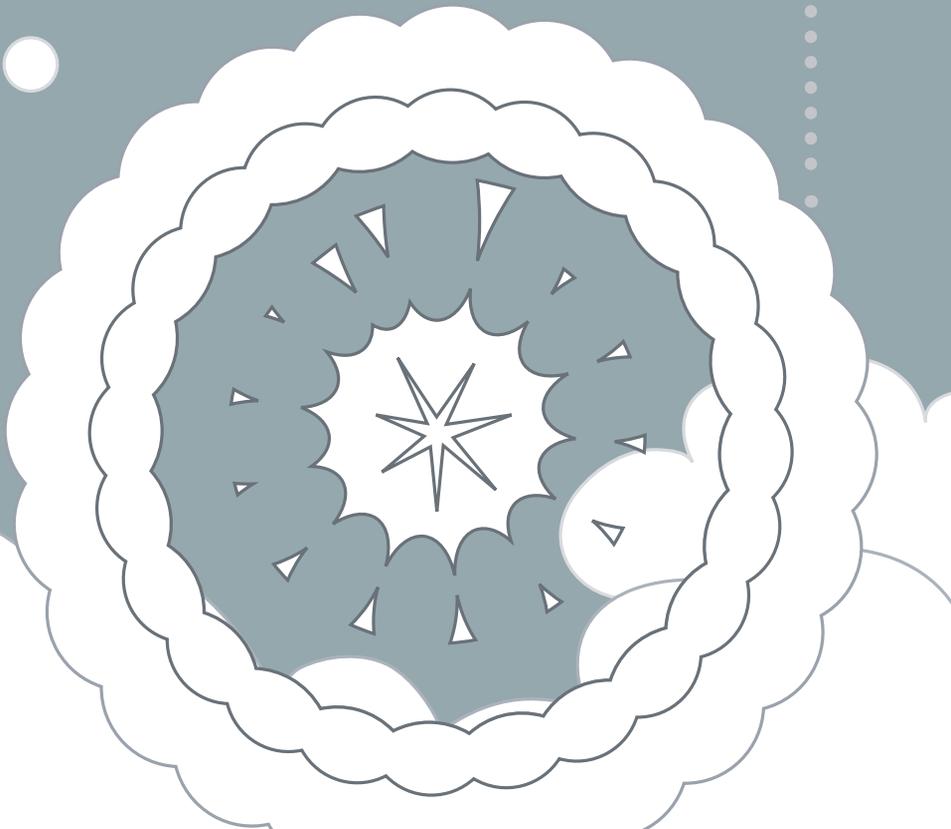


**THE FINAL  
FIGHT...  
IN DARK SPACE**

**9**  
**STAGE**

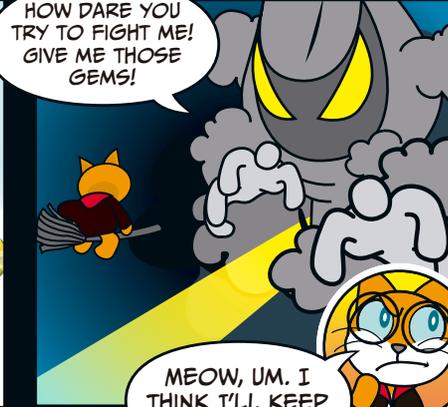


STAGE

9



MEOW!  
I CAUGHT  
THE SECOND  
MAGIC GEM!



HOW DARE YOU  
TRY TO FIGHT ME!  
GIVE ME THOSE  
GEMS!

MEOW, UM. I  
THINK I'LL KEEP  
THEM INSTEAD.



NOOO!



MUA HA HA!  
WELCOME TO MY  
DARK REALM!



GIVE UP  
ALREADY! CAN'T  
YOU SEE I'VE  
CAUGHT YOUR  
FRIEND?

DON'T GIVE  
HIM THE  
GEMS!



POW!

USE THE  
MANUAL,  
SCRATCHY!

BANG!



DARK WIZARD,  
WON'T YOU STOP?

WE STILL  
HAVE TIME TO  
RELEASE ITS  
POWER!



RATA! I'M  
GOING TO  
DESTROY YOU,  
TRAITOR!



