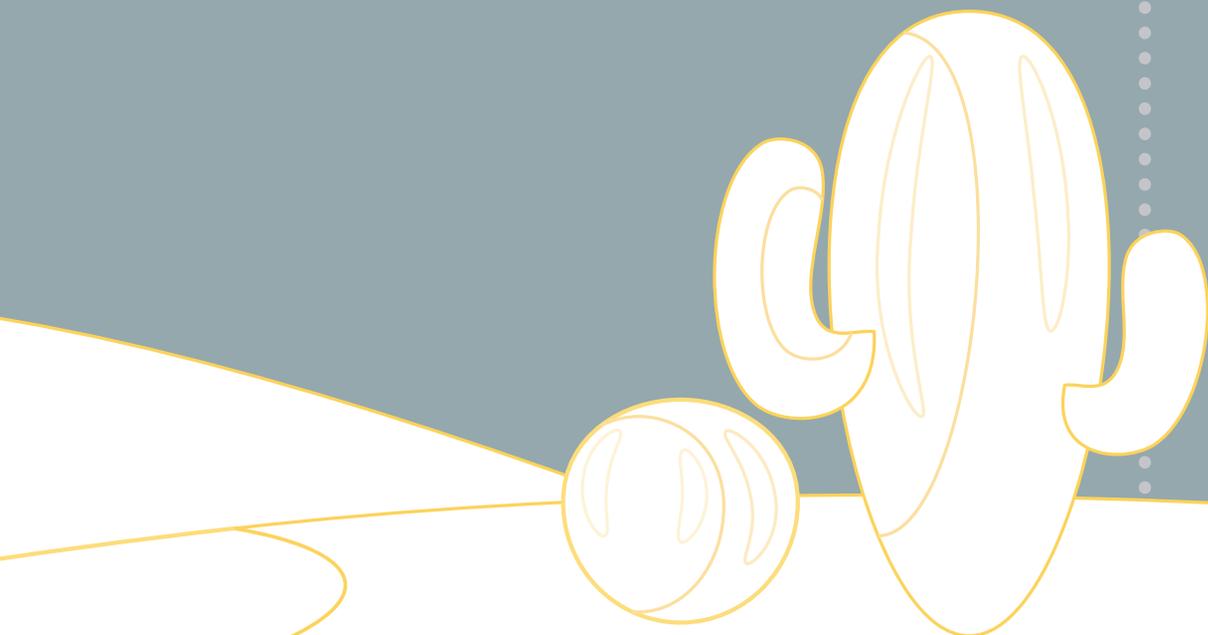


**SCRATCHY'S  
WILD RIDE**

**6**  
**STAGE**



STAGE

6



**GOAL!!!**

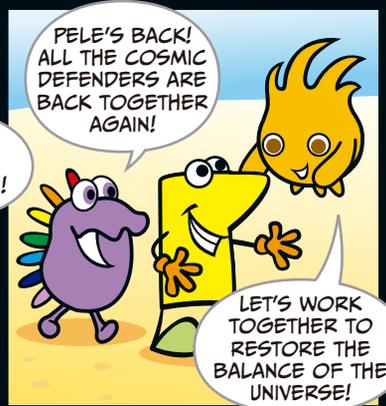
THE LOCK IS BROKEN! WHAT AWESOME SOCCER SKILLS!

BAM!



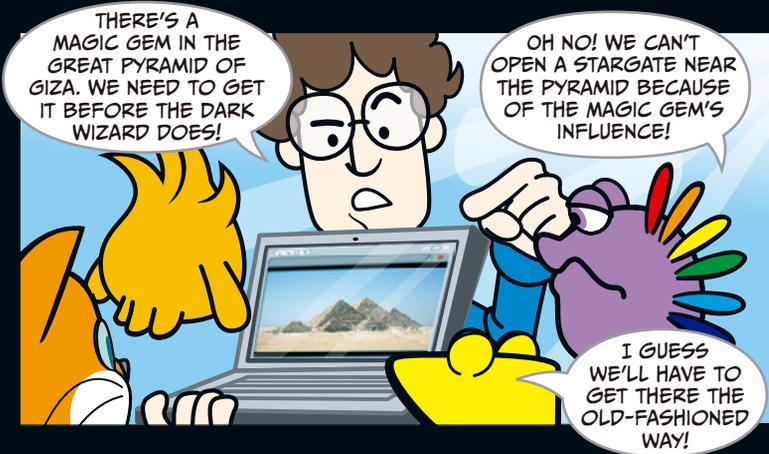
WELL... I ONLY WON BECAUSE OF EVERYONE'S HELP AND THE DIRECTIONS FROM THE SECRET MANUAL!

