

Problem Set 0: Scratch

due by noon on Thu 9/13

Per the directions at this document's end, submitting this problem set involves filling out a Web-based form that may take a few minutes, so best not to wait until the very last minute, lest you spend a late day unnecessarily.

Objectives.

- Introduce some fundamental programming constructs.
- Empower you to design your own animation, game, or interactive art.
- Impress your friends.

Academic Honesty.

All work that you do toward fulfillment of this course's expectations must be your own unless collaboration is explicitly allowed in writing by the course's instructor. Collaboration in the completion of problem sets is not permitted unless otherwise stated by some problem set's specification.

Viewing or copying another individual's work (even if left by a printer, stored in an executable directory, or posted online) or lifting material from a book, website, or other source—even in part—and presenting it as your own constitutes academic dishonesty, as does showing or giving your work, even in part, to another student or soliciting the work of another individual. Similarly is dual submission academic dishonesty: you may not submit the same or similar work to this course that you have submitted or will submit to another. Nor may you provide or make available solutions to problem sets to individuals who take or may take this course in the future. Moreover, submission of any work that you intend to use outside of the course (*e.g.*, for a job) must be approved by the course's instructor or preceptor.

You are welcome to discuss the course's material with others in order to better understand it. You may even discuss problem sets with classmates, but you may not share code. In other words, you may communicate with classmates in English, but you may not communicate in, say, C. If in doubt as to the appropriateness of some discussion, contact the course's instructor or preceptor.

You may turn to the Web for instruction beyond the course's lectures and sections, for references, and for solutions to technical difficulties, but not for outright solutions to problems on problem sets or your own final project. However, failure to cite (as with comments) the origin of any code or technique that you do discover outside of the course's lectures and sections (even while respecting these constraints) and then integrate into your own work may be considered academic dishonesty.

All forms of academic dishonesty are dealt with harshly. If the course refers some matter to the Administrative Board and the outcome for some student is *Admonish*, *Probation*, *Requirement to Withdraw*, or *Recommendation to Dismiss*, the course reserves the right to impose local sanctions on top of that outcome for that student that may include, but not be limited to, a failing grade for work submitted or for the course itself.

Fine Print.

Your work on this problem set will be evaluated along two axes primarily.

Scope. To what extent does your code implement the features required by our specification?

Correctness. To what extent is your code consistent with our specifications and free of bugs?

All students, whether taking the course Pass/Fail or for a letter grade, must ordinarily submit this and all other problem sets to be eligible for a passing grade (*i.e.*, Pass or A to D–) unless granted an exception in writing by the course's instructor or preceptor. No more than one late day may be spent on this, or any other, problem set.

Getting Started.

- ☒ O hai! Know that help with this and future problem sets is available not only at office hours, per the schedule on the course's website, but also via `cs50.net/discuss`. We'll do our best to reply to posts on the latter within 24 hours outside of office hours!

A Section of Questions.

- ☐ Head to

`https://www.cs50.net/shorts/`

and watch the shorts on ASCII and binary, if not too familiar.

Consider these questions rhetorical for now, but odds are they'll come up again!

- ☐ How do you represent the (decimal) integer 50 in binary?
- ☐ How many bits must be "flipped" (*i.e.*, changed from 0 to 1 or from 1 to 0) in order to capitalize a lowercase 'a' that's represented in ASCII?
- ☐ How do you represent the (decimal) integer 50 in, oh, "hexadecimal," otherwise known as base-16? Recall that decimal is simply base-10, and binary is simply base-2. Infer from those base systems how to represent this one!

Itching to Program?

- ☐ Surf on over to the URL below and sign up for an account on MIT's website by clicking **Signup** atop the page.

`http://scratch.mit.edu/`

Any username (that's available) is fine, but take care to remember it and your choice of password.

- ☐ Then head to the URL below and take note of the resources available to you before you dive into Scratch itself.

`http://info.scratch.mit.edu/Support`

In particular, you might want to skim the **Getting Started Guide**.

- ☐ Next, read through the tutorial below, which presents Scratch in terms a (soon-to-be) programmer should understand.

