

# Advanced Scratch

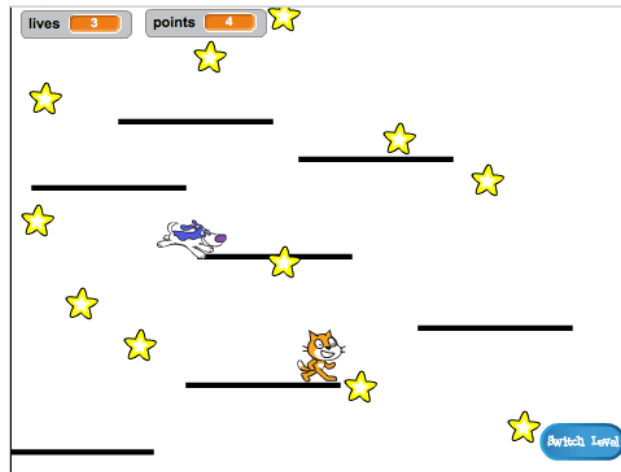


## SETTING THINGS UP

Card 1 of 8

I'm Learning: Scratch

- 1 These are the Advanced Scratch Sushi Cards, and in them you'll be making a platform-style game which you can later add your own levels, powers and characters to.



Feel free to improve on my "art"

Because you're learning Scratch and not how to build a physics engine (code that makes things behave at least a little like the real world—e.g. not falling through floors), you'll be starting with a project I've created that already has the basics of movement, jumping and detecting platforms built in.

You should take a quick look at it including the details on the other side of this card, since you'll be making some changes to it later, but you don't need to understand everything it's doing.

- 2 The first thing you'll need to do is to get a copy of the code from <http://dojo.soy/advanced-scratch>

You can download this code by clicking "See Inside", then the **File** menu and then "Download to your computer" and then open it in Scratch on your computer.

You can use it directly in Scratch in your browser by just clicking "See Inside" and then "Remix".

# Advanced Scratch



## SETTING THINGS UP

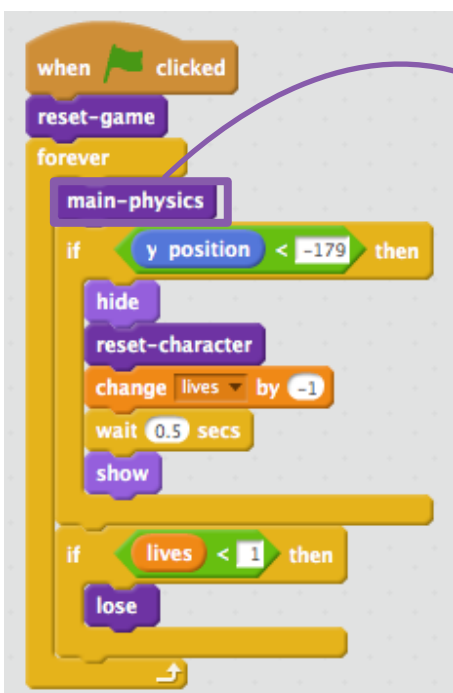
Card 1 of 8

I'm Learning: Scratch

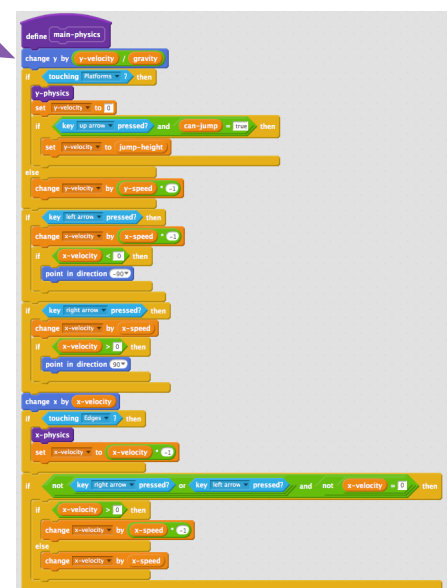
3 The physics engine of the game has a variety of pieces in it, some of which work right now and some of which don't. You can find out which by running the game and trying to play it.

You can lose lives, but nothing happens when you run out. Also, the game has only got one level, one type of thing to collect and no enemies. You're going to fix all of that, and a bit more!

For now, take a look at how the code is put together. It uses lots of **more blocks**, which are great for splitting your code up into pieces so you can manage it better. It's like having a block made up of a lot of other blocks, which you can give some basic instructions to.



The main game **forever** loop calls the “main-physics” block to do a whole lot of stuff! Keeping them separated like this makes it easy to read the main loop and understand what happens when, without worrying about *how* it happens.



4 Now look at “reset game” and “reset character” and notice:

They do pretty normal things—setting up variables, making sure the character rotates properly

“reset-game” **calls** “reset-character”—you can use a **more block** inside another **more block**!

“reset-character” gets used in two different places, but to change it you only have to change the code of the **more block** in one! This can save you a lot of work and help you avoid mistakes.

# Advanced Scratch



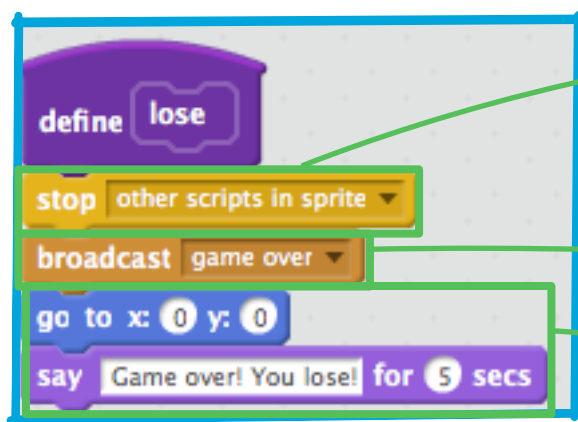
## LOSING THE GAME

Card 2 of 8

I'm Learning: Scratch

- 1 You may have noticed that the “lose” **more block** on the “Player Character” sprite is empty. You’re going to fill this in and setup all the pieces needed for a nice “Game Over” screen.

