliders. You can do this using the color palette in the Paint Editor.



The color you create will be shown in this box.



Click the drop-down to see the sliders:

Move the Color slider to select one of the colors shown here.

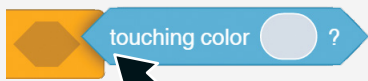
Adjust the Saturation slider to make the colors richer and more intense.

Move the Brightness slider down to 0 to make very dark colors (like black or dark gray).

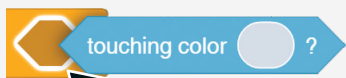
Draw as transparent.

Use the eyedropper to pick a color that's already on the Stage.

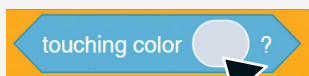
SET TOUCHING COLOR BLOCKS



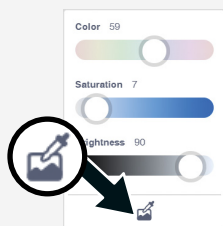
Drag a touching color block over a repeat until or if block.



Drop it when a white line appears.



Click the color.



Choose the eyedropper from the pop-up that appears.



Move the mouse around the Stage to find the color you want to check for—then click the mouse.



Now you've set the color to test for!

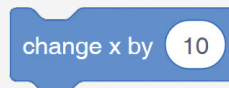
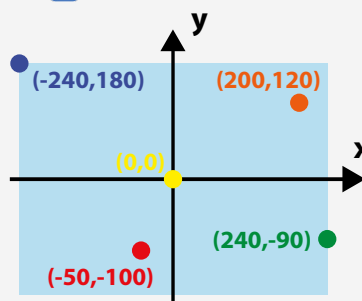
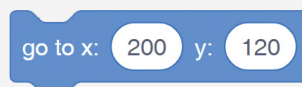
Here's how to check the color a sprite is touching on the Stage.



Use the *change x* and *change y* blocks to move a sprite.

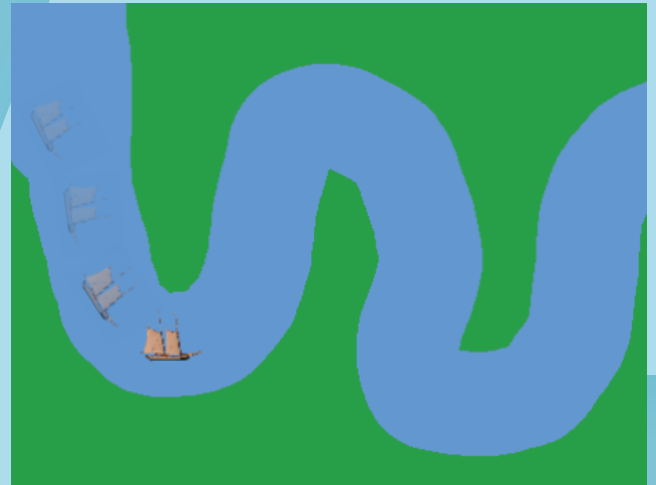
SET X- AND Y-COORDINATES

The go to block sends a sprite to specific coordinates on the Stage.

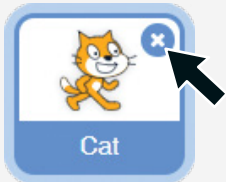


DOWN THE RIVER

In this game, you'll create your own backdrop. You'll draw a river for a boat to sail down, and you'll need to make sure it stays in the river. To do this, you'll use code to detect any colors that the boat sprite touches. You'll use keyboard inputs to steer the boat.

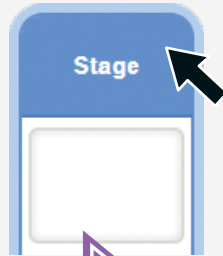


1. NO CATS



Click the blue **X** in the corner of the sprite to delete it.

2. SELECT THE BACKDROP



In the sprites pane, click the **Stage** icon. (It's at the bottom right of the screen.)

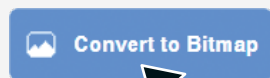


Click the **Backdrops** tab. (It's at the top left of the screen.)



Turn to page 16 for help with setting colors.

3. START DRAWING



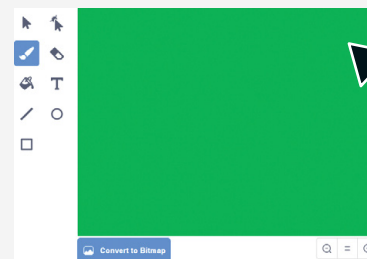
Click the **Convert to Bitmap** button. (This will make it easier to start drawing.)



Choose the **Fill** tool.



Click the **Fill** drop-down and set it to green.



Click the Drawing Area to fill it with green.

4. SET THE BRUSH STYLE



Choose the **Brush** tool.



Choose a blue color.




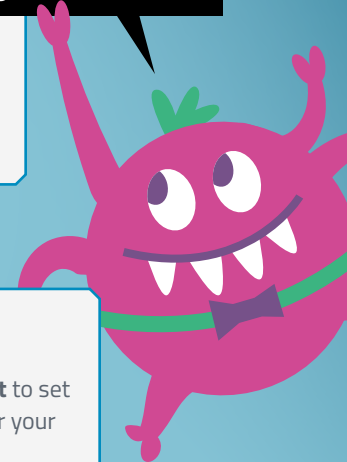
Make the brush very thick by typing in **100** above the Drawing Area.

5. DRAW THE RIVER

Use the Brush tool to draw a river. Make sure the river is wide enough for the sailboat you'll be adding. (If the boat gets stuck too easily, come back and make the river wider by drawing more water.)



 If you make a mistake while drawing, click the Undo button and try again!



6. ADD A SPRITE



Click the **Choose a Sprite** button.

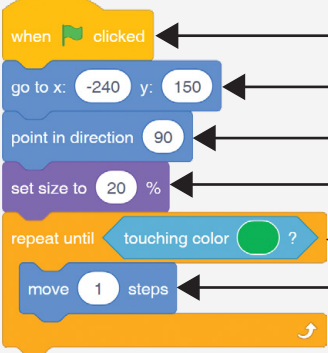
7. FIND A BOAT



Click **Sailboat** to set the image for your new sprite.

8. ADD CODE TO MAKE THE BOAT MOVE

Make the Sailboat move by dragging this code into the Coding Area:



Run the following code when the green flag is clicked.


Start the sailboat in the top-left corner of the screen.

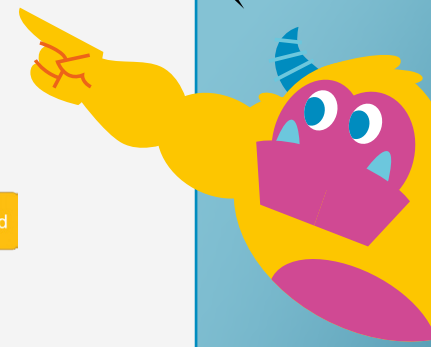
Make the sailboat point to the right.

Shrink the sailboat down to 20 percent of its normal size.

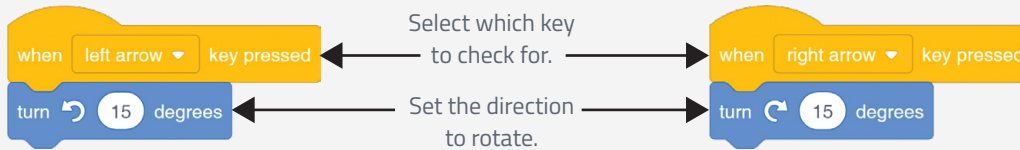
Keep repeating the next code block until the boat hits some land.

Slowly move the boat forward.

 Use the eyedropper to select the green color. See page 17 for help.



Now drag in two more scripts to steer the boat when the left or right arrow key is pressed:



Select which key to check for.

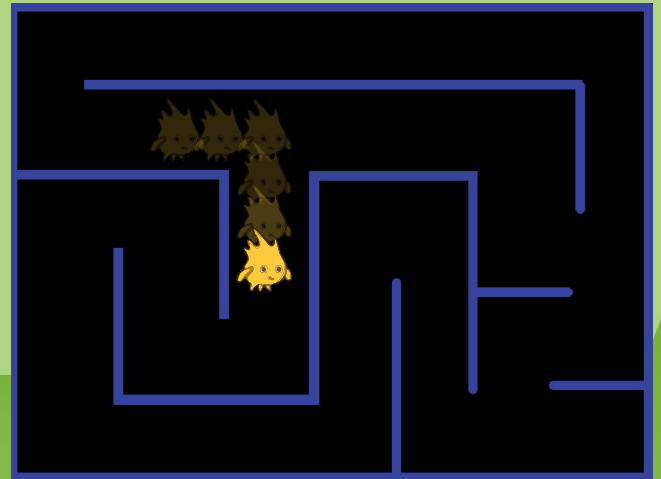
Set the direction to rotate.

Now test your code!

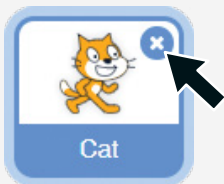
This game showed you how to draw your own backdrops. You used the touching color block to check whether a sprite was bumping into part of your background picture. You'll be using this technique in lots of ways throughout this book. Ready for a challenge? Turn to page 32 and try to build the Down the Road game.

MONSTER MAZE

This game will give you a chance to practice your drawing skills. You'll use code to detect whether your sprite has touched part of the maze. You'll use a different technique to move the sprite. A loop will keep the sprite moving all the time, and pressing the arrow keys will change its direction.

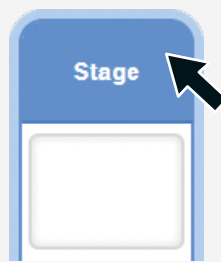


1. NO CATS



Click the blue X in the corner of the sprite to delete it.

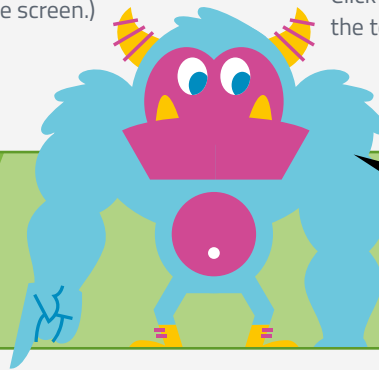
2. SELECT THE BACKDROP



In the sprites pane, click the **Stage** icon. (It's at the bottom right of the screen.)

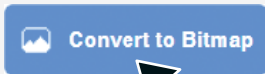


Click the **Backdrops** tab at the top left of the screen.



Turn to page 16 for help with setting colors.

3. START DRAWING



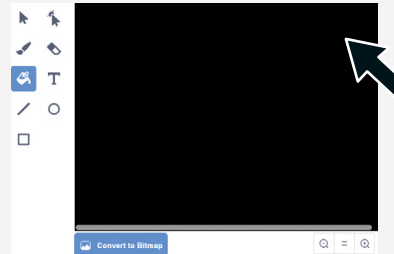
Click the **Convert to Bitmap** button. (This will make it simpler to start drawing.)



Choose the **Fill** tool.



Click the **Fill** drop-down, and set it to black.



Click the Drawing Area to fill it with black.

4. SET THE LINE STYLE



Choose the **Line** tool.



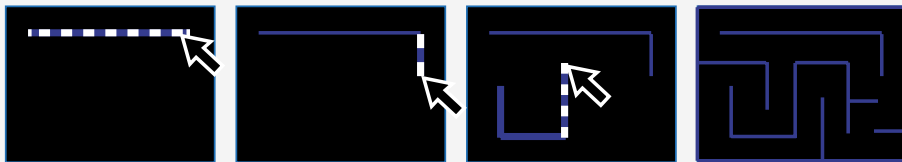
Choose a blue color.



20

Make the line thick by typing in **20** above the Drawing Area.

5. ADD SOME WALLS



Use the Line tool to gradually build up the maze.

6. ADD A SPRITE



Click the **Choose a Sprite** button.

7. FIND GOBO



Click **Gobo** to set the image for your new sprite.

8. ADD CODE TO MAKE THE MONSTER MOVE

Make Gobo the monster move by dragging this code into the Coding Area:

when green flag clicked → Run the following code when the green flag is clicked.

set rotation style to left-right → Set the way the monster sprite rotates.

set size to 40% → Shrink the sprite down to 40 percent of its size so it fits in the maze.

forever loop → Keep repeating the following code forever.

move 4 steps → Move the sprite 4 steps forward.

if touching color ? then → If the sprite hits a wall, then run this code.

move -5 steps → Move the sprite 5 steps backward.

Use the eyedropper to select the wall color. See page 17 for help.

Now drag in four more scripts to make the sprite change direction when one of the arrow keys is pressed:

Select which key to check for.

Set the direction to move.



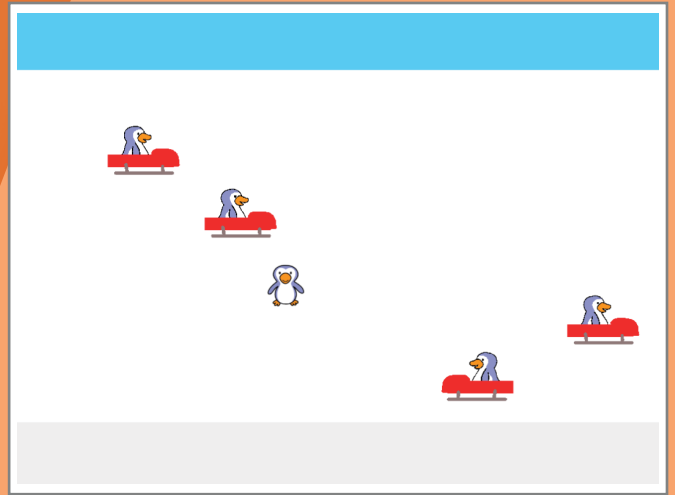
Now test your code!



In this game you created your own maze backdrop. You used the same technique as in Down the River to check for collisions—the color touching block. But the code to move the sprite used some new ideas, making it go back when it hit a wall. Ready for a challenge? Turn to page 32 and try to build the Magic Maze game.

PENGUIN BOBSLED CROSS

In this winter variation of a road-crossing game, the penguin has to cross the ice while dodging bobsleds in order to reach the sea on the other side! The four duplicated bobsled sprites travel back and forth across the Stage, checking to see if they bump into the penguin. You'll use color testing to check when the penguin has reached the sea.



Turn to page 16 for help with setting colors.

1. NO CATS

Click the blue X to delete the cat.

2. SELECT THE BACKDROP

Click the **Backdrops** tab at the top left of the screen.

3. START DRAWING

Click the **Convert to Bitmap** button.

Choose the **Rectangle** tool.

Click and drag to draw a thin rectangle at the bottom.

Add a blue rectangle at the top to be the sea.

4. ADD A SPRITE

Click the **Choose a Sprite** button.

Scroll through the sprites and click **Penguin 2**.

5. CODE THE PENGUIN

Click the **Code** tab and then drag this code into the Coding Area:

- when green flag clicked → When the green flag is clicked, run the following code.
- set size to 30% → Set penguin size to 30 percent.
- set rotation style left-right → Stop the penguin from fully rotating.
- go to x: 0 y: -160 → Start the penguin in the middle of the bottom part of the Stage.
- repeat until touching color → **Repeat this code until the penguin reaches the sea.**
- move 2 steps → Move the penguin 2 steps forward.
- move 20 steps → Move the penguin 20 steps into the sea.
- say Wahoo! for 2 seconds → The penguin made it! Show a message.

6. SET A CHANGE OF DIRECTION

Now add more code to make the penguin change direction when one of the arrow keys is pressed:

when left arrow key pressed
point in direction -90

when up arrow key pressed
point in direction 0

when down arrow key pressed
point in direction 180

when right arrow key pressed
point in direction 90

Select which key to check for.

Set the direction to move.

Test your code!

7. ADD A BOBSLED

Repeat step 4 to add another penguin.

Click the **penguin2-c** costume showing the penguin from the side. Find this in the Costume List under the Costumes tab.

Click the **Convert to Bitmap** button.

Choose the **Brush** tool.

Make the brush thick by typing in **30**.

Draw a simple bobsled shape. Use the **Line** tool to add some skis.

8. CODE THE BOBSLED

Click the **Code** tab and then drag this code into the Coding Area:

when green flag clicked

set rotation style left-right

repeat until touching Penguin 2

move 5 steps

if on edge, bounce

say Game Over! for 2 seconds

stop all

When the green flag is clicked, run the following code.

Stop the bobsled from fully rotating.

Keep repeating the following code until it hits the penguin.

Move the bobsled 5 steps forward.

If the bobsled reaches the edge of the screen, make it change direction.

The penguin has been hit! Show a message.

Stop the penguin code and the other bobsleds.

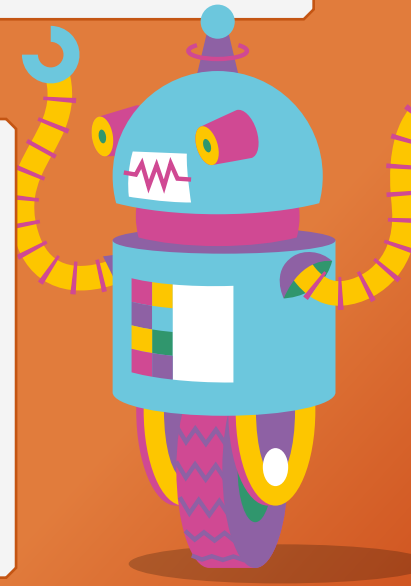
9. ADD MORE BOBSLEDS

In the Sprite List, right-click the **bobsled** icon.

Click **duplicate** to make a copy. The copy appears in the Sprite List and above it on the Stage.

Duplicate more bobsleds and place them where you want on the ice.

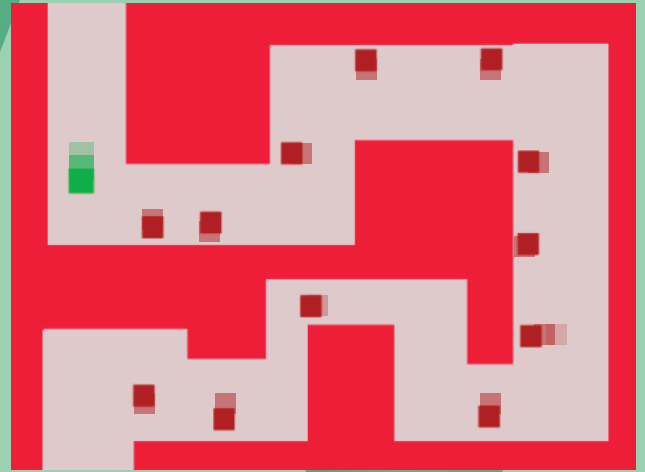
Click the **green flag** to test your code.



Ready for a challenge? Turn to page 33 and try creating your own road-crossing game.

COLOR BOUNCE

In Color Bounce, you'll build on the idea of a maze to make a challenging and exciting game. You'll create small squares that slowly bounce off the walls of a maze, creating extra obstacles to get past. Code will check whether each square hits the maze wall. If it does, it changes direction by 180 degrees, bouncing back the other way. A green player sprite has to get through the maze without hitting the squares!



1. NO CATS



Delete the cat sprite.

2. SELECT THE BACKDROP



In the sprites pane, click the **Stage** icon.

Code

Backdrops

Click the **Backdrops** tab in the top left of the screen.

Turn to page 16 for help with setting colors.

3. START DRAWING

Convert to Bitmap

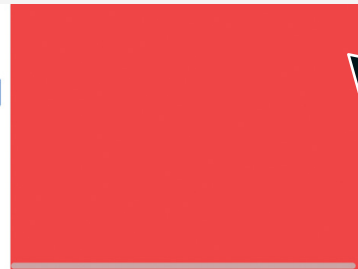
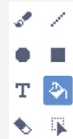
Click the **Convert to Bitmap** button.



Choose the **Fill** tool.

Fill

Choose a pinkish red.



Color the **Drawing Area**.



If you make a mistake, click the Undo button and try again.



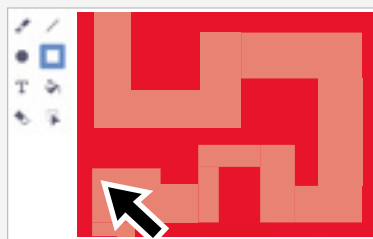
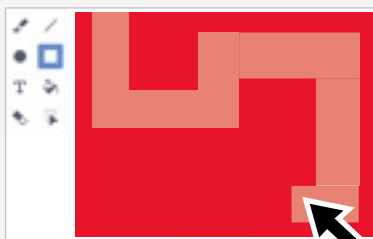
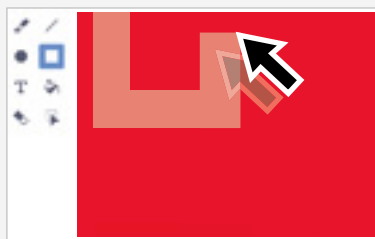
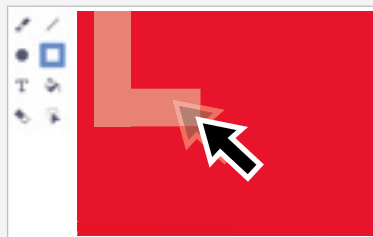
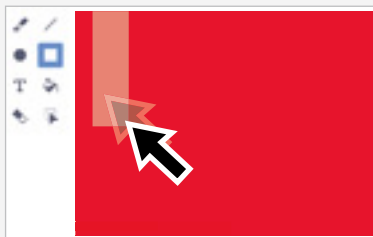
4. DRAW A PATH



Select the **Rectangle** tool.



Choose a very light gray.



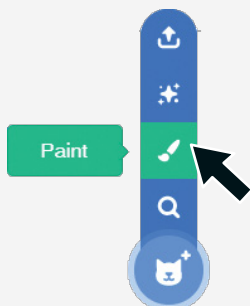
Use the mouse to draw a path across the screen.

5. ADD A MOVING SQUARE SPRITE

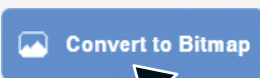


Let your mouse hover over the **Choose a Sprite** button.

Select the **Paint** option (the brush icon).



6. DRAW A SQUARE



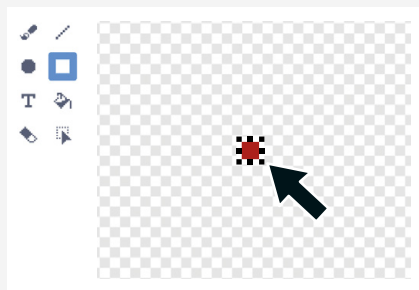
Click the **Convert to Bitmap** button.



Choose the **Rectangle** tool.



Choose a dark red color.



Draw a very small square in the center of the Drawing Area.

7. ADD CODE TO MAKE THE SQUARES MOVE



Click the **Code** tab and then drag the following code into the Coding Area:

```

when green flag clicked
  set rotation style to don't rotate
  point in direction 180
  forever loop
    move 1 steps
    if touching color red then
      turn 180 degrees
  
```

Run the following code when the green flag is clicked.

Stop the square from rotating.

Point the square downward.

Keep repeating the following code forever.

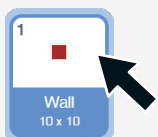
Move the square forward slowly.

If the square hits the maze wall, then run this.

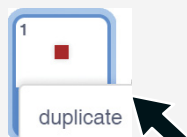
Rotate the square around so it moves the other way.

Drag the square into place. Click the **green flag** to test your code. The square should bounce up and down.

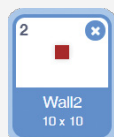
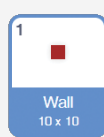
8. DUPLICATE SQUARES



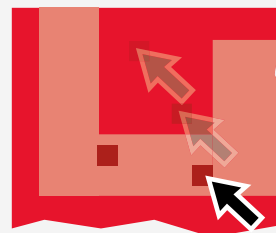
In the Sprite List, right-click the **square** icon.



Click **duplicate** to make a copy.

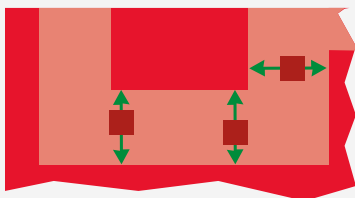


The sprite will appear in the Sprite List and above it on the Stage.



On the Stage, drag the sprite onto the path.

9. SET THE CODE TO CHANGE DIRECTION



Some of the squares need to move up and down, while others need to move left and right. To make them move left and right, change the direction the square starts moving in.

```

when green flag clicked
  set rotation style to don't rotate
  point in direction 180
  forever loop
    move 1 steps
    if touching color red then
      turn 180 degrees
  
```

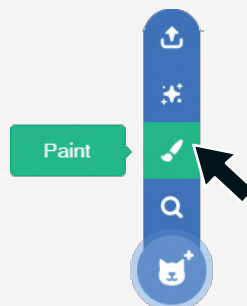
Change the direction to **90**.

10. ADD THE PLAYER SPRITE



Let your mouse hover over the **Choose a Sprite** button.

Select the **Paint** option (the brush icon).



11. DRAW THE PLAYER SPRITE



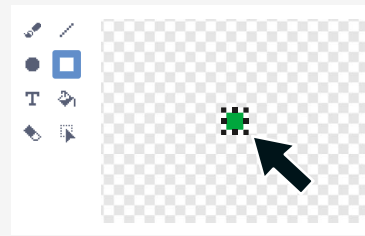
Click the **Convert to Bitmap** button.



Choose the **Rectangle** tool.



Choose a green color.



Draw the player sprite in the center. It should be roughly the same size as the red square obstacle sprites.

12. ADD CODE TO CONTROL HOW THE PLAYER MOVES

Code

Click the **Code** tab and then drag the following code into the Coding Area:

when green flag clicked

go to x: -180 y: 170

set rotation style to don't rotate

repeat until touching color red or touching color red

if key right arrow pressed? then

change x by 2

if key left arrow pressed? then

change x by -2

if key up arrow pressed? then

change y by 2

if key down arrow pressed? then

change y by -2

stop all

Click the **green flag** to play the game.

Run this code when the green flag is clicked.

Start the player in the top-left corner.

Stop the player from rotating.

Repeat the following code until the player hits a moving square or the side of the maze.

If the **right arrow** key is pressed, then run this code.

Move the player *right* by increasing its x-coordinate.

If the **left arrow** key is pressed, then run this code.

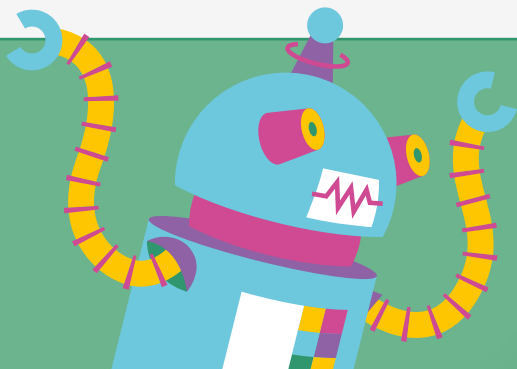
Move the player *left* by decreasing its x-coordinate.

If the **up arrow** key is pressed, then run this code.

Move the player *up* by increasing its y-coordinate.

If the **down arrow** key is pressed, then run this code.

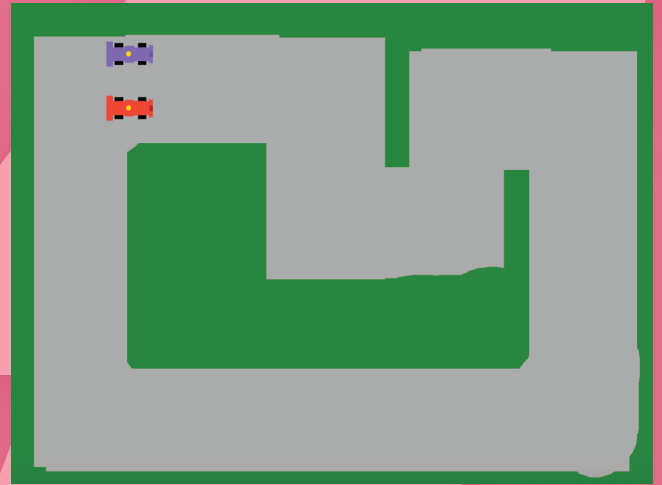
Move the player *down* by decreasing its y-coordinate.



Ready for a challenge? Turn to page 33 and try to make Color Spinners.

RACETRACK

Two-player games are great fun to play and can be quite simple to code. For this game, you'll design and code one car, and then duplicate it to create the second player's car. You'll use color sensing to keep the cars on track. After duplicating the first car, you'll modify the code controlling the second one to respond to different key inputs.



1. NO CATS

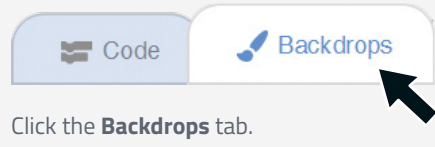


Delete the cat sprite.

2. SELECT THE BACKDROP

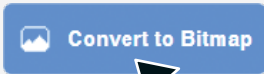


In the sprites pane, click the Stage icon.



Click the **Backdrops** tab.

3. DRAW THE BACKGROUND



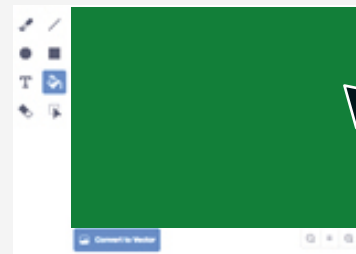
Click the **Convert to Bitmap** button.



Choose the **Fill** tool.



Click the **Fill** drop-down and choose a green color.



Color the Drawing Area.

Turn to page 16 for help with setting colors.



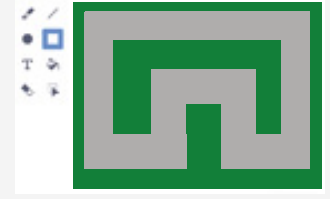
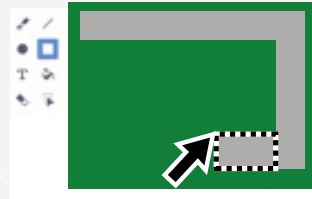
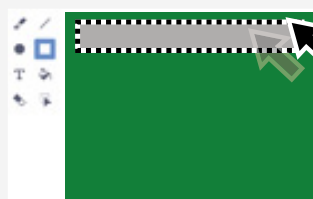
4. DRAW THE RACETRACK



Select the **Rectangle** tool.



Choose a gray color.



Draw a series of rectangles to make a track for your game. You can always make the track thicker or thinner later.

5. ADD A CAR SPRITE



Let your mouse hover over the **Choose a Sprite** button.

Paint



Find the Paint option. Click **Paint**.



If you make a mistake, click the Undo button and try again!

