

Table of Contents

[tl;dr](#)

[Help](#)

[Itching to Program?](#)

[How to Submit](#)

Problem Set 0

tl;dr

0. Watch [Week 0's lecture \(https://video.cs50.net/2016/fall/lectures/0\)](https://video.cs50.net/2016/fall/lectures/0).
1. Program in Scratch.
2. Submit a form.

Help

For help with Week 0 and Problem Set 0:

- Watch [Zamyla's walkthrough \(https://youtu.be/697pD31GCZg\)](https://youtu.be/697pD31GCZg).
- Start a [discussion \(https://courses.edx.org/courses/course-v1:HarvardX+CS50+X/a7ec0c0a7b6e460f877da0734811c4cd/\)](https://courses.edx.org/courses/course-v1:HarvardX+CS50+X/a7ec0c0a7b6e460f877da0734811c4cd/) with classmates.

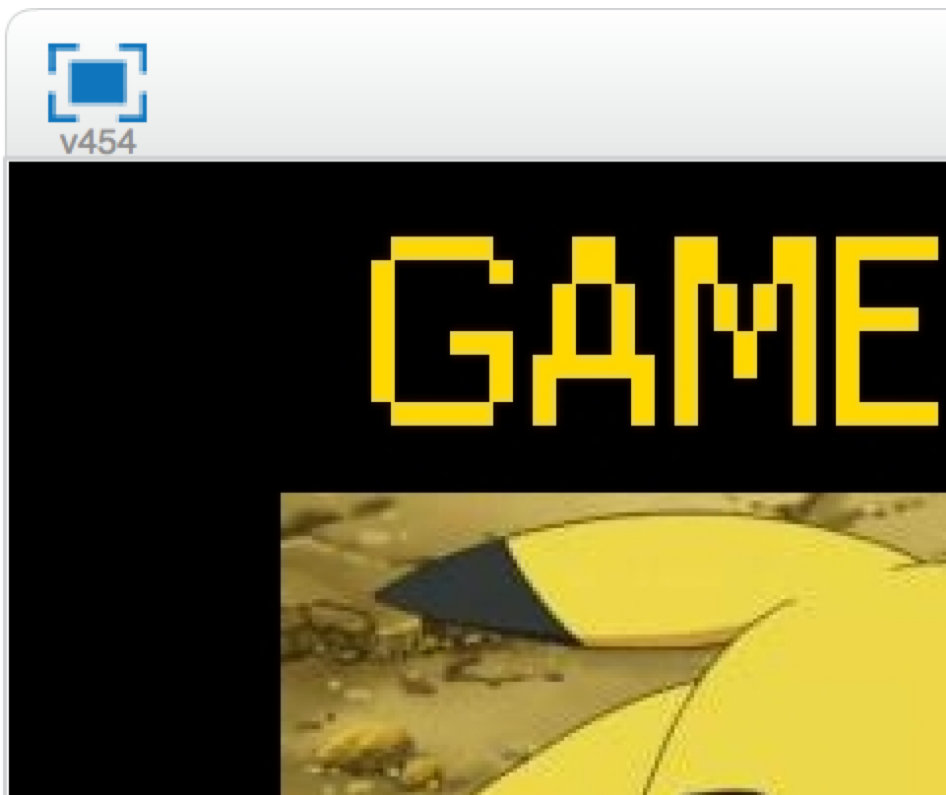
Itching to Program?

Head to <https://scratch.mit.edu/> (<https://scratch.mit.edu/>) and sign up for an account on MIT's website by clicking **Join Scratch** atop the page. Any username (that's available) is fine, but take care to remember it and your choice of password.

Then head to <https://scratch.mit.edu/help/> (<https://scratch.mit.edu/help/>) and take note of the resources available to you before you dive into Scratch itself. In particular, you might want to skim the [Getting Started Guide](#)

(https://cdn.scratch.mit.edu/scratchr2/static/_95f8025b5d5663c8eca07b96a66ef8d6_/pdfs/help/Getting-Started-Guide-Scratch2.pdf).

Next try your hand at *Pikachu's Pastry Catch* by (former student) Gabe Walker! Click the green flag and then, per Gabe's instructions, hit your keyboard's space bar, at which point the game will begin! Feel free to procrastinate a bit.



