Table of Contents

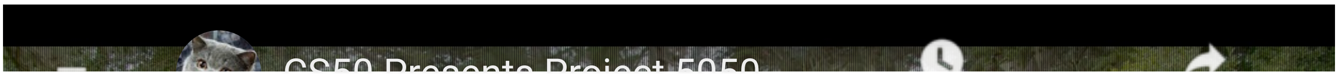
Problem Set 0

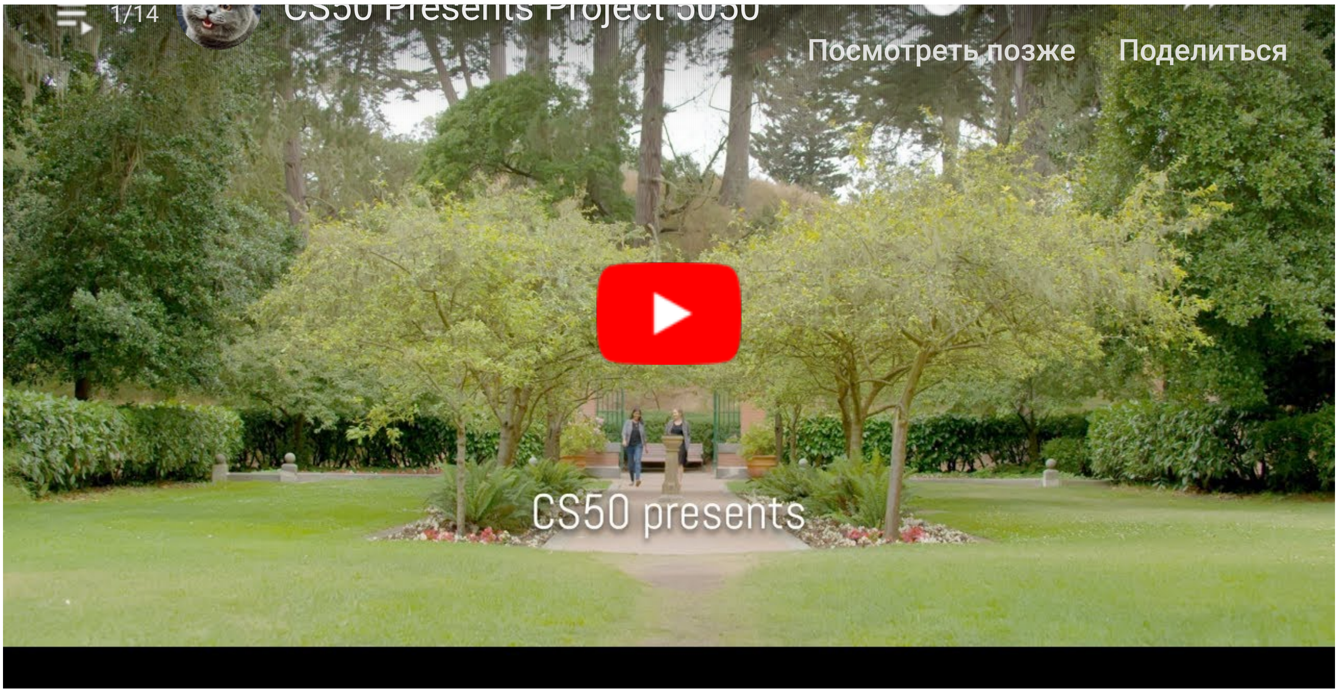
tl;dr

0. Watch Week 0’s lecture (https://video.cs50.net/2017/fall/lectures/0).
1. Program in Scratch.
2. Upload a file.
3. Submit a form.

Project 5050

If you’d like to learn a bit more about others who’ve come before you, take a look at Project 5050 (http://project5050.org/) by CS50’s own Hailey James '19 et al.





Hello, World

Suffice it to say it’s a bit harder to meet classmates when taking a course online. But, thanks to technology, everyone can at least say hello!