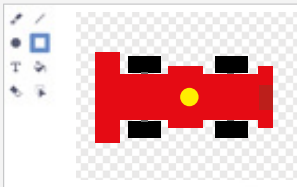
6. DRAW THE CAR

Convert to Bitmap

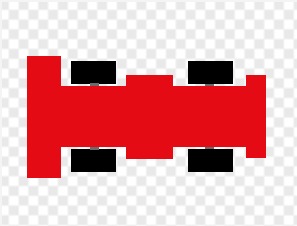
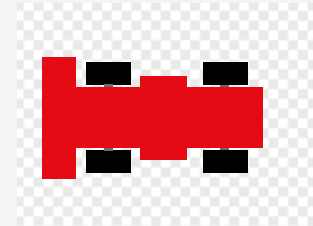
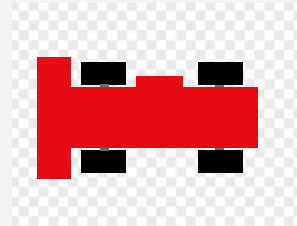
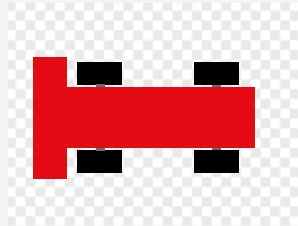
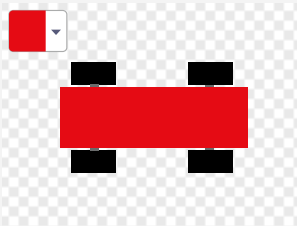
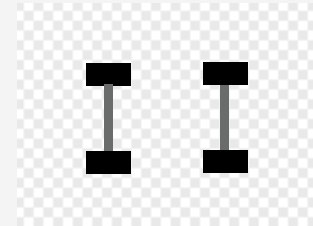
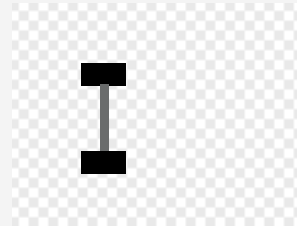
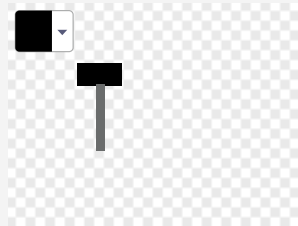
Click the **Convert to Bitmap** button.



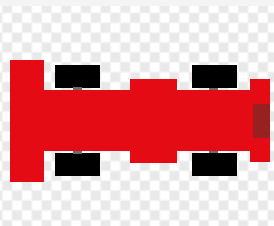
Choose the **Rectangle** tool.



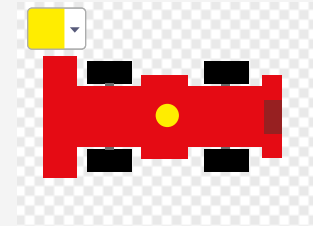
To create your racing car, you'll first draw a large car. This will make it easier to add details. You can shrink it later with code. Make sure you draw it so it nearly fills the Drawing Area.



Use a much darker red to add a box at the front. This will help you check for collisions.



Use the **Circle** tool to add a yellow driver's helmet to the car.



7. CODE THE CAR

Code

Click the **Code** tab at the top left of the screen, and then drag this code into the Coding Area:

when green flag clicked

set size to 10 %

go to x: -150 y: -140

point in direction 90

forever

if key left arrow pressed? then

turn 5 degrees

if key right arrow pressed? then

turn 5 degrees

When the green flag is clicked, run the following code.

Shrink the car down to 10 percent of its size.

Move it to the top-left corner of the screen.

Point the car to the right.

Keep repeating the following code forever.

If the *left arrow* key is pressed, then run this code.

Rotate the car counterclockwise.

If the *right arrow* key is pressed, then run this code.

Rotate the car clockwise.

This first part of the code will allow you to steer the car around corners. You also need some code to make it move forward.

when up arrow key pressed

repeat until not color red is touching ?

move 3 steps

move -3 steps

When the up arrow key is pressed, run this code.

Repeat this code until the car leaves the track.

Move the car forward 3 steps.

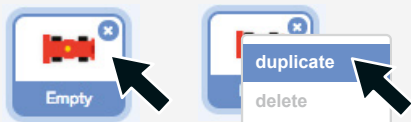
The car has hit something, so move it back 3 steps.



Click the **green flag** to test your code. Press the **up arrow** to start the car moving. Use the **left** and **right arrows** to steer it around the track. If the car hits anything green, it should stop, until you press the up arrow again.



8. ADD ANOTHER CAR



In the Sprite List, right-click the **car** icon.

Click **duplicate**.

9. CHANGE THE KEYS

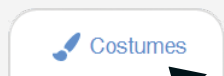
You'll need different keys to control the new car.

Change: key left arrow pressed? to: key z pressed?

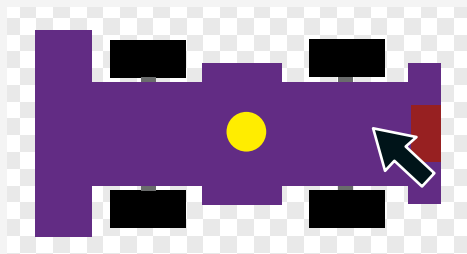
Change: key right arrow pressed? to: key x pressed?

Change: when up arrow key pressed to: when s key pressed

10. SET A NEW COLOR



Click the **Costumes** tab.



Use the **Fill** tool to change the main body color of the second car. (Make sure to leave the dark red part at the front the same!)

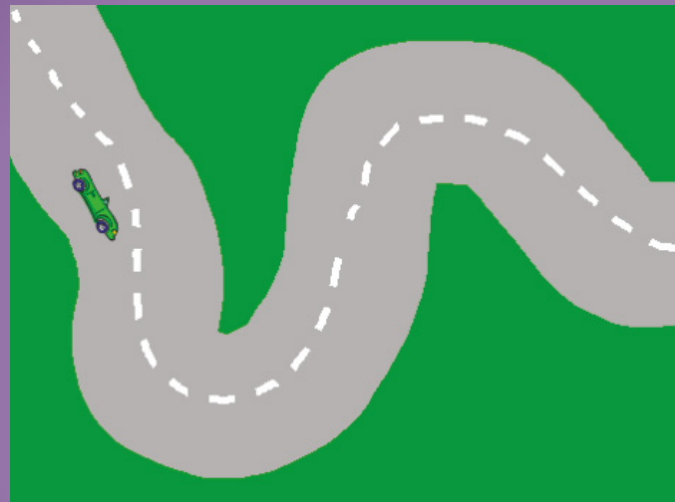
You've learned how to duplicate sprites to create a simple two-player game. When you're ready for something more challenging, turn to Moon Race on page 33.



CHAPTER 2

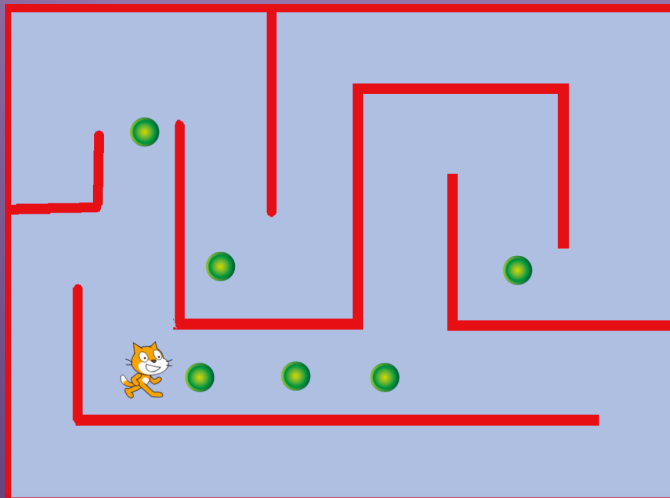
CHALLENGES

This chapter taught you how to make your games more personal by drawing your own backdrops and sprites. You used touching color blocks to detect when a sprite hits a color in the backdrop and then used that event to change what happens in your game. Using coordinates to set or change a sprite's position gives you more control over your games.



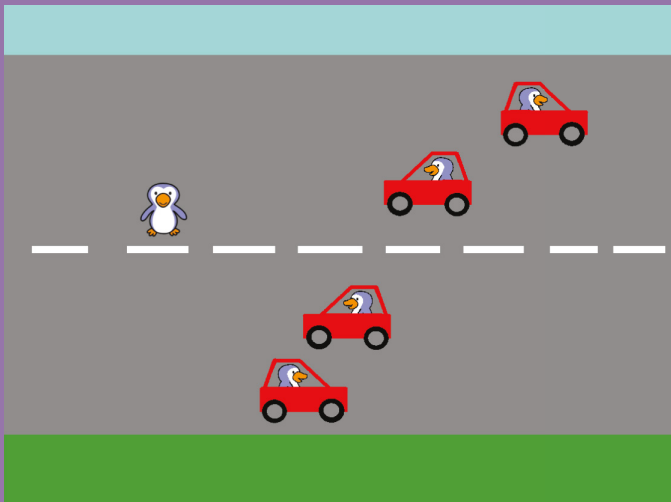
CHALLENGE 1 DOWN THE ROAD

Use the ideas you learned in Down the River to build a similar game. In this game, a car has to drive along a road. Start with a green background and draw a gray road. Add a car sprite and then code it to move by using the ideas on page 18. The car should stop when it hits green.



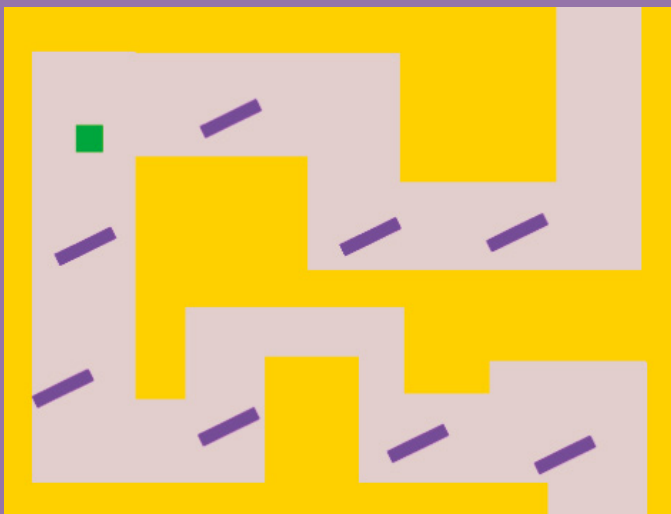
CHALLENGE 2 MAGIC MAZE

Use the Monster Maze game on page 20 to help you with this challenge. Start a new project and draw a maze of your own. Use code to move the cat around the maze—unless it hits one of the maze walls. Once the game is working, add magic jewels for the cat to collect. Use the concepts from the Kiran and the Magic Crystals game on page 12 to make the jewels disappear as the cat finds them.



CHALLENGE 3 CROSS THE ROAD

Start a new project in Scratch. Draw a road going across the Drawing Area. Add code to make the penguin move when the keys are pressed. Use the code on page 23 to help you. Add a car sprite and then add code to make it drive from left to right (the way the bobsleds moved). Once the car is working, duplicate it to add another car. Your goal is to get the penguin across the road, just like in Penguin Bobsled Cross.



CHALLENGE 4 COLOR SPINNERS

Draw a simple path as you did for the Color Bounce game. Draw a new sprite that is a small purple rectangle. Add code to make it rotate slowly. Use a forever loop with a turn 1 degree block inside it. Duplicate the sprite several times. Drag each one to a different place on the path to create moving obstacles. Add a new green square sprite as the player. Add code to move the player when the arrow keys are pressed (see page 23 for help).



CHALLENGE 5 MOON RACE

Make a racing game similar to the Race Track game on page 28. Instead of a car, draw a rocket. Add code to make it move and to set keys that will steer it. You'll need to change the code from the Race Track game so the rocket checks for black instead of gray. When the game is working, duplicate the rocket to add another one. Change the keys used to fly the second rocket.

WHAT'S THE SCORE?

To make a game more engaging, you can do several things. You can make the game more unpredictable by generating random numbers to change the way sprites move around. You can also add a time limit to give players more incentive to compete against the clock. Most importantly, you'll learn how to show the score in a game by using a variable.

RANDOM NUMBERS

set x to 0

In Chapter 2, you learned how to set the starting coordinates of a sprite.

set x to pick random 0 to 100

Using a pick random block, you can change the x value to a range of random numbers.

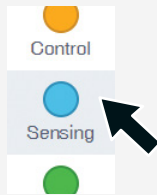


It's like rolling a giant die!

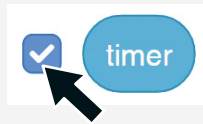
1 23 45 ^ 96

SHOWING THE TIMER

You can display the timer on the Stage to show how long a program has been running.



Click the **Sensing** button.

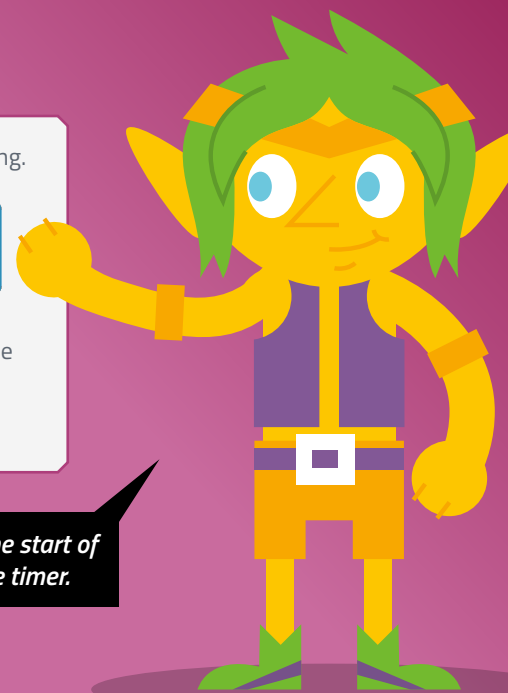


Find the timer code block and click its checkbox.



This block sets the time back to 0.

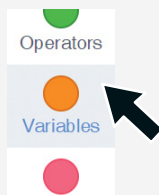
Use this block at the start of a game to reset the timer.



VARIABLES

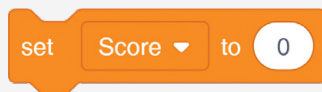
Variables are a special way that we can store a number or value in our program.

Score 12



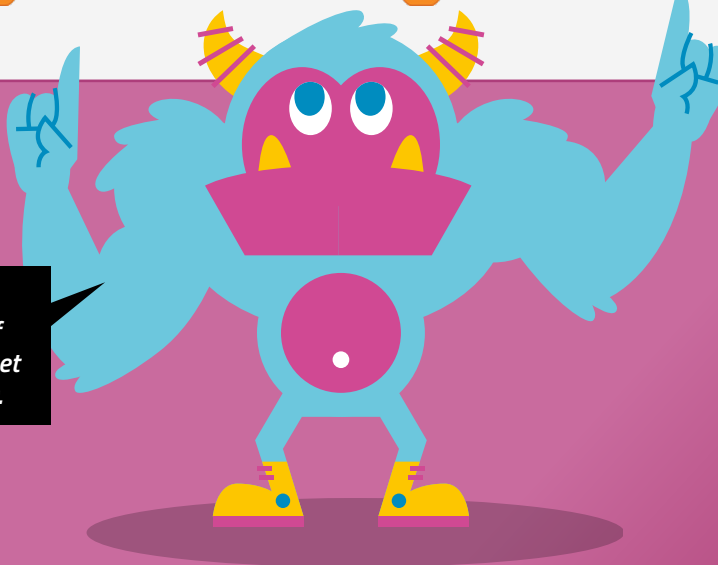
The value that you store in variables can *change*. This makes it perfect to store the score for a game!

You can find the variable blocks in the Variable group.



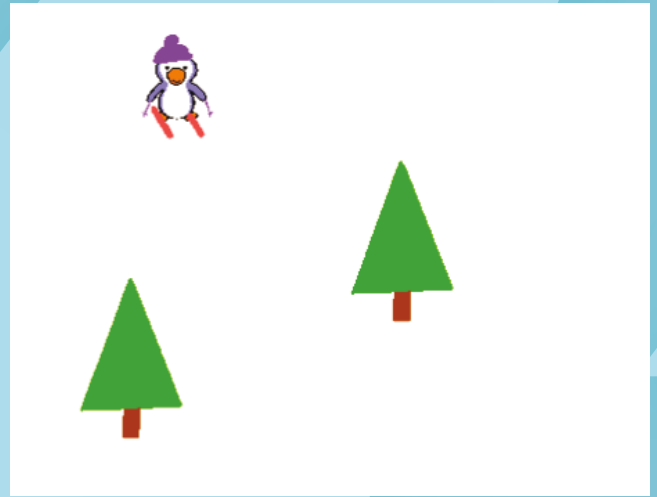
This code increases the score by 1.

Use this code at the start of a game to reset the score to 0.



PENGUIN SKI SCHOOL

In this game, a penguin has to ski down a mountain while avoiding trees. You'll adapt a ready-made sprite to create a skiing penguin. You'll place the trees in random positions on the Stage.



1. NO CATS



Click the **X** to delete the cat.

2. ADD A PENGUIN SPRITE

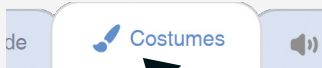


Click the **Choose a Sprite** button.



Scroll through the sprites and click **Penguin 2**.

3. GET READY TO EDIT



Click the **Costumes** tab at the top left of the screen.



Click the **Convert to Bitmap** button.

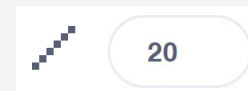
4. SET UP THE LINE TOOL



Select the **Line** tool.



Choose a purple color.



Type in **20** to make the line thick.



5. ADD SOME SKIS



Use the **Line** tool to draw two skis for the penguin.

6. ADD A HAT



Paint a hat with the **Brush** tool.

7. ADD ANOTHER SPRITE

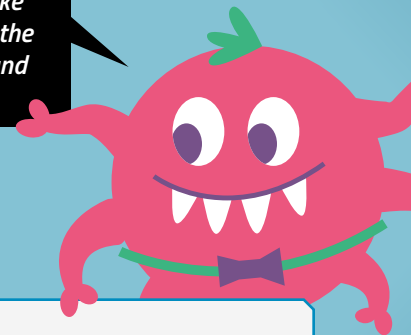


Let your mouse hover over the **Choose a Sprite** button.



Click the **Paint** option.

*If you make a mistake drawing, click the **Undo** button and try again!*



8. DRAW A TREE



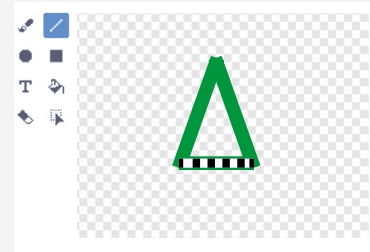
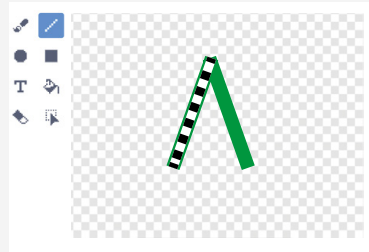
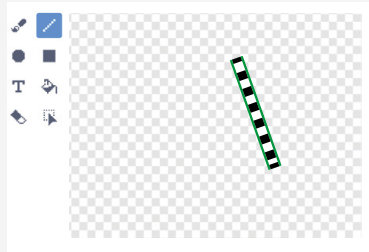
Click the **Convert to Bitmap** button.



Choose the **Line** tool.



Choose a green color.



Click the **Fill** tool.



Fill in the tree with green.



Choose a brown color.



Select the **Rectangle** tool.

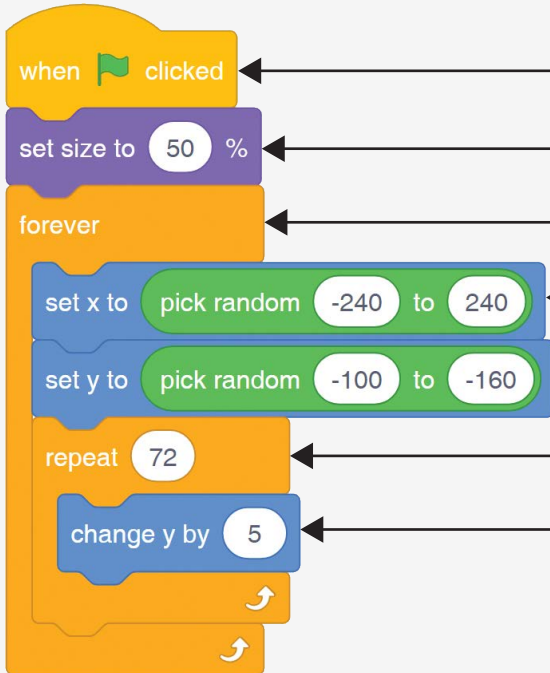


Use the mouse to draw the tree trunk.

9. CODE THE TREE

 Code

Click the **Code** tab at the top left and then drag this code into the Coding Area:



The code consists of the following blocks:

- when green flag clicked** (yellow block)
- set size to 50 %** (purple block)
- forever** loop (orange block) containing:
 - set x to pick random -240 to 240** (green block)
 - set y to pick random -100 to -160** (green block)
 - repeat 72** loop (orange block) containing:
 - change y by 5** (blue block)

When the green flag is clicked, run the following code.

Shrink the tree to half its size.

Repeat the following code forever.

Set x to a random value to determine how far across the Stage the tree is positioned.

Set y to a random value near the bottom of the Stage.

Repeat the following code 72 times.

(This is because $5 \times 72 = 360$ and the screen is 360 steps high.)

Move the tree up the screen.

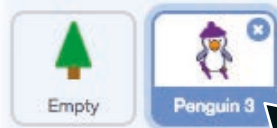


Click the **green flag** to test your code so far. Only the tree should move, traveling up the Stage. Once it reaches the top, it should start again from a random place at the bottom.

Now that you've coded the tree, let's program the penguin.



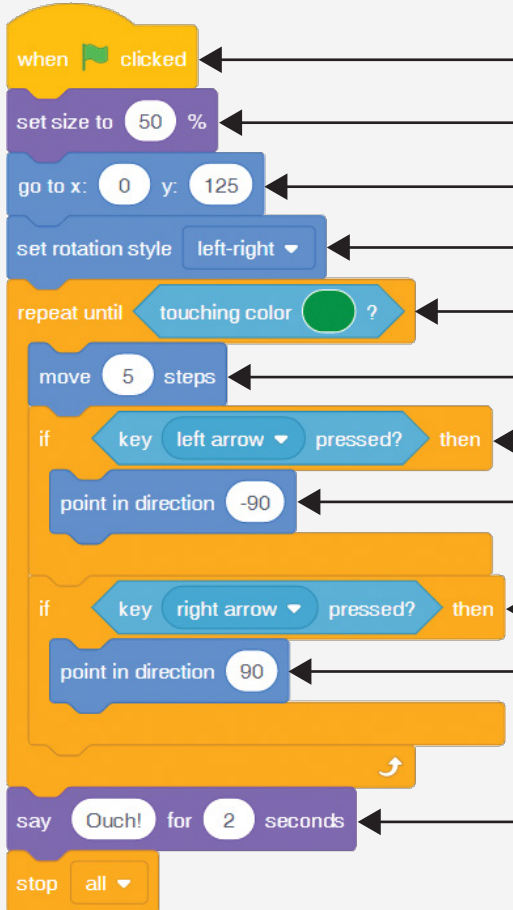
10. SELECT THE PENGUIN



Click the **Penguin** icon in the Sprite List.

11. CODE THE PENGUIN

Drag the following code into the Coding Area:

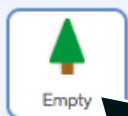


The code consists of the following blocks:

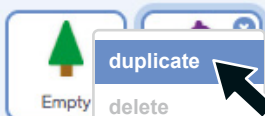
- when green flag clicked**: Run the following code when the green flag is clicked.
- set size to 50 %**: Shrink the penguin to half its normal size.
- go to x: 0 y: 125**: Move the penguin to the top of the Stage, in the center.
- set rotation style left-right**: Set the penguin image to flip left and right instead of rotating.
- repeat until touching color green ?**: Repeat the following code until the skiing penguin hits the tree. (See page 16 for help setting the color.)
- move 5 steps**: Move the penguin in the direction it's facing.
- if key left arrow pressed? then**: If the left arrow key is pressed, then run this code.
- point in direction -90**: Set the penguin to move to the left.
- if key right arrow pressed? then**: If the right arrow key is pressed, then run this code.
- point in direction 90**: Set the penguin to move to the right.
- say Ouch! for 2 seconds**: The penguin has hit a tree, so show a message.
- stop all**: Stop the code.

Duplicating the tree copies its code as well as its picture.

12. ADD ANOTHER TREE



In the Sprite List, right-click the tree icon.



Click **duplicate**.



Click the **green flag** to test your code. Press the **left** and **right arrow** keys to steer the penguin left and right as you avoid the trees.

This game shows you how to use random numbers to make your games more exciting. You could try using this technique with some of the games you made in the previous chapters. When you're ready for the next challenge, turn to page 56 and try building the Cross-Country Skier game.

SHARK DIVER

In this game, the diver has to find all the treasure in the sea. But the diver will need to avoid sharks swimming around! You'll control the diver by using your mouse. You'll create multiple shark sprites to make the game even harder. A score variable will keep track of the treasure found.



1. NO CATS

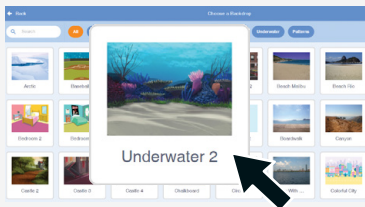


Cats don't like sharks. Click the X to delete the cat.

2. UNDERWATER

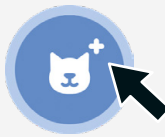


Click the **New Backdrop** button.



Find **Underwater 2** and then click it to set the backdrop.

3. ADD A DIVER



Click the **Choose a Sprite** button.

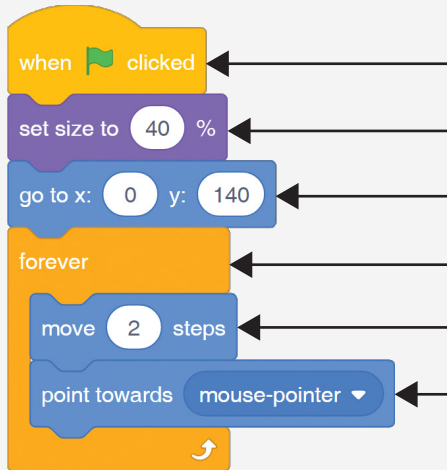


Scroll through the sprites and click **Diver 2**.



4. CODE THE DIVER

Drag this code into the Coding Area:



The code consists of the following blocks:

- when green flag clicked**: When the green flag is clicked, run the following code.
- set size to 40 %**: Shrink the diver to 40 percent of its size.
- go to x: 0 y: 140**: Start the diver at the top center of the Stage.
- forever loop**: **Keep repeating this code forever.**
 - move 2 steps**: Move it 2 steps forward.
 - point towards mouse-pointer**: Point the diver toward the mouse.

Click the **green flag** to test your code. The diver should swim toward your mouse as you move it around the screen.

5. ADD SOME TREASURE



Click the **Choose a Sprite** button.

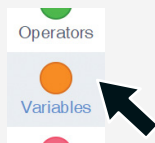


Click the **Ball** sprite.

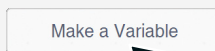
6. MAKE A SCORE VARIABLE



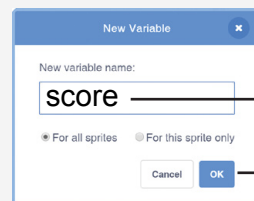
Click the **Code** tab.



Click the **Variables** button.



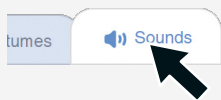
Click **Make a Variable**.



Type **score**.

Click **OK**. Now the score will appear at the top-left corner of the Stage.

7. IMPORT A SOUND EFFECT



Click the **Sounds** tab.
(It's at the top left.)



Click the **Choose a Sound** button. (It's at the bottom left.)



Scroll through the sounds to find the **Fairydust** icon. Click it to import it.

8. CODE THE TREASURE

Click the Code tab and then drag this code into the Coding Area:

```
when green flag clicked
  set size to 30 %
  set Score to 0
  forever loop
    go to random position
    repeat until touching Diver2 ?
      change color effect by 2
    change Score by 1
    start sound Fairydust
```

Run the following code when the green flag is clicked.

Set the size of the treasure to 30 percent.

Reset the score to 0.

Keep repeating all of the following code forever.

Go to a random place on the Stage.

Repeat the next code block until the diver gets to the treasure.

Make the treasure gently change color.

The diver has found the treasure, increase the score.

Play the sound effect.

9. ADD A SHARK



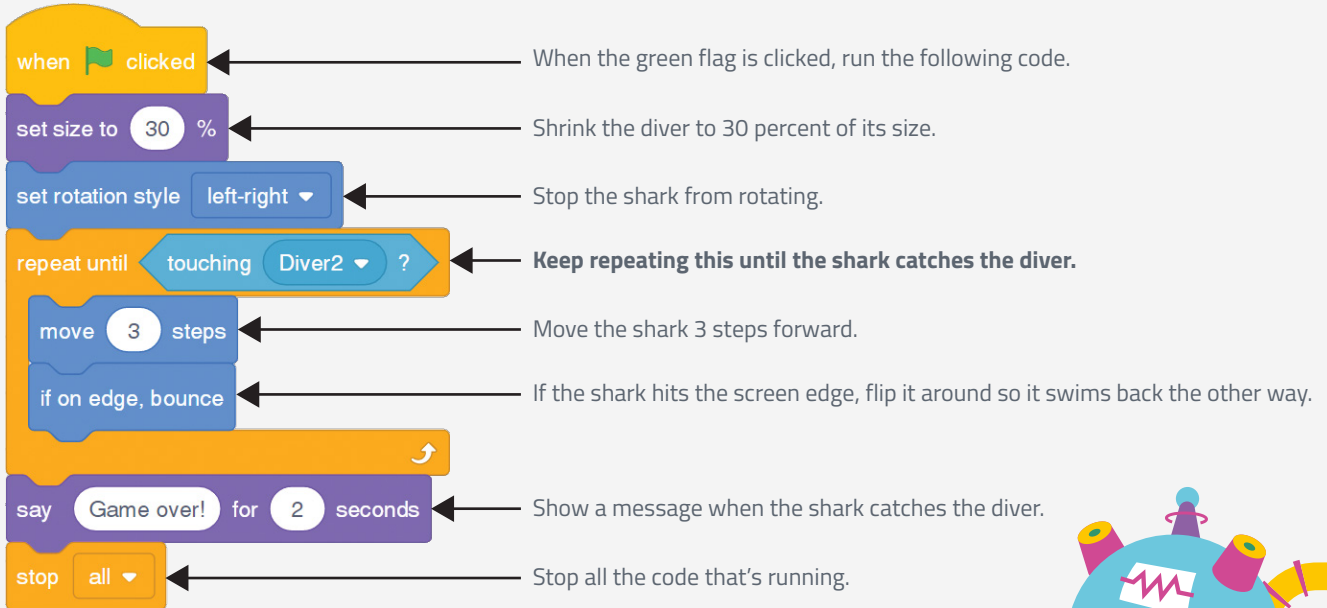
Click the **Choose a Sprite** button.



