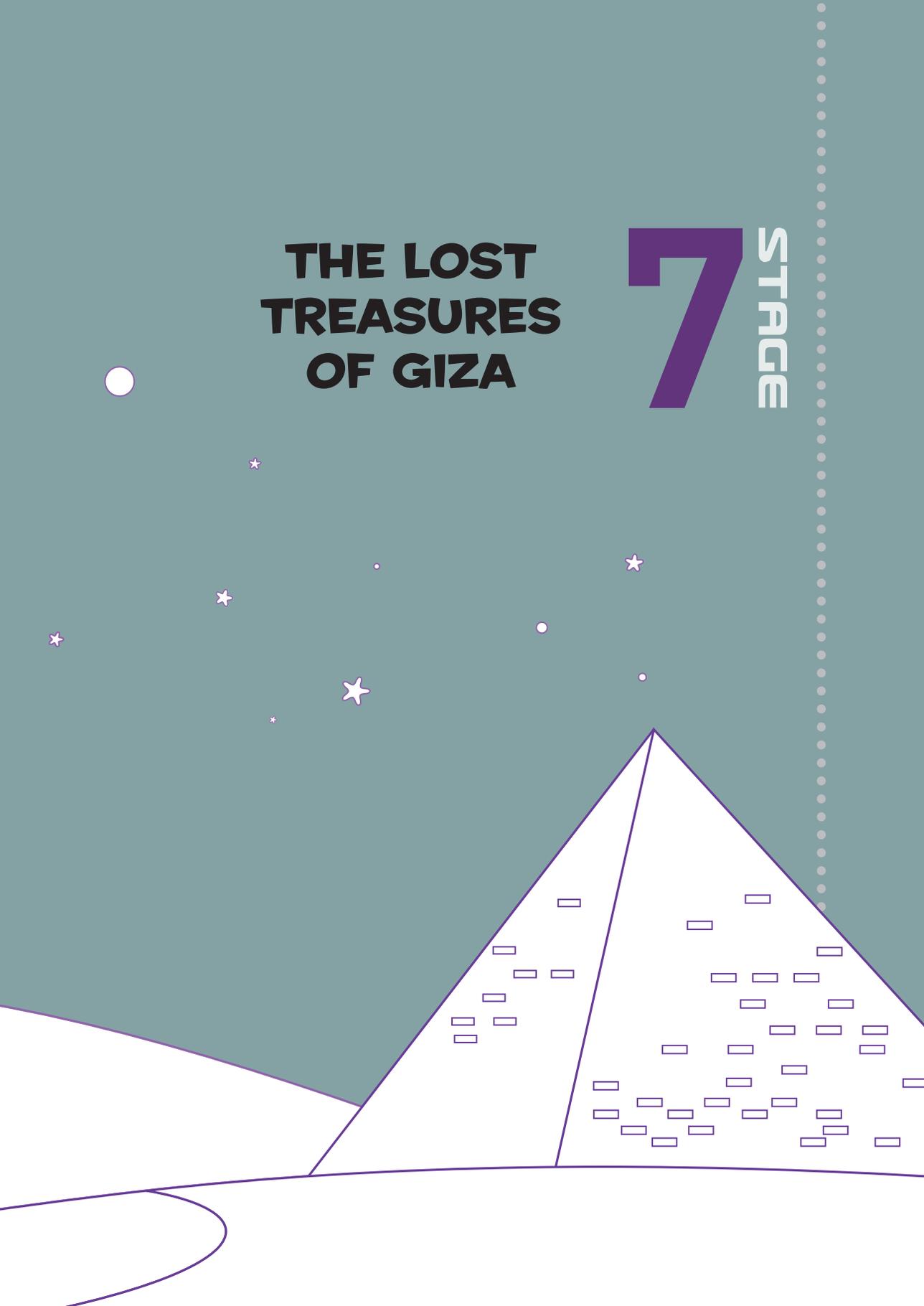


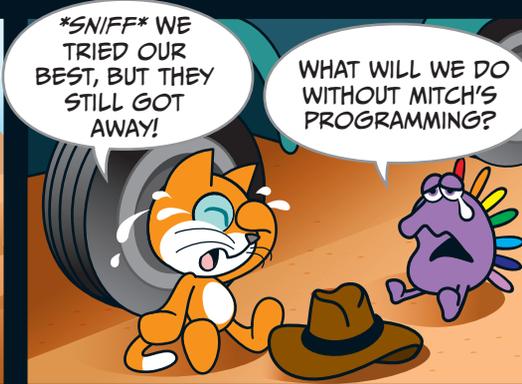
**THE LOST
TREASURES
OF GIZA**

7 STAGE



STAGE

7





ESCAPE THE MAZE!