WHY DON'T  
YOU PICK ON  
SOMEONE  
YOUR OWN  
SIZE?



## THE FINAL FIGHT

# 9 STAGE

### + Chapter Focus

Learn how to design a *fighting game*. We'll create two characters with unique fight moves, custom health counters, and more. To make custom animations for Scratchy's three fight moves, we'll use a special trick to swap between four different sprites.

### ✎ The Game

Take control of Scratchy for the final fight with the Dark Wizard. Use his saber spin, saber throw, and force attack to defeat the Dark Wizard.

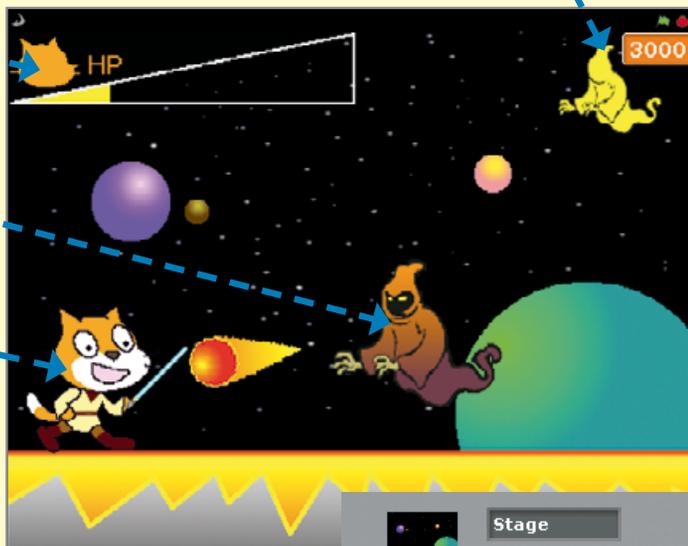
Here's a look at the final game we'll create. You'll need to jump over the Dark Wizard's dangerous fireballs and launch a counterattack!

This sprite represents the Dark Wizard's health.

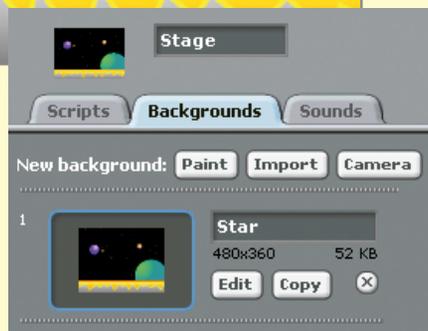
This sprite represents Scratchy's health.

The computer controls the Dark Wizard.

The player controls Scratchy.



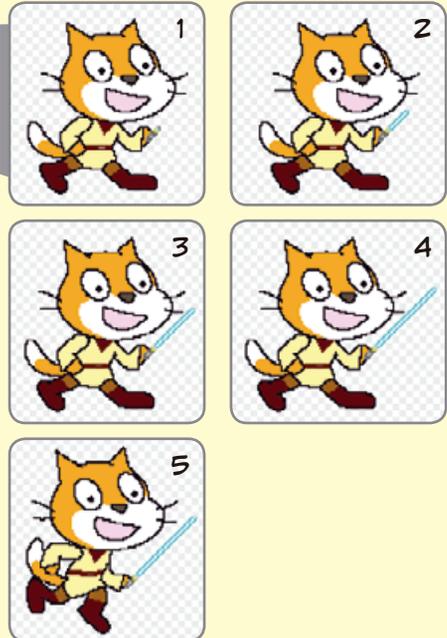
Let's start by importing a blank project called *FinalFight*. This project has all the sprites we'll need, even the Stage. Now let's move on to the exciting stuff—programming!



# 9 STAGE



Let's take a look at the Cat sprite. We'll use these five costumes at the start of the game to make the saber look like it's extending! There's also a costume we'll use for Scratchy's jump animation.



Make sure you click the correct cat sprite in the Sprite List—it's the one named **Cat**. This game has a few different sprites for Scratchy! You'll see why soon.

We also added three sound effects to this sprite's **Sounds** tab. Don't forget that you can record your own!