Scroll through the sprites and click **Shark**.

10. CODE THE SHARK

Drag this code into the Coding Area:



when green flag clicked ← When the green flag is clicked, run the following code.

set size to 30% ← Shrink the diver to 30 percent of its size.

set rotation style left-right ← Stop the shark from rotating.

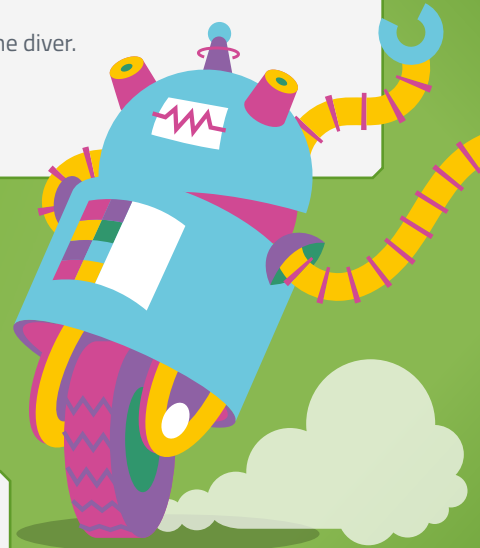
repeat until touching Diver2? ← **Keep repeating this until the shark catches the diver.**

move 3 steps ← Move the shark 3 steps forward.

if on edge, bounce ← If the shark hits the screen edge, flip it around so it swims back the other way.

say Game over! for 2 seconds ← Show a message when the shark catches the diver.

stop all ← Stop all the code that's running.



11. ADD MORE SHARKS



In the Sprite List, right-click the **Shark**.

Click the **green flag** to test your code.

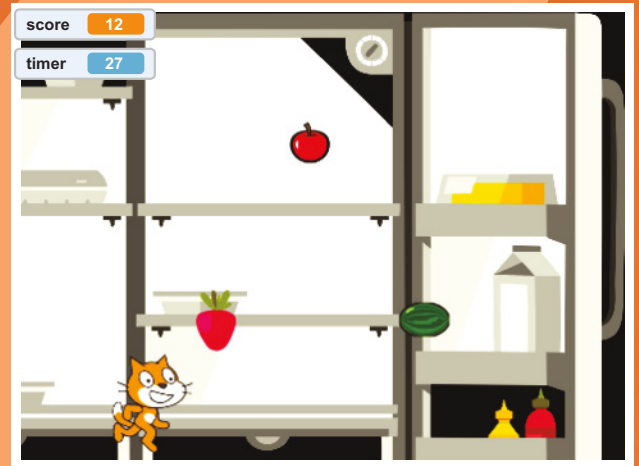
Click **duplicate**.

Duplicate more sharks and drag them apart. This will make sure they start in different places.

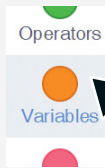
Ready for a challenge? Turn to page 56 and try to create your own Dino City game to rescue people from flying pterodactyls.

FRUIT CATCHER

This is a basic catching game. The player, positioned at the bottom of the screen, can move left or right. The player's job is to catch fruit that's falling down. You'll use a variable to keep the score and a timer to limit the game to 30 seconds. To move the fruit, you'll change its x- and y-coordinates.



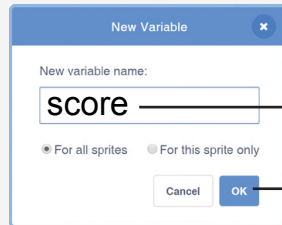
1. MAKE A SCORE VARIABLE



Click the **Variables** button.



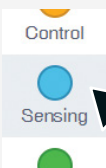
Click the **Make a Variable** button.



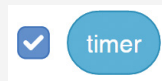
Type in **score**.

Click **OK**.

2. SHOW THE TIMER



Click the **Sensing** button.



Find the timer code block and click its checkbox.

This places the timer onscreen to show how long the game has been running.



3. CODE THE CAT

Drag this code into the Coding Area to make the cat move:

when clicked ← Run the following code when the green flag is clicked.

set to ← Reset the score to 0.

go to x: y: ← Start the cat at the center of the Stage at the bottom.

reset timer ← Reset the timer.

set size to % ← Shrink the cat to 70 percent of its size.

set rotation style ← Stop the cat from fully rotating—just let it flip left or right.

repeat until ← **Keep repeating the following code until 30 seconds are up.**

move steps ← Move the cat 5 steps forward.

next costume ← Show the next cat costume to make the cat look as though it's walking.

stop ← Stop any other code from running, like the falling fruit.

when key pressed ← Depending on which key is pressed, run one of these blocks.

point in direction ← Point the cat to the left or the right.

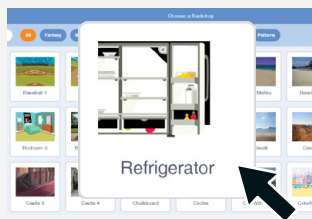
when key pressed

point in direction

4. COOL IT DOWN



Click the **Choose a Backdrop** button.



Click the **Refrigerator**.

5. ADD A SPRITE



Click the **Choose a Sprite** button at the bottom right.



Scroll through the sprites and click **Apple**.

Now let's add a sprite to be the falling fruit. You'll click Choose a Sprite button on the right-hand side. You'll need to change the sprite into different fruits. Add an extra costume to your fruit sprite for each type of fruit. The Choose a Costume button is at the bottom left, under the Costumes tab.

6. ADD A BANANA COSTUME



Click the **Choose a Costume** button at the bottom left.

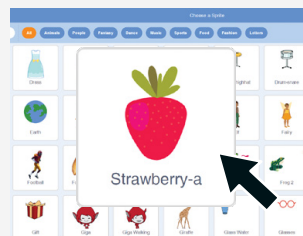


Scroll through the costumes and click **Bananas**.

7. ADD MORE COSTUMES



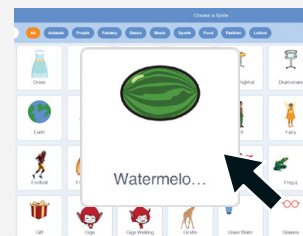
Click **Choose a Costume**.



Add a **Strawberry** costume.



Click **Choose a Costume**.



Add a **Watermelon** costume.



8. CODE THE FRUIT

Code

Click the **Code** tab and then drag this code into the Coding Area:

The code consists of the following blocks:

- when green flag clicked** (yellow)
- set size to 50 %** (purple)
- forever** loop (yellow)
- go to x: pick random -220 to 220 y: 160** (blue)
- repeat pick random 1 to 3** (green)
- next costume** (purple)
- repeat until y position < -160** (orange)
- change y by -3** (blue)
- if touching Sprite1 ? then** (yellow)
- start sound Chomp** (purple)
- change Score by 1** (orange)
- set y to -200** (blue)

Annotations:

- When the green flag is clicked, run the following code.
- Shrink the fruit to half its size.
- Keep repeating this code forever.
- Start the fruit somewhere random at the top of the screen.
- Repeat the next line a random number of times.
- Show the next costume. (These blocks make sure the fruit has a different costume each time.)
- Repeat this code until the fruit reaches the bottom.
- Move the fruit down slowly.
- If the cat catches the fruit, then run this code.
- Play the sound effect.
- Increase the score by 1.
- Move the fruit down to the bottom of the Stage so you can break out of the repeat until loop.

9. ADD MORE FRUIT



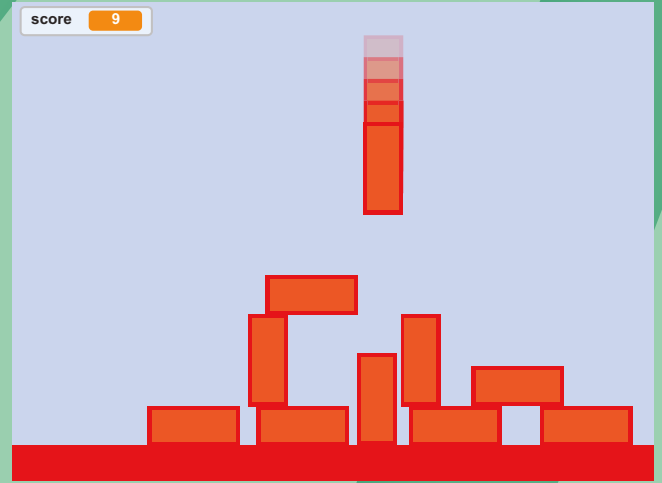
Right-click the **Apple** in the Sprite List.

Click **duplicate**.

Now turn to page 57 and try to create your own Mystery Catch game.

BRICK TRICKS

In many computer games, such as Tetris, bricks fall down the screen to form patterns. In this game, the player has to guide a falling brick to fit into a space. The goal is to fit together as many bricks as possible on the Stage. When the brick sprite reaches the bottom or hits another brick, you simply copy it back onto the Stage. Even though it looks like the game has many bricks, it really has just one brick!

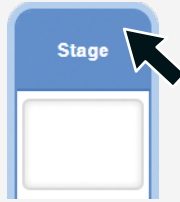


1. NO CATS

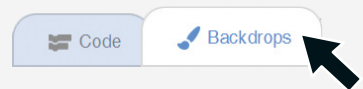


Click the X to delete the cat.

2. SELECT THE BACKDROP



In the sprites pane, click the **Stage** icon. (It's at the bottom right.)



Click the **Backdrops** tab (at the top left of the screen).

Turn to page 16 for help with setting colors.

3. START DRAWING



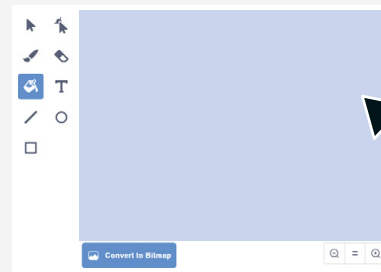
Click the **Convert to Bitmap** button.



Choose the **Fill** tool.



Click the **Fill** drop-down and set it to a light blue.



Color the Drawing Area.

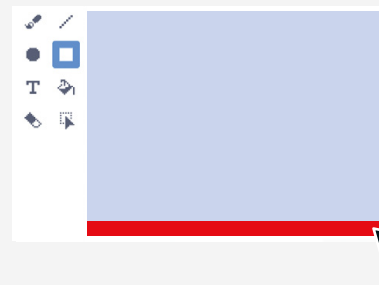
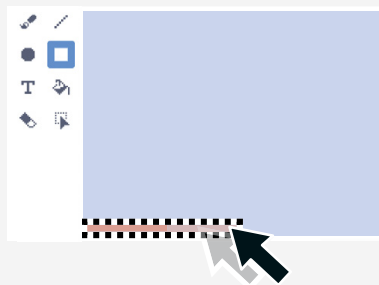
4. DRAW THE FLOOR



Select the **Rectangle** tool.



Choose a red color.

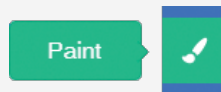
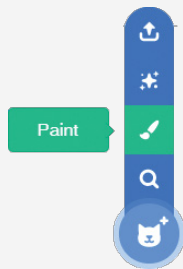


Use the mouse to draw a thin, wide rectangle at the bottom of the Drawing Area.


5. PAINT A NEW SPRITE



Let your mouse hover over the **Choose a Sprite** button.



Find the **Paint** option and click it.

 *If you make a mistake drawing, click the **Undo** button and try again!*



6. DRAW A BRICK



Click the **Convert to Bitmap** button.



Choose the **Rectangle** tool.



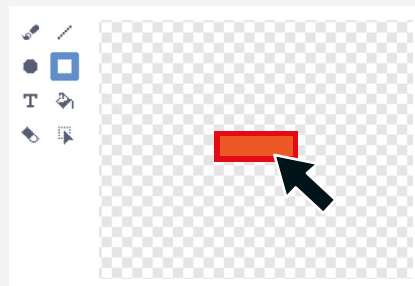
Choose a red.



Draw a small red rectangle in the center of the Canvas.



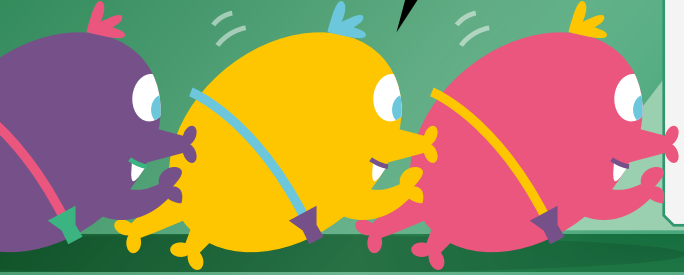
Choose an orange.



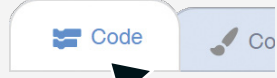
Now draw an orange rectangle inside the red one.



Now that you have your backdrop and sprite ready, it's time to add some code!



7. START CODING

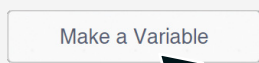


Click the **Code** tab at the top left of the screen.

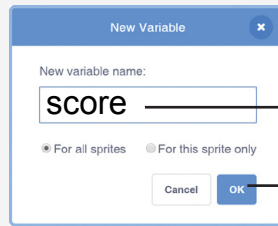
8. MAKE A VARIABLE



Choose **Variables**.



Click the **Make a Variable** button.



Type in **score**.

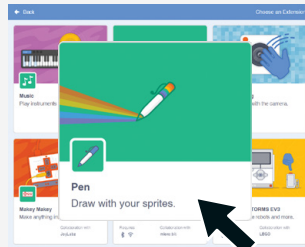
Click **OK**.

An extension is a set of extra code blocks. Use these blocks to do a special task such as drawing, making sounds, or connecting to a robot.

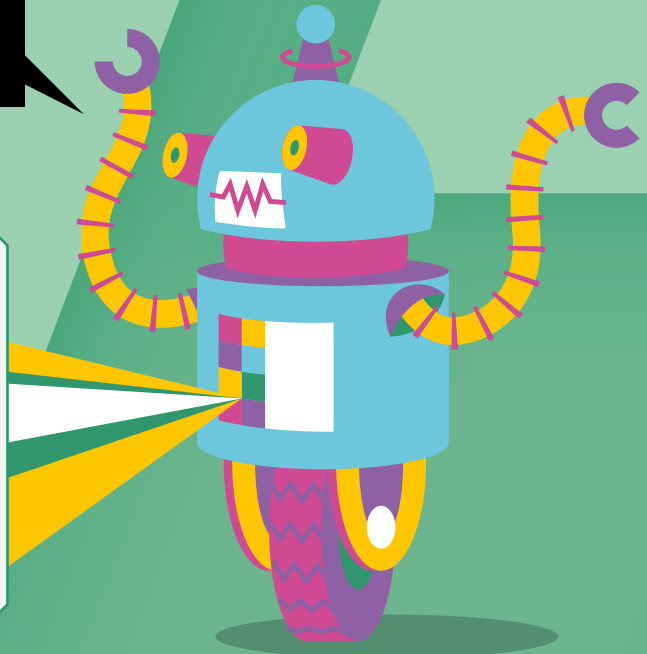
9. ADD THE PEN EXTENSION



Click the **Add Extension** button (at the bottom left of the screen).



Click the **Pen Extension**.



10. MAIN CODE

Drag this code into the Coding Area:

when green flag clicked ← When the green flag is clicked, run the following code.

set Score to 0 ← Reset the score to 0.

erase all ← Clear any bricks from the previous game.

go to x: 0 y: 145 ← Move the brick to the top of the Stage in the center.

repeat until y position = 150 ← **Keep repeating the following code until there's no room for any more bricks.**

go to x: 0 y: 150 ← Move the brick to the top of the Stage in the center.

repeat until touching color red ? ← **Keep repeating the following code until the brick hits the floor or another brick.**

change y by -5 ← Make the brick fall to the bottom of the Stage.

stamp ← Now that the brick has hit something, you draw it on the backdrop in that exact location. Then you move the brick up to the top of the Stage.

change Score by 1 ← Increase the score by 1.

start sound pop ← Play a sound effect.

say Game over! ← The game is over, so show a message.

when left arrow key pressed ← When the *left* arrow key is pressed:

change x by -20 ← Move the brick to the left.

when right arrow key pressed ← When the *right* arrow key is pressed:

change x by 20 ← Move the brick to the right.

when space key pressed ← When the spacebar is pressed:

turn 90 degrees ← Rotate the brick 90 degrees.

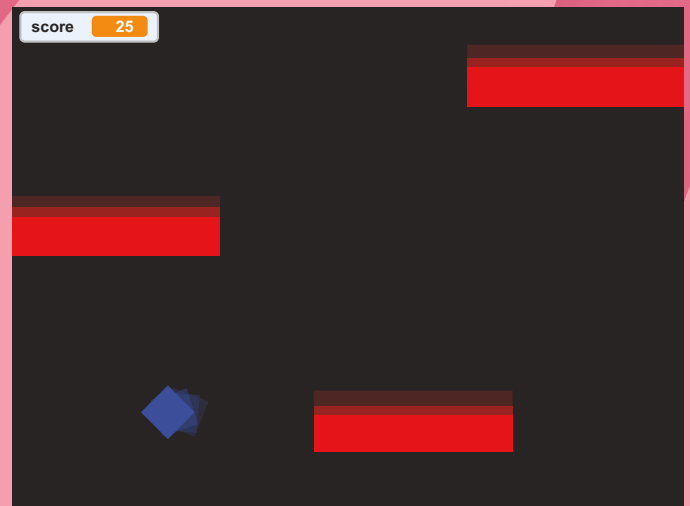


Click the **green flag** to test your code. Use the arrow keys to guide the bricks into an empty space.

Although this game has only one sprite, the Pen extension code lets you use it over and over again as long as the game runs. By using the stamp command, the location the brick stops becomes part of the backdrop. When you're ready to take this to the next level, turn to page 57 and try the Mega Brick Builder challenge!

COLORED WALLS

Lots of games use simple colored shapes as graphics. Usually in those games, the player has to travel past a series of colored obstacles to get to their goal. In this game, you'll use cloning to create walls that move toward the player. To make the game get gradually harder, you'll enlarge the walls every time a new one is added. You'll also raise the pitch of the sound effect with each new wall, increasing the tension!

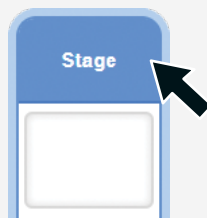


1. NO CATS



Click the X in the corner of the sprite to delete it.

2. SELECT THE BACKDROP

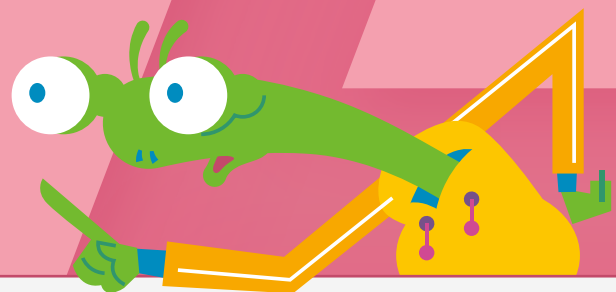


In the sprites pane, click the **Stage** icon. (It's at the bottom right.)

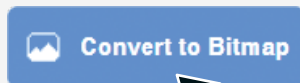


Click the **Backdrops** tab at the top left.

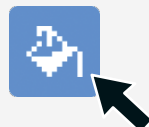
Turn to page 16 for help with setting colors.



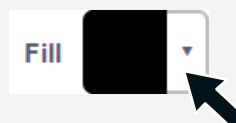
3. PAINT THE BACKGROUND BLACK



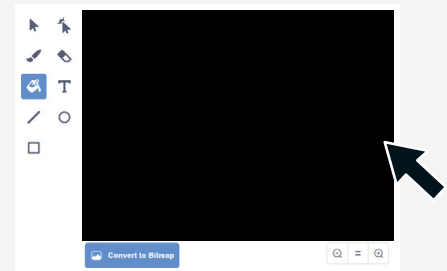
Click the **Convert to Bitmap** button.



Choose the **Fill** tool.



Click the **Color** drop-down and select black.

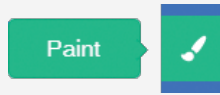
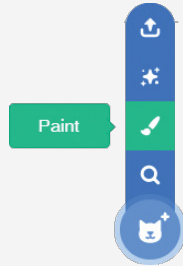


Color the Drawing Area.

4. ADD THE WALL SPRITE



Let your mouse hover over the **Choose a Sprite** button.



Click the **Paint** option.

5. DRAW THE WALL



Click the **Convert to Bitmap** button.



Choose the **Rectangle** tool.

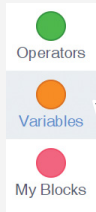


Choose red.

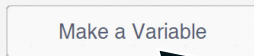


Draw a thin rectangle about one-quarter of the width of the Drawing Area. Make sure it's in the center.

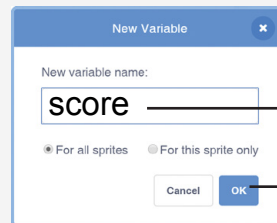
6. MAKE A VARIABLE



Choose **Variables**.

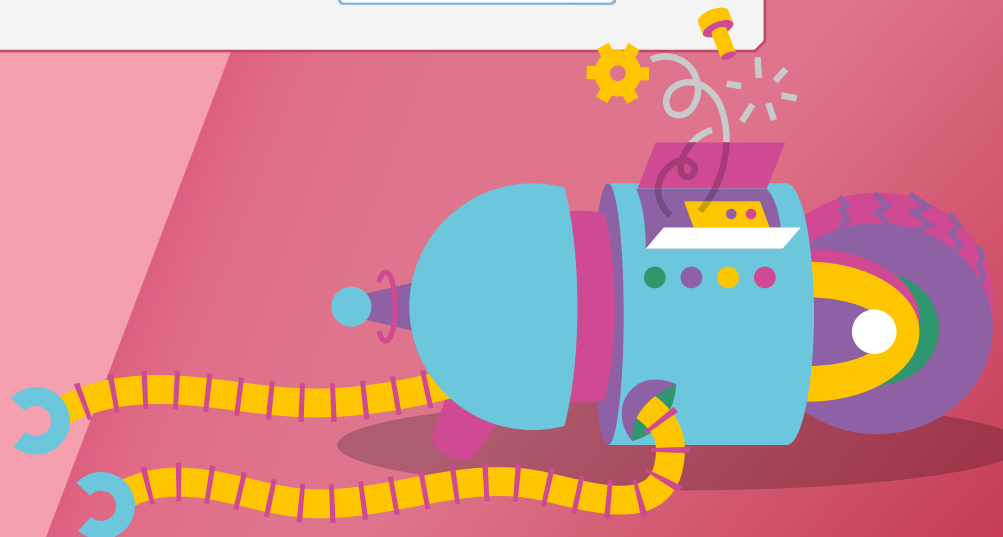


Click the **Make a Variable** button.



Type in **score**.

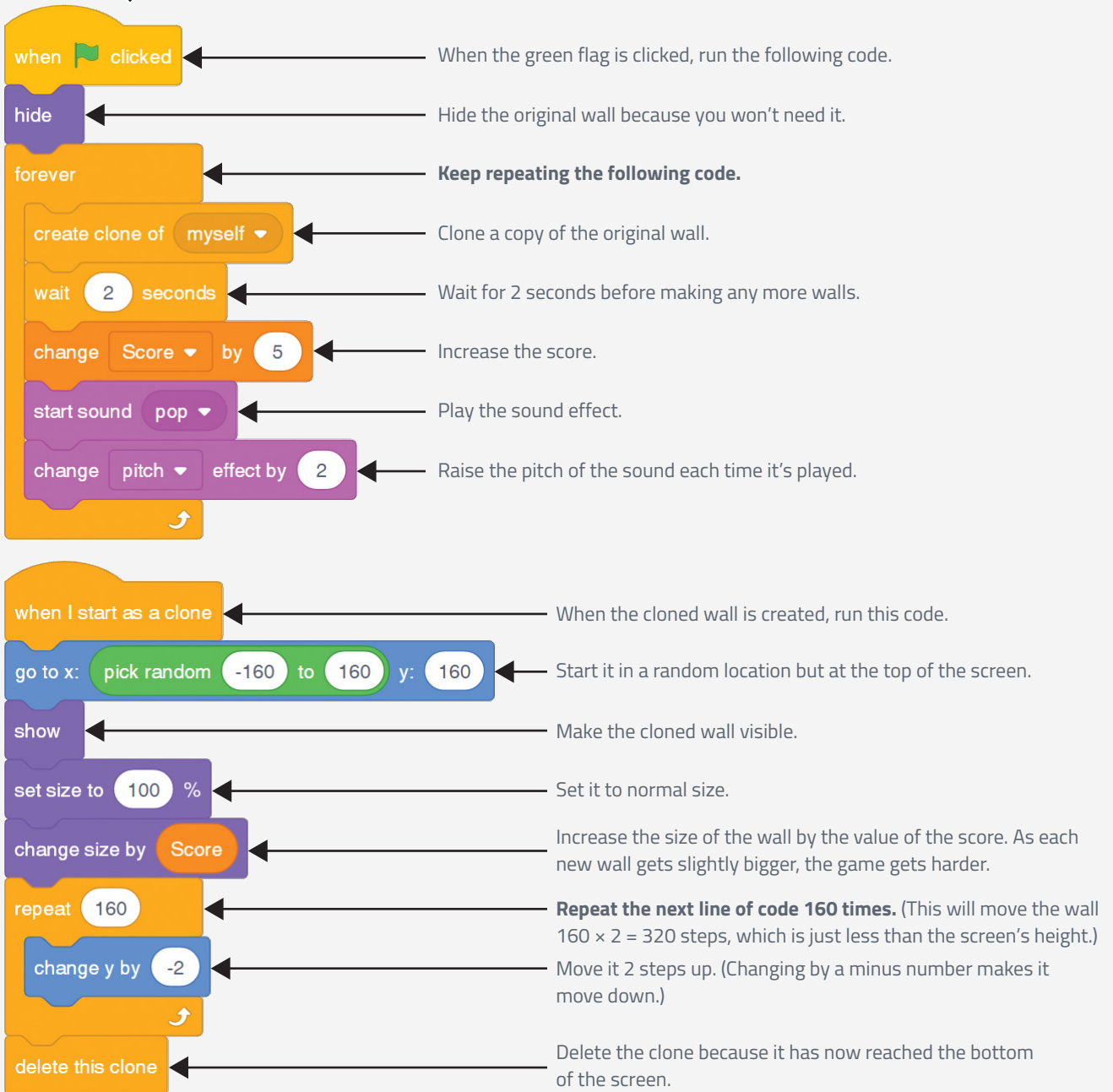
Click **OK**.



7. CODE IT

Code

Click the **Code** tab and then drag this code into the Coding Area:



The image shows two sets of Scratch code blocks with arrows pointing to specific parts and explanatory text. The first set is a 'when green flag clicked' event block followed by a 'hide' block, a 'forever' loop containing 'create clone of myself', 'wait 2 seconds', 'change Score by 5', 'start sound pop', and 'change pitch effect by 2'. The second set is a 'when I start as a clone' event block followed by 'go to x: pick random -160 to 160 y: 160', 'show', 'set size to 100%', 'change size by Score', a 'repeat 160' loop containing 'change y by -2', and 'delete this clone'.

When the green flag is clicked, run the following code.

Hide the original wall because you won't need it.

Keep repeating the following code.

Clone a copy of the original wall.

Wait for 2 seconds before making any more walls.

Increase the score.

Play the sound effect.

Raise the pitch of the sound each time it's played.

When the cloned wall is created, run this code.

Start it in a random location but at the top of the screen.

Make the cloned wall visible.

Set it to normal size.

Increase the size of the wall by the value of the score. As each new wall gets slightly bigger, the game gets harder.

Repeat the next line of code 160 times. (This will move the wall $160 \times 2 = 320$ steps, which is just less than the screen's height.)

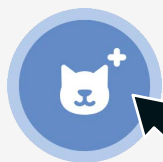
Move it 2 steps up. (Changing by a minus number makes it move down.)

Delete the clone because it has now reached the bottom of the screen.

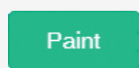


Click the **green flag** to test your code so far. A series of red rectangles should slowly move down the screen from the top to the bottom. Each rectangle should be slightly bigger than the previous one. Now let's add the player.

8. ADD THE PLAYER SPRITE



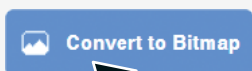
Let your mouse hover over the **Choose a Sprite** button.



Select the **Paint** option.



9. DRAW THE PLAYER



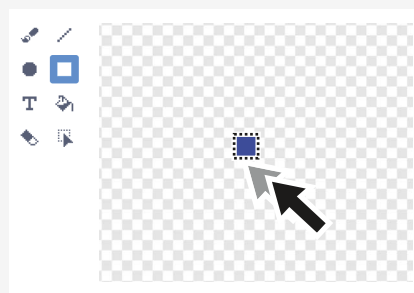
Click the **Convert to Bitmap** button.



Choose the **Rectangle** tool.



Choose a blue color.

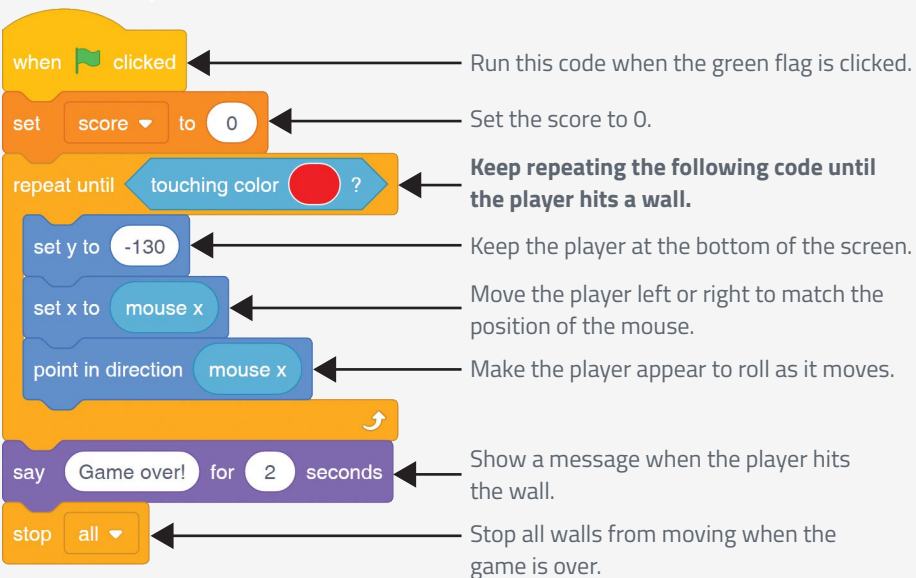


Draw a very small square in the center.

