

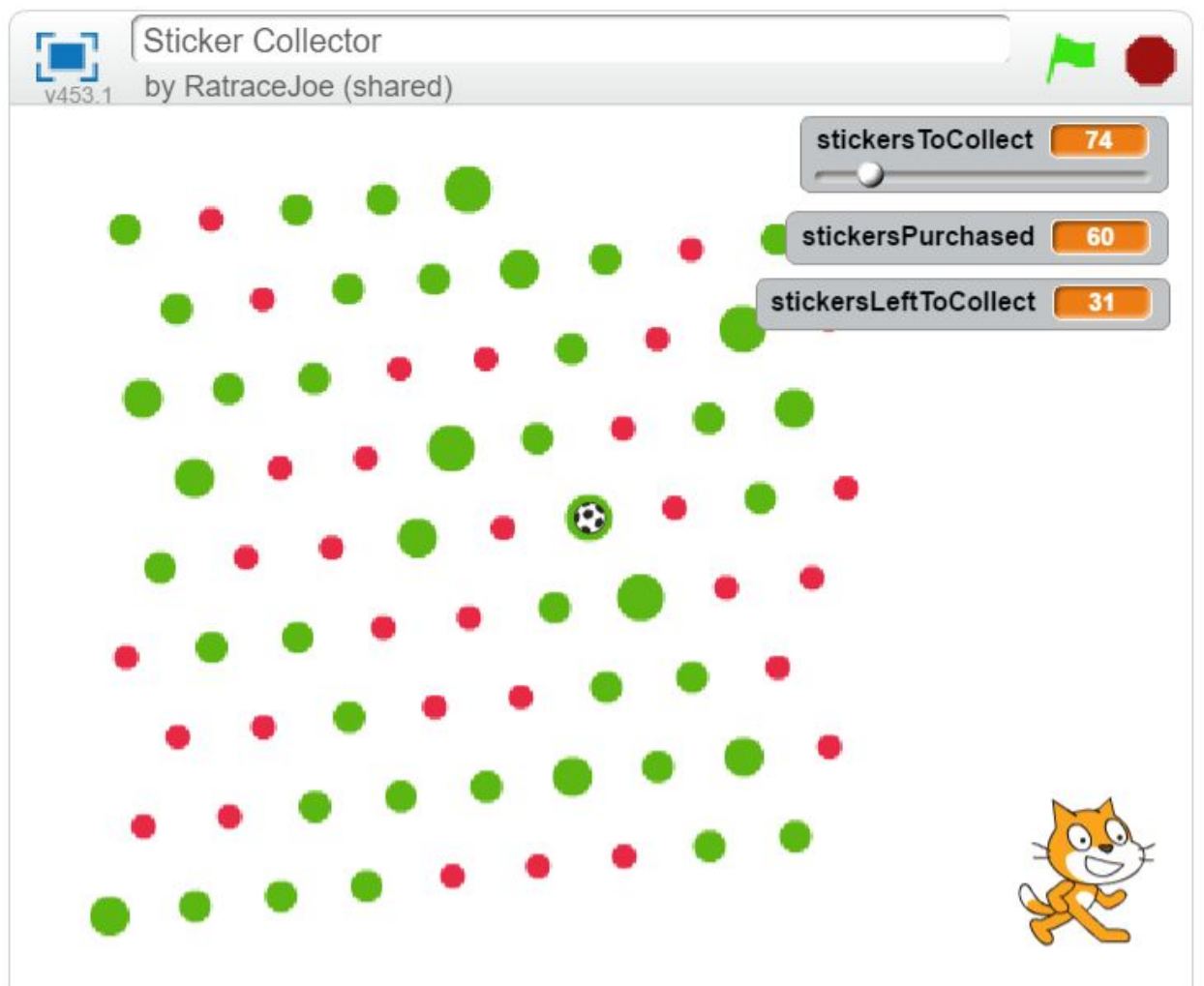
Scratch - Sticker Collection

Have you ever collected stickers? I remember collecting Premier League 94/95 stickers and swapping them with friends. We would invariably end up with a great big wad of 'swaps' and spend lunch and break time hunting for ones we did not have in our friends swap piles. After the sticker collection got so far, I would invariably end up with 4 or 5 copies of the same people. Do the sticker companies do this deliberately? Are the stickers truly random? Let us experiment using code to find out.

