# Avoid the Thieves and Bad Bananas!

#

In this lesson, you’ll extend the monkey game from last week to make it a little harder. Now, the monkey will have to avoid two types of obstacles: rotten falling bananas and thief monkeys.

In this game, some of the bananas falling from the tree will be bad; they will be red instead of yellow. Your monkey needs to avoid those rotten bananas; if one of those rotten bananas touches him, all of his current bananas are spoiled and his banana count goes to zero. There is also now a baby thief monkey sliding across the bottom of the Stage; if the thief touches the monkey, then the thief steals all of the bananas! The monkey should jump over the thief to avoid it.

We will continue to explore **if statements** and **variables**. Remember, an **if** statement asks a question and then executes some code **if and only if** the answer to the question was true. A **variable** is a way for your program to remember something. In this game, the number of points (or, the number of bananas the monkey has caught) is stored in a variable.

This lesson has four steps:

1. Make rotten bananas fall from the tree (from random locations and moving at random speeds). If the rotten bananas touch the monkey, then take away all of his points.
2. Make a thief that slowly slides across the bottom of the stage. If the thief touches the monkey, then the thief takes all of the Monkey’s bananas.
3. Make the monkey jump when the user presses the up arrow key.
4. Make instructions.

## Make Rotten Bananas Fall

Last week we created Scripts for good yellow bananas. The bananas would appear at random locations and fall to the ground. If the bananas touched the monkey, then the number of Banana points was increased by 1. If the bananas touched the edge of the Stage, the bananas appeared at a new random location and started falling again.

We now want to make rotten red bananas. The rotten red bananas should behave just the same as the good bananas, EXCEPT if they touch the monkey, the monkey should lose all of his bananas.

To do this, we will **duplicate** the yellow banana Sprite and make a few changes. The first change will be to change their color. You can do this by **Editing** the **Costume** and using the paint tool.

Now we need to change the banana **Scripts** so that if the bananas touch the monkey, the monkey loses all of his bananas. How would you change the previous code (shown below) to make this happen?



You will want to replace the block **change Bananas by 1** with the block **set Bananas to 0.**

Try this out. You could make rotten bananas do different things than ruin all of the caught bananas; for example, you could just make the rotten bananas spoil 1 or 2 bananas. How would you do this? (Advanced Issue: How can make sure you don’t make the Monkey have a negative number of bananas?)

You might find that it is difficult seeing when the rotten bananas touch the monkey. To make this more obvious, you could make the rotten bananas say “Ha ha” or make a noise or some effect when they touch the Monkey.

You can have as many rotten bananas as you like by duplicating that Sprite.

## Make Thief Monkeys

Let’s make the game much harder by adding a thief monkey that slides across the bottom of the stage.

While the game is playing, we want the thief to forever move to the left or the right. When it gets to the edge of the screen, it should bounce back the other direction. How can you build a Script that tells the thief monkey how to do that?



Now, if the thief touches the Monkey, we want the Monkey’s points to go away. How can you tell the thief to do this?



Now, we want the thief to be able to collect bananas too. We want the game to remember how many bananas the thief has collected so far. So, what do we need to add to the game? A new **Variable**; let’s call this one **Stolen Bananas!**  When the thief touches the Monkey, how can we give the Monkey’s points to the thief?

You can use the block **set Stolen Bananas to Bananas!** Note that you can set a variable to any number or to any other variable. You can drag the **variable** named **Bananas** into the **set** block. Note that this block must come before the block where you **set Bananas to 0.** What happens if you reverse the order?

Finally, Remember to initialize your new variable! Where should you set the initial value of this variable? *(As one of the scripts with the Stage.)*

You might want the Thief to say something to the Monkey when he steals the bananas to make it more obvious this has happened. Have the thief say something like “Thank You!” To make the game even more challenging, you can **duplicate** the thieves too.

## Jump over the Thief Monkeys

We need to have some way for the Monkey to avoid the thieves. When the user presses the Up Arrow Key, we want to tell the Monkey jump up.

When the monkey jumps, does it change its **x** or its **y** position? *Y*. When it jumps up, should y get bigger or smaller? *Bigger*. When it falls back down, should y get bigger or smaller? *Smaller.* Lets try specifying this with the **change y by** block. Let’s make him jump up (and back down) about 100 pixels.

When you run these scripts, how does the game look? The jumping monkey has a couple of problems. First, the monkey needs to pause at the top of his jump. You can use the **wait** block for this. Second, you want the user to be able to see some of the steps of the jump. So, you want to tell the monkey to change **y** by just a little bit at a time. One way you could do this is by replacing the block **change y by 100**  with five blocks that all say **change y by 20**. This will tell the monkey to do the right thing, but it is a pain for us to drag all of those blocks.

A much easier way to tell the monkey what to do is to use the **repeat** block. You can tell it to **repeat 5 times** the block **change y by 20**. And then, repeat 5 times the block **change y by -20.** Now, if you think the monkey is still skipping too much, you can easily change the code (e.g., **to repeat 10 times** the block **change y by 10** ; as long as it still jumps a total of 100 pixels).

You might want to resize your Thief Sprites to make it easier or harder to jump over them.

## Add Instructions

The game is now getting pretty complicated. You probably should add an Instruction Sprite that tells the user what they are supposed to do. Write a script to tell the Instruction Sprite to disappear when the user starts playing the game.

# Things to Remember

There are many ways to move sprites. In this lesson, you made a Sprite jump up and down with the block **change y by**. The **repeat loop** repeats the blocks inside the loop the specified number of times.

An **if** statement asks a question and then executes some code **if and only if** the answer to the question was true. With an if statement, your program can ask questions about what is happening and then react to the current situation. There are many interesting questions your Sprites can ask; most are under the **Sensing** Menu.

A **variable** is a way for your program to remember something. A variable holds a value for you; this value can change (it can vary) as your program runs. You should name your variables with something descriptive so you can remember what it means. You can show the value of a variable by checking the box next to its name. Variables should always be set (or **initialized**) to some value when the program starts (that is, when the Green Flag is Clicked). A logical place to initialize variables that are shared across all Sprites is with the Stage.

You often want to make new Sprites that behave similarly to other Sprites. To do this, **duplicate** the existing Sprite (click the right mouse button over the existing Sprite) and change the Scripts (or Costumes) of the new Sprite.

# Final Scripts: Monkey



# Final Scripts: Bad Bananas

#

# Final Scripts: Thief



# Final Scripts: Stage



# Final Scripts: Instructions

