

Compte rendu projet Arduino

METTATON

Koralie Porcel
François Canava
G4

Table des matières

Explication globale du projet.....	2
1) Objectif.....	2
2) Comparaison avec ce que nous avons fait.....	2
3) Vision globale	3
4) Planning	4
Explications précises du projet.....	5
1) Problèmes rencontrés et solutions	5
2) Algorithme	7
3) Développement des parties du projet	8
4) Ce que nous ferions maintenant différemment	8
Conclusion.....	9
1. Conclusion.....	9
2. Perspectives	9
3. Bibliographie	10

Explication globale du projet

1. Objectif :

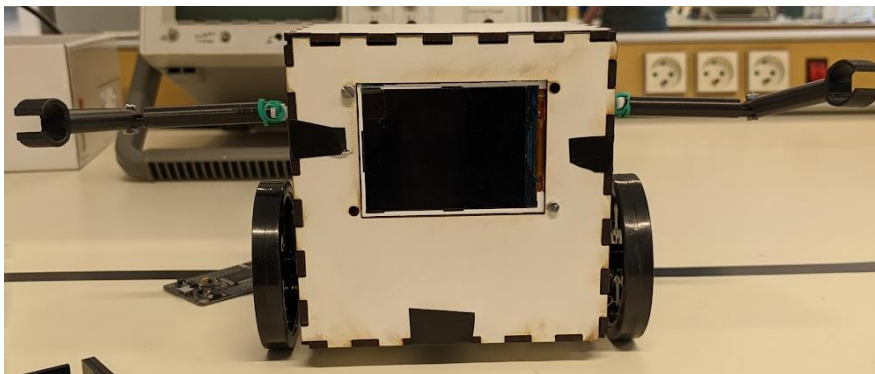
L'objectif du projet est de réaliser un robot "quiz" fonctionnel dans le temps imparti, satisfaisant au mieux le client ainsi que nous même.

Le jeu commence à l'aide du téléphone. Une partie est constituée de 10 questions choisies aléatoirement. Ces questions vont être affichées sur l'écran de Mettaton (notre robot) et sur l'écran du téléphone. En même temps, sur le téléphone, il y a 4 choix de réponses, le joueur devra cliquer sur une des réponses.

Mettaton a différentes réactions selon les réponses du joueur. Puis la question suivante est posée.

A la fin des 10 questions, le score du joueur est affiché sur l'écran et Mettaton a différentes réactions. De plus, lors du quiz, il y aura une musique.

