The Design of Naming Features in App Inventor 2

Franklyn Turbak
Wellesley College

David Wolber
University of San Francisco

Paul Medlock-Walton
MIT

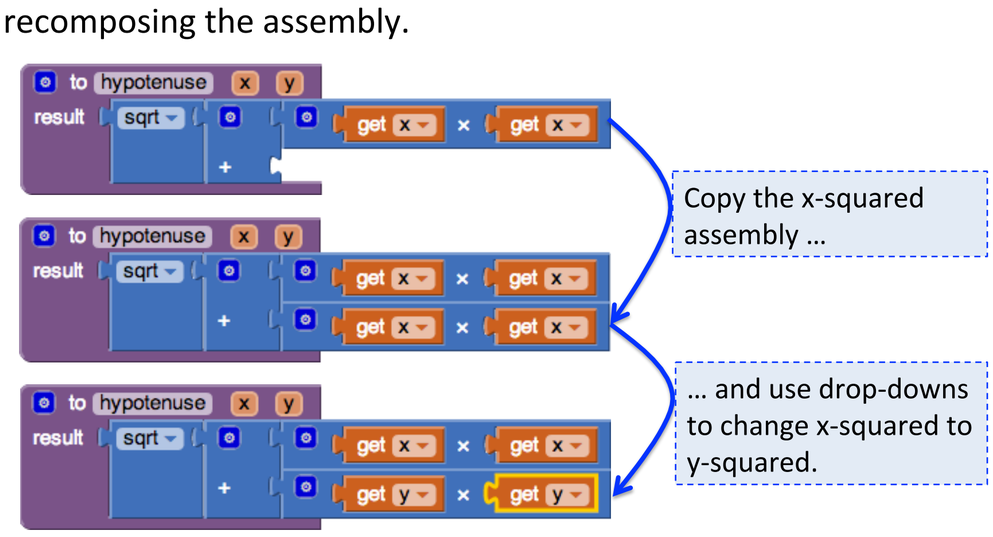
Project Goals

MIT App Inventor [1] is a blocks-based programming environment for creating Android mobile apps. App Inventor 2 (AI2), with its browser-based blocks editor, supercedes App Inventor Classic (AI1) and its Java-based editor and also makes improvements to the blocks programming language.

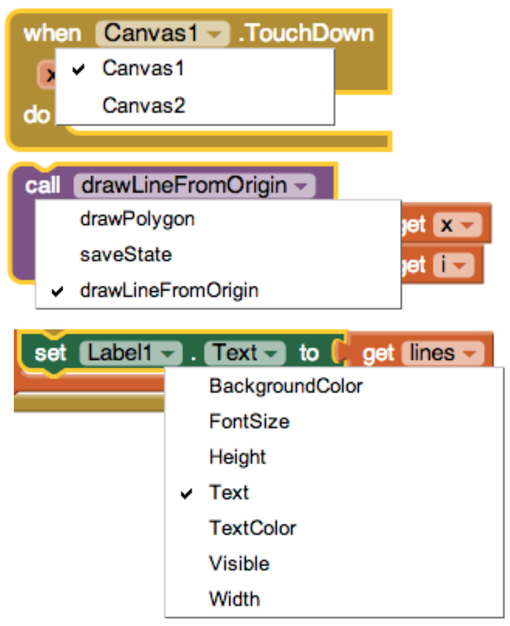
Our project was to design and implement AI2 naming features (for variables and other entities) that fixed problems with such features in AI1 and other blocks languages. Our goals were to respect key naming concepts in computer science and mathematics and (in retrospect) to improve along three dimensions in the **cognitive dimensions of notations** [2]: reducing **error proneness**, reducing **viscosity** (the difficulty of changing a program), and increasing the **consistency** of notations.