First, complete the “lose” block:



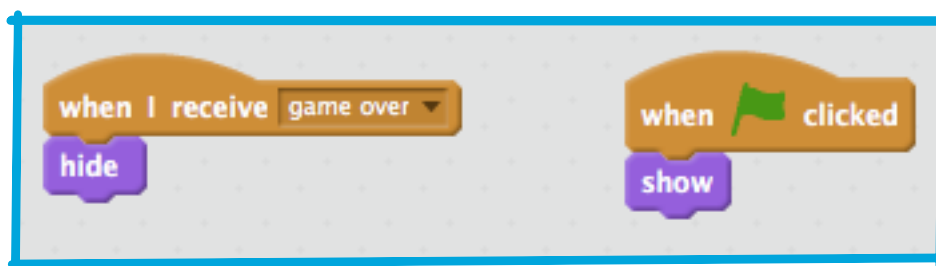
Stop the physics and other game scripts on the Player Character

Tell all the other sprites that the game is over, by broadcasting a message so they can change based on that

Move the Player Character to the centre of the screen and have them tell the player the game is over

- 2 Now you need to make sure all the sprites know what to do when the game is over, and how to reset themselves when the player starts a new game. **Don't forget that any new sprites you add may need code for this too!**

Start with the easy ones, the “Platforms” and “Edges” sprites both need code for appearing when the game starts and disappearing at game over.



# Advanced Scratch



## LOSING THE GAME

Card 2 of 8

I'm Learning: Scratch

3 Now, for something a little more tricky! If you look at the code for the “Collectable” sprite, you’ll see it works by **cloning** itself. That is, it makes copies of itself, which follow the special **when I start as a clone** instructions.

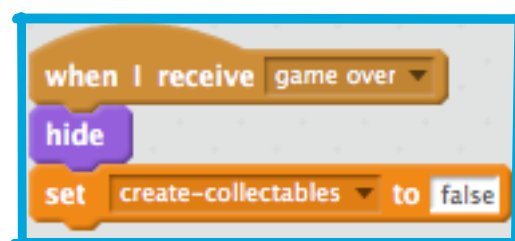
We’ll talk more about what makes clones special when we get to the card about making new and different collectables, but for now what you need to know is that clones can do *almost* everything a normal sprite can, including receiving **broadcast** messages.

Let’s look at how the “Collectable” sprite works:

The image shows a Scratch code block for a 'Collectable' sprite. The code starts with a 'when clicked' event, followed by a 'hide' block. Then, four 'set' blocks are used to initialize variables: 'collectable-value' to 1, 'collectable-speed' to 1, 'collectable-frequency' to 1, and 'create-collectables' to true. The 'collectable-type' is set to 1. A 'repeat until' loop is used, with the condition 'not create-collectables = true'. Inside the loop, there is a 'wait collectable-frequency secs' block, a 'go to x: pick random -240 to 240 y: 179' block, and a 'create clone of myself' block. Green arrows point from specific code blocks to explanatory text boxes on the right.

- Make the original collectable invisible
- Setup the control variables. We’ll come back to these later.
- This “create-collectables” variable is the on/off switch for cloning
- This creates clones if “create-collectables” is true, and does nothing if it’s not

4 Now what you need to do is setup a block like the ones you had on the “Edges” and “Platforms” sprites on the “Collectable” sprite. The only difference is you’re also setting the “create-collectables” variable to “false” so that no new clones are created. Notice how you can use the variable to pass messages from one part of your code to another!



# Advanced Scratch



POWERUPS

Card 3 of 8

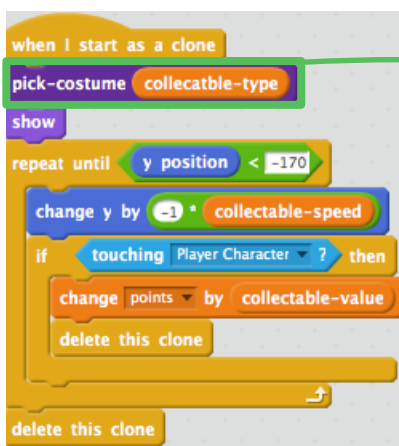
I'm Learning: Scratch

1 On the last card you saw the collectable I created. It's a star that just adds one point when you grab it. That's pretty boring.

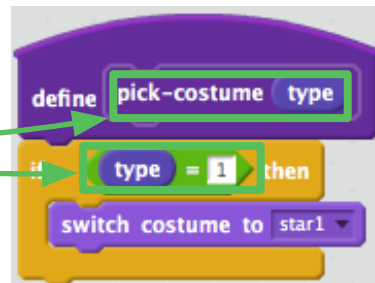
On this card, you're going to create a new collectable, but you're going to do it in a way that makes adding more collectables easy, so you can invent your own powers and bonuses and really make the game your own!

2 Add a new costume to the "Collectable" sprite for your new power-up. I like the lightning bolt, but pick whatever you like.

