Scratch Programming Lesson 12
Mini Mario Game Part 3 - Adding Game Rules

In this lesson, I will show you how to create additional sprites and scripts to apply the game rules. We will add scripts based on Sprite Interaction Table. Moreover, we will create random motion scripts for Mario's enemies.

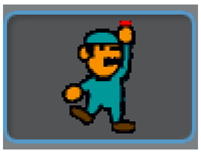
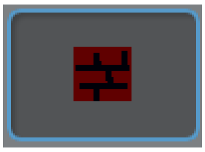
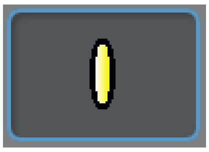
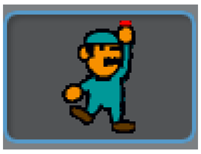
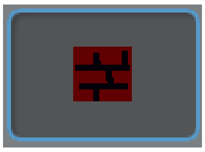
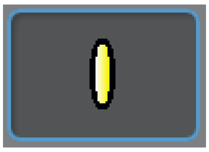
1. Create Scripts based on Sprite Interaction Rules

1.1 Rule: How Mario can Score

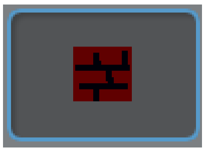
The rule for Mario to score is as followed:

> If Mario hand hits Brick, then Brick would crack. When the Brick cracks, Coin would start spinning above it. When Coin disappears, Mario gets one point.

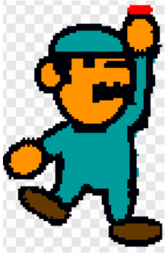
The Sprite Interaction Table below lists all relevant interactions for this rule.

Sprite1	**Sprite2**	**Interaction Rule**
Mario 	Brick 	If Mario hand hits Brick, then Brick would crack.
Coin 	Mario 	Each time a coin appears, Mario scores 1 point.
Brick 	Coin 	When Brick cracks, Coin would show spinning above Brick

You may have noticed that the JUMPING UP costume has a bit of RED on the fist. It's not for decoration. The RED bit is added so that the Brick sprite can know when Mario's fist has hit it. I call this color (RED in this case) a *sensitive color* for the Brick, and the Brick is a *color-sensitive sprite* of the RED color. A sensitive color of a color-sensitive sprite is a color which such a sprite is sensitive to and responds to when touched. A good sensitive color for a color-sensitive sprite is a color that is not yet used by any sprite in the project. In the case of Brick, which contains burgundy and black, it can have all other colors as sensitive colors. I just picked RED. You can certainly use BLUE.



The Brick has burgundy and black.

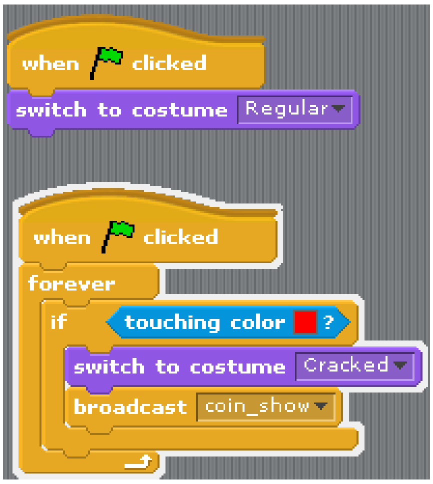


I picked RED to be one sensitive color.

You can certainly use BLUE.

For Brick Sprite:

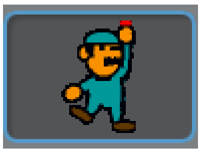
To simulate Brick cracking when hit by Mario, add the script as shown at right to Brick so that it would show when game starts and would cracked when touched by RED, one of Brick's sensitive color. Also, the Brick will send out a message, *coin_show*, so that Coin knows when to appear.



