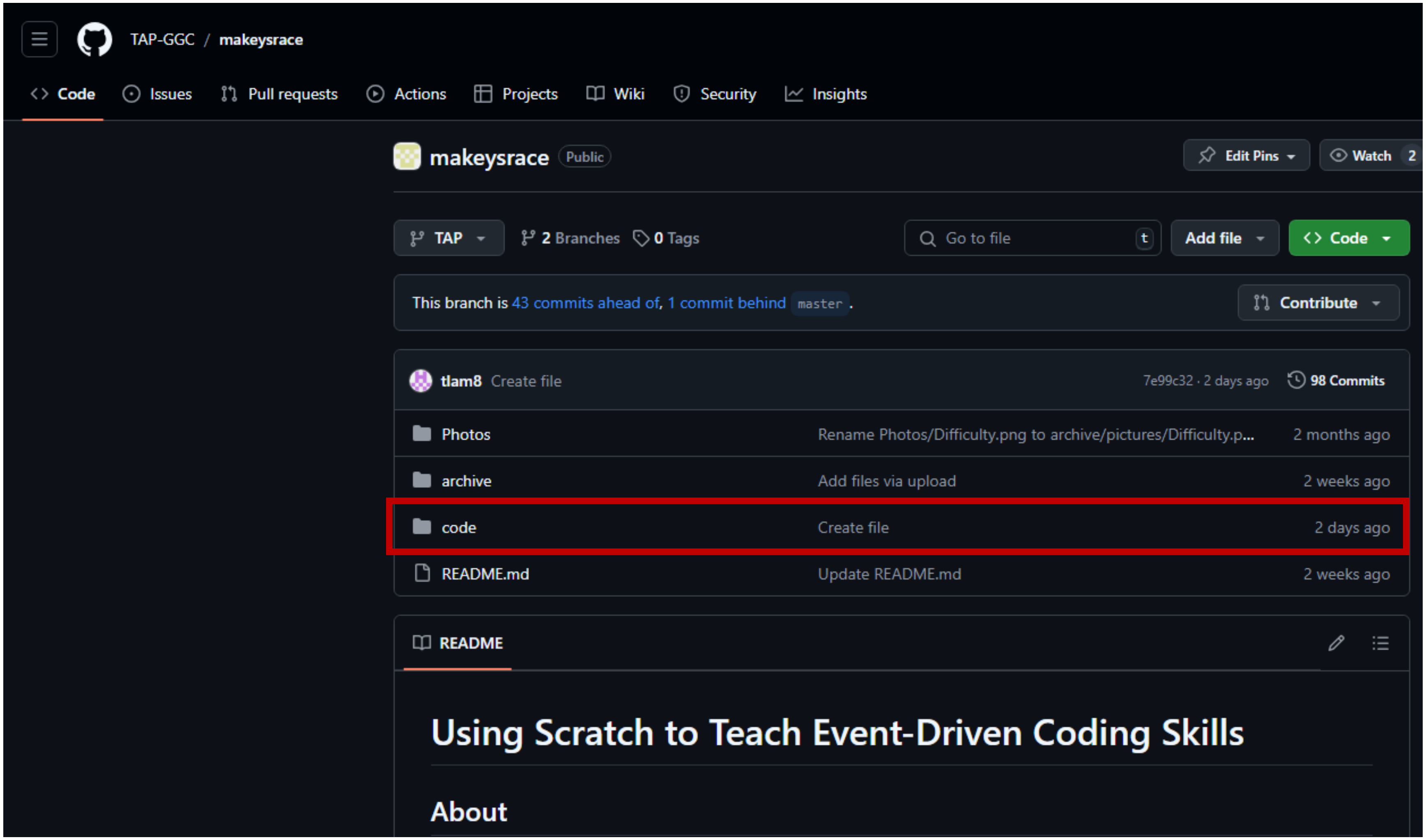
Using Scratch to Teach Event-Driven Coding Skills

Step-by-Step

https://tapggc.org/

Start by going to https://github.com/TAP-GGC/makeysrace. Then click on the code folder and select the incomplete code. This will contain all 3 levels of the game.



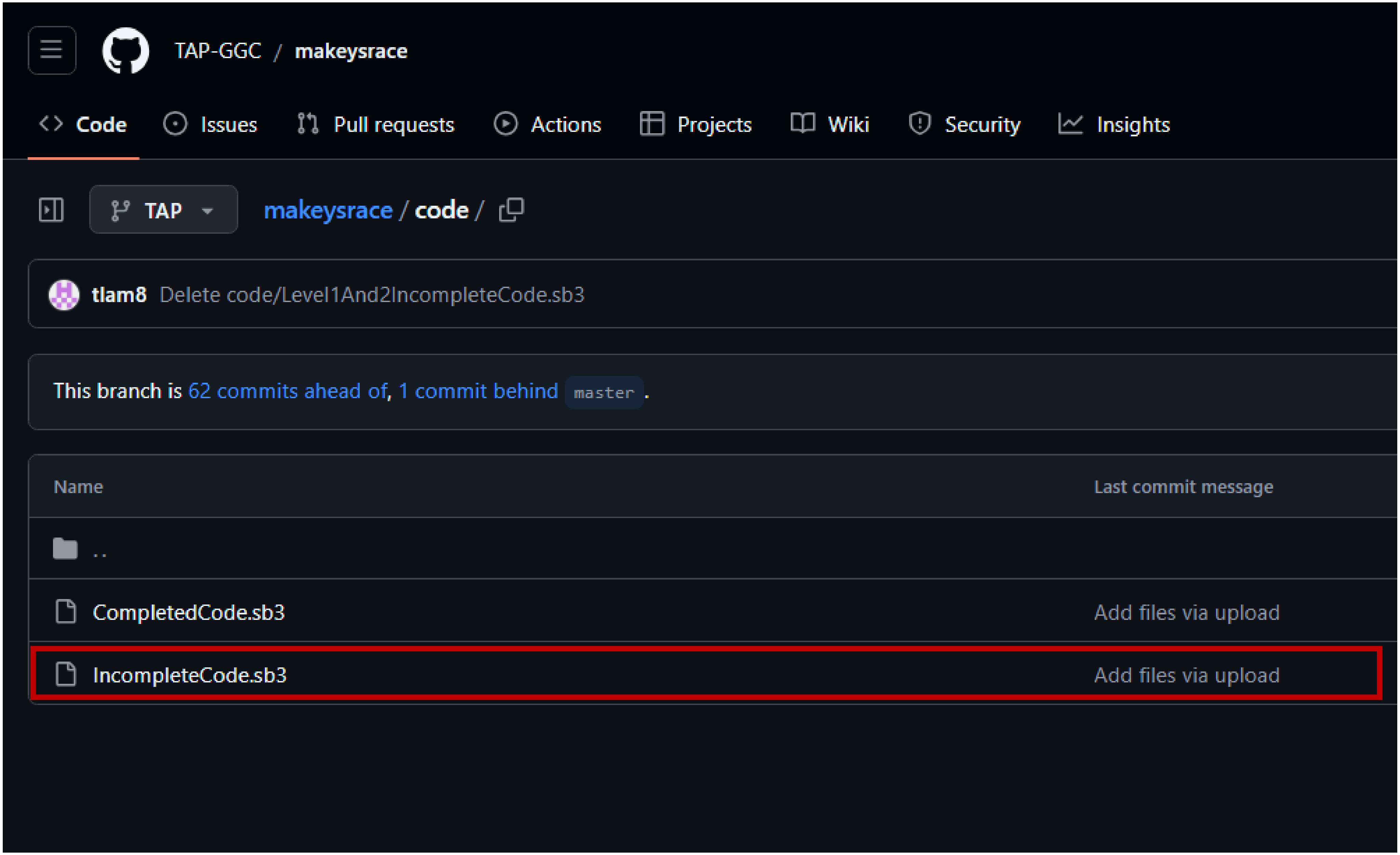
This branch is 43 commits ahead of, 1 commit behind `master`.

tlam8 Create file 7e99c32 · 2 days ago

Photos	Rename Photos/Difficulty.png to archive/pictures/Difficulty.p...	2 months ago
archive	Add files via upload	2 weeks ago
code	Create file	2 days ago
README.md	Update README.md	2 weeks ago

Using Scratch to Teach Event-Driven Coding Skills

About

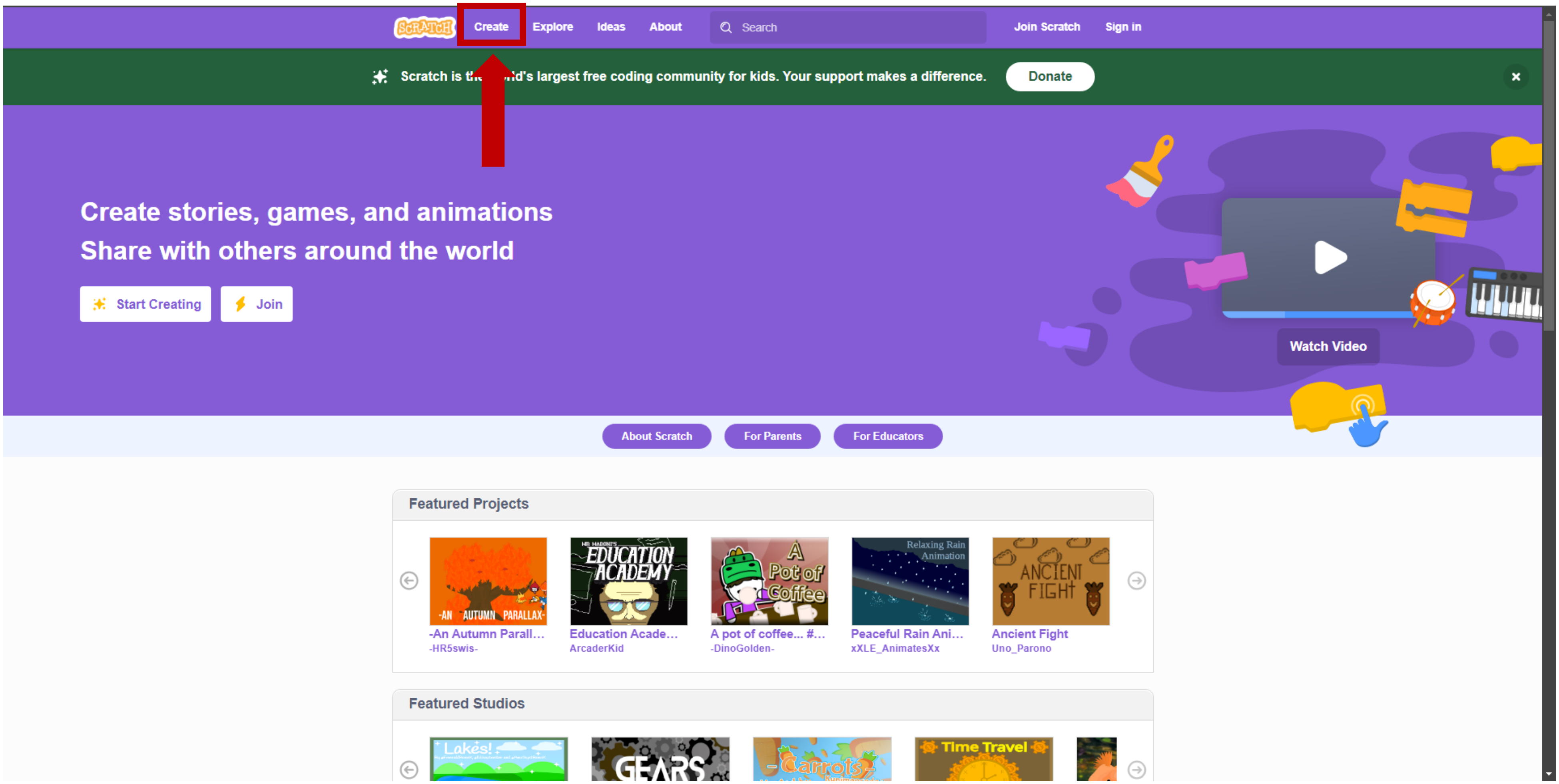


tlam8 Delete code/Level1And2IncompleteCode.sb3

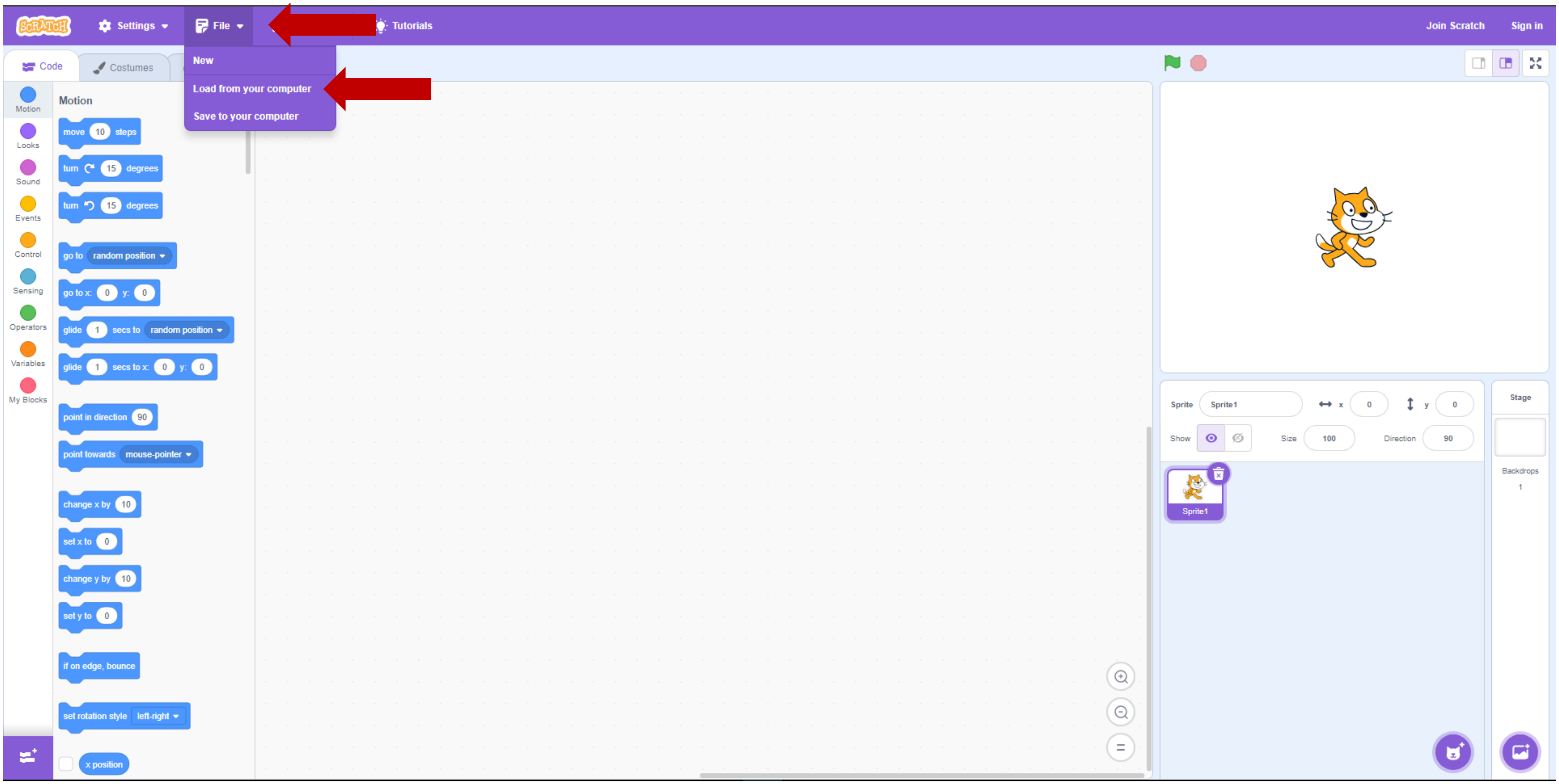
This branch is 62 commits ahead of, 1 commit behind `master`.

Name	Last commit message
..	
CompletedCode.sb3	Add files via upload
IncompleteCode.sb3	Add files via upload

After you download the file, go to https://scratch.mit.edu/ and then click on create.

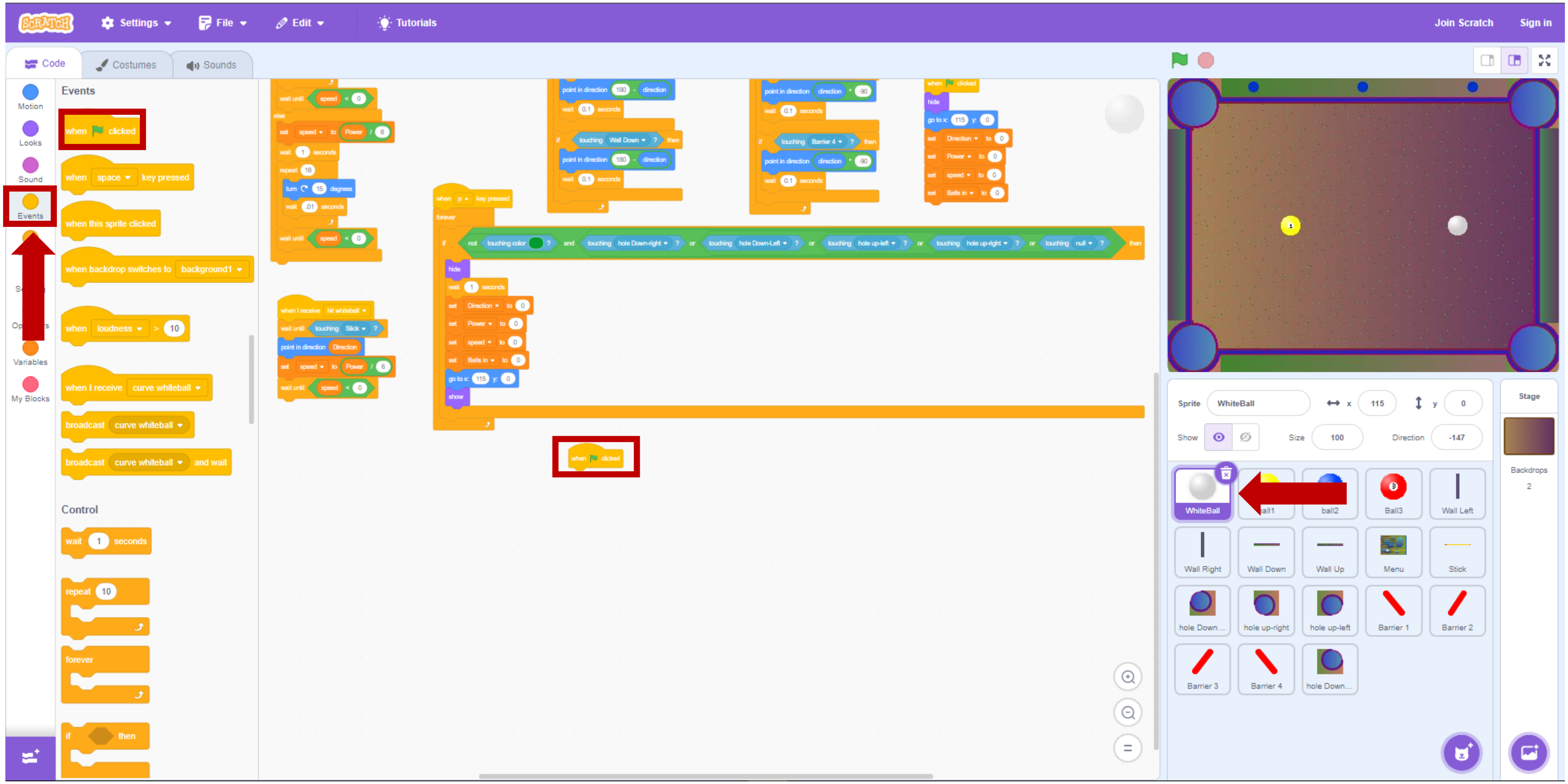


Now click on “file” and then “Load from your computer.” Then select the file that was just downloaded from GitHub. It should be called “IncompleteCode.sb3”.

