

Projet d'interface Homme-Machine : QUELLE EST CETTE NOTE

Thibault BÉZIER LA FOSSE, Dennis BORDET

21 mars 2016



Sommaire

1	Introduction	3
1.1	Motivation et but du projet	3
1.2	Interface générale	3
1.3	Plan	3
2	Différents éléments de l'interface	4
2.1	Portée	4
2.1.1	Variabilité de la portée	4
2.1.2	Clef de sol & Notes	4
2.1.3	Curseur de suivi	4
2.1.4	Correction	5
2.2	Menu	5
2.3	Couleur de fond	6
2.4	Piano	6
2.4.1	Pourquoi le Piano ?	6
2.4.2	Interactions	6
2.4.3	Couleurs	7
2.4.4	Bulles d'information	7
2.5	Raccourcis clavier	7
2.6	Affichage des raccourcis	7
2.7	Affichage du nom des notes	7
2.7.1	Représentation	8
2.8	Annuler la Note	8
2.9	Masquer les options	9
2.9.1	Pollution visuelle	9
2.9.2	Optimisation de l'espace	9
2.9.3	Nom	9
2.9.4	Position	9
2.10	Note de fin de session	9
2.11	Progression	10
2.12	Choix de partitions	10
2.13	Aide pour les débutants	10
2.13.1	Aide note après note	10
2.13.2	Partition modèle	10
2.14	Tutoriel au premier lancement	10
3	Paper Prototype	11
3.1	Lien	11
3.2	Essais et conséquences	11
4	Test du logiciel par des cobayes	11
4.1	Cobaye numéro 1 : Johnna, 6ans1/2, CP	11
4.2	Cobaye numéro 2 :Charline	11

5 Conclusion**12**

1 Introduction

1.1 Motivation et but du projet

Pourquoi apprendre le solfège ? Afin d'éviter des catastrophes auditives comme le thème de titanic par Marc Mulholland.

L'idée est de réaliser un petit logiciel pour initier les débutants au solfège. Il fallait donc au départ restreindre le programme, le domaine de la musique étant assez vaste. Il a donc été décidé collégialement que le programme aurait un niveau assez faible et simple, niveau 6e en musique.

1.2 Interface générale

Il faut donc que l'utilisateur sache lire une partition simple en clef de sol, la partition ne contient que deux gammes.

L'outil choisi pour entrer les notes est un clavier de piano. Le piano étant un instrument universel en musique.

Il a également été décidé de représenter les \sharp et les b-mol, ceux-ci étant présent dans de très nombreuses partitions, il est donc important pour un débutant d'en prendre connaissance.

1.3 Plan

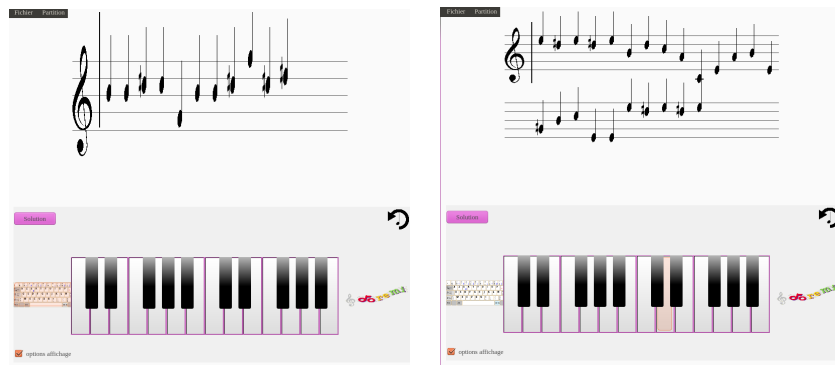
- Explications des choix de l'interface
- Paper prototype et compte rendu des essais
- Test du logiciel par des cobayes

2 Différents éléments de l'interface

2.1 Portée

2.1.1 Variabilité de la portée

Le but de notre application étant d'apprendre à lire une partition, il va de soi que notre logiciel devait contenir une portée de notes. Nous voulions pouvoir efficacement des portées de moins d'une dizaine de lignes, et des portées plus longues. Si bien que notre application adapte son affichage au nombre de notes. Arbitrairement, nous avons fixé le nombre de notes par ligne à 14. Nous estimons que c'est un juste milieu entre une portée trop pleine, et une portée trop vide.



