**Doomsday**

**CSCD350 Project Report**

**Doomsday**

**Submitted by:**

**Joshua Lester (jlester3)**

Dr. Sanmeet Kaur

**Computer Science and Electrical Engineering Department**

**Eastern Washington University**

**June 2025**

**ABSTRACT**

This application is a CLI tool that provides a quick and easy way to highlight the rhyme scheme in hip hop (or other genres) lyrics. The current functionality is to save lyrics in a plaintext file, and have the app open the file and analyze the rhyme scheme. The app stores information on syllables, phonemes, and pronunciation in a local database and queries the database at runtime. The information on the syllables, phonemes, and pronunciation is used through an open-source pronouncing dictionary funded by CMU. There is a custom algorithm that maps colors to each individual syllable based on its pronunciation and the pronunciation of the surrounding syllables. The CLI tool then takes the output of this mapper algorithm and prints each syllable, and its corresponding color, to the screen. By making rhyme schemes apparent through color coding, songwriters, poets, and music enthusiasts can better understand theirs or others music.

i

**ACKNOWLEDGEMENT**

The most helpful piece of info in creating this application was Wall Street Journal’s report on their own algorithm to analyze ‘Hamilton’ the Musical (<https://graphics.wsj.com/hamilton-methodology/?gaa_at=eafs&gaa_n=ASWzDAi74VpfEJQQN-3rM_SFO71uAXjR-PXGTKd60Yi-591OygwSmSrirwTcxPEuLAo%3D&gaa_ts=68371e64&gaa_sig=DcxjiQ_GVemBGQY-Shy7Xv60m3w5M62NI2at-jkD3OoMWmWJqhaAseudQAOYAPicsWR1HAIN8SxmMIj_53Cv-g%3D%3D>). Through their description of their algorithm, I was able to piece together a similar solution to mapping colors to syllables.

This wouldn’t have been possible without the Python TUI framework, Textual (<https://textual.textualize.io/>), .NET’s ORM, Entity Framework Core (<https://learn.microsoft.com/en-us/ef/core/>), or the in-memory SQL database, SQLite (<https://www.sqlite.org/>).

ii

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Figure No.** | **Caption** | **Page No.** |
|  |  |  |  |
|  | Figure 1 | Sample YouTube video screenshot of lyrics being color coded. | 1 |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | | Figure 2 | Use case diagram. | 3 |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | | Figure 3 | System architecture diagram | 5 |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | | Figure 4 | Sequence Diagram | 6 |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | | Figure 5 | Class diagram | 7 |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | | Figure 6 | Activity diagram on getting color coded lyrics | 8 |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | | Figure 7 | Frontend code | 9 |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | | Figure 8 | Backend code | 9 |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | | Figure 9 | **Title screen with sample color coded test input.** | 10 |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | | Figure 10 | File finding screen | 11 |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  |  | | --- | --- | --- | --- | |  | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | | Figure 3 | Desc | 3 | |

iii

**LIST OF ABBREVIATIONS/GLOSSARY**

**ABBR1** Abbreviation 1

**ABBR2** Abbreviation 2

iv

**TABLE OF CONTENTS**

**ABSTRACT…………………………………………………………………………………. i**

**ACKNOWLEDGEMENT……………………………………………………………….... ii**

**LIST OF FIGURES………………………………………………………………………... iii**

**LIST OF ABBREVIATIONS……………………………………………………………... iv**

# Team Members - Bios and Project Roles

1. **INTRODUCTION**
   1. Overview of the Project
   2. Problem Statement
   3. Objectives and Scope
   4. Importance and Potential Impact
2. **REQUIREMENT ANALYSIS**
   1. User Stories with Acceptance Criteria
   2. Use Cases and Use Case Diagram
   3. Requirements
      1. Functional Requirements
      2. Non-Functional Requirements
   4. Constraints and Assumptions
3. **SYSTEM ARCHITECTURE AND DESIGN**
   1. System Architecture (Eg. Block Diagram / Component Diagram)
   2. UML Diagrams (Class, Activity, Sequence, etc.)
   3. Database Design
   4. UI Design
   5. Technology Stack Used
4. **IMPLEMENTATION AND TESTING**
   1. System/Code Screenshots
   2. Test Plan and Cases
   3. Test Results
5. **CONCLUSIONS AND FUTURE DIRECTIONS**
   1. Conclusions
   2. Reflections
   3. Future Work

**Team members:**

**Name:** Joshua Lester

**Bio:** Senior at Eastern Washington University in the Computer Science program.

**Project Role:** Team Lead, Software Developer/Designer

1. **INTRODUCTION**
   1. Overview of the Project

A CLI tool dedicated to color coding rhyme schemes based on rhyme structure and pronunciation. The tool outputs the rhyme scheme to the console screen.