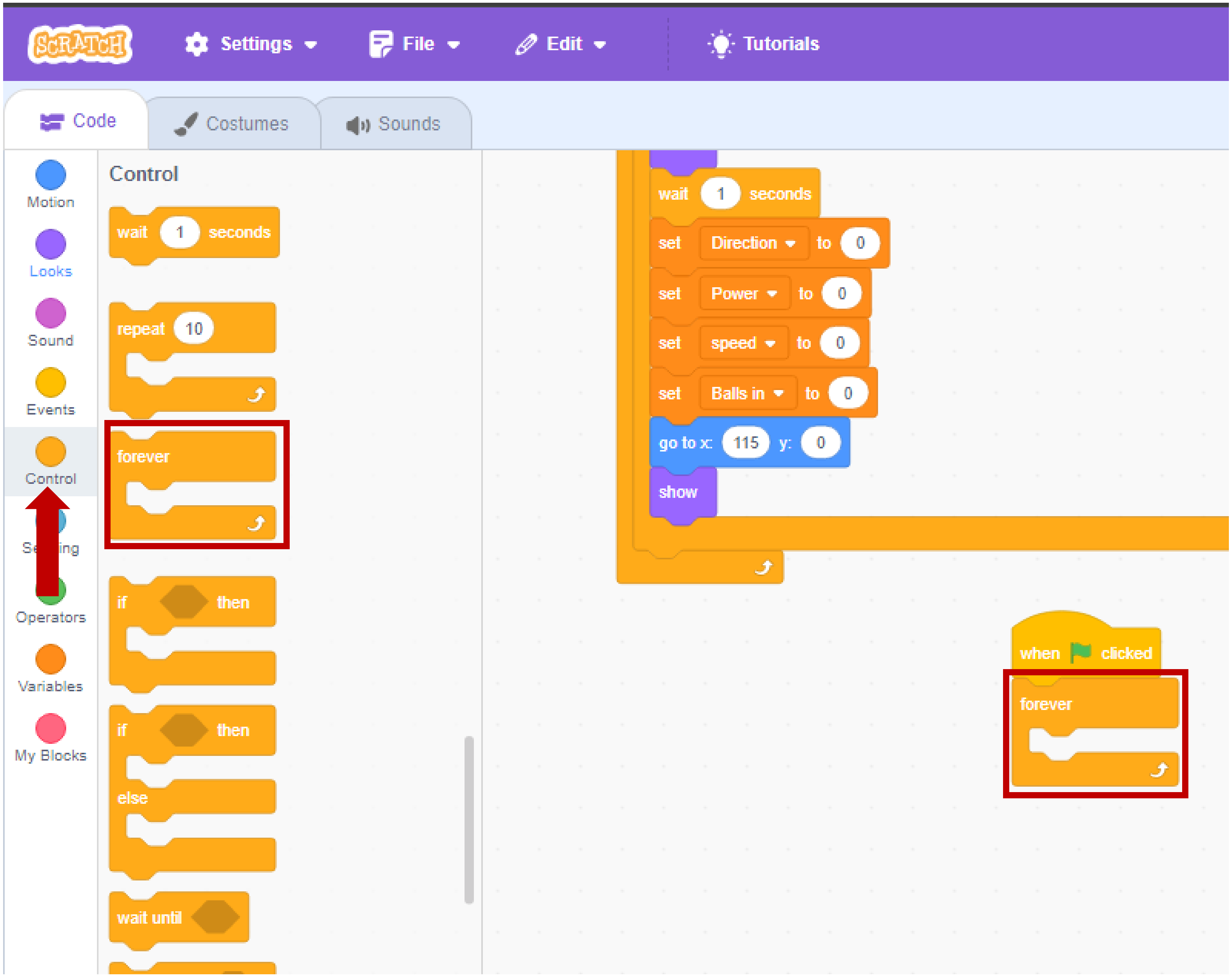


This section will cover level 1 of the game.

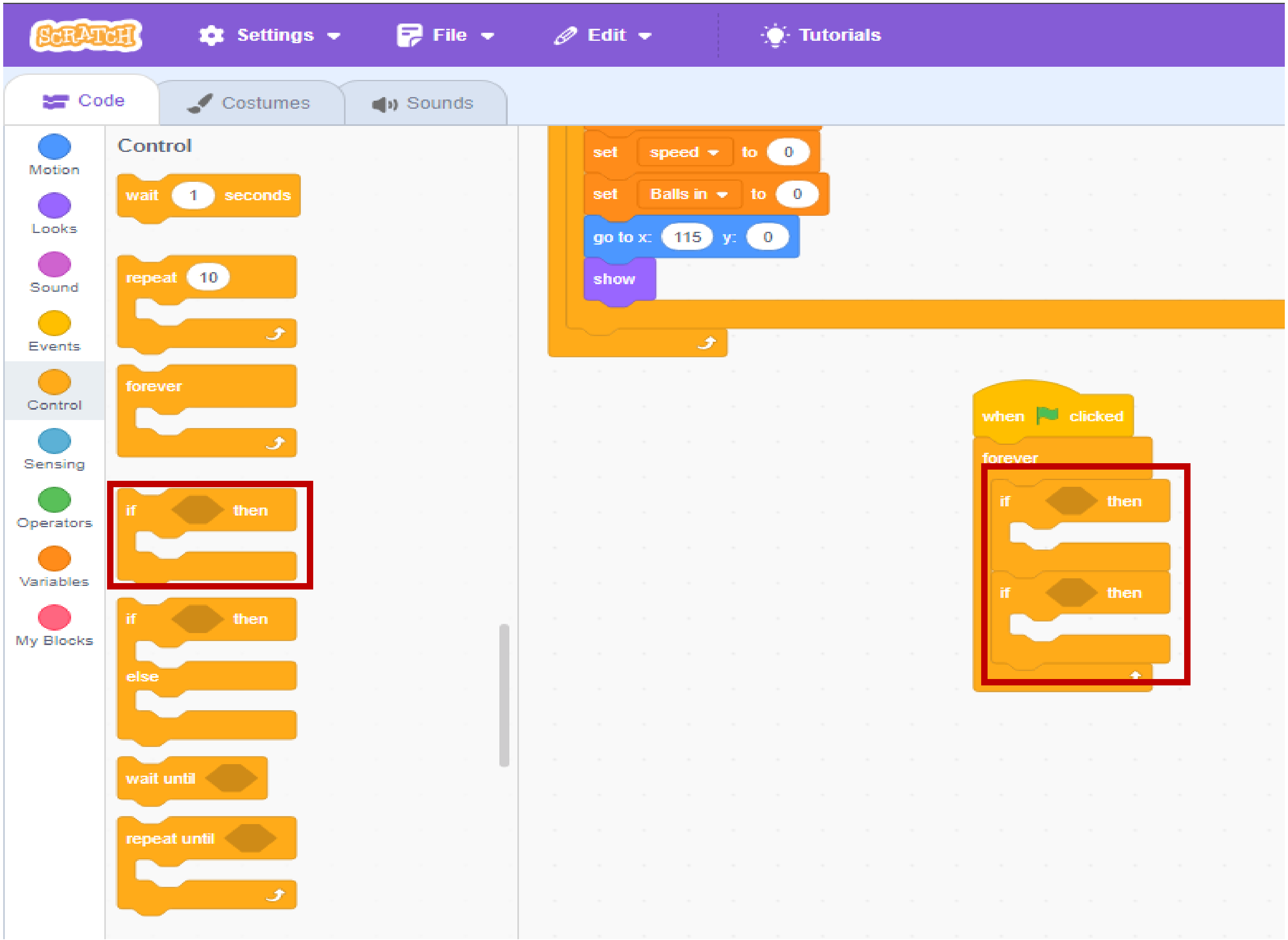
Make sure that the white ball is the selected sprite. Now click on “Events” on the left side of the screen. Then click and drag the “when flag clicked” and drop on an empty space on the canvas.



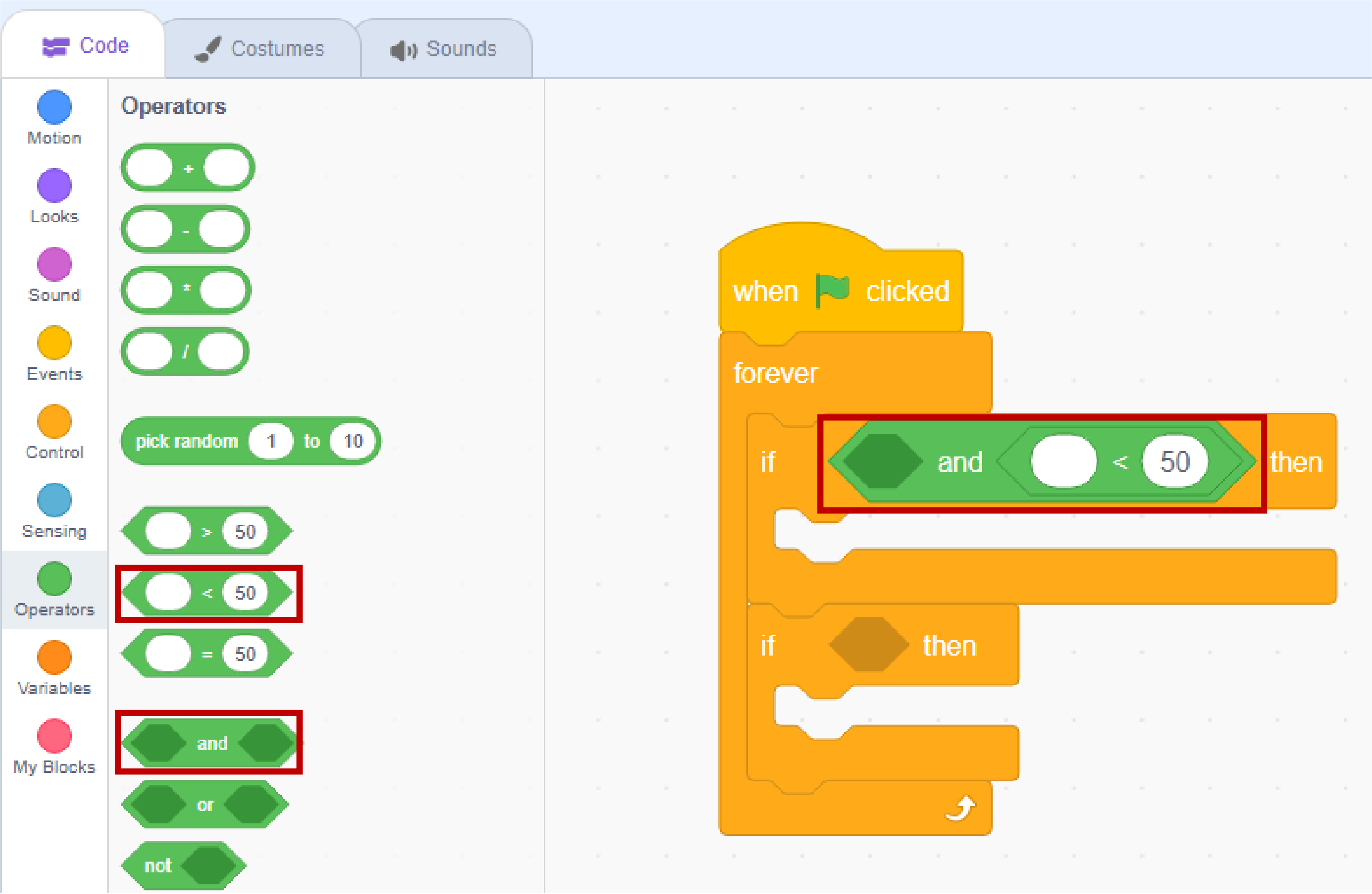
Now go to “Control” and grab the “forever” loop and attach it to code block.



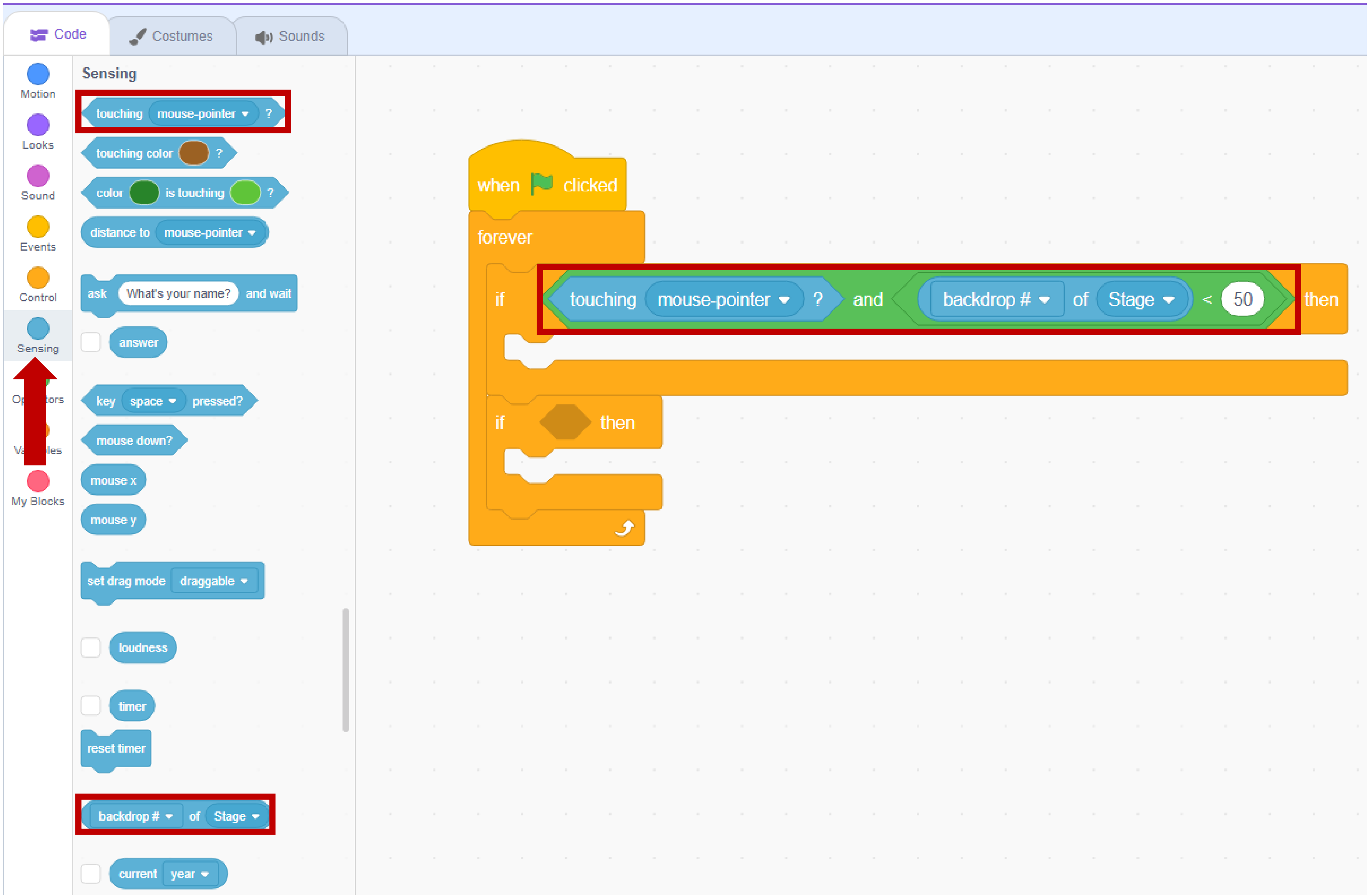
Now grab 2 “if-then” blocks and put them into the “forever loop” like shown below.

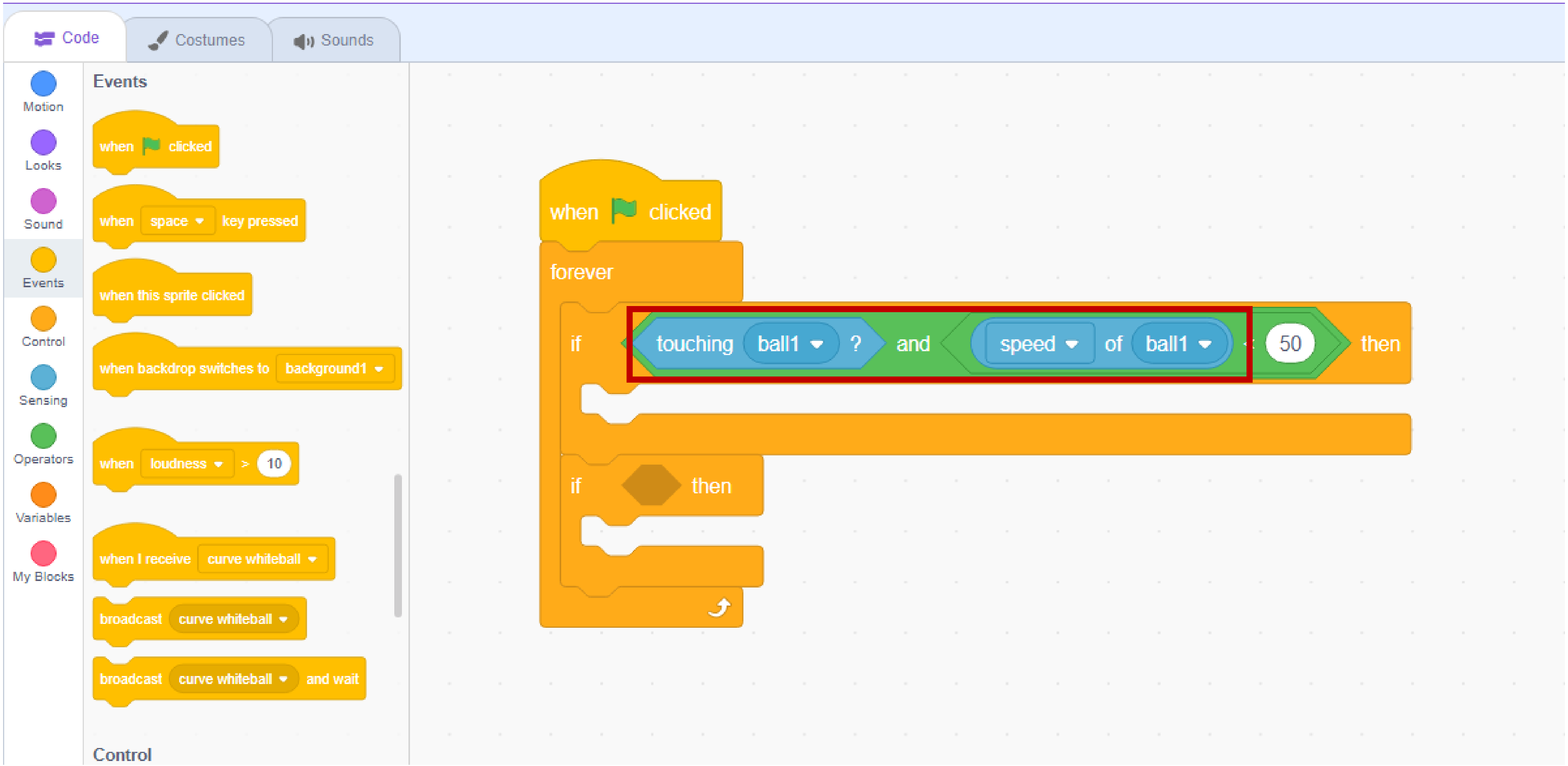


We will focus on the top “if-then” block for now. Go to “Operators”. Then grab the “__ and __” block and put it into the blank in the “if-then” block. Then grab the “__ < 50” block and put it into the second blank of the “__ and __” block.