7 STAGE

+ Chapter Focus

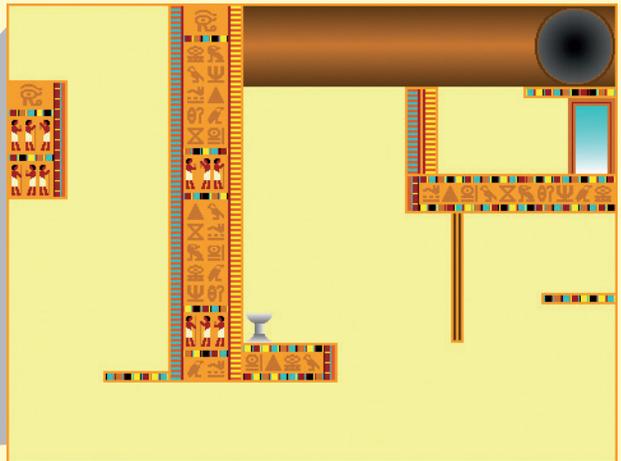
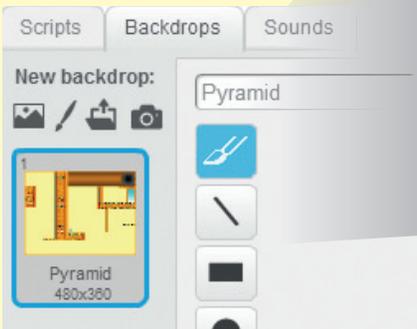
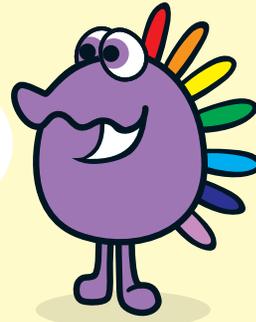
Learn how to design an interactive maze with a guard, booby traps, and treasure!

Game

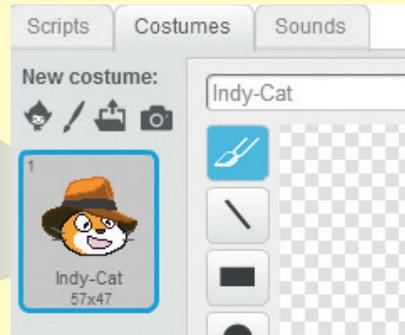
Guide Scratchy through the maze, and into the treasure room to collect the Magic Gem. After he picks up the Magic Gem, other traps in the pyramid are sprung, and he must escape!

For this game, begin by uploading a project file called **07 - The Maze.sb2** (File ► Upload from your computer). This project file has all the images you need for the game, but none of the sprites have any programs yet.

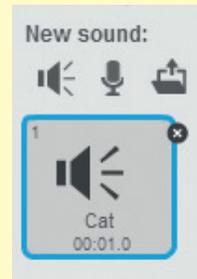
Take a look around, and especially take notice of the Stage. You can see that all of the walls in our maze have the same orange color. We'll use that color as the boundary, so Scratchy can't walk through walls!



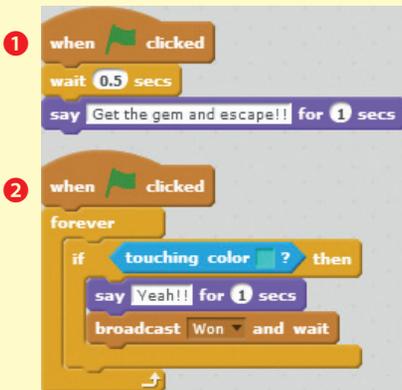
7 STAGE



Click the sprite for Scratchy called **Indy-Cat** in the Sprite List. Then click the **Sounds** tab and add a sound effect for him. Either record a “meow” yourself or use the **Cat** sound effect. We’ll write a program to make Scratchy meow whenever he bumps into a bad guy or trap.



Let’s begin by thinking about how the game should start and how the player will win at the end of the game.



