



**EXPLORATION OF THE INFLUENCE OF ESOTERIC PROGRAMMING
LANGUAGES ON THE INVOLVEMENT IN THE LEARNING PROCESS**

E. Nabigazinova¹, B. Meraliyev²

^{1,2}*Suleyman Demirel University, Kaskelen, Kazakhstan*



Abstract

In this paper several types of esoteric programming languages have been chosen to analyze whether they can help to learn programming and arouse interest of non-programmers. Esoteric programming languages were designed to explore the possibilities of developing any ideas with programming languages, in other words, to implement an idea in any human-understandable way. We chose programming languages which were as close as possible to the spheres of real life, such as cooking - to write the code as a recipe of any dish, literature - the code format looks like a literary work, movies - using quotes from movies to write the code, painting - the written code looks like a picture and also jokes - using slang to make the code more understandable to a certain people. During the work we used observational research methods to understand the influence of esoteric programming languages on students' involvement. Also the selected esoteric programming languages helps to understand whether it is worth learning programming and whether learning one of the esoteric programming languages facilitates the learning process.

Keywords: Esoteric programming languages, esolangs, meme-based learning, education methods

Introduction

Many esoteric languages have now been created to help prove, create and understand conceptual things. They can help to represent how human intellect can be creative, while building new esoteric programming languages. Also, esoteric languages can be radically different from each other and be multidimensional. Esoteric programming languages can be divided into four different classes:

Recreational

Comedy

Computationally interesting

Transgressive programming or Multi Coding

Let's describe each class, Recreational esolangs created by inspiration of another language or esoteric language, as an example for the recreational esoteric language can be yaball. It is inspired by the Brainf**k, yaball has a 2-dimensional code flow and 2 different flow modes.



describes the usage of the esoteric languages and their performance on art of code. This work used three strategies to express an idea with the esoteric languages. The first, and perhaps more obvious given that much of the code is a form of text, lies in their vocabulary. The other two we can think of as logically oriented esolangs and conceptual languages [2]. As an example for the esoteric language itself, the author used LOLCODE, Befunge, Piet, Chef and Brainf**k languages. He describes their usage and tracks the performance for comparison. Let's list some details for the esoteric languages used in the work. LOLCODE is the esoteric programming language which was created by influence of the lolcat meme. The language uses elements of English Internet slang, so a person familiar with this slang can understand software code even without knowing the syntax of the language. Befunge is one of the older esoteric languages which used only two dimensions for representing code. Initially, its cells are filled with programme commands. Befunge allows you to modify the program code at runtime, so the field cells can also be used as random access memory. Brainf**k characterized by its particular minimalism, it consists of eight commands, each written with a single character. As a conclusion, the author presents the idea that all the esoteric languages help to represent creativity of the esolangs creators, and are useful for easily understanding the main concept of the problem which needs to be solved. Another work "Esoteric Programming Languages" by Sebastian Morr, tries to define five categories of the esolangs. On this work also used esoteric languages like Brainf**k and Befunge. As a minimal esolang type chosen Brainfuck, the weird one was Intercal esoteric languages, as a multidimensional esolangs defined Befunge, as the category of the hard one used Malbolge, and the poetic esolang used Shakespeare [3]. Also on work used unexpected examples of Turing complete systems which are best viewed as a subset of "discovered" or "found" esoteric programming languages. The languages were checked on Turing completeness and described the origin of the languages. Turing completeness is a fundamental concept in computer science. It helps to answer many key questions, such as why it is impossible to create a perfect anti-virus program. But at the same time it is a strikingly common phenomenon. Conclusion of the work is that esoteric languages present language users with interesting, thought-provoking puzzles that can sharpen their minds and help them better understand different ways of approaching and solving problems.^[3] Next work on education and joke programming languages, how they can improve the interest on learning programming languages. "I CAN HAS SUPERCOMPUTER? A Novel Approach to Teaching Parallel and Distributed Computing Concepts Using a Meme-Based Programming Language" by David A. Richie and James A. Ross.



