Problem Set 8: Programming

due Monday, 5 January 2008, by 7:35 P.M.

You must submit this problem set via email to **psets@computerscience1.org**.

Survey Says!

1. (15 points.) Surf on over to the URL below.

   `http://computerscience1.org/survey/`

   Please take a few minutes to provide candid answers to the survey's questions. No need to provide your name; you may remain anonymous.

"Waiter, there's a bug in my ~~soup~~ pancake."

2. (10 points.) Suppose that the International House of Pancakes (IHOP) that just opened in Harvard Square has just purchased a robot to serve as its hostess. That is, the job of this robot is to seat the restaurant's patrons as they arrive. The restaurant's manager has configured this robot with the following program.

```
while restaurant is open for business
    if a group of patrons is waiting to be seated then
        if the group contains fewer than five patrons then
            seat the group at a small table
        else if the group contains more than five patrons then
            seat the group at a large table
    else
        wait for a group of patrons to arrive
```

   Sadly, this program contains a bug, the result of which is that certain groups are never seated. Figure out which type of group is never seated and suggest how to fix the bug.

Programming Your Lunch.

3. (25 points.) Suppose that IHOP has decided to replace its newly purchased hostess with a human being. Rather than discard the robot, the restaurant has decided to put it to work in the kitchen. You have been hired to re-program this robot to make peanut butter and jelly (PBJ) sandwiches. (The pancake business has been slow.)

   Write, in English, a program that instructs this robot how to make PBJ sandwiches. Think carefully about the steps involved in this process; leave nothing out of your instructions. Your program must comprise at least ten steps, and it must include at least one condition and at least one loop.

Its lines should be nicely indented and numbered. Your program will be graded on the basis of its correctness and thoroughness. That is, we will grade your answer by debugging your program.[1]

The first step of your program must be the following.

```
Locate jars of peanut butter and jelly, a loaf of bread, and a knife.
```

Itching to Program?

4. (0 points.) Surf on over to the URL below and sign up for an account on MIT's website.

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

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

5. (0 points.) Then head to the URL below and take note of the resources available to you before you dive into Scratch itself.

   `http://scratch.wik.is/Support`

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

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

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

6. (0 points.) If not already installed on the computer you're using, download Scratch itself via the URL below.

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

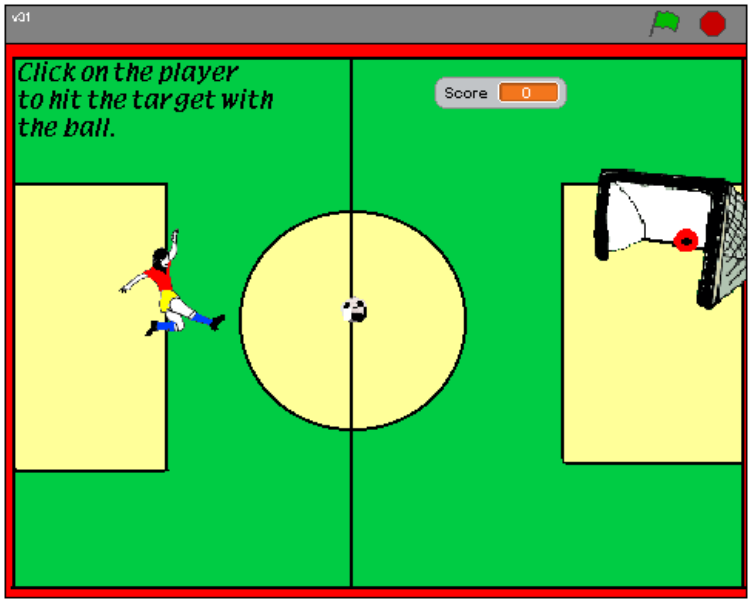
   Once downloaded, install the software by double-clicking its icon and/or following any on-screen directions.

[1] Frankly, we'll be surprised if you can write a correct and thorough program for making PBJ sandwiches in only ten steps. Try to break (and then fix) your program before we get the chance!

7. (0 points.) Head to

`http://scratch.mit.edu/projects/malan/37216`

and you should find yourself at MIT's website with Carlos Herrera's game (screenshot below) embedded in your browser.



If you do not see Carlos's game, you may need to download and install a "Java Runtime Environment" (aka "Java Virtual Machine" or "JVM"). To do so, head to

`http://java.com/en/download/manual.jsp`

and click the link appropriate for your operating system, following any on-screen instructions. Once done downloading and installing the software, return to MIT's website. If you run into any trouble, do not hesitate to contact the course's staff.

Once Carlos's game has loaded in your browser, give it a try![2]

Now let's have you download the "source code" (*i.e.*, the `.sb` file) for Carlos's game. 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** toward the page's top-left corner 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, or wherever you

[2] If the game doesn't start, try clicking the green flag in the project's top-right corner.

installed it. Click **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.

8. (0 points.) Next open up one or more of the projects that come with Scratch by clicking **Open** toward Scratch's top-left corner followed by the icon labeled **Projects** in the window that ultimately 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.

9. (50 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 must have at least two sprites, at least one of which must resemble something other than a cat.
   ii. Your project must have at least three scripts total (*i.e.*, not necessarily three per sprite).
   iii. Your project must use at least one condition, one loop, and one variable.
   iv. Your project must use at least one sound.
   v. 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 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!

10. (0 points.) Once finished with your project, upload it to your account on MIT's server by clicking the **Share!** button toward the top of Scratch's window. Provide "your Scratch website 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 at scratch.mit.edu," 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 the project you just uploaded among **My Projects**. Go ahead and click its name or icon. Your project should be embedded (and start playing) 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.

11. (0 points.) When done admiring your work, head on over to the URL below.

`http://scratch.mit.edu/galleries/view/36081`

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**. In the window that 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 the course's Fall 2008 gallery for others to enjoy. If not, try once more or contact the 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!**), per the previous step. And you must still "submit" your project to us, per the next step.

12. (0 points.) Rather than email us your project (which is likely to be pretty large), simply tell us, in your email to **psets@computerscience1.org**, what the URL of your project is on MIT's website.

[3] If informed that your project is larger than 10 MB (and thus too large to be uploaded), try to decrease its size by clicking **Extras** atop Scratch's window followed by **compress sounds** and/or **compress images**. You may need to experiment with different levels of compression in order to get your project below 10 MB.