

GL40 - Interface homme/machine et perception
Université de Technologie de Belfort-Montbéliard

Application des méthodes GOMS et Keystroke à la navigation vocale

Paul LOCATELLI et Pierre ROGNON



Printemps 2013

Sommaire

Introduction	3
1 Analyse du projet	4
2 L'outil de reconnaissance vocale	7
3 L'interface logicielle	9
4 Les tests d'efficacité	15
5 Bilan	19
Conclusion	20

Introduction

Lors de ce semestre, des méthodes de calcul du temps nécessaire pour réaliser une tâche sur un ordinateur ont été étudiées. Ces calculs mettent en avant une comparaison afin de savoir dans quels cas un moyen est plus intéressant comparé à un autre. Cependant, lors de ces études, seuls deux moyens ont été étudiés : le clavier et la souris. Une troisième solution éventuelle était la combinaison des deux.

Cependant, si ces deux moyens sont généralement suffisant pour la vie de tous les jours afin de piloter un ordinateur de façon aisée, il convient de penser aux personnes pouvant éprouver des difficultés à l'utilisation de ces périphériques. Afin de palier à d'éventuels problèmes de mobilité des mains, qui peuvent empêcher d'utiliser de manière efficace la souris et le clavier, une solution peut-être la reconnaissance vocale afin de piloter un navigateur de fichiers par exemple.

Le but de ce projet est donc de mettre en place des tests et l'interface nécessaire pour les réaliser afin de mesurer l'efficacité de cette troisième méthode. Ce rapport va donc détailler tout d'abord la composante de recherche vocale, puis se concentrer sur le navigateur et donc l'interface construite dans lequel les tests se dérouleront. Enfin, les fonctionnalités des tests en eux-même seront abordés ainsi qu'une synthèse sur l'efficacité de la méthode.

1 Analyse du projet

Afin de mieux formaliser les étapes qui doivent être suivies tout au long du projet, il a été choisi de suivre ce qui a été appris lors de ce semestre en GL40. La première étape consiste donc à lister les différentes entités attachées au sujet. Un modèle ARCH/Amodeus sera mis en place afin de mieux délimiter les buts et tâches à effectuer.

1.1 Liste des entités

1.1.1 Liste des postes de travail

Un seul poste de travail est requis pour l'utilisation de l'application : le poste de l'utilisateur de celle-ci. Ce poste peut donc être n'importe quel ordinateur ou machine capable de faire fonctionner le projet.

1.1.2 Liste des buts et sous-buts

Il n'y a qu'un but principal à cette application mais beaucoup d'autres en découlent. Ce principal but est *le test de la méthode de navigation par la voix*.

En découlent donc les sous-buts suivants :

- naviguer dans une arborescence de fichiers ;
- éditer un fichier texte ;
- lancer un test ou l'arrêter ;
- consulter les résultats du test effectué.

1.1.3 Liste des tâches et sous-tâches

Pour pouvoir réaliser les buts et sous-buts énoncés plus haut, des tâches sont requises. Les tâches principales sont :

- l'analyse des sons entendus ;
- la recherche de l'arborescence à afficher ;
- mesurer le temps lors d'un test.

Ces tâches sont aussi composées de sous-tâches :

- pour l'analyse des sons entendus :

