

---

# **Documentation Technique de Caiman**

***Version 1.1.0***

**Lorenzo Bauduccio**

**juin 08, 2021**



---

## table des matières

---

<b>1</b>	<b>Résumé</b>	<b>1</b>
<b>2</b>	<b>Abstract</b>	<b>3</b>
<b>3</b>	<b>Remerciement</b>	<b>5</b>
<b>4</b>	<b>Légalité du projet</b>	<b>7</b>
4.1	Caiman . . . . .	7
4.2	Utilisation d'émulateurs . . . . .	7
<b>5</b>	<b>Installation de Caiman utilisateur</b>	<b>9</b>
<b>6</b>	<b>Installation de Caiman serveur</b>	<b>11</b>
<b>7</b>	<b>Introduction</b>	<b>13</b>
7.1	Contexte . . . . .	13
7.2	Motivations . . . . .	13
<b>8</b>	<b>Cahier des charges</b>	<b>15</b>
8.1	Objectifs . . . . .	15
8.2	Spécification . . . . .	15
8.3	Limites du Projet . . . . .	17
8.4	Calendrier . . . . .	17
8.5	Matériel . . . . .	18
<b>9</b>	<b>Étude d'opportunité</b>	<b>19</b>
9.1	RetroPie . . . . .	19
9.2	RetroArch . . . . .	20
9.3	RomStation . . . . .	21
9.4	Steam . . . . .	22
9.5	Conclusion de l'analyse de l'existant . . . . .	22
<b>10</b>	<b>Analyse fonctionnelle</b>	<b>23</b>
10.1	Base de données . . . . .	23

<b>11 Site web Caiman</b>	<b>27</b>
11.1 Création de compte . . . . .	27
11.2 connexion . . . . .	27
11.3 modification des informations d'un utilisateur . . . . .	27
11.4 affichage des jeux . . . . .	28
11.5 affichage d'un profil utilisateur . . . . .	28
11.6 ajout d'un jeu à la base de données / sur le Bunker pour le fichier .ISO . . . . .	28
11.7 modification d'un jeu . . . . .	28
11.8 Administration . . . . .	29
11.9 Téléchargement . . . . .	29
11.10 Fonctionnalités manquante . . . . .	29
<b>12 API Caiman</b>	<b>31</b>
12.1 Structure de l'API . . . . .	32
12.2 Categories . . . . .	33
12.3 GET . . . . .	33
12.4 Games . . . . .	33
12.5 GET . . . . .	33
12.6 GET(?byName) . . . . .	33
12.7 GET(?byCategory) . . . . .	34
12.8 GET(?byFavoriteUser) . . . . .	34
12.9 GET(?byUserTime) . . . . .	34
12.10 GET(?gameFileName) . . . . .	35
12.11 GET(?gameConsole) . . . . .	35
12.12 GET(?idGame&apiKey) . . . . .	35
12.13 GET(?idGameTime&idUser) . . . . .	35
12.14 GET(?idEmulator&idUser&apiKey) . . . . .	35
12.15 POST(?idEmulator&idUser&apiKey) . . . . .	36
12.16 POST(idGameAdd&idUser) . . . . .	36
12.17 POST(idGameRemove&idUser) . . . . .	36
12.18 POST(idGameCheck&idUser) . . . . .	36
12.19 POST(idGameTimeAdd&idUser) . . . . .	36
12.20 Users . . . . .	36
12.21 GET(sans paramètres) . . . . .	36
12.22 POST(avec apitoken) . . . . .	37
12.23 User/connection . . . . .	37
12.24 POST(?username, password) . . . . .	37
12.25 POST(caimanToken) . . . . .	37
<b>13 Application Caiman C#</b>	<b>39</b>
13.1 Connexion . . . . .	39
13.2 Inscription . . . . .	40
13.3 Téléchargement de jeu . . . . .	40
13.4 Lancement d'un jeu . . . . .	41
13.5 Synchronisation des sauvegarde entre le pc client et le Bunker . . . . .	41
13.6 Modification de la configuration utilisateur . . . . .	41
13.7 Application de la configuration utilisateur . . . . .	43
13.8 Ajout/supresion de jeu en favoris . . . . .	43

13.9	Affichage de jeux par catégories . . . . .	44
13.10	Nombre d'heures de jeu . . . . .	45
13.11	Gestion des manettes . . . . .	45
<b>14</b>	<b>Interface utilisateur utilisable à la manette</b>	<b>47</b>
14.1	Réception des input des manettes connecté . . . . .	47
14.2	Transformation des input de la manette en événement . . . . .	47
14.3	Structure de l'affichage . . . . .	48
14.4	Déplacement dans un panel de l'application . . . . .	48
14.5	Déplacement de panel en panel . . . . .	48
<b>15</b>	<b>Logbook</b>	<b>51</b>
15.1	19.04.2021 . . . . .	51
15.2	20.04.2021 . . . . .	53
15.3	21.04.2021 . . . . .	53
15.4	22.04.2021 . . . . .	54
15.5	23.04.2021 . . . . .	55
15.6	26.04.2021 . . . . .	56
15.7	27.04.2021 . . . . .	58
15.8	28.04.2021 . . . . .	59
15.9	29.04.2021 . . . . .	59
15.10	30.04.2021 . . . . .	60
15.11	notes pour le premier rendu . . . . .	61
15.12	03.05.2021 . . . . .	61
15.13	04.05.2021 . . . . .	65
15.14	05.05.2021 . . . . .	65
15.15	06.05.2021 . . . . .	67
15.16	07.05.2021 . . . . .	68
15.17	10.05.2021 . . . . .	69
15.18	11.05.2021 . . . . .	70
15.19	12.05.2021 . . . . .	71
15.20	13.05.2021 . . . . .	72
15.21	14.05.2021 . . . . .	72
15.22	17.05.2021 . . . . .	73
15.23	18.05.2021 . . . . .	74
15.24	19.05.2021 . . . . .	74
15.25	20.05.2021 . . . . .	75
15.26	21.05.2021 . . . . .	76
15.27	24.05.2021 . . . . .	77
15.28	25.05.2021 . . . . .	77
15.29	26.05.2021 . . . . .	78
15.30	27.05.2021 . . . . .	79
15.31	28.05.2021 . . . . .	79
15.32	30.05.2021 . . . . .	80
15.33	31.05.2021 . . . . .	81
15.34	01.06.2021 . . . . .	81
15.35	02.06.2021 . . . . .	81
15.36	03.06.2021 . . . . .	82

15.37	04.06.2021	.....	82
15.38	07.06.2021	.....	83

# CHAPITRE 1

---

## Résumé

---

Ce document décrit le processus de conception de mon projet Caiman. Caiman est inspiré par des applications comme Retropie, RetroArch ou Steam. RetroPie est une distribution pour raspberry comprenant des émulateurs pour d'anciennes consoles de jeu. RetroArch est lui une application servant de frontend pour émulateur, RetroArch est disponible sur plusieurs plateformes(Windows, macOS, Xbox).

Caiman est un projet comprenant deux parties distinctes. La première partie consiste en un site web PHP permettant de se créer un compte et de s'identifier. Le site web permet de consulter des informations sur les jeux présent sur le store ainsi que de pouvoir administrer les sites et le store.

La deuxième partie du projet est une application C# inspirée par le mode Big Picture de steam. Caiman sert de frontend pour différents émulateurs (Dolphin, PCSX2,etc). L'application permet de télécharger les différents jeux qui ont été ajoutés au store prévu pour le projet. Les jeux pris en charge sont ajoutés par des administrateurs via le site internet de Caiman. L'application permet de synchroniser les sauvegardes de l'utilisateur et cela peut importe sur quel pc, il lance l'application.



# CHAPITRE 2

---

## Abstract

---

This document describes the design process of my project Caiman. Caiman is inspired by applications like Retropie, RetroArch or Steam. RetroPie is a raspberry distribution including emulators for old game consoles. RetroArch is a frontend application for emulators, RetroArch is available on several platforms (Windows, macOS, Xbox).

Caiman is a project with two distinct parts. The first part consists of a PHP website allowing the visitor to create an account and to identify himself. The website allows you to consult information about the games on the store and to administer the site and the store.

The second part of the project is a C# application inspired by the Big Picture mode of Steam. Caiman is used as a frontend for different emulators (Dolphin, PCSX2, etc). The application allows you to download the different games that have been added to the store provided for the project. Supported games are added by administrators via the Caiman website. The application allows synchronizing the user's save, no matter on which PC the user runs the application.



# CHAPITRE 3

---

## Remerciement

---

Pour commencer, je tenais en particulier à remercier les personnes qui ont pris le temps de me faire des retours sur Caiman. Sans les retours que j'ai reçus de leur part, Caiman ne serait pas aussi bien peaufiné qu'il l'est actuellement. Ensuite, j'aimerais remercier M.Maréchal et M.Schmid pour leurs conseils et leurs retours tout le long de mon travail de diplôme. Pour finir, j'aimerais remercier les autres élèves de ma classe avec qui j'ai passé de très bon moment et qui m'ont aussi donné des conseils pertinents. Je tiens en particulier à remercier M.Borel-Jaquet qui m'a énormément aidé avec la structure de l'API.



# CHAPITRE 4

---

## Légalité du projet

---

### 4.1 Caiman

Je pense qu'il est nécessaire d'aborder le sujet de la légalité de Caiman. Selon le droit Suisse, il n'est pas légal de mettre à disposition des fichiers sous droit d'auteur. Il est par contre légal de télécharger des films, séries ou jeux à condition de les utiliser dans un but privé. Caiman ne répondant pas à ces descriptions l'utilisation de l'application n'est pas conforme au droit Suisse.

### 4.2 Utilisation d'émulateurs

L'utilisation d'émulateurs n'est pas illégale en soi, en sachant que le code des émulateurs n'est pas la propriété de Sony ou de Nintendo. Par contre, il y a un petit point technique à détailler. L'émulateur PCSX2 doit pour fonctionner utiliser un BIOS de console de Playstation 2. Le BIOS de la Playstation 2 étant soumis à des droits d'auteur, il n'est en théorie pas légal de le distribuer en Suisse.

Je tiens donc à spécifier que Caiman est développé dans un seul but pédagogique. Il n'a aucune vocation à être distribué après la fin de mon travail de diplôme. L'application sera supprimée du serveur de l'école à la fin de mon travail. Par contre, il serait légal de distribuer le projet si je supprime la fonctionnalité de téléchargement de jeu.



# CHAPITRE 5

---

## Installation de Caiman utilisateur

---

Pour utiliser Caiman, il faut le télécharger depuis [caiman.cfpt.info](http://caiman.cfpt.info). Quand on télécharge l'application, l'utilisateur télécharge un fichier caiman.zip contenant l'application ainsi que les émulateurs, il n'y a donc rien d'autre à télécharger.

Quand l'utilisateur décomprime le fichier caiman.zip, il se retrouve avec un dossier Caiman contenant deux dossiers. Un contenant les émulateurs et un autre contenant les fichiers de Caiman.



Pour pouvoir exécuter Caiman, il faut se rendre dans le dossier “\bin\Debug\” et ensuite l'utilisateur va trouver le fichier “Caiman.exe”. Ce fichier permet de lancer l'application.

Caiman est donc une application “portable”, c'est-à- dire qu'elle ne nécessite pas d'une installation pour pouvoir être lancée.



