

## Jazzy jumpers

Create a memory game involving jazzy jumpers



### Step 1 Introduction

---

Create a memory game involving jazzy jumpers.

#### What you will make

Below is an example of the game you will be making. Click the flag to begin. Memorise the jazzy jumper, then click on each part of the jumper template you're shown to change the colours and recreate the jumper you've memorised. When you think you've got it, press the button to see if you were right or not.

#### What you will need

##### Hardware

- A computer capable of running Scratch 3

##### Software

- Scratch 3 (either **online** (<https://scratch.mit.edu/projects/editor/>) or **offline** (<https://scratch.mit.edu/download/>))

#### What you will learn

This project covers elements from the following strands of the **Raspberry Pi Digital Making Curriculum** (<http://rpf.io/curriculum>):

- **Use basic programming constructs to create simple programs** (<https://curriculum.raspberrypi.org/programming/creator/>).
- **Design basic 2D and 3D assets** (<https://curriculum.raspberrypi.org/design/creator/>).



### Additional information for educators

If you need to print this project, please use the **printer-friendly version** (<https://projects.raspberrypi.org/en/projects/jazzy-jumpers/print>).

Download the finished project code at:

<http://rpf.io/p/en/jazzy-jumpers-get> (<http://rpf.io/p/en/jazzy-jumpers-get>).

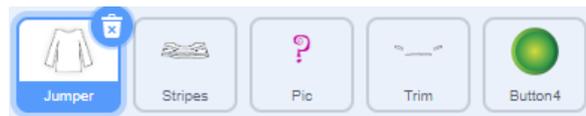
## Step 2 Parts of the jumper

---

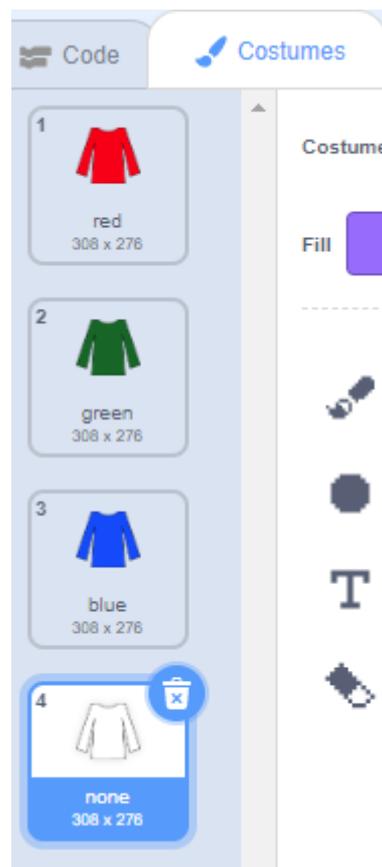
- Open the starter project online at [jumpto.cc/jazzy-go](http://jumpto.cc/jazzy-go) (<http://jumpto.cc/jazzy-go>) or use the **offline starter project** (<https://rpf.io/p/en/jazzy-jumpers-go>).

If you are using Scratch online and you have a Scratch account, click on **Remix** in the top right-hand corner to save a copy of the project to your account.

- Take a look at the sprites in the project: you have five sprites in total, one for each of the four parts which make up the jumper and one for the button.



- Click on the **Jumper** sprite and then on the Costumes tab. You will see three coloured costumes and a white “none” costume.



- If you like, you can use the **fill** tool to change the colours of the costumes. Change the names of the costumes too so that they match the new colours.
- However, make sure to leave the “none” costume at the bottom, and make sure it remains completely white. This costume is shown when the player is trying to remember the jumper.
- If you want to, you can change the colours or pictures of all four sprites which make up the jumper: **Jumper**, **Stripes**, **Trim**, and **Picture**.

## Step 3 New jumper

---

In this project there are a lot of sprites, and a good way of telling lots of sprites what to do is to send a **broadcast**.

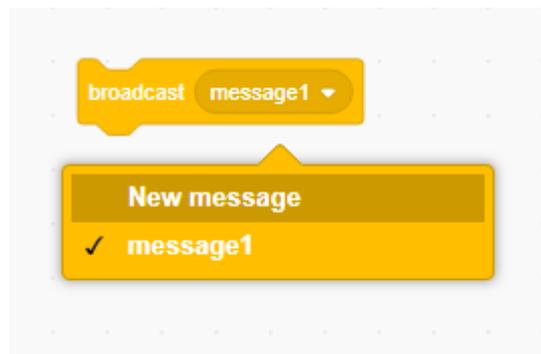
### Broadcast a message in Scratch

A **broadcast** is a way of sending a message which can be heard by all sprites. Think of it like an announcement made over a loudspeaker.