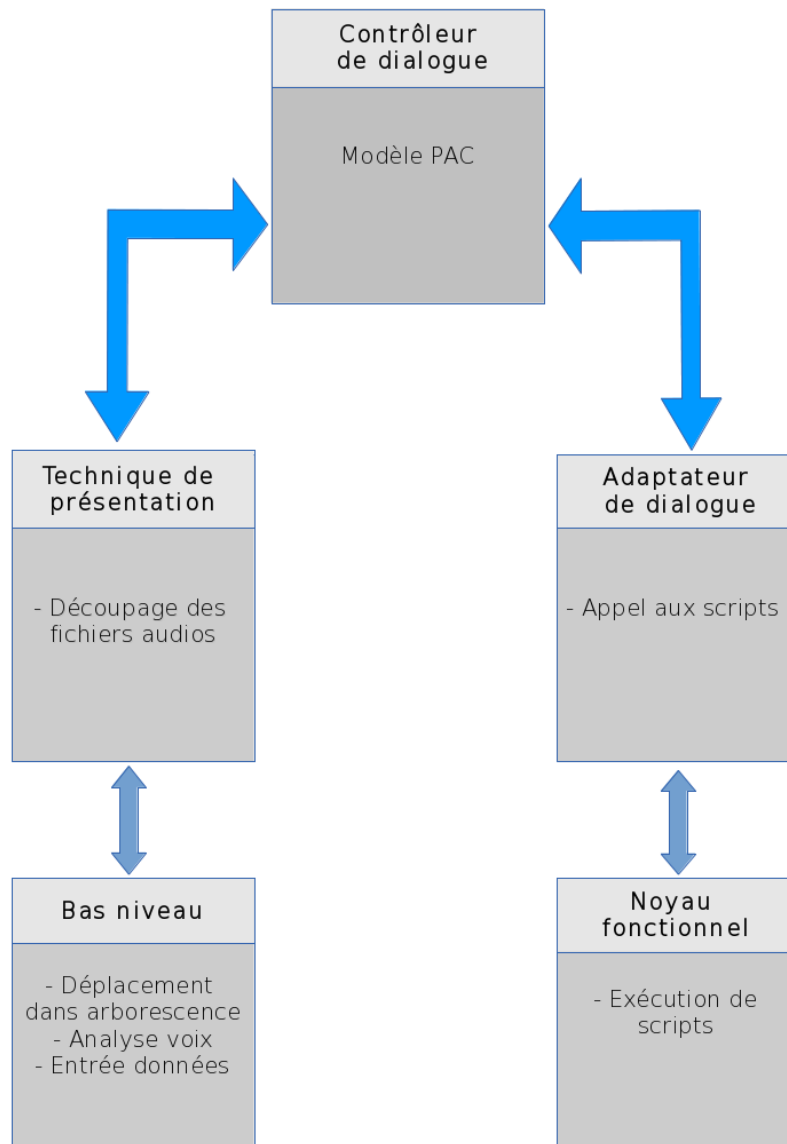
- enregistrer le son ;
- analyser le son ;
- retourner le texte analysé.
- pour la recherche de l’arborescence à afficher :
 - parcourir le dossier en cours ;
 - retourner tous les fichiers et dossiers présents.
- pour la mesure des temps lors d’un test :
 - démarrer un timer lors du lancement du test ;
 - reconnaître la fin d’un test et arrêter le timer ;
 - calculer les temps attendus avec une autre méthode pour informer l’utilisateur.

1.1.4 Liste des données

De nombreuses données sont nécessaires au fonctionnement du logiciel :

- la position actuelle dans l’arborescence du navigateur ;
- la position précédente dans l’arborescence du navigateur ;
- les phrases analysées ;
- le temps en cours si un test est en cours ;
- le nom du fichier à atteindre lors d’un test ;
- le dossier dans lequel se situe le fichier à atteindre ;
- le document ouvert si on est en mode éditeur ;

1.2 ARCH/Amodeus



Ci-dessus, le modèle ARCH/Amodeus du projet

2 L'outil de reconnaissance vocale

Le développement d'un outil de reconnaissance vocale prenant énormément de temps, de ressources et demandant des connaissances solides dans le domaine, il aurait été impossible d'en développer un au cours du projet. C'est pourquoi une recherche sur les modules de reconnaissance vocale a dû être effectuée. Ces logiciels n'étant pas très répandus et souvent propriétaires et payants, le choix n'a pas été aisé. Cependant, un module s'est détaché de tous les autres : Google Speech. Ce dernier est disponible dans plusieurs langues et sur de nombreuses plate forme ce qui en fait un outil flexible. Sa puissance réside surtout dans le fait que c'est un module exécuté sur des serveurs distants de la marque, qui sont efficaces.

Cette puissance est donc aussi un inconvénient puisqu'elle implique une connexion effective avec les serveurs de Google lors de son utilisation. Le logiciel naissant de ce projet nécessitera donc un accès à internet permanent. N'ayant pas d'autre alternative crédible à la solution de Google, du fait d'une absence d'API chez les principaux concurrents, nous avons choisi Google Speech. Les modules de Windows ou d'Ubuntu par exemple sont en effet très efficaces, mais pas utilisables par des logiciels tiers. L'utilisation de Google Speech peut s'effectuer par l'intermédiaire d'un terminal (sous Linux ou Mac). Il n'y a pas de commande pour ce module à proprement parler mais Google Speech demande l'envoi du fichier audio où la reconnaissance vocale doit être appliquée. Une simple requête HTTP suffit donc pour l'utilisation du service. La solution la plus simple ici consistait donc à envoyer par l'intermédiaire d'un terminal le fichier audio enregistré à décoder. Le serveur Google Speech va se contenter alors de renvoyer le texte déchiffré.