# CHAPITRE 6

---

## Installation de Caiman serveur

---

Pour pouvoir déployer le site et l'API de caiman, il faut que le serveur ait les services suivant :

- Apache2
- PHP 7.4 (minimum)
- mysql

Pour la base de données, il faut l'importer dans mysql sans faire de manipulation particulière.

Le site web et l'api doivent se trouver dans le même dossier, en sachant qu'il a des chemins relatifs entre les deux dossiers. ils ne peuvent donc pas être séparés pour l'instant.

Pour pouvoir uploader des jeux, il faut augmenter la taille que le PHP peut recevoir par formulaire, il faut également le configurer pour recevoir minimum 8GB. Pour finir, Il faut autoriser l'écriture dans les dossiers des sauvegardes et des jeux.



# CHAPITRE 7

---

## Introduction

---

### 7.1 Contexte

Caiman a été réalisé dans le cadre du travail de diplôme de technicien deuxième année. La réussite du travail de diplôme est l'une des conditions pour recevoir son titre de technicien en informatique.

### 7.2 Motivations

Le choix de ce travail de diplôme découle de mon envie de faire un travail de diplôme qui serait intéressant, ludique et en lien avec les jeux vidéo. La réalisation d'un jeu vidéo étant trop compliquée pour un travail de diplôme, j'ai décidé de créer une application à mi-chemin entre un launcher de jeu comme Steam ou GOG et un gestionnaire de VM.

J'utilise depuis des années des émulateurs et je me heurte souvent aux mêmes problèmes, perte de sauvegarde dû à une réinstallation de windows, problème pour ripper mes jeux, téléchargement de chaque émulateur un par un. Donc, J'ai eu l'idée de créer Caiman pour pallier ces différents problèmes.

Un autre aspect important pour moi est la simplicité. Je voulais que l'utilisation de l'application soit le plus simple possible, une personne sans connaissance particulière en informatique doit pouvoir utiliser Caiman sans difficulté. Une autre spécification est que l'utilisation de Caiman puisse se faire à la manette. Pour créer une interface ergonomique, je me suis basé sur l'interface Big Picture de Steam qui a justement été créé pour être utilisé à la manette.



# CHAPITRE 8

---

## Cahier des charges

---

### 8.1 Objectifs

Caiman est une application pour Windows regroupant plusieurs émulateurs de console de jeu. L'utilisateur peut exécuter les jeux depuis Caiman.

Caiman a pour but d'être simple d'utilisation, aucune connaissance n'est requise pour l'utiliser. Caiman permet une utilisation complète à la manette ou au clavier/souris. L'avantage de Caiman est qu'il ne requiert aucune configuration. Caiman est utilisable instantanément par l'utilisateur, contrairement d'un émulateur traditionnel qui requiert une configuration relativement compliquée.

Le téléchargement des jeux se fait directement depuis Caiman, un serveur nommé le "Bunker" contient les fichiers des différents jeux.

Un site web permet de créer un compte pour accéder à l'application , de voir les informations des joueurs ainsi que de télécharger Caiman.

### 8.2 Spécification

#### 8.2.1 Emulateurs et contrôles

Les différents émulateurs qui permettent d'exécuter des jeux dans Caiman sont :

- PCSX2 (playstation 2)
- Dolphin (gamecube / wii)

L'utilisateur doit utiliser une manette de Xbox pour utiliser Caiman à la manette (une seule manette est requise pour jouer mais plusieurs peuvent être connectées, pour jouer en multijoueurs local).

Caiman est utilisable à la manette à l'exception des formulaires (création de compte, importation de jeux).

### 8.2.2 Création d'un compte utilisateur

L'utilisateur a l'obligation de créer un compte pour pouvoir utiliser Caiman.

L'utilisateur peut créer un compte directement depuis Caiman s'il n'en possède pas.

### 8.2.3 Interface

Dans l'interface, un système de catégories existe pour permettre de se retrouver plus aisément dans la liste des jeux à télécharger. Les différentes catégories sont créées par un administrateur.

Exemple de catégories :

- jeux Mario
- jeux Zelda
- jeux d'aventure
- jeux de réflexion
- jeux multijoueur
- jeux favoris
- etc..

L'utilisateur peut ajouter depuis Caiman des jeux favoris pour pouvoir les retrouver plus facilement. ( une catégorie "favoris" est visible contenant les jeux favoris de l'utilisateur authentifié )

### 8.2.4 Paramètre graphiques

L'utilisateur a la possibilité de modifier certains paramètres d'émulation :

- La définition des jeux (définition native, proche de 1080p, proche de 4K).
- La langue des machines.
- Si les jeux doivent être lancés en plein écran ou non.
- Format d'écran 16/9 ou 4/3.

### 8.2.5 Gestion des sauvegardes

Il existe plusieurs cas où les sauvegardes de l'utilisateur se synchronisent :

1. Le joueur sauvegarde sa partie directement dans un jeu depuis Caiman. La sauvegarde que l'utilisateur vient de créer est envoyée sur le Bunker et remplace la version présente.
2. Quand le joueur lance Caiman, une vérification est faite pour savoir si une version plus récente des sauvegardes du joueur sont présentes, si c'est le cas, elles sont téléchargées sur la machine du joueur.

### **8.2.6 Spécifications du “Bunker”**

Le Bunker contient les fichiers des différents jeux au format (.ISO).

Le Bunker contient les fichiers de sauvegardes des utilisateurs. Ces sauvegardes seront envoyées de la machine de l'utilisateur vers le Bunker quand l'utilisateur sauvegarde sa partie depuis un jeu.

Les données des utilisateurs (email, jeux favoris, heures de jeux) sont stockées dans une base de données.

### **8.2.7 Spécification site web (site présent dans le Bunker)**

Sur le site web, il est possible de créer un compte en spécifiant un mail et un mot de passe.

L'interface web permet de rechercher des utilisateurs pour voir leur profil.

L'affichage d'un profil permet de voir les jeux favoris d'un utilisateur ainsi que son nombre d'heures de jeu sur chaque jeu.

Le site web permet à un administrateur d'ajouter des jeux et de modifier les informations des jeux.

Le site web permet de modifier les informations de compte d'un utilisateur.

Un utilisateur peut faire la demande de réinitialiser son mot de passe.

### **8.2.8 Installation**

Caiman est téléchargeable depuis le site web. L'installateur contient les émulateurs, il n'y a donc rien à télécharger d'autre.

## **8.3 Limites du Projet**

Caiman sera limité aux jeux que j'ai pu ajouter moi même ainsi qu'aux émulateurs que j'aurais importé, la totalité des consoles de jeu ne sera donc pas prise en charge.

## **8.4 Calendrier**

Le début du travail de diplôme est fixé au lundi 19 avril et le rendu est fixé au 11 juin à 12h00.

Le travail est à exécuter soit en présentiel et en télétravail du lundi au vendredi et cela 8h00 par jour.

## **8.5 Matériel**

J'ai à ma disposition pour la réalisation de Caiman :

- Mon pc pour le développement
- Intel i7-7700 3.6 GHz
- 32 GB de ram
- GTX 1060 3GB
- Un serveur pour héberger le site et les fichiers du store
- Une manette de Xbox Series

# CHAPITRE 9

---

## Étude d'opportunité

---

### 9.1 RetroPie



RetroPie est une distribution Linux prévue pour Raspberry et PC. C'est un frontend pour jouer aux jeux d'arcade, d'anciennes consoles et de jeux pc. Retropie est à la différence de Caiman un OS à part entière et chaque jeu doit être ajouté indépendamment par l'utilisateur sur le Raspberry/PC.

## 9.2 RetroArch



RetroArch est un frontend pour émulateurs, moteur de jeu et media players. Il embarque un très grand nombre d'émulateurs et a l'avantage d'être disponible sur un très grand nombre de plateformes.

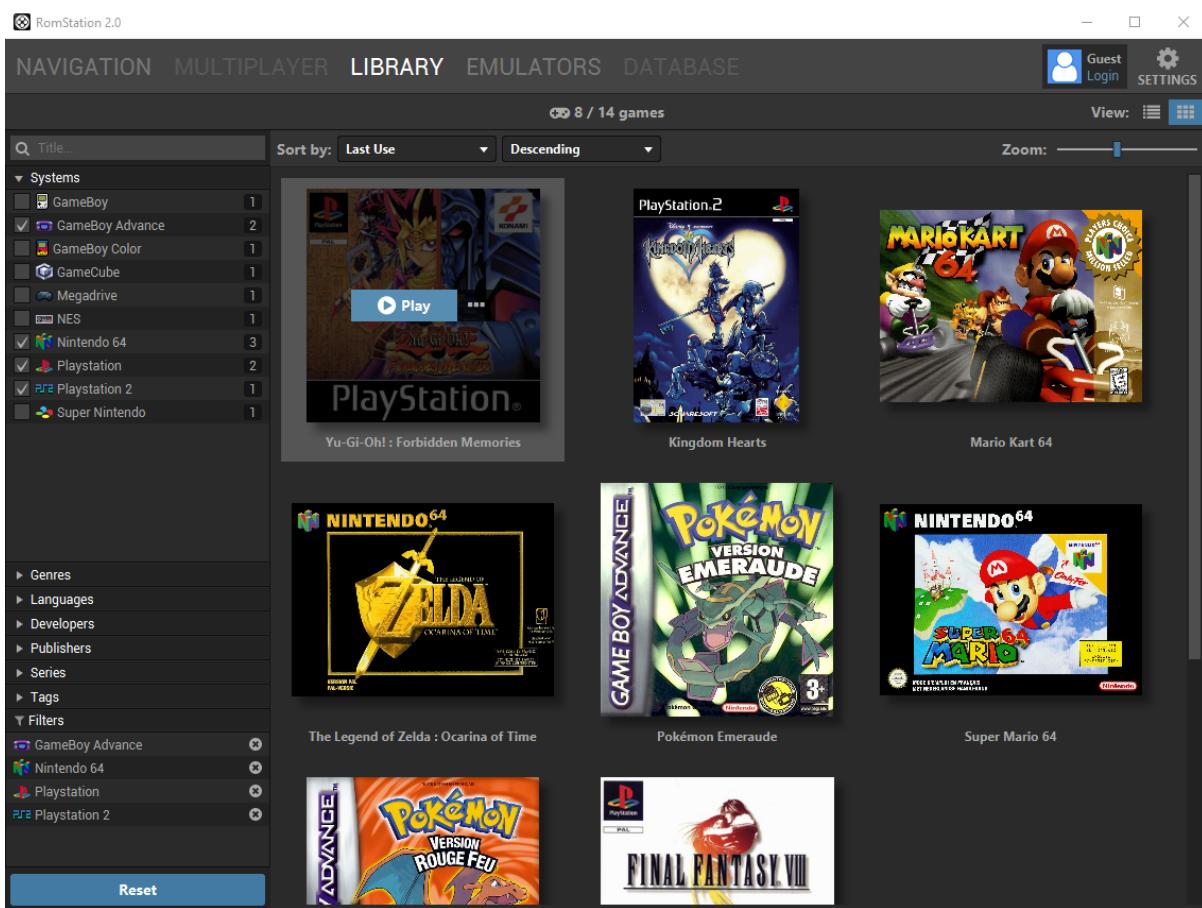
RetroArch possède des fonctionnalités très utiles comme des Shaders intégré, le jeux en réseau, le calcul du temps entre les frames, et d'autres. C'est sûrement le Frontend pour émulateur le plus complet disponible.

