AISSANOU Sarah

CODAZZI Rama

GUO Kai

POLYTECH GRENOBLE

SERIOUS GAME v2

Rapport final

Table des matières

Présentation du projet

Contexte

Dans le cadre d'un projet, nous avons choisi de poursuivre le travail effectué l'année précédente en nous plongeant dans l'implémentation du projet Serious Game en apportant une seconde version. Réalisé par 3 étudiants de 4ème année en Réseaux Informatique Communication et Multimédia à Polytech'Grenoble, et tuteuré par Richard Olivier, ce projet vise le développement d'un logiciel utilisant la notion de jeux vidéo pour aider principalement des enfants ayant des difficultés dans la maîtrise de la parole, où le geste est utilisé pour compenser ce déficit.

Objectif

L'objectif du projet est d'élaborer une batterie de tests auditifs sous forme de mini jeux pour des enfants handicapés (problèmes auditifs et de parole) âgés de 7 à 10 ans. Les contraintes seront proches des tests psycho-acoustiques utilisés en laboratoire pour évaluer les compétences auditives centrales des enfants. La batterie est actuellement composée de 7 tâches :

- une tâche de latéralisation du son.
- une tâche de discrimination de sons du point de vue de la fréquence.
- une tâche de discrimination de la durée et de l'intensité.
- une tâche de reconnaissance et d'identification de sons de l'environnement.
- une tâche de détection d'un son dans du bruit.
- une tâche de ségrégation de flux sonore.

Nous avons reçu, de la part des chercheuses, des scénarios de jeux à effectuer, dont les spécifications vous seront détaillées dans la suite du rapport.

Outils

La première version utilisait le langage Python ainsi que le Framework Kivy, nous avons fait de même pour cette version. Python est un langage de programmation objet et Kivy est une librairie Python open source, pratique pour créer des interfaces graphiques. La gestion du multi-touch fait de lui un des principaux framework Python. Kivy a donc été pensé pour permettre du prototypage d'interface graphique.

Cahier des charges

Nous avons établi un cahier de charges en prenant en compte les exigences demandées par les chercheuses disponible sur notre page wiki : http://air.imag.fr/index.php/Serious_Game:_Handicap,_parole_et_geste_v2.

Exigences

- Mettre en place différents jeux à base de sons : Gestion des sons sur Python/Kivy.
- Possibilité d'ajout de jeux facilement : Un développeur doit pouvoir ajouter un ou plusieurs jeux facilement, il faudra donc bien commenter l'ensemble du code.
- Enregistrer les données de chaque utilisateur dans une base de données.
- Identification des utilisateurs et des chercheurs par un système login : Afin que le chercheur puisse analyser la base de données de chaque enfant.
- Réactivité de l'application.
- Eviter les pertes de données.

Contraintes générales

- Design épuré, efficace, application facile à prendre en main : Ici l'utilisateur est un enfant qui a entre 7 et 10 ans. Il est important de mettre en place une IHM la plus claire possible afin que l'utilisateur puisse interagir avec les jeux sans fournir trop d'efforts de compréhension.
- Gameplay travaillé : L'enfant doit pouvoir trouver un amusement à ces jeux. Pour cela il faudra mettre en place des jeux intéressants qui ne l'ennuieront pas, tout en respectant les exigences scénarios de jeux des chercheuses.

Déroulement du projet

Le déroulement du projet a commencé par une analyse complète du code de la version1. Cela nous a permis de comprendre en partie le fonctionnement de Python et de Kivy, et nous a aidé à mettre en place un premier prototype.

