Visual OO Debugger

A VS Code extension for visualizing debugger information at runtime

Graduate

**Alexandre Ichiro Lagadec**

Pascal Schürmann

Gino Cardillo

Introduction: Object-oriented programming can be a challenge for inexperienced or new developers. And teaching object-oriented programming can be just as challenging. In the autumn term of 2021, we created the VS Code extension Visual OO Debugger, VOOD for short, whose goal was to ease the process of learning and teaching the concepts of object-oriented programming. It achieves this by using debugger information at runtime to visualize objects and variables in a graph.

Objective: The goal of this project is to extend the VOOD with more valuable features, as well as to adapt the code to facilitate further extension. The current library used for visualization, vis.js, is a great starting point, but it has its limitations. Thus, this project's main feature is adding the option to change the visualization style. Currently, only Java is supported by VOOD. While the support of other languages is out of scope, it should be possible to add support for them. Since the Java-specific parts are intertwined with the rest of the debugger, they must be separated.

Result: The result was a new version of VOOD with many new features and improvements. A new visualization was added that uses the JointJS library. JointJS was already evaluated in the term project and was deemed fit as an alternate visualization. It offered more flexibility for customization but at the cost of increased complexity. Another new feature was the option to choose a stack frame of the call stack in a dropdown and visualize it. By clicking on a node, it collapses with its referenced nodes, and they form a cluster. Developers can open those clusters either by clicking on them individually or clicking the left button in the upper right-hand corner to open all clusters

simultaneously. Dragging a node or cluster of nodes over the right button in the upper right-hand corner hides it from the visualization. Clicking this button will show all hidden nodes and clusters again. These are just a fraction of the features and improvements implemented in this project.
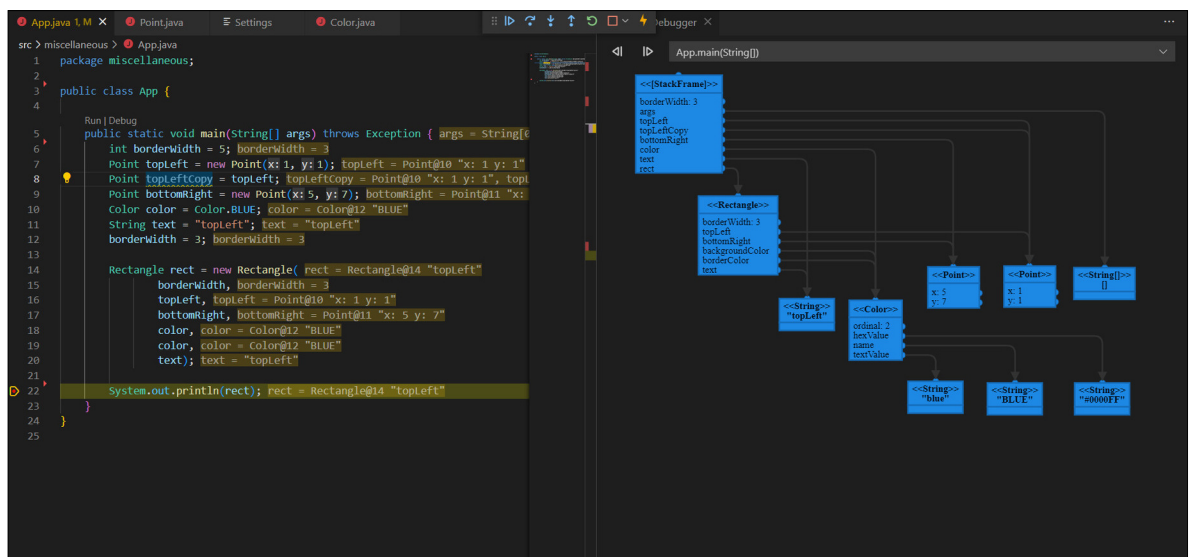
Stack frame selection and new buttons (upper right-hand corner)
Own presentment
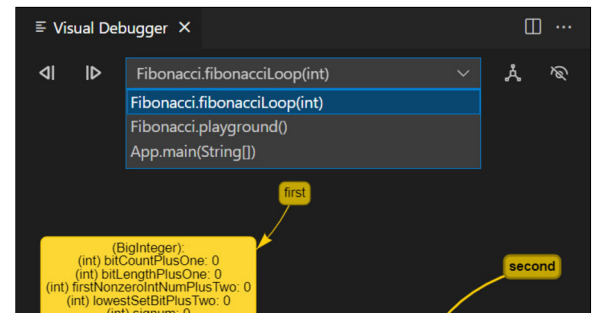
Clustered node
Own presentment
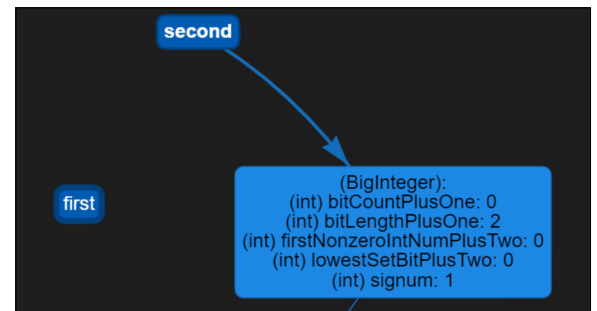
JointJS visualisation
Own presentment

Advisor
Prof. Mirko Stocker

Co-Examiner
Leo Büttiker, yonesu GmbH, Olten, SO

Subject Area
Software Engineering - Core Systems