L'interface de RetroArch est inspirée par le XMB de la Playstation 3 et de la PSP, ce qui le rend compatible avec une manette.

RetroArch est disponible sur :

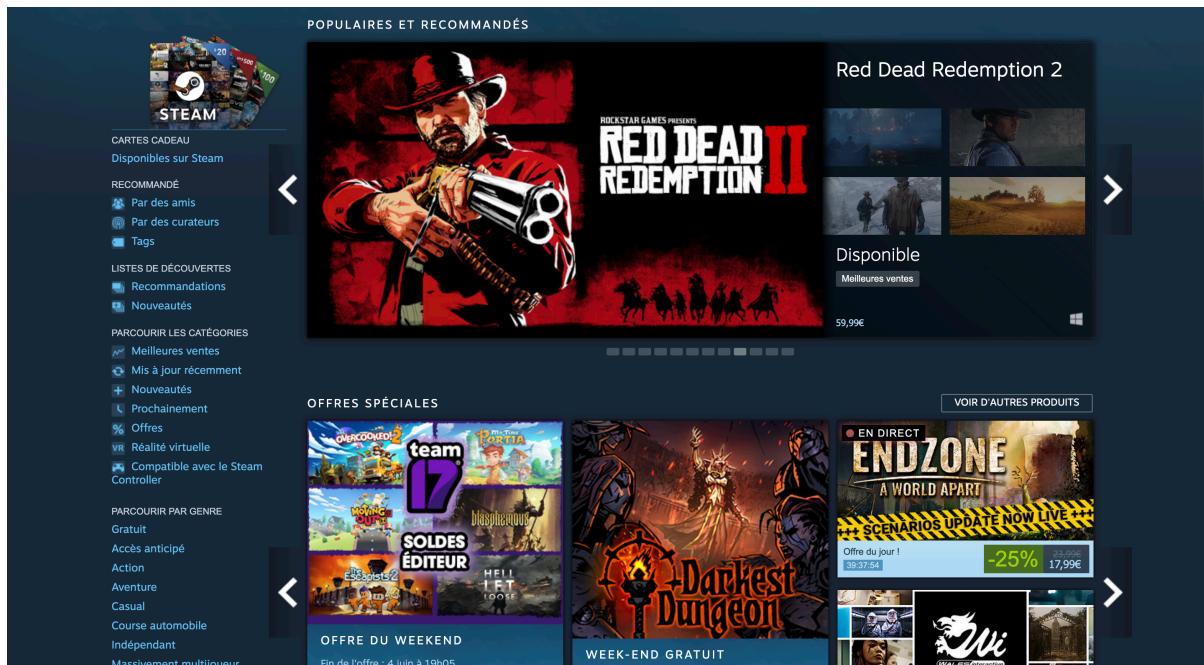
- Windows
- Linux
- Raspberry
- Android
- Apple macOS (ARM/x64)
- Apple macOS/OSX
- IOS/Apple TV
- Xbox Series / One
- OpenDingux
- PSVita
- PSP
- PS2
- PS3
- PS4
- Switch
- WII U
- WII
- Gamecube
- 3DS / 2DS

## 9.3 RomStation



RomStation est un frontend pour émulateur facilitant l'émulation de nombreuses consoles. RomStation a la différence de RetroPie ou RetroArch, l'utilisateur a la possibilité de télécharger des jeux directement depuis l'application. L'application a une formule payante qui permet d'améliorer la vitesse des téléchargements des jeux. RomStation est disponible sur Windows (x64)(x86), macOS.

### 9.4 Steam



Steam est une plateforme de distribution de jeux et d'applications en ligne. Steam est spécialisé dans les jeux vidéo contrairement aux autres projets que j'ai cité ci-dessus. C'est une application commerciale à but lucratif.

Steam n'est pas directement liée à l'utilisation d'émulateurs, mais j'ai décidé d'en parler car je m'inspire de son interface. Elle est créée spécifiquement pour une utilisation à la manette et la gestion que fais steam dès sauvegardes des utilisateurs.

### 9.5 Conclusion de l'analyse de l'existant

Il existe un grand nombre de frontend pour émulateur, ils permettent de simplifier l'utilisation pour l'utilisateur mais je n'en ai pas trouvé qui essaie d'atteindre le même but que moi. Le but de Caiman est de permettre la simplification de l'utilisation d'émulateurs et de permettre de synchroniser des sauvegardes tout en fournissant la possibilité de télécharger des jeux.

Steam représente un niveau de finition dans son interface BigPicture que j'aimerais atteindre. La synchronisation des sauvegardes sur steam est peut-être la plus performante, si l'on compare avec d'autres boutiques comme l'Epic Game Store ou Origin.

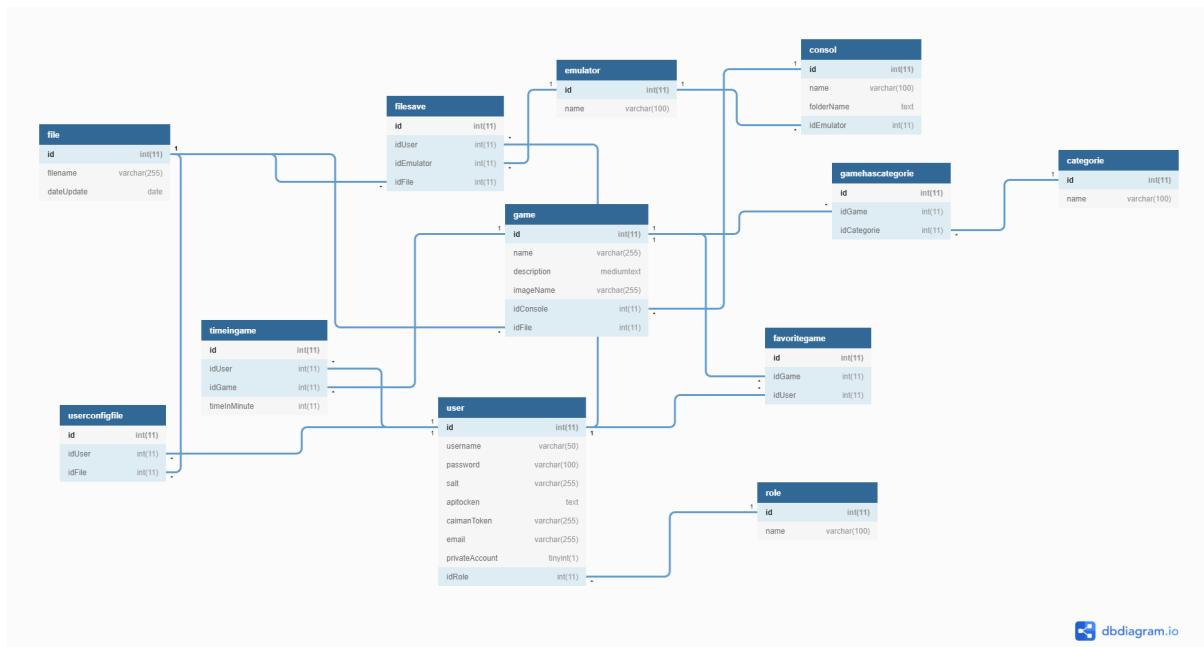
# CHAPITRE 10

## Analyse fonctionnelle

### 10.1 Base de données

#### 10.1.1 Schéma

Le schéma suivant représente la structure de la base de données. La base de données est commune au site web et à l'application Caiman. Les tables sont principalement utilisées pour stocker les informations des utilisateurs et des différents jeux.



dbdiagram.io

### **10.1.2 Table categorie**

Cette table est utilisée pour stocker les différentes catégories auxquelles les jeux peuvent appartenir.

### **10.1.3 Table console**

Cette table sert à stocker les différentes consoles prises en charge par l'application Caiman. Chaque console doit être reliée à un émulateur.

### **10.1.4 Table emulator**

La table émulateur sert à lister les différents émulateurs disponibles pour l'application, il est possible qu'un émulateur soit compatible avec plusieurs consoles.

### **10.1.5 Table favoritegame**

Sert à lister les jeux favoris des utilisateurs.

### **10.1.6 Table file**

La table file sert à lister le nom d'un fichier, cela peut être un fichier .iso d'un jeu, un fichier de sauvegarde ou un fichier de configuration utilisateur.

### **10.1.7 Table filesave**

Sert faire le lien entre un émulateur, un utilisateur, et un fichier de sauvegarde. Alors elle permet de connaître les sauvegardes pour un émulateur particulier d'un utilisateur.

### **10.1.8 Table game**

Table qui liste les jeux disponibles. Chaque jeu a plusieurs informations : un nom, une description, une console. La console permet à l'application de savoir quel émulateur doit être utilisé.

### **10.1.9 Table gamehascategorie**

Sert à assigner des catégories aux jeux.

### **10.1.10 Table rôle**

Sert à lister les différents rôles (utilisateur, administrateur), elle ne sert que pour le site web sachant que toute la partie administration est sur le site.

### **10.1.11 Table timeingame**

Sert à stocker le temps de jeu en minutes de chaque utilisateur. Le temps de jeu est spécifique à chaque jeu. Il est mis à jour directement depuis l'application Caiman.

### **10.1.12 Table user**

Cette table sert à stocker les informations de compte de chaque utilisateur de Caiman. Le compte est commun au site web et à l'application. Le mot de passe de l'utilisateur est crypté avec les fonctions de sécurité de php.

### **10.1.13 Table userconfigfile**

Cette table sert à faire le lien entre un fichier de configuration et un utilisateur. Le fichier de configuration sert à connaître la configuration.



# CHAPITRE 11

---

Site web Caiman

---

## 11.1 Cration de compte

L'utilisateur du site a la possibilite de crer un compte qui sera commun au site et a l'application. La cration de compte necessite de renseigner son email, de donner un nom d'utilisateur ainsi que un mot de passe.

## 11.2 connexion

La connexion a son compte utilisateur permet de modifier nos informations de compte et d'ajouter ou de supprimer des jeux a la liste de favoris.

Si l'utilisateur oublie son mot de passe, il a la possibilite de le reinitialiser. L'utilisateur qui se decide de reinitialiser son mot de passe reoit un mail contenant un lien de reinitialisation.

## 11.3 modification des informations d'un utilisateur

Un utilisateur connect a la possibilite de modifier ces informations :

- mot de passe
- liste de jeux favoris
- la visibilite de son profil pour les autres utilisateurs

## **11.4 affichage des jeux**

Tous les utilisateurs ont la possibilité d'afficher la liste de jeux disponible. Il n'y a pas de restriction particulière.

Les informations disponibles pour chaque jeu sont les suivantes :

- Nom
- Description
- Catégories du jeu

## **11.5 affichage d'un profil utilisateur**

Il est possible de consulter la page personnelle d'un utilisateur si celui-ci a rendu son compte publique. Les informations disponibles sont celle-ci :

- Nom d'utilisateur
- Jeux favoris
- Nombres d'heures de jeux sur chaque jeu

## **11.6 ajout d'un jeu à la base de données / sur le Bunker pour le fichier .ISO**

L'ajout d'un jeu se fait grâce à un formulaire, plusieurs champs sont à renseigner :

- le nom du jeu
- une description
- une image
- la console du jeu qui est uploadé
- le nom que va porter le jeu sur le Bunker
- le fichier de base du jeu

L'ajout à la base va créer deux entrées. Une dans la table Game et une autre dans la table file. Après avoir été ajouté depuis le site web, le jeu devient accessible depuis l'application Caiman et le site web pour de la consultation.

