An Analysis of Programmers' Visual Attention Patterns During Problem-Reading in A Competitive Programming Context

Jenilyn L. Agapito, Jonathan DL. Casano, Joshua C. Martinez, and Ma. Mercedes T. Rodrigo

Department of Information Systems and Computer Science, Ateneo de Manila University

Abstract

This study is an attempt to explore the problem reading patterns of intermediate and expert programmers and solvers and non-solvers through their eye gazes in a competitive programming setting. An eye-tracker was used to capture visual attention data while participants were reading a provided problem specifications. Subsequently, they were asked to code their programming solutions. Results showed that solvers have higher dwell time on the Problem Description while non-solvers on the Input Description. Fixation is highest on the Problem Description for all observed groups. However, no significant observations on intermediate and expert programmers were deduced. Generally, solvers pay more visual attention to the stimulus than the non-solvers.

Objectives

The study attempts to answer the following questions:

1. What are the fixation points of intermediate and expert programmers when reading competitive programming problems and how are they similar or different?
2. What are the fixation points of solvers and non-solvers when reading competitive programming problems prior to solving and how are they similar or different?

Methodology

Participants Profile. The participants were undergraduate students of the Ateneo de Manila University who, at least, have knowledge in solving programming problems and writing computer programs. They were classified into two groups: (a) *intermediate programmers* (non-programming varsity members) and *expert programmers* (programming varsity members).

Stimulus and Regions of Interest. The experiment used a 1366 x 768 screen capture of a sample competitive programming problem taken from codeforces.com as stimulus. A competitive programming problem typically contains 6 major parts and these were mapped to consist the 6 regions of interest (see figure 1):

1. **Region 1: Problem Description.** Describes clearly and concisely the problem.
2. **Region 2: Input Description.** Statements on how input data is read.
3. **Region 3: Constraints.** Statements, usually in a form of mathematical equations, that state the bounds of variables. It tells what and what not to do.
4. **Region 4: Output Description.** Details the output format of the solution.
5. **Region 5: Sample Input.** A sample input data as defined in the input description.
6. **Region 6: Sample Output.** The expected result.

