

Visual OO-Debugger

A VS Code extension for visualizing debugger information at runtime

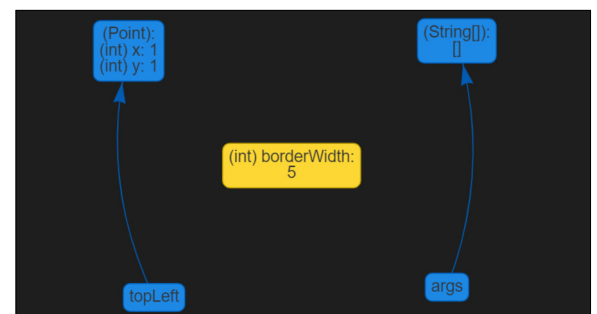
Introduction: Object-oriented programming can be a challenge for unexperienced or new developers. The relations between objects, variables, and the concept of call-by-reference in methods is difficult to comprehend for a lot of people, sometimes even for more experienced developers. Teaching object-oriented programming can be just as challenging as learning it. One of the best ways to teach this topic, is to visualize the relations between objects and variables. Creating such visualizations is time consuming.

Objective: The goal of this project is to create a tool for developers, with any level of experience, as well as teachers, to ease the process of learning and teaching the concepts of object-oriented programming. This tool uses debugger information at runtime to visualize objects and variables in a graph. The nodes of the graph represent variables and objects while the edges represent references between variables and objects, as well as references between two objects.

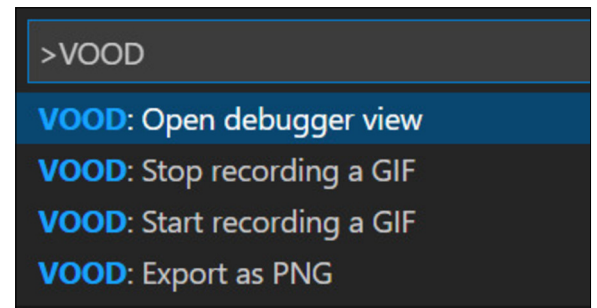
Result: The result was an extension for the widely used and free IDE Visual Studio Code. For the visualization of the graph, we used the open-source visualization library vis.js. In case of an object, a node consists of the name of the class in parentheses and, if present, every instance field with a primitive data type including its value. References to other objects are displayed with edges/arrows to other nodes. In case of a variable referencing an object or null, a node simply contains the name of the variable. A variable with a primitive data type contains the type, name of the variable and the value. Newly added nodes and edges are colored in yellow. Multiple options to export the visualization were implemented,

as a PNG, PlantUML or GraphViz of the current state or as a GIF of multiple steps. Using the two buttons in the upper left-hand corner of the debugger view, it is possible to load the previous/next state of the visualization, all the way back to the first visualization.

Different types of nodes
Own presentation



Implemented commands for the extension
Own presentation



Students



Alexandre Ichiro Lagadec



Pascal Schürmann



Gino Cardillo

Examiner Prof. Mirko Stocker

Subject Area
Software, Software Engineering - Core Systems

Example graph with code
Own presentation