## **11.7 modification d'un jeu**

La modification d'un jeu ne peut être faite que par un administrateur. Les modifications possibles sont les suivantes :

- nom
- description
- console
- catégories

## 11.8 Administration

Les administrateurs ont la possibilité de faire plusieurs choses. Donc, je vais les lister ici, il n'est pas nécessaire de les détailler.

- modifier un jeu
- ajouter des catégories
- ajouter ou supprimer des catégories à un jeu

## 11.9 Téléchargement

L'un des intérêts du site est de pouvoir télécharger l'application Caiman. Le téléchargement de cette application nécessite d'être authentifié sur le site. Si un invité se rend sur la page de téléchargement sans être authentifié, une invitation lui sera faite de s'authentifier ou de créer un compte.

## 11.10 Fonctionnalités manquante

Malheureusement dû au temps imposé j'ai dû faire des choix organisationnel. J'ai donc décidé de me focaliser sur L'application Caiman au lieu du site internet ce qui explique que certaines fonctionnalités ne soient pas finalisées ou mises en place. Je vais lister les ce qui n'a pas été fini complètement ou abandonné.

- Modification complète des données des jeux
- Modification d'une catégorie
- Réinitialisation de mot de passe
- Création de compte administrateur



# CHAPITRE 12

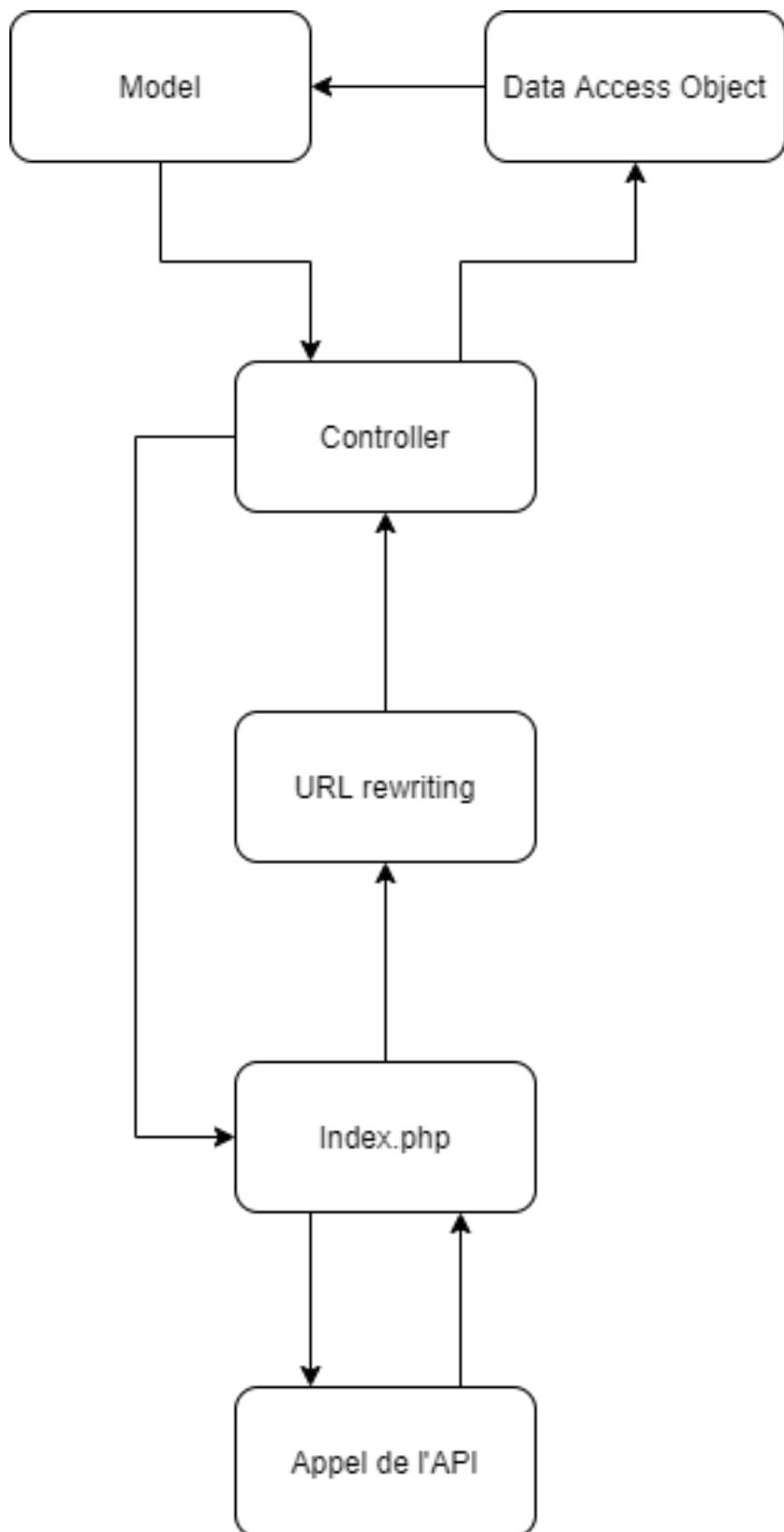
---

## API Caiman

---

L'API sert à pouvoir accéder à la base de données depuis l'application Caiman. Je vais détailler les endpoint et la structure de l'API.

## 12.1 Structure de l'API



Pour expliquer la structure de l'API, je vais expliquer étape par étape comment un appel se passe.

1. L'utilisateur envoie une requête à la page index.php de api.caiman.cfpt.info.

2. La requête est réceptionnée par index.php. L'url est ensuite traité par le .htaccess pour savoir où doit envoyer à quel controller.
3. Le contrôleur décide selon les informations reçues quelle méthode il doit exécuter.
4. Le DAO est appelé et va rechercher dans la base de données les données demandé
5. Le DAO crée la réponse grâce au model.
6. La réponse est envoyée à l'utilisateur par l'intermédiaire de la page index.php

## 12.2 Categories

### 12.3 GET

Permet de recevoir la liste des catégories disponibles.

Les informations reçues sont les suivantes :

- id
- nom

### 12.4 Games

### 12.5 GET

Retourne la liste des jeux.

Les informations reçues sont les suivantes :

- id
- description
- nom de l'image
- id de la console
- id du fichier du jeu

### 12.6 GET(?byName)

Retourne la liste des jeux qui dans le nom contient ce que l'utilisateur a demandé.

Les informations reçues sont les suivantes :

- id
- description
- nom de l'image
- id de la console
- id du fichier du jeu

## 12.7 GET(?byCategory)

Retourne la liste des jeux qui appartiennent à une catégorie.

Il faut spécifier l'id de la catégorie qui est demandée.

Les informations reçues sont les suivantes :

- id
- description
- nom de l'image
- id de la console
- id du fichier du jeu

## 12.8 GET(?byFavoriteUser)

Retourne la liste des jeux favoris d'un utilisateur.

Il faut spécifier l'id de l'utilisateur.

Les informations reçues sont les suivantes :

- id
- description
- nom de l'image
- id de la console
- id du fichier du jeu

## 12.9 GET(?byUserTime)

Retourne la liste des jeux auxquels un joueur a joué.

Il faut spécifier l'id de l'utilisateur.

Les informations reçues sont les suivantes :

- id
- description
- nom de l'image
- id de la console
- id du fichier du jeu
- nombre de minutes en jeu

## 12.10 GET(?gameFileName)

Retourne le nom du fichier d'un jeu.

Les informations reçues sont les suivantes :

- filename

## 12.11 GET(?gameConsole)

Retourne la console d'un jeu.

Les informations reçues sont les suivantes :

- name
- folderName

## 12.12 GET(?idGame&apiKey)

Retourne le fichier d'un jeu.

Les informations reçues sont les suivantes :

- fichier.iso

## 12.13 GET(?idGameTime&idUser)

Retourne le temps de jeu sur un jeu.

Les informations reçues sont les suivantes :

- minutes

## 12.14 GET(?idEmulator&idUser&apiKey)

Retourne un fichier zip contenant les sauvegardes d'un joueur pour un émulateur particulier

Les informations reçues sont les suivantes :

- fichier.zip

## 12.15 POST(?idEmulator&idUser&apiKey)

Upload un fichier zip contenant les sauvegardes d'un joueur pour un émulateur particulier

## 12.16 POST(idGameAdd&idUser)

Ajouter un jeu en favoris pour un utilisateur particulier

## 12.17 POST(idGameRemove&idUser)

Supprime un jeu en favoris pour un utilisateur particulier

## 12.18 POST(idGameCheck&idUser)

Vérifie si un jeu est déjà en favoris et retourne un booléen

## 12.19 POST(idGameTimeAdd&idUser)

Ajouter une minute de jeu à un jeu particulier pour un utilisateur

## 12.20 Users

### 12.21 GET(sans paramètres)

Retourne la liste des utilisateurs.

Les informations reçues sont les suivantes :

- id
- username

## 12.22 POST(avec apitoken)

Retourne un utilisateur en particulier

Les informations reçues sont les suivantes :

- id
- username
- password
- salt
- apitoken
- caimanToken
- email
- idRole

## 12.23 User/connection

### 12.24 POST(?username, password)

Permet de vérifier les informations de connexion d'un utilisateur

Les informations reçues sont les suivantes :

- id
- username
- password
- salt
- apitoken
- caimanToken
- email
- idRole

### 12.25 POST(caimanToken)

Permet de recevoir les informations d'un utilisateur grâce à un token généré à chaque connexion.

Les informations reçues sont les suivantes :

- id
- username
- password
- salt
- apitoken
- caimanToken
- email
- idRole



# CHAPITRE 13

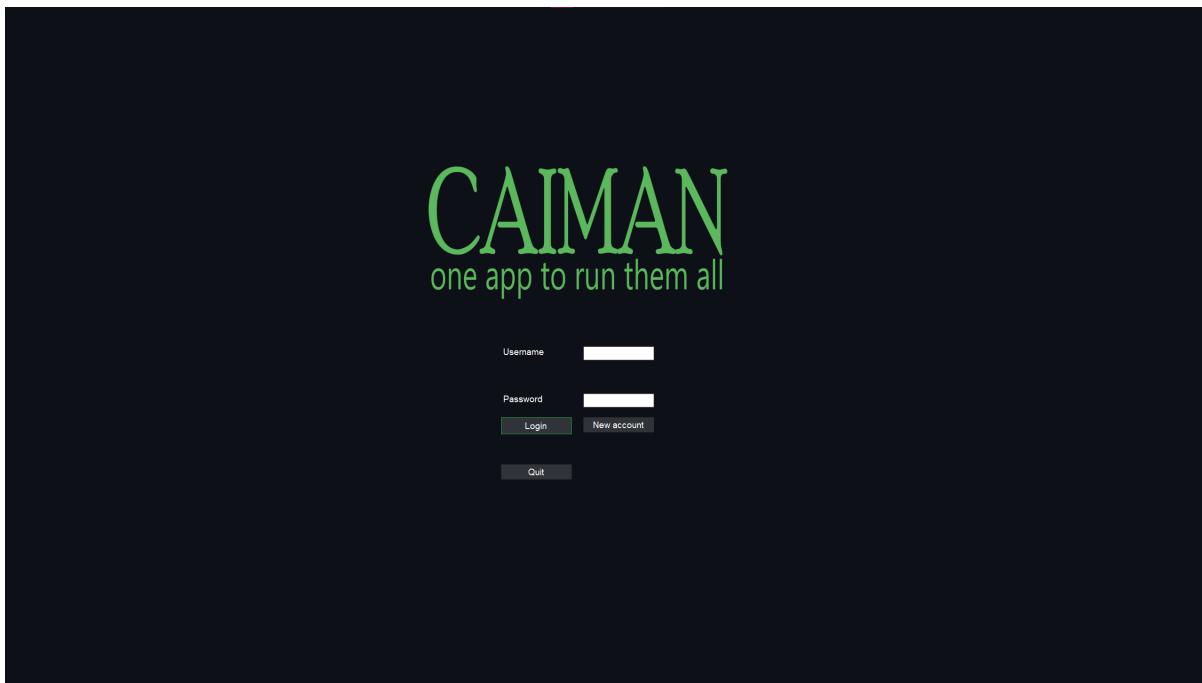
---

## Application Caiman C#

---

### 13.1 Connexion

L'utilisateur de Caiman doit obligatoirement être connecté pour pouvoir utiliser caiman. La connexion se fait avec le nom d'utilisateur et le mot de passe. Une fois la connexion valable, elle est maintenue tant que l'utilisateur ne se déconnecte pas ou tant que l'utilisateur ne s'est pas connecté sur un autre ordinateur.

