

Eoin Davey - Final Year Project Outline

Details

Student	Eoin Davey
Student Number	16334926
Supervisor	Dr. Barak Pearlmutter
Credits	5
Repository	https://github.com/EoinDavey/Final-Year-Project

Primary Goal

The primary goal of this project is to create a new programming language built on the Irish language.

Secondary Goals

Secondary goals include:

- The domain of the language should be education.
- The language should be as easy to access as possible, avoiding excessive installation steps or cross platform issues
- The language should be designed under the influence of the Irish language, It should be allowed to be influenced by Irish language structures, both grammatical and cultural.
- The language should be simple and constrained in scope and design, to keep the language accessible to all.
- *Standard* features of most languages should be present. This is to ensure that learning the language teaches fundamental programming concepts.
 - Variables (mutable / non mutable to be decided later)
 - Functions
 - Integers
 - Floats
 - Strings
 - Booleans
 - Conditional execution

Tertiary ideas

These are some ideas that might be taken forward into the project.

- Postfix notation for function application. The inspiration for this is the fact that Irish is a VSO (Verb Subject Object) language rather than an

SVO language like English. This means that Irish lists the verb (*function*) before the subject and object. In contrast in most modern OOP languages the subject is listed first, then the verb, then the arguments. This could have a syntactic effect on our language, but also a semantic one, as if the function is syntactically more prominent then a more functional paradigm might suit the language best.

- Genitive word order - In English the order of genitive case words is left to right. E.g. “Sean’s coat”, “The President of Ireland’s dog” . This property is translated across into English programming languages very directly, in most languages with structural data types these ideas could be expressed like `Sean.coat` or `Ireland.President.Dog`. This is in direct contrast with Irish however, in Irish genitive word order is right to left. “Sean’s coat” is expressed more akin to coat of Sean, (*cóta Sheain*). The same goes for “The President of Ireland’s dog”, in Irish this is *Madra Uachtaráin na hÉireann*. The word order here is dog, president, Ireland.
- Non technicality. As the language is not English, there is no value in over saturating the user with technical words for things like programs, strings, floats, functions etc. These could be replaced with familiar concepts like stories, words, actions, characters etc.
- LOGO/Scratch like visual elements: As the domain of the language is education, it would benefit greatly from a visual interface, a canvas where the user can draw pictures or make animations. This could be done in many ways including using the turtle graphics approach of LOGO, the more scratch like approach of using sprites and movement, or using programmable virtual robots.
 - Possibility to enable easier drawing of celtic knots / symbols
- Story telling: Irish has a great history of storytelling, both written and oral. Example programs can be written to play a “game” where the user plays through a historical Irish tale. e.g. Oisín and Tír na nÓg, Diarmuid and Gráinne etc.
- Class inheritance denoted using “Ó” keyword. This is a small feature that could be added to the language that comes from the cultural and historical fact that names in Irish have historically used *Ó* meaning *from* to denote ancestry.
- Use in Gaelscoil, Gaelcholaiste: Gaelscoils and Gaelcholaistes are Irish speaking primary and secondary schools respectively. As computer science is being brought into the Irish education curriculum it might be possible to trial use of the language in such schools, as part of a single CS class.
- Initial mutations: The Irish language is heavily dependent on **initial mutations**. The first consonant of a noun can be mutated in many contexts to convey possession, subject gender and other more complex

properties. In the context of a programming language this would imply some method of having multiple variable names refer to the same underlying variable.

Possible Inspirations

- **Scratch**
- **LOGO**
- **Linotte**: A french education language.
- **RoboMind**: Education language using virtual programmable robots

Technical details

My initial plan for the project is a Web App, hosted on some cloud platform, possibly Google's App Engine.

I would like the code to be executed client side, to allow for interactivity and speed. For this reason the language cannot have any network interface access or raw javascript evaluation etc.

For personal reasons I want to avoid writing as much Javascript as possible so I am thinking of looking into the WASM architecture. This would allow an interpreter to be written in a preferable language like Go or Python and be executed client side.

An alternate possible approach would be to have a compiler run server side that would return bytecode to be executed on a VM in the browser, in some custom instruction set. The VM could be written in TypeScript or Javascript or some browser language.

Initially I am not planning on having user accounts, making the site more like **Ideone**, code can be written and saved to a unique URL, where it can be forked and edited further.