PROJET THÉÂTRE DMX



DMX 512

[**PARTIE 1 : Présentation commune du projet** 3](#_Toc36562677)

[Présentation et Introduction du Sujet 3](#_Toc36562678)

[Le but du projet 3](#_Toc36562679)

[Le Principe de réalisation du projet 4](#_Toc36562680)

[*Synoptique simplifié du système* 4](#_Toc36562681)

[*Synoptique simplifié de la console matérielle* 5](#_Toc36562682)

[Analyse Fonctionnelle du système 5](#_Toc36562683)

[*Diagramme de cas d’utilisation* 5](#_Toc36562684)

[*Diagramme d’exigences* 7](#_Toc36562685)

[*Diagrammes de classe* 8](#_Toc36562686)

[Organisation du projet 9](#_Toc36562687)

[*GANTT Prévisionnel* 9](#_Toc36562688)

[*GANTT Réel* 10](#_Toc36562689)

[*Répartition des tâches* 10](#_Toc36562690)

[Organisation de l’équipe 11](#_Toc36562691)

[*Compte rendu d’activité (CRA) et Cahier de bord* 11](#_Toc36562692)

[*GITHUB et Versionning* 12](#_Toc36562693)

[*Logiciel d’analyse et de développement* 12](#_Toc36562694)

[*Maquettage et Prototype* 15](#_Toc36562695)

[Choix technique et Étude Physique 16](#_Toc36562696)

[*Choix du boitier DMX & de la lampe* 16](#_Toc36562697)

[*Choix des composants de la console matérielle* 17](#_Toc36562698)

[Recette 19](#_Toc36562699)

[Tests d’intégration du prototype 20](#_Toc36562700)

[Avancement et Conclusion 21](#_Toc36562701)

[**PARTIE 2 : Présentation Etudiant 4 / BOURY** 22](#_Toc36562702)

[Analyse Fonctionnelle 22](#_Toc36562703)

[Diagramme de cas d’utilisation 22](#_Toc36562704)

[Diagrammes de Séquence 23](#_Toc36562705)

[Séquence : Ajouter une adresse d’équipement 23](#_Toc36562706)

[Séquence : Modifier une adresse d’équipement 23](#_Toc36562707)

[Séquence : Créer un équipement 24](#_Toc36562708)

[Séquence : Modifier un équipement 24](#_Toc36562710)

[Séquence : Supprimer un équipement 25](#_Toc36562711)

[Diagramme de classe / Classes utilisées 26](#_Toc36562712)

[Modules de Test 27](#_Toc36562713)

# **PARTIE 1 : Présentation commune du projet**

## Présentation et Introduction du Sujet

Le théâtre de la providence va bénéficier en 2020 d’une importante rénovation aussi bien au niveau de l’architecture d’intérieur que de sa scène avec la mise en place d’un dispositif de son et de lumière.

La partie “scène” sera confiée à Sono+, une société amiénoise reconnue pour son expérience dans le monde du spectacle. Le BTS Système Numérique sera associé à ce projet par la réalisation d’un système capable de gérer l’ensemble des lumières à distance, ainsi que par la mise en place d’un réseau pour piloter via le wifi l’ensemble du dispositif de lumière. Le groupe projet participera aux choix différents composants de la scène pour vérifier l’intégration avec leur système.

## 

## Le but du projet

La scène est mise à disposition de tous. Il est donc nécessaire d’avoir un système simple et compréhensif facilement pour que chacun puisse profiter au maximum des jeux de lumières qui seront installés par la société audio-visuel. Aujourd’hui, tous les équipements professionnels de lumière sont équipés d’un protocole de communication de type DMX (512 1024 etc)

Une application en C++ permettra de paramétrer les lumières (adresse, canaux de paramétrages etc) et servir de serveur TCP pour faire la passerelle entre le réseau de lumière DMX et l’IHM Web de l’application client.

Une IHM web/interface WEB, qui sera responsive pour mobile, sera développé pour communiquer avec le serveur C++ pour envoyer via TCP les différentes scènes de lumière que pourra préparer, créer ou modifier un enseignant.

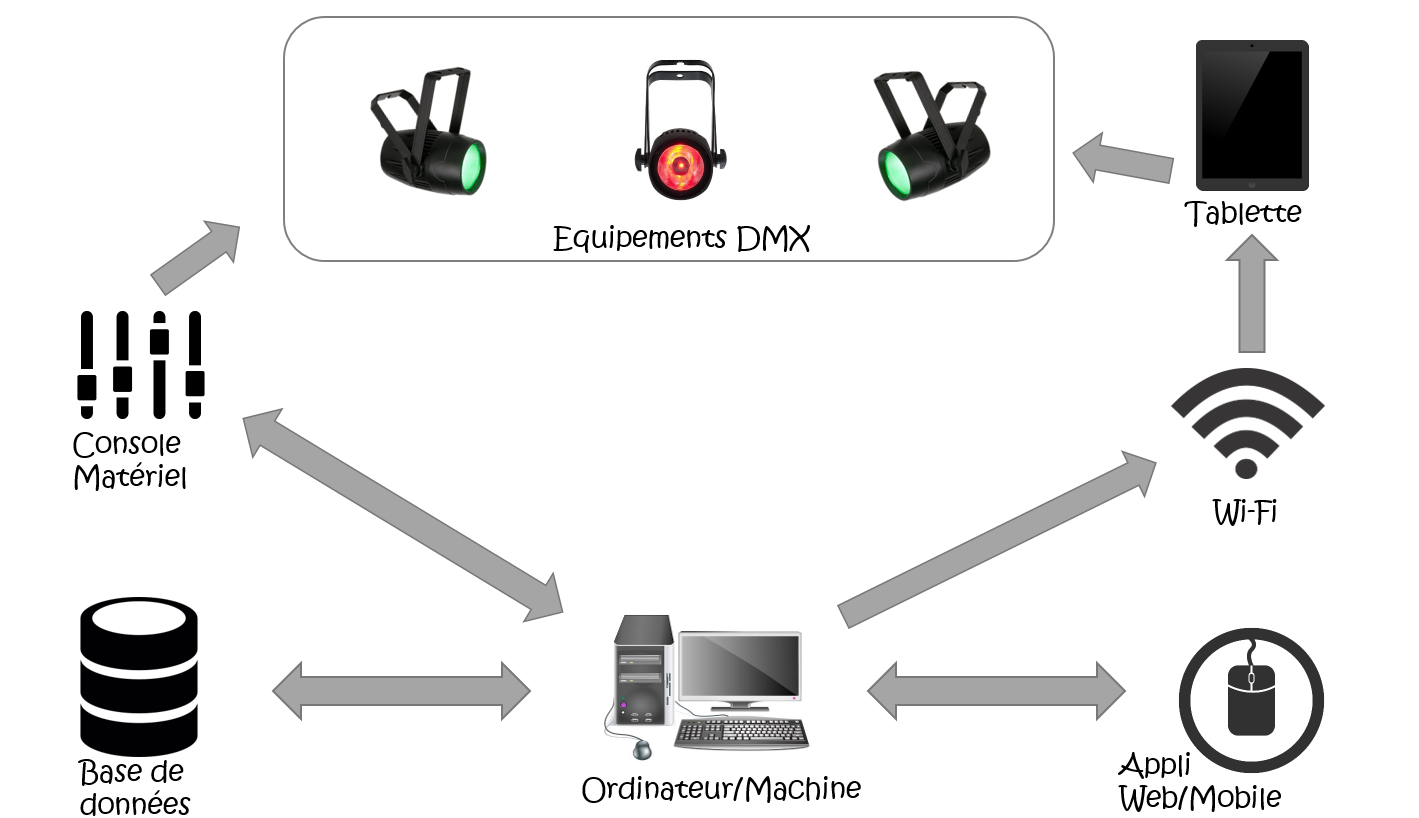
Via la méthode Drag&Drop, il sera facile de lancer les scènes d’un programme (qui seront stockés dans une Base De Données).

Pour faciliter la création des scènes, un prototype de type “console matérielle” sera effectué.

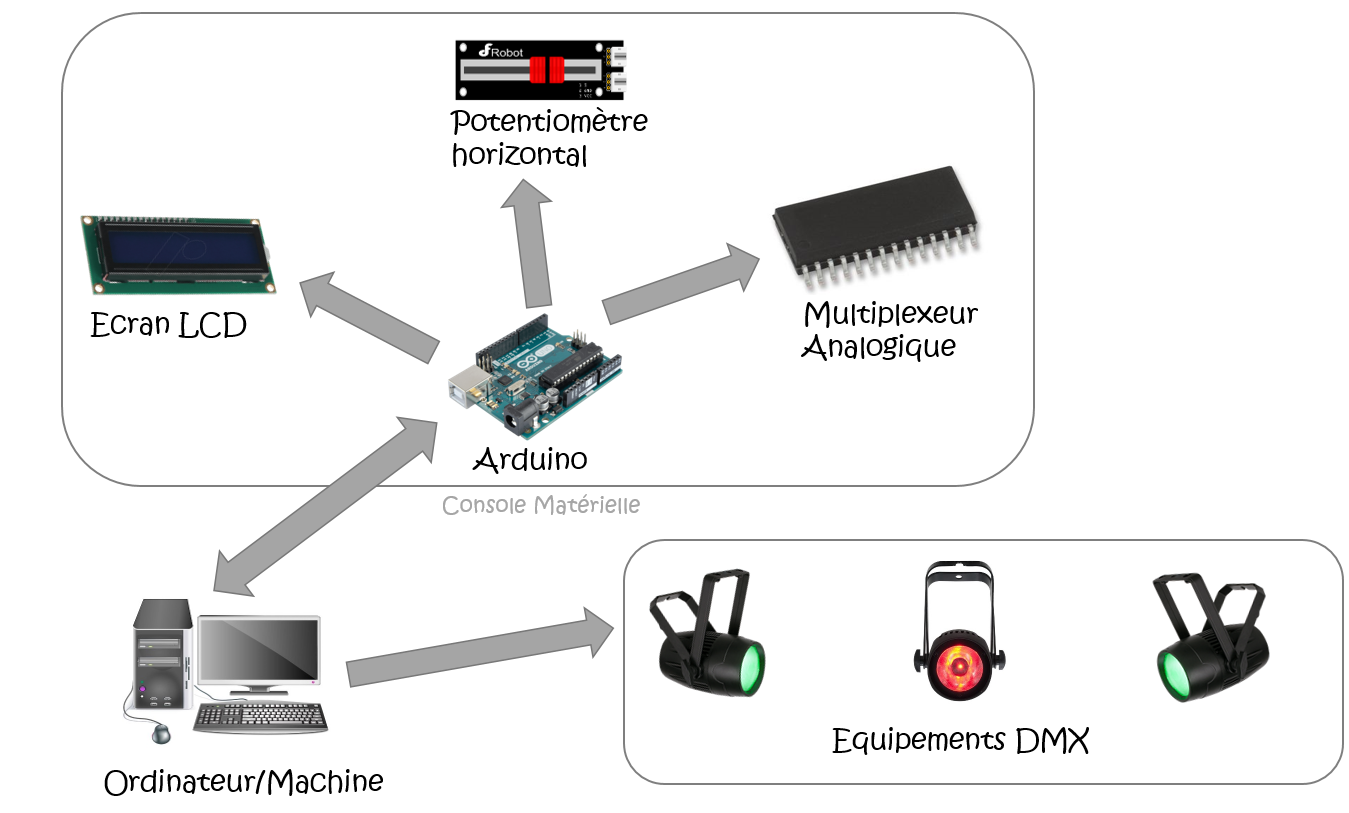
Elle sera reliée en série ou en USB pour contrôler l’application via des boutons, des potentiomètres et un afficheur LCD.