3 Notice that I've already included some pieces to make this easier for you with the "collectable-type" variable and the "pick costume" **more block**. You're going to need to improve on them though.



"collectable-type" is passed to "pick-costume" when it's called, where it becomes "type" and can be used inside the more block



4 First, you need to set the collectable type. It's just a number, used to tell the program what costume, rules etc. to use for the collectable. You're going to want to pick it at random, to keep things interesting. This example gives a 1/50 chance:



## Pro Tip!

There can be a different value set as the "collectable-type" for each clone. Think of it like creating a new copy of the variable on the main "collectable" sprite with the value that was in "collectable-type" the instant that clone was created. One of the things that makes clones special is that they cannot change the values of any variables they start with. They are effectively **constant** values.

# Advanced Scratch



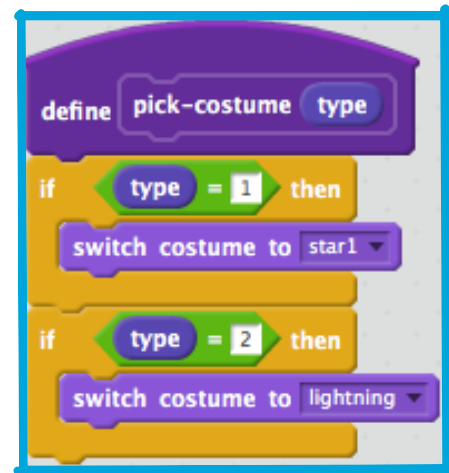
POWERUPS

Card 3 of 8

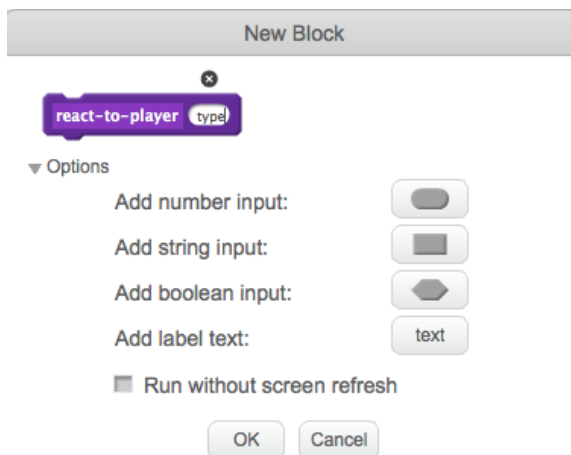
I'm Learning: Scratch

5 Great! Now you're setting a different value for the collectable type, but none of the code knows what to do with it yet!

First, just teach the "pick-costume" more block to set the new costume when it gets the new type, like this (using whatever costume you picked):

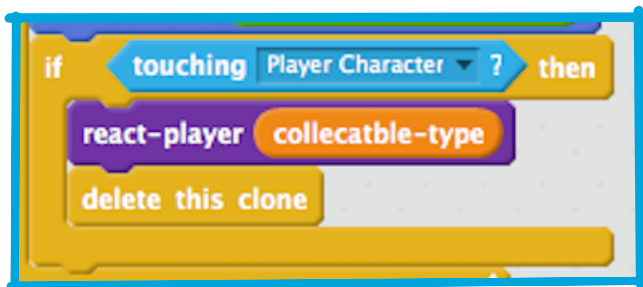
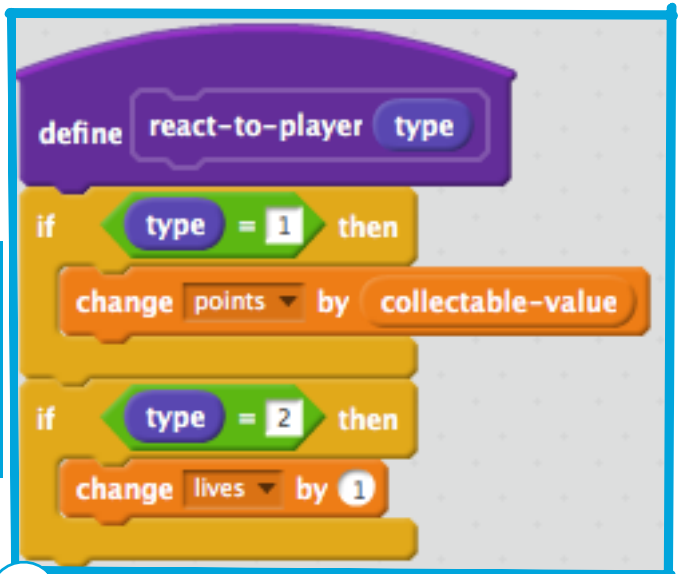


6 Now you need to decide what the powerup will do. We'll start with something simple: giving the player a new life. On the next card, you'll make it do something cooler.



1 Go into more blocks and Make a Block. Expand the options section and add a number input. Name the block "react-to-player" and the number input "type"

2 Make the "react-to-player" block either give the same points prize that the star is already giving, or increase the player's lives, depending on the "type" of powerup.



3 Update the when I start as a clone code to replace the points increase with a call to "react-to-player", passing "collectable-type". Stars still boost points but the new powerup adds lives.

# Advanced Scratch



**SUPER POWERUPS!**

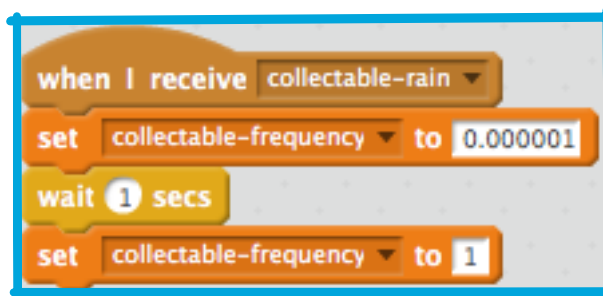
Card 4 of 8

I'm Learning: Scratch

- 1 Now that you have a new powerup working, it's time to make it do something cool! How about making it rain powerups for a few seconds, instead of just giving out an extra life?

