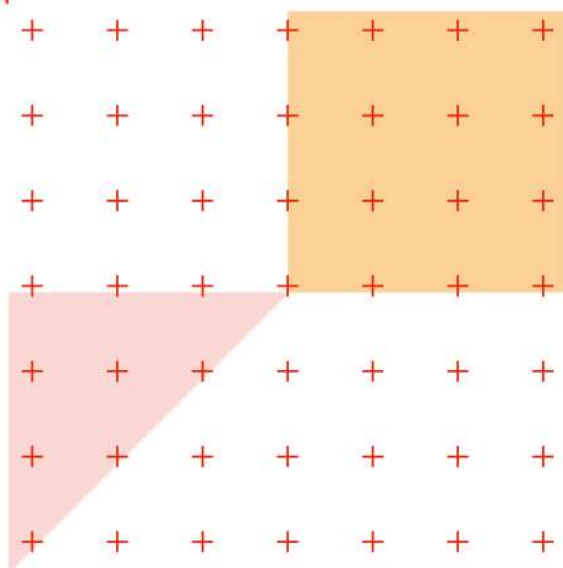
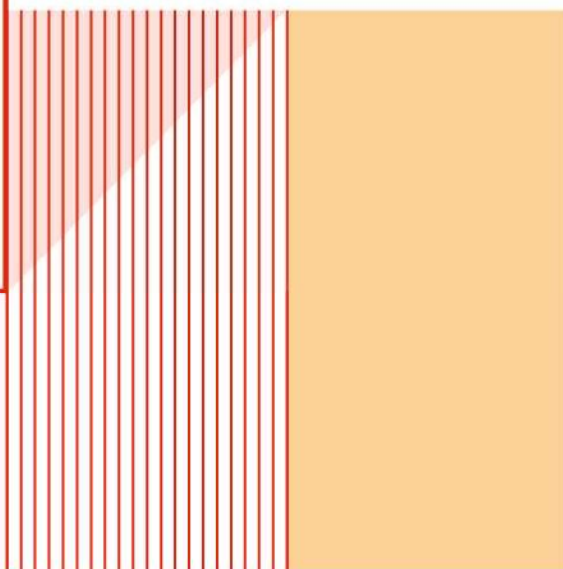


PIANO AUTOMATISE

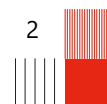
Cyprien Heusse

Emily Holmes



SOMMAIRE

Piano automatisé	1
SOMMAIRE	2
Introduction.....	3
Choix techniques	3
Conception orientée objet.....	4
Fonctionnalités du programme.....	6
Conclusion.....	6



INTRODUCTION

Dans ce projet, nous avons décidé de travailler sur un piano automatisé. Le but est de pouvoir jouer à peu près n'importe quel morceau existant en quelques clics.

Nous imaginons des utilisations de l'ordre du divertissement principalement. Par exemple, dans le cadre d'une maison hantée, d'une attraction ou d'un *escape game*. A cause de ces utilisations, il était important pour nous de préserver l'apparence des touches appuyées quand l'on joue de la musique. On peut également l'imaginer dans un bar où le client pourrait jouer la musique de leur choix s'ils gagnent un jeu, par exemple.

Dans ce rapport, nous présenterons à la fois des choix techniques pour l'implémentation en C++ de notre projet, ainsi que les diagrammes de conception que nous avons utilisé pour nous aider à concevoir le code.

CHOIX TECHNIQUES

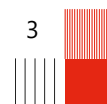
Un fichier MIDI est un simple fichier qui contient les informations permettant de jouer un morceaux de musique. Il contient événements qui correspondent à l'activation ou la désactivation d'une note avec un timing associé.

Nous avons décidé d'utiliser un Raspberry Pi plutôt que l'ESP8266 car nous avons besoin de pouvoir traiter des fichiers MIDI ce qui est plus simple sur un Raspberry ayant un OS.

Nous avons utilisé deux bibliothèques pour le projet : MidiParser de MStephan99 (repo [ici](#)) et PiGPIO de abyz (site [ici](#)).