## Le Principe de réalisation du projet

### *Synoptique simplifié du système*

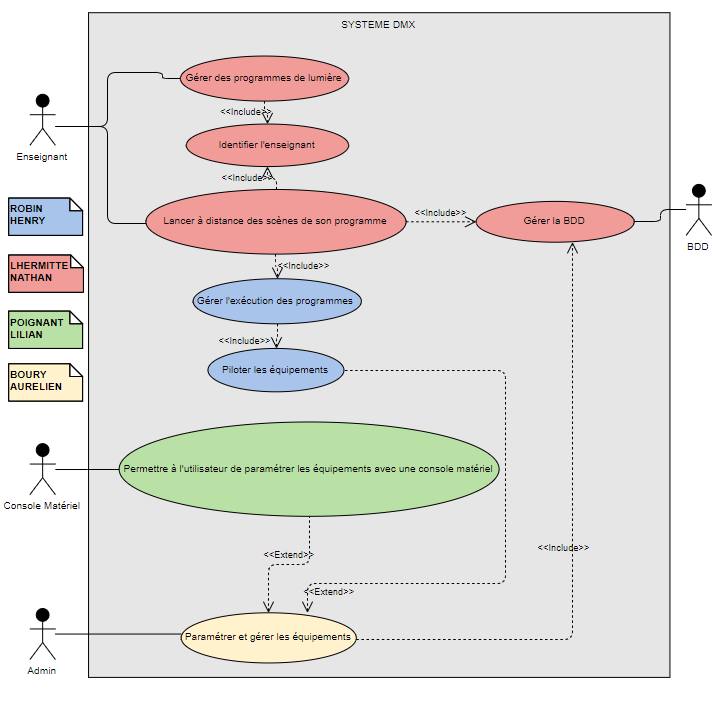


### *Synoptique simplifié de la console matérielle*



## Analyse Fonctionnelle du système

### *Diagramme de cas d’utilisation*



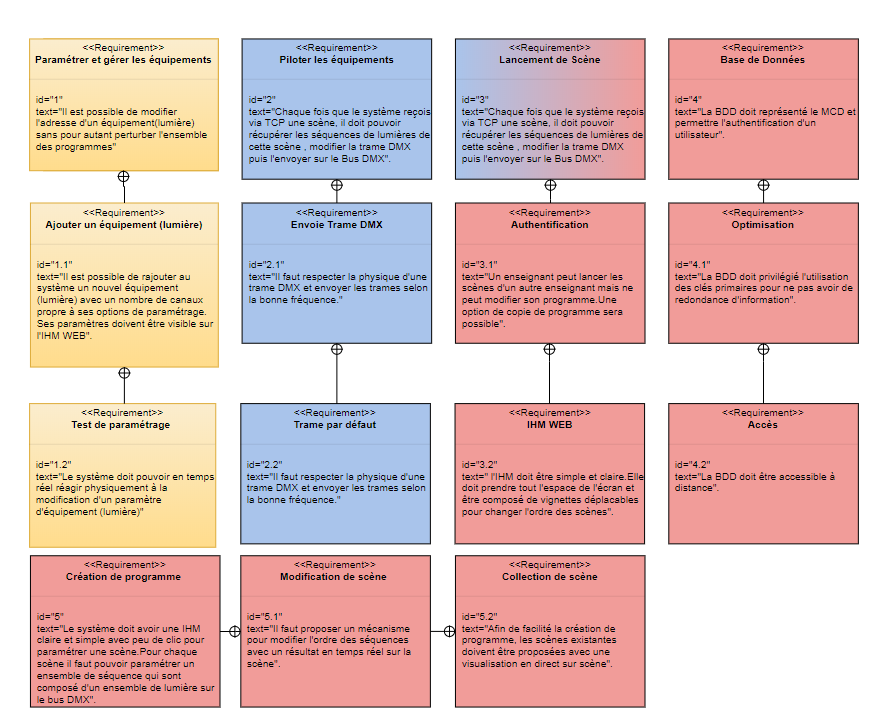
Les bulles représentent les fonctionnalités que propose le système

Ici l’Enseignant ne pourra agir que sur le système de deux façons. La première c’est en gérant des programmes de lumières sur l’interface web. La seconde est de lancer à distance des scènes de son programme via l’interface web.

Ensuite un Administrateur pourra paramétrer et gérer les équipements qui fera en sorte que tout soit paramétré pour que l’enseignant n’ai rien à faire.

L’enseignant pourra utiliser la console matérielle pour faciliter l’utilisation des équipements.

### *Diagramme d’exigences*

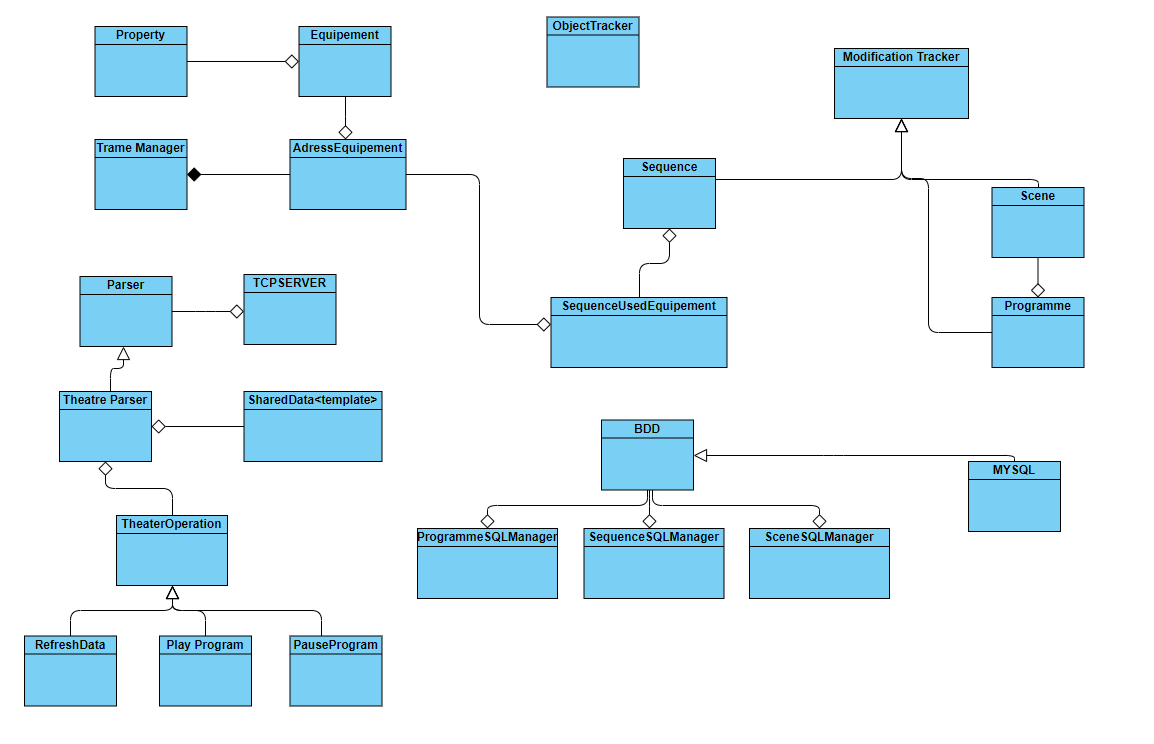


Aurélien Boury.

Nathan Lhermitte.

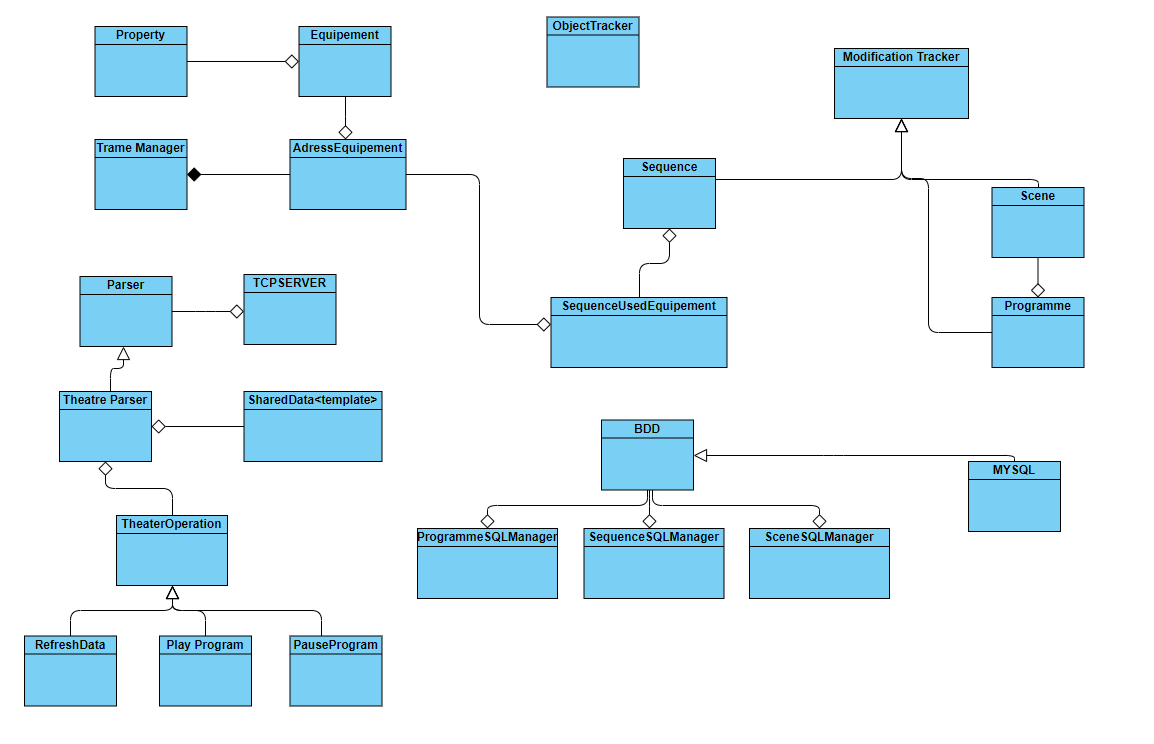
Robin Henry.

