

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.

The screenshot shows the GitHub repository page for 'makeysrace' by 'TAP-GGC'. The repository is public and has 2 branches and 0 tags. The 'code' folder is highlighted with a red box. Below the repository name, there is a table of files and folders:

| File/Folder | Commit Message | Commit Date |
|-------------|--|--------------|
| 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 |

Below the table, there is a 'README' section with the title 'Using Scratch to Teach Event-Driven Coding Skills' and a sub-section 'About'.

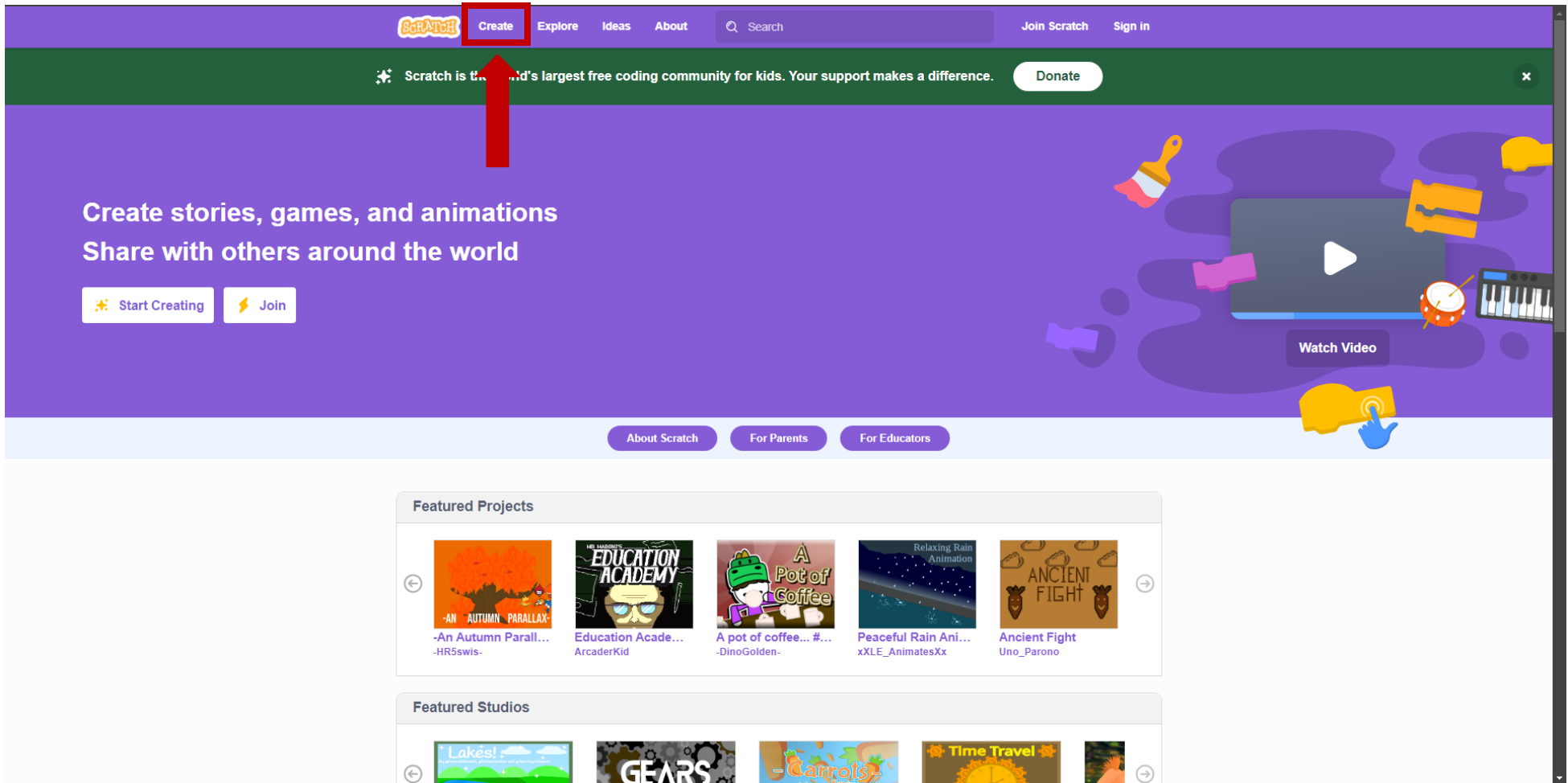


The screenshot shows the GitHub interface for the repository 'TAP-GGC / makeysrace'. The navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and Insights. The breadcrumb trail indicates the current location is 'makeysrace / code'. A notification from user 'tlam8' about deleting a file is visible. A status bar indicates the current branch is 62 commits ahead of and 1 commit behind the 'master' branch. Below this is a table listing repository files:

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

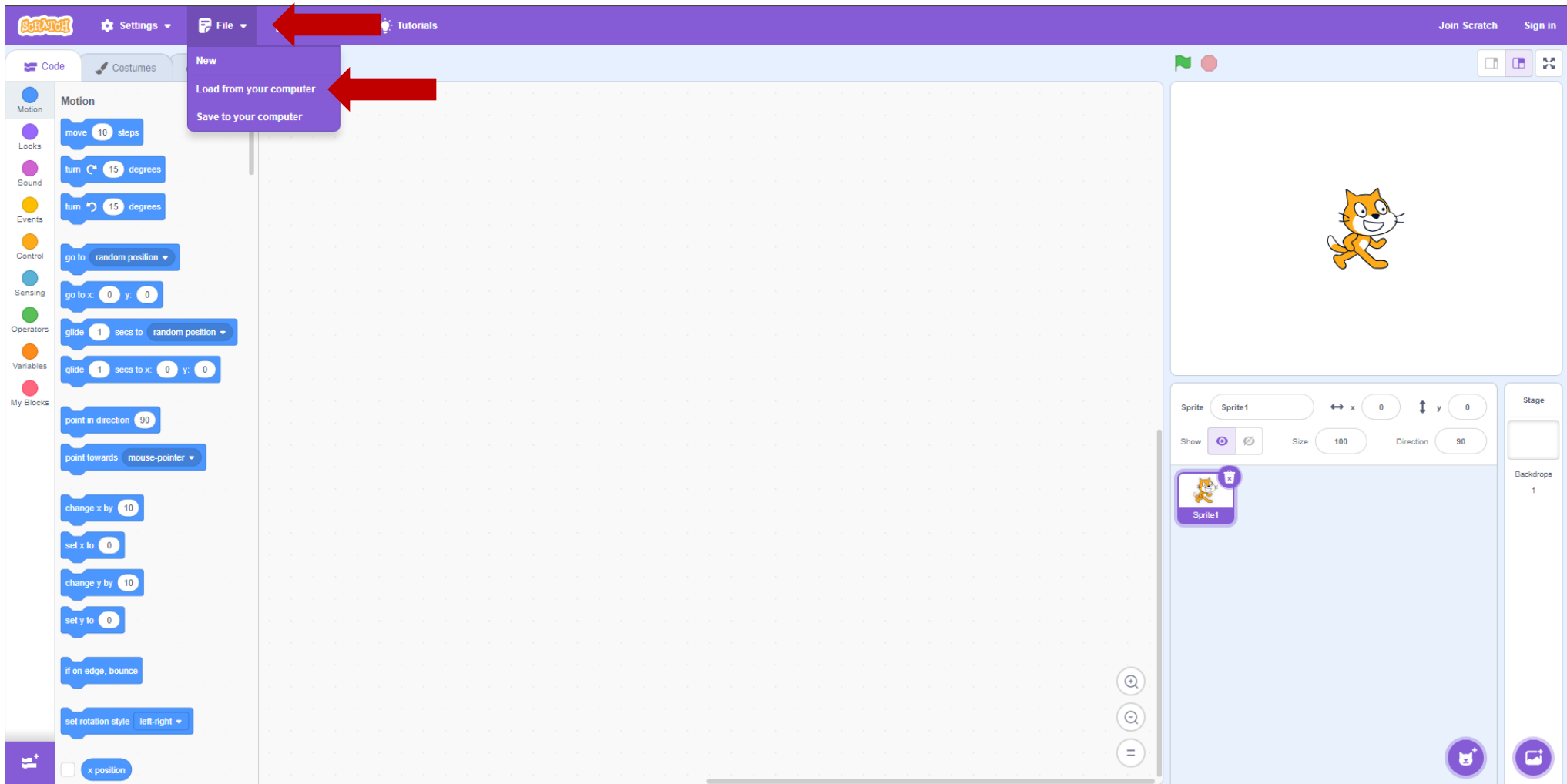
The row for 'IncompleteCode.sb3' is highlighted with a red border.

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



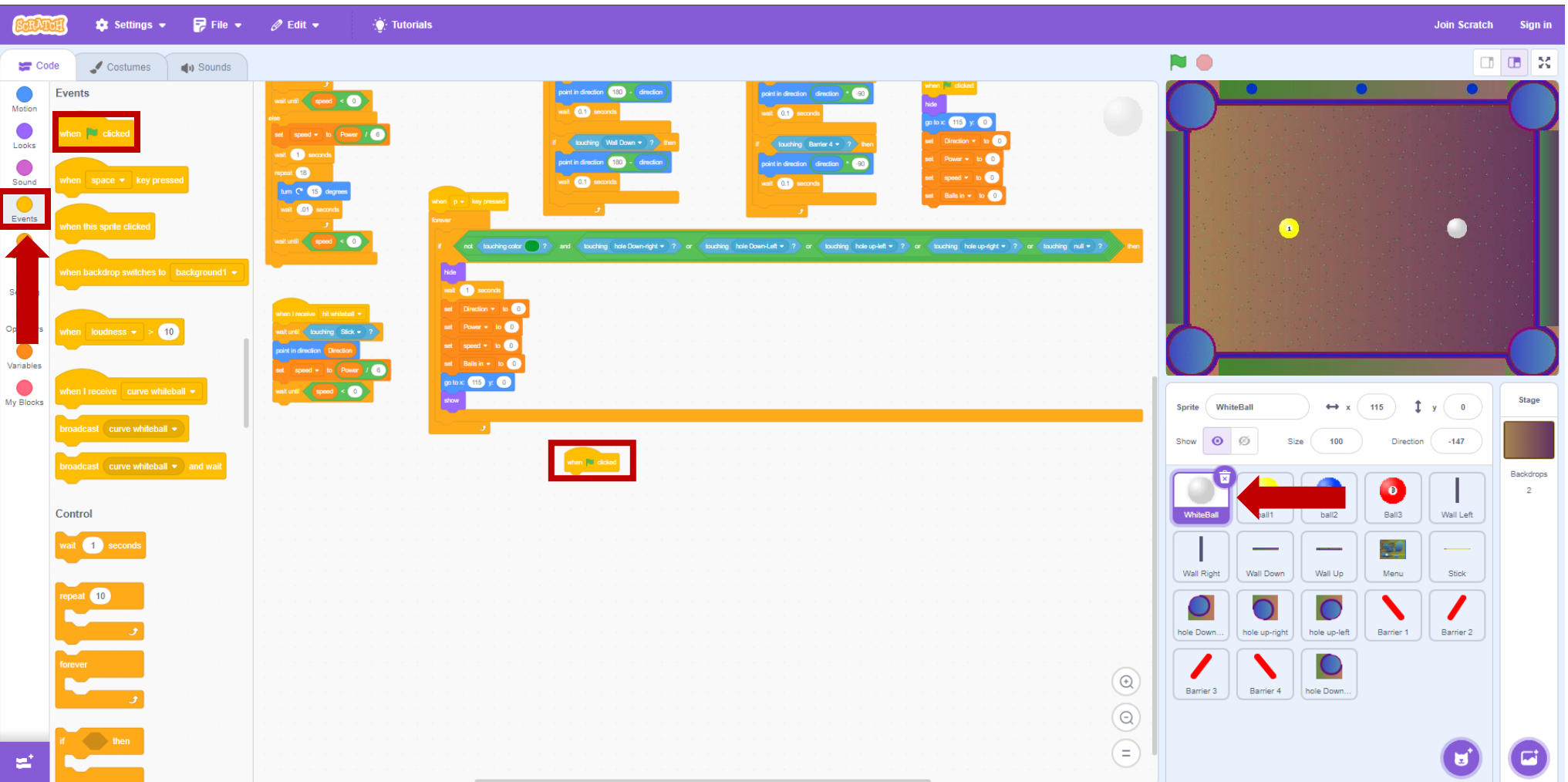
<https://tapggc.org/>

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”.

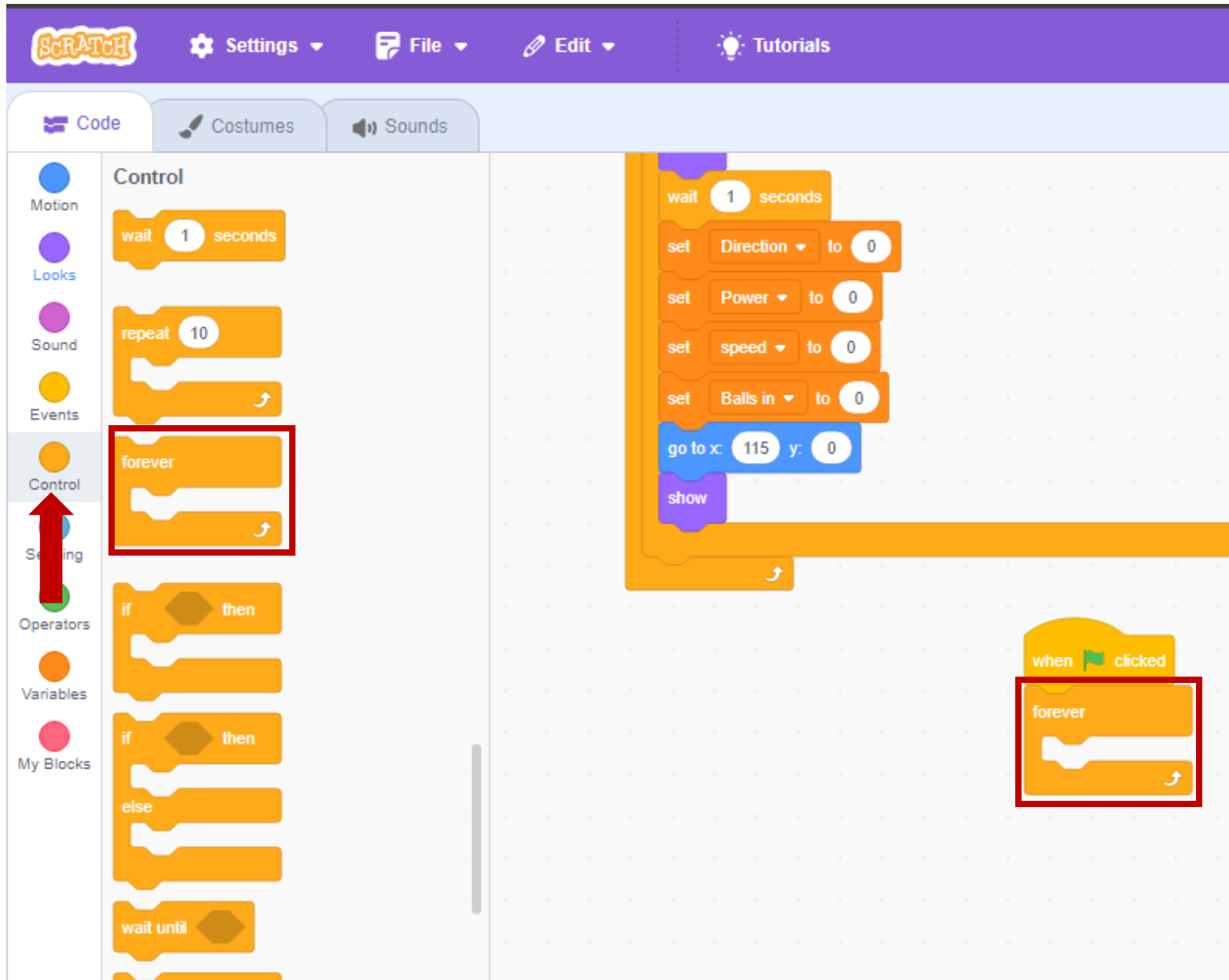


This section will cover level 1 of the game.

