

A1: Top-down Problem Solving

Due	Jan 23 by 11:59pm	Points	100	Submitting	a file upload	File Types	pdf
------------	-------------------	---------------	-----	-------------------	---------------	-------------------	-----

Goal

The goal of this assignment is to practice the type of precise problem solving required in computer science, as well as top-down problem solving.

You will use an online flash game, called Light Bot 2, during the assignment. Light Bot 2 is available from a number of locations. My recommended location is

<http://coweb.cc.gatech.edu/ice-gt/1835> (<http://coweb.cc.gatech.edu/ice-gt/1835>)

Please check early on during the assignment period whether or not you are able to successfully load the game - it uses Flash, which is a common plugin for browsers and likely already available the machine you choose.

Light-Bot Background

In the game, hit Play and then start in the Basics area, where you learn the mechanics of the game. Along the upper right are available instructions you can use. Along the right side are slots where you can place instructions for the robot to follow. Note that some instructions or slots may be colored pink to show they are unavailable for a particular task.

You may want to view a short helper video from <http://ice-web.cc.gatech.edu/dl/?q=node/628> (<http://ice-web.cc.gatech.edu/dl/?q=node/628>) to get started, but the tutorial part of the game is also helpful.

You can place instructions in two different ways. You can either click the instruction, then click the destination, or (I find this easier) you can click the larger destination area containing the slots and then just click on instructions to repeatedly fill the slots. The Basics area starts with some Tutorial levels that have you work through the mechanics of working the game.

This game provides some elements of a computer language. The main instructions perform some operation on the robot. Your ability to solve the task is limited by the types of instructions and limits on the number of instructions used, similar to limited memory on a computer. The puzzles require you to creatively problem-solve.

Along with the built-in instructions are ways of grouping sets of instructions and reusing them, called Function1 and Function2 in the game. The functions act as the more detailed levels in a top-down design. When you see a pattern in the game, it is useful to put that in a Function area. There are also ways of conditionally performing actions by using colors, and ways of repeating actions over and over, called recursion in this game. These are some of the fundamental building blocks of other computer languages.

Describing Your Solution

You will be asked to solve certain levels and to answer some additional questions about them. In order to record your answer, you will need to take a screenshot of the program area. On most PC computers, the standard Snipping Tool is an easy way to capture a part of your screen (search for it on your machine). On a Mac, command-shift-4 allows you to drag a rectangle and capture to a file, command-control-shift-3 saves an area to the clipboard for pasting into a document).

Note that when you successfully complete a level, the game moves on without giving you a chance to write down your answer. Keep track of what you are doing as you test and refine your solution or pause it just before the final bit and capture your solution.

Assignment

1. For a warm-up exercise on top-down problem solving, do a top-down description of how you get from your home to this class. The top block should be called "Get to CS1060", while other levels should break the process into steps. Make sure that steps at the same level have roughly the same amount of detail. Rather than having to draw blocks, please use the indented list style shown in the lecture notes and during lecture. **(20 pts.)**
2. Then, work with the Light-Bot game to answer the following questions. First do the first 5 levels in the Basic area in order to gain familiarity with the system, without recording your solutions. Once you understand the basics of the game, solve Basics level 6. This level is best solved by using a top-down approach, where a general problem is broken into smaller pieces. **(40 pts.)**
 1. Using the indented list style demonstrated in the lecture notes, write down what conceptual blocks you use in your answer to solve this problem. These are not the LightBot commands, but rather, your own description of what the game character is doing. Give each line a terse heading that describes it. Your answer should start with a main block called "Solve Puzzle", then be followed with at least two more levels of indenting detailing your solution.
 2. Now, give your solution to the puzzle by pasting in a screenshot of your solution. Your description should match your solution.
3. Do level 1 in the Recursion area for practice, then solve Recursion, level 2. Recursion is when a function calls itself. Normally, in a programming language, you have a test to end the recursion or it goes forever, while in the game, lighting all the blocks causes the program to terminate. **(40 pts.)**
 1. Solve the puzzle for level 2 and provide a screenshot of the solution.
 2. How many LightBot instructions did you use? How many times were instructions called? This information is available at the bottom of the screen. You may want to slow the game down by clicking on the leftmost setting of the "System Clock" in the game so you can more easily read this information. On a real computer, programs might need to run for trillions of steps to solve a problem. What does what you see in this example from LightBot imply for how big a program has to be to solve a problem that requires many steps like that? Does a program that runs a long time require a very large set of instructions? Write a paragraph with your thoughts on that question.

Type your answers and submit them as a pdf file. If you are unsure how to create a pdf, Mac computers can do this as one of the print or save as options, while on Windows you can use a number of tools. I have installed CutePDF Writer as a fake printer, so when you print to it, it makes a pdf for you

(<http://www.cutepdf.com/products/cutepdf/writer.asp>). If you are using a campus machine, look for Acrobat as a way to convert. MS Word has a pdf option, press "Save As" then at the bottom of the pop-up window in the "Save as Type" box, select "PDF".

Contact me for more help if you have specific issues with this conversion. Students at the U currently have free access to install the MS Office suite on a personal Mac or Windows machine, so a word doc (saved as a pdf) is an easy way to include text and screenshot images.

Files are submitted through the Canvas website.