10. CODE IT

Code

Click the **Code** tab and then drag this code into the Coding Area:



Click the **green flag** to play your game. How many points can you score?

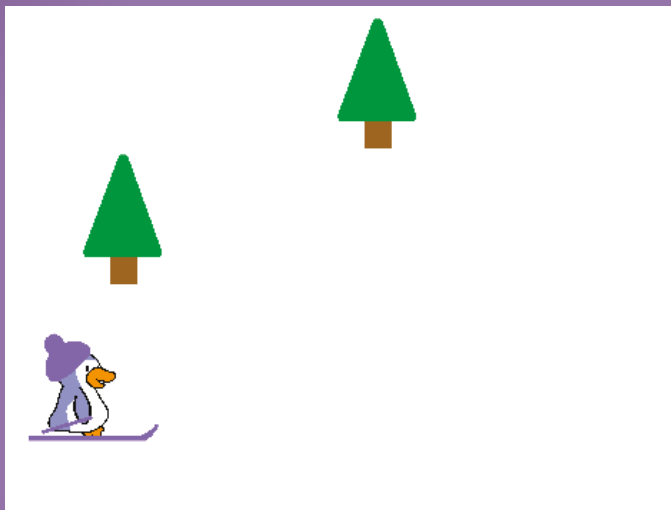
This game uses cloned sprites to move multiple objects on the screen. The player moves by following the x-coordinate of the mouse. The score increases by 5 each time a wall is passed. The score is used to set the size of the next wall. Increasing by 5 makes the change in size more noticeable than changing the score by just 1. Use these ideas to build the Colored Walls 2 game on page 57.

CHAPTER 3 CHALLENGES

In this chapter, variables are key to how the games work. You learned how to use a timer to limit how long a game runs, and to give it a sense of competition. You used touching blocks to check when sprites had been caught and then increased the score by changing the score variable.



After coding the programs in this chapter, try these challenges.



CHALLENGE 1 CROSS-COUNTRY SKIER

Make a side scroller version of the Penguin Ski School game. Look through the code on page 36 to help you. Make the trees start on the right side of the Stage, and change their x value by -5 to make them scroll across. You'll need to make the penguin go up and down by changing its y value when keys are pressed.



CHALLENGE 2 DINO CITY

Use the ideas from Shark Diver (page 40) to make a new game. This time, make the player fly a helicopter around to rescue people from flying dinosaurs. Instead of using a diver, draw a helicopter for the player, and then add similar code to make it fly. Add flying pterodactyls instead of sharks, using similar code to what was used for the shark sprites. Finally, instead of a treasure sprite, draw a person and code it.



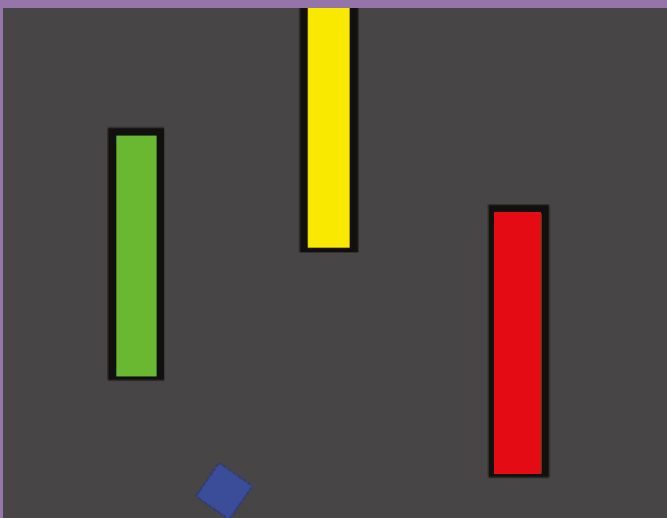
CHALLENGE 3 MYSTERY CATCH

In Fruit Catcher (page 44), you learned to make a simple catching game with different kinds of fruit that fell down the screen. Use the same ideas to build your own game. Choose a background and then choose a sprite to start with. What will fall down? How many points will you get for each object you catch? Add more costumes to randomize what falls. Refer to the code in Fruit Catcher when you need help.



CHALLENGE 4 RANDOM BRICK TRICKS

Extend the Brick Tricks game. Add extra costumes to the brick in different shapes and colors. Getting the graphics right is key to this game. Add code to randomly change the costume (see the Fruit Catcher game on page 44 to help with that part). How else can you improve the game?



CHALLENGE 5 COLORED WALLS 2

Can you code a new version of the Colored Walls game to make the blocks move across the screen? Use the ideas from the game on page 52 to help you. You'll need to swap around many of the x and y values and code blocks. Add more costumes and make them change color at the start of the loop, the way Fruit Catcher does. Change the background color and border of the colored rectangle to make it simpler to test for the end of the game.

MATH MATTERS

As your programs get more sophisticated, you need to use more complex math. It takes a little bit of math to make a game get gradually faster or slower or to change the pitch of sound effects. But don't worry; we'll go through this step-by-step!

To combine blocks, drag them by the left side. A white glow will show around the space that the block can drop into.

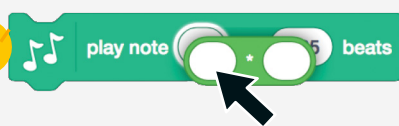
MATH SYMBOLS

All coding languages, including Scratch, use special symbols for multiplication and division:



COMBINING LONGER CALCULATIONS

You can use math to create special sound effects. In the Robot Power Up game, you'll combine calculations to change the pitch of the sound as the robot moves up and down.



Start by dragging a **multiplication** block into a **play note** code block.



Type **0.1** into to the left side of the multiplication block.



Drag an **addition** block to the right side of the multiplication block.



Drop a **y position** block to the left side of the addition block.



Type **100** into the right side of the addition block.

Instead of playing the same note all the time, the note value will change by 0.1 times the y+100 position of the sprite.

When a calculation has operator blocks within operator blocks, Scratch will calculate the innermost block first, as if it was in brackets.

USING MATH TO MAKE A GAME GET HARDER

The longer you play the game, the faster the donuts appear!

```

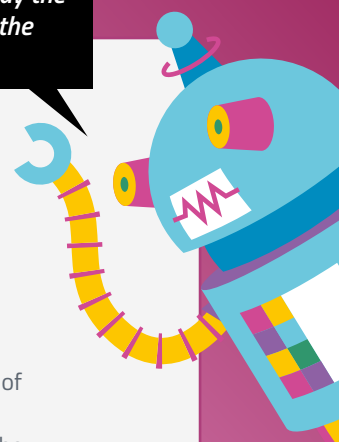
set Delay to 4
forever
  create clone of myself
  wait pick random 0.5 to Delay seconds
  if Delay > 0.5 then
    change Delay by -0.1
  
```

In Invasion of the Donuts on page 76, you'll use math to make the game get progressively harder.

You'll use a delay variable to pause between creating each new donut clone.

The delay variable is decreased by 0.1 (changed by *minus* 0.1) until it gets to 0.5.

You'll use the pick random block to introduce a random element into the game. Instead of waiting for the value of the delay variable, the code waits for a random number between 0.5 and the value of the variable. This makes the donuts start arriving more quickly and at slightly random times.



BROADCASTING MESSAGES

In this chapter, you'll learn a new way to let sprites communicate with each other. This is very useful when you have lots of sprites doing more complex things. No math required!

```

broadcast message1
  
```

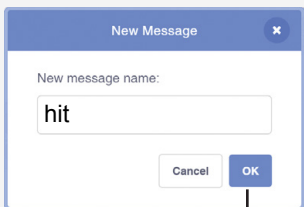
Start by dragging a broadcast block into your code.

```

broadcast message1
  
```



Click **New Message**.



Type the message you want to share with other sprites and then click **OK**.

CLONING

You've used the duplicate option to make copies of your sprites. But there's a better way to do this using code. To clone sprites, use the create clone of myself block:

```

when clicked
  repeat 5
    create clone of myself
  
```

Mmm, I love donuts. Could you make some more?

Decide how many cloned sprites you want to create.

Clone the sprite.

Now, to make each clone do something, use the when I start as a clone block:

```

when I start as a clone
  go to random position
  forever
    move 2 steps
  
```

Move each clone to a random place on the Stage.

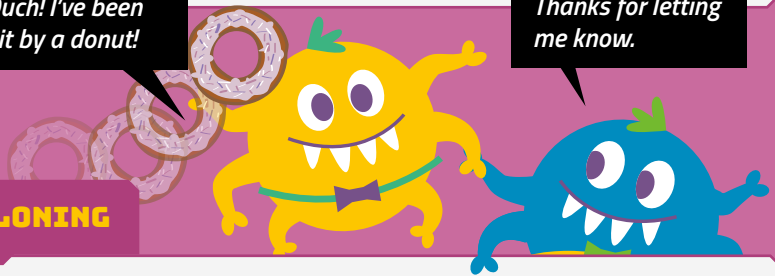
Repeat the following code forever.

Move the clone 2 steps forward.

Are these donuts cloned?

Ouch! I've been hit by a donut!

Thanks for letting me know.



SUPER SNAKE

This is a version of the classic computer game called *Snake*. The player has to control a snake as it moves around the screen while avoiding colliding with itself. As the snake gets longer, the game gets harder. You'll use a coding technique called *cloning* to make the snake longer. Cloning is a bit like copying and pasting a sprite by using code.



1. NO CATS



Click the **X** in the corner of the sprite to delete it.

2. SELECT THE BACKDROP



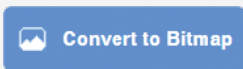
In the sprites pane, click the **Stage** icon. (It's at the bottom right.)



Click the **Backdrops** tab (at the top left of the screen).

Turn to page 16 for help with setting colors.

3. DRAW A BACKGROUND AND BORDER



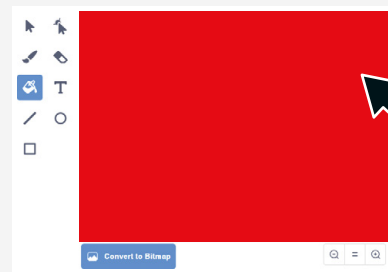
Click the **Convert to Bitmap** button.



Choose the **Fill** tool.



Click the **Fill** drop-down and select red.



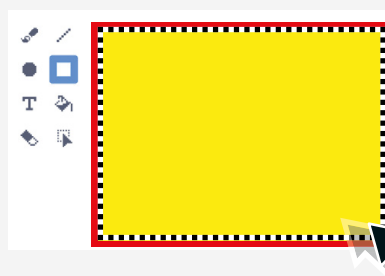
Color the screen.



Select the **Rectangle** tool.



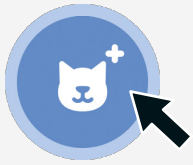
Choose yellow.



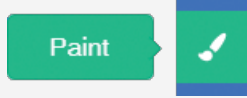
Use the mouse to draw a large yellow rectangle. It should leave a thin red border around the edge.

The border must be exactly the same color as the snake. You'll be using this to check for collisions.


4. ADD THE SNAKE SPRITE

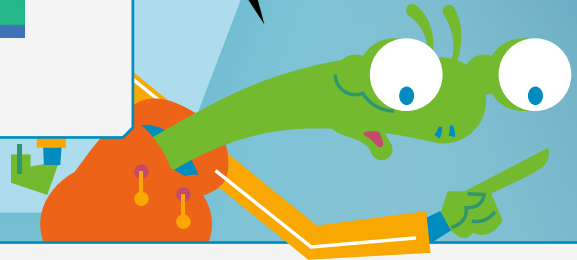


Let your mouse hover over the **Choose a Sprite** button.



Find the **Paint** option and click it.

 *If you make a mistake, click the Undo button and try again!*



5. DRAW THE BODY



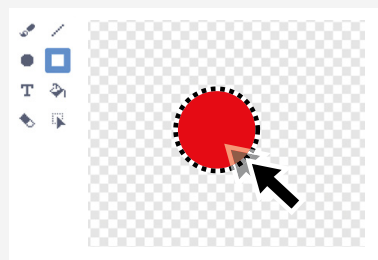
Click the **Convert to Bitmap** button.



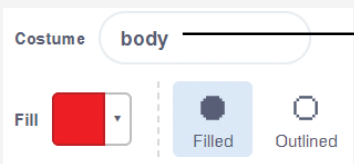
Choose the **Circle** tool.



Choose red.

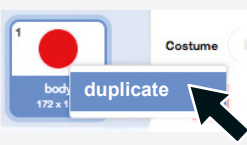


Draw a large red circle to be a section of the snake. (You'll shrink it later.)

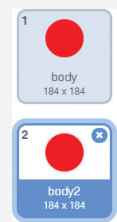


You'll need to draw two costumes for the snake. This one will be the body, so type **body** as its name.

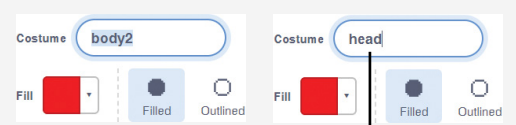
6. DUPLICATE THE BODY COSTUME



Right-click the **body** icon in the Sprite List. Click **duplicate** to create another copy of the body.

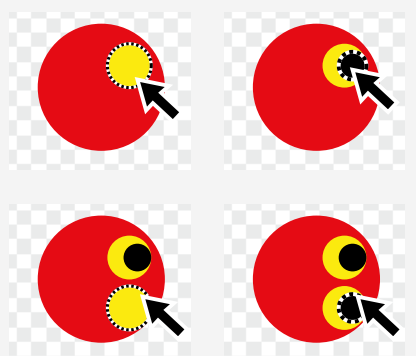


A new costume called **body2** should appear.




Rename the new costume **head**.

7. DRAW THE EYES



Use the **Circle** tool to draw two eyes.

8. DRAW THE TONGUE

Choose the **Brush** tool.  Select a green color.



Draw a tongue. Leave a small gap between it and the head.

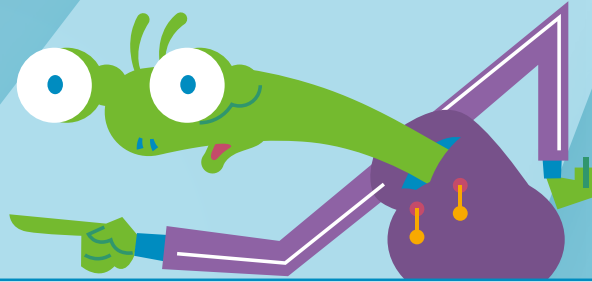
If you don't leave a gap, the snake will be touching red as soon as the game starts! That would make the game stop working.



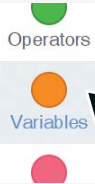
9. START CODING

Code

Click the **Code** tab at the top left of the screen.



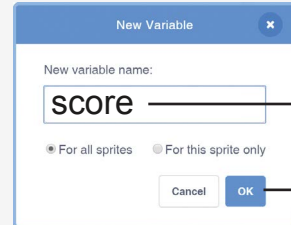
10. MAKE A VARIABLE



Click the **Variables** button.

Make a Variable

Click the **Make a Variable** button.



Type in **score**.

Click **OK**.

11. CODE THE SNAKE'S HEAD

Drag this code into the Coding Area:

when green flag clicked

When the green flag is clicked, run the following code.

set score to 0

Reset the score to 0.

set size to 15 %

Shrink the snake to 15 percent of its size.

switch costume to head

Show the costume with the snake's head.

point in direction 90

Make sure the snake is pointing to the right.

go to x: -100 y: 0

Move it just left of the center of the screen.

repeat until color green is touching red ?

Keep looping the following code until the snake's tongue hits part of its body or the red border.

move 5 steps

Move the snake's head forward 5 steps.

if key left arrow pressed? then

If the left arrow key is pressed, then run this block.

turn 15 degrees

Rotate the snake's head counterclockwise.

if key right arrow pressed? then

If the right arrow key is pressed, then run this block.

turn 15 degrees

Rotate the snake's head clockwise.

say Game over! for 2 seconds

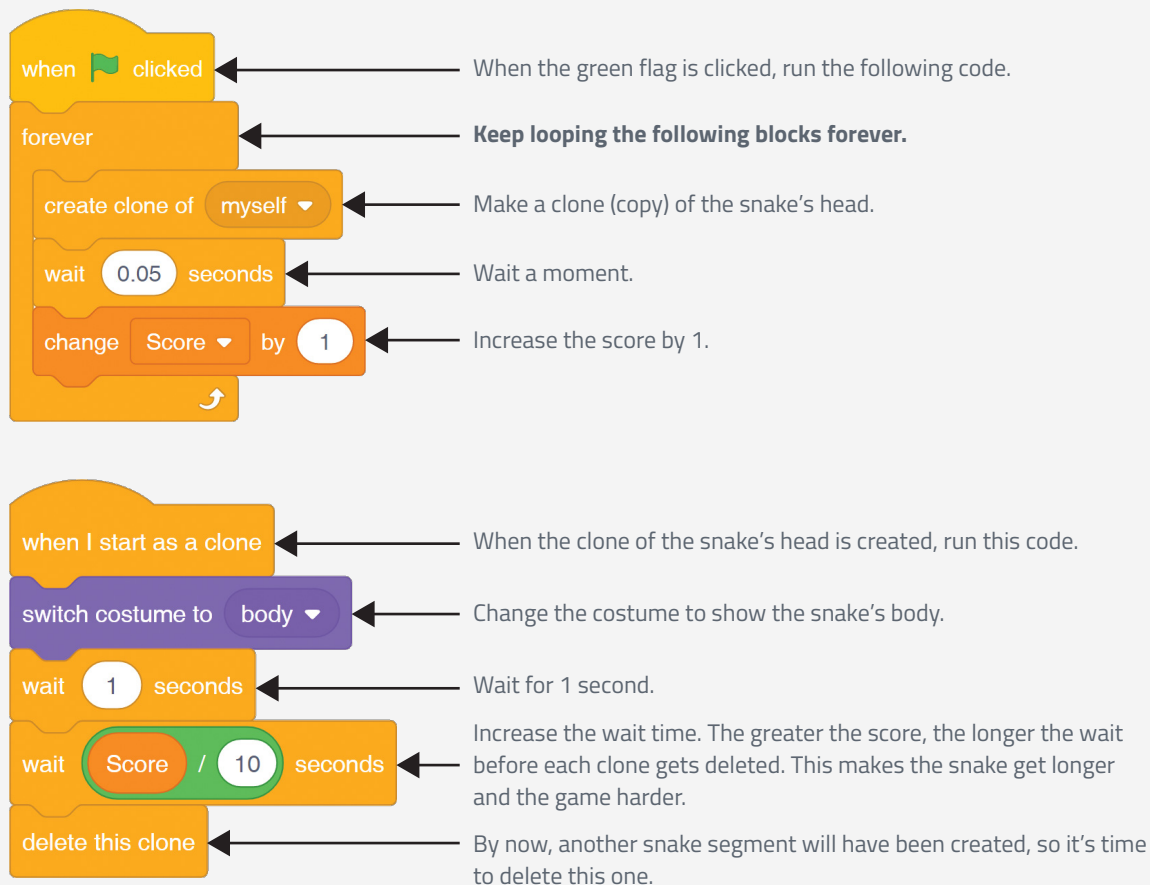
The snake has hit something, so show a message.

stop all


Stop all running code when the game is over.

12. CODE THE BODY


Add these two sections of code to control the body of the snake:




The image shows two sets of Scratch code blocks with arrows pointing to their respective descriptions. The first set is a 'when green flag clicked' event block followed by a 'forever' loop containing three blocks: 'create clone of myself', 'wait 0.05 seconds', and 'change Score by 1'. The second set is a 'when I start as a clone' event block followed by four blocks: 'switch costume to body', 'wait 1 seconds', 'wait Score / 10 seconds', and 'delete this clone'.

when  clicked ← When the green flag is clicked, run the following code.


forever ← **Keep looping the following blocks forever.**

create clone of  ← Make a clone (copy) of the snake's head.


wait seconds ← Wait a moment.

change  by ← Increase the score by 1.

when I start as a clone ← When the clone of the snake's head is created, run this code.

switch costume to  ← Change the costume to show the snake's body.

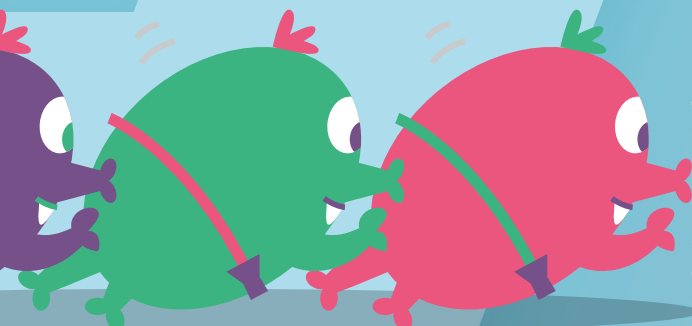
wait seconds ← Wait for 1 second.

wait  seconds ← Increase the wait time. The greater the score, the longer the wait before each clone gets deleted. This makes the snake get longer and the game harder.

delete this clone ← By now, another snake segment will have been created, so it's time to delete this one.



Click the **green flag** to test your code. Use the **left** and **right arrow** keys to steer your snake around the screen. Avoid bumping into anything!



By cloning segments of the snake and deleting them, you made it appear as if the whole snake was moving. To make the snake get longer, you changed the time before each clone was deleted. Ready for a challenge? Turn to page 82 and create your own Hungry Snake game.

CACTUS JUMPER

For this project, you'll make a side-scrolling game. Cacti will scroll across the screen toward the cat. You can press the spacebar to make the cat jump over each cactus. You'll use cloning to create multiple cacti. To make the cat jump, you'll add a variable to store how fast it's jumping. This creates a much more exciting and realistic jump effect.



1. MAKE A SCORE VARIABLE

Click **Variables**.

Click **Make a Variable**.

New Variable

New variable name:

For all sprites For this sprite only

Cancel OK

Type **score**. Click **OK**.

2. ADD ANOTHER VARIABLE

Click **Make a Variable** again.

New Variable

New variable name:

For all sprites For this sprite only

Cancel OK

Type **speed**. Click **OK**.

3. CODE THE CAT

Drag this code into the Coding Area to make it seem like the cat is running in place:

when green flag clicked → Run the following code when the green flag is clicked.

set size to 60 % → Shrink the cat to 60 percent of its normal size.

go to x: -190 y: -70 → Move the cat to the left side of the Stage, near the bottom.


forever → **Keep repeating the following code forever.**

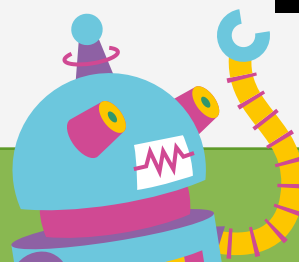
wait 0.1 seconds → Wait for a tenth of a second. (This makes the animation look more realistic.)

if y position = -70 then → **If the cat is touching the ground, then run this code.**

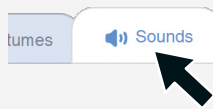
next costume → Show the next costume to animate the cat's legs.

change Score by 1 → Increase the score by 1.

 *This code makes the cat run in place. Click the green flag to test your code!*



4. IMPORT A SOUND EFFECT



Click the **Sounds** tab (top left).



Click the **Choose a Sound** button.



Scroll through the sounds to find the **Boing** icon. Click it to import it.

5. MAKE IT JUMP

Click the **Code** tab and drag in this code to make the cat jump:

When the spacebar is pressed, run this code.

Play the sound effect.

Set jump speed to 15.

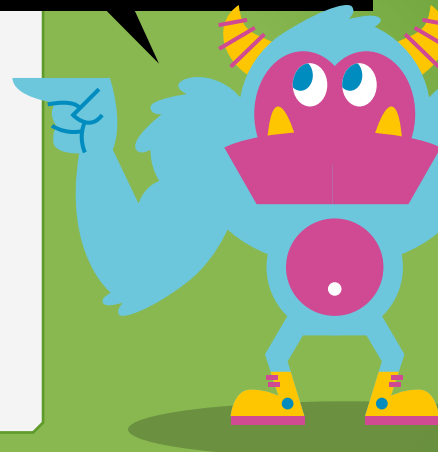
Repeat the following code 29 times.

Decrease the speed variable by 1.

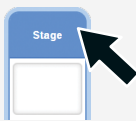
Change the y-coordinate by the value of the speed variable.

Click the **green flag** to test your code.
Press the **spacebar** and watch the cat jump in the air!

You use a speed variable to store the cat's jump speed. The jumping speed starts out quick, with a value of 15. Then you use a loop to reduce the speed until it drops to 0 when the cat briefly stops in midair. As the speed becomes more negative, the cat falls down to the ground faster and faster.



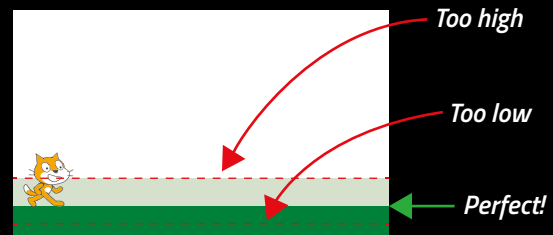
6. SELECT THE BACKDROP



Click the **Stage** icon at the bottom right of the screen.



Click the **Backdrops** tab (top left of the screen).



Make sure the ground is at the same level as the cat's feet.

7. DRAW THE GROUND



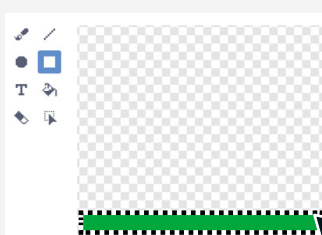
Click the **Convert to Bitmap** button.



Choose the **Rectangle** tool.



Choose a green color.



Draw a thin, wide rectangle at the bottom of the screen to be the ground.



8. DRAW THE SKY



Choose the **Fill** tool.



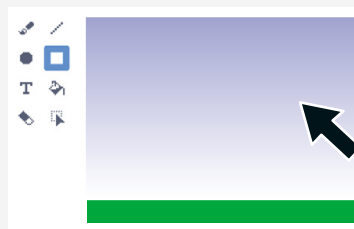
Select the **fill down gradient**.



Set the left color to a light blue.



Set the right color to white.



Click to fill in a gradient color for the sky.

9. ADD A CLOUD



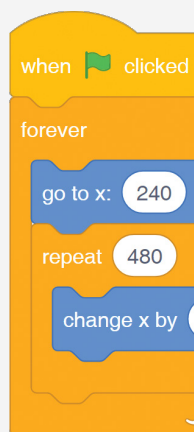
Click the **Choose a Sprite** button.



Scroll through the sprites and select **Cloud**.

10. CODE THE CLOUD

Drag these code blocks into the Coding Area to make the cloud drift slowly across the Stage:



Run this code when the green flag is clicked.

Repeat the following code forever.

Move the cloud to the top right of the Stage.

Repeat this code **480** times to cover the width of the Stage.

Move the cloud to the left slowly.



Click the **green flag** to test your code.

11. ADD A NEW SPRITE



Let your mouse hover over the **Choose a Sprite** button.



Find the **Paint** option and click it.

12. IMPORT A SOUND



Import the sound effect called **Bonk**. (See step 4 for help.)

13. DRAW A CACTUS



Click the **Costumes** tab. Click the **Convert to Bitmap** button.



Choose the **Brush** tool.



Make the brush thick by typing in **30**.



Make the brush thinner and then add any final touches.



14. CODE THE CACTUS

The first section of code creates the cactus clones. Click the **Code** tab and then drag this code into the Coding Area:

when green flag clicked

set Score to 0

set size to 25 %

hide

forever loop:

- wait pick random 1 to 3 seconds
- repeat loop (pick random 1 to 3):
 - create clone of myself
 - wait 0.05 seconds

Run the following code when the green flag is clicked.

Reset the score to 0.

Shrink the cactus to a quarter of its drawn size.

Make it invisible.

Repeat the following code forever.

Wait a random amount of time. This prevents spacing out the cacti equally.

Repeat this loop a couple of times (randomly).

Create a cloned copy of the cactus.

Wait a moment.

Every few seconds, this code will create one to three clones of the cactus. Waiting for 0.05 seconds means they won't be too close together.

The second section of code makes each clone move toward the cat:

when I start as a clone

go to x: 200 y: -90

show

repeat until touching edge ?

- change x by -5
- if touching Cat ? then:
 - start sound Bonk
 - say OW! for 1 seconds
 - stop all

delete this clone

When the cactus is cloned, run the following code.

Move the cloned cactus to the bottom right of the screen.

Make it visible.

Keep repeating this code until it reaches the other side.

Move the cactus to the left.

If the cat bumps into the cactus, then run this code.

Play the sound effect.

The game is over. Show a message.

Stop all other code.

Once the cactus clone has reached the edge, delete it.

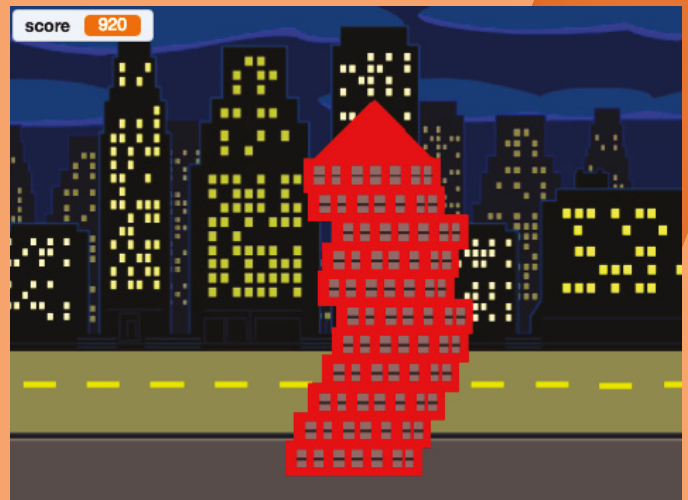
Click the **green flag** to test your code.



You've built your first side-scrolling game! Ready for a challenge? Turn to page 82 and try the T-Rex Jumper.