Afin d'utiliser au mieux le service de Google, il a été choisi de segmenter un enregistrement permanent en fichiers audio de courte durée. Après quelques essais, il s'est avéré que 3 secondes étaient largement suffisantes pour les enregistrements puissent réunir une commande vocale entière sans qu'une partie de la commande soit sur un premier fichier et l'autre sur un second. Ce problème fait en effet échouer la reconnaissance puisqu'elle n'est alors pas pertinente.

Le fait de devoir envoyer en permanence des informations afin de garder un enregistrement audio continu implique l'utilisation dans le projet de threads.

Les threads permettent d'envoyer alternativement un fichier audio au serveur de Google et de continuer à enregistrer dans le même temps, ce qui évite des coupures.

Lors des résultats sur les durées des tests passés dans le logiciel, il faudra donc prendre en compte le fait qu'une latence peut apparaître due à l'envoi des données au serveur Google, l'analyse de celles-ci et le retour vers la machine cliente.

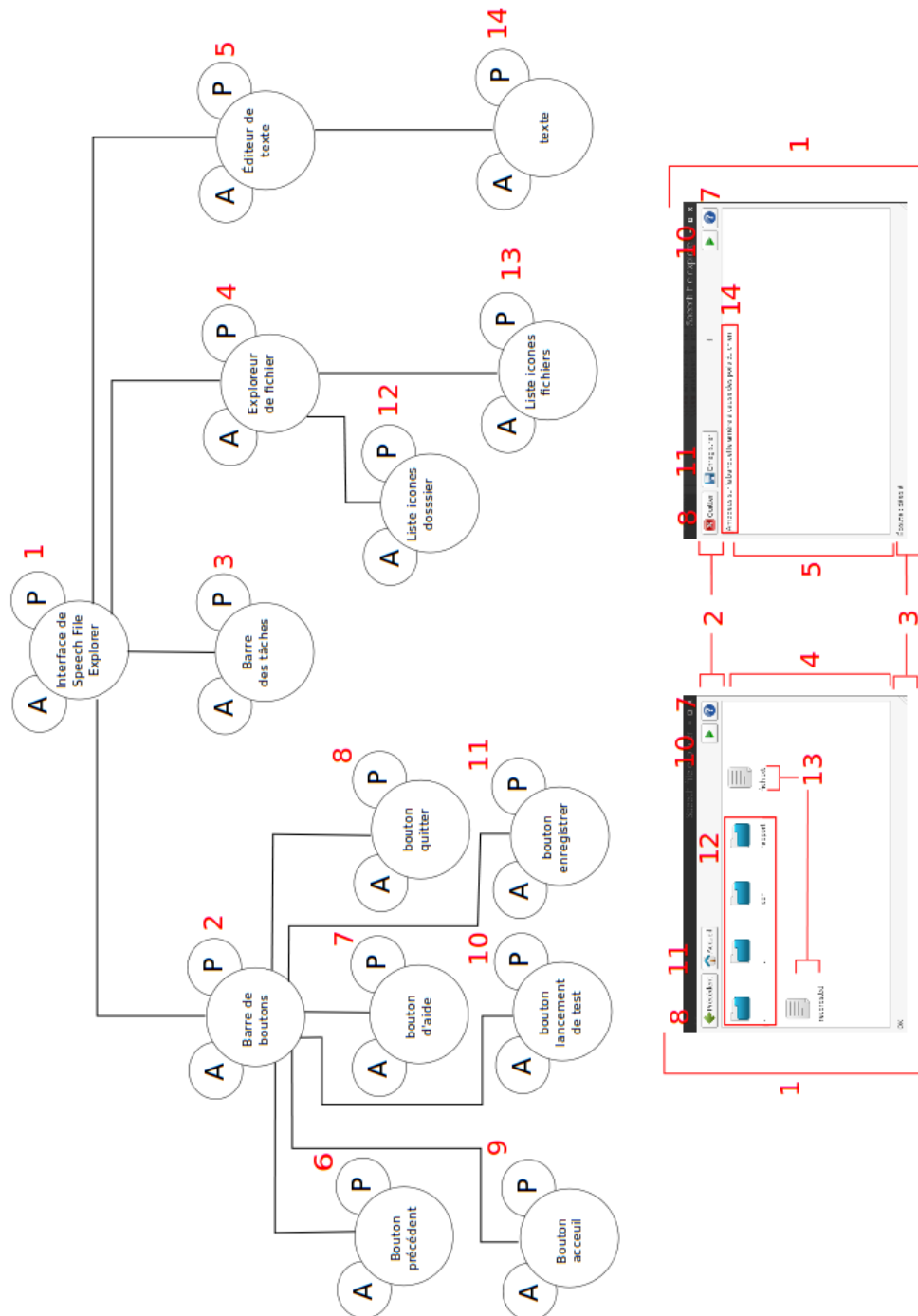
3 L'interface logicielle

Afin de pouvoir réaliser des tests de calcul de la performance de la navigation vocale, il est nécessaire de mettre en place un support permettant de les mettre en pratiques. Ces actions pourront être chronométrées ou non pour laisser à l'utilisateur le choix d'utiliser le logiciel dans sa fonction secondaire utilisée pour le support. Cette fonction est un navigateur de fichiers avec visualisation de fichiers texte intégré. Ainsi, si l'utilisateur