Projet d’Algorithmique et Structure de données

Sujet n°6 : **Historique d’action dans un éditeur de texte** (Undo/Redo)

**Membres du groupe :**

* TSAFACK NGOUFACK Ernis Merkel
* NOUBISSI MEIKEM Indira Leslie
* NANA Laurianne Audrey
* TANKAM SOH Raoul Daril

**Chargé** **de** **cours** : Ing. ATANGANA Guy Martial

**Plan**

1. **Introduction**
   1. But du document
   2. Portée du projet
   3. Définitions, Acronymes et Abréviations
   4. Références
   5. Vue d’ensemble du reste du document
2. **Description Générale** 
   1. Contexte du produit
   2. Besoins des utilisateurs
   3. Caractéristiques du produit
   4. Contraintes
   5. Hypothèses et Dépendances
3. **Exigences spécifiques** 
   1. Exigences fonctionnelles
   2. Exigences non fonctionnelles
   3. Exigences d’Interface Externe
4. **Préparation à la livraison** 
   1. Plan de déploiement
   2. Exigences de formation
   3. Exigences de Documentation
   4. Exigences de Maintenance
5. **Annexes**

**1.Introduction**

* 1. **But du document**

Ce document est destiné à toute personne souhaitant réaliser un éditeur de texte simple développé avec le Framework Qt de A à Z que vous soyez débutant ou aguerri. Il est également fait pour présenter le résultat de notre travail et les méthodes employés pour réaliser notre éditeur de texte et y intégrer un historique d’action afin de permettre à nos camarades de pouvoir reproduire ce projet de façon ludique et efficace.

* 1. **Portée du projet**

Le projet vise à fournir un éditeur de texte basique sous Windows, permettant d’effectuer des opérations d’édition classique (ajouter, supprimer, copier/coller) et de revenir en arrière ou rétablir des actions (Undo/Redo). Il intègre aussi des fonctionnalités de mise en forme de texte (police, taille, couleur), la sauvegarde de fichier et l’ouverture de fichier texte.

* 1. **Définitions, Acronymes et Abréviations**
* Les fonctions Undo (Annuler) et Redo (Rétablir) permettent à l’utilisateur d’annuler ou de de rétablir la ou les dernières actions qu’il a effectuées, généralement sur un document, afin de le faire revenir à un état antérieur.
* Qt est un Framework C++ pour interface graphique qui permet d’utiliser le principe d’abstraction pour écrire du code sans pour autant connaitre toutes les ficelles du langage UI (Interface Utilisateur).
* Le principe d’abstraction est un concept qui consiste à cacher les détails de fonctionnement d’un système à l’utilisateur pour lui fournir uniquement le nécessaire.
  1. **Références**

Documentation officielle Qt : https://doc.qt.io

Spécifications IEEE 830

Création de logo : logomark.com, remove.bg, online-convert.com

Outil d’installation : Inno Setup Compiler

* 1. **Vue d’ensemble du reste du document**

Ce document présente les spécifications fonctionnelles et techniques de l’application, les exigences de maintenance et de formation, notre plan de déploiement, les contraintes liées au développement et les perspectives d’évolution.

**2.Description Générale**

**2.1 Contexte du produit**

Ce logiciel peut remplacer un éditeur de texte basique dans un environnement Windows ; notre produit sera développé grâce à la bibliothèque Qt qui nous fournira une interface graphique, il est donc compatible avec tout environnement de développement où Qt est installé et fonctionnel.

**2.2 Besoin des utilisateurs et objectifs du projet**

* Ce projet est destiné au grand public, étudiants ou professionnels ayant besoin d’un éditeur simple
* Ce projet vise à faciliter la saisie et la mise en forme de texte pour tous nos utilisateurs, les différents éléments qui sont intégrés dans notre éditeur de texte (supprimer, ajouter des caractères, copier, colle, changer la couleur ou la taille du texte, Annuler, Rétablir) permettent de rendre la saisie de texte beaucoup plus rapide et optimale.

