

Homework: Structures and variants, part II

COMP 50

Fall 2013

This homework is due at 11:59PM on **Monday, September 30**.

Submit them using the COMP 50 Handin button on DrRacket; the homework is the `structures-ii` homework.

All the exercises should be done using the *Beginning Student Language*.

Finger Exercises

For this homework I am recommending the *same* finger exercises as for the “Structures I” homework:

- In the First Edition *How to Design Programs*, Exercises 6.4.1, 6.4.3, 6.5.1, 6.6.1 (but instead of a symbol for color, please use a string), 6.6.3, 7.1.1, 7.1.3, 7.2.1, 7.2.2, 7.5.1, 7.5.3
- Develop data examples for the structure definitions in Exercise 6.4.1
- Write a function that converts a GPS position (as you define it below) to English

Problems to submit

1. While riding in his convertible with the top down, Professor Jacob is struck in the head by an errant baseball. When he recovers, he decides to hire you to create an experiment designed to confirm the validity of [Fitts’s Law](#), which you can look up on Wikipedia.

Professor Jacob asks you to design a `big-bang` program that reacts to mouse clicks (“button-down” mouse events). In case the experimental subject decides not to click the mouse, the experimenter must also be able to limit the number of seconds for which the program runs.

The program should draw the first mouse click as a seven-pointed, solid green star. At that point it should place a red circle at a random location on the canvas (`display`). The radius of the circle should be chosen randomly between 2 and 20 pixels.

The program should then wait either for time to expire or for the experimental subject to click *within* the red circle.

- If time expires, the program should write a consoling message over the canvas.
- If the experimental subject successfully clicks within the red circle, the program should display a congratulatory message that includes the time that elapsed between the initial mouse click and the successful mouse click. Professor Jacob would really like to measure time to the nearest tenth of a second.

After the message is written, the program should return a structure containing the information relevant to an experiment on Fitts’s Law. (DrRacket will continue to display the final result on the screen.) Professor Jacob is still feeling a little foggy from the effects of the baseball, so you will have to use the [Wikipedia page](#) to figure out what information to return and how to represent it.

Use all of the design recipes above, plus a *wish list*.

Domain knowledge: The Wikipedia page talks about a “device”, a “target”, and a “movement.” The device is not under your control and you can’t measure any of its properties directly. The “target” is the red circle you create and the starting point is the star. You are able to gather data about these aspects of the problem.

Domain knowledge: You will need the `2hntp/image` and `2hntp/universe` teachpacks. In the second edition, read [Section 3.6 \(Designing World Programs\)](#), especially [Figure 10: Signatures for interaction functions](#).

- To understand how a program be limited to a given number of seconds, read the documentation for `on-tick`.
- To understand how to respond to the mouse, read the first part of [Section 3.7 \(A Note on Mice and Characters\)](#) in the second edition.

Karma problems

- A. Write a program that is similar to the line-drawing program we talked about in lab, except that if the program is doing the red rubber-band thing, clicking the mouse should turn the rubber band into a black line and should instantly start a *new* rubber band anchored at the point of the mouse click. The program should stop rubber-banding when the Esc key is pressed.

This program will make it easy for you to draw figures with connected line segments.

- B. Run the Fitts’s Law experiment multiple times in succession and use the results to estimate the characteristics of your device.