Reducing Viscosity

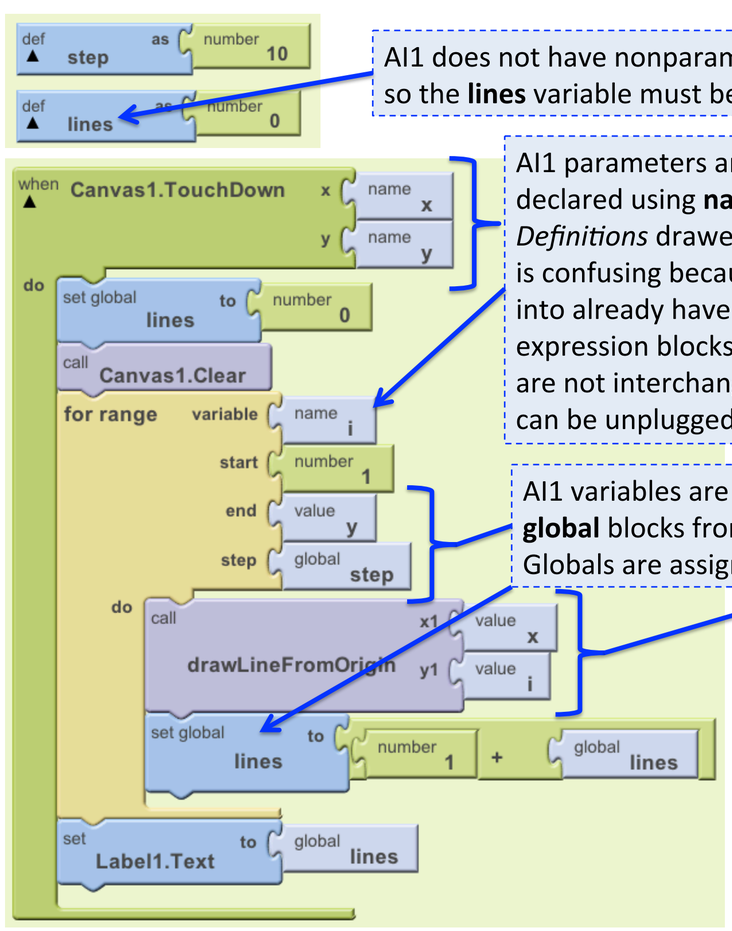
AI2 uses drop-down menus for references to variables. This reduces viscosity compared to AI1 by allowing in-place editing of variable reference blocks via drop-down menus rather than decomposing an assembly, selecting a different reference block from a drawer, and recomposing the assembly.



AI2 also uses drop-down menus for procedures, components, and component properties to reduce viscosity by in-place edits.



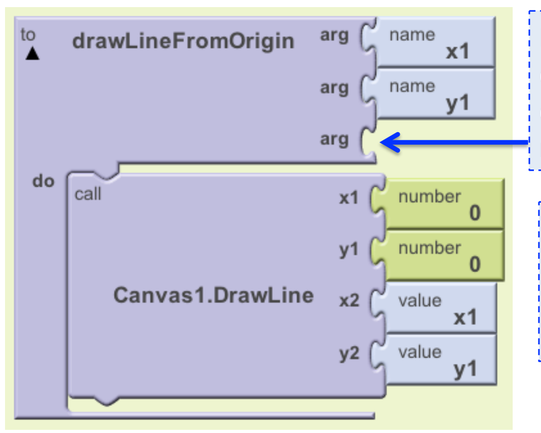
Sample AI1 Program



AI1 does not have nonparameter local variables, so the **lines** variable must be global in AI1

AI1 parameters and loop index variables are declared using **name** blocks from the *BuiltIn Definitions* drawer. Using blocks for this purpose is confusing because: (1) the sockets they plug into already have names; (2) they look like expression blocks (esp. variable **value** blocks) but are not interchangeable with them; and (3) they can be unplugged, which causes a runtime error.

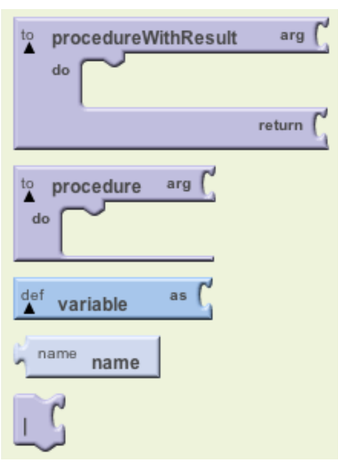
AI1 variables are referenced with **value** and **global** blocks from the *My Definitions* drawer. Globals are assigned via **set global**.



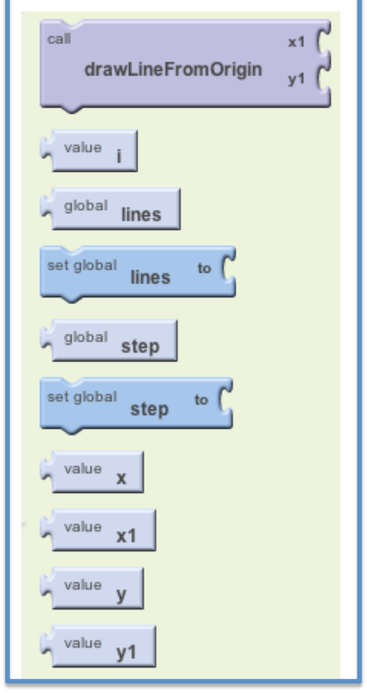
Procedure parameters are added with so-called **extensible sockets**. These can be confused with empty value sockets, which must be filled to avoid an error.

The *My Definitions* drawer list getters and setters for all variables, which must be uniquely named. This violates the **name locality** principle for naming systems.