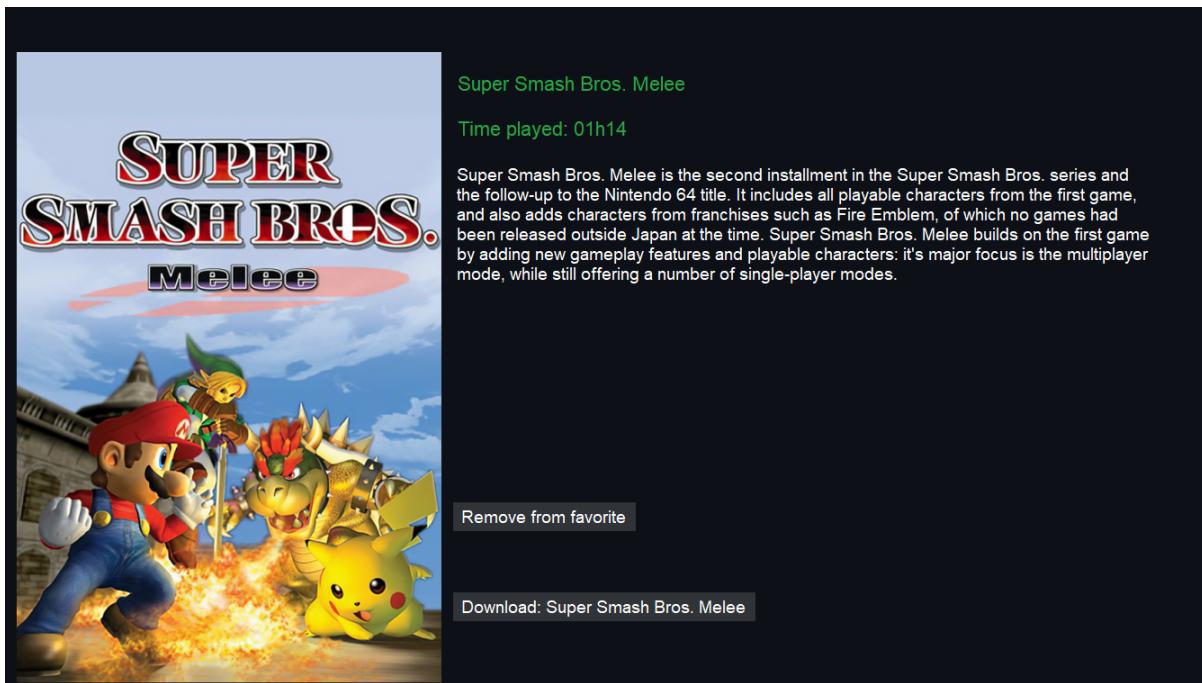


## 13.2 Inscription

L'inscription n'est pas disponible sur caiman mais un bouton est disponible pour être redirigé sur le site web de Caiman.

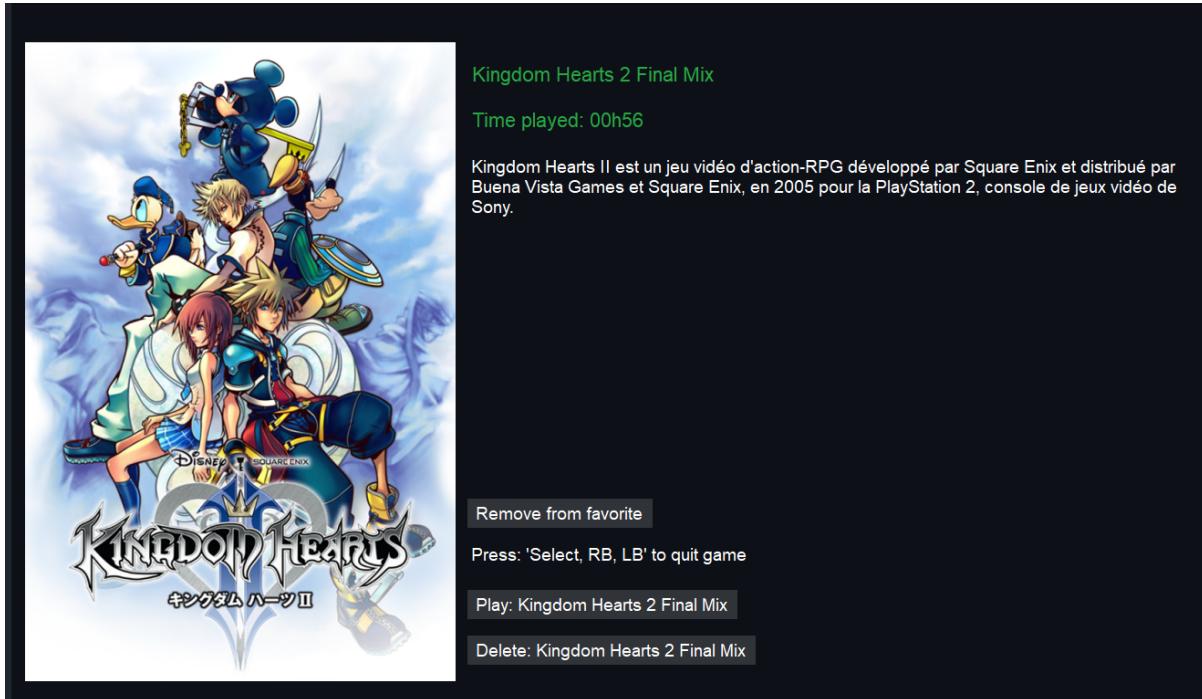
## 13.3 Téléchargement de jeu

L'utilisateur a la possibilité de télécharger des jeux. Les jeux disponibles dans l'application sont ajoutés depuis le site internet de Caiman. Les téléchargements des jeux se font les un après les autres.



## 13.4 Lancement d'un jeu

Caiman inclut deux émulateurs Dolphin un PCSX2. Ces deux émulateurs permettent d'exécuter des jeux de Gamecube et Wii pour Dolphin et de Playstation 2 pour PCSX2. Pour lancer un jeu, il suffit de le télécharger, puis de cliquer sur Play. Il n'est donc pas nécessaire de lancer soit même un émulateur.



## 13.5 Synchronisation des sauvegarde entre le pc client et le Bunker

Les sauvegardes de l'utilisateur sont synchronisées entre les différents pc qu'il utilise. La synchronisation se fait à la connexion, les sauvegardes sont stockées sur les serveurs de Caiman. La copie des sauvegardes des utilisateurs est envoyée automatiquement donc l'utilisateur n'a pas de manipulation à faire. L'envoie se passe dès que l'utilisateur sauvegarde, cela permet d'éviter de perte de sauvegardes si un problème se produit durant le moment où le joueur est en train de jouer.

## 13.6 Modification de la configuration utilisateur

L'utilisateur a la possibilité de modifier plusieurs paramètres graphiques. Il a la possibilité de choisir en trois mode de configuration global :

- Original
- 1080p
- 4K

C'est différent mode modifie l'antialiasing et la définition native de l'émulateur qui va être utilisé.

L'utilisateur a aussi la possibilité de choisir entre lancer le jeu plein écran ou non et il peut choisir si le jeu doit être en 16/9 ou en 4/3.

