Improving the Usability of App Inventor through Conversion between Blocks and Text

Karishma Chadha, kchadha@wellesley.edu
Franklyn Turbak, Advisor

In blocks programming, users compose programs by combining visual fragments (blocks) shaped like jigsaw-puzzle pieces. The shapes suggest how the blocks fit together, reducing syntactic frustrations experienced by novices when learning textual programming. MIT App Inventor 2 (AI2), a popular online environment for Android app development, democratizes programming through its easy-to-use blocks language. However, while simple blocks programs are easy to read and write, complex ones become overwhelming. Creating and navigating nontrivial blocks programs is tedious, and AI2's current inability to copy blocks between projects inhibits sharing.

To address these issues, I have created a new textual language, TAIL, that is isomorphic to AI2's blocks language and provided a means for converting between them. TAIL syntax is designed to provide users with a systematic way to translate the visual information on the blocks into text. I have extended AI2 with a set of *code blocks* (for expressions, statements, and top-level declarations) in which users can type TAIL code representing AI2 blocks. These code blocks have the same meaning as the larger block assemblies they represent. Programmers can also convert back and forth between these code blocks and the original AI2 blocks. Language isomorphism guarantees that a round-trip conversion (from text to blocks and back, or blocks to text and back) begins and ends with the identical program.

To implement the TAIL language, I wrote a grammar for the ANTLR parser generator to generate a JavaScript lexer and parser for TAIL. I use actions in the grammar to translate the TAIL parse tree into AI2's XML tree representation for blocks.

This project aims to (1) increase AI2's usability by providing an efficient means for reading, constructing, and sharing programs, and (2) ease users' transitions from blocks programming to text programming.