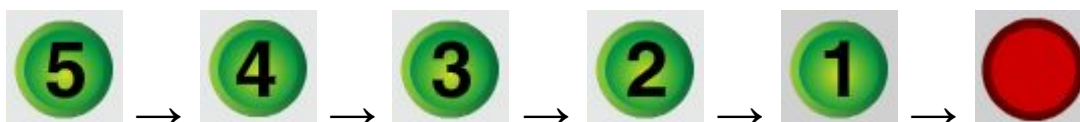


## Week 7

There was no 'week 6' worksheet because we used Week 6 to catch up and do challenges from the previous weeks.

This week we will add an idea somebody came up with in the first couple of sessions: A cell which at the start is fine for Waldo to go on, but after some time, it turns into a cell Waldo is *not* allowed to go on.

As Waldo steps round the grid, the new sprite will change like this:



Waldo is allowed to go on the sprite while it's counting down, but once the sprite has reached the final 'blocked' state, Waldo is not allowed on that cell any more.

Either start from your previous work, or use [my demo 'end of week 5' project](#) (that's a version with most challenges done) and remix it.

### Add 'countdown blocker' sprite

The new sprite behaves very like the arrows and so on, so **duplicate one of the arrow sprites**, give it a sensible name, and move it to somewhere sensible on your game stage.

#### Countdown costumes

Next we need to choose costumes for when it's OK for Waldo to go on the sprite. I chose the 'Button1' costume from the library. You might need to resize it so it fits in a cell OK.



For the countdown, we need to duplicate this *costume* four times, to get five copies altogether. **Make sure you're duplicating the costume not the sprite!**



Then use the text tool to put the countdown numbers in the middle of each costume.

The first costume should have '5', then counting down so the last costume should have '1'. You can make the countdown longer or shorter if you like, but you'll need to make other changes to the rest of the work in this sheet.

Finally add a separate 'blocked' costume at the end. Make sure it has a good area of a colour not used in the 'allowed' costumes — you'll see why later. I used the same costume, but changed the colours with the paint bucket tool shown at the right.



Delete any other costumes that the sprite might have from whatever you duplicated it from. You should end up with *just* a set of costumes like those shown at the start of this worksheet.

You can of course design your own set of costumes; maybe the size or colour changes instead of just a number counting down.

## Starting costume

While in 'costumes', click on the highest countdown number to set the costume for the main sprite. This is what whoever is designing the level will drag onto the grid when setting up the level.

Try it!



In full-screen mode, you should be able to put countdown-blockers onto the grid, but when you start Waldo off, they won't count down, and Waldo will ignore them.

**For this whole sheet, when testing, you need to have some atoms on the level, otherwise it thinks you've won the level straight away.**

This is the same pattern we've seen when adding other things, so hopefully it's starting to look familiar now. We need to put scripts together to give everything the right behaviour.

## Countdown behaviour

Let's get the countdown working first.

Countdown should reset when Waldo is sent back to the start

When the player clicks 'stop' to send Waldo back to the start, we need all countdown blockers to go back to their starting count.

**Add this script to the Countdown Blocker:**



(Here you can see I've been a bit lazy and not given the costumes sensible names. If you're less lazy you can give better names to the costumes!)

Countdown should count down every time Waldo takes a step

We want the counter to tick down one count each time Waldo finishes moving from one cell to the next. This is like the atoms, which have to check whether they've been collected after Waldo takes a step.

So we can use the same message which Waldo broadcasts after every step it takes, which I called '*check whether collected*' but you might have called something else.

As a first try, **add this script to the Countdown Blocker:**



Try it!



If you go to full-screen and try this, it sort of works, but you'll see two problems:

- The 'original' countdown blocker in the 'toolbox' counts down too.
- The countdown blocker doesn't stop once it gets to the 'blocked' costume.

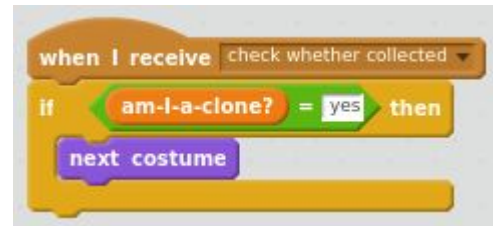
We'll fix these bugs next.

## Only clones should count down

Each sprite already has a variable which lets it remember whether it's a clone or the original — 'am-I-a-clone?'. So before switching costume, a countdown blocker should check it's a clone and not the original.

**Change the Countdown Blocker's script you just added to include an 'if' test:**

If you try this, only the clones you drag onto the grid should count down. But they still don't stop in the 'blocked' costume.



## Stop changing once in 'blocked' costume

In the 'Looks' category, Scratch has a **costume #** reporter block, which lets your script tell what costume the sprite is currently using. We only want to do 'next costume' if the sprite is in an 'earlier' costume than the 'blocked' one. The 'blocked' costume is the sprite's 6th costume, so we can ask the question with the block shown at the right.



We need to test both 'am I a clone?' and 'am I in a counting costume?'.

**Change the Countdown Blocker's script again to make the 'if' test check both things:**



Try it!



In full-screen, you should now be able to drag some countdown blockers onto the grid, and see them count down, then stay in their 'blocked' costume once they reach it.