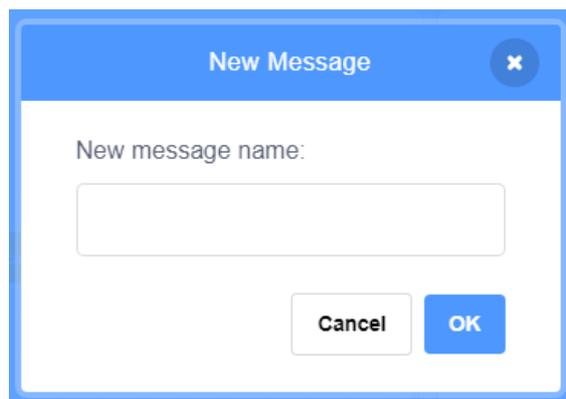
**Broadcasting spells:** Use the magic wand to click on the buttons and cast spells. What does each spell do to the characters? **See inside** (<https://scratch.mit.edu/projects/518413238/editor>).

You can create a message to be **broadcast**. The message text can be anything you like, but it is useful to give it a sensible description.

- Find the **broadcast** block under **Events**
- Select **New Message** in the drop-down menu.

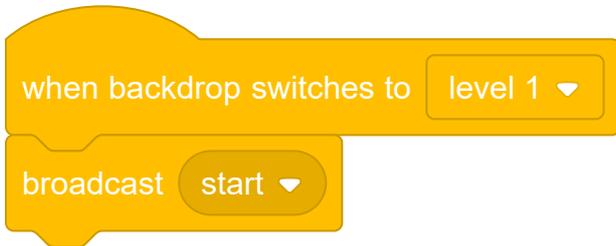
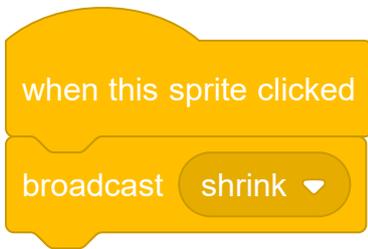


- Then type your message



### Send a broadcast

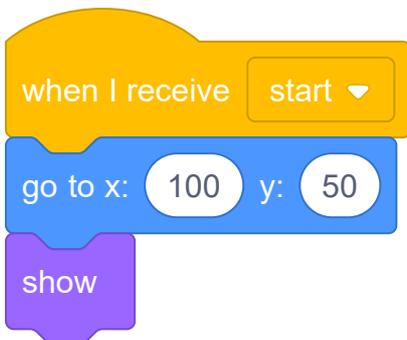
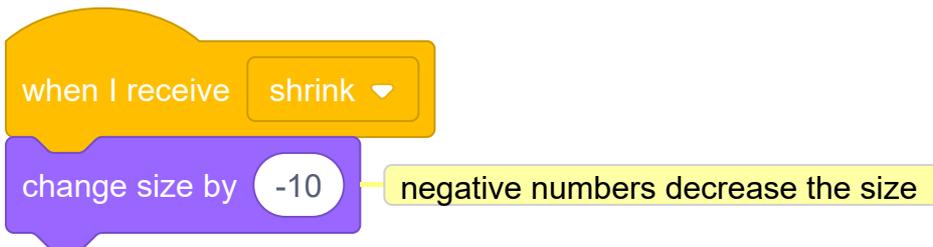
You can decide when to **broadcast** your message. For example:



### Receive a broadcast

Sprite can react to a **broadcast** by using a **when I receive** block. Multiple sprites can respond when they receive the same message.

You can add blocks below a **when I receive** block to tell the sprite(s) what to do when they receives the message.



Let's create a broadcast which tells the various sprites forming the jumper to each choose a random costume, so that they form a new jumper.

- Click on the **Stage** and make sure you are on the **Scripts** tab.

The **Stage** is going to send a broadcast called **new jumper** to all of the other sprites. Think of this like a person with a megaphone giving instructions to lots of people at once.

- Add the following code to the **Stage**. If you can't remember how to create a new broadcast message, have a look at the information in the "Broadcast a message in Scratch" section above to refresh your memory.