If curious, Gabe's source code can be seen at <http://scratch.mit.edu/projects/26329354/> (<http://scratch.mit.edu/projects/26329354/>). (You can also full-screen the game at that same URL, as full-screening the embedded game here might not work.)

Next, be sure you know what's recyclable and compostable these days by trying out this remix of *Oscartime* by Jordan Hayashi!



Jordan's source code can be found at <https://scratch.mit.edu/projects/71161586/> (<https://scratch.mit.edu/projects/71161586/>). (You can also full-screen that game at that same URL.)

If you've no experience (or comfort) whatsoever with programming, rest assured that Gabe's and Jordan's projects are more complex than what we expect for this first problem set. (Click **See inside** in Scratch's top-right corner to look at each project's underlying "implementation details.") But they do reveal what you can do with Scratch.

In fact, for a gentler introduction to Scratch (and programming more generally), you might want to review some of the examples that we looked at in Week 0, the "source code" for which can be found at <https://scratch.mit.edu/studios/3003963/> (<https://scratch.mit.edu/studios/3003963/>). Once you can say to yourself, "Okay, I think I get this," you're ready to proceed.

Now it's time to choose your own adventure! Your mission is, quite simply, to have fun with Scratch and implement a project of your choice (be it an animation, a game, interactive art, or anything else), subject only to the following requirements.

- Your project must have at least two sprites, at least one of which must resemble something other than a cat.
- Your project must have at least three scripts total (i.e., not necessarily three per sprite).
- Your project must use at least one condition, one loop, and one variable.

- Your project must use at least one sound.
- Your project should be more complex than most of those demonstrated in lecture (many of which, though instructive, were quite short) but it can be less complex than *Oscartime* and *Pokemon Go*. As such, your project should probably use a few dozen puzzle pieces overall.

Feel free to peruse some of last year's projects (<https://scratch.mit.edu/studios/1493562/>) for inspiration, but your own project should not be terribly similar to any of them. Try to think of an idea on your own, and then set out to implement it. But don't try to implement the entirety of your project all at once: pluck off one piece at a time. Gabe, for instance, probably implemented just one pastry first, before he moved onto the game's other sprites. In other words, take baby steps: write a bit of code (i.e., drag and drop a few puzzle pieces), test, write a bit more, test, and so forth.

If, along the way, you find it too difficult to implement some feature, try not to fret; alter your design or work around the problem. If you set out to implement an idea that you find fun, you should not find it hard to satisfy this problem set's requirements.

Alright, off you go. Make us proud!

Incidentally, if you don't have the best Internet access, you're welcome to download Scratch's "offline editor" at <https://scratch.mit.edu/scratch2download/> (<https://scratch.mit.edu/scratch2download/>). But when done with your project offline, be sure to upload it to your account at <http://scratch.mit.edu/> (<http://scratch.mit.edu/>) via **File > Share to website** in the offline editor.

Once finished with your project, click **See project page** in Scratch's top-right corner. Ensure your project has a title (in Scratch's top-left corner), some instructions (in Scratch's top-right corner), and some notes and/or credits (in Scratch's bottom-right corner). Then click **Share** in Scratch's top-right corner so that others can see your project. Finally, take note of the URL in your browser's address bar. That's your project's URL on MIT's website, and you'll need to know it later.

Oh, and if you'd like to exhibit your project in CS50x 2017's studio, head to <https://scratch.mit.edu/studios/3678335/> (<https://scratch.mit.edu/studios/3678335/>), then click **Add projects**, and paste in your own project's URL.

How to Submit

1. Submit [this form \(https://newforms.cs50.net/2017/x/psets/0\)](https://newforms.cs50.net/2017/x/psets/0)!
2. Visit [cs50.me/account \(https://cs50.me/account\)](https://cs50.me/account), and follow the link to connect your edX account to your GitHub account.

Your submission should be graded within 2 weeks, at which point your score will appear in [CS50.me \(https://cs50.me/gradebook/course/cs50/2017/x\)](https://cs50.me/gradebook/course/cs50/2017/x)!

This was Problem Set 0.