DON'T SELL YOURSELF SHORT, MITCH!



PELE'S BACK! ALL THE COSMIC DEFENDERS ARE BACK TOGETHER AGAIN!

LET'S WORK TOGETHER TO RESTORE THE BALANCE OF THE UNIVERSE!



THERE'S A MAGIC GEM IN THE GREAT PYRAMID OF GIZA. WE NEED TO GET IT BEFORE THE DARK WIZARD DOES!

OH NO! WE CAN'T OPEN A STARGATE NEAR THE PYRAMID BECAUSE OF THE MAGIC GEM'S INFLUENCE!

I GUESS WE'LL HAVE TO GET THERE THE OLD-FASHIONED WAY!



THE SAHARA DESERT

THE TRUCK'S READY TO GO. HOP ON, EVERYBODY!



HOLD ON TIGHT!

AHHH!



NO!! THEY TOOK MITCH!

HURRY UP AND START THE CAR! WE HAVE TO CATCH THEM!



## DESERT RALLY RACE

# STAGE

### + Chapter Focus

Learn how to create a scrolling game, program complex movements for the sprites, and make a background change over time.

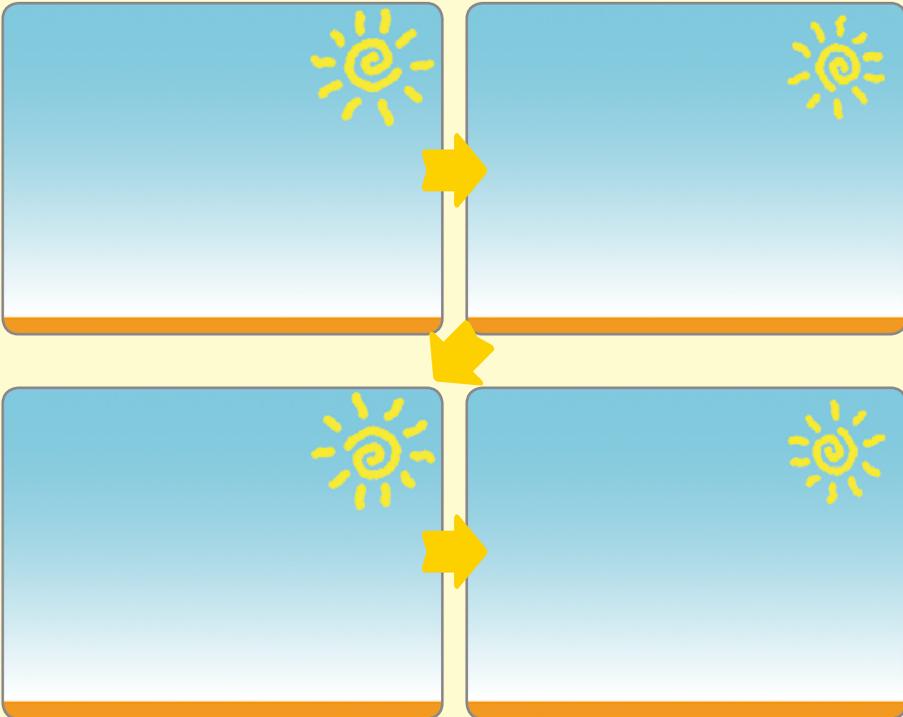
### Game

Control Scratchy's car to avoid obstacles and to run away from the Dark Minions in order to reach the Great Pyramid of Giza. Each time you crash your car, one of the Cosmic Defenders will jump out. If you crash your car four times, your car will break down!



Let's start by importing a project called *DesertRace*, which already has a bunch of sprites in it. It doesn't have any programs yet, but we'll add some soon.

First, let's look at the Stage. If you click the **Stage** in the Sprite List, you can see that we have a lot of different backgrounds.



# 6

STAGE



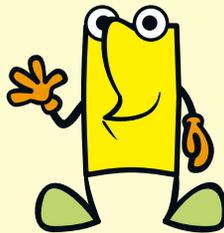
You Win!!