### *Diagrammes de classe*



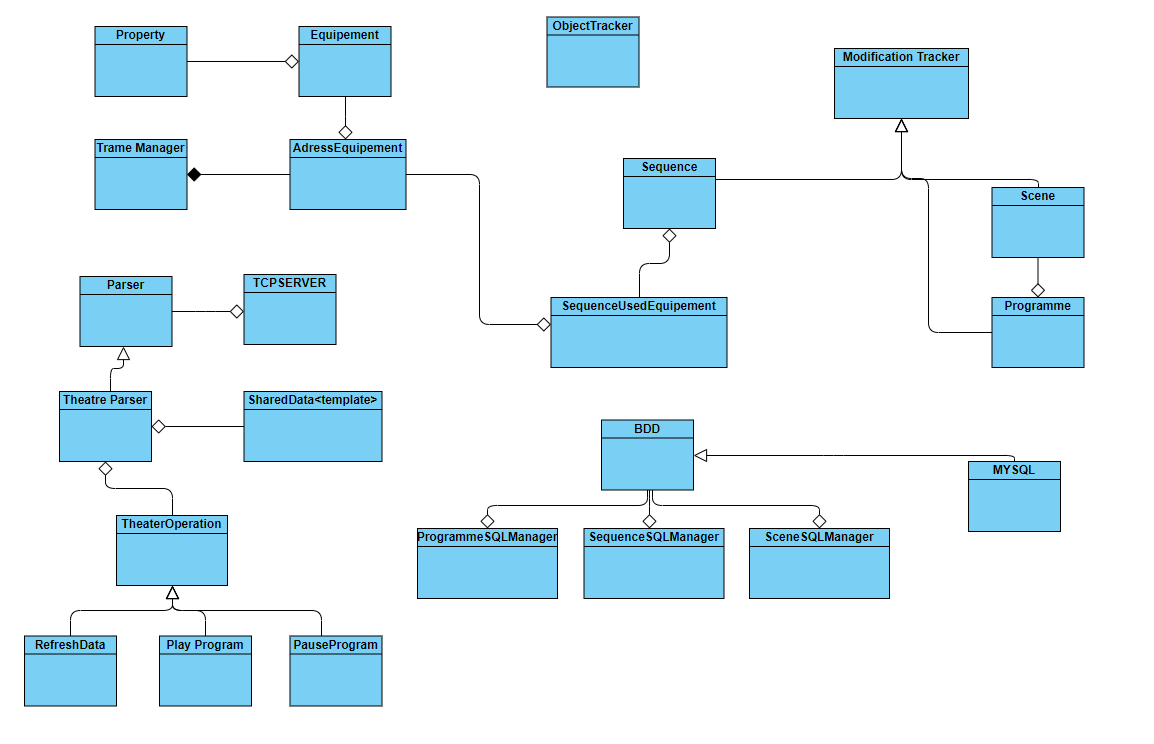
Partie TCP / IP :

Ces classes serviront à communiquer entre l’IHM Web donc la partie web avec la partie C++ qui pilotera les équipements



Partie BDD / MYSQL :

Ces classes serviront à ajouter / modifier / supprimer des programmes / scènes / séquences dans la Base De Données via le C++



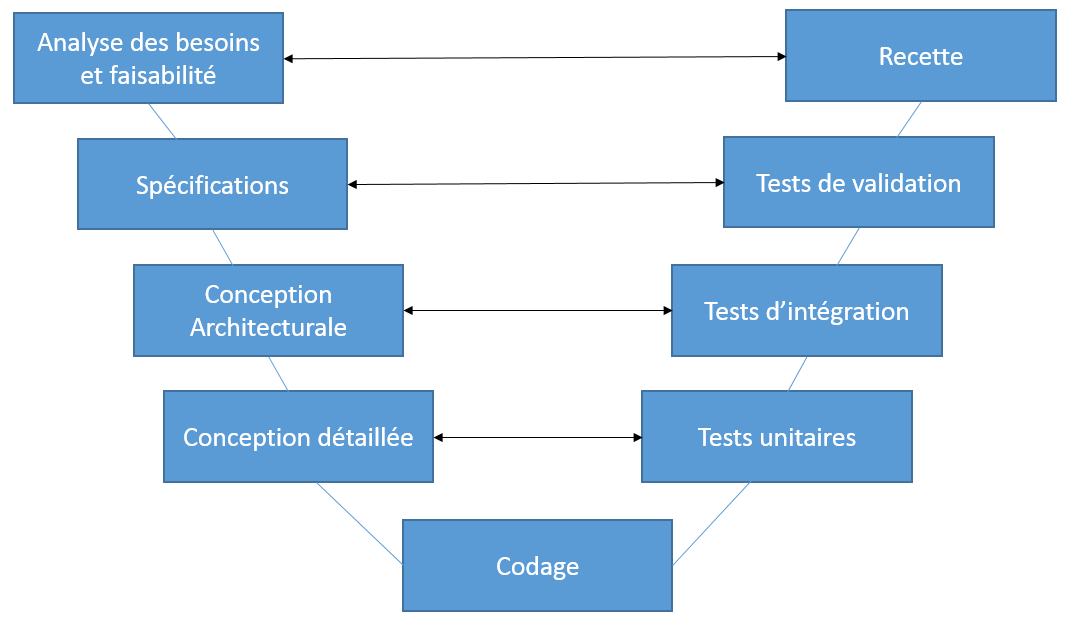
Partie C++ :

Voici le cœur du code C++, ces classes serviront à piloter les équipements grâce à des Programmes, des Scènes et des Séquences.

## Organisation du projet

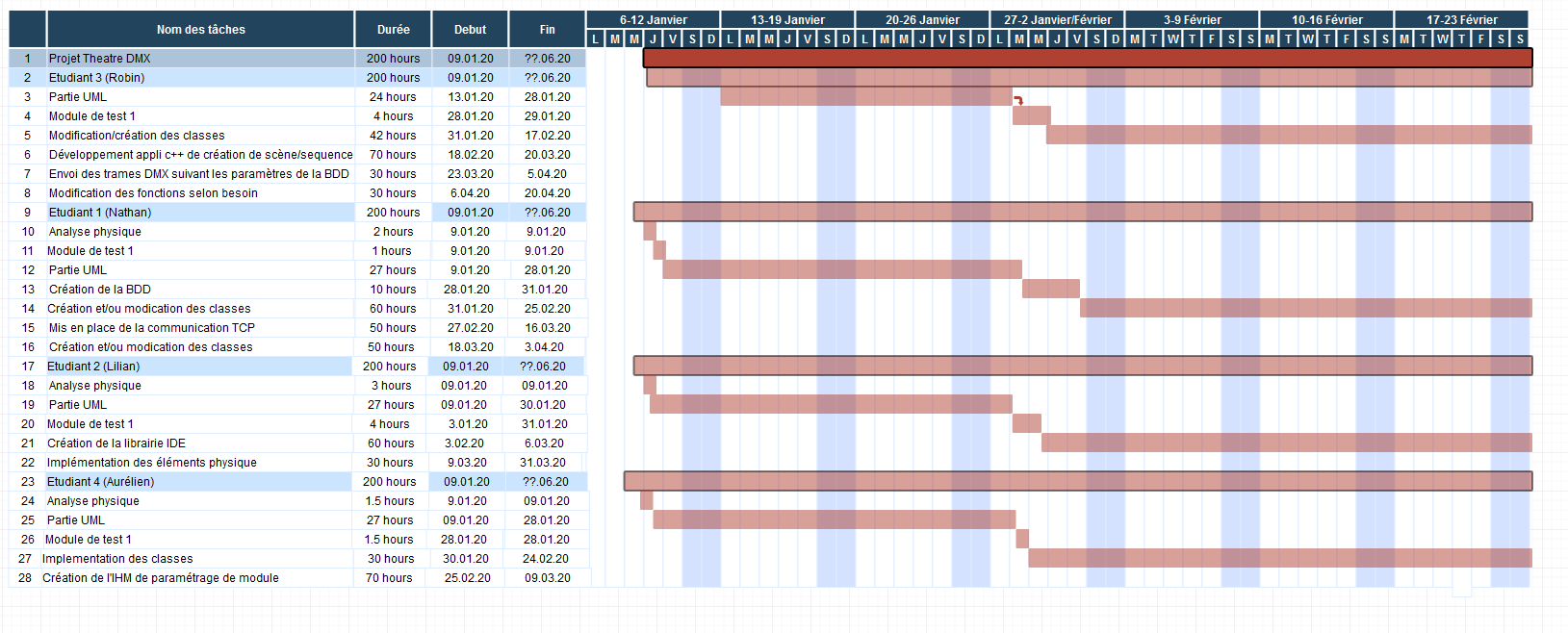
Afin de bien faire le projet, nous avons mis en place une organisation qui permettra de gérer notre projet, nos tâches...

