



PICK UP YOUR RUBBISH!!!

SCRATCH INSTRUCTIONS

AN INTRODUCTION
TO VALUE BASED
GAME DESIGN

YEARS 7 - 10
GROUPS OF 1 - 2



OMGTECH.CO.NZ

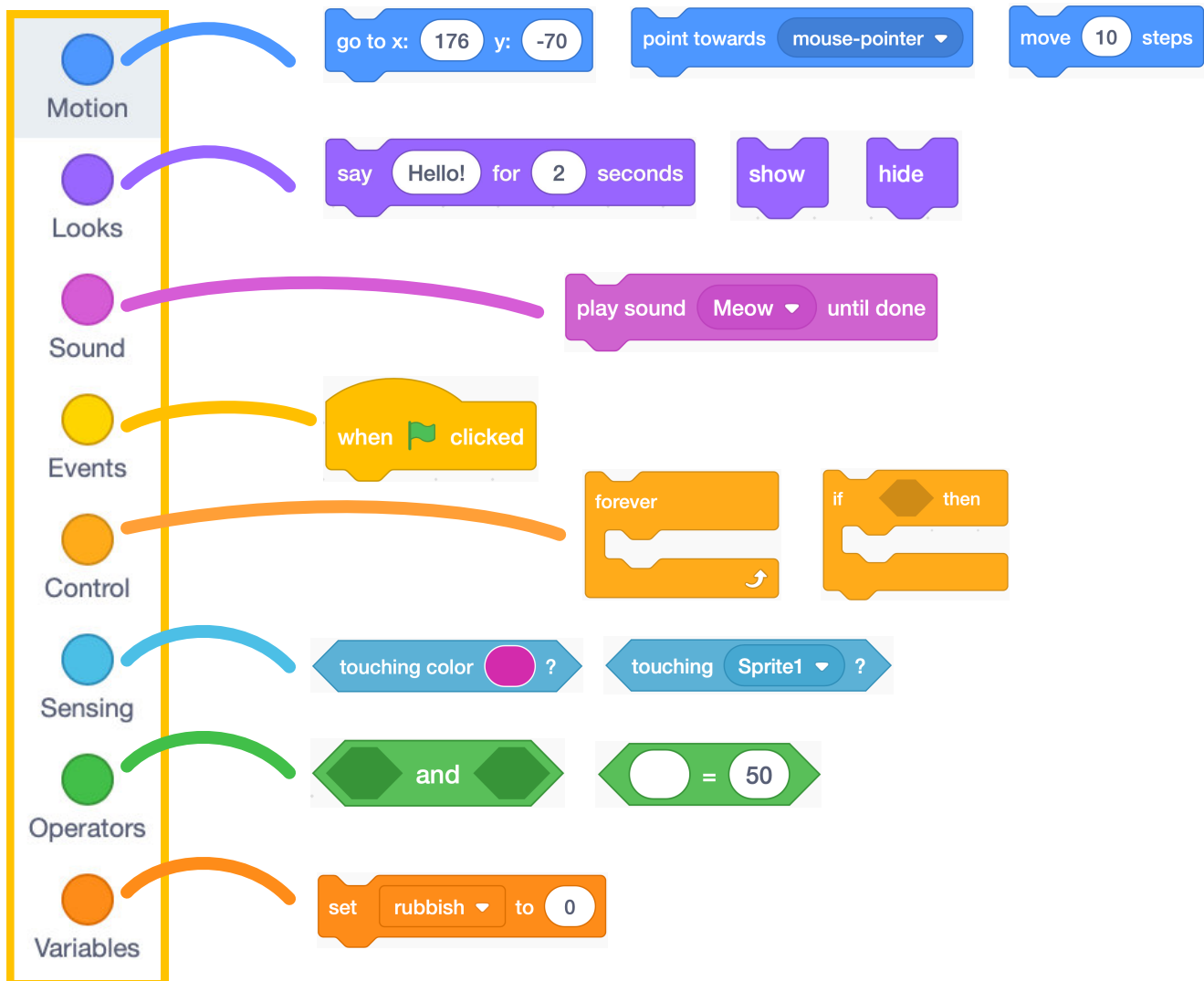
"Pick up your Rubbish!!!"

In "Pick up your rubbish we explore the concept of value based game design. We are going to apply our traditional values of guardianship and caretaker of the land into our game design. In this game we challenge players to collect all the rubbish that has been carelessly tossed onto our beautiful whenua.

Find our example here:

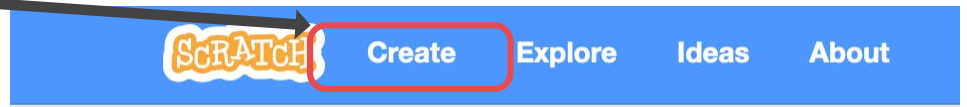
<https://scratch.mit.edu/projects/305916116/>

Blocks you need for this project

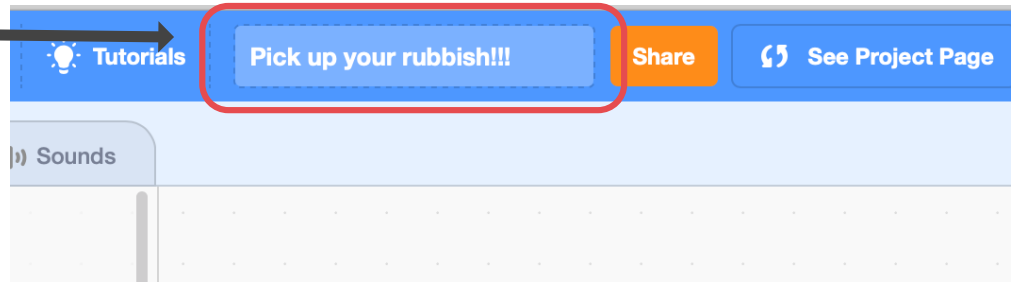


Setting up our project

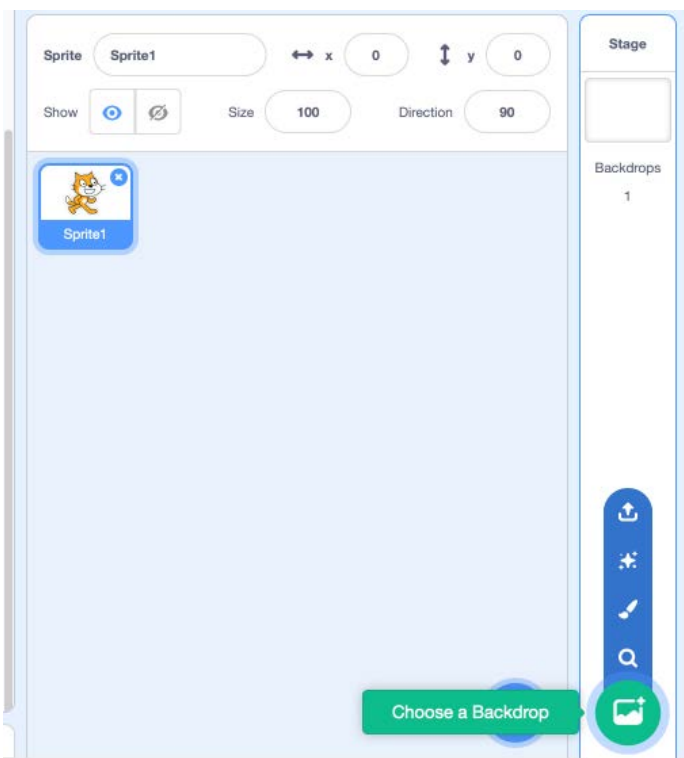
1. First create a new project in Scratch



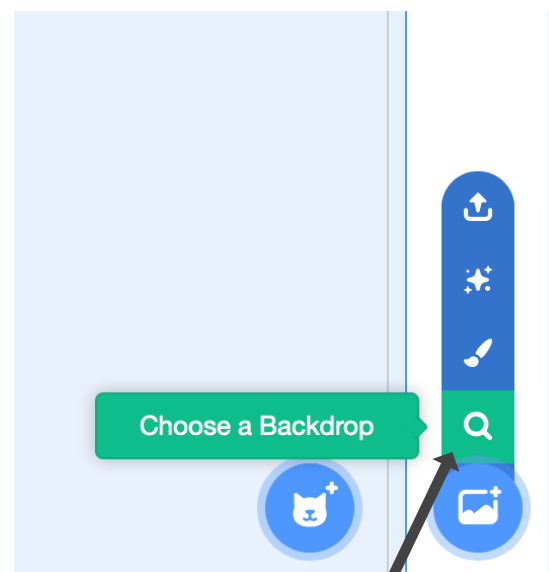
2. Name your project. We are going to call ours "Pick up your rubbish!!!"



3. Now we are going to add a backdrop to make our level more interesting. Click the Backdrop button at the bottom right of the screen then select the choose a Backdrop button.

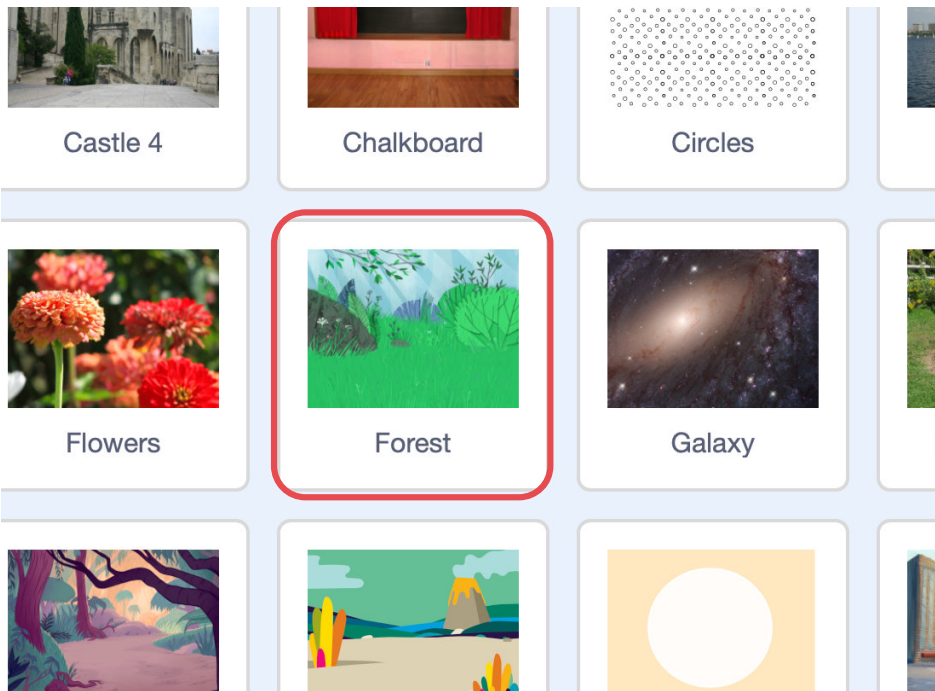


"Backdrop button"



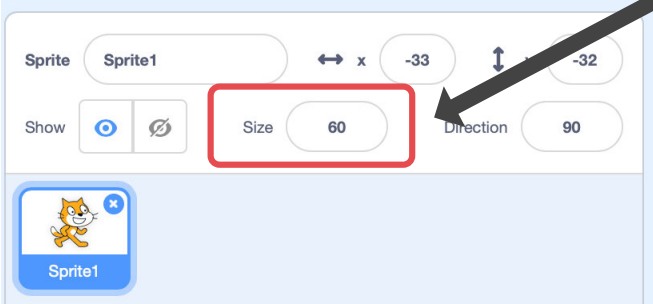
"Choose backdrop"

Setting up our project



4. We are going to choose the "Forest" backdrop as it best represents the game we are trying to create. You may choose or make your own backdrop for your game.

5. We need to change Scratch's size so it better fits our game. Change Scratch's size from 100 to 60. This will reduce Scratch's size by 40%.



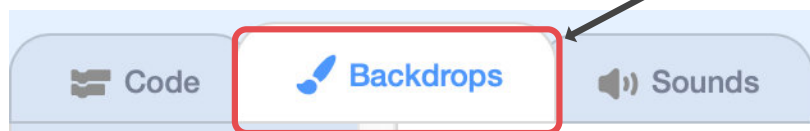
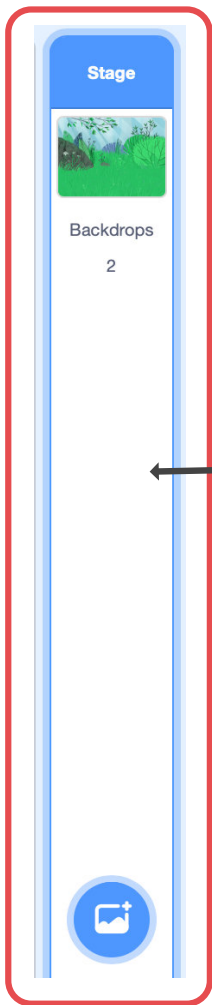
"Change from 100 to 60"

Adding the maze

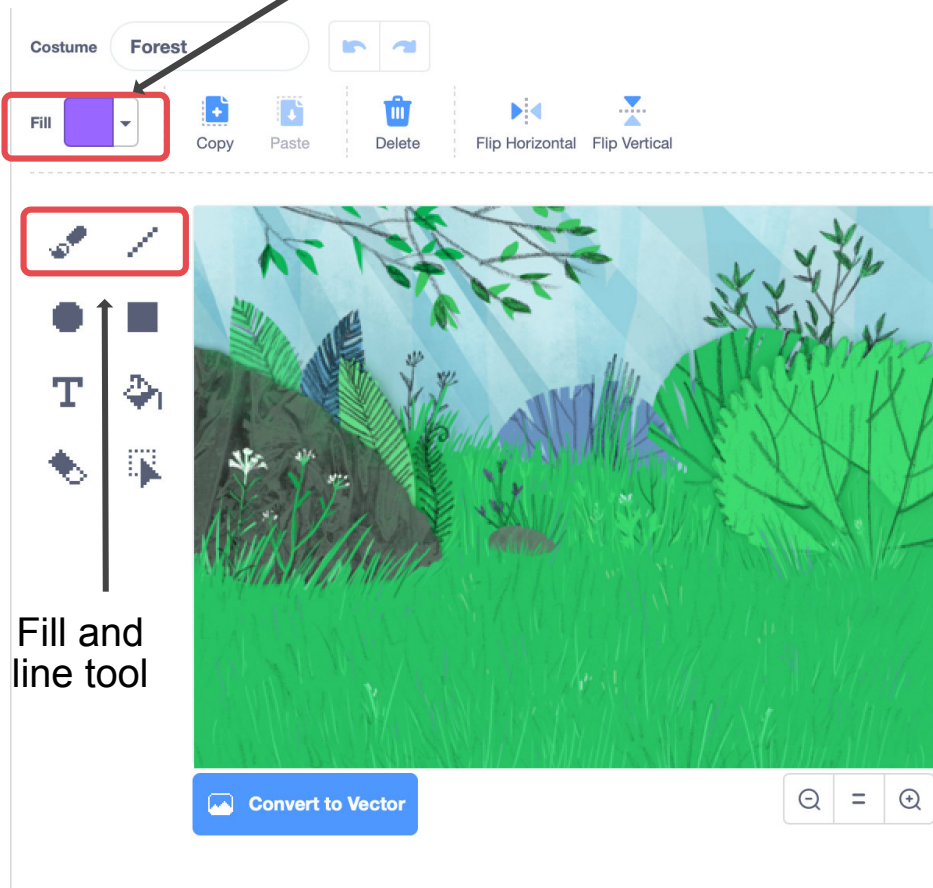
To make our game more interesting, we are going to add a maze. When our player hits any maze wall, they will be sent back to the start of the maze.

6. To add the maze, we first have to select the backdrop. Click on the Stage bar to select the backdrop.

7. Once you have the Stage bar selected, click on the backdrops tab at the top of the screen so we can add our maze walls.