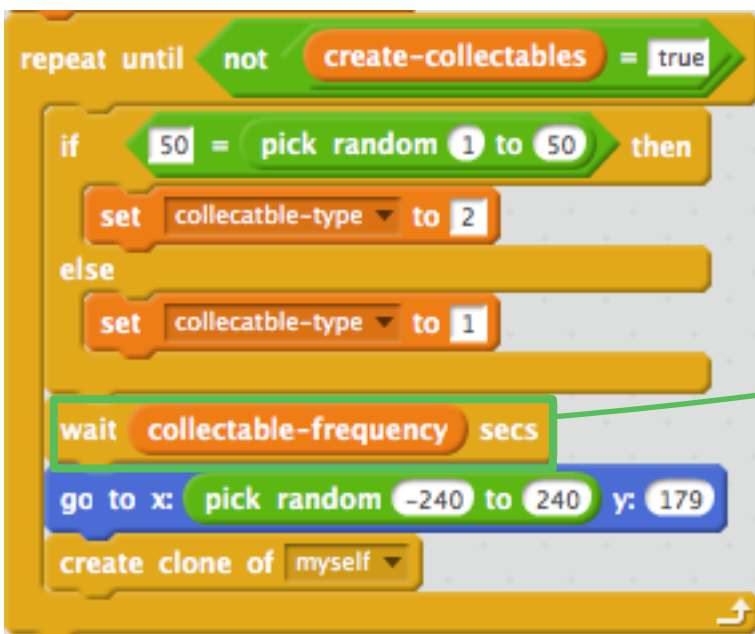
To make that work, you need to create another piece of code that you can start while the “react-to-player” block finishes running. The way to make this happen is to use the **broadcast** block to send a message to another piece of code inside this sprite. Let's call it “collectable-rain”, since that's basically what it does!

- 2 Create this block on the “collectable” sprite.



This block just sets “collectable-frequency” to a very small number (change it to different values, see what happens!) and then waits a second and changes it back to 1.

This doesn't look like it should do much, but if you think about what's happening during that second, the **when (green flag) clicked** code is still running, and the **repeat until** loop in it is looping. Look at the code in that loop:



Instead of pausing the code here for a second, it's only pausing for *one millionth* of a second, meaning that the loop will run many more times than normal because of the smaller value of “collectable-frequency”.

This means that the code is going to create *a lot* more powerups in that second than it normally would.

Can you think of any problems that might cause? There'll be a lot more lightning bolts... what if I kept catching them?

# Advanced Scratch



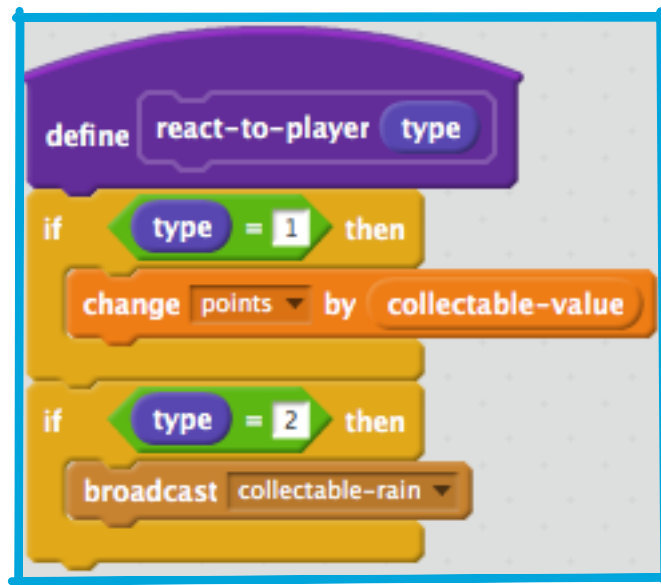
## SUPER POWERUPS!

Card 4 of 8

I'm Learning: Scratch

3

Now you have that broadcast block ready, but it's not being used yet. This next part's easy. Just update "react-to-player" to look like this, so it broadcasts "collectable-rain" when the player touches a type 2 powerup.



## Get Creative!

Based on this card and the previous one, you can now make as many powerups as you want!

What about one that gives out 20 times the usual number of points, adds three lives, or maybe means the player can't run out of lives while it's on?

Come up with some and see if you can make them!