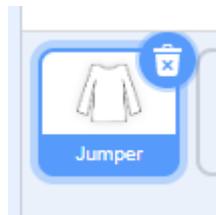


- Click the green flag. What happens? Do you see a new jumper appear?

**i Answer**

Absolutely nothing happens! This is because you haven't told the sprites making up the jumper what to do when they hear this broadcast, so they won't do anything yet. Let's tell them what to do.

- Click on the **Jumper** sprite in the **Sprites** panel.

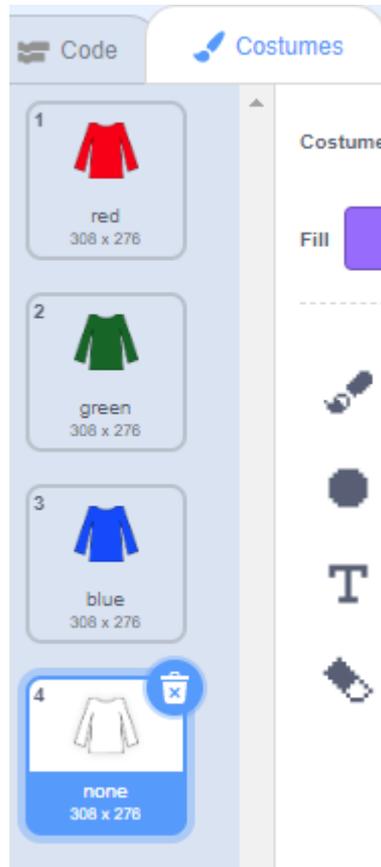


- Add this block to the scripts for the **Jumper** sprite:



You can tell the **Jumper** sprite exactly what to do when it hears the broadcast by attaching the blocks you want to happen below this block.

When the **Jumper** sprite hears the broadcast, it should randomly pick one of the coloured costumes – but not the white one.



- Attach some code to your **when I receive** block to tell the **Jumper** sprite to pick a random costume out of the first three only.



- Click the green flag. Does the **Jumper** sprite colour change? Click the green flag a few more times to check.

The colour might not change every single time you click the green flag, because sometimes the randomly chosen costume will be the same as the previous one.

- Now add some code to the **Stripes**, **Picture**, and **Trim** sprites so that, when they hear the broadcast **new jumper**, they also choose a random costume.

### I need a hint

The code you will need to use is exactly the same for all of the sprites!

You can save time by duplicating the code. Drag all the code attached to the **when I receive** block on top of one of the other sprites in the **Sprites** panel, and let go to drop it.

Look inside that sprite, and you should see a copy of the code.

As you add more parts to the jumper, it becomes much harder for the player to memorise each one, making the game more of a challenge!

## Step 4 Hide the jumper

---

The aim of the game is for the player to memorise the jumper, so you need to make it disappear after a few seconds.

- Go back to the **Stage** and add some code so that, when the **Stage** receives the **new jumper** broadcast, it waits for three seconds and then broadcasts a new **hide jumper** message.

### I need a hint

Here is the code you will need to add to the stage:



- Switch to the **Jumper** sprite. Start a new block of code so that, when this sprite hears the **hide jumper** broadcast, it switches to the blank "none" costume.



- Add similar code to the other sprites making up the jumper.
- Click the green flag and test your code. Does the jumper switch to all white with a question mark picture three seconds after the green flag is clicked?

## Step 5 What was on the jumper?

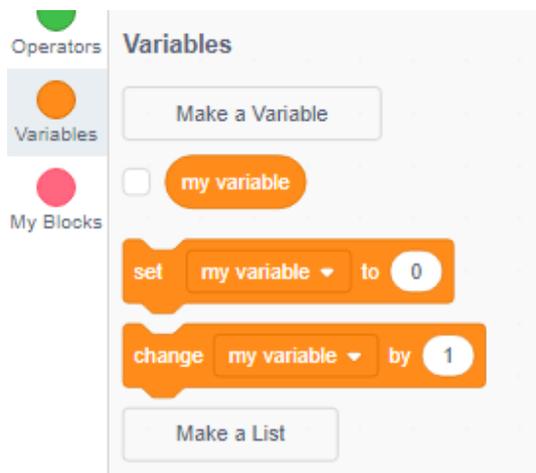
So far the program creates a random jumper for the player to remember, and then hides it. However, we've made a mistake! We didn't note down anywhere which costumes were randomly chosen, so how will we know if the player has recreated the jumper correctly?