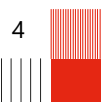
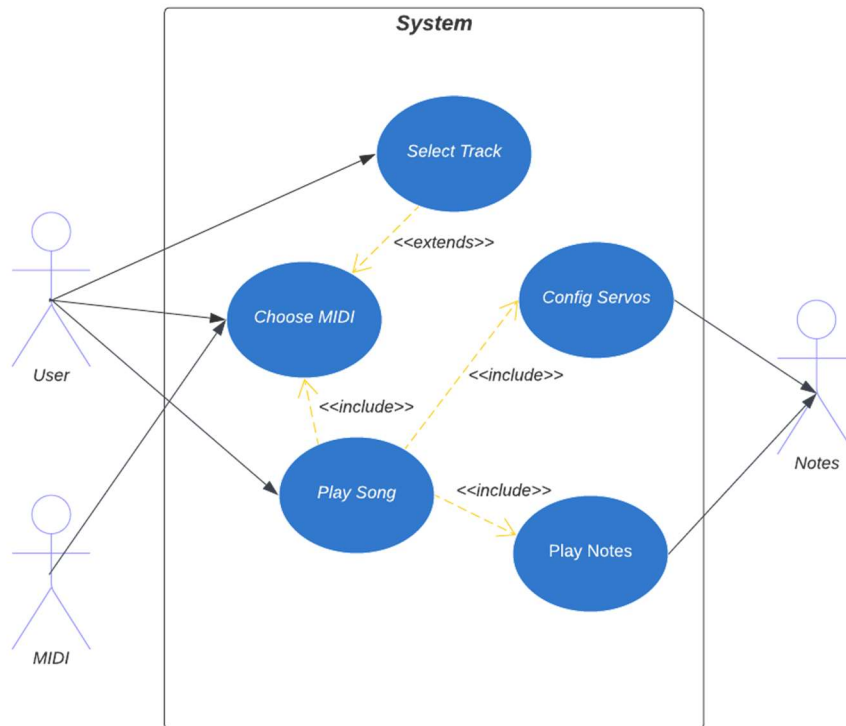
Du point de vue mécanique, nous avons utilisé un servo moteur par touche du piano. Chaque servo tire une ficelle qui tire la note vers le bas, ce qui produit le son de la note. Les servo sont les [SER0053](#) que nous avons choisis notamment pour leur faible bruit. Etant limité par le nombre de GPIOs sur le Raspberry Pi, nous utilisons un total de 24 servo, soit 24 notes.

Ainsi, notre programme prend la forme d'une commande qui prend le fichier MIDI à jouer en paramètre et qui lance le morceau sur le piano. Ce fonctionnement est pour l'instant très simple mais pourrait être amélioré en ajoutant une bibliothèques de morceaux, une application de contrôle du Piano et de nombreuses autres fonctionnalités.

