

Hervo Pierre-Yves

Rapport de Stage de 4ème année.

Projet de création d’algorithme génétique

Pour la synthèse vocale

Année universitaire 2012-2013

Département informatique

Spécialité Informatique Décisionnelle

Polytech Nantes





Remerciements

Je tiens à remercier toute l’équipe du centre de recherche qui m’as si gentiment accueillie et qui s’est montré si sympathique tout au long de mon stage.

Je remercie plus particulièrement le professeur Eduardo Miranda, mon maitre de stage qui m’a fourni un sujet très intéressant sur lequel travailler et qui s’est montré très disponible et très à l’écoute tout au long du projet.

Je tiens également à remercier le Dr Duncan Williams qui a toujours pris le temps de m’aider et de répondre à mes nombreuses questions lorsque la barrière culturelle se faisait sentir.

Introduction

Ce rapport constitue le compte rendu de mes quatre mois de stage dans le laboratoire de recherche ICCMR de l'université de Plymouth sous la direction d'Eduardo Miranda durant la période du 03 mai au 21 septembre 2013.

Dans le cadre du cursus d'élève ingénieur en 4ème année à Polytech Nantes, nous devons effectués un stage d’un minimum de deux mois à l’étranger. J'ai choisi pour ma part d’en réaliser un de quatre mois à Plymouth au Royaume-Uni.

Ma principale motivation fut tout d’abord le coté culturel. Cela m’a permis de découvrir une nouvelle culture, une autre façon de vivre. Et puis, quoi de mieux que le Royaume-Uni pour améliorer mon anglais, langue aujourd’hui utilisée quotidiennement pour communiquer à l’international ? Ma seconde motivation fut de travailler dans le domaine de la musique, domaine qui me tient particulièrement à cœur. C’est pour cette raison que j’ai postulé à L’ICCMR, sous la direction du professeur Miranda.

Enfin, ma dernière motivation pour choisir ce stage est m’aura permis d’expérimenter le travail dans le monde de la recherche et de voir en quoi il consistait au quotidien, sujet sur lequel on nous a sensibilisé lors de la quatrième année.

Afin de retracer au mieux mon expérience, je vais commencer par vous présenter le contexte du stage et je vous parlerais ensuite du travail que j’ai réalisé. Une fois cela fait, je vous expliquerais comment j’ai gérer le projet sur le long terme, les difficultés que j’ai rencontré puis je vous exposerais les apports que j’ai retiré de ce stage avant de conclure ce rapport.

Sommaire

[I/ Présentation du contexte 6](#_Toc366596184)

[1. Présentation du cadre de travail 6](#_Toc366596185)

[1.1) L'université de Plymouth 6](#_Toc366596186)

[1.2) Le laboratoire ICCMR 7](#_Toc366596187)

[2. Présentation du projet 8](#_Toc366596188)

[1.3) Praat 8](#_Toc366596189)

[1.4) Projet global 9](#_Toc366596190)

[1.5) Algorithme génétiques 10](#_Toc366596191)

[1.6) Intitulé final 10](#_Toc366596192)

[I/ Présentation de mon travail 10](#_Toc366596193)

[3. Les différentes étapes 11](#_Toc366596194)

[1.7) Bibliographie 11](#_Toc366596195)

[1.8) Interaction avec Praat 11](#_Toc366596196)

[1.9) Algorithme génétique 11](#_Toc366596197)

[1.10) Intégration 12](#_Toc366596198)

[4. Conception 13](#_Toc366596199)

[1.11) Algorithme Génétique générique 13](#_Toc366596200)

[1.12) Ma population 13](#_Toc366596201)

[1.13) Ma cible 14](#_Toc366596202)

[1.14) Récapitulatif de mon architecture 15](#_Toc366596203)

[1.15) Ma fonction fitness et ma condition d’arrêt 15](#_Toc366596204)

[1.16) Mes opérateurs d’évolution 16](#_Toc366596205)

[5. Interface graphique 18](#_Toc366596206)

[II/ Gestion du projet 20](#_Toc366596207)

[6. Gestion générale 20](#_Toc366596208)

[1.17) Système de fonctionnement du projet 20](#_Toc366596209)

[1.18) Documentation 20](#_Toc366596210)

[7. Gestion du code 21](#_Toc366596211)

[1.19) Le code en lui-même 21](#_Toc366596212)

[1.20) Système de gestion de version 21](#_Toc366596213)

[III/ Les difficultés rencontrées 21](#_Toc366596214)

[IV/ Ce que ce stage m’as apporté 22](#_Toc366596215)

[V/ Conclusion 23](#_Toc366596216)

# Présentation du contexte

Dans ce premier point je vais commencer par vous présenter le cadre de travail dans lequel j’ai évolué pendant ces quatre mois puis poursuive en vous présentant mon projet.