To save information you will need to use later on, you can create **variables**.

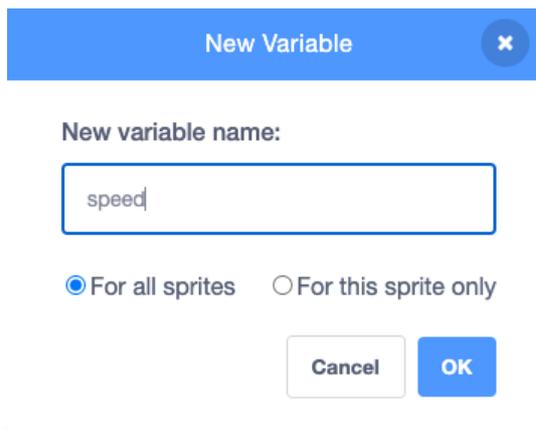
- Click on the **Jumper** sprite and create a variable called **jumper**.

### Add a variable in Scratch

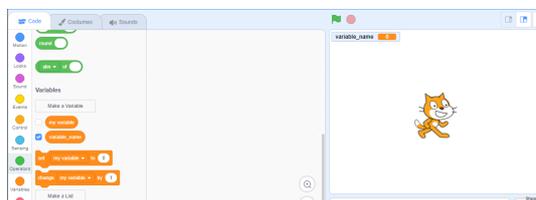
- Click on **Variables** in the Code tab, then click on **Make a Variable**.



- Type in the name of your variable. You can choose whether you would like your variable to be available to all sprites, or to only this sprite. Press **OK**.



- Once you have created the variable, it will be displayed on the Stage, or you can untick the variable in the Scripts tab to hide it.



You're going to save the random costume that was chosen for the **Jumper** sprite inside this variable, so that the program remembers it for later.

- Set the value of the **jumper** variable as the green **pick random 1 to 3** block to save the number that was chosen.
- Then set the costume to the value of the variable **jumper**, which now holds a random number between 1 and 3.
- Click the green flag and check that your **Jumper** sprite still chooses a random colour each time.
- Create three more variables, one named after each other sprite making up the jumper.
- Add some code to the other three sprites so that their costume number is saved in the variable named after it. Your code will be slightly different for each sprite, because each sprite will have its own variable.
- Make sure that you hide the variables from the stage by unselecting the boxes in the variables menu, otherwise the game will be very easy for your player!

## Step 6 Recreate the jumper

---

The player needs to be able to recreate the jumper they saw by clicking on the different parts until they match the original jumper.

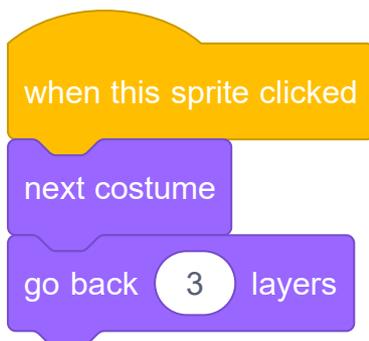
- Switch to the **Jumper** sprite.
- Add some code so that, when the sprite is clicked on, it will display the next available costume.



- Add the same code to all four sprites making up the jumper.
- Test your code by waiting for the blank jumper to display, and then clicking on the different parts. Do they change colour each time you click?

### My jumper went blank!

You might notice that sometimes when you click on the **Jumper** sprite, you accidentally drag it a little bit too, and that means it covers up the **Picture** and the **Stripes** sprites. Add the following block at the end to send your **Jumper** sprite behind the other sprites when it is clicked. Then you can always see the other sprites, even if you accidentally drag the **Jumper** sprite.



## Step 7 The correct jumper

---

When the player thinks they have recreated the jumper correctly, they should click on the button to be told whether they were correct or wrong.

- Click on the **Button** sprite and look at its costumes.

The **correct** and **wrong** costumes will be used to display whether the player's jumper was the same as the one that was displayed at the start.

- Add some code to the **Button** sprite so that, when it is clicked, it broadcasts a new message called **check**.