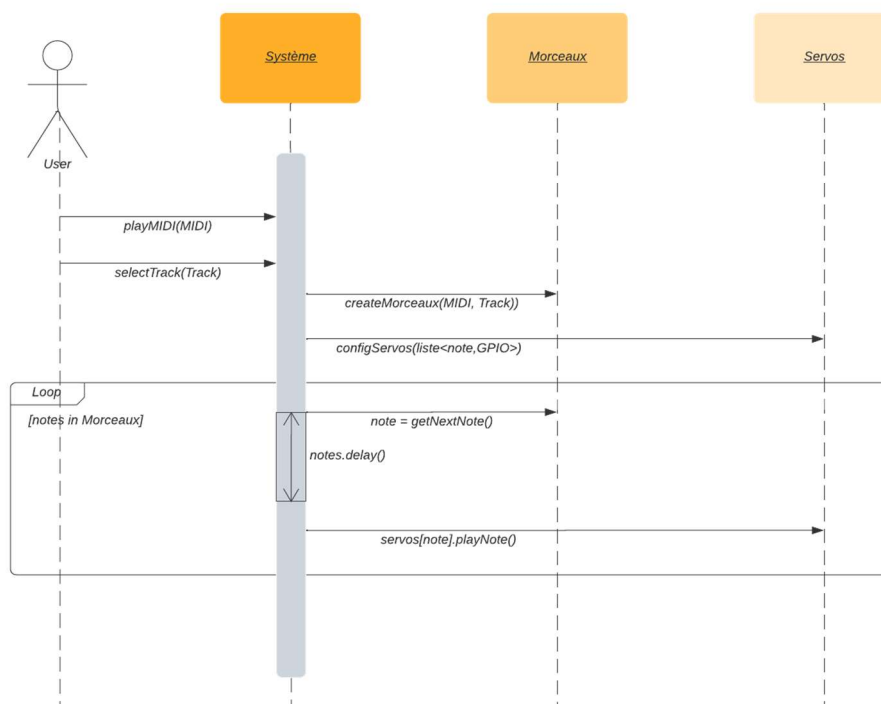


CONCEPTION ORIENTEE OBJET

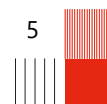
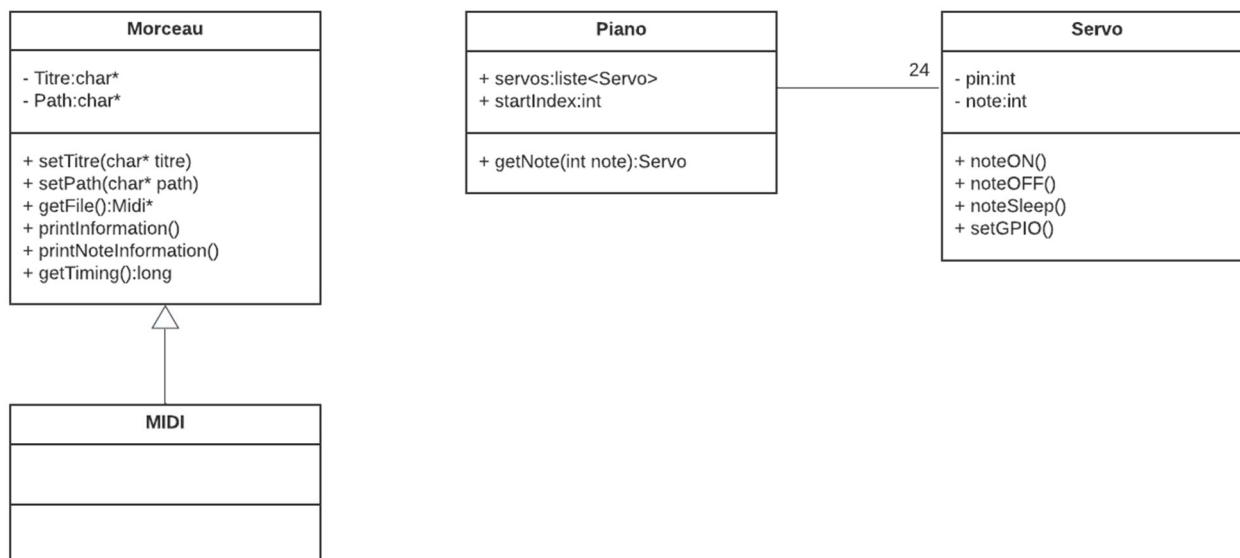
Nous avons 3 acteurs principaux : l'utilisateur du piano, le fichier MIDI et les notes du piano. Ces acteurs interagissent de la façon suivante :



Le fonctionnement typique du programme suit le diagramme de séquence suivant :



Tous ces éléments permettent de créer le diagramme de classe suivant :



FONCTIONNALITES DU PROGRAMME

Comme exigé dans le cahier des charges (sujet du projet), notre code contient les éléments suivants :

- **Plusieurs classes** (Morceau, Piano, Servo)
- **Héritage d'une classe** (Morceau hérite de MIDI)
- **Redéfinition d'opérateurs** (Plusieurs classes redéfinissent l'opérateur « << » et Piano redéfinit l'opérateur « [] »)
- **Création d'une exception** dans le cas où l'on essaie de créer un Servo avec un pin GPIO invalide
- **Utilisation de la STL** via plusieurs vecteurs (« vector »).

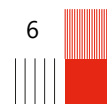
La bibliothèque pigpio doit être précompilée (guide sur <https://abyz.me.uk/rpi/pigpio/download.html>) puis le programme peut être compilé et utilisé en suivant les commandes décrites dans le README.md du repository [GitHub](#).

Note : A cause de notre setup SSH pour communiquer avec le Raspberry Pi, tous les commits du repository apparaissent depuis un seul compte GitHub, mais le travail a été réalisé en binôme.

CONCLUSION

Ce projet fut l'occasion d'implémenter les concepts du paradigme orienté objet tel que nous les avons abordé à la fois en cours de C++ mais aussi Conception Orienté Objet. Nous avons également dû prendre en compte des contraintes électriques (par exemple, s'assurer que nous allions fournir un courant suffisant pour alimenter l'ensemble des GPIO). Nous avons également appris à travailler avec d'autres librairies pour créer un nouveau projet.

Nous sommes plus que satisfaits du travail que nous avons accompli et nous comptons continuer à travailler sur notre piano afin de l'améliorer avec, par exemple, une application pour le choix des musiques et une meilleure expérience utilisateur.



INSA TOULOUSE

135 avenue de Rangueil
31400 Toulouse

Tél : + 33 (0)5 61 55 95 13

www.insa-toulouse.fr



INSA | INSTITUT NATIONAL
DES SCIENCES
APPLIQUÉES
TOULOUSE