Ce découpage se rapproche du modèle de projet en cycle en V

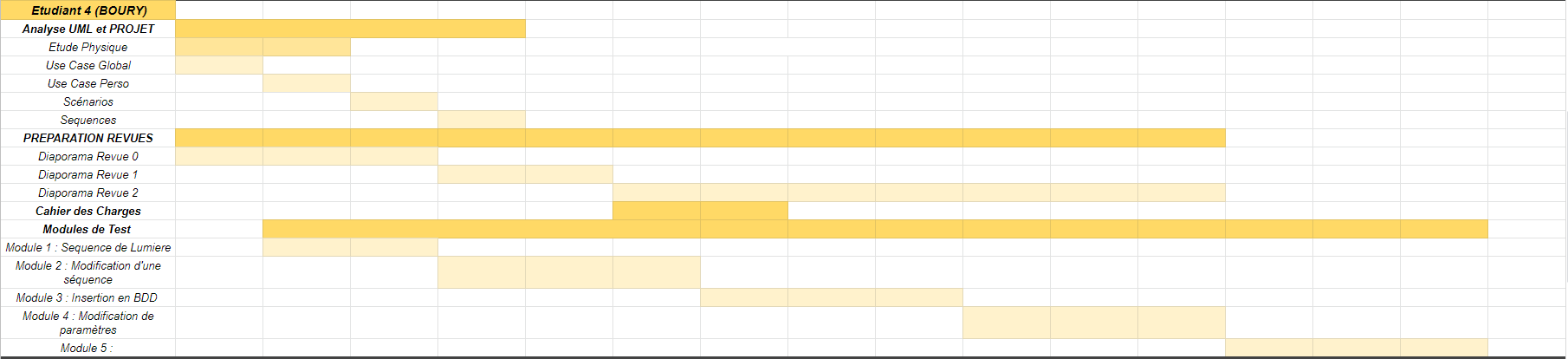
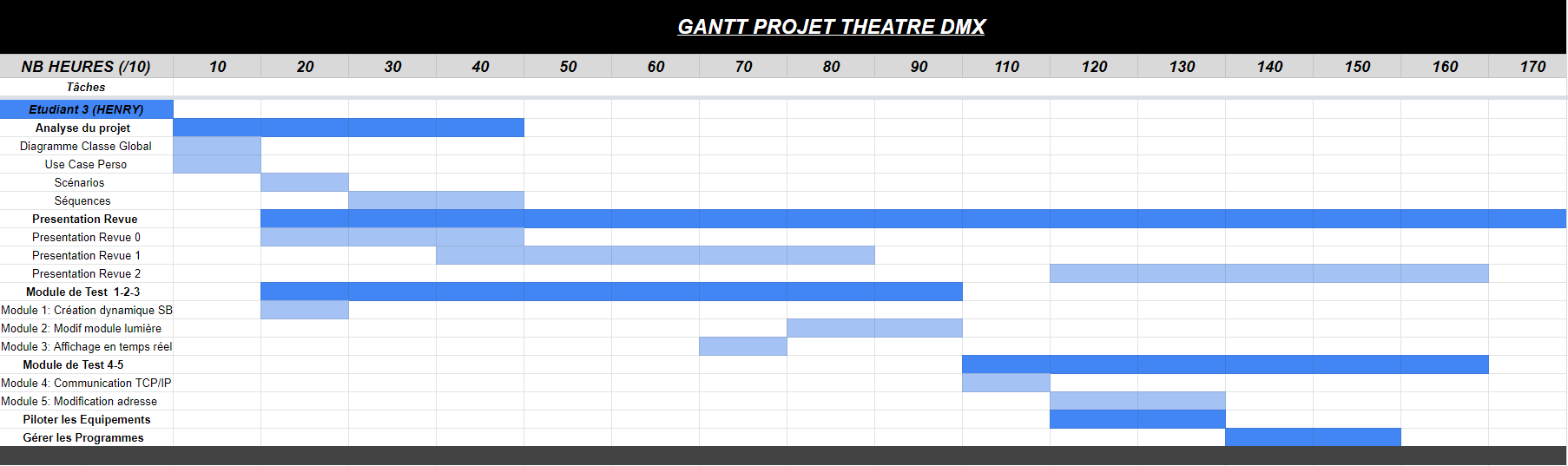


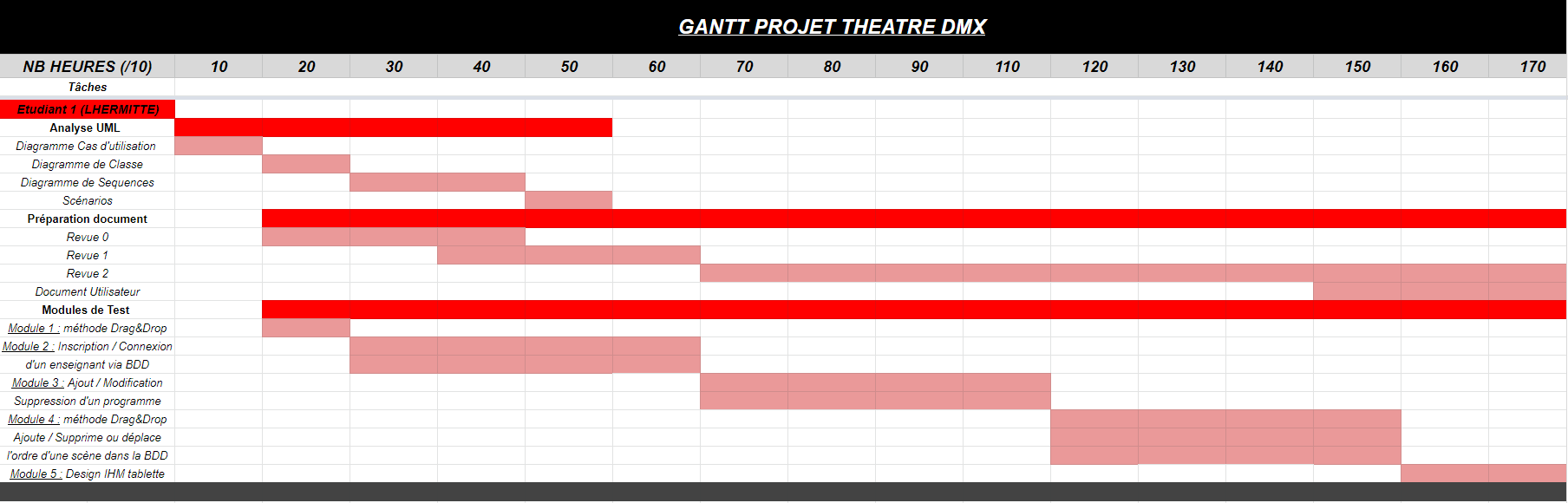
En parallèle de la méthode de projet nous devons travailler en équipe. Nous allons donc détailler ce qui nous a été conseillé et comment réellement nous avons travaillé.

### *GANTT Prévisionnel*

Voici la prévision fait après analyse du projet : le GANTT présenté ici est une version simplifiée pour le rendre plus lisible sur format papier. 

### *GANTT Réel*





### *Répartition des tâches*

Étudiant 1 (Vues) : Partie Web et IHM Mobile

Étudiant 2 (Système) : Partie C++ : faciliter le paramétrage des équipements

Étudiant 3 (Contrôleur) : Partie C++ : Paramétrer les équipements

Étudiant 4 (Modèle) : Conception console matérielle

## Organisation de l’équipe

Pour communiquer entre nous, nous utilisons la plateforme GitHub pour nos fichiers de code et la plateforme Discord pour communiquer entre nous à distance hors des heures projets en atelier. Nous avons une copie de nos fichiers sur Google drive.

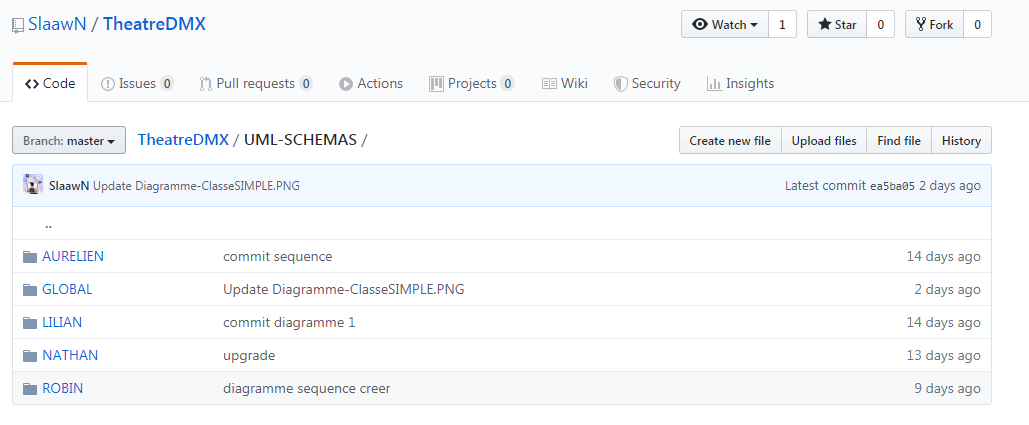
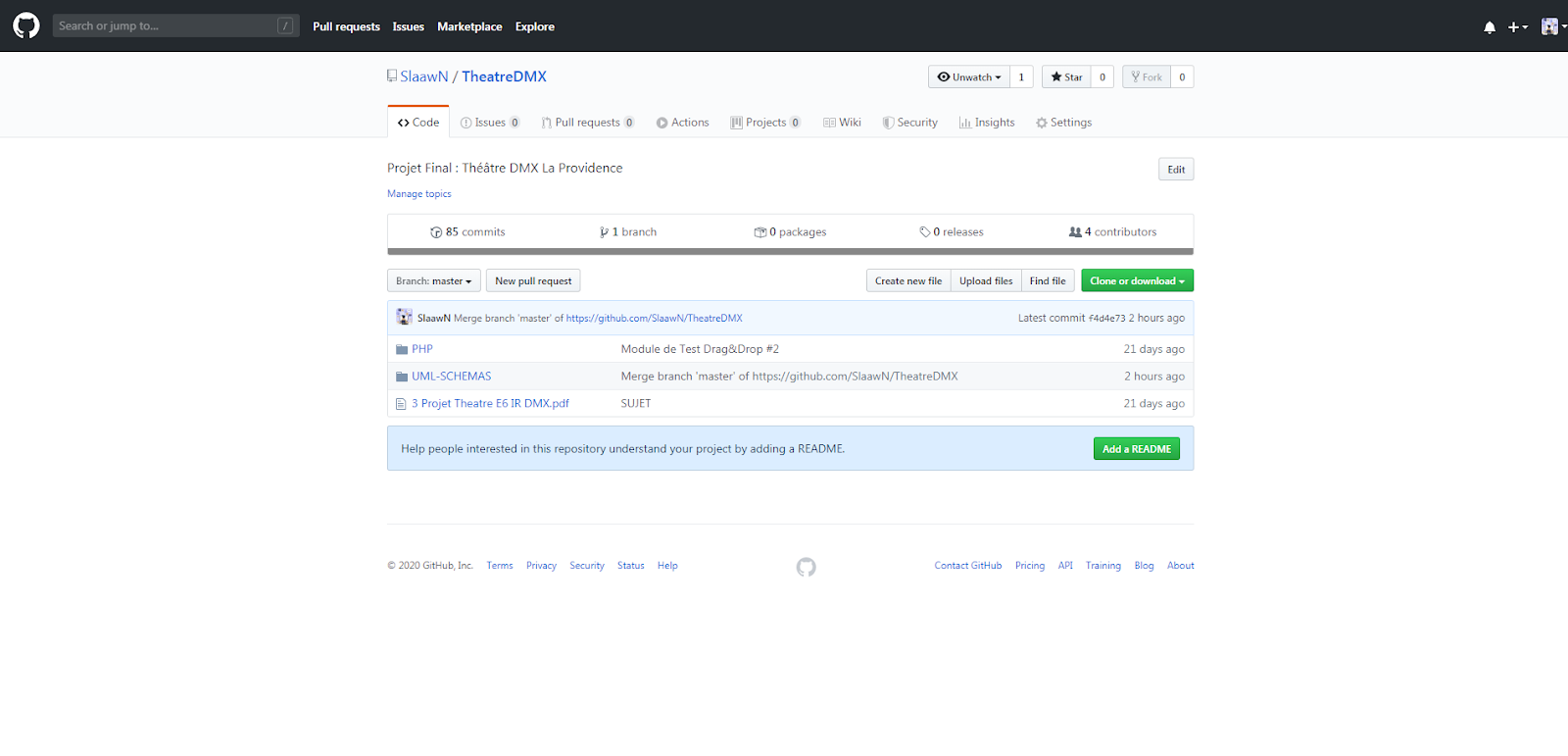
### 