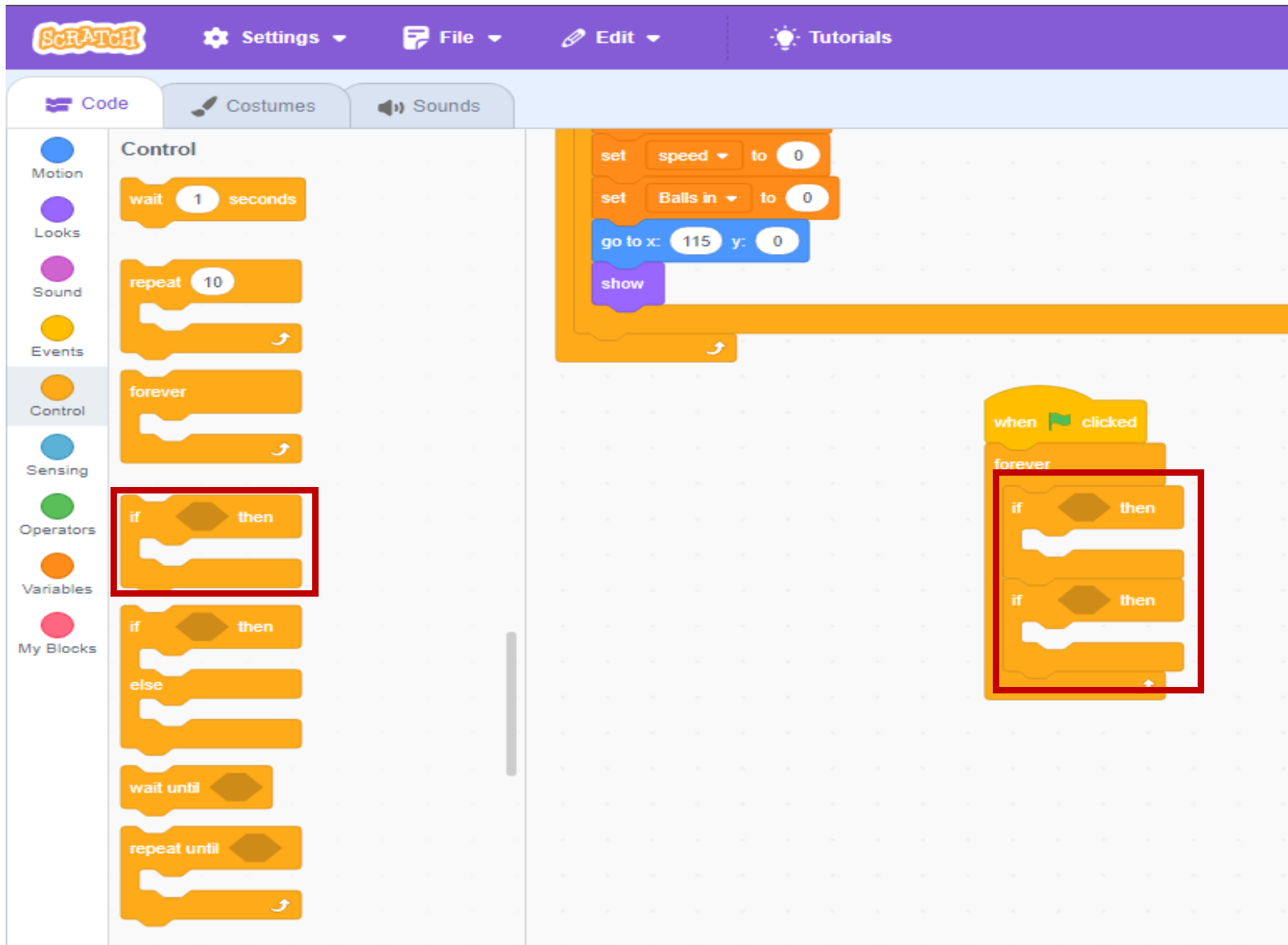
Make sure that the white ball is the selected sprite. Now click on “Events” on the left side of the screen. Then click and drag the “when flag clicked” and drop on an empty space on the canvas.



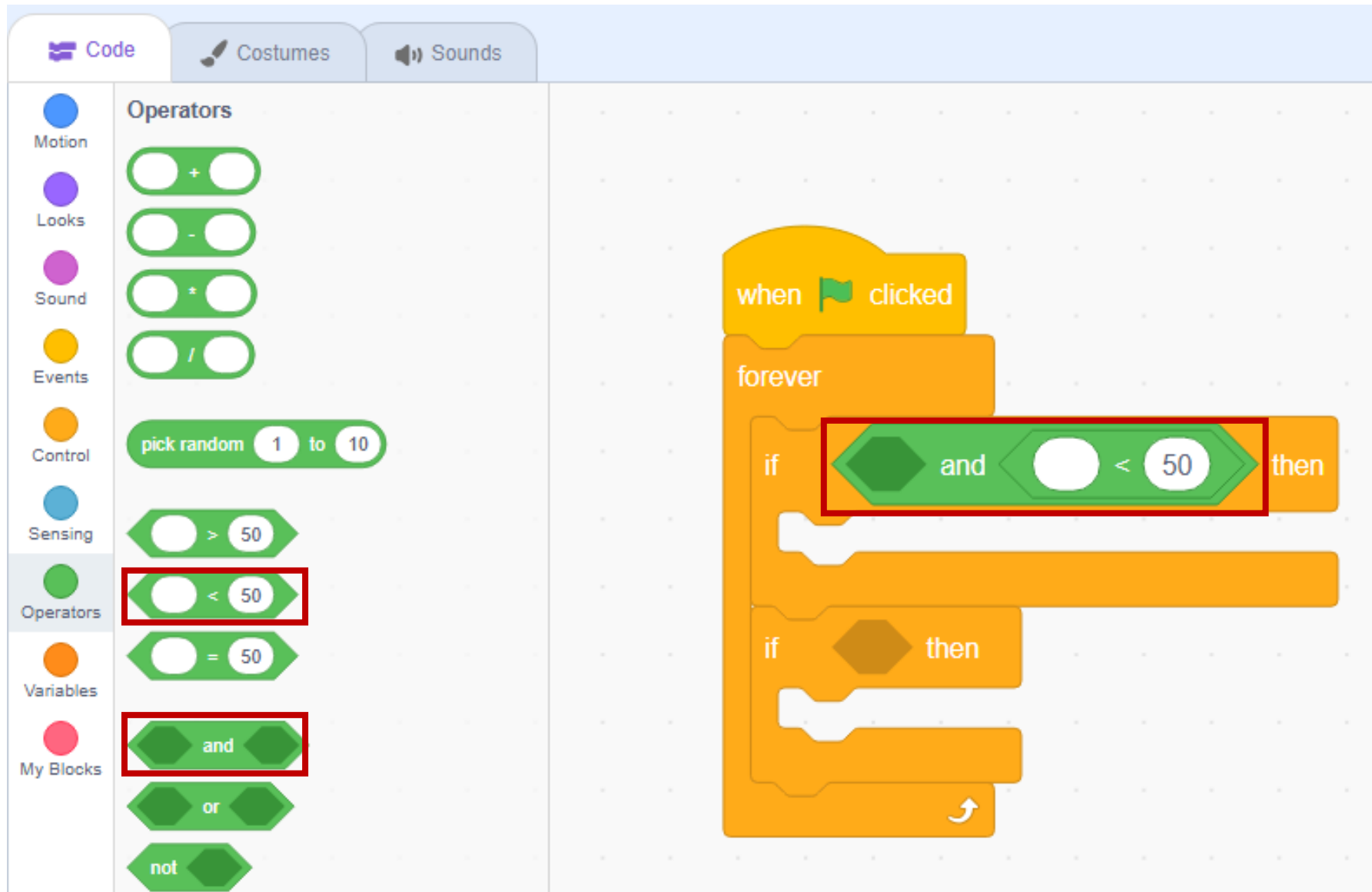
Now go to “Control” and grab the “forever” loop and attach it to code block.



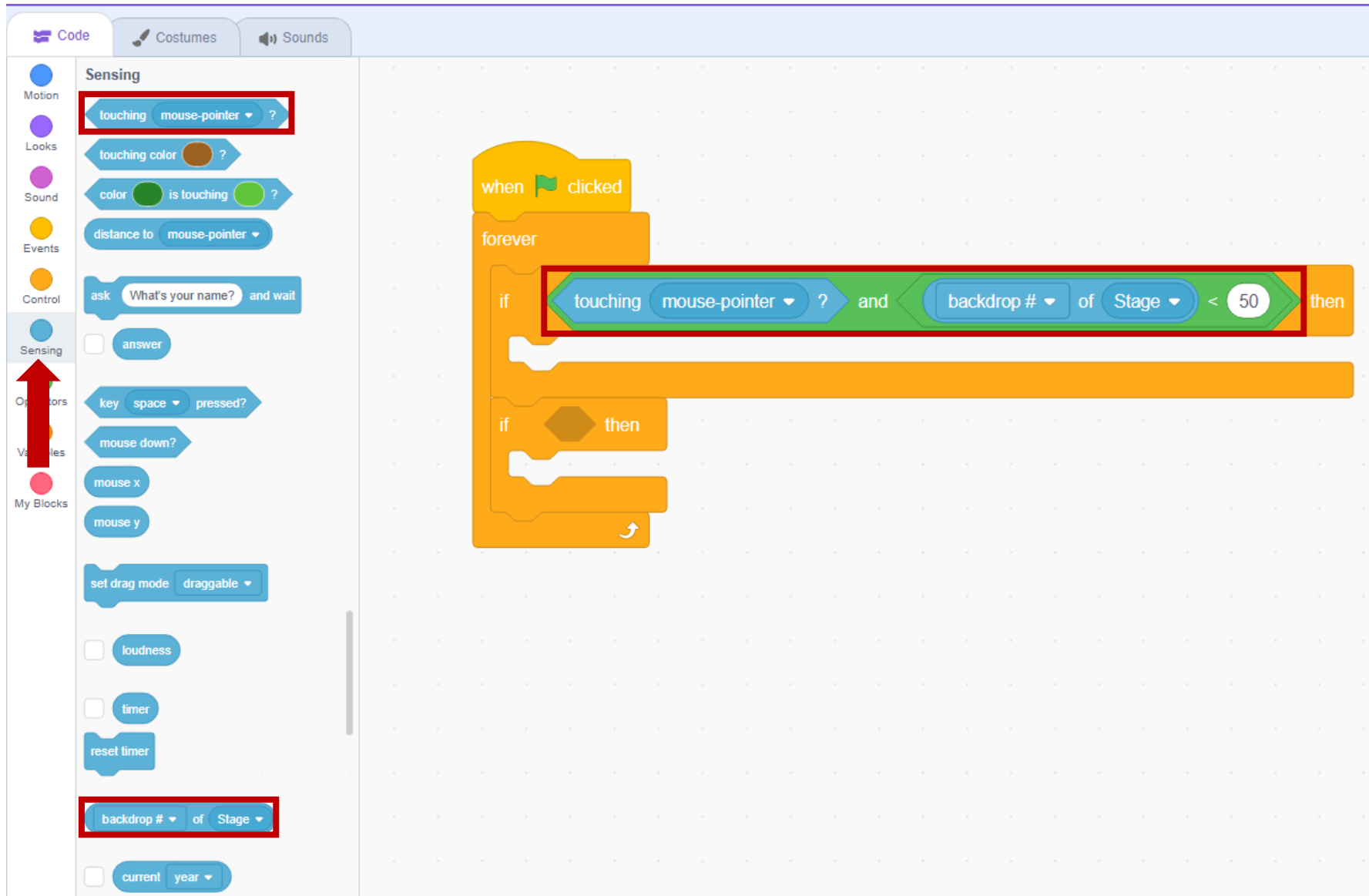
Now grab 2 “if-then” blocks and put them into the “forever loop” like shown below.

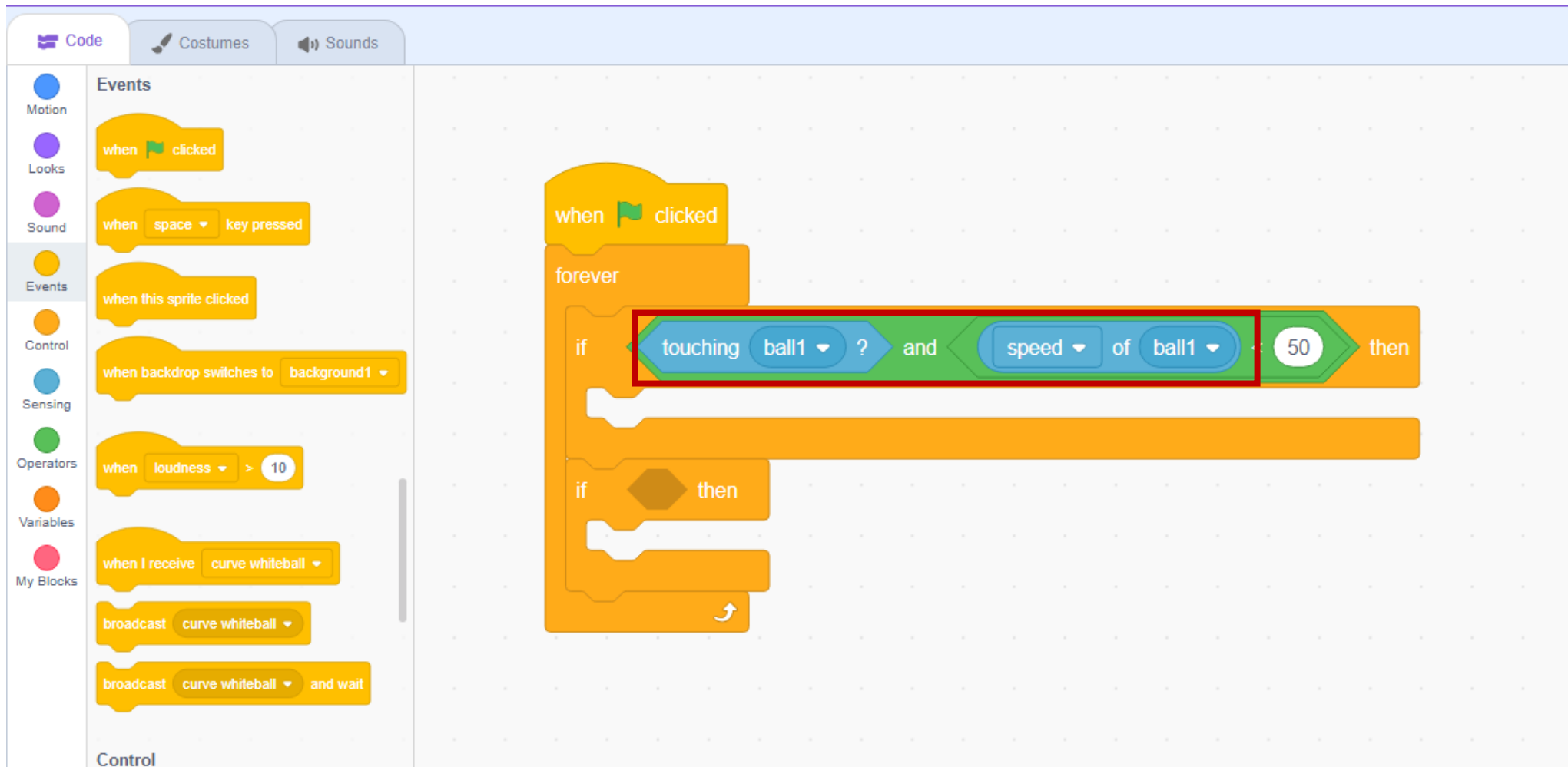


We will focus on the top “if-then” block for now. Go to “Operators”. Then grab the “__ and __” block and put it into the blank in the “if-then” block. Then grab the “__ < 50” block and put it into the second blank of the “__ and __” block.