2. Comparaison de ce que nous avons finalement fait :



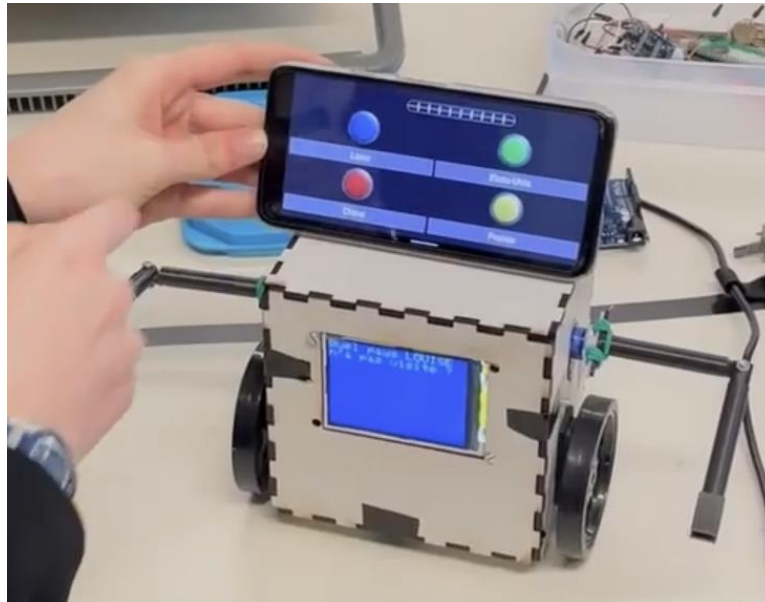
Apparence finale de Mettaton



Exemple de l'affichage de l'interface

Globalement ce que nous avons fait ressemble à ce que nous nous attendions. Nous n'avons pas pu suivre le planning que nous nous étions fixés car nous étions en retard dans beaucoup de domaines ce qui fait que notre projet n'est pas totalement finalisé. Mais il respecte en grande partie ce que nous avions prévu même si le rendu est moins complet que ce que nous avions imaginé. Contrairement aux objectifs fixés, nous n'avons pas de musique. Le robot manque de décorations et le score n'est pas affiché sur l'écran. Il n'y a que deux réactions à la fin des dix questions posées. La question n'est posée que sur l'écran mais pas sur le téléphone. Pendant les réactions, l'écran n'affiche pas d'image.

3. Vision globale du projet avec les différents modules :



Le projet est une boîte en bois comportant 4 servomoteurs, deux permettant d'actionner les bras en plastique, deux autres permettant de faire tourner les roues. Il a un module Bluetooth HC-06 qui permet la communication Bluetooth entre l'Arduino et le téléphone. Il comporte une interface graphique avec quatre choix de réponses à la question sur l'application bluetooth Electronics. Le projet comporte une alimentation de 6 volts et enfin un écran LCD permettant d'afficher les questions. La carte Arduino utilisée est une carte Mega.

4. Plannings :

Diagramme de Gantt initialement prévu :

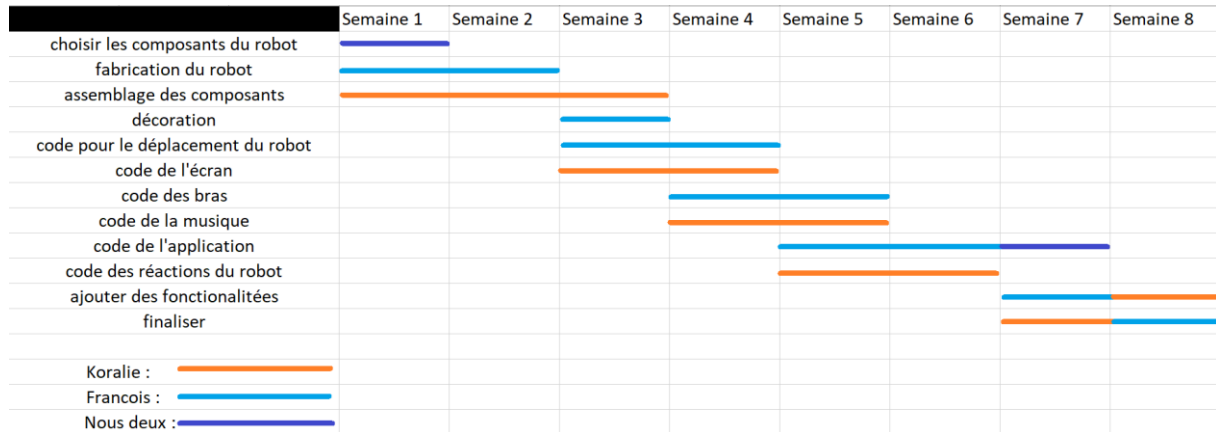
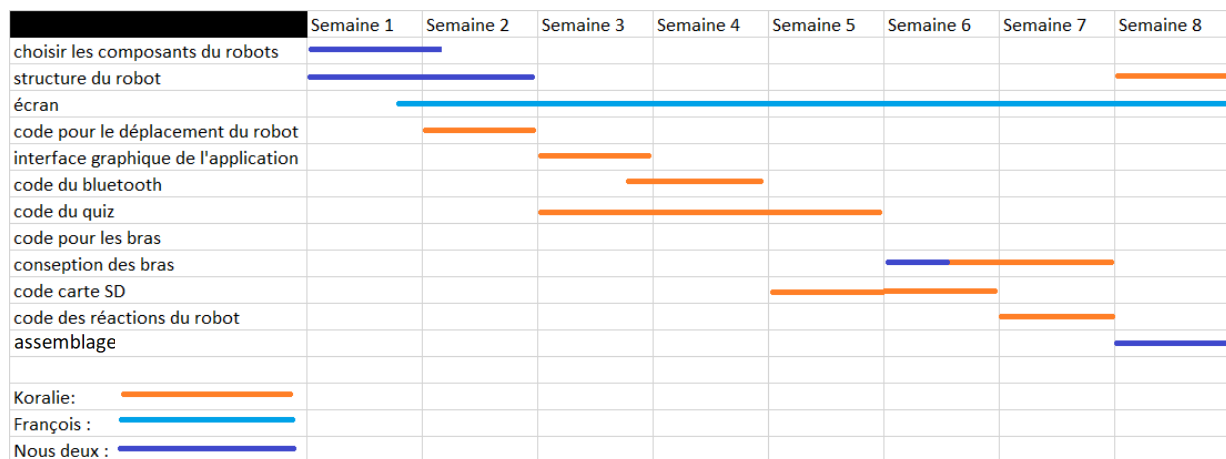


