Projects

Balloons

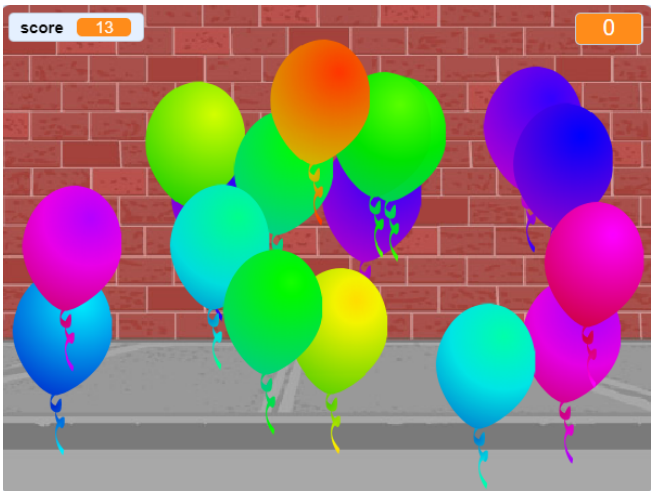
Learn how to make a balloon-popping game!

Scratch

Step 1 Introduction

You are going to make a balloon-popping game!

What you will make



What you will need

Hardware

- A computer capable of running Scratch

Software

- Scratch 3 (either **online** (http://rpf.io/scratchon) or **offline** (http://rpf.io/scratchoff))

What you will learn

- How to use animation to make sprites move
- How to use random numbers
- How to draw sprites
- How to create clones of a sprite

Additional information for educators

You can **find the solution for this project here** (http://rpf.io/p/en/balloons-get).

Step 2 Animating a balloon

Open a new Scratch project.

Online: open a **new online Scratch project** (http://rpf.io/scratch-new).

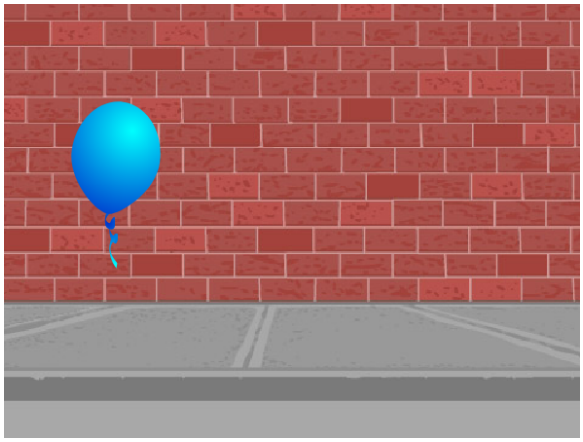
If you have a Scratch account you can make a copy by clicking **Remix**.

Offline: open a new project in the offline editor.

If you need to download and install the Scratch offline editor, you can find it at **rpf.io/scratchoff** (http://rpf.io/scratchoff).

Delete the cat sprite.

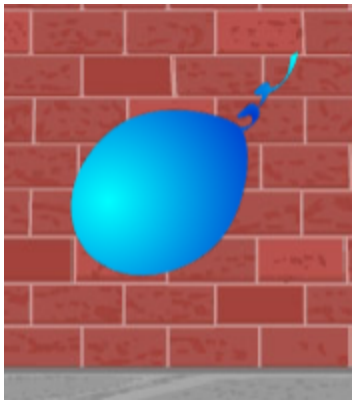
Add in a new balloon sprite, and a suitable stage backdrop.



Add this code to your balloon, so that it bounces around the screen:

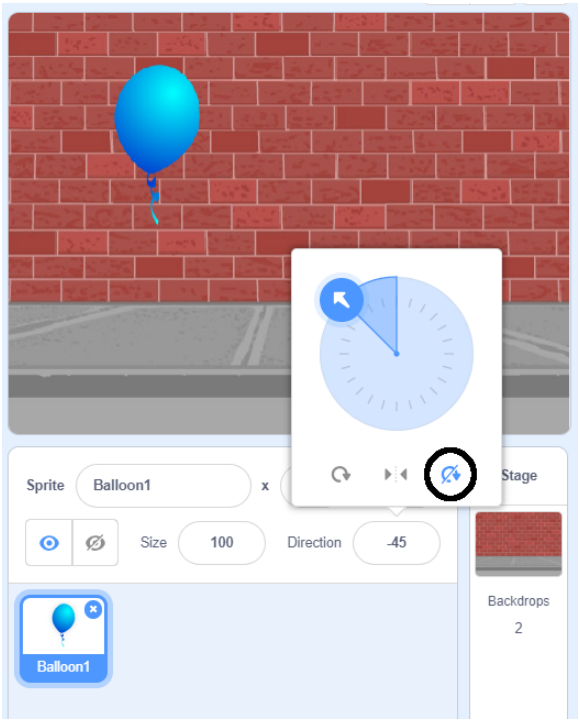
Test out your balloon. Does it move too slowly? Change the numbers in your code if you want to speed it up a bit.