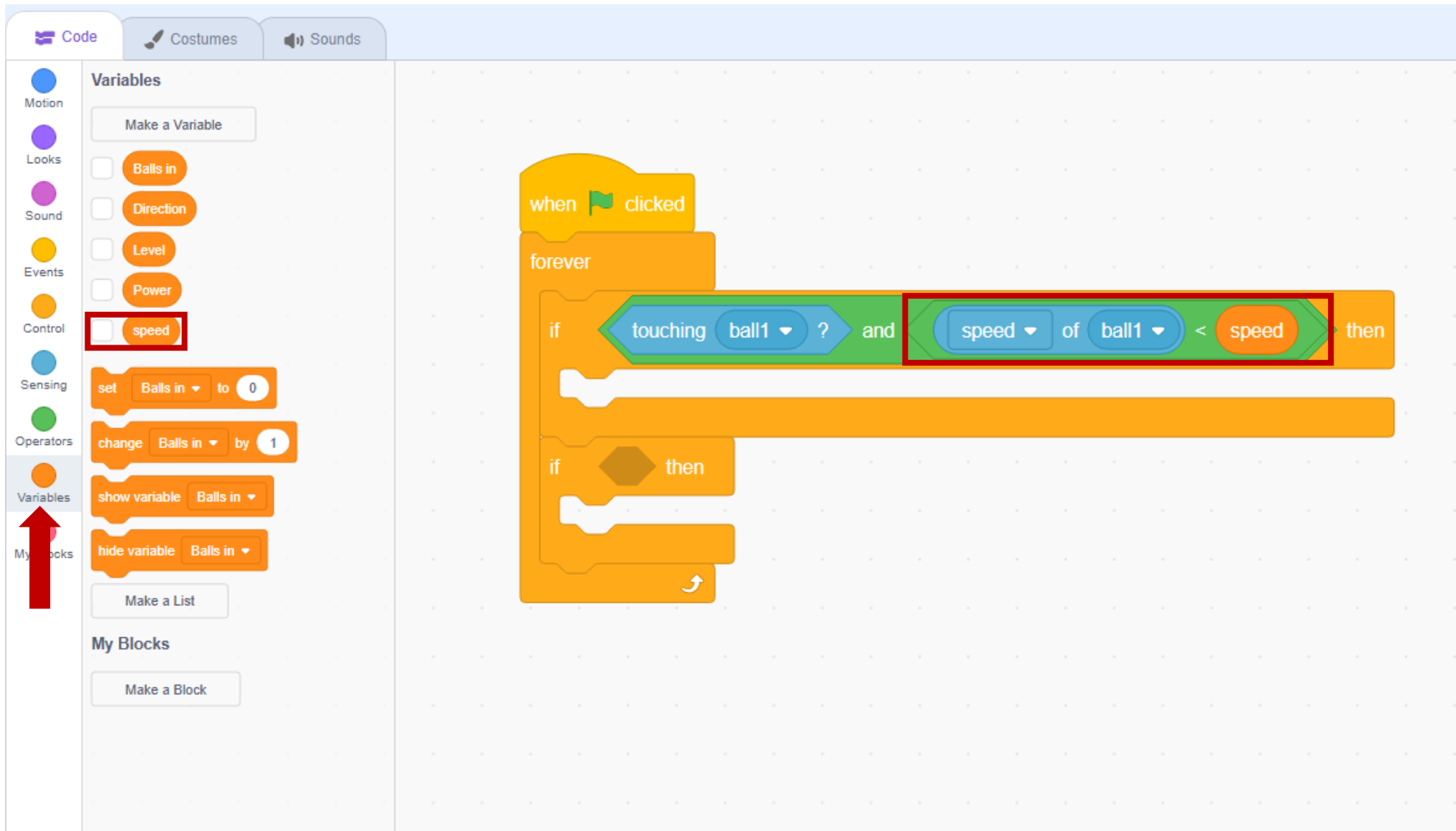


Go to “Sensing”. Grab the “touching mouse pointer?” block and insert it into the first blank of the “if-then” block. Then grab a “backdrop # of Stage” block and insert it into the “__ < 50” block. Change the mouse pointer to “ball 1” and “backdrop # of Stage” to “speed of ball 1”.



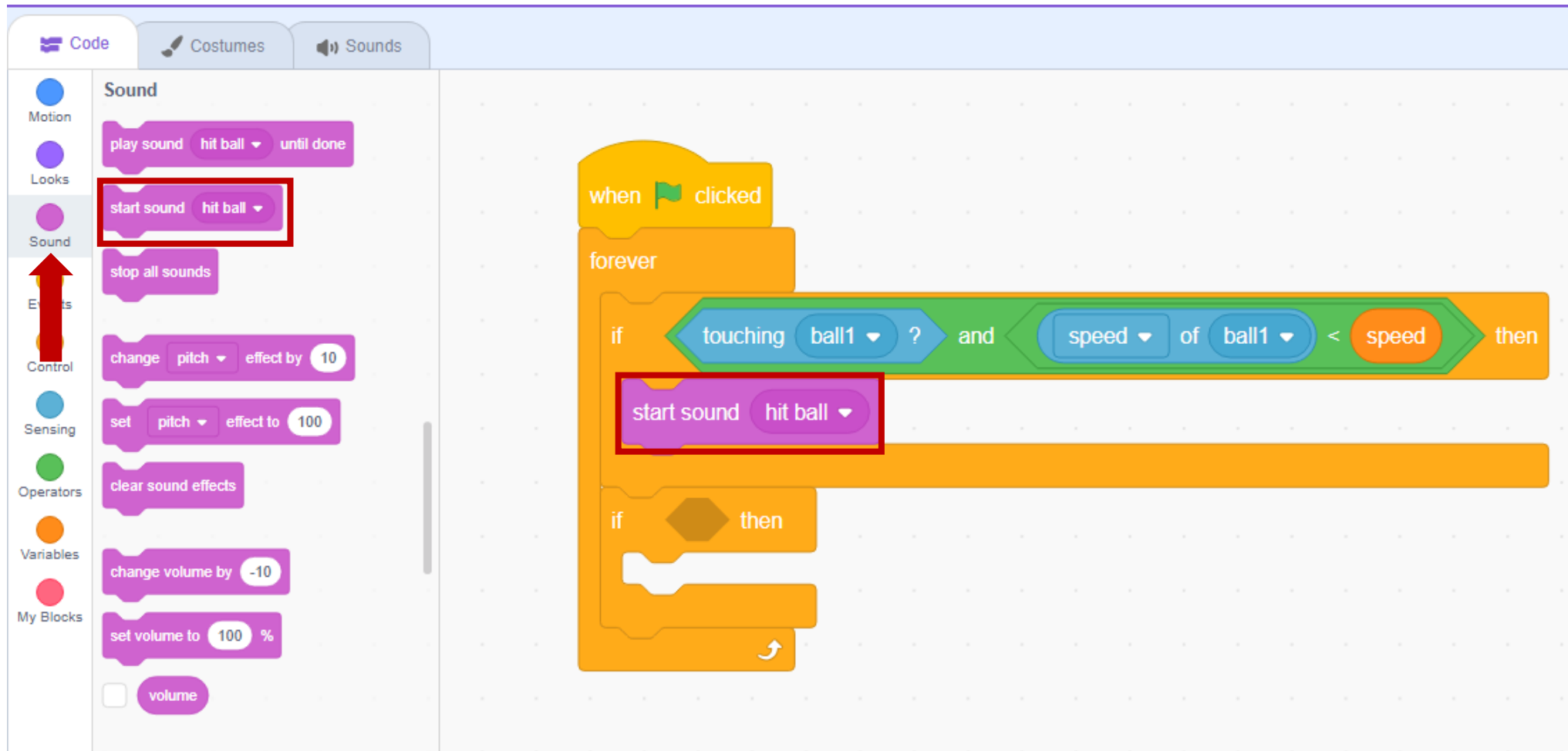


Now go to the “Variables” tab and grab the “speed” variable. Insert it into the “__ < 50” block.

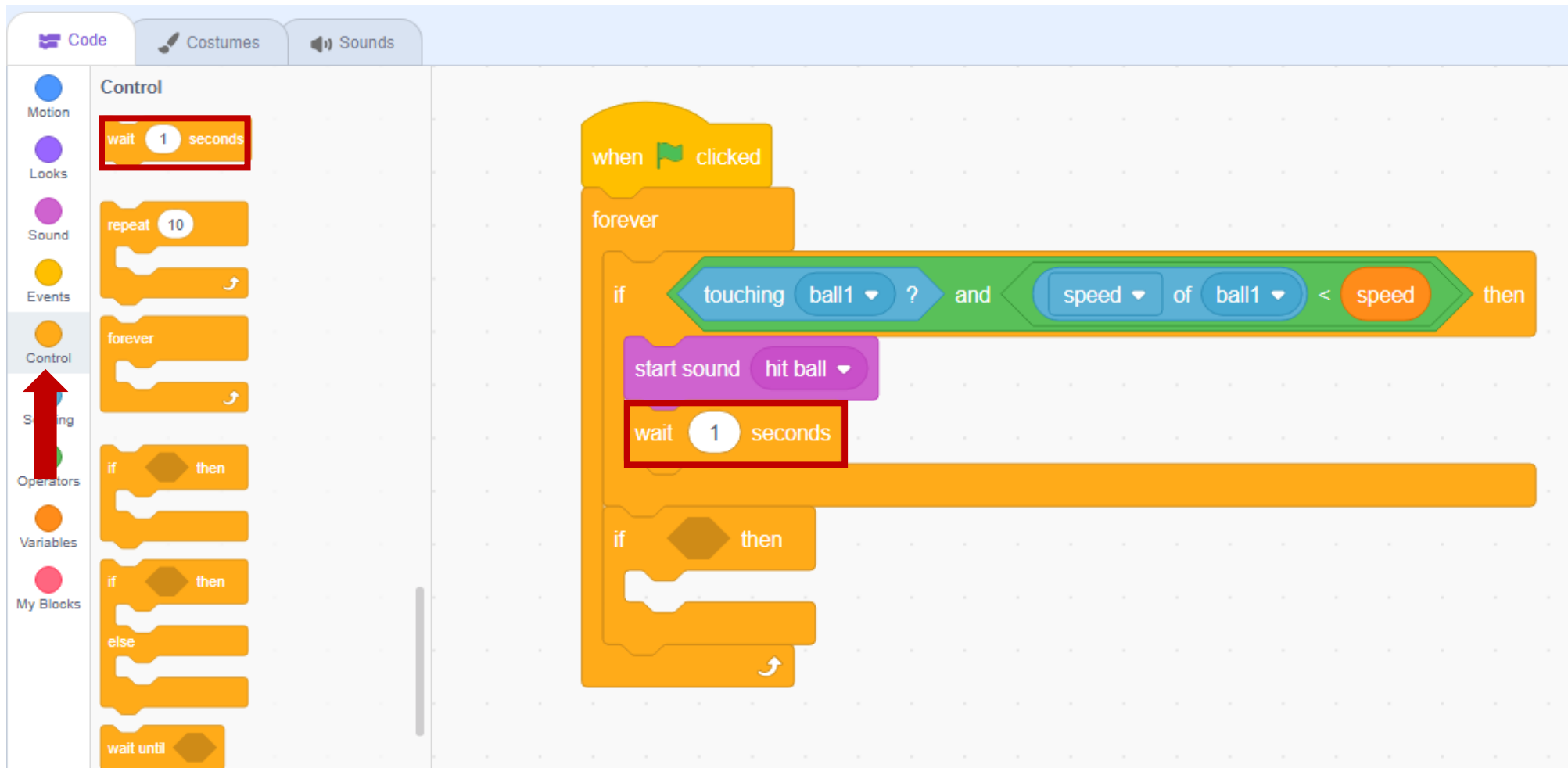


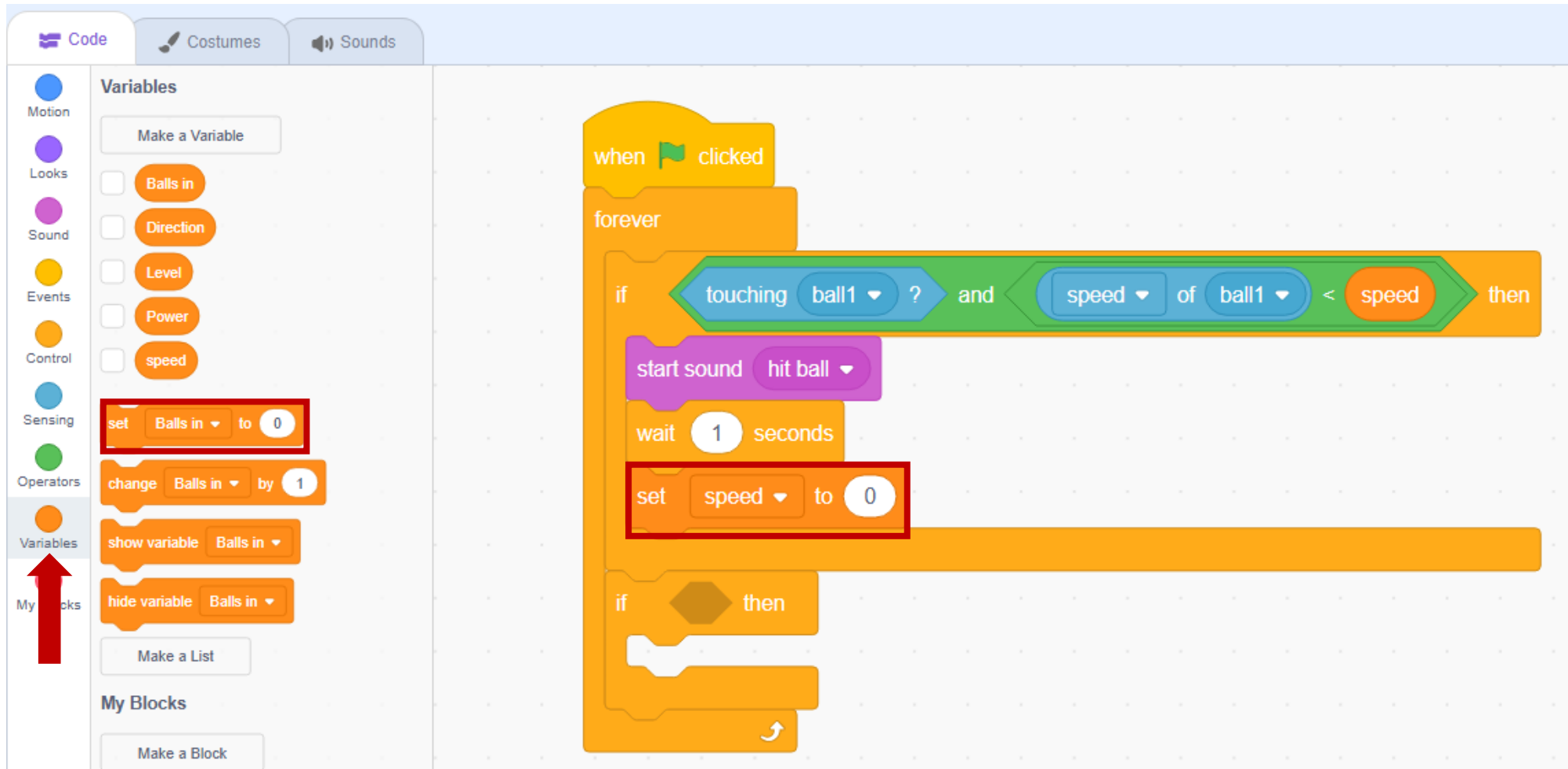
The image shows the Scratch code editor interface. On the left, the 'Variables' tab is selected, and the 'speed' variable is highlighted with a red box. A red arrow points to the 'Variables' tab. The main workspace shows a script starting with 'when green flag clicked', followed by a 'forever' loop. Inside the loop, there is an 'if' block with the condition 'touching ball1?' and 'speed of ball1 < speed'. The 'then' block of the 'if' statement is currently empty.

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



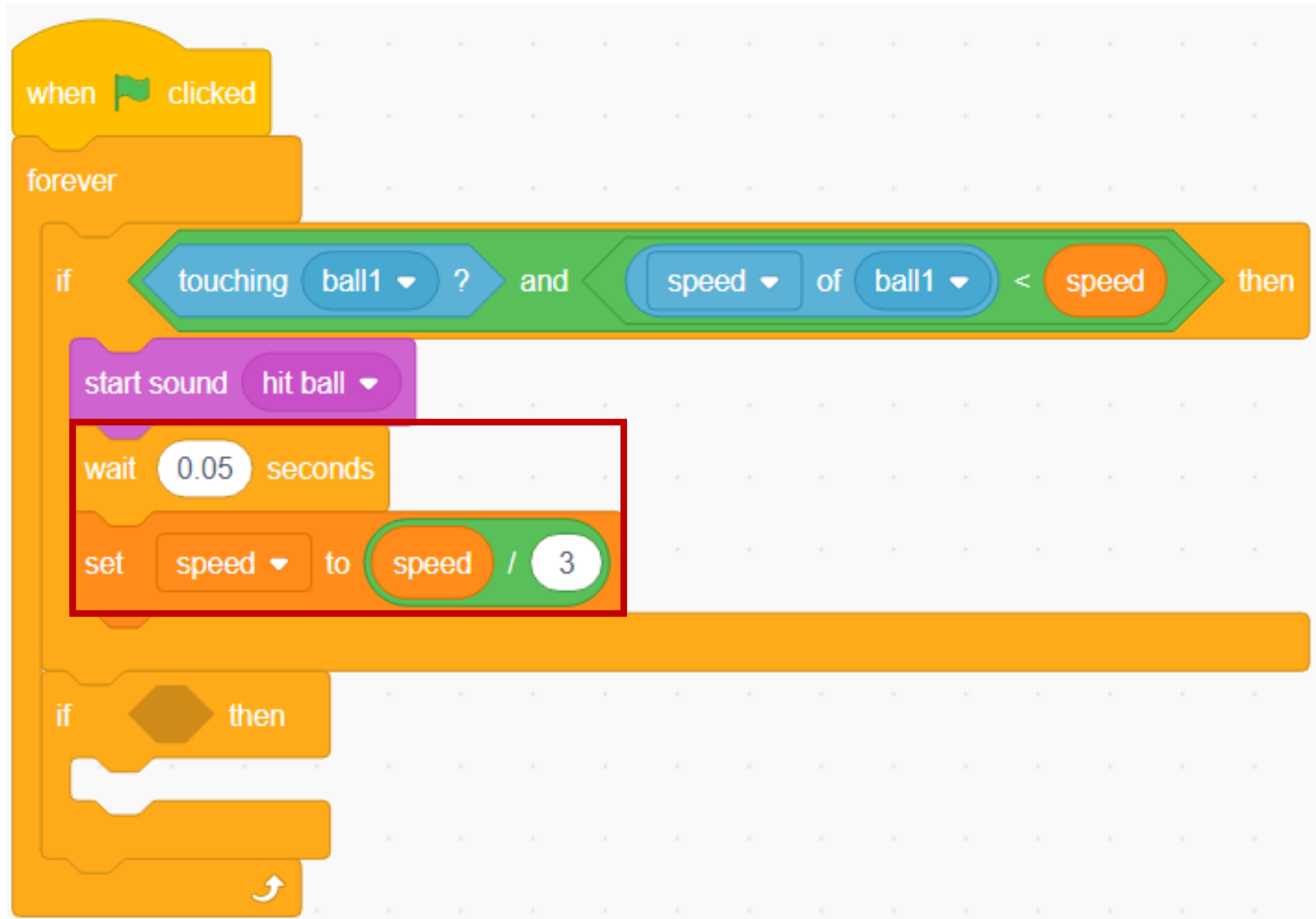
Go to “Control” and then grab the “wait 1 seconds” block and insert it into the “if-then” statement. Then go to “Variables” and grab the “set balls in to 0” block and put it into the “if-then” statement. Change the variable to speed.



