Setanta - An Irish programming language and learning environment

Interim Report - CS460 Final Year Project 2019/2020 - 5 credits

Student: Eoin Davey - 16334926 Supervisor: Dr. Barak Pearlmutter

December 15, 2019

1 Goals of the project

The primary goal of this project is to create a new, modern programming language built around the Irish language, *Setanta*, with an intuitive online learning environment, allowing the language to be used as an educational tool, with no installation required.

The goal for *Setanta* is to be a modern, dynamic, object oriented language, whose domain is education. It should be easily accessible and intuitive, feeling familiar to anyone with some programming experience, while being friendly enough to learn as a first language. *Setanta* should take large inspiration from the Irish language and culture.

The goal for the *Setanta* learning environment is to provide an intuitive interface to program *Setanta* with. It provides a modern text editor, an I/O console, as well as a powerful API to interact with a custom graphics environment, all running in the browser.

The project also has the extra goal of building an expressive, powerful parser generator for TypeScript. This parser generator is then used to develop the interpreter for *Setanta*.

2 Background

English is well established as the lingua franca of the programming world, even languages developed outside the Anglosphere are designed to be written in English, for example *Ruby* is from Japan, and *Lua* is from Brazil.

The Irish language is in daily use around Ireland. It is the primary language in many schools, primary and secondary, around the country. Ireland's education system is beginning to bring more and more Computer Science education into the school curriculum. For this reason and many others, there will be more and more Irish speakers looking to get started learning programming, and what better way than by learning on an accessible platform, in a modern language, and in their native tongue.

As *Setanta* is to be run in the browser, it is written in a web friendly language, specifically TypeScript, a strongly typed JavaScript variant. However, there was no existing parser generator for TypeScript in existence. Part of this project therefore became to create a fully functional parser generator for TypeScript, and to then use it to build the *Setanta* interpreter.

3 Progress to Date

3.1 *tsPEG* - The TypeScript Parser Generator

The TypeScript parser generator, named *tsPEG* is effectively complete, It is published on the NPM package repository, currently at version 1.1.6. *tsPEG* is a fully featured parser generator, built around the PEG grammar formalisation (a variant of Context Free Grammars). *tsPEG* is capable of generating parsers for any context free language, and even some context sensitive ones. It's built on top of an innovative regex based implicit lexing system, meaning no separate lexical analyser is required.

My supervisor thinks the parser generator itself is of sufficient independent interest, and value to the community, to be project-worthy on it's own.

tsPEG is self-hosting, the grammar for *tsPEG* grammars is defined itself in the *tsPEG* grammar. This serves as a strong verification of *tsPEG*'s power.

When I announced *tsPEG* on relevant channels online, it received strong positive attention, peaking at around 350 weekly downloads.

3.2 *Setanta* - The Irish Programming Language - Teanga Ríomhchlárúchain as Gaeilge

At the time of writing *Setanta* is well under way to completion. It's code can be found on Github. *Setanta* supports a whole list of features that you expect from a modern language. It has a REPL, supports all standard maths operators, dynamically typed, lexically scoped, first class functions, higher order functions, closures, a few types of loops and conditionals etc. It will eventually also support classes and inheritance, as well as asynchronous execution.

The interpreter for *Setanta* is being written in TypeScript, using a parser generated by *tsPEG*. It is an interpreted language that can be executed client-side in the browser, as well as locally through the node JavaScript engine. The syntax of *Setanta* will be familiar to those used to the C family of languages, but is an original syntax that is built upon the Irish language, not only in it's keywords and functions, but also is directly built on the grammatical structure of Irish.

3.3 The *Setanta* learning environment

The online learning environment for *Setanta* has a live demo online and available now. The environment has a text editor, an IO console, a graphics display panel, and a high level OOP API for interacting with the drawing panel, allowing the user to program simple games and demos. The learning environment was developed before the *Setanta* interpreter was ready, so originally used JavaScript as the language that could be written and executed. Now that the *Setanta* interpreter is operational, it now supports it. However as *Setanta* does not yet support the OOP operations required to interact with the graphics library API, the graphics are currently disabled.

4 Problems Encountered

The biggest initial problem for *Setanta* was the lack of a sufficient parser generator for TypeScript. I had decided to do the project in TypeScript, as I wanted to run the project in the browser, but I didn't want to use JavaScript because of it's weak typing system. WebAssembly was considered, but I found it wasn't mature enough to be worth experimenting with for this project. After some discussion with my supervisor it was decided it would be a good addition to the project to create my own TypeScript parser generator, *tsPEG*. I took inspiration from some other parser gnerators, largely from textX, PEG.js and pgen, to create a fully featured one for TypeScript. This also served as a great way to learn the ins and outs of the TypeScript type system.

5 Planned Steps to Completion

There are a few main steps left to complete the project. Firstly classes and inheritance constructs must be added to *Setanta*. This is important to allow interfacing with the graphics engine API in the browser learning environment, as well as being an important feature for an educational programming language. I also need to add async execution to *Setanta*, to allow things like concurrency and I/O to be done.

The learning environment also requires more work, more advanced API functionality needs to be added, to allow more interesting programs to be written, as well as some backend work to be done to allow saving of programs, along the lines of the Go playground.

The domain of the project is education, so it will benefit largely from a series of beginner-targeted documentation articles, teaching the user how to use the language. As the language runs in the browser it could be possible to make this an interactive process.