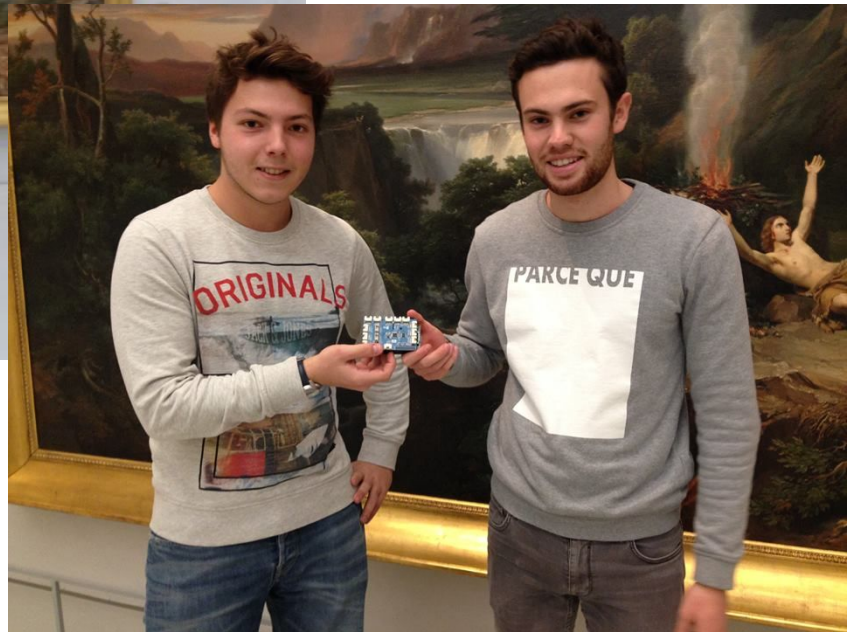


# PROJET FAS

-

## Dialogue interactif avec une oeuvre



# Explication:

## Projet initial :

Tout d'abord nous répondons à l'appel d'offre du musée Fabre. Notre projet se traduit par la proposition de différentes activités interactives proposés au public lié à une oeuvre d'art. Tout d'abord l'activité principale est une diffusion d'une explication historique de l'oeuvre. Le public peut interagir avec l'explication. C'est à dire que le volume changera en fonction de l'ambiance dans la pièce. Mais aussi le public pourra relancer l'explication, faire pause etc... ou revenir aux choix des activités. Devant l'oeuvre sera installé un capteur de distance pour vérifier si une personne est présente ou non devant l'oeuvre. Si une personne est présente on interagit avec elle, sinon on attend en essayant d'attirer les gens qui pourraient être autour. Ces interactions seront gérés par des capteurs de mouvements et/ou de boutons. De plus on ajoutera des petits jeux en lien avec l'oeuvre choisie.

## Projet final :

La version finale diffère quelque peu de notre version initiale prévue.

Dans la version finale l'utilisateur aura la possibilité de tout d'abord choisir d'écouter la description d'une oeuvre ou de répondre à un questionnaire portant sur cette dernière. Si il choisit la description du tableau de Remond il aura la possibilité d'avoir une description de la partie droite ou gauche du tableau en fonction de sa position. ( il pourra choisir entre 2 positions qui seront indiquées par des pastilles sur le sol.)

Une fois la description terminée on passe directement au questionnaire correspondant.

Tout au long du projet le son de l'explication du tableau se fera en fonction du son actuel dans la salle. Si le son émis par les personnes dans la salle augmente le son produit par l'explication augmente aussi proportionnellement et inversement.

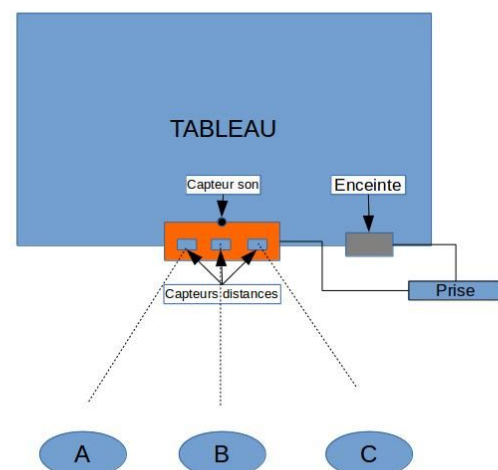
On peut à tout moment changer de descriptions, de quizz ou d'arrêter le projet pour contempler le tableau lui même.

# MODE D'EMPLOI

## MOYENS MATÉRIELS :

Tout d'abord nous utilisons 4 capteurs différents sur notre raspberry ainsi qu'un shield. Une fois le shield posé sur le raspberry il faudra brancher les capteurs sur le shield.