Did you also notice that your balloon flips as it moves around the screen?



Balloons don't move like this! To fix this, click on the balloon sprite icon, and then click the direction.

In the 'rotation style' section, click 'Do not rotate' to stop the balloon rotating.



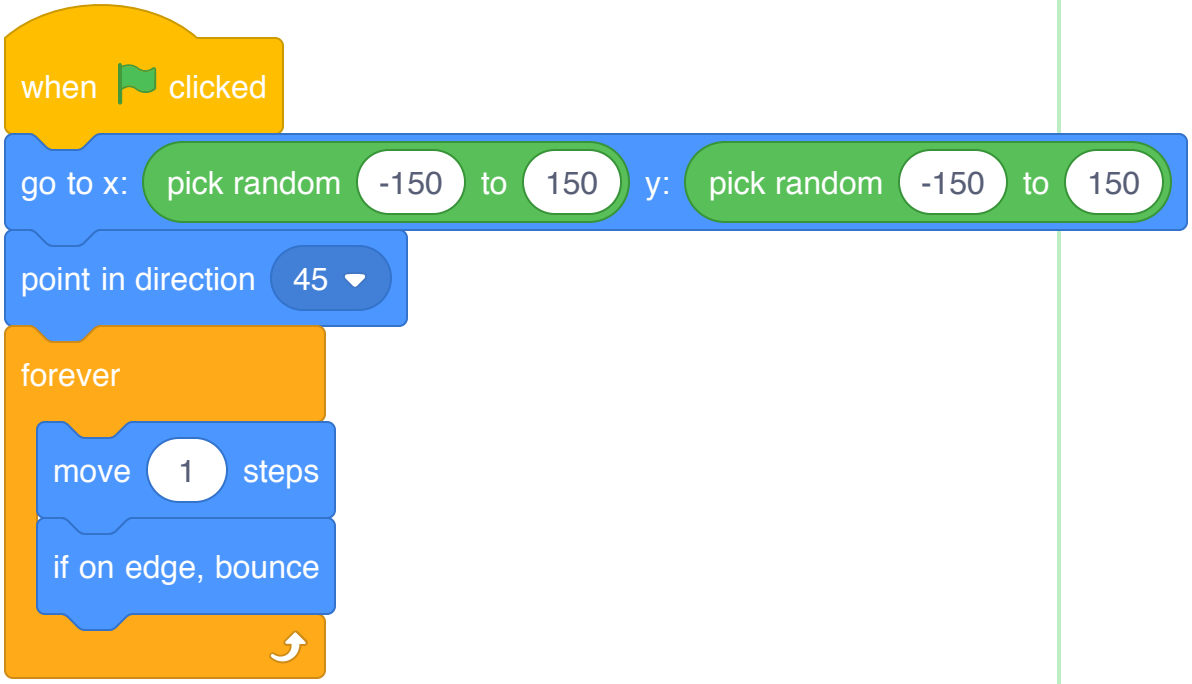
Test your program again to see if the problem is fixed.

Step 3 Random balloons

With the code you have now, your balloon will always start in the same place and move in the same path.

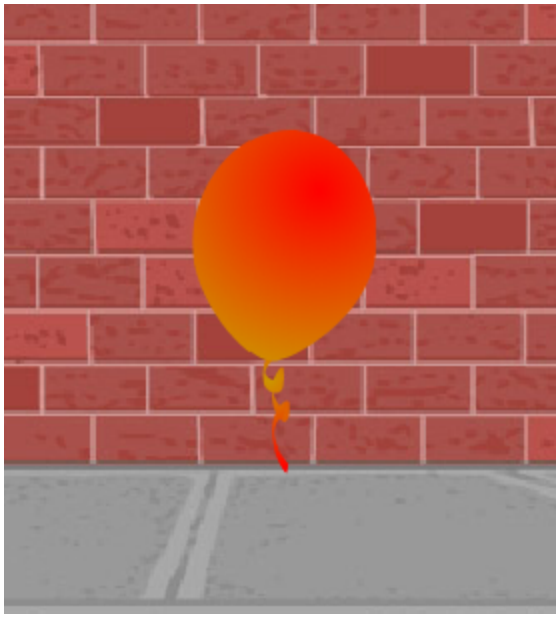
Click the flag a few times to start your program, and you'll see it's the same every time.

