QtGroove Documentation

Object Oriented Programming

Software Engineering Program,

Department of Computer Engineering,

School of Engineering, KMITL

67011090 Chanunyu Chinnawuth
67011352 Theepakorn Phayonrat

Preface

This project, QtGroove, was undertaken as part of Object Oriented Programming course in Software Engineering at KMITL. Throughout this project, We have gained valuable insights into developing Qt C++ based application, teamwork management and developing workflow. This preface serves to outline the journey that led to the final outcome, which aims to contribute to the field of Software Engineering by providing and demonstrating the power of C++ in building efficience multi-platform graphical user interface (GUI) application.

Abstract (Waiting for finalization)

This project, titled RUSH: Rust Shell Terminal, presents the design and implementation of a command-line shell application developed in the Rust programming language. As part of the Elementary System Programming course in Software Engineering at KMITL, RUSH was created to develop a fast, and efficient custom shell using the Rust programming language that provide users with typical features found in standard command shells (such as Bash, Zsh, etc.), including executing programs, navigating directories, file management, environment management while utilizing Rust's safety guarantees to minimize common security vulnerabilities.

The project demonstrates a range of system programming concepts, including process management, command parsing, and file handling, showcasing the capabilities of Rust in building performant and secure applications. RUSH supports essential command-line functionalities such as a help system, history logging, and automatic configuration of necessary files, providing a user-friendly and highly customizable terminal experience.

This work aims to contribute to the field of Software Engineering by offering a flexible and safe alternative to traditional shell interfaces, especially suited for developers seeking to understand system programming through the Rust language. The final outcome highlights Rust's effectiveness in system software development, encouraging further exploration of Rust for similar high-performance applications.

Contents

1 Introduction **4**
- 1.1 Project Overview . . . 4
- 1.2 Background . . . 4
- 1.3 Objective . . . 4

2 Project Overview **5**
- 2.1 Design . . . 5
  - 2.1.1 asdf . . . 5
  - 2.1.2 jkl; . . . 5
  - 2.1.3 qwerty . . . 5
- 2.2 Database . . . 6
  - 2.2.1 playlist.db . . . 6

3 Installation and Execution Guide **7**
- 3.1 Git Clone from the Remote Repository . . . 7
- 3.2 Build and Run the program (Waiting for finalization) . . . 7

4 Summary **8**
- 4.1 Learning Outcomes . . . 8
- 4.2 Accomplishment . . . 8

5 References **9**

6 Appendix **10**
- 6.1 Demonstration Video . . . 10

Chapter 1

Introduction

1.1 Project Overview

QtGroove is a graphic-based music player written in C++ using the Qt framework. The project aims to be a lightweight music player with a friendly user interface.

QtGroove will have the functions of a typical music player like a file browser, the ability to make playlists, showing music file info, and having a bit of extra functions like speed up playback or player customization.

1.2 Background

We wanted to created our own multi-platform GUI music player, which is efficience to navigate through the UI with low learning curve.

1.3 Objective

This project aims to create a lightweight and multi-platform music player as an alternative to other music players. The app can great for listening to local music files. The making of this app also serves as an experience for us to learn C++ and work with the qt framework.

Since this is a duo project, it is a great opportunity to learn teamwork and strive to make the best products.

Chapter 2

Project Overview

2.1 Design

2.1.1 asdf

asdf

2.1.2 jkl;

jkl;

2.1.3 qwerty

qwerty

2.2 Database

DB

2.2.1 playlist.db

playlist.db

Chapter 3

Installation and Execution Guide

3.1 Git Clone from the Remote Repository

```
git clone https://github.com/Pottarr/QtGroove.git
cd QtGroove
```

3.2 Build and Run the program (Waiting for finalization)

```
cargo build
cargo run
```

Chapter 4

Summary

4.1 Learning Outcomes

- We have learnt fundamental of concepts of creating good UX and UI.
- We have learnt how to develop multi-platform application using C++ Qt.
- We have learnt the workflow of project developing.
- We have learnt how to use Version Control to help developing application.

4.2 Accomplishment

We have created a user friendly multi-platform music player application.

Chapter 5

References

- Qt Group. (2025). *Qt Documentation*. Retrieved from `https://doc.qt.io/`

Chapter 6

Appendix

6.1 Demonstration Video