Problem Set 0: ~~Scratch~~ BYOB

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.

diff pset0.pdf hacker0.pdf.

- Hacker Edition requires BYOB instead of Scratch.
- Hacker Edition recommends more technical reading on hard disk drives.

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?

☐ Rather than use Scratch for this Hacker Edition, you'll instead use BYOB,[1] an "advanced offshoot" thereof. Among BYOB's features that Scratch does not have are procedures (*i.e.*, functions), custom blocks, and recursion, along with more sophisticated lists and sprites. The only catch is that you won't be able to share your project on `scratch.mit.edu`, as in the standard edition. But you can still invite friends over to your laptop to see your creation!

☐ First, head to

`http://byob.berkeley.edu/#tutorials`

and watch Tutorials 1, 2, and 3. Then, visit

`http://byob.berkeley.edu/#download`

to download version 3.1.1 of BYOB and proceed to install!

Take care <u>not</u> to use Panther or Snap! 4.0, which are browser-based.

[1] No, not that. Build Your Own Blocks.

□ Now it's time to choose your own adventure! Your mission is, quite simply, to have fun with BYOB[2] 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 `surname.sb`, where `surname` is your last name.
- □ 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).
- □ Your project must use at least one custom block.

Feel free to peruse additional projects online or those that come with BYOB 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.

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 to `cs50.net/discuss`!

How Stuff Works.

□ Head to

`http://en.wikipedia.org/wiki/Hard_disk_drive`

and read up on how hard disk drives (aka HDDs) work. You may also want to re-watch the two videos we saw in 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

[2] Still not that.

this world who know *what* they're talking about but do not know how to explain what they know to non-technical people, so consider it very good practice![3]

Submitting Your Work.

☐ When you're ready to "submit" your project to us, surf on over to the URL below to answer some questions.

```
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!

[3] Plus, we may ask you to explain them to us before long!