8. Chose a strong colour for our maze walls.

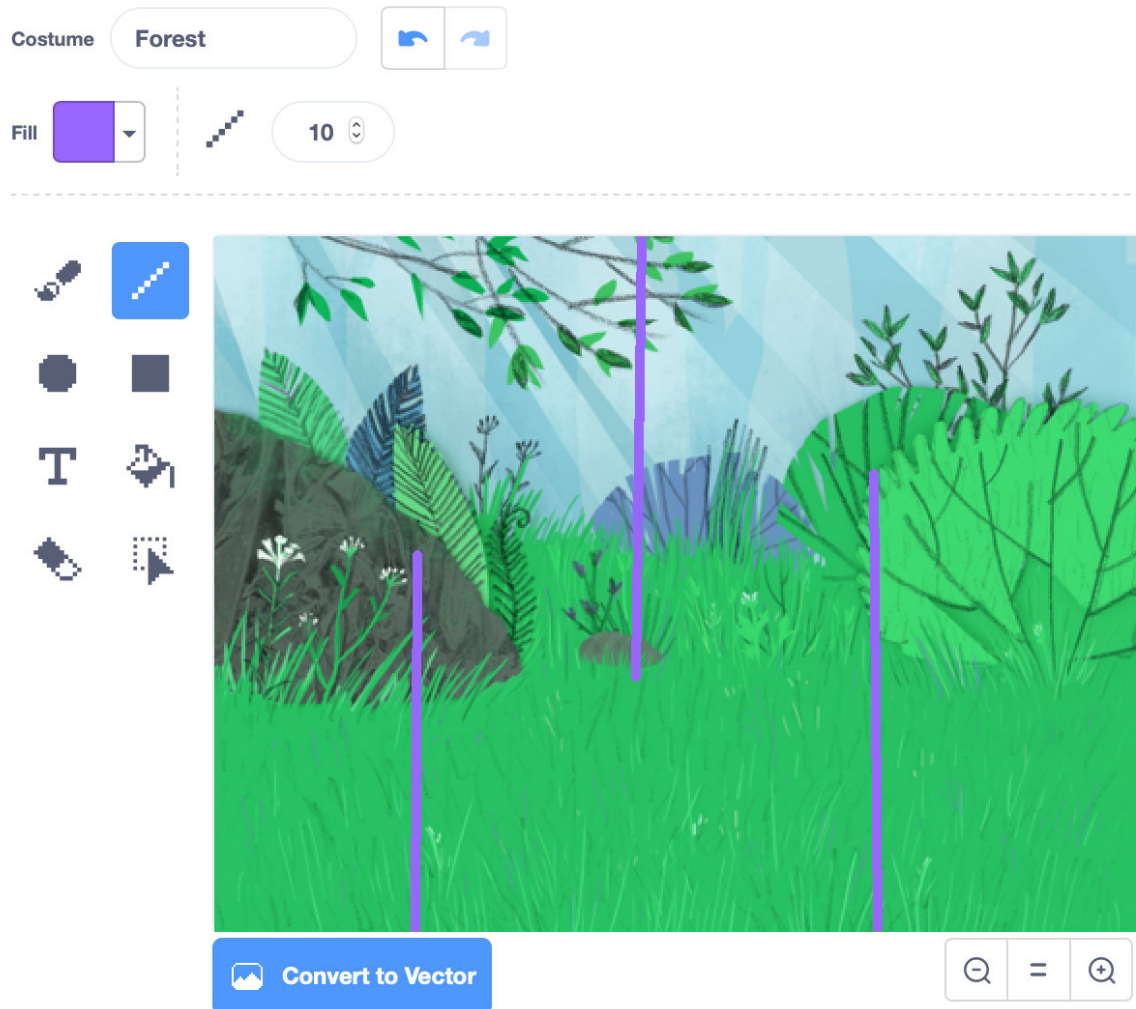


9. Select the line or pen tool and draw a few simple walls for your maze.

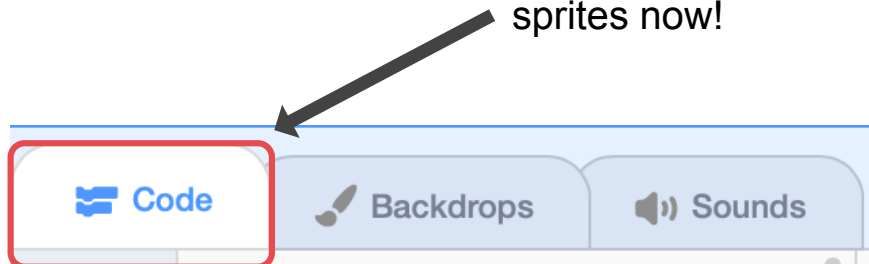
Think about how Scratch will move through the maze. Will the walls be far enough apart for it to move around?

Make sure you select a colour that contrasts from the backdrop. We chose purple because it really pops off the green!

Here is our finished maze. Simple is always better for our first level.

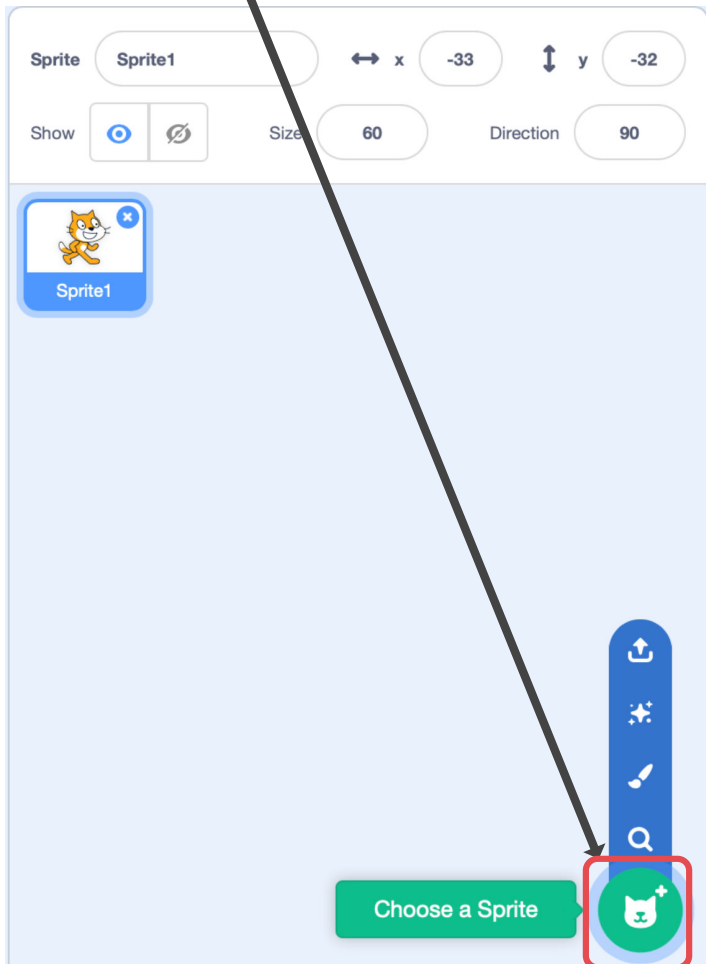


10. Now select the Code tab. We are going to add the rest of our sprites now!

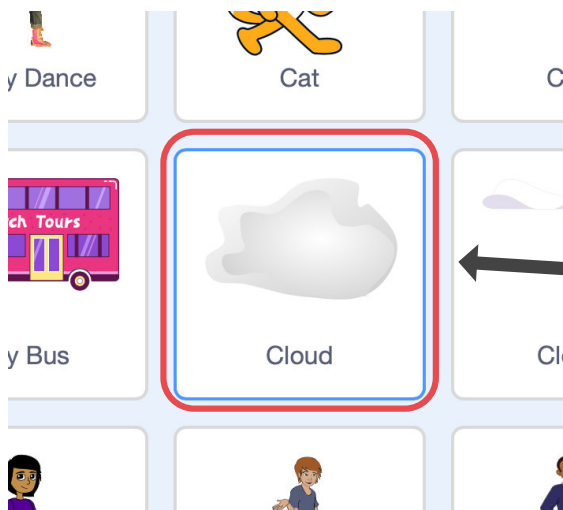
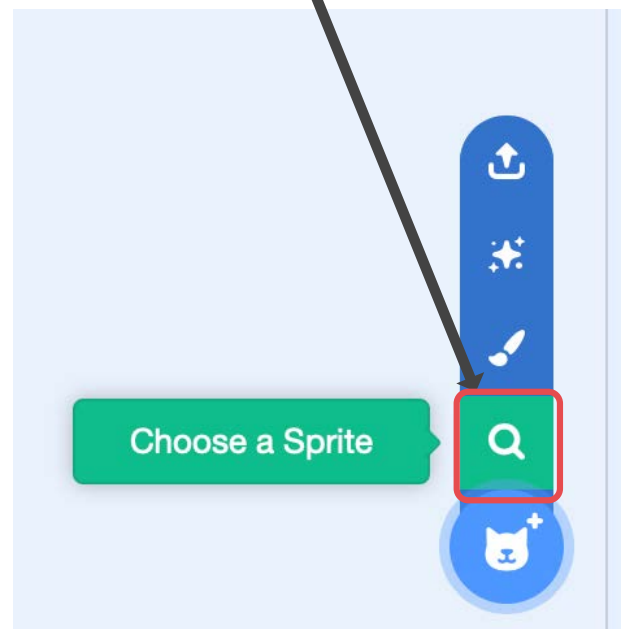


Adding rubbish sprites

11. Click on the Add sprite button at the bottom of the screen.

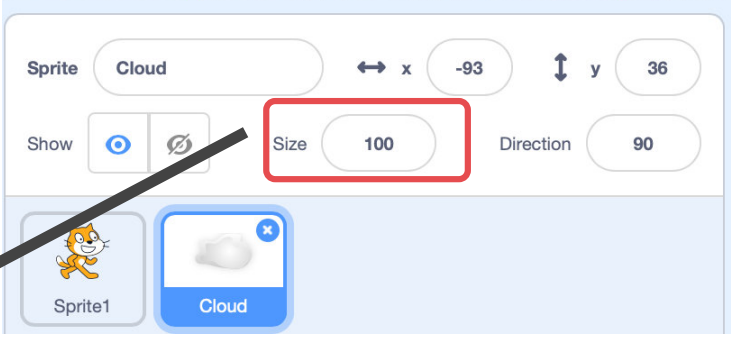


12. Select the "Choose a Sprite" button

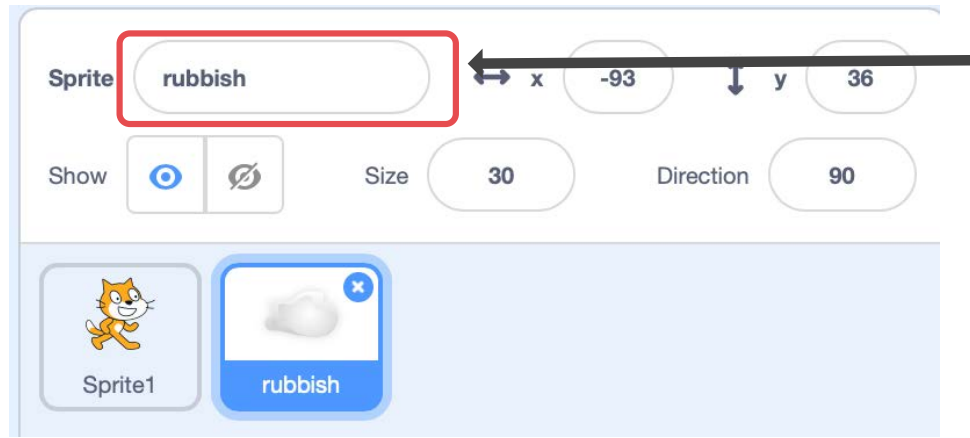


13. First we are going to add a sprite to represent the rubbish that our player will collect. We decided to go with the Cloud sprite because it looked like scrunched up paper. You can choose any sprite you want to represent your litter

14. We need to make our "rubbish" smaller so that it will fit into our Maze. Using the size property, shrink the rubbish sprite from 100 to 30.

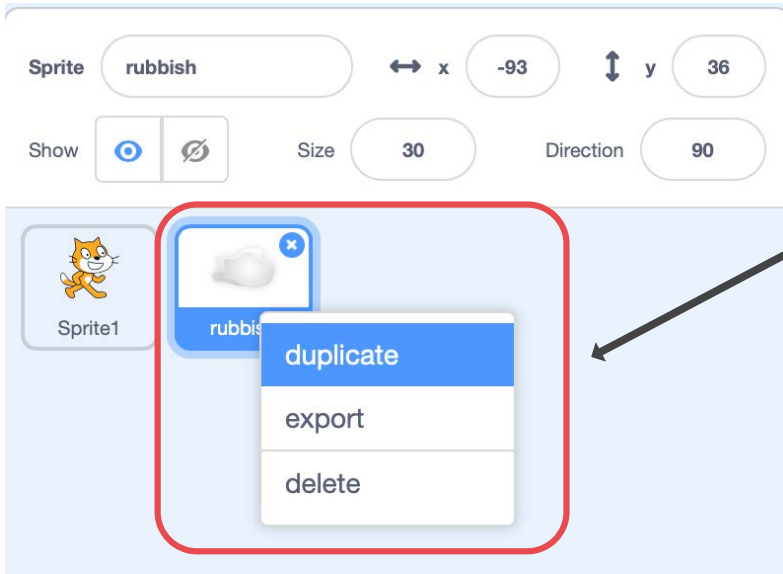


"Change from 100 to 30"



15. Rename the sprite to rubbish. This will make it easier to understand our game as we create more rubbish or if someone wants to remix our Scratch project.





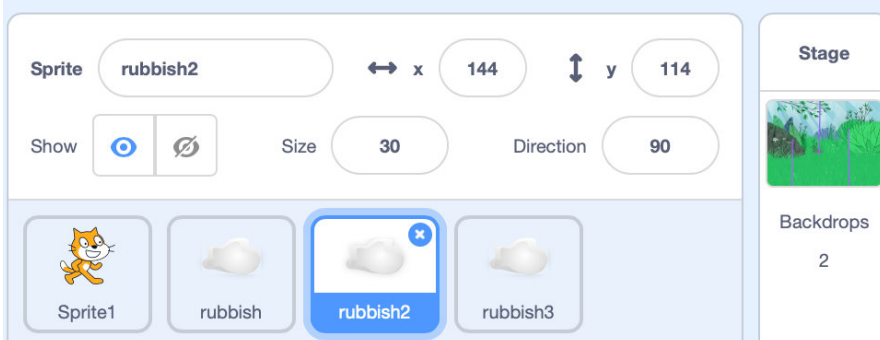
16. Let's create some more rubbish. We do this by right clicking on our rubbish sprite and select duplicate.

Duplicate your rubbish sprite three times!