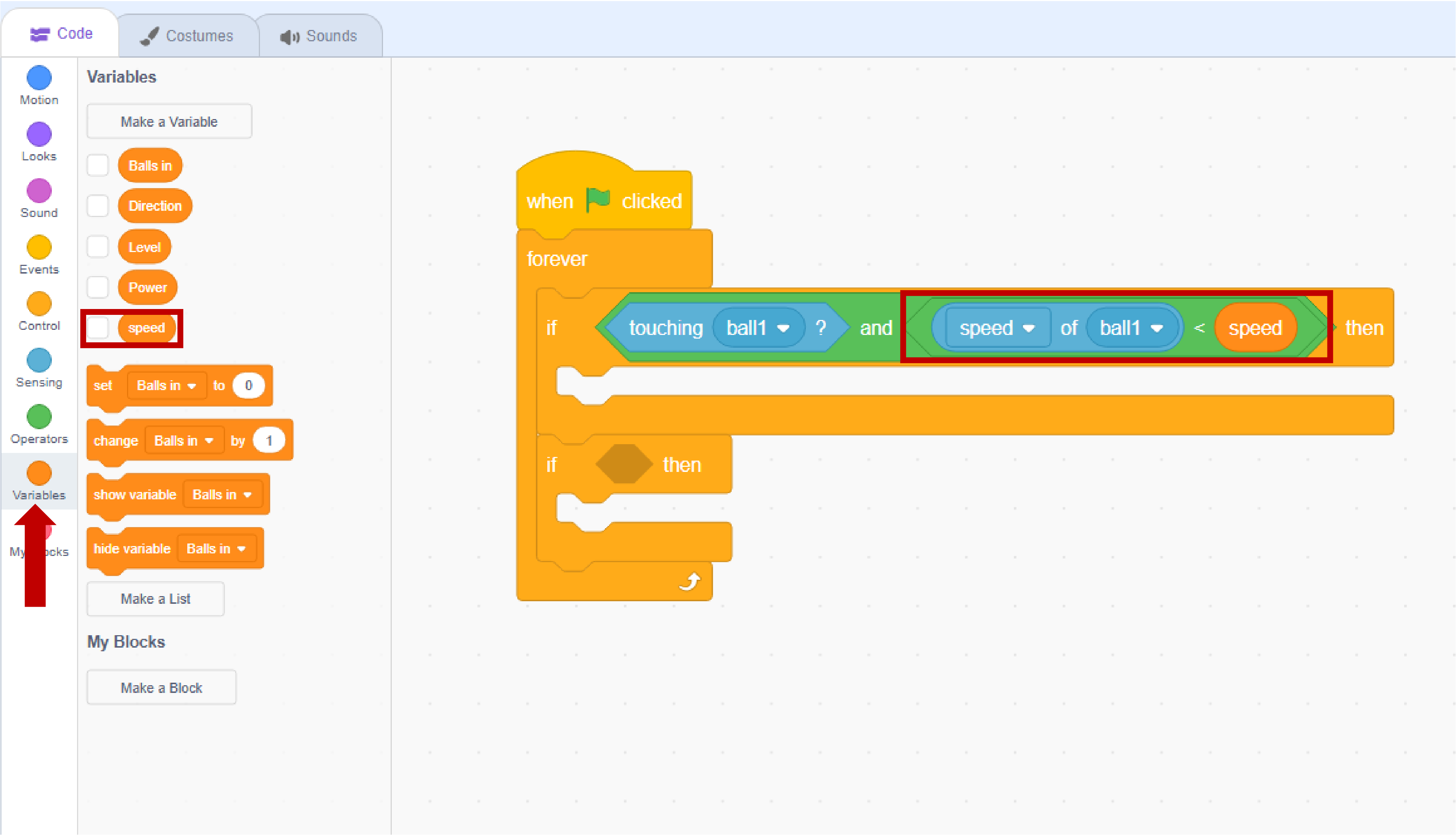
Go to “Sensing”. Grab the “touching mouse pointer?” block and insert it into the first blank of the “if-then” block. Then grab a “backdrop # of Stage” block and insert it into the “__ < 50” block. Change the mouse pointer to “ball 1” and “backdrop # of Stage” to “speed of ball 1”.



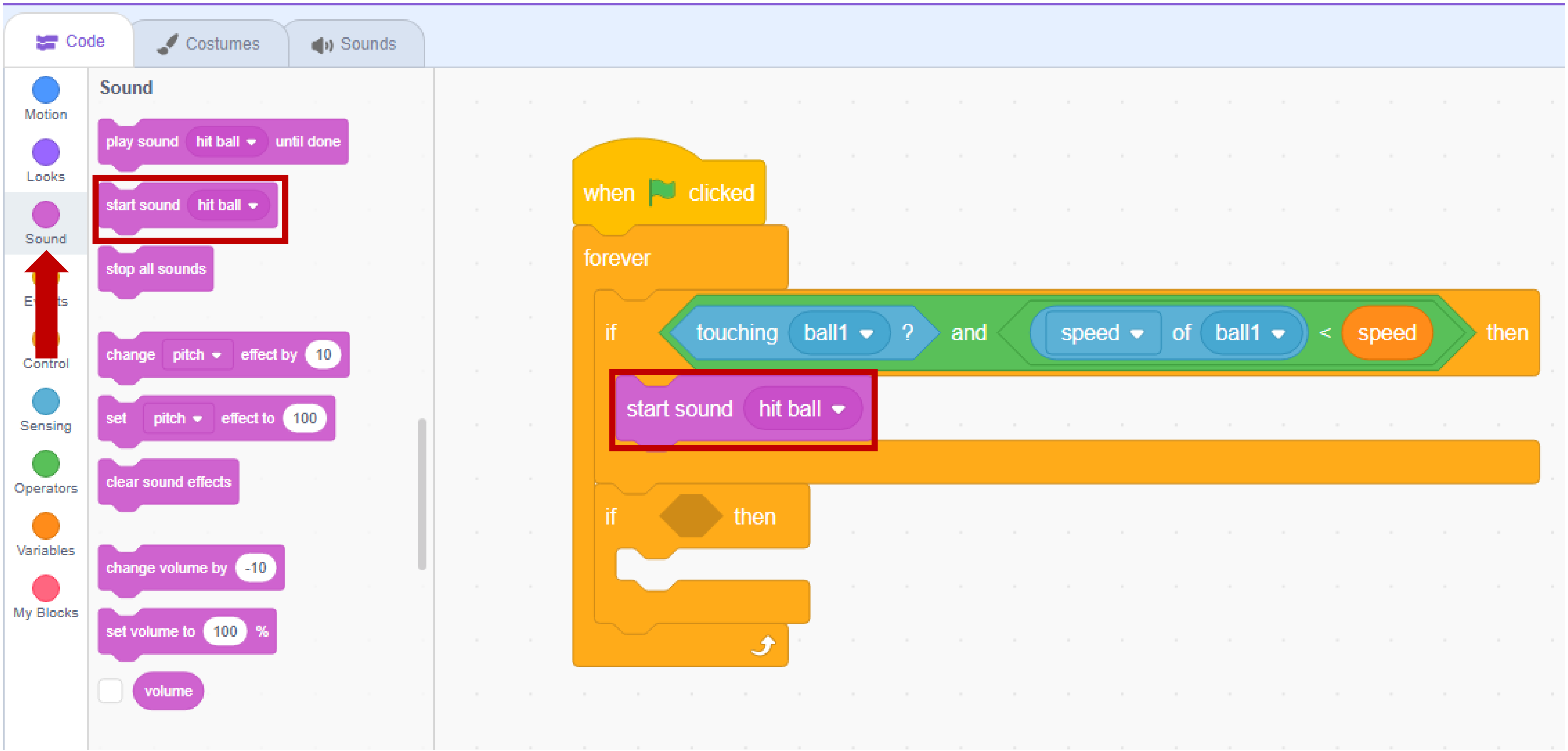


```
when ⚑ clicked
forever
    if < touching ball1 ▾ ? and speed ▾ of ball1 ▾ < 50 > then

    if < > then

↻
```

Now go to the “Variables” tab and grab the “speed” variable. Insert it into the “__ < 50” block.

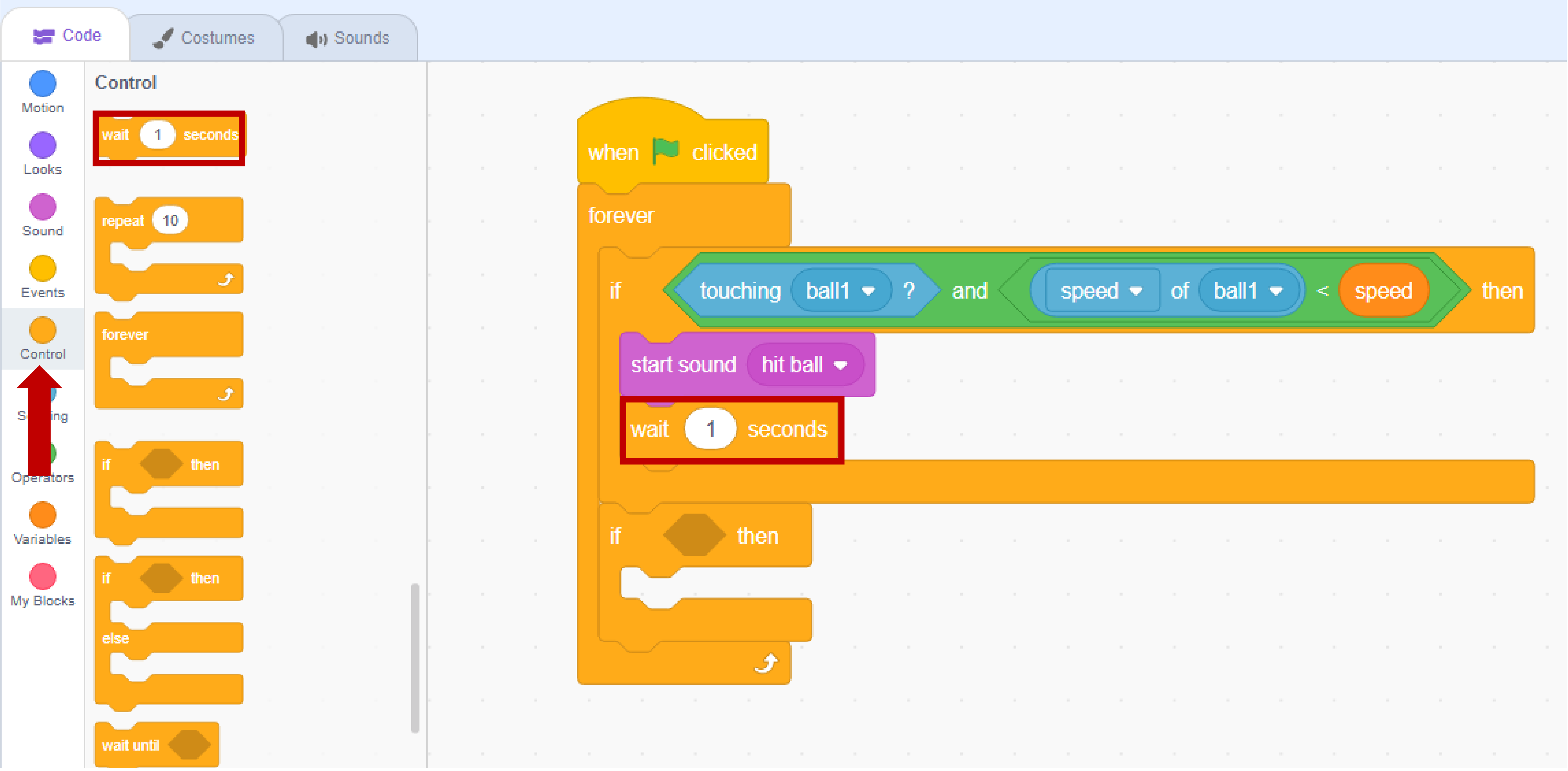


Go to “sounds” and grab the “start sound hit ball” and put it into the “if-then” statement.

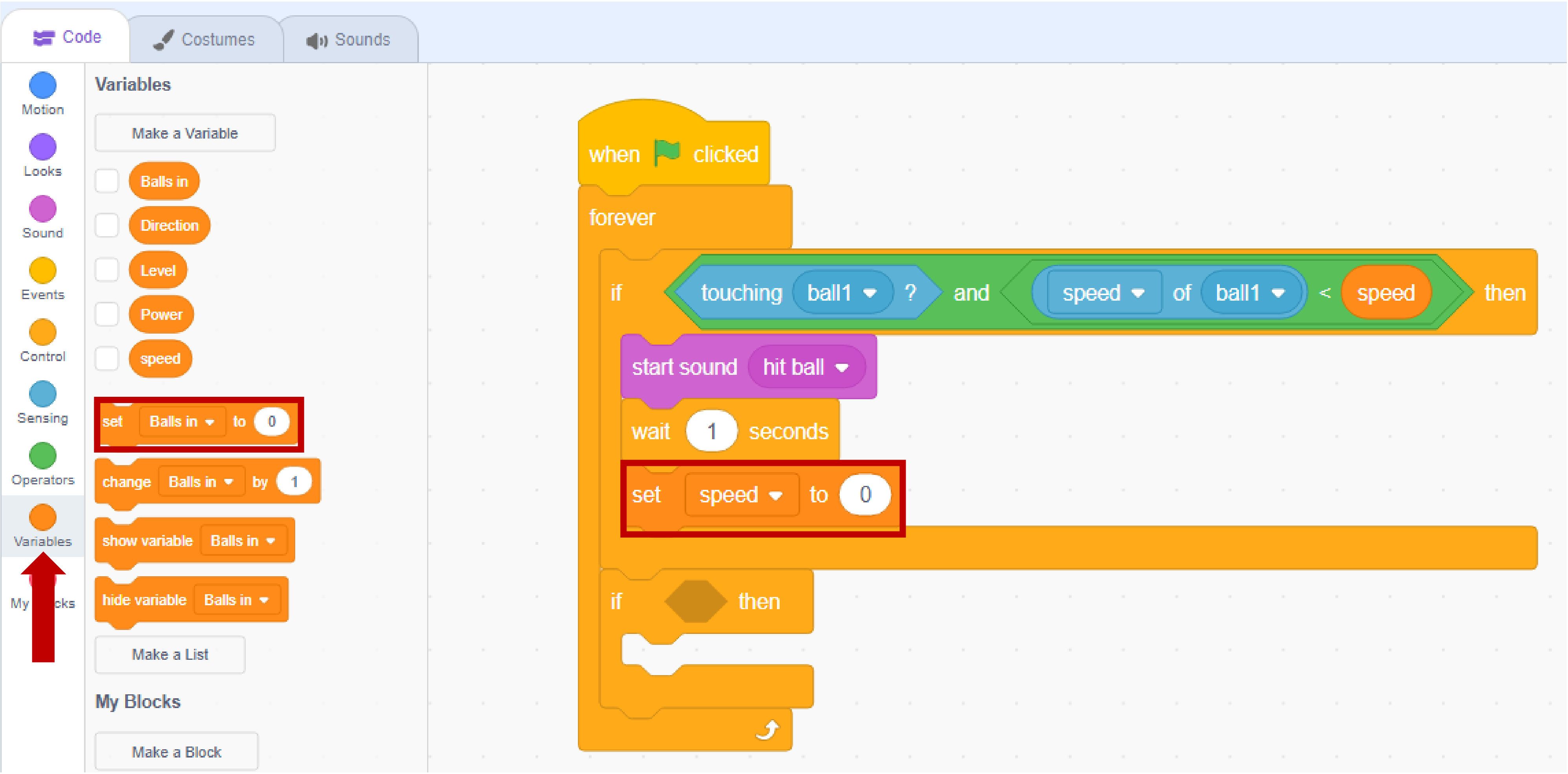


```
when 🏁 clicked
forever
    if touching ball1 ? and speed of ball1 < speed then
        start sound hit ball
    if then
```

Go to “Control” and then grab the “wait 1 seconds” block and insert it into the “if-then” statement. Then go to “Variables” and grab the “set balls in to 0” block and put it into the “if-then” statement. Change the variable to speed.