le souhaite, il peut naviguer dans ses fichiers et utiliser le logiciel comme un navigateur habituel. La différence ici est que le navigateur ne permet que de se déplacer dans l'arborescence de fichiers et d'ouvrir les fichiers dans l'éditeur intégré. L'utilisateur ne pourra donc pas ouvrir un fichier dans un autre logiciel. Malgré ceci, le but premier du projet étant de permettre la mise en place de tests chronométrés, il faut donc bien garder en tête que le navigateur a pour fonction première d'être un support aux tests.

La matière étudiée pour ce projet se focalisant sur l'interface homme/machine, il est important d'attacher de l'importance à la composition de l'interface afin quelle soit utilisable aisément. C'est pourquoi il a été choisi de réaliser une interface simple, avec peu de boutons et répartis de manière astucieuse.

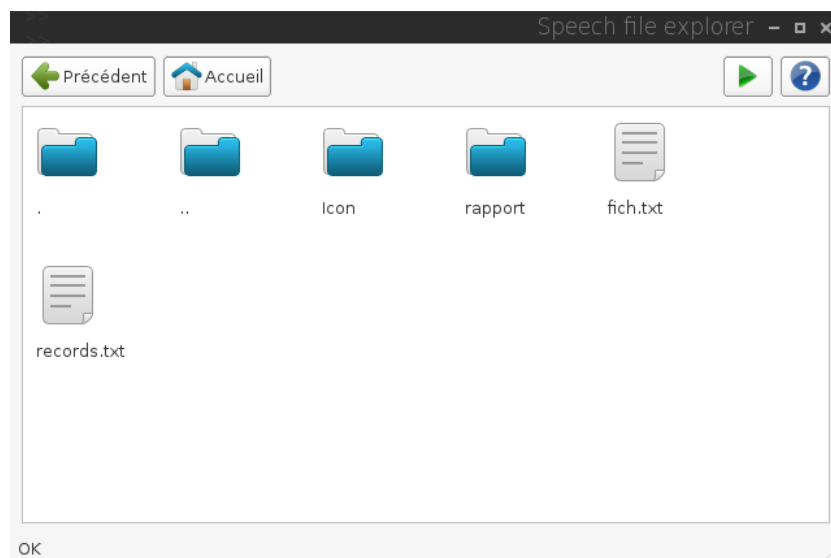
Pour pouvoir mettre en place plus facilement cette interface, un modèle PAC a été utilisé. Aussi, deux modes sont ressortis et coexistent donc dans ce logiciel : un mode de navigation qui permet de se déplacer dans une arborescence de fichiers et un mode d'édition, réservée aux fichiers texte puisque c'est le seul utile aux tests, qui permet de lire, sélection, écrire et sauvegarder un fichier texte.

3.1 Le modèle PAC appliqué au projet



3.2 Le mode navigation

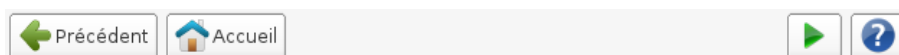
Le mode navigation est constitué de trois parties : la partie navigation "pure", la partie "boutons" et la partie "retour" de la reconnaissance vocale.