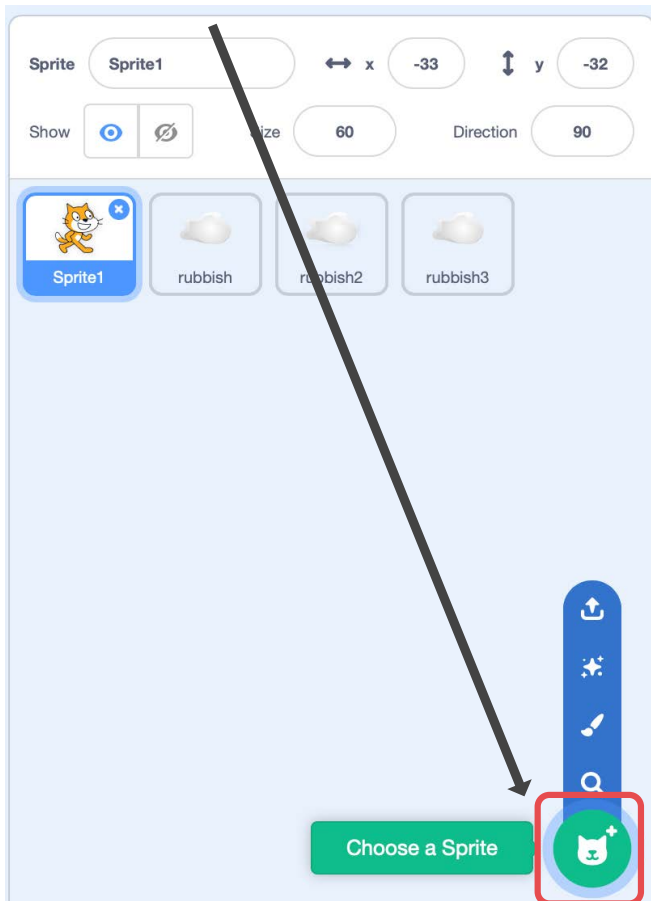
17. Spread the rubbish sprites around the maze.

We can add more rubbish in the future, but for now only add three.

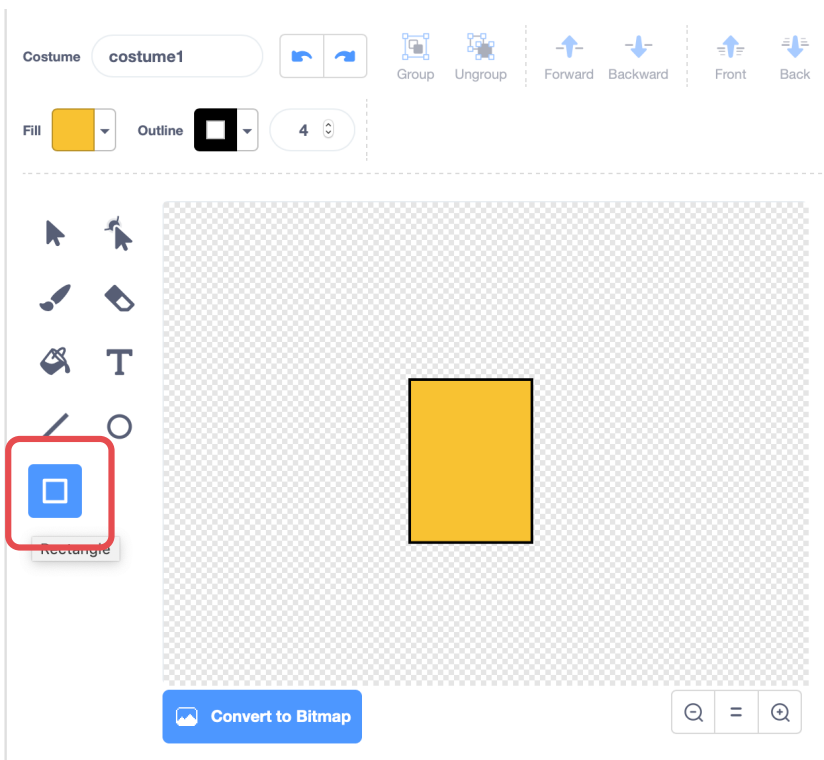
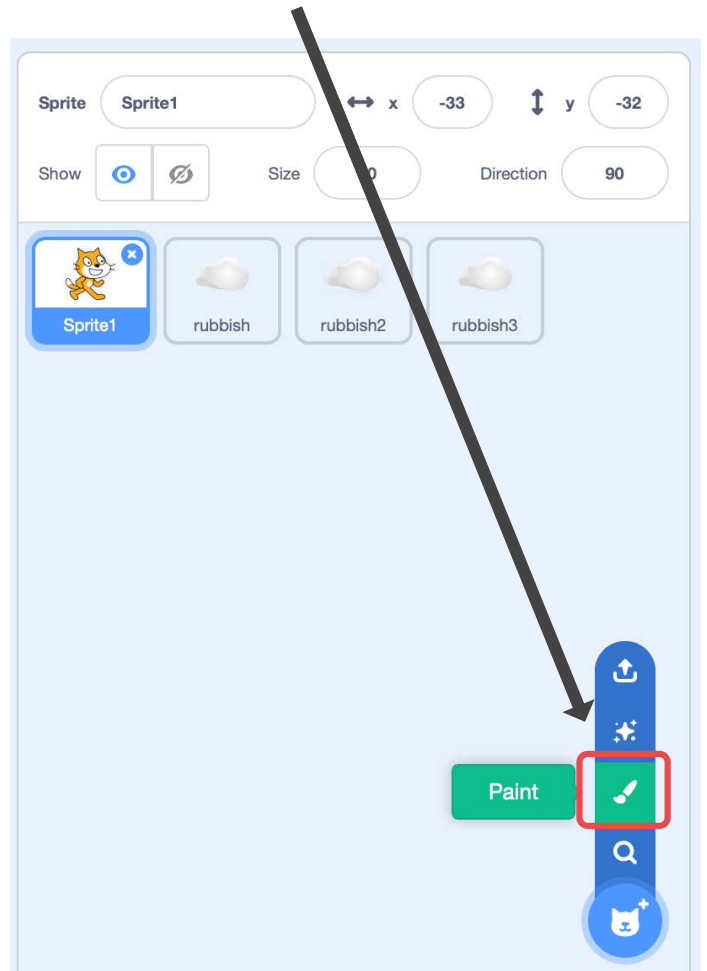


Adding the rubbish bin

18. Click on the Add sprite button at the bottom of the screen.



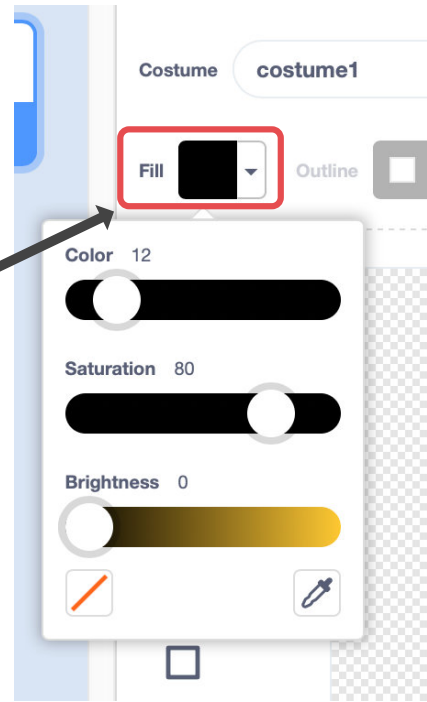
19. Select the "Paint" button



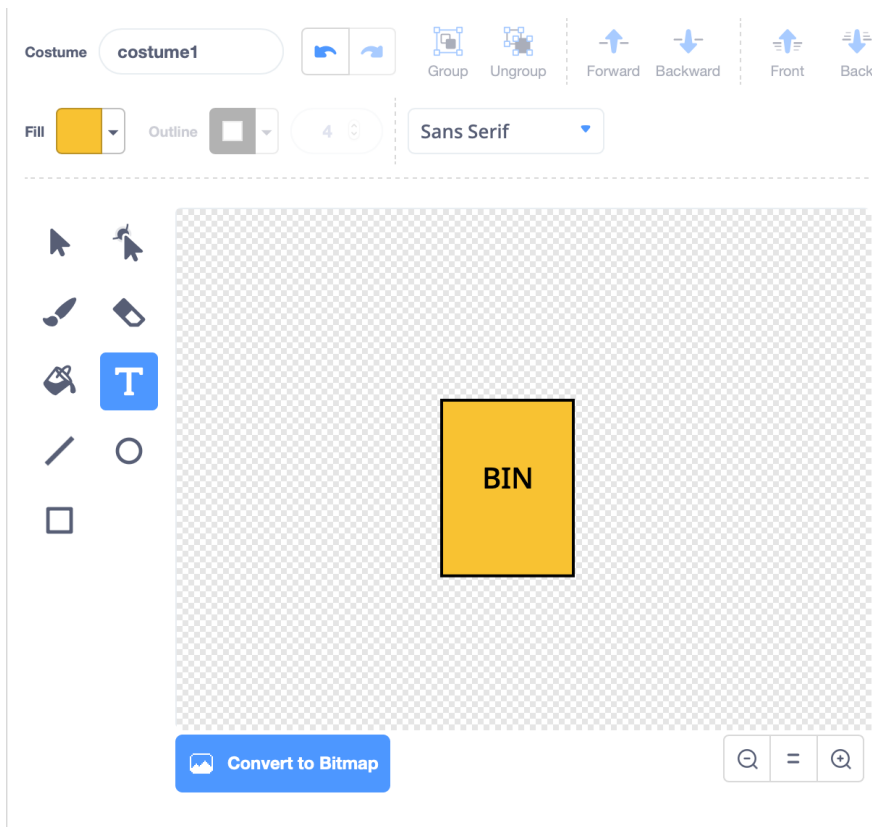
20. Next select the Rectangle tool and draw an rectangle. We will use this as the base of our bin. Choose a nice strong colour to help it stick out from the backdrop.



21. Let's add some text to our rubbish bin. Select the text tool.

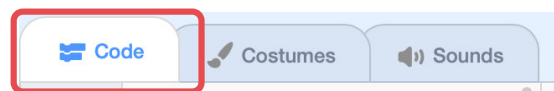


22. Select the Fill tool and change the colour to black.

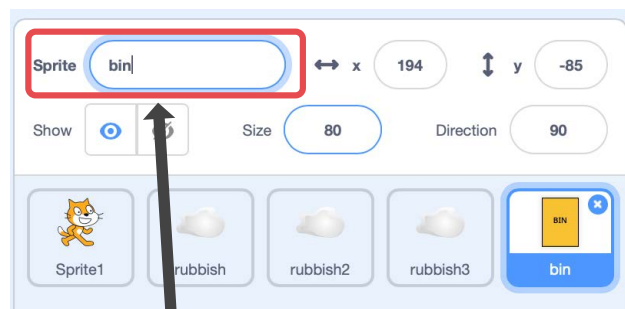
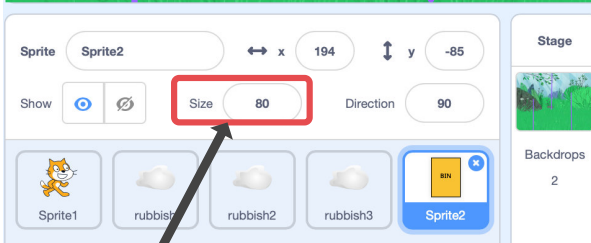
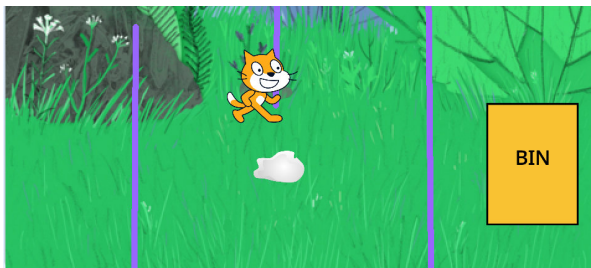
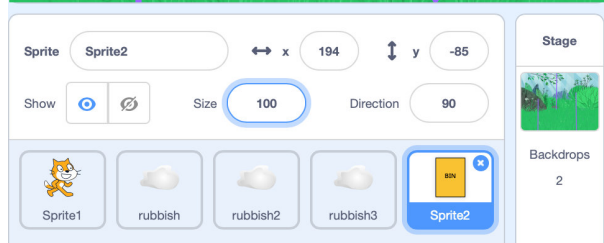
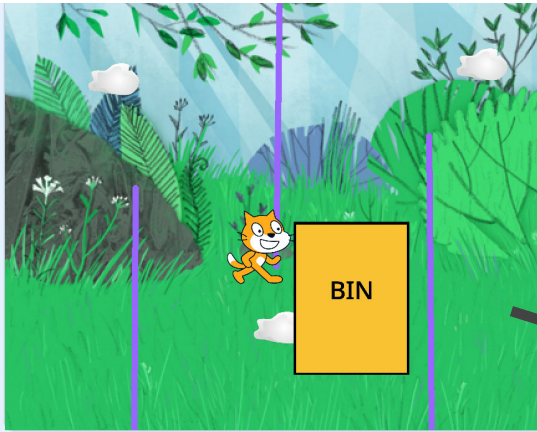


23. Now, with the text tool still selected, we are going to click in the centre of the rectangle we made. Type the word BIN in capitals! You can use the arrow tool to move the text around to better center it.

24. Select the "Code" tab. We need to place our new bin at the end of the maze.



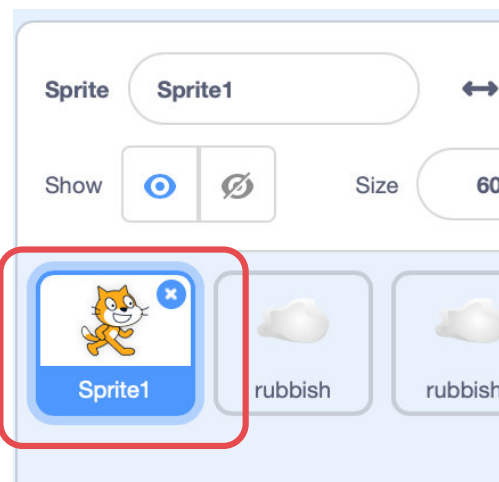
25. Drag and click our new bin sprite to the end of the maze. When Scratch touches the bin the game will end



26. Make our bin sprite smaller by changing it's size to 80.

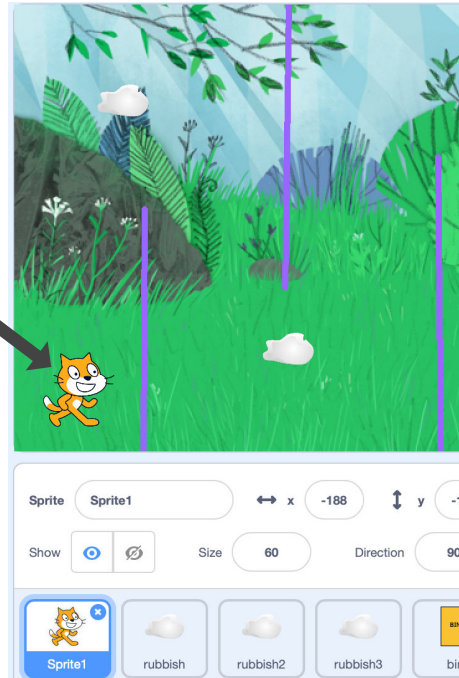
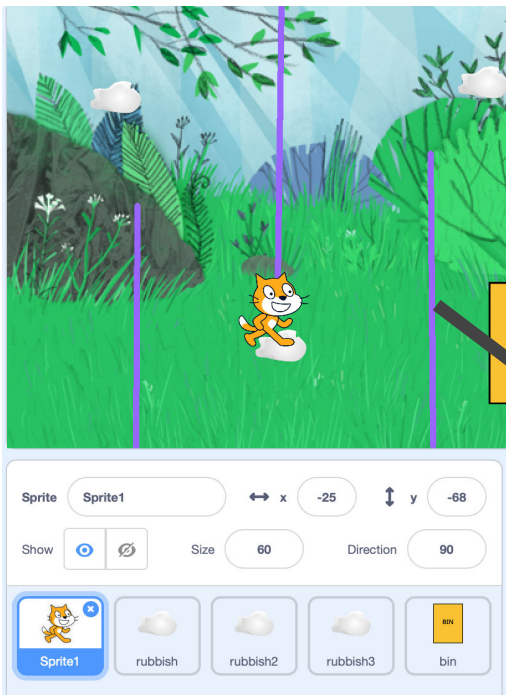
27. Change the bin sprites name to "bin"

28. Now select Scratch the cat. We are going to add our code!!

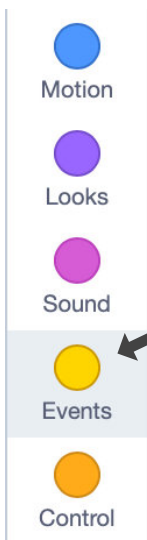


Lets add some Code!!!!

