Computer Science 3

MorseMind: Interactive translation between Morse code and natural language

Torres Mendieta David Santiago
Universidad Distrital Francisco José de Caldas

Index

Introduction

Why Morse code in the digital age?

Morse code, created in the 19th century, was essential in the history of telecommunications. Although its use is limited today, it still has relevant applications in emergency situations, accessibility, and educational settings. However, there are few modern, functional tools that allow for its learning or practical use, especially in training environments or with limited resources.

What are we trying to solve?

Is it possible to create a simple, educational, and functional tool that translates text into Morse code and vice versa?

Our goal is to develop Python software that enables this bidirectional translation, integrating input validation, file reading and writing, and an intuitive graphical interface, with a focus on accessibility and learning.

What is MorseMind?

MorseMind is an educational program developed in Python, designed to translate bidirectionally between Morse code and natural language. Its modular design allows the program to be divided into functional components: input, validation, translation, and output. With a graphical interface developed in Tkinter, the user can easily interact, enter text or code, view results, and export them. Its architecture is extensible, allowing for future integrations, such as audio or web input.

How are we going to build it?

Development is organized into independent modules that allow the code to remain clean and scalable. It begins with a text input or Morse code, which is validated and processed, and then translated into its equivalent using dictionary structures. Finally, the result is displayed on the screen or saved to a file. This structure allows for individual work on each component and facilitates future improvements.

Initial phase

A modular architecture based on Python was defined, with an educational focus. Appropriate libraries, such as Tkinter, were selected, and the logical flow was designed.

Development phase

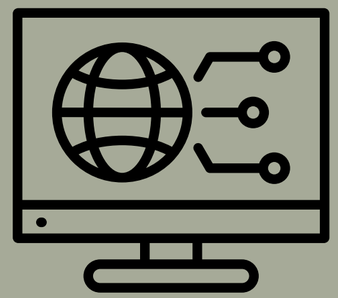
Input, validation, translation, and output modules will be programmed. File upload and export functions will be integrated, with functional testing for each component.

End phase

Integrated testing of the entire system will be conducted, evaluating translation, the interface, and error robustness. The system is being prepared for future enhancements, such as audio input.

Computational techniques

Python was chosen for its simplicity, readability, and compatibility with educational environments. Libraries such as Tkinter were used to create a functional graphical interface and operating systems for file management. Text processing was based on basic string and list techniques, ensuring a fluid user experience. This combination of technologies enables the development of intuitive, efficient, and cross-platform software.



How will we know it works?

To validate the system, unit tests were applied to each module (translation, files, interface), as well as integrated functional tests to verify the complete system flow. Test cases with valid, empty, and error inputs will be used to ensure robustness. Interface usability and correct output export will also be evaluated.

1

Two-way translation

The expected result is an accurate translation or one with errors of less than 5%.

2

Input validation

The software is expected to handle errors appropriately to ensure the translation is as accurate as possible.

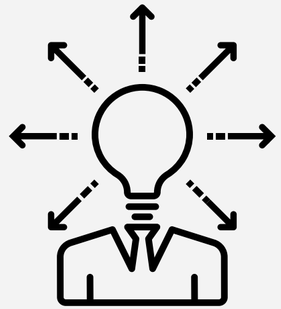
3

Exporting files

The software output will be executed in a readable and correct format to the extent possible.

Conclusions

MorseMind represents a functional, accessible, and educational solution for translating between text and Morse code. Its modular design and intuitive interface allow for a comprehensive user experience. Furthermore, the project offers opportunities to strengthen programming and logical thinking skills, providing a versatile tool for education, accessibility, and alternative communication.



References

[1] López, J., Pérez, M., Rodríguez, F., "Uso del código Morse en sistemas de comunicación de emergencia", Revista de Tecnología y Seguridad, 2020.

[2] Van Rossum, G., & Drake, F. L., Python 3: The Programming Language, O'Reilly, 2009.

[3] Smith, A., & Johnson, P., "Aplicaciones de código Morse en la era digital", Journal of Communication Engineering, 2018.

[4] Jackson, R., "Implementación de interfaces gráficas con Tkinter en Python", Programación y Desarrollo, 2019.

Thank you so much

David Torres
dstorresm@udistrital.edu.co

Computer Science 3
Universidad Distrital Francisco José de Caldas