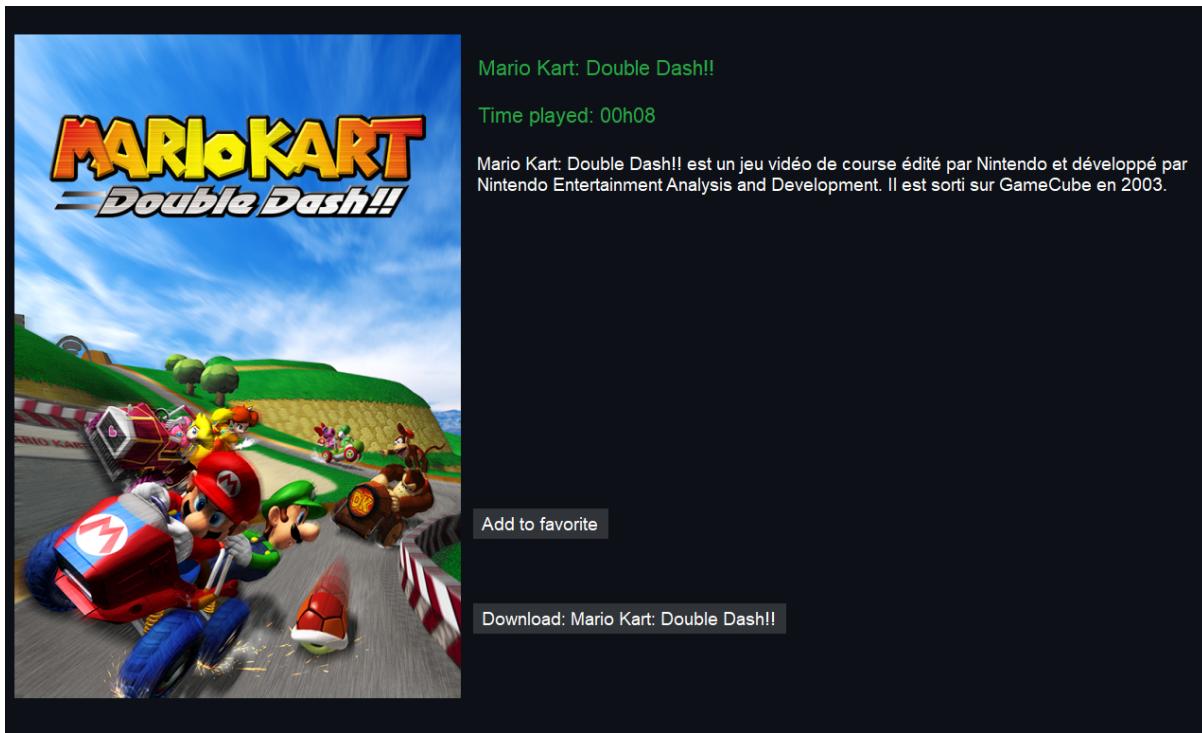


## 13.7 Application de la configuration utilisateur

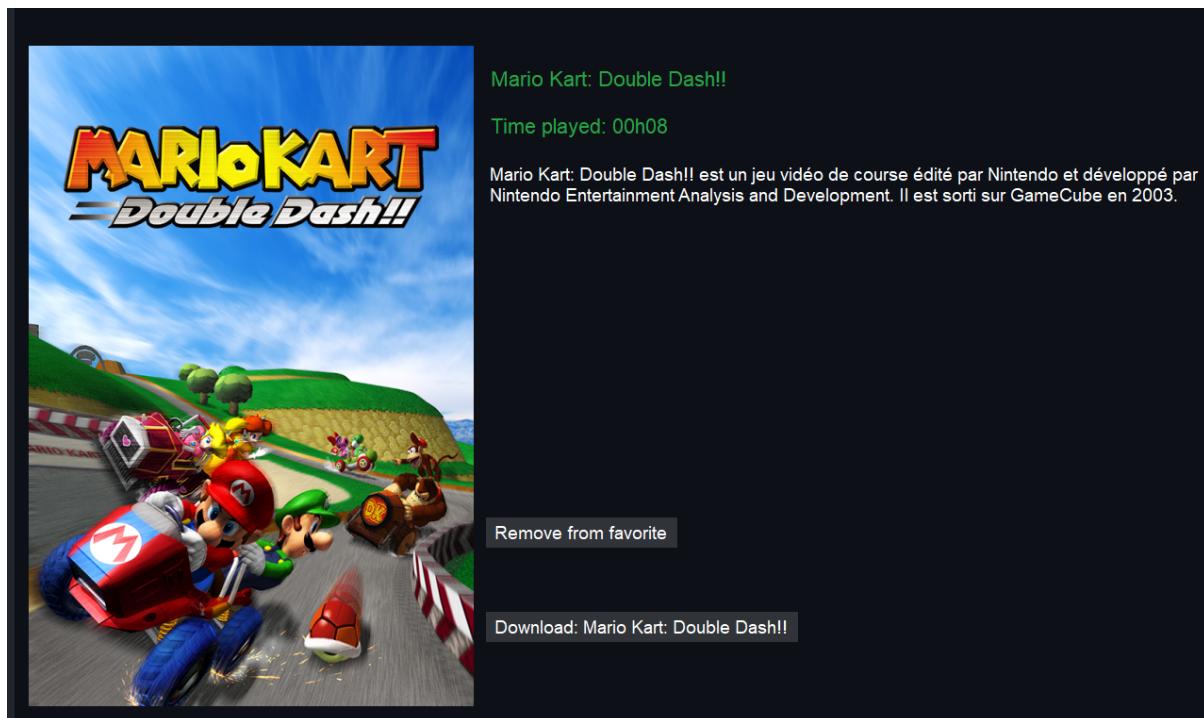
La configuration est appliquée avant de lancer un jeu, c'est à dire que la configuration n'est pas appliquée si l'utilisateur est déjà en jeu.

## 13.8 Ajout/suppression de jeu en favoris

L'utilisateur a la possibilité de modifier ses jeux favoris directement depuis caiman.

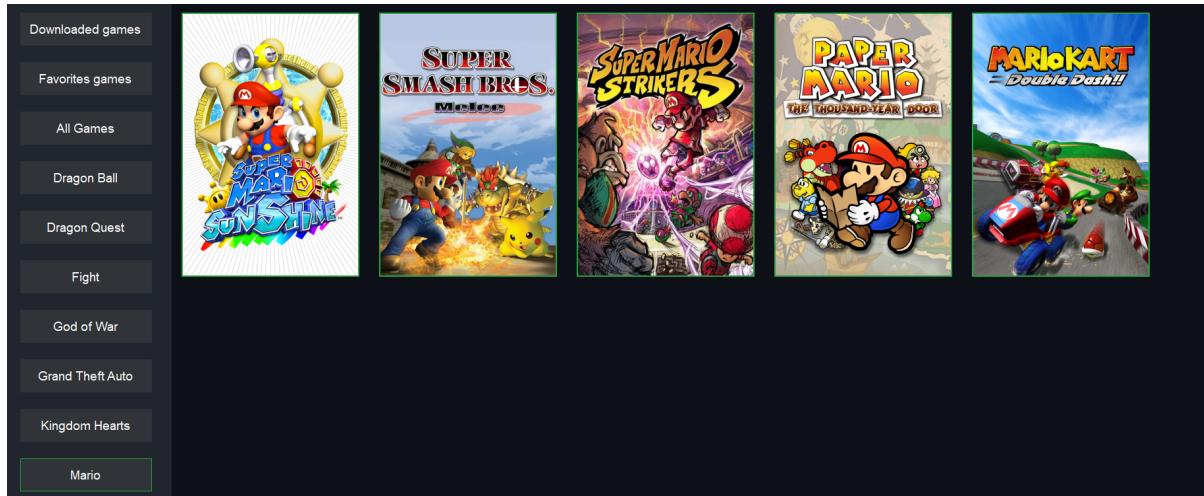


Suppression de jeu des favoris



### 13.9 Affichage de jeux par catégories

L'utilisateur a la possibilité d'afficher les jeux par catégories. Les catégories sont créées manuellement par un administrateur. Il y a certaines catégories « spéciales », par exemple les jeux favoris et les jeux uniques téléchargés par chaque utilisateur.



## 13.10 Nombre d'heures de jeu

Le nombre de minutes de jeu d'un utilisateur est mis à jour à chaque minute de jeu. le nombre de minutes de jeu est visible sur la page de chaque jeu si l'utilisateur n'a pas encore joué.

Dragon Quest VIII

Time played: 00h58

## 13.11 Gestion des manettes

Pour Caiman, les manettes supportées sont les manettes pour xbox qui fonctionne avec Xinput. Il est possible d'utiliser d'autres manettes pour cela, il faut passer par un programme qui va convertir les inputs de la manette non compatible en input de manette xbox.

Pour gérer les déplacements, j'utilise les touches "haut, bas, gauche, droite", le stick gauche, la validation se fait avec la touche "A" et le retour arrière avec la touche "B".



# CHAPITRE 14

---

## Interface utilisateur utilisable à la manette

---

### 14.1 Réception des input des manettes connecté

L'utilisateur de Caiman a la possibilité de pouvoir utiliser l'application au clavier souris mais aussi à la manette. Pour ce faire, j'ai utilisé le paquet nuGet "SharpDX.XInput". Ce paquet me permet de connaître les manettes connectées au pc ainsi que les touches appuyées par l'utilisateur.

La seule manette qui peut se déplacer dans l'application est la manette 1. Pour connaître les boutons de la manette, j'utilise la fonction getInput(). Cette fonction me permet de connaître les touches qui sont pressées à un instant T. Je vais chercher les inputs toutes les 2ms pour être sûr de ne pas louper d'inputs.

Les inputs sont ensuite traités par l'interface de l'application qui décide quoi en faire selon le contexte.

### 14.2 Transformation des input de la manette en événement

Les inputs de la manette sont analysés par la form principale de Caiman. Selon la ou les touches qui sont pressées, l'application exécute des actions différentes. Par exemple, quand la touche "A" est pressée, alors le programme envoie la touche ENTER à l'application ce qui me permet de cliquer sur des boutons.

Pour gérer les événements, je fais un test pour savoir si l'utilisateur utilise l'application ou non . Si l'application n'est pas focus par l'utilisateur, seul une partie des actions sont possibles pour éviter que des actions inattendues puissent arriver alors que l'application n'est plus visible.

## 14.3 Structure de l'affichage

L'affichage est constitué d'un "XboxMainForm". Il sert à contenir tous les autres panels, il est aussi chargé de la gestion des inputs de la manette de l'utilisateur 1.

Un "XboxMainForm" contient une liste de XboxUserController qui elles contiennent différentes choses comme des boutons des images ou des labels.

Le XboxMainForm est aussi responsable de la gestion des demandes de l'utilisateur, par exemple si l'utilisateur veut afficher l'accueil de Caiman, il va lui passer un objet contenant sa demande. Il va donc traiter les demandes et afficher les panels selon les besoins de l'utilisateur.

## 14.4 Déplacement dans un panel de l'application

L'application est conçue avec des "panels", c'est-à-dire une liste de listes de controls. Cette liste de controls est propre à chaque panel. Les panels contiennent aussi une variable position\_x et position\_y qui permettent de connaître le control actuellement sélectionné par l'utilisateur. Quand l'utilisateur décide de se déplacer, il demande au panel de modifier ses variables x et y. Avant de valider ce changement, le panel regarde si le déplacement demandé par l'utilisateur est possible ou non.

Il existe 3 possibilités :

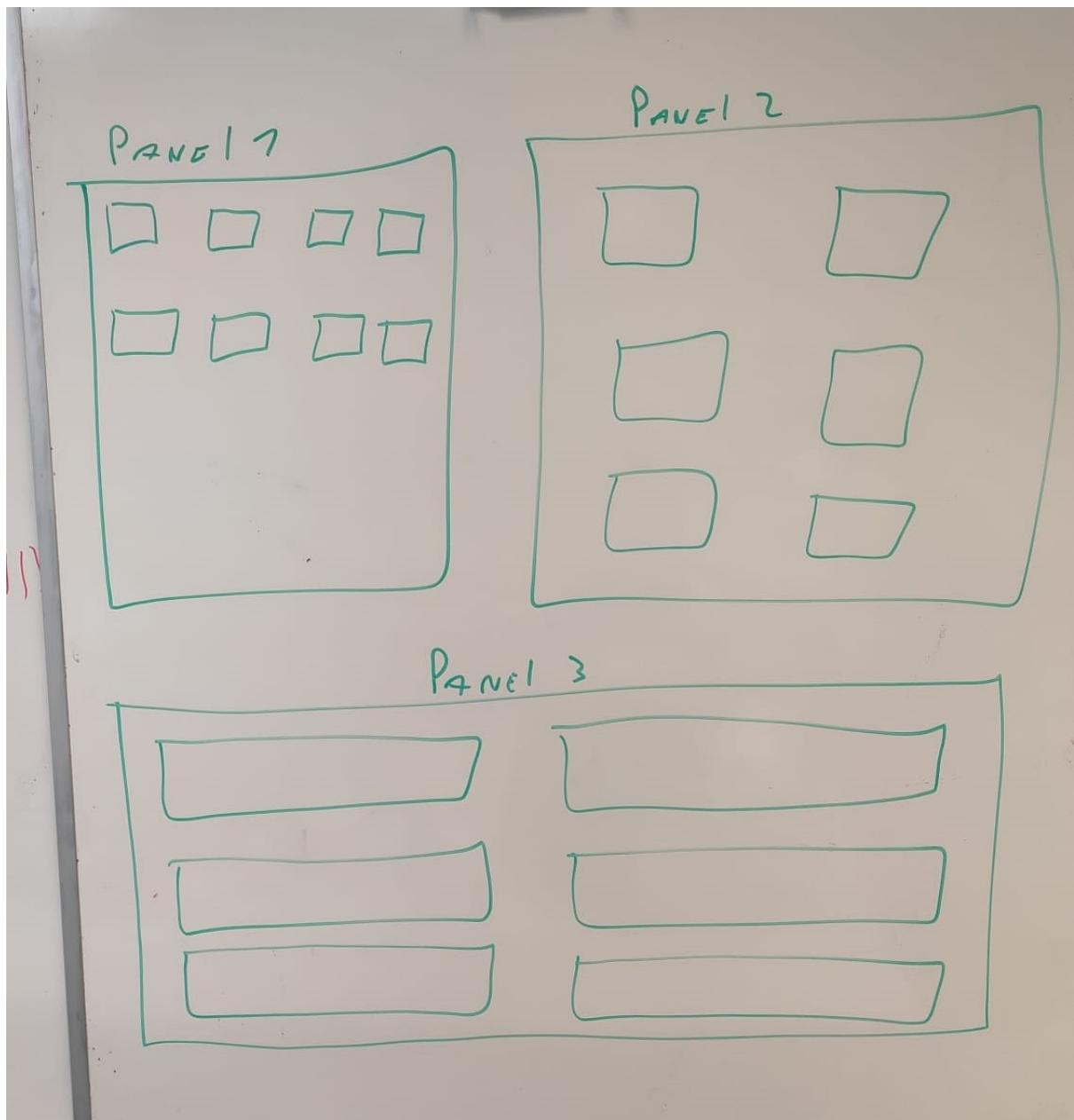
1. Le déplacement est possible, alors la position sur l'axe x,y est modifiée.
2. Le déplacement est impossible car il n'y a rien à l'emplacement demandé. Dans ce cas, le panel va décider de bouger le curseur sur un des emplacements possibles.
3. L'utilisateur est à la fin du panel et "sors du panel" dans ce cas il va se diriger dans un autre panel s'il y en a un dans la direction demandée.