## Présentation du cadre de travail

J’ai passé les 4 mois qu’ont duré mon stage dans le laboratoire de recherche de L’ICCMR de l’université de Plymouth. Je vais commencer par présenter ces deux organismes dans ce point.

### L'université de Plymouth

L’université de Plymouth fut fondée en 1862. Cette institution intègre 6 pôles de compétences différents:

* La faculté des arts dans laquelle j’ai réalisé mon stage.
* La faculté d'éducation.
* La faculté de santé et d'action sociale.
* La faculté de science.
* La faculté de sciences sociales et de commerce.
* La faculté de technologie.

Contrairement à l’université de Nantes, tous les bâtiments de la faculté sont rassemblés sur un seul campus qui n’est pas très large mais très concentré avec de nombreux immeubles au design très moderne.



L’université de Plymouth accueille environ 30,000 étudiants, compte 3000 salariés et propose en conséquent de très nombreux services universitaires. Elle génère chaque année un revenu de 160 millions de livres sterlings.

Ses principales missions sont :

* Fournir un enseignement de qualité à tous ses étudiants.
* Favoriser l’insertion de ses étudiants en étant en contact direct avec les entreprises par le biais de projets collaboratifs.
* Favoriser la recherche et l’innovation avec de nombreux laboratoires de recherches dans chaque faculté.
* Agir sur l’activité locale en sensibilisant ses élève, futurs salariés et habitants de demain ainsi qu’en apportant une dynamique économique à la ville de Plymouth par l’activité de tous ses étudiants.
* Sensibiliser les élèves et le personnel au développement durable et au commerce équitable.

### Le laboratoire ICCMR

ICCMR signifie «Interdisciplinary Centre for Computer Music Research ». Ce qui se traduit par centre de recherche interdisciplinaire sur la musique générée par ordinateur.

Comme son nom l’indique, ce centre de recherche comprend de nombreux domaines de spécialités. On trouvera par exemple de la biologie, de l’étude cognitive, de la composition musicale, de la cinématographie, de la robotique, du développement applicatif, etc. Tous ces pôles de compétences divers sont réunis par un point commun : leur application dans le domaine de la musique.

Le laboratoire de recherche ICCMR se situe dans l’université et fait partie de la faculté des arts. Il comprend une vingtaine de personnes : 11 membres permanents et une quinzaine d’étudiants (masters ou doctorants). La direction du centre est assuré par le professeur Eduardo Miranda.

La philosophie du centre est d’être :

* Un pôle de croisement entre la créativité musicale, la science et la technologie
* Le premier acteur en termes de développement de la technologie musicale utilisant les nouvelles technologies de l’informatique et les dernières découvertes musicales.

## Présentation du projet

Dans cette sous-partie, je vais commencer par vous présenter le projet sur lequel j’ai travaillé. Pour permettre au lecteur de comprendre au mieux les intérêts du projet, je l’ai divisé en plusieurs sous-points complémentaires qui permettent au final de pleinement comprendre l’intitulé de mon stage.

### Praat

La première chose à faire est de présenter le logiciel qui se trouve au centre de mon projet : Praat.

Praat est un logiciel open source développé par deux professeurs de l’université d’Amsterdam : Paul Boersma and David Weenink. Il s’agit d’un logiciel d’analyse phonétique assistée par ordinateur. Ses principales fonctionnalités sont l’enregistrement, l’édition et l’analyse de sons. A partir de ce logiciel, on peut réaliser de nombreuses opérations permettant l’analyse phonétique. On citera par exemple l’analyse de formants, de la hauteur vocale, de l’intensité, etc.

Praat possède une deuxième fonctionnalité très intéressante qui est la synthèse vocale. Il possède un modèle qui lui permet de reproduire la synthèse de la voix chez l’humain en recréant artificiellement les différents organes et éléments du corps qui jouent un rôle dans la création de sons. Ainsi on peut par exemple la production d’air par les poumons ou encore la tension a un moment donné. On peut ainsi théoriquement recréer n’importe quel son en définissant correctement les bons paramètres.