Ici, on peut voir l'interface lors du lancement du logiciel. Il ressemble à un navigateur de fichier standard simplifié.

Comme dans tout navigateur de fichier, la plus grande partie de la fenêtre doit être affectée à la navigation de fichier. C'est pourquoi cette partie est insérée au centre afin d'attirer immédiatement l'attention de l'utilisateur. Les autres éléments doivent se faire suffisamment discrets pour ne pas perturber la concentration sur la navigation. Cependant, ils doivent être identifiables facilement et atteignable facilement.

Les boutons ont donc été regroupés dans la partie haute de la fenêtre et divisés en deux parties : la partie aide à la navigation classique, la deuxième en lien avec notre but du projet, la navigation vocale.



La barre de boutons : à gauche les boutons d'aide à la navigation classique (accueil et précédent), à droite les outils pour la navigation vocale et les tests (lancer/arrêter le test, aide à la commande vocale).

Le fait de diviser en deux les boutons permet de bien identifier les différentes fonctions du logiciel. La partie gauche existe sur la grande majorité des navigateurs de fichiers et sont placés au même endroit, l'utilisateur ne doit donc pas être perdu et l'ont ne cherche pas à bousculer cette règle.

A droite, on peut se permettre de placer des fonctions spécifiques à notre logiciel puisqu'en lien avec le vocal.

Le bouton "play", facilement identifiable de par son icône, lance un test. L'icône devient un carré rouge ensuite pour indiquer l'arrêt possible du test mais surtout qu'un test est en cours.

Le bouton "aide", reconnaissable lui aussi permet d'ouvrir une fenêtre (voir ci-dessous) qui apporte une aide non négligeable pour un utilisateur non habitué du logiciel. On y indique les mots-clés reconnus par le logiciel pour la reconnaissance vocale. L'utilisateur peut s'y référer afin de mémoriser ces mots-clés.

La partie du milieu de ce bandeau est laissée vide en temps normal, mais lors du pilotage à la voix, une interaction avec l'utilisateur est prévue ; essayez donc de remercier le logiciel !



La fenêtre d'aide permet d'indiquer les mots-clés à utiliser pour la navigation par la voix à l'utilisateur.

Le navigateur contient une troisième partie, tout en bas de la fenêtre, sur toute la longueur.

Cette partie, très discrète puisqu'utile uniquement dans certains cas précis,

a pour but d'indiquer ce que le logiciel reconnaît durant la reconnaissance. On peut donc y voir la dernière interprétation renvoyée par Google Speech, qu'elle ait été interprétée correctement ou non.

3.3 Le mode édition

Ce second mode, nommé "édition", ne peut apparaître en démarrant le logiciel puisqu'il affiche le contenu d'un fichier. Ce mode est donc activé lorsque l'utilisateur, dans l'arborescence du navigateur, choisit non pas un dossier mais un fichier texte.

Ici aussi, un intérêt a été porté au niveau de la simplicité de l'interface. Ainsi, pour ne pas dérouter l'utilisateur, la disposition reste la même : ni la barre de boutons, ni la partie de retour sur la reconnaissance ne changent de proportions. Et logiquement, l'édition étant le sujet principal de ce mode, il a été choisi de donner la plus grande partie de la fenêtre à l'espace édition et de centrer celui-ci. Ayant le même but pour la navigation, il a été choisi de simplement placer la surface d'édition à l'emplacement où se trouve la partie navigation dans le mode du même nom. Ci-dessous, une capture du mode d'édition.



Le mode édition est composé d'une partie édition prenant la majeure partie de la place, d'une partie boutons et d'une partie "retour". La proximité avec le mode navigation donne une cohérence au logiciel.

La partie d'édition est très simple puisqu'il s'agit simplement d'une "boîte" où écrire du texte.

Pour le bandeau de boutons, la partie de gauche évolue puisqu'elle est dédiée à la fonction principale actuelle qui est ici l'édition de fichiers et non plus la navigation. Les boutons donnent donc ici la possibilité de quitter le mode

mais aussi de sauvegarder le fichier actuel pour ne pas perdre d'éventuelles modifications récentes. Des icônes ont été intégrées pour identifier plus facilement les actions en accord avec l'ensemble des boutons du logiciel. La partie inférieure de la fenêtre n'évolue pas par rapport au mode de navigation puisque la reconnaissance vocale a toujours lieu quelque soit le mode du logiciel.