TOWER BUILDER

In this game, the players have to build the straightest and tallest tower they can. Each floor of the tower moves back and forth across the screen. Pressing the spacebar makes the floor drop down until it hits the rest of the tower. If the tower reaches a certain level, the player must place a roof on top. Although the rest of the code is fairly straightforward, some complex math is used to calculate how accurately the player drops each floor. The more accurate the drop, the higher the score given.



1. NO CATS!

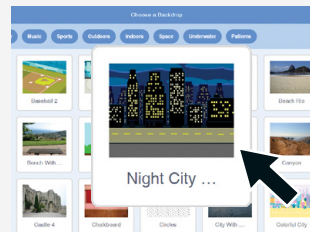


Start by deleting the cat sprite.

2. NIGHT CITY

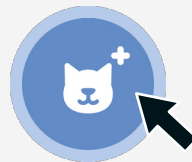


Click the **Choose a Backdrop** button.

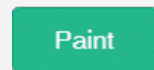


Find **Night City with Street** and then click it to set the backdrop.

3. ADD THE PLAYER SPRITE



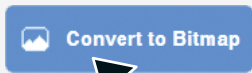
Let your mouse hover over the **Choose a Sprite** button.



Move up to the **Paint** option and click it.



4. DRAW A FLOOR OF THE TOWER



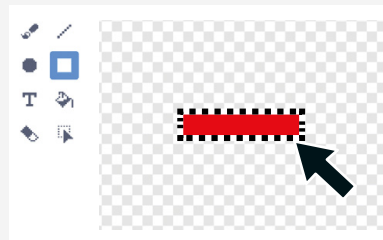
Click the **Convert to Bitmap** button.



Choose the **Rectangle** tool.



Choose red.



Draw a red rectangle in the center. It should be about half the width of the Drawing Area.



Click out of the red rectangle first. Choose a light gray.



Draw a window.



Choose black.



Choose the **Line** tool. Choose the thickness you want.



Add details to the window.

5. COPY THE WINDOW



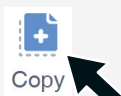
Click the **Select** tool.



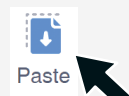
Draw a dotted line around the window to select it.



Blue lines should appear around the window.



Click the **Copy** button.

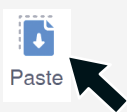


Click **Paste**.



A copy of your window should appear. Drag it into place next to the first one.

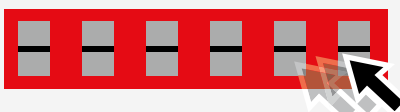
6. KEEP PASTING WINDOWS



Click **Paste**.

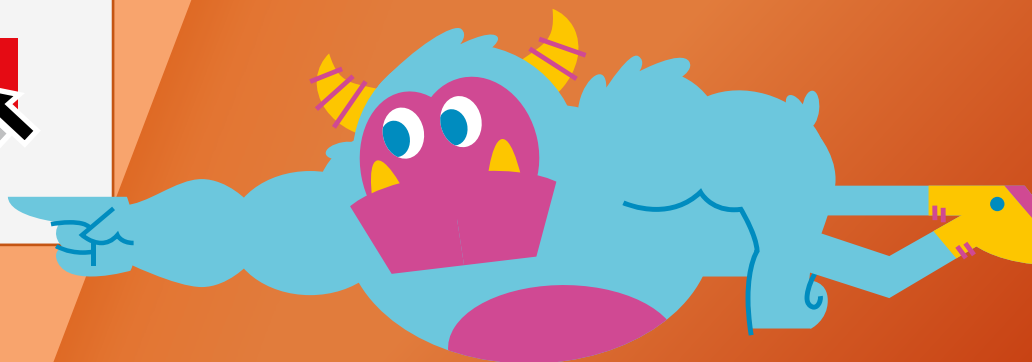


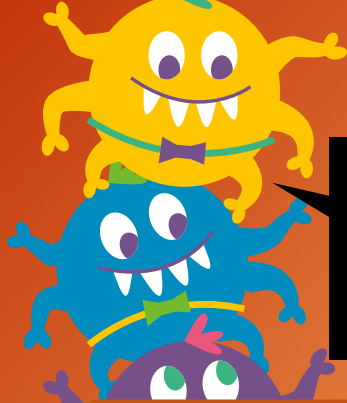
Drag the new window into place.



Keep pasting windows until you've filled up the entire floor.

Using Copy and Paste is a good way to reproduce parts of your sprite images. It will make your games look more professional.





You need to make a special top floor for the tower. You'll make another costume for this. First, set a name for the current costume.

7. NAME THE COSTUME

Costume **Floor** — Type **floor** as the name for this first costume. Naming your costume makes it easier to organize your code.

Fill Copy Paste

8. DUPLICATE THE WHOLE FLOOR

Right-click **Floor** icon in the Sprite List. Click **duplicate** to create another copy of the floor.

A new costume should appear.

Costume **Floor2** Fill Filled Outlined

Costume **Roof** Fill Filled Outlined

Rename the new costume **Roof**.

9. DRAW THE ROOF

Choose the **Line** tool.

Draw two lines to outline the top of the tower.

Select the **Fill** tool.

Click to fill in the roof.

10. MAKE A SCORE VARIABLE

Click the **Code** tab.

Click the **Variables** button.

Click **Make a Variable**.

New Variable dialog: New variable name: **score** For all sprites For this sprite only Cancel OK

Type **score**. Click **OK**.

11. MAKE ANOTHER VARIABLE

Click **Make a Variable**.

New Variable dialog: New variable name: **tower x** For all sprites For this sprite only Cancel OK

Type **tower x**. Click **OK**.

12. IMPORT A SOUND EFFECT

Click the **Sounds** tab (top left).

Click the **Choose a Sound** button.

Scroll through the sounds to find the **Dance Celebrate** icon. Click it to import that sound.

13. THE COMPLETE CODE

Drag this code into the Coding Area:

when green flag clicked ← When the green flag is clicked, run the following code.

set size to 50 % ← Shrink the sprite to half its size.

switch costume to Floor ← Show the normal floor costume.

set Score to 0 ← Reset the score to 0.

set Tower X to 0 ← Set the tower x variable to 0.

erase all ← Clear the background image.

set rotation style don't rotate ← Stop the floor sprite from rotating.

go to x: 0 y: -140 ← Move the sprite down to the bottom center of the screen.

stamp ← Draw a copy of the floor sprite on the screen.

repeat 10 ← **Keep repeating the following code 10 times.**

go to x: 0 y: 130 ← Start the floor sprite at the top center of the screen.

repeat until key space pressed? ← **Repeat this code until the spacebar is pressed.**

move 10 steps ← Move the sprite 10 steps in its current direction.

if on edge, bounce ← If it hits the edge, then bounce back the other way.

repeat until touching color red ? ← **Keep repeating the next block until it hits something.**

change y by -10 ← Move the floor down 10 steps.

stamp ← Draw a copy of the floor in the place it hits another one.

change Score by 100 - abs of Tower X - x position ← Increase the score according to how closely stacked the floor sprite is to the previous floor.

set Tower X to x position ← Store the current position of the floor sprite.

start sound Dance Celebrate ← Play a celebratory tune.

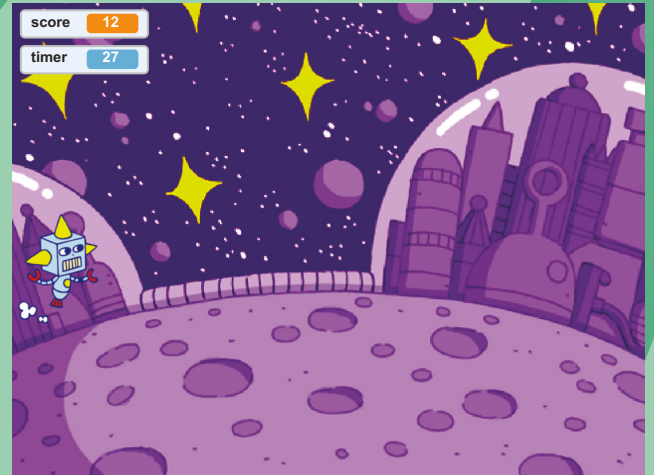
switch costume to Roof ← Change the final sprite to be the roof of the tower.

Click the **green flag** to play the game.

Ready for a complex challenge? Turn to page 83 and try making the Super Tower game.

ROBOT POWER UP

In this next game, you'll program a robot that floats around in space. The goal of this game is to control the robot by using the spacebar to collect as many fuel cells as you can! You'll use cloning to create multiple targets to collect and a timer to stop the game after 30 seconds. You'll use math to create a background sound effect that changes as the robot moves up and down.



1. NO CATS!

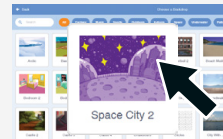


Space is no place for a cat. Delete it.

2. SPACE CITY



Click the **Choose a Backdrop** button.

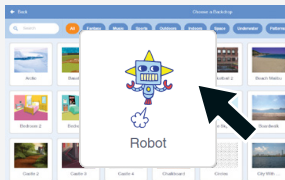


Scroll to find **Space City 2** and then click it to set the backdrop.

3. ADD A ROBOT



Click the **Choose a Sprite** button.

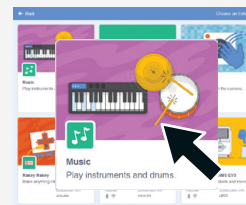


Scroll through the sprites and click **Robot**.

4. ADD AN EXTENSION



Click the **Add Extension** button (bottom left of the screen).

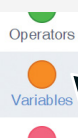


Click the **Music Extension**.

The Music Extension adds extra code blocks to make and control sounds.



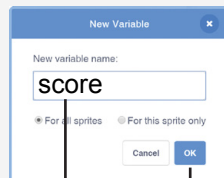
5. MAKE A VARIABLE



Click the **Variables** button.

Make a Variable

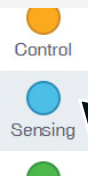
Click the **Make a Variable** button.



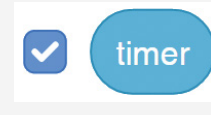
Type in **score**.

Click **OK**.

6. SHOW THE TIMER

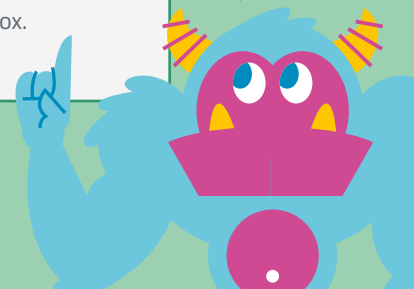


Click the **Sensing** button.



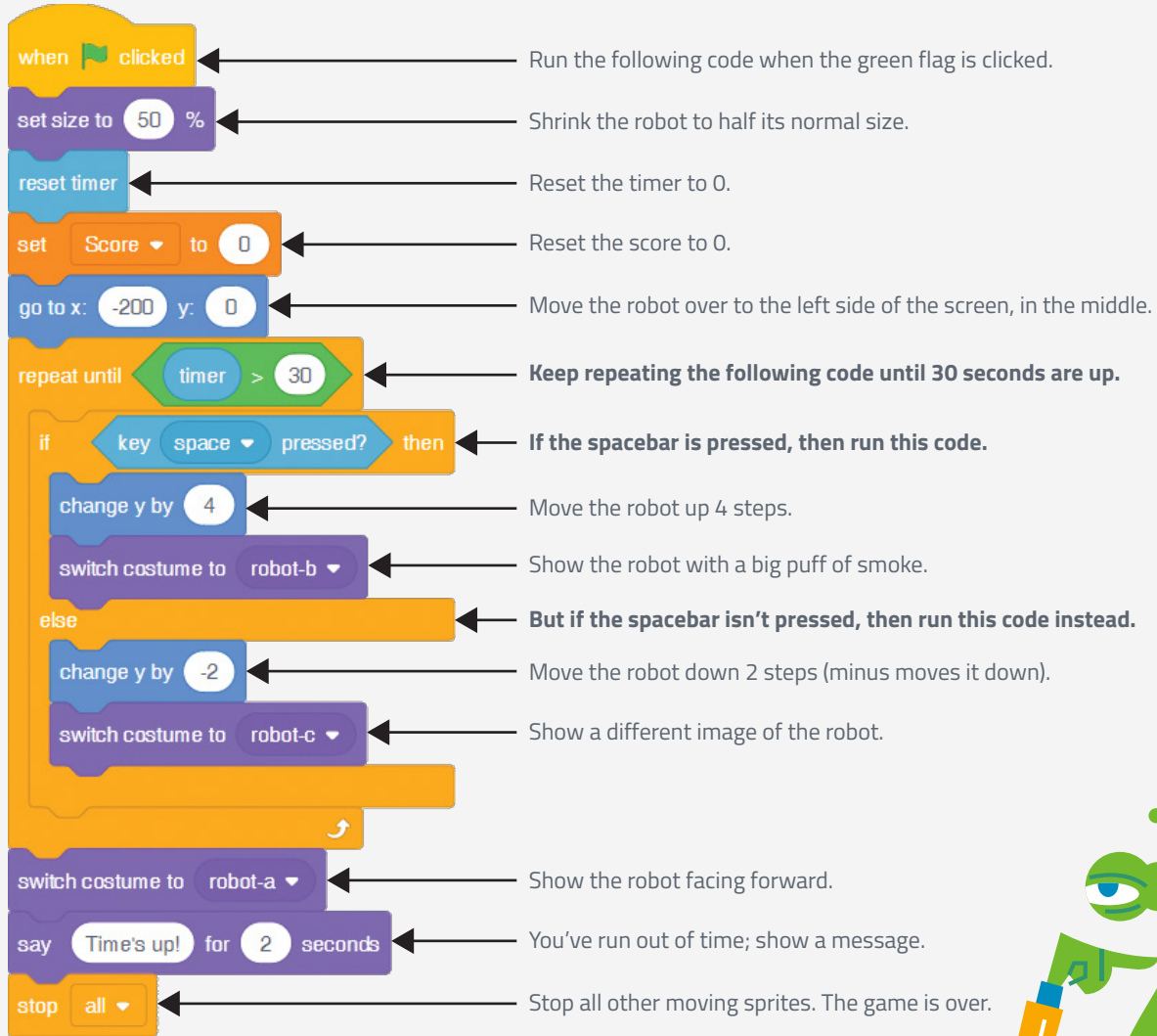
Find the **timer** code block and click the checkbox.


The timer shows how long the game has been running.



7. CODE THE ROBOT

Control the robot by dragging this code into the Coding Area:




when  clicked ← Run the following code when the green flag is clicked.


set size to % ← Shrink the robot to half its normal size.

reset timer ← Reset the timer to 0.

set to ← Reset the score to 0.

go to x: y: ← Move the robot over to the left side of the screen, in the middle.

repeat until  ← **Keep repeating the following code until 30 seconds are up.**

if  then ← **If the spacebar is pressed, then run this code.**

change y by ← Move the robot up 4 steps.

switch costume to ← Show the robot with a big puff of smoke.

else ← **But if the spacebar isn't pressed, then run this code instead.**

change y by ← Move the robot down 2 steps (minus moves it down).

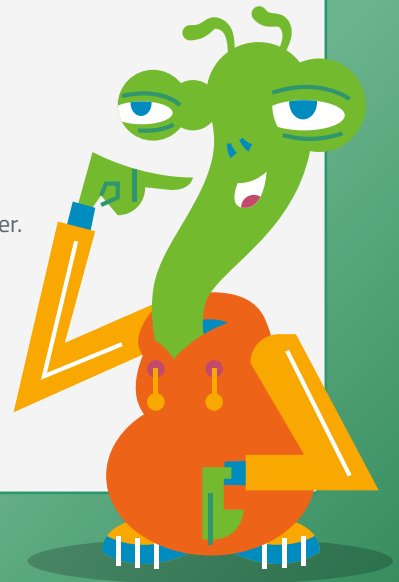
switch costume to ← Show a different image of the robot.

switch costume to ← Show the robot facing forward.

say for seconds ← You've run out of time; show a message.

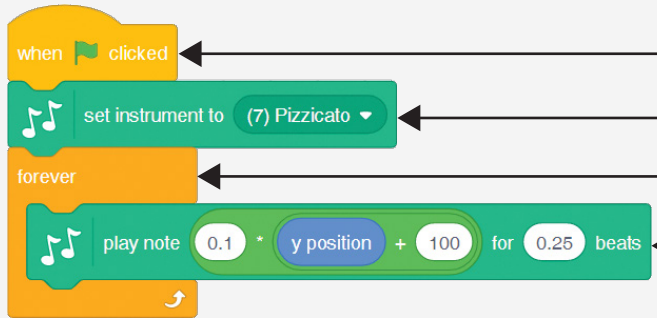
stop ← Stop all other moving sprites. The game is over.

While the game is running, you'll play a sound effect. To avoid delaying the main loop, you'll add this sound effect as a separate section of code.



8. MAKE SOME NOISE

Drag this code into the scripts area.



The code consists of three main blocks: a 'when green flag clicked' block, a 'set instrument to (7) Pizzicato' block, and a 'forever' loop containing a 'play note' block. The 'play note' block is configured with a duration of 0.1, a pitch of 'y position + 100', and a duration of 0.25 beats.

Also run this code when the green flag is clicked.

Set the sound effect to that of a violin being plucked.

Keep repeating the following block of code forever.

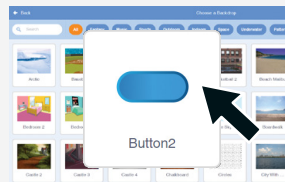
Play a short note. Make the pitch higher if the robot's y position is higher on the Stage.

Click the **green flag** to test your code. The robot should slowly move down. Pressing the spacebar will move it up. You should hear a sound effect that changes its pitch as the robot moves.

9. ADD A FUEL CELL SPRITE



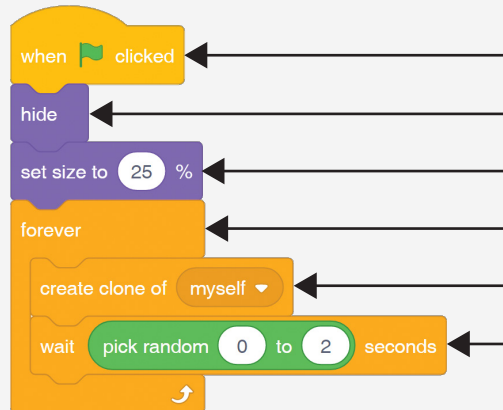
Click the **Choose a Sprite** button.



Scroll through the sprites and click **Button2**.

10. CODE THE FUEL CELL

Control the fuel cell by dragging this code into the Coding Area:



The code consists of several blocks: a 'when green flag clicked' block, a 'hide' block, a 'set size to 25%' block, a 'forever' loop containing a 'create clone of myself' block and a 'wait pick random 0 to 2 seconds' block.

Run the following code when the green flag is clicked.

Hide the original fuel cell since you'll be making a copy of it.

Shrink it down to 25 percent of its original size.

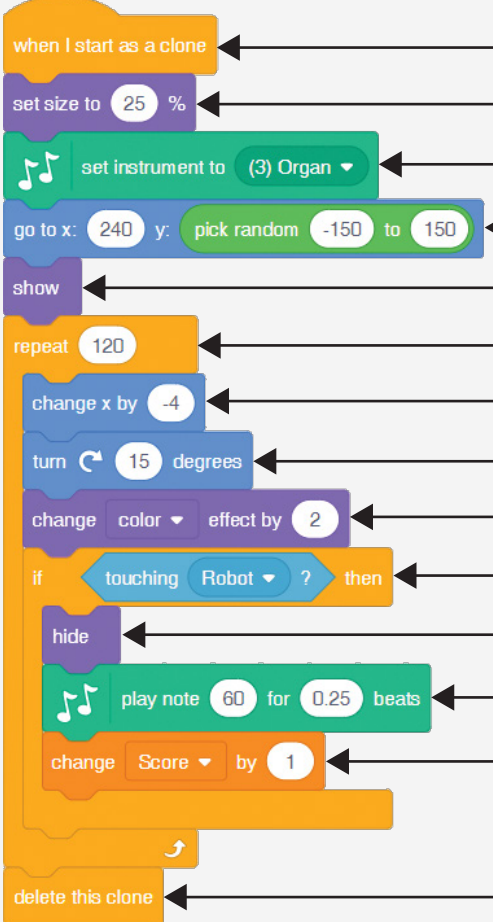
Keep repeating the following code forever.

Clone a copy of the fuel cell.

Wait a moment or two.

11. CODE THE FUEL CELL

Add these blocks to code the clones as they are created:



When I start as a clone

set size to 25 %

set instrument to (3) Organ

go to x: 240 y: pick random -150 to 150

show

repeat 120

change x by -4

turn 15 degrees

change color effect by 2

if touching Robot ? then

hide

play note 60 for 0.25 beats

change Score by 1

delete this clone

When the clone is created, run this code:

Shrink it down to 25 percent of its original size.

Select the organ sound effect.

Start the clone on the right side in a random place.

Make sure the clone is visible.

Repeat the following code 120 times to move the clone all the way across the Stage.

Move the clone to the left 4 steps (*minus 4*).

Make the clone spin around slowly.

Make it gradually change color.

If the robot catches the clone, run this.

Hide the clone—it has been caught!

Play a sound effect to let players know they caught a fuel cell.

Increase the score by 1.

Delete the clone—it's reached the edge of the Stage or has been caught.

 Click the **green flag** to test your code!



You've already used most of the techniques in this game before. The timer lets you check how long the game has been running. You use the position of the robot to determine the pitch of the sound effect. When you're ready for a challenge, use these ideas to create your own collecting game on page 83.

INVASION OF THE DONUTS

A spaceship is under attack—from donuts! By rotating left and right, you can take aim at the donuts. Fire at them before they hit the ship. You'll clone each donut so that several can appear at the same time. You'll use a color effect to make them pulse as they move toward the ship. Variables store the score and the player's number of lives. Each time a donut hits the ship, the number of lives decreases.



1. NO CATS!

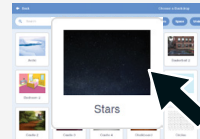


Cats can't eat donuts. Let's delete the cat.

2. STARS



Click the **Choose a Backdrop** button.

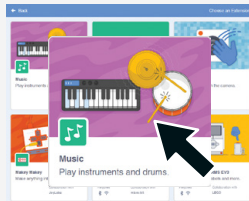


Scroll down to find **Stars** and then click it to set the backdrop

3. ADD AN EXTENSION

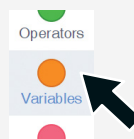


Click the **Add Extension** button.



Click **Music Extension**.

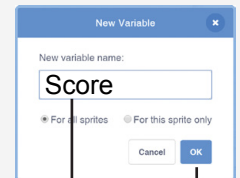
4. MAKE A SCORE VARIABLE



Click the **Variables** button.

Make a Variable

Click the **Make a Variable** button.

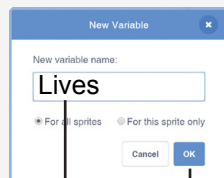


Type in **Score**. Click **OK**.

5. ADD ANOTHER VARIABLE

Make a Variable

Click the **Make a Variable** button.

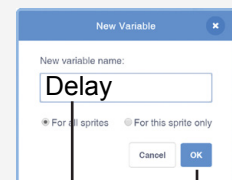


Type in **Lives**. Click **OK**.

6. ADD ANOTHER VARIABLE

Make a Variable

Click the **Make a Variable** button.



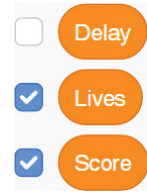
Type in **Delay**. Click **OK**.



You'll use the delay variable to set how often a new donut appears onscreen. To watch the variable in action, select the checkbox again later!

7. HIDE OR SHOW

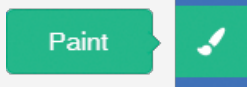
The player needs to see the score and the number of lives left. You don't need the player to see the variable called **Delay**, so unselect it.



8. ADD THE SHIP SPRITE



Let your mouse hover over the **Choose a Sprite** button.



Move up to the **Paint** option and click it.

If you make a mistake drawing, click the **Undo** button and try again!



9. DRAW THE SHIP



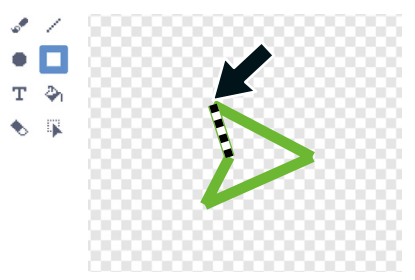
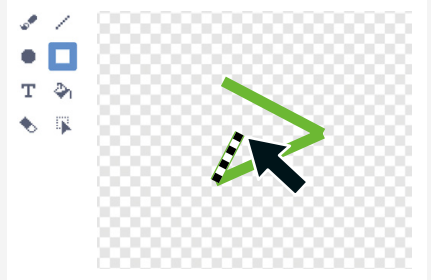
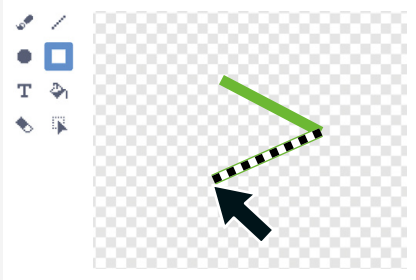
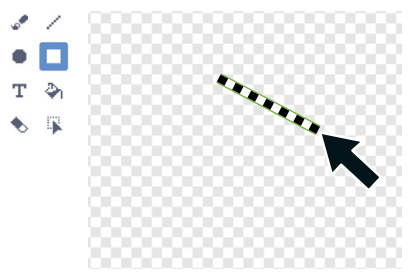
Click the **Convert to Bitmap** button.



Choose the **Line** tool.



Choose a green color.



Click the **Fill** tool.



Fill in the spaceship.

For now, draw a simple spaceship. You can add more details once you have all the code working.



10. START CODING

Click the **Code** tab at the top left of the screen and then drag this code into the Coding Area:

The code consists of the following blocks:

- when green flag clicked** (yellow block)
- set Lives to 3** (orange block)
- set Score to 0** (orange block)
- set size to 25 %** (purple block)
- forever** loop (yellow block) containing:
 - if key left arrow pressed? then** (blue block) containing **turn 5 degrees** (blue block)
 - if key right arrow pressed? then** (blue block) containing **turn 5 degrees** (blue block)

Run the following code when the green flag is clicked.

Set the number of lives at the start of the game to 3.

Reset the score to 0.

Shrink the spaceship to a quarter of its normal size.

Keep repeating this code forever.

If the *left arrow* key is pressed, then run this code.

Rotate the spaceship *counterclockwise*.

If the *right arrow* key is pressed, then run this code.

Rotate the spaceship *clockwise*.

Rename your sprite to make it easier to organize your code.

11. RENAME YOUR SPRITE

Sprite **Ship**

Click inside the **Sprite** name box. Type in **Ship** and press **Enter**.

12. ADD THE LASER SPRITE



Click the **Choose a Sprite** button.



Scroll through the sprites and click **Ball**. (This will be the laser.)

13. START CODING THE LASER

Drag this code into the Coding Area:

when clicked ← Run this when the green flag is clicked.

set instrument to (19) Marimba ← Choose the instrument for the sound effect.

hide ← Make the ball invisible.

set size to 20 % ← Shrink it to 20 percent of its size.

when space key pressed ← Run this code when the spacebar is pressed.

create clone of myself ← Create a cloned copy of the ball.

play note 24 for 0.1 beats ← Play a bleep sound.

when I start as a clone ← Once the cloned ball is created, run the following code.

go to Ship ← Move the clone to the same place as the spaceship.

show ← Make sure the clone is visible.

point in direction direction of Ship ← Point the ball in the same direction as the spaceship.

repeat until touching edge ? ← **Keep repeating the following code until the ball clone reaches the edge of the screen.**

move 10 steps ← Move it forward 10 steps.

delete this clone ← Once the ball clone hits the edge of the Stage, delete it.

Start with a point in direction block. Drag the following sensing block inside it.

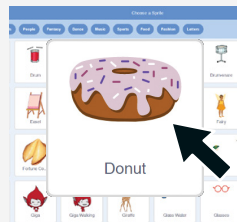
backdrop # of Stage

Change Stage to Ship. Next select direction from the left drop-down.

14. ADD THE DONUT SPRITE



Click the **Choose a Sprite** button.



Scroll through the sprites and click **Donut**.

15. START CODING THE DONUT

Now you need to clone the donut and control it. Drag the following code into the Coding Area:

when green flag clicked

set instrument to (19) Marimba

hide

set size to 50 %

set Delay to 4

forever

create clone of myself

wait pick random 0.5 to Delay seconds

if Delay > 0.5 then

change Delay by -0.1

Run the following code when the green flag is clicked.

Choose the instrument to play the sound effects.

Make the original donut invisible.

Shrink it to half its size.

Start the delay value at 4.

Keep repeating the following code forever.

Make a cloned copy of the donut.

Wait a short, random amount of time.

If the delay is more than half a second, then run this code.

Reduce the delay by 0.1 seconds. (These code blocks will keep reducing the delay before a new donut appearing until it gets to 0.5 seconds.)

when I start as a clone

go to x: 0 y: 0

point in direction pick random 0 to 360

move 400 steps

show

point towards Ship

turn pick random -15 to 15 degrees

repeat 700

move 1 steps

change color effect by 5

if touching Ball ? then

play note 36 for 0.1 beats

change Score by 1

delete this clone

if touching Ship ? then

broadcast hitShip

delete this clone

When each donut clone is created, run this code.

Move the donut to the middle of the screen.

Point it in a random direction.

Move the donut quickly to the very edge of the Stage.

Make the donut visible.

Now turn the donut and make it point to the spaceship.

Turn the donut a random number of degrees to make it hard for players to predict where it will go.

Keep repeating the following code 700 times.

Move the donut 1 step forward.

Make the donut change color a little.

If a laser hits the donut, then run this code.

Play a short sound effect.

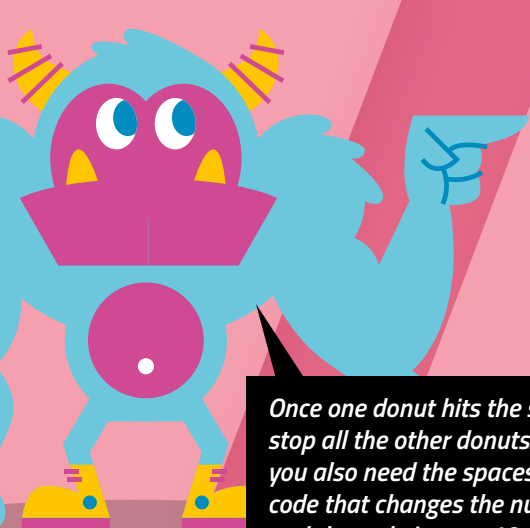
Increase the score by 1.