Diagramme de Gantt finalement réalisé :



Explications précises du projet

1. Problèmes rencontrés et solutions :

Lors de ce projet, nous avons rencontré chacun un problème important.

- Le problème de François était l'écran :

Il y a eu de nombreux problèmes liés à l'écran, le plus handicapant fut le choix de l'écran (ST7920 LCD). En effet, lors de la première séance nous pensions que celui-ci fonctionnait comme celui du cours ce qui n'était pas du tout le cas. De plus, nous nous sommes rendu compte lors de nos recherches que les caractéristiques de l'écran (exemple sa taille en pixels) ainsi que la bibliothèque nécessaire à son utilisation (u8g2) n'étaient pas optimisées pour notre projet. Nous avons réalisé à quel point cela allait être problématique que trop tardivement car nous pensions pouvoir trouver une solution à chaque problème sans avoir à changer d'écran. Cependant cela était bien trop complexe voire infaisable la plupart du temps, et ce malgré les recherches sur les forums spécialisés, ce qui a engendré une perte de temps monumental et nous a amené à échanger l'écran trop tardivement (environ deux séances avant l'oral). Même si le nouvel écran (ILI9341 TFT) dont nous disposions était bien plus pratique grâce à ses caractéristiques plus appropriées, certaines bibliothèques censées fonctionner avec le ILI9341 étaient obsolètes ou bien certaines de leurs fonctions ne marchaient simplement pas (comme ILI9341_t3 et ces exemples qui ne compilent pas). Pour palier à ces problèmes, nous avons opté pour ADAFRUIT_GFX et ADAFRUIT_ILI9341 qui ont été plus adaptés cependant le temps pour un apprentissage approfondi de leur fonctionnement fut malheureusement trop court pour pouvoir répondre à nos spécifications initiales.

On pourra de plus mentionner que rechercher la référence du ILI9341 TFT redirigeait vers un écran légèrement différent ce qui ralentit aussi sa mise en place.

- Le problème de Koralie était la mémoire de la carte Arduino :

