Titel der Arbeit

Optionaler Untertitel der Arbeit

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Pretitle Forename Surname Posttitle
Matrikelnummer 0123456

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Pretitle Forename Surname Posttitle
Mitwirkung: Pretitle Forename Surname Posttitle
Pretitle Forename Surname Posttitle
Pretitle Forename Surname Posttitle

Wien, 1. Jänner 2001

Forename Surname   Forename Surname

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Title of the Thesis

Optional Subtitle of the Thesis

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Pretitle Forename Surname Posttitle
Registration Number 0123456

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Pretitle Forename Surname Posttitle
Assistance: Pretitle Forename Surname Posttitle
Pretitle Forename Surname Posttitle
Pretitle Forename Surname Posttitle

Vienna, 1st January, 2001

Forename Surname   Forename Surname

Erklärung zur Verfassung der Arbeit

Pretitle Forename Surname Posttitle
Address

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. Jänner 2001

Forename Surname

Danksagung

Ihr Text hier.

Acknowledgements

Enter your text here.

Kurzfassung

Ihr Text hier.

Abstract

Enter your text here.

Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

1.1 Motivation

- software -> bug -> statistics
- code gets written one time and read 4 times
- program comprehension
- debugging -> different kind of paradigms and languages

1.2 problem statement (which problem should be solved?)

- much work and tools on oo-languages
- not so much on stack oriented languages
- applicability of oo-methods for stack oriented languages at the example of forth
- applicability of oo-visualization methods

1.3 aim of the work

- identify important information
- visualization of information
- demo approach

1.4 methodological approach

- qualitative approach
- outcome is a subjectiv view of the available methods, and proposed enhancements which have been implemented and suggestions of further enhancements

1.5 structure of the work

- summary on the available methods for program comprehension in gforth
- summary and applicability of available methods for other paradigms and languages
- enhancement of existing methods and proposal for further enhancements
- poc

CHAPTER 2

State of the art / analysis of existing approaches

2.1 literature studies

- about program comprehension
  - about debugging
- about debugging in other paradigms
  - (?)about some tools
- about debugging in stack oriented languages
  - (?)about some tools
- (?)about visualization maybe some examples and tools
- (?)about realtime/interactive vs post mortem

2.2 analysis

- existing methods abstract(abstract like print debugging and stepping and so on)
- applicability for so-languages

2.3 comparison and summary of existing approaches

- existing methods(actual methods)
  - dump
  - . / type
  - dbg
  - see/ code-see
  - ∼∼

CHAPTER 3

Methodology

3.1 used concepts

- prototyping
- reading codes
- print-debugging
- step-debugging

3.2 methods and/or models

- prototyping

3.3 languages

- postscript
- forth
- shell script
- c
- m2

3.4 design methods

?

3.5 data models

?

3.6 analysis methods

- reading code
- tail and error

3.7 formalisms

?

CHAPTER 4

Suggested solution/implementation

- kind of an ide
  - interactive program manipulation: state of the system before a word, after a word and by clicking on the word jumping to its definition and there also providing those features
  - stepping debugger mode: simply stepping through the whole code word by word
  - other data structures and variables should be displayed
  - display of the 'vocabulary'
- proof of concept by enhancement of stepping debugger on forth code level(cause it has turned out to be the fastest and simples approach) by showing additional data: the other stacks

CHAPTER 5

Critical reflection

5.1 comparison with related work

light table ide(js) eclipse ide(java)

5.2 discussion of open issues

- nature of gforth
  - interpretation/compilation mix
  - implementation within the executing system
  - lack of static(and dynamic?) information
- quantitative data on the effects

CHAPTER 6

Summary and future work

summary of what has been done and the subjective conclusion

- see suggested solution
- using a standard data type to store traces
- display of variable content
- display of allocated memory areas
- display of color diff with tooltip of previous values for stacks and memory areas
- better visualization of loops and control structures
- display of the full program as a graph
- customizable inspection depth
- static code analysis
  - stack depth per word
  - type system for forth
  - ...

Bibliography

[Tur36] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *J. of Math*, 58:345–363, 1936.