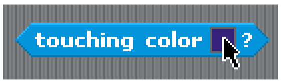
```
when 🏁 clicked
switch to costume Regular

when 🏁 clicked
forever
  if touching color 🟥 ?
    switch to costume Cracked
    broadcast coin_show
```

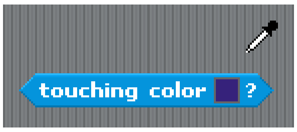
NOTE: To copy the color of the RED from Mario's fist, first select Mario's JUMPING UP costume so Mario shows up on Stage in JUMPING UP costume.



Select Brick's script panel and drag out a "touching color … " block. Click the color box (arrow cursor would turn into an eyedropper) and move the eyedropper

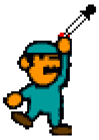


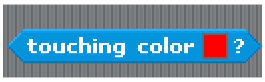
```
touching color ■ ?
```



```
touching color ■ ?
```

Move the eyedropper to the Stage to copy the RED on Mario’s fist. The “touching color …” block would change to reflect the color just copied.

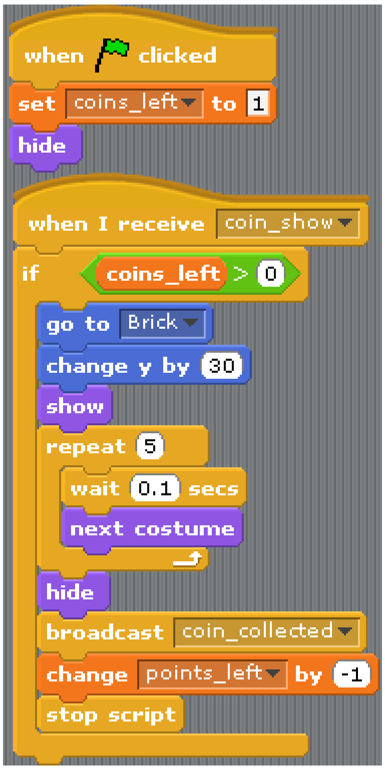




```
touching color 🟥 ?
```

For Coin Sprite:

Create a for-this-sprite-only variable called coins_left and set it to 1 when game starts. Hide Coin when game starts and check if coins_left is greater than 0 when it receives a *coin_show* message. If YES, then show Coin, spin it (simulated by changing costumes), hide it, send out a *coin_collected* message, decrease coins_left by 1, and finally stop script. If NO (points_left is less or equal to 0), then just do nothing.

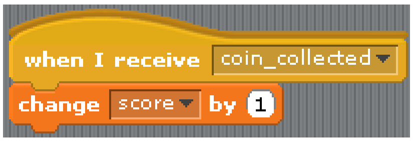


```
when ⚑ clicked
set coins_left to 1
hide

when I receive coin_show
if coins_left > 0
    go to Brick
    change y by 30
    show
    repeat 5
        wait 0.1 secs
        next costume
    hide
    broadcast coin_collected
    change points_left by -1
    stop script
```

For Mario Sprite:

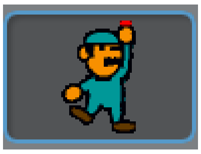
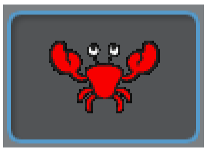
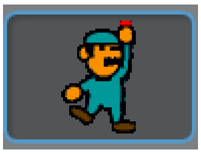
Add the script shown at right to Mario’s script so that the score would go up one when he collects a coin (when received a coin_collected message).



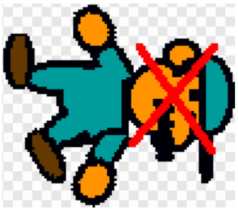
```
when I receive coin_collected
change score by 1
```

1.2 Rule: If Mario is attached or touched by a Bat or a Crab, he would die.

The Sprite Interaction Table below lists all relevant interactions for this rule.

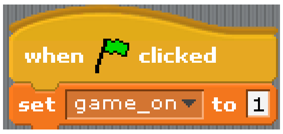
Sprite1	Sprite2	Interaction Rule
Mario 	Crab 	If Mario touches Crab, he would die.
Mario 	Bat	If Mario touches Bat, he would die.