29. Drag and click Scratch and move it to the start of the maze.

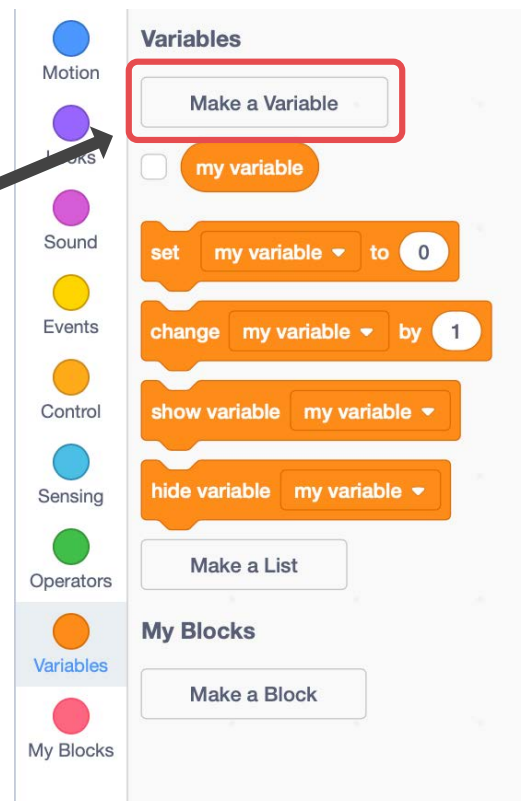


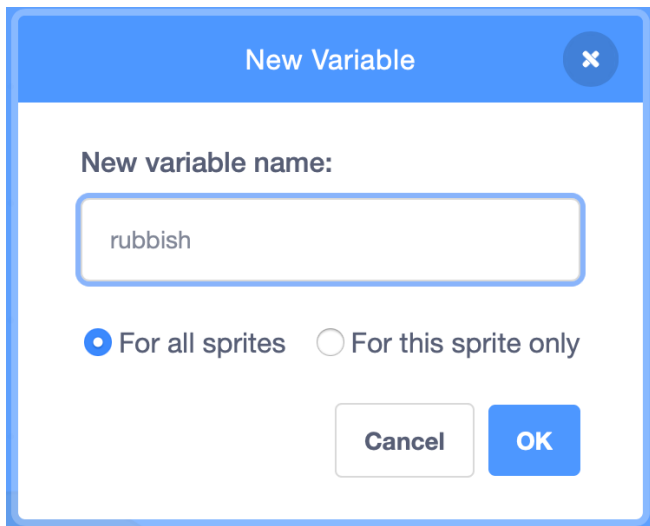
30. To start, we need add an event to trigger our code. Go to the Events code draw and drag the when green flag clicked block into the code zone.



31. Our player needs to collect 3 pieces of rubbish so we need to create a variable to hold this information. A variable is a type of container we can use to hold data.

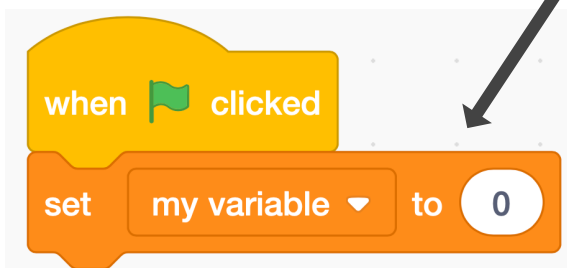
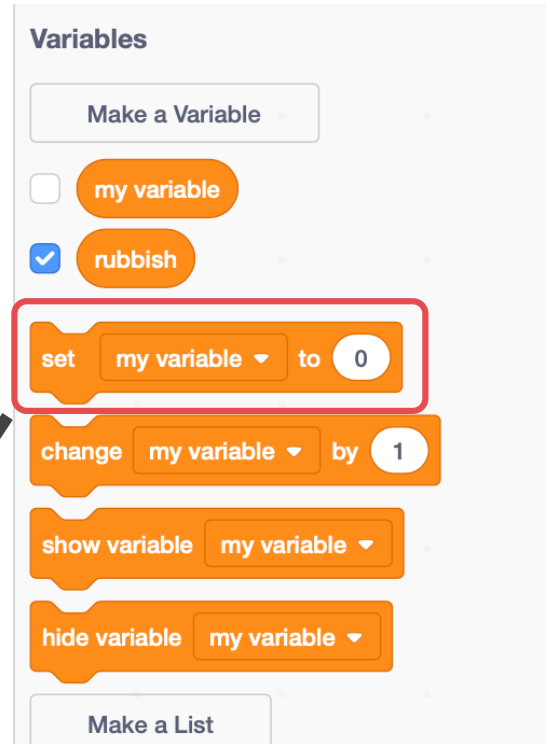
32. Go to the variables code draw and click the "Make a Variable" button.



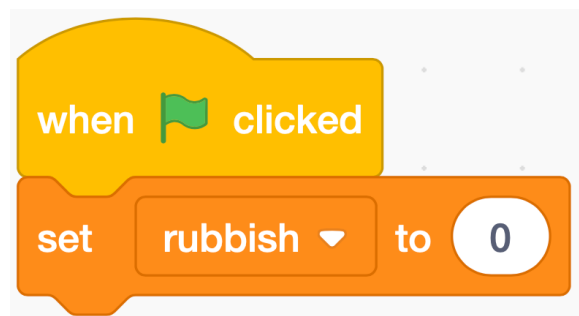
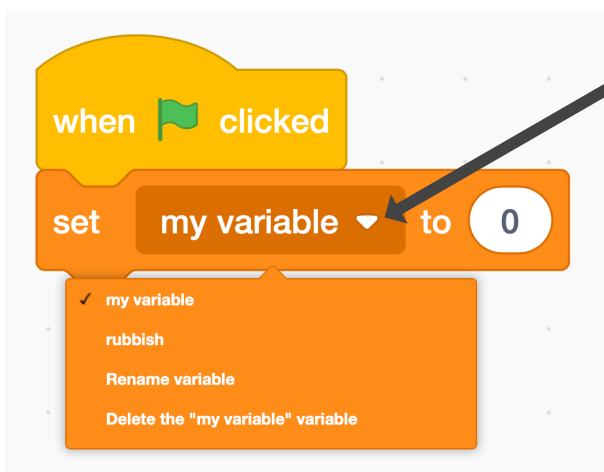


33. We need to give our variable a good name. We are going to name our new variable "rubbish". Click OK when you are done.

34. You will now see our new variable "rubbish"! Drag the 'Set 'my variable' to 0' underneath the "When green flag clicked" event.

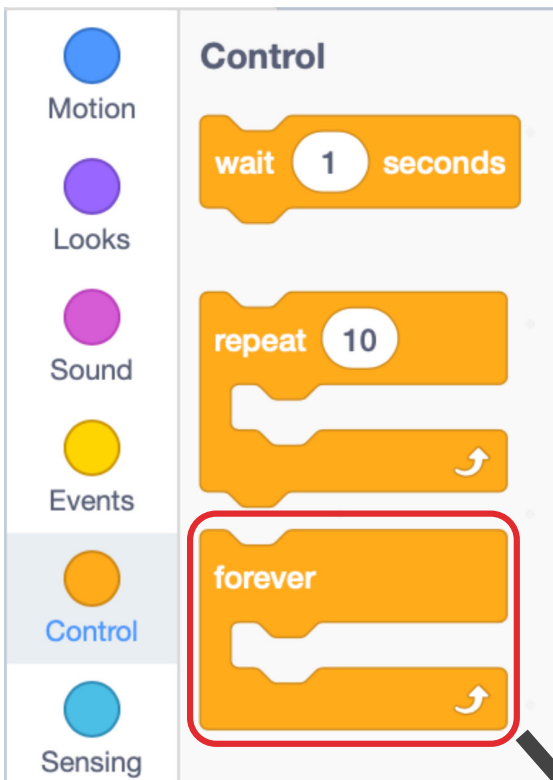
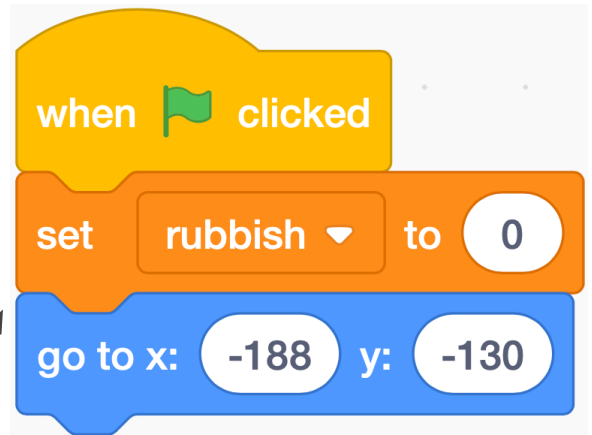


35. Click the the white down arrow and select our new variable called rubbish.





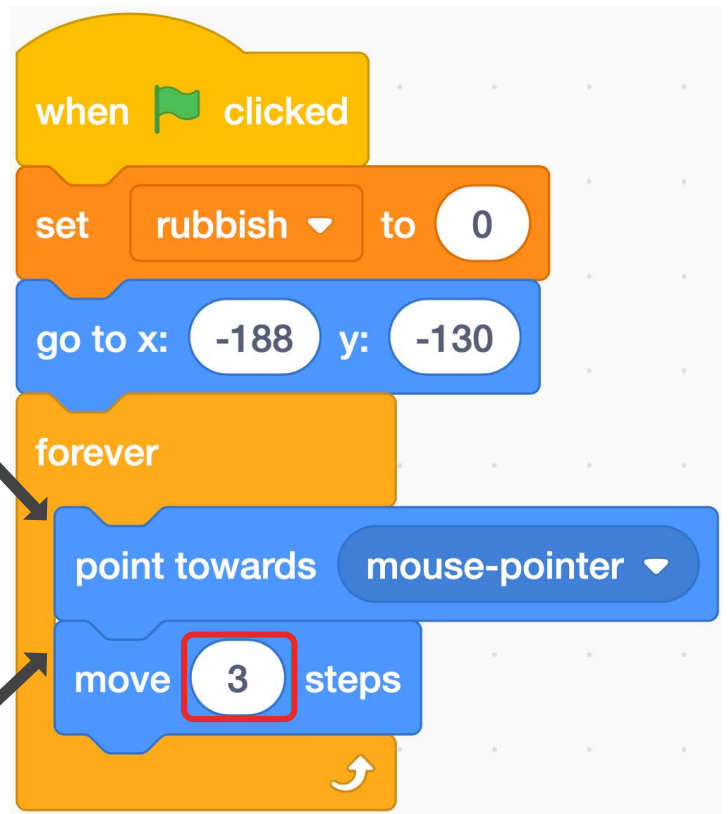
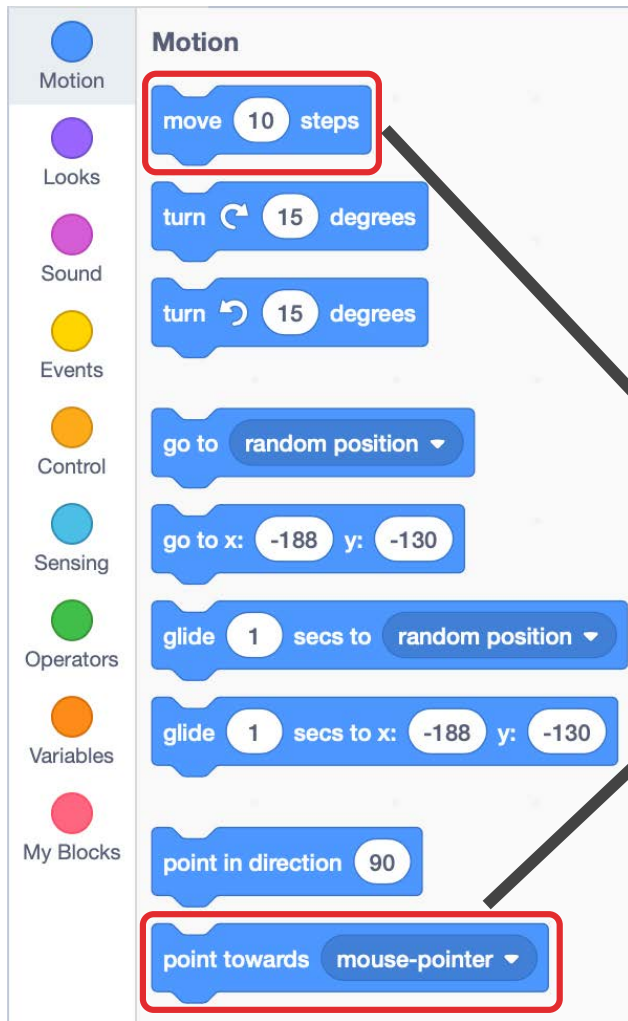
36. We need to add a reset location for our player when we start a game. Go to the motion code drawer and drag the "Go to x:0 y:0" code block under the variable block. This block will automatically show the location of Scratch's current location on the backdrop.



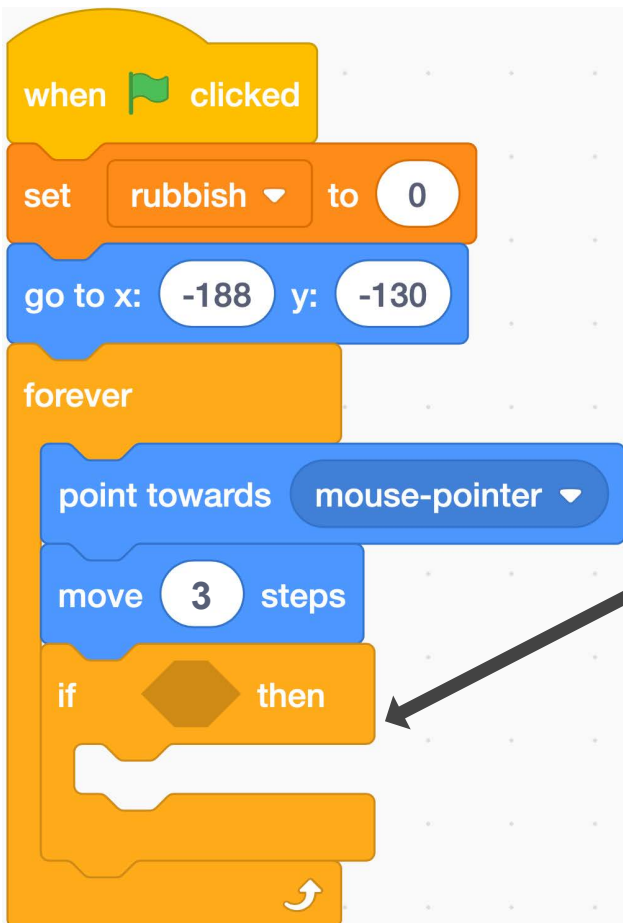
37. Go to the Control code draw and drag the "forever" loop block under the "go to" block.

We use the "forever" block to run our main game code.