# Advanced Scratch



## ADDING SOME COMPETITION

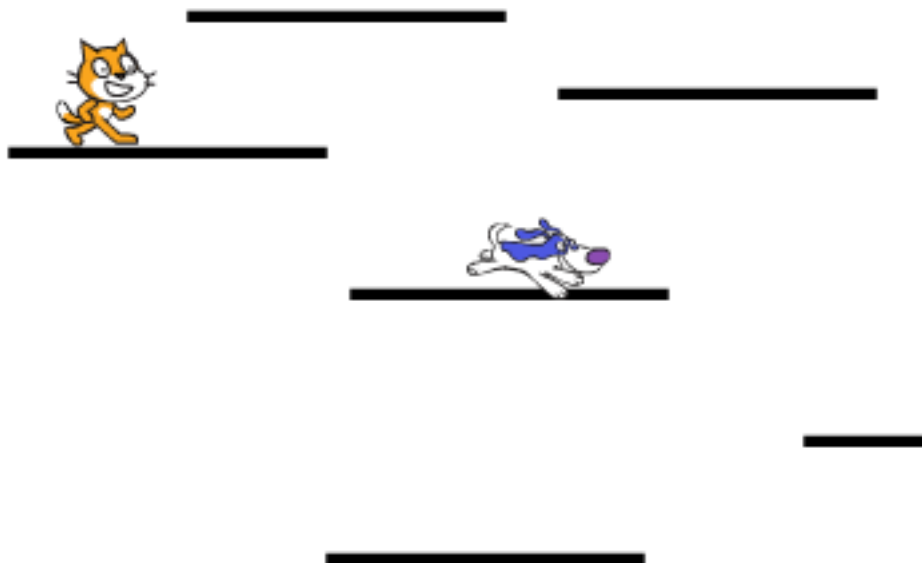
Card 5 of 8

I'm Learning: Scratch

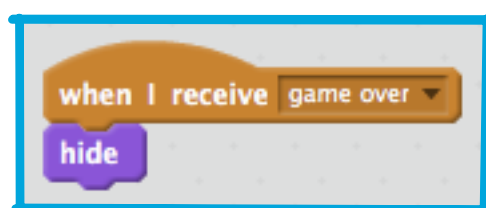
1 Your game works and now you can collect points, get special powers from powerups and lose. We're getting somewhere! Maybe it'd be fun to add some competition though... what about including a character who moves around a little, but you can't touch? Like the enemies in the traditional platformer games (like Super Mario) we're being inspired by here.

Well, first, pick your enemy and add their sprite. Because our character is a cat, I chose a dog. There are lots of other sprites you could add though. I also renamed the sprite to "Enemy" just to make things clearer for me.

Resize the sprite to the right size and place it somewhere appropriate to start. Here's what mine looks like:



2 Write the easier code first: Setup the block for the "game over" message, so the enemy will disappear when the player loses the game.



# Advanced Scratch

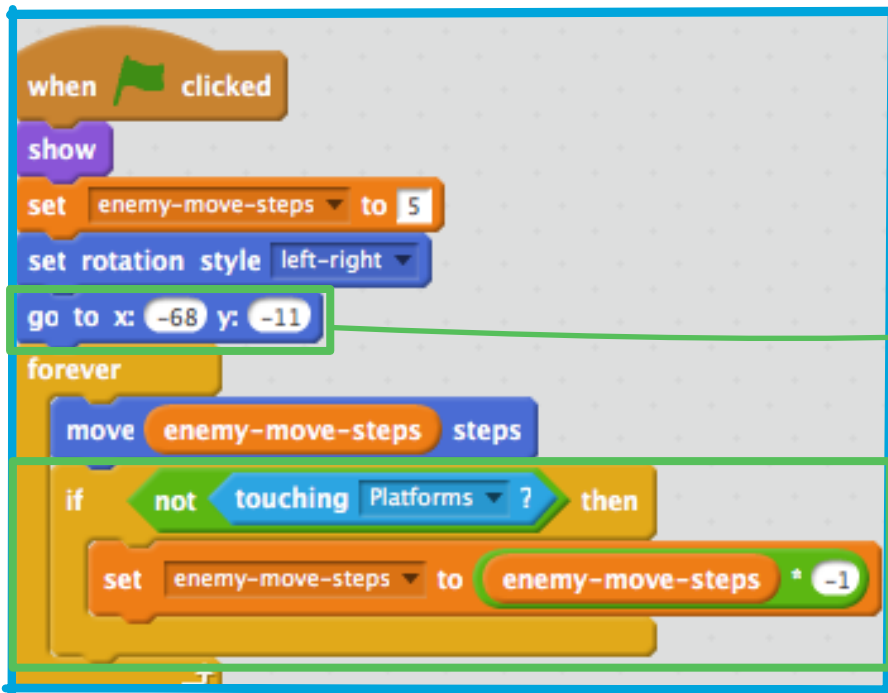


## ADDING SOME COMPETITION

Card 5 of 8

I'm Learning: Scratch

- 3 Now you need to write the code for what the enemy does. You can use mine from this card, but don't be afraid to add more! What if they teleport around to different platforms? Or what if there's a powerup that makes them move faster, or slower?



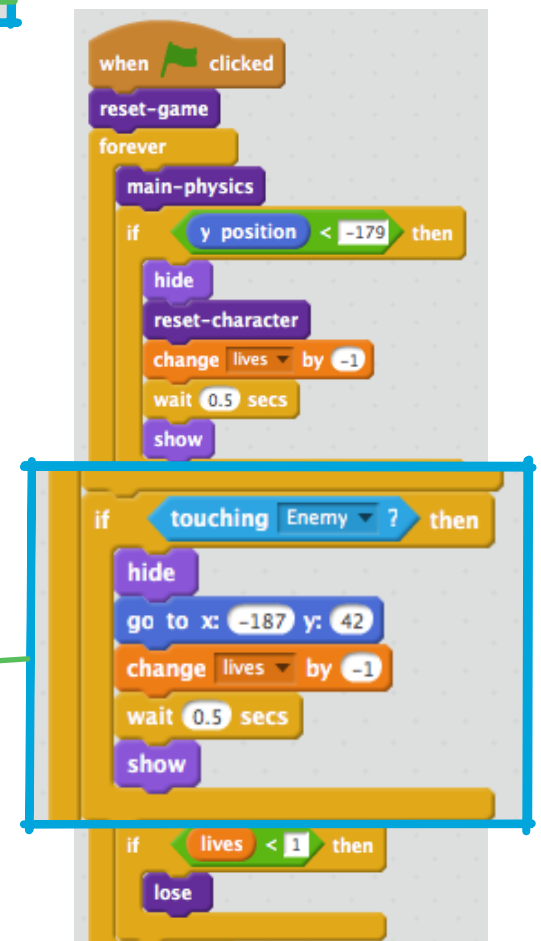
If you just drag the go to block and don't change the x & y values, they'll be the values for the current location of the sprite!

