Scratchy Etch-A-Sketch Project

This Scratch project will demonstrate the use of a simple data structure in Scratch. We will create a project that will keep track of the steps that Scratchy takes, and places the X,Y coordinates in a pair of parallel stacks. Then when Scratchy reaches the border, reverse the stacks and follow the path back to the starting point.

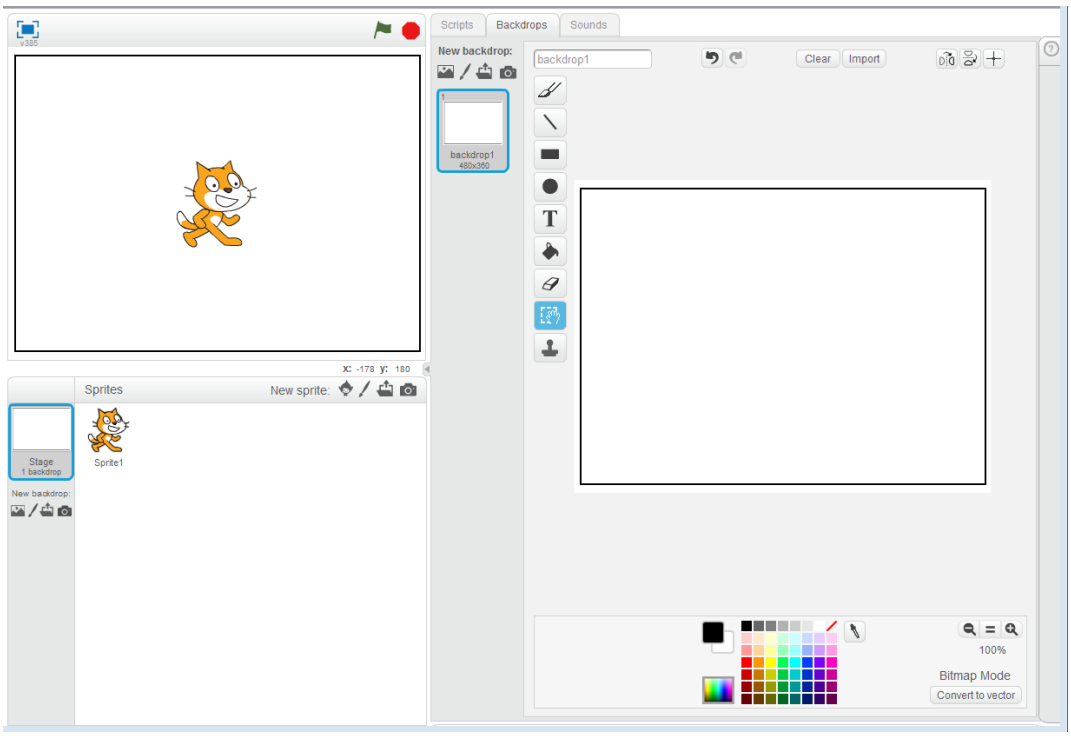
Step 1: Start Creating!

Create a new project

Step 2: Set Boundaries

Create the border that will contain Scratchy and be his destination. Scratchy's journey will end when he runs into a border.

- Select the "Stage", go to the "Backdrops" and create a square border along the edge of the backdrop.
- Keep track of the color used for the border. Black is the default and assumed moving forward.



Step 3: Create Lists

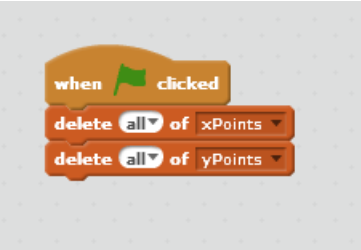
Create two lists that will be used for the parallel stacks.

- Go back to “Scripts” tab and click on the “Data” palette.
- Click on the “Make List” button and make two lists, xPoints and yPoints. Remember to make sure they are set for all sprites.

Step 4: Reset Stacks

Next, we need to make sure that our stacks will always be reset to empty each time the game is restarted.

- Go to the “Scripts” tab for the Stage and enter blocks to do this.



Step 5: Add Scratchy

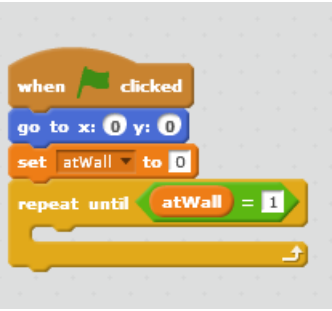
Now, let’s make Scratchy a little smaller, because his size will get in the way of seeing his path.