Instead of using the same x and y position each time, you can let Scratch `pick a random number` instead. Change your balloon's code, so that it looks like this:

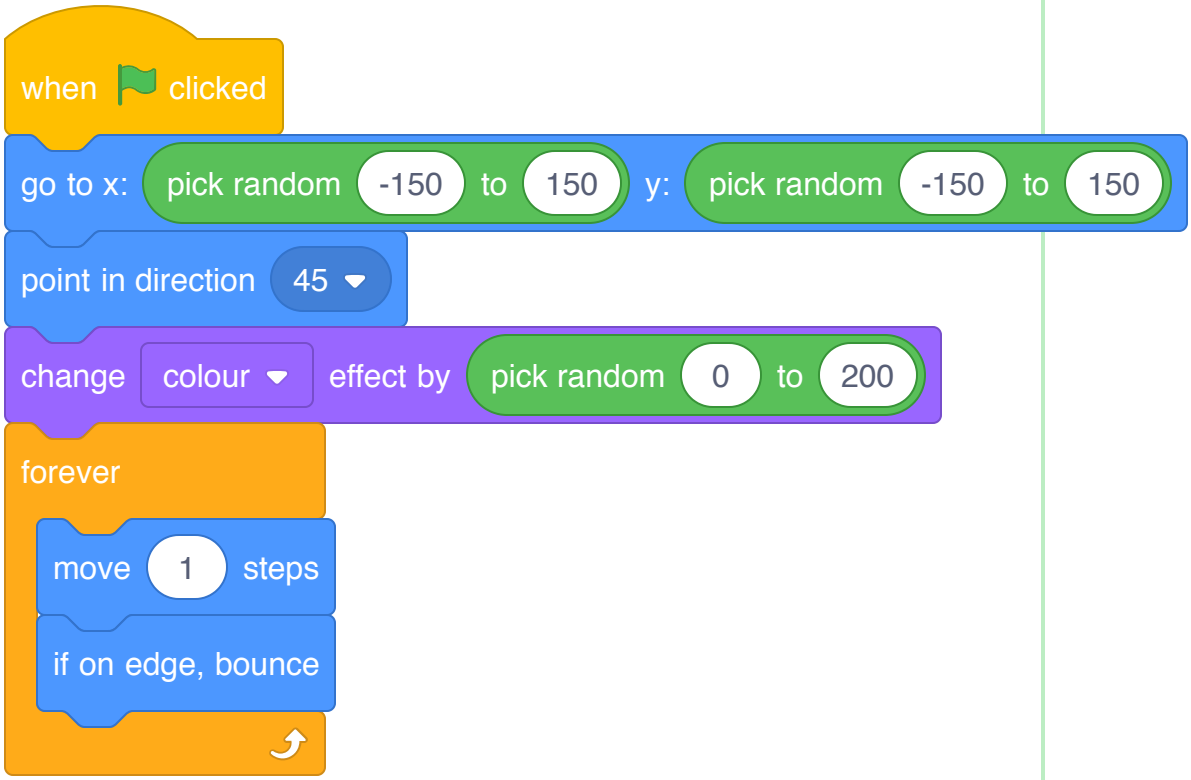


If you click the green flag a few times, you should notice that your balloon starts in a different place each time.

You could even use a random number to choose a random balloon colour each time:



You code should look like this:



What happens if this code is put at the start of your program? Does anything different happen if you put this code *inside* the `forever` loop? Which do you prefer?

Challenge!

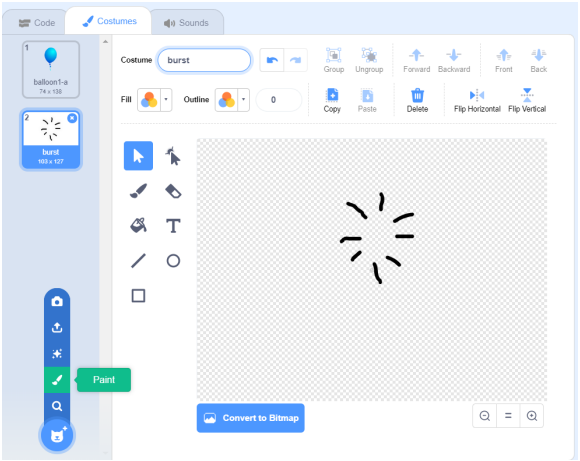
Challenge: More randomness

Can you make your balloon start by pointing in a random direction (between -90 and 180)?