### *Compte rendu d’activité (CRA) et Cahier de bord*



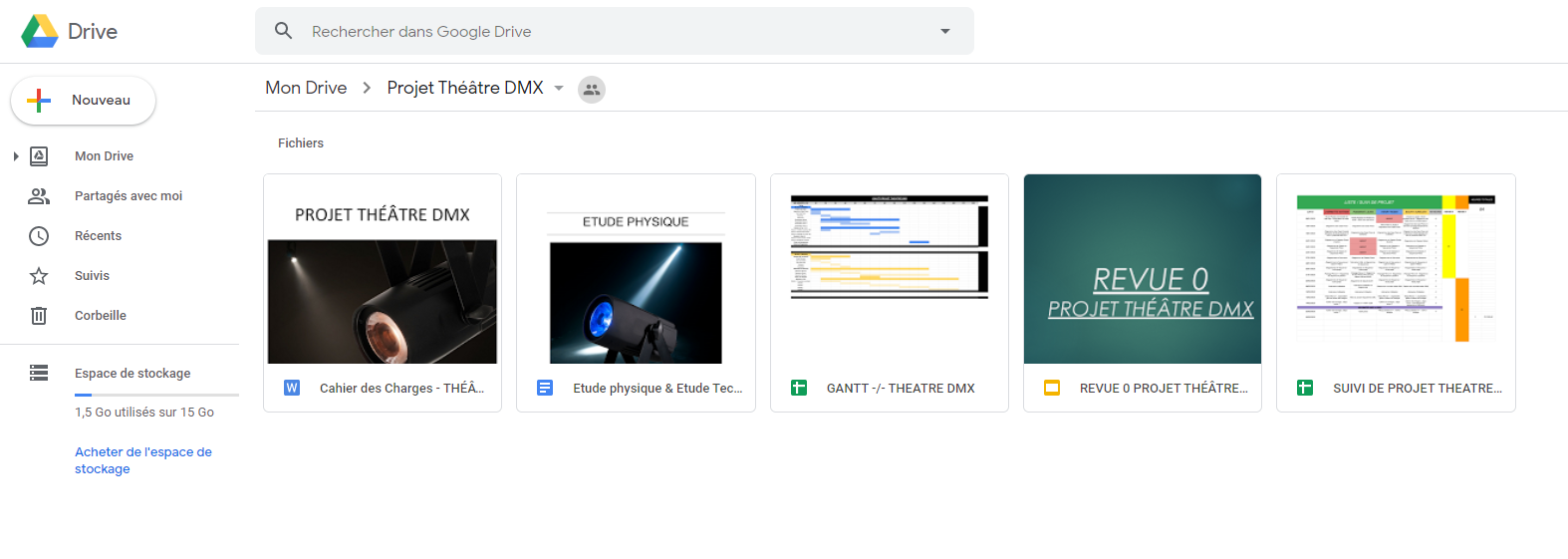
### *GITHUB et Versionning*

Pour travailler en collaboration nous avons utilisé le logiciel de Versionning « ***GITHUB DESKTOP*** » Ainsi que la Plateforme d’hébergement Git-hub.com. Sur nos PC de développement nous avons une version de notre code source avec nos différents "Commit" une fois qu’une fonctionnalité est opérationnelle nous la poussons sur le site hebergement en faisant un "Push" pour que tous les membres du projet puissent avoir la fonctionnalité.



A chaque instant sur l’hébergeur il y a la version la plus à jour de notre projet.

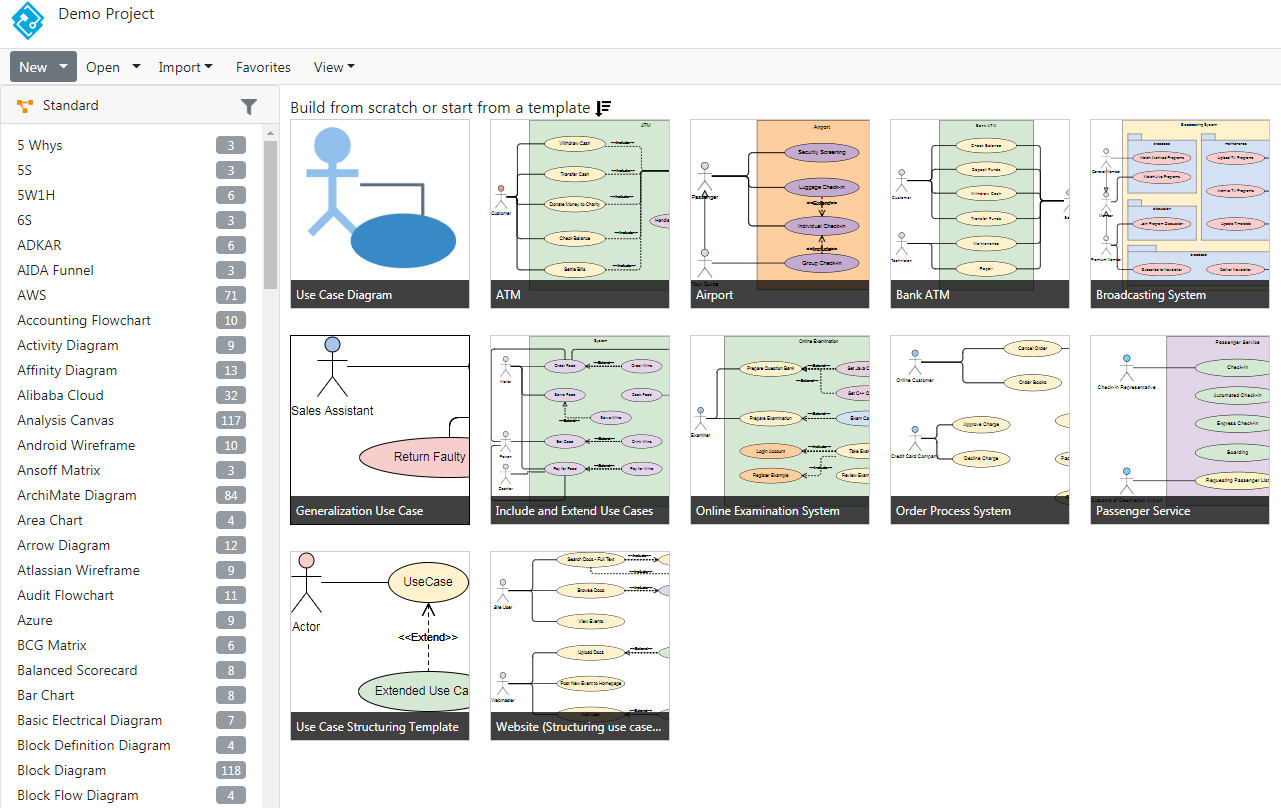
Nous utilisons aussi ***Google Drive*** qui permet de réaliser et stocker nos rapports ainsi que nos diaporamas. Google Drive nous permet de travailler à plusieurs sur un seul fichier. 