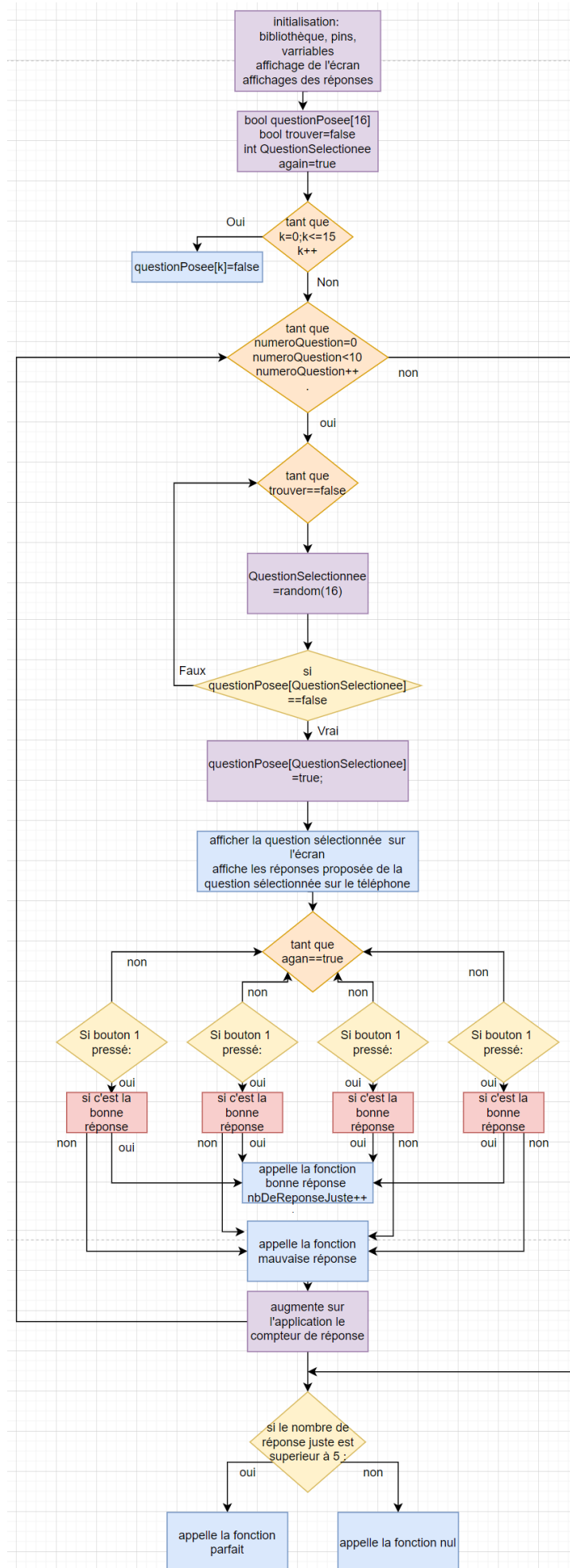
En effet, notre projet possède une base de données de questions qui prend énormément de mémoire. De plus, la communication entre le téléphone et l'Arduino ne marchait plus au-dessus de 60% de mémoire de variables globales utilisées. Il fallait donc trouver une solution pour réduire la mémoire.

Pour pouvoir contrer ce problème de place, j'ai passé deux séances pour prendre connaissance de la bibliothèque de la carte SD et réussir à lire des fichiers dedans. J'ai eu beaucoup de problèmes pour réussir à manipuler la carte SD, je n'arrivais pas à

lire les fichiers à l'intérieur mais suite à des recherches et de la patience, j'ai réussi à régler ces problèmes. Je ne peux stocker sur la carte SD que des String ou des entiers pour pouvoir ensuite les lire. Je pouvais donc stocker dans des fichiers différents seulement les questions et les réponses. Cependant, lors de l'implémentation du programme du code principal avec le programme de la carte SD, je me rends compte que cela n'a pas du tout réglé le problème c'était même pire. En effet, l'importation de la librairie permettant la lecture des données sur la carte SD prenait 30% des variables globales ce qui dépassait la limite des 60% pour que la communication Bluetooth puisse fonctionner. Je n'ai pas trouvé de solution pour réduire cette utilisation de place. Face à cela, la seule solution trouvée a été de changer de carte SD pour prendre une carte Mega, réglant ainsi le problème de place puisque la carte Mega possède une plus grande mémoire.

Il y a également eu d'autres problèmes minimes qui ont pu être réglés facilement tel que le bras 3D en plastique qui s'est cassé, il a donc fallu le coller. Il y a eu également des problèmes liés à la communication entre l'Arduino et le Bluetooth.

2. Algorithmes:



3. Développement de chacune des parties du projet :

Le projet comporte quatre grandes parties :

- La structure est faite en bois découpé à la découpe laser et modélisé grâce à onshape comportant 6 pièces détachées.
- Les réactions du robot sont faites grâce à quatre servomoteurs :
 - 2 micro-servomoteurs ayant une rotation de 180 degrés qui permettent dans notre projet, de faire bouger les bras qui sont attachés au moteur à l'aide d'élastiques. Ils comportent 4 pièces en plastique, deux pour chaque bras. La première pièce est l'avant-bras et la seconde, l'arrière-bras. Les deux pièces sont reliées par une vis permettant de régler l'inclinaison de l'avant et de l'arrière-bras à la main. Les pièces du bras ont été imprimées en 3D et modélisées grâce à onshape.
 - 2 servomoteurs à rotation continue à 360 degrés. Contrairement à un servo moteur classique qui garde un angle précis, celui-là tourne comme un moteur classique, ce qui permet de faire tourner les roues pour pouvoir faire avancer ou bien reculer le robot.