Delete the donut clone—it has been destroyed!

If the donut hits the spaceship, then run this code.

Tell the other sprites that the ship has been hit. (See page 59 for help with creating a new message.)

Delete the donut clone—it has reached the edge of the screen.



16. MORE DONUT CODE

```

when I receive hitShip
  delete this clone
  
```

When any donut receives a broadcasted message, run this code.
Delete the donut clone that receives the message.

Once one donut hits the ship, you need to stop all the other donuts. At that moment, you also need the spaceship to run some code that changes the number of lives and the code in step 19. You'll do this by using a technique called broadcasting to send a message to the other sprites.

17. SELECT THE SPACESHIP

To add more code to the spaceship, click it in the Sprite List.

18. IMPORT A GAME OVER SOUND EFFECT

Click the **Sounds** tab (top left of the screen).
Click the **Choose a Sound** button.
Scroll through the sounds to find the **Boom Cloud** icon. Click it to import that sound.

19. ADD CODE TO HANDLE THE BROADCAST MESSAGE

Add some code to decide what happens when the spaceship is hit. Drag the following code into the Coding Area:

```

when I receive hitShip
  change Lives by -1
  start sound Boom Cloud
  if Lives = 0 then
    say Game over! for 4 seconds
    stop all
  repeat 80 times
    change color effect by 25
  
```

Run the following code when this message is received.
Reduce the number of lives by 1.
Play the Game Over sound effect.
If the player runs out of lives, then run this code.
Show a message.
Stop all the donuts and any other code.
Keep repeating this code 80 times. This will make the spaceship flash for a few seconds and then go back to green.
Make the spaceship change color.
Click the **green flag** to play the game. How long can you survive without hitting the deadly donuts?

Ready for an even more complex challenge? Turn to page 83 and try making the Killer Burger Invasion game.

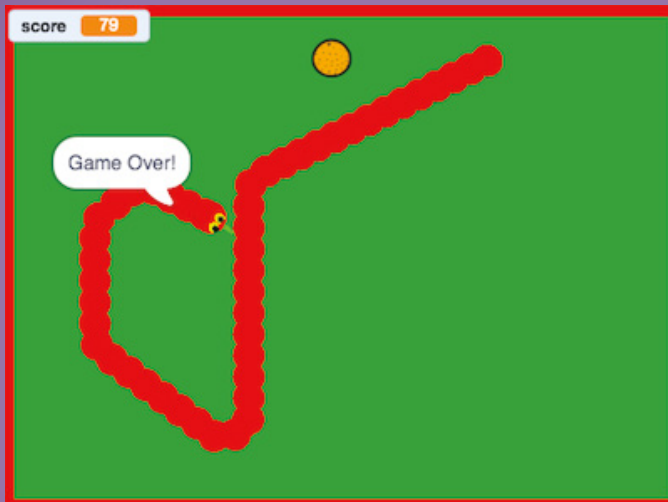
CHAPTER 4

CHALLENGES

In this chapter, you used more complex math to make the programs more engaging. Think about how you can use math to make your own games harder or to add new challenges, costumes, or special sound effects.

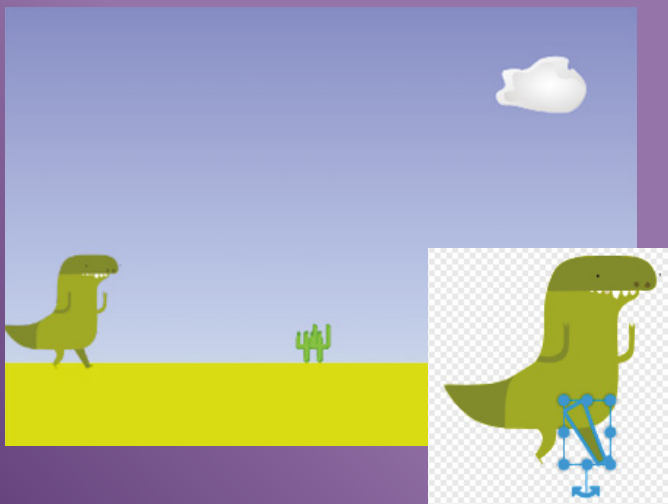


After coding the programs in this chapter, try these challenges.



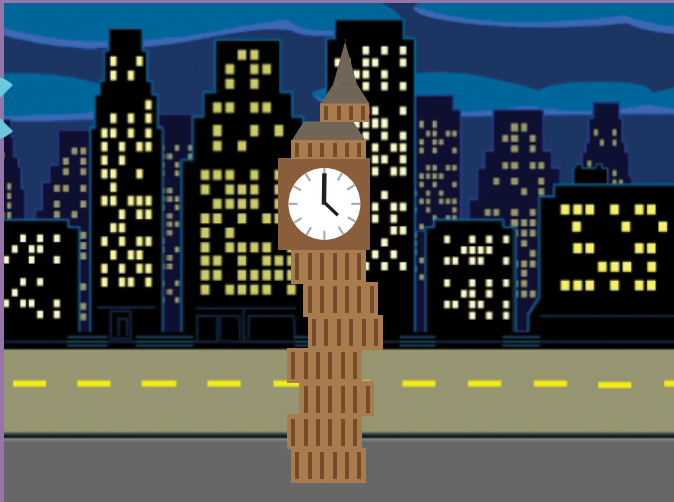
CHALLENGE 1 HUNGRY SNAKE

Build a version of the snake game in which the snake has to eat pieces of food on the screen. These should appear in random places. When the snake eats one piece of food, increase the score by 10 points and then move the food to a new place. You could make multiple costumes for the food sprite. (See the Fruit Catcher game on page 44 for help.)



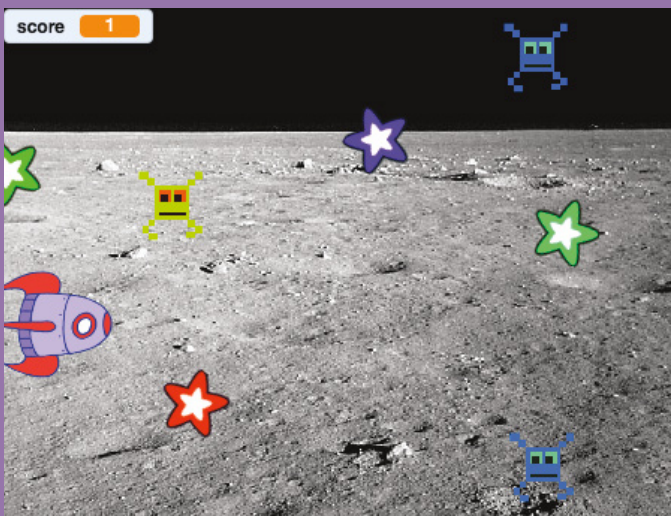
CHALLENGE 2 T-REX JUMPER

Use the ideas from the Cactus Jumper game but with a T-Rex as the main sprite. Delete some of the T-Rex costumes and then duplicate one of them. Edit the graphics to make the leg look like the one shown here. Make the game harder by doubling the speed of the cacti's movement. Make the cacti get bigger as the game goes on. (See Colored Walls on page 52 for ideas on how to increase size as a game progresses.)



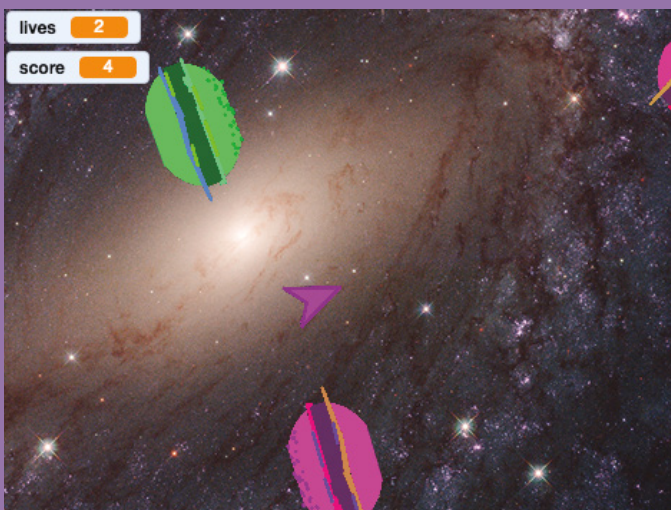
CHALLENGE 3 SUPER TOWER

Improve the tower game on page 68 by designing a more elaborate tower. Use a real-life monument or skyscraper as your inspiration. Add extra levels to make the tower look different. Create a variable called *floor level*. Reset the variable to 1 at the start and increase it every time a floor is added. Add an if statement that checks whether the floor level is greater than 6 or 7. If it is, switch to the next costume.



CHALLENGE 4 COLLECTING GAME

Look at the code in Robot Power Up and design your own collecting game. Choose your own backdrop and sprites. How many points will you get for collecting each sprite? Why not add some alien sprites? Hitting one of the aliens should make your score go down. Do this by changing the score by a minus number. (See the code on page 72 for help.)



CHALLENGE 5 KILLER BURGER INVASION

Use the code from the Invasion of the Donuts game on page 76 to help you create your own invasion game—killer burgers, maybe? How many lives will you have, and how many points for each burger destroyed?

MAKING YOUR OWN BLOCKS

As your programs get bigger and more complex, problems may occur. It can get harder to debug or fix them. If you come back to a program after a few weeks, remembering how it works and which part does what can be tricky. In this chapter, you'll make your program easier to read by splitting it into separate parts. To do this in Scratch, you need to make your own blocks called *functions*.

CREATING A FUNCTION

To create your own function block in Scratch, follow these steps:

Click **My Blocks**.

Click **Make a Block**.

Type in the name of your function. Make sure to choose a name that describes what this part of your program will do.

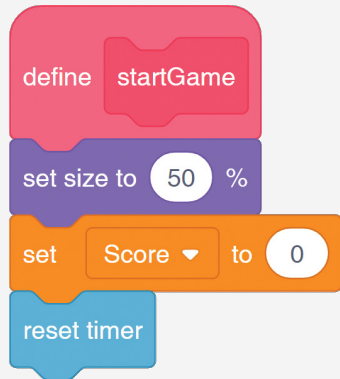
Click **OK**.

A new block with the name of your function should show up in the My Blocks group.

Some programming languages call functions procedures or methods.

DEFINING THE FUNCTION

Follow these steps to tell Scratch what your function block will do:

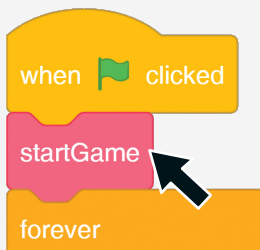


Drag in any blocks you want to run when the function is used.

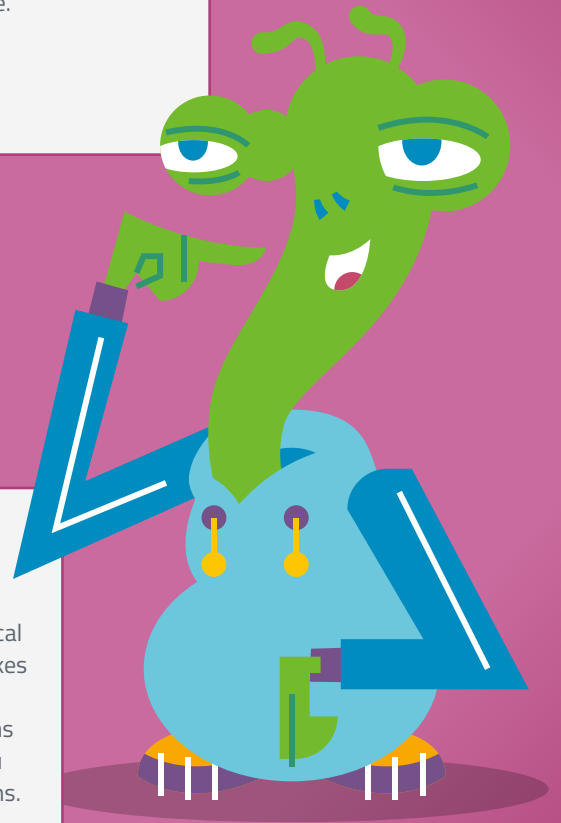
Here, you'll use the startGame function when the game starts. That means you need to put in any code you want to run at the start of the game, like resetting the score.

USING THE FUNCTION

To use the function you created, just drag it into your code.

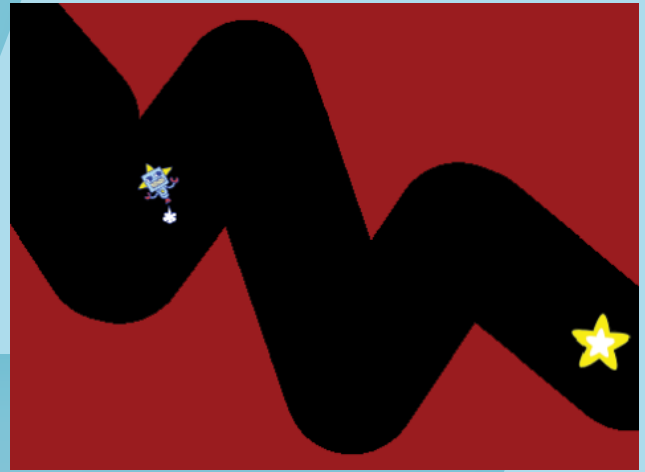


Defining your own blocks and breaking your program into logical parts has lots of benefits. It makes your code much clearer to read. You don't have to repeat sections of code over and over again. You can reuse code in other programs. You can adapt functions to do different things.



CAVE OF STARS

You'll create this game in a slightly different way. By making your own blocks, you'll make the program simpler to understand. Making your own blocks will also allow you to reuse some of the code without having to build it twice. In the game you'll guide a robot through a cave toward a star. The mouse pointer will be used to steer which way the robot goes.



1. NO CATS!



Delete the cat.

2. SELECT THE BACKDROP



In the sprites pane, click the **Stage** icon. (It's at the bottom right.)



Click the **Backdrops** tab.

3. START DRAWING



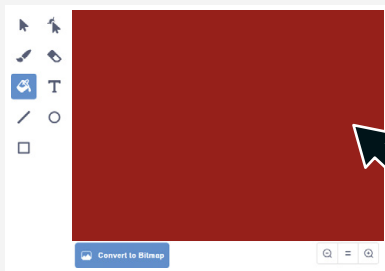
Click the **Convert to Bitmap** button.



Click the **Fill** tool.



Set the color to dark red.



Fill the Drawing Area with dark red.

Turn to page 16 for help with setting colors.



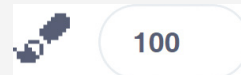
4. SET THE BRUSH STYLE



Select the **Brush** tool.



Choose black.



Make the brush extremely thick by typing in **100**.



If you make a mistake drawing, click the Undo button and try again!

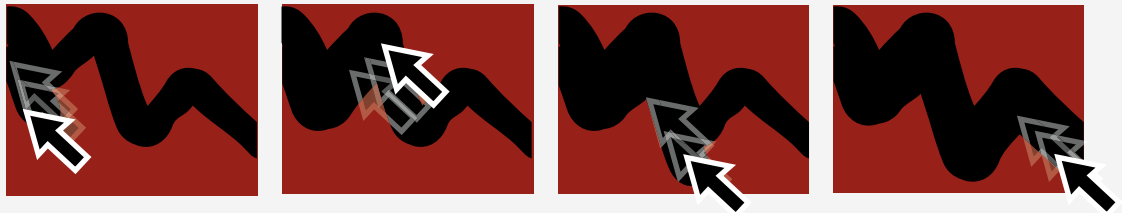


5. DRAW A PATH

Use the **Brush** tool to draw a path through the cave.



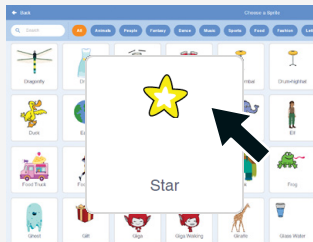
Draw a second line below the first to make the path thicker.



6. ADD A STAR



Click the **Choose a Sprite** button.



Scroll through the sprites and click **Star**.

7. ADD A ROBOT



Click the **Choose a Sprite** button again.



Scroll through the sprites and click **Robot**.

8. MAKE A BLOCK



Variables

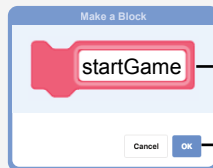


My Blocks

Click **My Blocks**.

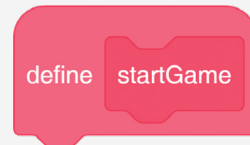
Make a Block

Click **Make a Block**.



Type **startGame**.

Click **OK**.



A pink block called **startGame** should appear in the Coding Area.



You'll run this code when the game starts. This makes sure the robot is in the correct place.

9. ADD CODE TO DEFINE THE BLOCK

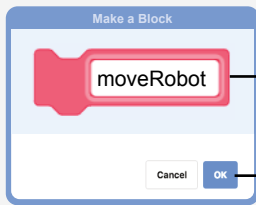
Drag in the following blocks to define the startGame code block:

Run this code when the startGame block is used.

Move the robot to the top left of the Stage.

Shrink the robot to a quarter of its size.

10. MAKE AND DEFINE ANOTHER



Click **Make a Block** and type in **moveRobot** as a function name.

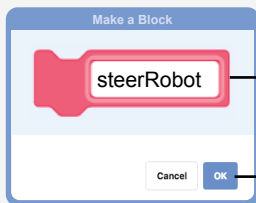
Click **OK**.

Drag in this code:

Run this when the moveRobot block is used.

Move the robot 2 steps forward.

11. MAKE AND DEFINE A CODE BLOCK TO STEER THE ROBOT



Click the **Make a Block** button and type in **steerRobot** as a function name.

Click **OK**.

Drag in this code to define the block:

Run this when the steerRobot block is used.

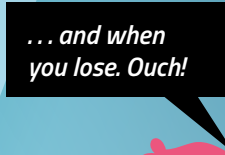
Point the robot toward the mouse pointer.



So far, you've defined blocks to start the game and to move and steer the robot.

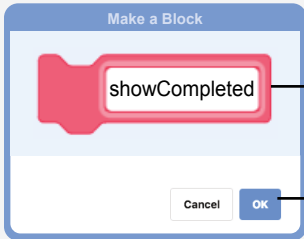


Now you need to define a block for when you win!



... and when you lose. Ouch!

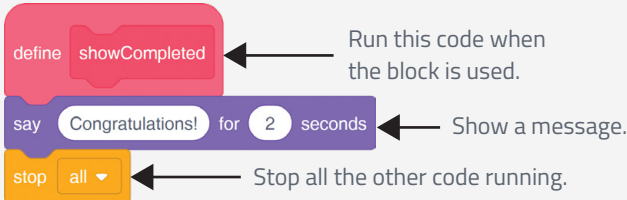
12. WHAT HAPPENS WHEN YOU WIN



Click **Make a Block** and type in **showCompleted** as a function name.

Click **OK**.

Drag in this code:

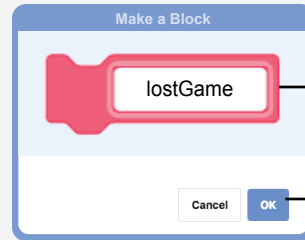


Run this code when the block is used.

Show a message.

Stop all the other code running.

13. WHAT HAPPENS WHEN YOU LOSE



Click the **Make a Block** button and type in **lostGame** as a function name.

Click **OK**.

Drag in this code:

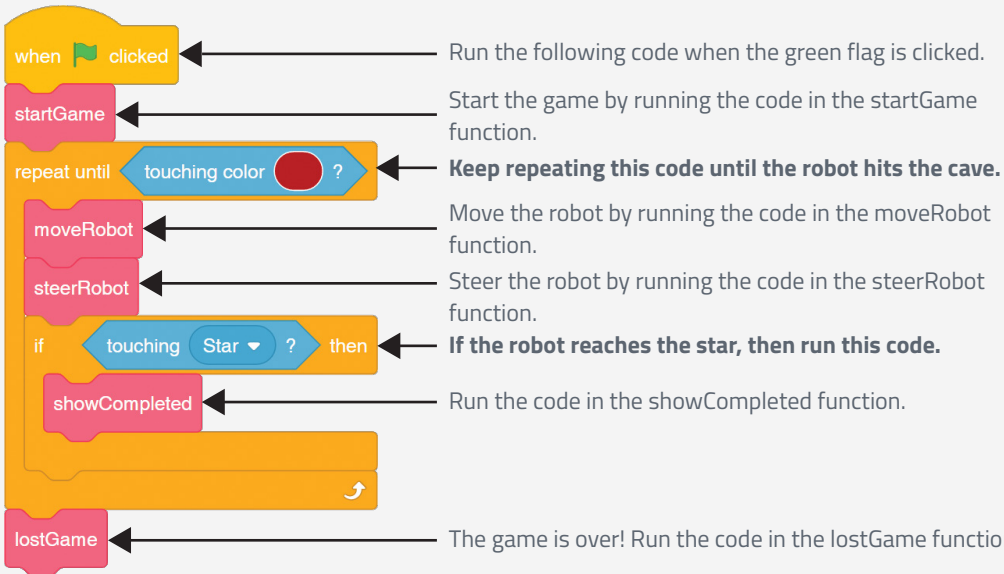


Run this when the lostGame block is used.

Show a message.

14. BRING ALL THE CODE TOGETHER

Now that you've defined the blocks you need, let's use them! You'll combine them with other simple control and sensing blocks to complete the code. Drag in these blocks:



Run the following code when the green flag is clicked.

Start the game by running the code in the startGame function.

Keep repeating this code until the robot hits the cave.

Move the robot by running the code in the moveRobot function.

Steer the robot by running the code in the steerRobot function.

If the robot reaches the star, then run this code.

Run the code in the showCompleted function.

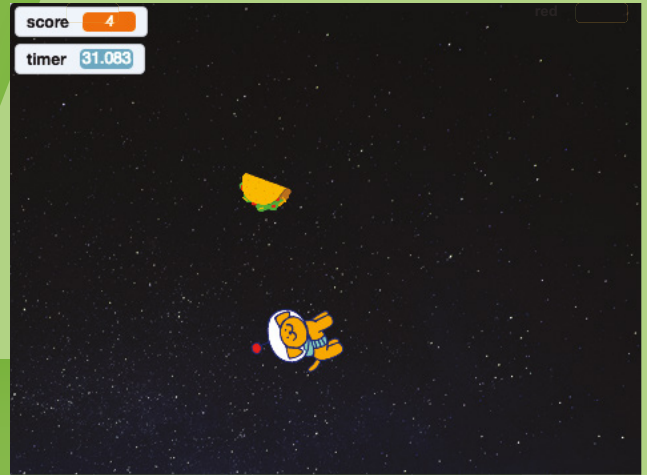
The game is over! Run the code in the lostGame function.

Click the **green flag** to test your code. Try guiding the robot to the star.

Ready for a challenge? Turn to page 112 and try to build the Going Underground game.

SPACE TACOS

This game is similar to some of the simpler games you made in Chapter 3, like Shark Diver or Fruit Catcher. This time, you'll create your own blocks to make the game simpler to understand and adapt. In this game the player has to guide the dog toward the taco using the mouse. There is a limit of 30 seconds to eat as many as possible!



1. NO CATS!

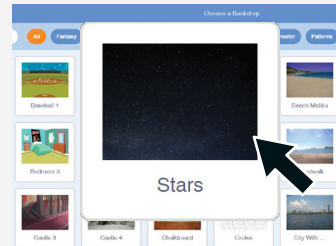


Start by deleting the cat sprite.

2. SET IN SPACE



Click the **Choose a Backdrop** button.

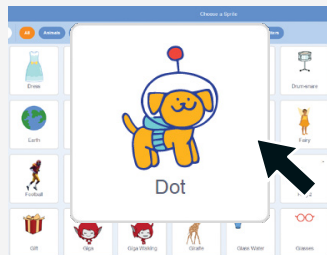


Find **Stars** and set it as the backdrop.

3. ADD A SPRITE

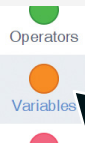


Click the **Choose a Sprite** button.



Scroll through the sprites and select **Dot** the dog.

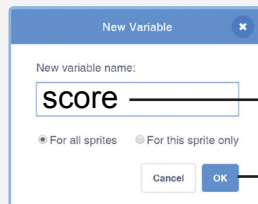
4. MAKE A SCORE VARIABLE



Click **Variables**.

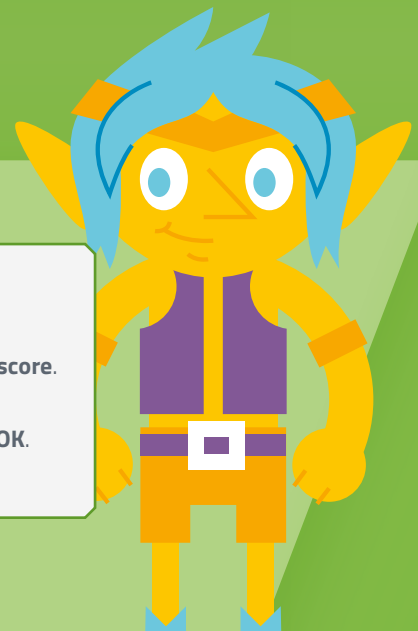
Make a Variable

Click **Make a Variable**.

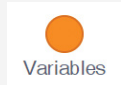


Type **score**.

Click **OK**.



5. MAKE A BLOCK



Variables

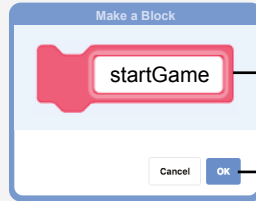


My Blocks

Click **My Blocks**.

Make a Block

Click **Make a Block**.



Type **startGame**.

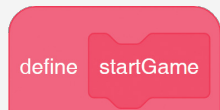
Click **OK**.



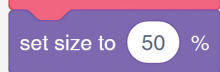
A pink block called **startGame** should appear in the Coding Area.

6. ADD CODE TO DEFINE THE BLOCK

Drag in the following blocks to define the startGame code block:



Run this code when the startGame block is used.



Shrink Dot's size by half.



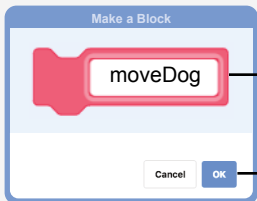
Reset the score to 0.



Reset the timer.

You'll run this code when the game starts to reset the score and get things ready.

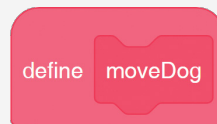
7. MAKE AND DEFINE ANOTHER



Click **Make a Block** and type in **moveDog** as a function name.

Click **OK**.

Drag in this code:



Run this when the moveDog block is used.

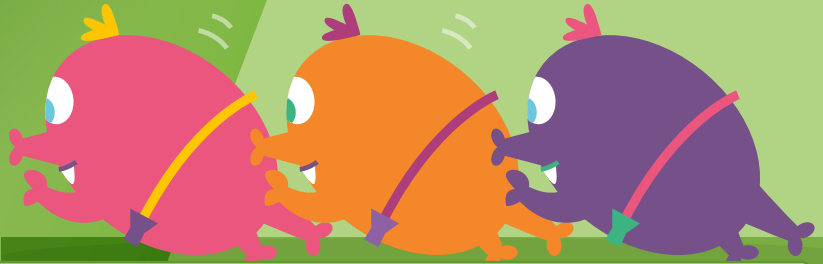


Point Dot toward the mouse.



Move Dot 2 steps forward.

Now let's code by using the blocks you just made.



8. BRING IT ALL TOGETHER

You've defined the blocks you need to start the game and move the dog. Now you'll join them together with other code. Drag in these blocks:

when green flag clicked ← Run the following code when the green flag is clicked.

startGame ← Start the game by running the code in the startGame function.

repeat until timer > 30 ← Keep repeating this code until 30 seconds are up.

moveDog ← Move Dot by running the code in the moveDog function.

say Game over! ← Show a message that the game is over.

Click the **green flag** to test your code! The dog should move slowly toward your mouse pointer.

9. ADD A TACO

Click the **Choose a Sprite** button.

Scroll through the sprites and click **Taco**.

10. CREATE A BLOCK TO POSITION THE TACO

You need a function block that will move the taco to a random place.

Click the **Make a Block** button and name the block **repositionTaco**.

Click **OK**.

Drag in this code:

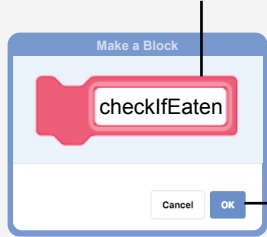
define repositionTaco ← Run this code when the repositionTaco block is used.

go to random position ← Move the taco to a random place on the Stage.

show ← Make sure it's visible.

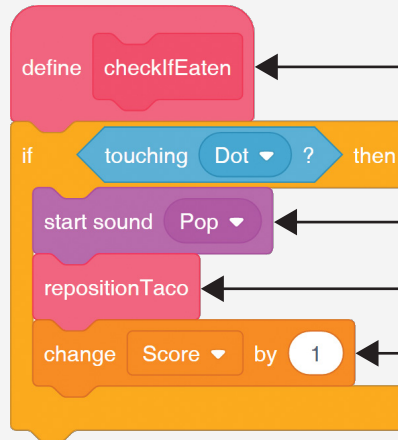
11. CHECK WHETHER THE TACO HAS BEEN EATEN

Click **Make a Block** and type in **checkIfEaten** as a function name.



Click **OK**.

Drag in this code to check whether the taco has been eaten by Dot the dog:



Run the following code when the block is used.

If Dot has reached the taco, then do this.

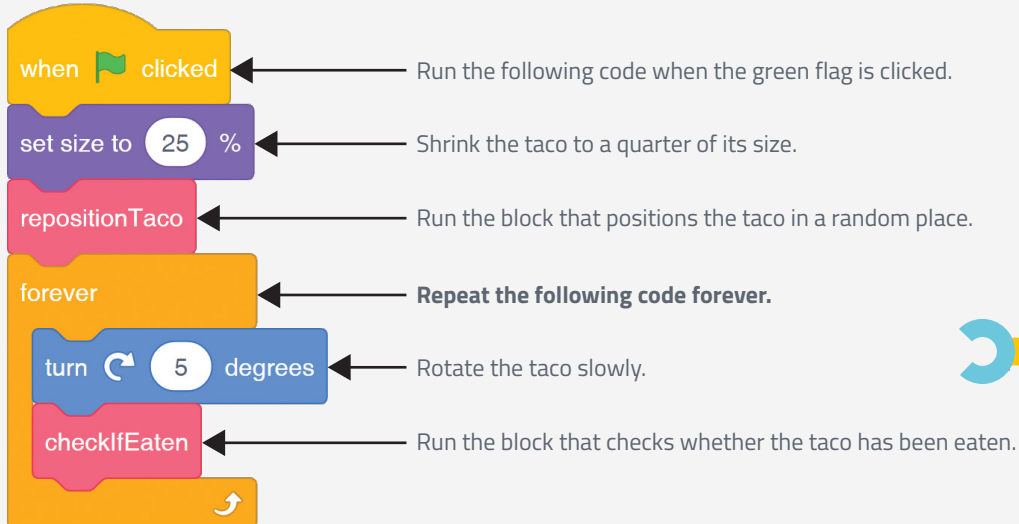
Play a sound effect.

