

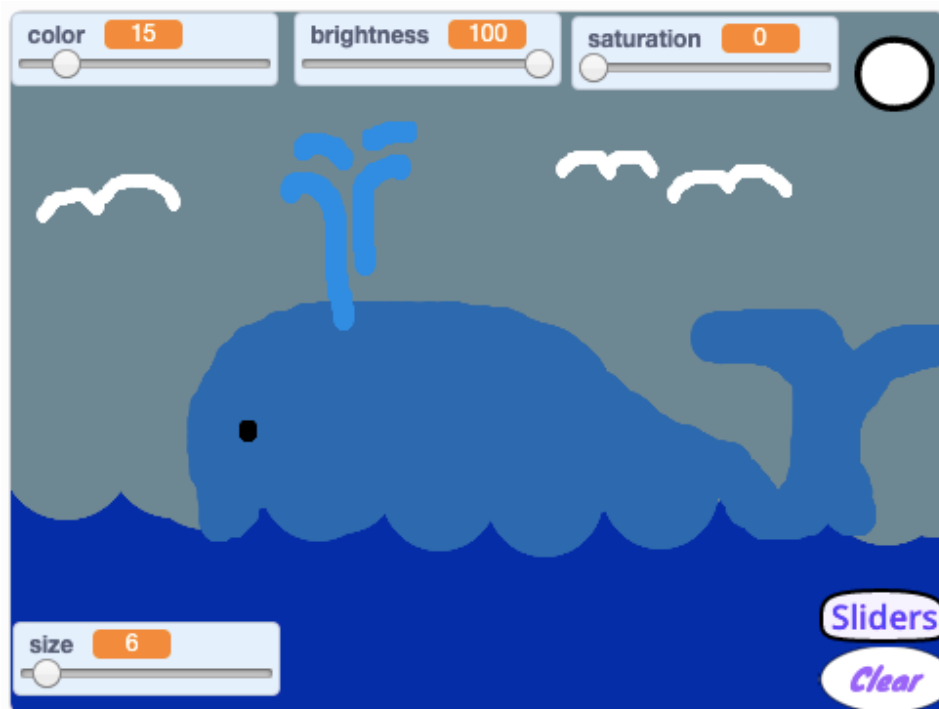


GameMaker Tutorial

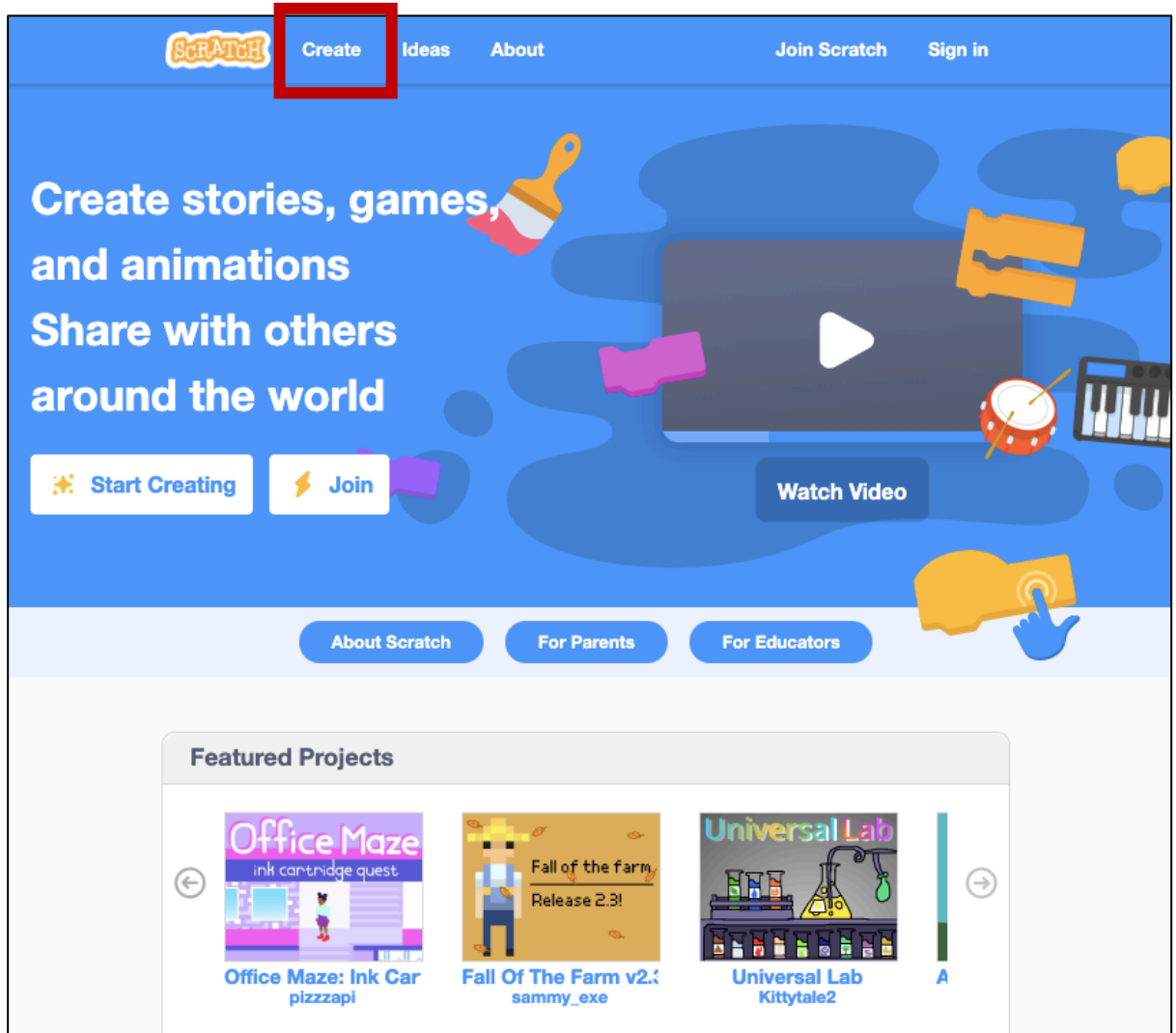
How to Make a Paint Game in Scratch

Written by Rebekah Ilave
October 2019

Source: Learn Learn Scratch Tutorials
<https://www.youtube.com/watch?v=9J44i3Xh2HQ>



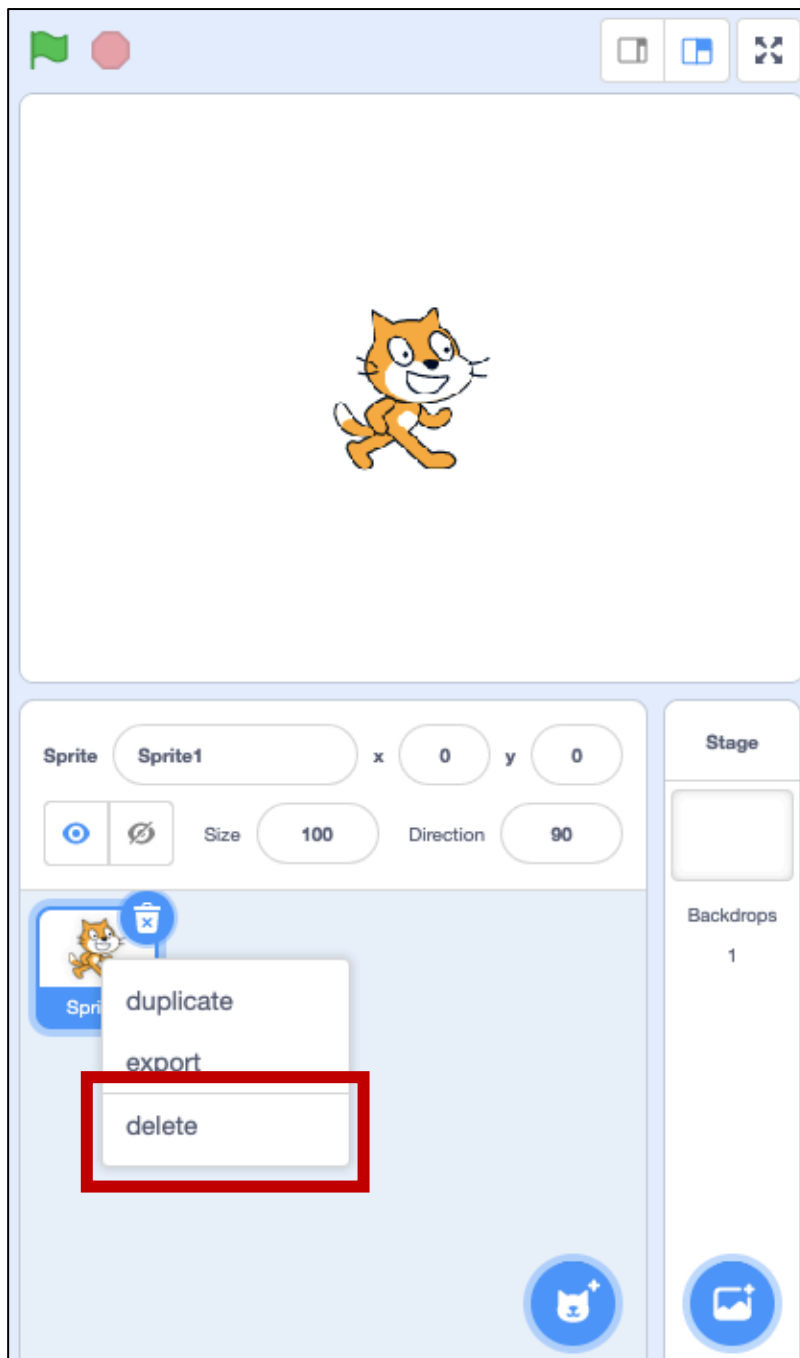
1. Create a new Scratch project. Open Scratch > Click Create.



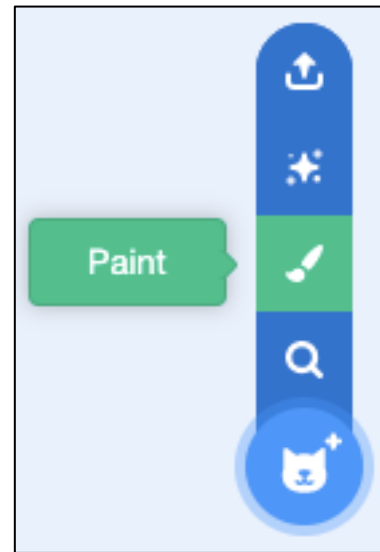
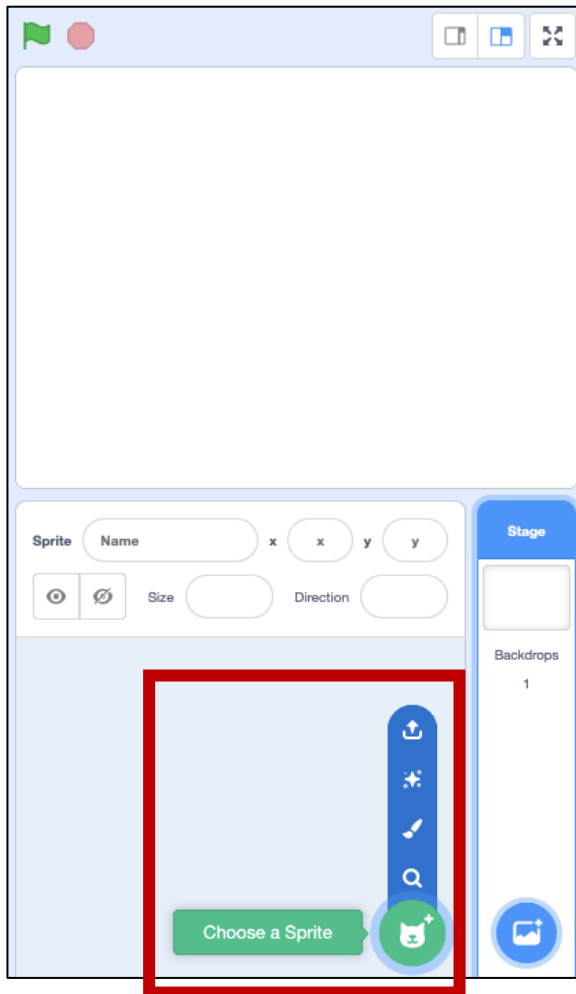
2. We don't need Scratchy the Cat today, so right-click on Scratchy > delete Scratchy the sprite.

Remember: Sprites

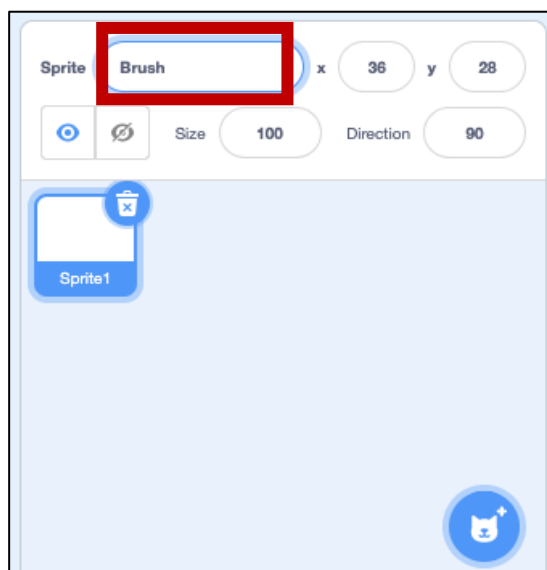
Sprites are images on a Scratch computer program screen. Every Scratch program is made up of sprites and the scripts (instructions) that control them. Scripts are programmed to make the sprites do things. Sprites can be made to move around, change their appearance, react when they touch things, and be controlled by the game player.



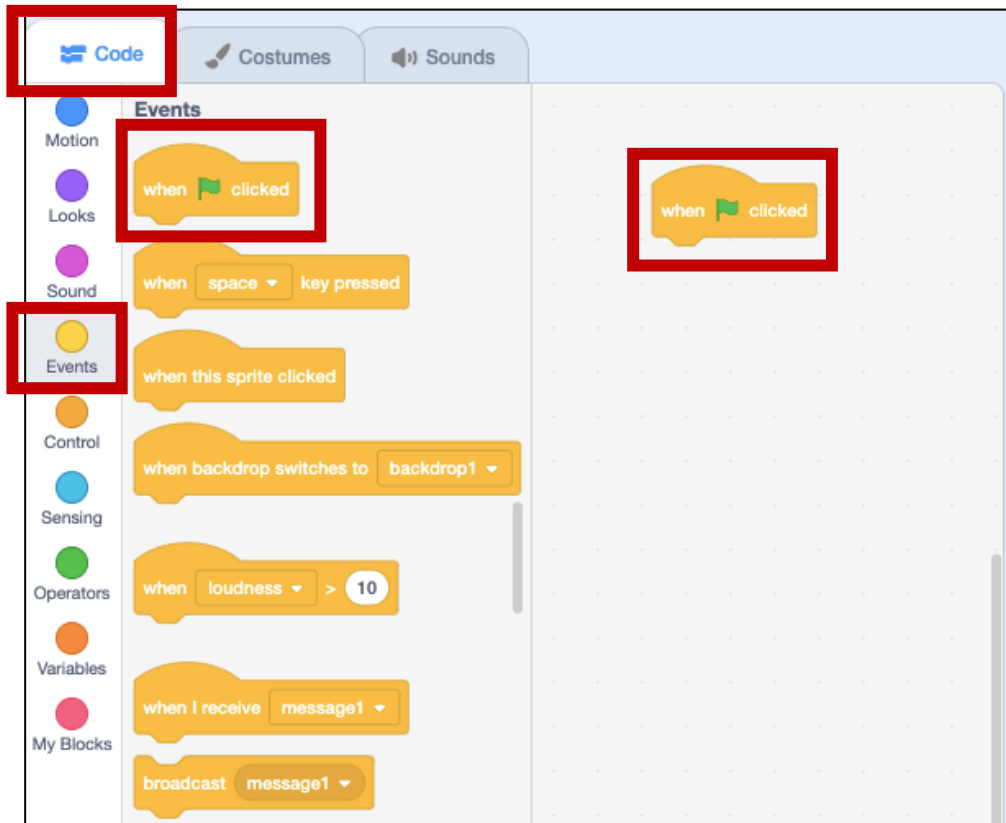
3. We'll make a new sprite, the paintbrush. So hover over the *Choose a Sprite* button, then click *Paint*.



4. Rename the Sprite to 'Brush'. The brush will be invisible, just like in other digital paint games.



5. Now we're going to let our invisible Brush sprite move around the game when using the mouse, and the sprite will paint a colour when the mouse button is pressed down. Click on the *Code* tab. Click the yellow *Events* button, and click and drag "When green flag is clicked" block to the panel.



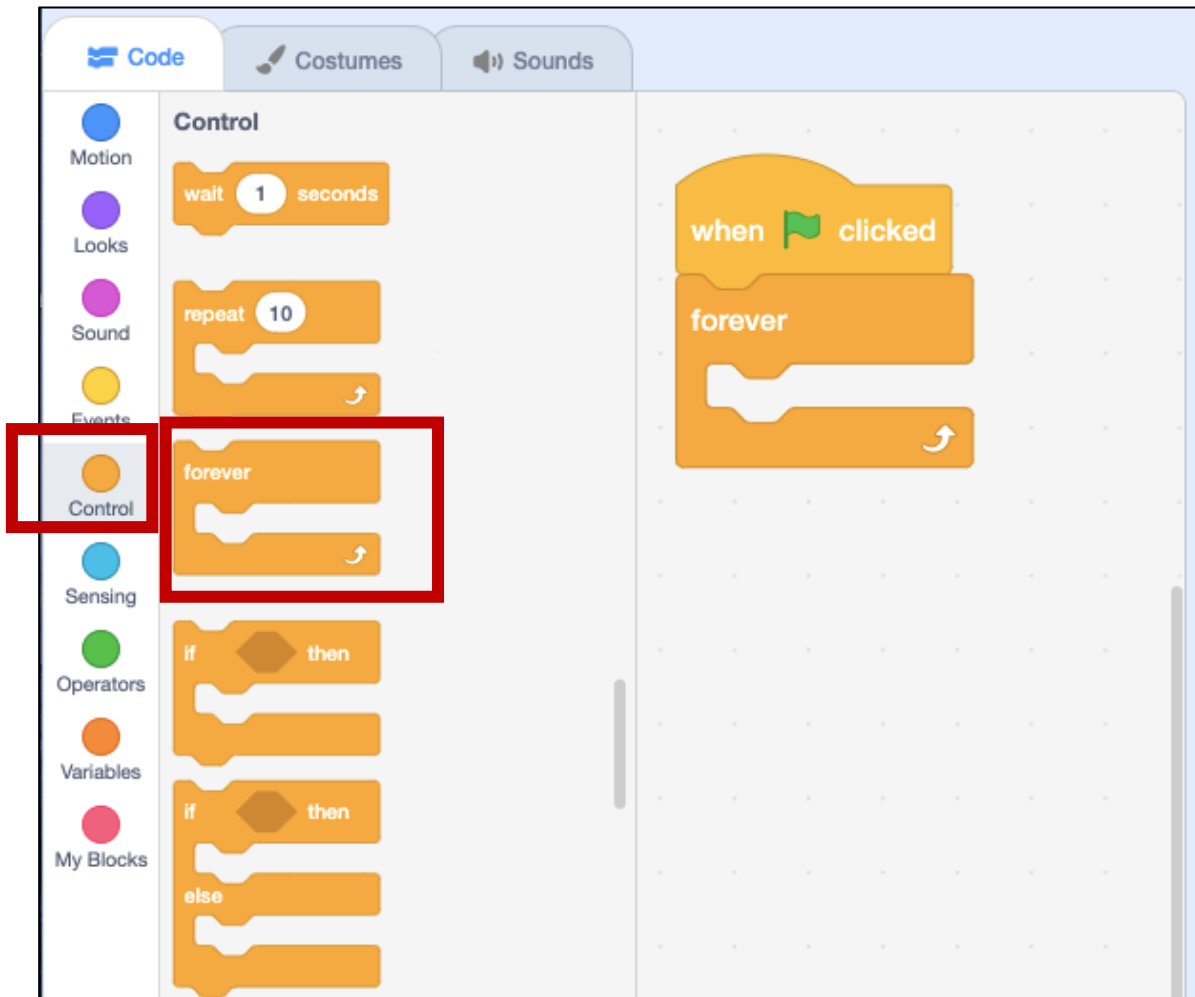
Remember: Blocks

Blocks are puzzle-piece shapes that are used to create code in Scratch. The blocks connect to each other like a jigsaw puzzle. Each block has its own shape and specially shaped slots. Connected blocks are called scripts. There are 9 kinds of blocks you can see the Scratch screen: Motion, Looks, Sound, List, Event, Control, Sensing, Operators, Variables and My Blocks.