What will the Finished Project look like?

Here is a screenshot of the finished project working. The user will set the number of stickers in the collection using the slider bar, then hit the Green Flag to start the collection simulation.

The football sprite (called Pen) puts red dots down for each sticker, the colours them in green as those stickers are collected. When the player gets duplicates/swaps, the green dots get bigger.

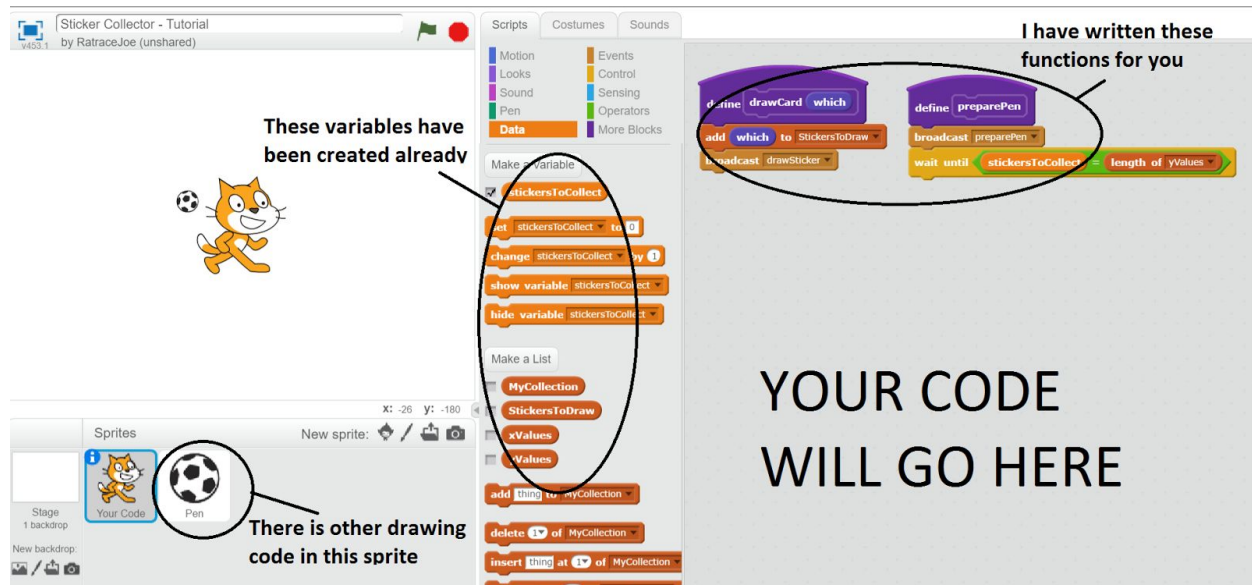


Remix My Base Project

The base project can be found here

<https://scratch.mit.edu/projects/140973726/>

Please 'remix' this project so that you have the drawing code built in. The project should look like this



You should not need to edit the code under the football sprite named **Pen**. You should also leave the two functions called **preparePen** and **drawCard** alone. You will be using those to draw the stickers as you collect them.

I have created some variables for you, they are as follows.