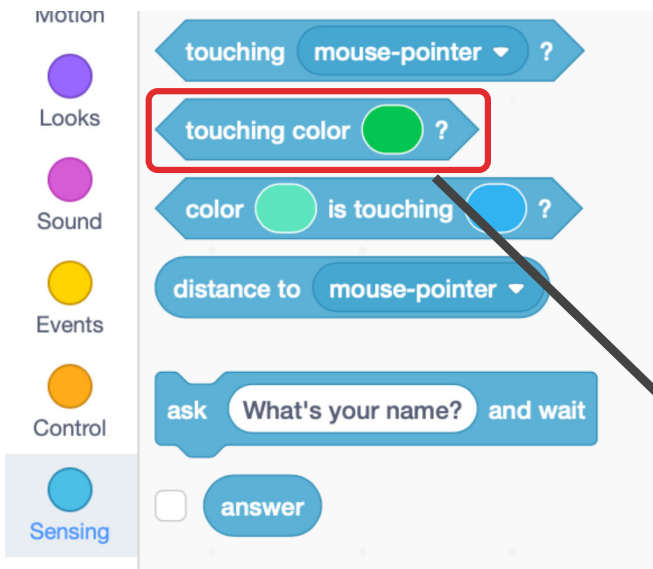




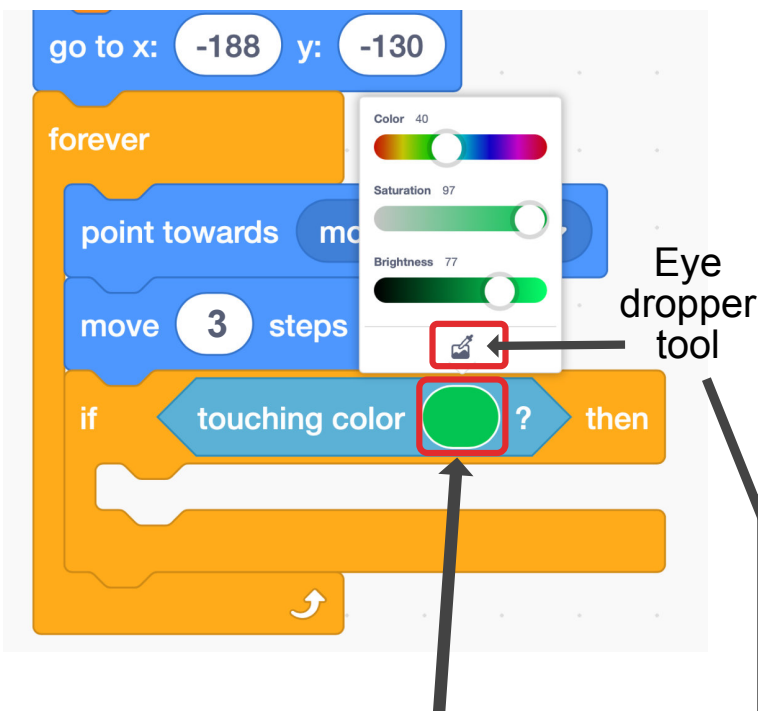
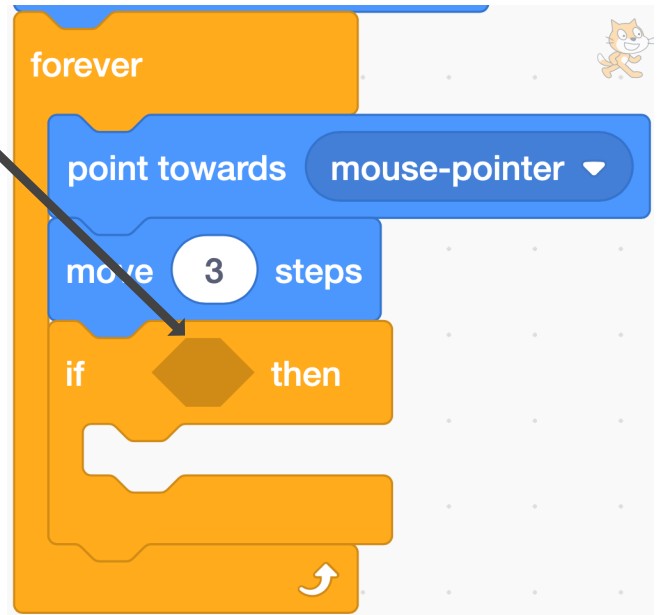
38. Go to the 'Motion' code draw and drag the "Point towards mouse-pointer" block. Now add a "move 10 steps" block and change the 10 to a 3 by clicking on it's input box. We use this block to control the players movement speed. The bigger the number, the faster our player will move.



39. Go to the control code draw and drag a IF block into out code. We use IF blocks to help our code make decisions. We are going to use this IF block to detect if the player touches the maze wall

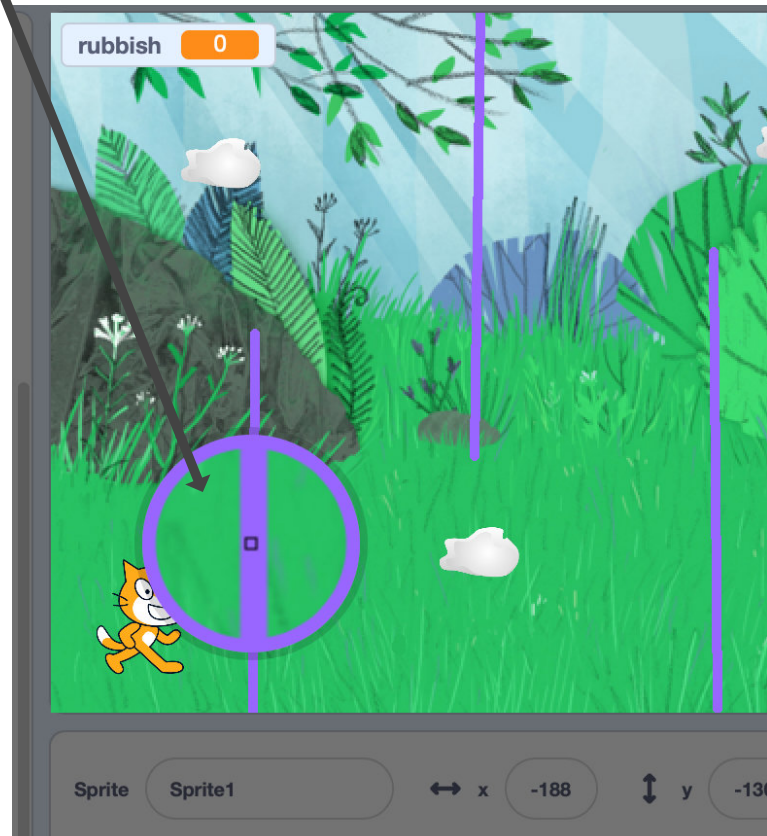


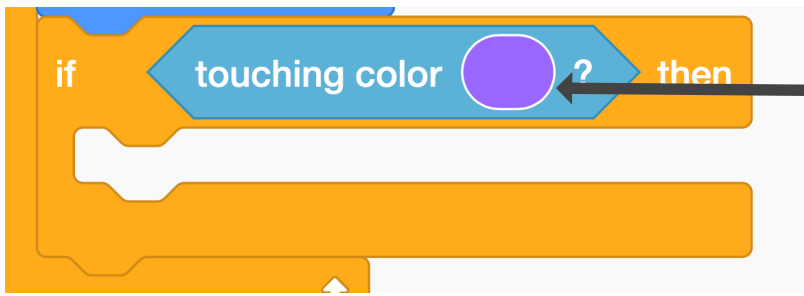
40. Select the Sensing code draw and drag the "touching colour?" condition block. We need to drag this block inside the IF blocks input space. You will notice that the condition blocks are shaped like hexagons.



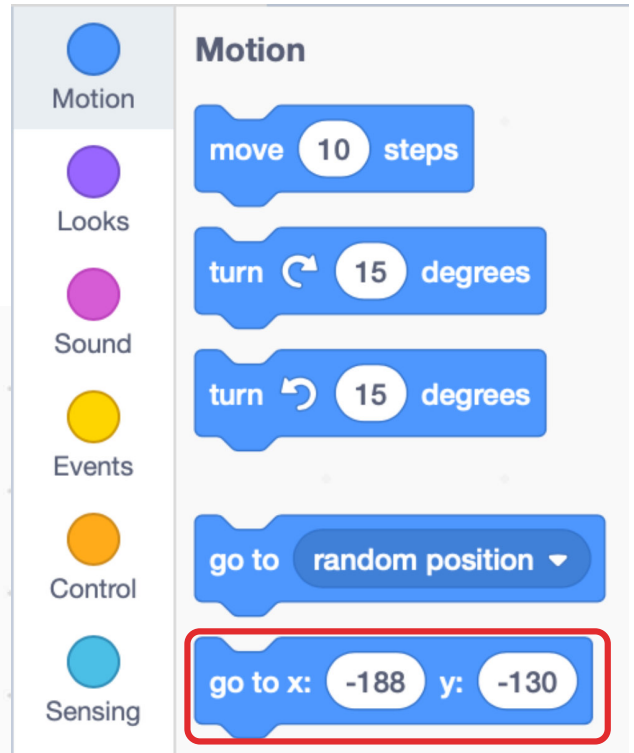
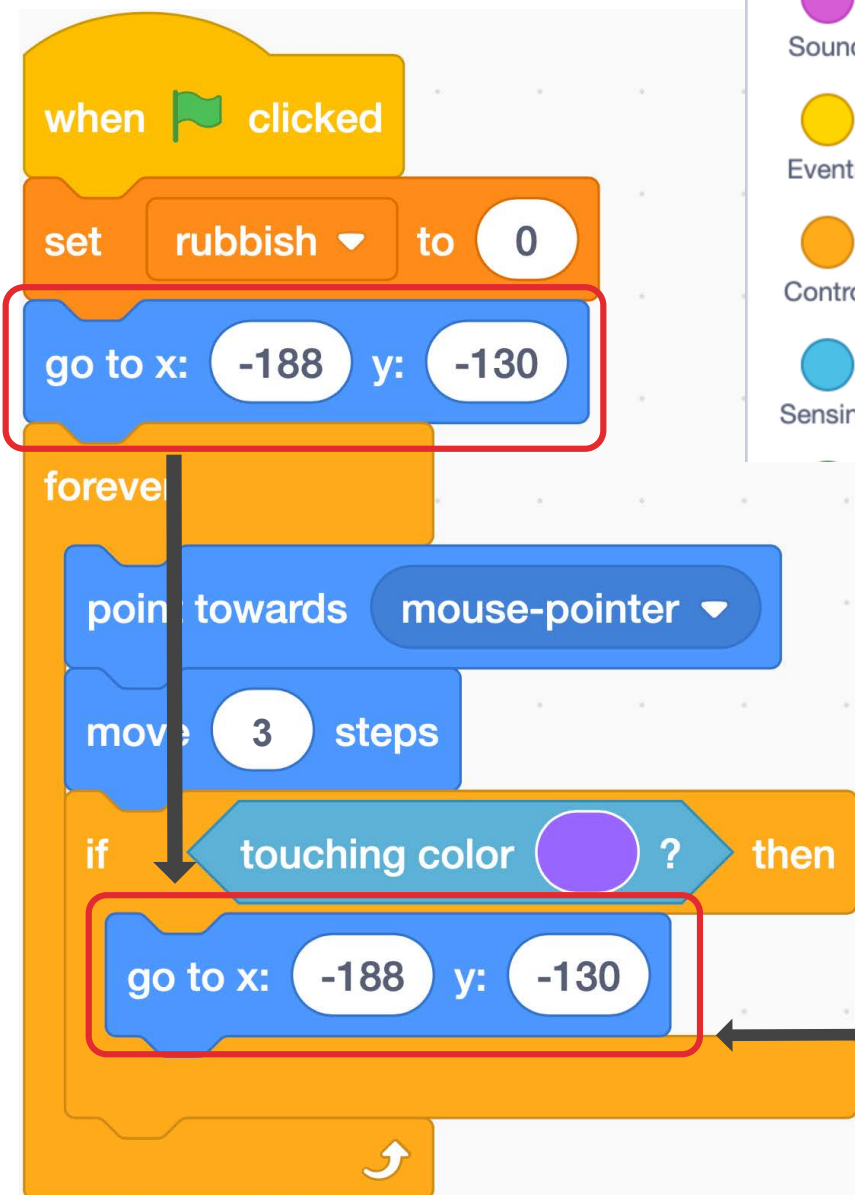
41. We need to change the colour of the "touching colour?" block to match the colour of our maze walls. Click on the colour circle of the "touching colour?" block and select the eye dropper tool.

Now use the eye dropper tool to select the colour of the maze walls.



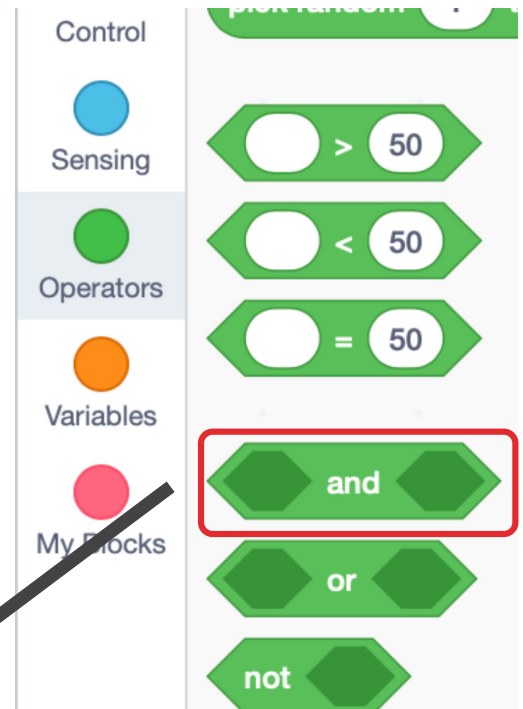
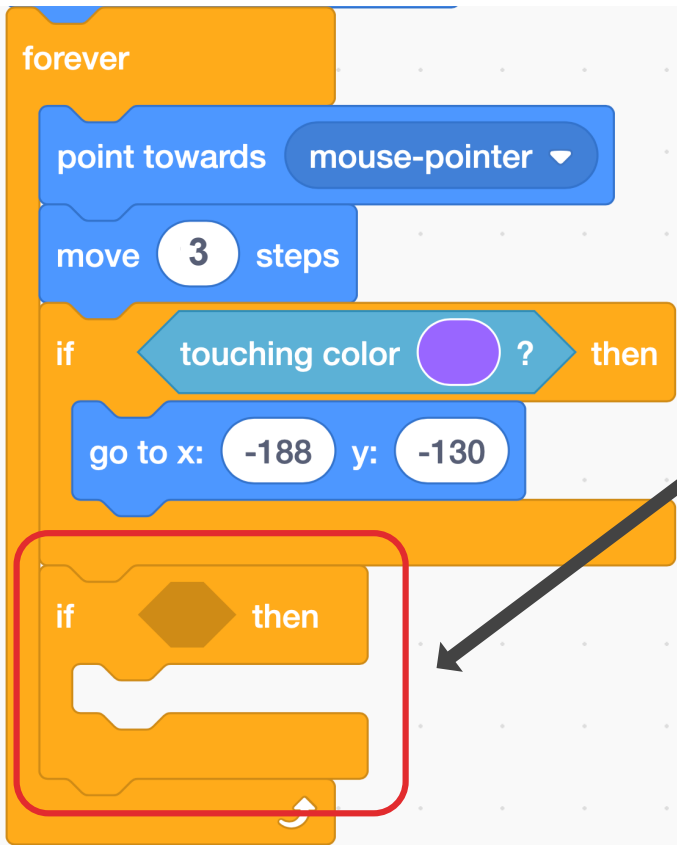


