

Displaying data structures and algorithms in a graphical user interface

Brief

Novice programming students usually have a hard time visualizing changes in data structures as an algorithm proceeds.

The main goal of this system is to provide a graphical representation of these changes in two ways: displaying the behavior, of an existing algorithm and moving graphical objects around the screen to simulate an algorithm and the system automatically generates the code.

It's not easy to implement these kinds of systems as it requires access to the runtime environment of an executing program. Also translating the manipulation of graphical objects to code is challenging. A set of manipulation primitives must be defined that will correspond to particular code blocks in the underlying language. The application of these primitives must be recognizable and the translation must be made. Some options for undoing previous operations must be implemented that will reset the screen as well as the underlying run-time representation. Using Genesis as an open source and simple programming language should make it easier to implement. Genesis implements a single data structure: the list data structure. Accessing and displaying these structures will be challenging because Genesis allows the same sticky note to appear in more than one list, lists can be nested, and lists can be heterogeneous.

Design

Layer 1: Genesis Runtime Environment (Basically Sticky Notes, Lists, Values).

Layer 2: A middle layer written in java to handle generating the code according to the objects dragged on the graphical user interface and to generate objects on the graphical user interface when a code written.

Layer 3: A graphical user interface built using **JavaFX** technology which provides high flexibility in building interfaces.

Example

Use Cases:

Case 1:

- The user drags a sticky note from the tool box and names it “a” and gives it a value of “5”
- The user clicks on “Show Code” → Generated code appears in the code area

Case 2:

- The user writes “Let a name 5;” in the code area
- The user clicks “View” → it views a Sticky note with name “a” and value “5” on it