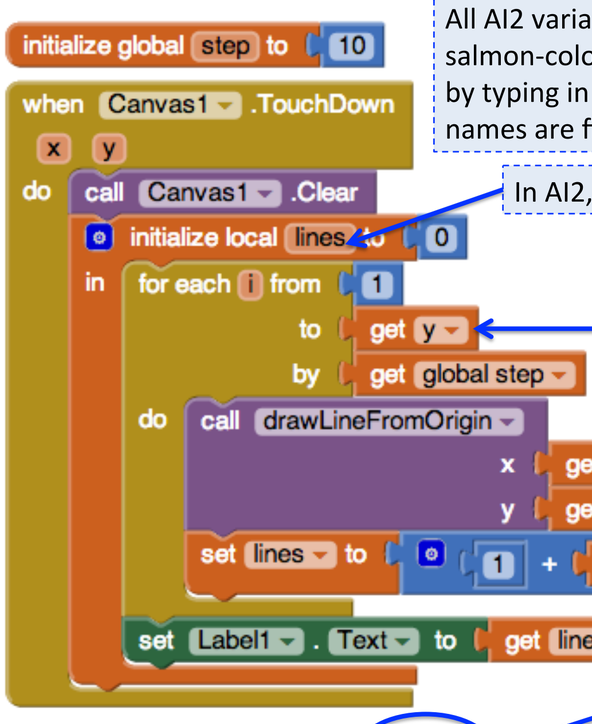
BuiltIn Definitions Drawer



My Definitions Drawer



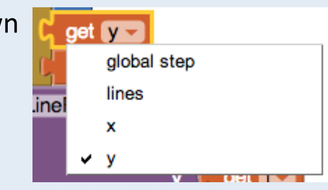
Corresponding AI2 Program



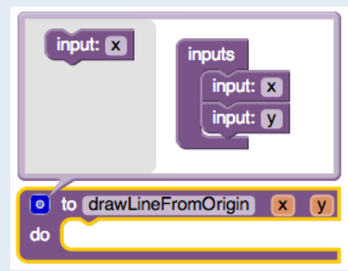
All AI2 variable declarations are consistently represented as salmon-colored rounded boxes. The names can be changed by typing in the boxes (except for event parameters, whose names are fixed), which consistently renames all references.

In AI2, **lines** is a local variable.

A variable drop-down lists only names in scope at this point.



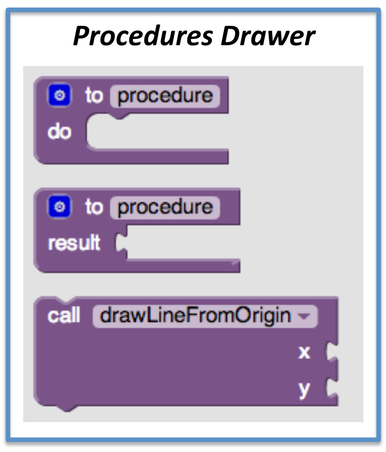
AI2 procedure parameters are declared using Blockly's **mutator**, a mini-block-editor.



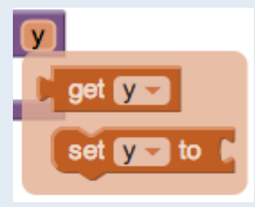
AI2 respects the **name locality** principle, so drawLineFromOrigin can use the same parameter names as Canvas1.TouchDown.

Variables Drawer

Procedures Drawer

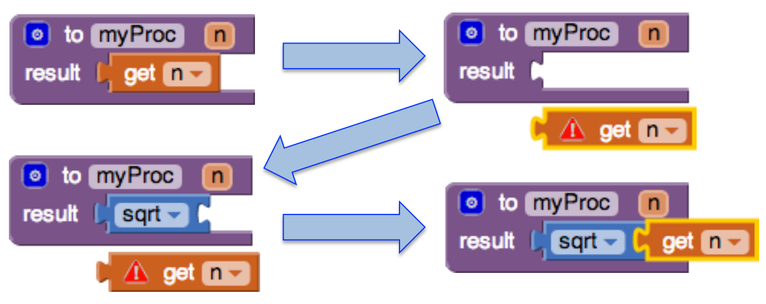


AI2 variable getter and setter blocks can be selected from the *Variables* drawer or from a flyout from hovering over a declaration. For example:



Flagging Unbound Variables

Unlike AI1 & other blocks languages, AI2 flags unbound variables (with a red error triangle). This makes variable errors more obvious and reduces viscosity when editing programs:



Acknowledgments

This work was supported by Wellesley College Faculty Grants, by sabbatical funding from Wellesley College and the University of San Francisco, and by the National Science Foundation under Grant Numbers DUE-1226216 and DUE- 1225745.

References

[1] MIT App Inventor home page: http://appinventor.mit.edu

[2] T. R. G. Green, "Cognitive dimensions of notations," in *People and Computers V,* A. Sutcliffe and L. Macaulay, Eds. Cambridge, UK: Cambridge University Press, 1989, pp. 443–460.