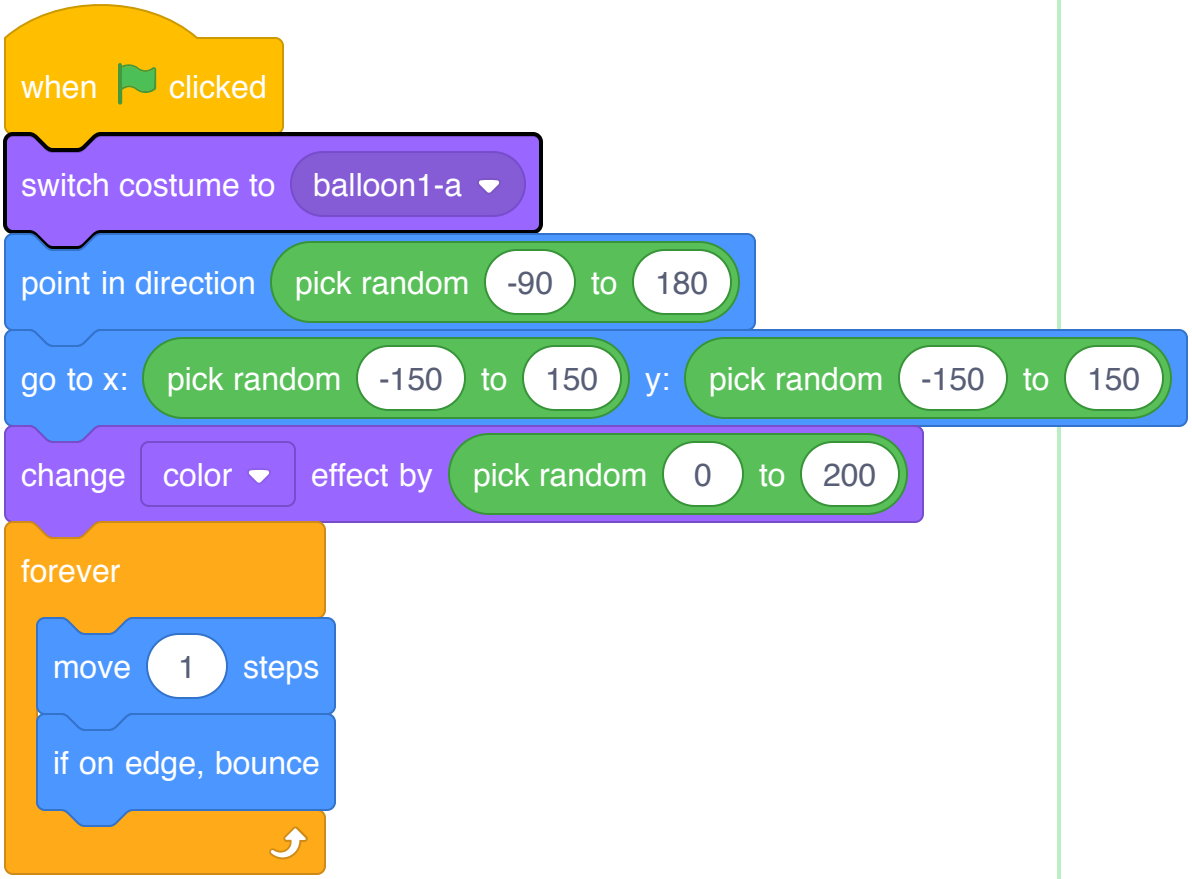
Step 4 Popping balloons

Lets allow the player to pop the balloons!

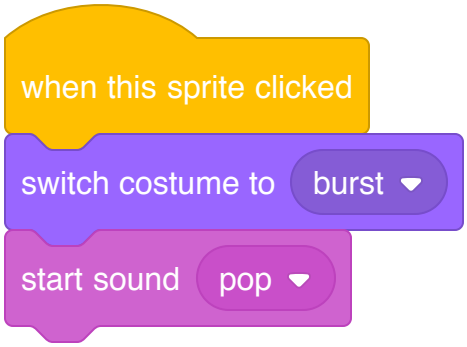
Click on your balloon sprite, and then click the **Costumes** tab. You can delete all of the other costumes, just leaving 1 balloon costume. Add a new costume, by clicking **Paint new costume** and create a new costume called `burst`.



Make sure that your balloon switches to the right costume when the game starts. Your code should now look like this:



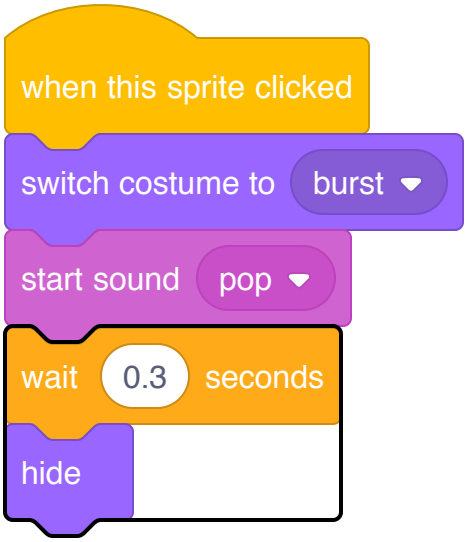
To allow the player to burst a balloon, add this code:



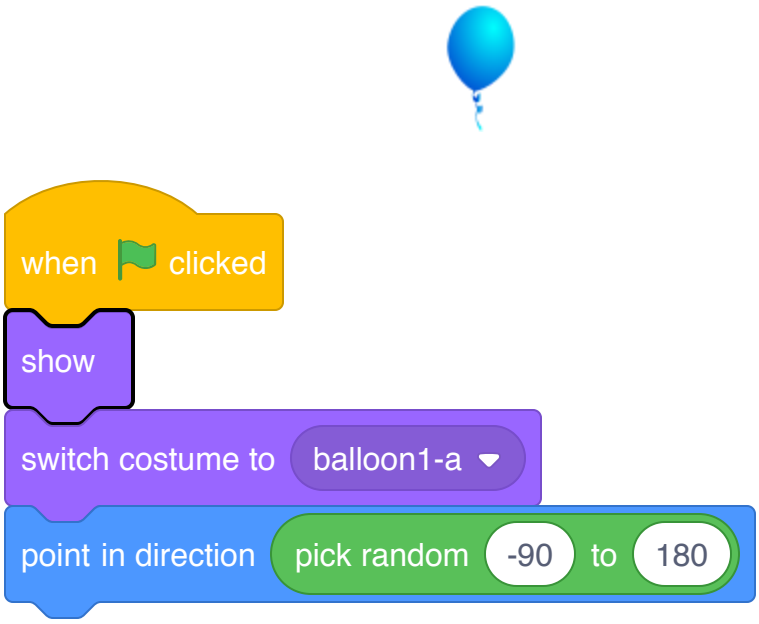
Test out your project. Can you pop the balloon? Does it work as you expected?