Examples of blocks:



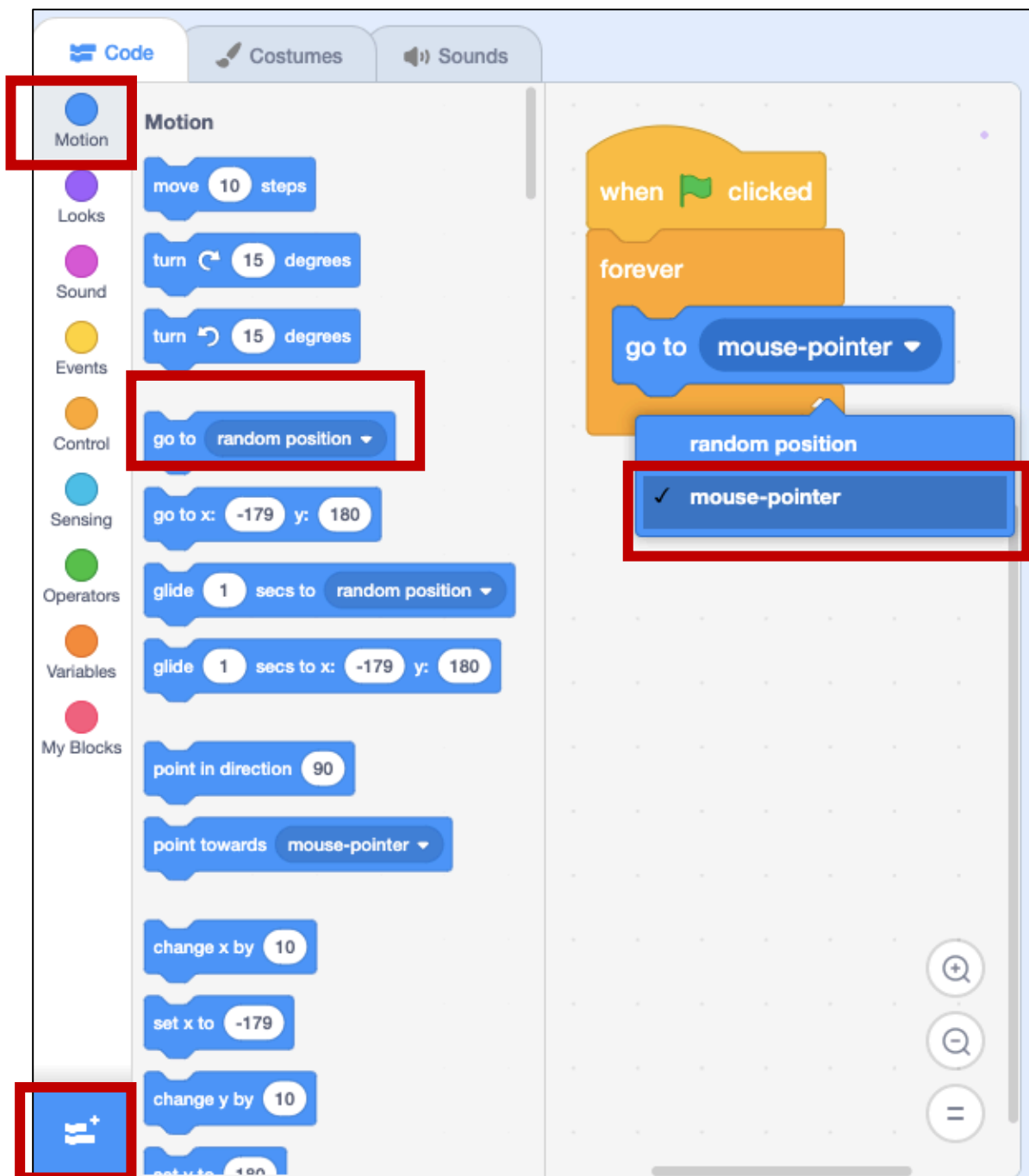
6. We want the Brush sprite to keep moving whenever the mouse moves, so we are going to use a forever loop. Click the Control button, then click and drag the “forever” block to the bottom of the block to the bottom of the “when green flag clicked” block.



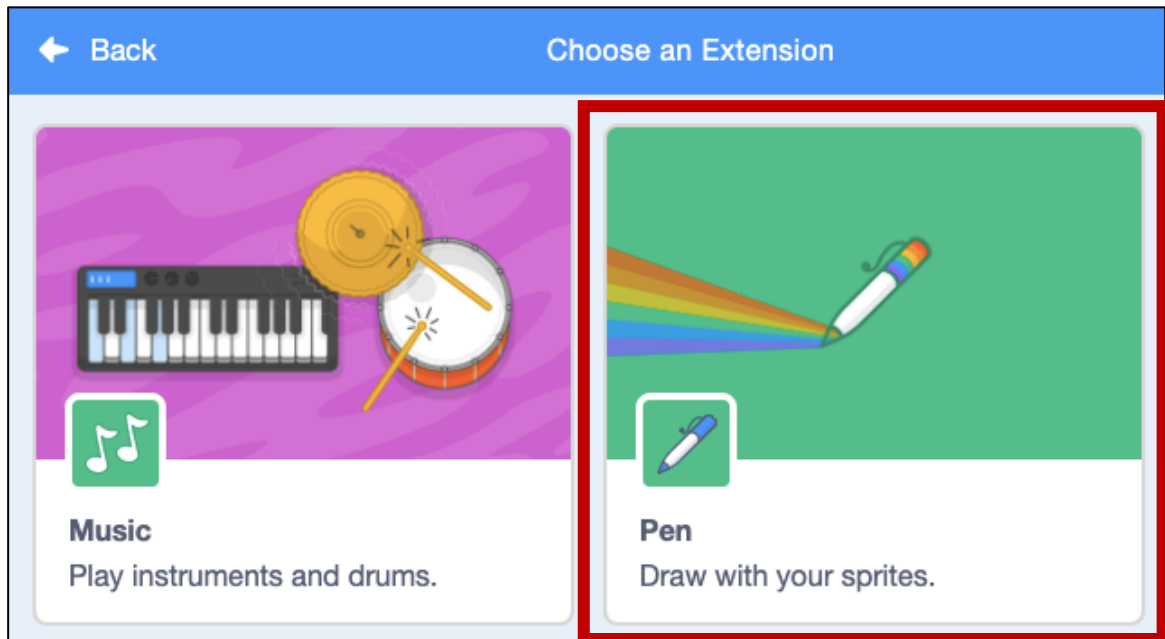
7. To make the Brush sprite move, we need Motion. Click the Motion button, then click and drag the “go to random position” block and put it inside the “forever” block. On the “go to random position” block, click and change *random position* to *mouse-pointer*.

Now we need to add an extension for this Paint game. Click the Extensions button in the bottom left corner of the screen:

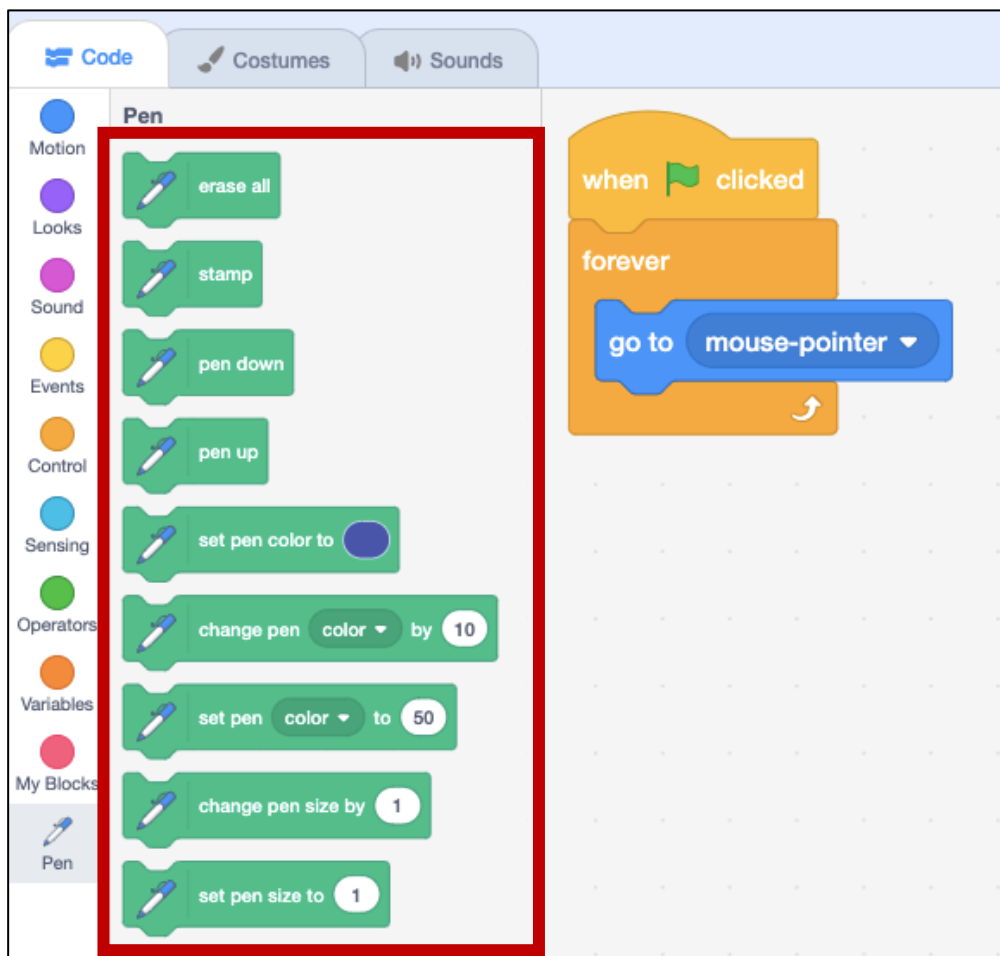
Extensions button:



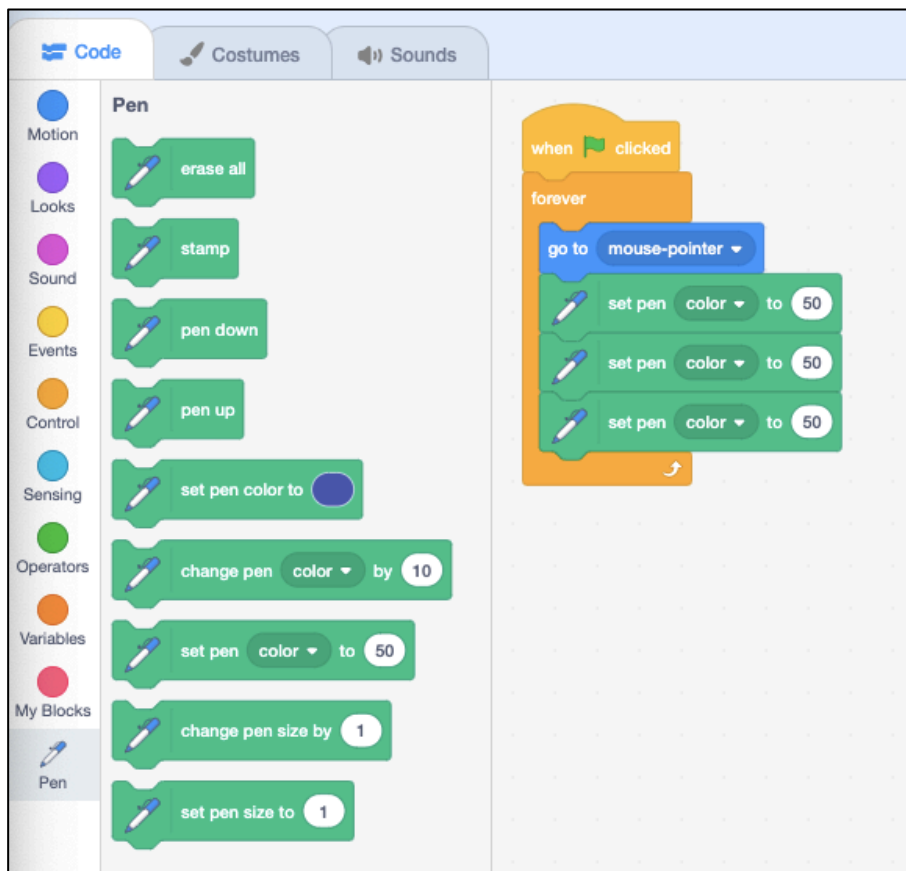
8. Choose the Pen Extension by clicking on it.



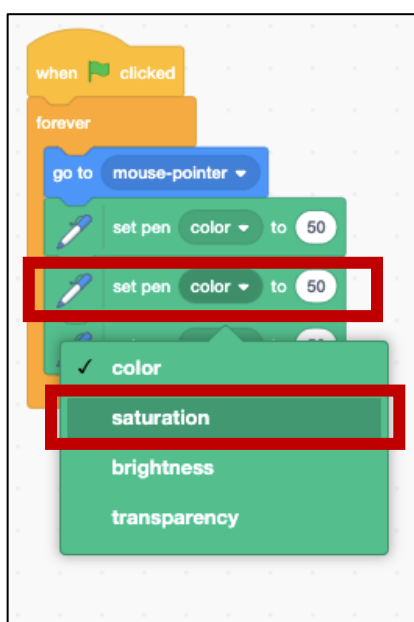
After you click Pen Extension, green blocks will appear for Pen in your main screen.



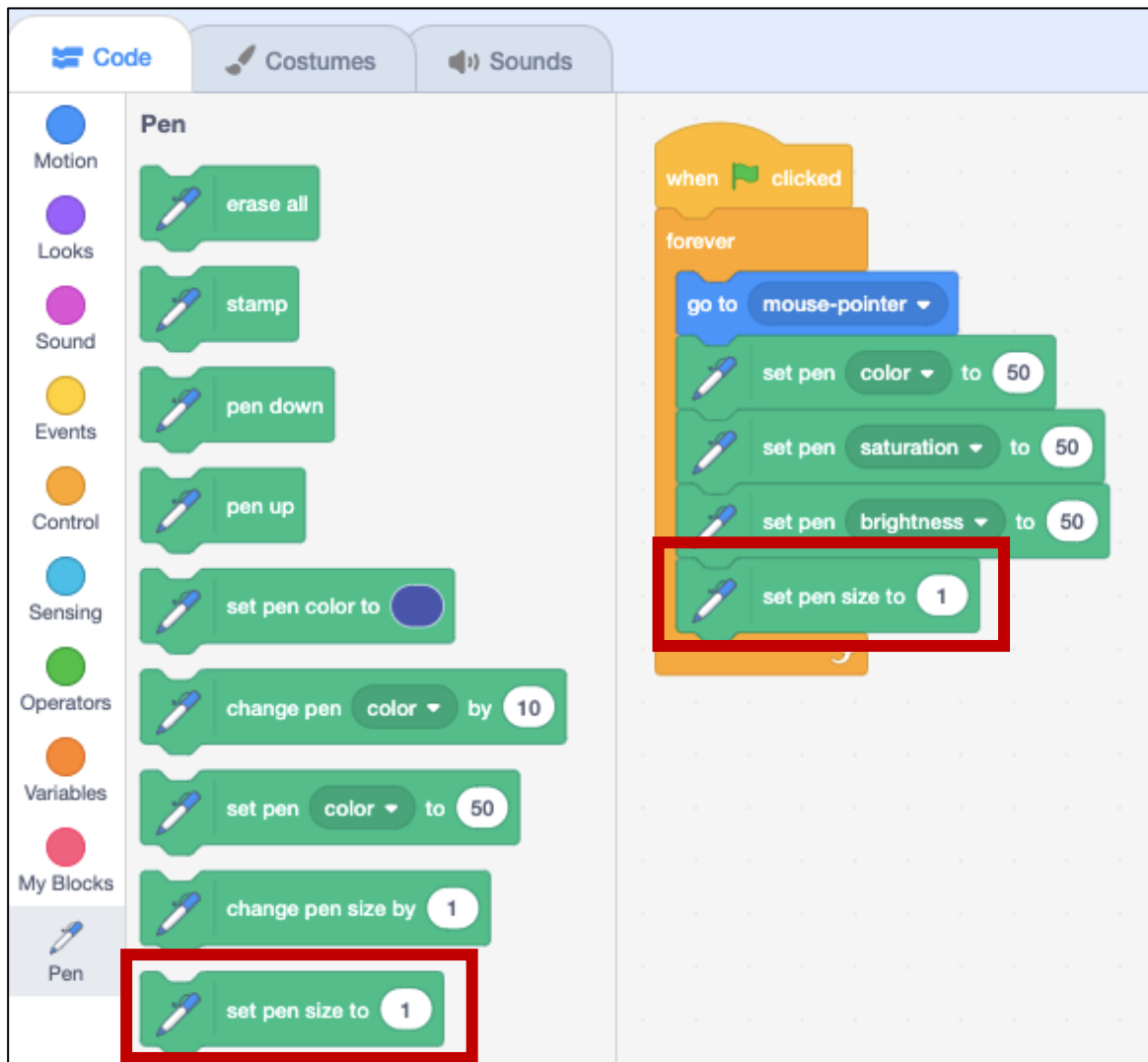
9. Now we're ready to add painting features for the Brush! Click and drag the "set pen color to ___" block into the "forever" block, under the "go to mouse-pointer" block. Repeat this 2 times, to have 3 "set pen color to ___" blocks.



