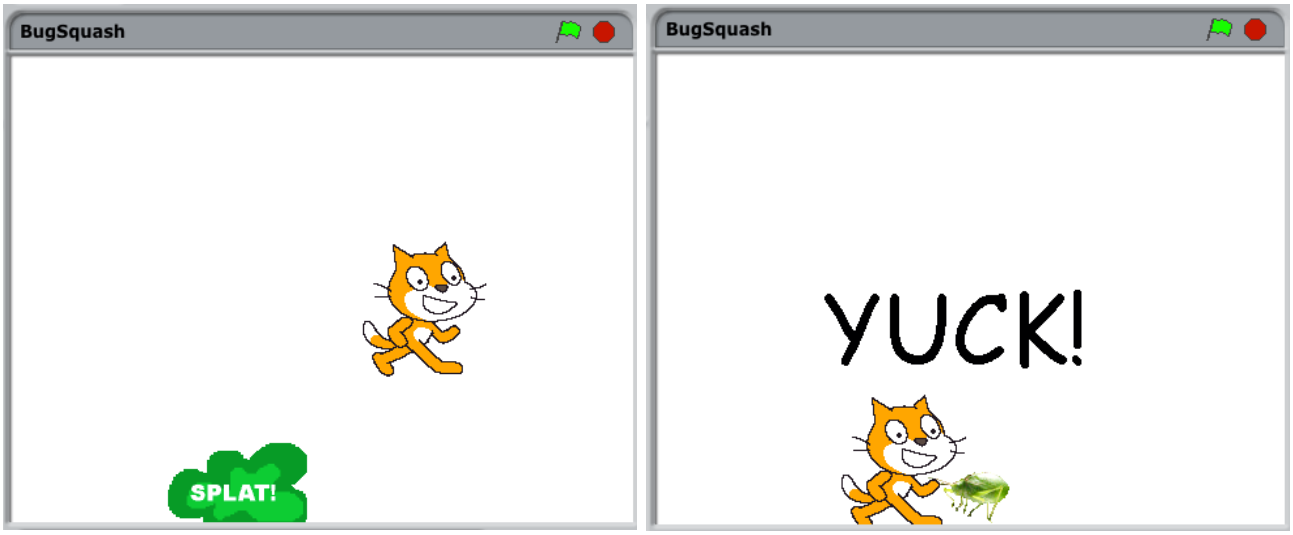
Bug Squash

There are bugs everywhere! They must be squashed at all costs. You play as Scratch the cat, and have to squash the bugs that appear by jumping on top of them. If a bug touches you... yuck! That's game over.

Use the **left and right arrow keys to move**, and press the **space bar to jump**.

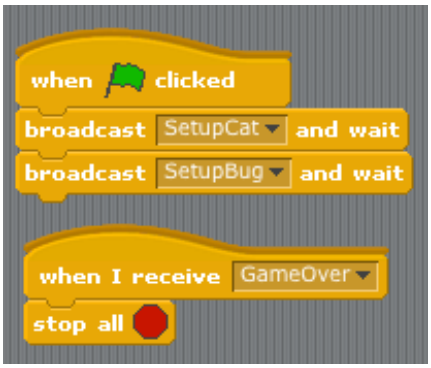


There are a few different bits of work that go into making this game:

- We want **Scratch the cat** to be able to **move around** and **jump** realistically.
- We want the **bug** to **move around by itself**, react to being **squashed**, and **regenerate** itself in a new place afterwards.
- We need to **set up** the game properly, and make sure it ends with a **YUCK!** if Scratch gets touched by the bug.

Setting up the Stage

For each bit of work, we'll write a script that listens for a specific **broadcast**. This is a good way of keeping your program organised and easy to understand.



Put this code in the Stage.

It makes sure that when the **start** button is clicked, the broadcasts called **SetupCat** and **SetupBug** are sent. The code also listens out for a broadcast called **GameOver** and **stops** the game when it hears it.