1

```

when clicked
  point in direction 90
  go to x: -180 y: -60
  clear graphic effects
  show
  switch to costume Saber_on1
  wait 0.15 secs
  switch to costume Saber_on2
  wait 0.15 secs
  switch to costume Saber_on3
  wait 0.15 secs
  switch to costume Saber_fight1
  say Fight!! for 0.5 secs
  forever
    point towards Dark
  
```

Write program 1, which will make a cool starting animation for the game. First, we put Scratchy where he needs to go. Then we use `switch to costume` blocks to change among his three costumes. Next, we use the `say` block to tell Scratchy to say "Fight!" Finally, we use the `point towards` block in a `forever` loop to make Scratchy always face his enemy, the Dark Wizard.

Next, we'll add programs 2, 3, and 4 so that we can move Scratchy to the left and right.

2

```

when clicked
  wait 1 secs
  forever
    if key left arrow pressed?
      broadcast left and wait
    if key right arrow pressed?
      broadcast right and wait
  
```

Try clicking the flag to make sure all your programs work as expected. The game won't really work yet, but you should be able to move Scratchy back and forth.

3

```

when I receive left
  change x by -40
  
```

4

```

when I receive right
  change x by 40
  
```

5

```

when clicked
  wait 1 secs
  forever
    if key up arrow pressed?
      switch to costume Saber_fight2
      broadcast jump and wait
      repeat until y position = -60
        change y by -10
      switch to costume Saber_fight1
  
```

Programs 5, 6, and 7 are for Scratchy's jump ability. Program 5 animates the jump by switching costumes, broadcasts `jump` to control programs 6 and 7, and also creates "gravity" in the `change y by -10` block. When Scratchy lands, he changes back to his original saber fight costume. In program 6, we determine how high Scratchy can jump. Program 7 is just a sound effect for the jump.

6

```

when I receive jump
  broadcast jump sound
  repeat 6
    change y by 30
    wait 0.02 secs
  
```

Tip: Notice how we used the `broadcast and wait` block in program 2. That's to make sure the player doesn't jump too often or jump right off the screen! Scratchy must reach y position -60 to jump again. That's the platform's height.

7

```

when I receive jump sound
  play sound Jump
  wait 2 secs
  stop all sounds
  
```

Tip: Since we're adding so many programs to Scratchy's sprite, you may want to make the Stage small by clicking  so there's more room to program.

# 9 STAGE

Now let's use some new broadcasts to make Scratchy's fight moves! We'll use a cool trick. Whenever Scratchy uses a fight move, he'll actually change into a new sprite instead. Each fight move will get its own sprite, as you'll see.

So we'll hide the Cat sprite and broadcast a unique signal for each move—**Attack1**, **Attack2**, and **Attack3**—in program **8**.

```
9 when I receive show1
   go to Saber Spin
   show

10 when I receive show2
   go to Saber Throw
   show

11 when I receive show3
   go to Force Attack
   show
```

```
8 when clicked
   wait 1 secs
   forever
     if key 1 pressed?
       hide
       broadcast Attack1 and wait
     if key 2 pressed?
       hide
       broadcast Attack2 and wait
     if key 3 pressed?
       hide
       broadcast Attack3 and wait
```

Programs **9**, **10**, and **11** use broadcasts called **show1**, **show2**, and **show3**. We'll use these broadcasts at the end of each attack sequence. These will make Scratchy **show** up again on the screen. The **hide** and **show** blocks are like partners—one makes a sprite disappear, and the other makes it reappear.