42. Our "touching colour?" block should now match the maze wall.

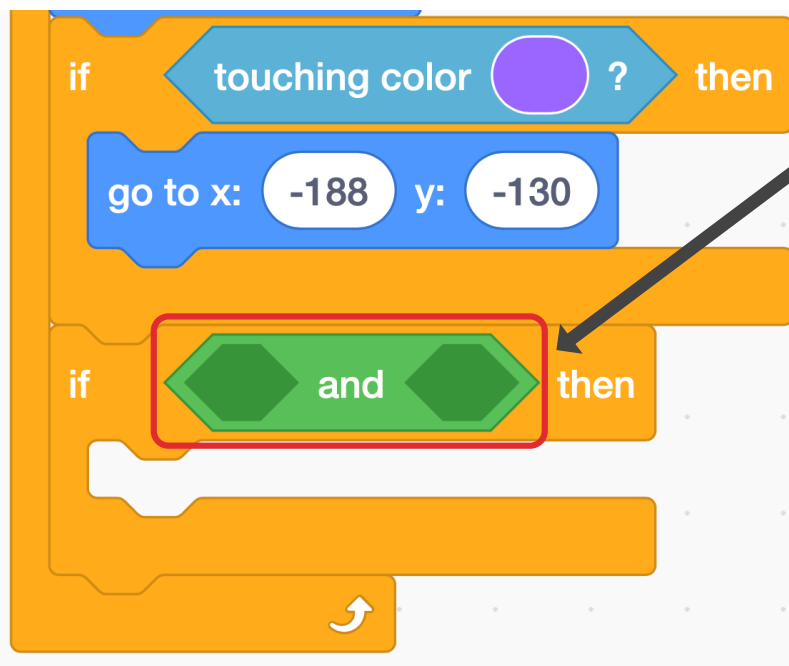


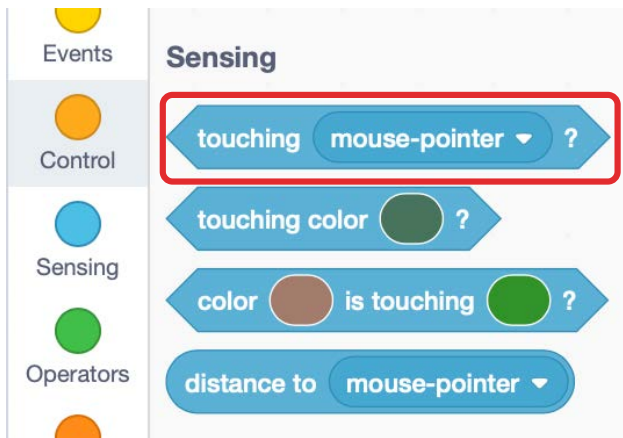
43. We need to add a reset location for our player when they touch the maze wall. Go to the motion code drawer and drag the "Go to x:0 y:0" code block under the variable block. Make sure the X and Y coordinates match the "go to X:0 Y:0" block we placed in the beginning.

44. We now need to add another IF block underneath our previous IF block. This IF block will be used to decide if our player has collected all the rubbish when they touch the rubbish bin we created.

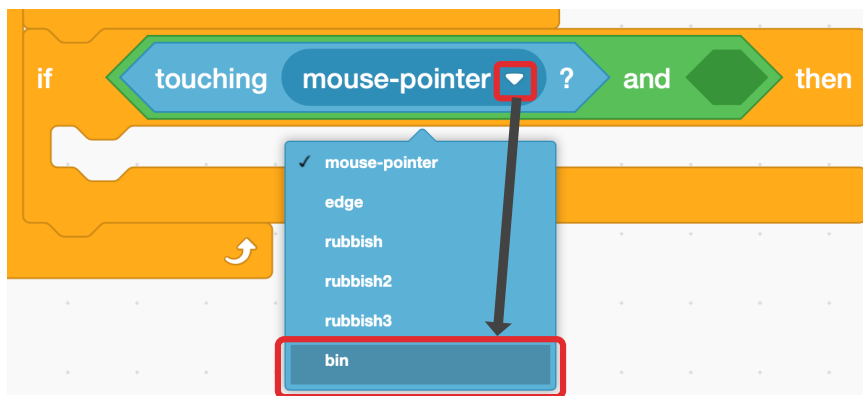
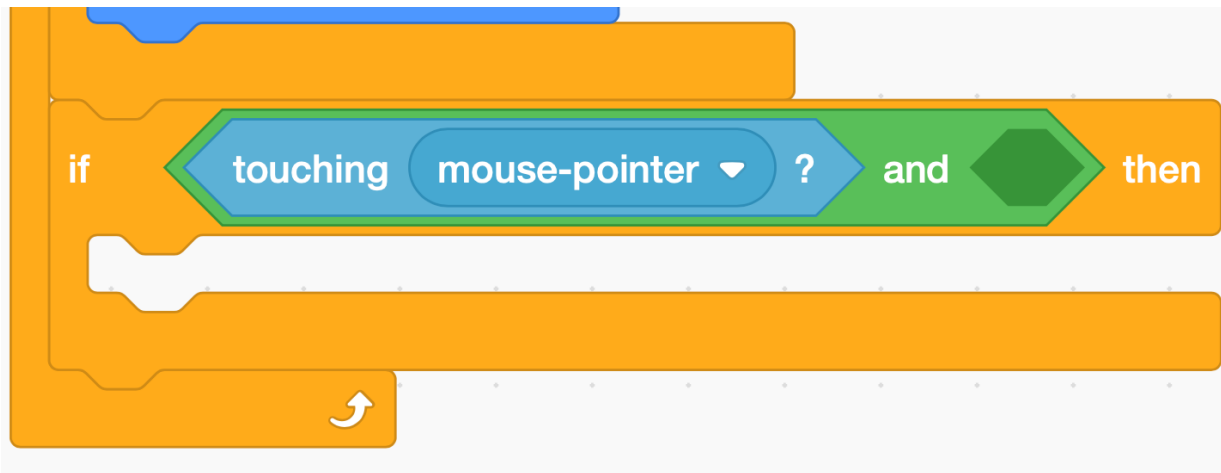


45. Our IF block needs to check if the player is touching the bin AND has collected all the rubbish. To have our code check both of these conditions at the same time we use the AND block. Go to the operators code draw. We need to drag this block inside of the IF conditions input space.

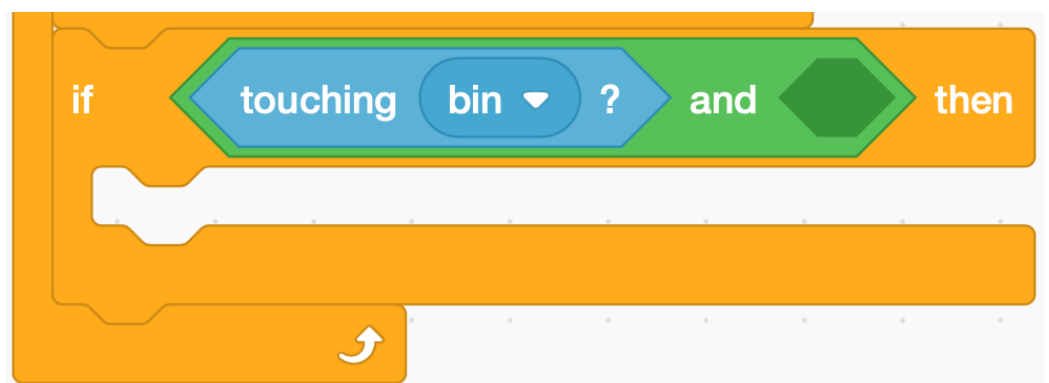


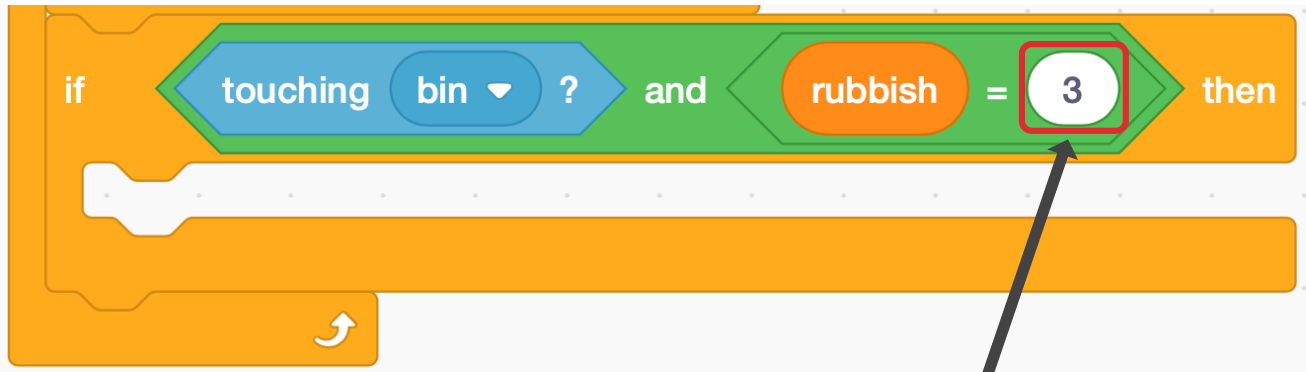


46. Lets add the first condition to our IF block. Go the sensing draw and add the "touching object?" block. Drag this block into the first input space of the AND block.

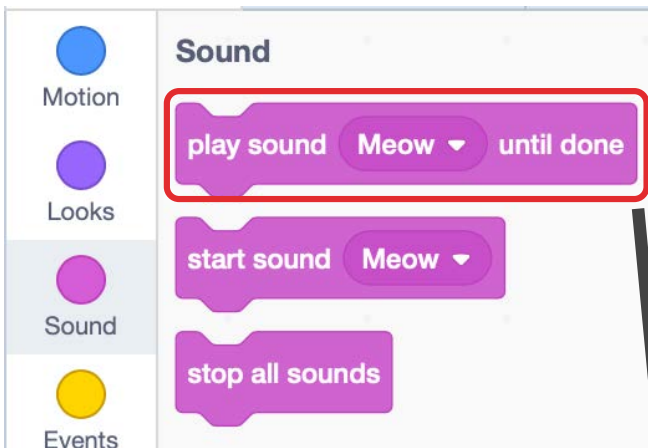


47. We need to change the "touching object?" block to our bin object. Click on the white drop down arrow and select our bin.



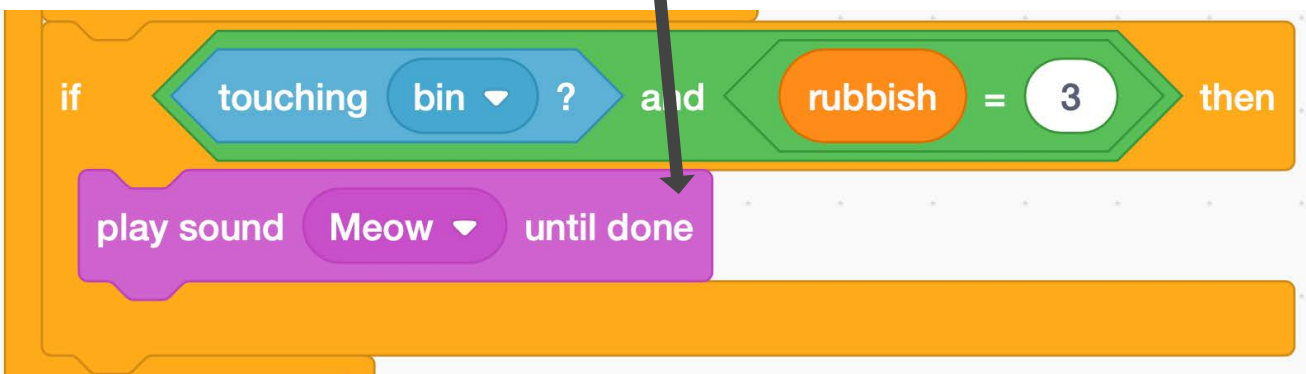


50. We need to change the number in the "Equal to" block to match the total amount of rubbish the player must collect our level. In this example we have added three pieces of rubbish so we are going to change this number to match.



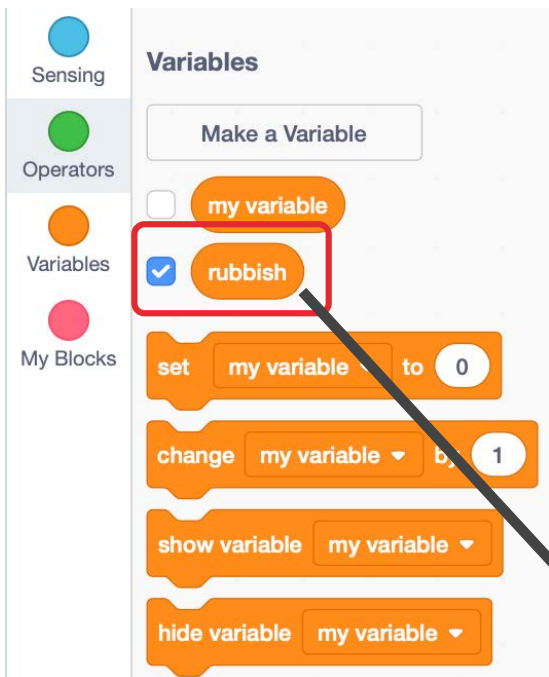
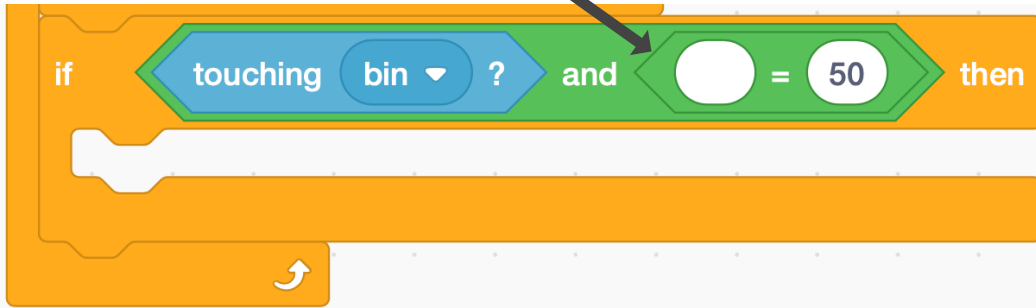
51. To show our player that they have successfully collected all the rubbish. We are going to have our code play a sound to show this.

Go to the sound code draw and drag the "play sound Meow until done" into the IF block we just added.

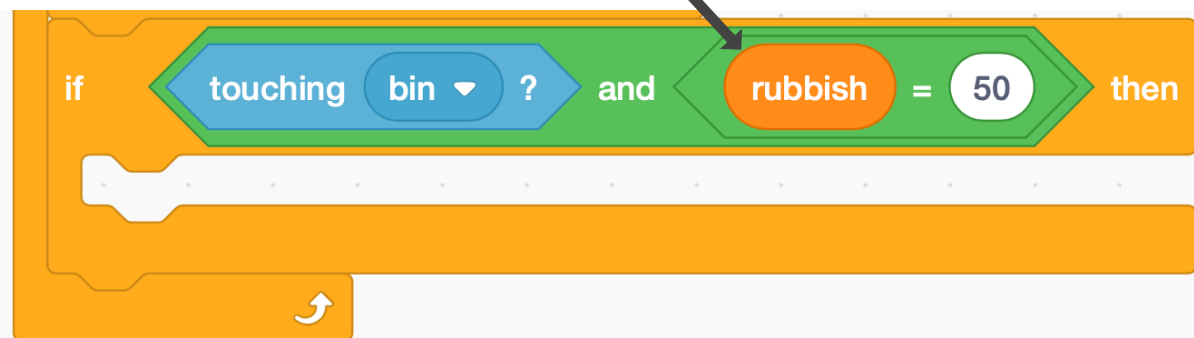




48. We now need to add a operator block that checks the rubbish variable to make sure players have collected all the rubbish. Go to the operator code draw and drag the "Equal to" block into the second input in the AND block.



49. Go to the the Variables code draw and drag the rubbish block into the empty input in the 'Equal to' block.



```

if touching bin ? and rubbish = 3 then
  play sound Meow until done
  say You win! for 2 seconds

```

Looks

- say Hello! for 2 seconds
- say Hello!
- think Hmm... for 2 seconds
- think Hmm...