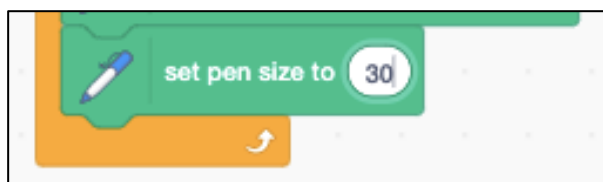
Change *color* the second "set pen color to ___" block to *saturation*.
Change *color* on the third "set pen color to ___" block to *brightness*.



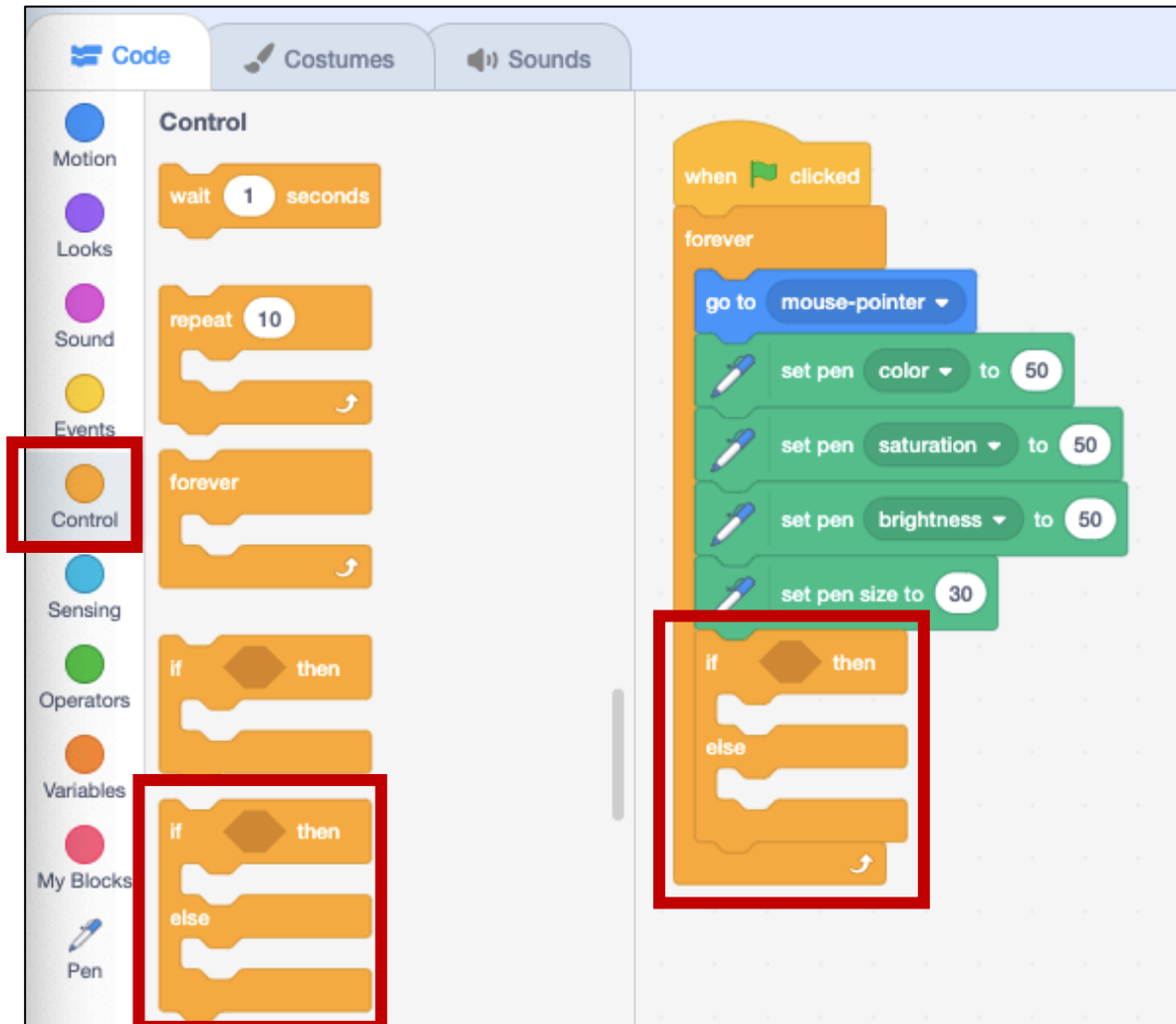
10. Click and drag the “set pen size to ___” block to the bottom of the “set pen brightness to ___”. Make sure it’s inside the “forever” block!



Change 1 in the “set pen size to ___” block to 30, so that the paint brush starting size will be big.



11. Now we're going to code the part that draws using the Brush **if** the mouse button is pressed down, so we are going to use an "if" block. Click the Control button, then click and drag the "if ___ then ___ else" block into the "forever block", under the "set pen size to 30" block.



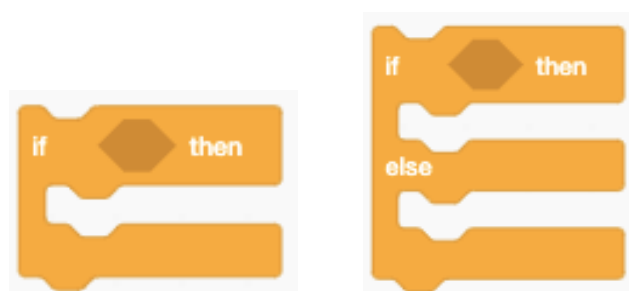
Remember: "If" blocks

"If" blocks are used many times in Scratch games or other Scratch programs.

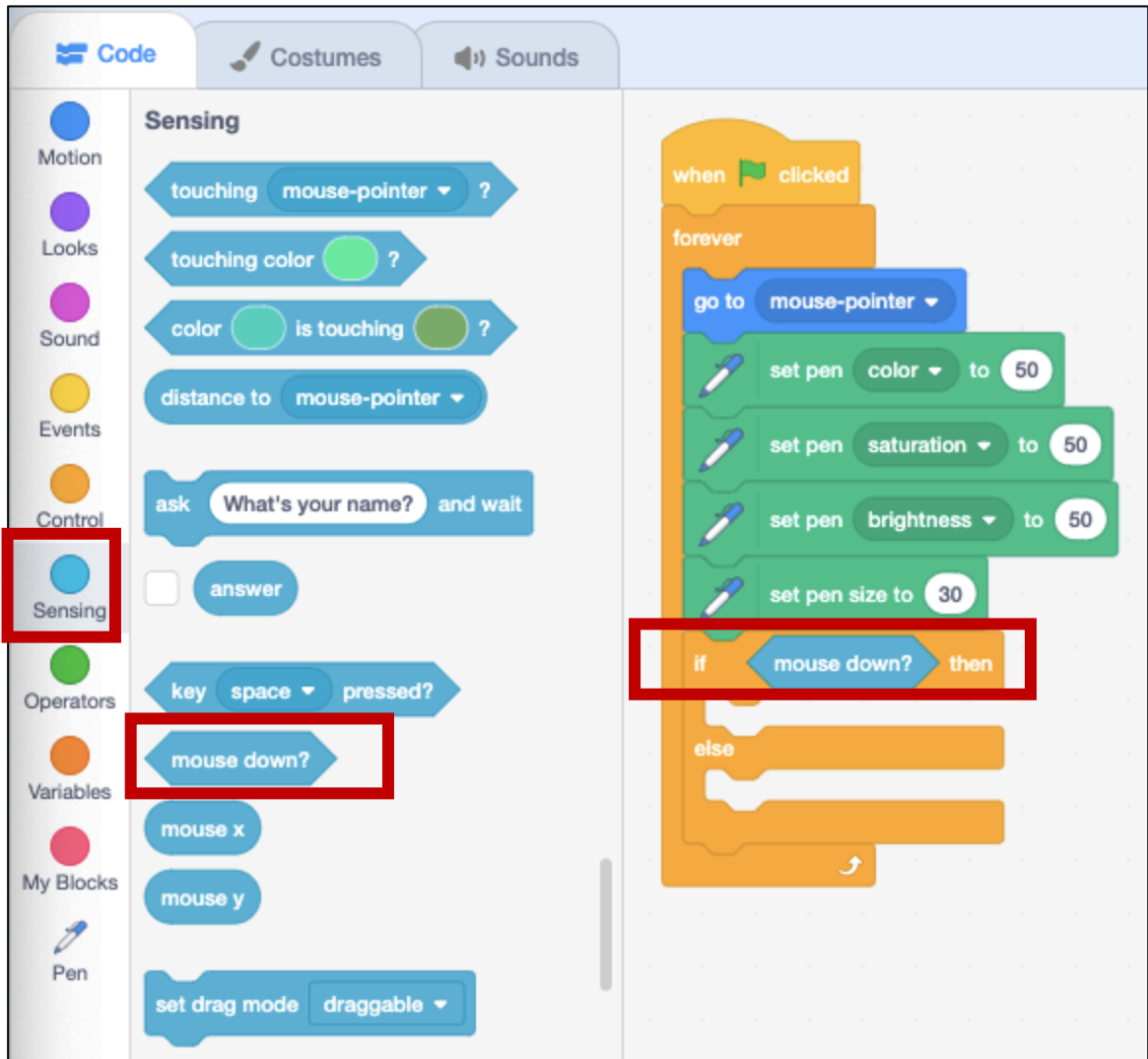
"If" blocks check to see if a condition is true or false.

If the condition is true, then the "if" script will run.

If the condition is false, then the "if" script will not run.



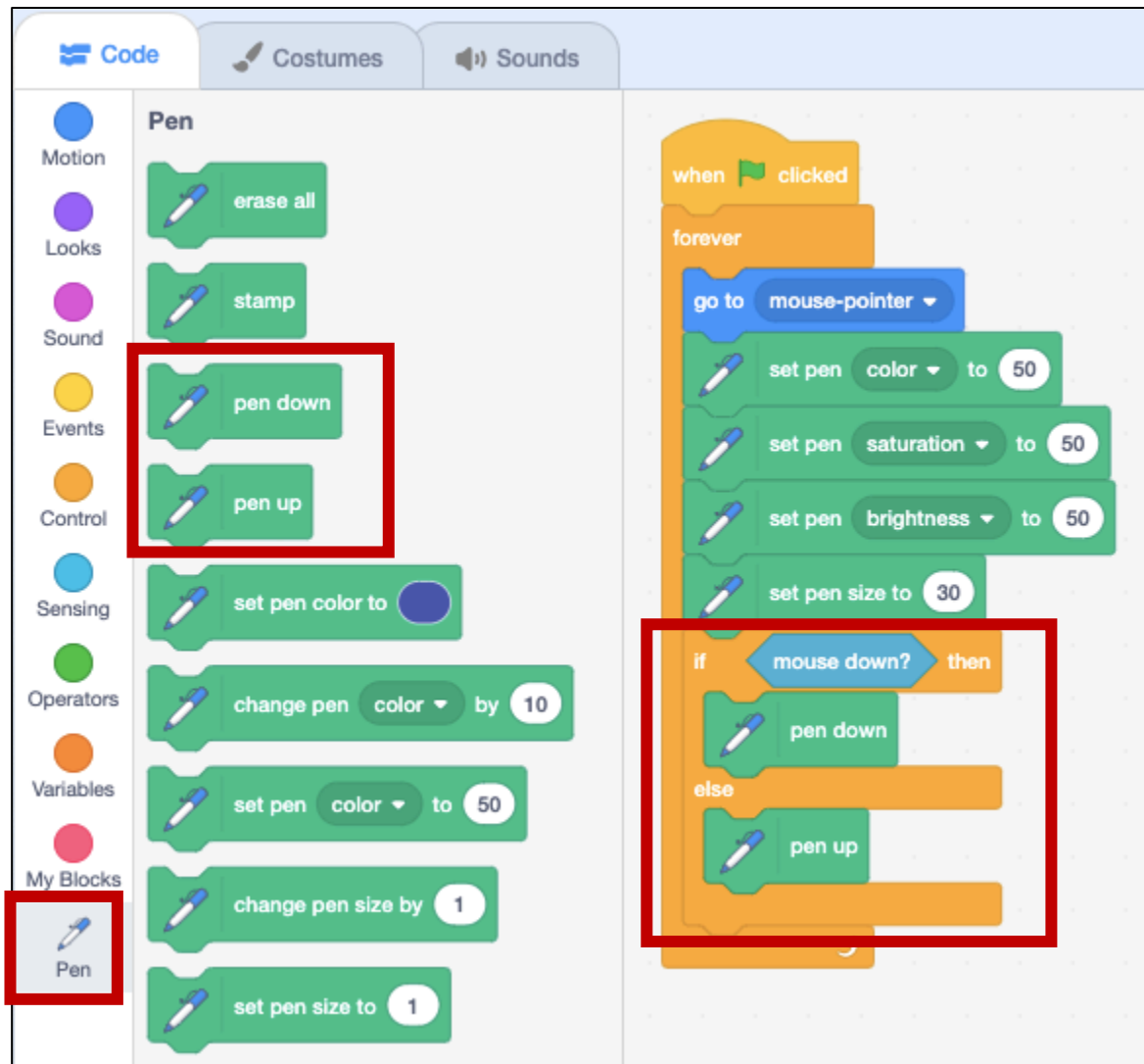
12. Click the Sensing button, then click and drag the “mouse down” block into the “If ____ then” block.



13. Now we're ready to do our condition to run the script that says:
if the mouse is pressed down, then the brush is pressed down, or else the brush is up.

Click the Pen button, then click and drag the "pen down" block into the first slot in the "if mouse down? then ___ else ___" block.

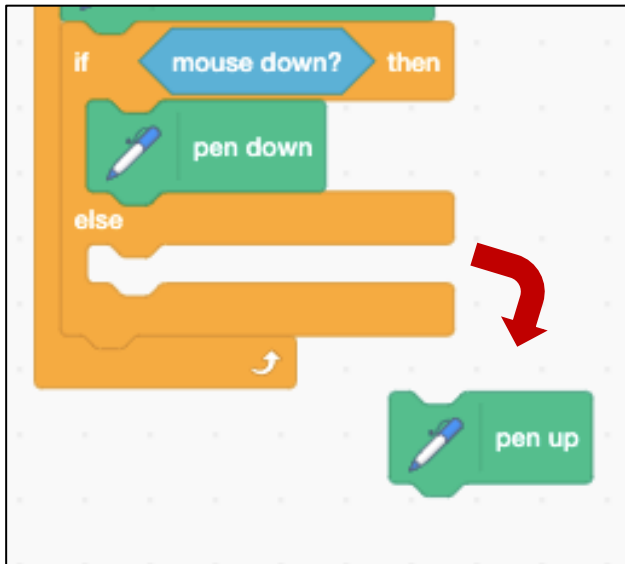
Click and drag the "pen up" block into the second slot in the "if mouse down? then pen down else ___" block.



14. What will happen if we don't put a "pen up" block into the "else ____" space?

Try it now!

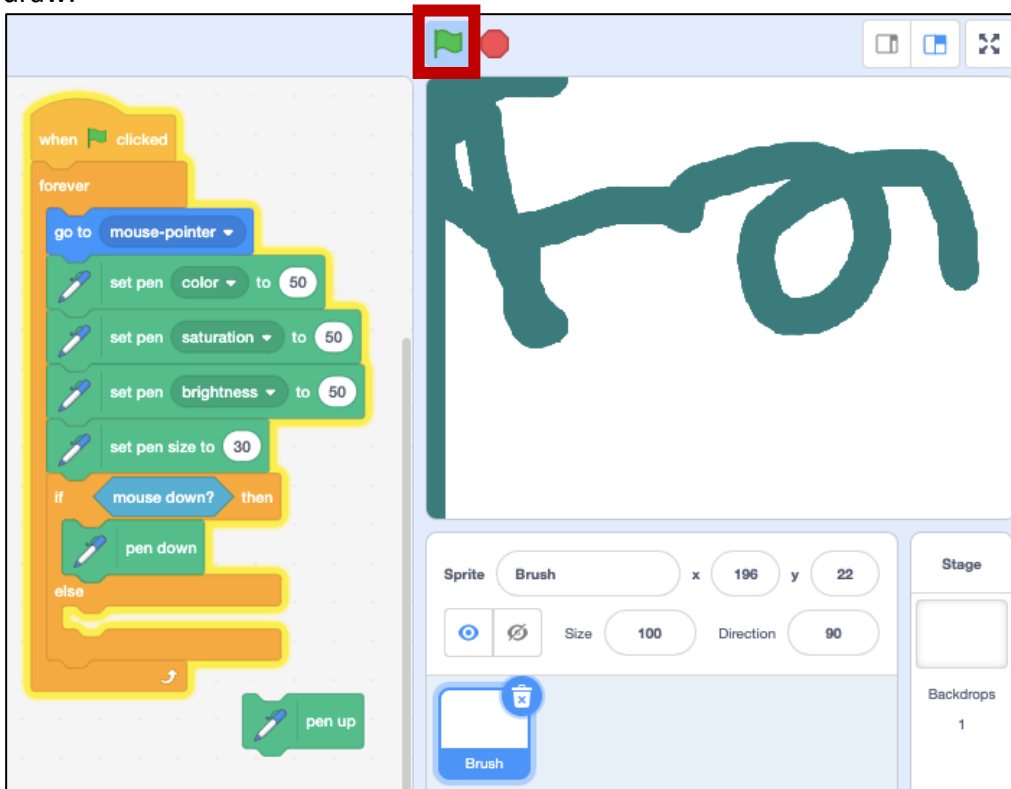
Take the "pen up" block out of the "if mouse down then pen down else pen up" block.



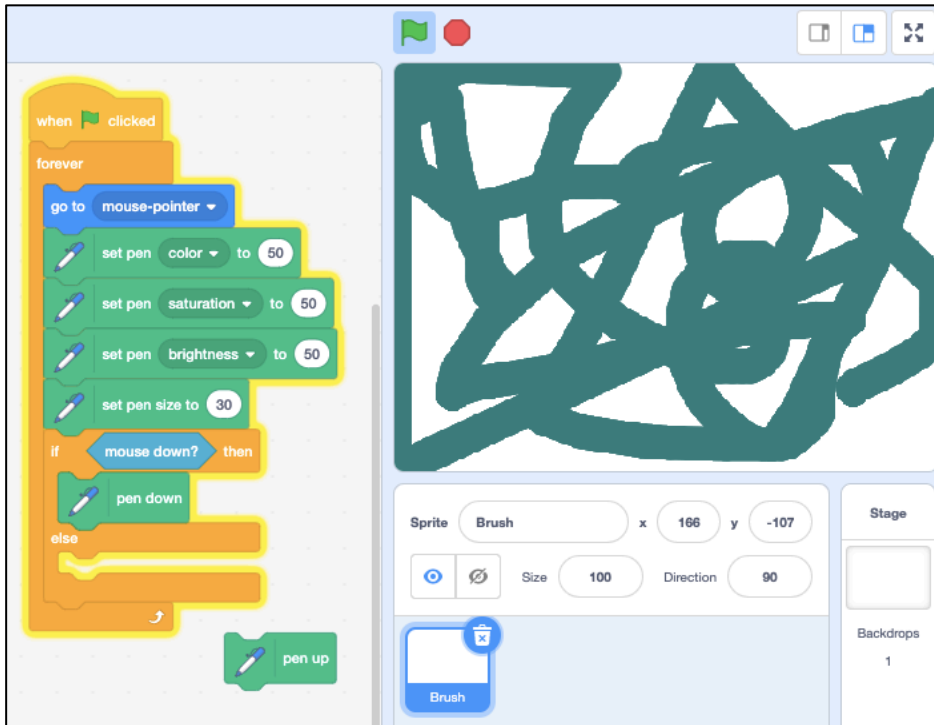
Now click the green flag to run your program:



Press your mouse down to start to draw using the invisible Brush. It should start to draw.