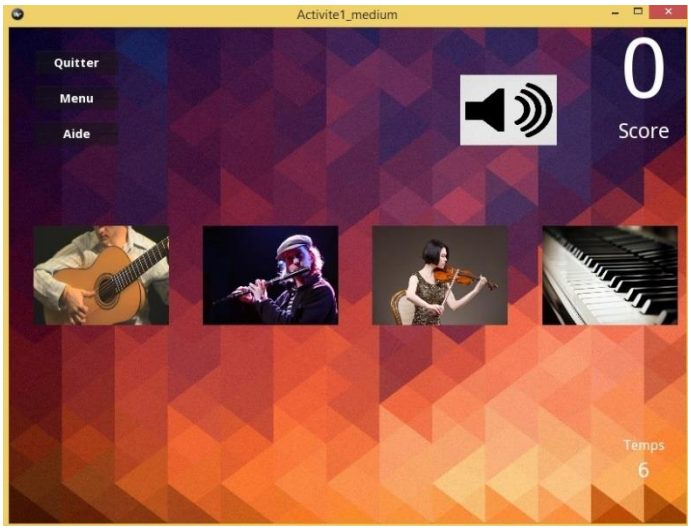
Prototype

Nous avons établi un prototype un peu tard, car on a passé plus de temps que prévu à analyser le code de la version 1 et comprendre le langage python et le Framework Kivy qui sont nouveaux pour nous cette année. Au lancement du prototype l'utilisateur entend un son (des gens qui applaudissent) et doit cliquer sur l'une des 4 images correspondantes. En cliquant sur la bonne image un son « bien » se lance pour montrer que l'utilisateur a la bonne réponse. En cliquant sur une mauvaise image un son « mauvais » se lance pour faire comprendre à l'utilisateur que ce n'est pas la bonne réponse.

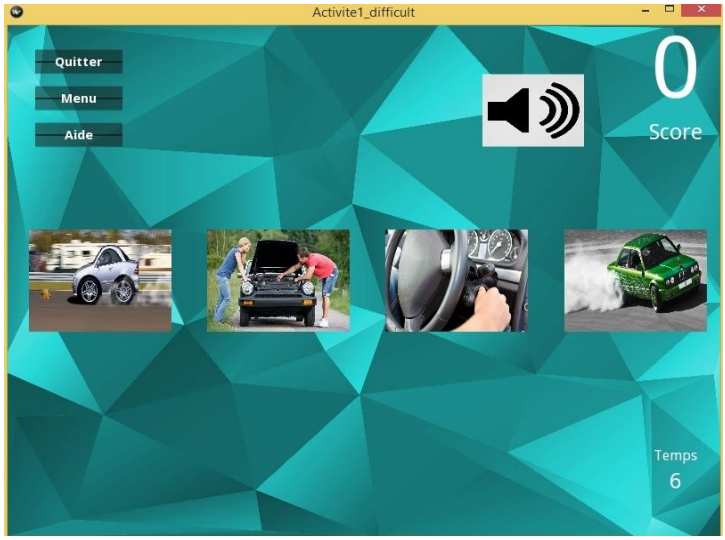
Jeu 1

Après la réalisation du prototype et en constatant notre retard nous nous sommes vite lancés dans la réalisation des jeux en commençant par le premier :