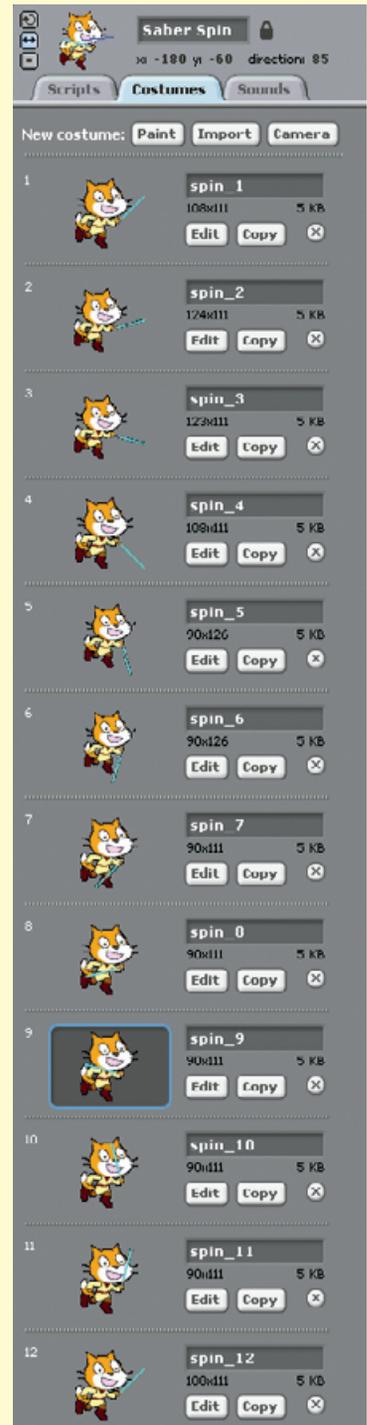
Next, create a new variable using the **Variables** palette, and name it **HP** (for Health Points). Write program **12** to determine Scratchy's starting HP and how dangerous the Dark Wizard's attacks are. Every time Scratchy touches the Dark sprite or Fireball sprite, he loses 5 HP and plays the **Hurt** sound, and the **change color effect** block animates him.

The last program, **13**, determines what happens when all of Scratchy's HP is gone: A broadcast called **lose** is sent.

```
12 when clicked
   set HP to 100
   hide variable HP
   play sound Saber until done
   forever
     if touching Fireball? or touching Dark?
       change HP by -5
       play sound Hurt
       repeat 10
         change color effect by 25
       clear graphic effects

13 when clicked
   wait 1 secs
   forever if HP < 0 or HP = 0
     broadcast lose and wait
```

Now let's set up some costumes for Scratchy's attacks. But instead of adding even more costumes to the Cat sprite, we'll use a new sprite, called Saber Spin, for the spinning saber attack. (Remember how we made a program to hide the Cat sprite in program 8 on the previous page?)



Then add a sound effect for the Saber Spin sprite called Spin in the **Sounds** tab.



# 9 STAGE

Next, use these four programs to control the saber spin attack. Program 1 makes this sprite **go to** the location of the original Cat sprite. Program 2 is just a sound effect when the sprite receives **Attack1**.

Program 3 makes the light saber swirl around three times—by using the block **next costume** in a **repeat 36** loop—and then broadcasts **show1** to tell the Cat sprite that the attack move is finished.

Program 4 determines how much damage the saber does to the Dark Wizard's **Dark HP** variable.

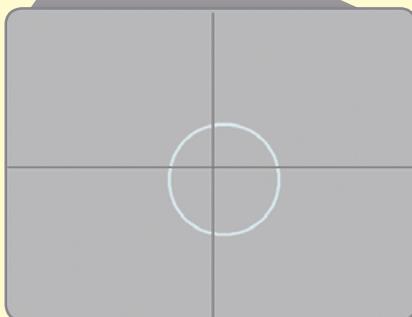
We'll use that **Dark HP** variable to keep track of the Dark Wizard's health. Recall that Scratchy already has his health variable, called **HP**. Take a moment to create **Dark HP** in the **Variables** palette now—we'll need to use this variable in all three of Scratchy's attacks!

```
1 when green flag clicked
   hide
   forever loop
     go to Cat
     point towards Dark

2 when I receive Attack1
   play sound Spin until done

3 when I receive Attack1
   show
   repeat 36
     next costume
   hide
   broadcast show1 and wait

4 when green flag clicked
   forever if touching Dark?
     change Dark HP by -100
     wait 1 secs
```



To give our program a cool look, we can add a ring around the saber, with the Ring sprite.

