Level 3

Paint Box

Introduction:

The project creates a drawing tool for you to make your own art. You can change the colour of the line, clear the screen, make stamps and much more!

Activity Checklist – Follow these **INSTRUCTIONS** one by one

Test Your Project – Click on the green flag to **TEST** your code

Save Your Project – Click on this to **SAVE** your work

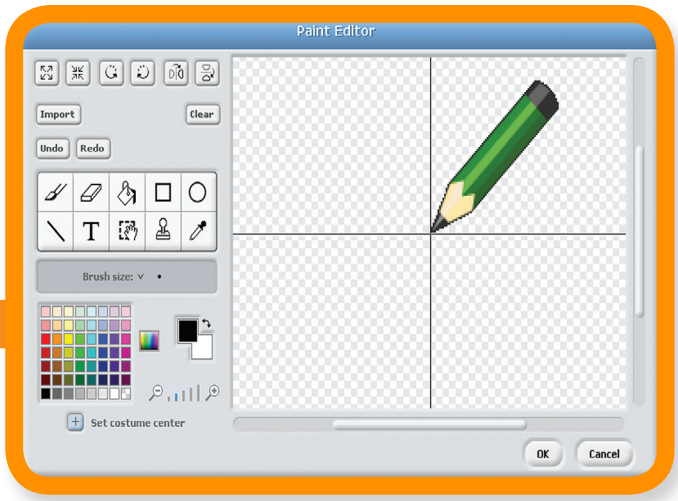
STEP 1: Drag and draw

Keep track of your progess by ticking off the boxes below:

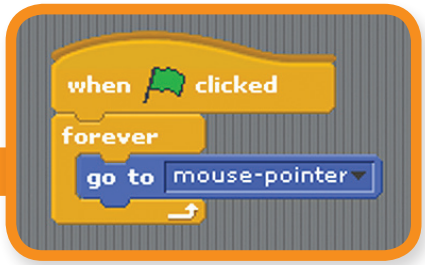
Activity Checklist

We'll start with a pen that draws when you drag it around the Stage.

1. **Start a new Scratch project.** Delete the cat by **right clicking** it and clicking **Delete** ☐
2. Click **Stage** and then the **Backgrounds** tab. Import the **indoors/chalkboard** background. ☐
3. Create a new sprite called **pencil**, using **resources\green-pencil.** ☐
4. Switch to the **costumes** tab and click **edit** from the **Paint Editor,** change the center of the image to be at the tip of the pen. To do this, click **Set costume center** and drag the lines until they are at the tip. ☐



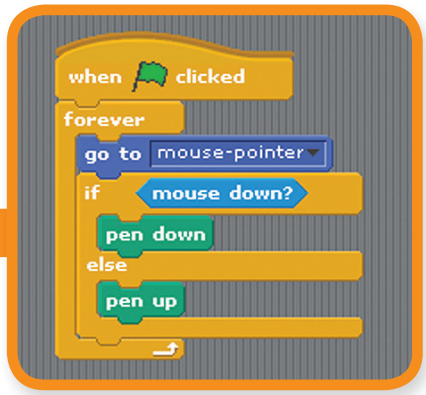
5. Make the pencil follow the mouse around the **stage** using the **forever** and **go to** mouse-pointer blocks. ☐



Now we want to use this pen sprite as an actual pen. If you look under the pen section you'll see all sorts of drawing related blocks. The ones we're initially interested in are **pen down** and **pen up**

Keep track of your progess by ticking off the boxes below:

6. **We want to use the mouse button to control the pen** – whenever the mouse button is down the pen should be down, and when it is up the pen should be up. We can do this using an **if... else** and **mouse down?** blocks.

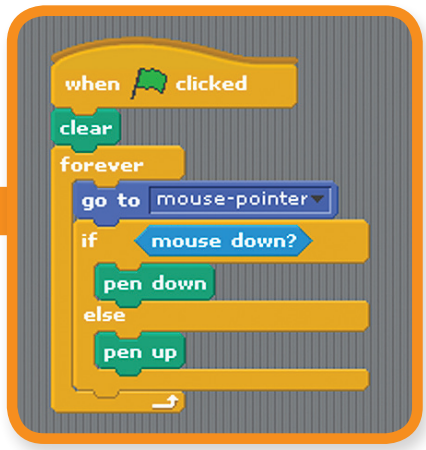


Test Your Project

Click on the green flag.