But what happens when you stop pressing on the mouse? It still draws! (oh no!)

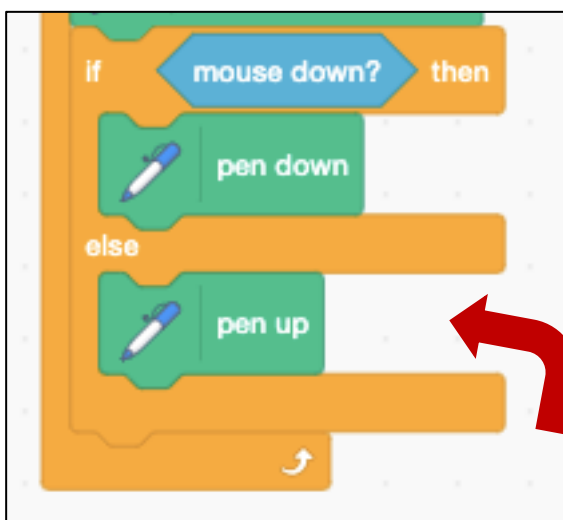


So that's why we need the "pen up" block in the "else____" space!

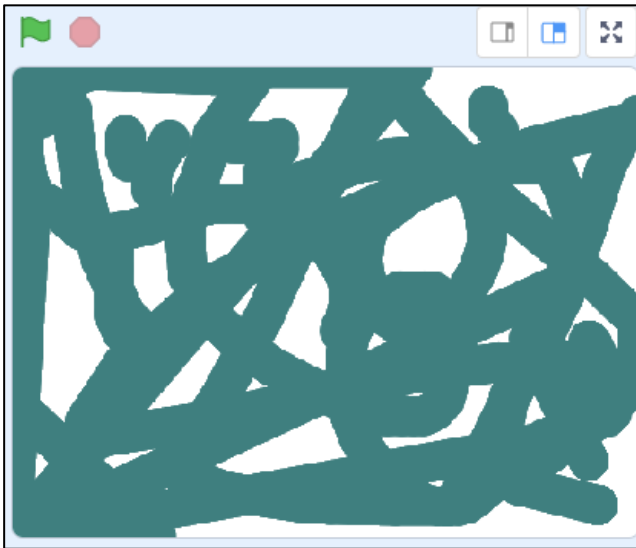


Press the red stop button:

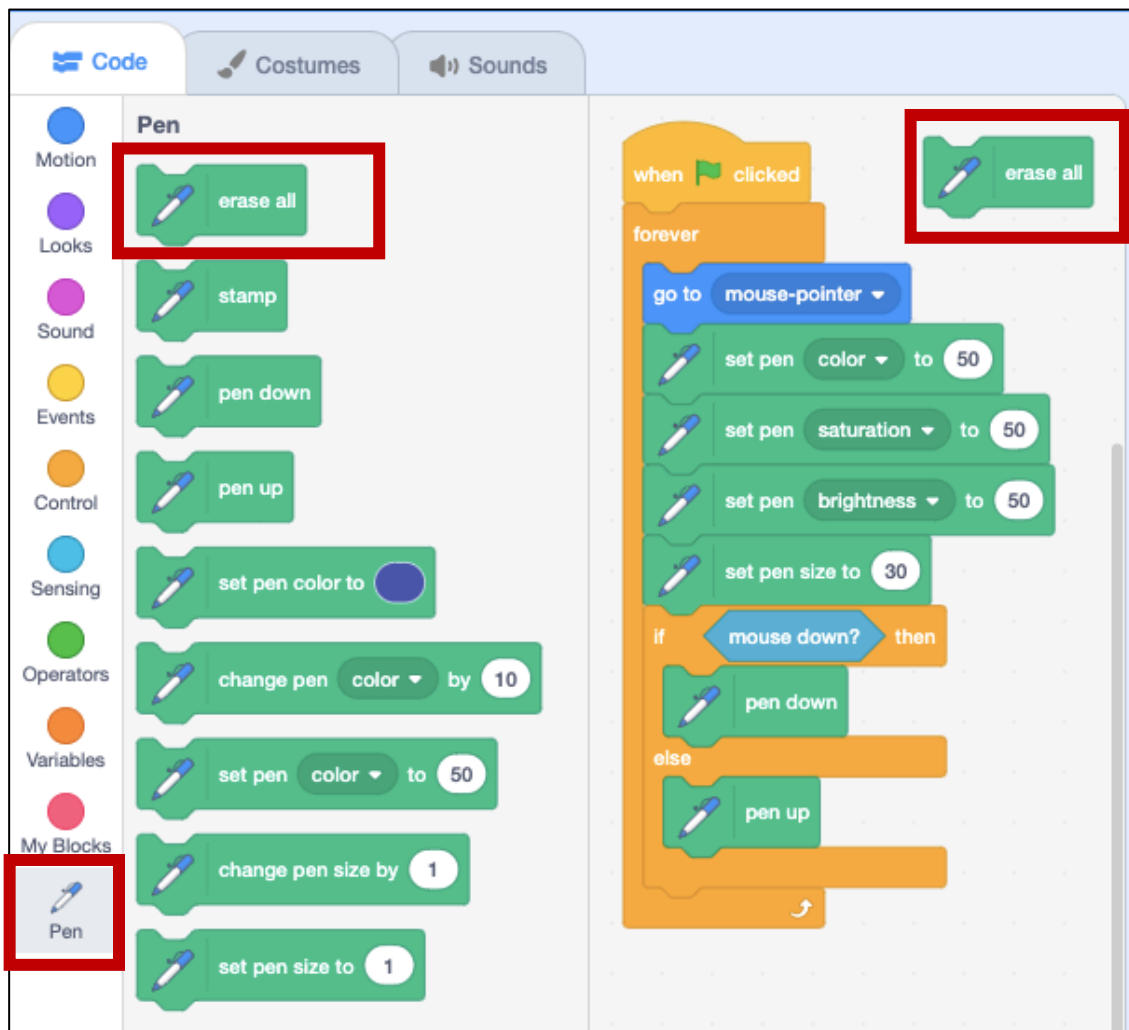
and put the "pen up" block back where it belongs:



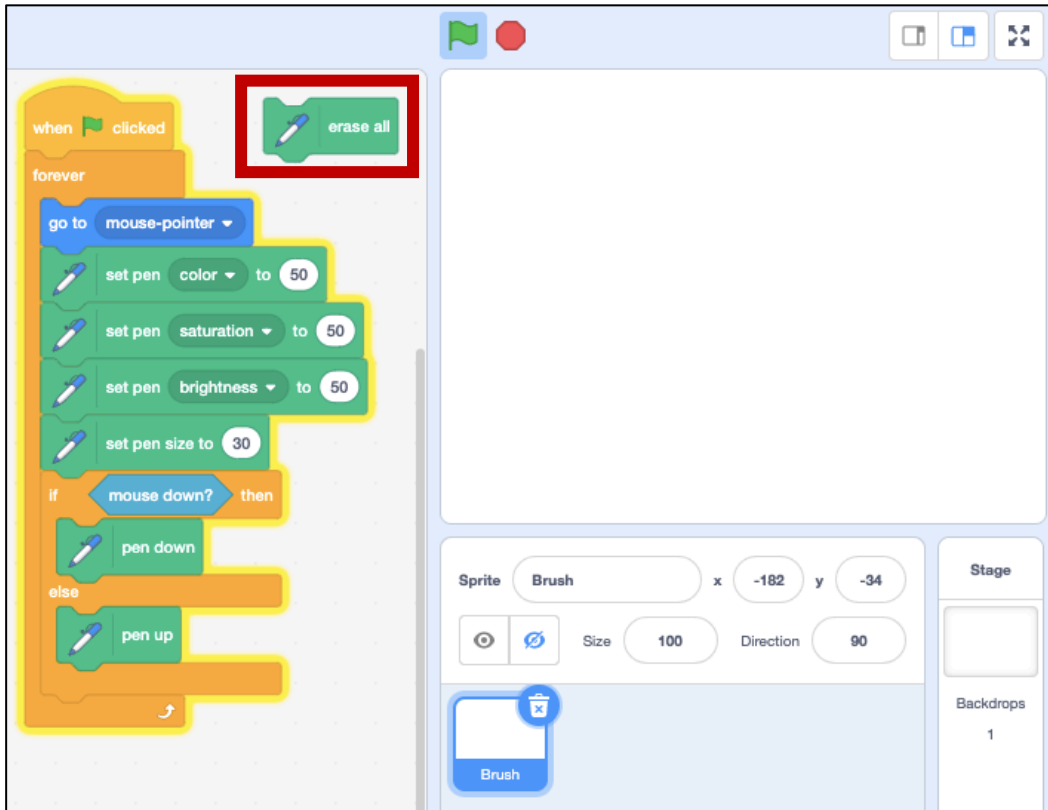
15. Now our game screen is covered in paint, so we need to find a way to erase it all.



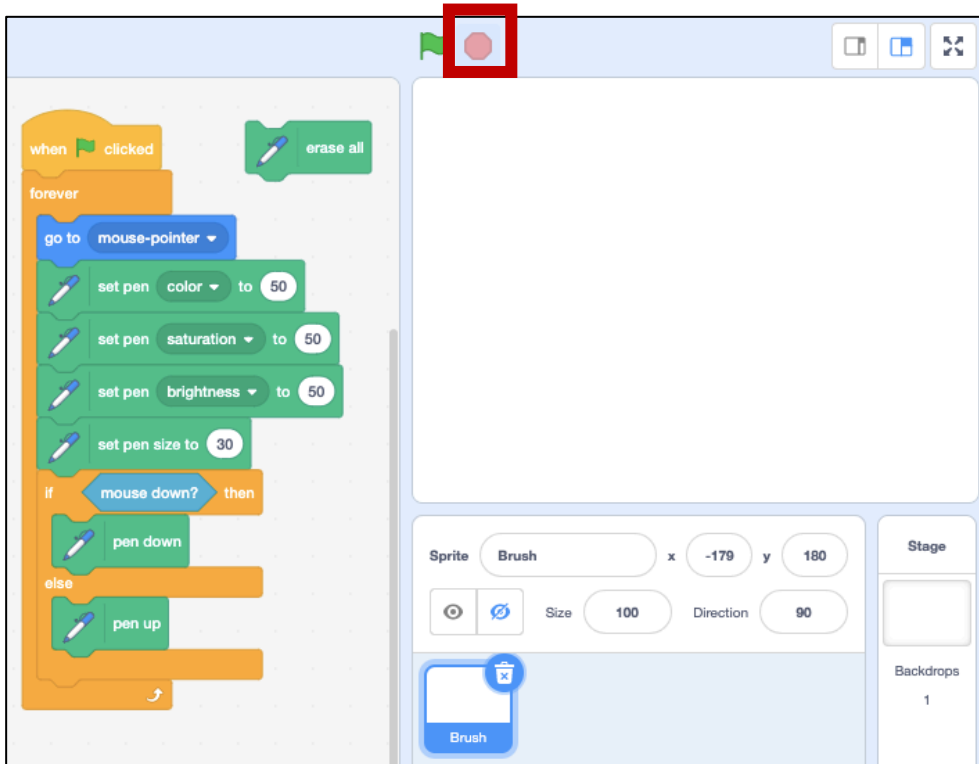
Click the Pen button, then click and drag the “erase all” block and put it by itself on the side.



Now whenever you want to erase your test drawings, you can click “erase all” and all the paint will disappear! Try it now.

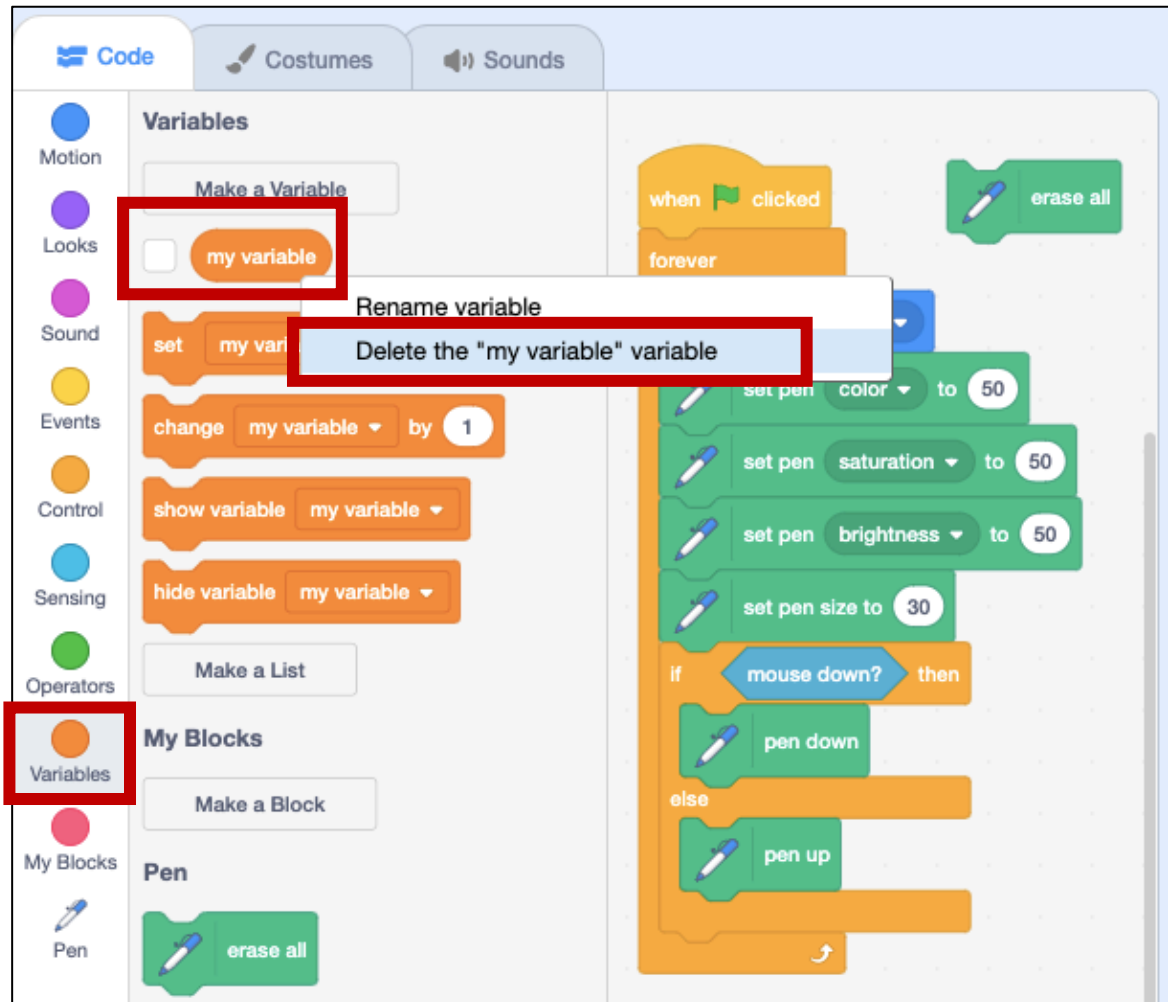


Click the red stop button to stop running the code:

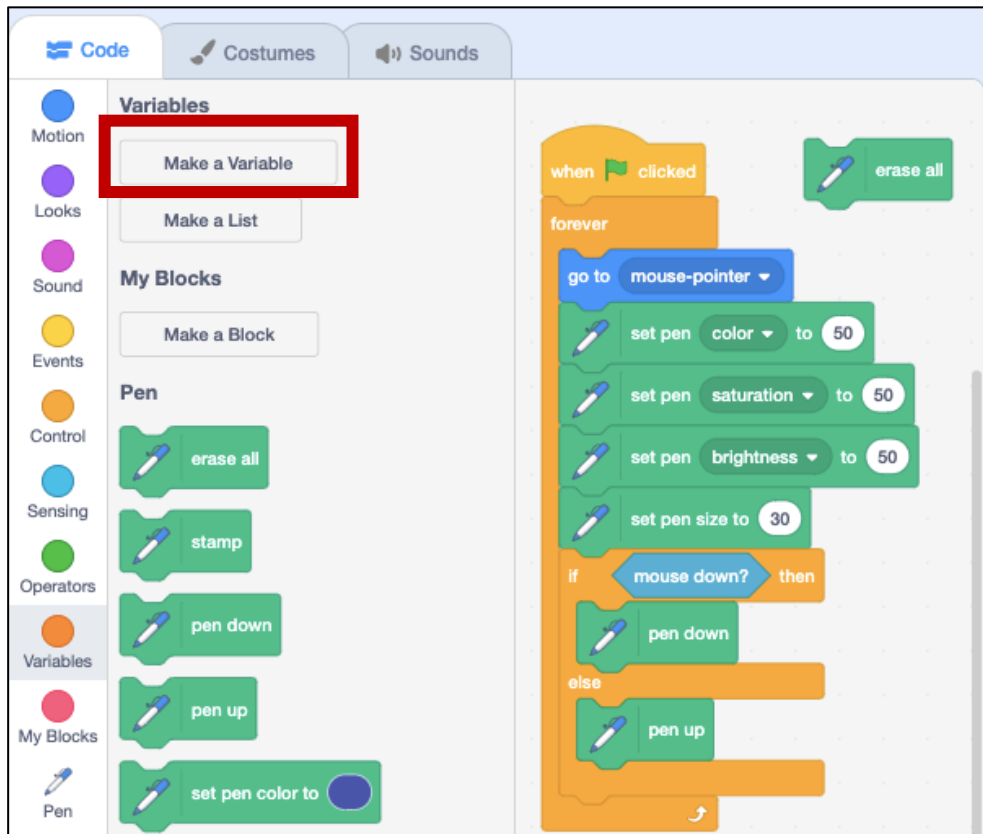


16. Now we're going to create sliders in the game to change the color, saturation and brightness of the paints. Because these sliders allow us to **change (vary)** the paints, we need to use Variables.