## 14.5 Déplacement de panel en panel

Chaque "panel" possède un pointeur sur le panel du haut, du bas, de droite et de gauche.

Ces pointers ne sont pas forcément utilisés, ils ont le droit d'être nul.

Si on prend l'exemple suivant :



Nous avons 3 panels différents qui contiennent chacun plusieurs controls.

Le panel 1 possède donc deux pointeurs différents, un sur le panel 2 et un autre sur le panel 3.  
trop confus

Si l'utilisateur se trouve en bas du panel 1 et qu'il décide de se déplacer encore plus bas, il ne pourra pas car aucune case n'est disponible dans ce panel. C'est pourquoi une vérification sera faite pour savoir si un panel est indiqué comme le panel "down", si tel est le cas le focus va changer de panel.

Un autre cas possible est que l'utilisateur va peut-être décider de retourner sur le panel 1. La contrainte est de savoir où doit pointer le panel haut du panel 3. Actuellement, un seul panel peut être défini par côté, mais la solution est de créer de petits panels pour éviter que ces situations arrivent.



# CHAPITRE 15

---

## Logbook

---

**15.1 19.04.2021**

**15.1.1 8h05**

Entretiens avec M. Garcia

**15.1.2 9h05**

Copie de mon disque pour Gawen

**15.1.3 9h20**

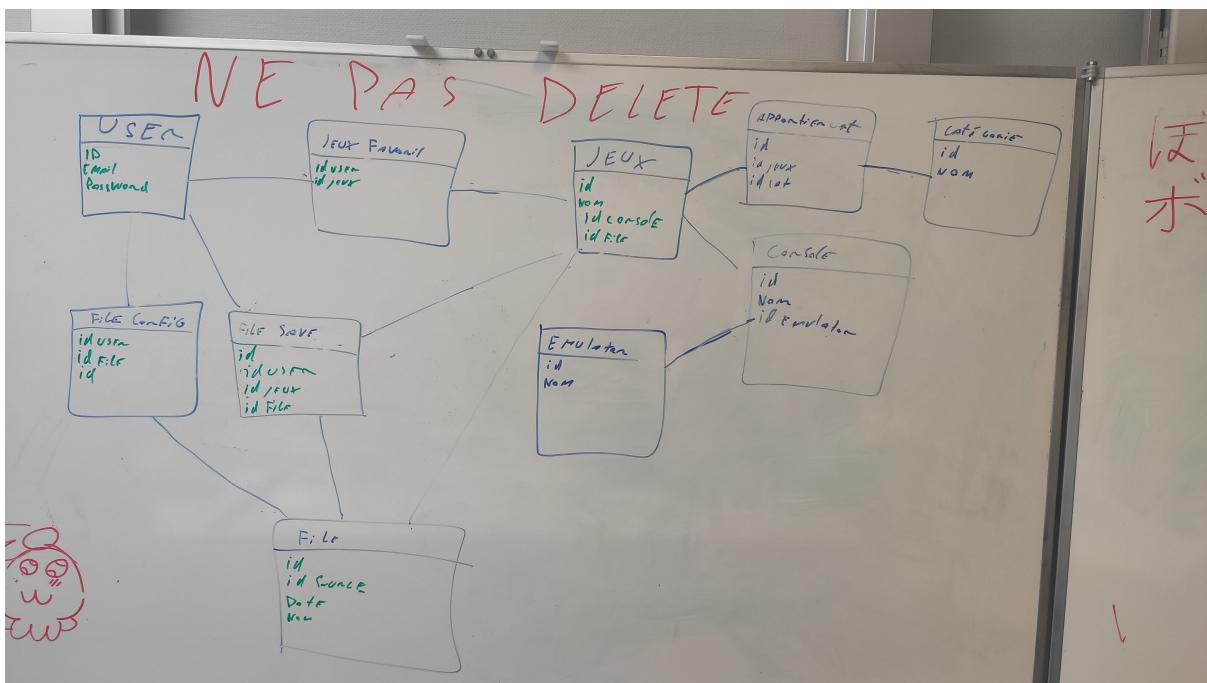
Création du git

**15.1.4 9h30**

je réfléchi à ce par quoi je vais commencer j'hésite entre commencé entre le site web ou les téléchargement

### 15.1.5 9h40

Modélisation de la BDD



### 15.1.6 10h30

Installation de laragon

### 15.1.7 10h40

Création de la base de données

### 15.1.8 12h40

Création de la structure du site web

### 15.1.9 14h35

Création des différentes pages et mise en place de bootstrap

### **15.1.10 résumé**

j'ai créer la base de donnée et le site web

---

## **15.2 20.04.2021**

### **15.2.1 8h05**

je continue à créer le site web je crée les formulaires pour la connexion

### **15.2.2 15h20**

J'ai impliqué la création de compte et la connexion, le mail doit être unique et le username aussi.  
j'ai selon les indication de M. Schmid utilisé les fonctions  
password\_hash et password\_verify de php.

### **15.2.3 15h30**

j'ajoute des donnée a la main dans la bdd

### **15.2.4 15h56**

aide de M.Schmid pour du sql

---

## **15.3 21.04.2021**

### **15.3.1 8h10**

Modification de la structure du site et ajout de l'update de mots de passe

### **15.3.2 13h00**

l'affichage des jeux est disponible ainsi qu'une recherche sur les jeux grâce à leurs noms.

### **15.3.3 13h20**

Ajout de champs dans la table game

- description
- imageName

### **15.3.4 15h00**

l'affichage de la recherche et du détail d'un jeu fonctionne mais n'est pas beau.

---

## **15.4 22.04.2021**

### **15.4.1 8h05**

Ajout de l'affichage des catégories de chaque jeu

### **15.4.2 9h00**

Affichage des jeux qui appartiennent à une catégorie.

### **15.4.3 10h20**

modification de l'interface de recherche

### **15.4.4 notes personnelles**

- je dois ajouter une table pour savoir le nombre d'heure de jeu de chaque utilisateurs
- je dois ajouter une gestion des messages d'erreurs.
- je peux ajouter une photo de profil

### **15.4.5 11h0**

modification de l'interface de connexion et de d'inscription

### **15.4.6 11h50**

suppression de la page de création de compte

### **15.4.7 12h40**

Ajout d'un jeux en favoris

### **15.4.8 15h00**

suppression d'un jeu en favoris

### **15.4.9 a faire demain**

- l'affichage des card dans le dashboard n'est pas bon
- 

## **15.5 23.04.2021**

### **15.5.1 8h05**

modification de la l'interface du Dashboard

### **15.5.2 8h40**

Ajout d'un champ dans la table utilisateur pour spécifier si l'utilisateur est privé ou non si l'utilisateur n'est pas privé tout le monde va pouvoir voir son profils

### **15.5.3 9h30**

la modification du paramètre pour savoir si le compte est privé ou non

### **15.5.4 12h20**

test de Git Hook

### **15.5.5 13h30**

modification de l'interface theme blanc -> dark

### **15.5.6 15h20**

Création de la page de téléchargement et mise en place de du téléchargement de Caiman depuis le site

### **15.5.7 notes pour la prochaine fois**

- Je dois créer la partie Administrateur du site
  - Je dois créer une fonctionnalité qui me permet de gérer les messages d'erreurs
  - Je dois sécuriser l'accès au pages
  - je dois sécuriser les différents formulaires
  - je dois me renseigner comment uploader des gros fichier depuis un poste clients
  - je dois changer de navbar
- 

## **15.6 26.04.2021**

### **15.6.1 8h05**

notes personnelles :

- je dois ajouter la possibilité d'afficher la page d'un utilisateur
- je dois corriger mon script d'export de base de données

## **15.6.2 8h10**

Création de la page dédiée aux administrateurs.

## **15.6.3 8h30**

Ajout de catégorie

## **15.6.4 9h00**

Ajout de jeu

## **15.6.5 notes personnels**

j'ai regardé plusieurs méthodes pour envoyer un fichier depuis un formulaire en php. Pour l'instant j'utilise les fonctions de base de php et elle fonctionne donc je vais faire des tests une fois le site uploadé sur le Bunker.

## **15.6.6 11h00**

le fichier .iso est uploadé avec le bon nom mais pas encore dans la base de données

## **15.6.7 15h00**

Le jeu est bien ajouté avec le bon nom ainsi que la bonne image.

## **15.6.8 15h45**

il est maintenant possible de mettre à jour le nom, la description ou la console d'un jeu.

## **15.6.9 15h50**

modification de la structure du git

---

## **15.7 27.04.2021**

### **15.7.1 8h05**

ajout/ suppression de catégories à un jeu

### **15.7.2 9h20**

modification mineur de l'interface

### **15.7.3 10h05**

recherche d'un profil utilisateur

### **15.7.4 12h15**

la recherche et l'affichage d'un profil utilisateur est fonctionnel

### **15.7.5 13h00**

modification de l'interface pour que les jeux s'affiche correctement

### **15.7.6 13h30**

Le site est fonctionnel mais il manque des détail comme les message d'erreur et les droit sur les pages

### **15.7.7 notes personnelles**

pour finir le site il me reste les choses suivante à faire :

- Sécuriser les pages
- afficher des messages d'erreur
- suppression de catégories
- suppression de jeu
- mot de passe oublié
- commenter mon code
- pagination pour les recherches

## **15.7.8 13h40**

documentation

---

## **15.8 28.04.2021**

### **15.8.1 8h05**

documentation

### **15.8.2 10h40**

correction d'un bogue sur le nombre d'heure de jeu et triage des jeux par heure de jeu

### **15.8.3 11h00**

gestion des droit d'accès au page

### **15.8.4 13h10**

modification de la navbar

### **15.8.5 14h00**

gestion des erreurs de du login

---

## **15.9 29.04.2021**

### **15.9.1 8h30**

Documentation

### **15.9.2 10h20**

Mr. Garcia m'a aidé à mettre en place ma documentation doxygen

### **15.9.3 11h00**

documentation

---

## **15.10 30.04.2021**

### **15.10.1 8h05**

Correction du logbook

### **15.10.2 8h20**

supression de code inutile sur le site et mis a jour de la documentation

### **15.10.3 9h30**

ajout de contrainte dans la base de données

### **15.10.4 10h45**

Création du planning effectif

### **15.10.5 13h00**

Configuration de debian

user tfp : FTPdiplomant password ftp : SuperCfpt@

J'ai pus installer apache, php, et mysql mais je n'arrive pas à me connecter en ftp. A la connexion je bloque sur l'erreur : "Impossible de récupérer le contenu du dossier". Malgré l'aide de Mr.Schmidt je n'ai toujours pas réussi.