If you have a phone (or digital camera) and would like to say hello to classmates, record a 1- to 2-minute video of yourself saying hello, perhaps stating where in the world you are, why you’re taking CS50x, and something interesting about you! Try to begin your video by saying "hello, world" and end it with "my name is …., and this is CS50." But, ultimately, it’s totally up to you.

If you do record a video, upload it to YouTube (unless blocked in your country, in which case you’re welcome to upload it elsewhere) so that you can provide us with its URL when you submit!

GitHub

In order to submit problem sets, you'll need an account on GitHub (https://github.com/), a popular service with which to store code and collaborate with others. If you don't already have one, visit https://github.com/join (https://github.com/join) and create an account (for free). Take care to remember your username and password! If you'd prefer not to provide GitHub with a personally identifiable username and email address (i.e., one that includes your name), you're welcome to create an account using an anonymous username and some other email address (e.g., that you only use for GitHub). If you choose to provide a personally identifiable username or email address, material you submit for the course to GitHub and our feedback on it also will be personally identifiable by GitHub. See GitHub's Privacy Statement (https://help.github.com/articles/github-privacy-statement/).

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.

Make sure that your web browser has Flash enabled. If you're using Google Chrome, you can go to `chrome://settings/content`, click on Advanced → Content settings, and make sure you "allow" Flash!

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.

Oh no! We're having trouble displaying this Scratch project.
If you are on a mobile phone or tablet, try visiting this project on a computer.
If you're on a computer, your Flash player might be disabled, missing, or out of date. Visit this page to update Flash.

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!

Oh no! We're having trouble displaying this Scratch project.
If you are on a mobile phone or tablet, try visiting this project on a computer.
If you're on a computer, your Flash player might be disabled, missing, or out of date. Visit this page to update Flash.

Get ADOBE® FLASH® PLAYER

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*. 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/3009443/) 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 Fall 2017's studio, head to https://scratch.mit.edu/studios/4248580/ (https://scratch.mit.edu/studios/4248580/), then click **Add projects**, and paste in your own project's URL.

How to Submit

Step 1 of 2

1. After creating a GitHub account (https://github.com/join), visit CS50.me (https://cs50.me), log in with your GitHub account, and click **Authorize submit50**.
2. You should receive two emails from GitHub inviting you to join CS50's GitHub organizations. Click the button to accept both of these invitations.
3. On your Scratch editor page (e.g. https://scratch.mit.edu/projects/123456789/#editor (https://scratch.mit.edu/projects/123456789/#editor)), go to the **File** menu and click **Download to your computer**. Make sure to save your file with a `.sb2` extension!
4. Go to `https://github.com/submit50/USERNAME (https://github.com/submit50/USERNAME)`, replacing `USERNAME` with your own GitHub username.
5. On the left side of the screen, click on **Branch: master**.
6.

In the field that says **Find or create a branch…**, copy and paste the `cs50/2018/x/scratch`, and then click **Create branch**.

7. Click the button that says **Upload files**.
8. Drag your `project.sb2` Scratch file into the box that says **Drag files here**.
9. Click the green **Commit changes** button.
10. You're done (with step 1)! If you go back to CS50.me (https://cs50.me), you should see your Scratch submission. If you click on the **check50** link next to the submission, you should see (after a minute or two) whether your project met all of the requirements. You are welcome to resubmit as many times as you'd like (before the deadline)!

If you don't see results on cs50.me[https://cs50.me/ (https://cs50.me/)], make sure that your branch is named `cs50/2018/x/scratch`. If you mistyped it the first time, you should create a new branch (repeating steps 5 and 6), making sure to name it `cs50/2018/x/scratch`!

Step 2 of 2

Submit https://forms.cs50.net/2018/x/psets/0 (https://forms.cs50.net/2018/x/psets/0).

This was Problem Set 0.

FAQs

Can't find the "Branch: master" button.

If you don't see a "Branch: master" button upon going to your submit50 page on GitHub, click the "README" button next to "We recommend every repository include a…" and then click the green "Commit new file" button. After that, the "Branch: master" button should appear!