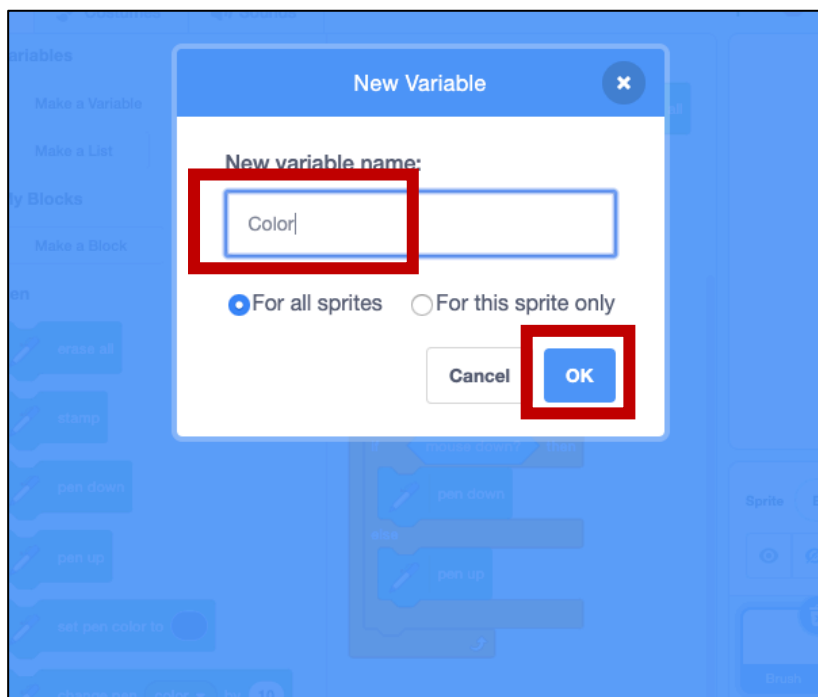
Click the Variables button. Right-click the "my variable" block and select *Delete the "my variable" variable*. Don't worry, we'll create some more variables.



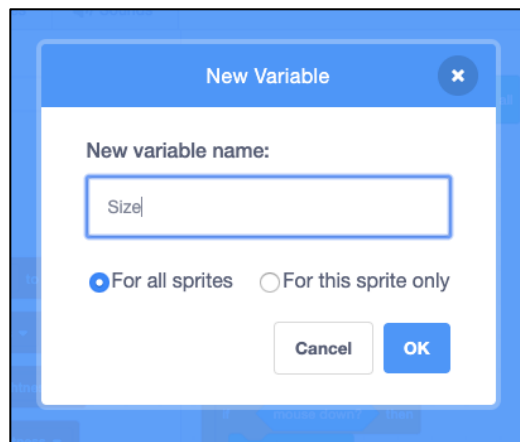
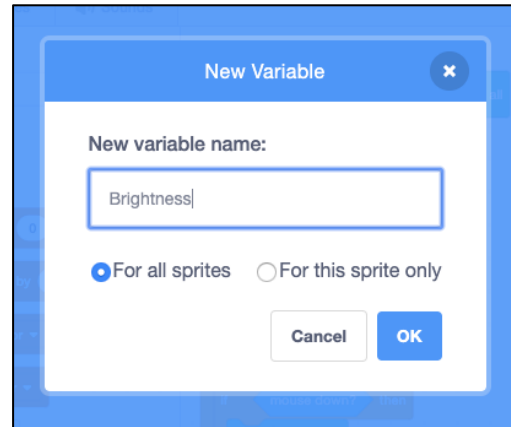
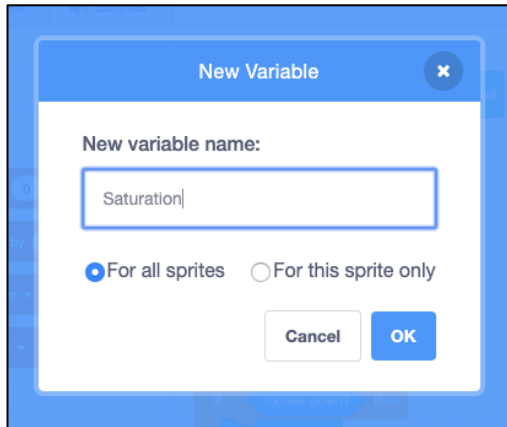
17. Click the *Make a Variable* button.



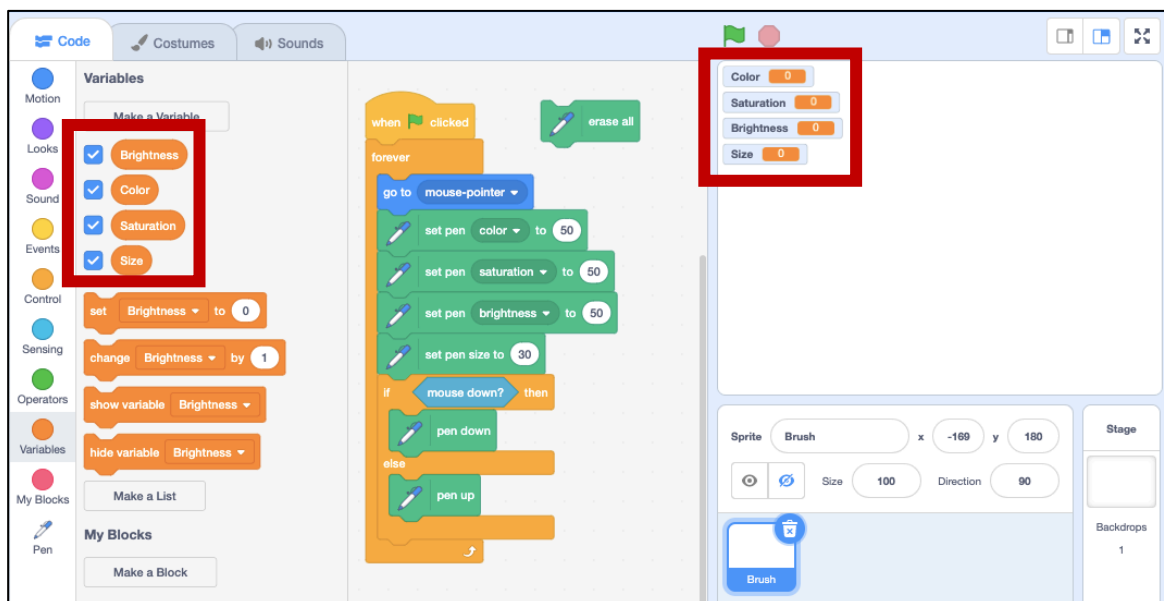
Your screen will go blue, and ask you for a **New variable name**. Type the name of this variable as *Color* (this is American spelling, you can use British/Australia spelling as *Colour* if you like). Click OK.



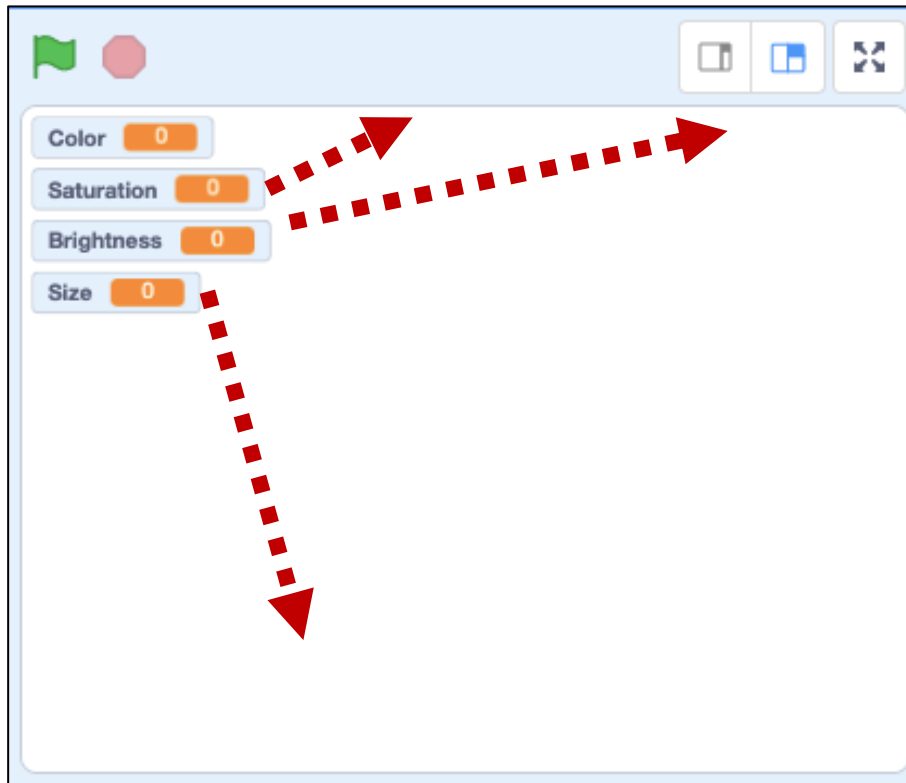
18. Repeat Step 17 to create 3 more variables: *Saturation*, *Brightness* and *Size*.



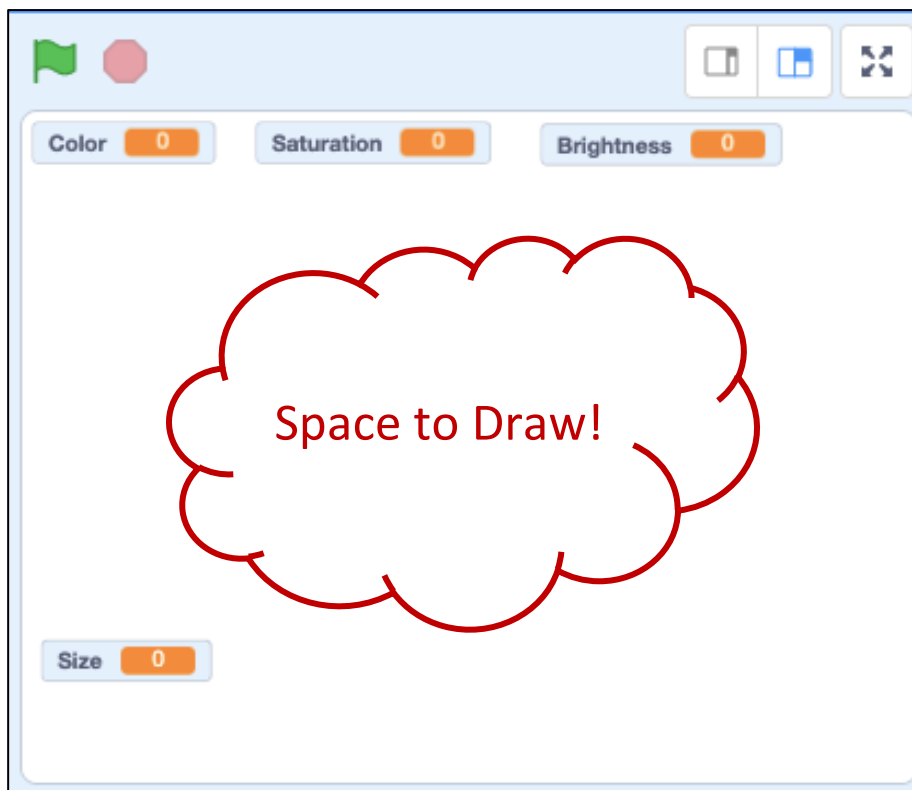
You should now have 4 new variables: *Color*, *Saturation*, *Brightness* and *Size*.
You will see these variables also appear on your game screen:



19. Now we can start to make sliders for these variables. On the game screen, click and drag the variables around to the sides of the screen to space them out.



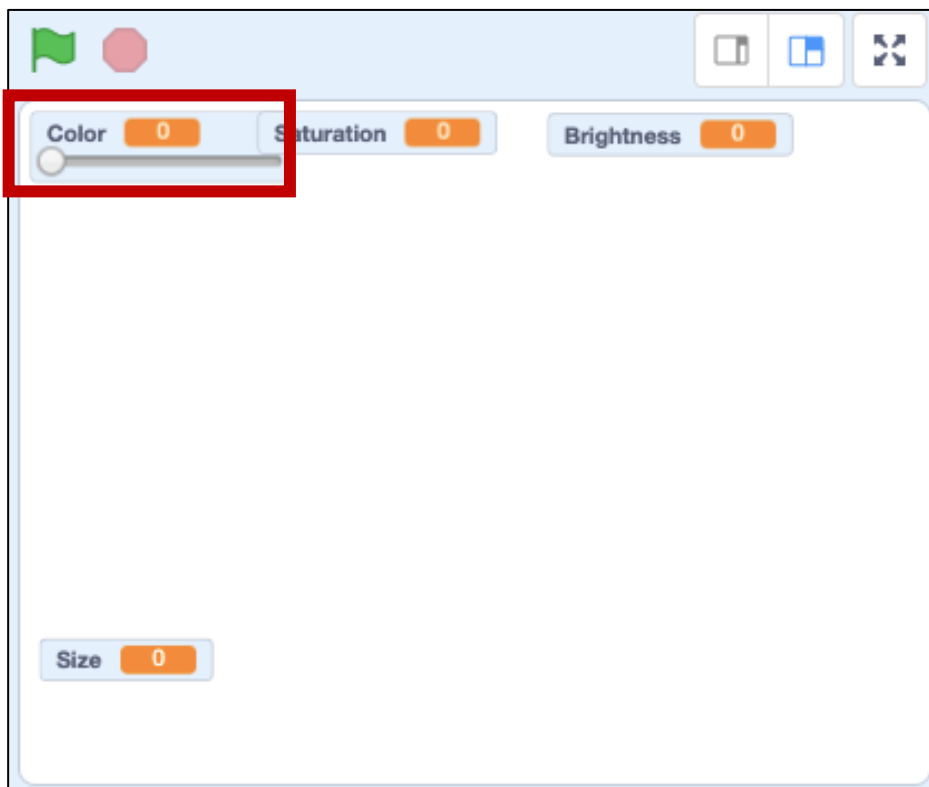
You can click and drag them to where-ever you like, but leave some space in the middle of the screen for drawing 😊



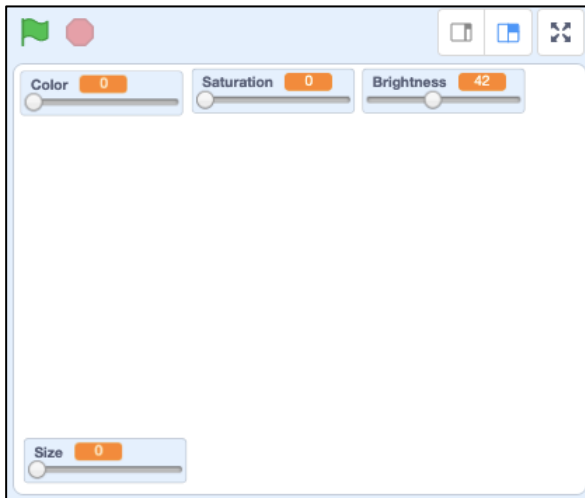
20. Right-click the *Color* variable on the game screen and select *slider*.



The *Color* variable will now be a slider!



21. Make the *Saturation* variable, *Brightness* variable and *Size* variable on the game screen a sliders as well. Click and drag the variable sliders on the game screen so that they don't overlap.



But if you move the sliders, nothing happens!

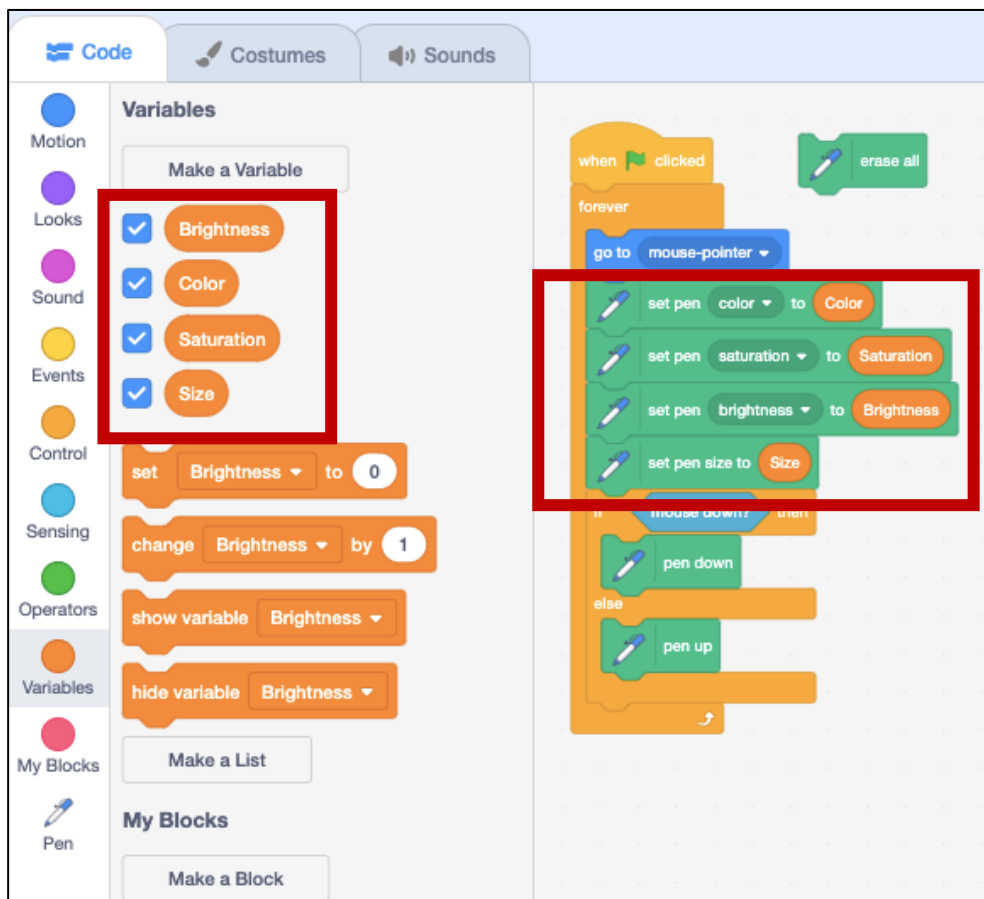
That's because we need to link the variables and our Pen code blocks.

Click and drag the Color block into the "set pen color to ____" block.

Click and drag the Saturation block into the "set pen saturation to ____" block.