- **stickersToCollect** - This is the full size of the sticker collection. The larger the sticker collection the longer it will take to collect them all.
- **MyCollection** - This is a list of the stickers that you have managed to collect. At the start of the program we will be building this list.
- **StickersToDraw** - This is used by my code to tell the pen which stickers should be redrawn. *You will not have to interact with this list.*
- **xValues, yValues** - These are used by the drawing code to work out where to put the dots. *You will not have to interact with either of these lists.*

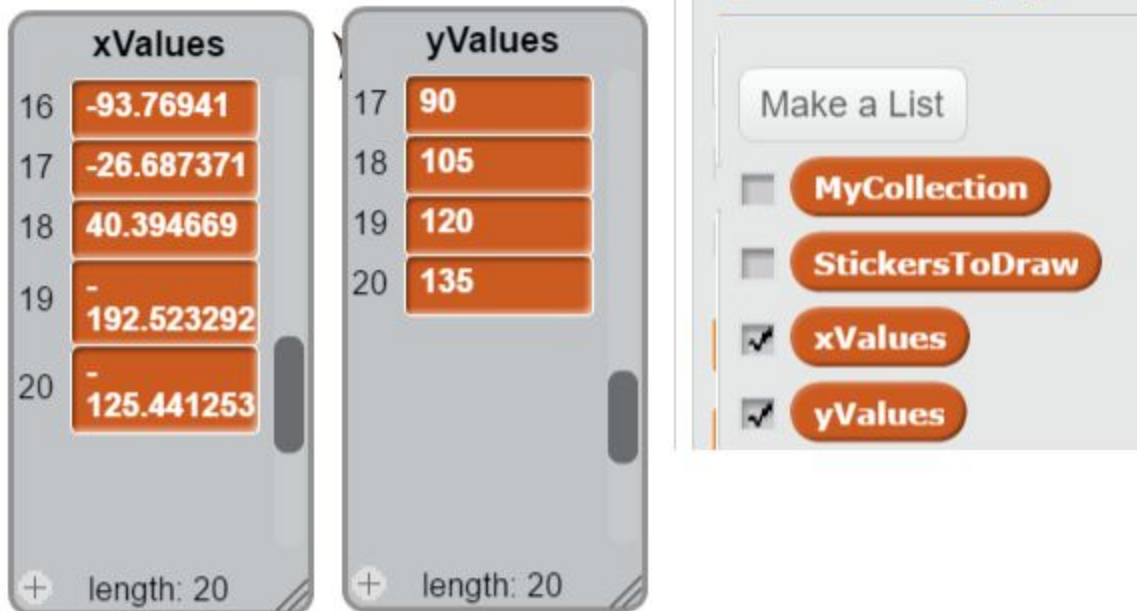
Get the Simulation Started

We will be using the Green Flag to kick off proceedings, so from **Control** drag a **When <Green Flag> is clicked** event handler into our code area.

Next we need to call **preparePen**, this should cause the pen to fill in the xValues and yValues.



If you want to see this working, you can tick the boxes next to the **xValues** and **yValues** lists under **Data**. Then click on the Green Flag. Once you have verified this, untick the boxes to hide those lists.

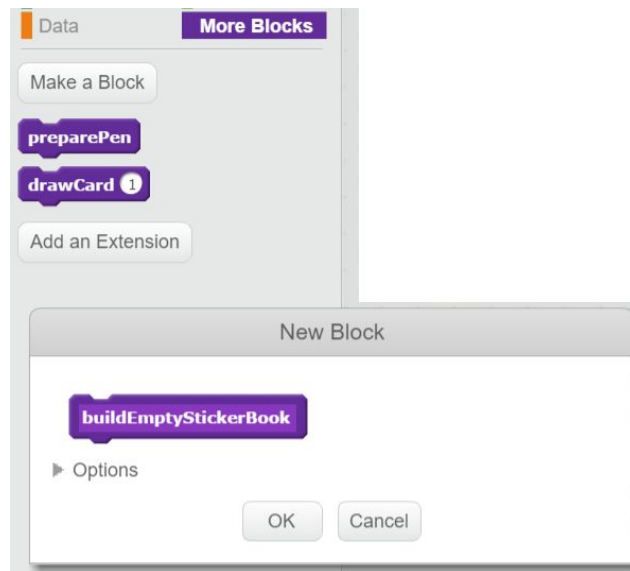


Build Empty Sticker Book

Now we need to build our empty sticker book. I am going to put this into a separate **Block** (or function) called **buildEmptyStickerBook**.

