CodePanorama: a language agnostic tool

for visual code inspection

Paper doi: <https://dl.acm.org/doi/10.1145/3524610.3527874>

Team members: Ciama Andreea Elena, Abrudan Rebeca Rafaela

CodePanorama uses an alternative approach to summarize code not into numbers, but into images. By generating zoomed-out images of the code-base, the human eye can quickly spot anomalies without the need to rely on numerical metrics and statistics. In contrast to reductionistic metrics, CodePanorama generates zoomed-out images (so-called code panoramas) of the entire selected code-base of a software project, thereby allowing the reviewer to take advantage of their innate image processing skills to instantly get a good first impression of thousands of lines of code.

The idea behind this visualization originated out of a need to evaluate the code quality of large student projects. Our initial search for such a tool did not reveal anything that we could use. After developing the tool, we were made aware that similar visualizations had been attempted before. SeeSoft , developed in the 1990s, is often cited as the first implementation of this so-called “code-map metaphor” . Bacher et al. have published a review paper, comparing 21 different implementations of the code-map metaphor . Their study concludes that such a visualization has great benefits in the software engineering process. Nevertheless, we observe that its use is virtually nonexistent in the standard software engineering process.CodePanorama tool will allow this form of visualization to be used more frequently in modern software engineering projects.

CodePanorama is publicly available as a web application. CodePanorama was designed to be language-agnostic, therefore it is able to visualize any repository, regardless of which programming language is used. CodePanorama can display more lines per image than its predecessors. This is achieved through a more compact arrangement of the visualization.

CodePanorama takes code from a Git repository and produces interactive images by generating a line of pixels for each line of code. This abstraction generates a “zoomed-out” view of the code. This view can also be thought of as the silhouette of the code, as indentation and whitespace are preserved. CodePanorama is built with a server component and a user interface component. The backend server is implemented using Haskell2. The web interface is implemented using Elm3.

After CodePanorama has cloned the supplied repository and gathered some initial information, the user is presented with the filter view. Most software repositories contain many non-source-code files. CodePanorama provides the user with various filter options to tailor the generated images to the user’s needs. Using these filter options, the user can specify exactly which files of the repository the user would like to have included or excluded in the generated images.

The tool was originally intended as an aid for an instructor to grade student projects. Instructors often have multiple student projects to grade. Having a tool to more quickly identify key locations of the code to assess, drastically cuts down on the work required.

CodePanorama eases the transition from metrics (represented as overlays) to the code, by presenting said code in a visual zoomed-out representation. With the use of CodePanorama, we believe that code reviews will become more efficient, as less time is spent scrolling through uninteresting code. As a next step, we plan to investigate to what extent this claim is valid.