Backgrounds for the Stage are just like costumes for any other kind of sprite. So let's write a program that controls how they change.

Program 1 will make the background change over time in two loops, day and night. You can use the Duplicate tool to save time with the programming! This animation will give the Stage a cool look as Scratchy drives.

Program 2 will make the Stage change its background to the Win costume when the **finish** broadcast is received.



```

1 when clicked
  forever
    repeat 8
      switch to background Day_1
      wait 0.5 secs
      switch to background Day_2
      wait 0.5 secs
      switch to background Day_3
      wait 0.5 secs
      switch to background Day_4
      wait 0.5 secs
    repeat 4
      switch to background Night_1
      wait 0.5 secs
      switch to background Night_2
      wait 0.5 secs
      switch to background Night_3
      wait 0.5 secs
      switch to background Night_4
      wait 0.5 secs
      switch to background Night_5
      wait 0.5 secs
      switch to background Night_6
      wait 0.5 secs

2 when I receive finish
  switch to background Win
  stop all
  
```

# 6 STAGE

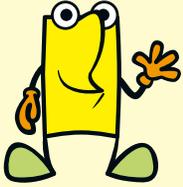
We'll also have the Stage keep track of the time in program 3. So create a variable called **Time** from the **Variables** palette. We set **Time** to 0 and then change it by 1 with each second. We'll use the **Time** variable again later.

```
3 when clicked
  set Time to 0
  forever
    wait 1 secs
    change Time by 1
```



Next, let's look at the road. Try to use the whole width of the Stage if you're drawing it!





Adding these programs to the Road1 sprite will make it appear on the screen and scroll to the left.

```
1 when green flag clicked
  set Scroll to 0
  forever loop
    change Scroll by -1

2 when green flag clicked
  go to front
  go back 1 layers
  set y to 10
  forever loop
    set x to Scroll

3 when green flag clicked
  forever loop
    if Scroll < -479
      set Scroll to 0
```

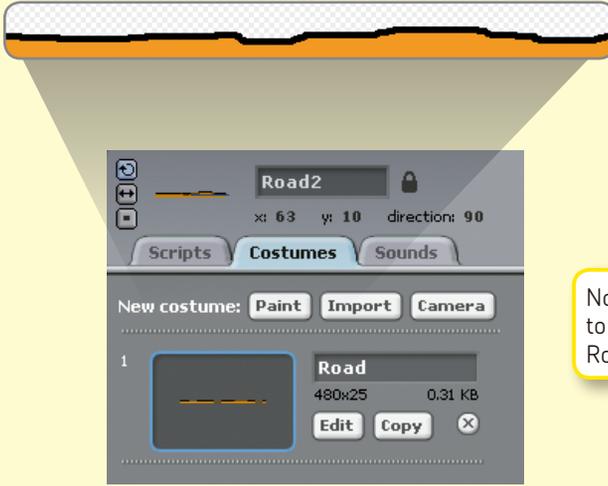
Write program **1** to make the **Scroll** variable continuously decrease by 1 (that is, **change Scroll by -1**).

Program **2** will set the road's position. Set the y coordinate to **10** so it won't move up or down, and then add **set x to Scroll** in a **forever** loop. By doing this, the road will continuously move to the left as the **Scroll** variable changes.

Program **3** will make the **Scroll** variable reset to a 0 value once it reaches a value less than -479.

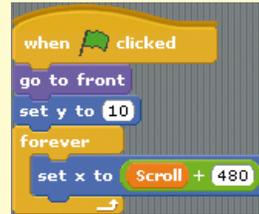
Tip: Why did we use the number -479? The width of the entire Scratch Stage is 480 pixels, so that's when it will roll off the Stage.

# 6 STAGE



Now duplicate the Road1 sprite to create a second sprite called Road2.

Add this program to use the **Scroll** variable from the first road sprite. This time, we use a trick to make Road2 follow right behind Road1. By setting the x coordinate to **Scroll + 480**, we know Road2 will always follow behind Road1. This means that the player always has a road to drive on, no matter what!



Next, add Scratchy's Car sprite.



Program 1 for the Car does a lot of work. First, it sets the costume, size, and position.