Does the pen follow the mouse around? What happens if you hold the mouse button down and move the mouse? Don't worry about the pen colour for now.

7. Eventually the screen is going to get pretty filled with scribbles. The **clear** block can be used to clear the screen.



Test Your Project

Click on the green flag.

Does your drawing disappear when you click on the green flag?

SAVE YOUR PROJECT

STEP 2: Clearing up

Keep track of your progess by ticking off the boxes below:

Rather than having to stop and start the whole project, let's add a button that clears the drawing. It will do that using the clear block.

Activity Checklist

1. Create a **new sprite** from the **resources/cancel button** costume.
2. Change the sprite's name to **clear**.
3. Position the sprite near the bottom-left corner of the stage.
4. **Give the clear sprite this simple script:**

Test Your Project

Click on the green flag.

Does the clear button clear all your drawing?

SAVE YOUR PROJECT

STEP 3: Changing colour

So far, we can only draw blue lines. Let's draw with some different colours! We'll add some sprites at the bottom of the frame. The sprites will look like coloured buttons. If we click on a button, it will change the colour of the line we draw. So we know what colour we're using, the button will also change the colour of the pencil sprite.

Activity Checklist

1. Add a **new sprite**, called **red**, using the **resources/red-selector** costume.
2. Place it somewhere along the bottom of the frame, near the **clear button.**

Keep track of your progess by ticking off the boxes below:

3. When the red sprite is clicked, it should broadcast the message **red**.

Yes, that's all it does. The hard work is done by the pencil.

In the pencil, **import** a **new costume**, **resources/red-pencil**. Set the costume centre to be the tip of the pencil as you did for the original costume.

4. **Add a new script to the pencil.** When the pencil receives the message **red**, it should change to the red pencil costume and change the pen colour to red (using the set pen color to block).

Hint: if you click on the coloured square in the set pen color to block, you can click the **eyedropper** on the **red sprite** to make sure it's the same colour.

Test Your Project

Click on the green flag.

Start by drawing a line. Then click on the red selector sprite and draw some more. Does the pencil change costume? Does it now draw red? Does it draw from the tip of the red pencil?

SAVE YOUR PROJECT

5. **Repeat what you just did for the blue, yellow, and green selector sprites.**

Keep track of your progess by ticking off the boxes below:

Test Your Project

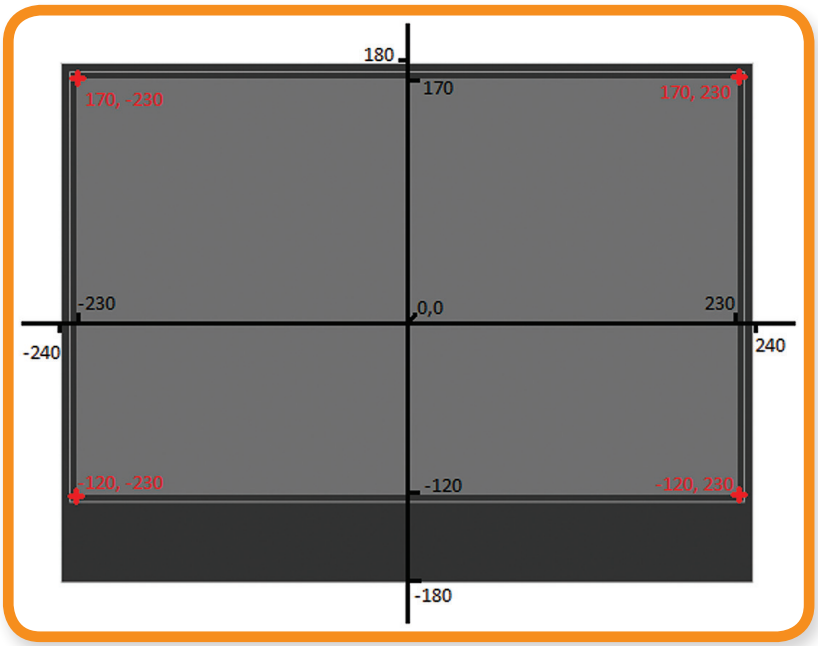
Click on the green flag.

Do all the selector buttons work? Do they all change the pencil's costume to the right colour? Do they all make the pencil draw in the right colour? Do all the costumes draw with the tip of the pencil?