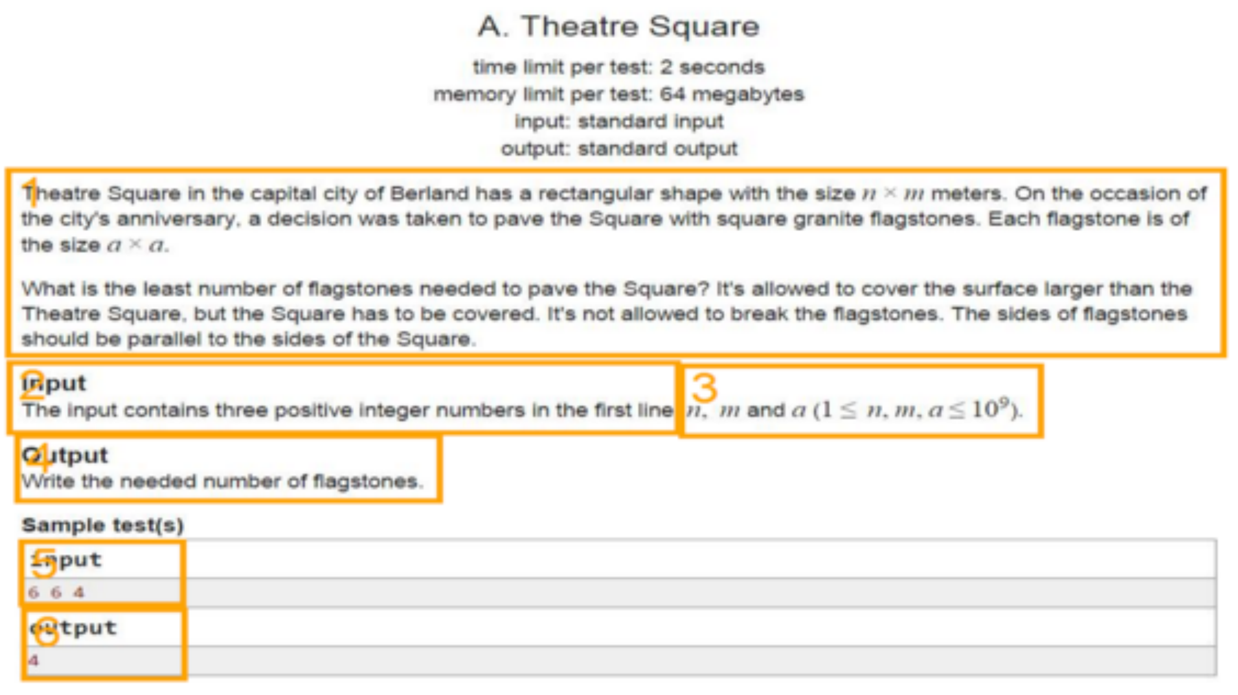

Figure 1. Stimulus and Regions of Interest

Data Gathering. The participants were asked to sit on a chair and place their head on the metal chin-rest of the eyetracker where a laptop is placed in front. Calibration was done next to make sure the eye-tracking device can accurately tell where the participant was looking and record the correct data. The participant was then shown the stimulus and given 2 minutes to design a solution to the problem while using the device. Thereafter, 10 minutes was given to the participant to code the solution on a different machine.

Results

Seventy (70) percent of the participants were able to correctly provide a programming solution to the problem. 71% of these solvers are experts, the other 29% are intermediate.

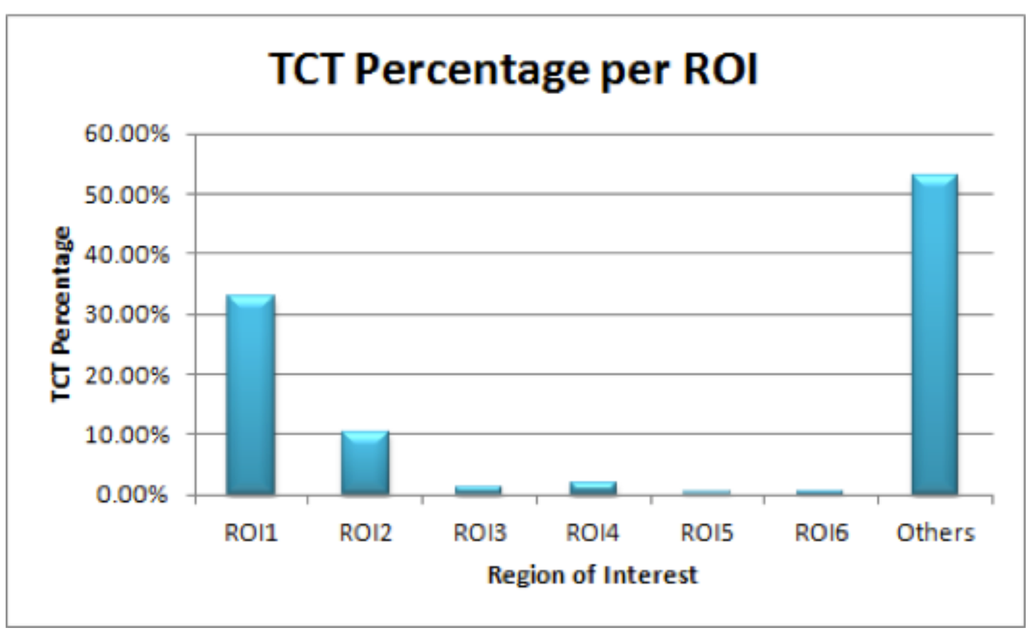

Figure 2. Total Contact Time (TCT) per Region of Interest

Among the 6 defined regions of interest, Problem Description (ROI1), Input Description (ROI2), and Output Description (ROI4) are the top 3 attention-receivers (figure 2) which are considered as the three most necessary information in formulating a solution to a problem. Thus, making these regions heavily susceptible to the participants' attention.