`http://cs.harvard.edu/malan/scratch/`

- ☐ If not already installed on the computer you're using, download Scratch itself via the URL below.^{1,2}

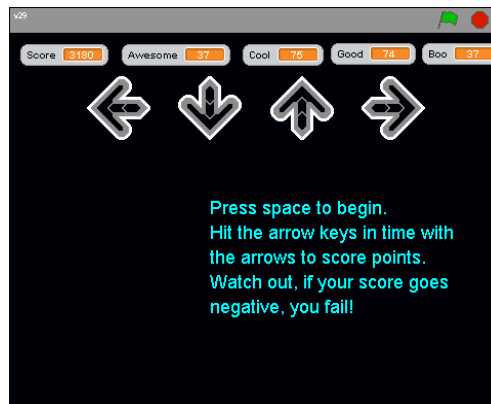
<http://scratch.mit.edu/download>

Take care to download version 1.4. Once downloaded, install the software by double-clicking its icon and/or following any on-screen directions. If unsure how to install, don't hesitate to post with a **pset0** label at cs50.net/discuss!

- ☐ Next head to

<http://scratch.mit.edu/projects/lindorniel/40241>

to see **Scratch Scratch Revolution** by Ann Chi '08 (aka `lindorniel`). You should find yourself at MIT's website with Ann's project (screenshot below) embedded in your browser.



If you do not see Ann's project, you may need to download and install a "Java Runtime Environment" (aka "Java Virtual Machine" or "JVM"). To do so, follow the link to **Java** at <https://www.cs50.net/software/>, follow any on-screen instructions, then return to MIT's website. If unsure how to install, don't hesitate to ask at cs50.net/discuss!

Once Ann's project has loaded in your browser, press your space bar to begin!³ See if you can't beat some of the high scores claimed in the page's **Comments**. Better yet, email the page's URL to a friend, and see if you can't beat him or her!

- ☐ If you've no experience (or comfort) whatsoever with programming, rest assured that Ann's project is more complex than what we expect for this first problem set. But it does reveal what you can do with Scratch.

¹ Installers for Linux are available at http://info.scratch.mit.edu/Linux_Installer.

² Students running an operating system other than a supported version of Linux, Mac OS, or Windows should please find a computer with a supported operating system to use for this problem set (e.g., in the Science Center, a house's lab, or a friend's room).

³ If the game doesn't start, try clicking the green flag in the project's top-right corner, then hit your space bar again.

Let's take a look at one other project. Head to

<http://scratch.mit.edu/projects/cs50/37413>

to see a project you probably haven't yet seen by Carlos Herrera. This game, too, you can play within the confines of your browser. But let's download its "source code" (*i.e.*, its `.sb` file). Click the link labeled **cherrera** in the page's top-right corner, just below **Download this project!** If told by a pop-up to log in, click the link labeled **Login** atop the page and log in with the username and password that you created earlier for MIT's website, then click **cherrera** again. If prompted to open or save the file (called `cherrera.sb`), go ahead and save it to your desktop so that it's easy to find.

Then launch the non-browser version of Scratch (*i.e.*, the one that you downloaded and installed earlier) by clicking its own icon on your desktop, in your Applications folder, in Program Files, or wherever you installed it. Click **File → Open...** toward Scratch's top-left corner, and navigate your way to Carlos's project, wherever it ended up on your hard drive.

Spend some time looking over Carlos's scripts. Don't forget that each sprite has its own set of scripts. Try to get a sense of how the overall program works. Try making some changes, even while the program is running, to see how the program responds. Note that this project is probably a bit simpler than we expect of you for this problem set, but it's a good one to learn from because it's pretty easy to follow. And do appreciate that this game, like all Scratch projects, reduces quite literally to some basic building blocks.

- ☐ Feel free to download the source code for a few more projects, either from

<http://scratch.mit.edu/galleries/>

or from Week 0 at

<https://www.cs50.net/lectures/>

even if you already saw it in lecture. Or pull up some of the projects that come with Scratch by clicking **File → Open...** toward Scratch's top-left corner followed by the icon labeled **Examples** in the window that appears.