Dans le cadre de ses recherches, le laboratoire ICCMR utilise le logiciel Praat pour analyser et reproduire la synthèse de paroles chez l'être humain.

### Projet global

Le laboratoire a actuellement un projet à long terme qui consiste à pouvoir utiliser Pratt pour réaliser une synthèse vocale sous forme de chant. Cela correspond à une de leur interprétation possible de la musique de demain.

Le problème est que Praat est loin d’être intuitif ou simple à contrôler. Comme je le disais précédemment, il simule les éléments du corps humain responsables de la production de la voix. Il y a 29 de ces éléments définis dans Praat. Le modèle physique de Praat est basé sur une interprétation temporelle des valeurs fournis. Si l’on considère le modèle de base, l’interprétation linéaire, il faut deux points pour représenter chaque élément, un pour le début du son et un pour la fin. Ce qui fait 58 valeurs distinctes. Comme je le disais il s’agit du modèle de base, on peut complexifier en rajoutant des valeurs intermédiaires pour faire des fluctuations vocales.

Chacune de ces valeurs est à renseigner à la main. Il faut donc connaitre à l’avance les valeurs que doivent prendre chaque paramètre à un instant donné. Ce qui fait que cela est relativement compliqué pour des sons cours et est impossible en l’état pour réaliser de sons excédants quelques secondes.

Le premier pas dans le projet du laboratoire est donc d’être capable de contrôler Praat. Pour cela, il a divisé ce grand projet en plusieurs sous projets. Mon projet est le premier de cette liste.

Il consiste à poser les bases des interactions avec Praat et à permettre de trouver automatiquement les valeurs des variables pour une voyelle donnée. Ainsi l’utilisateur n’as plus à se soucier des valeurs de chaque paramètre uniquement de fournir la voyelle à trouver, mon logiciel s’occupant du reste. Le choix des voyelles a été fait car il s’agit de sons « connus » c’est à dire dont on connait avec certitudes les valeurs de formants, d’intensité, … . Cela permet de poser les bases pour les stagiaires suivants.

### Algorithme génétiques

Praat fonctionne par intervalle de valeurs pour chaque paramètre et nous n’avons pas d’idée à priori sur les intervalles dans lesquels nos voyelles peuvent se trouver. Sachant que ces intervalles sont très étendus et ce pour chaque variable, cela fait donc des milliards de combinaisons possibles. Il faut donc fournir un algorithme qui permette de trouver une solution sans explorer tout l’espace de recherche et ce dans un temps raisonnable. C’est là qu’interviennent les algorithmes génétiques qui permettent d’obtenir une solution acceptable en explorant une partie de l’espace de recherche sans pour autant bloquer sur un optimum local grâce aux différents opérateurs d’évolutions. => a expliquer mieux, parler de l’evoltuoin naturelle et tout le bazar

### Intitulé final

Maintenant que j’ai présenté tous ces éléments, je peux dire que mon objectif était donc de créer un programme utilisant les algorithmes génétiques pour pouvoir trouver les valeurs des variables Praat afin de pouvoir récréer une voyelle donnée.

# Présentation de mon travail

Dans cette partie, je vais vous présenter le travail que j’ai réalisé tout au long de mon stage. Je vais commencer par vous présenter ce que j’ai réalisé durant les différentes phases.l’aspect gestion de projet. Enfin, je conclurais ce point en vous parlant des difficultés que j’ai rencontré durant mon stage.

## Les différentes étapes

Je vais ici vous expliquer comment j’ai découpé le temps qui m’était impartit en différentes phases et le travail que j’ai réalisé dans chacune de ces phases.

### Bibliographie

Comme je l’ai indiqué dans la présentation de mon travail, mon projet met en jeu à la fois la synthèse vocale et les algorithmes génétiques. J’avais des connaissances sur les deux grâces aux cours reçues à Polytech mais il me fallait les revoir et les approfondir avant de me lancer dans mon projet. J’ai donc passé mes premières semaines à faire des recherches bibliographiques et à me documenté sur ces deux domaines. J’ai mis en annexes une sélection d’ouvrages qui m’ont été très utiles pour réaliser mon projet.