The `forever` loop holds the rest of the program. The `change y by -5` block will pull the car down, giving it gravity. The `if touching color` block makes the car bounce up whenever it touches the black part of the road, making it seem like they're driving on a very bumpy road. The `if key up arrow pressed?` block will broadcast `jump` and then wait.

```
1 when clicked
  switch to costume Car_1
  set size to 60 %
  go to front
  go to x: -150 y: -105
  forever
    change y by -5
    if touching color [black] and y position < -105
      change y by 10
      wait 0.05 secs
    if key up arrow pressed?
      broadcast jump and wait

2 when I receive jump
  repeat 15
    change y by 12
  repeat until touching color [black] and y position < -105
    change y by -5
```

Program 2 makes the car "listen" for the `jump` broadcast and makes the car jump up.

The `broadcast jump and wait` block in program 1 will temporarily stop the first program so the second program can run.

# 6 STAGE

Now add program 3 so that the car can move left and right.

```
3 when clicked
  forever
    if key right arrow pressed?
      move 5 steps
    if key left arrow pressed?
      move -5 steps
```

In program 4, we add some speech bubbles as instructions for the player.

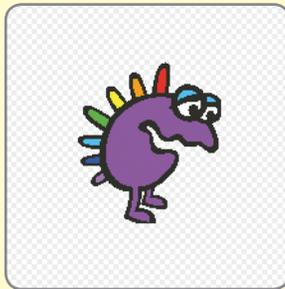
In program 5, we create a new variable called **Life**. When the **Life** value is less than 1, we'll set the car's costume to Boom! and then end the game with the **stop all** command.

```
4 when clicked
  say Press L or R keys to move, UP key to jump! for 2 secs
  say Avoid the obstacles! for 2 secs

5 when clicked
  set Life to 4
  wait 1 secs
  forever
    if Life < 1
      switch to costume Boom!
      stop all
```

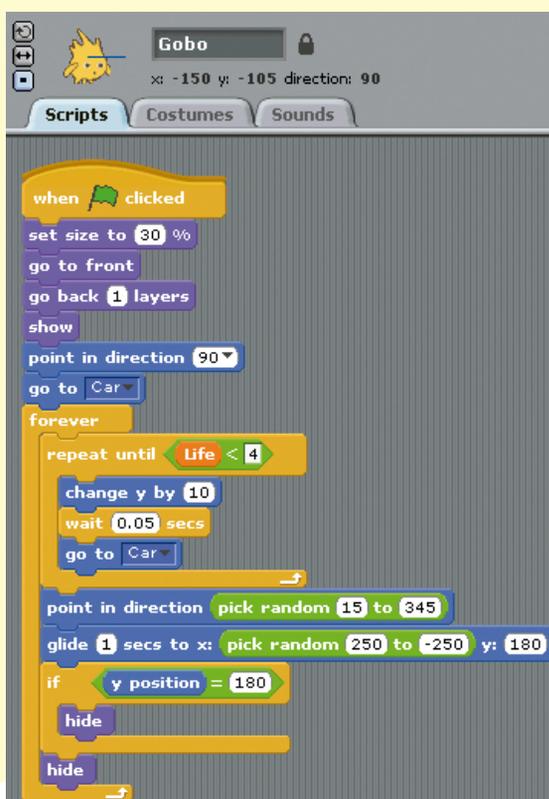
Once you're finished with the Car sprite's programming, you can add some passengers—the Cosmic Defenders!

Add these three sprites, and then drag them onto the car. Gobo is at the back, Fabu is in the middle, and Pele is in the front. It's okay if they overlap a bit.

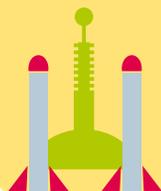


# 6 STAGE

Write this program for Gobo. It sets his size and position and uses the `go to` block so he'll always follow the Car sprite. Once the variable `Life` drops to less than 4 (`Life < 4`), he'll shoot to a random area. When he touches the top of the screen (`y position = 180`), we make him disappear by using the `hide` block.



```
when green flag clicked
  set size to 30 %
  go to front
  go back 1 layers
  show
  point in direction 90
  go to Car
  forever
    repeat until Life < 4
      change y by 10
      wait 0.05 secs
      go to Car
    point in direction pick random 15 to 345
    glide 1 secs to x: pick random 250 to -250 y: 180
    if y position = 180
      hide
    hide
```



```

when clicked
  set size to 30 %
  go to front
  go back 2 layers
  show
  point in direction 90
  go to Car
  forever
    repeat until Life < 3
      change y by 10
      wait 0.05 secs
      go to Car
    point in direction pick random 15 to 345
    glide 1 secs to x: pick random 250 to -250 y: 180
    if y position = 180
      hide
    hide
  
```

Drag and copy Gobo's program onto Fabu in the Sprite List. You'll need to change only a few things. Most important, change the `repeat until` block to `Life < 3`, so Fabu will bounce out at a different time.