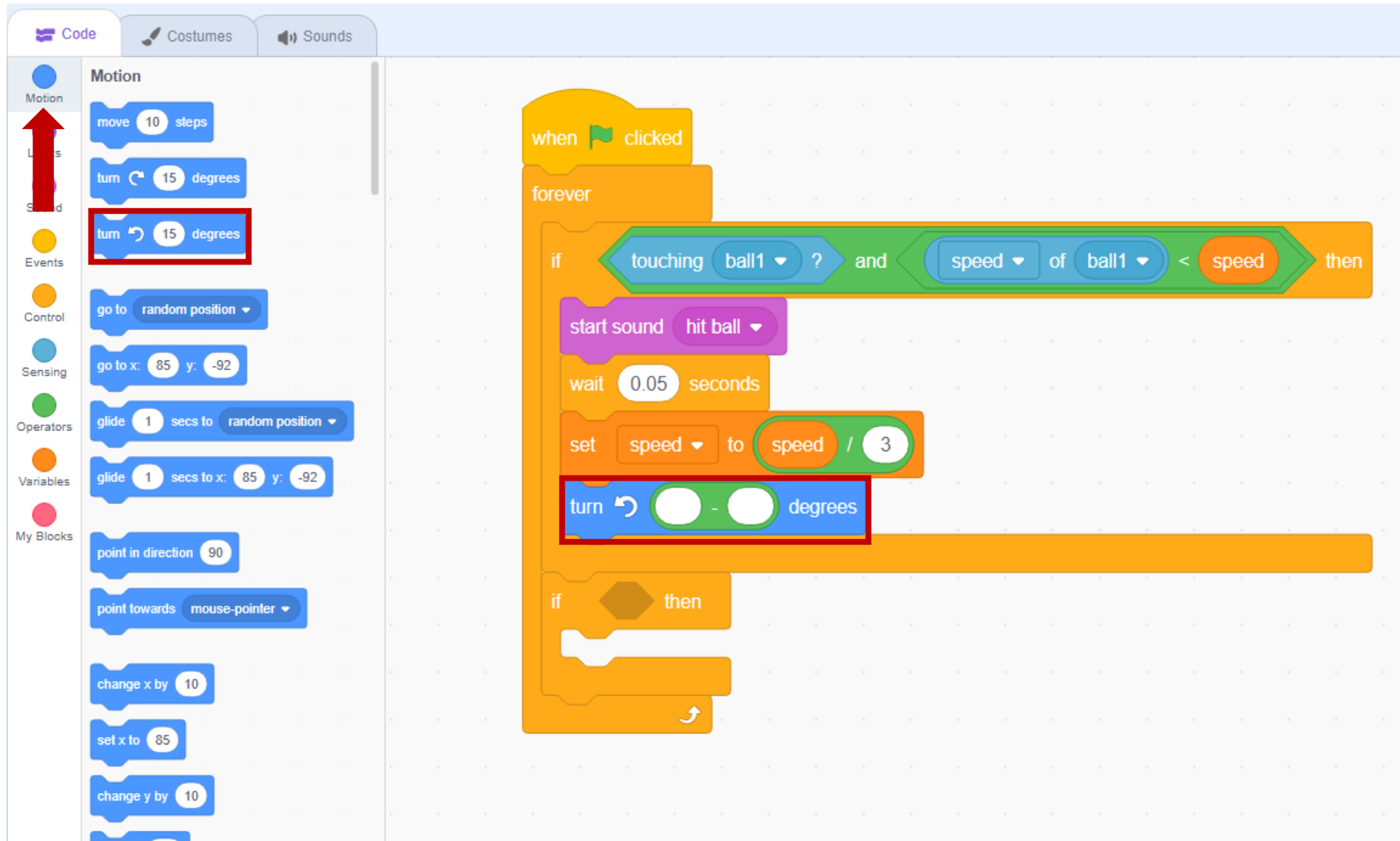


The image shows the Scratch code editor interface. On the left sidebar, the 'Variables' category is highlighted with a red arrow. The 'Variables' panel shows a variable named 'Balls in' set to 0. The main workspace contains a script starting with 'when green flag clicked', followed by a 'forever' loop. Inside the loop, there is an 'if' condition: 'touching ball1?' and 'speed of ball1 < speed'. If true, it triggers 'start sound hit ball', 'wait 1 seconds', and 'set speed to 0'. A red arrow points to the 'Variables' category in the left sidebar, and two red boxes highlight the 'set Balls in to 0' and 'set speed to 0' blocks.

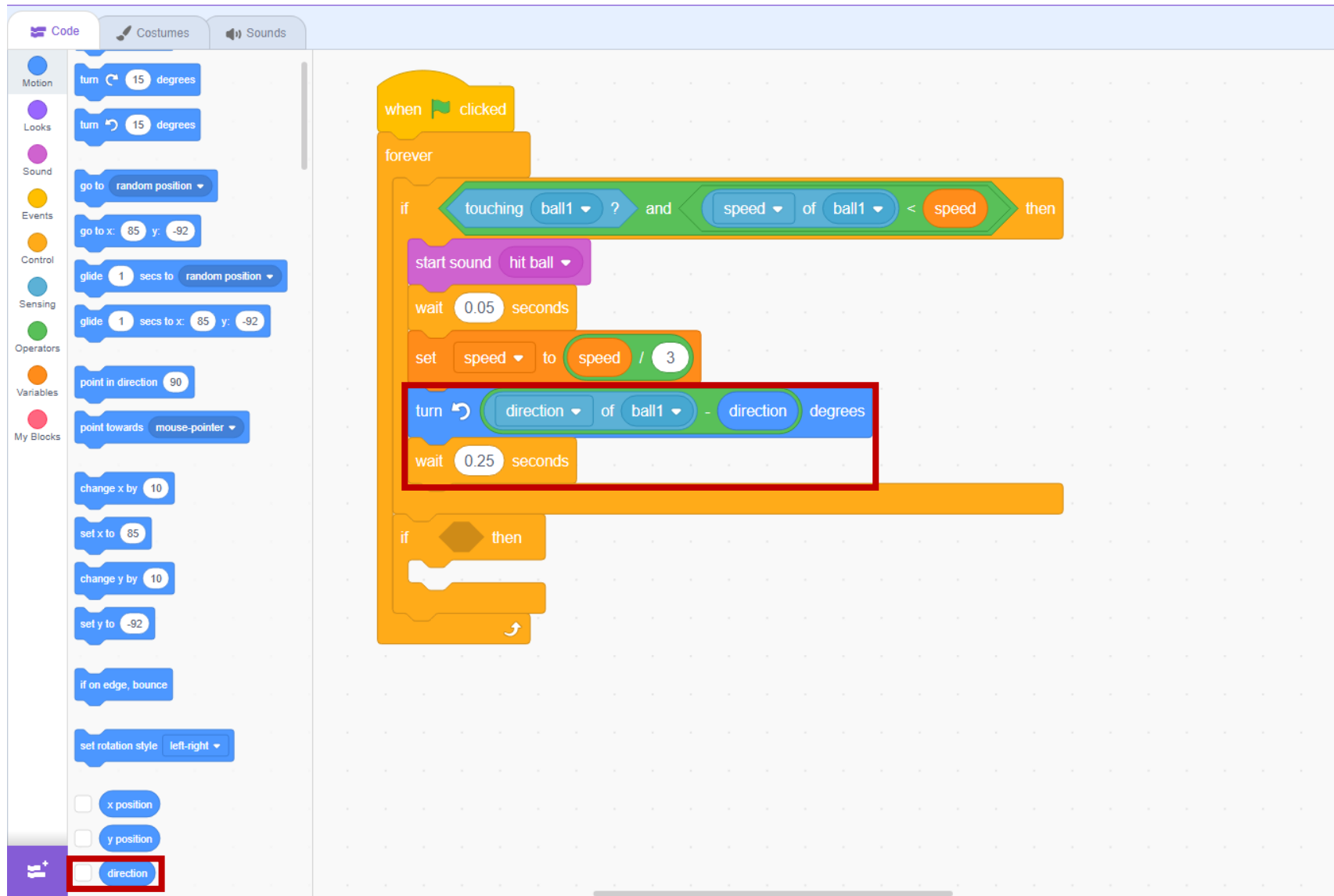
Change the wait time from 1 to 0.05 seconds. Then go to “Operators” and grab the “__ / __” block and put it into the “set speed to __” block. Then go to “Variables” to grab the speed variable and put it into the first blank of the “__ / __” block. Then type 3 for the last blank. It should look like the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



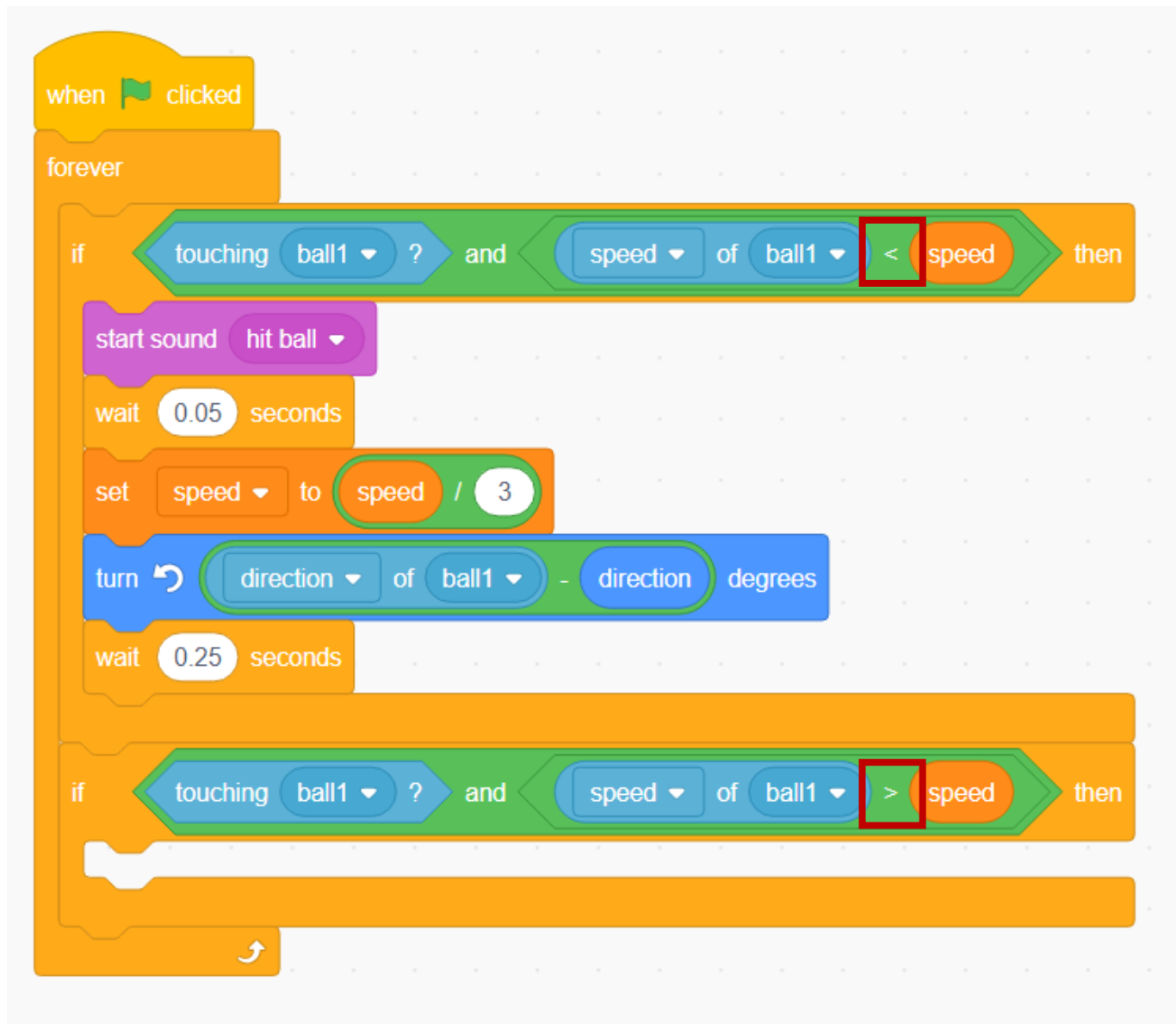
Go to “Motion” and grab the “turn counterclockwise 15 degrees” block and put it into the “if-then” statement. Then go to “Operators” and grab the “_ - _” block and put it into the “turn counterclockwise 15 degrees” block.



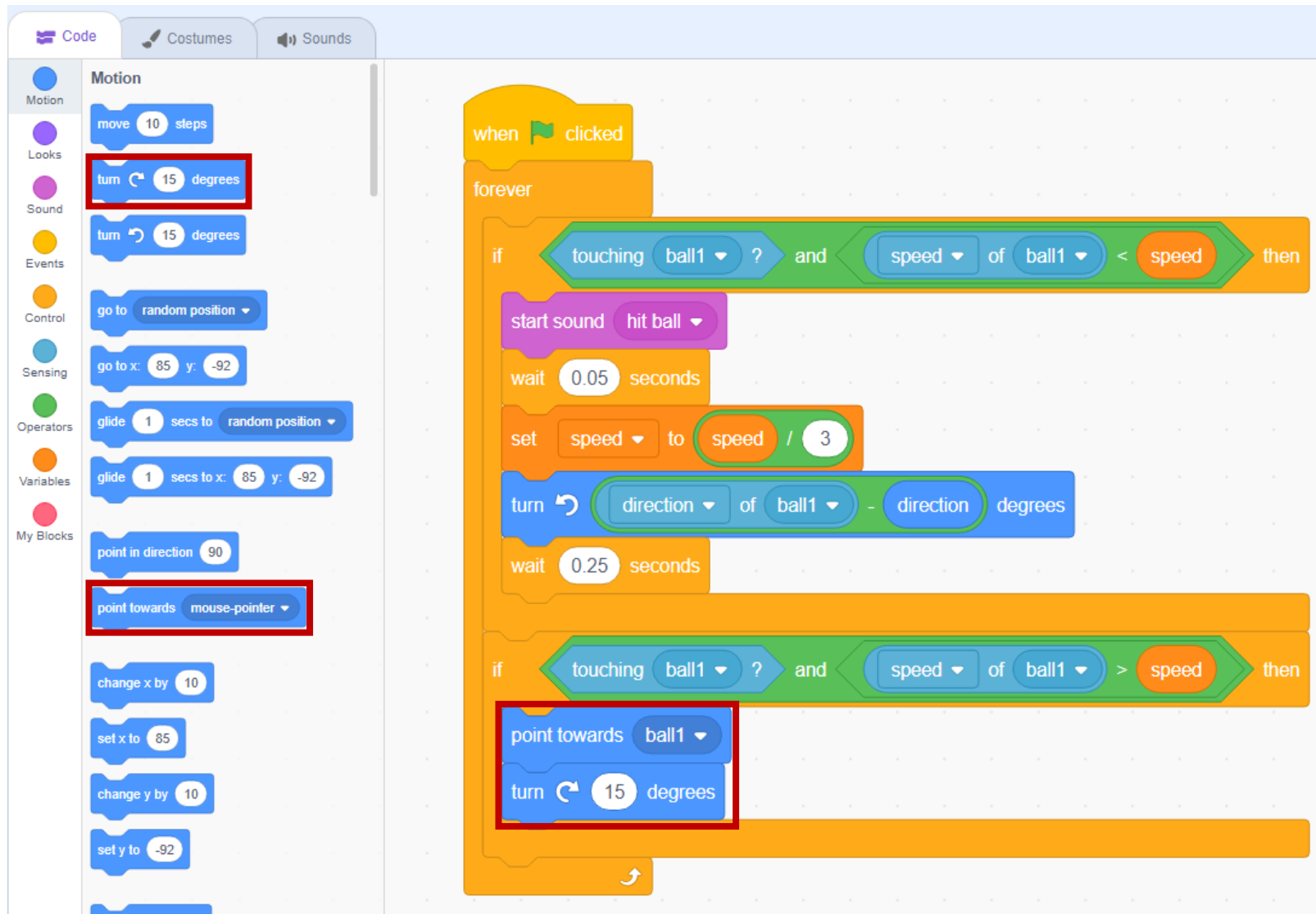
Go to “Sensing” and grab the “backdrop # of Stage” and add it to the first blank. Change variables so it says, “direction of ball 1”. Then go to “Motion” and grab the “direction” block and attach it to the second blank. Then go to control add grab the “wait 1 seconds” block and change it to 0.25.