Move the taco to a new place.

Increase the score by 1.

12. BRING ALL THE TACO CODE TOGETHER

You've defined the blocks you need to reposition the taco and to check whether it's been eaten. Now let's put them together with other code. Drag in these blocks:



Run the following code when the green flag is clicked.

Shrink the taco to a quarter of its size.

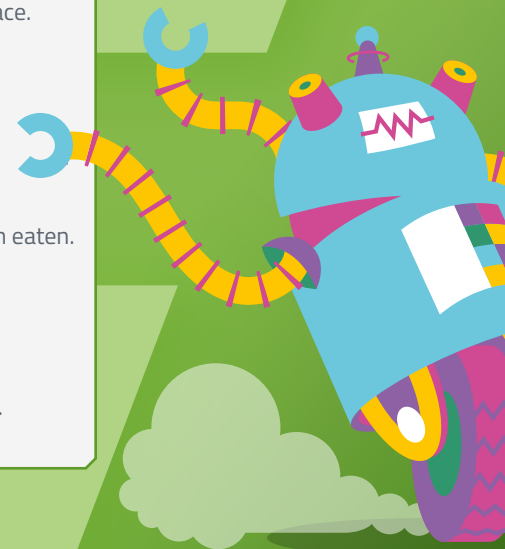
Run the block that positions the taco in a random place.

Repeat the following code forever.

Rotate the taco slowly.

Run the block that checks whether the taco has been eaten.

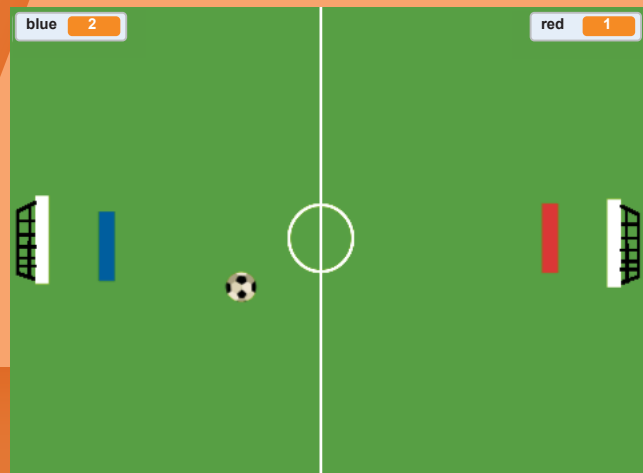
Click the **green flag** to test your code! Guide Dot around the screen to eat up the tacos.



Ready for a challenge? Turn to page 112 and try to build the Collecting Game with Functions.

TWO-PLAYER SOCCER

This is a great two-player game, in which each player has to try to score goals. After drawing the soccer field and the goal nets, you'll build the code. Using function blocks will make the code much clearer and simpler to adapt.

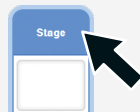


1. NO CATS!

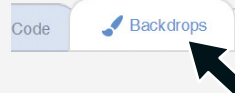


Delete the cat sprite.

2. SELECT THE BACKDROP

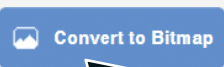


In the sprites pane, click the **Stage** icon. (It's at the bottom right.)



Click the **Backdrops** tab.

3. START DRAWING



Click the **Convert to Bitmap** button.



Click the **Line** tool.



Choose white.



Set the thickness to 5.



Carefully draw a line in the middle of the Drawing Area (look for the center mark).

4. DRAW THE CENTER CIRCLE



Select **Circle**.



Choose **Outlined**.



Drag the mouse to draw a circle in the center.

Hold down the Shift key to make an exact circle.

5. FILL IN THE FIELD

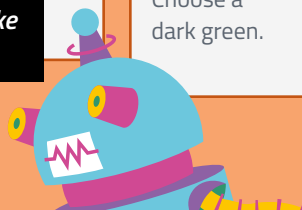


Fill in the field with dark green.

Click the **Fill** tool.



Choose a dark green.



6. ADD A SPRITE



Let your mouse hover over the **Choose a Sprite** button.

Paint



Click the **Paint** option.

To make it easier to see the goal, you'll draw it in black first and then change it to white at the end.



7. DRAW A GOAL



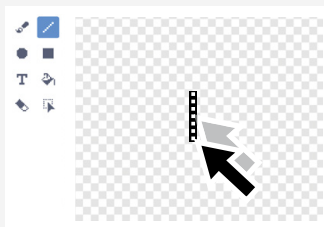
Click the **Convert to Bitmap** button.



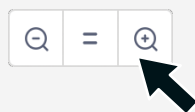
Choose the **Line** tool.



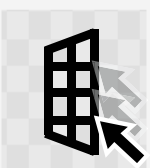
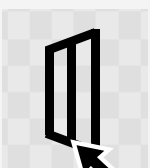
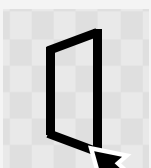
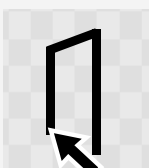
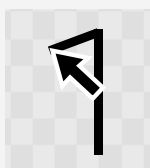
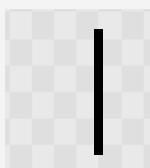
Choose black.



Draw a short vertical line in the center of the Drawing Area.



Click zoom three times so you can see the line better.



Use the mouse to draw the rest of the goal net.



Click the **Fill** tool.



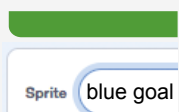
Choose white.



Fill in the goal with white.

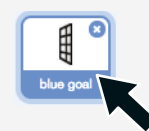
Rename the goal sprite to make your code easier to follow.

8. RENAME IT

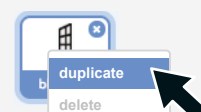


Click inside the Sprite name box, type **blue goal**, and then press **Enter**.

9. DUPLICATE THE GOAL



In the Sprite List, right-click the **blue goal** icon.



Click **duplicate**.

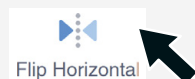
10. FLIP IT



Chose the **Select** tool.



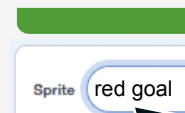
The goal should now look like this.



Flip Horizontal

Click **Flip Horizontal**.

11. RENAME IT



Click in the Sprite name box, type **red goal**, and then press **Enter**.

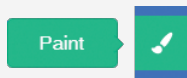


The Sprite List should now look like this.

12. ADD A SPRITE



Let your mouse hover over the **Choose a Sprite** button.



Click the **Paint** option.

13. DRAW A SIMPLE PLAYER



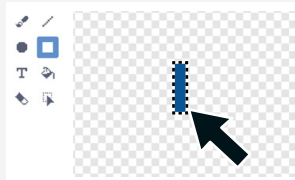
Click the **Convert to Bitmap** button.



Click the **Rectangle** tool.



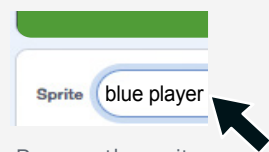
Choose a blue color.



In the center of the Drawing Area, draw a small blue rectangle about the same size as the goal.



14. RENAME IT



Rename the sprite in the Sprite List.

15. CODE THE PLAYER



Click the **Code** tab and then drag this code into the Coding Area:

```
when green flag clicked
  forever loop
    if key a pressed? then
      change y by 5
    if key z pressed? then
      change y by -5
```

Run the following code when the green flag is clicked.

Keep repeating this code forever.

If the **A** key is pressed, then run this.

Move the blue player up by 5 steps.

If the **Z** key is pressed, then run this.

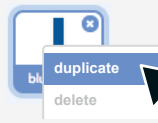
Move the blue player down by 5 steps.

Test your code. Use the **A** and **Z** keys to move the player up and down.

16. DUPLICATE THE PLAYER



Right-click the **blue player** icon in the Sprite List.



Click **duplicate**.

17. CHANGE THE CODE

This player will be controlled by different keys. Edit the code controlling the player:

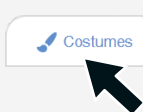
Change: key a pressed? to: key up arrow pressed?

Change: key z pressed? to: key down arrow pressed?

18. RENAME AND RECOLOR THE PLAYER



In the Sprite List, rename it **red player**. Press **Enter**.



Click the **Costumes** tab.



Click the **Fill** tool.



Choose red.



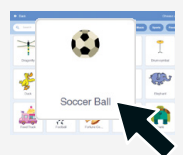
Click inside the rectangle to color it red.

19. ADD A BALL

Click the **Choose a Sprite** button.

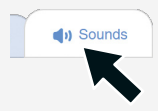


Select the **Soccer Ball**.




20. IMPORT SOUND EFFECTS

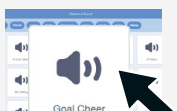
Click the **Sounds** tab.




Click the **Choose a Sound** button.



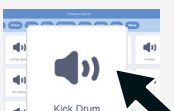
Scroll to find the **Goal Cheer** icon. Click it to import that sound.



Click **Choose a Sound** again.



Scroll to find the **Kick Drum** icon. Click it to import that sound.



21. MAKE A VARIABLE FOR RED

Click the **Code** tab.



Click **Variables**.



Click **Make a Variable**.




Type **red**.



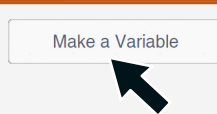
Click **OK**.

22. MAKE ONE FOR BLUE

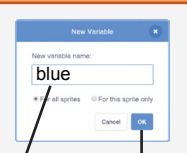
Click the **Code** tab.



Click **Make a Variable**.

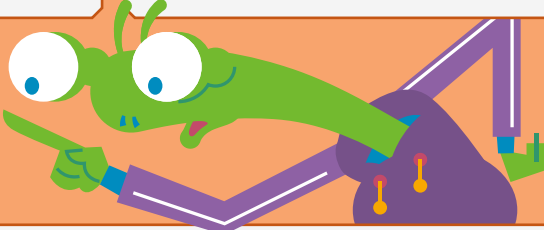


Type **blue**.



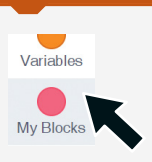
Click **OK**.

Now let's make a block that controls what happens at the start of the game.

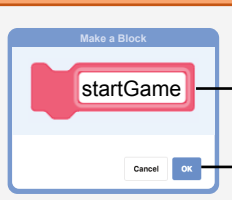


23. MAKE A BLOCK

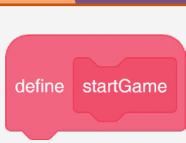
Click **Make a Block**.



Type **startGame**.



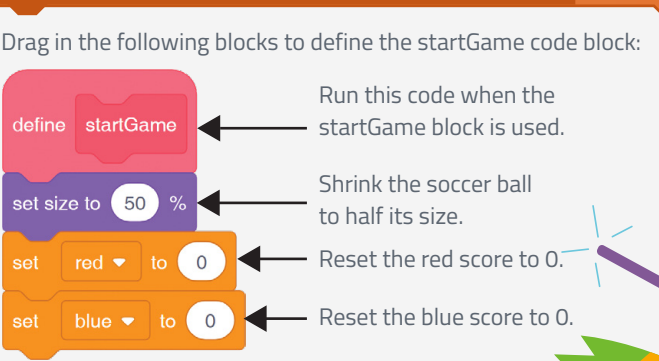
Click **OK**.



A pink block called **startGame** should appear in the Coding Area.

24. ADD CODE TO DEFINE THE BLOCK

Drag in the following blocks to define the startGame code block:



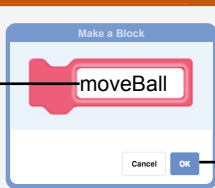
- Run this code when the startGame block is used.
- Shrink the soccer ball to half its size.
- Reset the red score to 0.
- Reset the blue score to 0.

You'll run this code when the game starts. The code resets the score and get things ready.



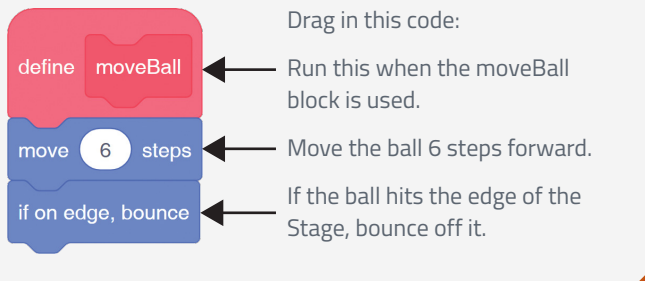
25. MAKE AND DEFINE ANOTHER

Click **Make a Block** and then type in **moveBall** as a function name.



Click **OK**.

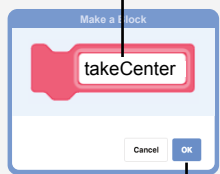
Drag in this code:



- Run this when the moveBall block is used.
- Move the ball 6 steps forward.
- If the ball hits the edge of the Stage, bounce off it.

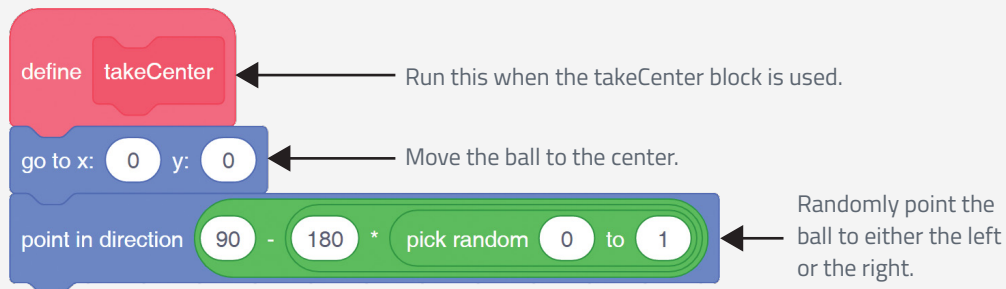
26. MAKE AND DEFINE TAKECENTER

Click **Make a Block** and then type in **takeCenter** as the function name.



Click **OK**.

Drag in this code to define the takeCenter function:



```
define takeCenter
  go to x: 0 y: 0
  point in direction 90 - 180 * pick random 0 to 1
```

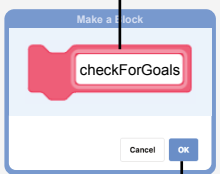
Run this when the takeCenter block is used.

Move the ball to the center.

Randomly point the ball to either the left or the right.

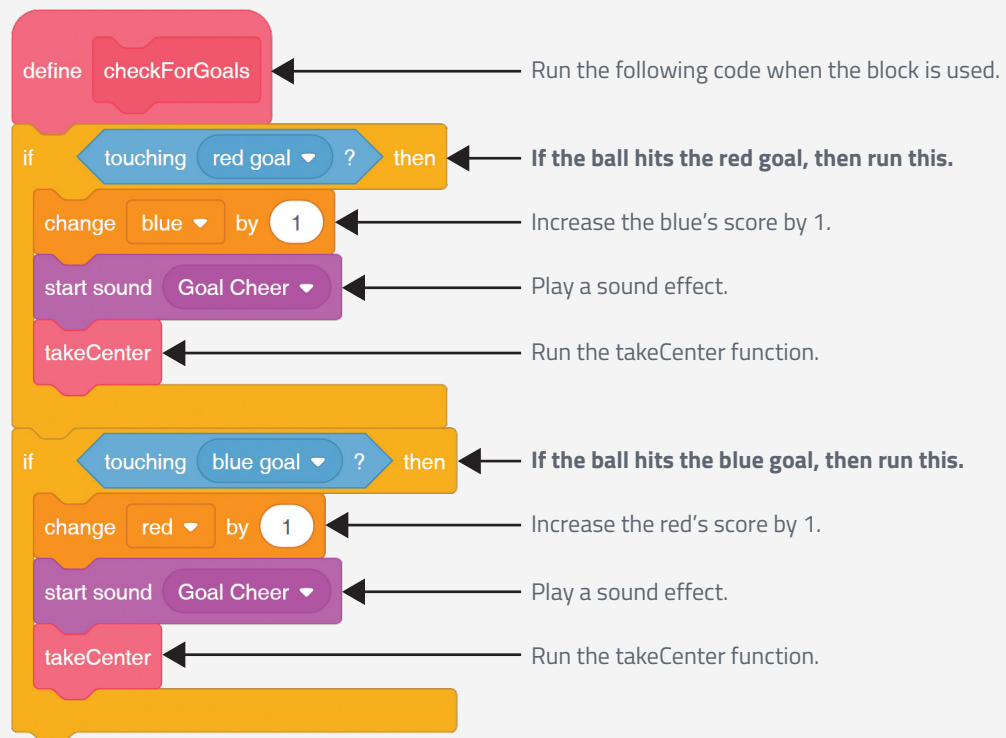
27. HOW TO CHECK FOR GOALS

Click **Make a Block** and then type **checkForGoals** as the function name.



Click **OK**.

Drag in this code to check whether either player has scored a goal.



```
define checkForGoals
  if touching red goal ? then
    change blue by 1
    start sound Goal Cheer
    takeCenter
  if touching blue goal ? then
    change red by 1
    start sound Goal Cheer
    takeCenter
```

Run the following code when the block is used.

If the ball hits the red goal, then run this.

Increase the blue's score by 1.

Play a sound effect.

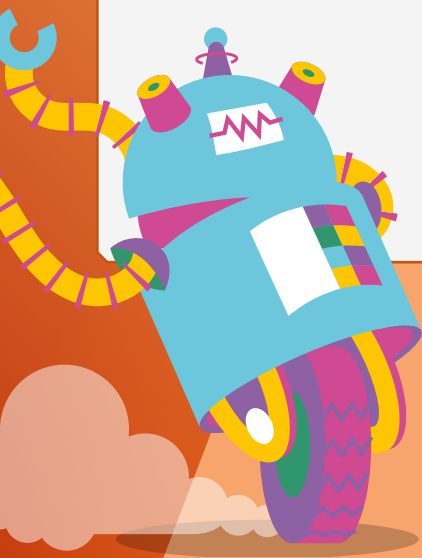
Run the takeCenter function.

If the ball hits the blue goal, then run this.

Increase the red's score by 1.

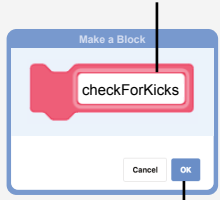
Play a sound effect.

Run the takeCenter function.



28. HOW TO CHECK WHEN A PLAYER KICKS THE BALL

Click **Make a Block** and type in **checkForKicks** as a function name.



Click **OK**.

Drag in this code to check whether either player has kicked the ball. If the player has kicked the ball, the code needs to change the direction of the ball.

```

define checkForKicks
  if touching blue player ? then
    point in direction 90 - y position - y position of blue player
    start sound Kick Drum
  if touching red player ? then
    point in direction -90 - y position of red player - y position
    start sound Kick Drum
  
```

Run this code when the block is used.

If the blue player hits the ball, then run this.

Kick the ball.

Play a sound effect.

If the red player hits the ball, then run this.

Kick the ball.

Play a sound effect.

This code will make the ball move back in the other direction. The code changes the angle depending on whether the ball hits the top, middle, or bottom of the player.

29. BRING IT ALL TOGETHER

Drag in the following code to complete the program. This code will run the other functions that you defined:

```

when green flag clicked
  startGame
  takeCenter
  forever
    moveBall
    checkForGoals
    checkForKicks
  
```

Run the following code when the green flag is clicked.

Get things ready for the start of the game.

Move the ball to the center of the field.

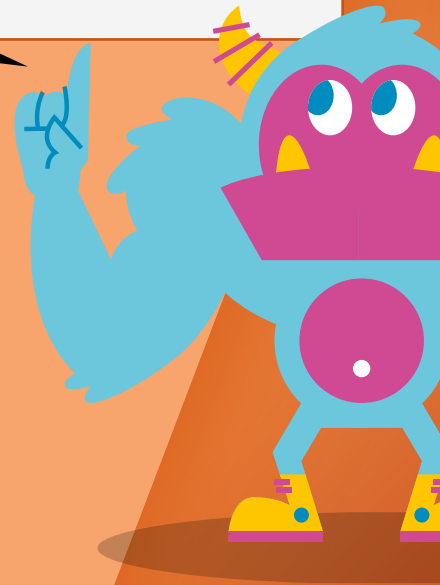
Keep repeating this code forever.

Move the ball in the current direction.

Check to see whether any goals have been scored.

Test to see whether either player has kicked the ball.

Now test your code!



Ready for a challenge? Turn to this chapter's third challenge on page 113 and try rebuilding one of the Chapter 3 games by using function blocks.

ROCKET RACERS

In Chapter 2, you built a two-player game called Race Track. Here, you'll create Rocket Racers, another kind of racing game in which players race rockets across the Stage. You'll create function blocks to move the rockets and to move asteroids toward them.



1. NO CATS!

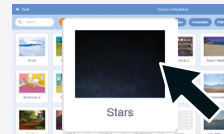


Start by deleting the cat sprite.

2. SET IT IN SPACE



Click the **Choose a Backdrop** button.

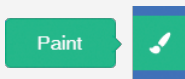


Find **Stars** and click it to set the backdrop.

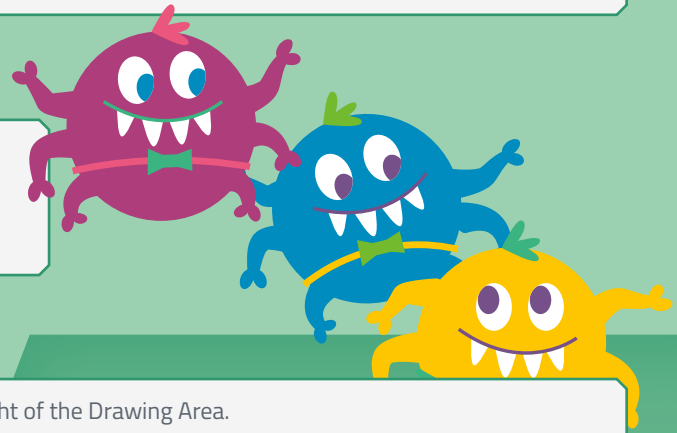
3. ADD A SPRITE



Let your mouse hover over the **Choose a Sprite** button.

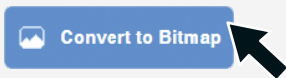


Click the **Paint** option.



4. DRAW AN ASTEROID

Draw an asteroid by using the Line tool. It should be about half the height of the Drawing Area.



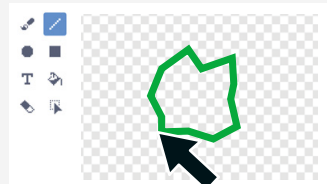
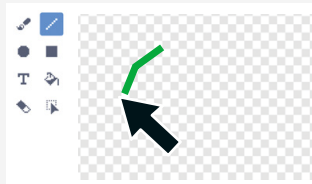
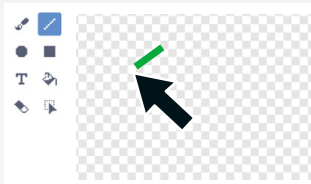
Click **Convert to Bitmap**.



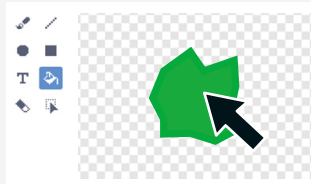
Choose the **Line** tool.



Choose a green color.

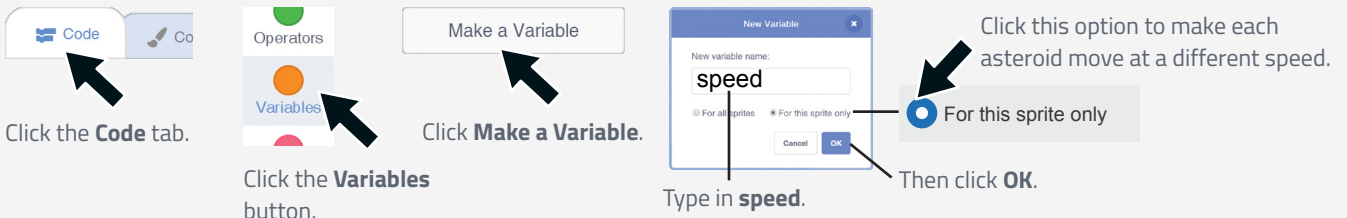


Choose the **Fill** tool.



Fill in the asteroid with green.

5. MAKE A SPEED VARIABLE



Click the **Code** tab.

Click the **Variables** button.

Click **Make a Variable**.

Click **Make a Variable**.

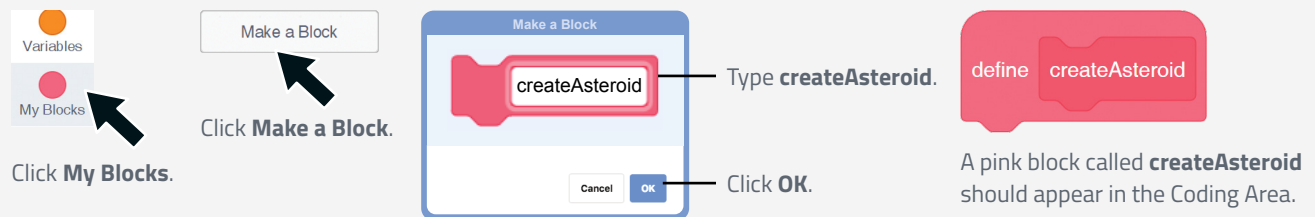
Type in **speed**.

Click this option to make each asteroid move at a different speed.

For this sprite only

Then click **OK**.

6. MAKE A BLOCK



Click **My Blocks**.

Click **Make a Block**.

Type **createAsteroid**.

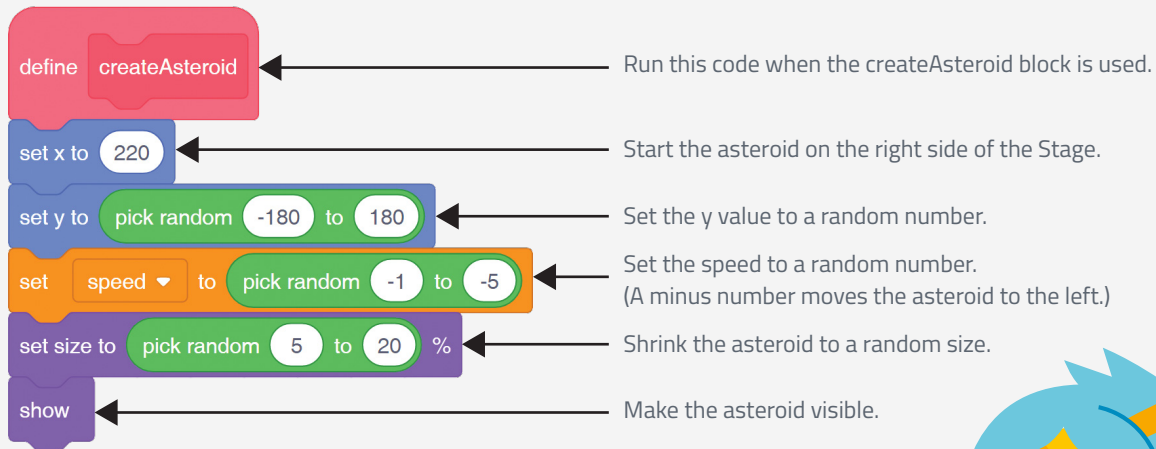
Click **OK**.

define createAsteroid

A pink block called **createAsteroid** should appear in the Coding Area.

7. ADD CODE TO DEFINE THE BLOCK

Drag in the blocks shown to define the createAsteroid block:



define createAsteroid

Run this code when the createAsteroid block is used.

set x to 220

Start the asteroid on the right side of the Stage.

set y to pick random -180 to 180

Set the y value to a random number.

set speed to pick random -1 to -5

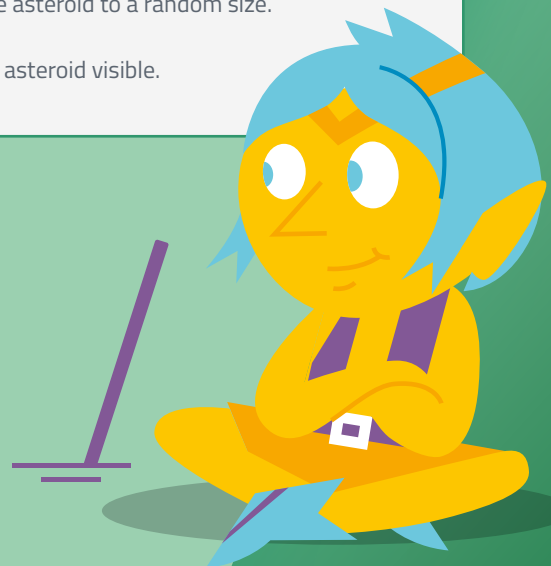
Set the speed to a random number.
(A minus number moves the asteroid to the left.)

set size to pick random 5 to 20 %

Shrink the asteroid to a random size.

show

Make the asteroid visible.



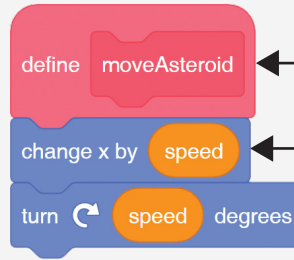
8. DEFINE A BLOCK TO MOVE THE ASTEROID



Click **Make a Block** and then type in **moveAsteroid** as a function name.

Click **OK**.

Drag in this code:



Run the following code when the block is used.

Move the asteroid left according to the value of the speed variable.

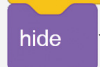
Rotate the asteroid according to the value of the speed variable.

9. COMBINE THE ASTEROID CODE

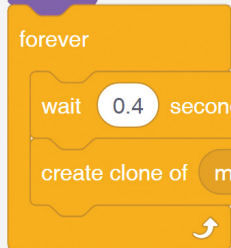
Now that you've defined all the blocks, let's put them together using other code. Drag in these blocks:



Run the following code when the green flag is clicked.