4 Les tests d'efficacité

La partie la plus importante de ce projet puisque représentant son but premier est la partie de tests de la reconnaissance vocale. Malgré le fait qu'elle est l'objectif principal, il faut pour une utilisation la moins dérangeante possible que ces tests restent discrets. C'est pourquoi cette partie test est accessible à partir d'un simple bouton. Ce bouton à la façon d'un play/pause permet de lancer un test et de savoir lorsque celui-ci est terminé ou bien de l'arrêter.

Deux tests ont été développés, en écho à ce qui a été étudié lors du semestre dans la matière : un premier test dans le navigateur qui a pour but de se rendre dans un fichier le plus rapidement. Le second test, dans l'éditeur, a pour but d'atteindre un certain endroit dans le fichier à partir d'un autre.

4.1 Le test du navigateur

Le test du navigateur a pour but de mesurer le temps que met l'utilisateur pour atteindre un fichier cible. Pour cela, il faut que le logiciel puisse enregistrer la cible à atteindre.

Lorsque l'utilisateur veut lancer un test de navigateur, il faut donc qu'il clique sur le bouton de lancement de test. Cependant, pour que le test soit identifié, il faut en toute logique qu'il se situe en mode navigateur. C'est le fait qu'il soit en mode navigateur qui lance le bon test. Une fenêtre s'ouvre alors afin de recueillir les informations nécessaires (voir ci-dessous la fenêtre).



La fenêtre de test ci-dessus s'ouvre lorsqu'un test de navigateur est lancé.

La fenêtre demande de remplir le chemin ainsi que le nom du fichier. Le chemin est simplement le dossier où le fichier est contenu. Cela permet d'éviter d'éventuelles erreurs de manipulation si l'on part d'un dossier où le même nom de fichier existant. Comme on ne vise pas ce dernier, il ne faut pas que le test s'arrête si l'on sélectionne ce fichier.

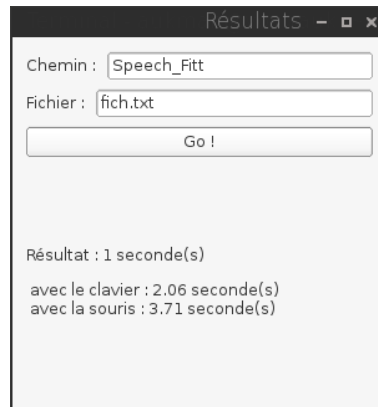
Une fois les informations nécessaires renseignées, l'utilisateur lance le test. Ce test fait démarrer un chronomètre et on décompte donc le temps. Le lancement du chronomètre est visible par l'icône du bouton de test qui change (●).

La fenêtre reste cependant ouverte pour plusieurs raisons. Tout d'abord, cela permet de rappeler à l'utilisateur qu'un test est en cours en plus du changement d'icône dans le bouton de test. Cela permet aussi de pouvoir afficher une fois le test terminé les résultats. Afin d'indiquer à l'utilisateur qu'il n'a plus à entrer d'informations, on grise les champs comme le montre la capture ci-dessous.



Une fois le test lancé, la fenêtre reste ouverte mais les champs sont grisés.

Lorsque l'utilisateur arrive au fichier qui été visé, le test s'arrête automatiquement et les résultats sont instantanément affichée dans la fenêtre de test. Les résultats retournés sont affichés de la manière suivante :



Lorsque le test est terminé, on affiche les résultats dans l'encart prévu à cet effet.

Le résultat obtenu lors du test effectif est comparé à des résultats calculés de manière théorique avec Keystroke. Cela permet à l'utilisateur de voir directement sa navigation a été plus efficace que le temps théorique. Cependant, si le but de ce test est de mesurer le temps mis par une navigation vocale, l'utilisateur devra s'imposer une discipline puisque le logiciel ne fait pas la différence entre clavier, souris ou voix. S'il veut réellement mesurer le temps de la navigation vocale, il devra donc lors du test ne pas "tricher". L'avantage est que la mesure au clavier ou à la souris sont possibles aussi. On peut alors comparer les temps réel mis par rapport au temps théorique.