2.1.2 Clef de sol & Notes

Le placement des notes et de la clef de sol dans la portée pouvait se faire de deux manières différentes avec *Qt* : En dessinant les notes, ou en insérant des images. Nous avons tout d'abord choisi d'insérer la clef de Sol au début de la portée, car il aurait été impossible d'en dessiner une d'une manière propre. Ensuite nous avons commencé en dessinant les notes. Enfin nous avons vite changé d'avis, et inséré des images à la place, la variété des notes (Couleur, Dièse, normale, barrées ...) rendait le code illisible. L'insertion d'images nous a paru bien plus évidente, en plus d'être graphiquement plus agréable.

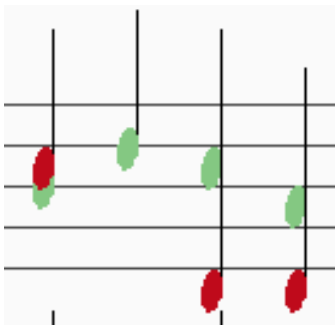
2.1.3 Curseur de suivi

Afin de simplifier la lecture des notes par l'utilisateur, nous avons ajouté un curseur de lecture sur la portée. Il se présente sous la forme d'une barre verticale, de largeur 2px, et se positionne avant la prochaine note à jouer. A chaque note jouée, le curseur avance d'une note, ou bien recule d'une note si l'utilisateur annule sa dernière entrée.



2.1.4 Correction

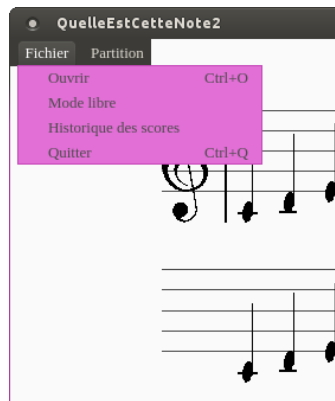
Lorsque l'utilisateur entre la dernière note de la portée, la correction débute automatiquement. Toutes les notes sont redessinées dans un vert léger, et volontairement non-agressif. Ensuite les entrées de l'utilisateur sont analysées. Chaque note est comparée à la partition. Lorsque la note est bien jouée, la note verte est donc simplement présente. Lorsque l'utilisateur s'est trompé, on place une note rouge correspondant à celle qu'il a jouée, et une verte là où il aurait dû jouer. Ainsi il peut comprendre ses erreurs, et éventuellement recommencer en évitant de faire les mêmes erreurs.



2.2 Menu

Afin de naviguer dans notre application nous avons implémenté une barre d'outils. Cette barre propose 2 onglets, le premier, "Fichier", contient :

- Ouvrir un fichier, ou bien CTRL+O pour les utilisateurs de raccourcis, ouvre un fichier contenant une partition et l'affiche sur la portée. L'utilisateur peut ensuite la jouer.
- Jouer en mode libre, qui permet de n'utiliser aucune partition, et de jouer librement du piano. Pour composer, par exemple.
- Historique des scores, afin d'afficher sa progression, notamment au niveau des temps de réalisation, ainsi que des pourcentages de réussite.
- Quitter, ou CTRL+Q, afin de fermer l'application.



Néanmoins suite à un problème d'implémentation nous avons fait un second onglet, "Partition", qui permet de choisir la partition dans un menu déroulant. Cependant, l'ouverture de partition étant constamment répétée, nous avons rapidement remarqué qu'un onglet de la barre d'outil destiné à cet effet était extrêmement plaisant.

Finalement nous avons même ajouté un bouton $+$ en haut à gauche, accélérant d'autant plus l'accès au choix de partition.

2.3 Couleur de fond

Nous avons choisit de mettre le fond en blanc, le blanc rendant la portée plus visible. Le l'arrière plan du piano est quant à lui légèrement grisé pour faire une petite séparation entre la partition et le "groupe utilisateur" (piano + boutons).