You'll need to improve this code, so that when the balloon is clicked, it shows the `burst` costume for a short time, and is then hidden.

You can make all of this happen by changing your balloon `when sprite clicked` code to this:



Now that you're deleting the balloon when it's clicked, you'll also need to add a `show` block to the start of the `when flag clicked` code.



Try popping a balloon again, to check that it works properly.

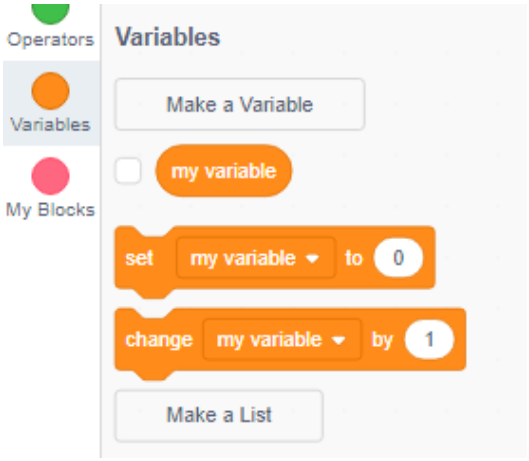
Step 5 Adding a score

Let's make things more interesting by keeping score.

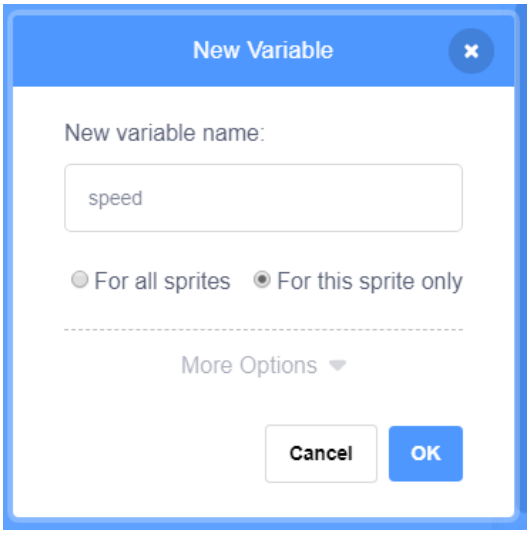
To keep the player's score, you need a place to put it. Create a new `variable` called `score`.

Add a variable in Scratch

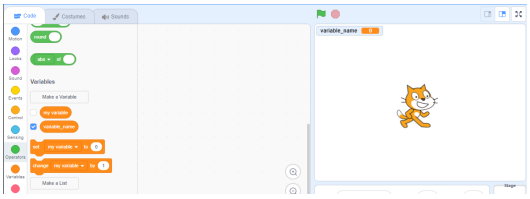
- Click on **Variables** in the Code tab, then click on **Make a Variable**.



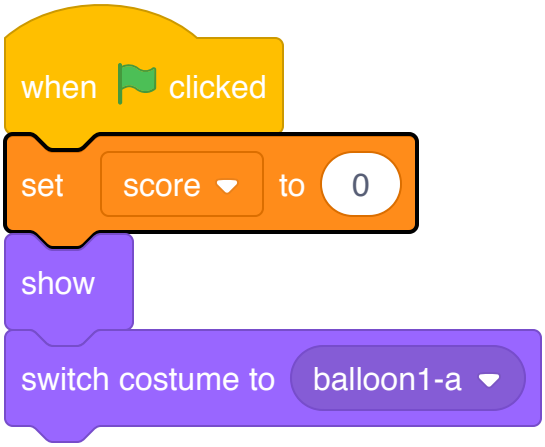
- Type in the name of your variable. You can choose whether you would like your variable to be available to all sprites, or to only this sprite. Press **OK**.



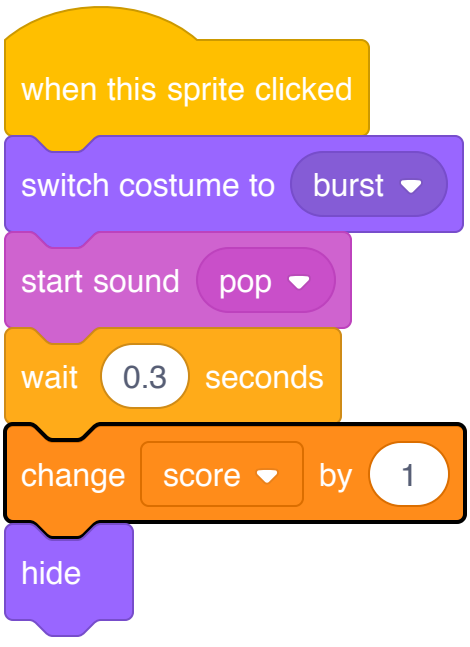
- Once you have created the variable, it will be displayed on the Stage, or you can untick the variable in the Scripts tab to hide it.