For each program, run it to see how it works overall and then look over its scripts to understand how it works underneath the hood. Feel free to make changes to scripts and observe the effects. 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's filename must be `username.sb`, where `username` is your MIT username in all lowercase.

- ☐ 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, say, Scratch Scratch Revolution. As such, your project should probably use a few dozen puzzle pieces overall.

Feel free to peruse additional projects online or those that come with Scratch 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. Ann, for instance, probably implemented just one arrow first, before she moved on to her game's other three. And Carlos probably implemented a stationary goal before he tried to make it move up and down on its own.

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. If you suspect your program might fall short of our expectations, feel free to ask a member of the staff for an opinion prior to submitting.

Alright, off you go. Make us proud! If you have questions or want a hand making your project even better, do take advantage of office hours. Alternatively, post questions at cs50.net/discuss!

- ☐ Once finished with your project, upload it to your account on MIT's server by clicking **Share → Share This Project Online...** toward the top of Scratch's window. Provide "Your Scratch website login name" (*i.e.*, username) and password that you chose earlier, some project notes (*e.g.*, some instructions or a description), and zero or more tags; ensure that your "Project name" is your MIT username. Best to leave **Compress sounds and images** checked. Then click the **OK** button. Assuming you're informed that "Your project is now online at scratch.mit.edu," head on over to the URL below.⁴

<http://scratch.mit.edu/login>

Log in (if you aren't already logged in) with your username and password. Once logged in, click **my stuff** toward the page's top-right corner. You should see the project you just uploaded among **My Projects**. Go ahead and click its name or icon. Your project should be embedded (and may

⁴ If informed that your project is too large to be uploaded, try to decrease its size by clicking **Edit → Compress Sounds...** and/or **Edit → Compress Images...** atop Scratch's window. You may need to experiment with different levels of compression. Also take care to remove from each of your sprites any sounds that you imported but ended up not using. If, despite many attempts, you are still unable to upload your project to MIT's website because of its size, best to turn to cs50.net/discuss for advice.

start running) in the window that appears. 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.

Note that some projects, particularly those with sounds or lots of sprites, do not function properly when embedded in MIT's website. Not to worry! We'll be sure to download your `.sb` file and open it in the non-browser version of Scratch.

- ☐ When done admiring your work, head on over to the URL below.

`https://www.cs50.net/galleries/2012/fall/scratch`

Make sure that you're still logged into MIT's website. (If you see "Welcome" followed by your username atop the page, you are.) If not, click **Login** to log in again.

Toward the page's right-hand side, click **add my projects** and wait for a window to appear. Once that window appears, click your own project's name, wait for a checkmark to appear to in the box to the left of it, then click **Accept**. If you return to the URL above, you should find that your project has been added to Fall 2012's gallery for others to enjoy. If not, try once more or reach out to the course's staff for assistance.

If you really, really don't want your work included in the course's gallery, you may opt out of this particular step. But you must still upload your project to MIT's server (by clicking **Share** → **Share This Project Online...**), per the previous step. And you must still "submit" your project to us, per this document's end.

How Stuff Works.

- ☐ Head to

`http://www.explainthatstuff.com/harddrive.html`

and read up on how hard disk drives (aka HDDs) work. You may also want to re-watch the two videos we saw on Friday of Week 0, both of which are linked under Week 0 at:

`https://www.cs50.net/lectures/`

Once you feel you've wrapped your mind around the technology, try explaining how HDDs work in a few sentences verbally to a roommate or friend. (For real!) There are far too many people in this world who know *what* they're talking about but do not know how to explain what they know to non-technical people, so it's very good practice!⁵

⁵ Plus, we may ask you to explain them to us before long!

How to Submit.

- ☐ To submit this problem set, head to the URL below.

`https://www.cs50.net/psets/0/`

You'll find that a few questions await. Be extra-sure that your answers are correct, particularly your email address(es) and your project's URL, else we may overlook your submission! And be sure to click **Submit** on the last page of that form in order to submit this, your first CS50 problem set!

- ☐ kthxbai!