52. We want to say something encouraging to our player. Something that celebrates their success at completing the level.

Note: you can say something in any language you want!

```

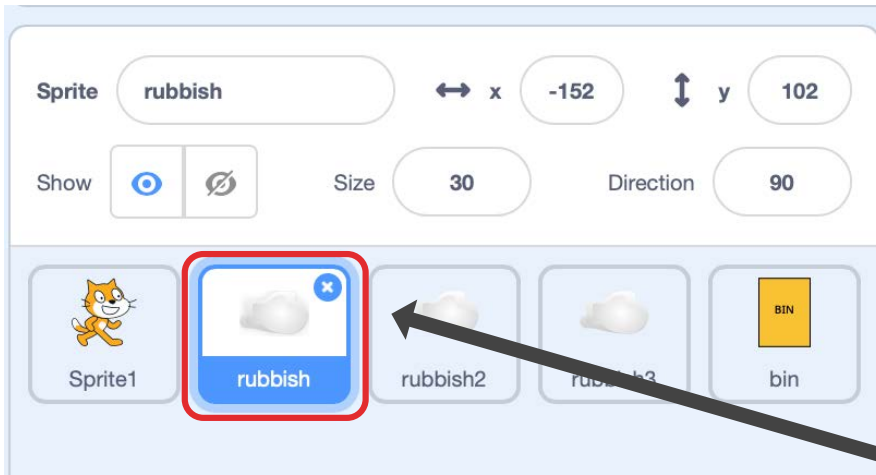
if touching bin ? and rubbish = 3 then
  play sound Meow until done
  say You win! for 2 seconds
  stop all

```

Events

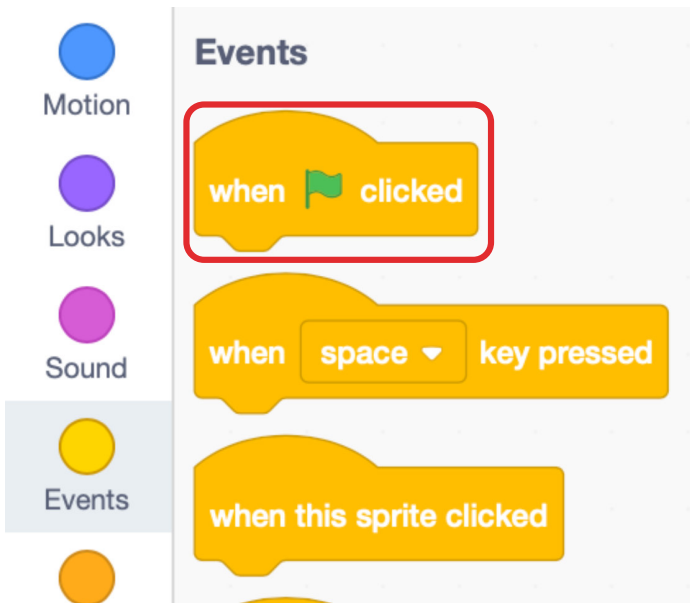
- stop all
- when I start as a clone
- create clone of myself
- delete this clone

53. To finish off our player code we need to add a “stop all” block which will stop our program after we win the game.

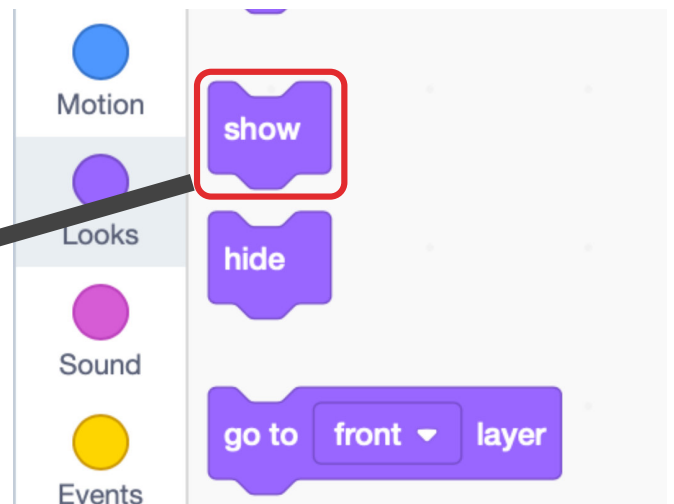
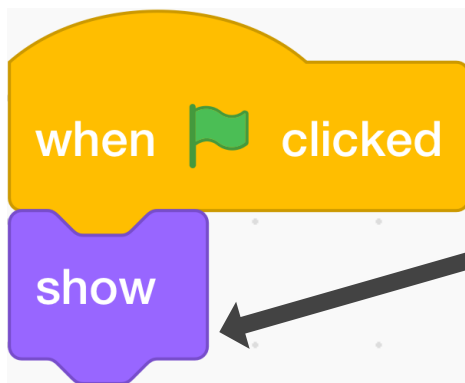


To finish our game, we need to add code to the rubbish sprites that add one to our rubbish variable and make the rubbish hide.

54. Select the first rubbish sprite.

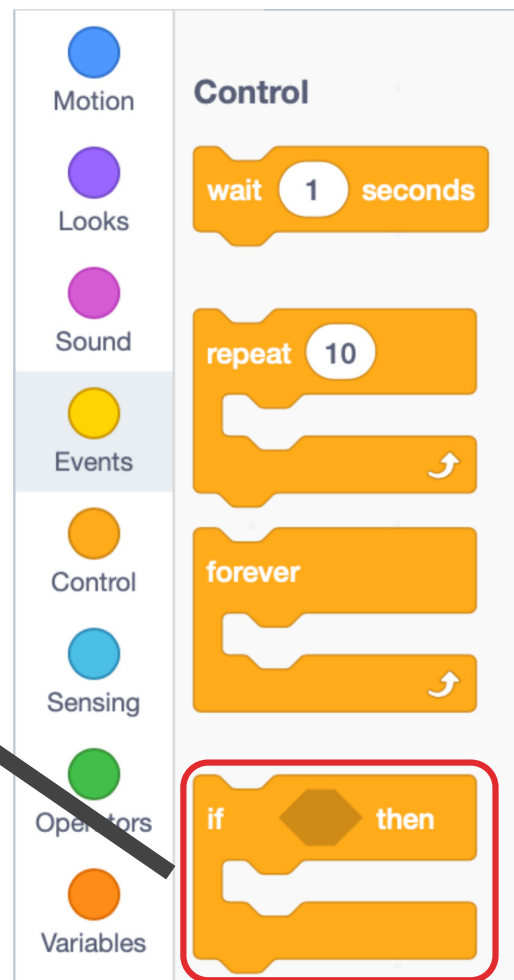
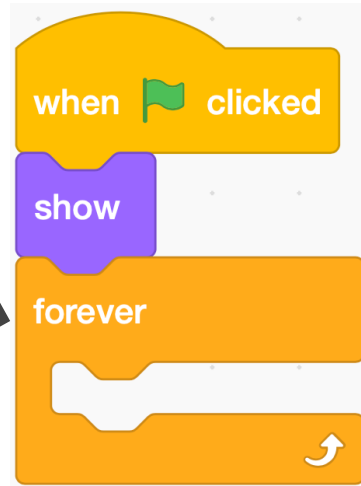
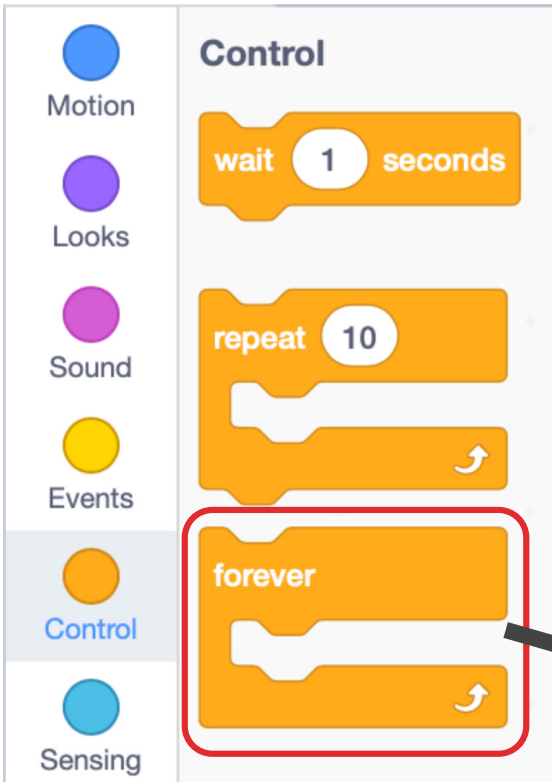


55. Go to the the Event code draw and drag a “When Green flaged clicked” event into the coding area.

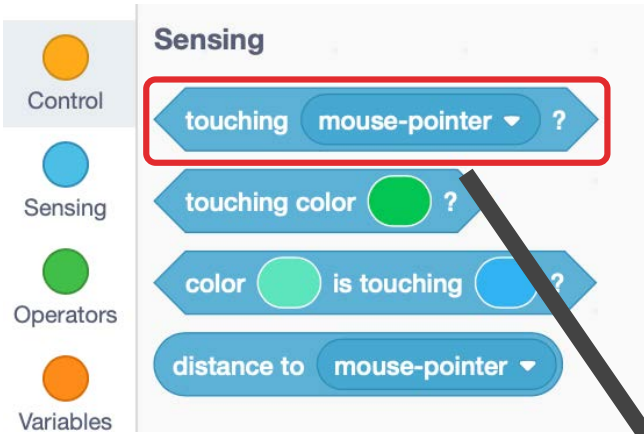


56. Now go to the Looks code draw and drag a show block under the event we just added. This makes the rubbish visible when we restart the game.

57. Now go to the Control code draw and drag a forever block under the show code block. We need this block to keep checking if the rubbish has been collected by the player.

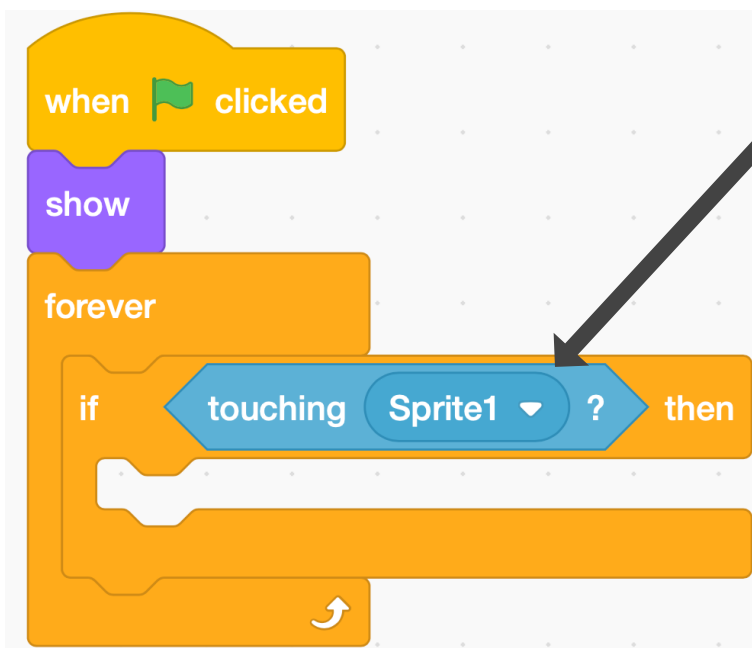
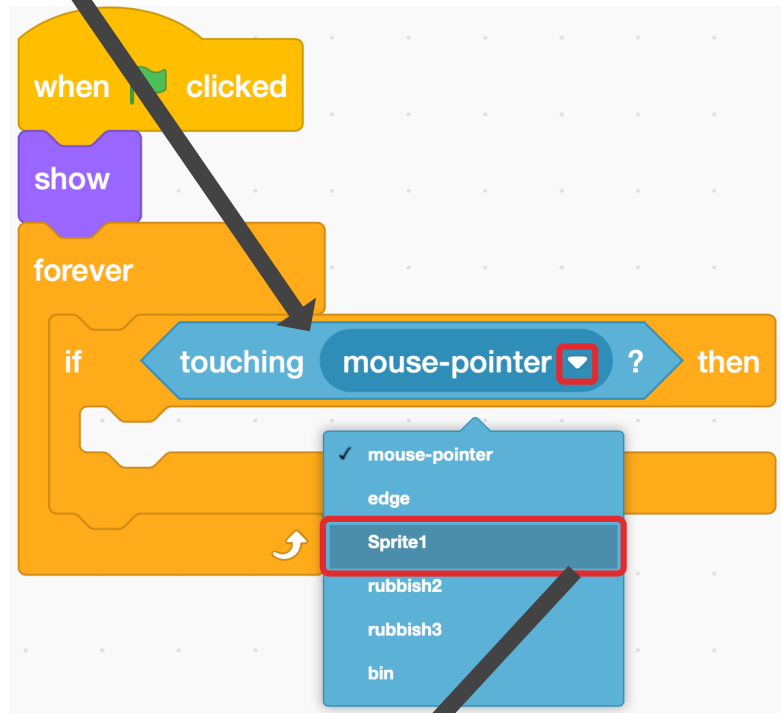


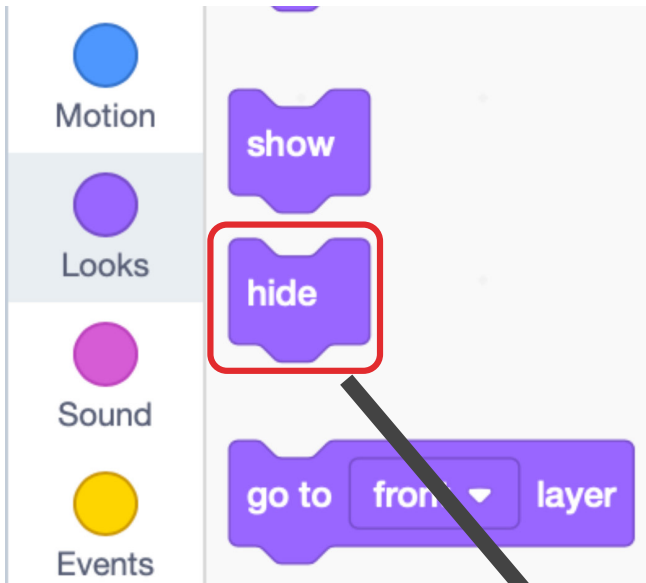
58. While still in the control code draw, drag an IF block inside of the Forever mloop we added previously.