When a new game is started (by clicking the flag), you should set the player's score to 0. Add this code to the top of the balloon's `when flag clicked` code:



Whenever a balloon is popped, you need to add 1 to the score:



Run your program again and click the balloon. Does your score change?

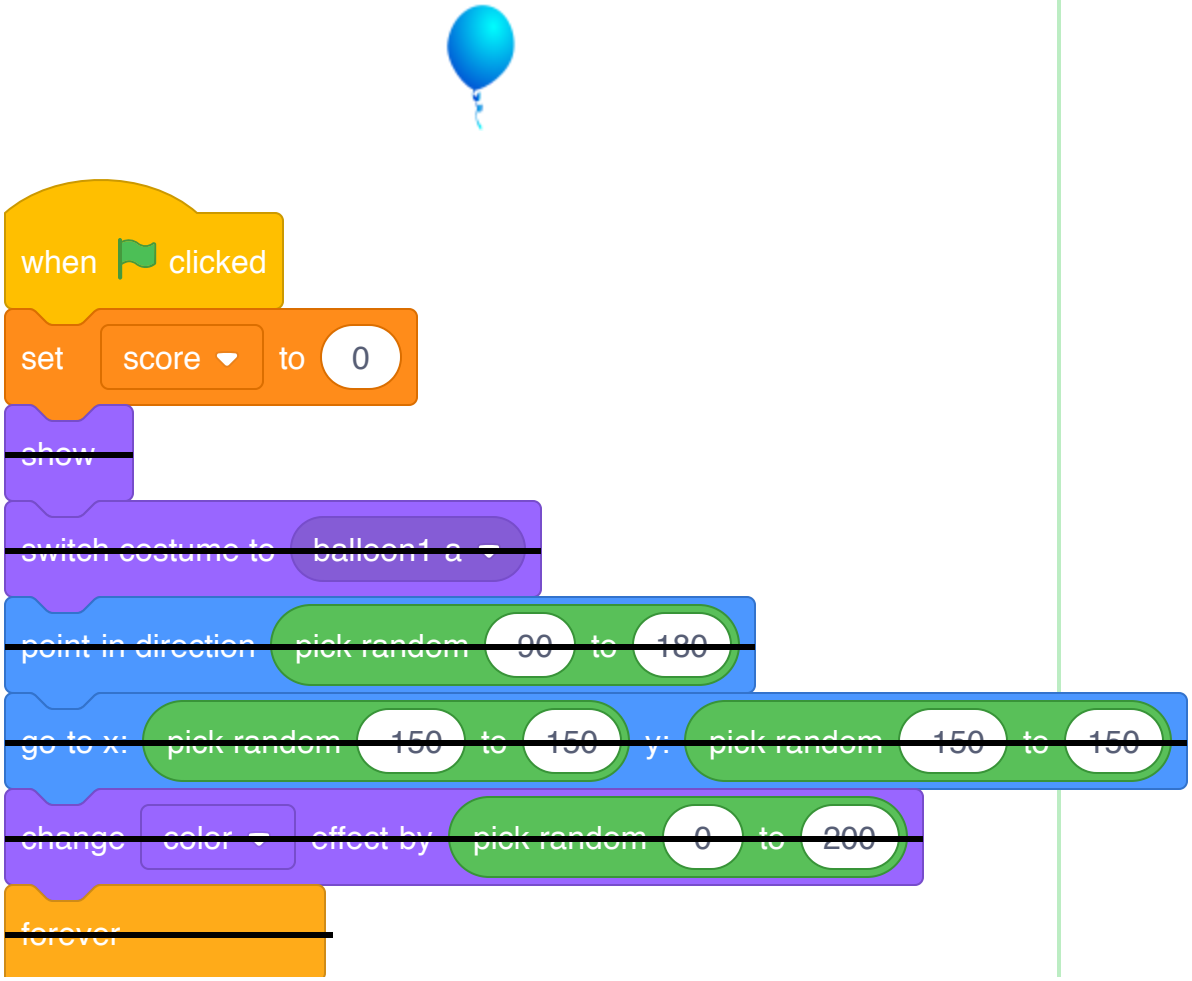
Step 6 Lots of balloons

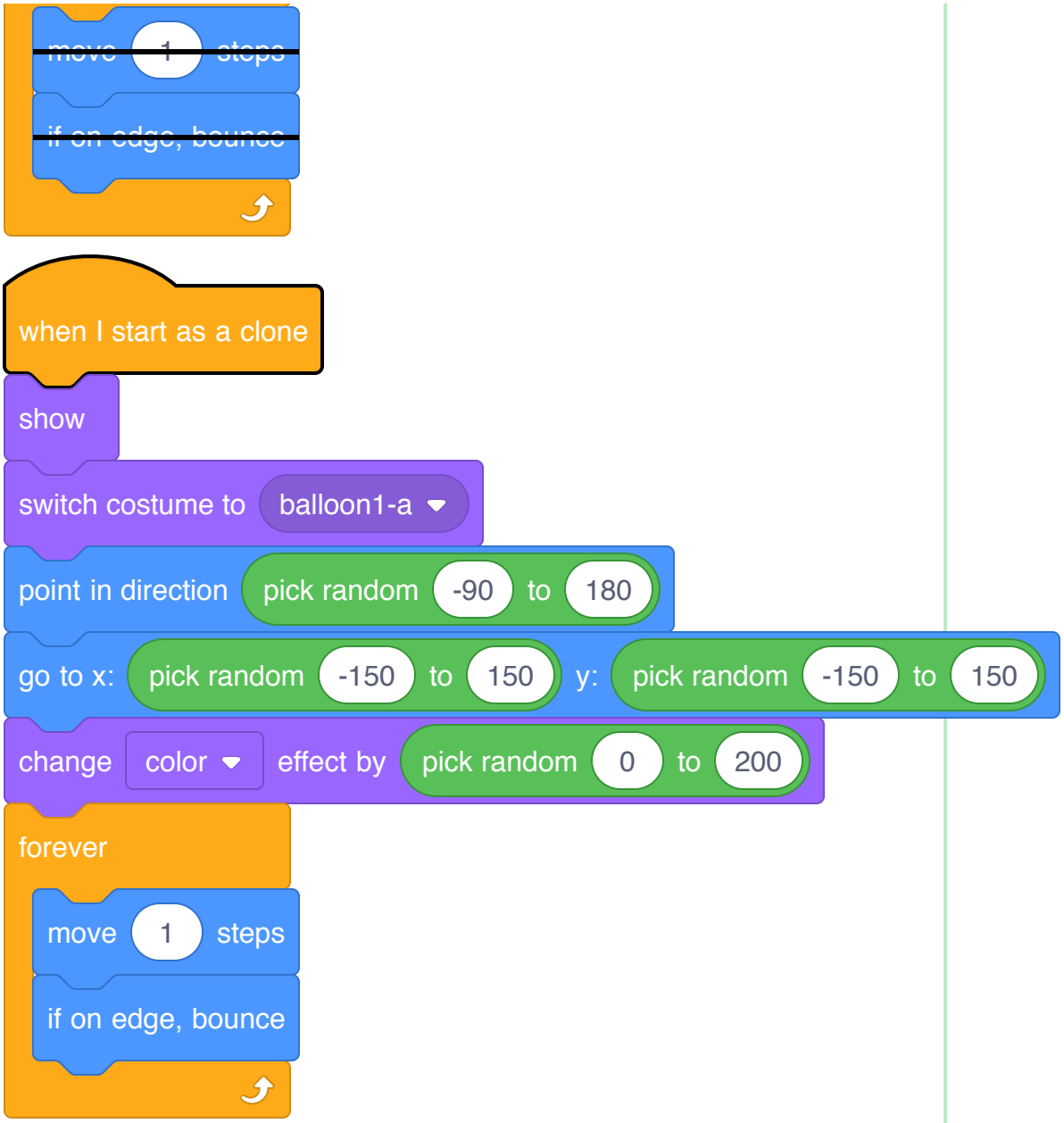
Popping 1 balloon isn't much of a game, so let's add lots more!