First add a new costume call *Dead*.



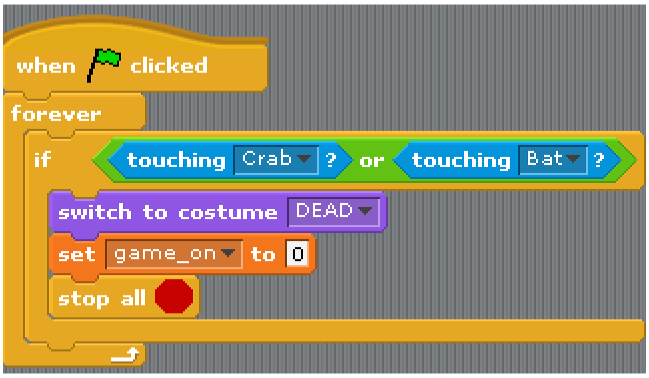
For Mario Sprite:

Create a variable called game_on and set it to 1 when game starts



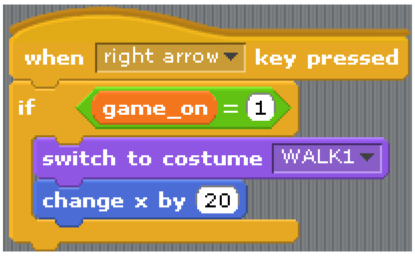
```
when ⚑ clicked
set game_on ▾ to 1
```

Add the script shown at right to Mario so that he would die when touched by a Crab or a Bat. To die, Mario switches costume to DEAD, sets game_on to 0, and then stop all scripts.



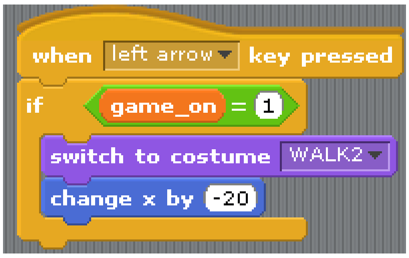
```
when ⚑ clicked
forever
  if < touching Crab ▾ ? or touching Bat ▾ ? >
    switch to costume DEAD ▾
    set game_on ▾ to 0
    stop all ●
```

Modify the script to move right so that Mario responds to right arrow key click only when game_on variable is 1.



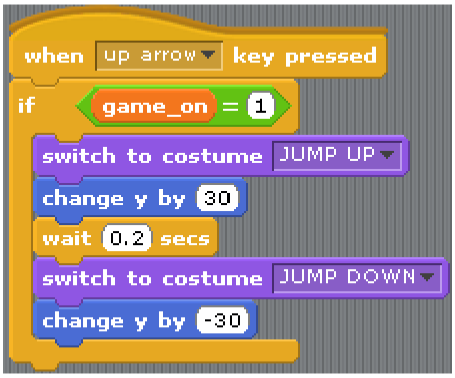
```
when right arrow key pressed
if game_on = 1
    switch to costume WALK1
    change x by 20
```

Modify the script to move left so that Mario responds to left arrow key click only when game_on variable is 1.



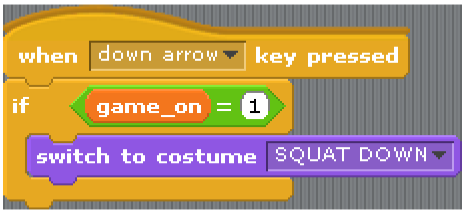
```
when left arrow key pressed
if game_on = 1
    switch to costume WALK2
    change x by -20
```

Modify the script to jump so that Mario responds to up arrow key click only when game_on variable is 1.



```
when up arrow key pressed
if game_on = 1
    switch to costume JUMP UP
    change y by 30
    wait 0.2 secs
    switch to costume JUMP DOWN
    change y by -30
```

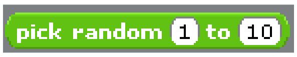
Modify the script to squat so that Mario responds to down arrow key click only when game_on variable is 1.



```
when down arrow key pressed
if game_on = 1
    switch to costume SQUAT DOWN
```

2. Create Motion Scripts for Mario's Enemies

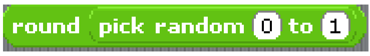
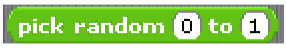
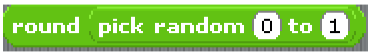
In this section, we will add script to Mario's enemies (Crab and Bat for now). We want these enemies to move randomly. To do so, we will be using the "pick random X to Y" from Math Tool Kit.



