



[Date]

Editeur de texte

Plan de test

SI-programing

UFR MI

1. INTRODUCTION

1.1 DESCRIPTION DU PROJET

Ce projet s'inscrit dans le cadre d'un exercice de génie logiciel, le but du projet est de concevoir un éditeur de texte pour le cours de génie logiciel.

1.2 BUTS DU PLAN DE TESTS

Le but de ce document est de faire un plan de test pour le logiciel afin de vérifier la bonne marche des classes, méthodes et fonctionnalités définies afin de mieux gérer les éventuelles erreurs.

1.3 PORTÉE DU PLAN DE TEST

Ce plan de test s'adresse en premier lieu aux membres de l'équipe de développement du logiciel ensuite au professeur de génie logiciel.

Notre logiciel contient des outils élémentaires pour le traitement des fichiers texte

Date de début du projet : 16/03/2021, nous avons défini des boutons pour les différentes options de saisie (copier, coller, gras ...), des options de sauvegarde, d'ouverture et d'impression ; ces différentes fonctionnalités feront l'objet de tests.

1.4 RÉFÉRENCES

Les différents documents utiliser pour la rédaction de ce plan test sont :

- *Ref-1* Compte Rendu de réunion
- *Ref-2* SAD (version 1.0) projet Editeur de texte département de mathématique et informatique
- *Ref-3* SDP (version1.0)

2. INVENTAIRE DES TESTS

PREMIÈRE VERSION

❖ Fonctionnalité Ouvrir

Le test sera porté sur la fonction `askopenasfilename ()` du module `filedialog` qui est appelée dans notre méthode `ouvrir ()` associée à la fonctionnalité ouvrir

❖ Fonctionnalité Enregistrer/ Enregistrer sous

Le test sera porté sur la fonction `asksaveasfilename ()` du module `filedialog` qui est appelée dans la méthode de classe `Enregistrer_sous`

la méthode `Enregistrer` est en majeure partie dérivée de la méthode `Enregistrer_sous`.

❖ Fonctionnalité Nouveau/Quitter

Ces deux fonctionnalités ne nécessitent pas de tests en tant que tel car elles ne sont composées que d'instructions natives au langage donc relativement sûres.

❖ Fonctionnalité imprimer

❖ Les fonctionnalités de modification des formats de texte (souligner ; barré ; mise en gras ...)

Les tests sont effectués sur le bout de texte à modifier

En effet il nous faut vérifier que le champ de texte est non et que des caractères de texte sont bien sélectionnés afin d'assurer que les actions se seront bien exécutés.

3. STRATÉGIE DE TESTS

Nous effectuons des tests unitaires sur les différentes méthodes susmentionnées afin de gérer les éventuels cas d'exceptions et aussi s'assurer de l'intégrité des données

3.1 <LES FICHIERS >

3.1.1 INTÉGRITÉ DE DONNÉES

Objectif:	TESTER L'EXISTENCE DES DONNEES QUI SONT DANS NOTRE CAS ICI LES FICHIERS A OUVRIR A SAUVEGARDER LES TEXTES SELECTIONNEES POUR ETRE MIS EN GRAS ;ITALIQUE ;SOULIGNEE ;BARRE
Technique:	NOUS UTILISONS LE SYSTÈME DES EXCEPTIONS NATIVES A PYTHON POUR GERER LES CAS D'EXISTENCE DES FICHIERS DE SAUVEGARDE OU DES FICHIERS D'OUVERTURE, S'ASSURER DE LA BONNE EXECUTION DES COMMANDES ASSOCIEES AUX BOUTONS
Critère de succès:	LE RESULTAT DES EXECUTIONS EST CELUI ATTENDU
Considérations particulières:	PAS DE NOTES

3.1.2 FONCTIONNEL

Objectif:	VIRIFIER QUE LES FONCTIONS MENTIONNÉES PLUS HAUT TRAITENT LES DONNÉES QUI LEURS SONT TRANSMISES
Technique:	VÉRIFIER L'AUTHENTICITÉ DES FICHIERS DE SAUVEGARDE VERIFIER QUE LE FICHIER OUVERT EST BIEN AUTHENTIQUE À CELUI OUVERT DEPUIS LA MÉMOIRE DE MASSE
Critère de succès:	LE RESULTAT DES EXECUTIONS EST CELUI ATTENDU
Considérations particulières:	AUCUNES