The enemy will turn around when they get to the end of the platform!

- 4 The next thing you'll need is for the player to lose a life when they touch the enemy. You need to make sure they stop touching really quickly, though, since otherwise the touching code will keep running and they'll keep losing lives.

This code is the way I did modified the "Player Character" sprite's main block, though feel free to try to improve on it!

Here's the new code. It hides the character, moves them back to their starting position, reduces lives by one and, after half a second, makes them re-appear.



# Advanced Scratch



LEVEL 2

Card 6 of 8

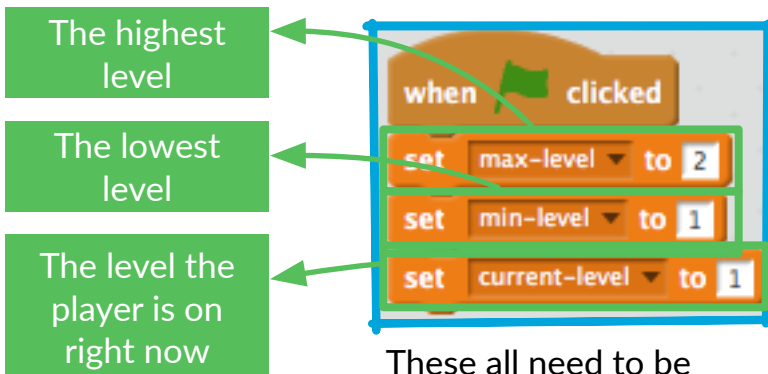
I'm Learning: Scratch

- 1 What you're going to do on this card is add a new level to the game that the player can get to just by pressing a button. Later, you can change it so they need a certain number of points, or something else, to get there.
- 2 First, create a new button sprite by either adding it from the library or drawing your own. I did a bit of both and came up with this:

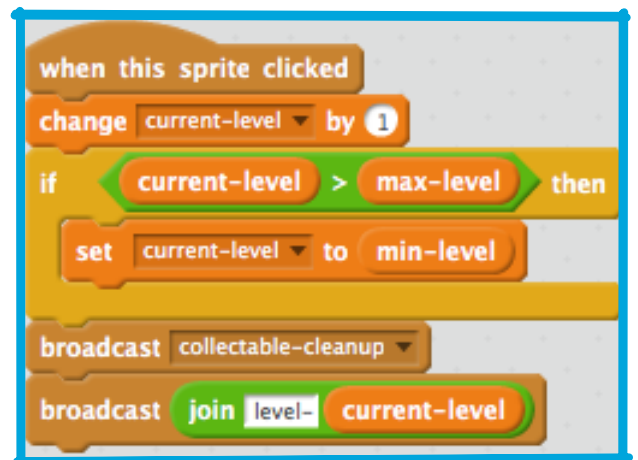
Switch Level

- 3 Now, the code for this button is kinda clever; it's designed so that every time you click it, it will take you to the next level, how ever many levels there are.

Add this to your button sprite



These all need to be set by the programmer, so if you add a third level, don't forget to change "max-level"!



This uses broadcasts to tell the other sprites which level to display, and to clear up the collectables.

- 4 Now you need to get the other sprites to respond to those broadcasts! Start with the easy one: clearing all the collectables. If you just tell them to **hide**, all the existing clones will. So add this to the "collectable" sprite:



Since one of the first things any new clone already does is show itself, that means you don't even have to worry about turning this off for them!

# Advanced Scratch



LEVEL 2

Card 6 of 8

I'm Learning: Scratch

5

Now to switch the “platforms” sprite! You can design your own new level later, if you like, but for now let’s use the one I’ve already included (you’ll see why on the next card!).

You just need this code to take the messages sent out by **joining** the “level-” and the “current-level” variable and use them to change the “platforms” costume.



6

For the “Enemy” sprite, you just need to make sure it disappears on level 2 (or move it to another platform!), like this:



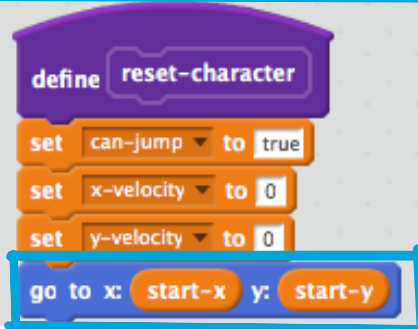
7

Finally, the player character needs to separate out the coordinates from the “reset character” **more block**, so the character goes to the right place, and call the first level when the game starts.