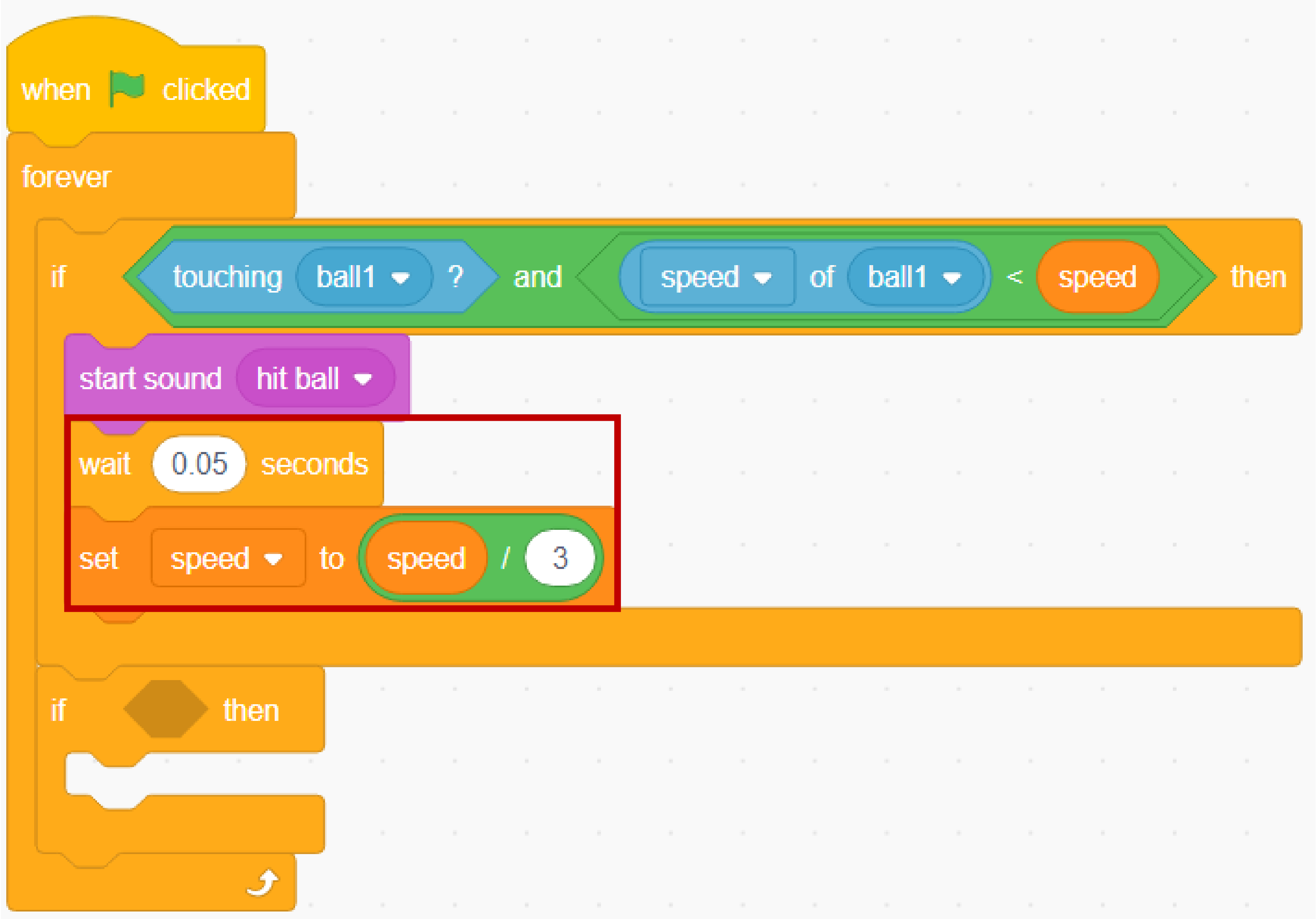


```
when ⚑ clicked
forever
    if <touching ball1 ? and speed of ball1 < speed> then
        start sound hit ball
        wait 1 seconds
    if <> then
```



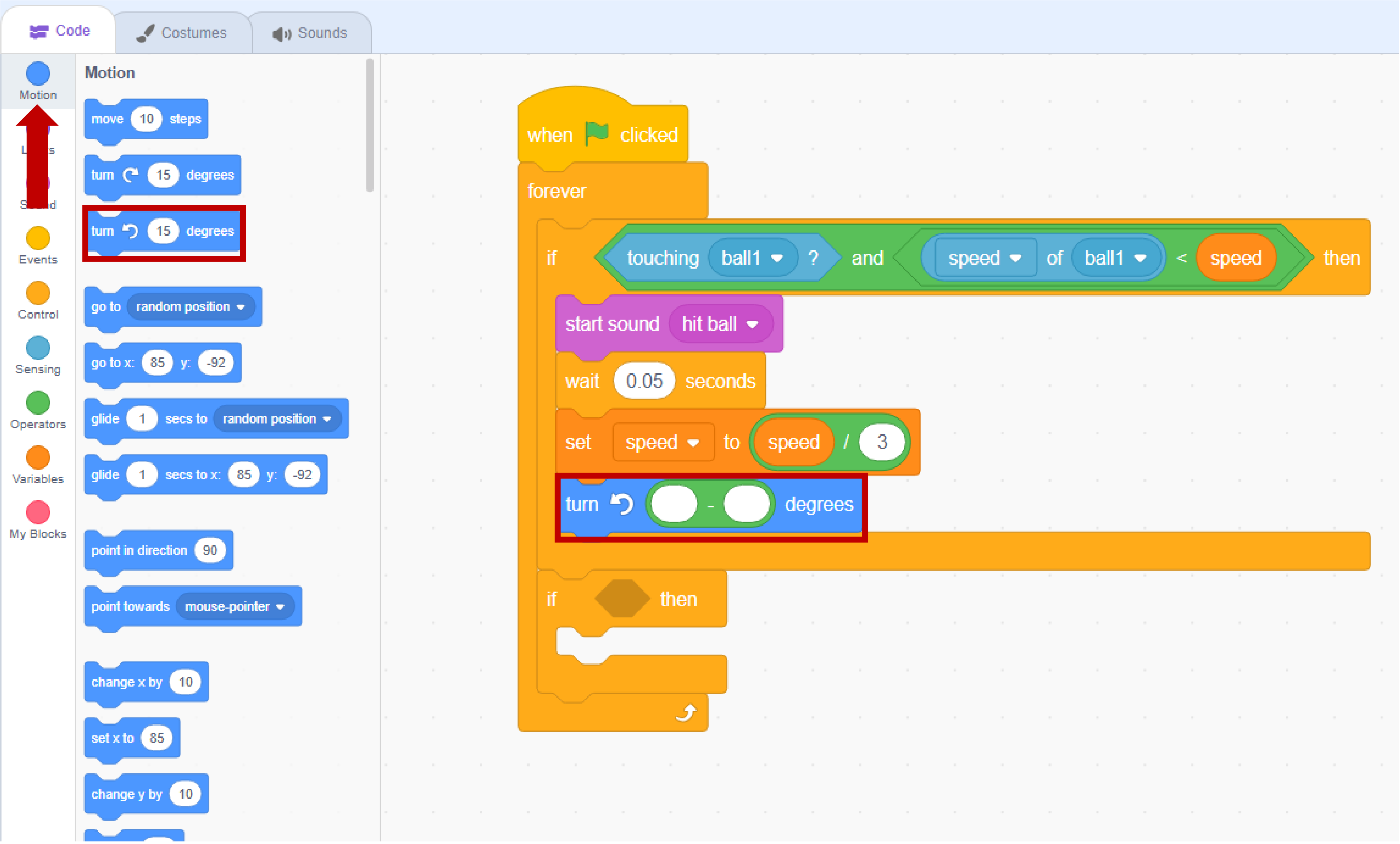
```
when ⚑ clicked
forever
    if <touching ball1 ? and <speed of ball1 < speed>> then
        start sound hit ball
        wait 1 seconds
        set speed to 0
    if < > then
↺
```

Change the wait time from 1 to 0.05 seconds. Then go to “Operators” and grab the “__ /__” block and put it into the “set speed to __” block. Then go to “Variables” to grab the speed variable and put it into the first blank of the “__ /__” block. Then type 3 for the last blank. It should look like the section shown below.



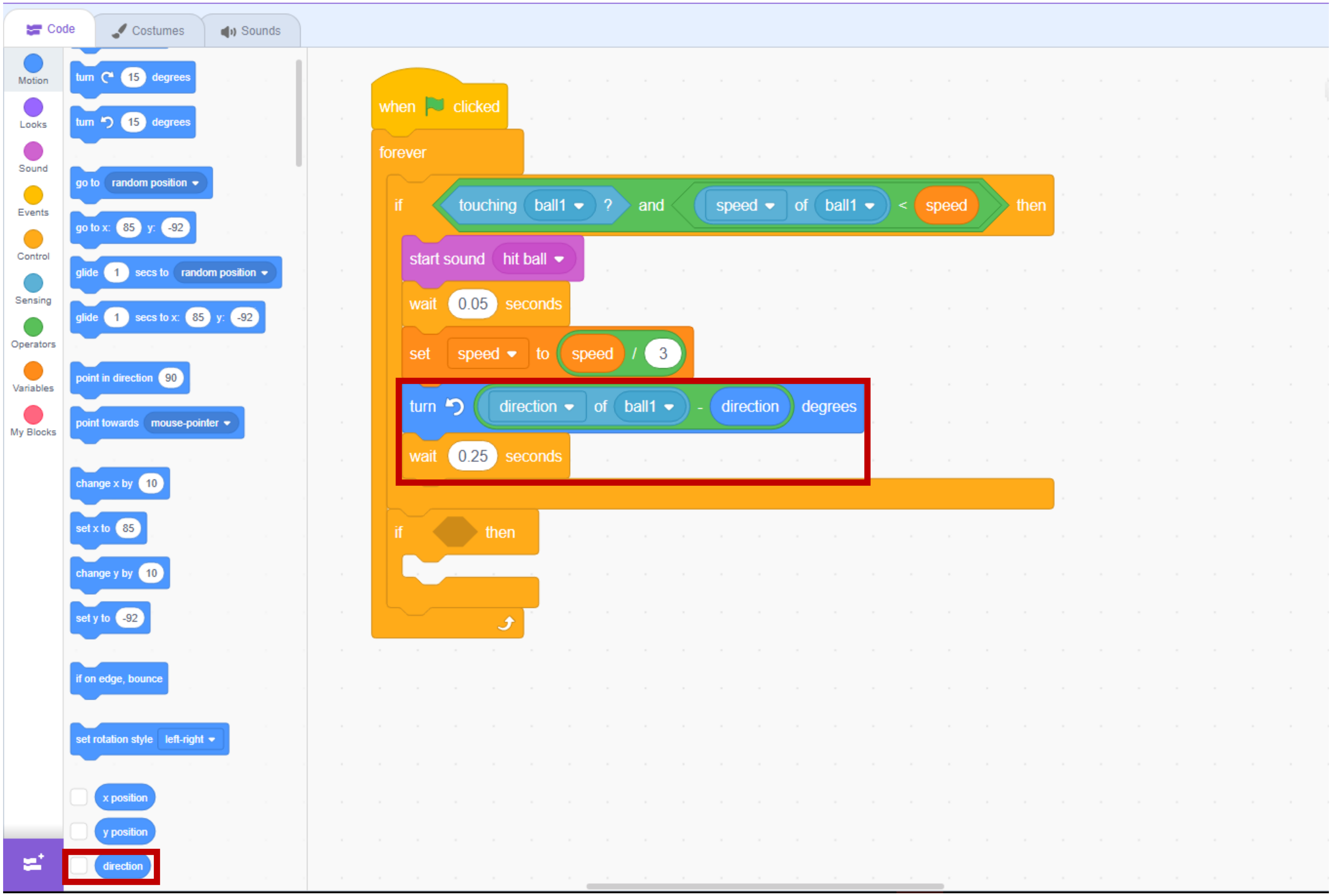
```
when ⚑ clicked
forever
    if touching ball1 ▾ ? and speed ▾ of ball1 ▾ < speed then
        start sound hit ball ▾
        wait 0.05 seconds
        set speed ▾ to speed / 3
    if then
```

Go to “Motion” and grab the “turn counterclockwise 15 degrees” block and put it into the “if-then” statement. Then go to “Operators” and grab the “__ - __” block and put it into the “turn counterclockwise 15 degrees” block.



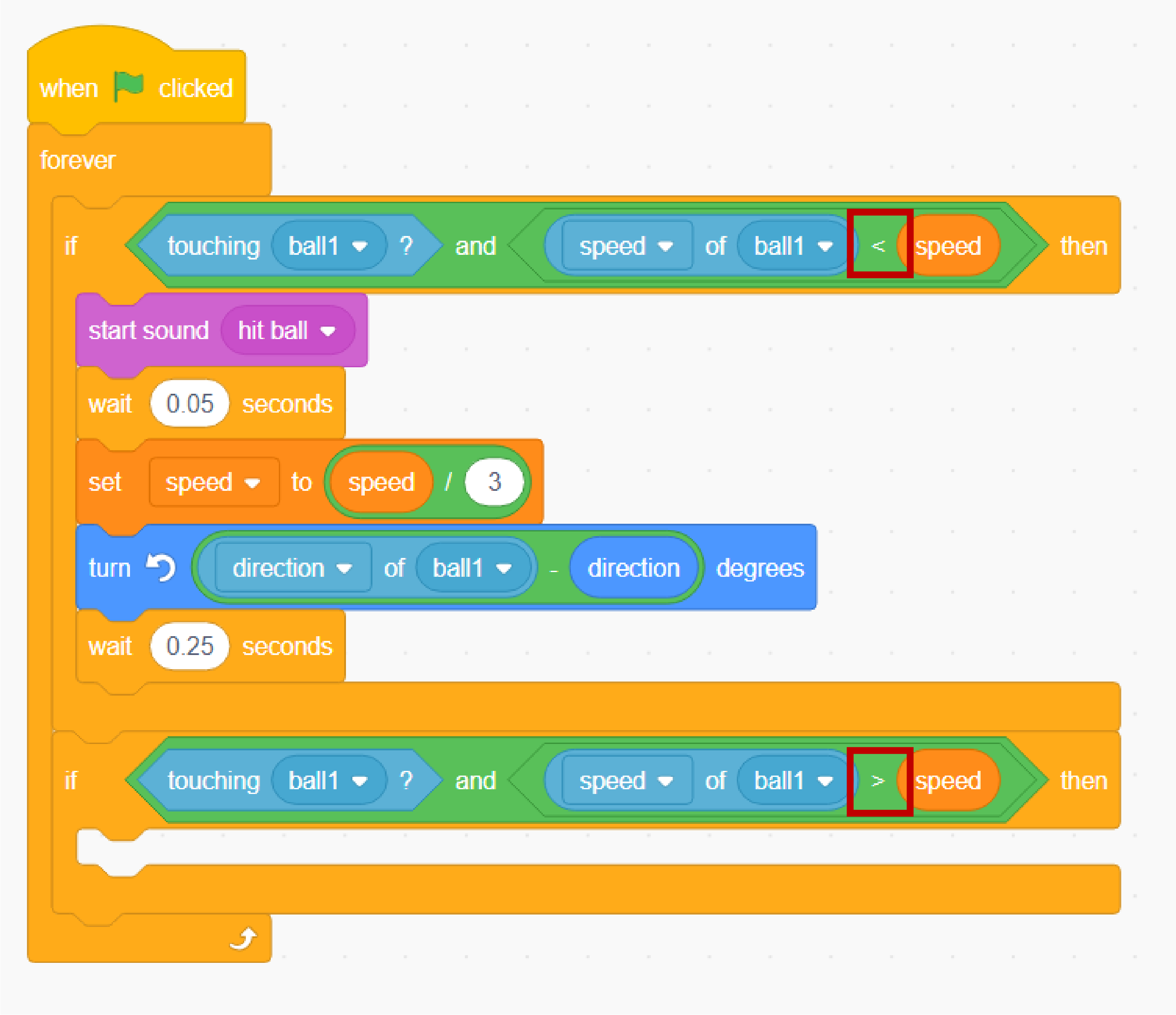
```
when ⚑ clicked
forever
    if <<touching ball1 ?> and <(speed of ball1) < (speed)>> then
        start sound hit ball
        wait 0.05 seconds
        set speed to (speed) / 3
        turn ↺ ( ( ) - ( ) ) degrees
    if < > then
```

Go to “Sensing” and grab the “backdrop # of Stage” and add it to the first blank. Change variables so it says, “direction of ball 1”. Then go to “Motion” and grab the “direction” block and attach it to the second blank. Then go to control add grab the “wait 1 seconds” block and change it to 0.25.



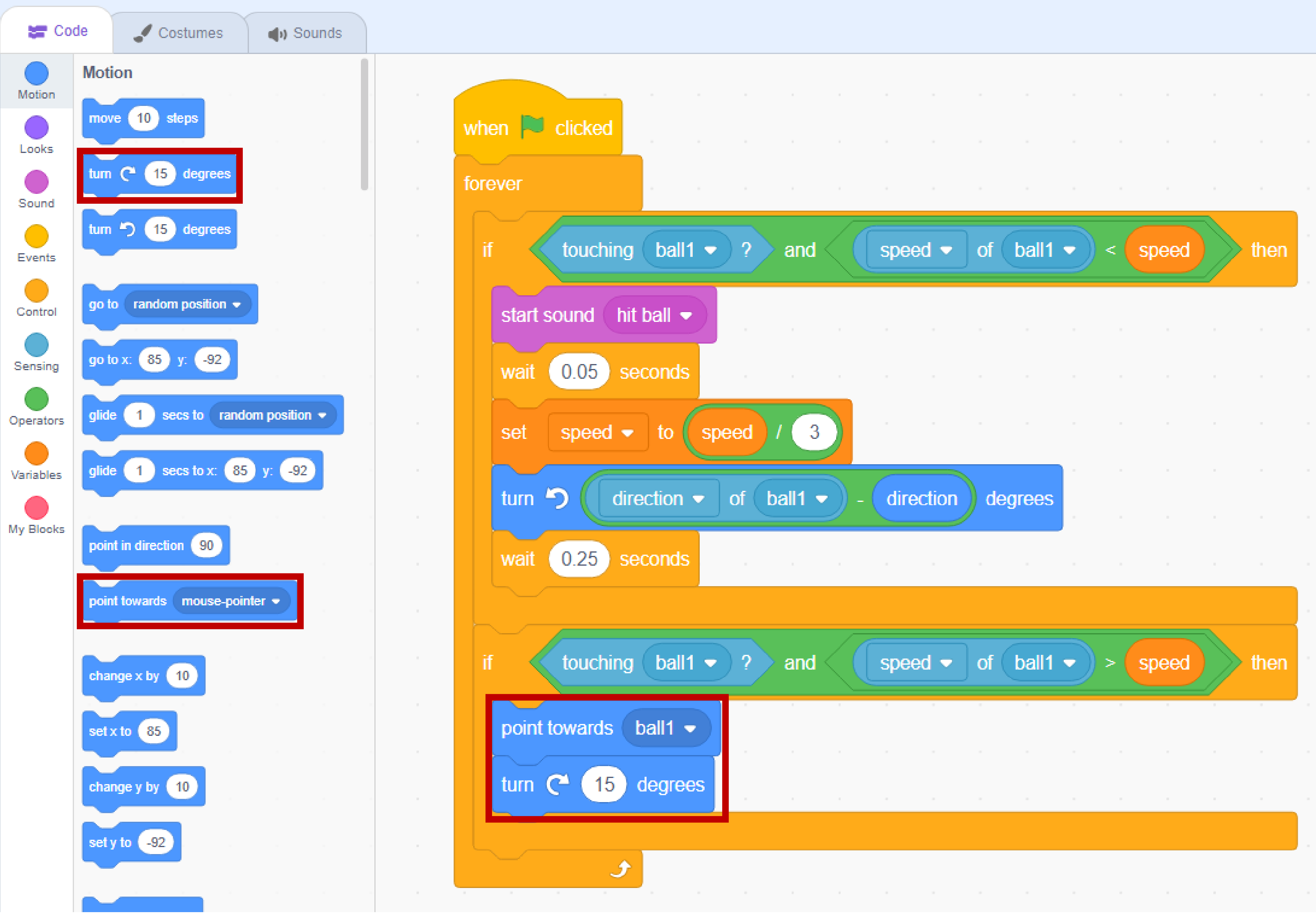
```
when ⚑ clicked
forever
    if <touching ball1 ? and speed of ball1 < speed> then
        start sound hit ball
        wait 0.05 seconds
        set speed to speed / 3
        turn ↺ direction of ball1 - direction degrees
        wait 0.25 seconds
    if <> then
```

The second “if-then” block is very similar to the first. Insert the same blocks in the “if-then” blank, but grab “__>__” instead of the “__<__”.