Program **1** gives the player the instructions for the game using the **say** block. Now when the game starts, the player will know he needs to grab the Magic Gem to win.

And, of course, to end the game, Scratchy needs to escape the maze with the Magic Gem. Now let’s write a program for the end of the game. Program **2** uses a special kind of block within a **forever if** loop. If Scratchy touches the color blue—that is, the blue sky of the exit door—he’ll say “Yeah!!” and broadcast **Won**, which will cause the game to end. (Because the maze itself doesn’t have any blue, we don’t have to worry about ending the game accidentally.)

To write program **2**, drag the **touching color** command from the **Sensing** palette into the **if** block. Click the color inside the block, and an eyedropper appears. Click the blue of the doorway, and you’re all set. We’ll use the **touching color** command for another neat programming trick next.

Now take a look at program 3. It looks pretty complicated, but it's really not so hard. Can you tell what it does just by reading it?

First, we set the direction and position of Scratchy. That's simple enough. But what about the big `forever` loop? That holds all of the rest of the program, and that's how we'll program Scratchy's movements. First, if you press the up key, you can see there's a command that will `change y by 3`. But then *inside* that `if` loop, there's a second `if` loop!

If Scratchy is touching orange, the computer tells Scratchy to `change y by -3`. What's that all about? Well, did you notice that the walls of the maze are all orange? So if Scratchy bumps into the orange wall, we want the wall to stop him. And what does $3 + (-3)$ equal? That's right, 0. So when Scratchy touches the orange wall, he doesn't change his y position at all. He won't move! Cool.

The down, left, and right `if` loops work in just the same way, and they have a second `if` loop inside them as well. Make sure to pick orange with the eyedropper for every `if touching color` command.

Now Scratchy can't walk through the maze's walls or gates. Notice that the edge of the Stage has a thin band of orange, too. Scratchy can't walk off the Stage either! He's trapped in our maze, just like we want.

```

3
when clicked
  point in direction 90
  go to x: -205 y: 150
  go to front
  go back 1 layers
  forever
    if key up arrow pressed? then
      change y by 3
      if touching color orange? then
        change y by -3
    if key down arrow pressed? then
      change y by -3
      if touching color orange? then
        change y by 3
    if key left arrow pressed? then
      point in direction -90
      change x by -3
      if touching color orange? then
        change x by 3
    if key right arrow pressed? then
      point in direction 90
      change x by 3
      if touching color orange? then
        change x by -3
  
```

Finally, for program 4, we use the `forever if` block and the `or` block to program what will happen whenever Scratchy bumps into a trap or a bad guy. A speech bubble will say "Oh!", the sound effect Cat will play, and Scratchy returns to his starting position.

Tip: The second `say` block is blank. This makes the "Oh!" disappear.

```

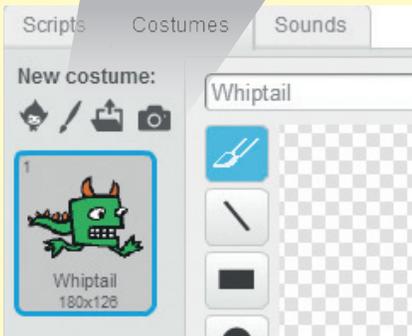
4
when clicked
  wait 1 secs
  forever
    if touching Turnstile? or touching Whiptail? or touching Wall_L? or touching Wall_R? or touching Stone? then
      say Oh!
      play sound Cat
      glide 1 secs to x: -205 y: 150
      say 
  
```

7 STAGE

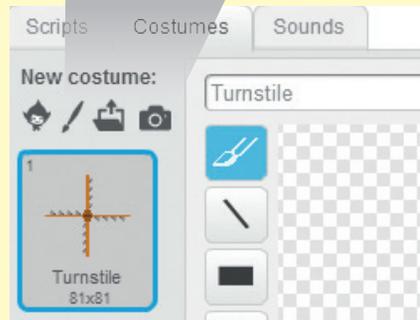
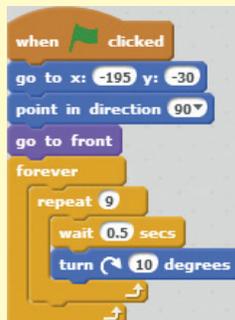
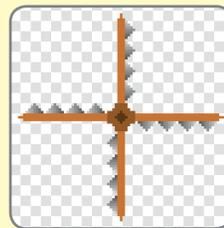
Now is a good time to make sure that your programs work as you expected. Click , and make sure Scratchy moves up, down, left, and right. Try bumping into the walls of the maze. Does Scratchy stop moving once he hits a wall in all four directions? If not, go back and double-check your programming. (Remember that if Scratchy touches the orange wall, his movement should add up to 0.) Try hitting an obstacle or a bad guy to make sure Scratchy returns to the start of the maze.