- Select “Sprite1” and use the “Shrink” tool to make him smaller

Step 6: Get Scratchy Ready To Draw

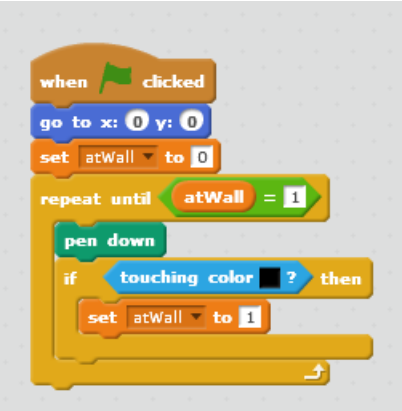
Next, let’s create the logic to have Scratchy start from the center of the Stage and walk out to the border, while drawing a line, and then stop, when he gets to the border.

- Create a new variable atWall
- Add a “When flag clicked” block
- Add a “go to x:0 y:0” block
- Add a “set atWall to 0” block
- Add a “repeat until atWall = 1” loop



But that is only part of what need to do for this step. Next, we need to put some blocks inside of the repeat loop.

- Add a “pen down” block
- Add an “if touching color black” block or whatever color you used for the border
- Inside of the if, add a “set atWall to 1” block



Step 7: Record Scratchy’s Steps

At this point, Scratchy really is not doing anything. If you run the program, nothing will happen, and that is because we have not added the logic for scratchy to move around in the stage. We will take care of that next.

Let’s think about what we want to do here, and write it down in English and then translate that into Scratch blocks. Let’s start with what happens when the left or right arrow gets pressed.

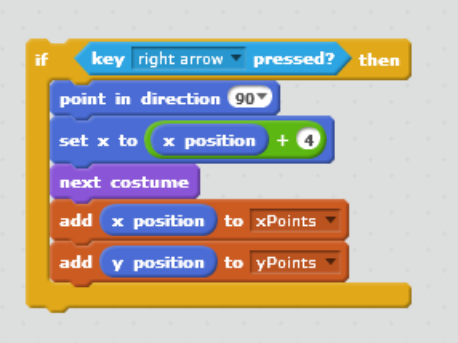
If right or left arrow pressed
    Point in the direction of the arrow
    Move in the direction of the arrow
    Make Scratchy look like he moved (Scratchy will look like he’s walking)
    Save the current position
Endif

It would be very similar when the up or down arrows are pressed, except it is easier because we don’t need to worry about how Scratchy looks. When going up or down, Scratchy does not move his legs.

If up or down arrow pressed
    Move in the direction of the arrow
    Save the current position
Endif

For the right arrow

- Add an “if right arrow pressed then” block
- Add a “point in direction 90 degress” block
- Add a “set x to x + 4” block
- Add a “next costume” block
- Add a “add x position to xPoints” block
- Add a “add y position to yPoints” block



Now you try it for the left arrow. The left arrow will be very similar, except you will minus 4 and point in -90 degrees. Why?

For the up arrow

- Add an “if right up pressed then” block
- Add a “set x to y + 4” block
- Add a “add x position to xPoints” block
- Add a “add y position to yPoints” block

Now you do the blocks for the down arrow. The down arrow will be very similar, except you will minus 4. Why?

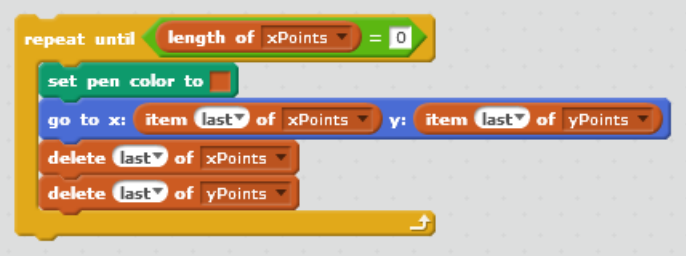
Note: Remember to set the rotation style on Scratchy to be horizontal, or he will turn upside down.

- Click the small “i” in the upper left corner
- The rotation style needs to be set to the arrows and not the circle

Step 8: Scratchy Retraces His Path

Okay, so now are getting somewhere. After finishing the previous step Scratchy now walks around, draws a line out to the border, and then stops when he reaches it. Now what we want to do is to use our parallel pair of stacks that contain the X, Y coordinates of all the steps that Scratchy took, and do those steps in reverse order.

- Add a “repeat until length of xPoints = 0” block
- Add a “set pen color to red” block or some other color than what Scratchy drew earlier
- Add a “go to x: item last of xPoints y: item last of yPoints” block
- Add a “delete last of xPoints” block
- Add a “delete last of yPoints” block



Step 9: Get Scratchy Ready to Start Again

In this final step, we just need to do some clean up. This needs to be the final step, outside of the previous repeat until.

- Add a “wait 3 secs” block
- Add a “set pen color to green” block
- Add a “pen up” block
- Add a “clear” block