### Interaction avec Praat

J’ai passé la seconde phase de mon stage à étudier l’Api de Praat pour voir comment je pouvais comment je pouvais l’utiliser avec les options courantes et ensuite comment le faire interagir avec un programme extérieur. Praat est un programme très complet qui propose de très nombreuses options et il n’est pas aisé de le prendre en main, même avec l’interface graphique fournie. De plus, la partie de Praat que j’utilise est à rédigé sous forme de script. J’ai donc pris du temps pour apprendre la syntaxe, me familiarisé avec la documentation et résoudre les problèmes de compatibilité.

### Algorithme génétique

Une fois familiarisé avec les algorithmes génétiques, encore fallait-il trouver une librairie qui les gérait et se familiarisé avec. En java, il existe trois grandes librairies qui le permettent : JGAP, ECJ et Watchmaker. J’ai choisi Watchmaker pour plusieurs raisons :

* Tout d’abord car cette librairie était très bien documentée.
* Ensuite car elle est mise à jour régulièrement.
* Elle est très appréciée des développeurs sur les forums traitant du sujet.
* La documentation est importante sans pour autant être étouffante.
* Les auteurs sont très réactifs à répondre aux questions des utilisateurs.

D’un point de vue technique, elle permet aussi à l’utilisateur de pouvoir implémenter ses propres objets à faire évoluer et cela est un point très fort par rapport aux autres ou les modèles de données sont calquées sur la structure des chromosomes humains et où il faut donc tout transformer. On gagne donc un précieux temps de calcul.

Ce qui est vraiment appréciable dans cette librairie est que les parties importantes telles la définition des opérateurs, des structures manipulées sont laissées au soin de l’utilisateur et que les structures de calculs utilisées par derrière auxquelles on a apriori pas besoin de toucher sont cachées et tiennent en peu de ligne de code.

J’ai donc pu implémenter une application en Java qui utilise l’API Watchmaker pour tout ce qui est la partie exécution de l’algorithme génétique.

### Intégration

La dernière et la plus grosse partie de mon temps fut consacrée à la l’intégration de tout ce que j’avais vu précédemment. Cela afin de concevoir et d’implémenter un programme d’algorithme génétique qui fait appel à un programme extérieur afin de lui déléguer la partie de calcul de score. Je vais expliquer cela en détail dans le point suivant.

Je ne traiterais pas de la partie implémentation technique dans ce rapport, un rapport spécial accessible sur le dépôt GitHub (cf plus bas) lui est dédié.

## Conception

Maintenant que je vous ai présenté comment s’était découper mon projet, je vais vous expliquer comment fonctionne mon algorithme et quelle est la solution que j’ai mise en place.

### Algorithme Génétique générique

Pour commencer, je vais faire un petit rappel sur le fonctionnement d’un algorithme génétique.

Voici l’algorithme générique d’un algorithme génétique :

1) Générer la population de candidats.

2) Fixer un score à atteindre pour arrêter l’algorithme.

3) Tant que score non atteint {

a) Pour chaque individu de la population {

-> évaluer le score (ou « Fitness »).

} Fin pour

b) Si pas de solution trouvée lors évaluation {

-> Utiliser les meilleurs individus comme parents pour la génération suivante.

-> Leur appliquer les opérateurs d’évolution.

-> On obtient une nouvelle population qui sera évalué lors de l’itération suivante.

} Sinon {

-> Quitter la boucle

}

} Fin tant que

4) Retourner le meilleur résultat

### Ma population

La structure que je vous ai présentée ci-dessus est commune à tous les algorithmes génétiques, elle est juste à adapter pour la faire correspondre à nos besoins.

Je vais maintenant vous présenter les points spécifiques de mon programme.

Commençons tout d’abord par parler des individus que je fais évoluer. Comme je l’ai expliqué dans le point précédent, l’objectif de mon algorithme est de trouver les valeurs des variables Praat pour une voyelle donnée. La structure que je fais évoluer est donc une séquence de nombres sous formes de doubles.

Comme nous n’avons pas d’idée à priori des valeurs que ces variables vont prendre, j’ai défini un alphabet de toute les valeurs possibles pour chaque paramètre. Lors de l’initialisation de la population, l’algorithme va aller créer de manière aléatoire des individus en piochant dans ces alphabets.