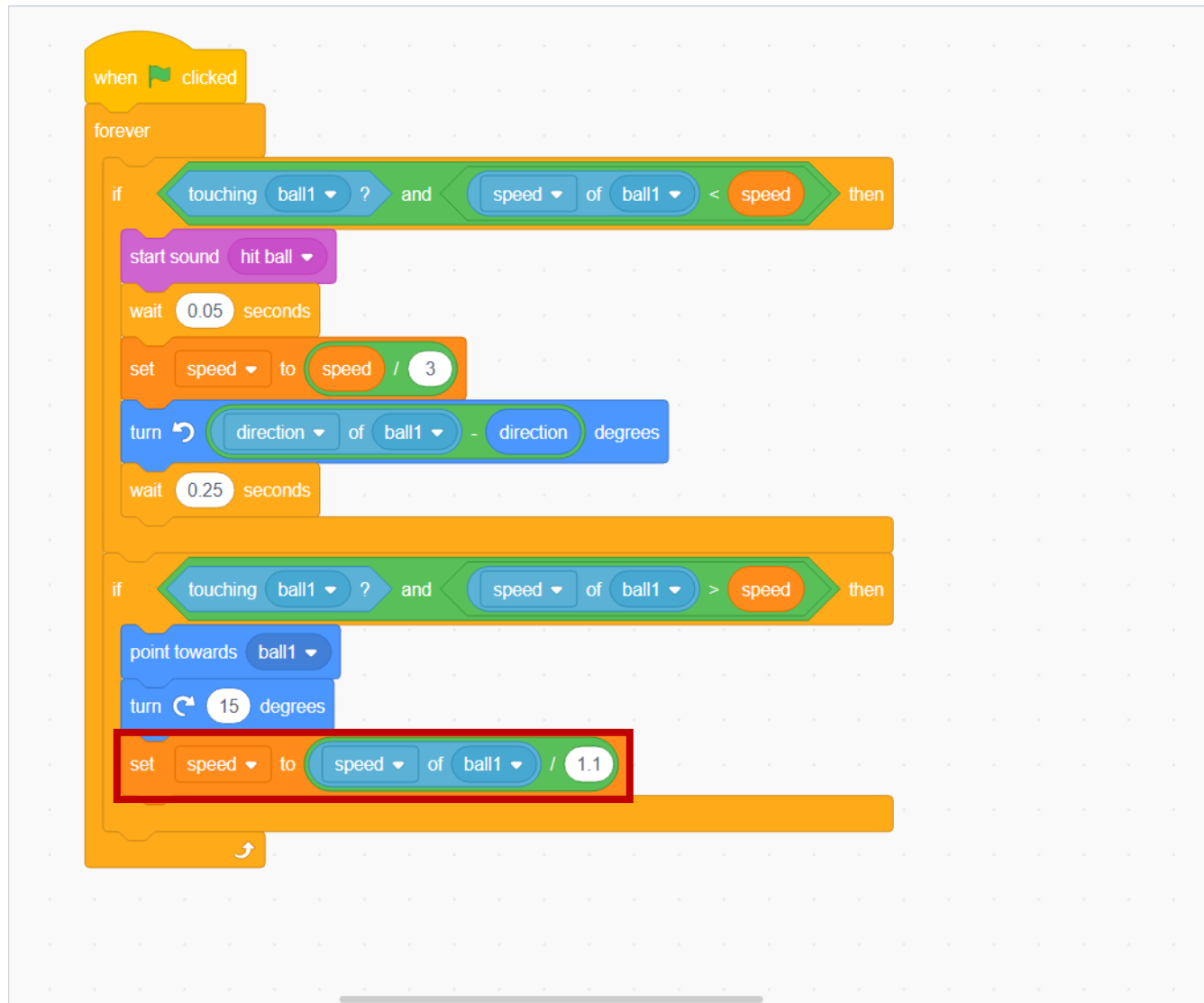
The second “if-then” block is very similar to the first. Insert the same blocks in the “if-then” blank, but grab “>” instead of the “<”.



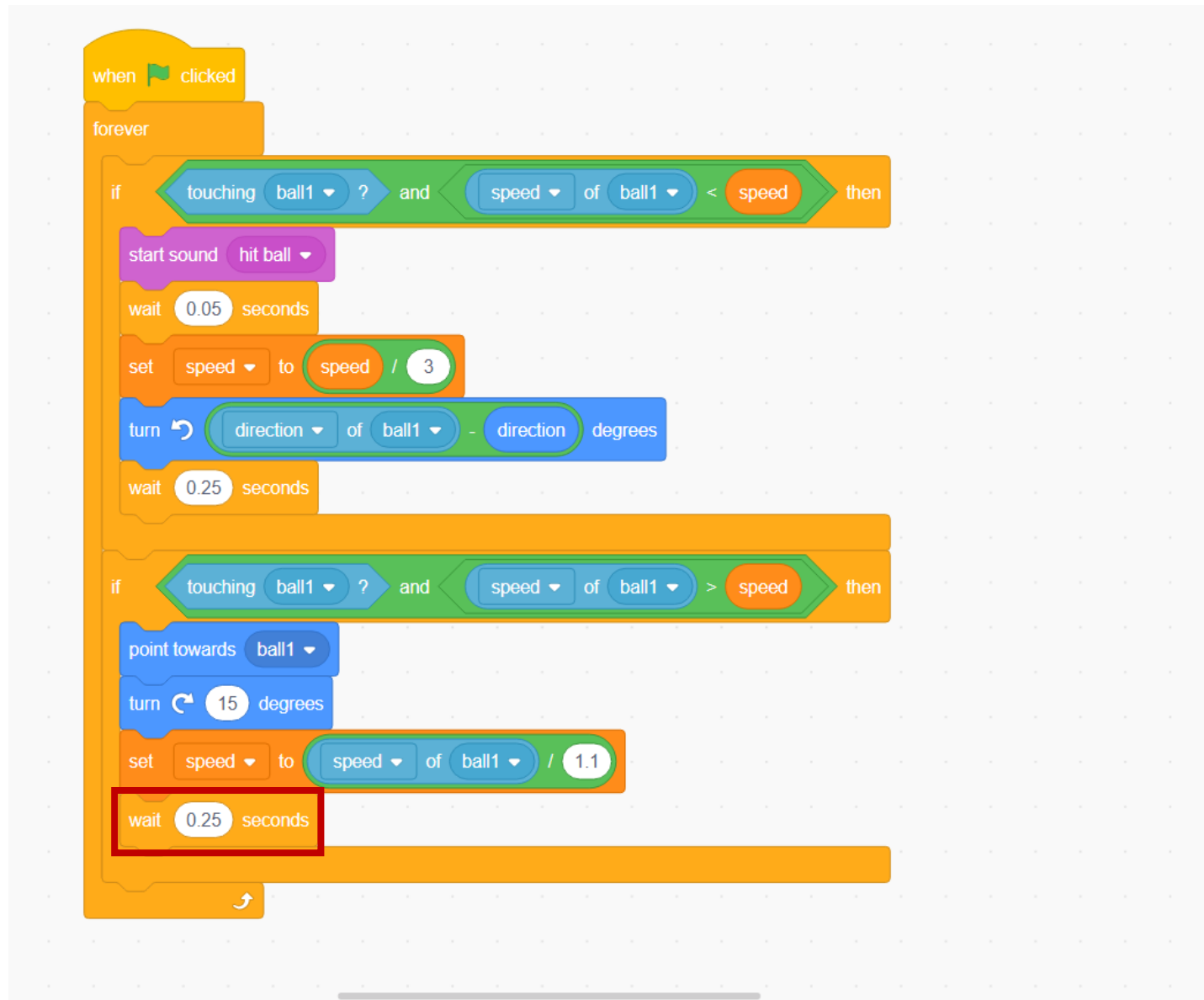
Go to the “Motion” tab and grab the “point towards mouse-pointer” block and change it “point towards ball 1”. Then add a “turn clockwise 15 degrees” block. Keep the same degree.



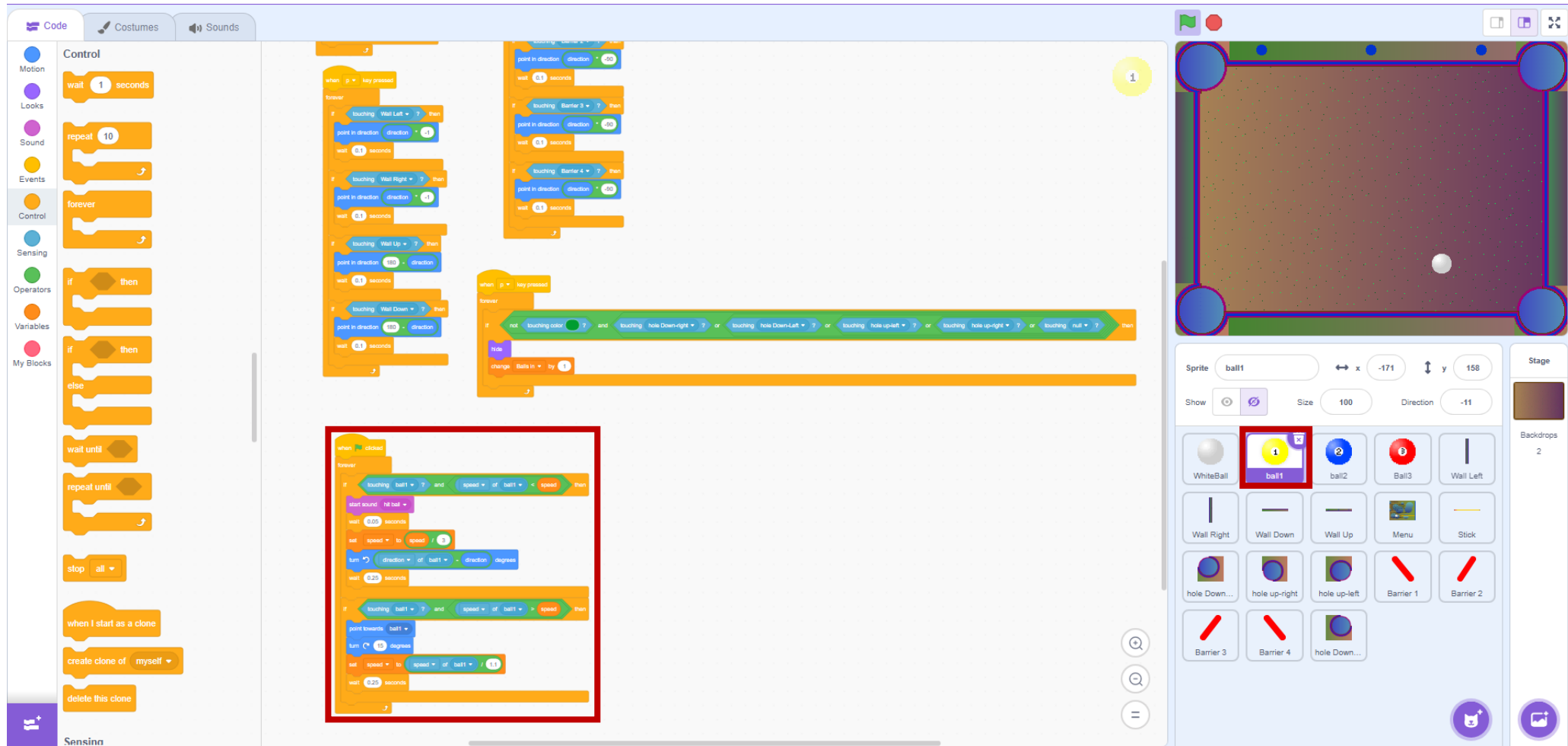
Go to the “Variables” tab and grab the “set balls in to 0” block and change the variable to “speed”. Then go to “Operators” and grab the “_/_”. Then go to “Sensing” and grab the “backdrop # of Stage” and insert it into the first blank. Change it to “speed of ball 1”. Then type 1.1 in the second blank.



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

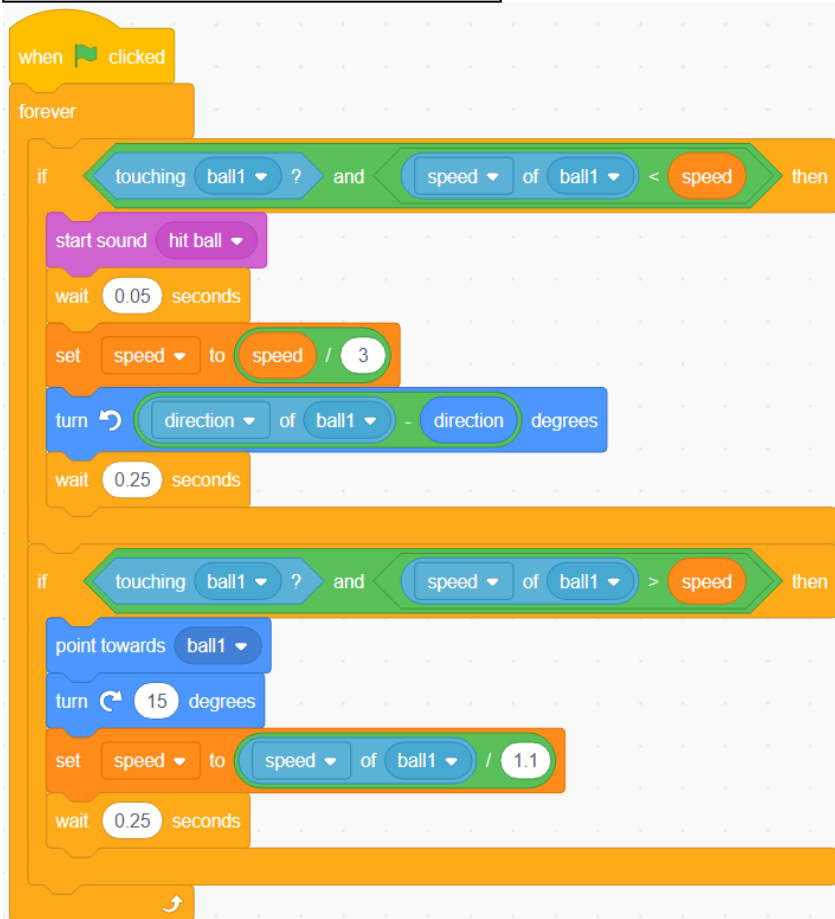


Click on the “when flag clicked” block in the code and then use ctrl+c to copy the code block. Now select the “ball 1” sprite. Use ctrl+v to paste the code onto the canvas. You may need to move around the canvas to find the code block.

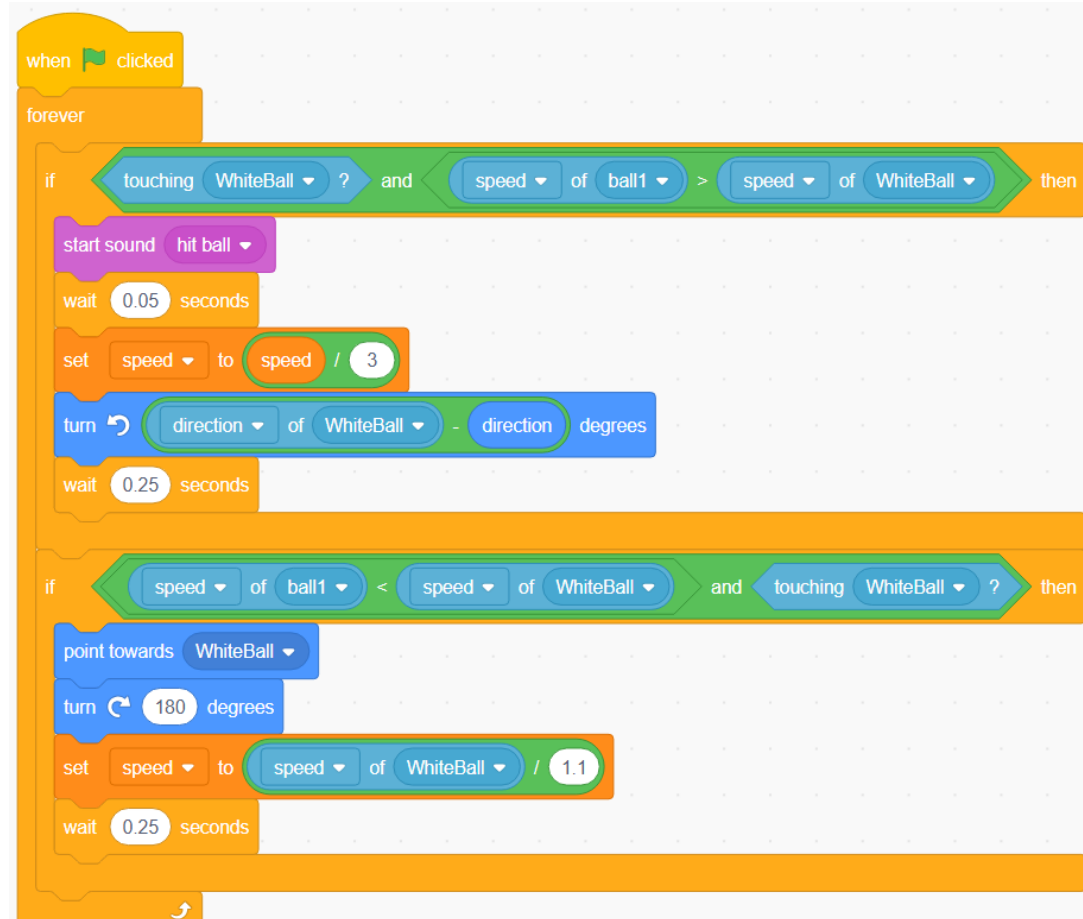


Now we have to change all instances of “ball 1” to “white ball”. Make sure that the block is still checking for speed not direction. We also have to change the “turn clockwise 15 degrees” block to “turn clockwise 180 degrees”. Use the two pictures below to compare the different code blocks.