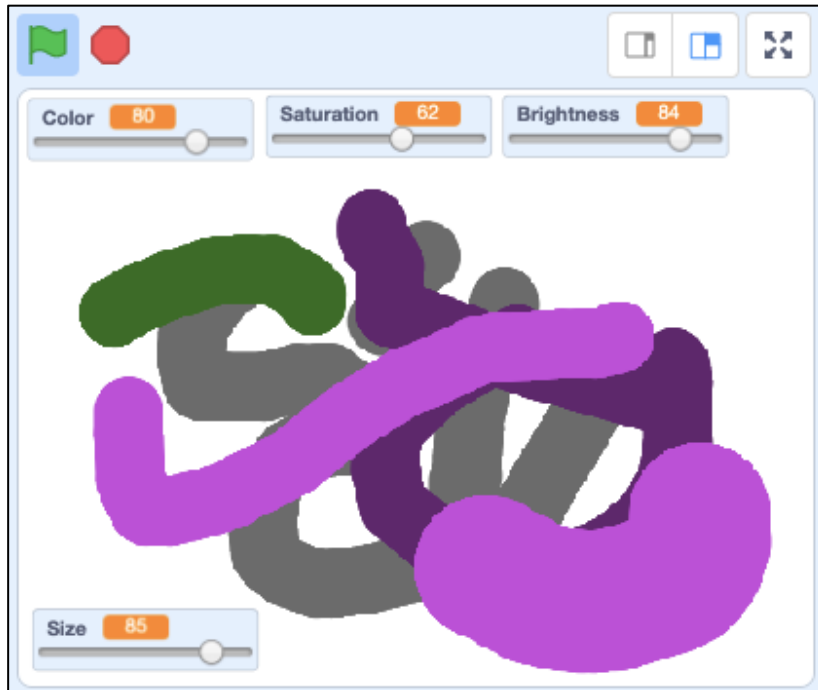
Click and drag the Brightness block into the "set pen brightness to ____" block.

Click and drag the Size block into the "set pen size to ____" block.



22. Now if you move the sliders on the game screen, you can change the color, saturation (intensity), brightness (light or dark), and size (big or small) of the paintbrush.

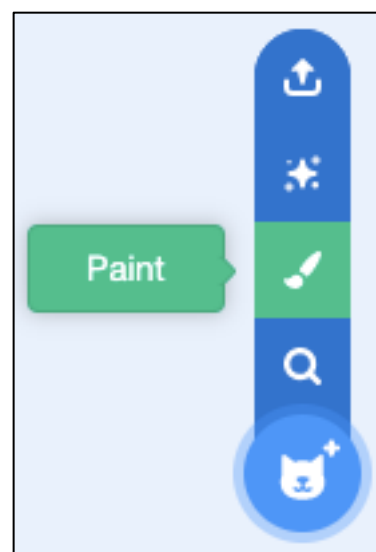
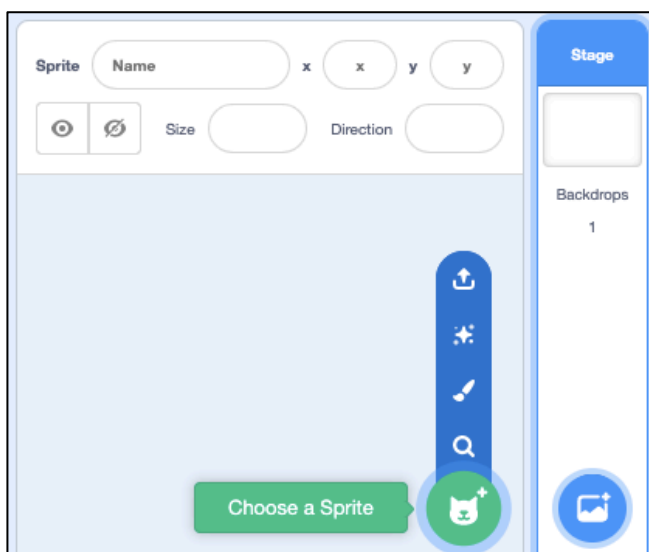
Try it now! Click the green flag button, and try to paint and change the sliders to get different colors, saturation, brightness and size.



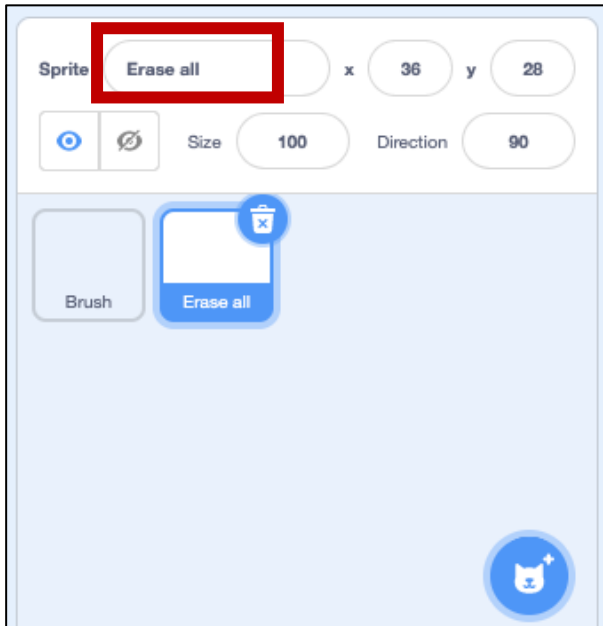
23. But what if we want to erase all painting on the game screen? How do we erase all without clicking on the “erase all” block?

We do this by creating an erase all sprite to be a button you can click on the game screen.

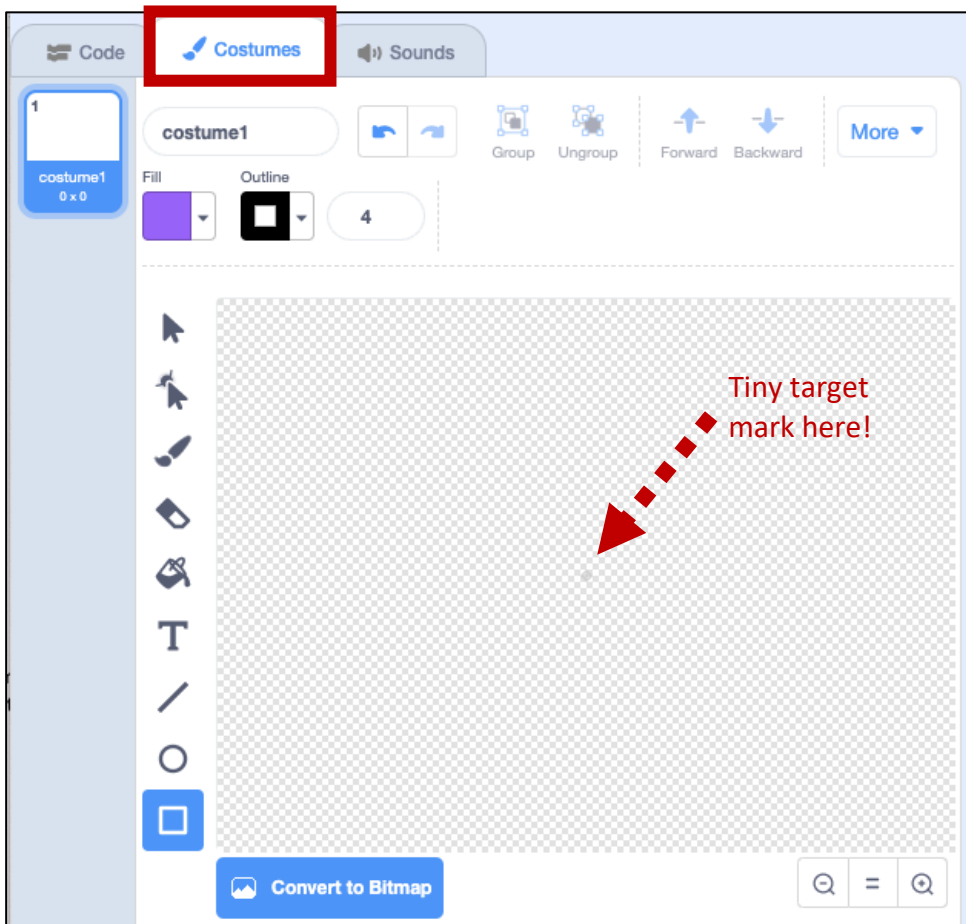
Hover over the *Choose a Sprite* button, then click *Paint*.



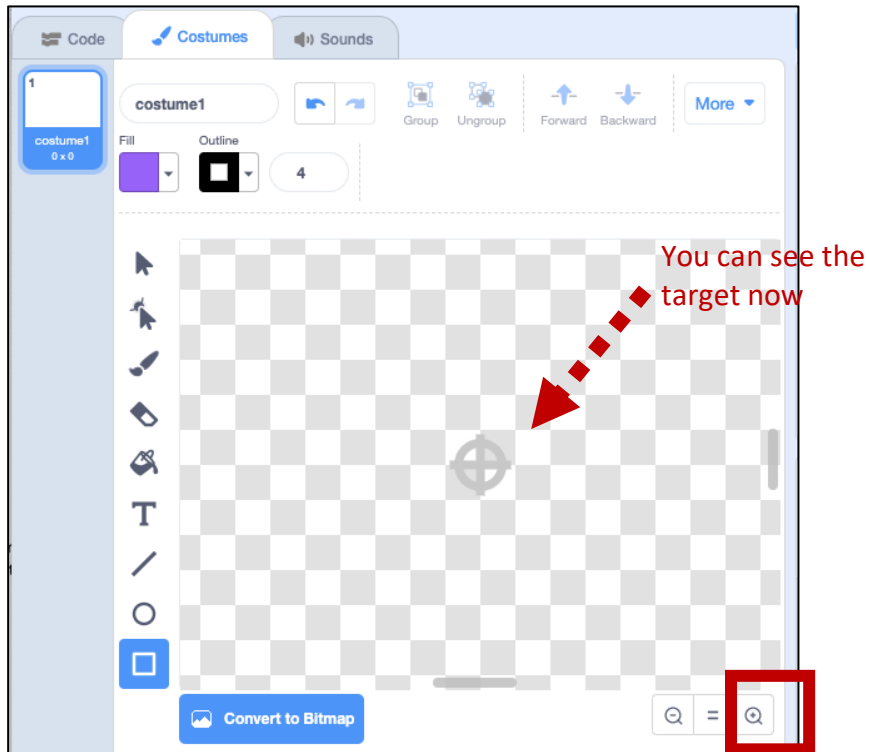
24. Name this Sprite: Erase all



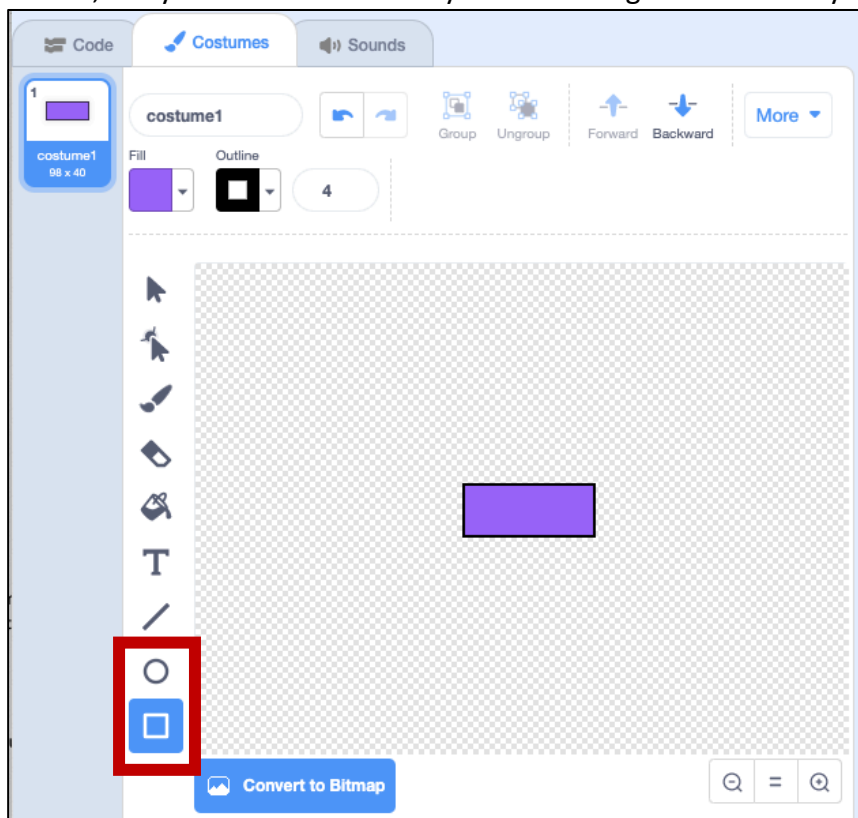
We'll now draw our Erase all button using Costumes. Click the Costumes Tab. You'll see a small dot in the middle of the white-and-grey checked panel. This dot is actually a target mark.



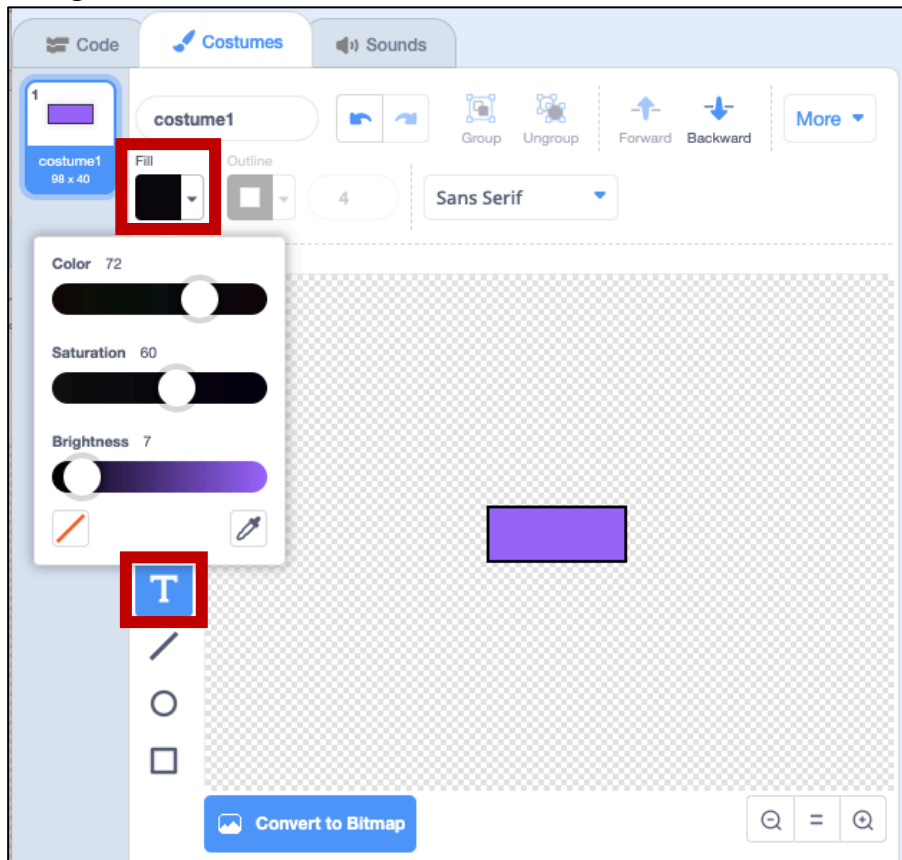
25. To see the target clearly, zoom in using the zoom in button (magnifying glass with +)
When you draw your *Erase all* button, you need to make sure your button covers this target mark.



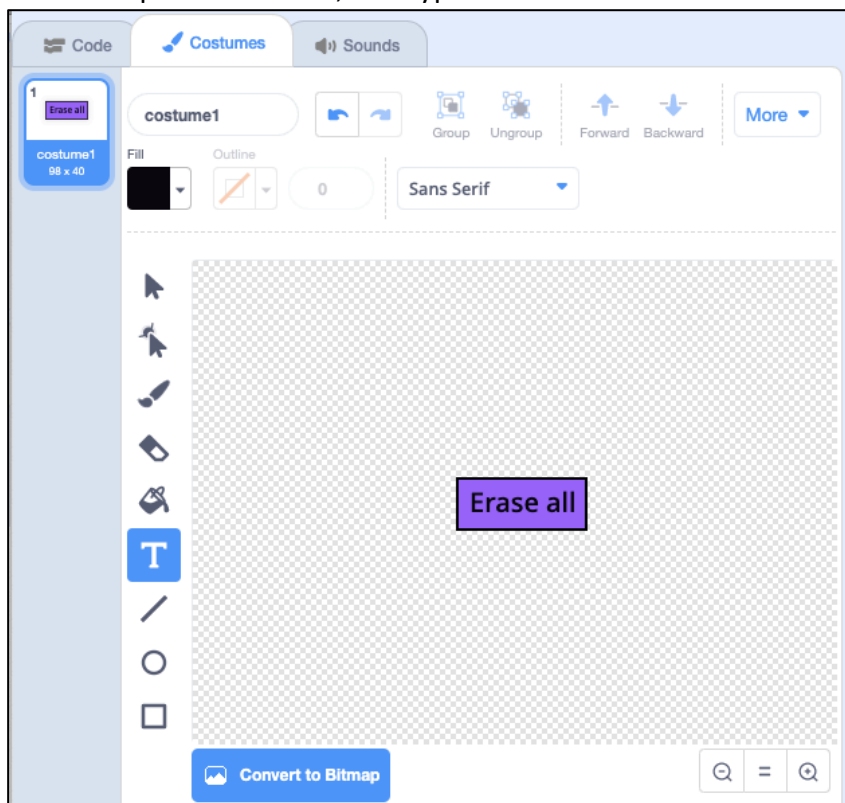
Zoom back out again. Draw a shape over the target mark. I used a rectangle to make my button, but you can use a circle if you like too. Change the colour if you like too!



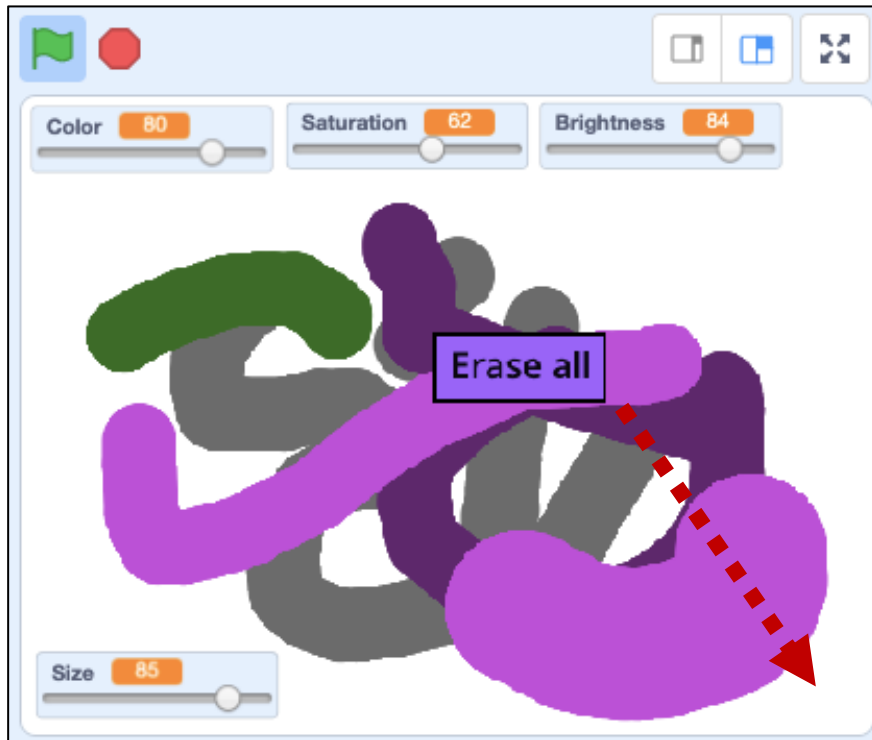
26. Now type the words “Erase all” on top of your button: select the Text button “T”. Change the Text Fill colour to black so it stands out.