SAVE YOUR PROJECT

STEP 4: Only drawing inside the border

You've probably noticed that you can draw all over the Stage, even in the border. We don't want that to happen. We want to keep the drawing in the middle of the Stage. We can do this by making sure the pen isn't allowed to leave the **drawing area** - the light grey part on the **Stage**.



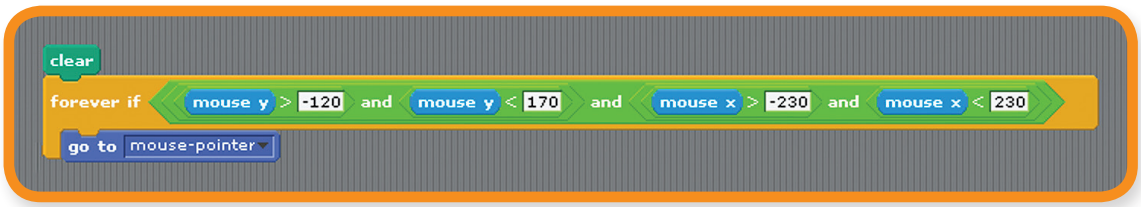
Remember that Scratch defines points using **x** and y axis. Our **drawing area** lies between **230** and **-230 on the x-axis** and **170** and **-120 on the y-axis**. We can use these values in an **if** block, making sure the mouse is inside this area before we move the pencil to it.

Keep track of your progess by ticking off the boxes below:

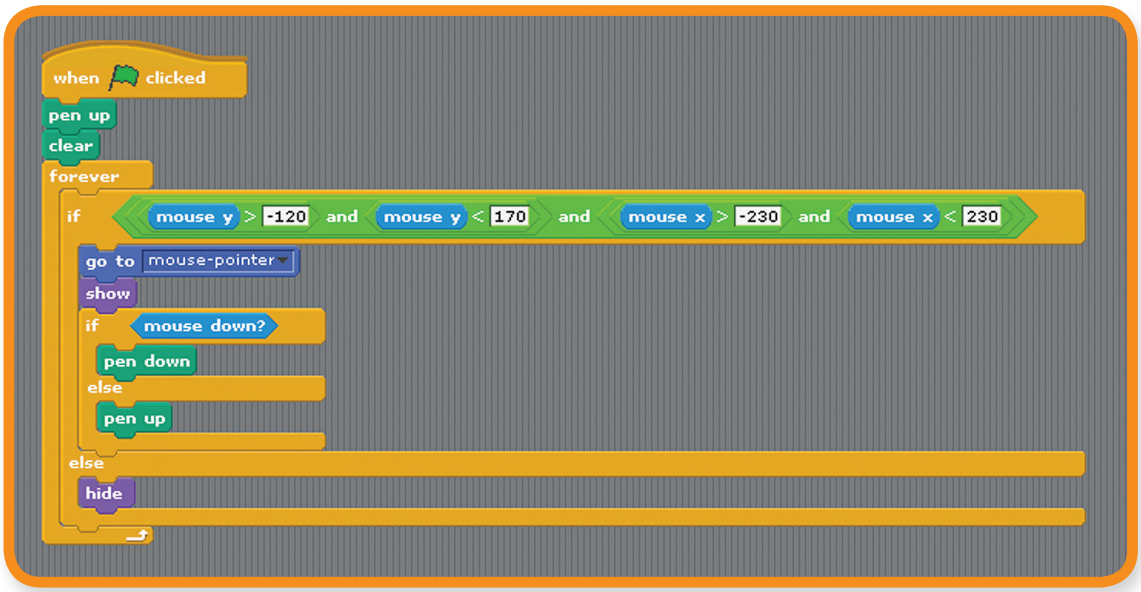
To do this, wrap a new **if** block around your existing **go to... if** block, and inside this new **if** check for the following:

mouse y is greater than **-120** and **mouse y** is less than **170** and **mouse x** is greater than **-230** and **mouse x** is less than **230**

Note to do this you'll need to use multiple **and** operator blocks, one for the two **mouse x** conditions, one for the two **mouse y** conditions and a final one to join these all together:



Since we can't draw outside of the drawing area, we could hide the pencil tool whenever we leave it. To do this, replace the **if** with an **if else** block. Keep the same condition for the **if**, and **show** the pencil if it's true, otherwise hide it.



Test Your Project

Click on the green flag.

Can you still draw inside the drawing area? Can you draw outside the drawing area? What happens to the pencil when you leave the drawing area and go back in?