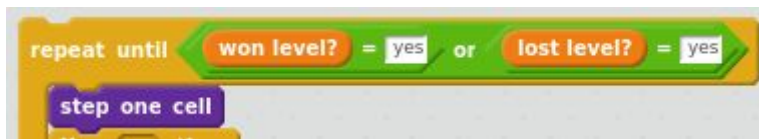
## Waldo behaviour

Our next job is to make it so Waldo can go on the countdown blocker when it's counting down, but Waldo should lose the level if it goes onto a countdown blocker in the 'blocked' stage.

## Reminder: Waldo's main loop

The main script Waldo has is the one which keeps taking steps until the player has either won the game or lost the game.

At the moment, the 'repeat until' loop in that script looks like this, leaving out the details:



*[ in here, there are some 'if ... then' tests for the different ways of winning or losing ]*



## Test for hitting a blocked countdown blocker

We need to add another test, to check whether Waldo has ended up on a countdown blocker which is in its 'blocked' costume. We can think of this as:

- Waldo is touching one of the countdown blocks  
and
- That countdown block is in its 'blocked' costume.

Scratch doesn't give us a way of telling what costume a particular clone of another sprite is in, so we cheat a bit and use the costume colour. This is why we needed a different colour for the 'blocked' costume.

The question we want to ask is



Here, the 'touching color □' block should use whatever special colour your 'blocked' costume has.

If the answer to this question is 'yes', the player has lost the level.

**Add this test into the 'repeat until' loop:**



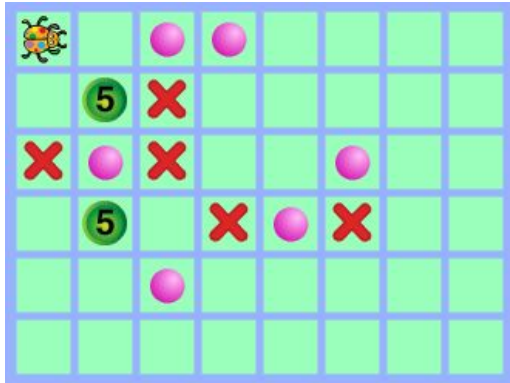
Try it!



If you go to full-screen mode, everything should now work. The countdown blockers should count down into the 'blocked' state, and Waldo should be able to walk over them while they're counting down, but lose the level if it goes over one which has counted all the way down to its 'blocked' state.

## Make a good level using the new Countdown Blocker

Make a level which the player has to think about which order to collect the atoms. If the player tries to collect the wrong atoms first, the 'timed blocker' should make it impossible to finish the level. Here's one I came up with but see what levels you can design:



If the player lets Waldo go straight ahead and pick up the two atoms in the top row, the player will never be able to get Waldo back to the atom between the countdown blockers in time.

The player has to send Waldo downwards straight away, before the atom between the countdown blockers make that atom impossible to get.

## Challenges

If you get all of this working, here are some challenges to work on:

### Make the block actually disappear [fairly easy]

The original idea the person had was that the cell should disappear after some time. Make it look like the block is disappearing instead of just changing to a 'blocking' appearance.

### A block which starts off 'blocking' but changes to 'allowed' [medium]

Make a sprite which works backwards to the one we made in this worksheet. It should start off in a state which blocks Waldo, but then after some number of Waldo-steps, it should change into something which lets Waldo pass over it. Design a level using both types of counting blocker.

### A block which flips between 'blocking' and 'allowed' [bit harder]

Put both these ideas together, to make another sort of sprite which spends some number of Waldo-steps in the 'OK to go over' state, and then some in the 'blocks Waldo' state.

### Only count down when Waldo passes over it [medium]

Instead of counting down every step, make the sprite count down only when Waldo goes over it.

Have your own ideas!

## Key points

Use a series of costumes to animate appearance.

Check 'costume#' to stop at the last costume.

Work round Scratch limitations using 'touching color' testing block.

Know where to extend existing code (Waldo's main loop) to add new behaviour.

---

## Extra information (not part of main worksheet)

It's not very clean to use the same message to make the atoms check whether they've been collected and to make the countdown blockers count down. A separate message for countdown could be better design; for example, it would let us choose whether the countdown happens before or after the test for 'has Waldo hit a blocked countdown blocker?'. Or we could use a different message name, but unfortunately Scratch doesn't allow you to change the name of a message once you've created it.

In the countdown blocker's scripts, we don't test we're not a clone when reacting to the 'reset' message. We get away with this, because the script says to switch to the '5' costume, which is what the original is using anyway. But this is a bit sloppy.



For finding out which costume another sprite is using, Scratch does have the block shown here. This looks like it would work for our situation. But that block reports the costume of the original of that sprite. I don't think there's a way to find out which costume a particular clone is using.

### About this document

Main content is copyright 2017 Ben North, and is hereby licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/). This and other worksheets available at <https://bennorth.github.io/scratch-worksheets-2017-2018/>.

Images contain material copyright The Scratch Team, used under a [Creative Commons Attribution-ShareAlike 2.0 license](https://creativecommons.org/licenses/by-sa/2.0/).

Scratch is developed by the Lifelong Kindergarten Group at the MIT Media Lab. See <http://scratch.mit.edu>.