# Io: Enhancing Input/Output of Visual Programming Environment

Our plan for UIST2010 Student Innovation Contest

Jun Kato, Akira Ohgawara, Noh Seung-Tak

The University of Tokyo

There have been developed various kinds of visual programming environments such as Scratch, LabVIEW, Microsoft Visual Programming Language and so on. Their users make and edit a diagram to construct a program. Mathias et al. investigated gestures for diagram editing on interactive surfaces [1] which can be implemented on these environments to enhance their usability.

When a brand-new input/output device is invented, we think there is always a chance to rethink and create an interesting new programming environment. For instance, a pressure-sensitive programming environment with visual feedbacks of a built-in 2D physical simulator called "Pressing" was presented by Jun at the Student Innovation Contest last year [2]. Thus, for this year, we would like to propose a visual programming environment named "Io" (pronounced /ˈiː.oʊ/) which we think is very innovative and can only be implemented on the adaptive keyboard. In this proposal paper, we will explain how the user can make and debug a program on Io.

## Io Programming

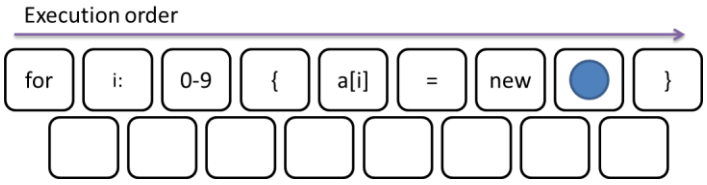


**Figure 1**

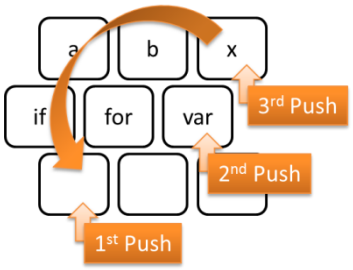


**Figure 2**

By default, source code of Io is displayed on the keyboard as shown in Figure 1. A vertical scroll bar is shown in the touch display above the keyboard to allow the user to increase or decrease the starting number of line of the source code. Each of the keys basically represents one syntax element. When the user 1) pushes a key for short time as shown in Figure 2, possible syntax elements at the key position will be displayed on the keys in the above line of the pushed one; 2) pushes a key of "{" or "}" for short time, the scope will be folded as shown in Figure 3. 3) pushes the key twice in a short time, the syntax element displayed on the key will be removed; 4) pushes the key for long time, a blank element will be inserted at the key position and all elements in the right are shifted to right one by one. The user can switch the editing mode by tapping the touch screen above the keyboard and the source code will be

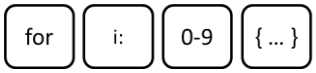


**Figure 3**

displayed in the touch screen with the keyboard of the normal layout. In this “normal” mode, the user can do the old-fashion text programming.

## Io Debugging

**Figure 4**

When the user runs the source code, current stack trace will be displayed on the keyboard in real-time. Figure 4 shows the state when the function *h* is called by *g* which is called by *f*. The user can run the code line by line for debug purpose and check the state of the program visually. When the user 1) pushes a button for short time, the parameters passed to the corresponding function will be shown on the keys in the above line; 2) pushes a button for long time, the definition of the corresponding function will be shown on the keyboard.

## References

1. Frisch, M., Heydekorn, J., and Dachselt, R. 2009. Investigating multi-touch and pen gestures for diagram editing on interactive surfaces. In Proc. of ITS’09 (Banff, Alberta, Canada, November 23 - 25, 2009). pp.149-156.
2. Pressing. http://digitalmuseum.jp/en/software/pressing/