### Ma cible

Maintenant que je vous ai présenté les individus que je fais évoluer, je vais vous expliquer sur quel référentiel je m’appuie pour déterminer la proximité de ces individus obtenu par rapport à ma cible et comme je détermine ce score de proximité.

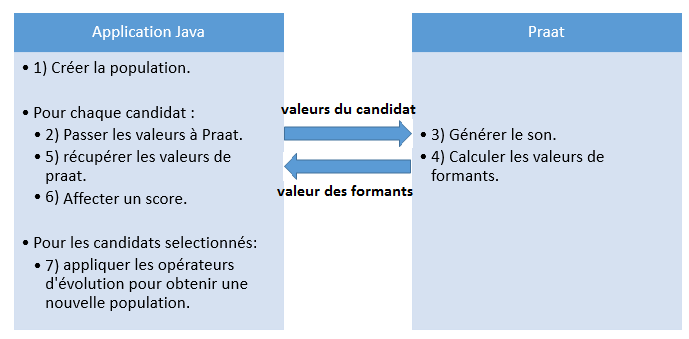
Dans le domaine de l’analyse sonore, il existe ce que l’on appelle les formants. Il s’agit des maximas d’énergie en termes de spectre sonore pour ce son. Chaque son possède un certain nombre de formants allant de F1 (basses fréquences) à FX, X variant pour chaque son. Ces valeurs de formants sont bien connues et permettent d’identifier un son donné.

Ainsi, j’ai défini les valeurs des formants pour plusieurs sons dans mon algorithme et je m’en sers pour les comparer à ceux de l’individu candidat.

Praat permet de calculer les formants pour un son donné. Ainsi pour l’évaluation de chaque individu, je vais donc générer le son correspondant aux valeurs de l’individu en question, récupérer les valeurs de formants correspondants et lui attribué un score selon la proximité avec la cible.

### Récapitulatif de mon architecture

Voici un récapitulatif de ce que je vous ai présenté précédemment sous forme de schéma :



### Ma fonction fitness et ma condition d’arrêt

J’ai parlé plus haut du fait que j’utilise les valeurs des formants pour déterminer si le son correspondant à mon individu candidat est fidèle ou non à la cible. En théorie, il faudrait que je trouve exactement les mêmes valeurs que ma cible pour arrêter l’algorithme. En pratique, deux raisons font que je suis obligé de procéder un peu différemment.

La première est que les valeurs données pour les formants sont une approximation. Chaque être humain a sa propre intonation et on estime en pratique que les valeurs des formants peuvent varier à +/- 10%. Je tiens compte de cela dans mon algorithme et autorise donc une marge de manœuvre.

La seconde raison est que les variables de Praat possèdent un très grand nombre de valeurs possibles. En effet, ce sont des doubles avec plusieurs chiffres après la virgule, ce qui fait plusieurs milliards de possibilités d’individus candidats. De plus Praat fait une approximation dans le calcul des formants et on n’est donc pas sûr de trouver un jour la valeur exacte. Or le but d’utiliser un algorithme génétique est d’obtenir un résultat acceptable dans un temps raisonnable.

Pour ces deux raisons j’autorise donc une marge de +/- 10% autour de la valeur du formant.

Ma fonction fitness consiste donc à calculer la différence entre le formant attendu et le formant du candidat. Plus ce chiffre sera bas, plus le candidat sera proche de mes attentes et donc meilleur il sera.

En conséquent, la condition d’arrêt est d’atteindre un score qui soit inférieur à la somme des marges de chaque formant de la cible et que chaque formant candidat se retrouve dans sa propre marge.

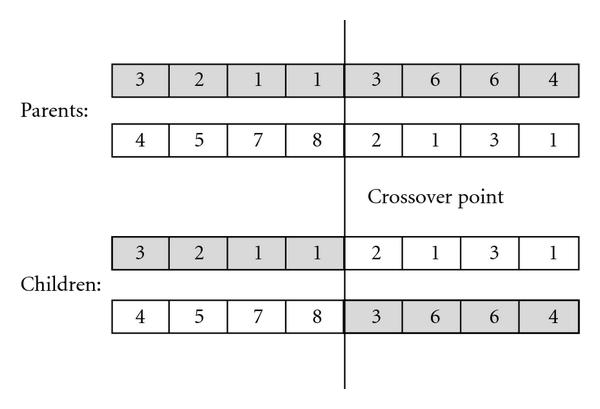
### Mes opérateurs d’évolution

Pour faire évoluer ma population initiale et obtenir un meilleur score de génération en génération, j’ai besoin d’opérateurs d’évolution à appliquer sur les « bons » candidats.