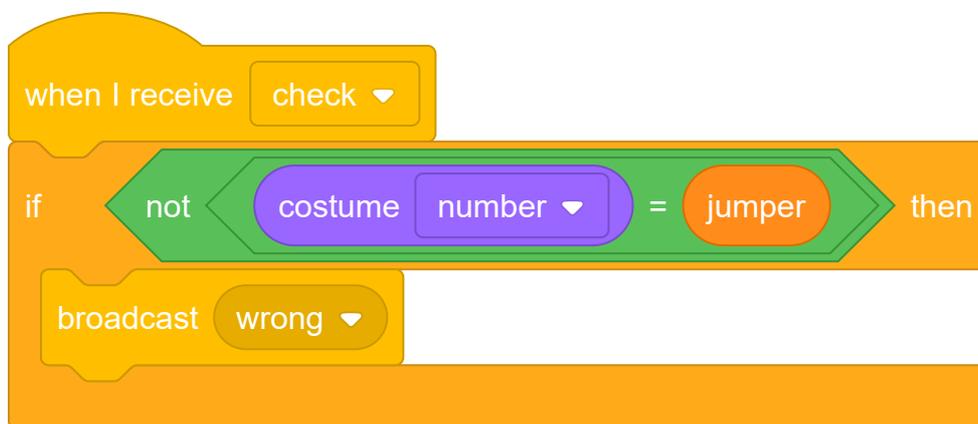
### I need a hint

Here is the code you will need to add to the **Button** sprite:

When the other sprites hear the message **check**, they should each check whether the current **costume number** is the same as the costume number saved in the **variable** named after them.

If the costume numbers do **not** match, the sprites should broadcast the message **wrong**.

- Switch to the **Jumper** sprite and add some code to check whether the player selected the correct colour. If they were wrong, broadcast **wrong**.

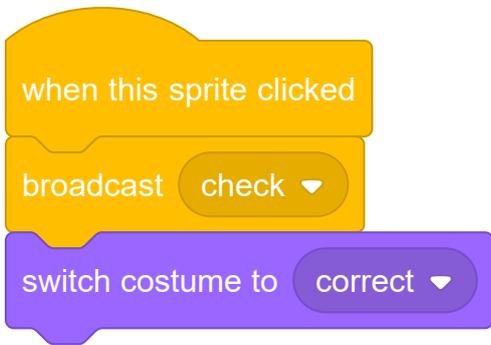


Note that you will need two green blocks: one for **not**, and another one inside it for **=**.

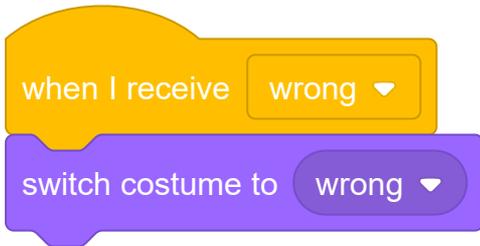
- Add similar code to the other three sprites making up the jumper, but be careful to check the current costume number against the correct variable for that sprite.
- Now switch back to the **Button** sprite.

We will assume the player is correct unless we receive a broadcast saying they were wrong.

- Add a block after you broadcast **check** to switch to the **correct** costume.



- Also add blocks to switch to the **wrong** costume if the **wrong** broadcast is received.



If any of the sprites making up the jumper broadcasts that its costume was wrong, the player will see the word 'wrong'. If not, they will see the word 'correct'.

## Step 8 Challenge: enhance the game

---

- When the player has pressed the button and seen the result, can you show them another jumper to memorise so that they can keep playing the game over and over again?

### I need a hint

The broadcast `new jumper` starts the process all over again. Perhaps you could wait for a few seconds after showing the result, and then broadcast this message?

- Can you add a `winning streak` variable to keep track of how many correct jumpers in a row the player has remembered? If they get one wrong, the streak counter should go back to zero.
- Add a high score variable to keep track of the longest streak.



### Create a high score in Scratch

It's fun to keep track of a high score in a game.

Let's say you have a variable called `score`, which gets set to zero at the beginning of each game.

Add another variable called `high score`.

At the end of the game (or whenever you want to update the high score), you'll need to check whether you have a new `high score`.

- Add more costumes to each of the sprites so that there are more possible colours to remember. If you do, don't forget to update the code that chooses a random costume so that it chooses between all the available costumes including your new ones.
- Add another sprite to represent a different part of the jumper to make the game even harder. Perhaps you could have sleeve colours, or a circle on the tummy?

---

Published by Raspberry Pi Foundation (<https://www.raspberrypi.org>) under a Creative Commons license (<https://creativecommons.org/licenses/by-sa/4.0/>).

View project & license on GitHub (<https://github.com/RaspberryPiLearning/jazzy-jumpers>).