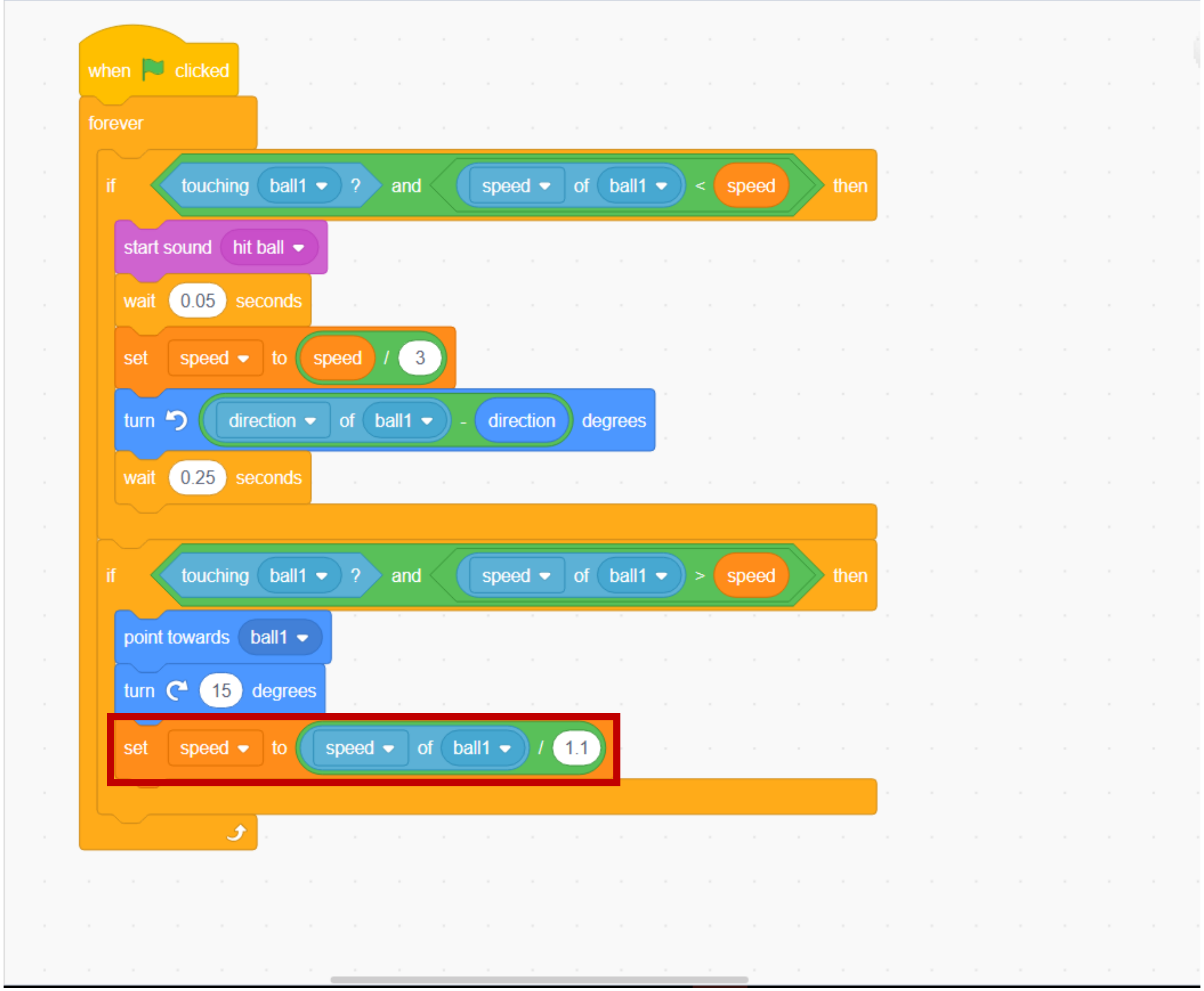
```
when 🏴 clicked
forever
    if <<touching ball1 ?> and <[speed of ball1] < (speed)>> then
        start sound hit ball
        wait 0.05 seconds
        set speed to ((speed) / 3)
        turn ↺ (([direction of ball1]) - (direction)) degrees
        wait 0.25 seconds
    if <<touching ball1 ?> and <[speed of ball1] > (speed)>> then
```

Go to the “Motion” tab and grab the “point towards mouse-pointer” block and change it “point towards ball 1”. Then add a “turn clockwise 15 degrees” block. Keep the same degree.



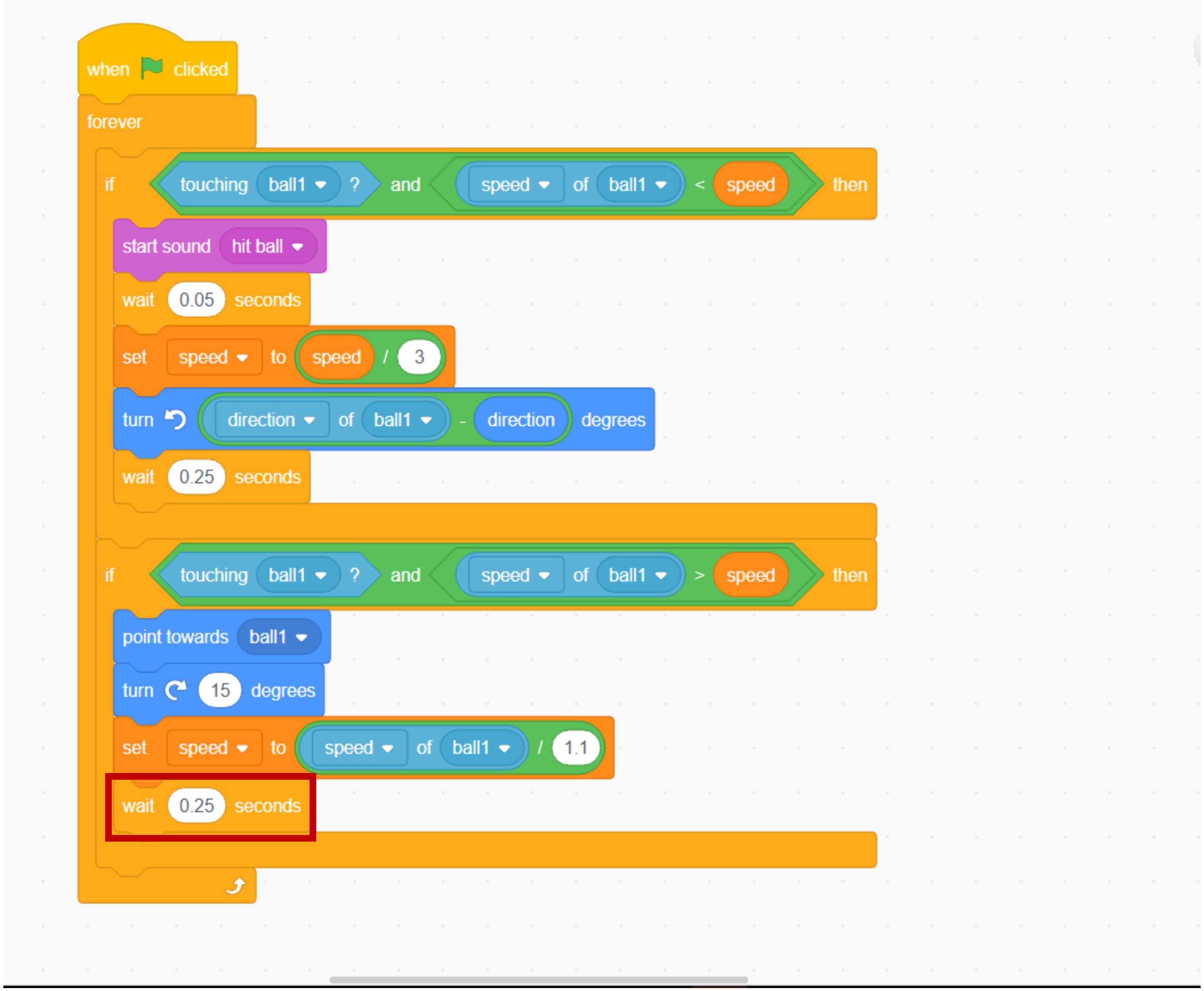
```
when 🏁 clicked
forever
    if touching ball1 ? and speed of ball1 < speed then
        start sound hit ball
        wait 0.05 seconds
        set speed to speed / 3
        turn ↺ direction of ball1 - direction degrees
        wait 0.25 seconds
    if touching ball1 ? and speed of ball1 > speed then
        point towards ball1
        turn ↻ 15 degrees
```

Go to the “Variables” tab and grab the “set balls in to 0” block and change the variable to “speed”. Then go to “Operators” and grab the “__/__”. Then go to “Sensing” and grab the “backdrop # of Stage” and insert it into the first blank. Change it to “speed of ball 1”. Then type 1.1 in the second blank.



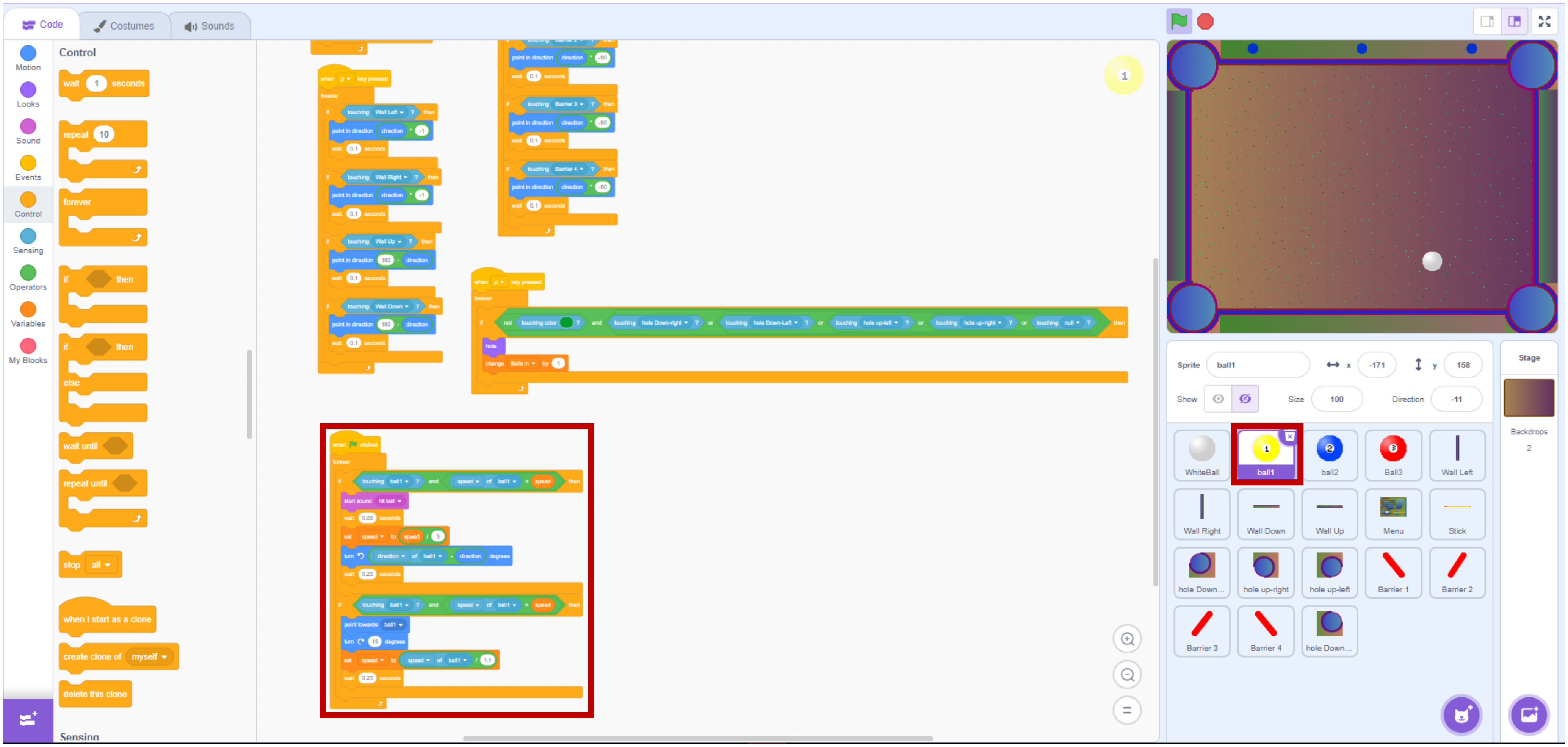
```
when ⚑ clicked
forever
  if touching ball1 ? and speed of ball1 < speed then
    start sound hit ball
    wait 0.05 seconds
    set speed to speed / 3
    turn ↺ direction of ball1 - direction degrees
    wait 0.25 seconds
  if touching ball1 ? and speed of ball1 > speed then
    point towards ball1
    turn ↻ 15 degrees
    set speed to speed of ball1 / 1.1
```

Go to the “Control” tab and grab the “wait 1 second” block. Change the block so it will wait 0.25 seconds.



```
when ⚑ clicked
forever
    if <touching ball1 ? and speed of ball1 < speed> then
        start sound hit ball
        wait 0.05 seconds
        set speed to speed / 3
        turn ↺ direction of ball1 - direction degrees
        wait 0.25 seconds
    if <touching ball1 ? and speed of ball1 > speed> then
        point towards ball1
        turn ↻ 15 degrees
        set speed to speed of ball1 / 1.1
        wait 0.25 seconds
```

Click on the “when flag clicked” block in the code and then use ctrl+c to copy the code block. Now select the “ball 1” sprite. Use ctrl+v to paste the code onto the canvas. You may need to move around the canvas to find the code block.



```
when p key pressed
forever
    if touching Wall Left ? then
        point in direction (direction * -1)
        wait 0.1 seconds
    if touching Wall Right ? then
        point in direction (direction * -1)
        wait 0.1 seconds
    if touching Wall Up ? then
        point in direction (180 - direction)
        wait 0.1 seconds
    if touching Wall Down ? then
        point in direction (180 - direction)
        wait 0.1 seconds
```

```
        point in direction (direction * -90)
        wait 0.1 seconds
    if touching Barrier 3 ? then
        point in direction (direction * -90)
        wait 0.1 seconds
    if touching Barrier 4 ? then
        point in direction (direction * -90)
        wait 0.1 seconds
```

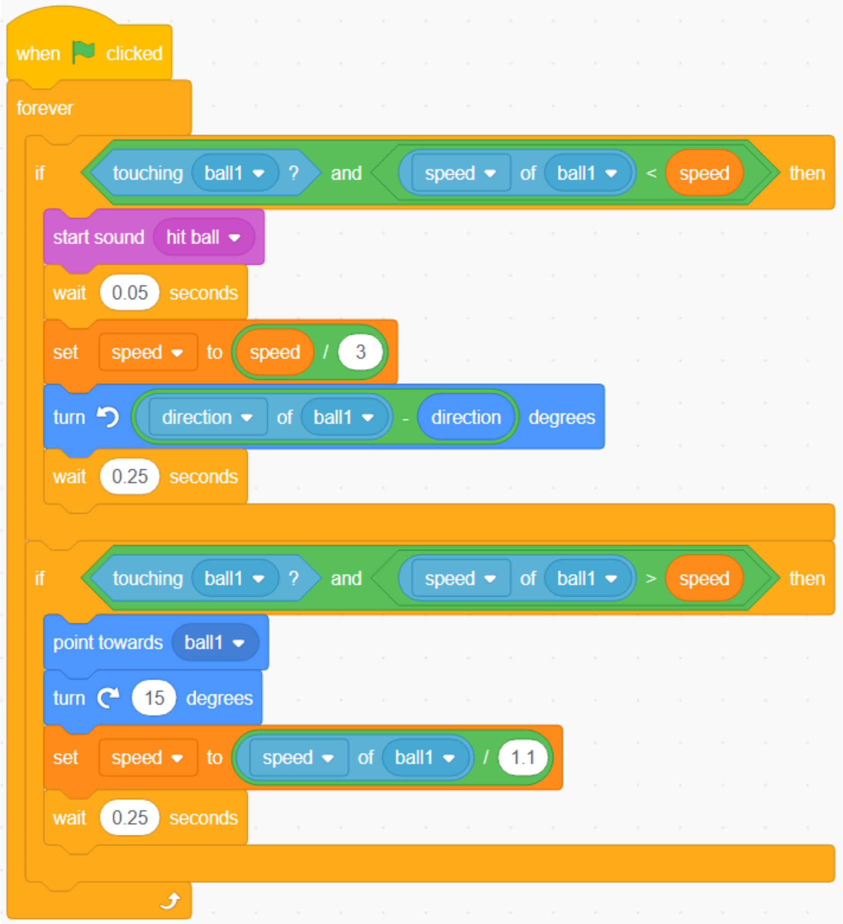
```
when p key pressed
forever
    if not touching color ? and touching hole Down-right ? or touching hole Down-Left ? or touching hole up-left ? or touching hole up-right ? or touching null ? then
        hide
        change Balls in by 1
```

```
when 🏁 clicked
forever
    if touching ball1 ? and (speed of ball1) < speed then
        start sound hit ball
        wait 0.05 seconds
        set speed to (speed / 3)
        turn ↺ ((direction of ball1) - direction) degrees
        wait 0.25 seconds
    if touching ball1 ? and (speed of ball1) > speed then
        point towards ball1
        turn ↻ 15 degrees
        set speed to ((speed of ball1) / 1.1)
        wait 0.25 seconds
```