2.4 Piano

2.4.1 Pourquoi le Piano ?

Nous avons choisit de représenter uniquement un piano comme instrument, le métallophone étant très similaire, nous avons jugé inutile d'en faire un. Pour d'autre instruments tel que le trombone, les notes auraient été plus difficiles à représenter, moins parlantes. Le logiciel doit rester simple et pour débutants.

2.4.2 Interactions

Quand l'utilisateur appuie sur une touche du piano (avec la souris ou le clavier), le programme lui confirme son action par une réponse visuelle et auditive.

En effet, la note est jouée et la touche du piano est enfoncée. De plus si l'utilisateur était en plein test sur une partition, le curseur avance à chaque note tapée.

2.4.3 Couleurs

Les couleurs sont bien évidemment noires et blanches comme un piano normal. Nous avons décidé de donner un effet de lumière sur la partie haute du piano, les touches noires étant donc blanchies aux sommets. Ceci est un choix purement esthétique et apporte un effet plus réaliste au piano.

2.4.4 Bulles d'information

Pour chaque touche du piano, si l'utilisateur peut afficher la bulle d'information (tooltip), qui lui indiquera le nom de la note et son raccourci clavier. Eh oui, il y a des raccourcis clavier !

2.5 Raccourcis clavier

Le but du logiciel est d'apprendre les bases du solfège, mais de les apprendre le plus professionnellement possible. Il ne s'agit donc pas de savoir lire des partitions simple mais de les lire rapidement ! Et pour savoir si l'utilisateur lit une partitions rapidement, il faut qu'il ait pu la jouer rapidement. Jouer une partition à la souris étant particulièrement sportif et imprécis lorsque l'on joue rapidement, il est nécessaire de pouvoir la jouer avec le clavier, comme un musicien aurait fait avec un vrai piano.

2.6 Affichage des raccourcis

Si chaque touche du piano correspond à une touche du clavier, il serait judicieux d'indiquer à l'utilisateur quelles sont les touches qui sont associées. Et ceci sans avoir à afficher les bulles d'informations les unes après les autres. Nous avons donc décidé d'implémenter un moyen d'afficher sur chacune des notes, son raccourci clavier. Cette fonction est lancé par un bouton situé à proximité du piano, ce qui est évident puisque la fonction agit sur le piano par principe de proximité.

Ce bouton est représenté par un clavier, ce qui est assez parlant pour un utilisateur qui aurait envie d'utiliser son clavier pour jouer du piano.



Bien sûr, au cas où, une bulle d'information indique ce que fait ce bouton si un utilisateur moins futé se posait la question.

2.7 Affichage du nom des notes

De façon similaire, nous avons décidé d'implémenter une fonction indiquant la note de chaque touche. Le bouton qui la lance est lui aussi proche du piano. Par soucis de symétrie (et donc de confort visuel des utilisateurs), il est situé de l'autre coté de son semblable.

Une question se pose alors : est-il possible d'afficher les raccourcis et les notes en même temps. Oui, pourquoi pas après tout, si l'utilisateur en a envie, il ne faut pas l'en empêcher. Et ceci est tout à fait justifié si l'utilisateur débute avec le logiciel et le piano.

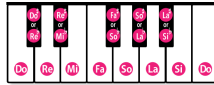
Cependant, nous n'avons pas pu l'implémenter sur notre programme car nous n'avons pas réussi à tout faire tenir sur la largeur de la touche tout en étant lisible, ni à écrire sur plusieurs lignes dans la touche. Le temps qui nous était imparti étant assez court.

2.7.1 Représentation

À quoi doit ressembler ce bouton : un petit dessin de clé de sol, avec les notes "do re mi" écrites est assez communicant puisque l'utilisateur peut voir le nom des trois premières notes sur ce bouton. Bien sûr si ce n'est pas assez évident, le bulle d'information est là. Cependant, au cours du test sur le cobaye numéro 1, celui-ci s'attendait à ce que le PC lui joue les notes do, ré et mi.



Nous avons donc décidé de changer la représentation de ce bouton, un professeur compétent en IHM nous a conseillé de mettre un aperçu du piano avec les notes écrites dessus. Ceci indique alors exactement l'action du bouton.