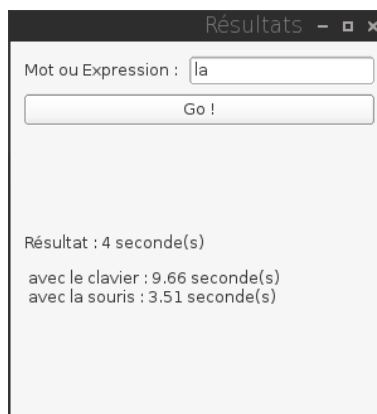
4.2 Le test de l'éditeur

Le second test, relatif au mode éditeur, doit mesurer le temps que met un utilisateur à sélectionner un mot ou une expression dans un fichier texte. De la même manière que le test du navigateur, l'utilisateur doit cliquer sur le bouton du lancement d'un test. Il faut alors qu'il soit en mode éditeur pour que le logiciel le reconnaisse que c'est ce test qu'il doit lancer. La fenêtre ci-dessous s'ouvre alors afin de permettre à l'utilisateur de rentrer ses informations.



L'utilisateur est invité à entrer le mot qu'il vise.

Une fois le mot à atteindre indiqué, l'utilisateur clique sur "Go!". Il peut ensuite atteindre la sélection. Dès que l'utilisateur sélectionne le bon mot, le test s'arrête automatiquement et les résultats sont affichés dans la fenêtre de test comme ci-dessous :



Les résultats sont affichés de la même manière que le test du navigateur.

Tout comme le test du navigateur, le test est comparé aux résultats qui seraient attendus par les autres méthodes (clavier et souris).

5 Bilan

Le développement de ce logiciel montre que les résultats sont assez inégaux et difficiles à synthétiser. En effet, dès que le cadre devient quelque peu bruyant, ou si la voix n'est pas assez forte, la compréhension par les algorithmes est difficile et il faut s'y reprendre à de nombreuses fois avant d'obtenir l'action souhaitée.

Cependant, avec un peu d'entraînement et dans un environnement calme, la réponse se fait souvent assez rapidement, au premier essai. Cette méthode devient alors intéressante même si elle bat rarement les deux autres.

Il y a un cas où la méthode devient particulièrement efficace : lorsque dans l'arborescence ou dans le fichier, la zone à sélectionner se trouve loin. En effet, dans ce cas, l'utilisateur au clavier ou à la souris se voit obligé de rechercher l'objet voulu avant de pouvoir l'atteindre. On a donc deux étapes à franchir, particulièrement longues. La méthode de la voix devient alors très aisée à utiliser.

Seul bémol : si on a plusieurs fois la même expression dans un fichier ou deux noms de dossier à la prononciation semblable, il devient difficile de sélectionner l'expression ou le fichier voulu.

C'est donc un bilan mitigé pour cette méthode. Mais si l'on prend en compte que la navigation à la voix est destinée à des personnes qui en ont le besoin et ne peuvent pas utiliser le clavier ni la souris, son intérêt grandit fortement.

Conclusion

Ce projet a été fort enrichissant sur différents aspects, premièrement la réalisation d'un projet avec PyQt, un mélange entre le langage objet Python et la bibliothèque Qt. Dans un second temps, l'utilisation de Google speech a été une forte instruction.

Dans les améliorations possibles de ce projet, on peut noter une amélioration de la reconnaissance vocale, dans certains logiciels spécialisés comme Dragon Dictation qui apprend à reconnaître la voix de l'utilisateur. Une seconde amélioration possible est l'élargissement des fichiers acceptés et reconnus par le navigateur afin d'offrir une expérience totale.

La question que l'on peut se poser est si notre futur est dans cette technologie comme on peut le voir dans la plupart des films de science fiction, ou dans une autre voie encore peu exploitée comme la reconnaissance de mouvement ?

Table des matières

Introduction	3
1 Analyse du projet	4
1.1 Liste des entités	4
1.1.1 Liste des postes de travail	4
1.1.2 Liste des buts et sous-buts	4
1.1.3 Liste des tâches et sous-tâches	4
1.1.4 Liste des données	5
1.2 ARCH/Amodeus	6
2 L’outil de reconnaissance vocale	7
3 L’interface logicielle	9
3.1 Le modèle PAC appliqué au projet	10
3.2 Le mode navigation	11
3.3 Le mode édition	13
4 Les tests d’efficacité	15
4.1 Le test du navigateur	15
4.2 Le test de l’éditeur	17
5 Bilan	19
Conclusion	20