Sprite ball1 x -171 y 158
Show Size 100 Direction -11

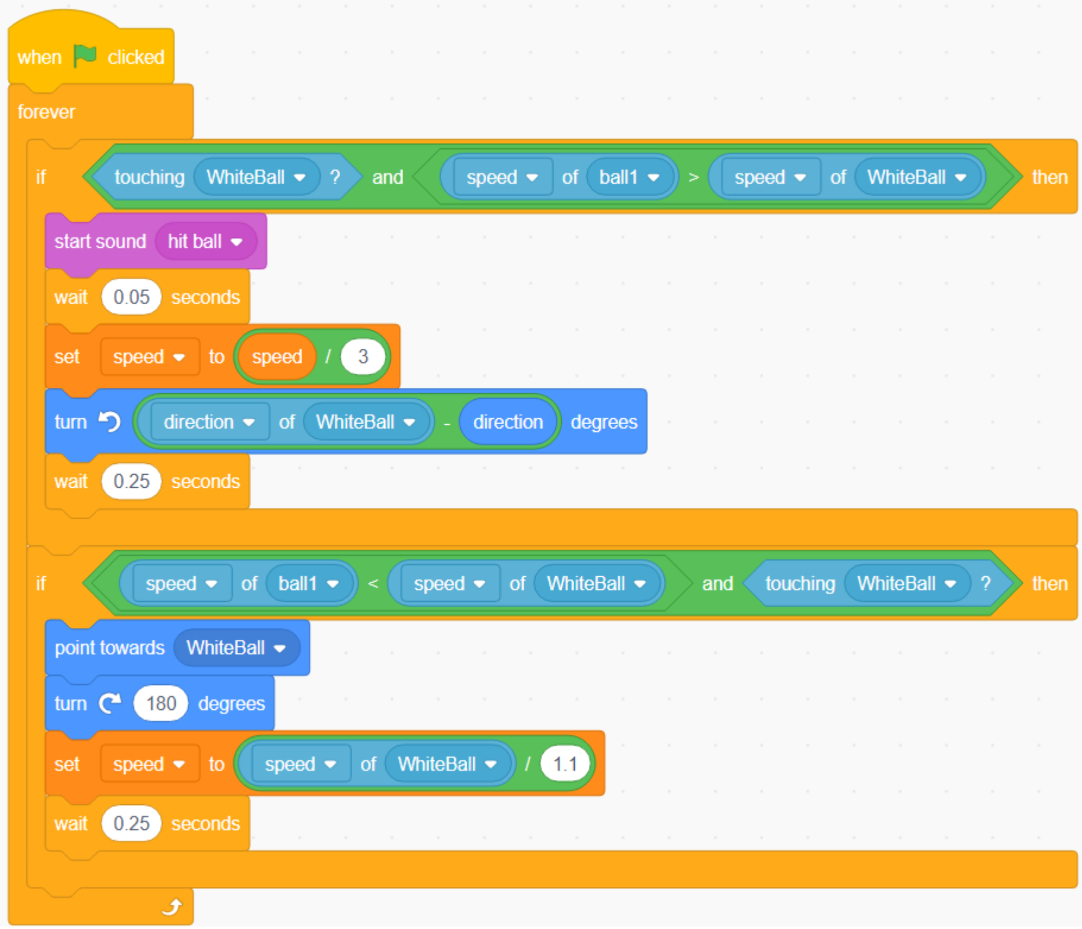
Now we have to change all instances of “ball 1” to “white ball”. Make sure that the block is still checking for speed not direction. We also have to change the “turn clockwise 15 degrees” block to “turn clockwise 180 degrees”. Use the two pictures below to compare the different code blocks.

Code block in “white ball” sprite



```
when 🏁 clicked
forever
    if <touching ball1 ? and speed of ball1 < speed> then
        start sound hit ball
        wait 0.05 seconds
        set speed to speed / 3
        turn ↺ direction of ball1 - direction degrees
        wait 0.25 seconds
    if <touching ball1 ? and speed of ball1 > speed> then
        point towards ball1
        turn ↻ 15 degrees
        set speed to speed of ball1 / 1.1
        wait 0.25 seconds
```

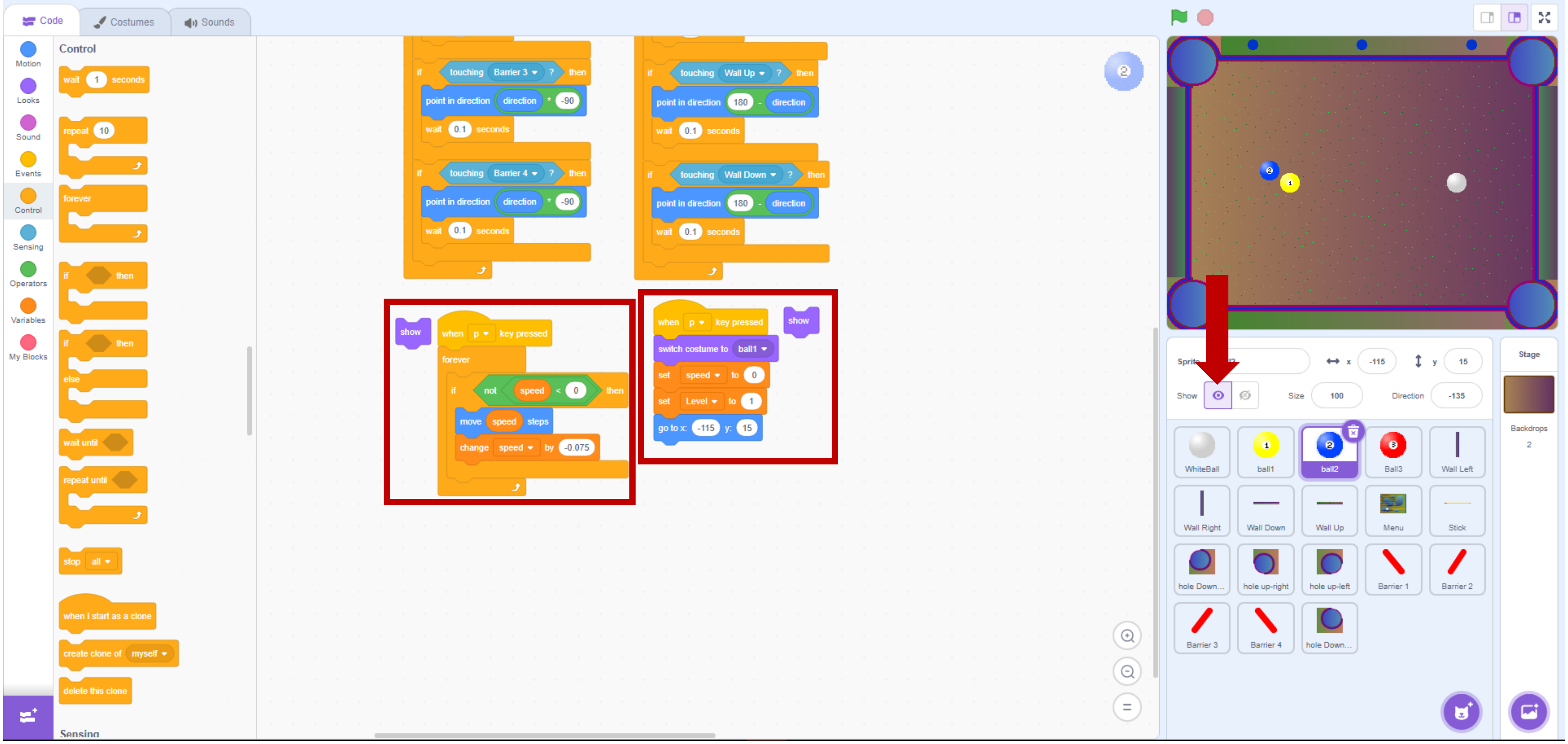
Code block in “ball 1” sprite



```
when 🏁 clicked
forever
    if <touching WhiteBall ? and speed of ball1 > speed of WhiteBall> then
        start sound hit ball
        wait 0.05 seconds
        set speed to speed / 3
        turn ↺ direction of WhiteBall - direction degrees
        wait 0.25 seconds
    if <speed of ball1 < speed of WhiteBall and touching WhiteBall ?> then
        point towards WhiteBall
        turn ↻ 180 degrees
        set speed to speed of WhiteBall / 1.1
        wait 0.25 seconds
```

This section will cover level 2 of the game

First go to the “ball 2” sprite and click on the eyeball to make the ball visible. Then attach the two “show” block to the adjacent code blocks they are next to.



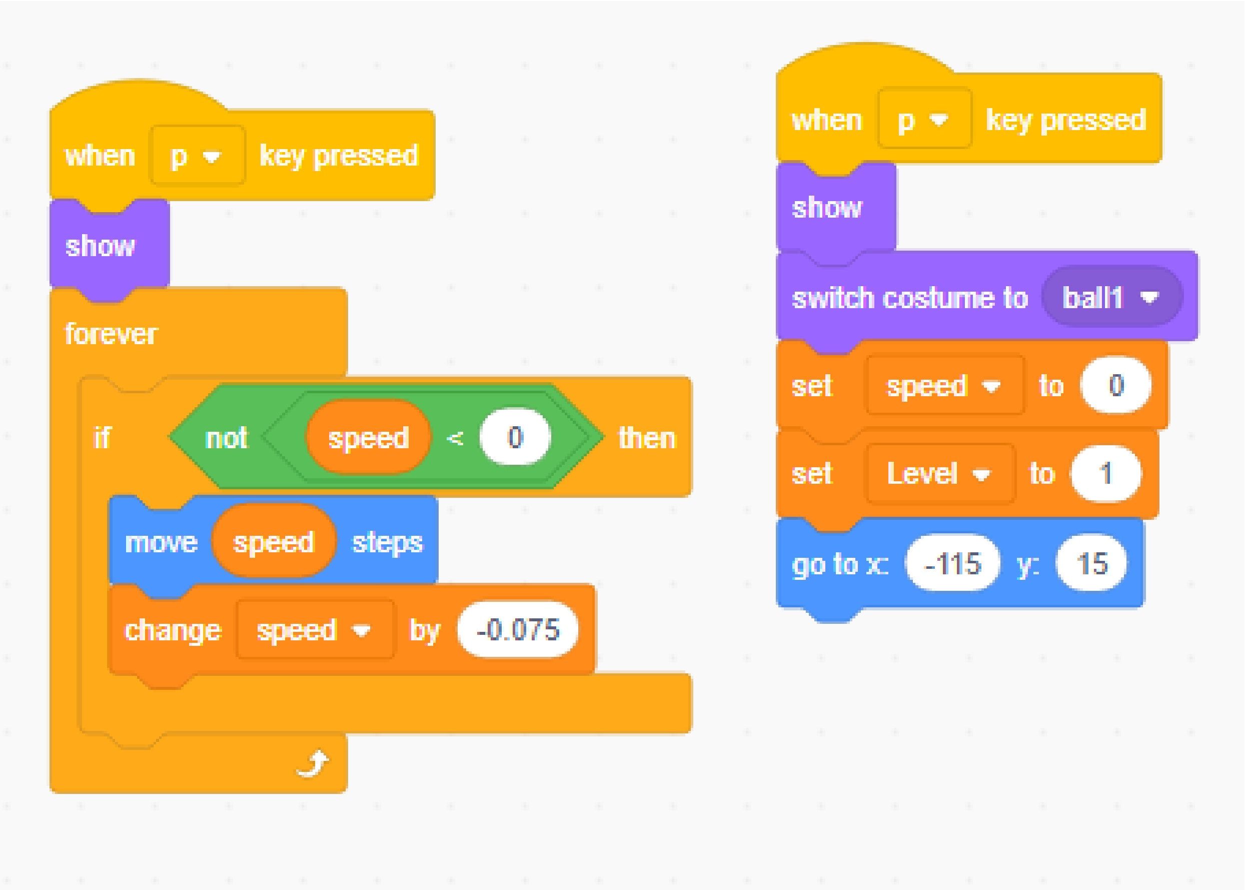
```
if touching Barrier 3 ? then
    point in direction (direction * -90)
    wait 0.1 seconds
if touching Barrier 4 ? then
    point in direction (direction * -90)
    wait 0.1 seconds
```

```
if touching Wall Up ? then
    point in direction (180 - direction)
    wait 0.1 seconds
if touching Wall Down ? then
    point in direction (180 - direction)
    wait 0.1 seconds
```

```
show

when p key pressed
forever
    if not (speed < 0) then
        move speed steps
        change speed by -0.075
```

```
when p key pressed
switch costume to ball1
set speed to 0
set Level to 1
go to x: -115 y: 15

show
```

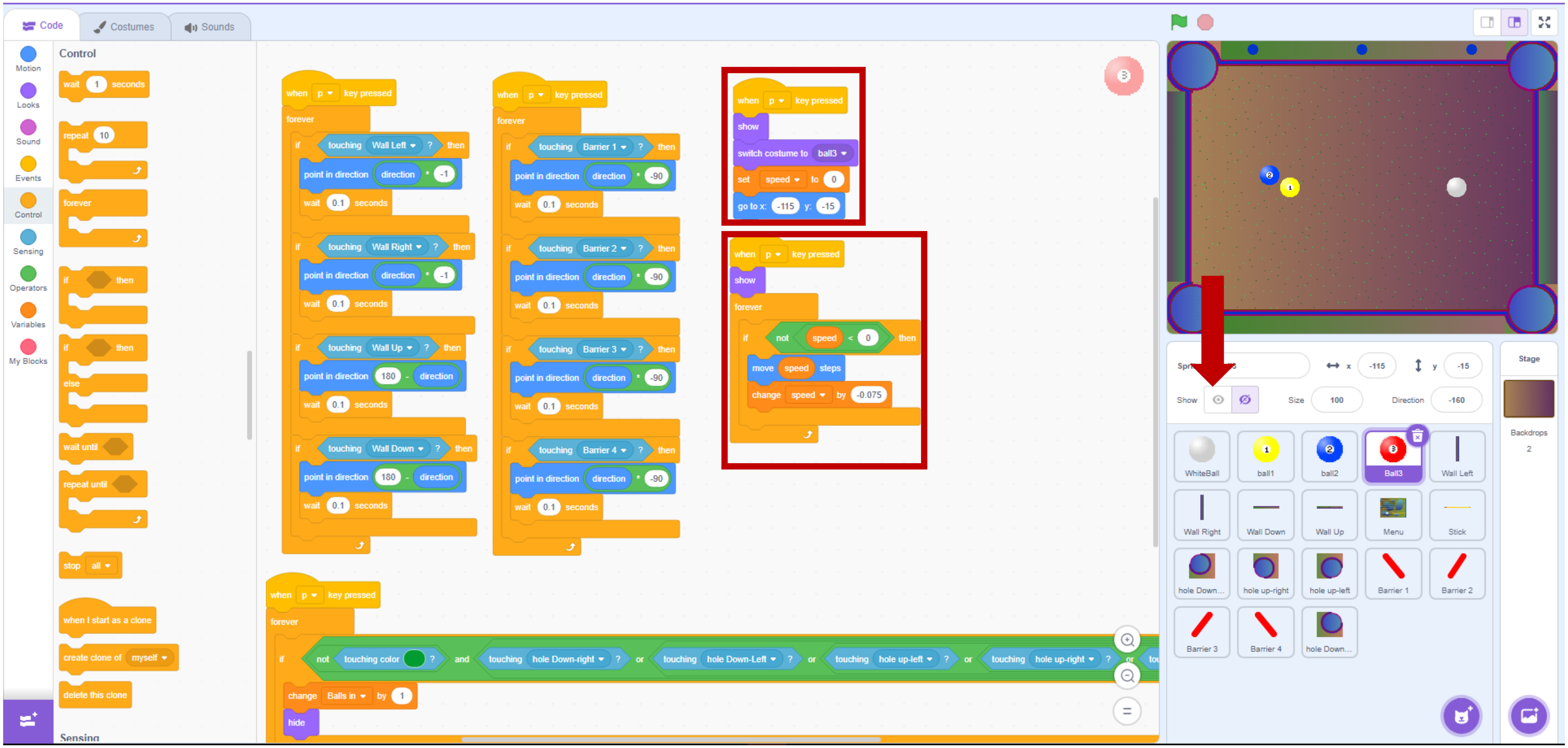
x -115 y 15
Size 100 Direction -135



```
when p key pressed
show
forever
    if not speed < 0 then
        move speed steps
        change speed by -0.075
```

```
when p key pressed
show
switch costume to ball1
set speed to 0
set Level to 1
go to x: -115 y: 15
```

Repeat this process for ball 3.



```
when p key pressed
forever
    if touching Wall Left ? then
        point in direction (direction * -1)
        wait 0.1 seconds
    if touching Wall Right ? then
        point in direction (direction * -1)
        wait 0.1 seconds
    if touching Wall Up ? then
        point in direction (180 - direction)
        wait 0.1 seconds
    if touching Wall Down ? then
        point in direction (180 - direction)
        wait 0.1 seconds
```

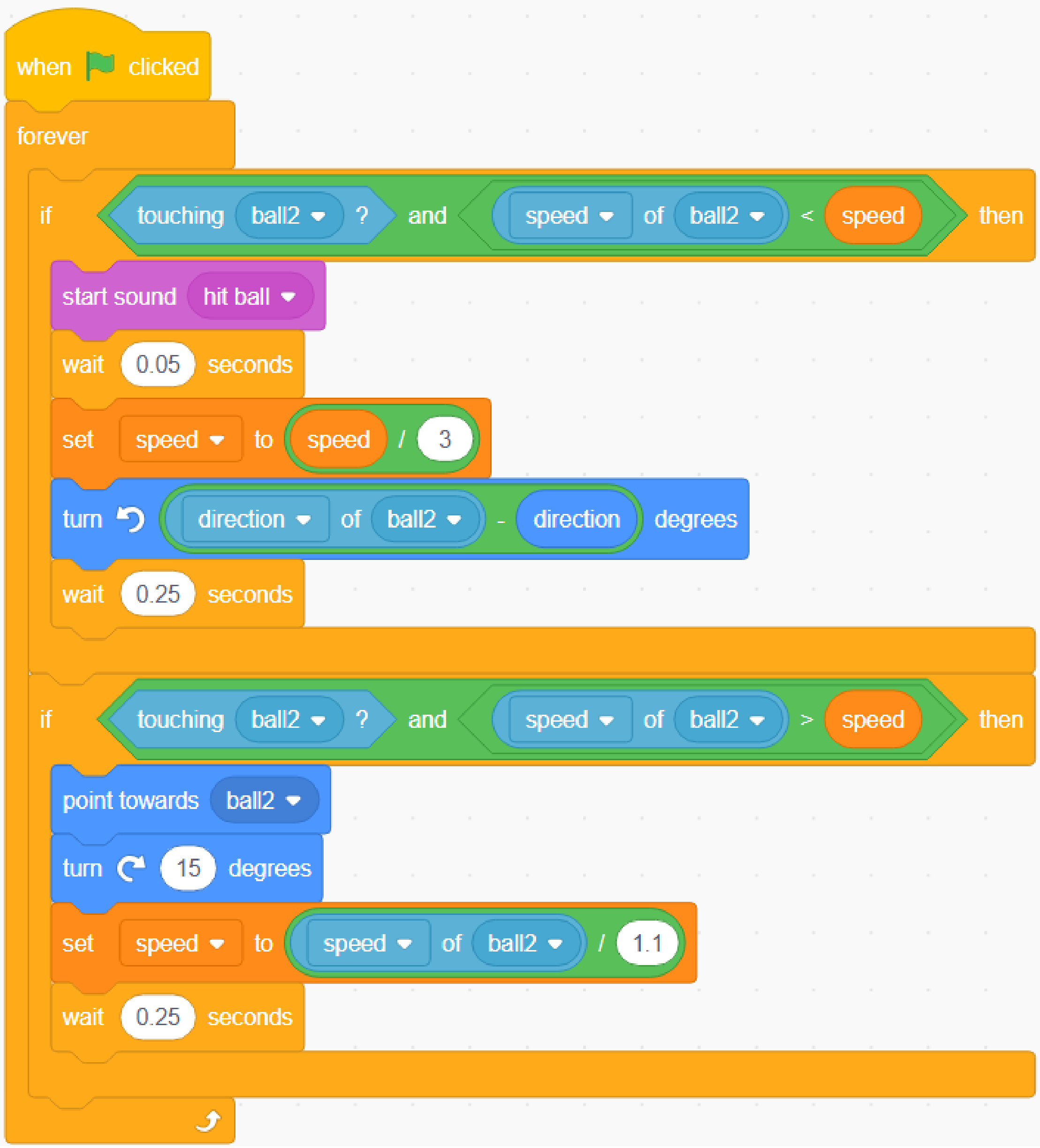
```
when p key pressed
forever
    if touching Barrier 1 ? then
        point in direction (direction * -90)
        wait 0.1 seconds
    if touching Barrier 2 ? then
        point in direction (direction * -90)
        wait 0.1 seconds
    if touching Barrier 3 ? then
        point in direction (direction * -90)
        wait 0.1 seconds
    if touching Barrier 4 ? then
        point in direction (direction * -90)
        wait 0.1 seconds
```

```
when p key pressed
show
switch costume to ball3
set speed to 0
go to x: -115 y: -15
```

```
when p key pressed
show
forever
    if not (speed < 0) then
        move speed steps
        change speed by -0.075
```