Jeu 1 facile



Jeu 1 moyen

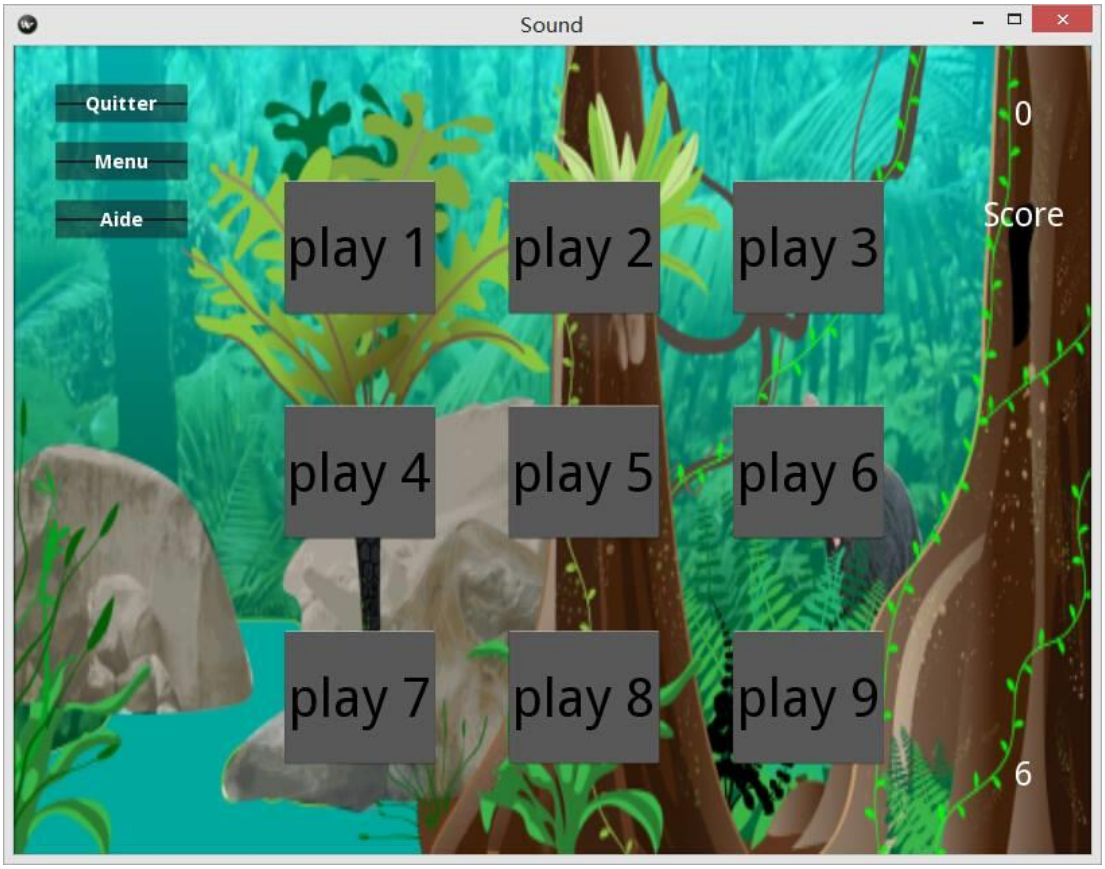


Jeu 1 difficile

Le scénario du jeu 1 est le suivant : en lançant le jeu, l'enfant entend un son, il doit cliquer le plus vite possible sur l'image qui correspond au son. Les images défilent de gauche à droite. Il a la possibilité de réécouter le son en cliquant sur l'image « Haut-parleur » en haut à droite. Les différents niveaux se distinguent par la similarité des images proposées par rapport au son. Les sons se lancent aléatoirement et son enregistrés dans des listes. Si l'utilisateur clique sur la bonne image son score augmente de 1 point pour le niveau facile, 3 points pour le niveau moyen et 5 pour le niveau difficile ; sinon il baisse d'un point. En plus

du score nous avons ajouté la notion du temps. Chaque niveau de jeu est codé dans un fichier différent.

Jeu 2



Le scénario est le suivant : En lançant le jeu l'enfant aperçoit un ensemble de cartes retournées. Chacune d'entre elles émet un son. Seules 2 de ces cartes émettent le même son, et le but est de les trouver. Le joueur découvre une première fois toutes les cartes sans perdre de points, puis en recliquant sur une « mauvaise » carte son score est décrémenté d'un point. L'enfant gagne la partie en cliquant successivement sur les deux « mêmes » cartes. Les sons sont générés aléatoirement à partir d'une liste. Nous avons mis en place un seul niveau de difficulté pour ce jeu.

Jeu 3

Scénario : L'utilisateur entend un son et a la possibilité de le réécouter en cliquant sur l'image en haut à droite « Haut-parleur ». Si le son est généré sur la droite il doit cliquer sur la porte droite, s'il est généré sur les deux oreilles avec la même intensité il doit cliquer sur la porte en face, sinon sur la prote de gauche. En cliquant sur la mauvaise porte un son « mauvais » se lance et il perd un point, et s'il trouve la bonne porte un son qui montre qu'il a gagné se lance, il gagne 5 points et une fenêtre lui propose de rejouer. Dans ce jeu on voit si l'enfant est capable de reconnaître qu'un son est généré avec une forte intensité sur une des deux oreilles.

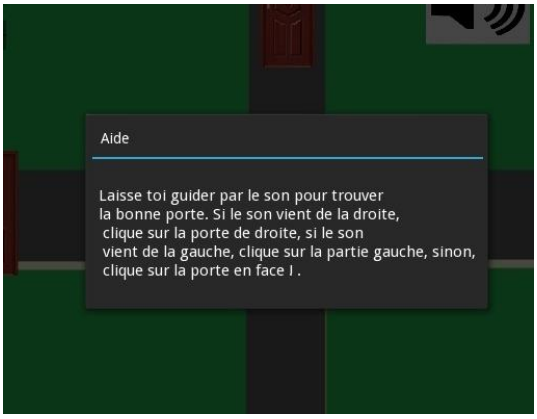
Ce jeu nécessite le port d'un casque audio stéréo afin de distinguer les nuances dans les sons émis par le jeu. De plus les sons peuvent etre généré a l'aide d'un programme python où il faut choisir la fréquence et l'intensité sonore dans chaque oreille. Pour l'instant, seule une dizaine de sons on étés générés pour ce jeu.

