

Problem Set 0: Scratch

out of 60 points

due by 7:00 P.M. on Friday, 28 September 2007,
via submission on `nice.fas.harvard.edu`
per the directions at this document's end

Goals.

The goals of this problem set are to:

- Empower you to implement your own animation, game, or interactive art.
- Introduce you (gently) to SSH, SFTP, Linux, and Nano.

Getting Started.

0. Per the course's syllabus, obtain an FAS (Faculty of Arts and Sciences) Computer Account, if you haven't one already, by visiting the URL below and following the on-screen instructions.

`https://www.fas.harvard.edu/computing/utilities/activate-pin/`

Not only will this account will provide you with access to FAS's computer facilities, it will also provide you with an email address of the form `username@fas.harvard.edu`, where `username` is your FAS username. Additionally, this account will allow you to access via SFTP and SSH `nice.fas.harvard.edu`, FAS's New Linux Computing Environment, use of which is required by this and future problem sets. SFTP and SSH clients are available for various platforms via the course's website. Usage thereof will be taught in section and office hours.

Itching to Program?

1. Surf on over to the URL below and sign up for an account on MIT's website.

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

Take care to remember your choice of username and password.

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

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

Of particular help may be the links to **Getting Started**, **Help Screens**, **Reference Guide**, **Scratch Cards**, and **Videos**.

You might also want to skim the tutorial below, which presents Scratch in terms a (soon-to-be) programmer should understand.

<http://www.fas.harvard.edu/~cs50/resources/scratch/>

3. If not already installed on the computer you're using, download Scratch itself via the URL below.¹

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

Once downloaded, install the software by double-clicking its icon and following any on-screen directions.

4. Head to

<http://www.fas.harvard.edu/~cs50/lectures/weeks/0/src/>

and download the source code for Raining Men (*i.e.*, `Raining Men.sb`), written by CS 50's own Andrew Berry, to your desktop. Open the file in Scratch, either by double-clicking its icon or by double-clicking Scratch's (whiskered) icon and navigating your way to Andrew's file by way of the **Open** button toward Scratch's top-left corner.

Summon two friends to your computer, click the green flag toward Scratch's top-right corner, sit back, and enjoy.

Tell your friends they can play Andrew's project on their own computers (without downloading Scratch itself) simply by visiting the URL below.²

<http://www.fas.harvard.edu/~cs50/shortcuts/men>

5. Send your friends away and spend some time looking over Andrew'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. It's perfectly okay if you don't quite follow how everything works. Bear in mind that this project was implemented by someone who has already completed CS 50. But appreciate that the entire program (and its humor) reduces, quite literally, to some basic building blocks.

¹ Students running an operating system other than Mac OS or Windows should please find a computer with either of those operating systems to use for this problem set (*e.g.*, in the Science Center, a house's lab, or a friend's room).

² To play programs like Andrew's within a browser, they'll need to have a "Java Runtime Environment" installed on their computers, which they probably do. If not, they (and you) can download Java 6 Update 2 for free via the link to **Software** on the course's website.

6. Go ahead and download the source code (*i.e.*, *.sb files) for a few more projects, either from

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

or from

<http://www.fas.harvard.edu/~cs50/lectures/weeks/0/src/>

even if you already saw it in lecture. For each program, run it to see how it works and then look over its scripts to understand how it works. 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.

7. (40 points.) Now your own adventure begins. Your task for this problem 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.
- i. Your project’s filename must be `username.sb`, where `username` is your FAS (not MIT) username.
 - ii. Your project must have at least two sprites, none of which may be a cat.
 - iii. Your project must have at least three scripts total (*i.e.*, not necessarily per sprite).
 - iv. Your project must use at least one condition, one loop, and one variable.
 - v. Your project must use at least one sound.
 - vi. 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, *Oscartime* or *Raining Men*. 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. 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 you find fun, you should not find it hard to satisfy this problem’s requirements.

If you suspect your program might fall short of our expectations, feel free to ask us for our opinion prior to submitting.

Alright, off you go. Impress us.

There shall be awards for **The Cutest**, **The Funniest**, and **The Most Amazing** submissions.

Do take advantage of the course’s bulletin board as well as office hours (virtual or otherwise), the schedule for which is available on the course’s website.

8. Once finished with your project, prepare to upload it to your account on MIT's server by clicking the **Share!** button toward the top of Scratch's window. Provide the login name (*i.e.*, username) and password that you chose earlier, along with a name for your project and some notes (*e.g.*, some instructions or a description). Then click the **OK** button. Assuming you're informed that "Your project is now online," head on over to the URL below.³

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

Log in (if you aren't still logged in) with your username and password. Once logged in, click **my stuff** toward the page's top-right corner. You should see your project among **My Projects**.

Then point your browser to the URL below.

`http://www.fas.harvard.edu/~cs50/galleries/scratch/`

Make sure that you're still logged in. (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**. In the window that appears, click your own project's name, wait for a checkmark to appear 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 CS 50's gallery for others to enjoy. If not, do try again or contact the staff for assistance.

If you absolutely do not want others to see your work, you may skip this whole sharing process altogether without penalty. But, even if you think your project The Ugliest, The Unfunniest, or The Least Amazing, you are encouraged to share it anyway!

If you thought that was nice...