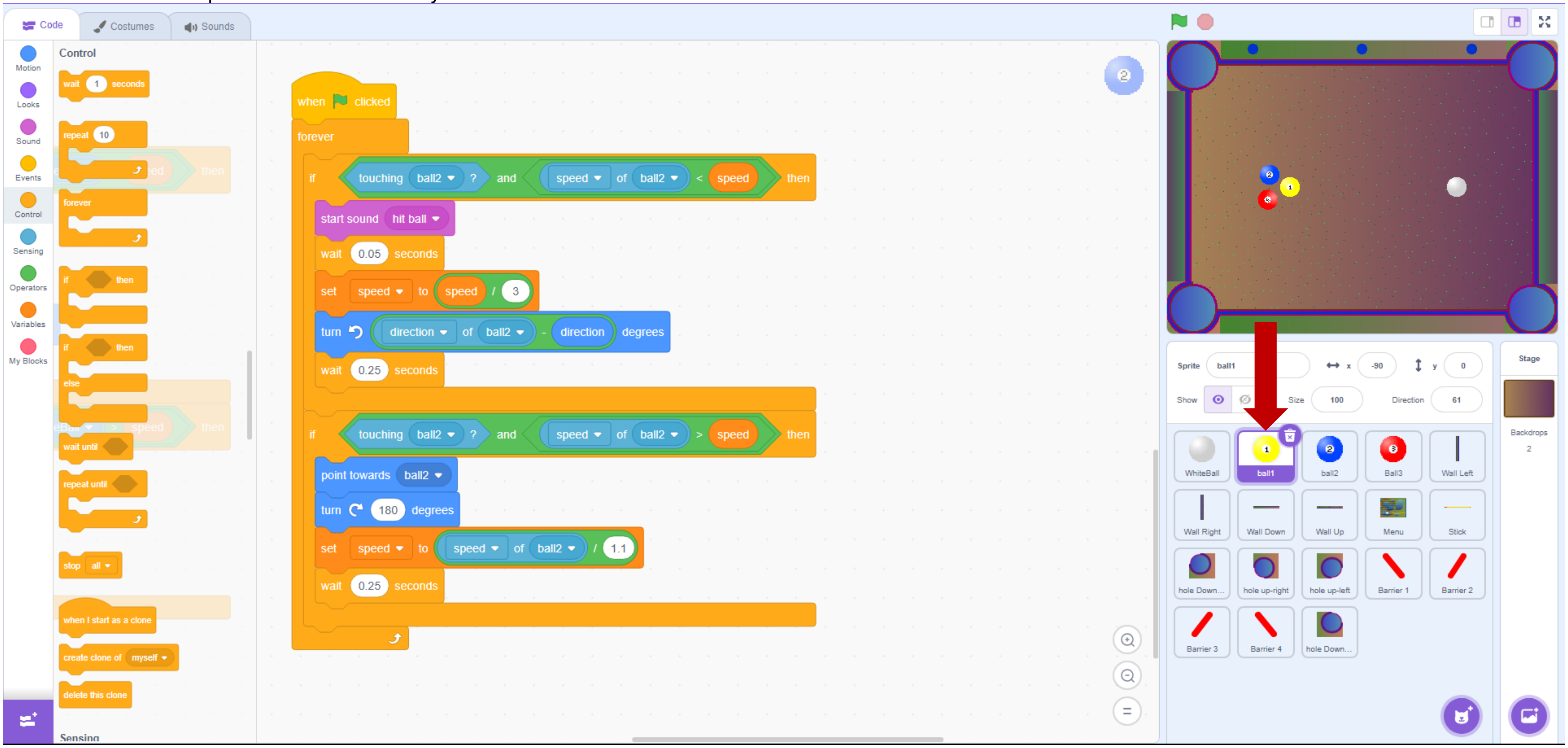
```
when p key pressed
forever
    if not touching color ? and touching hole Down-right ? or touching hole Down-Left ? or touching hole up-left ? or touching hole up-right ? or tou
        change Balls in by 1
        hide
```

Select the “white ball” sprite and create the same code block as the one in level 1. Change all instance of “ball 1” to “ball 2”.



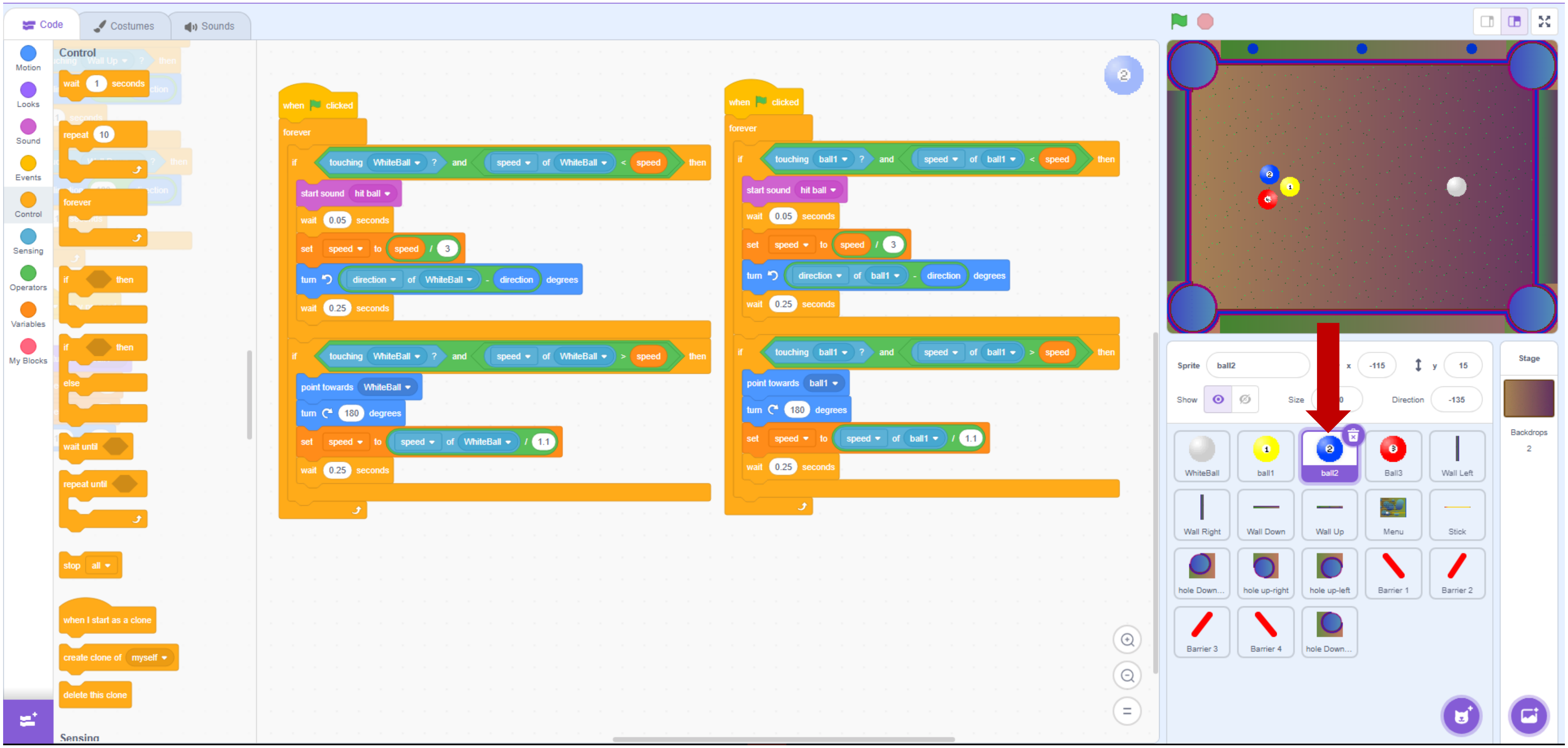
```
when ⚑ clicked
forever
    if < touching (ball2 ▾) ? > and < ( speed ▾ of (ball2 ▾) ) < (speed) > then
        start sound (hit ball ▾)
        wait (0.05) seconds
        set (speed ▾) to ( (speed) / (3) )
        turn ↺ ( ( direction ▾ of (ball2 ▾) ) - (direction) ) degrees
        wait (0.25) seconds
    if < touching (ball2 ▾) ? > and < ( speed ▾ of (ball2 ▾) ) > (speed) > then
        point towards (ball2 ▾)
        turn ↻ (15) degrees
        set (speed ▾) to ( ( speed ▾ of (ball2 ▾) ) / (1.1) )
        wait (0.25) seconds
```

Now repeat the same process in the ball 1 sprite and the ball 2 sprite. The correct code for the interactions for the white ball, ball 1, and ball 2 are provided below. Now try to create the code for ball 3.



```
when [flag] clicked
forever
    if <<touching ball2 ?> and <(speed of ball2) < (speed)>> then
        start sound hit ball
        wait 0.05 seconds
        set speed to ((speed) / 3)
        turn ↺ ((direction of ball2) - (direction)) degrees
        wait 0.25 seconds
    if <<touching ball2 ?> and <(speed of ball2) > (speed)>> then
        point towards ball2
        turn ↻ 180 degrees
        set speed to ((speed of ball2) / 1.1)
        wait 0.25 seconds
```

Sprite ball1 x -90 y 0
Show Size 100 Direction 61



```
when ⚑ clicked
forever
    if <<touching WhiteBall ?> and <(speed of WhiteBall) < (speed)>> then
        start sound hit ball
        wait 0.05 seconds
        set speed to ((speed) / 3)
        turn ↺ ((direction of WhiteBall) - (direction)) degrees
        wait 0.25 seconds
    if <<touching WhiteBall ?> and <(speed of WhiteBall) > (speed)>> then
        point towards WhiteBall
        turn ↻ 180 degrees
        set speed to ((speed of WhiteBall) / 1.1)
        wait 0.25 seconds
```

```
when ⚑ clicked
forever
    if <<touching ball1 ?> and <(speed of ball1) < (speed)>> then
        start sound hit ball
        wait 0.05 seconds
        set speed to ((speed) / 3)
        turn ↺ ((direction of ball1) - (direction)) degrees
        wait 0.25 seconds
    if <<touching ball1 ?> and <(speed of ball1) > (speed)>> then
        point towards ball1
        turn ↻ 180 degrees
        set speed to ((speed of ball1) / 1.1)
        wait 0.25 seconds
```

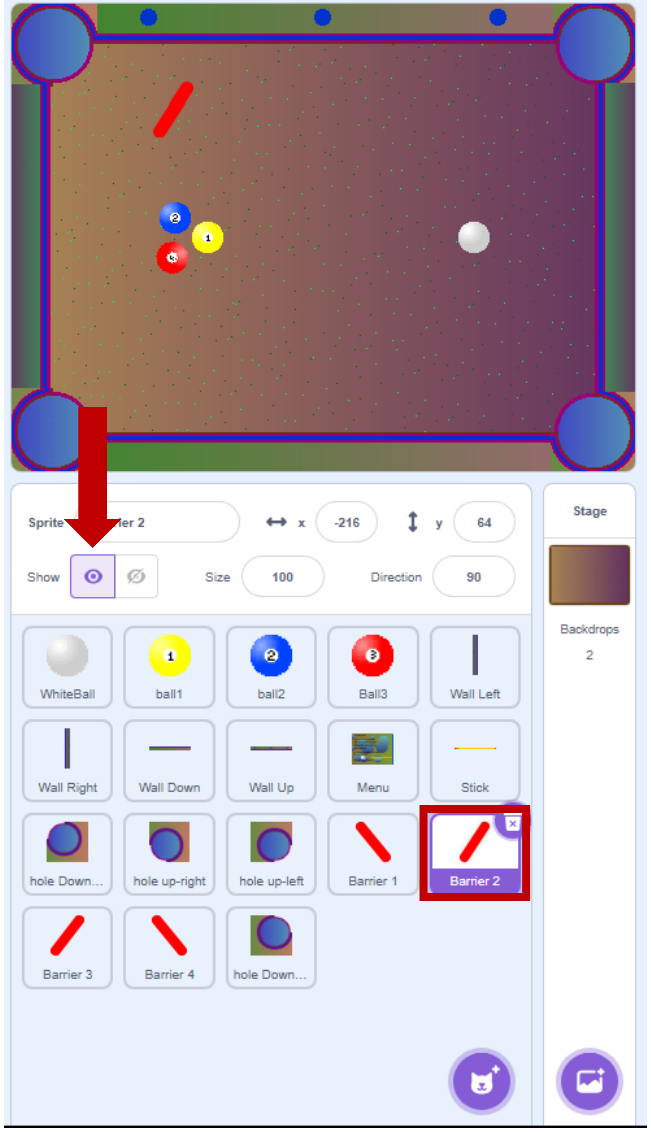
Sprite ball2 x -115 y 15

Show Size Direction -135

https://tapggc.org/

This section will cover level 3 of the game

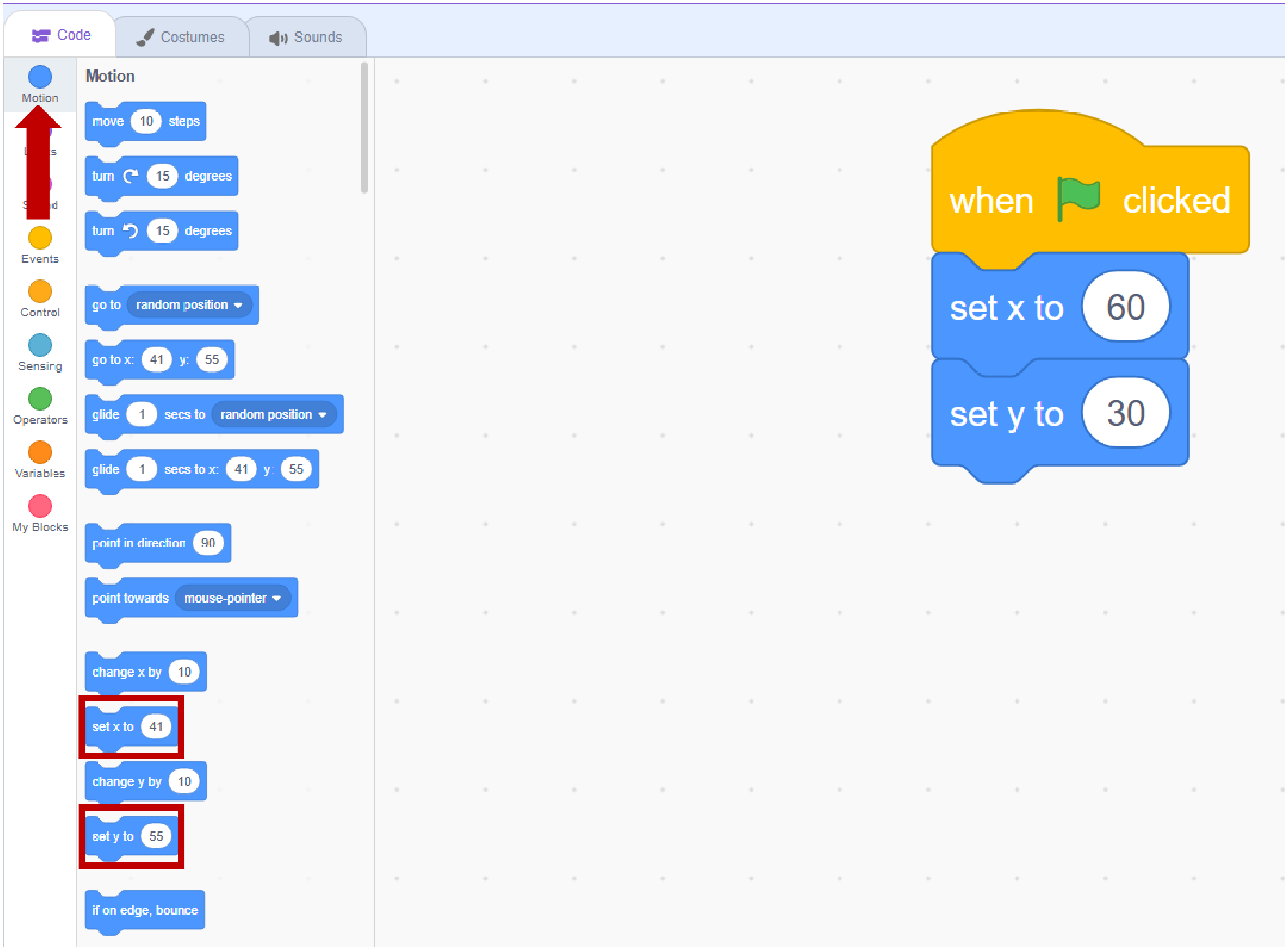
Go to each of the barriers and click on the eyeball to make them visible.