### *Logiciel d’analyse et de développement*

***Pour réaliser nos diagrammes nous avons utilisé l’outil en ligne Visual Paradigm Online :***

<https://online.visual-paradigm.com/>



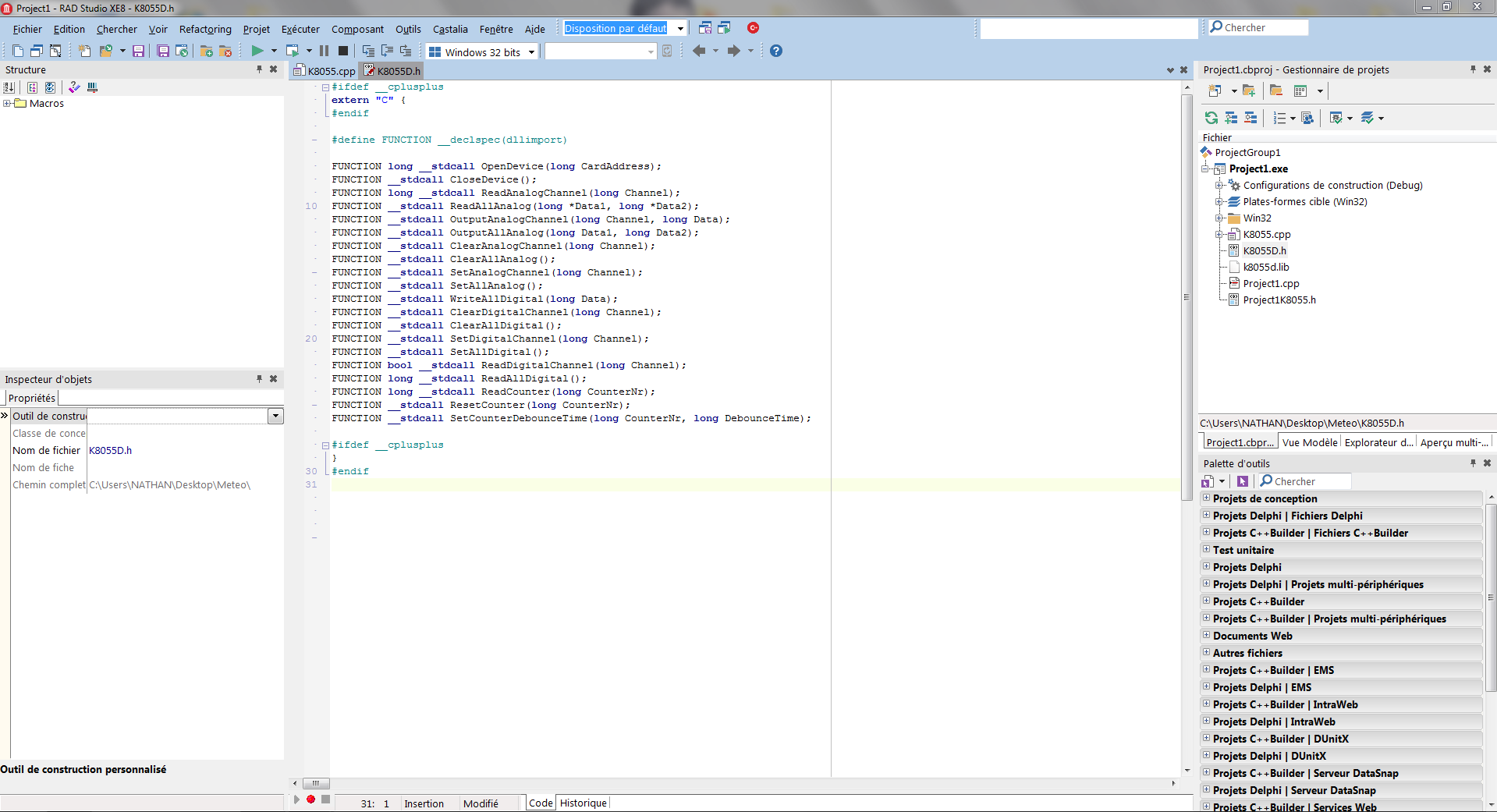
Cette application en ligne permet de retrouver nos diagrammes sur n’importe qu’elle machine en mode édition. Pas besoin d’installer d’application lourde. L’espace gratuit est suffisant pour nos besoins du projet.

Nous l’avons choisie car elle est simple d’utilisation et permet un stockage soit en ligne , soit sur Google Drive ou encore permet de l’enregistrer en format .vpd.

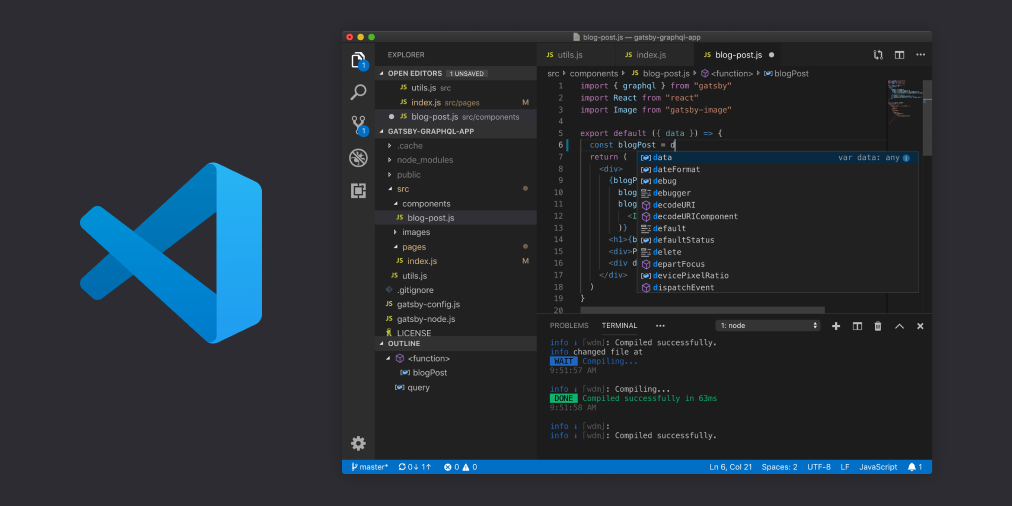
***Pour réaliser notre code C++ nous avons utilisé l’outil Embarcadero XE8 :***

<https://www.embarcadero.com/fr/>

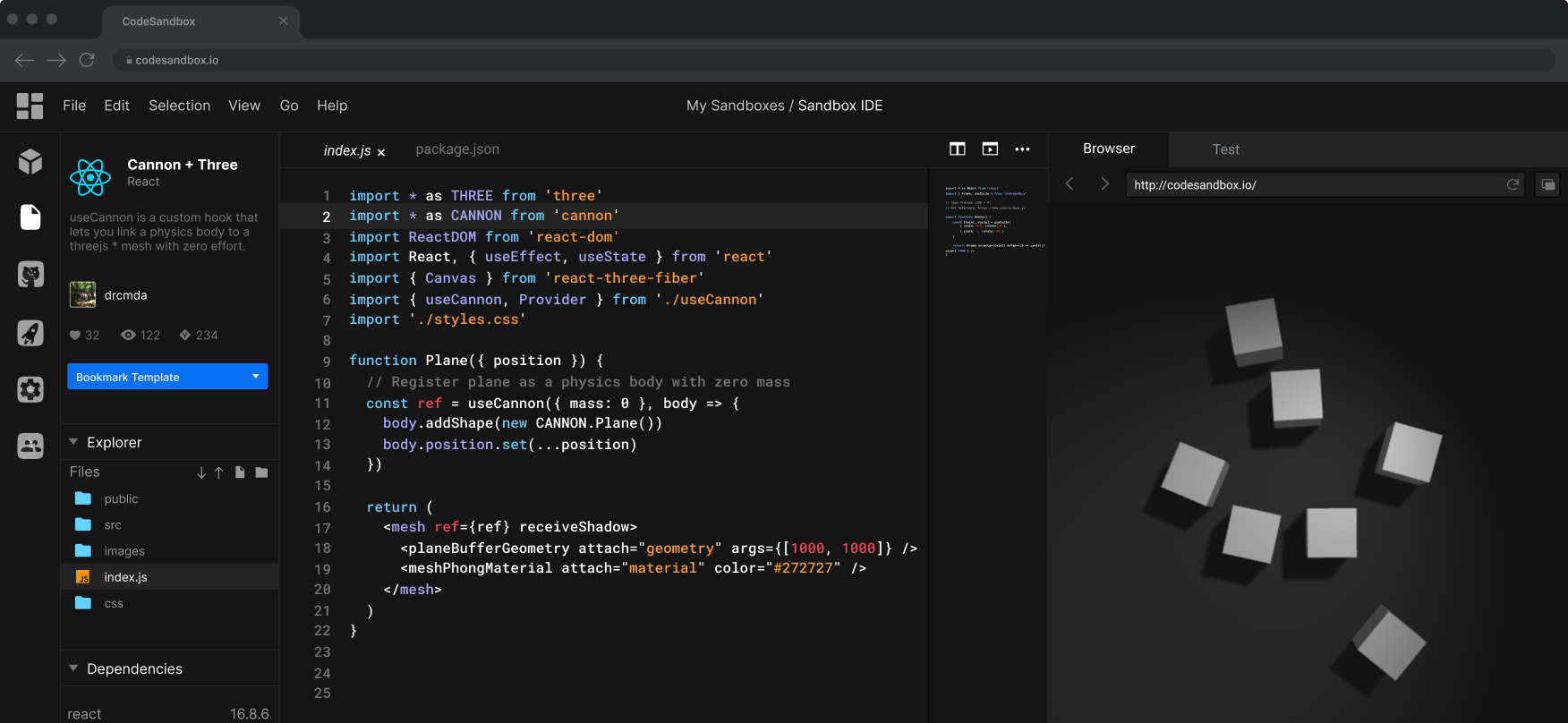




Cette application permet de pouvoir coder en C++ et de pouvoir le compiler. Grâce à cette application on peut contrôler des équipements tels que des GPS, des stations météo etc... Embarcadero permet aussi de faire des IHM C++ pour piloter toute sorte d’équipement.

***Pour réaliser notre code Web (Php / React) nous avons utilisé l’outil CodeSandbox ainsi que Visual Studio Code :*** 

Virtual Studio Code permet de pouvoir programmer dans n’importe quelle langage grâce à des extensions. Ce logiciel fournit une large liberté de programmation.



CodeSandbox est un logiciel en ligne qui permet de programmer à l’aide de la librairie React qui permet de créer des interfaces interactives. Ce logiciel permet également d’avoir un aperçu direct.

### 

### 

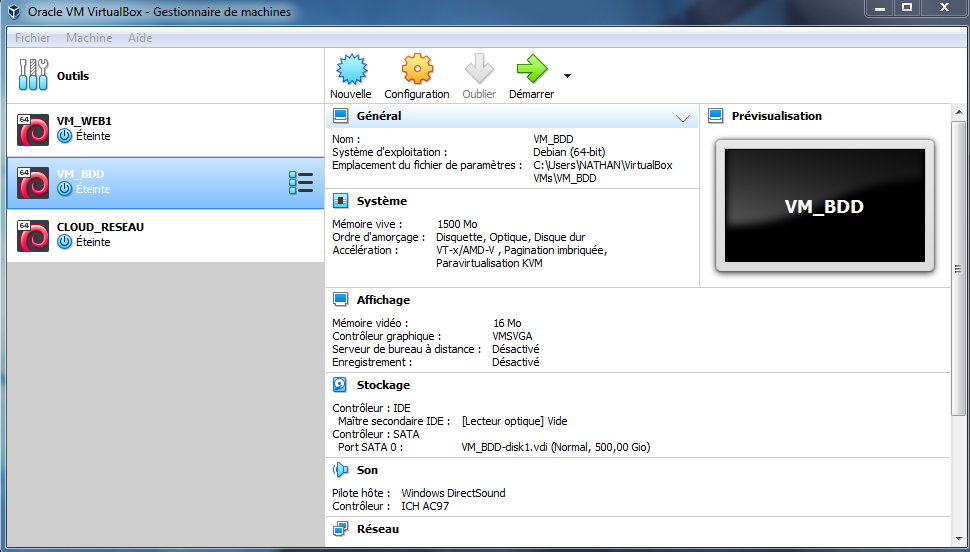
### 

### *Maquettage et Prototype*

**Partie Centrale de Gestion (IHM Web)**

Dans ce projet nous allons utiliser des Machines Virtuelles qui permettent d’héberger notre Base De Données ainsi que l'IHM WEB.