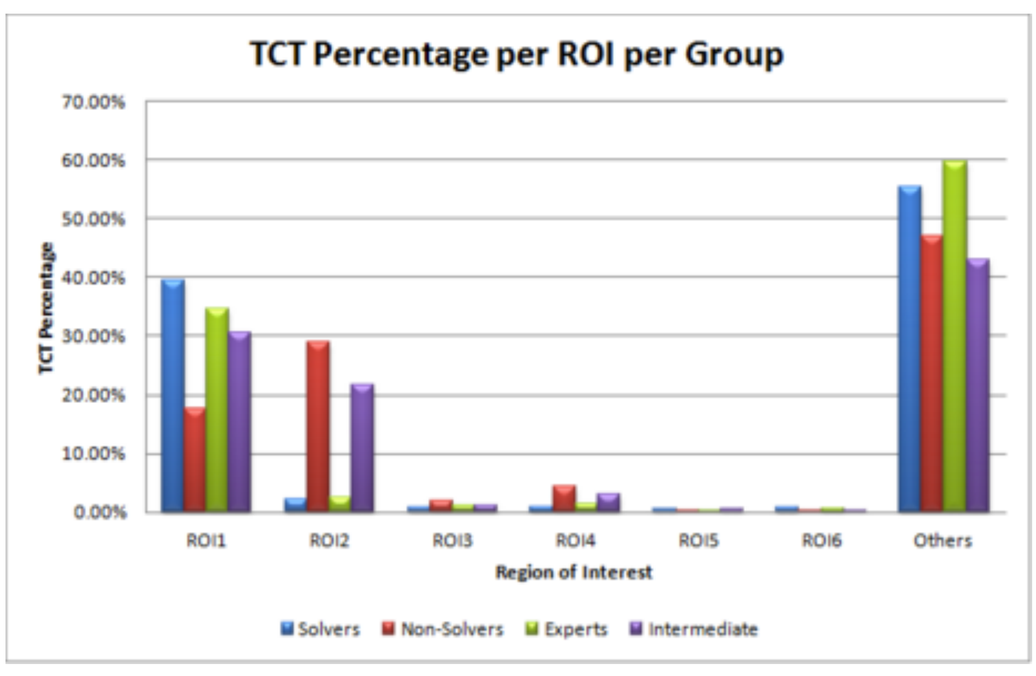

Figure 3. TCT of Groups per Region of Interest

On a per group analysis(figure 3), solvers tend to spend more contact time on the Problem Description (ROI1) while non-solvers on the Input Description (ROI2).

Group	Visual Pattern
Solvers	ROI1 → ROI4 → ROI5 → ROI3 → ROI2 → ROI6
Non-Solvers	ROI1 → ROI3 → ROI5 → ROI4 → ROI2 → ROI6
Experts	ROI1 → ROI4 → ROI5 → ROI3 → ROI2 → ROI6
Intermediate	ROI1 → ROI6 → ROI5 → ROI3 → ROI4 → ROI2

Table 1. Visual Patterns of Groups

Solvers and experts tend to have similar visual patterns (table 1). This may be attributed to the fact that 71% solvers are actually members of the programming varsity. As a general observation, solvers spend more visual attention on stimulus than non-solvers.