Tip: To make sure the Ring shows up in the right place during the game, use the **Set costume center** button in the Paint Editor to center it at Scratchy's hand.

```

1 when green flag clicked
  forever
    go to Cat
    point towards Dark

2 when I receive ATK1
  show

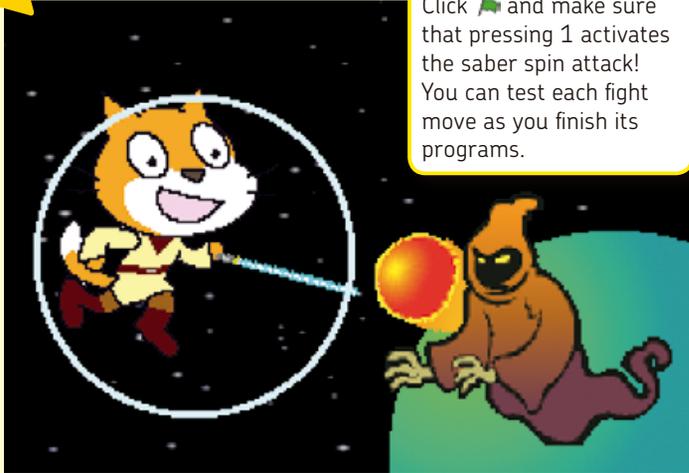
3 when I receive show1
  hide

4 when green flag clicked
  clear graphic effects
  hide
  forever
    change fisheye effect by 50
    wait 0.01 secs
    change fisheye effect by 50
    wait 0.01 secs
    change fisheye effect by -50
    wait 0.01 secs
    change fisheye effect by -50
    wait 0.01 secs
  
```

Then add some simple programs to the Ring. Program 1 makes the Ring appear in the right place, and programs 2 and 3 make sure that the Ring appears only during the **Attack1** sequence. The **fisheye** effect in program 4 makes the Ring expand and contract in a cool animation.

We'll give all of Scratchy's attacks some major defensive power by skipping the health (HP) programming. (Remember that after the end of the saber spin attack, the script broadcasts **show1**, which shows the original Cat sprite, which is vulnerable to attack! This defensive power is only temporary.)

Let's check our work. Click  and make sure that pressing 1 activates the saber spin attack! You can test each fight move as you finish its programs.



```

when green flag clicked
  hide
  forever
    go to Cat
    point towards Dark

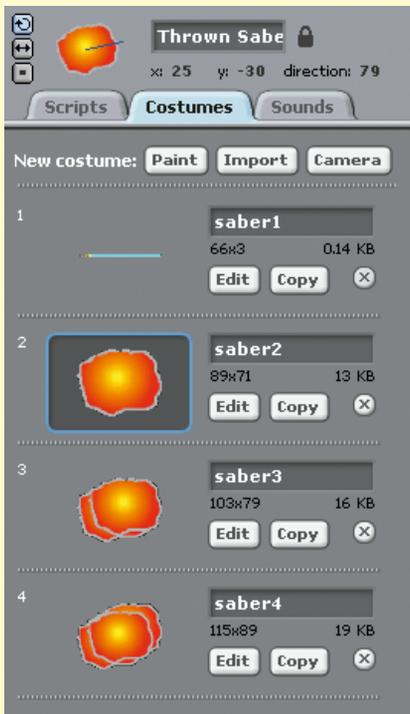
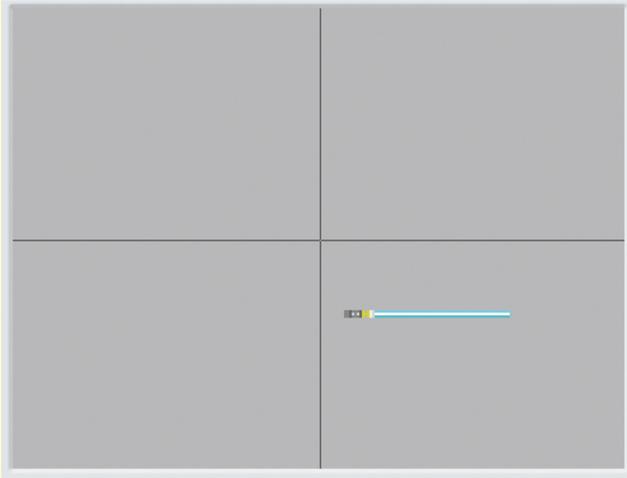
when I receive Attack2
  show

when I receive show2
  hide
  
```

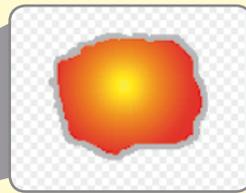
Next, we can design a new sprite for the second fight move—the saber throw attack. It's a simple sprite with just one costume. We'll add some programs to it to make sure this sprite faces the right way and listens for the broadcast **Attack2** to start (and the broadcast **show2** to **hide**).