2.8 Annuler la Note

Parfois l'utilisateur peut se tromper malencontreusement de note et souhaiter revenir en arrière puisqu'il s'est rendu compte de son erreur, plutôt que de relancer la partition, une touche pour revenir à la note précédente serait très pratique. Un genre "d'undo" en quelques sortes. Pour représenter ce bouton, nous avons initialement choisi du texte (cf paper prototype) : annuler la note". Cependant, l'interface devenait beaucoup trop textuelle, une icône aurait été beaucoup plus agréable à l'œil. Nous avons donc eu l'idée de reprendre la représentation culturelle de la commande undo (une flèche courbée qui revient en arrière) et d'y ajouter une note de musique à l'intérieur pour bien comprendre le sens de l'action.



2.9 Masquer les options

2.9.1 Pollution visuelle

Certains utilisateurs, sûrement une majorité, une fois qu'ils seront familiers avec les notes et les raccourcis, ne voudront plus être embêtés par ces deux boutons qui ne leurs sont plus d'aucune utilité, nous avons donc décidé de pouvoir les cacher en décochant une option.

Au cours d'une seconde réflexion, nous nous sommes demandé si les fonctions d'annulation de note et de solution devaient aussi être masqués, après débat musclé nous avons décidé de les cacher afin que l'utilisateur un peu plus expérimenté aie le moins de pollution visuelle possible.

2.9.2 Optimisation de l'espace

Ces deux boutons disparus, il reste de la place inoccupée de chaque coté du piano, il est donc judicieux d'augmenter la taille du piano en conséquence, ce qui donne un meilleur confort visuel.

2.9.3 Nom

Cette fonction permettant d'améliorer l'espace de saisie des notes doit avoir un nom, pour indiquer à l'utilisateur ce qu'il va se passer. Le terme "options d'affichage" est plutôt parlant et à proximité du piano, il est probable que les options concernent le piano. Le terme "options" lui est plus court mais il est moins précis. Quelles sont ces options ? Des touches supplémentaires ?

2.9.4 Position

Au début de notre implémentation (cf paper prototype), cette checkbox était situé juste en haut du piano, avec les boutons solution, annuler la note et correction. Ces 3 derniers boutons avait un rapport avec la partition, contrairement au checkbox, nous avons donc décidé de la descendre sous le piano, pour la séparer de la partition. Ainsi L'utilisateur n'aura pas la sensation que son action agit sur celle-ci.

De plus, avec la suppression du bouton de correction, la checkbox cassait la symétrie axiale en haut du piano.

2.10 Note de fin de session

Le logiciel ayant un but éducatif, il faut donc évaluer l'utilisateur afin de déterminer son niveau de connaissance. Il y a deux critères pour l'évaluer :

- Le nombre de bonnes réponses (en pourcentage)
- La vitesse à laquelle il a pu lire la partition

À la fin de chaque test (=partition), le récapitulatif de ces deux points est affiché accompagné d'une appréciation. En plus, le personnage accompagnateur émet une petite remarque ayant pour but d'aider l'utilisateur à s'améliorer.

Par exemple, si l'utilisateur met plus de temps à trouver les notes en “#” ou “b-mol”, le personnage lui conseillera de les réviser. Pour implémenter ceci, nous nous servons de différents fichiers de log.

De plus, pour créer une redondance dans le résultat de la session, nous ajoutons des couleurs correspondantes au score. Exemple : vert pour les bons scores, orange pour les moyens et rouge pour les mauvais.

2.11 Progression

En plus de la note de fin de session, les utilisateurs peuvent être intéressés par la progression de leurs notes et de leurs vitesse de jeu. Il est donc intéressant de stocker un historique des scores, mais cette historique n'est pas consulté régulièrement, il ne faut donc pas que cette commande soit accessible directement sur l'écran principal et pollue l'espace.

2.12 Choix de partitions

Pour juger efficacement de l'évolution de l'utilisateur, il faut qu'il ait accès à un grand nombre de partitions, il ne faut donc pas les enregistrer “en dur” dans le programme, mais pouvoir les charger à partir d'un fichier ou autre. Cette commande étant utilisée souvent (enfin normalement), on peut la représenté par un petit “+” discret en haut a gauche de la partition.

Bien sûr, cette commande est également présente dans le menu “fichier->ouvrir” pour les habitués aux logiciels de bureau. Il serait intéressant de tenir à jour un site qui répertorie de nombreuses partitions simples.