This work used LOLCODE esolang to teach parallel computing approaches. Authors solve computationally demanding problems with distributed systems and parallel computing and the main problems on solving problems by using LOLCODE was, identifying and using the extensions on LOLCODE which were created for just demonstrating the algorithms not for solving it. They created the PGAS memory model (figure 3), which will allocate memory while doing some computational operations. The PGAS programming model with symmetric memory in the system may be one of the simplest distributed computing models.^[4]

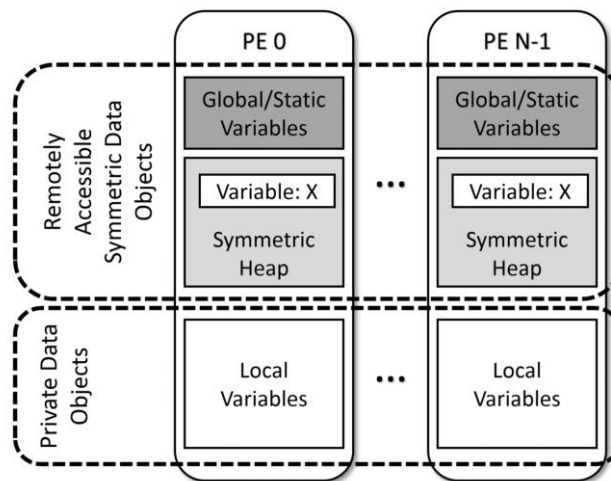


Figure 3. PGAS memory model

Also, authors used OPENSMMEM and other APIs for parallel computation, which would be helpful on writing implementation. Result of the work was the implementation of parallel computing and distribution by using LOLCODE esoteric language, and the authors wanted to create a compiler with examples to make teaching easier. LOLCODE is a meme-based slang esoteric language which can be understandable for young learners [4].

Methods and Results

After understanding different classes of esoteric languages we can divide them into two categories, for programmers and for non programmers. Esolangs for programmers can be used to train their brain to solve extraordinary problems. Esolangs for the non-programmer can be used to help them to understand programming by already known concepts. There criterias for choosing esoteric programming language for non programmers:

Readability

Writability



Reliability

Cost

Also, while selecting esoteric programming language the main point was the Turing completeness of the language. Turing completeness is the property of a system, with some simple input and output representation, to realize any computable function.

By using them selected LOLCODE, reMorse, ArnoldC, Piet, Aheui, Code Golf and Velato esoteric programming languages. But five of them are for learners with specific knowledge on music and must know Morse alphabet. That means that if they already know specific terms it would be easier for them to learn programming by this way.

In this work as a method of teaching programming is language oriented with a combination with the instruction oriented. Language oriented is not enough to learn programming, and for learners it would be hard to learn another language. But in combination with other teaching methods it can be more complete and useful [5]. While learning programming students have huge amounts of difficulties, which help us to know strong and weak students, and find methodologies for teaching. As a measurement of the tracking students influence used monitoring and tracking methods. Which shows us influence on the learning process.

Discussion

During the work described types and classes of the esoteric programming languages. This is where these questions come from.

How to choose an esoteric language?

How can they help to learn programming languages?

How can they be used in education?

Outcome of the questions above is the complex combination of the teaching and learning methods in the educational process. For the selecting language on work described categories and classes of the esoteric languages. Understanding the advantages of esolangs in learning programming was a huge part of the research. As the answer for the last question, work described a complex of teaching methods.

Conclusion

Work as a result provided a combination of teaching methods and esoteric programming languages. Also given description for the esoteric languages concept, understanding how esoteric languages work. The usage of the specific esoteric languages for the non-programmer will be



helpful to understand the concepts of the programming. Not all the esoteric languages can be used as a first programming language, and not all the esoteric languages can be really interesting to understand the basic concepts of coding. But only with the right combinations of the teaching methods and life related esoteric languages can be supportive.

References

1. Hämeen Kirjapaino, Tampere, Finland ISSN 2323-8992
2. Esolangs as Experiential Art, Daniel Temkin, ISSN 2451-8611
3. “Esoteric Programming Languages”, Sebastian Morr,
4. “I CAN HAS SUPERCOMPUTER? A Novel Approach to Teaching Parallel and Distributed Computing Concepts Using a Meme-Based Programming Language”, David A. Richie and James A. Ross
5. Methods of teaching programming, Szlávi Péter, Zsakó László, DOI:[10.5485/TMCS.2003.0023](https://doi.org/10.5485/TMCS.2003.0023) // URL: <https://habr.com/ru/post/429602/>
6. “Innovative Approaches in Teaching Programming: A Systematic Literature Review”, Simone C. Santos, Patricia Azevedo Tedesco, Matheus Borba and Matheus Brito
7. Measuring Student Success: Innovative Approaches to Understanding Diverse Learners URL: <https://files.eric.ed.gov/fulltext/ED603739.pdf>