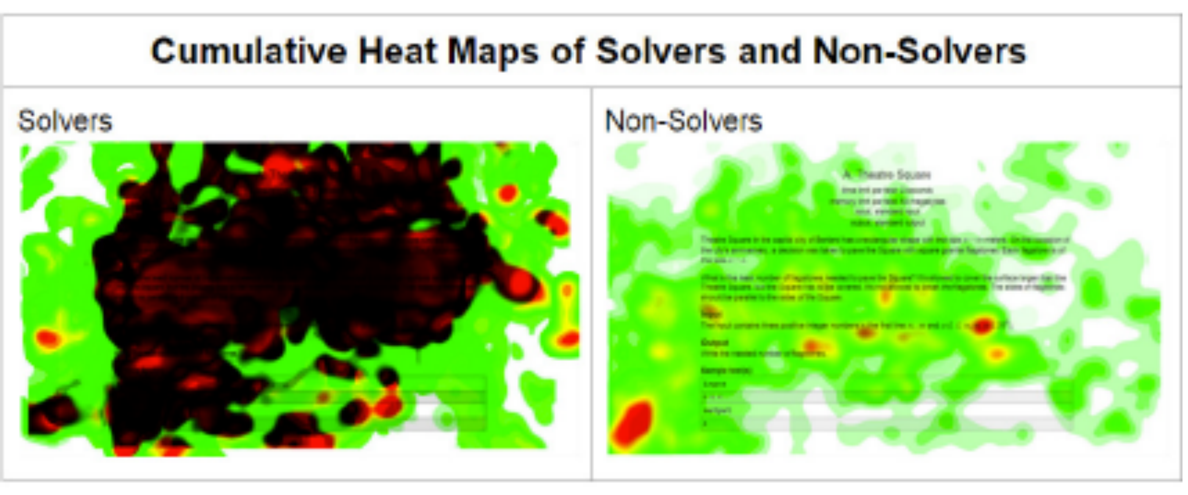

Figure 4. Heat Maps of Solvers and Non-Solvers

Figure 4 shows the heat maps of the two groups. The heavily burnt areas of the solvers' heat map indicate more dwell time spent looking at the problem presented. This elucidates a behavior that may be common to those who are able to solve the problem. They are inclined to repeatedly go through the specifications as they formulate a programming solution.

Conclusion

Visual patterns of solvers and non-solvers have minimal difference. However, notable variations have been found on which regions they give more attention to. Solvers have higher dwell time on the Problem Description (ROI1) while non-solvers on the Input Description (ROI2). Fixation is highest on the Problem Description (ROI1) for all observed groups. It is also the region that first attracts the attention of the participants. Taking into account the `Others' region yields it to have the highest contact time which was regarded as instances during which the participants were formulating their solutions. Generally, solvers give more visual attention to the stimulus than non-solvers. No significant observations on intermediate and expert programmers were deduced. Although the results are telling of the likely behavior of a programmer when reading a programming problem specifications, the study opens up a lot of room for further verification of said observations as well as more research undertakings in the field.

References

Ribeiro, P., & Guerreiro, P. (2008). Early introduction of competitive programming. Olympiads in Informatics, 2, 149-162.

Agapito, J., Casano, J., & Martinez, J. Xiphias: A Competitive Classroom Control System to Facilitate the Gamification of Academic Evaluation of Novice C++ Programmers. In Proceedings of the International Symposium on Computing for Education (ISCE2014), Boracay, Philippines, (23-24 October 2014), pp. 9-15.

Casano, J., An Analysis of Programming Languages and Competitive Programming Problems as Bound by the Limits of Non-Quantum Computation (2015), In Proceedings of Project Einstein 2015

Stein, R., & Brennan, S. E. (2004, October). Another person's eye gaze as a cue in solving programming problems. In Proceedings of the 6th international conference on Multimodal interfaces (pp. 9-15). ACM.

Bednarik, R., & Tukiainen, M. (2006, March). An eye-tracking methodology for characterizing program comprehension processes. In Proceedings of the 2006 symposium on Eye tracking research & applications (pp. 125-132). ACM.

Rodrigo, M. M. T., & Baker, R. S. (2009, August). Coarse-grained detection of student frustration in an introductory programming course. In Proceedings of the fifth international workshop on Computing education research workshop (pp. 75-80). ACM(2012).