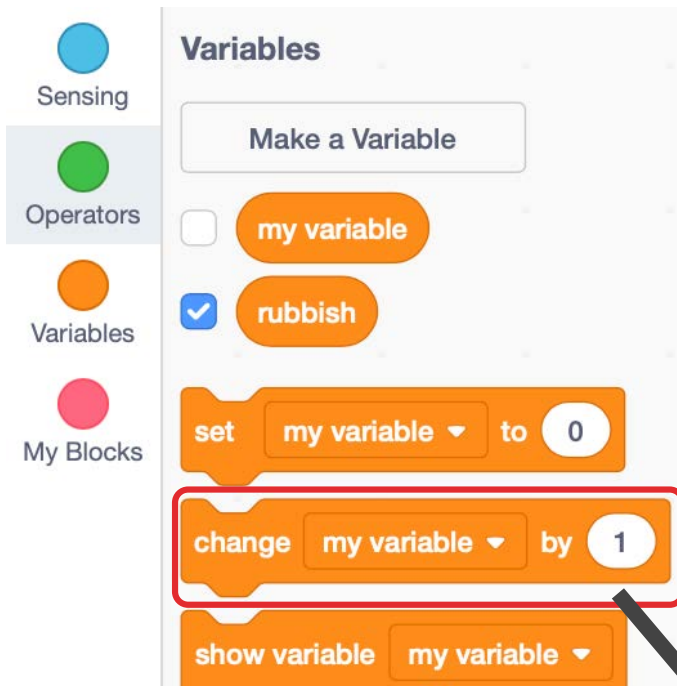
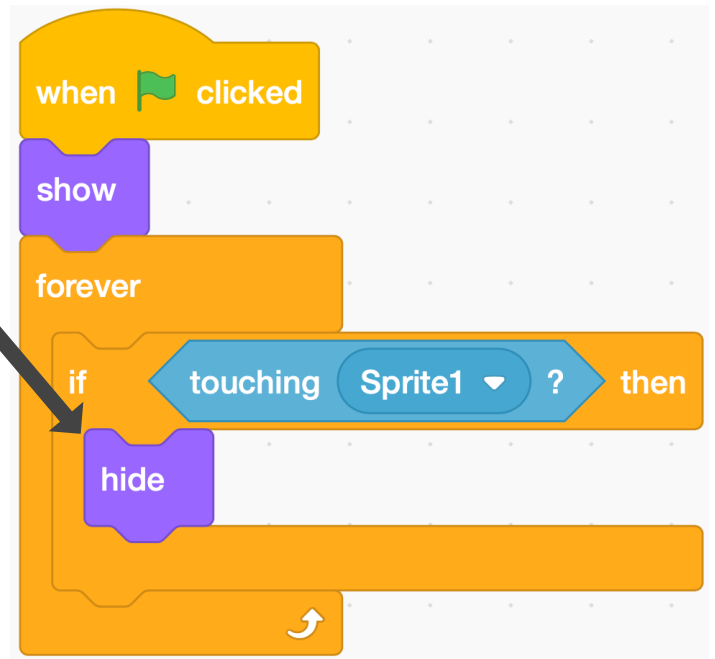
59. Go to the Sensing code draw and drag the “touching?” block into the IF blocks empty input space.

60. We need to change our “touching?” block to match Scratch the cat sprite. Click the white down arrow and select “sprite1” or whatever you have named your sprite.



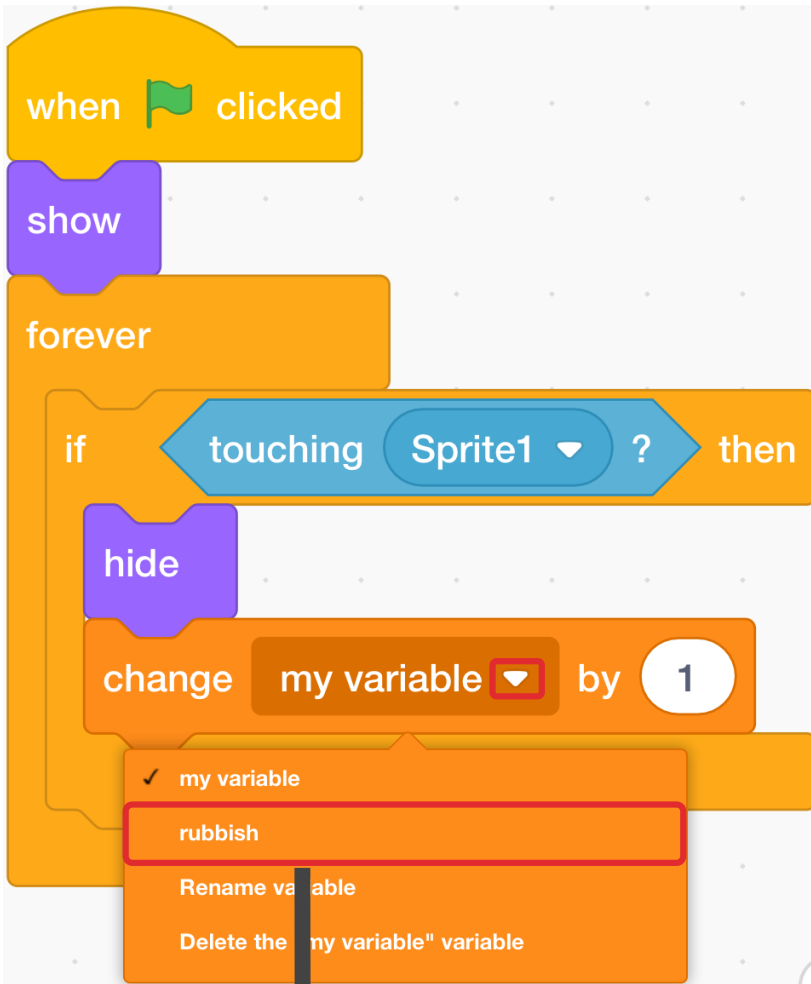


61. Lets add some code that now hides the rubbish after the player has picked it up.

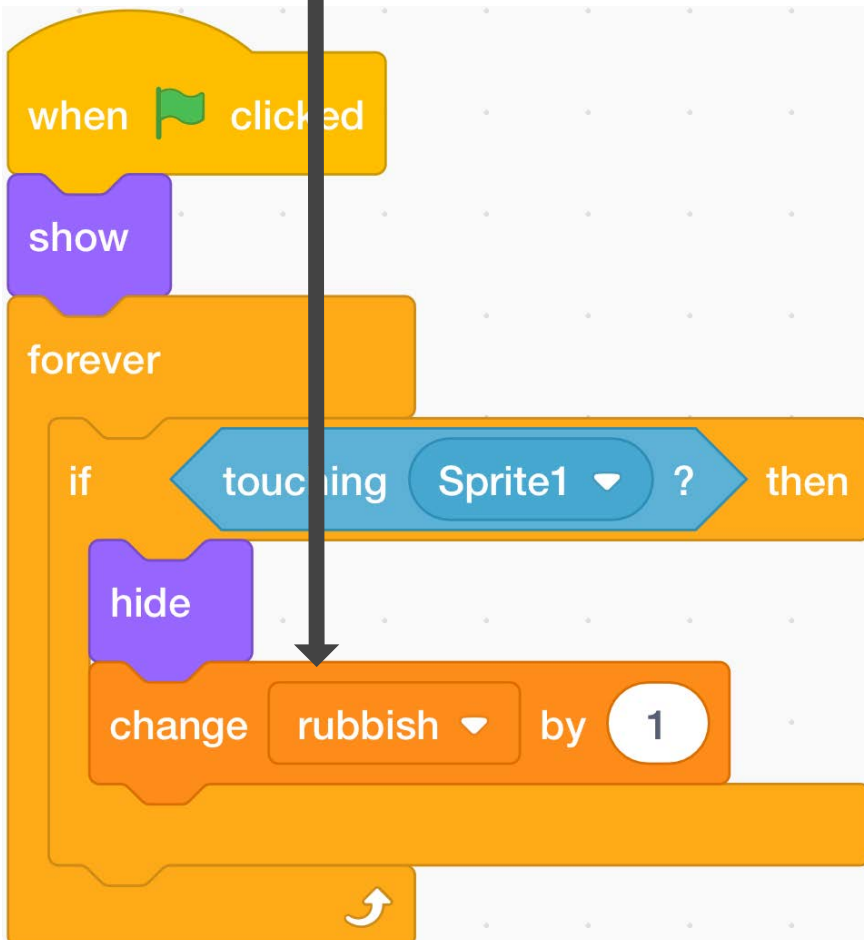


62. Go the Variables code draw and drag the “change variable by 1” under the hide code.





63. Click the white down arrow in the “change variable by 1” code block and select our rubbish variable.



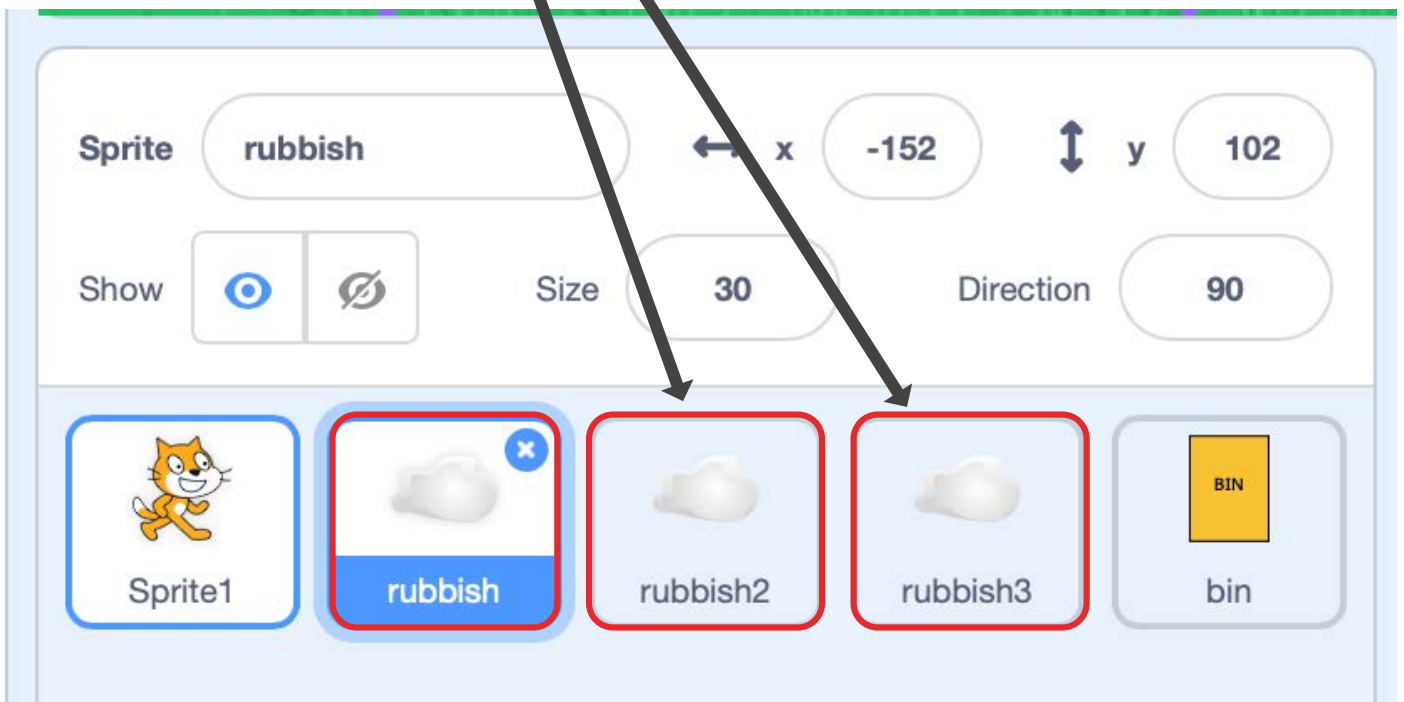
Your completed code for our rubbish sprite should look like this

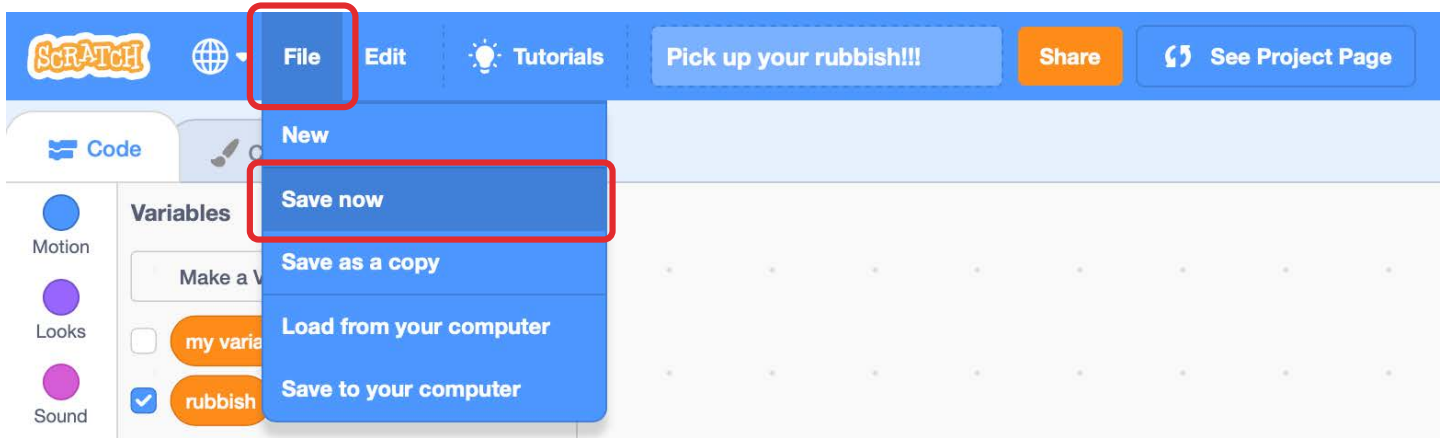


64. We need to copy all our rubbish code to all of our rubbish sprites.

Right click on the top part of our code block and select Duplicate. This will create a copy of all our code.

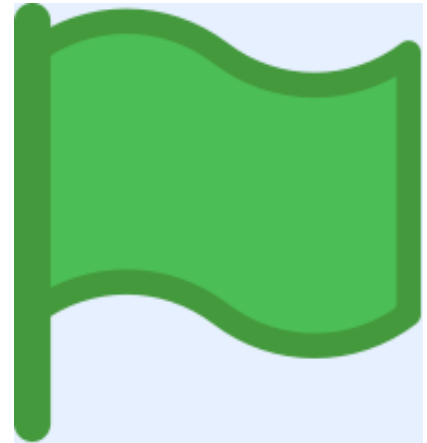
Drag this duplicated code, one at a time, into all of our rubbish sprites.





65. Before we finish our game, we need to make sure we have saved it to our accounts in the cloud. Go to file and select save now!

66. Finally, select the green flag and play the game!



Extra for experts:

1. Create more types of rubbish that give you more points
2. Make super secret levels that only you know how to access

Finished code examples

The image displays two examples of Scratch code blocks with yellow callout boxes explaining their functions.

Example 1 (Top):

- when clicked:** A script starting with a 'when clicked' trigger.
- set rubbish to 0:** Resets the 'rubbish' variable to 0. Callout: "Resets the variable for new games".
- go to x: -199 y: -123:** Moves the sprite to the starting coordinates. Callout: "Resets the sprite to it's starting location".
- forever loop:**
 - point towards mouse-pointer:** Points the sprite towards the mouse. Callout: "Scratch follows the mouse pointer and moves 3 steps forward."
 - move 3 steps:** Moves the sprite forward.
 - if touching color (purple)?:** Checks if the sprite is touching a purple color (representing a wall). Callout: "If Scratch touches the colour of the maze wall then it will go back to the starting coordinates".
 - then go to x: -199 y: -123:** Resets the sprite to the starting coordinates.
 - if touching Bin? and rubbish = 3?:** Checks if the sprite is touching a 'Bin' and if the 'rubbish' variable is 3. Callout: "If the sprite touches the bin and has collected all 3 pieces of rubbish the sprite will congratulate the player, play a sound, and then stop".
 - then:**
 - play sound Meow until done:** Plays a 'Meow' sound.
 - say Ka pai! You are helping to keep our whenua clean and free of rubbish! Tumeke! for 5 seconds:** Displays a congratulatory message for 5 seconds.
 - stop all:** Stops all scripts on the stage.

Example 2 (Bottom):

- when clicked:** A script starting with a 'when clicked' trigger.
- show:** Shows the sprite. Callout: "Makes the sprite viable when the game starts again".
- forever loop:**
 - if touching Sprite1?:** Checks if the sprite is touching 'Sprite1'. Callout: "When the sprite touches the player, hide, and then add one to the rubbish variable".
 - then:**
 - hide:** Hides the sprite.
 - change rubbish by 1:** Increments the 'rubbish' variable by 1.

Additional Callouts:

- "Need to change this number to match the number of rubbish pieces." (points to the '3' in the 'rubbish = 3' block).
- "All rubbish sprites need this code" (points to the 'if touching Sprite1?' block).