Do the same thing for Pele, but change the `Life` value to `2`. Because Pele's sprite is a little bigger than the others, we also set his size to `25%`.

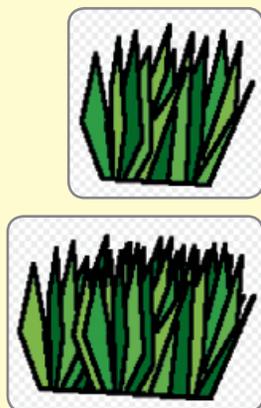


```

when clicked
  set size to 25 %
  go to front
  go back 3 layers
  show
  point in direction 90
  go to Car
  forever
    repeat until Life < 2
      change y by 10
      wait 0.05 secs
      go to Car
    point in direction pick random 15 to 345
    glide 1 secs to x: pick random 250 to -250 y: 180
    if y position = 180
      hide
    hide
  
```

# 6 STAGE

Now we can add the programming for the obstacles. First, let's take a look at the thorny and dangerous Bush! It has two costumes.



And then write these three programs:

Program 1 controls when the bush appears and makes sure it moves with the road. Once it touches the left edge of the screen, it'll disappear and switch to the next bush costume.

Program 2 programs the Car to change Life by -1 (that is, lose one life) whenever it touches an obstacle. Notice how we programmed the computer to check if the player still has enough Life value left using the and and not blocks.

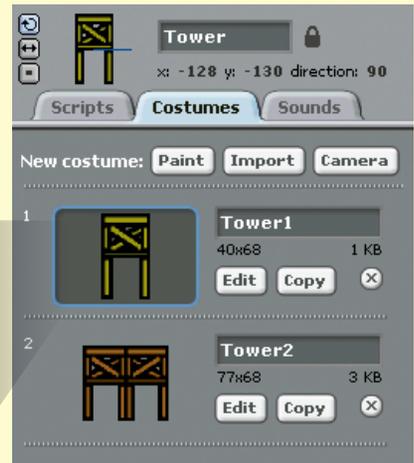
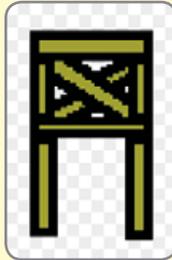
And program 3 makes the bush disappear once it receives the finish signal, which ends the game.

```
1 when green flag clicked
  switch to costume Bush1
  hide
  forever
    wait 8 secs
    go to x: 230 y: -130
    show
    repeat until x position < -230
      change x by -1
    hide
    next costume

2 when green flag clicked
  wait 1 secs
  forever
    if touching Car? and not Life = 0
      change Life by -1
      wait 6 secs

3 when I receive finish
  hide
```

Now let's look at the Tower sprite, which also has two costumes. This obstacle will be tough to jump!



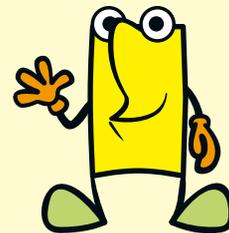
```

when green flag clicked
  switch to costume Tower1
  hide
  forever
    wait 18 secs
    go to x: 230 y: -130
    show
    repeat until x position < -230
      change x by -1
    hide
    next costume

when green flag clicked
  wait 1 secs
  forever
    if touching Car? and not Life = 0
      change Life by -1
      wait 6 secs

when I receive finish
  hide
  
```

We can once again copy the program we created for the bushes. Edit the costume name and the time it appears, and you're good to go!



# 6 STAGE

Create a new sprite for Legs, the evil octopus Dark Minion. But don't you think it's a little boring just to have one image for him?



Why don't we try animating him?

In the Paint Editor, use the **Select** tool to grab the end of his tentacle.



Next, click this button to flip his arm up and then drag it back into place.



Do the same for his other tentacles, and there you go—a new look!



Tip: Editing existing costumes is an easy way to animate a character without having to redraw it. The Select and Rotate tools let you quickly change the position of a sprite's arms and legs.