Create Empty Function

Under **More Blocks** click on **Make a Block** and call the new block **buildEmptyStickerBook** as shown.



To call this function, we should drag the new **buildEmptyStickerBook** block from **More Blocks** into our green flag handler.

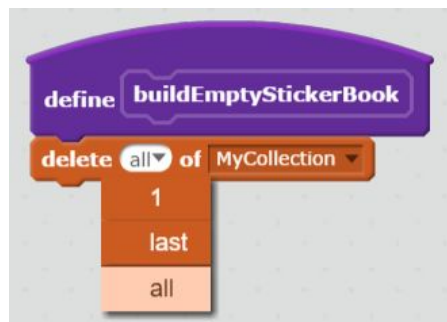


Build Empty Collection

We will be using our list **MyCollection** to hold the list of stickers we have. We will be using the index of the list items as the sticker numbers, so for a collection of 4 stickers the list would start off looking like this.

Sticker Number	Number Collected
1	0
2	0
3	0
4	0

Inside our **buildEmptyStickerBook** function, we should clear out any existing collection. From **Data** drag a **delete <all> of <MyCollection>** block into our function.



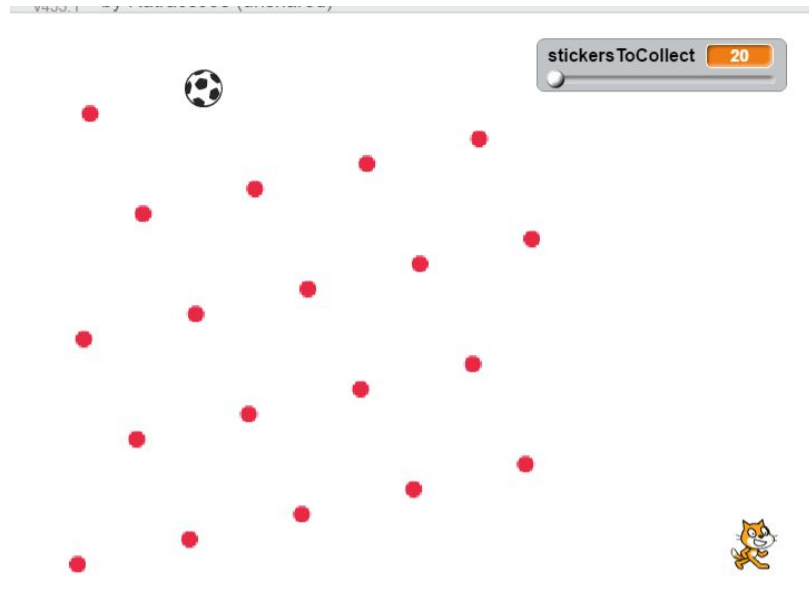
We now need to loop round for every sticker and add an item to the collection, then draw it.

- From **Control** drag a **repeat <>** block into our function
- From **Data** drag a **stickersToCollect** into our loop
- From **Data** drag a **add <0> to MyCollection** inside the loop
- From **More Blocks** drag **drawSticker <>** block into the loop
- We want to draw the sticker we just added, so from **Data** drag **length of <MyCollection>** into the **drawSticker** function.



Time to Test

Try clicking on the green flag and you should find the **pen** sprite draws out red dots to represent your collection!



You are now ready to start collecting stickers.

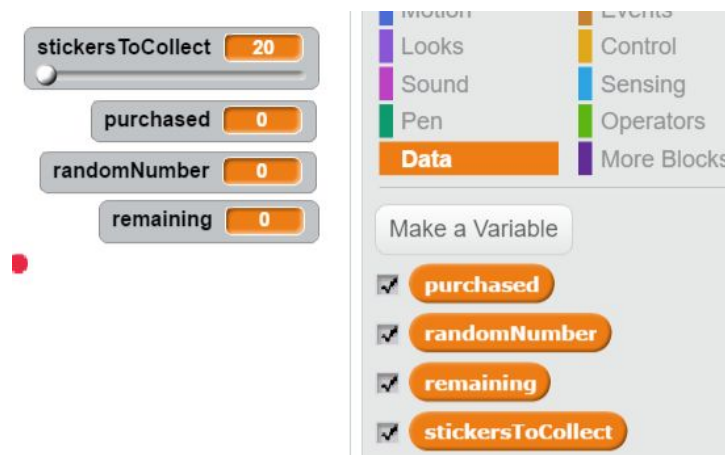
Create Variables to Track Progress

Now that we have an empty collection, it is time to start filling it up. We need to create some new variables to track our progress though. The main purpose of the program is to show just how many stickers you have to buy in order to complete a collection.