# 9 STAGE

The cool part of this attack is actually throwing the saber. We'll give it a second sprite, called Thrown Saber, just like we added a second sprite (the Ring) for the saber spin attack. The Thrown Saber sprite has four costumes: a simple saber, followed by three explosion animations.



We'll add a program to use these explosion costumes when we hit the Dark Wizard.



You can add a sound effect for the Thrown Saber and then add program 1 to make it play. Program 2 determines how much damage the saber throw attack does.



```

1 when I receive Attack2
   play sound Saber Throw until done

2 when green flag clicked
   forever if touching Dark?
     change Dark HP by -100
     wait 1 secs
  
```

```

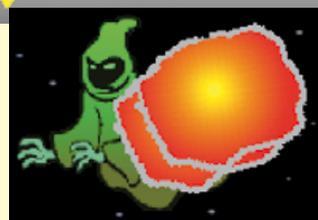
3 when green flag clicked
   hide

4 when I receive Attack2
   go to Cat
   point towards Dark
   switch to costume saber1
   go to front
   show
   wait 0.2 secs
   glide 0.5 secs to x: x position of Dark y: y position of Dark
   if touching Dark?
     switch to costume saber2
     wait 0.1 secs
     switch to costume saber3
     wait 0.1 secs
     switch to costume saber4
     wait 0.1 secs
     hide
     broadcast show2 and wait
   else
     wait 0.3 secs
     hide
     broadcast show2 and wait
  
```

Then write these programs. Program 3 hides the flying saber until we need it. Program 4 points the saber at the Dark Wizard and launches it! When it hits the Dark sprite, we make the sprite switch to its explosion costumes. Note the special glide command that finds the Dark Wizard, no matter where he is. At the end of this program, we broadcast show2. This will make Scratchy switch back to his original Cat sprite.



No matter where he goes, we can hit the Dark Wizard with the saber throw attack—pretty powerful! Give this attack move a test, too, and make sure it hits the Dark Wizard. Press 2 after clicking.



# 9 STAGE



Now let's program the final fight move, the force attack. Don't forget you can add a new sound effect for it in the **Sounds** tab.



```

1 when green flag clicked
  hide

2 when I receive Attack3
  go to Cat
  point towards Dark
  clear graphic effects
  go to front
  show
  repeat 5
    change ghost effect by 25
    wait 0.1 secs
    change ghost effect by 25
    wait 0.1 secs
    change ghost effect by -25
    wait 0.1 secs
    change ghost effect by -25
    wait 0.1 secs
  
```

Program 1 hides this costume until we launch the force attack. Program 2 uses the ghost effect to make the lights flash. Even though our sprite has only one costume, we created a cool effect—this program will make our attack pulse with energy!

Write program 3 to play your sound effect, and program 4 to make sure this attack will reduce Dark HP by 100 if the Force Attack sprite touches the Dark Wizard.

```

3 when I receive Attack3
  play sound Force until done
  hide
  broadcast show3 and wait

4 when green flag clicked
  forever if touching Dark?
    change Dark HP by -100
    wait 1 secs
  
```



The final program **5** will help Scratchy to land when he uses this attack while jumping.

```

5 when green flag clicked
    forever loop
        repeat until y position = -60
            change y by -10
    
```

Now Scratchy has all three of his fight moves. Click , and test your program to make sure it behaves exactly as you expected! Walk around; press 1, 2, and 3 to activate the fight moves; and try jumping around the screen. Now Scratchy is ready for this fight.



Finally, we can get to the Dark Wizard!