Quatre fonctions ont été rajoutées pour pouvoir coordonner les mouvements des bras et des roues pour les réactions.

- L'Interface est faite sur l'application Bluetooth électronique. Sur l'interface, il y a quatre propositions de réponse aux questions ainsi qu'un indicateur comportant une jauge indiquant l'avancée des questions (à dix, la jauge est pleine). Pour pouvoir communiquer entre l'Arduino (maître) et le téléphone (esclave), nous utilisons le module HC-06.
- L'écran LCD qui permet l'affichage des questions. Pour écrire plusieurs questions à la suite, le programme va réécrire le font et ensuite écrire la question pour ne pas avoir de superpositions.

4. Ce que nous ferions de différents maintenant :

Maintenant, comme nous avons un peu plus d'expérience dans la réalisation de projet Arduino, nous irions plus rapidement.

Nous avons appris à surmonter nos problèmes et nous ferions plus attention aux choix de nos composants (l'écran et la carte Arduino notamment) afin qu'ils correspondent vraiment à nos besoins, nous hésiterions moins à les changer pour faciliter l'avancée du projet plutôt qu'à s'obstiner à chercher des solutions plus complexes et se compliquer la vie. Notre gestion du temps serait, elle aussi bien meilleure ce qui permettrait sûrement de mieux remplir nos objectifs initiaux.

Conclusion

1. Conclusion :

Pour conclure, nous avons créé un robot quiz qui possède un écran où une question est choisie aléatoirement et sur le téléphone, il y a quatre propositions de réponse. Les fonctionnalités suivantes sont opérationnelles : l'affichage des questions sur l'écran et des réponses sur le téléphone, le choix des questions aléatoirement, l'enchaînement des questions, les réactions suite aux choix du joueur, la réaction finale du robot. Cependant, le robot avait des problèmes d'alimentation. Il n'y avait pas assez de courant pour alimenter convenablement tous les modules. De ce fait, les réactions n'étaient pas ce qu'elles devaient être.

2. Perspectives:

Avec 9 séances de plus, nous aurions pu régler les problèmes d'alimentation, mieux régler la police d'écriture et l'affichage des questions et les réactions.

Le projet manquait de finalité pour l'écran il aurait donc fallu régler ce manque.

Si nous avions eu des séances de plus, nous aurions pu rajouter de la musique sur le robot qui interagirait et changerait en fonction des réponses et de l'avancé des questions.

Mais dans sa globalité, le projet ne peut pas beaucoup évoluer par rapport à ce qu'il est actuellement. Nous aurions pu ajouter des articulations au niveau des avant-bras, pour avoir des réactions plus différentes. Cependant, nous n'aurions pas pu améliorer beaucoup de choses avec le modèle choisi de notre projet.

3. Bibliographie :

<https://www.carnetdumaker.net/articles/lire-et-ecrire-des-donnees-sur-une-carte-sd-avec-une-carte-arduino-genuino/>

<https://www.redohm.fr/2018/09/servomoteur-a-rotation-continue/>

<https://www.keuwl.com/apps/bluetoothelectronics/>

- 1^{er} écran :

<https://www.electronicshub.org/interfacing-128x64-graphical-lcd-with-arduino/>

<https://forum.arduino.cc/t/how-to-display-a-string-variable-with-the-u8g2-library/622278/4>

<https://github.com/olikraus/u8g2>

- 2eme écran :

[http://adafruit.github.io/Adafruit-GFX-](http://adafruit.github.io/Adafruit-GFX-Library/html/class_adafruit_gfx.html#a8c31e63d43f7285e47aa48c363d86ce9)

[Library/html/class_adafruit_gfx.html#a8c31e63d43f7285e47aa48c363d86ce9](http://adafruit.github.io/Adafruit-GFX-Library/html/class_adafruit_gfx.html#a8c31e63d43f7285e47aa48c363d86ce9)

https://electronoobs.com/eng_arduino_tut58.php(attention les pin DC et RST sont inversées dans le programme mis en exemple)

http://www.lcdwiki.com/2.8inch_SPI_Module_ILI9341_SKU:MSP2807

https://github.com/adafruit/Adafruit_ILI9341