GymInf | Informatikausbildung für Gymnasiallehrkräfte | Formation en informatique destinée aux enseignant-e-s au gymnase

Revealing Programming Language Abstractions

An Excerpt of nand-to-tetris – in Reverse – Using Smalltalk

GymInf Individual Project

Simon Bünzli
from
Bern, Switzerland

Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

01. August 2025

Prof. Dr. Timo Kehrer, Prof. Dr. Oscar Nierstrasz

Software Engineering Group
Institut für Informatik und angewandte Mathematik
University of Bern, Switzerland

UNIVERSITÄT BERN

UNIVERSITÉ DE NEUCHÂTEL

UNIVERSITÉ DE FRIBOURG
UNIVERSITÄT FREIBURG

Abstract

Not an *abstract* yet, but the original project description:

Ziel des Projekts ist, ein empirisch abgestütztes Instrument für den Programmier-Unterricht am Gymnasium zu entwickeln, in welchem Schüler:innen verschiedene Abstraktionsebenen interaktiv erleben können.

Auf der Basis von Processing mit Python Syntax (https://py.processing.org/) soll der einerseits der visuelle Ablauf eines Programms, aber auch die Parsing-Schritte und die Übersetzung in Byte-Code Seite-an-Seite sicht- und untersuchbar gemacht werden, damit Schüler:innen die Auswirkungen ihres Programmcodes auf die Maschine live erleben können.

Die Entwicklung des Produkts wird theoretisch begleitet und das Produkt selbst empirisch geprüft werden.

Als Basis der Umsetzung dient Glamorous Toolkit, eine Entwicklungsumgebung basierend auf Smalltalk/Pharo, welche u.a. von Oscar Nierstrasz für Master- und Doktoratsstudiengänge weiterentwickelt worden ist.

Contents

1 Introduction

Programming with Processing (by [6]) vs. ”Little Man Computer” or ”Human Resource Machine”

2

Leaky Abstractions when Teaching Programming

On the lack of connecting high-level languages with lower-level concepts

2.1 Didactic Approaches

See e.g. [7], [3], [1] or [2] only focusing on one aspect

2.2 Limitations of IDEs

Such as VS Code or Thonny not connecting the few views available

2.3 Leaky Abstractions

Where knowledge of lower abstraction levels might help

3
Dealing with Abstractions

3.1 Top Down

Working downwards from gaming, as in [9]

3.2 Bottom Up

Running Tetris on NANDs as described in [8], [5]

3.3 Moldable Development

Referring to [4].

4
Processing Abstractions

4.1 Development of ”Processing Abstractions”

Excerpts from gt-exploration Lepiter pages

4.2 Abstraction Levels

For each a short problem description and a presentation of the chosen approach:

4.2.1 Source Code

4.2.2 Abstract Syntax Tree

4.2.3 Transpilation/IR

4.2.4 Machine Code

4.2.5 Output

5 PA in Practice

How students reacted to using it

5.1 First Round

5.1.1 Setting

5.1.2 Observations

5.1.3 Student Feedback

5.1.4 Learnings

5.2 Second Round

5.2.1 Setting

5.2.2 Observations

5.2.3 Student Feedback

5.2.4 Learnings

6 Conclusion

6.1 Future Work

A

Installing and Using Processing Abstractions

B

Data from Questionnaires

Bibliography

[1] Stefan Hartinger. *Programmieren in Python*. Universität Regensburg, 2020.

[2] Irene Lee, Shuchi Grover, Fred Martin, Sarita Pillai, and Joyce Malyn-Smith. Computational thinking from a disciplinary perspective: Integrating computational thinking in k-12 science, technology, engineering, and mathematics education. *Journal of science education and technology*, 29(1):1–8, 2020.

[3] Eckart Modrow and Kerstin Strecker. *Didaktik der Informatik*. De Gruyter Studium. De Gruyter Oldenbourg, München, 2016.

[4] Oscar Nierstrasz and Tudor Gîrba. Moldable development patterns. 2024.

[5] Noam Nisan and Shimon Schocken. *The elements of computing systems : building a modern computer from first principles*. The MIT Press, Cambridge, Massachusetts, second edition edition, 2021 - 2021.

[6] Casey Reas and Ben Fry. *Processing: A Programming Handbook for Visual Designers and Artists*. The MIT Press, 2 edition, 2014.

[7] Sigrid Schubert and Andreas Schwill. *Didaktik der Informatik*. Spektrum Akademischer Verlag, Heidelberg, 2. auflage edition, 2011.

[8] Ünal Çakıroğlu and Mücahit Öztürk. Flipped classroom with problem based activities: Exploring self-regulated learning in a programming language course. *Journal of Educational Technology & Society*, 20(1):337–349, 2017.

[9] David Weintrop and Uri Wilensky. Playing by programming: Making gameplay a programming activity. *Educational Technology*, 56(3):36–41, 2016.