- Un capteur son pour détecter le volume sonore en temps réel.
- Un capteur de distance pour regarder si une personne se trouve à droite.
- Un capteur de distance pour regarder si une personne se trouve à gauche.
- Un capteur de distance pour regarder si une personne se trouve au milieu.
- Une enceinte ou des écouteurs pour pouvoir diffuser les explications.



Pour la partie logicielle nous avons eu recours à plusieurs librairies :

- Time. Cette librairie nous a permis de faire des calculs en temps réel. Nous l'utilisons quand nous voulons regarder pendant 2 secondes si une personne a bien interagit avec le tableau en se plaçant devant un des capteur par exemple.
- Fmod. FMOD est une bibliothèque multiplateforme de gestion du son, cela nous a permis de charger les différents fichiers mp3 qui vont être par la suite joués en fonction des actions du clients. Nous avons le choix entre différentes librairies mais FMOD nous permettait en plus de lancer des sons, d'en lancer plusieurs à la suite dans un même programme, de faire pause ou encore de regarder si un son était en cours à un moment précis.
- Grovepi. Cette librairie fournit les fonctions basiques pour utiliser les différents capteurs et le raspberry dans le langage C. Nous l'avons utilisé pour récupérer les données des capteurs de distances de son.
- Utilisation d'un makefile pour compiler: Se rendre dans le dossier RENDU, entrer la commande "make" puis pour lancer l'application entrer la commande "./bin/Musee", si il y a des problèmes de droits pour i2c, entrer la commande "chmod 777 /dev/i2c-1"

Ensuite dans un but purement pratique nous voulions pouvoir se connecter à notre raspberry par wifi. Mais n'étant pas sûre de pouvoir avoir un wifi stable au musée nous avons donc transformé notre raspberry en hotspot grâce aux packages hostapd et dhcp. Cela nous permet d'accéder à notre raspberry n'importe où grâce à notre téléphone en ssh. Nous utilisons aussi pulseaudio et de bluez pour que le son que le son produit par notre raspberry sorte sur une enceinte reliée en bluetooth.

## MOYENS HUMAINS

Nous avons travaillé souvent ensemble. Au début nous avons commencé par dessiner des schémas pour se représenter l'ordre dans lequel nous voulions que les actions s'enchaînent. Puis au niveau du code le plus compliqué était de savoir comment anticiper et contrer les différents problèmes pouvant arriver sur place. Nous avons donc listé ensemble ces différents problèmes.

Ensuite au niveau du code nous nous mettions d'accord sur certains points allions en salle TP puis pendant que l'un codait l'autre cherchait comment améliorer le projet. Vu que l'on ne pouvait pas vraiment travailler chez nous par manque de capteurs nous pouvions travailler seulement en salle TP donc ensemble.

Hugo: Le raspberry m'appartenait je l'ai donc totalement configuré pour notre projet. De plus ayant déjà travaillé sur la librairie FMOD et en particulier codé en C, j'avais des connaissances qui ont permis de faciliter la gestion du code de la partie audio du projet et le code du projet en général.

Melvil: Je me suis occupé entre autre de la gestion, c'est à dire prendre rendez vous avec le musée, récupérer les différents textes lié à l'oeuvre. Ayant aussi des connaissances en C j'ai pu aidé Hugo lors des différentes heures d'autonomie, notamment pour détecter certains problèmes dans le code étant donné que j'avais plus de recul sur le programme que mon camarade.

## Code :

Tout le code fourni dans le répertoire Github, mis à part les libraires, a été écrit par notre binôme. Vous retrouverez donc dans Musée.c le programme principal permettant de faire fonctionner notre projet.

Voici comment est hiérarchiser notre dossier RENDU:

- alerte (dossier)

#Contient les éléments audio "alertes" (nous en avons seulement un: trop de bruit)

- questions (dossier)

#Contient les éléments audio "questions"

- réponses (dossier)

#Contient les éléments audio des différents résultats en fonction des réponses

- titres (dossier)

#Contient les éléments audio des différents titres c'est à dire les explications etc...