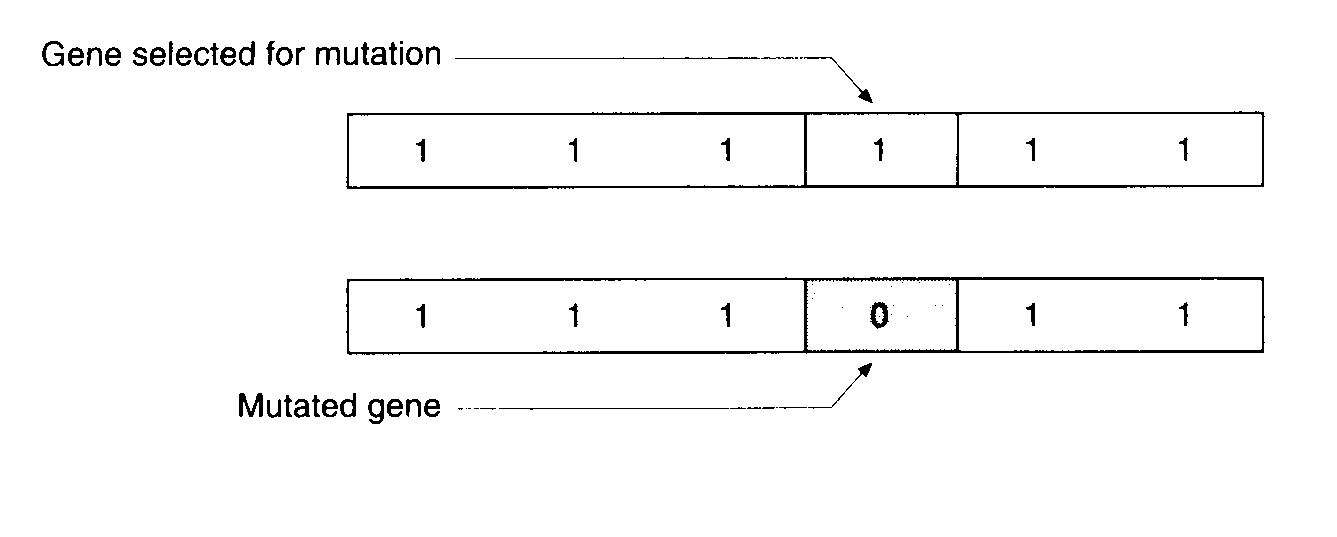
C’est opérateurs nous proviennent de l’observation de l’évolution génétique humaine et sont censés représentés au mieux l’adaptation de la population à son environnement d’une génération à une autre.

J’ai utilisé deux opérateurs d’évolution dans mon algorithme génétique. Tous d’abord, j’ai utilisé le cross over. Ce dernier permet de créer un nouvel individu à partir de deux parents. L’idée étant que les deux parents sélectionnés ont obtenus un bon score, alors en prenant une partie de chaque et en les assemblant, on est censé pouvoir obtenir un nouvel individu avec un score encore meilleur.

Un exemple est disponible dans la figure ci-dessous.



Le second opérateur que j’ai utilisé est la mutation. Avec cet opérateur, il y a une très faible probabilité que certaines variables dans ma séquence prennent d’elles même de nouvelles valeurs. Cela simule les mutations génétiques humaines qui apparaissent de manière aléatoires.

