Link to Github:

https://github.com/rpesek4/Eye-Tracking-Study-Project

Members:

Robert Spatz, Russell Pesek, Christian Molina

Title:

How do programmers understand regular expressions via eye tracking?

Abstract:

We want to use eye tracking on participants while they examine uses of regular expressions in code, in order to see if there are any notable patterns of behaviors.

This could help understand best practices for using regular expressions in code so that it is more readable.

Motivation:

Understanding how programmers read examples of regular expressions could help in understanding the best way to format code that uses regular expressions.

Related Work 1:

"Exploring regular expression comprehension"(Chapman et al., 2017)

This study investigates common usages of various regular expression features and how they affect comprehension for example using a range([0-9]) was found to be more often understood than a default character class([\d]). The authors conducted an empirical study on 42 pairs of behaviorally equivalent but syntactically different regular expressions using 180 participants.

C. Chapman, P. Wang and K. T. Stolee, "Exploring regular expression comprehension," 2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE), Urbana, IL, USA, 2017, pp. 405-416, doi: 10.1109/ASE.2017.8115653.

Related Work 2:

This study examines how use of regular expressions can result in ddos vulnerability when they take too long to process and input.

It also examines various AI development tools like Chat GPT as they will often generate insecure code and are rapidly gaining popularity.

The discussion talks about how these models will often generate code that works, but is not always completely secure.

Mohammed Latif Siddiq, Jiahao Zhang, and Joanna Cecilia Da Silva Santos. 2024.

Understanding Regular Expression Denial of Service (ReDoS): Insights from LLM-Generated Regexes and Developer Forums.

In Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension (ICPC '24).

Association for Computing Machinery, New York, NY, USA, 190–201. https://doi.org/10.1145/3643916.3644424

Related Work 3:

"How Programmers Read Regular Code: A Controlled Experiment Using Eye Tracking" (Jbara & Feitelson, 2017)

This study explores how programmers read and comprehend regular (repetitive) code using eye-tracking techniques. The authors conducted an experiment with 18 students and 2 faculty members, measuring eye movement and reading effort.

Jbara, A., Feitelson, D.G. How programmers read regular code: a controlled experiment using eye tracking. Empir Software Eng 22, 1440–1477 (2017). https://doi.org/10.1007/s10664-016-9477-x

Related Work 4:

"Regexes are Hard: Decision-Making, Difficulties, and Risks in Programming Regular Expressions" (Michael et al., 2019)

This paper investigates how developers use and perceive regular expressions (regexes) in real-world programming. The study combines a survey of 279 professional developers and interviews with 17 developers to understand regex challenges.

Louis G. Michael, James Donohue, James C. Davis, Dongyoon Lee, and Francisco Servant. 2020. Regexes are hard: decision-making, difficulties, and risks in programming regular expressions. In

Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering (ASE '19). IEEE Press, 415–426. https://doi.org/10.1109/ASE.2019.00047

Research Questions:

What constructs make it hard for novices to understand regular expressions?

Do certain formatting styles improve regex comprehension?

Hypothesis:

Participants will take longer to understand regexes with non-standard character classes compared to simpler alternatives.

Methods:

Showing programmers various code snippets that contain regular expressions and asking them to find an error or explain what it does.

Doing a post test survey to get participant's opinions on the different code snippets and how they were reading them.

Use programing students due accessability.

Data to be collected:

Task completion time

Fixations

Eye path

Survey data

Analysis Envisioned:

We want to examine the data for any potential patterns, like developer confusion. Hopefully allowing us to see what the best way to format regular expression in code is.

Possible threats to validity:

Survey response bias, participants may provide answers they think the researcher wants rather than their true experience or may just simply not report their experience accurately.

Selection bias, if participants are all from the same institution the results that come from them may not apply to the wider world.

Task design issues, poorly made tasks or instructions may cause the results to be inaccurate.

Prior familiarity with regular expressions may impact results.

Device/calibration errors in eye-tracking data.

Impacts Envisioned:

The main impact would be getting a eyetracking dataset that was created by regular expression tests. We also may be able to find a pattern in the data.

Time Log			
Russell Pesek			
Date	Start Time	End Time	Work Done
2/12/2025	11:00	2:45	Found research papers related to project, read and analyzed them, added them to project proposal. Updated proposal with minor corrections.
2/13/2025	1:30	1:45	Worked on Project proposal

Time Log			
Christian Molina			
Date	Start Time	End Time	Work Done
2/12/2025	16:00	17:00	Looked through and found research paper related to project,added related work to the project proposal
2/13/2025	13:15	13:45	Added the threats to validity and research question sections to the project proposal

Time Log			
Bobby Spatz			
Date	Start Time	End Time	Work Done
2/11/2025	15:00	15:45	Started project proposal and filled out a bunch of things.
2/13/2025	10:00	11:00	Did research on studies and filled out more project proposal things.