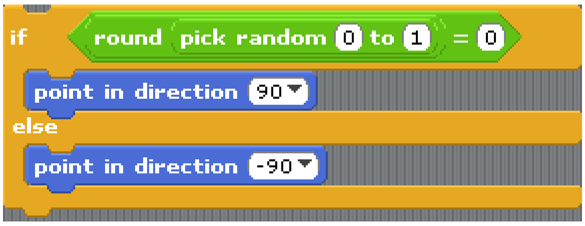
```
pick random 1 to 10
```

2.1 Rule: Crab Randomly Crawls to Left and Right

Since a real Crab can only move side to side, but not up and down, we will make Crab move toward Mario, but with some randomness in its direction when deciding to turn left or right.

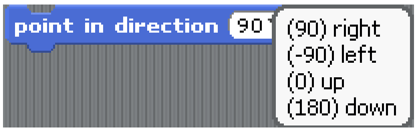
To do so, we need a way to pick left or right randomly. Create a variable named *dir* and set it to the value of  round pick random 0 to 1. The  pick random 0 to 1 block returns a random value between 0 and 1. And the  round pick random 0 to 1 block returns the round number of this random value. If the random number is smaller than 0.5, then the result would be 0, but if the random number returned is greater or equal to 0.5, then the result would be 1. This way, there is 50% chance the result would be 1, and 50% chance the result would be 0.

To use the result to determine the direction, use the "if … else …" block.



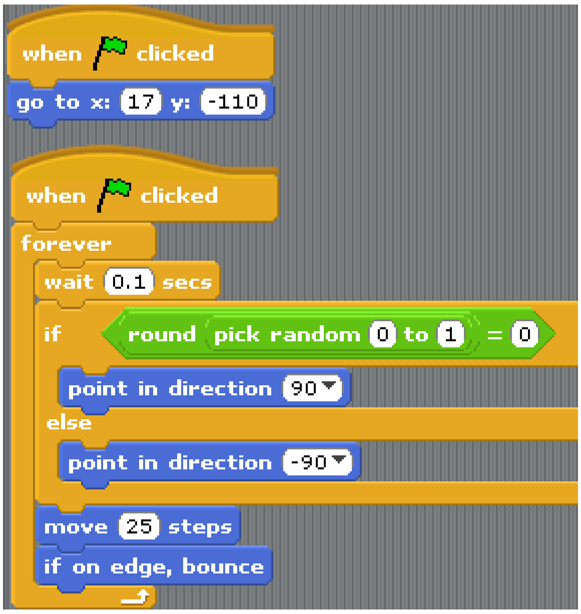
```
if round pick random 0 to 1 = 0
  point in direction 90
else
  point in direction -90
```

Remember that in Scratch, direction 90 is right -90 is left.



point in direction 90

Let's put Crab's script together so that it moves randomly to left and to right, and that it bounces if it hit the edge.

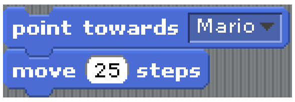


```
when ⚑ clicked
go to x: 17 y: -110

when ⚑ clicked
forever
    wait 0.1 secs
    if round pick random 0 to 1 = 0
        point in direction 90▾
    else
        point in direction -90▾
    move 25 steps
    if on edge, bounce
```

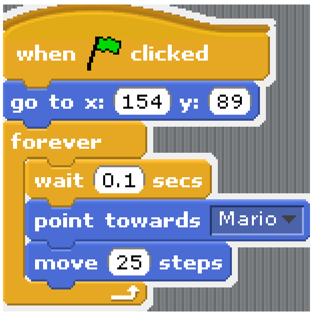
2.2 Rule: Bat Flies Toward Mario, with Some Randomness

Unlike Crab, Bat flies so it should move in all four directions. But there's much space on Stage so Bat may never touch Mario if it just flies randomly. What we want is to have Bat fly toward Mario but randomly adjusts its direction. To have it move toward Mario, use "point towards X" block and "move X steps" block.