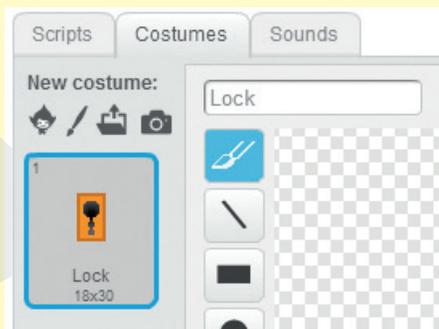
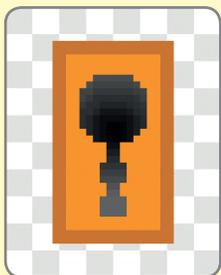
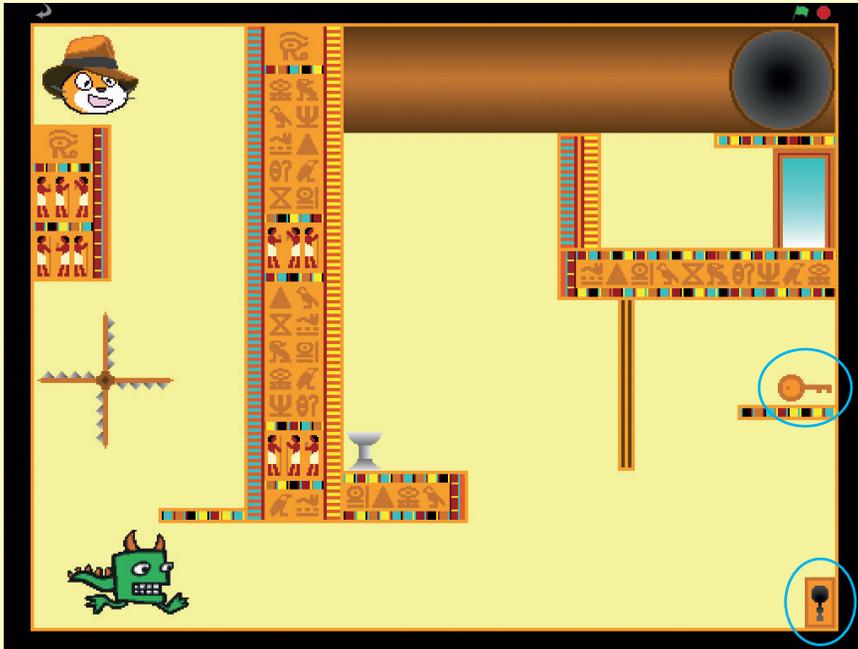
Next, click the sprite for **Whiptail**, the Dark Minion guarding the pyramid. Write a program that sets his size and starting position and then makes him pace back and forth in the maze.



Then click the **Turnstile** sprite, and write a program to make it spin using the `turn` block. The sprite doesn't move around at all, so we just need to set one position.



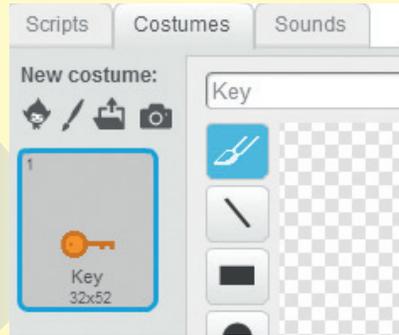
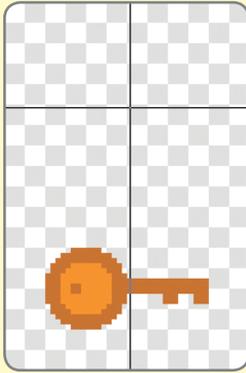
At this point, take a look at the **Lock** and **Key** sprites, which are circled in blue below. Scratchy will need to pick up the Key first, in order to open the Lock. Let's create some programs for them next.