Code block in “white ball” sprite

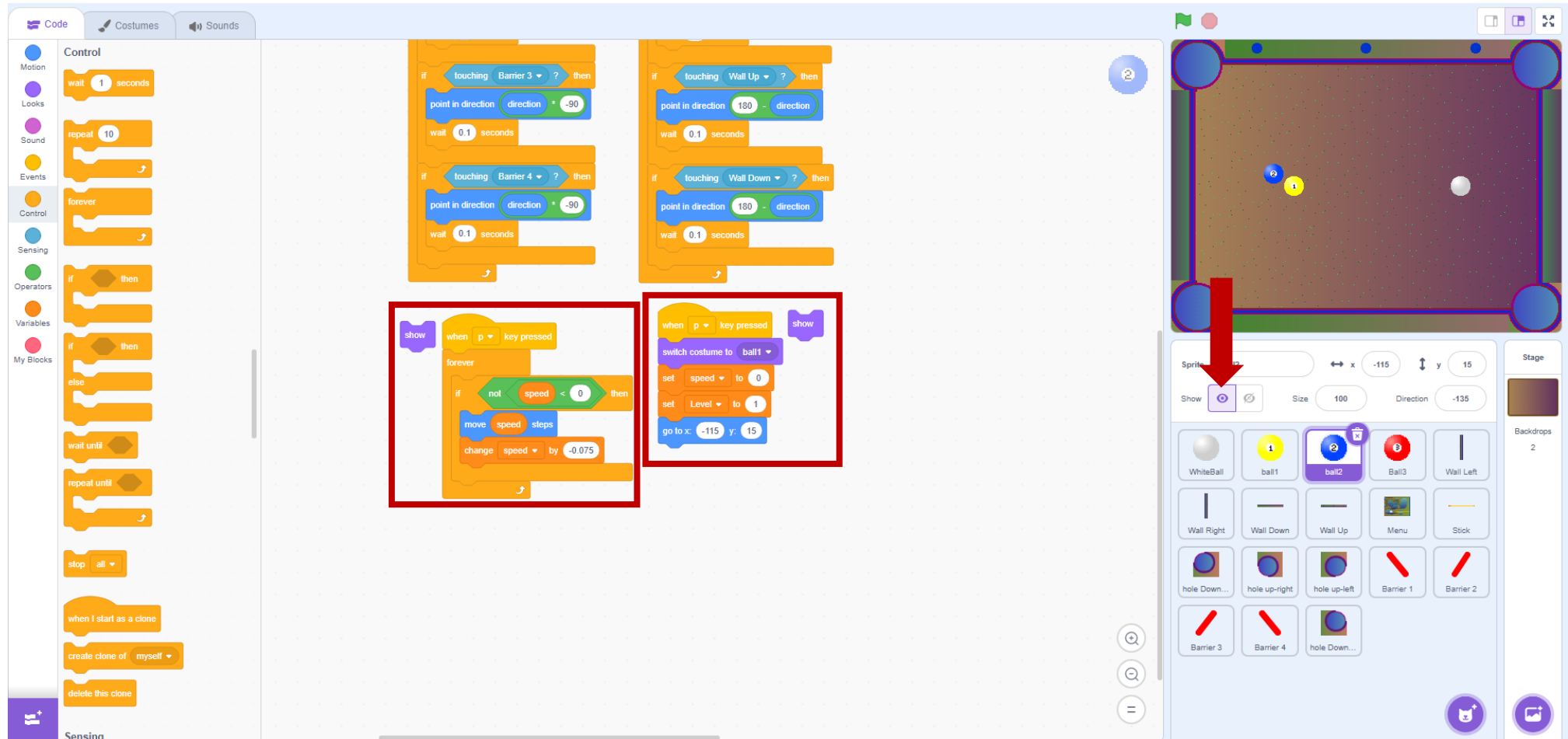


Code block in “ball 1” sprite

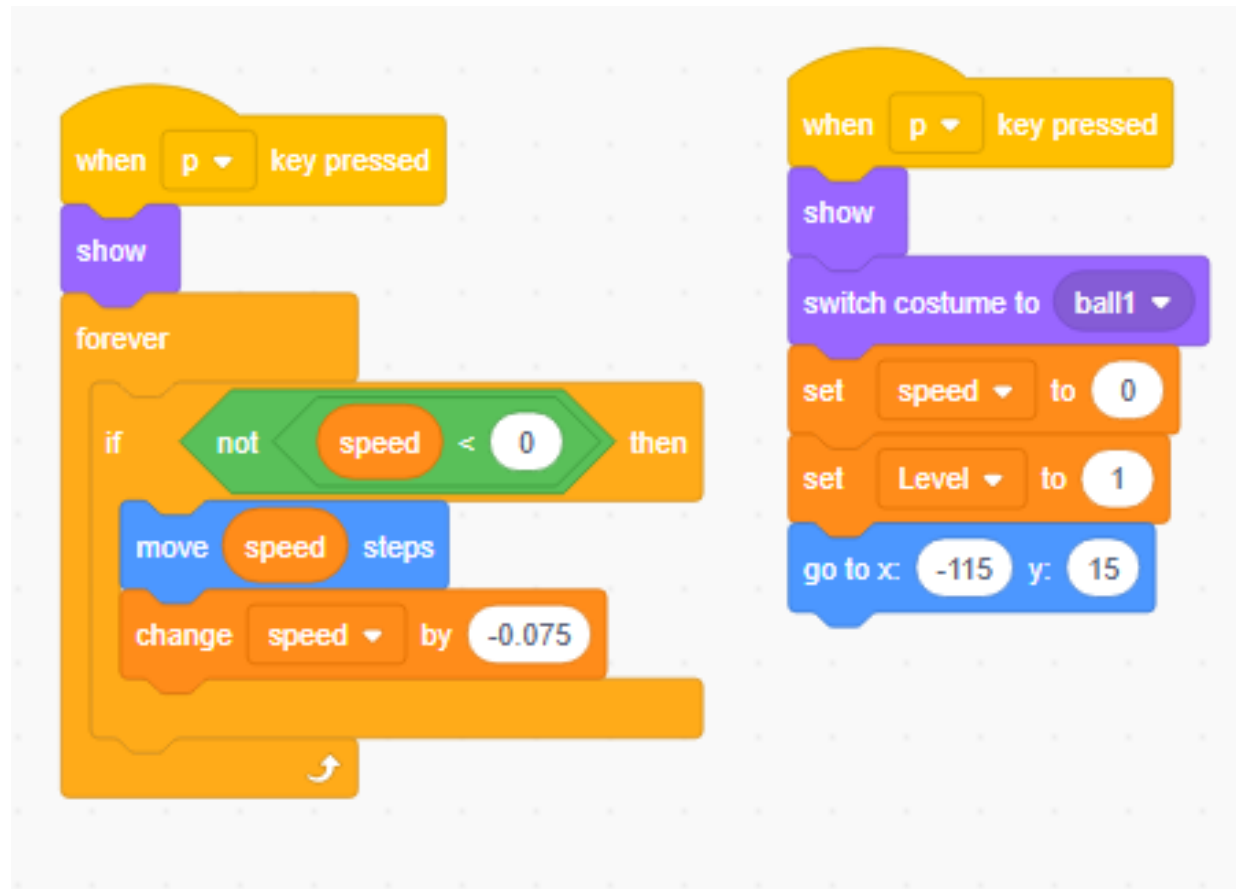


This section will cover level 2 of the game

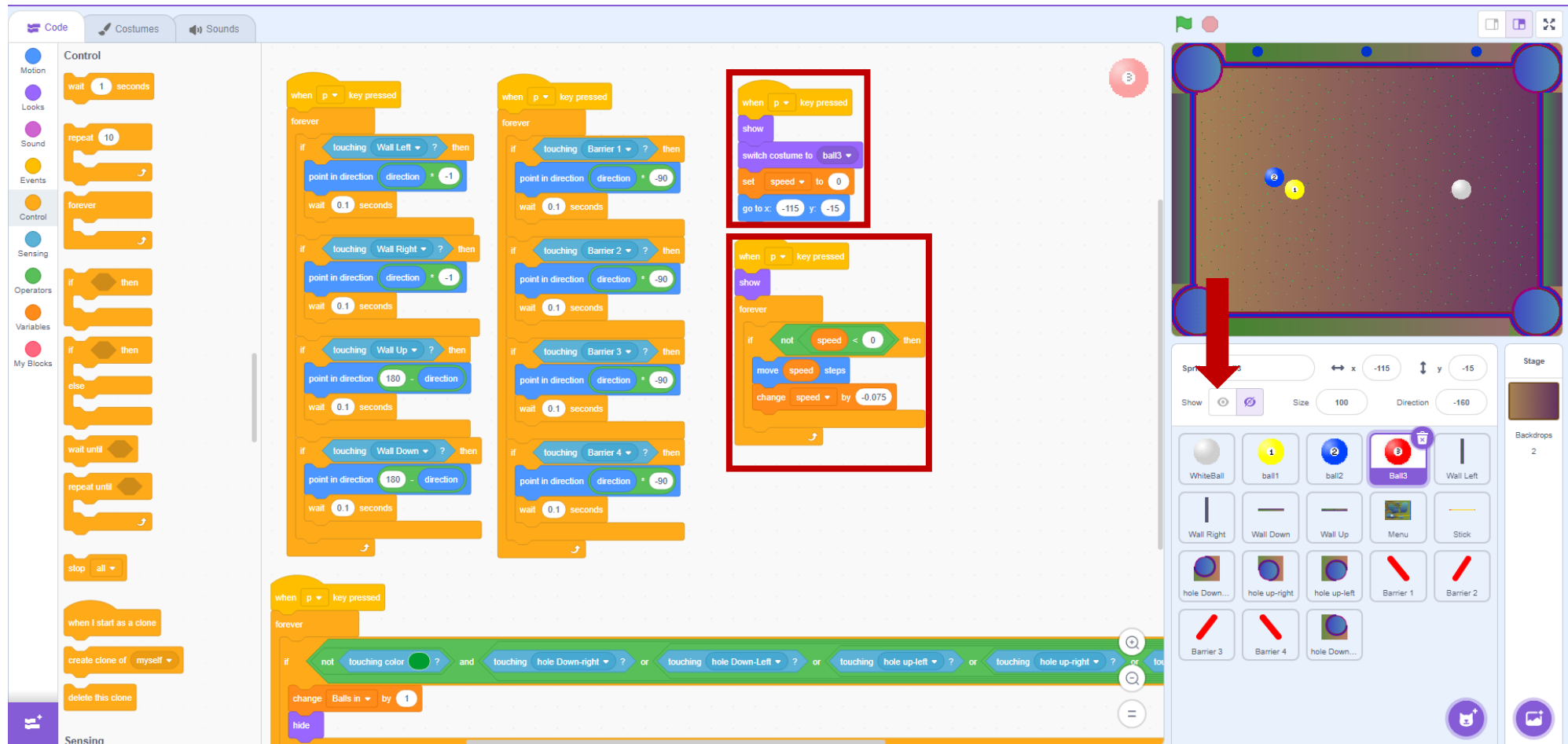
First go to the “ball 2” sprite and click on the eyeball to make the ball visible. Then attach the two “show” block to the adjacent code blocks they are next to.



The image shows the Scratch code editor and stage. The code editor has three tabs: Code, Costumes, and Sounds. The Code tab is active, showing a script area with several code blocks. The stage shows a game environment with a brown floor, green walls, and a white ball. A red arrow points to the 'ball2' sprite in the Sprites panel, which is currently hidden (indicated by a crossed-out eyeball icon). The 'ball2' sprite is selected, and its 'Show' button is highlighted. The code blocks are organized into two main sections, each enclosed in a red box. The first section (left) contains a 'when green flag clicked' event block, followed by a 'show' block, a 'when key pressed' block, a 'forever' loop containing a 'not speed < 0' condition, a 'move speed steps' block, and a 'change speed by -0.075' block. The second section (right) contains a 'when key pressed' block, a 'show' block, a 'switch costume to ball1' block, a 'set speed to 0' block, a 'set Level to 1' block, and a 'go to x: -115 y: 15' block. The stage shows a game environment with a brown floor, green walls, and a white ball. A red arrow points to the 'ball2' sprite in the Sprites panel, which is currently hidden (indicated by a crossed-out eyeball icon). The 'ball2' sprite is selected, and its 'Show' button is highlighted.



Repeat this process for ball 3.



The image displays the Scratch code editor with three event-driven scripts for ball movement, each triggered by a key press. The scripts are highlighted with red boxes in the original image.

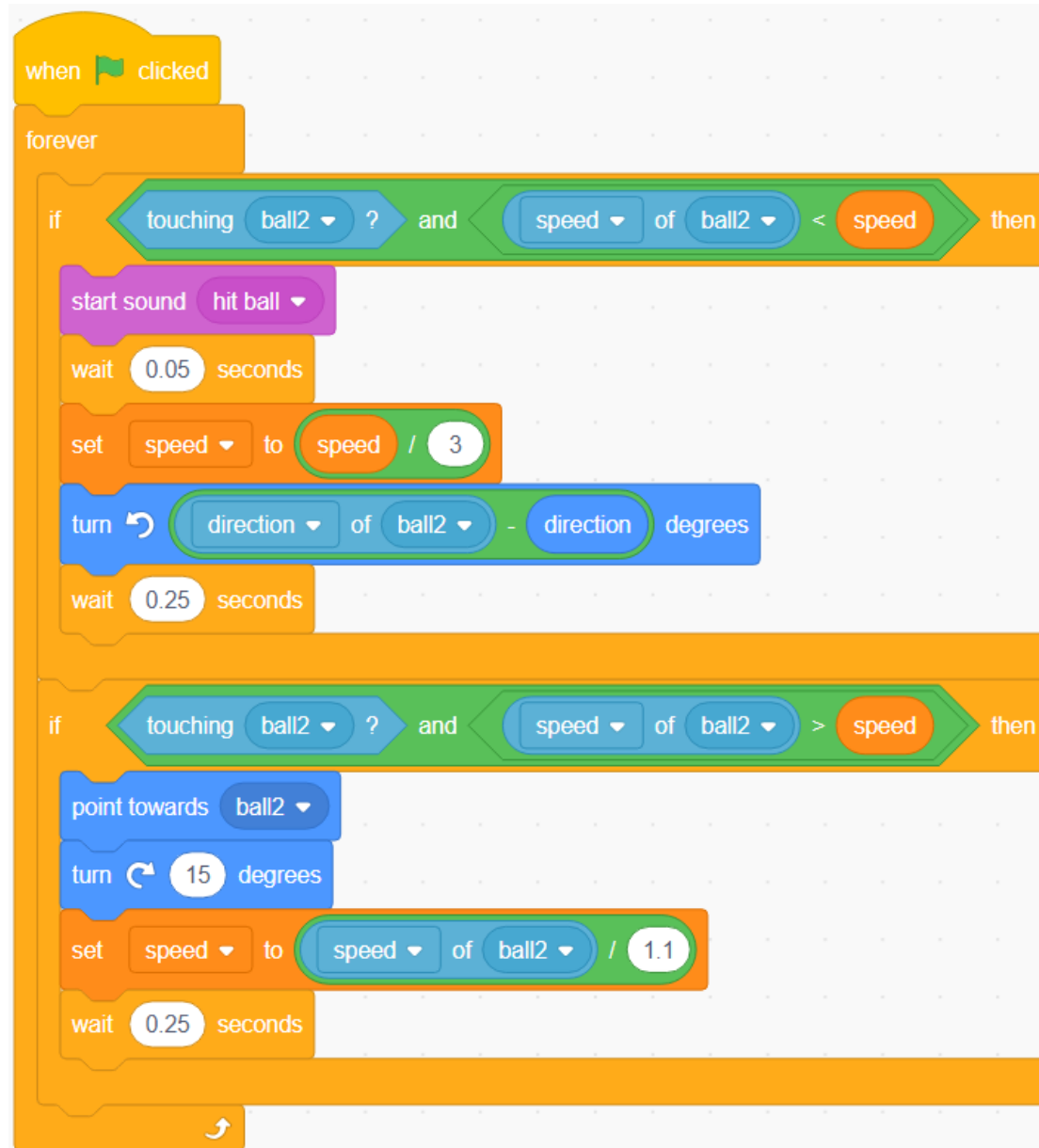
Script 1 (Top Left): When key pressed, show ball3, switch costume to ball3, set speed to 0, and go to x: -115 y: -15.