Sprite ...ier 2 ↔ x -216 ↕ y 64

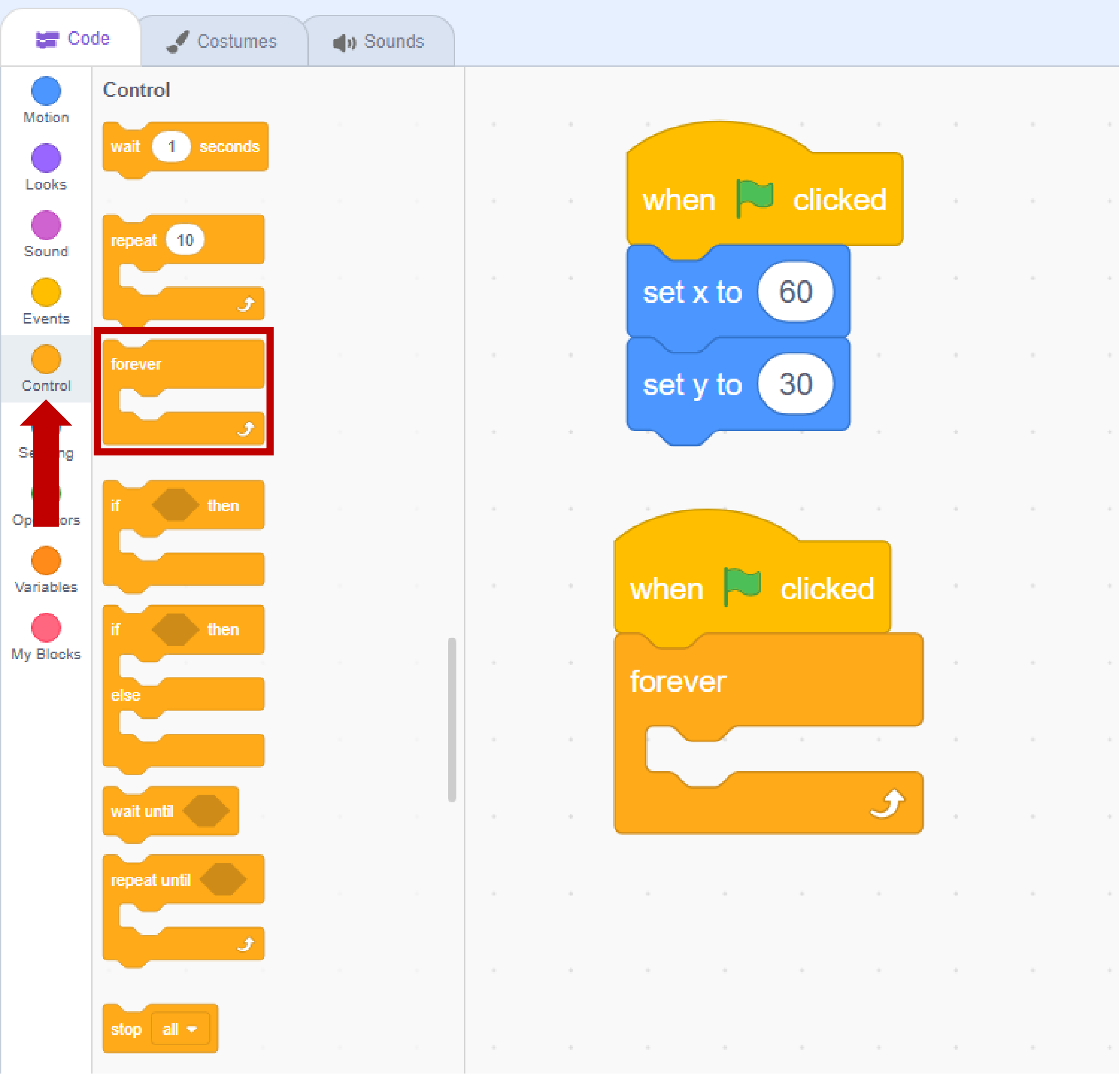
Show Size 100 Direction 90

Go to the “Events” tab and select the “when flag pressed” block. Then go to the “Motion” tab and grab a “set x to __” and a “set y to __” block. Change the x to 60 and the y to 30.



```
when 🏁 clicked
set x to 60
set y to 30
```

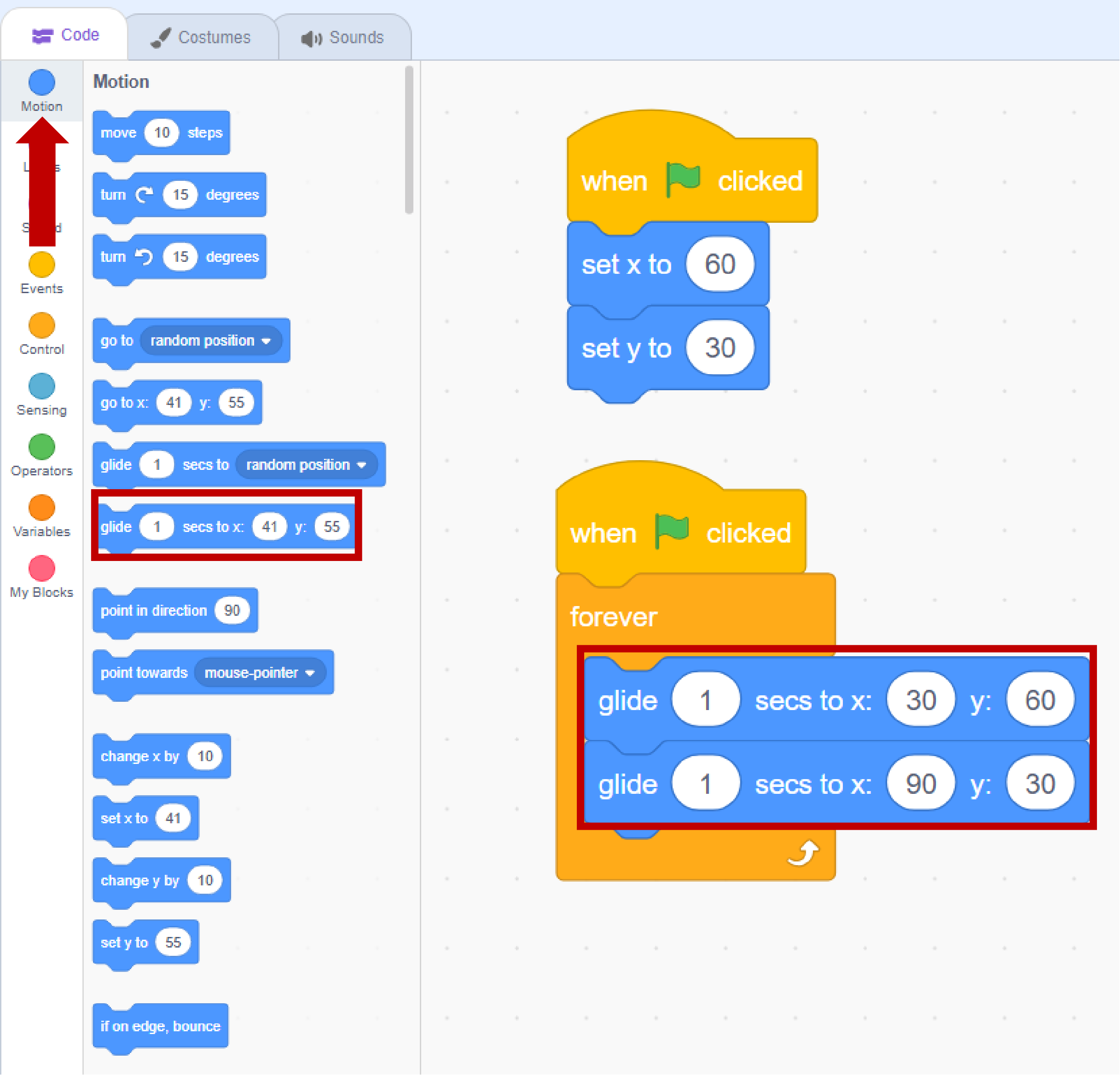
Go back to the “Events” tab and grab the “when flag pressed” block. Then go to the “Control” tab and grab a forever loop.



```
when 🏁 clicked
set x to 60
set y to 30
```

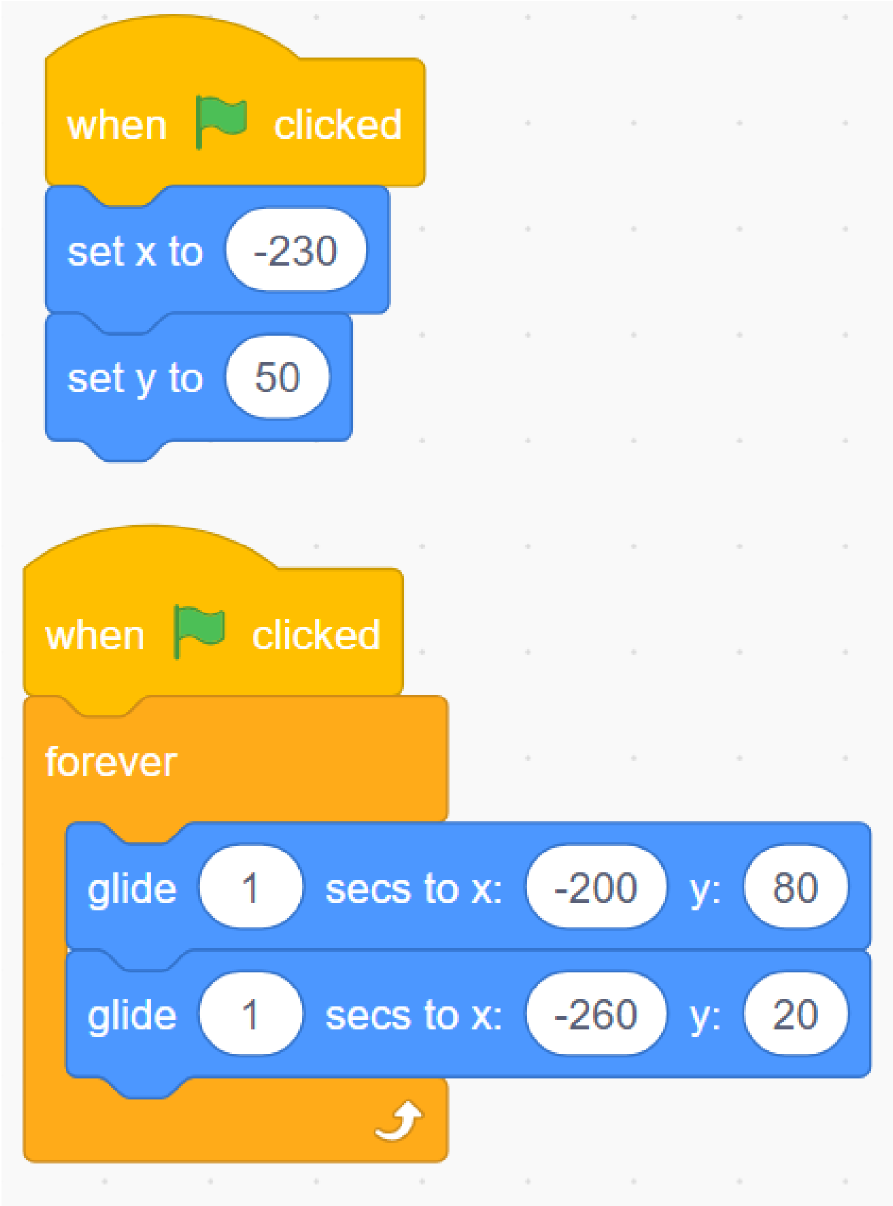
```
when 🏁 clicked
forever
```

Go to the “Motion” tab and grab two “glide __ secs to x:__ y:__” and insert them both inside the forever loop. Set the glide to 1 sec for both. Set the x to 30 and y to 60 for one of the glide blocks. Then set the x to 90 and y to 30 for the other.



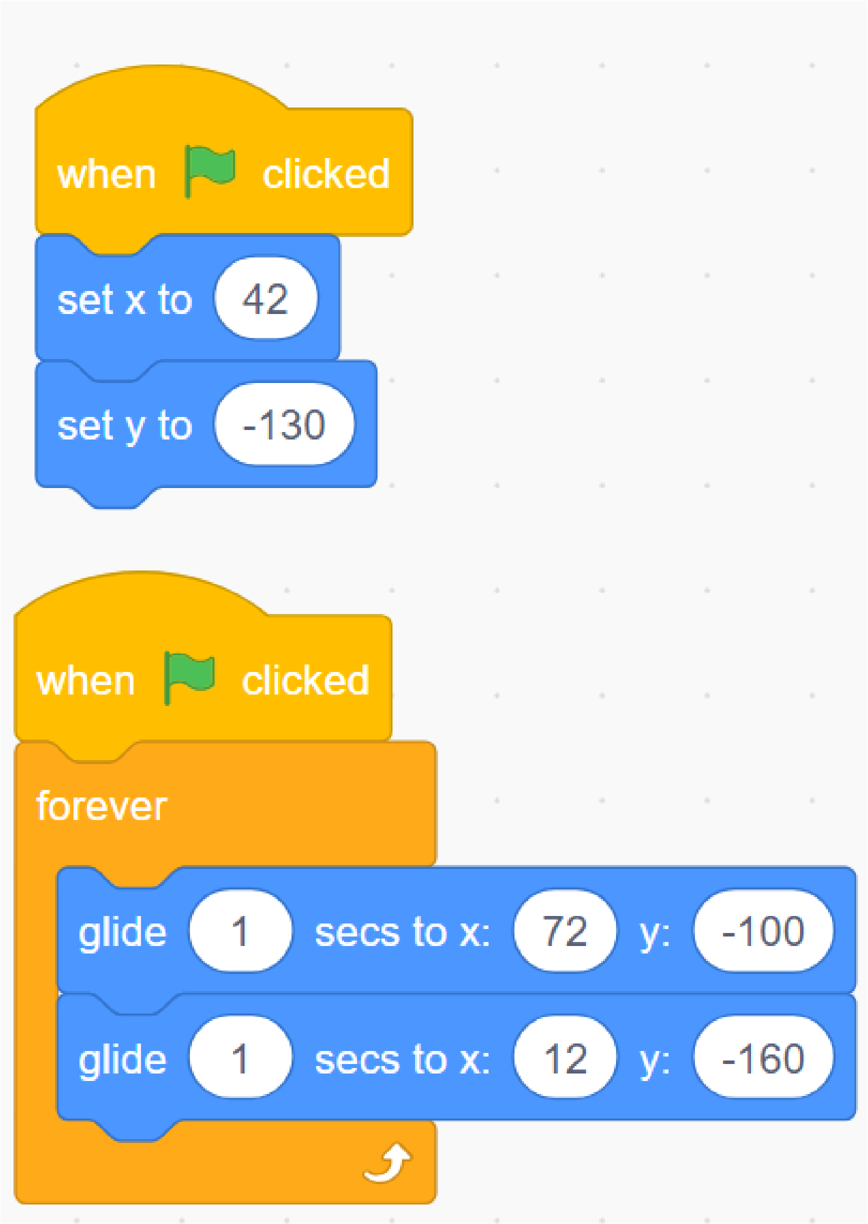
```
when ⚑ clicked
set x to 60
set y to 30

when ⚑ clicked
forever
    glide 1 secs to x: 30 y: 60
    glide 1 secs to x: 90 y: 30
```

This is a similar process for the remaining barriers. Below is the code for the barrier 2.



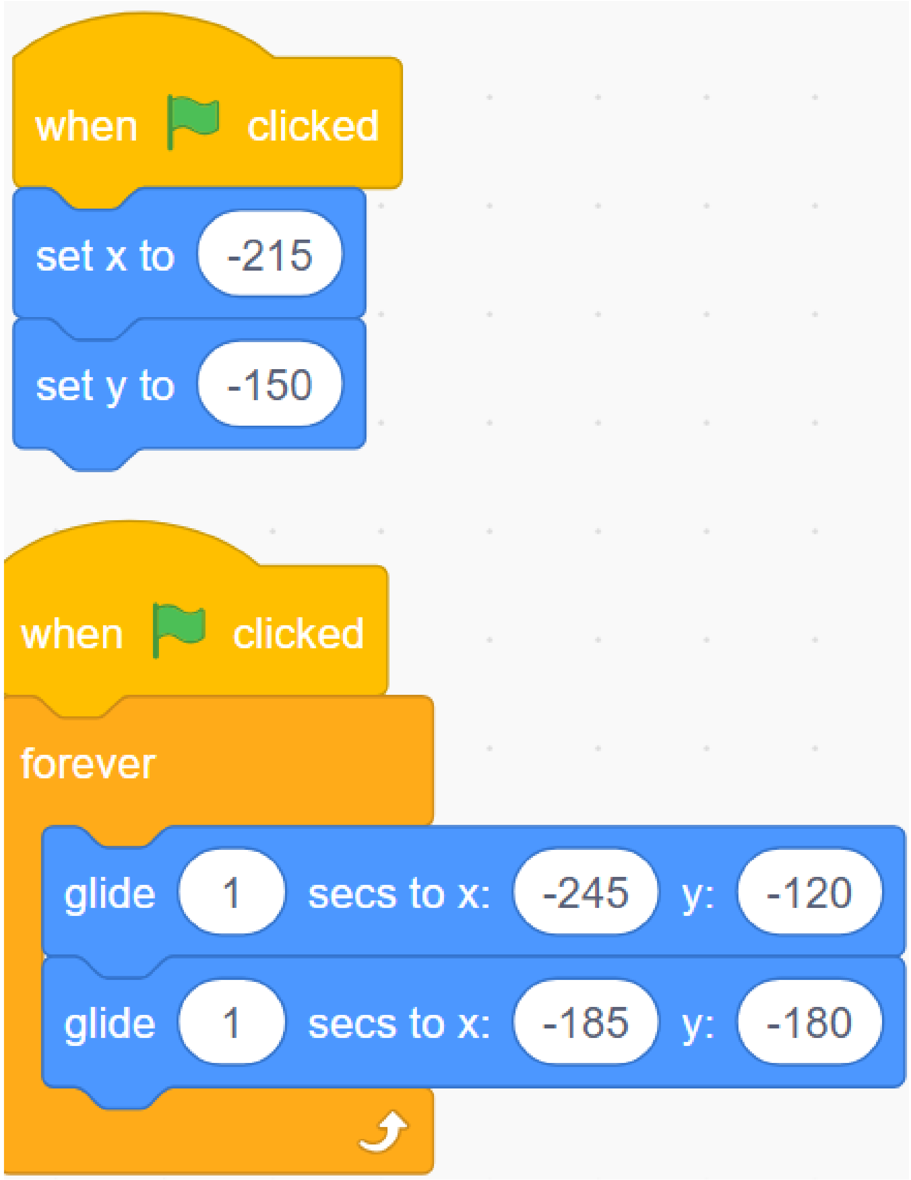
```
when ⚑ clicked
set x to (-230)
set y to (50)

when ⚑ clicked
forever
    glide (1) secs to x: (-200) y: (80)
    glide (1) secs to x: (-260) y: (20)
```

Below is the code for barrier 3.



```
when 🏁 clicked
set x to 42
set y to -130

when 🏁 clicked
forever
    glide 1 secs to x: 72 y: -100
    glide 1 secs to x: 12 y: -160
```

Below is the code for barrier 4.



```
when ⚑ clicked
set x to -215
set y to -150

when ⚑ clicked
forever
    glide 1 secs to x: -245 y: -120
    glide 1 secs to x: -185 y: -180
```