First, click the **Lock** in the Sprite List to give it a simple program—this just sets its location in the maze. The program that actually opens the gate is in the Key sprite.

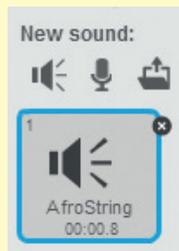


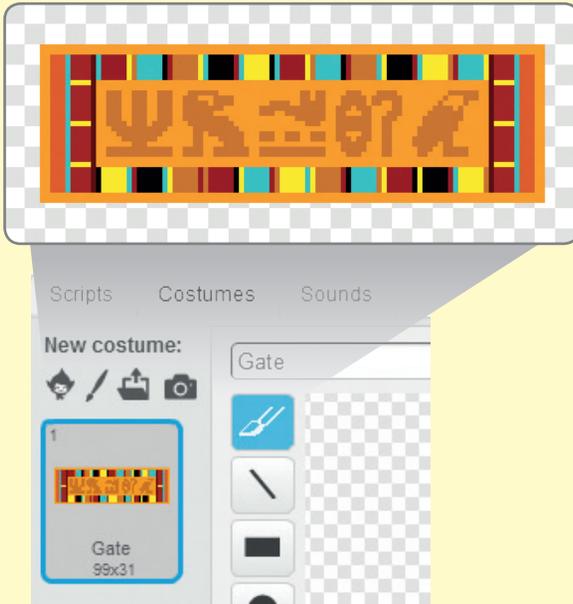
7 STAGE



Tip: When creating the Key sprite, I used the **Set Costume Center** button in the Paint Editor to make sure Scratchy and the Key don't overlap.

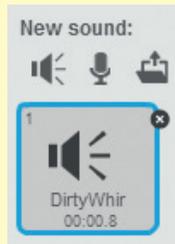
Click the **Key** in the Sprite List, and listen to its sound in the **Sounds** tab. Then click the **Scripts** tab to write this program. We want a sound to play when Scratchy picks up the Key and then have the Key follow Scratchy, using the **go to** command. When the Key touches the Lock, the **Gate Open** signal is broadcast.





Now to program the **Gate** sprite. Because it has an orange border just like our maze, Scratchy can't enter the treasure room unless it moves!

Click the **Gate** in the Sprite List, and then test out the DirtyWhir sound to the Gate in its **Sounds** tab.



Now for some programs. Program 1 just sets the Gate's location. Program 2 makes the Gate glide out of the way when the **Gate Open** broadcast signal is received. Program 3 plays a sound effect.

```

1 when green flag clicked
  go to x: 69 y: -70

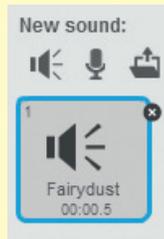
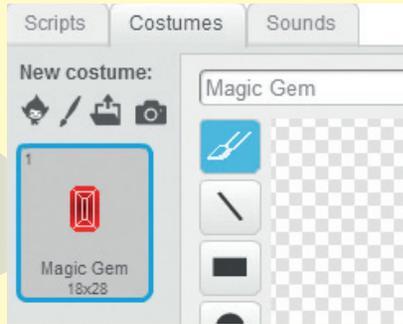
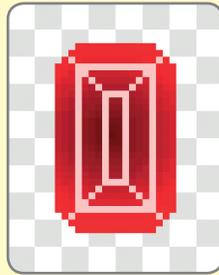
2 when I receive Gate Open
  think Gate Opened!! for 1 secs
  glide 2 secs to x: 69 y: 0

3 when I receive Gate Open
  play sound DirtyWhir
  
```

If you haven't tried out the game yet, give it a test now by clicking ! See if you can get Scratchy to enter the treasure room.



7 STAGE



Next, let's program the **Magic Gem** sprite. We'll use a sound effect called Fairydust in the **Sounds** tab.

