

# Applying Asynchronous Gaze Sharing and Social Navigation for Problem Solving Principles in an Programming Environment

Fabian Deitelhoff

Computer Science Department  
Fachhochschule Dortmund  
Fabian.Deitelhoff@fh-dortmund.de

## Abstract

Simplex is a simplified language with a reduced set of keywords and abstractions. It can currently be used to write programs for the LEGO® Mindstorms® EV3. The idea is to enhance this language and the programming environment as part of a PhD project to use gaze sharing data and social navigation for problem solving principles in programming education.

## 1. Language & Environment

Simplex is the result of a R&D-Project and a Master thesis. The goal is to develop and partially implement a text-based language and development environment. Simplex programs can run on the current version of the LEGO® Mindstorms® EV3 as some sort of proxy object to provide interaction with learners and to visualize the program execution. Figure 1 shows a screenshot of the programming environment. On the left a learner can write a Simplex program. The center shows an assembler like intermediate language and on the right the generated artifact in the RBF (Robot Byte Code File) format.

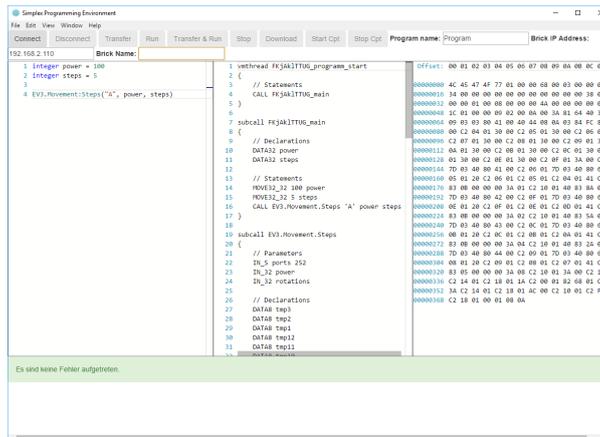


Figure 1 – Prototype of the programming environment for Simplex.

## 2. Gaze Sharing Ideas

One part of the PhD project is to use eye tracking data to capture learners gaze data to visualize and analyze how they interact with Simplex and the environment. For example, this data can be used to test the integration of learning material in a split or in an integrated design within the environment (Jarodzka, Holmqvist, & Gruber, 2017, P. 4f).

Other possibilities are to check how participants will interact with a dynamic help systems like animated images and videos, how they'll interact with error messages within the programming editor and how gaze paths are affected by an integrated simulation as a virtual proxy object.

Furthermore, the gaze data will be collected, analyzed and aggregated to help other learners. Therefore the gaze data will be integrated in the programming environment to provide following learners social navigation of the gaze data of previous participants. The hypothesis is that learners will benefit from information previous participants have looked at within the programming environment while solving tasks. For example, switches between the Simplex editor and the simulation or specific parts of the

integrated programming documentation. This overall scenario is an asynchronous one. There's the opportunity to test synchronous scenarios, too in which for example two learners try to solve a more extensive task in an cooperative session.

### 3. Programming Language Ideas

In addition to this gaze sharing ideas there are more programming language design related ones. The goal is to evaluate at least some of them within the PhD project.

One idea is to integrate a type-inference based type system in addition to the static type system currently implemented in Simplex. Combined with eye tracking it is worth a try to evaluate how missing types will affect the reading and comprehension of Simplex code. This idea can be extended to a "Unit of Measure" system known from many functional languages like F#. In such systems one can provide units like *cm* and *km* as well as rules for this units to further enhance the type system. Thereby the compiler can provide more precise type analysis and error messages to learners.

Additionally, it is possible to test ideas in the field of programming assistance like auto completion, icons for program structures like loops and conditions, and placeholder like input mechanisms for literals (Amorim, Luís Eduardo de Souza, Erdweg, Wachsmuth, & Visser, 2016) and sensors to mention just two examples. All supported by eye tracking data and analyses.

### 4. Analytic Ideas

The main goal on the analytic side is to provide useful insights and appropriate measures for the expertise domain of programming. Eye tracking data of a programming environment can be one part of a methodological triangulation (see (Denzin, 2012)). Eye tracking can be useful for the perceptual aspects but other data sources are needed to complete the picture. For example, this data sources can be performance and verbal data.

It is one goal of the PhD project to come up with results (analytic ways, data sources, ...) in terms of this methodological triangulation for programming in general and programming education.

### 5. References

- Amorim, Luís Eduardo de Souza, Erdweg, S., Wachsmuth, G., & Visser, E. (2016). *Principled syntactic code completion using placeholders*. ACM. Retrieved from [http://dl.acm.org/ft\\_gateway.cfm?id=2997374&type=pdf](http://dl.acm.org/ft_gateway.cfm?id=2997374&type=pdf) doi: 10.1145/2997364.2997374
- Denzin, N. K. (2012). Triangulation 2.0. *Journal of Mixed Methods Research*, 6(2), 80–88. doi: 10.1177/1558689812437186
- Jarodzka, H., Holmqvist, K., & Gruber, H. (2017). Eye tracking in Educational Science: Theoretical frameworks and research agendas. *Journal of Eye Movement Research*, 10(1). Retrieved from <https://bop.unibe.ch/index.php/JEMR/article/view/2959>