# 9 STAGE

First, let's set his starting position (x: 170, y: -30) and his size (65% of the original sprite, so he's not too big) in program 1. Program 2 controls how he moves on the platform. He just picks a random spot between x:-85 and x:170 and glides there in a **forever** loop.

```
1 when clicked
  go to x: 170 y: -30
  clear graphic effects
  set size to 65 %
  show

2 when clicked
  wait 1 secs
  forever
    glide pick random 0.5 to 2 secs to x: pick random -85 to 170 y: -30
  wait 1 secs
```

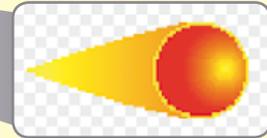
In program 3, we use the **Dark HP** variable we created earlier to keep track of the Dark Wizard's health. This program also makes sure he always faces his enemy, Scratchy.

```
3 when clicked
  set Dark HP to 3000
  show variable Dark HP
  forever
    point towards Cat

4 when clicked
  wait 1 secs
  forever
    if touching Saber Spin ? or touching Thrown Saber ? or touching Force Attack ?
      repeat 10
        change color effect by 25
        clear graphic effects
    if Dark HP < 0 or Dark HP = 0
      hide
      broadcast win and wait
```

In program 4, we add two sets of **if** blocks inside a **forever** command. If the Dark Wizard touches one of Scratchy's attacks, he'll **change color**. (Scratchy's attacks already have programs that subtract from the variable **Dark HP**.)

Now for the Dark Wizard's furious fireball attack! This is a new sprite called Fireball, and you can add a sound effect for it, too.



Write program 1 to give it a sweet animated look using a **fisheye** effect.

```

1 when green flag clicked
  clear graphic effects
  forever
    change fisheye effect by 20
    wait 0.01 secs
    change fisheye effect by 20
    wait 0.01 secs
    change fisheye effect by -20
    wait 0.01 secs
    change fisheye effect by -20
    wait 0.01 secs
  
```

Then add program 2 to control how often the Dark Wizard uses his attack and where the fireball goes once it's launched! Can you see how it works?

Program 3 plays our sound effect for the Fireball.

```

2 when green flag clicked
  hide
  wait 1 secs
  forever
    wait pick random 1 to 5 secs
    go to Dark
    point towards Cat
    show
    broadcast Dark Attack
    repeat 60
      move 8 steps
      if touching Cat?
        wait 0.25 secs
        hide
      if touching edge?
        hide
    if Dark HP < 0 or Dark HP = 0
      stop script

3 when I receive Dark Attack
  play sound Dark Attack until done
  
```

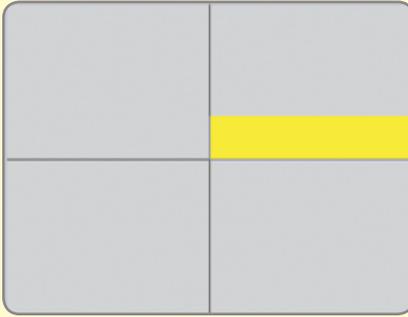
Tip: We used **move** instead of **glide** so that Scratchy has a chance to jump away. The **if touching Cat** and **if touching edge** statements make the fireball disappear once it touches Scratchy or the edge of the screen.

The **wait 0.25 secs** block in the **if touching Cat** loop makes sure that the fireball actually does damage before disappearing!

Don't forget to double-check your programming by making sure that these fireballs do damage, too. Click and let one of the fireballs hit Scratchy! Ouch!

# 9 STAGE

Now that the main programming is finished, let's add custom HP counters for each character, just like you'd see in any other fighting game. First, let's use the yellow bar sprite for Scratchy called Health.



```
when clicked
go to x: -241 y: 130
show
forever
  set color effect to 0
  set size to HP %
  if HP < 21
    set color effect to 170
  if HP < 0 or HP = 0
    hide
```

Add this program to make the health bar become smaller each time HP is subtracted, using the `set size` block. If Scratchy's HP goes lower than 21%, the bar will change color as a warning to the player. The final `if` loop hides this sprite if HP is completely depleted.

