###### **الجمهورية الجزائرية الديمقراطية الشعبية**

###### **وزارة التعليم العالي والبحث العلمي**

|  |  |  |
| --- | --- | --- |
| **جامعة باجي مختار – عنابـــــــــــــــة** |  | **Université Badji Mokhtar - Annaba**  **Badji Mokhtar – Annaba University** |
| **Faculté : Technologie** | | |
| **Département : Informatique** | | |
| **Domaine : Mathématique-Informatique** | | |
| **Filière : Informatique** | | |
| **Spécialité : systèmes informatiques** | | |

**Thème**

**Projet de lecteur audio avec PyQt5**

**Présenté par :** Abi Abde Raouf

**Domain :** Mathématique-informatique

**Filière :** Informatique

**Spécialité :** inélégance artificielle

**Niveau*:*** Master1

**Année scolaire : 2023-202**

**Table des** **matières**

[Introduction 2](#_bookmark0)

[Objectifs du Projet 3](#_bookmark1)

[Démarche Suivie 4](#_bookmark2)

T[echnologies Utilisées 6](#_bookmark3)

[Résultats Obtenus 7](#_bookmark4)

[Perspectives Futures 8](#_bookmark4)

[Conclusion 9](#_bookmark4)

**Introduction**

Le mini-projet réalisé consiste en un lecteur audio interactif développé en utilisant la bibliothèque PyQt5 de Python. Ce lecteur audio offre une interface utilisateur conviviale permettant à l'utilisateur de charger et de lire des fichiers audio, de régler le volume, la vitesse de lecture, et de naviguer entre les pistes de sa playlist. Le projet intègre des fonctionnalités telles que des curseurs de contrôle stylisés, des boutons de lecture/pause, des indicateurs de position et de durée, et une interface graphique élégante.

La conception du lecteur audio s'articule autour de la classe **AudioPlayer**, qui utilise le module **QMediaPlayer** de PyQt5 pour la gestion des fichiers audio. L'interface utilisateur offre une expérience visuelle intuitive avec des boutons d'action, des curseurs de contrôle stylisés, et des indicateurs de statut dynamiques.

La structure du code est modulaire et bien organisée, facilitant la compréhension, la maintenance et l'extension du projet. Le lecteur audio peut être intégré dans divers contextes, offrant une solution légère et flexible pour la lecture de fichiers audio.

Dans ce rapport, nous examinerons en détail les différentes fonctionnalités du lecteur audio, son architecture, ainsi que les choix de conception effectués tout au long du développement. En outre, nous aborderons les améliorations potentielles et les perspectives d'extension de ce projet.

**Objectifs du Projet**

Le projet de développement du lecteur audio avec PyQt5 s'articule autour de plusieurs objectifs fondamentaux, visant à créer une expérience utilisateur complète et personnalisée. Les principaux objectifs sont les suivants :

1. ***Lecture Audio Intuitive :***

Concevoir une interface utilisateur intuitive pour la gestion facile de la lecture audio, mettant l'accent sur la simplicité et l'accessibilité.

1. ***Contrôle Audio Avancé :***

Fournir des fonctionnalités de contrôle audio avancées, notamment le réglage du volume et la modification de la vitesse de lecture.

1. ***Gestion de Playlist :***

Intégrer une fonctionnalité de playlist pour permettre aux utilisateurs de créer, charger et écouter plusieurs fichiers audio de manière séquentielle.

1. ***Personnalisation de l'Interface :***

Utiliser des styles CSS pour personnaliser l'apparence des éléments graphiques tels que les curseurs et les boutons.

**Démarche Suive**

1. ***Analyse des Besoins*** :

Le projet a débuté par une analyse approfondie des besoins, identifiant les fonctionnalités essentielles telles que la lecture intuitive, le contrôle avancé, la gestion de playlist, et la personnalisation de l'interface.

1. ***Choix de la Technologie :***

PyQt5 a été sélectionné comme le framework principal pour la conception de l'interface utilisateur en raison de sa puissance, de sa flexibilité, et de son intégration harmonieuse avec le langage de programmation Python.

1. ***Conception de l'Interface Utilisateur :***

La conception de l'interface utilisateur a été réalisée en utilisant les outils fournis par PyQt5, en mettant l'accent sur l'esthétique et l'intuitivité. Des styles CSS ont été incorporés pour personnaliser l'apparence des éléments graphiques.

1. ***Implémentation du Modèle MVC :***

L'architecture du projet a été structurée selon le modèle MVC. La classe AudioPlayer a été développée en tant que modèle, encapsulant la logique métier. Les contrôleurs ont été implémentés à travers les méthodes de cette classe pour répondre aux interactions utilisateur.

1. ***Intégration des Fonctionnalités :***

Les fonctionnalités clés, telles que le chargement de fichiers audio, le contrôle de lecture, les ajustements de volume et de vitesse, ainsi que la gestion de playlist, ont été intégrées de manière modulaire.

1. ***Tests et Débogage :***

Des tests approfondis ont été effectués pour garantir le bon fonctionnement de chaque fonctionnalité. Les erreurs ont été identifiées et corrigées de manière itérative.

**Technologies Utilisées**

1. ***PyQt5 :***

Framework pour la création d'interfaces utilisateur en utilisant le langage de programmation Python.

1. ***QMediaPlayer :***

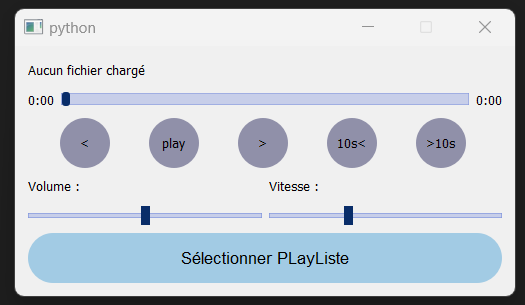
Module de PyQt5 pour la gestion de la lecture audio.

1. ***Styles CSS :***

Utilisés pour la personnalisation de l'interface graphique.

**Résultats Obtenus**

Le projet a abouti à la création d'un lecteur audio interactif avec une interface élégante. Les objectifs principaux, tels que la lecture intuitive, le contrôle audio avancé, la gestion de playlist, et la personnalisation de l'interface, ont été atteints. L'architecture MVC a contribué à une organisation claire du code, favorisant la maintenabilité et l'extensibilité.



**Perspectives Futures**

Les perspectives d'amélioration ont été identifiées, notamment l'ajout de métadonnées, la gestion persistante de playlist, l'intégration d'un égaliseur audio, et l'extension de la compatibilité avec d'autres formats audio. Ces améliorations sont envisagées pour répondre aux besoins évolutifs des utilisateurs.

**Conclusion**

En conclusion, le projet a été une réussite, démontrant l'efficacité de l'utilisation de PyQt5 dans le développement d'applications graphiques interactives en Python. Les résultats obtenus constituent une base solide pour des développements futurs et des évolutions fonctionnelles.