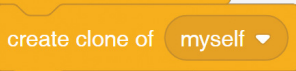
Hide the original asteroid sprite.



Keep repeating the following code forever.



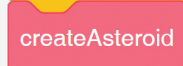
Wait for a fraction of a second.



Create a cloned copy of the asteroid.



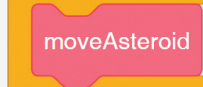
When each asteroid clone is created, run this code.



Create a new random asteroid by using the createAsteroid function.



Repeat the following code until the asteroid reaches the left edge of the Stage.

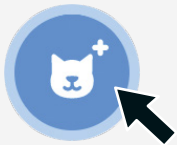


Move it by using the moveAsteroid function.

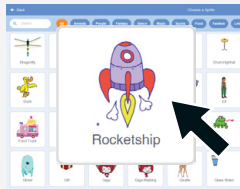


Delete the asteroid clone after it has reached the left edge of the Stage.

10. ADD A SPRITE

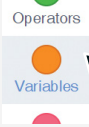


Click **Choose a Sprite**.



Click **Rocketship**.

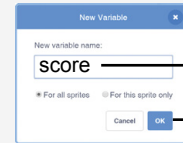
11. MAKE A SCORE VARIABLE



Click **Variables**.

Make a Variable

Click **Make a Variable**.

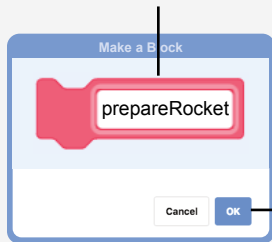


Type **score**.

Click **OK**.

12. PREPARE THE ROCKET

Click the **Make a Block** button and then type **prepareRocket** as the function name.



Click **OK**.

Drag in this code:

```
define prepareRocket
  set size to 25 %
  set color effect to 0
  go to x: -200 y: 60
  point in direction 180
```

Run this when the prepareRocket block is used.

Shrink the rocket to a quarter of its size.

Show its true color.

Start the rocket on the left side.

Change the way it's facing.

13. HANDLE KEY PRESSES

Click the **Make a Block** button and then type **handleKeys** as a function name.



Click **OK**.

Drag in this code:

```
define handleKeys
  if key a pressed? then
    change y by 5
  if key z pressed? then
    change y by -5
```

Run this code when the handleKeys block is used.

If the A key is pressed, then run this.

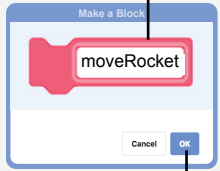
Move the rocket up.

If the Z key is pressed, then run this.

Move the rocket down.

14. DEFINE THE CODE TO MOVE THE ROCKET

Click **Make a Block** and type in **moveRocket** as the function name.



Click **OK**.

Drag in this code to move the rocket:

Run this code when the block is used.

If the rocket hits an asteroid, then run this.

Play a sound effect.

Repeat this code 10 times.

Rotate the rocket 36 degrees clockwise.

Move the rocket left 5 steps.

If the rocket doesn't hit an asteroid, then run this block.

Move the rocket 1 step to the right.

15. BRING ALL THE ROCKET CODE TOGETHER

You've defined all the rocket code blocks. Now let's combine them with other code. Drag in these blocks:

Run the following code when the green flag is clicked.

Run the code that gets the rocket ready.

Keep repeating this code until the rocket reaches the right edge of the Stage.

Run the handleKeys function to move the rocket up or down.

Use the moveRocket code block to move the rocket left or right.

Show a message to the winner.

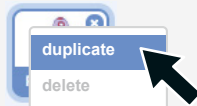
Play a sound effect.

Stop all code in the other sprites.

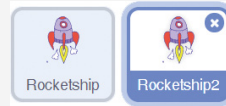
16. ADD ANOTHER ROCKET



In the Sprite List, right-click the **Rocketship** icon.



Click **duplicate**.



A new rocket should appear.

17. CHANGE THE CODE

You'll control the new rocket by using different keys, and you'll make it a different color. Edit the code like this:

In the **prepareRocket** block definition:

Change: `set color effect to 0` to: `set color effect to 120`

In the **handleKeys** block definition:

Change: `key a pressed?` to: `key up arrow pressed?`

Change: `key z pressed?` to: `key down arrow pressed?`

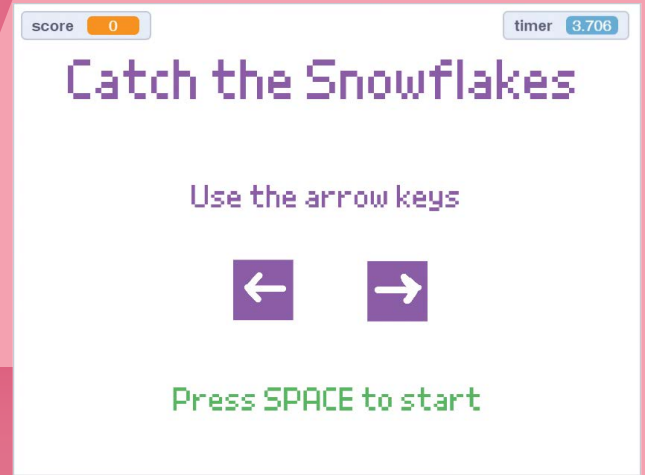
 Click the **green flag** to test your code.



Ready for a challenge?
Turn to page 113 and
try Challenge 4.

CATCH THE SNOWFLAKES

In this game, you'll take a simple catching game (similar to Fruit Catcher in Chapter 3) and take it to the next level. First, you'll build it with function blocks. Then you'll also add an instruction page and use function blocks to display it. You'll control a sprite that will walk left and right across the screen. The game will last for 30 seconds, during which you'll need to catch as many falling snow flakes as possible.



1. NO CATS!



Delete the cat sprite.

2. SELECT THE BACKDROP



Click the **Backdrops** tab.

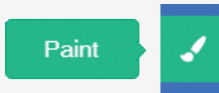


Rename the plain backdrop as **game**.

3. ADD A NEW BACKDROP



Hover over the **Choose a Backdrop** button.



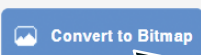
Click the **Paint** option.

4. RENAME IT



Rename this backdrop as **instructions**.

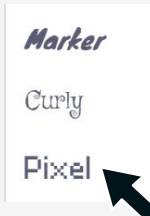
5. ADD A TITLE



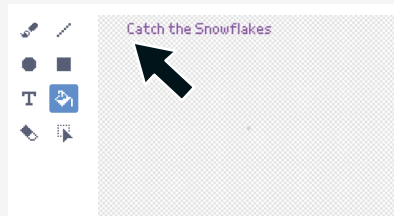
Click **Convert to Bitmap**.



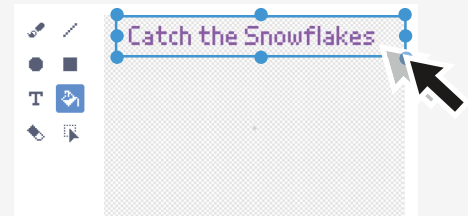
Click the **Text** tool.



Choose the **Pixel** font.



Click in the Drawing Area and then type in the game title.



Adjust the size and position of the text.

6. ADD MORE TEXT

Catch the Snowflakes

Use the arrow keys

Catch the Snowflakes

Use the arrow keys

Press SPACE to start

Add simple instructions on how to play the game.

7. DRAW ARROW KEYS

Catch the Snowflakes

Use the arrow keys

Press SPACE to start

Catch the Snowflakes

Use the arrow keys

Press SPACE to start

Use the **Rectangle** tool to draw two purple rectangles.

Use the **Brush** tool to draw a white arrow on each rectangle.

8. ADD A STAR



Click the **Choose a Sprite** button.



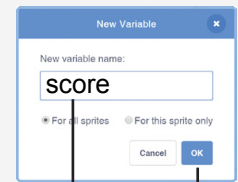
Scroll through and select the **Pico Walking** sprite.

9. MAKE A SCORE VARIABLE

Make a Variable

Click **Variables**.

Click **Make a Variable**.



Type in **score**. Click **OK**.

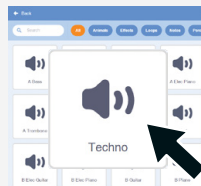
10. IMPORT A SOUND EFFECT

Sounds

Click the **Sounds** tab.



Click **Choose a Sound**.

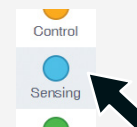


Scroll to find the **Techno** icon. Click it to import that sound.

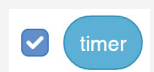
11. SHOW THE TIMER

Code

Click the **Code** tab.



Click the **Sensing** button.



Check the box next to the **timer** code block.

12. MAKE A BLOCK

Variables

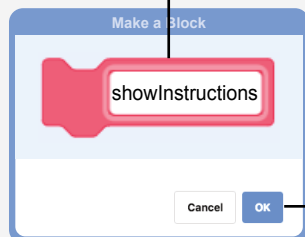
My Blocks

Click **My Blocks**.

Make a Block

Click **Make a Block**.

Click **Make a Block** and type in **showInstructions** as a function name.



Click **OK**.

define showInstructions

A pink block called **showInstructions** should appear in the Coding Area.

The timer will show onscreen how long the game has been running.

13. ADD CODE TO DEFINE THE BLOCK

Drag in the following blocks to define the showInstructions code block:

Run this code when the showInstructions block is used:

Hide the Pico player icon so it's not in the way.

Show the instructions backdrop.

Pause the code until the spacebar is pressed.

14. DEFINE THE STARTGAME BLOCK

Click **Make a Block** and then type in **startGame** as the function name.

Click **OK**.

Drag in this code to get things ready for the start of the game:

Run the following code when the block is used.

Let the other sprites know the game is starting. See page 59 to review how to create a new message.

Reset the score to 0.

Reset the timer.

Shrink the player sprite to half its size.

Stop the sprite from rotating.

Show the plain game background.

Show the player sprite.

15. WHAT HAPPENS WHEN KEYS ARE PRESSED

Click **Make a Block** and type in **handleKeys** as the function name.

Click **OK**.

Drag in this code to program the player's movement when the arrow keys are pressed:

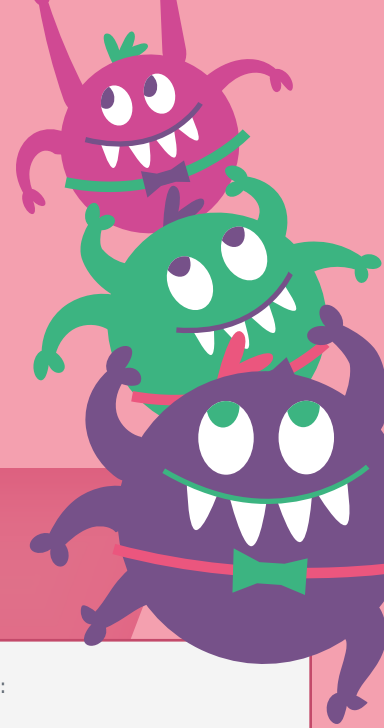
Run the following code when the block is used.

If the *left arrow* key is pressed, then run this code.

Point the Pico player sprite to the left.

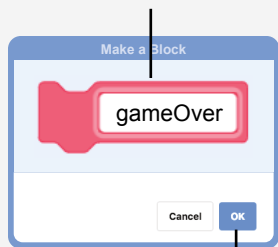
If the *right arrow* key is pressed, then run this code.

Point the Pico player sprite to the right.



16. DEFINE THE GAMEOVER BLOCK

Click **Make a Block** and then type in **gameOver** as the function name.



Click **OK**.

Drag in this code to define what happens at the end of the game:

define **gameOver** ← Run the following code when the block is used.

broadcast **game over** ← Let the other sprites know the game is over.

say **Game over!** for **3** seconds ← Show a short message.

say **join You scored: score** for **3** seconds ← Show the player's score.

17. BRING IT ALL TOGETHER

You've defined the blocks you need to start the game and move Pico. Now let's put them all together with other code. Drag in these blocks:

when **green flag clicked** ← Run the following code when the green flag is clicked.

forever ← **Keep repeating the following code forever.**

showInstructions ← Show the instructions and wait for the player to start the game by pressing the spacebar.

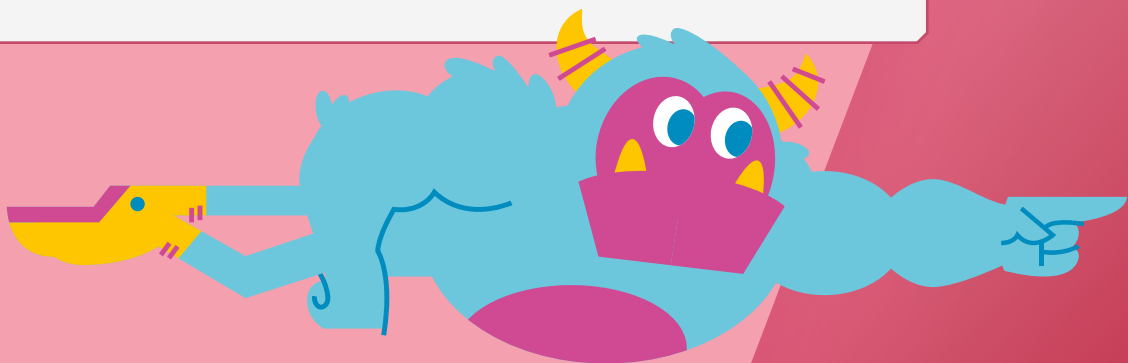
startGame ← Start the game by running the code in the startGame function.

repeat until **timer > 30** ← **Repeat this code until 30 seconds are up.**

handleKeys ← Handle what happens when the arrow keys are pressed.

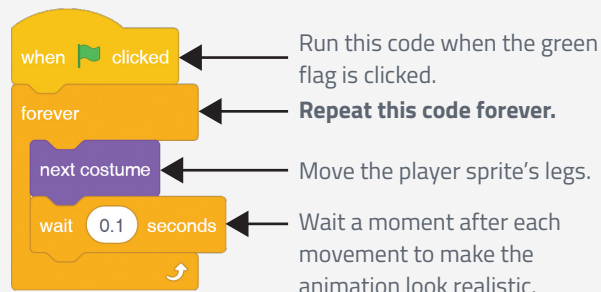
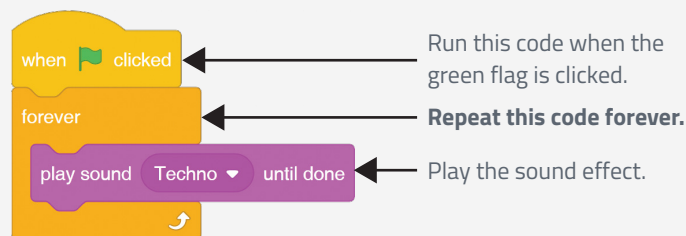
move **4** steps ← Move the Pico sprite 4 steps forward.

gameOver ← Run the code in the **gameOver** function.



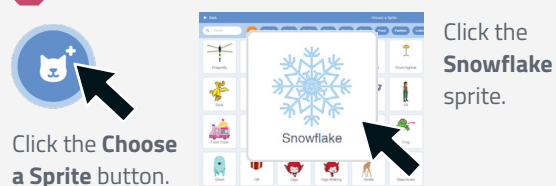
18. ADD SOUND AND ANIMATION

Add these two short sections of code:

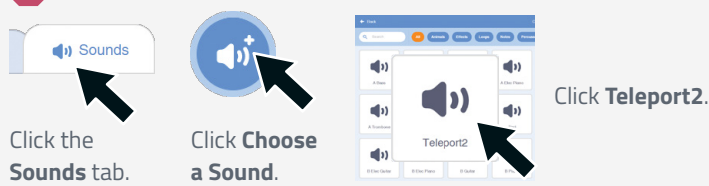


Test your code! You should see the instruction screen appear. Press the **left** and **right arrow** keys to try moving the player around.

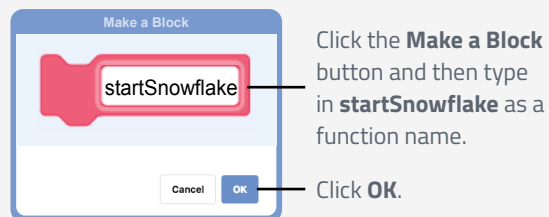
19. ADD A SNOWFLAKE



20. IMPORT A SOUND EFFECT



21. MAKE A BLOCK

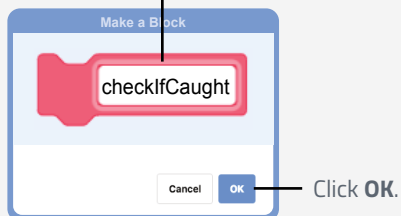


Drag in this code:

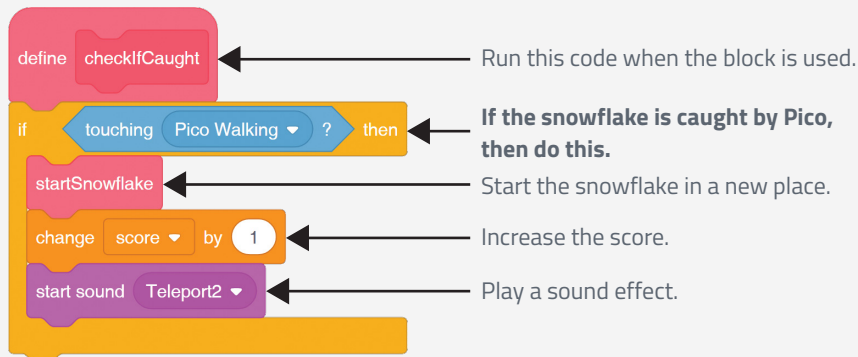


22. CHECK IF A SNOWFLAKE IS CAUGHT

Click **Make a Block** and then type in **checkIfCaught** as the function name.

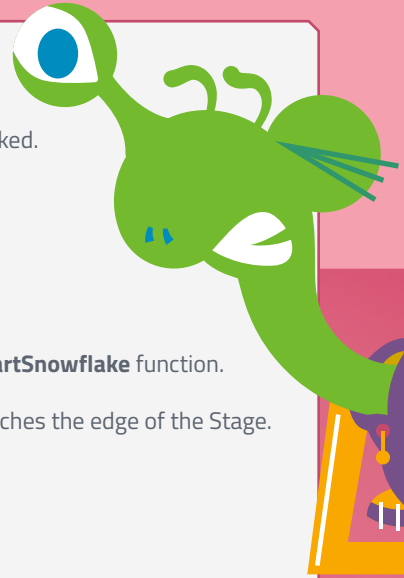


Drag in this code to get things ready for the start of the game:



23. COMBINE THE SNOWFLAKE CODE

Now let's use the blocks you made.



Drag in these blocks to bring the snowflake code together:

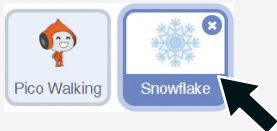
	←	Run the following code when the green flag is clicked.
	←	Hide the snowflake.
	←	Shrink it down to a quarter of its size.
	←	Keep repeating the following code forever.
	←	Start the snowflake by running the code in the startSnowflake function.
	←	Repeat the following code until the snowflake reaches the edge of the Stage.
	←	Move the snowflake down.
	←	Rotate it slowly.
	←	Check whether the snowflake has been caught.

24. LISTEN FOR MESSAGES

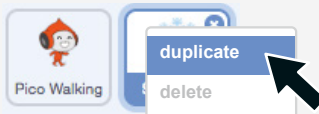
Add these two short sections of code:

	←	When a <i>new game</i> message is broadcast by Pico:		←	When a <i>game over</i> message is broadcast by Pico:
	←	Show the snowflake.		←	Hide the snowflake.

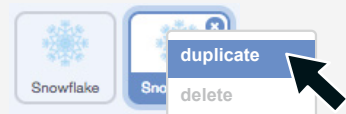
25. ADD ANOTHER SNOWFLAKE



In the Sprite List, right-click the **Snowflake** icon.



Click **duplicate**.



Duplicate another to make three snowflakes.

Click the **green flag** to test your code. Use the **left** and **right arrow** keys to collect as many snowflakes as you can.

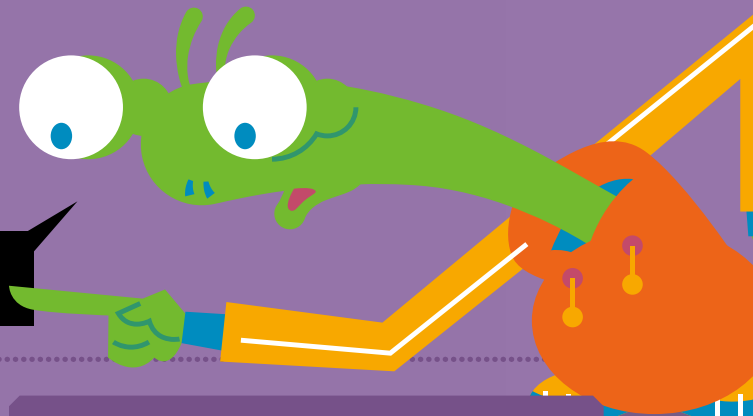
Ready for a challenge? Turn to page 113 and try Challenge 5.

CHAPTER 5

CHALLENGES

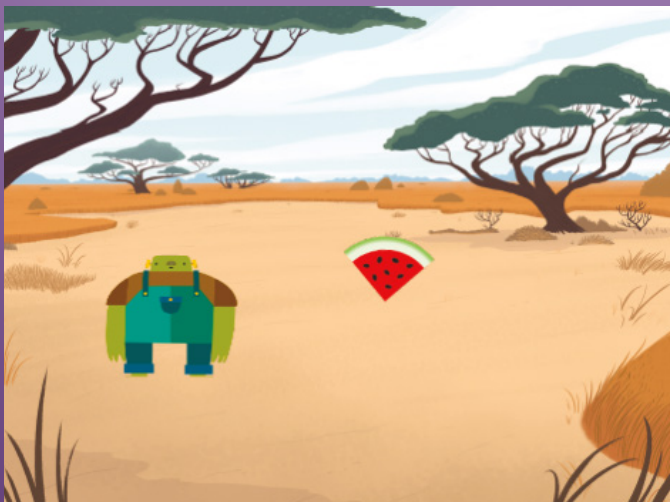
In this chapter, you learned how to make your programs easier to read by making your own blocks and functions.

Once you've completed the programs in this chapter, try the following challenges.



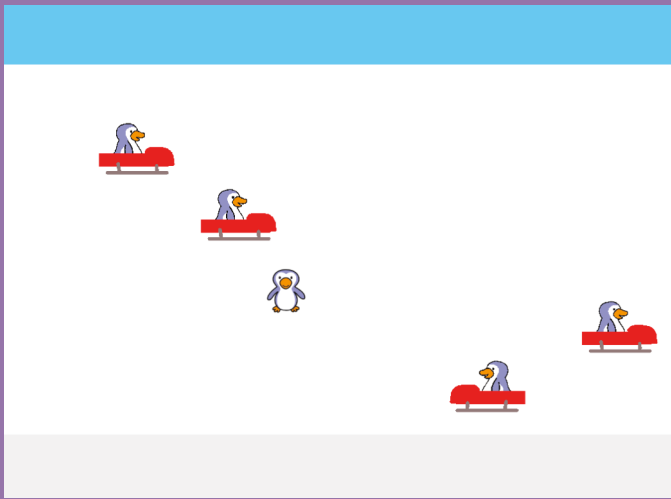
CHALLENGE 1 GOING UNDERGROUND

Make a game in which a robot travels through an underground tunnel. Look at the functions you built in the Cave of Stars game on page 86. Start a new project and then build up the function blocks you need to make your game work. Make changes to the startBlock function to make it start in the correct place and change any colors in the touching color blocks that you need to.



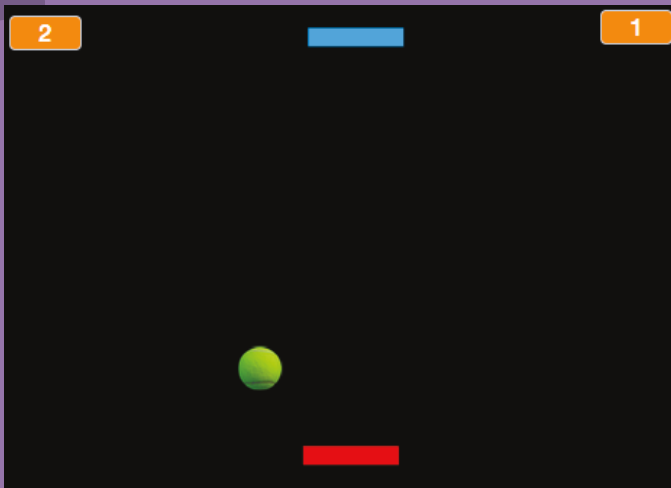
CHALLENGE 2 CRAZY COLLECTING

Once you've built the Space Taco game on page 90, see if you can build your own collecting game. Pick a background and a sprite for your player. Pick another sprite for the object that the player has to collect. Now look at the function blocks defined on pages 91–93 and add the code you need to make your game work. (It won't matter if you change the name of the function from moveDog to moveMonster, as long as you use the name consistently.)



CHALLENGE 3 FANTASTIC FUNCTIONS

To get better at using functions, try rebuilding one of the earlier games in the book by defining your own code blocks. For example, make the Penguin Ski School game by creating your own blocks, including **gameOver** and **startGame**. Add code to each function block to perform a specific task. Look back to the start of the chapter for help.



CHALLENGE 4 FUNCTION VS. FUNCTION

Create your own simple two-player game. Anyone up for tennis? Create the backdrop and player sprites you'll need. Add a ball sprite. Now create function blocks that will allow each player to move around. Look at the Two-Player Soccer code on page 94. Think about how you'll know when to change the score variables.



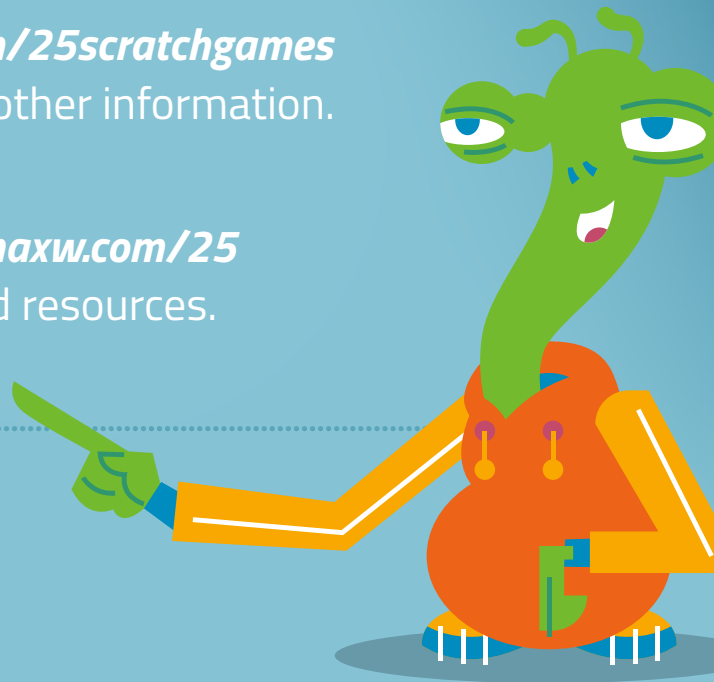
CHALLENGE 5 SPLASHDOWN

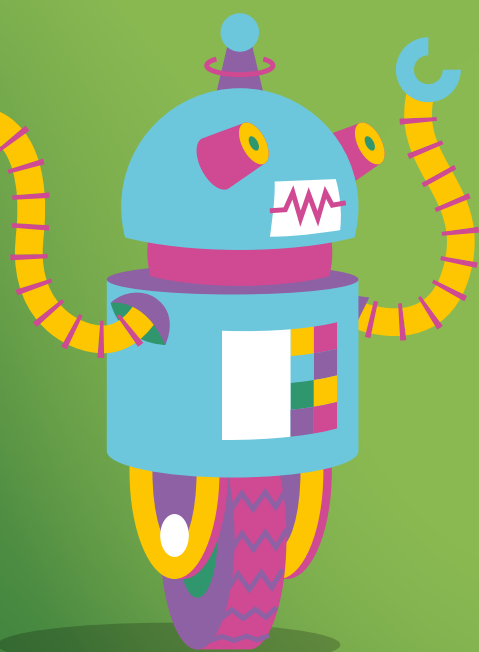
Make a simple game of your own featuring a splash screen. Plan how the main player sprite will move around. What other sprites will you need? How will you know when the game is over? Will you need a score variable? Adapt or use some of the functions you've already created to build your game. Include the showInstructions block from page 107 to add a splash screen, and allow players to restart the game.

UPDATES

Visit <https://nostarch.com/25scratchgames> for updates, errata, and other information.

Please also see <https://maxw.com/25> for more help, advice and resources.







The Official Scratch Coding Cards

Creative Coding Activities for Kids

by Natalie Rusk and the Scratch Team

May 2019, 76 cards, \$24.95, full color, box set
ISBN: 978-1-59327-976-9



Super Scratch Programming Adventure!

by The LEAD Project

Aug 2019, 160 pp., \$19.95, full color
ISBN: 978-1-7185-0012-9



Make Your Own Scratch Games!

by Anna Anthropy

Jul 2019, 192 pp., \$17.95, full color
ISBN-13: 978-1-59327-936-3



Mission Python

Code a Space Adventure Game!

by Sean McManus

Oct 2018, 280 pp., \$29.95, full color
ISBN: 978-1-59327-857-1



AGES 8+

BUILD YOUR OWN COMPUTER GAMES WITH SCRATCH 3



COVERS SCRATCH 3

Learn how to make fun games with Scratch, a free, beginner-friendly programming language from the MIT Media Lab. Create mazes, road-crossing games, and two-player games that keep score. Colorful pictures and easy-to-follow instructions show you how to add cool animations and sound effects. You'll have hours of fun catching snowflakes, gobbling up tacos, and dodging donuts in space—while learning how to code along the way!

ABOUT THE AUTHOR

MAX WAINEWRIGHT is an experienced elementary and high school teacher who has written dozens of educational books for children like the best-selling *How to Code* series, which has been translated into 20 languages. Wainewright has designed and developed coding software and online tools used by

London Grid for Learning and is the founder of 2Simple Software. His programs and websites have won a number of awards including BETT, ERA, and Practical Pre-School Gold Awards. He currently lives in London with his wife and two children and teaches part time in a London elementary school.



SHELVE IN:
Programming Languages/Scratch

ISBN: 978-1-59327-990-5



\$19.95 (US\$30.00 CAN)

9 781593 279905



THE FINEST IN GEEK ENTERTAINMENT™

www.nostarch.com

no starch
press