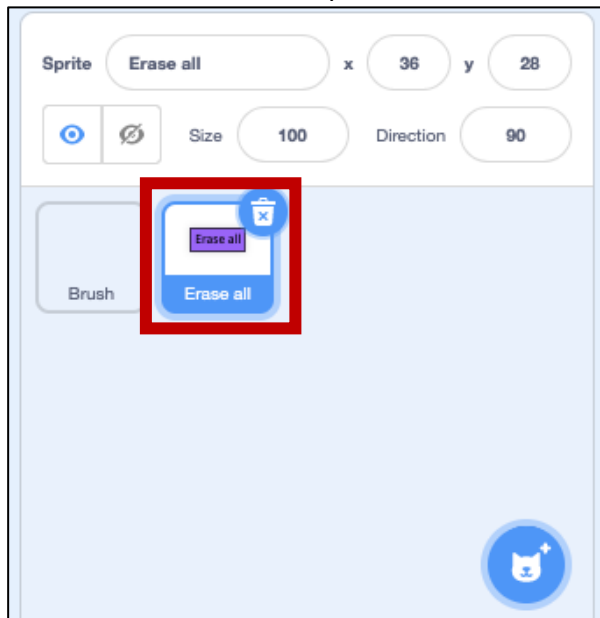
Click on top of the button, and type “Erase all”.



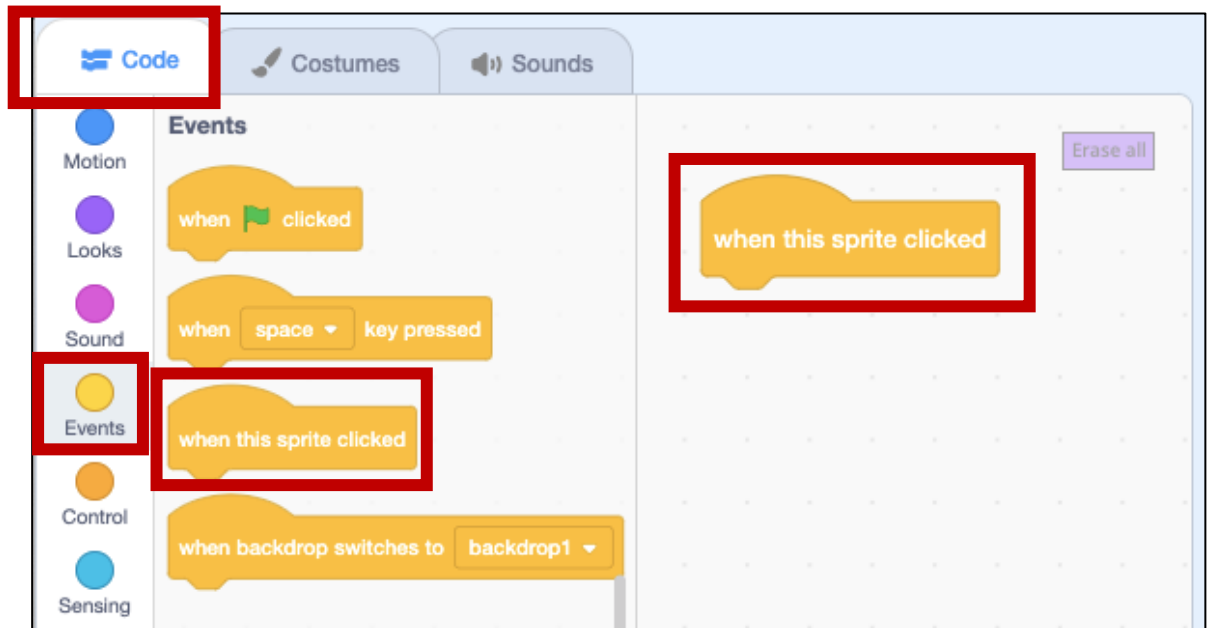
27. Your *Erase all* button had appeared on your game screen! Click and drag it to the corner.



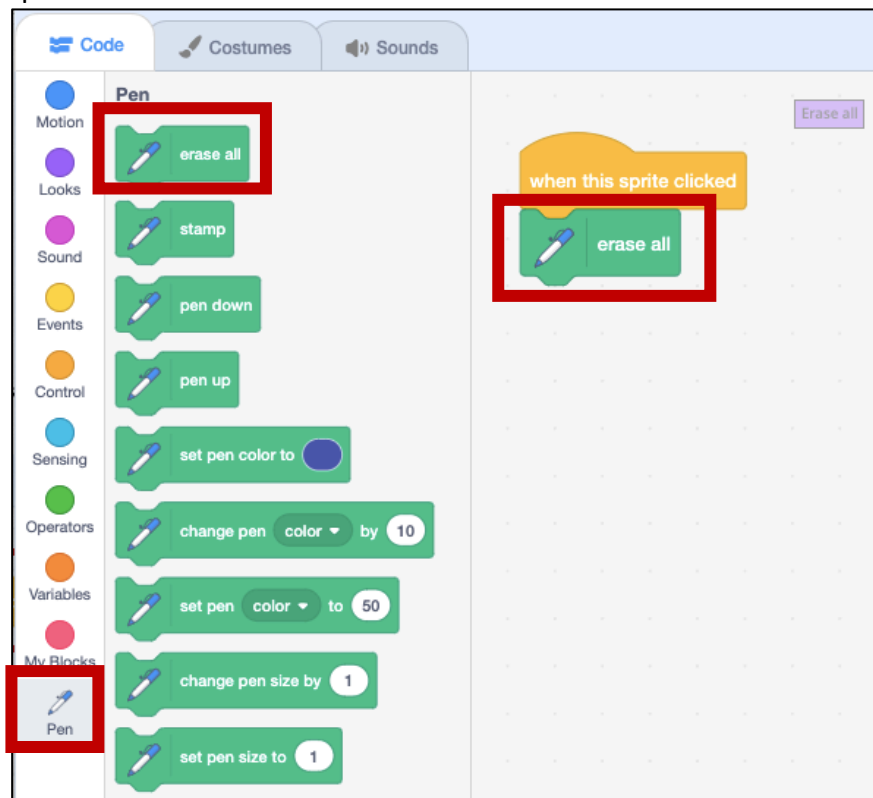
Make sure the *Erase all* sprite is selected in the Sprites panel.



Click on the Code tab. We're now going to code the Erase all sprite so that when it's clicked, all the painting on the game screen is erased. Click the events button, click and drag "when this sprite is clicked" block onto the coding panel.



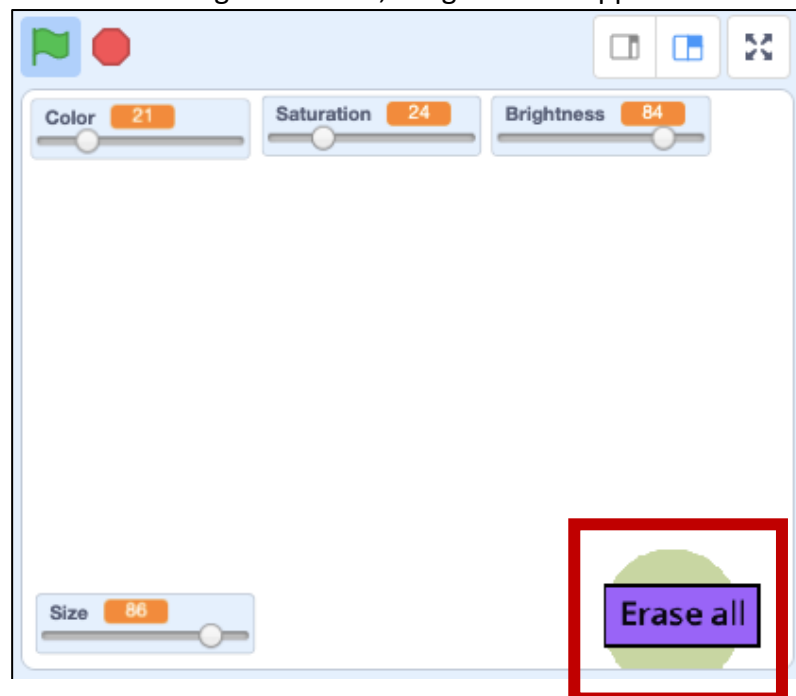
Click the Pen button, click and drag the "erase all" block and put it under the "when this sprite is clicked" block.



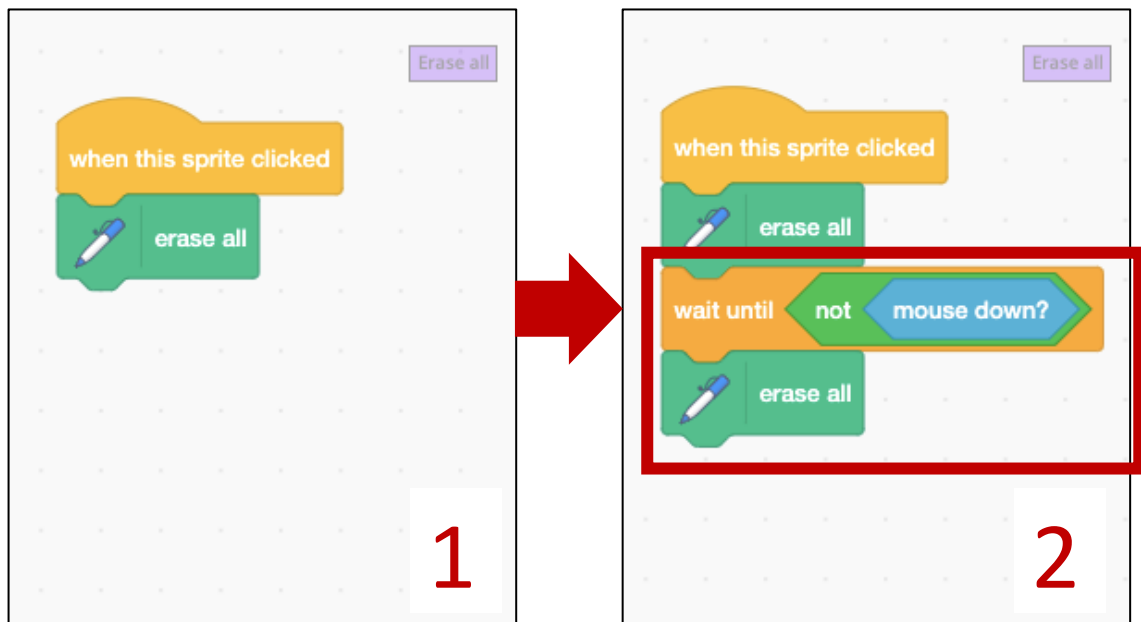
Now your Erase all button on the game screen should work. Try it now! Click the green flag button and paint something. Now click *Erase all* on the game screen. The painting is erased! Click the red stop button to stop the code.



28. You'll notice that whenever the brush size is really big (like 100), whenever you click Erase all on the game screen, a big blob still appears behind the Erase all button.

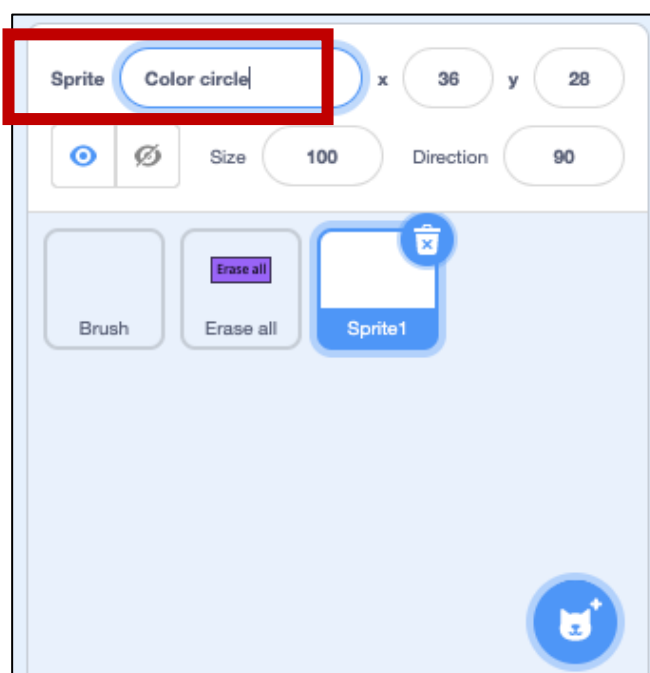


To fix this, we need to make sure that after the Erase all button is pressed, when the Erase all button is no longer pressed, the screen clears again. We do this by using a Control block “wait until”, Operator “not ___” block, a Sensing “mouse down?” block, and a second Pen “erase all” block. Can you recreate these code blocks from 1 to 2?

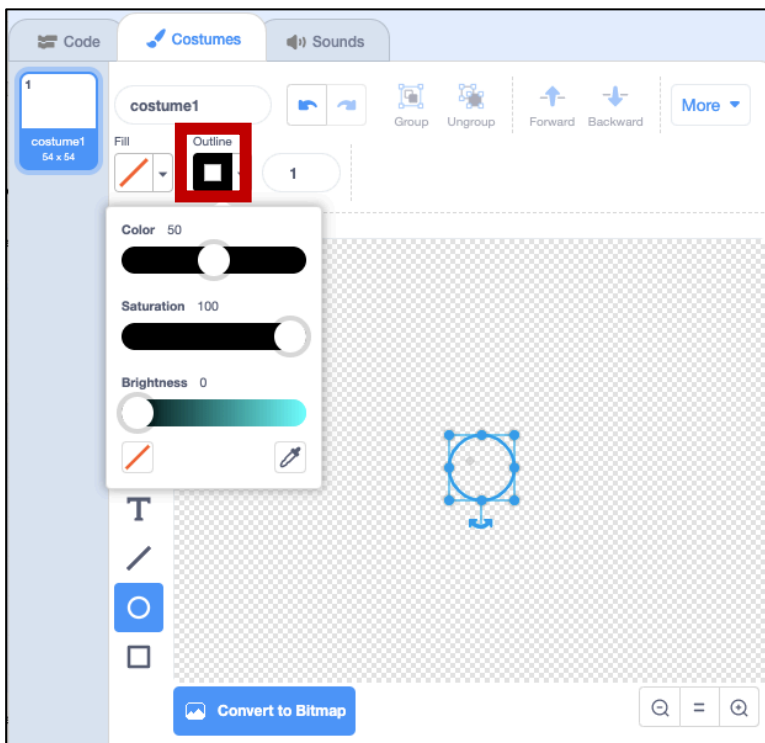
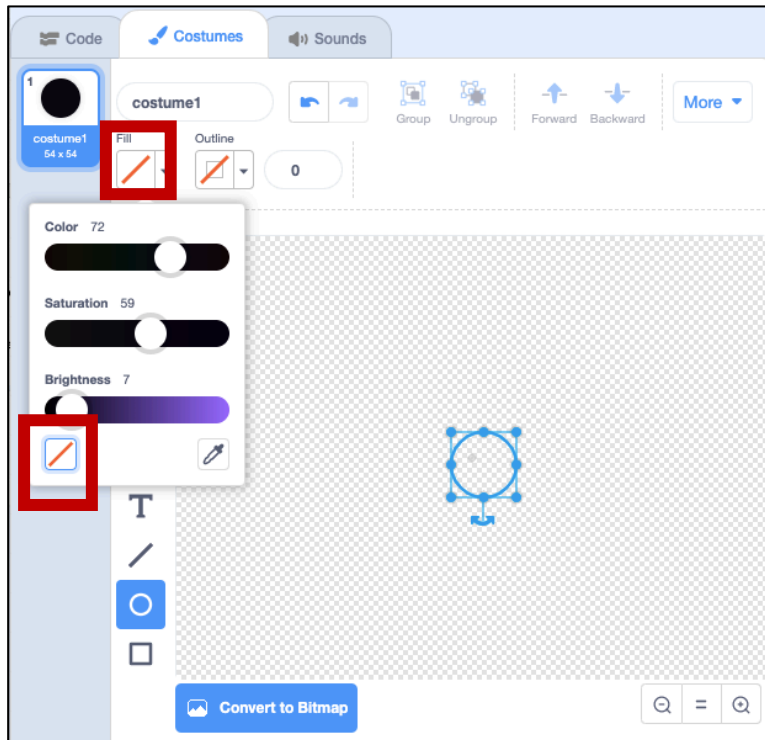


29. We're almost done! Now we just need to make a Color Circle to show us what colour we're using whenever the *Color* slider is changed.

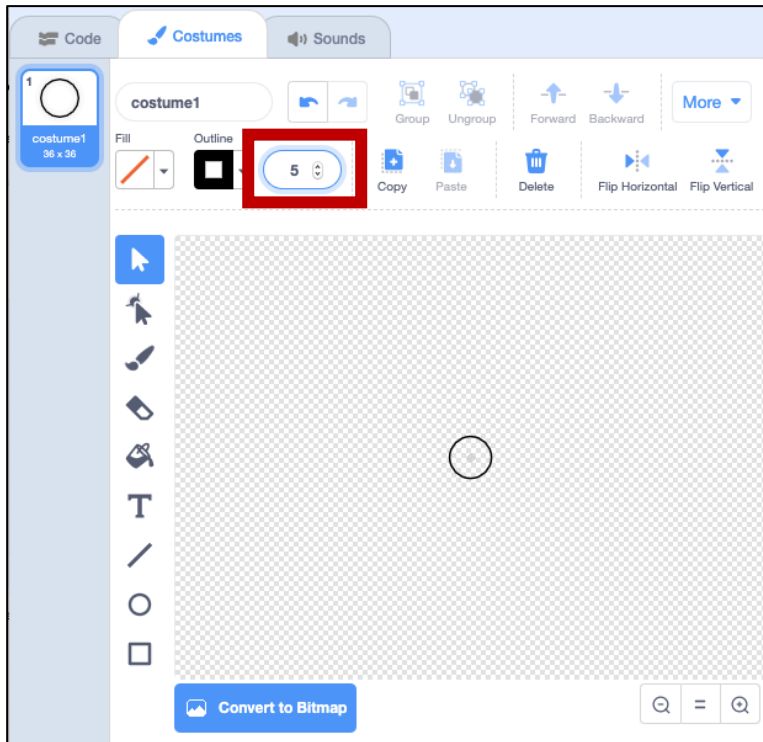
To do this, create a new sprite. Hover over Create a New Sprite, then select Paint. Name the sprite “Color circle”:




On the Costume tab, draw a circle. Set the Fill to transparent (white box with a red line).