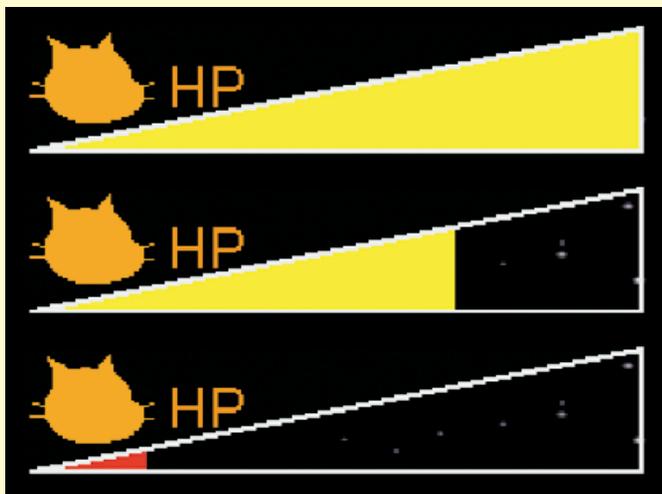
Add a sprite on top of the Health sprite called Health Box. The bottom half of the Health Box is transparent, which lets a triangular portion of the health bar show through. The Health Box gets a short program just to set its position.

```
when clicked
go to x: -264 y: 153
show
```

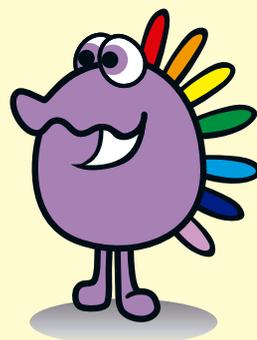




To hide the variable **HP** so it doesn't appear on the screen, just uncheck the **HP** variable in the **Variables** palette. There's also a `hide variable` command, if you want to add it to your programs.

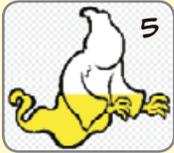
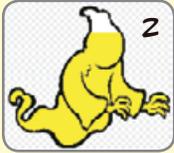
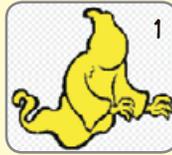


Now we can see how much HP Scratchy has left, just by looking at the top-left corner of the Stage.



# 9 STAGE

For the Dark Wizard's HP meter, we'll use a costume-switching program. The Dark HP sprite has seven costumes.



```

when clicked
  go to x: 180 y: 140
  switch to costume dark1
  set size to 40 %
  forever
    if 2500 > Dark HP and Dark HP > 2000
      switch to costume dark2
    if 2000 > Dark HP and Dark HP > 1500
      switch to costume dark3
    if 1500 > Dark HP and Dark HP > 1000
      switch to costume dark4
    if 1000 > Dark HP and Dark HP > 500
      switch to costume dark5
    if 500 > Dark HP and Dark HP > 0
      switch to costume dark6
    if 0 > Dark HP or Dark HP = 0
      switch to costume dark7
  
```

After taking a look at the Dark HP costumes, add this program. It sets the size, position, and conditions of the **Dark HP** variable when the sprite changes costumes.

Next, go to the Stage and find the **Dark HP** variable in the top-right corner. You can take your pick from one of three looks (just double-click to change it):

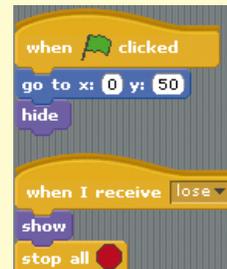
- Standard view
- Adjustable view (click and drag the ball to change a variable's value)
- Numeric view



Because we have a custom sprite, let's use the simplest view, the numeric one, to display the **Dark HP** variable.



Now add a sprite for the winning screen (Win) and another sprite for the losing screen (Lose). The winning screen gets the two programs below and shows itself only when it receives the **win** broadcast from the Dark Wizard sprite, once he's out of **Dark HP**.



The losing screen has two really similar programs. Now we're finished!

# 9

STAGE



After saving your file, give the game a try. You'll definitely want to play this one full screen. Step into Scratchy's shoes for the final battle.

## Scratchy's Challenge!!

Feel like playing the bad guy instead? Just program some movement controls for the Dark Wizard, and you'll have a two-player game. You can even add more fight moves! Give it a try!