- src (dossier)

  - Musee.c

  - grovepi.c

  - grovepi.h

- obj (dossier)

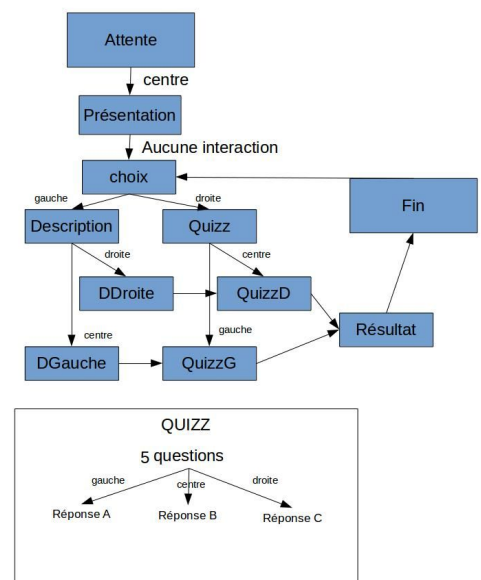
#Contient les éléments .o

- bin (dossier)

#Contient l'exécutable

- Makefile (fichier)

# Fichier Makefile



## Voici un récapitulatif des fonctions utilisées sous forme de spécification fonctionnelle :

initQuestions() :

# Charge les fichiers audio correspondant aux différentes questions et initialise les réponses à ces questions dans un tableau

lancerSons():

# Pré-conditions : Aucun fichier audio ne doit être en cours de lecture (vérification dans le main).

# Récupère le nom d'un fichier audio préalablement chargé en entrée.

# Lance la lecture de ce fichier audio.

initFMOD() :

# Charge les fichiers audio correspondant aux différentes réponses, descriptions et discours de présentations de l'oeuvre.

freeFMOD():

#Fonction appelée en tout fin de programme permettant de libérer l'espace mémoire utilisé lors du chargement des fichiers audio.

bonneDistance(): capteur -> Booléen

# Récupère le nom d'un capteur distance passé en paramètre et observe la distance devant ce dernier.

# Return : 1 si la personne se trouve entre le distance min et max, 0 sinon

estDansRond(): capteur -> Booléen

#Appelle la fonction bonneDistance() en fonction du capteur passé en paramètre durant un certain laps de temps

# Renvoie 1 si un personne se trouvait dans le rond pendant le laps de temps défini, 0 sinon

reinitialiser(): -> Booléen

#Fonction permettant de revenir au début du programme à n'importe quel moment. Si le capteur du milieu capte pendant plus de 2 secondes une distance de moins de 5 centimètres. Retourne vrai si on doit réinitialiser faux sinon

arreter(): -> Booléen

#Fonction permettant d'arrêter le programme à n'importe quel moment. Si les trois capteurs captent pendant plus de 2 secondes une distance de moins de 5 centimètres. Renvoi vrai si le programme doit s'arrêter, faux sinon

reponse(): -> char

#Renvoi une lettre correspondant à une réponse donnée par un utilisateur. En fonction du capteur elle renvoie une lettre différentes. Renvoi A ou B ou C si un personne est capté pendant un laps de temps sur un capteur, ou renvoi D

changerSon(): void

#Permet de changer le son en fonction du son capté. Et si le son est trop fort lance le son demandant au public de faire moins de bruit.

## Perspectives

Si nous avions eu plus de temps nous aurions voulu créer un logiciel permettant de générer du code pour rendre notre projet adaptable à tout oeuvre. À ce stade notre projet convient à notre utilisation, mais si le musée a besoin de ce type d'outil pour d'autres oeuvres il ne fonctionne plus. Dans l'idéal un des membres du personnel n'aurait besoin que de rentrer le nom d'une oeuvre et d'enregistrer les parties audio pour que notre code soit généré et prêt à l'emploi. Nous pensons quand même que nous avons fini notre projet notamment grâce aux retours très positifs que nous avons eu des membres du musée.

Au niveau de la commercialisation, on ne peut pas distribuer notre produit à grande échelle étant donné que celui ci a été développé spécialement pour l'oeuvre de Jean Charles Joseph Remond. Si nous voulions le commercialiser ce serait sous forme de commande. Les clients viendraient à nous et chaque produit serait créé au cas par cas . C'est pour cela que réaliser un générateur de code nous aurait permis de le vendre à grande échelle puis de réaliser des mises à jour suites aux différents retours de nos clients. D'après notre contact au sein du musée Fabre, il y aurait un marché potentiel étant donné que ce type de produit ajouterait une touche de technologie au sein d'un musée, cependant il faudra obligatoirement que ce dernier puisse s'adapter très vite à différentes oeuvres comme dit précédemment.

## Conclusion

Pour conclure, ce projet nous a apporté du concret. On a pu voir comment un projet peut être mené de a à z en répondant à une demande.

Cela nous a apporté beaucoup, en terme de connaissance dans le domaine de la programmation, en particulier avec le raspberry et les capteurs, mais aussi au niveau de la gestion et des relations humaines avec le client.

L'autre facette du projet à été de travailler en collaboration avec le musée Fabre ce qui nous a motivé à finir et peaufiner notre projet pour pouvoir le présenter lors de la nocturne se déroulant courant Février, nous sommes pressés de voir les retours des futurs utilisateurs.