On utilise un serveur Apache/MySQL (PHP/TCP).



Le serveur Mysql et le serveur Web sont pour le moment simulé pour le prototype sur une VM de type Debian. Avec Apache et Maria DB de configuré.

## Choix technique et Étude Physique

### *Choix du boitier DMX & de la lampe*

On a choisi le boitier Enttec DMX USB Pro Interface

Ce boitier à de multiple qualités :

1. A un bon retour client (4.8/5 sur amazon)
2. Est compatible avec Windows et Linux,
3. Possède 512 canaux,
4. Utilise le protocole DMX 512,
5. Peut-être utiliser sur un logiciel fourni par le constructeur pour le contrôler,
6. Est moins cher que les autres modèles (155 €).

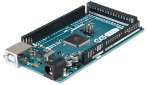
On a choisi ce Saber Spot RGBW



car :

1. C’est une LED RGBW ,
2. Elle a un changement de couleur fluide ,
3. Elle possède 12 canaux DMX différents avec plusieurs modes configurables,
4. La LED à une longue durée de vie ( 10K heures),
5. Même prix ou moins cher que les autres Saber Spot (115,00) €.

### *Choix des composants de la console matérielle*

***Carte programmable :*** 

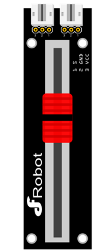
**Rôle :** Permettre le pilotage et l’utilisation des composants externes (potentiomètres, écran OLED, joystick)

**Choix :** ARDUINO MEGA 2560

**Avantages :** Performant, simple d’utilisation, nombre élevé de ports

**Tarif :** + ou - 40 €

***Potentiomètres linéaires :***

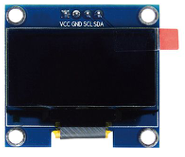


Rôle : Modifier une valeure (entre 0-1024) récupéré par l’arduino et envoyé à l’IHM

Choix : capteur linéaire gravity DFROBOT

Avantages : Tarif, simple d’utilisation

Tarif : + ou - 5 €/pièce

***Ecran monochrome OLED*** 

Rôle : Afficher un retour du paramétrage ou de la scène en cour de modification.

Choix : Afficheur OLED 1,3'' I2C TF051

Avantages : tarif, qualité d’affichage, simplicité de câblage

Tarif : + ou - 10€

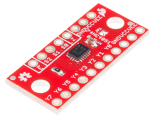
***Joystick Analogique***

Rôle : Choisir la scène / séquence que l’on souhaite modifier

Choix : JOYSTICK ANALOGIQUE GT1079

Avantages : tarif, plusieurs fonctionnalités, remplace plusieurs boutons

Tarif : + ou - 3€

***Module Multiplexeur Analogique***

Rôle : Connecter plusieurs capteurs à une même entrée analogique sur l’Arduino

Choix : Multiplexeur analogique BOB-13906

Avantages : tarif, possibilité de souder sur le composant afin d’éviter les faux contacts et les pertes de données

Tarif : + ou - 3€

## Recette

|  |  |  |
| --- | --- | --- |
| **Fonctionnalités du système + Nom du test pour validation** | **ETAT (OK ou NOK)** | **Commentaires** |
| Inscription d’un enseignant sur l’IHM WEB  +TESTInscription |  |  |
| Connexion d’un enseignant sur l’IHM WEB  +TESTConnexion |  |  |
| Gérer des programmes de lumière  +TESTProgramme |  |  |
| Séquence de Lumière aléatoire  + TESTSequence |  |  |
| Modification de Séquence aléatoire à l’aide de scrollbars  +TESTModif |  |  |
| Insertion en BDD des paramètres  +TESTInsertion |  |  |

## Tests d’intégration du prototype

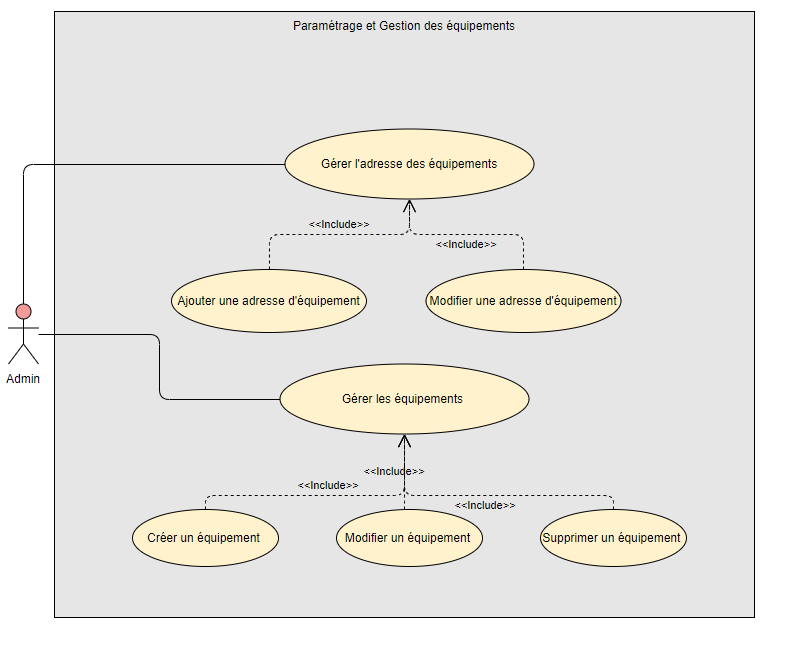
## Avancement et Conclusion

Ça avance plutôt bien mais la suite sera importante dans le projet car il faudra rassembler tout nos modules de test en 1.

# **PARTIE 2 : Présentation Etudiant 4 / BOURY**

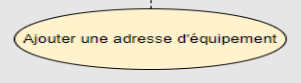
## Analyse Fonctionnelle

### Diagramme de cas d’utilisation

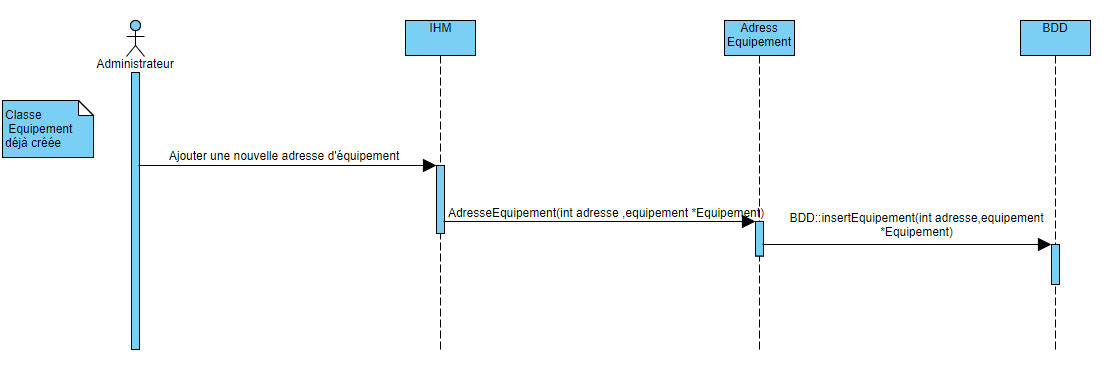


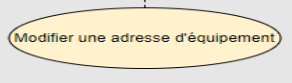
Ma partie consiste à Gérer et paramétrer les équipements (adresses,…) du DMX.