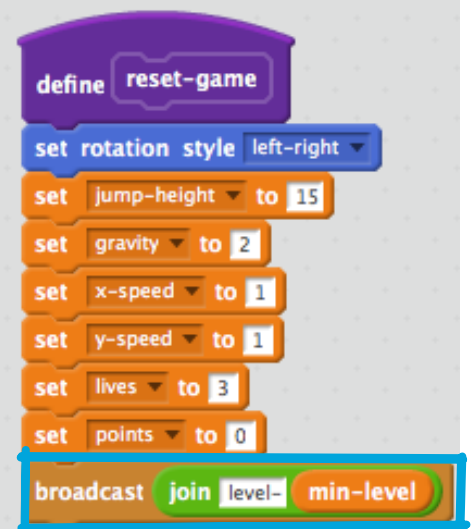
Set starting coordinates and call “reset-character”



Use variable starting coordinates instead of fixed x and y coordinates



Use the broadcast of the “min-level” to reset the character and game



# Advanced Scratch

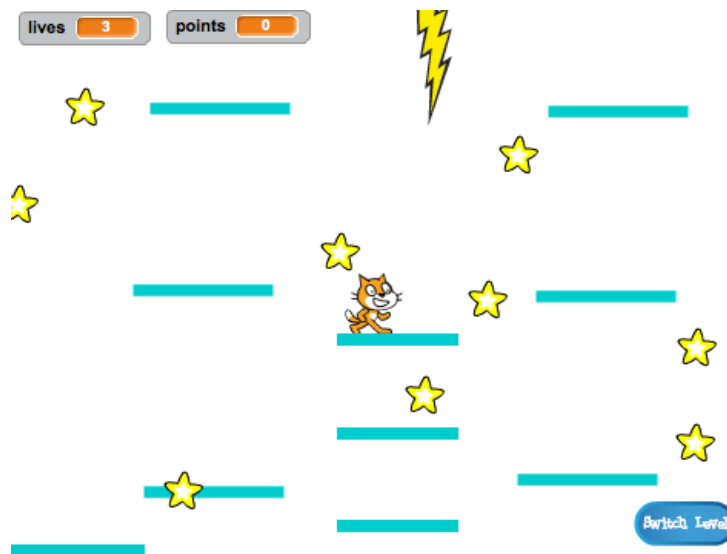


## MOVING PLATFORMS

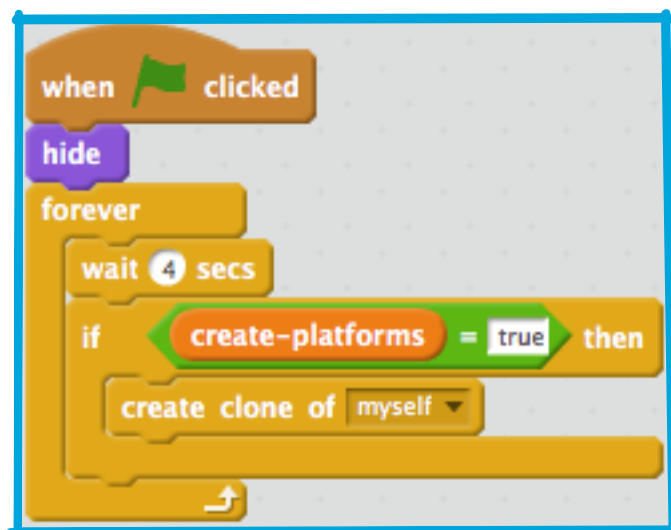
Card 7 of 8

I'm Learning: Scratch

- 1 The reason I asked you to use my version of level 2 is the gap you might have noticed in the middle. You're going to create a platform that moves through that gap, that the player can jump onto and ride.



- 2 First, you'll need the sprite for the platform. You can create this by adding any sprite, naming it "Moving-Platform" and using the costume customisation tools in the "costumes" tab to make it look like the other platforms (use vector mode).
- 3 Time to start adding some code! Begin with the basics: To make a never-ending set of platforms moving up the screen, you'll need to **clone** the platform at some sort of interval. I picked 4 seconds. You also need to make sure that there's an on/off switch for making the platforms, so they don't show up on level 1. I'm using a **variable** called "create-platforms".