One simple way to get lots of balloons is just to right-click on the balloon sprite and click **duplicate**. This is OK if you only want a few, but what if you need 20? or 100? Are you really going to click **duplicate** that many times?

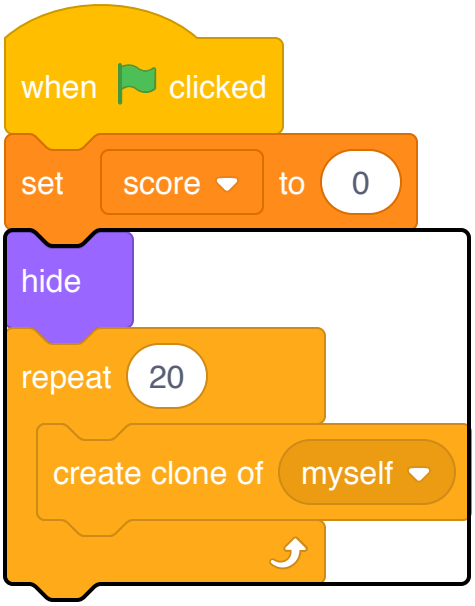
A much better way of getting lots of balloons is to *clone* the balloon sprite.

Drag your balloon `when flag clicked` code to a new `when I start as a clone` control block.