### Diagrammes de Séquence

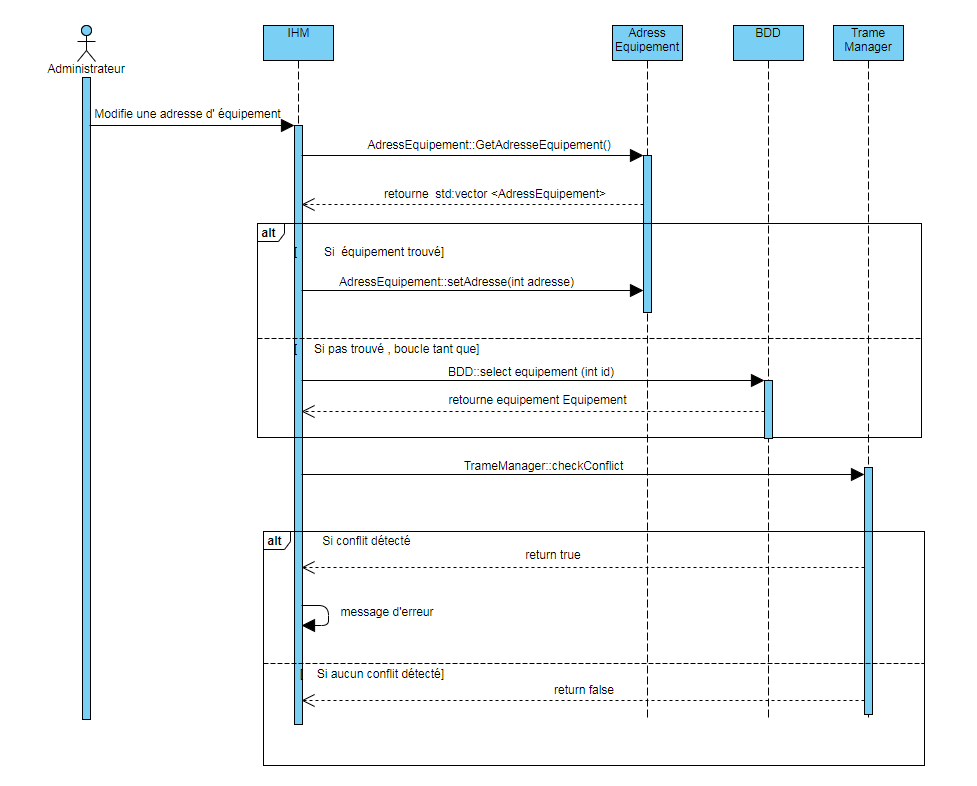


#### Séquence : Ajouter une adresse d’équipement

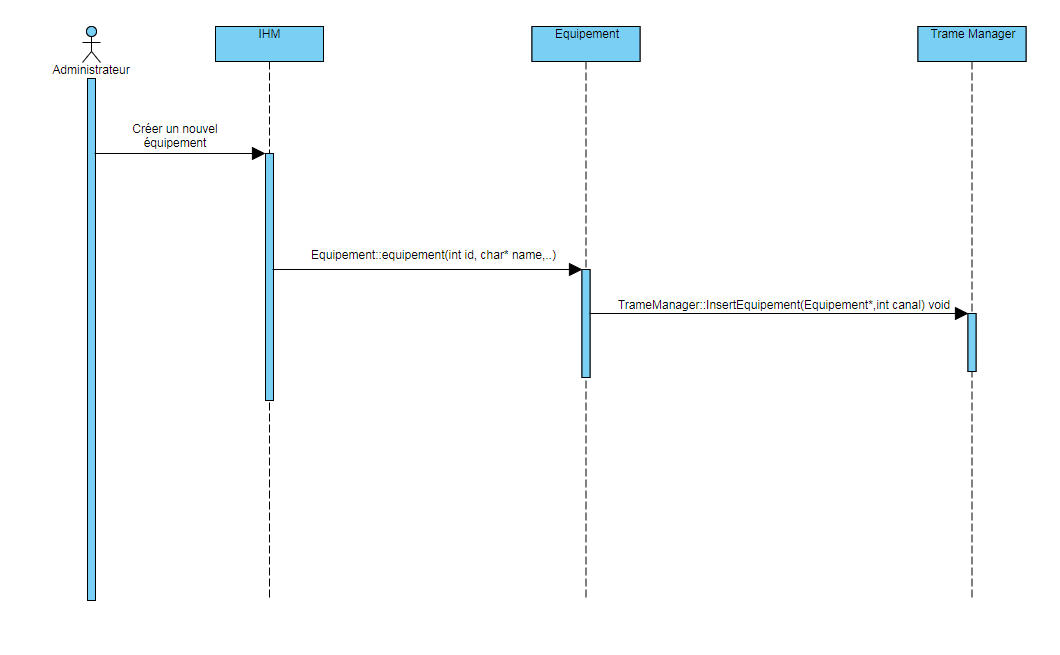




#### Séquence : Modifier une adresse d’équipement

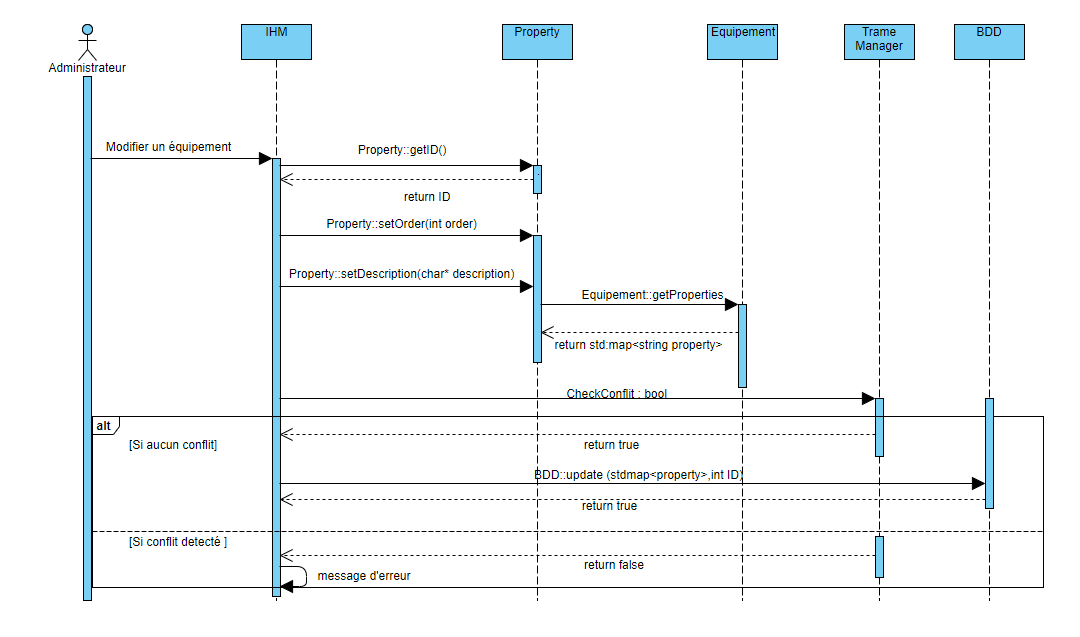


#### Séquence : Créer un équipement

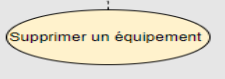


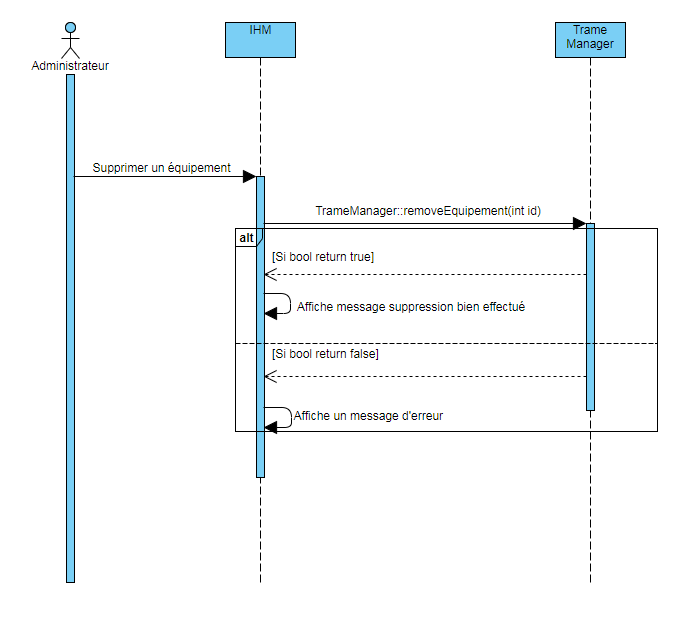
#### 

#### Séquence : Modifier un équipement

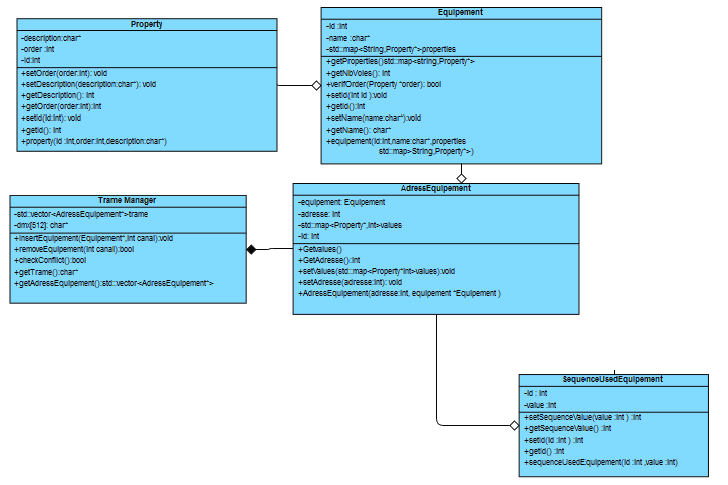


#### Séquence : Supprimer un équipement





### Diagramme de classe / Classes utilisées



### Modules de Test

#### Module de Test n°1 : Séquence de lumière

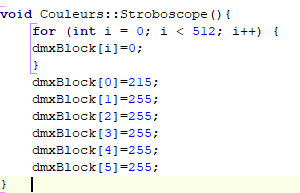
Mon 1er module de test représente une séquence de lumière.

Une séquence est un défilement ou la succession de scène de lumière.

Nous verrons par la suite que j’ai fait une séquence stroboscope.

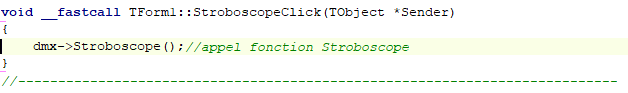
J’ai développé ce 1er module de Test à l’aide d’EMBARCADERO, ainsi lors d’un TP précédent, j’ai déjà fait ce genre de module.

C’est donc pourquoi j’ai gardé la même méthode.



Ce code ci-contre montre la méthode : On stocke des valeurs allant de 0 à 255, celles-ci correspondent à une action sur la lampe.

On définit donc les valeurs du tableau à 255(et correspondent à la séquence de stroboscope).



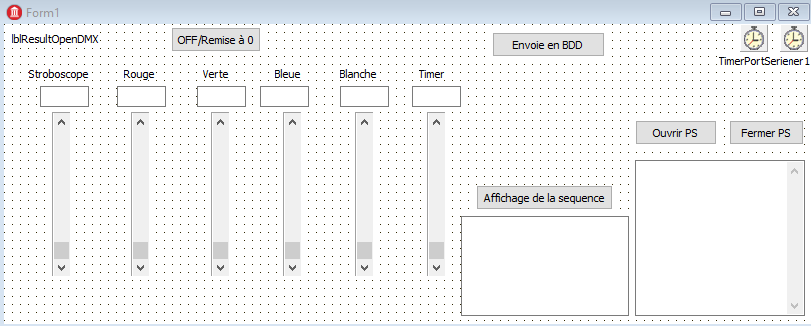
Ensuite dans un autre .cpp , au click d’un bouton sur l’IHM , on appelle la fonction nommée Stroboscope qui est expliquée au-dessus.

*Une vidéo est disponible dans le Diaporama de la revue pour voir le résultat.*

#### Module de Test n°2 : Modification de Séquence de lumière

Ce second module de test est un module permettant de modifier les valeurs expliqués au module n°1.

Grâce à une IHM , nous pouvons modifier les valeurs à l’aide de scroll bars, et donc permet de voir en temps réel les modifications apportées.

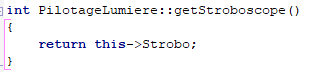


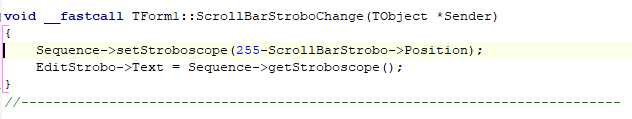
Cette IHM est simple de compréhension et donc simple d’utilisation.

Plus on monte la scroll bar vers le haut, plus la valeur se rapprochera de 255.

Ainsi les valeurs que nous mettons, s’affichent dans le « carré » en dessous de chaque scène.

Ainsi le paramétrage des valeurs se fait donc en direct, et non dans un tableau comme dans le module n°1.





Il est aussi possible via le bouton « Affichage de la séquence » de voir toutes les valeurs que nous avons paramétrés.

Le Bouton « OFF/Remise à0 » remet toutes les scroll bars, et donc toutes les valeurs à 0.

*Une vidéo est disponible dans le Diaporama de la revue pour voir le résultat.*