Developing and Assessing New List Operators in App Inventor

Soojin Kim
Advisor: Franklyn Turbak
Designer
Blockly Editor
Problem with Current List Operators
Current List Operators

- create empty list
- make a list
- add items to list
- is in list?
- length of list
- is list empty?
- pick a random item
- index in list
- select list item
- insert list item
- replace list item
- remove list item
- append to list
- copy list
- is a list?
- list to csv row
- list to csv table
- list from csv row
- list from csv table
- look up in pairs
Map, Filter and Reduce

>>> (map (lambda x: x + 1) [5, 3, 8, 10, 2])
[6, 4, 9, 11, 3]

>>> (filter (lambda x: x < 6) [5, 3, 8, 10, 2])
[5, 3, 2]

>>> (reduce (lambda x, y: x + y) [5, 3, 8, 10, 2])
28
## Berry’s Lemonade

<table>
<thead>
<tr>
<th>Date</th>
<th># of lemonades sold</th>
<th>Daily Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/1/13</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>6/2/13</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>6/3/13</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>6/4/13</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$2</td>
</tr>
</tbody>
</table>

*Total Profit: ?*
Old Design Using Loops

initialize global originalList to

make a list

13
20
"N/A"
18
"N/A"
10
16

Do It Result: (13 20 N/A 18 N/A 10 16)

---
Old Design Using Loops
Sort: Old Design

http://www.imagnity.com/tutorials/app-inventor/list-sorting-on-app-inventor/
Solution: Addition of Higher-Order Operators
Map, Filter and Reduce
Map Block

(1) map(lambda item: item * 2, [5,3,8,11,2])
New Design Using Map, Filter and Reduce Operators
Three Sort Blocks

1. make new sorted list from
2. make new sorted list from using key called on each item
3. make new sorted list from where item1 precedes item2 if
Basic Sort

Do It Result: (false true 19 23 bunny dog (17 fish) (cat 3))

Do It Result: (23 dog true (cat 3) (17 fish) false bunny 19)
Sort with Key

Do It Result: (Zoe Sam Alex David Megan Brendan)

Do It Result: (David Brendan Zoe Sam Alex Megan)

make new sorted list from
using key called on each length
get item

get global names
input list

key expression
Sort with Comparator
Destructive vs. Nondestructive Mechanism
Simulating Nondestructive Version of a Destructive Filter Operator
Simulating Destructive Version of a Nondestructive Filter Operator
Destructive vs. Nondestructive Mechanism
Design and Results of User Study
Design of User Study

- Short tutorial on each list operator
- Part 1: 8 tasks involving mapping, filtering, and/or reducing
- Part 2: 6 tasks involving sorting
User Study Participants

- 18 Wellesley students who had previous experience working with App Inventor
- 10 users (56%) knew map, filter or reduce previously and the remaining 8 users (44%) did not
User Study Results Part 1

Comparison of performance on mapping, filtering and reducing tasks between students who knew at least one of these three operators previously vs. those who did not.
Feedback

- “I have worked with map, filter, and reduce a lot in different languages, so the concepts were familiar and I was able to interpret pretty quickly what parts the blocks should have.”
- “These blocks were fairly simple to use, but I sometimes became frustrated because I would forget which block was useful in what kind of scenario. Reading the english on the blocks also helped with this though when I would get stuck.”
User Study Results Part 2

Comparison of performance on sorting tasks between students who knew map, filter or reduce previously vs. those who did not.
User Study Results Part 2

The type of sort operator used for tasks with multiple solutions

Proportion of students who completed the task correctly

Type of sorting task
- Element itself, ascending
- One key, ascending
- Two keys, both ascending

- Used basic sort and sort with key
- Used sort with comparator
- Used sort with key
- Basic sort
Feedback

• “I didn't like that there were three different blocks for three different kinds of sorting...I almost think it would be easier if you had to explicitly decide how you want a list to be sorted every time you want to sort a list.”
• “I liked that there were three options, so I could use the one I felt most comfortable with.”
• “There are multiple ways you can perform a single task, especially with these three specific sort blocks. That made it both easier (can use any) and more difficult (many options for how to execute) to use.”
Future Work

- finalizing the labels on the blocks
- additional user studies
  - compare loops vs. higher order operators
  - test destructive vs. nondestructive mechanism
  - study wider demographic base