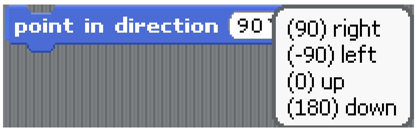
```
point towards Mario▾
move 25 steps
```

To keep flying toward Mario when game starts, expand the script to look like this:



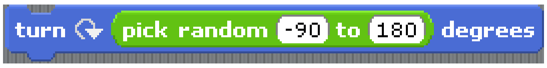
```
when ⚑ clicked
go to x: 154 y: 89
forever
    wait 0.1 secs
    point towards Mario▾
    move 25 steps
```

If you try it now, Bat would shoot straight toward Mario and the game will end before you have time to move Mario! What we need to add is a bit "wondering" by adding directional randomness. Just like Crab, Bat should randomly pick a direction to turn, but unlike Crab, Bat should pick a direction from -90 to 180, the whole range of rotation in Scratch.



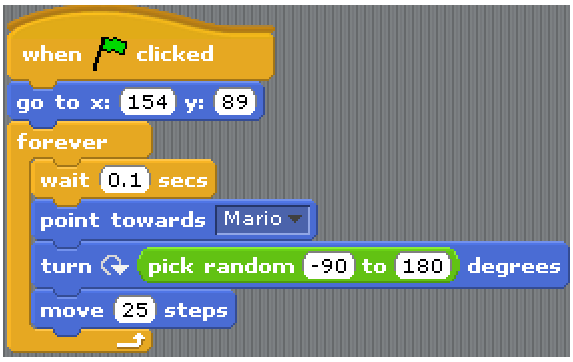
```
point in direction 90
(90) right
(-90) left
(0) up
(180) down
```

To do so, place this block immediately below “Move towards Mario”:



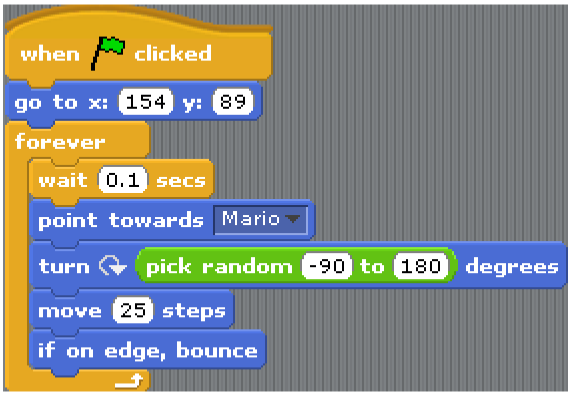
```
turn ↻ pick random -90 to 180 degrees
```

The updated script would look like this:



```
when ⚑ clicked
go to x: 154 y: 89
forever
    wait 0.1 secs
    point towards Mario
    turn ↻ pick random -90 to 180 degrees
    move 25 steps
```

Don’t forget to add the “if edge, then bounce” block. So the final script for Fly motion should be:



```
when ⚑ clicked
go to x: 154 y: 89
forever
    wait 0.1 secs
    point towards Mario
    turn ↻ pick random -90 to 180 degrees
    move 25 steps
    if on edge, bounce
```

!!TEST TIME!!

Now we are ready to test! Click Green Flag to start. Crab should crawl side to side, and Bat should fly in zig-zag motion. Try moving Mario under Brick and click up arrow key to jump to hit it. Coin should appear above Brick and disappear shortly after. The score would go up by 1. Also try move Mario toward Bat or Crab to “commit suicide”; when Mario touches either Bat or Crab, all should stop.