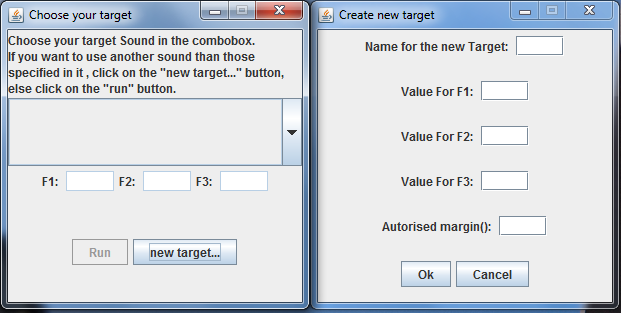


## Interface graphique

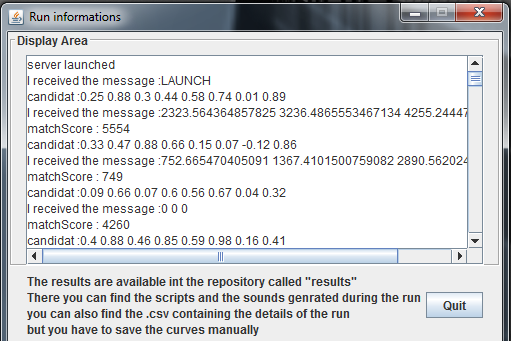
Une fois que j’ai fini de coder le programme, j’ai mis en place une interface graphique afin de permettre à l’utilisateur de pouvoir utiliser facilement et intuitivement mon application. Cette dernière est constituée de sept fenêtres.

La première permet de sélectionner la cible.

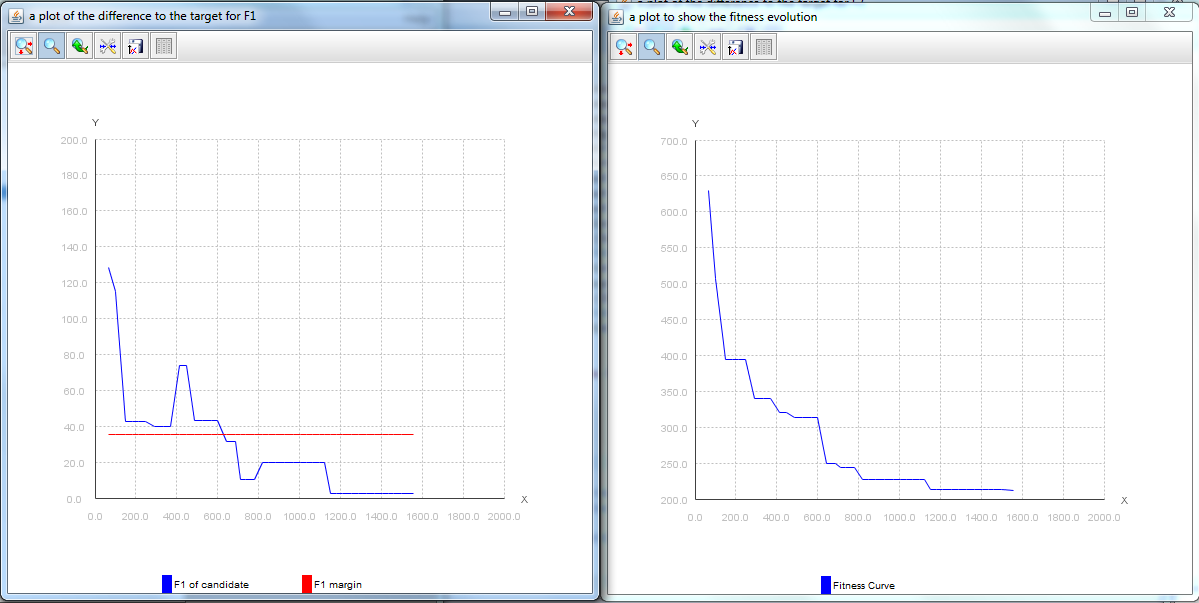
La seconde permet de créer sa propre cible si cette dernière n’existe pas.



La troisième permet d’afficher les informations sur le run en cours et de quitter l’application.



A la fin de l’exécution de l’algorithme génétique, quatre fenêtres avec les informations de monitoring sont affichées. Une fenêtre pour l’évolution de la distance pour chaque formant selon le temps et la dernière fenêtre pour afficher l’évolution globale selon le temps. Sur la capture d’écran ci-dessous je n’en ai fait figurer qu’une pour les formants.



# Gestion du projet

Maintenant que je vous ai présenté le travail que j’ai réalisé, je vais vous parler des méthodes et outils que j’ai mis en place afin de permettre au mieux le bon déroulement du projet.

## Gestion générale

### Système de fonctionnement du projet

Ce projet a été réalisé sous forme de méthode agile. Des réunions d’avancement périodiques permettaient d’évaluer la qualité de la solution proposée et de l’améliorer pour la fois suivante.