Amélioration de l'interface IHM

Durant ce projet, nous nous sommes concentrés sur 2 points : L'établissement des jeux proposés et l'amélioration de l'IHM. Nous venons de vous décrire le premier point. A présent, voici comment nous avons amélioré le design de l'application :

Sur chaque fenêtre de jeu, en haut à gauche apparaissent 3 boutons. « Quitter » pour quitter l'application à tout moment, « Menu » pour retourner sur la fenêtre du Menu et « Aide » qui est une fenêtre d'aide pour expliquer le principe du jeu courant.

De plus, sur chaque jeu, à chaque fois que l'utilisateur gagne, une fenêtre apparait avec plusieurs boutons pour faciliter le maniement du jeu.



Métriques

A partir de PyMetrics on a pu calculer le nombre de lignes de codes et de commentaires. Vous pourrez trouver le tableau correspondant dans le power point de présentation présente sur notre site page wiki.

Difficultés rencontrées et solutions

Dès le début du projet, la première difficulté que nous avons affrontée était la non maitrise de Python et Kivy pour les 3 membres du groupe. Pour y remédier nous nous sommes vite plongés dans différentes documentations et dans le code de la version 1 pour nous approprier ces langages le plus vite possible et avancer notre projet, mais cette partie a duré plus longtemps que prévu ce qui nous a retardé pour la suite.

Nous avons passé trop de temps sur la fonction m*ove* du jeu 1 qui permet de faire défiler les images. Nous avons alors épluché toutes les documentations possibles et nous sommes aidés d'un tutoriel de jeu en python/Kivy trouvé sur Internet.

La génération aléatoire et dynamique des sons (pour les jeux 1 et 3) et des images et sons pour le jeu 2 nous ont donné du fil à retordre. Il a fallu revoir en détail le jeu 3 de la version 1 du projet, qui gérait aléatoirement les images, et s'en inspirer. Nous avons donc aussi passé beaucoup de temps sur ce point.

Un dernier problème qu'on n'a pas résolu : Kivy bug lorsqu'on lance un fichier à partir d'un autre fichier. Par exemple, si l'utilisateur est en train de jouer, puis décide d'aller sur le menu, la fenêtre correspondante se lance, mais le « touch » ne marche pas : il a beau cliquer, rien ne se passe, et kivy bug. De même lorsqu'on joue par exemple au jeu 1 niveau facile, et qu'on veut switcher au niveau difficile : la fenêtre correspondante s'ouvre, les images défilent, le temps s'affiche correctement, le son se lance, mais kivy bug et « touch » ne marche pas.

Il y a très peu d'applications Kivy dont on aurait pu s'inspirer, ce qui a rendu toutes les difficultés précédentes encore plus complexes à résoudre. Nous avons essayé de contacter les développeurs Kivy sans réponses…

Pour aller plus loin

Par faute de temps, plusieurs points n'ont pas pu être abordés :

Etablir différents niveaux de difficulté pour les jeux 2 et 3, gérer les images du jeu 1 dynamiquement, et gérer le déplacement du « petit bonhomme » dans un labyrinthe pour le jeu 3. Avec plus de temps, nous aurions voulu régler le bug kivy lorsqu'on lance un fichier à partir d'un autre. De plus, nous n'avons pas abordé certains objectifs initiaux comme le system de login ou la base de donnée en ligne.

Conclusion

Ce projet a été très enrichissant pour nous. En effet il nous a permis de découvrir le langage Python ainsi que le Framework Kivy. Ce fut un défi pour nous de nous plonger dans un projet aussi concret et devoir le coder dans un langage qui nous était inconnu nous a encore plus motivés. Nous espérons pouvoir aider et inspirer les prochaines versions du Serious Game.