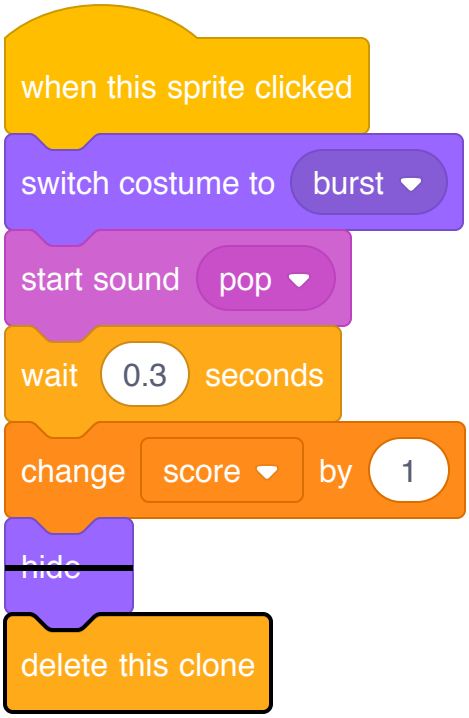




Add code to create 20 balloon clones to the `when flag clicked` code.



You should also replace the `hide` block in the balloon-clicking script with a `delete this clone` block.



Test your project! Now when the flag is clicked, your main balloon sprite will hide and then clone itself 20 times. When each of these 20 clones is started, they will each bounce around the screen randomly, just as they did before. See if you can pop the 20 balloons!

Step 7 Adding a timer

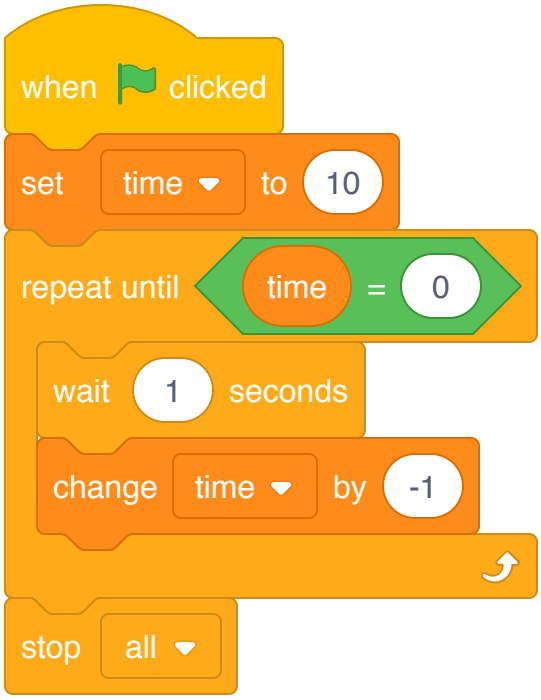
You can make your game more interesting, by only giving your player 10 seconds to pop as many balloons as possible.

You can use another variable to store the remaining time left. Click on the stage, and create a new variable called `time`.

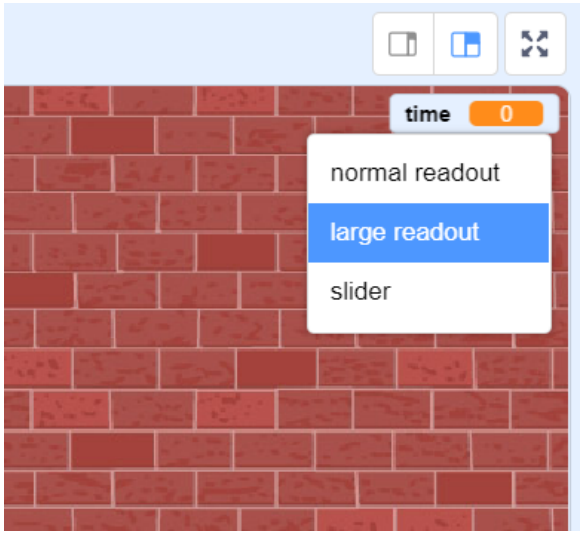
This is how the timer should work:

- The timer should start at 10 seconds;
- The timer should count down every second;
- The game should stop when the timer gets to 0.

Here's the code to do this, which you can add to your *stage*:



Drag your 'time' variable display to the right side of the stage. You can also right-click on the variable display and choose 'large readout' to change how the time is displayed.



Test your game. How many points can you score? If your game is too easy, you can:

- Give the player less time;

- Have more balloons;
- Make the balloons move faster;
- Make the balloons smaller.

Play your game a few times until you're happy that it's the right level of difficulty.

Challenge!

Challenge: More objects

Can you add in other objects to your game? You can add good objects, like donuts, that give you lots of points, or bad objects, like bats, that take points away.

You'll need to think about the objects you're adding. Think about:

- How many will there be?
- How big is it? How does it move?
- How many points will you score (or lose) for clicking it?
- Will it move faster or slower than the balloons?
- What will it look/sound like when it's been clicked?

If you need help adding another object, you can reuse the previous steps!

Step 8 What next?

Take a look at the **Tech Toys** (https://projects.raspberrypi.org/en/projects/tech-toys) Scratch project.