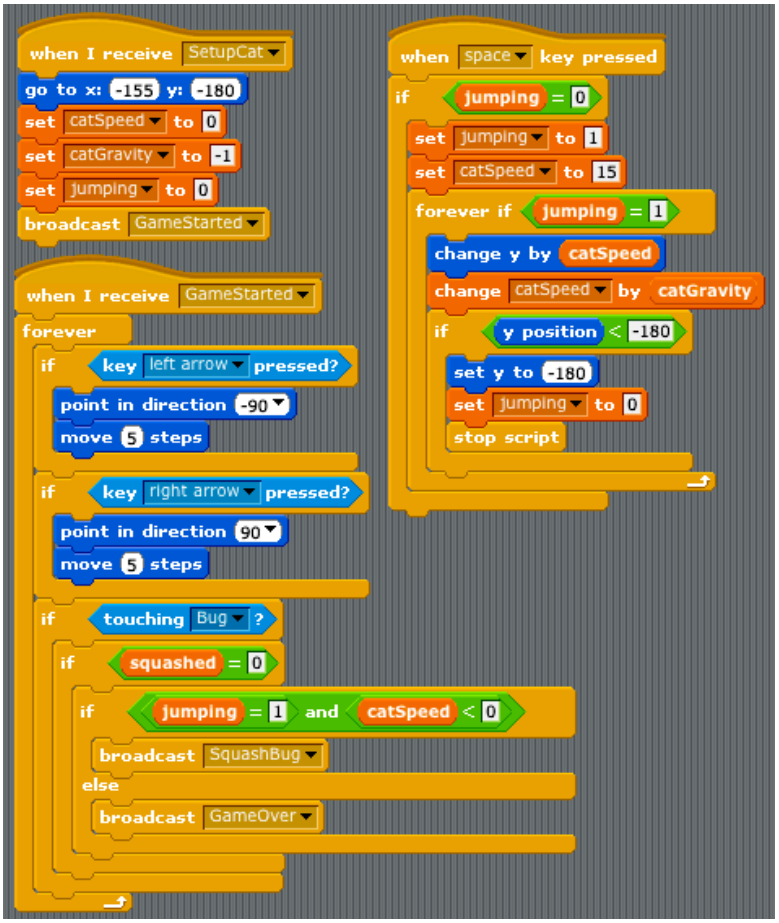
If you run the game now, nothing will happen yet! We need to write more code that listens out for the **SetupCat** and **SetupBug** broadcasts.

Scratch the cat

On the next page, you'll find all the code that needs to go into Scratch the cat. If you don't already have a cat sprite, you'll need to create it. Call your sprite **Cat**, and make sure **this** icon is clicked so that he **can only face left and right.**

Here's what the code does:

- The script that listens for **SetupCat** puts the cat in the **right place**, and sets up some **initial values** for our variables, including **catGravity** to 1. Then, it broadcasts **GameStarted**.
- The script that listens for the **space** key is in charge of **jumping**. If we're not already jumping, and we press the space key, then we set a variable called **catSpeed** to 15. The next bit then makes sure the cat **falls back down** by repeatedly taking the value of catGravity off catSpeed, until the cat reaches the ground. This is exactly what **gravity** does in real life!
- The script that listens for **GameStarted** handles our **movement** left and right, and what happens when we **touch a bug** depending on whether we're jumping or not.



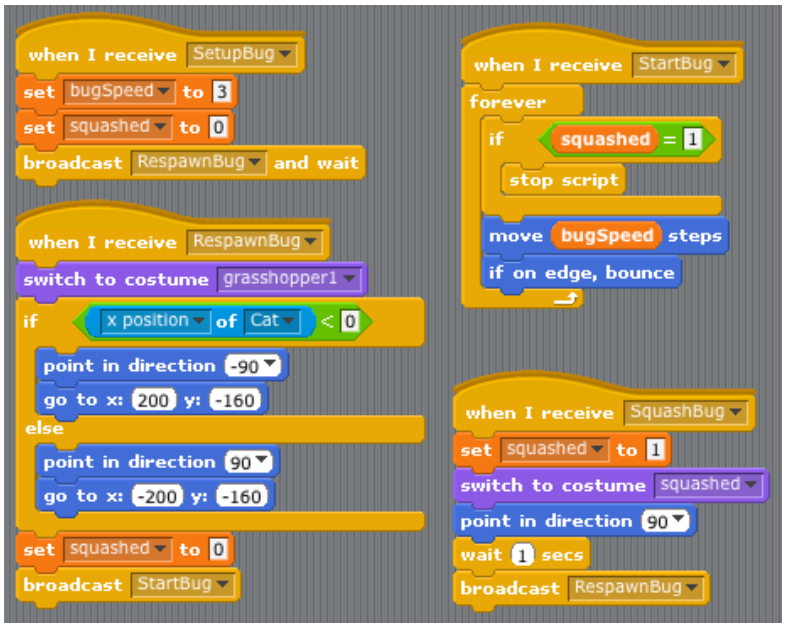
The bug

On the right, you'll find all the code that needs to go into the bug. You'll need to create a bug sprite with **two costumes**:

- the one called **grasshopper1** will be a normal bug.
- the one called **squashed** will be a big **SPLAT!** sign

Call your sprite **Bug**, and make sure the same icon as the cat is clicked so that he **can only face left and right.**

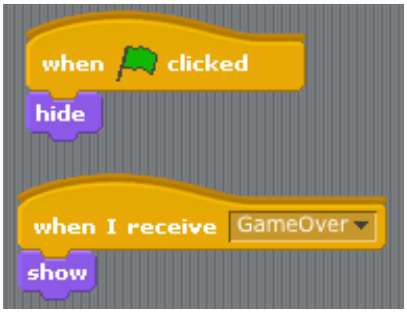
See if you can figure out for yourself how the code the bug works – bearing in mind that one bit has to **make a new bug** once it's been squashed (or at the beginning of the game), another bit has to **move the bug around** and another has to **react to being squashed!**



The YUCK! sign

You can see on the right that the YUCK! sign only has a little bit of code, which **hides** it when the game starts, and **shows** it hears the **GameOver** broadcast.

You'll need to make a new sprite called **Yuck** which is just some text saying **YUCK!**



...and you're done!