Design of user interfaces

Square matrix filtration

2018/19 course - Practice 2

David Bohmann
Petr Lukašík
September 19, 2018

Contents

1. Assigment

The objective of this practice is to introduce the panels as well as the components of the slide bar and text area. The application should show in an area of text the elements of a square matrix of dimension 10x10 that exceed the value selected by the slide bar.

This application should allow the user:

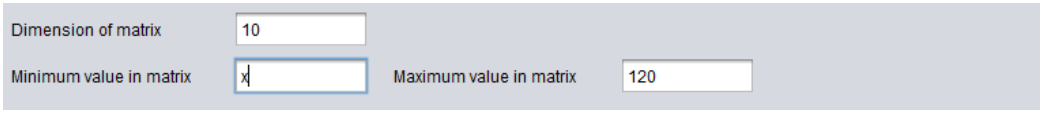
- Enter the maximum and minimum values contained in the matrix of whole.
- Show the square matrix in a text area with integer values generated randomly between the maximum and minimum values introduced.
- Include a sliding bar with marks and labels of values that allow to dynamically select the threshold. All the elements of the matrix with value less than or equal to that indicated on the slide bar is will show as a script (-) in the text area.

It should be noted that:

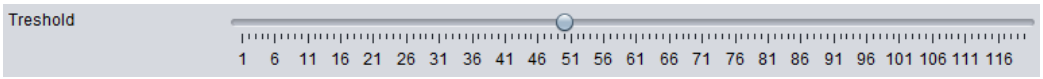
- The elements that must be separated in two different panels configure the result of the text area where the matrix is shown filtered.
- No button will be available to start processing.
- Any change in dimension, maximum or minimum value supposes initialize and show the new matrix.
- Changes in the threshold using the slide bar only will involve filtering the lower values and updating the area of text.
- Documentation to be delivered

2. Implementation

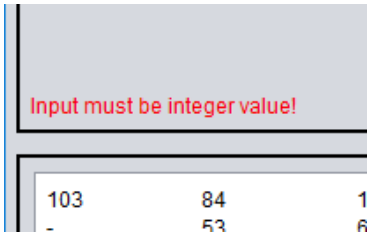
For programming we used Netbeans IDE with Java 8. We have programmed an application which is creating matrix based on inputs of matrix dimension, minimum matrix value and maximum matrix value as seen in 2.1a. Under input textboxes is placed threshold slider shown in figure 2.1b, which dynamically shows "-" for any number in matrix lower or equal than threshold value. For better user friendly environment we even created error label on figure 2.1c, that shows error messages when user inputs incorrect format inside textboxes (e.g. not integer). In the bottom part of application is located textarea shown in 2.2 for the output matrix.



(a) Detail on inputs



(b) Detail on slider



(c) Detail on error label

Figure 2.1: Single components in the application

Whole application was tested many times to eliminate all exceptions in the

program. The main concept was followed to the detail and it should be working perfectly.

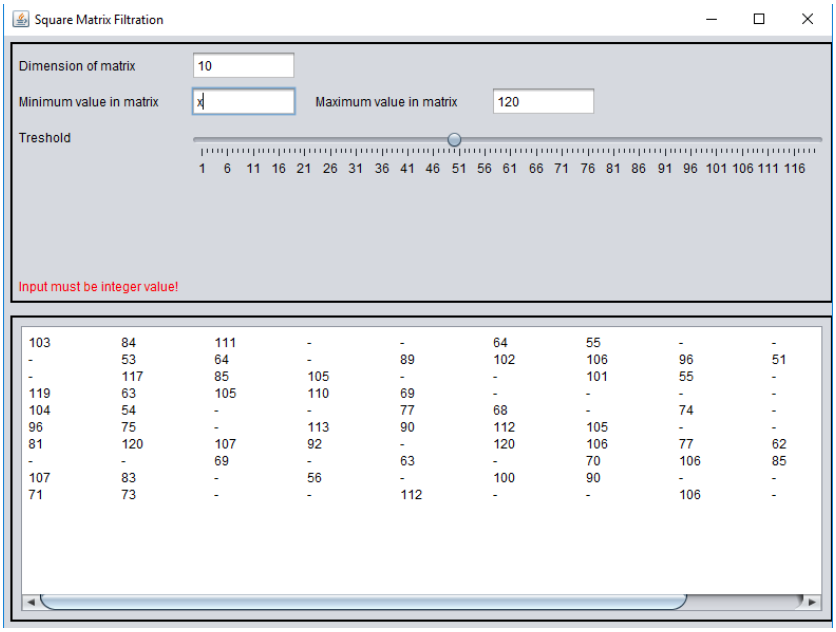


Figure 2.2: Our square matrix filtration application

3. Shneiderman and Plaisant principles

- **Consistency**
  Since we are creating only one application window there is not much space to be consistent at design. Window should be intuitive, upper half is for settings, bottom half is for displaying result.
- **Universal usability**
  For the purpose this application is created the usability of this application is pretty straight. There is almost nothing that user can do except of changing values of inputs. All inputs are executed instantly.
- **Informative feedback**
  If user writes wrong data into input textboxes, the error label automatically popup to show what is wrong with inputs. If input data are good, when changing, the textarea instantly shows result.
- **Design dialogue to yield closure**
  If every input is right, changing inputs displays results instantly in bottom textarea.
- **Simple error handling.**
  Implemented error label is in the bottom of settings to tell you what is wrong.
- **Permit easy reversal of actions**
  This is something we didn't implement in our application. There is no possibility to reverse current action.
- **Support internal locus of control**
  The user should be in full control of how big matrix and the dispersion of numbers he wants to create and what threshold to set for matrix values.

- **Reduce short-term memory load**
  This is hard to say for such a small application, but from a big perspective our application is fast enough for human needs.