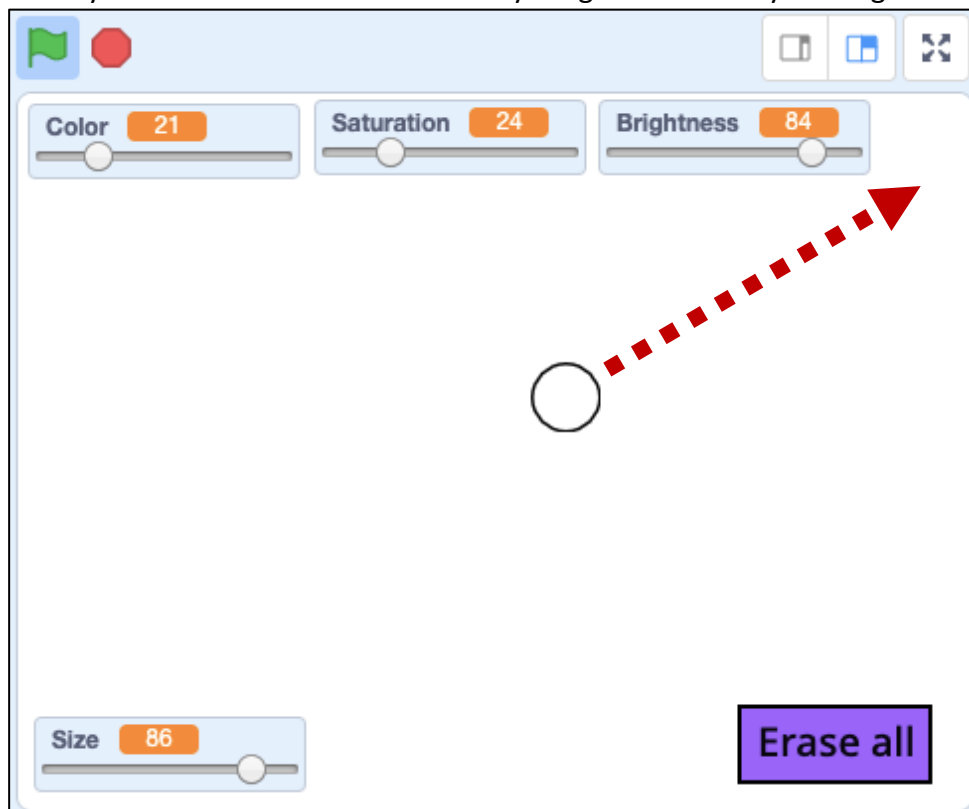
Change the outline to black.
This is so that you have
contrast to see the different
colours in the middle.



Change the *outline size* to 5 so you can see the circle clearly.

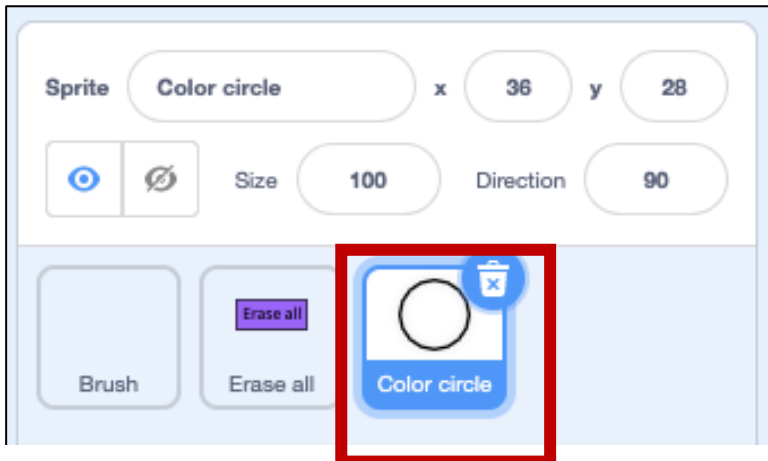
If the circle is not centred on the target mark, select the movement button: 
And click and drag the circle so its centred on the target mark.

Move your color circle to the corner in your game screen by clicking and dragging it:



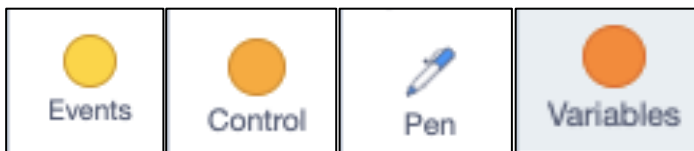
We have a transparent Color Circle sprite with a black border, ready to show our colors.

30. Now to code the Color Circle! Make sure the Color Circle sprite is selected:

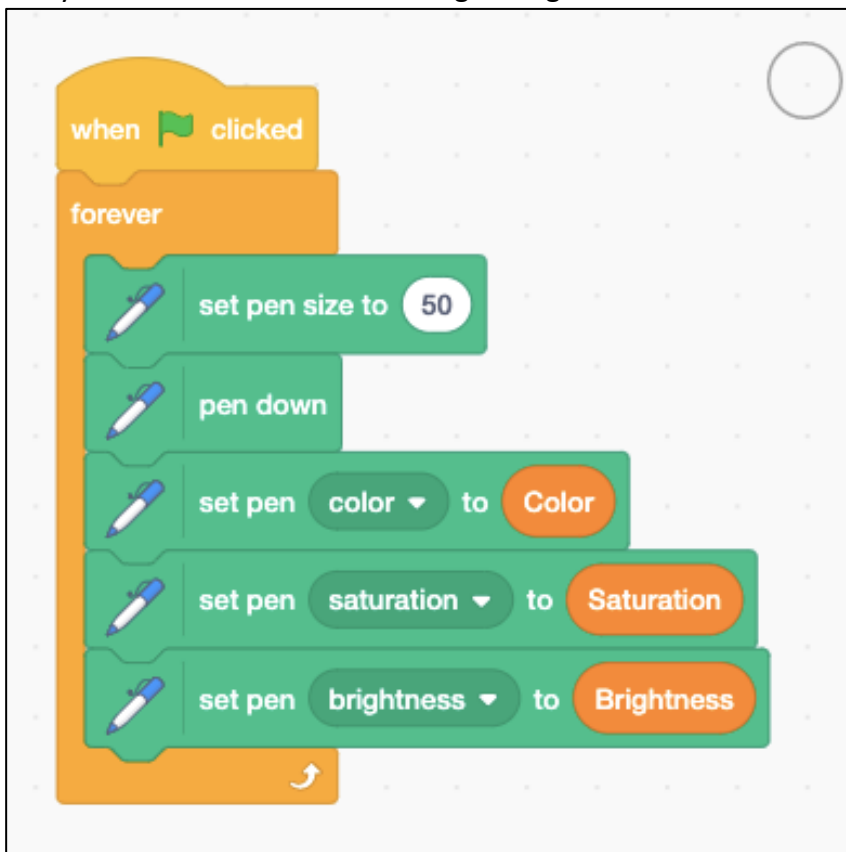


Click the Code tab. We want code that says: *Whenever this paint game starts, we always want to make the color circle change (vary) color whenever the color, saturation, or brightness sliders are changed.*

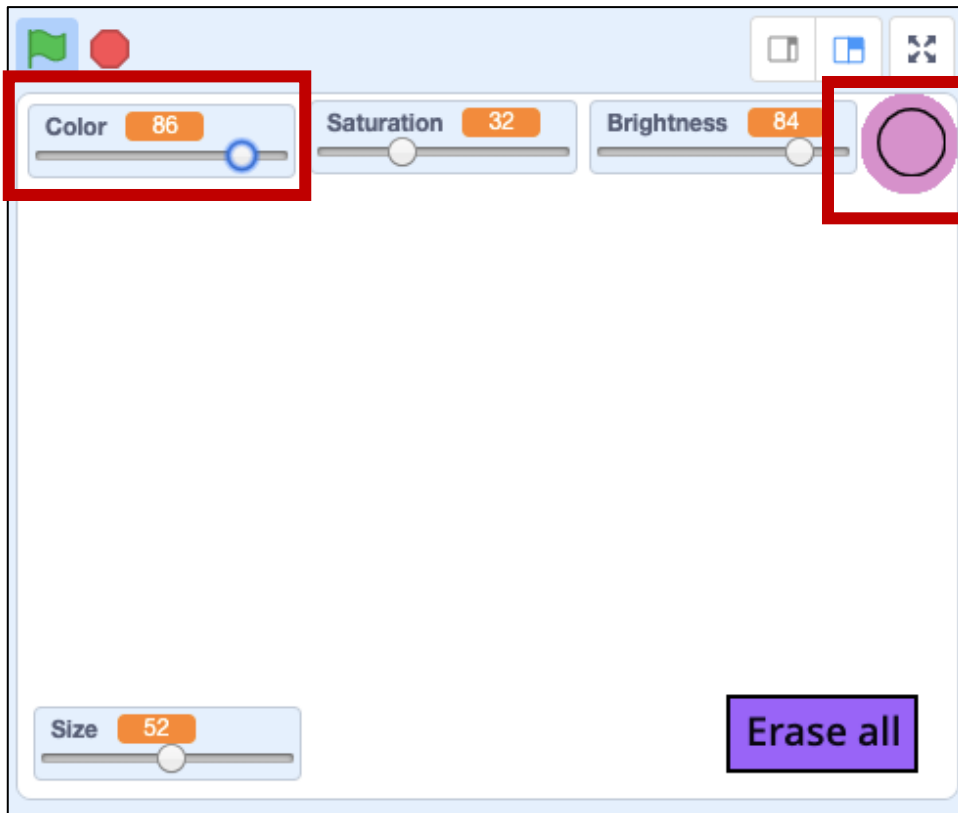
To do that, we build the following code block. It uses Event, Control, Pen and Variable blocks.



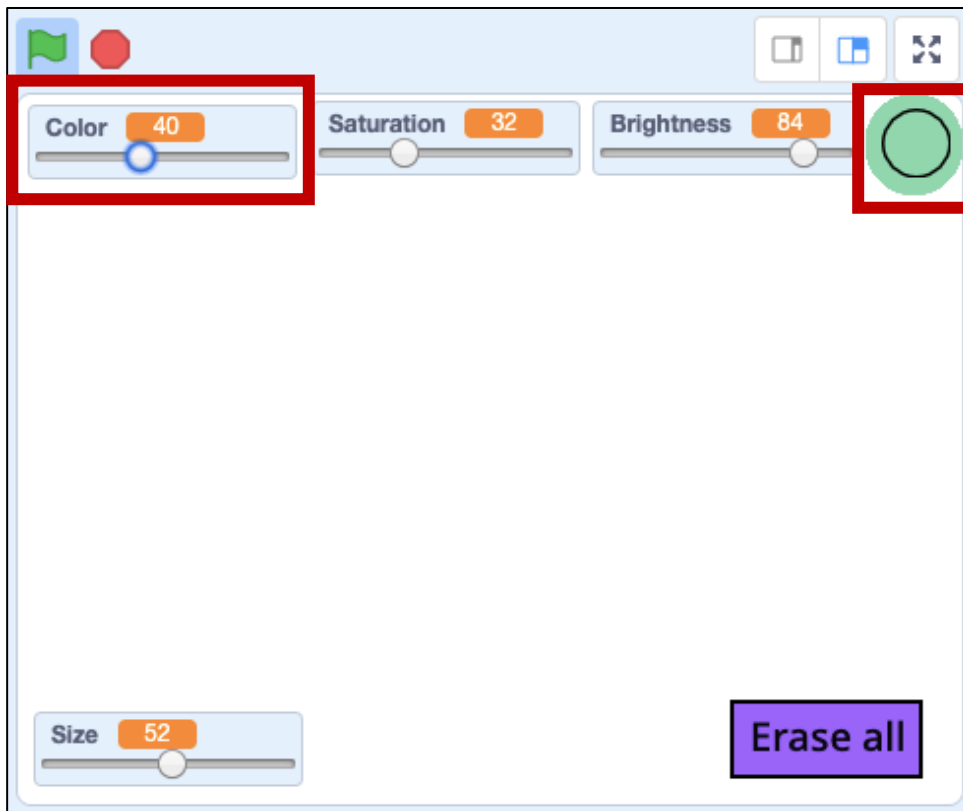
Can you make the code below using the right blocks?



36. Click the "green flag" button on the games screen. Now whenever you change the color, saturation or brightness *sliders*, the color, saturation, or brightness of the *color circle* should change.

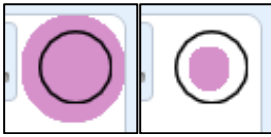


For example:
changing the
color slider
from 86 to
40...



... changes
the *color
circle* from
pink to
green!

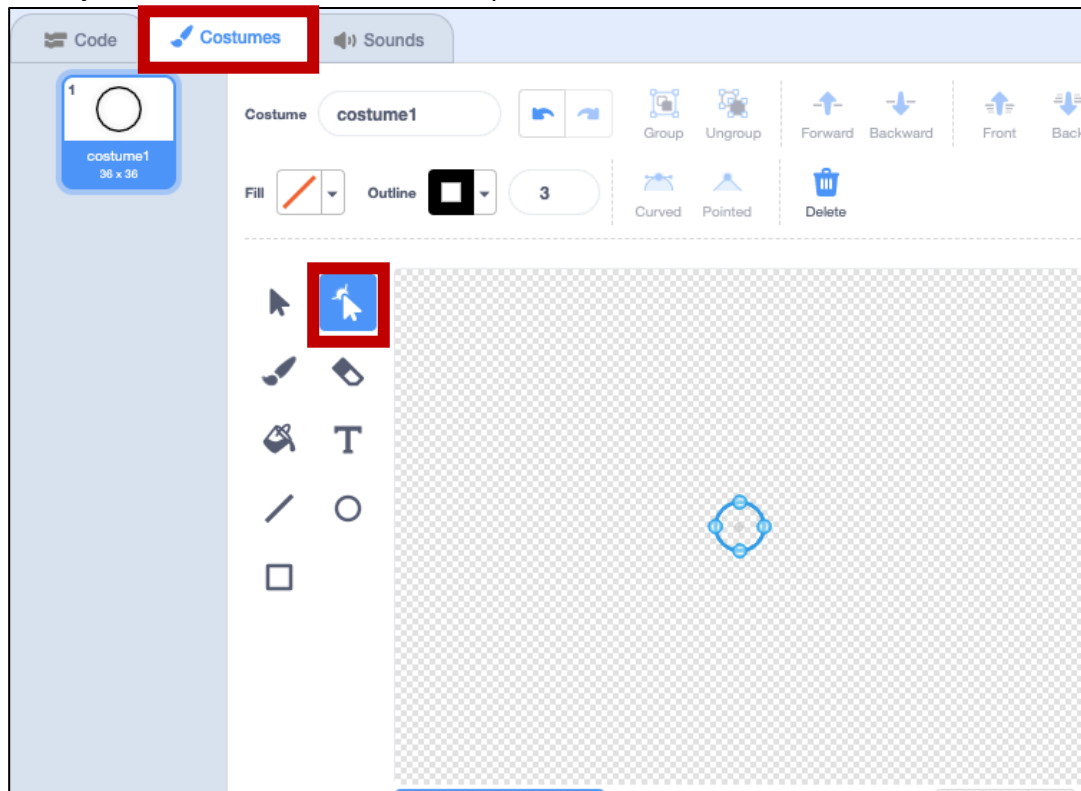
37. If your color circle is too big or too small for your black border:



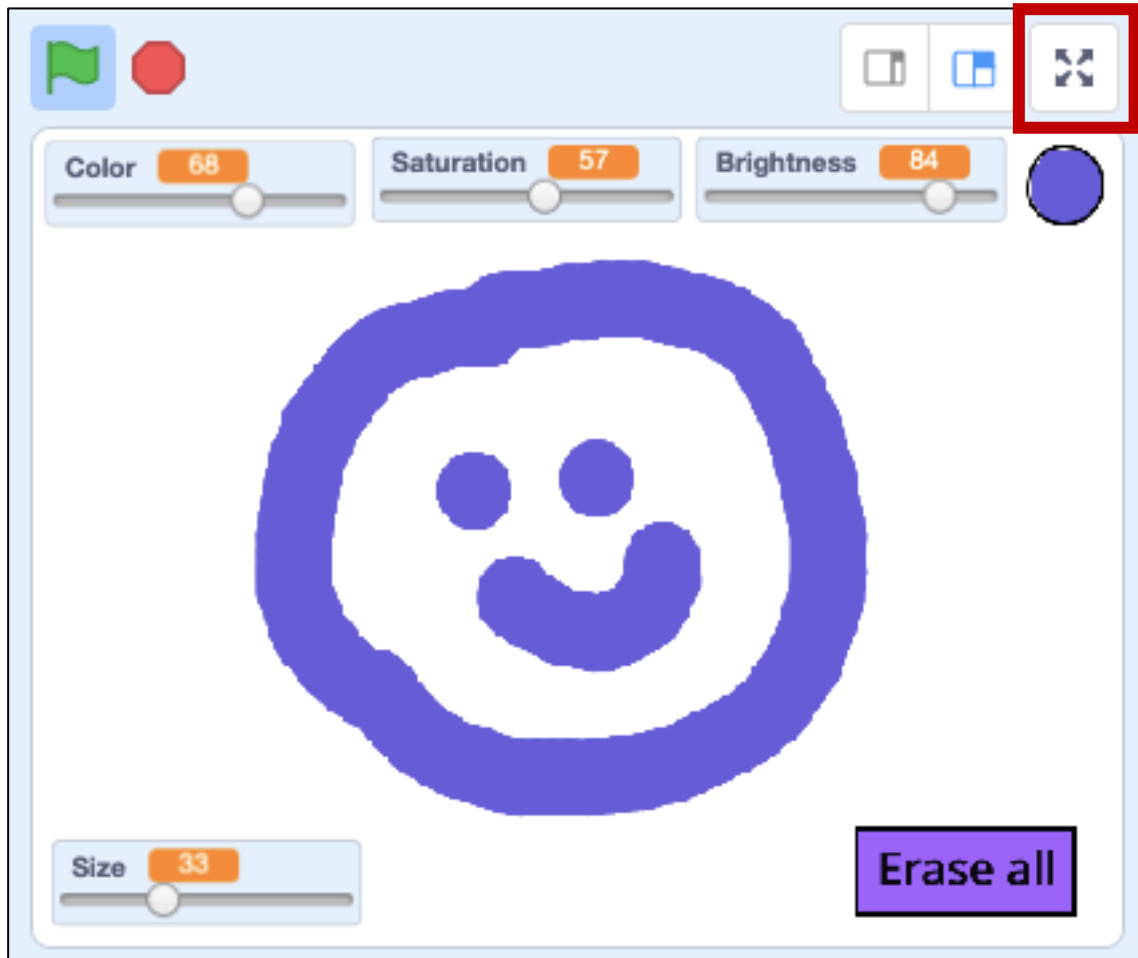
you can change the size of the “set pen size to ____” block for the color circle:



OR adjust the color circle size of the sprite in Costume:



38. And there you have your finished painting game! Click the full-screen button above your game screen to make the game screen full-size and paint whatever you want.



Congratulations, you just coded your own game by yourself!