# Advanced Scratch

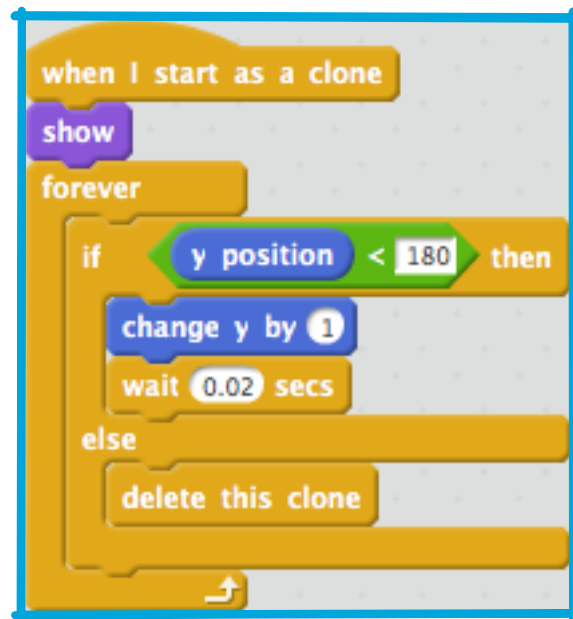


## MOVING PLATFORMS

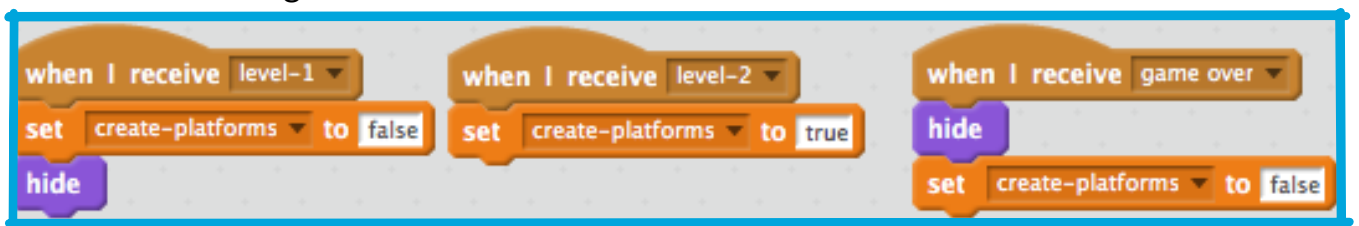
Card 7 of 8

I'm Learning: Scratch

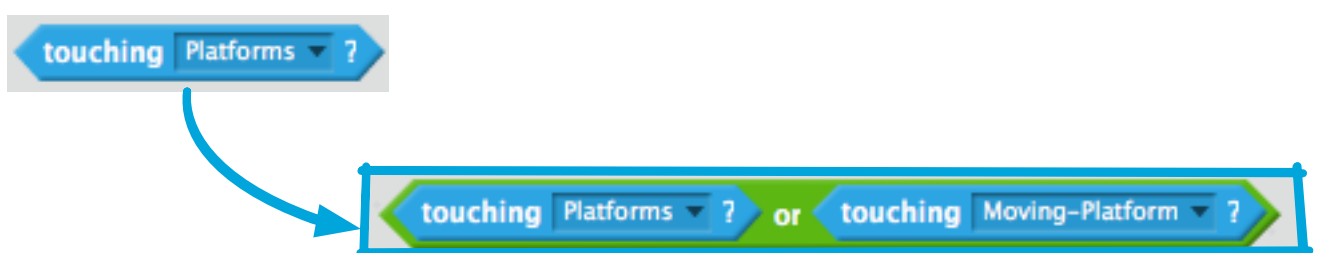
- 4 The clone's code is simple: Move up to the top of the screen, slowly enough for the player to jump on and off, and then disappear.



- 5 You need to make the platforms disappear/reappear based on the level changing and on the "game over" event.



- 6 Now, if you try to actually play the game, the Player Character falls through the platform! Any idea why?  
It's because the physics code doesn't know about the platform. It's actually an easy fix: Go through the code on the "Player Character" sprite and replace every **touching "Platforms"** with an **OR** operator, where you check for either **touching "Platforms"** OR touching "Moving Platform"



# Advanced Scratch



## WHAT NEXT?

Card 8 of 8

I'm Learning: Scratch

1

You've got a game now! But there's still a lot more you can do with it! Here's a few ideas to get you started:

### High Scores!

Keep a **list** of the names and scores of people who've gotten high scores in the game!

You'll need to use the **ask** block to get their name.

### New Powerups!

Try adding some new powerups. For example:

- Immunity to enemies
- More lives
- Bigger player character
- Smaller player character

### Scrolling Levels!

Can you figure out how to make the levels scroll along, so the player character can move through them from left to right?

Or at least *look* like that's what's happening?

### Completing Levels!

Right now, the levels never end. What if, instead of pushing a button, you needed a certain number of points to get to the next level?

### Play With Physics!

Try changing some of the values in the physics engine, like the gravity, jump height, x-speed and y-speed. How do they change the game?

Can you use them to make powerups?

### More Levels!

Add more levels! Make better art!

By using the stage background, make the game look cooler while still keeping platforms easy to work with as a coder.

### Sound Effects!

This game is totally silent right now!

Try adding background music and sound effects using the blocks in **sound**!

### Secrets!

Think of secret bonuses, cheat codes and other "easter eggs" you could hide in the game for players to discover.

Try to code some of them!

### Different Characters!

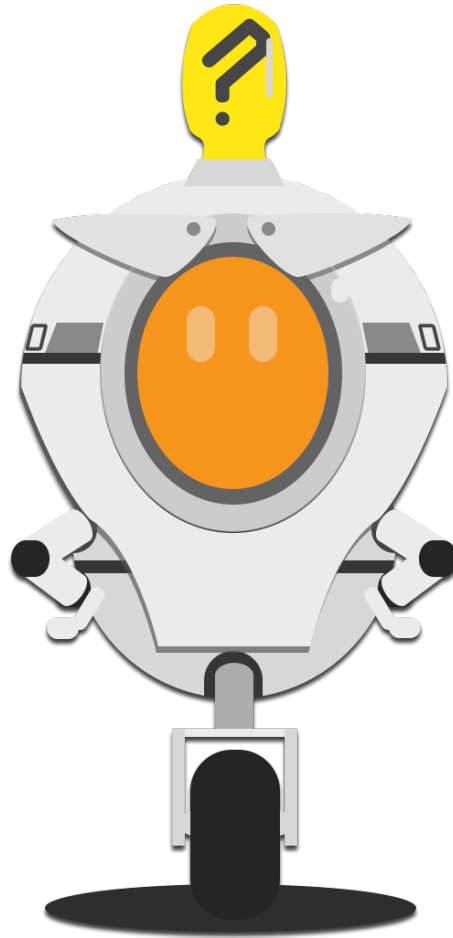
Let the player pick a character. Make them different in things like size, how high they jump, maybe even how many lives they have and points they get from collectables!

# Advanced Scratch



**WHAT NEXT?**

Card 8 of 8  
I'm Learning: Scratch



## Tell us what you thought!

Go to <http://dojo.soy/advanced-scratch-end> and let us know if you had fun with these Sushi Cards and what you'd like to see in future!