J’ai mis au point des tests unitaires et des tests d’intégration afin de pouvoir m’assurer que les nouveautés introduites n’apportaient pas de défauts de fonctionnement aux fonctionnalités déjà existantes.

### Documentation

Tout au long de ce stage j’ai rédigé plusieurs livrables permettant d’assurer au mieux la pérennité du projet. Tout d’abord un livrable en milieu de stage pour expliquer mon avancement et les techniques mises en place pour le fonctionnement de mon application.

A la fin du projet j’ai fourni :

* Ce rapport de stage pour relater mon expérience et le travail que j’avais mis en œuvre.
* Un rapport technique expliquant en détail l’architecture du logiciel et les technologies utilisées.
* Un manuel d’utilisation à l’intention des utilisateurs de mon logiciel.

## Gestion du code

### Le code en lui-même

J’ai pris grand soin tout au long du développement de l’application d’utiliser les patrons de conceptions et de coder afin de respecter au mieux les standards et permettre ainsi la réutilisabilité du code.

Ainsi chaque classe est fournie avec sa Javadoc afin de permettre une prise en main plus facile et j’ai respecté au mieux les conventions d’écritures Java afin de rendre le code le plus lisible possible.

Au niveau des patrons de conceptions voici la liste des modèles que j’ai utilisés : MVC, Etat, Observer, Singleton. Le but est de rendre le code le plus factorisé possible et d’éviter les copier-coller partout en cas de modifications. Cela permet aussi d’avoir une architecture propre et résistance à l’erreur utilisateur.

### Système de gestion de version

Afin de gérer au mieux mon code et les différents livrables produits, j’ai utilisé les services de git Hub, célèbre plateforme en ligne de type GIT.

Voici l’adresse du dépôt github : <https://github.com/phervo/ProjetEte2013>

Vous trouverez à cette adresse tout mon code ainsi que les différents rapports que j’ai produit tout au long du projet.

L’intérêt du dépôt Github est qu’il offre les opérations classiques d’un serveur git(historique des versions, branches, etc.) mais il permet également aux utilisateurs de télécharger rapidement le projet sous forme de .zip et aux futurs développeurs de reprendre rapidement mon travail en clonant ou en faisant un fork du répertoire.

# Les difficultés rencontrées

Comme je l’ai dit dans un point précédent, je ne parle pas de la partie purement technique du projet dans ce rapport. Je vais quand même exposer ici certain point qui m’ont posé des difficultés sans entrer dans le détail.

Le premier programme que j’ai rencontré vient de Praat lui-même. Comme je l’ai indiqué il s’agit d’un programme très complet. Le problème est qu’il souffre de très nombreux problèmes de compatibilité et que dès que l’on veut faire quelque chose d’un petit peu complexe, il faut parfois agir de manière détournée. Il y a beaucoup de choses qui nécessiteraient d’être recodées mais malheureusement j’ai manqué de temps lors de ce stage pour améliorer certains points qui auraient apporté un avantage non négligeable au programme. Un exemple est la possibilité de lancer plusieurs fois le programme, non disponible actuellement qui aurais permit de paralléliser l’algorithme et de gagner énormément en temps d’exécution.

La deuxième difficulté que j’ai rencontrée a été de synchroniser l’algorithme génétique et Praat. En effet ces derniers tournent sur deux processus différents. Il a donc fallu mettre en place une architecture spéciale pour pouvoir gérer Praat et cela a demander beaucoup de réflexions et de tests avant de pouvoir arriver à un premier résultat.

Cependant aucune de ces difficulté ne s’est révélée totalement bloquante et j’ai toujours réussit à trouver une solution.

# Ce que ce stage m’as apporté

Tout d’abord j’ai pu atteindre les objectifs que je m’étais fixé du point de vue linguistique et culturel, ce qui est vraiment appréciable.

J’ai également pu acquérir de nombreuses connaissances dans le domaine des algorithmes génétiques et de la synthèse vocale.

Je possède maintenant une expérience de travail à l’international et cela me sera je le pense vraiment profitable dans mon futur métier.

Paragraphe sur le travail qui fonctionne.

# Conclusion

J’ai donc pu réaliser un excellent stage qui a débouché sur un projet fonctionnel. Le code que j’ai produit peut facilement être repris par les étudiants suivants ou d’autres développeurs. La réelle problématique maintenant étant de paralléliser Praat afin de pouvoir gagner en temps de calcul.

Annexe