Under **Data** click on **Make a Variable** and make the following variables.

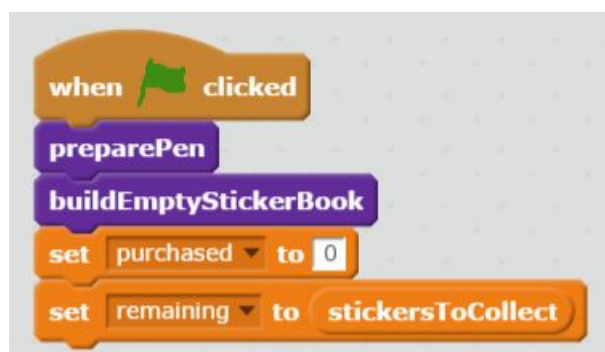
- **remaining** - This will be used to track how many stickers we still need to collect.
- **purchased** - This will be used to count the number of stickers purchased during the simulation.
- **randomNumber** - This will be used as a temporary variable to store the sticker received in a particular purchase.

Your variables should look like this.



We will initialise two of the variables in our green flag handler now.

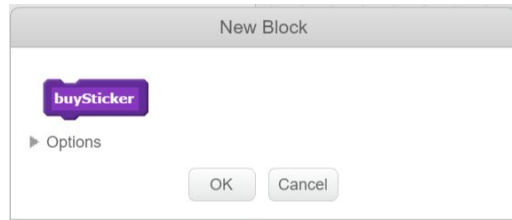
- From **Data** drag a **set <purchased> to <0>** into the Green Flag handler
- From **Data** drag a **set <remaining> to <>** block in
- From **Data** drag **stickersToCollect** into the setter that we just pulled in.



Create an empty Buy Sticker Function

We will create a separate code block for buying a single sticker. We can call this code block until we have no more stickers left to collect.

Under **More Block** click on **Make a Block** and give it the name **buySticker**.



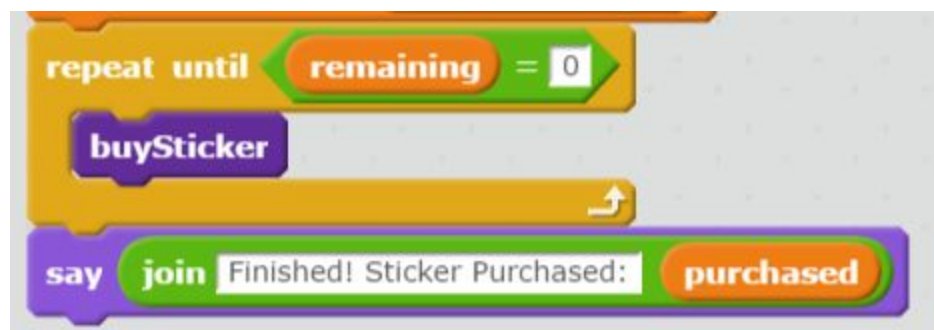
We will be calling this function in a loop

- From **Control** drag a **repeat until <>** block into our Green Flag handler.
- From **More Blocks** drag the **buySticker** function into our loop.
- Use a combination of **Operators** and the value of **remaining** to detect that we have collected all the stickers.



When the collection is finished, we should print out how many stickers were purchased to draw attention from the user. From **Looks** drag **say <Hello!>** block after the loop.

Use a combination of **Operators** -> **Join** and the value of **Data** -> **purchased** from to tell the user how many stickers were purchased.



