
Projet de mesure du temps passé par tâche

Objectifs

- brainstorming
- définition des tâches et séquences d'interaction
- mise en œuvre de prototypage rapide
- conception d'interfaces et implémentation en JavaFX

1 Contexte et problématique

Ce projet a pour objectif de concevoir et développer l'interface d'un logiciel destiné à la mesure du temps passé par tâche. Les tâches peuvent être de diverses natures : travail sur le projet d'IHM, rédaction d'un rapport, travail administratif, recherche de stage ou encore travail sur un projet pour un client. Le point commun de ces tâches est qu'elles requièrent de travailler sur un ordinateur de bureau (ordinateur portable ou de bureau). Les tâches peuvent comprendre des sous-tâches pour lesquelles on souhaite également mesurer le temps passé. L'utilisateur est libre de définir la granularité des tâches et sous-tâches : dans le cadre de la rédaction d'un rapport, une tâche peut être vue comme la rédaction du rapport lui-même ou la rédaction de chacun de ses chapitres ou encore la rédaction de sous-parties dans les chapitres.

Une exigence essentielle de l'interface est de pouvoir très rapidement et facilement démarrer et arrêter un compteur de temps pour une tâche en cours et passer d'une tâche à une autre. La prévention et la correction des erreurs (oubli d'arrêter ou démarrer un compteur) est également très importante pour garantir la fiabilité des mesures réalisées. D'autres caractéristiques essentielles sont la visualisation du temps passé sur chaque tâche, si possible en sachant comment le temps a été utilisé pour chacune des tâches, le renommage ainsi que le regroupement de tâches.

Les utilisateurs visés sont des employés qui souhaitent effectuer une analyse rétrospective de l'utilisation de leur temps de travail ou alors qui ont besoin de rendre des comptes à un supérieur sur le temps passé par projet, dans un objectif de facturation à un client ou pour vérifier qu'un projet ne va pas prendre de retard.

2 Matériel à disposition

Suivant les choix de conception adoptés, l'application à développer peut se prêter à l'utilisation de la barre d'état du système (System Tray). Un exemple d'application mettant en œuvre le System Tray est disponible sur cette page¹. Ce code a été testé avec succès sur macOS et Windows 10. Sous Linux, ne fonctionne pas avec le gestionnaire de fenêtres Gnome.

3 Travail demandé

Il vous est demandé de réaliser une interface qui permette de mesurer le temps passé par tâche selon le cahier des charges et les exigences définis ci-dessus.

1. <https://gist.github.com/jewelsea/e231e89e8d36ef4e5d8a>

4 Résultats attendus

L'application sera développée en utilisant JavaFX.

Vous devrez rendre sur Moodle une archive **ZIP** contenant :

- un lien vers le projet gitlab de l'IUT², paramétré en visibilité privée, avec votre intervenant de TD / TP ajouté en qualité de "Maintainer"
- un répertoire src avec les sources du projet
- un répertoire ressources avec les ressources nécessaires pour exécuter votre projet. Votre application doit démarrer avec un ensemble de données fictives pour mieux apprécier son utilisation.
- un jar exécutable qui fonctionne avec Java 8.
- un export au format pdf de vos mockups Balsamiq : dans votre pdf, il doit être possible de cliquer sur les éléments de l'interface pour enchaîner les écrans et ainsi explorer une séquence d'interaction.
- un compte rendu **au format pdf** qui suit ce modèle³
- Une vidéo de présentation de votre projet, conforme aux exigences disponibles ici⁴.

Le projet est à réaliser de manière individuelle. Des outils de détection du plagiat seront utilisés.

Barème indicatif de notation :

Critère	Points
Respect des consignes : archive zip et pas un autre format, compte rendu au format pdf, rendu code source, jar ...	1
Compte rendu	9
Qualité des mockups Balsamiq : pdf avec liens pour exécuter au moins deux séquences d'interaction	2
Ergonomie de l'interface. L'interface permet de réaliser les tâches demandées tout en répondant aux critères ergonomiques. Les utilisateurs finaux seraient a priori capables d'utiliser l'interface.	4
Utilisation correcte des gestionnaires de placement : l'interface est bien agencée avec des choix pertinents de gestionnaires de placement. Comportement correct des widgets au redimensionnement de la fenêtre. Les widgets sont alignés d'une manière pertinente.	1,5
Gestion correcte des événements : réponse correcte de l'interface aux événements. Utilisation correcte des event handlers et listeners. Utilisation de widgets plus avancées qu'uniquement des boutons, labels...	1,5
Qualité du code : lisibilité du code, commentaires...	1
Qualité de la vidéo de présentation du projet et conformité aux instructions	2

TABLE 1 – Barème indicatif de notation

2. <https://git-iut.univ-lille.fr>

3. https://univlillefr-my.sharepoint.com/:w:/g/personal/gery_casiez_univ-lille_fr/EQVqOWiyupHsyn-3V6YspgBqvLGRiG0QTancy1JjRxiEw?e=stUeIE

4. <https://git-iut.univ-lille.fr/gery.casiez/videomaker>