Script 2 (Middle Left): When key pressed, show ball3, and then enter a forever loop. Inside the loop, if speed is not less than 0, move speed steps and change speed by -0.075.

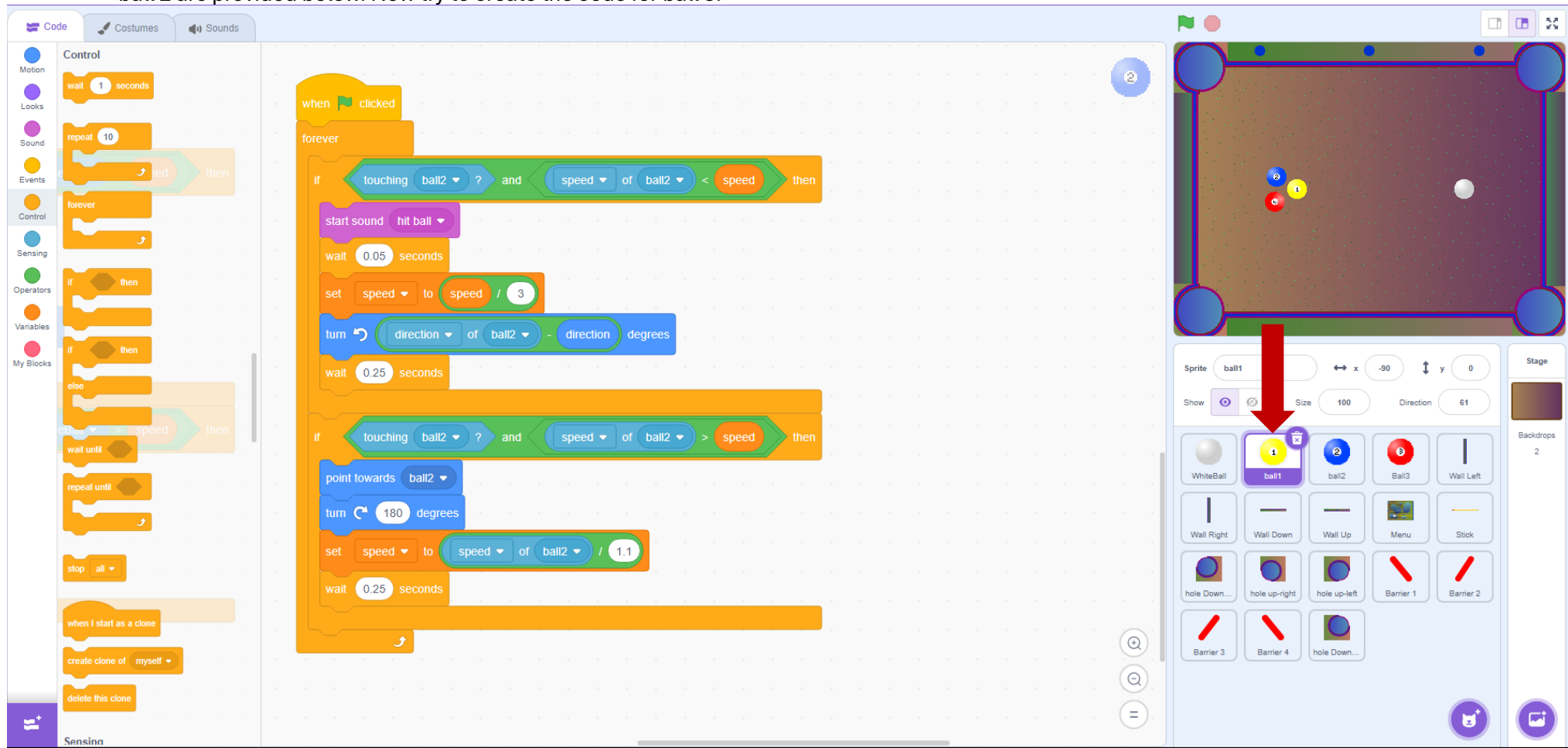
Script 3 (Bottom Left): When key pressed, show ball3, and then enter a forever loop. Inside the loop, if speed is not less than 0, move speed steps and change speed by -0.075.

The stage preview on the right shows a ball3 sprite on a stage with a grid. A red arrow points to the ball3 sprite in the stage preview.

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”.



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.



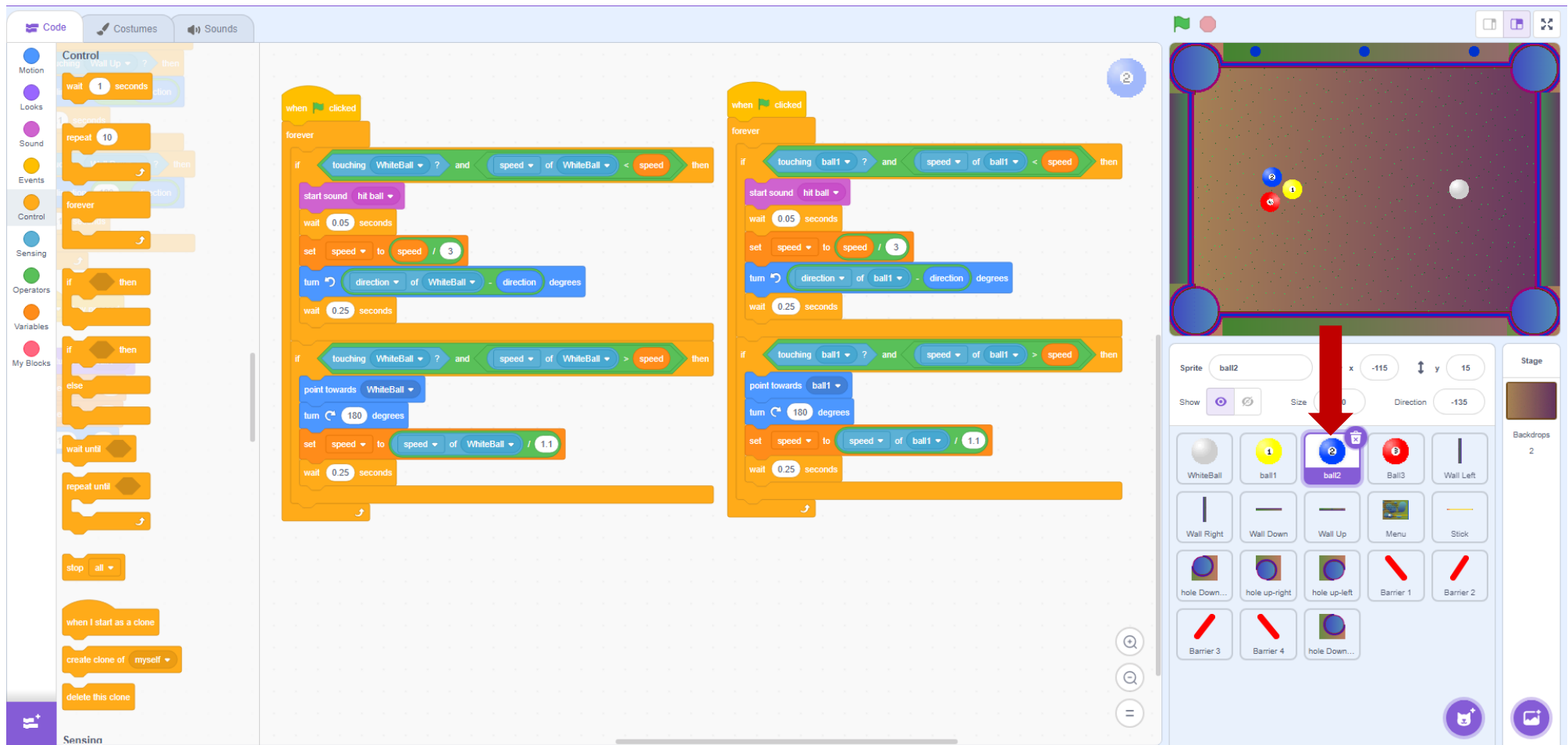
The image displays the Scratch code editor with a script for a ball collision and a stage view showing a pool table with three balls (ball1, ball2, ball3) and a white ball. A red arrow points to the 'ball1' sprite in the sprite list.

Code Editor:

- 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 180 degrees**
 - set speed to speed of ball2 / 1.1**
 - wait 0.25 seconds**

Stage View:

- Sprites:** ball1 (selected), ball2, ball3, WhiteBall, Wall Left, Wall Right, Wall Down, Wall Up, Menu, Stick, hole Down..., hole up-right, hole up-left, Barrier 1, Barrier 2, Barrier 3, Barrier 4, hole Down...
- Stage:** 2



The image displays the Scratch code editor with two scripts for ball movement and collision detection. The left script is for a 'WhiteBall' and the right script is for a 'ball1'.

Left Script (WhiteBall):

- when clicked
- forever loop:
 - 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

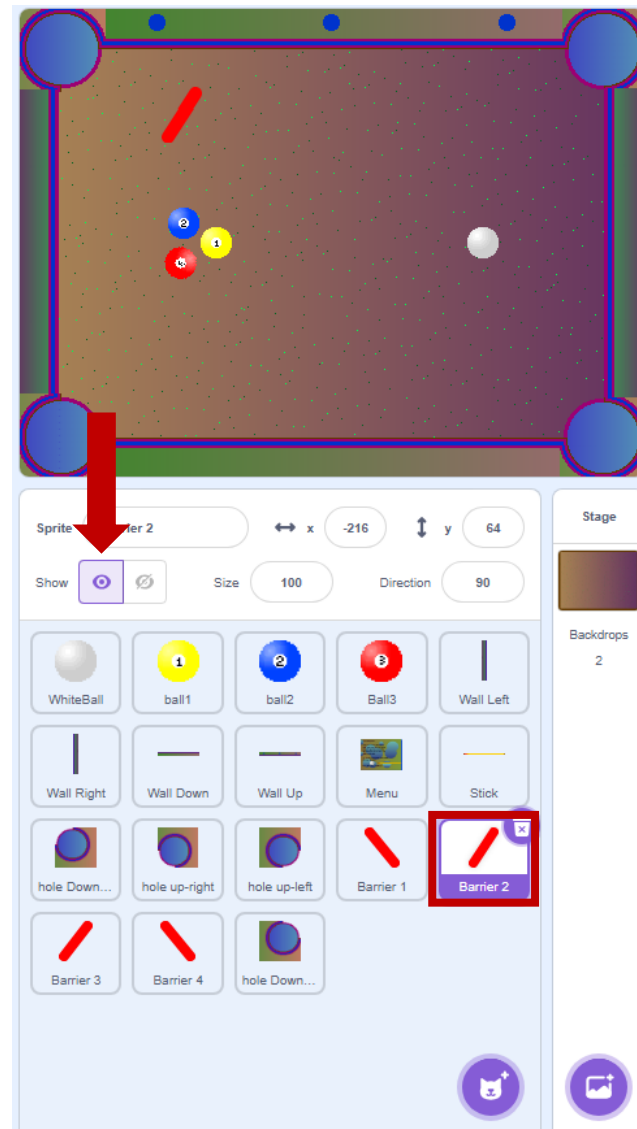
Right Script (ball1):

- when clicked
- forever loop:
 - 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

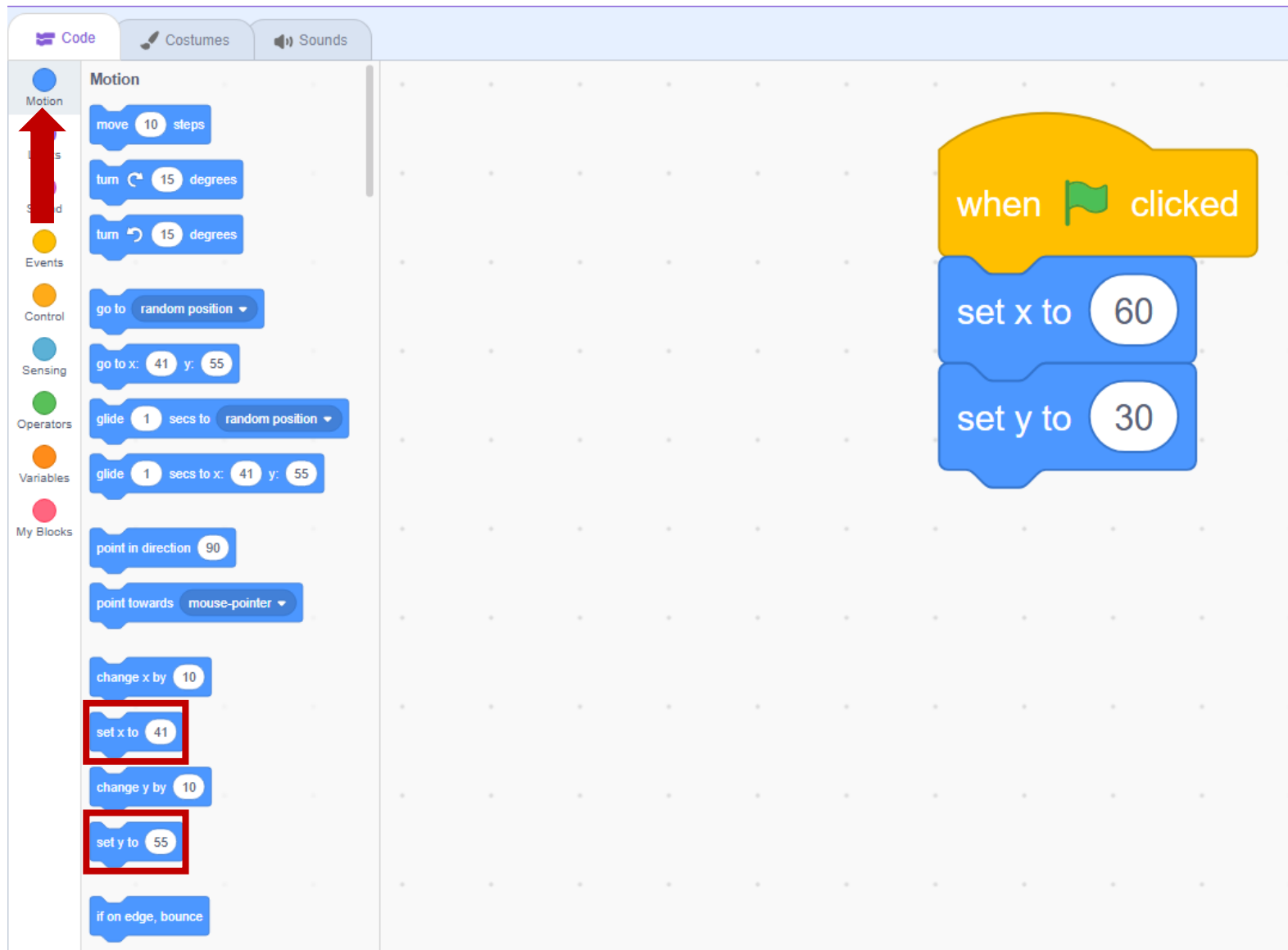
The right panel shows the stage with a ball2 sprite selected. A red arrow points to the 'ball2' sprite in the sprite list.

