

BACKGROUND



GitHub Copilot [1] is a AI pair programmer that generate code snippets from code contexts.

Our project is aimed at understanding programmer behavior when they program in collaboration with GitHub Copilot, including IDE behavior (direct) and eye tracking behavior (indirect).

DATA COLLECTION

We have developed a plugin for IntelliJ IDEA. The plugin has 3 main functions (Fig. 1).

(1) Detecting the developer's IDE behaviors, including IDE actions (e.g., reformat code, save all, go to declaration) and code editing by programmers.

(2) Tracking file content changes, compiles, test case, and debugging behaviors.

(3) Collecting and analyzing eye tracking gazes. Our plugin can use an eye tracking device to detect where a person looks in terms of line, column, and token in a source file [2].

All data samples collected are aligned by timestamps.

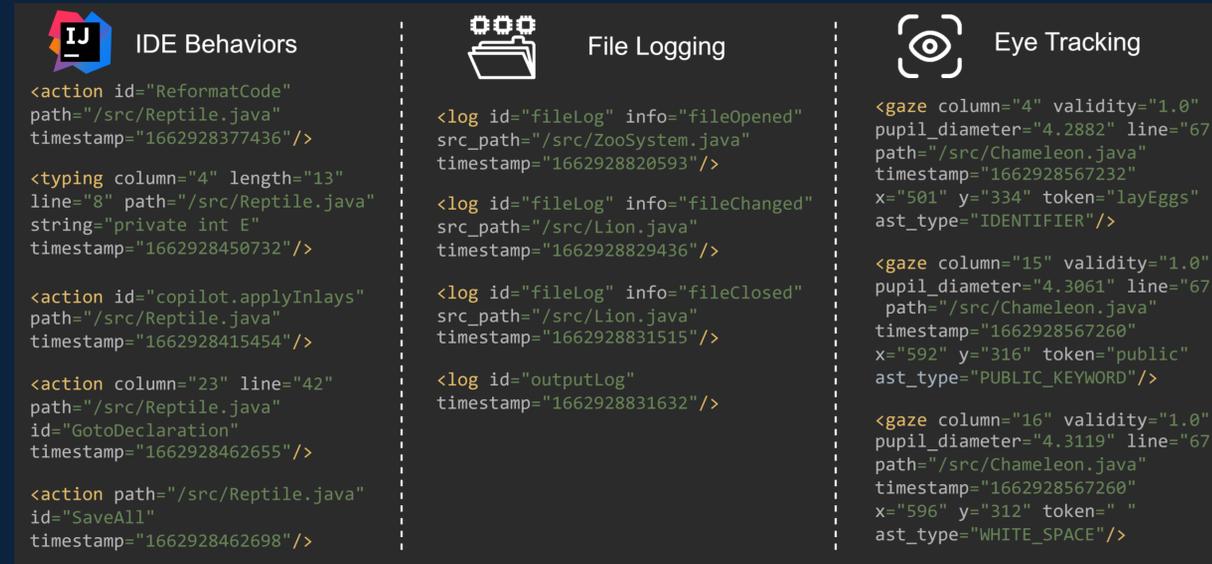


Figure 1. Example of programmer behavior data collected by our plugin.

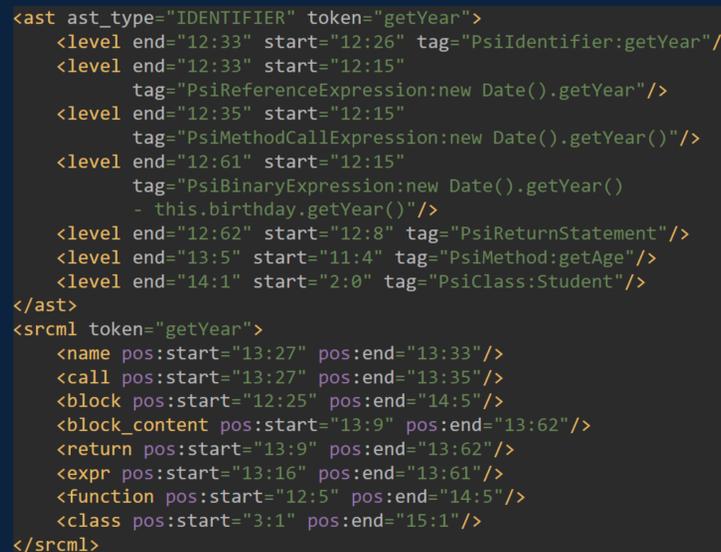


Figure 2. Example of extraction of normal AST and srcML hierarchy of token "getYear".

BEHAVIOR ANALYSIS

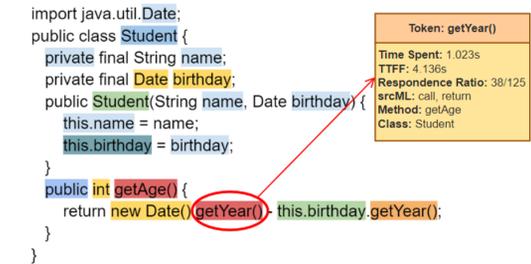


Figure 3. Analyzing gazes in token level. In the heatmap [3], the warmer colors represent higher percentage of fixation and the colder colors for lower percentage of fixation.

In an experiment, programmers will finish various kinds of programming tasks with GitHub Copilot: (1) algorithm design; (2) front-end and back-end development of a calculator; (3) an exercise of object-oriented programming. Eye tracking patterns based on AST/srcML hierarchy and IDE behavior patterns will be analyzed.

Reference

- [1] GitHub Copilot: <https://github.com/features/copilot>.
- [2] D. T. Guarnera, C. A. Bryant, A. Mishra, J. I. Maletic, and B. Sharif, "itrace: Eye tracking infrastructure for development environments," in Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, pp. 1–3, 2018.
- [3] B. Clark and B. Sharif, "itracevis: Visualizing eye movement data within eclipse," in 2017 IEEE Working Conference on Software Visualization (VISSOFT), pp. 22–32, IEEE, 2017.

FUTURE WORK

1. Model programmer behaviors and cognitions.
2. Augment ML models for source code with programmer behaviors.
3. Improve performance in downstream tasks e.g., code generation, code summarization, bug detection, requirement analysis.