3.1.3 PERFORMANCE

Objectif:	TESTER RAPIDITE ROBUSTESSE PORTABILITE
Technique:	NOUS AVONS EXECUTE LE LOGICIEL SUR DIFFERENTES CATEGORIES DE MACHINES IL S'EXECUTE MOYENNENT RAPIDEMENT SUR CES MACHINES ILS DISPONIBLE SUR WINDOWS LINUX ET MAC OS AUSSI NOUS TESTER LE LOGICIEL SUR DES MACHINES DE BASE PERFORMANCES (1GB RAM, CPU : PENTIUM)
Critère de succès:	IL S'EXECUTE SANS PROBLEME SUR TOUS LES SYSTÈME ET EN TEMPS RELATIVEMENT PETIT
Considérations particulières:	AUCUNES

3.1.4 SÉCURITÉ

Objectif:	RIEN
Technique:	RIEN
Critère de succès:	RIEN
Considérations particulières:	RIEN

3.1.5 INTERFACE UTILISATEUR

Objectif:	VERIFIER S'IL EST ASSEZ INTUITIF POUR DES UTILISATEURS
Technique:	TESTER PAR DES UTILISATEURS LAMBDA CHOISIS ALEATOIREMENT
Critère de succès:	ACCESSIBILITES ,INTUITIFS ;FACILITES DE MANIPULATIONS

Considérations particulières:	<p>SPÉCIFIQUES À CE TYPE DE TEST.</p> <p>CONSIDÉRER QUE LES TESTS DOIVENT AIDER À DÉMONTRER LES FACTEURS SUIVANTS, TIRÉS DE LA NORME ISO 25000 :</p> <ul style="list-style-type: none"> • PERTINENCE • FACILITÉ D'APPRENTISSAGE • OPÉRABILITÉ • PROTECTION DE L'UTILISATEUR FACE À UNE ERREUR • INTERFACE UTILISATEUR ESTHÉTIQUE, • ACCESSIBILITÉ
-------------------------------	---

3.1.7 OUTILS

- ❖ Test manuel
- ❖ Test des utilisateurs

DEUXIEME VERSION

❖ Fonctionnalité Ouvrir

Le test de la fonctionnalité ouvrir fut axé sur la fonction python que nous avons défini. Cette fonction se doit d'ouvrir un fichier et de convertir les données nécessaires (nom du fichier, contenu du fichier et chemin du fichier). Les tests se sont bien déroulé, ce qui indique les données attendu de la fonction y sont bien retournées après appel.

❖ Fonctionnalité Enregistrer/ Enregistrer sous

Les fonctionnalités Enregistrer/Enregistrer sous suivent les mêmes principes et utilise la fonction. Le test avait pour objectif de s'assurer que toute information est bien sauvegardée et bien écrire dans le fichier spécifier et aussi que le chemin du fichier est bien conservé comme défini par l'utilisateur.

❖ Fonctionnalité Nouveau

Le test de la fonctionnalité Nouveau consistait simplement à vérifier que tous les fonctionnalités de l'éditeur fonctionnent bien lorsqu'on crée

plusieurs autres instances du même éditeur (l'éditeur est vu ici comme un objet)

❖ Fonctionnalité imprimer

Pour cette fonctionnalité, nous nous sommes assuré que le contenu de l'éditeur est bien formaté et que l'appel de l'API print de l'objet Window est se déroule bien.

❖ Les fonctionnalités de modification des formats de texte (souligner ; barré ; mise en gras ...)

Les tests sont effectués sur des bouts de texte. Nous sommes assurées que les effets visuelles les actions s'exécutes bien comme chaque fonction est sous-entendu fonctionnée.

3. STRATÉGIE DE TESTS

Nous effectuons des tests unitaires sur les différentes méthodes susmentionnées afin de gérer les éventuels cas d'exceptions et aussi s'assurer de l'intégrité des données.

Cette version 2 du projet reprend exactement les mêmes stratégies de test que la version 1.