* 1. Problem Statement

There are many YouTubers online that manually create color coded rhyme scheme videos on their channels. This project aims to automate this process programmatically.

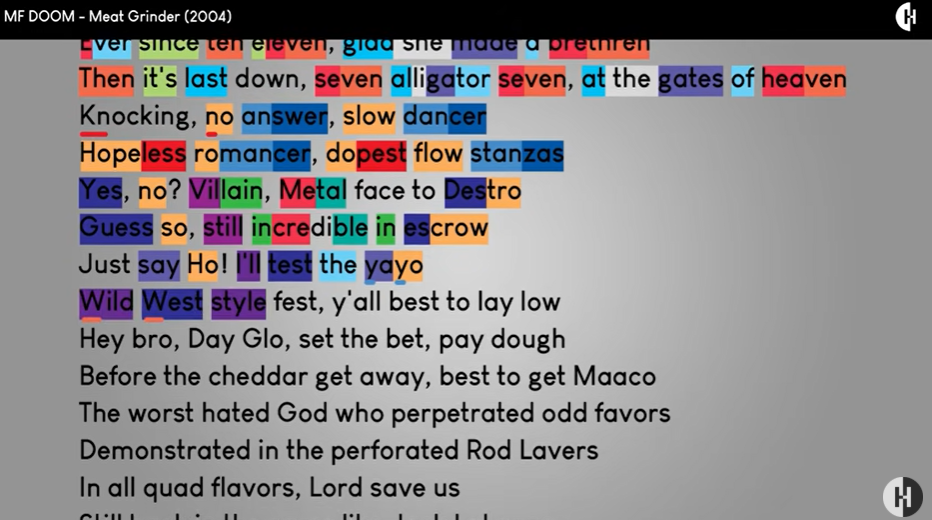


Figure 1: Sample YouTube video screenshot of lyrics being color coded.

* 1. Objectives and Scope

The main requirement is color coding a given set of lyrics, with each syllables color being based on their pronunciation and proximity to other similarly pronounced syllables and words.

1.4 Importance and Potential Impact

Could possibly be turned into a paid application, if advanced enough. It could also save YouTubers a lot of time of manually color coding rhyme schemes. These manual colorings also often have many errors in them. This project could help mitigate those errors.

1. **REQUIREMENT ANALYSIS**
   1. User Stories with Acceptance Criteria (in requirements below)

**1. As an app user, I want to be able to input my lyrics as a text file, so I don’t have**

**to type them in manually.**

**2. As an app user, I want to receive error messages when I’ve inputted the wrong**

**format, so I can fix the error with ease.**

**3. As an app user, I want to be able receive help from the app if I don’t understand**

**exactly how to use it.**

**4. As an app user, I want to be able to tell the app what to do.**

**5. As an app user, I want the app to separate the lyrics into words, syllables, and**

**pronunciation so that I can understand the pronunciation structure.**

**6. As an app user, I want the app to remember what words I’ve inputted before so it**

**can deliver my rhyme scheme faster.**

**7. As an app user, I want to be able to see my rhyme scheme visualized with colors**

**to benefit my writing process.**

**8. As an app user, I want to be able to go through an installation process that makes**

**everything easy to use so I don’t have to keep track of the internals of the app.**

**9. As an app user, I want to be able to look up potential rhymes so I can be assisted**

**in my writing process.**

**10. As an app user, I want to be able to use the app through a visual representation,**

**so I can use it easier.**

**11. As an app user, I don’t want to worry about formatting the lyrics perfectly (without punctuation), in order to make the process easier.**

**Acceptance criteria:**

**Given the user has the app and lyrics**

**When the app is run and is given the lyrics**

**Then the app returns some non-error output**

**Given the user has a text file of lyrics to input**

**When the app is run and given the text file**

**Then the app can parse the file and returns some non-error output.**

**Given the user has app and some form of lyrics**

**When app is run and takes lyrics**

**Then the app returns non-error output**

**Given app is run from command line**

**When help is specified**

**Then help style manual page is output for user**

**Given app takes in lyrics**

**When lyrics are in incorrect format**

**Then error message is cleanly output and program exits**

**Given app is run**

**When given specifications**

**Then the correct option is ran**

**Given app is given lyrics**

**When asked to color code rhyme scheme**

**Then app doesn't return error code during separation of lyrics**

* 1. Use Cases and Use Case Diagram

**A diagram of a person's process

AI-generated content may be incorrect.**

**Figure 2: Use case diagram.**

* 1. Requirements
     1. Functional Requirements

**1. There will be a feature to intake lyrics through the command line.**

**2. There will be a feature to intake lyrics through the command line as a file. This feature will use another software tool for reading files.**