**2.3 Caractéristiques du produit**

Les fonctionnalités de notre éditeur de texte sont : l’édition optimisée de texte, l’ouverture et la sauvegarde de fichiers .txt, les fonctions Copier/Couper/Coller, Undo/Redo (Annuler/Rétablir) pour une gestion plus rapide du texte. Ainsi que des options de rendu visuel pour modifier la couleur et la taille du texte avec une interface intuitive en Qt.

**2.4 Contraintes**

Les contraintes liées sont notamment :

* Application Desktop uniquement
* Propriétés intellectuelles : Qt impose des droits d’auteurs sur tout projet fait avec dans le cadre de la commercialisation.
* Le projet prend en pratique quelques heures (pour les débutants) à quelques minutes (pour les plus aguerris) pour être fait et nécessite juste d’avoir un PC pour pouvoir coder
* Sécurité : Le code source accessible peut poser des problèmes de sécurité
* Concurrence : Le produit est en compétition avec des éditeurs établis comme Bloc-notes ou Microsoft Word sous Windows.

**2.5 Hypothèses et Dépendances**

* L’utilisateur a installé Qt et ce dernier est fonctionnel sur son PC.
* L’utilisateur maitrise les fonctions de base d’un éditeur de texte et peut se servir correctement de chacune d’entre elle.
* Ce projet est conçu avec l’hypothèse d’une commercialisation future.

**3. Exigences Spécifiques**

**3.1** **Exigences fonctionnelles (ERF)**

* ERF001 : Ouvrir un fichier texte (.txt)
* ERF002 : Sauvegarder des modifications
* ERF003 : copier, couper, coller
* ERF004 : Modifier la couleur du texte
* ERF005 : Modifier la taille de la police
* ERF006 : Afficher une fenêtre de configuration
* Acteurs : Etudiants à ICT-U du Cameroun supervisés par leur enseignant en C++
* Préconditions : Système Windows 32 bits minimum avec Qt installé
* Déclencheur : Lancement de l’exécutable
* Séquence normale : Installation et lancement du logiciel avec toutes les dépendances
* Flux alternatif : Arrêt de l’installation ou annulation par l’utilisateur
* Priorité : Compatible Windows uniquement

**3.2 Exigences non fonctionnelles (ERNF)**

* ERNF001 : Interface simple et intuitive
* ERNF002 : Temps de chargement < 15s
* ERNF003 : L’application doit être stable et ne pas planter

**3.3 Exigences d’Interface Externe**

* UI : Interface Utilisateur basés sur Qt
* Logiciel : Interaction avec le système de fichiers local uniquement.

**4. Préparation à la livraison**

**4.1 Plan de déploiement**

L’application sera fournie sous forme exécutable Windows via licence gratuite. Possibilité de mise à jour vers une licence commerciale ultérieurement.

**4.2 Exigences de Formation**

Ce projet ne demande aucune formation spécifique.

L’interface simple permet une prise en main immédiate.

**4.3** **Exigences de Documentation**

L’utilisation de notre éditeur de texte est très intuitive et ne nécessite pas de mode d’emploi. Le principe étant assez simple et similaire au précèdent éditeur que vous avez déjà utilisé.

Un fichier README accompagnera le projet avec instructions de compilation, exécution et contributions.

**4.4 Exigences de maintenance**

Evolutivité prévue pour ajouter la gestion de nouveaux formats (.PDF), amélioration de l’interface et sécurisation du code.

**5.Annexes**

* **Glossaire**
* UI : Interface Utilisateur
* Qt : Framework graphique
* ERF : Exigence Fonctionnelle
* ERNF : Exigence Non Fonctionnelle
* PDF : Portable Document Format (format de fichier)
* **Prototype**

Voir l’interface dans Qt Creator

* **Historique**

Version initiale du projet réalisé dans le cadre d’un projet étudiant.