2.13 Aide pour les débutants

2.13.1 Aide note après note

Et pour les tout débutants en solfège? Qui n'ont eu aucun cours sur le solfège? Il faudrait pouvoir les aider lorsqu'ils sont vraiment bloqués dans l'ignorance. De ce fait, l'ajout d'un bouton qui donnerait indiquerait sur quelle touche appuyer. Appeler cette touche “solution” est assez signifiant. “Aide débutant” étant plus vague et plus long.

2.13.2 Partition modèle

Il est aussi plus simple, pour quelqu'un qui n'a pas suivi un seul cours de solfège, d'avoir la possibilité d'avoir sous les yeux une partitions avec toutes les notes des deux gammes du logiciel et leurs nom correspondant juste en dessous.

2.14 Tutoriel au premier lancement

Afin que l'utilisateur prenne parfaitement en main le logiciel, surtout les plus jeunes. Nous avons décidé (suite à un test sur cobaye n1), de lancer un petit tutoriel lors du premier démarrage. Ce tutoriel serait dirigé par un personnage

professeur accompagnant, pas trop humain, un robot ou une note de musique vivante, c'est en tout cas ce qui ressort dans nos sondages. Il expliquerait rapidement le but du programme, comment l'utiliser et comment avoir de l'aide en cas de besoin. Ce serait aussi ce personnage qui donnerait les conseils et remarque à la fin de chaque test.

3 Paper Prototype

3.1 Lien

Paper Prototype, 1er jet.

3.2 Essais et conséquences

Suite à un essai du paper prototype par un des cobaye, ce dernier nous a fait remarquer que le bouton de correction (présent dans la vidéo) est inutile puisque la correction est immédiatement lancée automatiquement lorsque le musicien en herbe joue sa dernière note. Et quel "ahuri" aurait l'idée de se faire évaluer sans avoir joué toutes ses notes ! ? En effet, ce bouton semble bien inutile, nous avons donc décidé de le retirer.

4 Test du logiciel par des cobayes

4.1 Cobaye numéro 1 : Johnna, 6ans1/2, CP

- À l'époque le logiciel n'avait pas de tutoriel. Le cobaye n'avait pas vraiment d'idée de ce qui allait se passer en cochant les options d'affichages. Mais une fois cliqué, il comprit.
- Bouton "do re mi" : le cobaye pensait que ce bouton allait lui jouer les notes do, ré et mi.
- Le bouton permettant d'afficher les raccourcis clavier et celui pour donner la solution font parfaitement ce à quoi s'attendait le cobaye.
- Partition : le cobaye ne comprend pas vraiment la partition, l'intérêt de celle-ci

C'est ici que l'idée du tutoriel nous vient, il permettrait donc à se cobaye de savoir ce que font les boutons dont il ne savait pas l'action et de le guider sur les partitions.

4.2 Cobaye numéro 2 :Charline

Curseur trop petit. Changement de taille du piano perturbant lors de l'affichage/masquage des options d'affichage.

5 Conclusion

Pour conclure ce rapport, ce fût vraiment un projet passionnant. Avant de commencer nous ne savions absolument rien faire avec Qt, hormis suivre les tutoriels, mais après avoir développé *Quelle est cette note*, nous savons les bases pour écrire des applications complexes. C'est désormais un plus non négligeable à nos compétences.

La conception de l'IHM de notre application nous a permis de réfléchir aux points vus en cours. Chaque partie fût l'objet d'une réflexion : Pourquoi ? Où ? Comment ? Afin d'obtenir un rendu final qui nous plaît. Même si notre application ne correspond pas tout à fait à notre *Paper Prototype*, nous avons essayé de le suivre le plus fidèlement possible.

Néanmoins, il reste toujours de nombreuses améliorations possibles à notre application : Amélioration des logs, une possible traduction des notes afin de s'ouvrir à l'internationale, avec un clavier compatible Qwerty. On pourrait aussi s'étendre sur plus de deux octaves, et même avec des clefs différentes. On pourrait aussi imaginer un outil de composition, écrivant sur la portée les notes entrées, au lieu de les lire.

Enfin, on pourrait aussi trouver un nom mieux que *Quelle est cette note*.