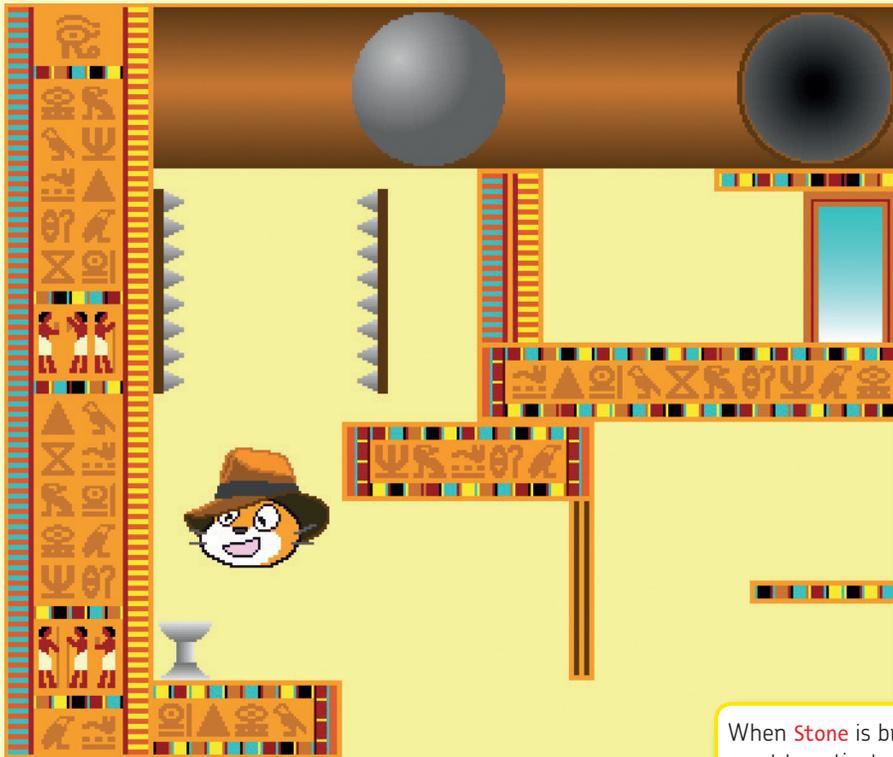
If it's not already there, you can just drag the sprite on top of its stand on the Stage.



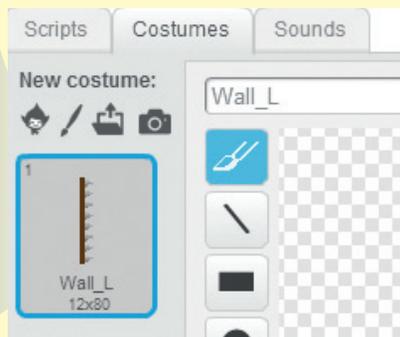
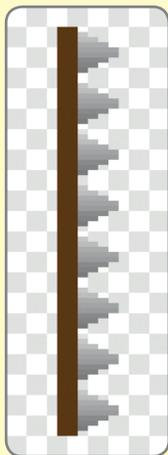
```
1 when clicked
  clear graphic effects
  forever
    change color effect by 25

2 when clicked
  go to x: -42 y: -48
  show
  wait until touching Indy-Cat ?
  play sound Fairydust
  think Gem Obtained!! for 1 secs
  broadcast Stone
  hide
```

Then write two programs for it. Program 1 makes the Magic Gem change colors. Program 2 sets the Magic Gem's position and then uses a **wait until** block to determine what happens when Scratchy grabs the Magic Gem. When Scratchy touches the Magic Gem, it broadcasts **Stone**. This will release the final traps in the maze!



When **Stone** is broadcast, we want to activate the rolling stone and the spiked wall traps.

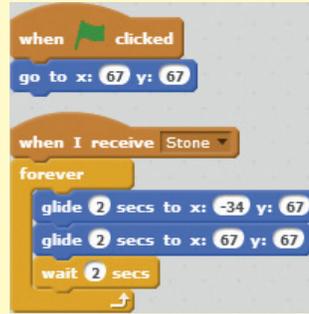
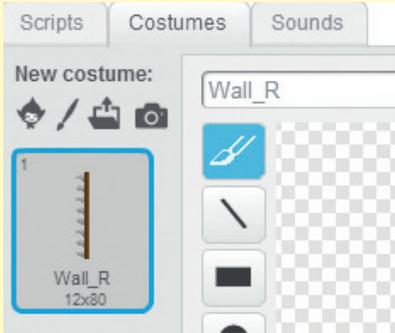


Our spiked wall trap will actually be two different sprites. **Wall_L** (the left side of the trap) gets one simple program to set its position.