Keep track of your progess by ticking off the boxes below:

SAVE YOUR PROJECT 

STEP 5: Eraser

Drawing lines is great, but there are times when you've made a mistake and you want to rub it out. We can do that with a new pencil tool that draws in grey (the same colour as the background).

Add a new button-sprite to the Stage to select the eraser. Use the **resources/eraser** costume for it, making it smaller to fit at the bottom of the **Stage**. It should work the same as the other colour-selection buttons, sending an eraser message.

The pencil sprite should respond to the eraser message by switching the pen colour to grey (remember you can use the **picker** to select the colour of the background). It will also need a new costume to represent the eraser tools: use the same **resources/eraser** costume. **Remember to reset the costume's centre.**

Test Your Project

Click on the green flag.

Does the eraser rub out lines? Does it work right up to the edges? Can you switch between eraser and pencil tools?

SAVE YOUR PROJECT 

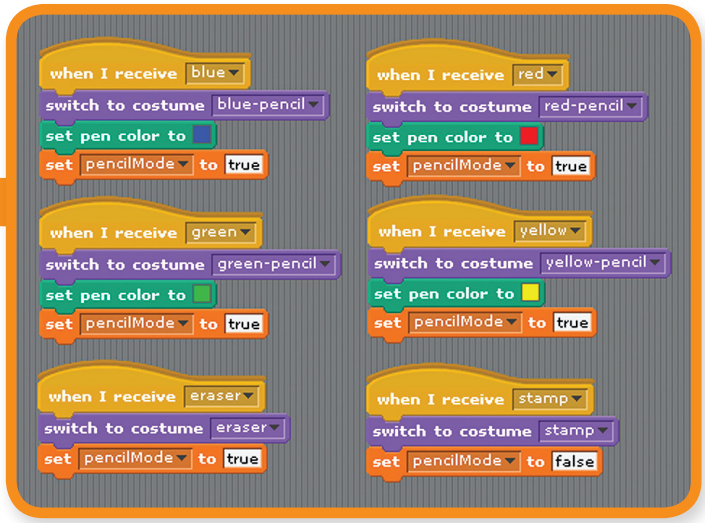
STEP 6: Stamps

Keep track of your progess by ticking off the boxes below:

The next thing to add is a stamp tool, to stamp small pictures on the drawing.

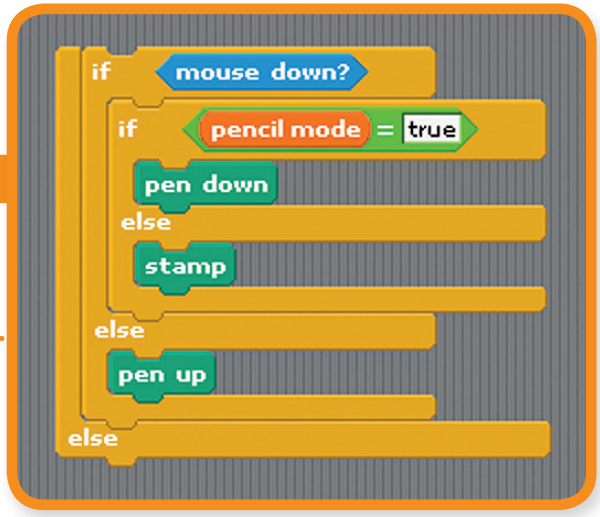
Activity Checklist

1. **Add a new sprite**, using whichever image or costume you want. Shrink the sprite down and place it at the bottom of the screen alongside the other tools. When this sprite is clicked, it should **broadcast stamp** ☐
2. **Add a new costume** for this pencil sprite, the same as the one you chose for the **stamp** button. ☐
3. **Select the pencil sprite** and create a new **variable** **pencil mode** for this sprite only. We'll use this variable to keep track of whether or not we are drawing or stamping. ☐
4. **Add a new script** to respond to the **stamp** message. It needs to set the costume to the stamp and set the pencil mode variable to false. ☐
5. **Change the other scripts** that respond to tool-selection messages (**red, green, blue,** and **eraser**) so that they each set the **pencil mode to true** ☐



Keep track of your progess by ticking off the boxes below:

6. Finally, lets check this variable **when the mouse button is down** to see if we should be drawing or stamping. If pencil mode = true we should use the existing pen down, if not we should stamp instead.



Test Your Project

Click on the green flag.

Does the stamp tool work correctly?

What happens when you switch back to one of the normal pencil tools?