**3. There will also be a feature to intake lyrics as the user types them out. This feature will use another software tool for keeping track of the lyrics.**

**4. There will be a feature let the user know about errors such as wrong file format.**

**5. There will be a feature to ask the app for help.**

**6. There will be a feature to find out what the user wants the app to do (as there might be multiple modes).**

**7. There will be a feature to divide the lyrics into lines, words, syllables, and then into pronunciation for better analyzation. This feature will rely on a pronunciation dictionary.**

**8. There will be a feature to store words from pronouncing dictionary and past words inputted and rhymes found into storage in the app.**

**9. There will be a feature to color the lyrics in the console text. This feature will use a tool for coloring text.**

**10. There will be a feature to query the application on words that rhyme with a usergiven word.**

**11. There will be a feature to parse the lyrics and ignore values that aren’t essential to the analyzation of the lyrical content (such as punctuation). The app will also put the nonessential characters back to maintain integrity of lyrics/input.**

* + 1. Non-Functional Requirements

**1. There will be a feature to run through an installation process, to keep the maintenance of the app clean and easy.**

* 1. Constraints and Assumptions

1. App has dictionaries stored in memory

1. Different runtime options are available (color coding or finding word to rhyme)
2. App is runnable from command line
3. App can take in some form of lyrics
4. App can accept text files
5. **SYSTEM ARCHITECTURE AND DESIGN**
   1. System Architecture (Eg. Block Diagram / Component Diagram)

A diagram of a computer program

AI-generated content may be incorrect.

**Figure 3: System architecture diagram.**

* 1. UML Diagrams (Class, Activity, Sequence, etc.)

A diagram of a computer program

AI-generated content may be incorrect.Figure 4: Sequence diagram on getting lyrics/splitting the words

A diagram of a computer program

AI-generated content may be incorrect.

Figure 5: class diagram for Doomsday project

A diagram of a computer program

AI-generated content may be incorrect.

Figure 6: Activity diagram on getting color coded lyrics

* 1. Database Design

Rhymes Table: main entity

Id: int

Word: string

PlainSyllables: string[]

PhonemeSyllables: string[]

* 1. UI Design

No specific UI, as this is a console application. There are several screens the app will go through including a title screen, a screen to select a file, and the final color coded lyrics screen.

* 1. Technology Stack Used

Frontend: Textual TUI framework in Python

Backend: .NET console application in C#; used Entity Framework Core to connect to DB

DB: SQLite database

1. **IMPLEMENTATION AND TESTING**
   1. System/Code Screenshots

A screen shot of a computer code

AI-generated content may be incorrect.

**Figure 7: frontend code**

A computer screen shot of a program code

AI-generated content may be incorrect.

**Figure 8: backend .NET color mapper algorithm code.**

A screenshot of a computer screen

AI-generated content may be incorrect.

**Figure 9: Title screen with sample color coded test input.**

A screenshot of a computer program

AI-generated content may be incorrect.

**Figure 10: file finding screen.**

* 1. Test Plan

Testing plan:

- FileUtils

- Test getLyrics()

- Test getWordsList

- Test getCmuDict

- Test getPlainSyllableDict

- For the above, test input, null input, and that correct functionality occurs

- RhymeUtils

- Integration test the entire process of color coding rhyme schemes

- app.py

- Unit test json to markdown

- Unit test custom widgets (Textual)

* 1. Test Results

All tests pass.

1. **CONCLUSIONS AND FUTURE DIRECTIONS**
   1. Conclusions

**This app is still a work in progress, but will continue to work on it to develop into a fully fledged app that YouTubers can use. I am proud of the work I completed as a one man team.**

* 1. Reflections

In the future I would rather work with a team. It is much better to bounce ideas off of someone and have someone share the load in developing an application. Plus you can get more progress completed.

* 1. Future Work
* Add voice recognition so songs (.mp4, .wav files) can be inputted or recording can be sung into to color code rhyme scheme
* Possibly port into easier to use application than console app (for non-technical users)
* Add ability to color code over time (live) to replicate manual-made YouTube videos