

Developing a Visual Studio Code Language Support Extension for the Snail Programming Language

Charles Reinhardt

June 5, 2023

Abstract

A high quality programming environment (often an integrated development environment, or IDE) can be vital to enhancing developer productivity. Visual Studio Code (VS Code) is a popular, open-source text editor maintained by Microsoft. VS Code delivers language-specific features through freely downloadable, community-built extensions on an online marketplace. Many of these extensions allow developers to take advantage of editing features such as syntax highlighting, code-autocompletion, or debugging support. The snail language (Strings Numbers Arrays and Inheritance Language) is a simple, object-oriented programming language meant to be implemented in a one-semester undergraduate course. We present a new VS Code extension to provide language support for the snail language. The extension implements support for syntax highlighting, rudimentary auto-completion, and static error-checking diagnostics using VS Code's Language Server Protocol. This report summarizes the contents of a VS Code extension and gives an overview of how an extension runs, particularly highlighting the functions of VS Code's Language Server Protocol. I also discuss how this extension can be further developed to make use of VS Code's Debug Adapter Protocol to implement a debugger with breakpoints, start/stop behavior, and variable inspection.

1 Introduction

Much of software development in the present day takes place in integrated development environments (IDEs) [6]. An IDE is a collection of software development tools, such as a code editor, debugger, and build system, often unified under a similar user interface, with the goal of simplifying the software development process and enhancing developer productivity [4, 13]. In addition to compiling these tools together, many IDEs also offer advanced features within their code editors such as syntax highlighting, code auto-completion, and error-checking diagnostics. Today, there are any number of different IDEs available for use with any given programming language, many offered as freely-downloadable for

public use. For example, a developer looking to write code in Java may choose to do so in Eclipse, IntelliJ IDEA, or BlueJ [3, 5, 7]. With so many high quality IDEs available today, it is no surprise that 75% of software developers today use an IDE in their everyday work [6]. Clearly, the IDE has become an integral part of the software development process today [14].

Visual Studio Code (VS Code), is a popular, open source text editor maintained by Microsoft [10, 12]. When first downloaded, VS Code is a lightweight text editor with minimal features. However, a number of community-built, freely downloadable extensions offer advanced language features. These extensions can be downloaded on VS Code’s online extension marketplace, and can turn VS Code into a very fast, robust, and powerful development environment for any programming task or language [11].

The snail programming language (Strings Numbers Arrays and Inheritance Language) is a simple, object-oriented programming language meant to be implemented in a one-semester undergraduate course [2]. In order to be implemented in a short time frame, snail is defined by limited features and a relatively annoying syntax. While this design makes snail easier to implement, it makes it hard for a developer to write programs in snail.

Currently, there are no tools to offer advanced language support for the snail language. This is no surprise, as snail has a small user base and was first released in February 2022 [1]. With no external support for the language, software developers are taken out of their comfort zone and are offered no guidance when navigating the snail language.

This report presents the Snail Language Support VS Code extension, which seeks to address the lack of programming support tools for the snail programming language. Snail Language Support provides several important features to make programming in the snail language easier. First, it features syntax highlighting to make reading snail code easier and help highlight key structures or keywords of the language. Next, it features rudimentary auto completion with auto-closing brackets, braces, and quotes, as well as if-else, while loop, and class definition snippets, reducing the burden of memorizing snail’s strict and unintuitive syntax. The extension also has automatic, real-time error checking diagnostics that allow a user to see syntax or parse errors in a snail program before running it for themselves. Finally, the extension is structured to support a full debugger with breakpoints, step-in, step-over, and step-out functionality.

This paper will outline the process of building the Snail Language Support VS Code extension. Specifically, we will introduce the structure of a VS Code extension that is meant to add support for new programming languages. We will also discuss how this extension uses VS Code’s language server protocol (LSP) to provide realtime error diagnostics [9]. We will address how the Snail Language Support extension may be further developed to include debugging support with breakpoints, step in/out behavior, and variable inspection, particularly highlighting the role of VS Code’s debug adapter protocol DAP [8]. Finally, we will review good software development practices such as version control and documentation.

2 Background

2.1 History of the Modern Integrated Development Environment

2.2 History of Modern Debugging Tools

2.3 What is Visual Studio Code?

3 Anatomy of a Visual Studio Code Extension

3.1 Directory Structure

3.2 Extension Contributions: The Extension Manifest

3.3 Extension Lifecycle: From Startup to Shutdown

3.4 Language Configuration

3.5 Rudimentary Autocompletion Through Snippets

3.6 Syntax Highlighting

4 Language Servers and the Language Server Protocol (LSP)

4.1 Theory

4.2 LSP in Snail Language Support

5 Debug Adapter Protocol

5.1 Theory

5.2 Debug Adapter Protocol in Snail Language Support

6 Good Practices in Software Development

6.1 Version Control

6.2 Documentation

7 Conclusions

8 Appendices

A Title of Some File

References

- [1] Kevin Angstadt. Version 1.0.0 released, February 2022.
- [2] Kevin Angstadt. Introduction, May 2023.
- [3] Eclipse Foundation. Eclipse installer 2023-03 r, Mar 2023.
- [4] Alexander S Gillis and Valerie Silverthorne. What is integrated development environment (ide)?, Sep 2018.
- [5] Kings Programming Education Tools Group, Sep 2022.
- [6] JetBrains. The state of developer ecosystem in 2019 infographic, 2019.
- [7] JetBrains. Download intellij idea, Mar 2023.
- [8] Microsoft. Debug adapter protocol, 2021.
- [9] Microsoft. Language server protocol, 2022.
- [10] Microsoft. Download visual studio code, Apr 2023.
- [11] Microsoft. Extensions for visual studio code, 2023.
- [12] Stack Overflow. Stack overflow developer survey 2022, May 2022.
- [13] Kateryna Shyniaieva. The most popular ide's for developers in 2023, Feb 2023.
- [14] Slava Vaniukov. 16 best ide's for software development: Overview for 2023, Apr 2023.