due in 5day 15h

Submission Phase

1. Do assignment ☐ (/programdesign-001/human_grading/view/courses/970256/assessments/6/submissions)

Evaluation Phase

2. Train 🔒 (/programdesign-001/human_grading/view/courses/970256/assessments/6/trainingSets)
3. Evaluate peers 🔒 (/programdesign-001/human_grading/view/courses/970256/assessments/6/peerGradingSe

Results Phase

4. See results 🔒 (/programdesign-001/human_grading/view/courses/970256/assessments/6/results/mine)

☐ In accordance with the Honor Code, I certify that my answers here are my own work, and that I have appropriately acknowledged all external sources (if any) that were used in this work.

Save draft Submit for grading

Learning Goals

- Be able to comprehend data definitions for compound data.
- Be able to design functions operating on compound data.
- Be able to design world programs on compound data.
- Be able to use the on-key option to big bang.
- Be able to use the “one task per function rule to identify when to wish for a helper function.
- Be able to use a wish list to keep track of work remaining to be done in a large program design.

Introduction

You will complete the design of a simple one line text editor similar to the one you see when you send a text message on your phone.

We have started the design of this program, and you will complete it. You will see how the data definitions and wish-lists make it possible for you to finish a program someone else has started.

Once your editor is complete, starting it with `(main (make-editor "abcdef" 0))` should display the following editor:

The red line is the cursor. In this text editor, you will be able to insert and delete text and move the cursor

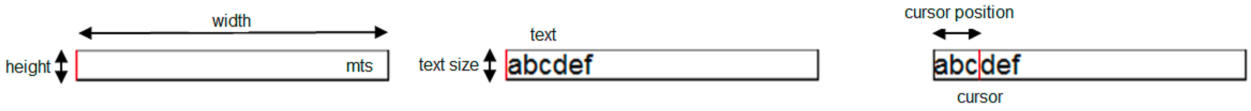
left and right. To help you with the project, we have made a demo video (https://class.coursera.org/programdesign-001/lecture/163) of what the program should do when it's done.

Main Project

One of the goals for this project is for you to get experience deciding when a function you are designing is doing "more than one task" and should therefore call a helper function. We are giving you a substantial start to the design. In some sense all you need to do is complete render-editor and handle-key. But note that in our solution, the handle-key function calls 4 helpers, which together call 2 more helpers. In all, we have 14 tests for handle-key and all of its direct and indirect helpers combined. Our render-key function calls two of the helpers that are indirectly called by handle-key.

Note that in the case of handle-key we are giving you part of the template, so you only need to complete the check-expects and code the function body.

Before you start programming, you should go through the domain analysis provided below.



Constant	Changing	big-bang options
width	text	on-key
height	cursor position	to-draw
text-size		
text-color		
cursor		
mts		

As you go through this domain analysis, be sure that you understand how we identified all of the constants, changing information and big-bang options. Then you should go through the starter file (https://spark-public.s3.amazonaws.com/programdesign/starters/editor-project-starter.rkt) to ensure you understand the constants, data definition, and main function provided there. Use the data definitions and wish-list entries in the program to determine what you need to do to finish it.

If you're unsure about how to complete the functions, one suggestion is to go through the examples and draw images of the examples to help you understand the correspondence between Editor and what appears on the screen. In other words **follow the recipe!**

When you are done the editor should implement at least the following functionality:

- The left arrow key should move the cursor left (unless already at left end of text).
- The right arrow key should move the cursor right (unless already at right end of text).
- The backspace key should delete the character before the cursor (if there is one).
- Key events of length 1 are normal characters, these should be inserted at the cursor position.

Challenge (Optional): Modify the program so that pressing the escape key ends the world program and returns the final editor state. You will need to change the world state data definition, and use the stop-when option to big-bang.

Grading

Note that the grading rubric will have the following criteria:

- **Correctness**
  - you can insert, delete, move the cursor left, and move the cursor right
- **Following Recipes**
  - handle-key has at least 5 check-expects (one for each of the four cases that have behavior and one for an ignored key like shift)

- all the helper functions have adequate check-expects
- all the functions have an appropriate signature
- all the functions have an appropriate purpose
- all the functions are based on the appropriate template
- you can insert, delete, move the cursor left, and move the cursor right

- **Using Helpers**
  - each case of handle-key calls a helper function
- **Coding Style**
  - there are no commented out tests or commented out code (except for stubs)
  - all the functions are named and indented properly

Correctness will be only worth 20% in our rubric. (But this does not mean you can hand in a blank file. You have to have made real progress on a solution to the problem to get any points.) You must follow the HtDF Recipe and the “one task per function” rule. While there are a number of reasonable solutions with different arrangements of helpers, a monolithic handle-key function and a monolithic render-editor function clearly do not follow the "one task per function" rule, and such solutions will not receive high scores according to our rubric.

Downloads

1. starter file (https://spark-public.s3.amazonaws.com/programdesign/starters/editor-project-starter.rkt)
2. demo video (https://class.coursera.org/programdesign-001/lecture/163)
3. subtitles (https://class.coursera.org/programdesign-001/lecture/subtitles?q=163_en&format=srt)

Submission

Upload your project below as a single .rkt file. If you don't submit your own assignment before the deadline, your work will not be graded by your peers, and you will not receive a grade. Make sure that your work is actually submitted, and not just saved.

Format

Words: 0

Upload file : Browse... No file selected.

Privacy choice: (read privacy policy (//class.coursera.org/programdesign-001/wiki/view?page=PrivacyPolicy))

I'd love to help out the course with my work. The staff may share my submission with my classmates

and other people who are curious, for things like peer assessment training, lecture slides, and publications.

In accordance with the Honor Code, I certify that my answers here are my own work, and that I have appropriately acknowledged all external sources (if any) that were used in this work.

Save draft | Submit for grading