9. If unfamiliar, peruse the document entitled **How to SSH to nice.fas.harvard.edu**, available via the link to **Resources** on the course's website.
10. Configure your FAS account for use in this course by SSHing to `nice.fas.harvard.edu` and executing the command below. Note that the tilde (~) is likely in your keyboard's top-left corner.

`~cs50/pub/bin/cs50setup`

³ If your project is larger than 10 megabytes (MB), your upload might be rejected by the server. Try to decrease its size (*e.g.*, by compressing any sounds with third-party software). But if shrinking your file's size would compromise your work's artistic integrity, you may skip this sharing process altogether without penalty.

You will then need to log out for the changes to take effect. Upon logging back in, you can confirm the changes' effect by executing the command below.

```
cs50check
```

If said command is “not found,” you probably failed to follow these directions correctly! For assistance with this process, simply contact the course's staff. Otherwise, proceed to type the command below.⁴

```
mkdir ~/cs50/
```

You've just created in a directory (*i.e.*, folder) called `cs50` in your so-called home directory (the shorthand for which is a tilde). All of your work will ultimately need to reside within this directory this semester for submission.

Next, execute the following command.

```
mkdir ~/cs50/ps0/
```

Perhaps needless to say, you've just created a directory called `ps0` within that `cs50` directory. Confirm as much by executing the command below.

```
find ~/cs50/
```

You should see output resembling the below, where `username` is your FAS username and `u` and `s` are the first and second characters thereof.

```
/home/u/s/username/cs50/  
/home/u/s/username/cs50/ps0
```

If you do not, you may want to re-try one or more of these steps or contact the staff for assistance.

11. (15 points.) Change your present working directory to `~/cs50/ps0/` by executing the command below.

```
cd ~/cs50/ps0/
```

To check that you are indeed in `~/cs50/ps0/`, execute the command below.

```
pwd
```

⁴ Note the difference between `~/cs50` in this command and `~cs50` in the previous command.

You should see output resembling the below.⁵

```
/nfs/home/u/s/username/cs50/ps0/
```

Needless to say, `~` is actually shorthand for `/nfs/home/u/s/username/`.

Open a new ASCII (*i.e.*, text) file called `questions.txt` by executing the command below.

```
nano questions.txt
```

Proceed to type answers for each of the questions below using Nano. Recall that you can move your cursor around within Nano using your keyboard's arrow keys but not your mouse. Useful keystrokes include `ctrl-v` (which pages down), `ctrl-y` (which pages up), `ctrl-o` (which saves), and `ctrl-x` (which quits).⁶

- i. What's your first and last name?
- ii. What's your FAS username?
- iii. What's your preferred email address?
- iv. What year are you?
- v. What's your (likely) concentration?
- vi. Why are you taking CS 50?
- vii. Any questions or concerns for your (soon-to-be) teaching fellow?
- viii. In a paragraph, tell us what your Scratch project does (or how to use it).
- ix. In one or more paragraphs, explain how your Scratch project works, noting the purpose of each sprite and script.
- x. In one or more paragraphs, point out any known bugs (*i.e.*, mistakes or quirks) in your Scratch project and explain why they're still there (*i.e.*, why you didn't or couldn't fix them).⁷
- xi. Floppy disks, as you (now) know, are a form of magnetic storage. CD-Rs, though, are not. In a short paragraph of your own words, how are bits stored on a CD-R?⁸

Upon quitting Nano, you can confirm that you did, in fact, create `questions.txt` by executing the command below.

```
ls
```

That command should list all files (or, in this case, the only file) in your present working directory.

⁵ Sometimes, though, `/nfs` is excluded from such output.

⁶ If prompted, upon quitting, to "Save modified buffer," you probably want to hit Y, since that's just an arcane way of asking you if you want to save your changes to the file.

⁷ We will not penalize any bug you disclaim. Only those bugs you don't disclaim (or even notice) will put points at risk!

⁸ For this question, you're welcome to consult *How Computers Work*, Google, Wikipedia, a friend, or anyone else, so long as your words are ultimately your own!

12. If unfamiliar, peruse the document entitled **How to SFTP to nice.fas.harvard.edu**, available via the link to **Resources** on the course's website.

SFTP to nice.fas.harvard.edu, navigate your way to ~/cs50/ps0/ (the hierarchy of directories you created earlier at the command line via SSH) by double-clicking the icon for cs50 followed by the icon for ps0, and upload your Scratch project (*i.e.*, username.sb) to the latter directory.

13. SSH to nice.fas.harvard.edu again (unless you left that window open, in which case you can continue using that session). Navigate your way at the command line to ~/cs50/ps0/. (Remember how?) Check that both questions.txt and username.sb are now in that directory. (Remember how?)

Submit those files by executing the command below.

```
cs50submit ps0
```

Thereafter, follow any on-screen instructions until you receive visual confirmation of your work's successful submission. You will also receive a "receipt" via email to your FAS account, which you should retain until term's end. You may re-submit as many times as you'd like; each resubmission will overwrite any previous submission. But take care not to re-submit after the problem set's deadline, as only your latest submission's timestamp is retained.

14. (4 points.) Section for the course anytime between 10:00 A.M. on Tuesday, 25 September 2007, and 7:00 P.M. on Friday, 28 September 2007 by visiting the URL below and following the on-screen instructions.

<http://www.section.fas.harvard.edu/>

You will be informed of your assigned section via email by 12:00 P.M. on Sunday, 30 September 2007.

15. (1 point.) That's it! Can you believe you're taking CS 50? Here's a free point for you.