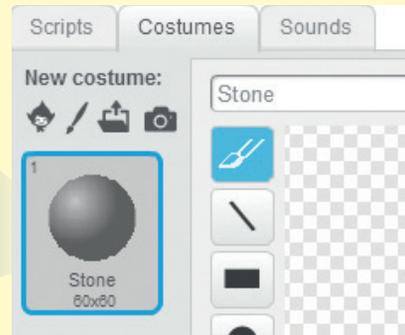
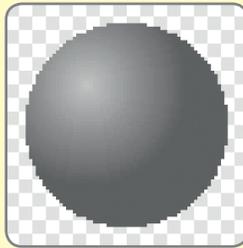


7 STAGE

The right side has its own sprite called **Wall_R**. Create these two programs to set the position and make it move. This wall listens for the **Stone** broadcast and begins to **glide** back and forth, most dangerously!

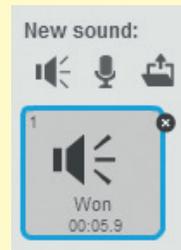
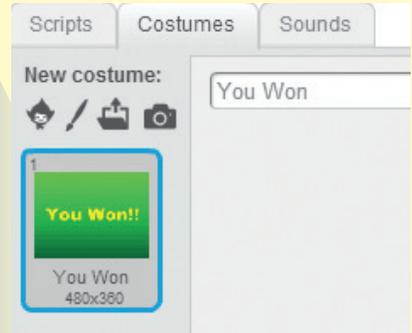


Waiting outside the passage is a rolling boulder sprite called **Stone**. I've used different shades of gray for the Stone to give it a 3D look.



Program 1 for the Stone will make the sprite appear to roll, giving it a realistic animation. Program 2 controls the movement of the Stone—it rolls down the passage and then appears again at the start, in a **forever** loop.

Finally, we have a sprite for the winning screen called **Won**.



```
1 when green flag clicked
  hide

2 when I receive Won
  go to x: 0 y: 0
  go to front
  show

3 when I receive Won
  play sound Won until done
  stop all
```

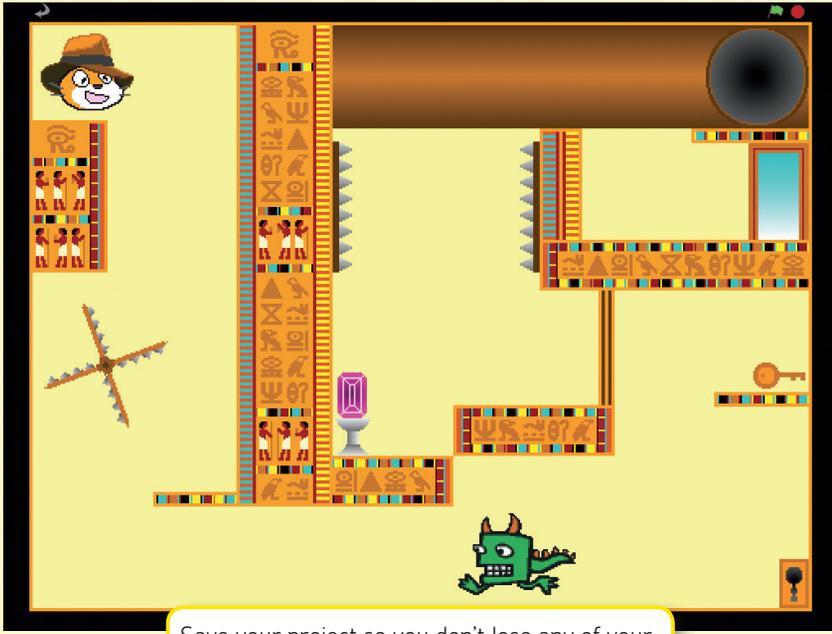
Write these three short programs. Program 1 hides the sprite, and program 2 displays it only when it receives **Won**. Program 3 plays the sound effect we added in the **Sounds** tab.

Tip: The **stop all** command in program 3 will make the Stone, Whiptail, and all other sprites stop moving.

Wondering where that **Won** broadcast will come from? Remember that Scratchy broadcasts **Won** when he touches the blue in the doorway. We added that way back in program 2 on page 108. So we're finished! Yes!

7

STAGE



Save your project so you don't lose any of your work! Now help Scratchy collect the Magic Gem and escape from the dangerous maze.

Scratchy's Challenge!!

By making the sprites smaller, you can create an even more complicated maze with more traps. Or you could add a second player and make it a race to the finish! Give it a try!