SAVE YOUR PROJECT

Keep track of your progess by ticking off the boxes below:

Well done, you have completed the basic steps for this project.

Try these challenges!

Challenge 1: Rainbow pencil

Let's add a special pencil that paints in rainbow colours. It's something that you can't do with ordinary pens and pencils, so it's nice to show off how drawing on a computer allows you do to different things. The secret to making it work is the change pen colour by block.

First, add the rainbow tool selection sprite and the rainbow tool costume to the pencil sprite:

1. **Create a new tool selection sprite** and place it at the bottom of the stage, alongside all the other pencil colour sprites. Use the **resources/rainbow-selector** costume and have it **broadcast rainbow** when clicked. ☐
2. Add the **resources/rainbow-pencil** costume to the **pencil sprite.** ☐

You need to build a script that will change the pen colour many times a second to give the rainbow effect **(I found that changing it by 5 every 0.05 seconds works well, but you should try out different values)**. The **timer** Scratch card shows how you can make something change every so often. Use a **change pen colour by 5** block instead of a **change timer by -1** block inside the loop. ☐

You also need to control that loop so that it only changes the pen colour when you've selected the rainbow pencil, otherwise all the pencils will have a rainbow effect! You can do this in a very similar way to how the pencil sprite changes between pencil and stamp modes. You need to create a **variable** called **rainbowChange** that has the value true when you want the rainbow effect and false otherwise. Every time the pencil responds to a tool-selection message, it should set the value of **rainbowChange** accordingly. ☐

Use what you learnt from the stamp step above to control the rainbow effect. The scripts that respond to the tool-selection messages will set two **variables** each: **pencilMode** and **rainbowChange**.

Keep track of your progess by ticking off the boxes below:

Test Your Project

Click on the green flag.

Does the rainbow tool work correctly?

What happens when you switch back to one of the normal pencil tools?

SAVE YOUR PROJECT

Challenge 2: Keyboard shortcuts

Rather than using the tool-selection sprites at the bottom of the stage, you can use the keyboard to select the different tools.

You can use the when [] key pressed blocks to respond to the keyboard. For each key you want to use, you'll need another when [] key pressed block, which sends the same message as the respective tool-selection sprite does when its clicked. **Add these scripts to the stage.**

I used these shortcuts:

Clear all	**a**
Eraser	**e**
Red pencil	**r**
Blue pencil	**b**
Yellow pencil	**y**
Green pencil	**g**
Rainbow pencil	**w**
Stamp	**s**

Test Your Project

Click on the green flag.

Do all the tools get selected with the correct keyboard shortcuts? Does each of the tools work correctly when you select it with keyboard? Are the correct tools still selected with the tool-selection sprites on the stage?

SAVE YOUR PROJECT

Keep track of your progess by ticking off the boxes below:

Challenge 3: Big and Small

Another feature that most drawing packages have is the ability to change the size of the pencil. Let's add that.

There's one complication, though, which is that sometimes the resizing needs to change the pen size and sometimes it needs to change the pencil sprite's costume size. It depends on whether you're using a pencil or a stamp.

Create two new tool-selection sprites, called **bigger** and **smaller**. They should have the **resources/bigger-selector** and **resources/smaller-selector** costumes and should send the bigger and smaller messages. ☐

The pencil sprite can respond to the messages by changing either the **pen size by 1** or the **costume size by 10**, depending on the value of pencil mode (use an if-else block, similar to how the sprite chooses between putting the pen down or stamping)

Don't forget the keyboard shortcuts for the bigger and smaller tools. I used the up and down arrows. ☐

☐

What you should have noticed is that changing the size of the stamp also changes the size of the pencil on-screen when you change to that tool. ☐

To stop that, you need to **set the size to 100%** every time you change to a pencil tool. so that the tools look the right size. ☐

To make it even better, have the stamp remember what size it was before you selected the pencil and go back to that size when you select the stamp tool again. The easiest way to do that is to create a new **variable** called **stampSize**, that is updated with the current size every time the stamp is resized. When the stamp tool is selected, it can set its size from the contents of this variable. ☐

Keep track of your progess by ticking off the boxes below:

Test Your Project

Click on the green flag.

Do the size controls work for the pencils?

What happens if you switch to the stamp, change the size and then switch back to a pencil?

SAVE YOUR PROJECT

Well done you've finished, now you can enjoy the game!

Don't forget you can share your game with all your friends and family by clicking on **Share** on the menu bar!