Now let's get back to programming! Program 1 makes Legs switch between his two costumes in a `forever` loop. Program 2 makes him `hide` when he receives the `finish` broadcast.

```

1 when green flag clicked
  forever loop
    wait 0.3 secs
    next costume

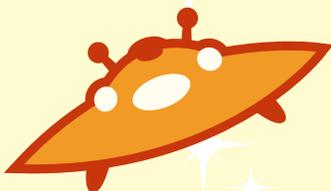
2 when I receive finish
  hide
  
```

Programs 3 and 4 control Legs's movements and make him an unpredictable obstacle for Scratch's car.

```

3 when green flag clicked
  set size to 50 %
  hide
  forever loop
    wait pick random 15 to 20 secs
    go to x: 230 y: 70
    show
    repeat until x position < -230
      change x by -3
    hide

4 when green flag clicked
  forever loop
    repeat 10
      change y by -5
      wait 0.05 secs
    repeat 10
      change y by 5
      wait 0.05 secs
  
```



# 6 STAGE

Lastly, program 5 for Legs adds a condition that will subtract life points from the Life variable, just as with the Bush and Tower obstacles.

```
5 when clicked
  wait 1 secs
  forever
    if touching Car? and not Life = 0
      change Life by -1
      wait 6 secs
```

And now we'll move on to the final sprite of the game: Egypt's Great Pyramid of Giza!



Let's start with this photo:





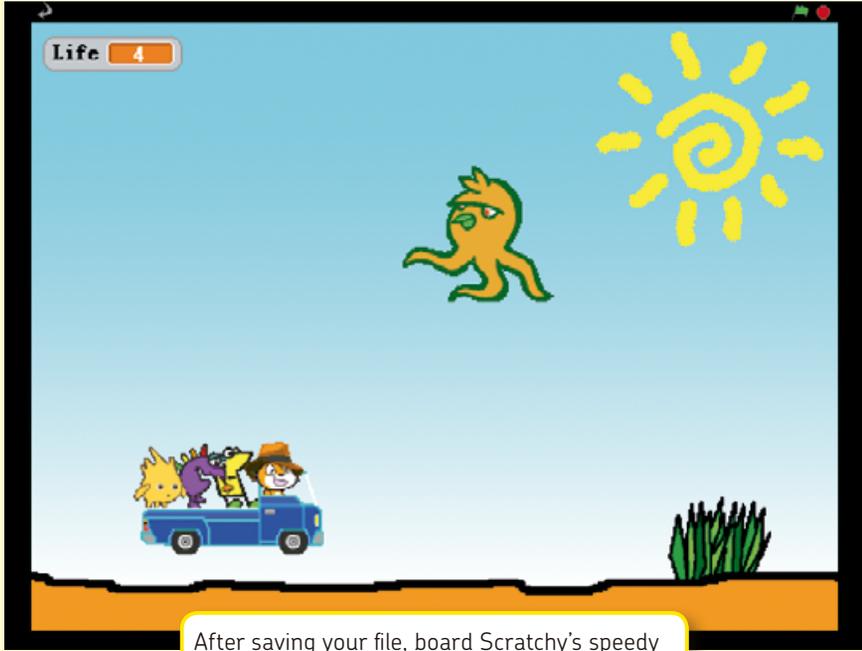
By adding this sprite, we'll make it look like Scratchy is "arriving" at the pyramids. Edit the Giza costume so that the cool backgrounds will show through and so that the bottom matches the orange of the road. Now we can make the photo fit into our existing game.



Write a script so that the pyramid slowly appears from the right, after the game is run for 60 seconds. Once it reaches the center of the screen ( $x$  position = 0), it broadcasts the **finish** signal. When the other sprites receive this signal, the game ends.

```
when green flag clicked
  switch to costume Giza
  set size to 70 %
  hide
  wait until Time = 60
  show
  go to x: 350 y: -75
  repeat until x position = 0
    change x by -1
  broadcast finish and wait
```

# 6 STAGE



After saving your file, board Scratchy's speedy car and drive into the Sahara Desert to begin your wild adventure!

## Scratchy's Challenge!!

Can you use these programs to create another scrolling game? Give it try! (Tip: The height of Scratch's screen is 360 pixels.) Make the game even more challenging by having the car go really fast!

