Good Project

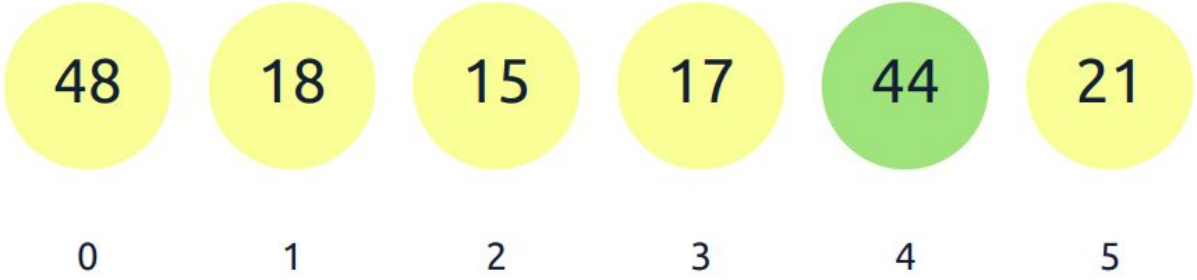
Shiven Sinha (2021111004)
Kommireddy Bhargav Srinivas (2021101065)

Goal

Create a language which can be used to define Virtual Labs experiments in a simple manner, removing the need to implement common functionalities from scratch.

Click on any numbered circle to toggle the selection. You can select upto two numbers.

NOT SORTED



Swap

Primitive Examples

Algorithmic:

- Arrays

- Graphs

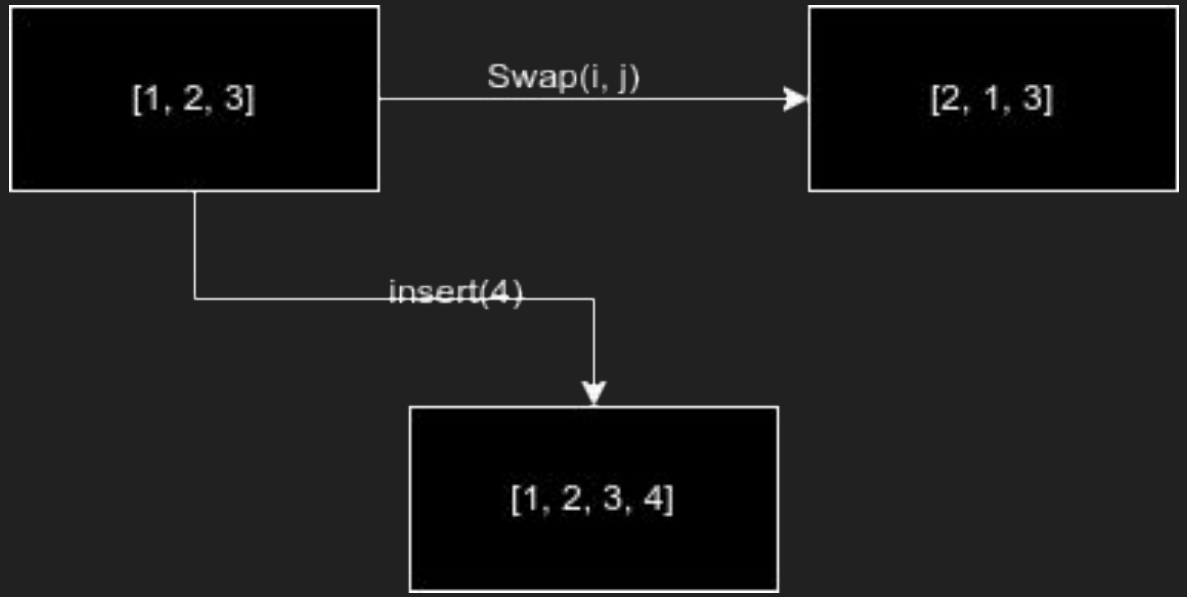
UI:

- Tables

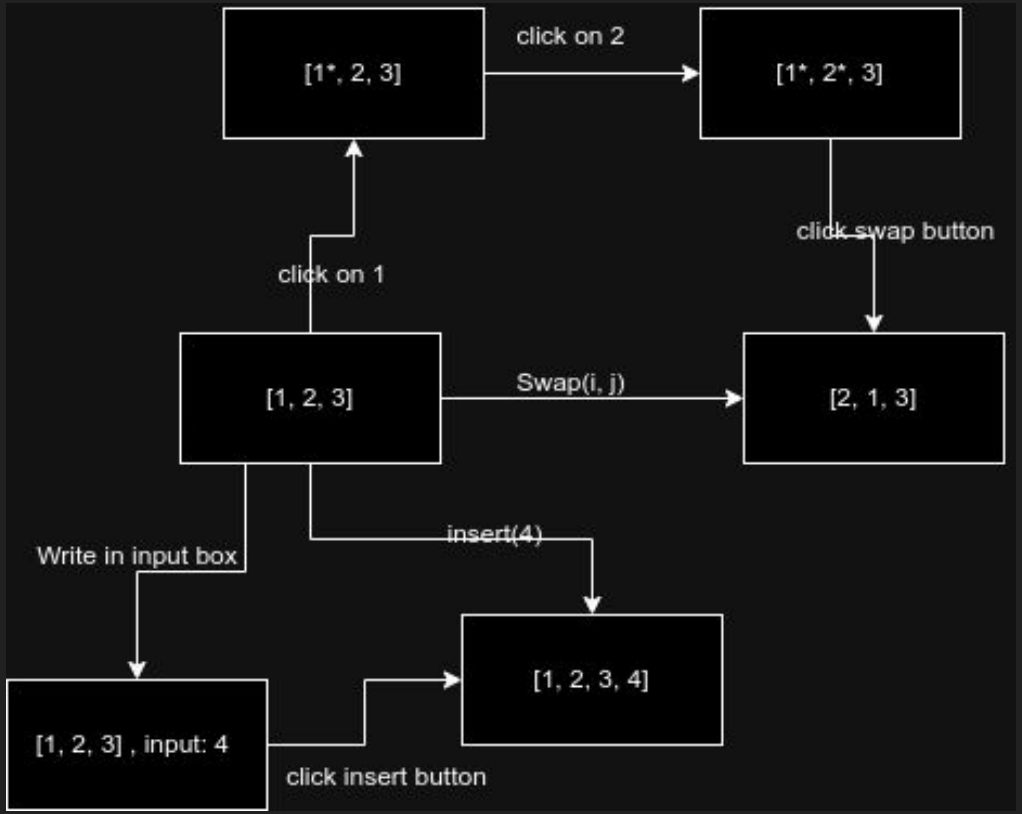
- Buttons

- Text bubbles

Thinking of it as a reduction system



Actions and UI actions



Solution

Step 1: Abstract representation of the experiment

```
(define (swap arr i j)
    (
        ;; code to swap elements
    ))
```

Step 2: Describe the application using good-project

State

```
(def-state
    [(data arr)
    (Array Button x)
    (Button act1)
    (Button act2)])
```

Reductions

```
(def swap (
    if arr.nselected == 2
      ;; do the swap
    else
      throw tantrum
)))
```

Step 3: Use good-project-compiler to get HTML, CSS, JS

```
<div class="header">
  <div class="logo">
    <img src="logo.jpg" alt="logo">
    <h1>Heading tag</h1>
  </div>
</div>
```

Using good-project to specify reduction systems

Defining the State

```
(def-ui (list
  (cons 'arr (array '(2 1 5 7 3)))
  (cons 'counter 0)
  (cons 'input (input "Placeholder"))
  (cons 'msg (text "Not sorted"))
  (cons 'nested-arr '((array '(1 2)) (array '(11 22))))))
```

Defining Reductions - Append

```
(add-action "append"
  (lambda (ui)
    (let ([arr (_ ui 'arr)])
        ((_ arr 'append) ((_ (_ ui 'input) 'get-value))))
      ui)
  (list 'input 'arr))
```

Defining Reductions - Swap

```
(add-action "swap"
    (lambda (ui)
      (let ([arr (_ ui 'arr)]) ;; extract array from the state
        (define swappers ((_ arr 'get-selected))) ;; get arguments (i, j) to swap
        (match swappers
          [(list a b)
           (let ([tmp (_arr-ref arr a)])
             ((_ arr1 'set) a (_arr-ref arr b))
             ((_ arr1 'set) b tmp))]

          ;; check if the array is sorted, update 'msg' in the state accordingly

          [_ ($$ window.alert "Select exactly two elements")])) ;; error message
      ui)
    (list 'txt 'arr))
```

How to handle multiple actions with overlapping UI actions?

We provide a drop down of the higher-level action (eg: swap or insert) that the user wants to do.

For a selected action, we allow a subset of possible actions to happen.

This solves problems like trying to predict what buttons should disabled, and what text boxes should be displayed, so on.


swap
2 1 3 4 5 6 7
Not sorted
swap


swap
1
2
3
4
5
6
7
Sorted
swap


append
2 1 3 4 5 6 7
lmao
append


pop
2 1 3 4 5 6 7
pop

Thanks