This section will cover level 3 of the game

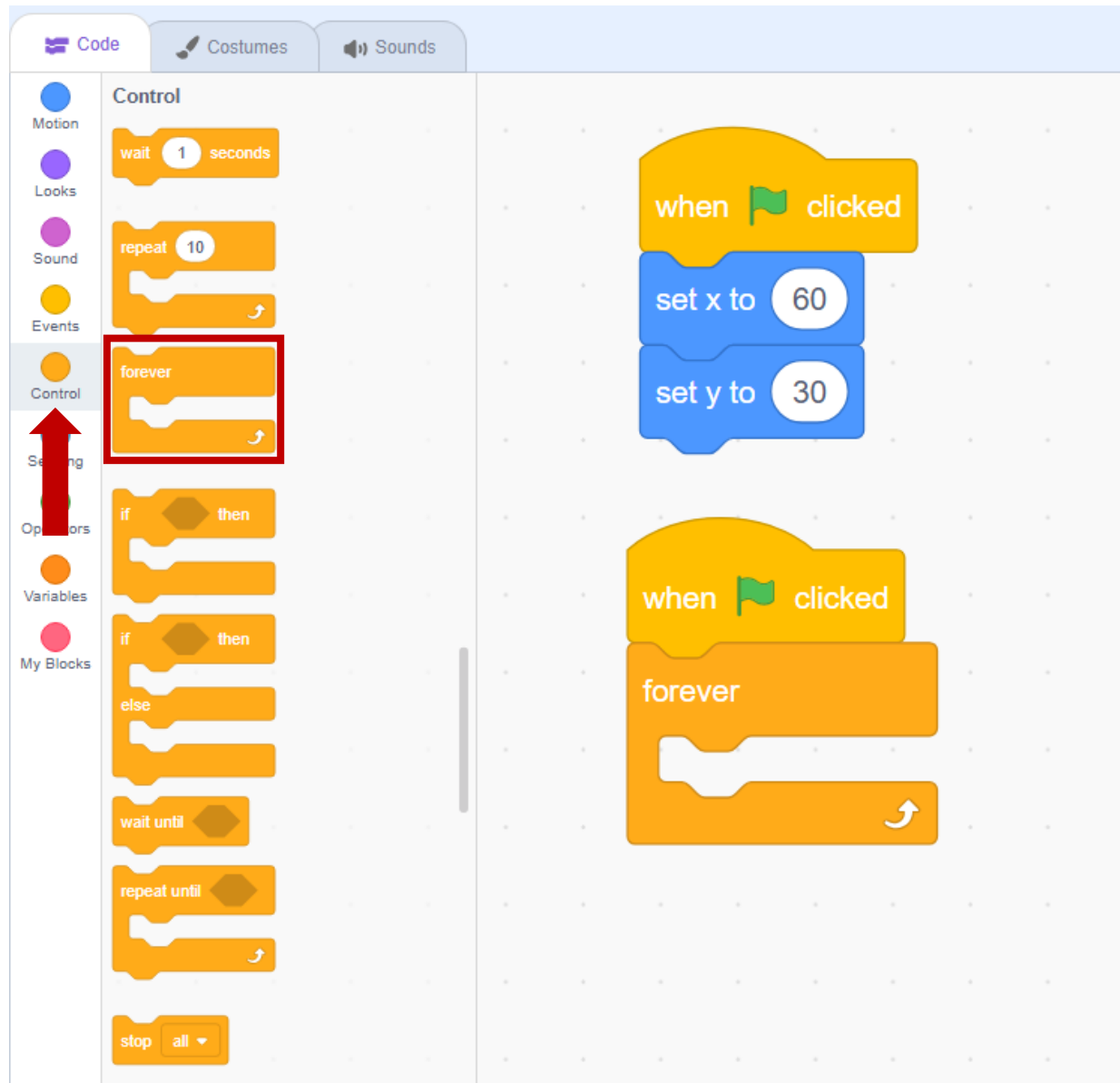
Go to each of the barriers and click on the eyeball to make them visible.



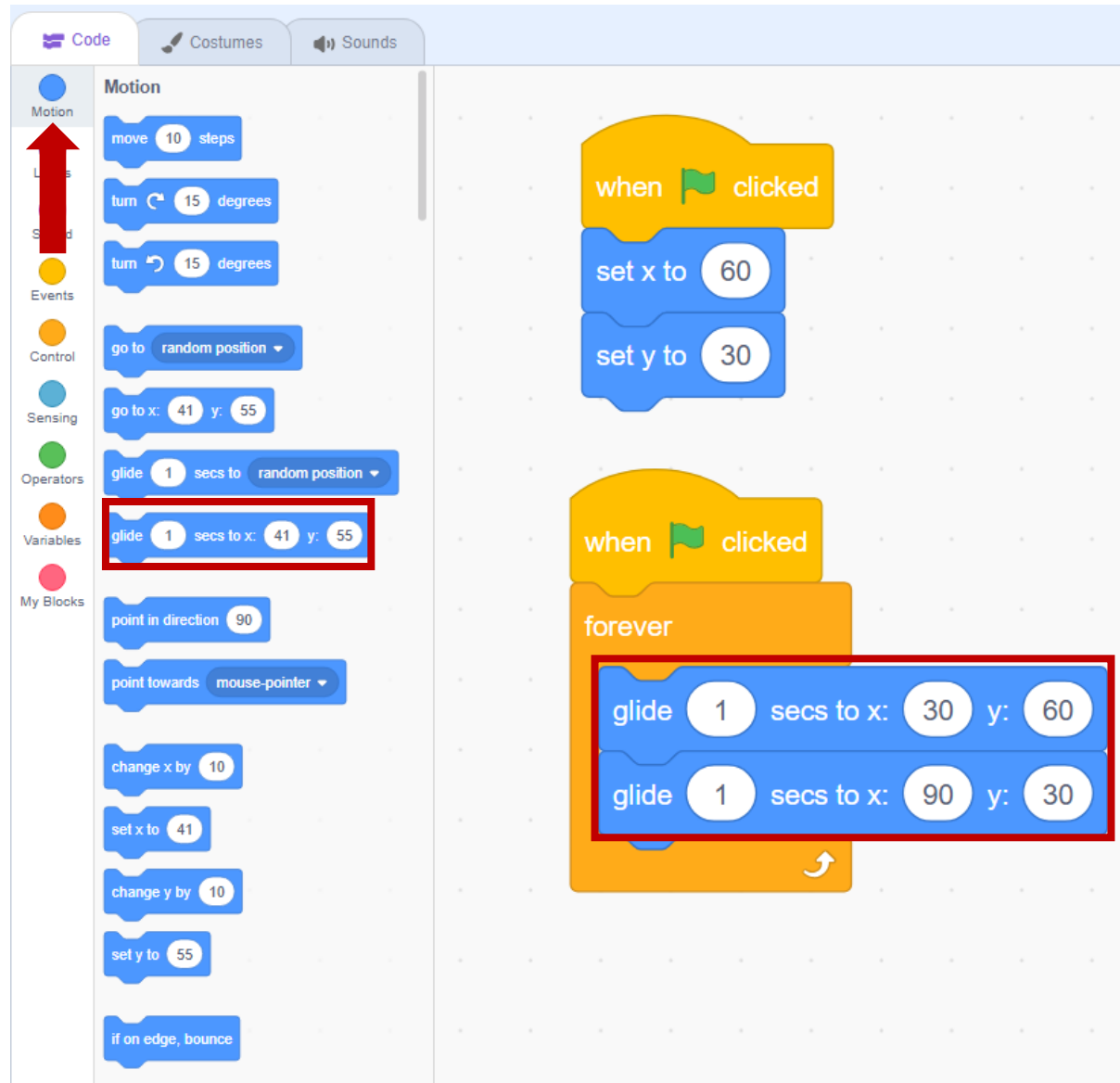
Go to the “Events” tab and select the “when flag pressed” block. Then go to the “Motion” tab and grab a “set x to ___” and a “set y to ___” block. Change the x to 60 and the y to 30.



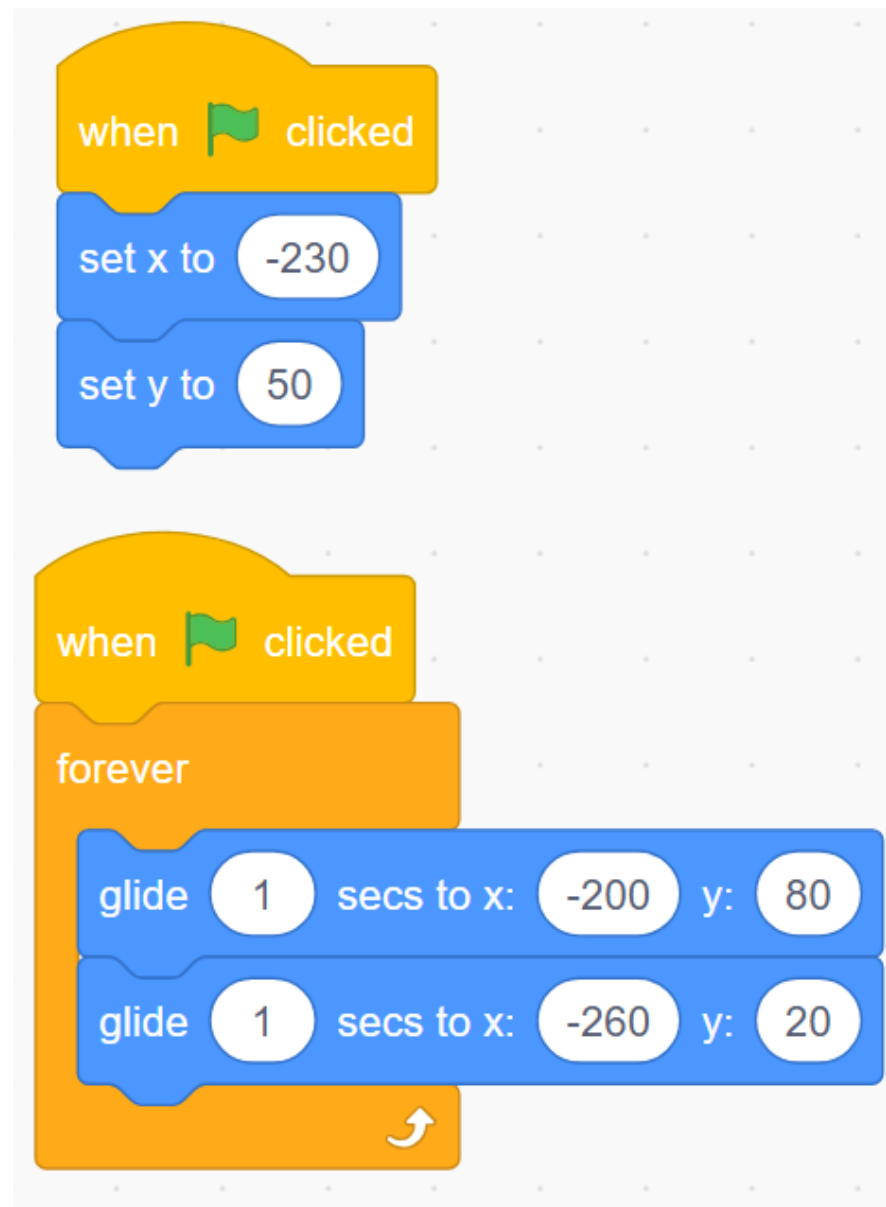
Go back to the “Events” tab and grab the “when flag pressed” block. Then go to the “Control” tab and grab a forever loop.



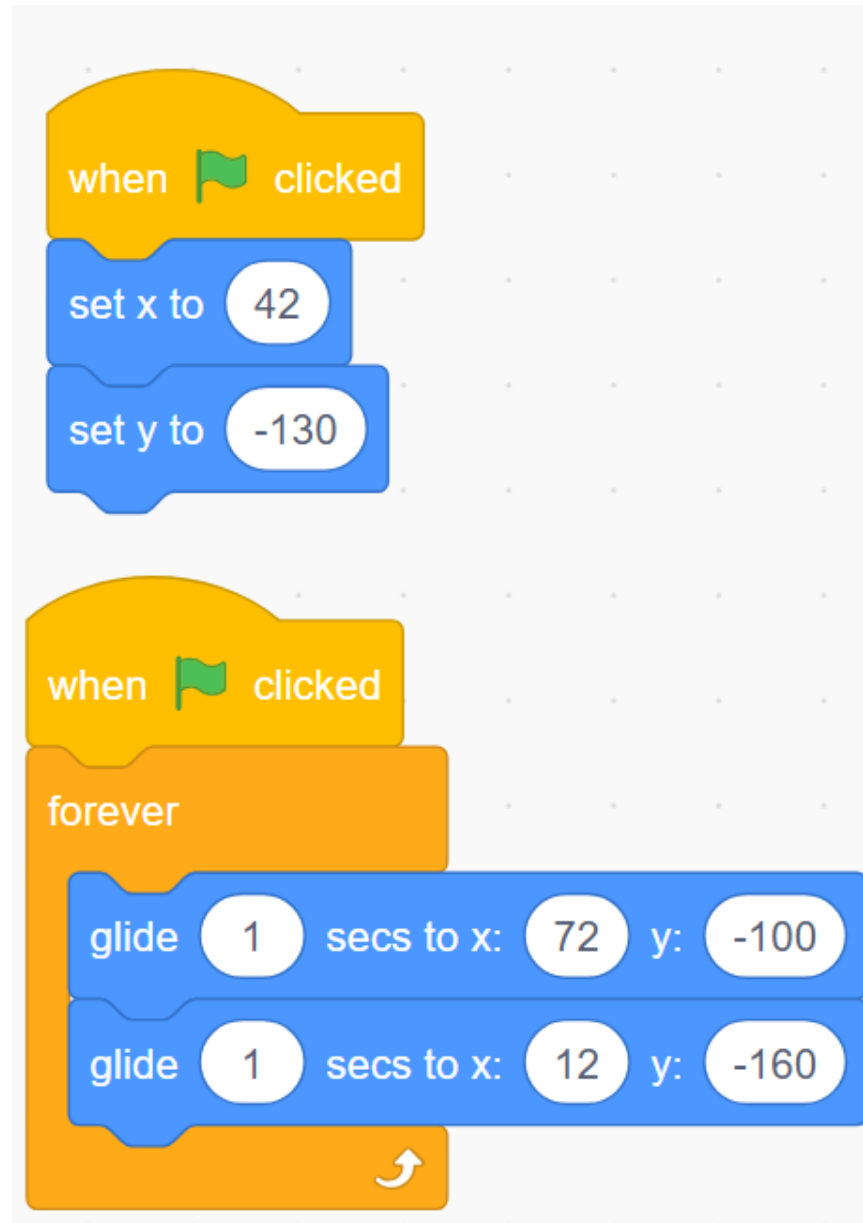
Go to the “Motion” tab and grab two “glide __ secs to x: __ y: __” and insert them both inside the forever loop. Set the glide to 1 sec for both. Set the x to 30 and y to 60 for one of the glide blocks. Then set the x to 90 and y to 30 for the other.



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

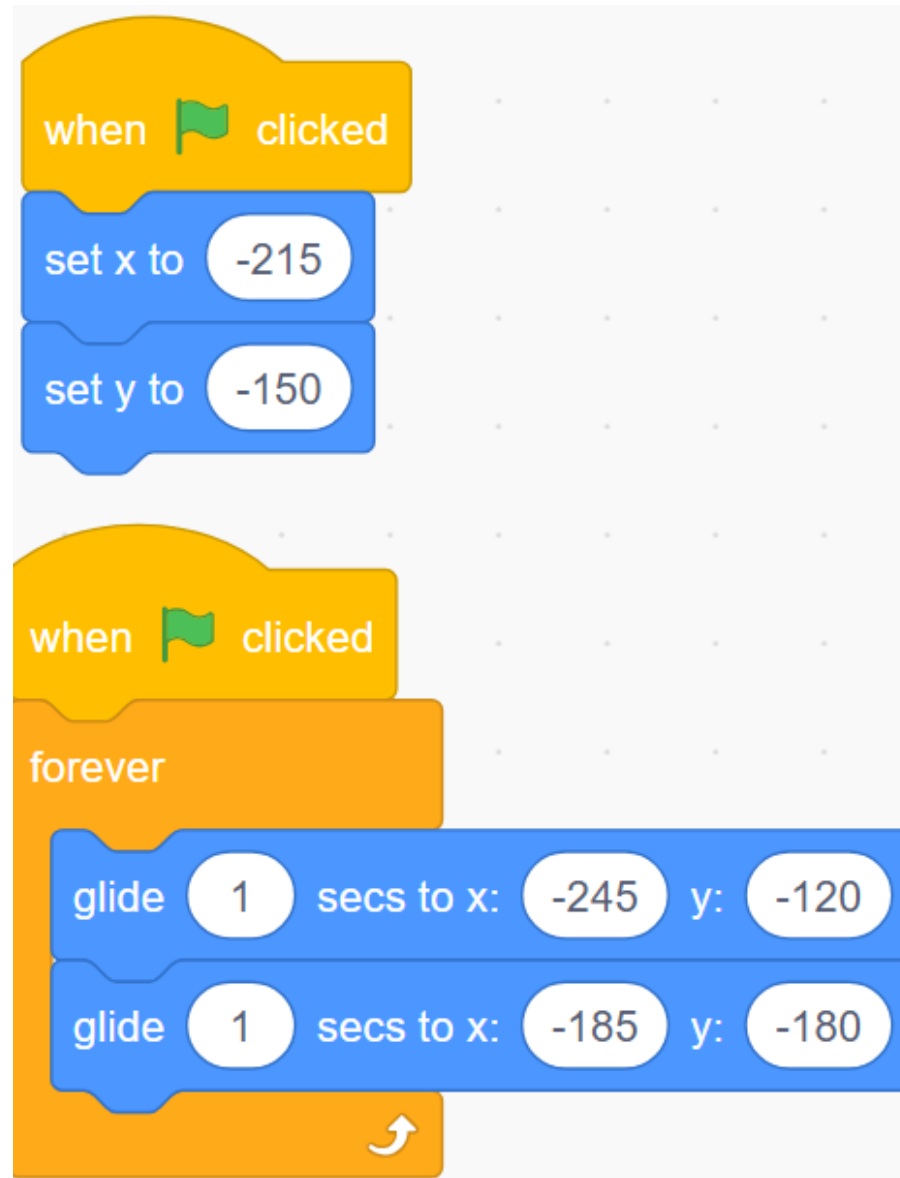


Below is the code for barrier 3.



<https://tapggc.org/>

Below is the code for barrier 4.



<https://tapggc.org/>