Write the Buy Sticker Function

This is the big one, the one we have all been waiting for. Time to buy some stickers. This function will be called each time the user simulates buying a single sticker. So we must increment the number of stickers **purchased**, pick a **randomNumber** as our next sticker, then add it to our collection. Finally we should **draw** the updated sticker.

- From **Data** drag a **change <purchased> by <1>** into the **buySticker** function.
- From **Data** drag a **set <randomNumber> to <>** into the function
- From **Operators** drag a **random from <1> to <>** into the setter
- From **Data** drag the value of **stickersToCollect** into the second box of the random generator.

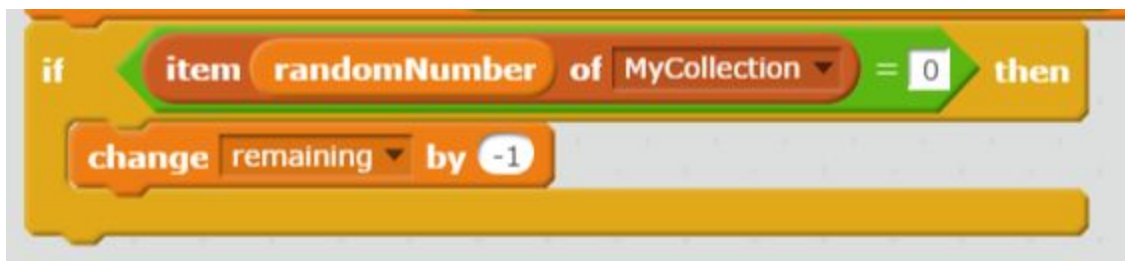
The code will look like this.



The **randomNumber** now contains a valid sticker number. We should detect if this is a sticker we already have. If it is, then we should decrement (reduce by 1) the **remaining** variable.

- From **Control** drag an **if <>** block in
- From **Operators** drag an equality operator in **<> = <0>**.
- From **Data** drag **item <> of MyCollection** into the left hand box of the equality operator.
- From **Data** drag the value of **randomNumber** into the **item <>** selector.
- From **Data** drag a **change <remaining> by <-1>** to decrement the number of stickers remaining.

The code should look like this.



Lastly we need to increase that sticker in our collection and draw the sticker. To increase the count of that sticker, we must replace the value in **MyCollection** with its current value + 1.

- From **Data** drag a **replace item <> of <MyCollection> with <>** after our **if <>** block
- From **Data** drag **randomNumber** into the item selection
- From **Operators** drag an addition block **<> + <1>** into the **<thing>** part of our **replace** block.
- From **Data** drag an **item <> of <MyCollection>** into the left hand block of our new addition block.
- From **Data** drag the value of **randomNumber** into this new selector.

The code should look like this.

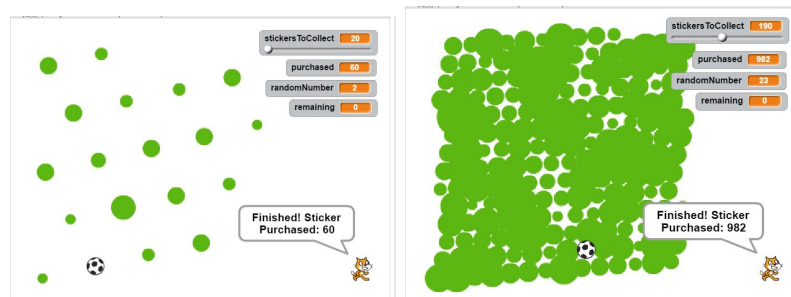


Lastly we shall call **drawSticker** and pass in the **randomNumber**. The code for that will look like this.



Time to Test!

Try changing the value of **stickersToCollect** using the slider bar and then run the simulation a few times. You should see stuff like this. Note that to collect 190 stickers I had to buy 982 in the simulation on the right! So do publishers need to make some stickers rare? Probably not.



CONGRATULATIONS, YOU HAVE FINISHED THE STICKER COLLECTOR