## **15.10.6 15h20**

Je n'arrive toujours pas à me connecter au ftp, je ne sais pas si le problème viens de ma configuration ou du firewall

## **15.11 notes pour le premier rendu**

J'ai durant ces deux premières semaines, créé le site internet de Caiman. Le site n'est pas fini à 100% mais les fonctionnalités de base sont toutes implémentées. Les fonctionnalités actuelles permettent de faire toutes les choses nécessaires au fonctionnement de l'application. J'ai pris plus de temps que prévu à réaliser le site mais durant la création de mon planning prévisionnel j'ai fait des erreurs, j'ai par exemple oublié de planifier la création des fonctionnalités d'administration ( Ajout de jeux, ajout de catégories, assignation de catégories à un jeu, upload de jeux, etc ).

La documentation du site n'est pas forcément touffue mais le site n'est pas particulièrement complexe, il m'a pris du temps dû au nombre de tables à gérer. Il reste à mettre en place la récupération de mot de passe mais j'ai décidé de passer à l'interface graphique de l'application C# dès la semaine prochaine. Étant donné que le projet est la partie la plus importante de l'application, je ne vais pas configurer la récupération de mot de passe maintenant.

---

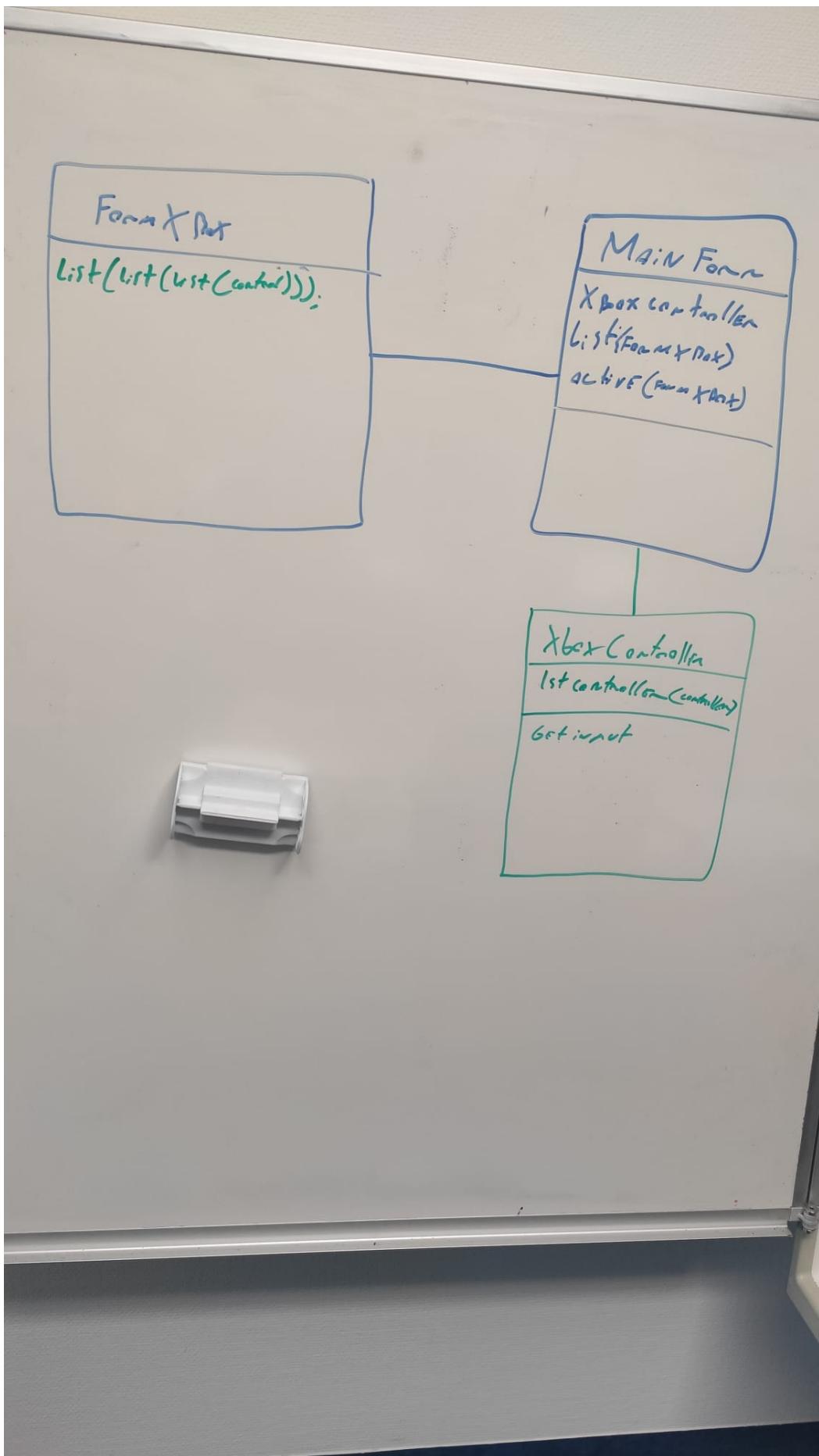
## **15.12 03.05.2021**

### **15.12.1 8h10**

Réflexion sur l'interface graphique et création du projet

### **15.12.2 8h15**

Importation de la classe XboxController.cs que j'ai créé précédemment.



La table MainForm contient un XboxController() cette classe permet de connaître les manettes connectés au pc et de recevoir leur inputs.

Elle contient aussi une liste de form, ces formes sont les différentes fenêtres de l'application (si possible j'aimerai faire que seul une fenêtre soit active).

Je fais des test avec les usercontrols

### **15.12.3 10h00**

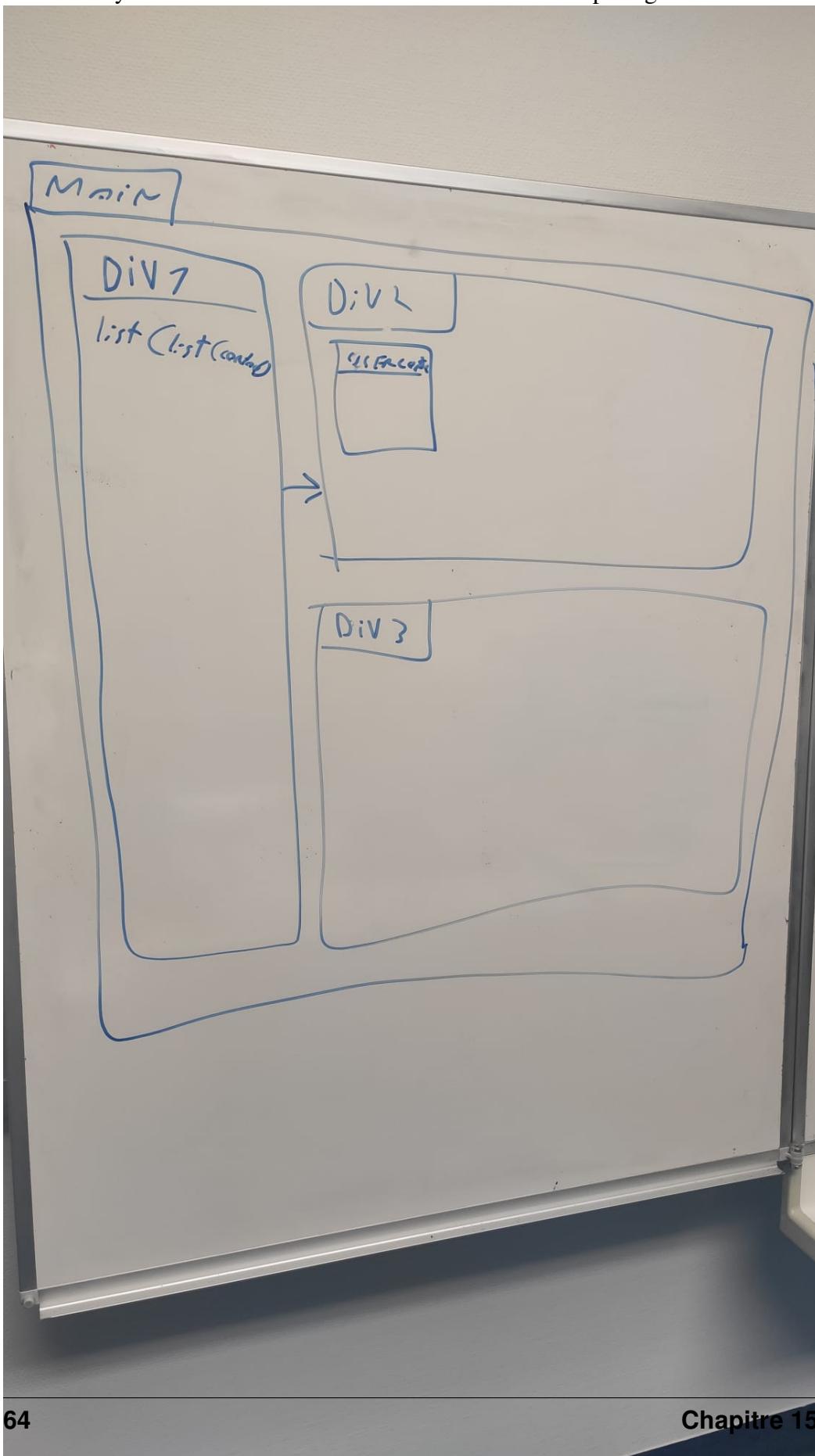
je n'arrive pas à afficher dynamiquement quand une action est faite.

### **15.12.4 12h00**

Mon problème venait du fait que je n'initialisais pas l "usercontrol.

### 15.12.5 13h00

Je vais essayer de me baser sur la structure des div en html pour gérer le contenu de l'affichage.



## **15.12.6 14h00**

Finalisation de la configuration du serveur

---

## **15.13 04.05.2021**

### **15.13.1 8h05**

Je continue à faire des test pour pouvoir bouger le focus d'une "div" à une autre

### **15.13.2 11h00**

Je continue à faire des test mais j'ai apporté des modifications :

le MainForm ne contient pas une liste de list de control, ce n'est pas nécessaire sachant que le lien entre les control est seulement connu des sous contrôle

dorénavant chaque sous control possède une liste de listes de control pour pouvoir se déplacer.

quand on déplace la position maximum dans un sous control il y a deux possibilité. : un control est disponible dans la direction souhaité rien n'est disponible Si un control est disponible alors la main form est informée qu'elle doit changer d'activeForm pour pouvoir se déplacer dans de bonne condition.

### **15.13.3 12h05**

Le déplacement dans chaque form fonctionne et l'on peut passer d'une form à une autre.

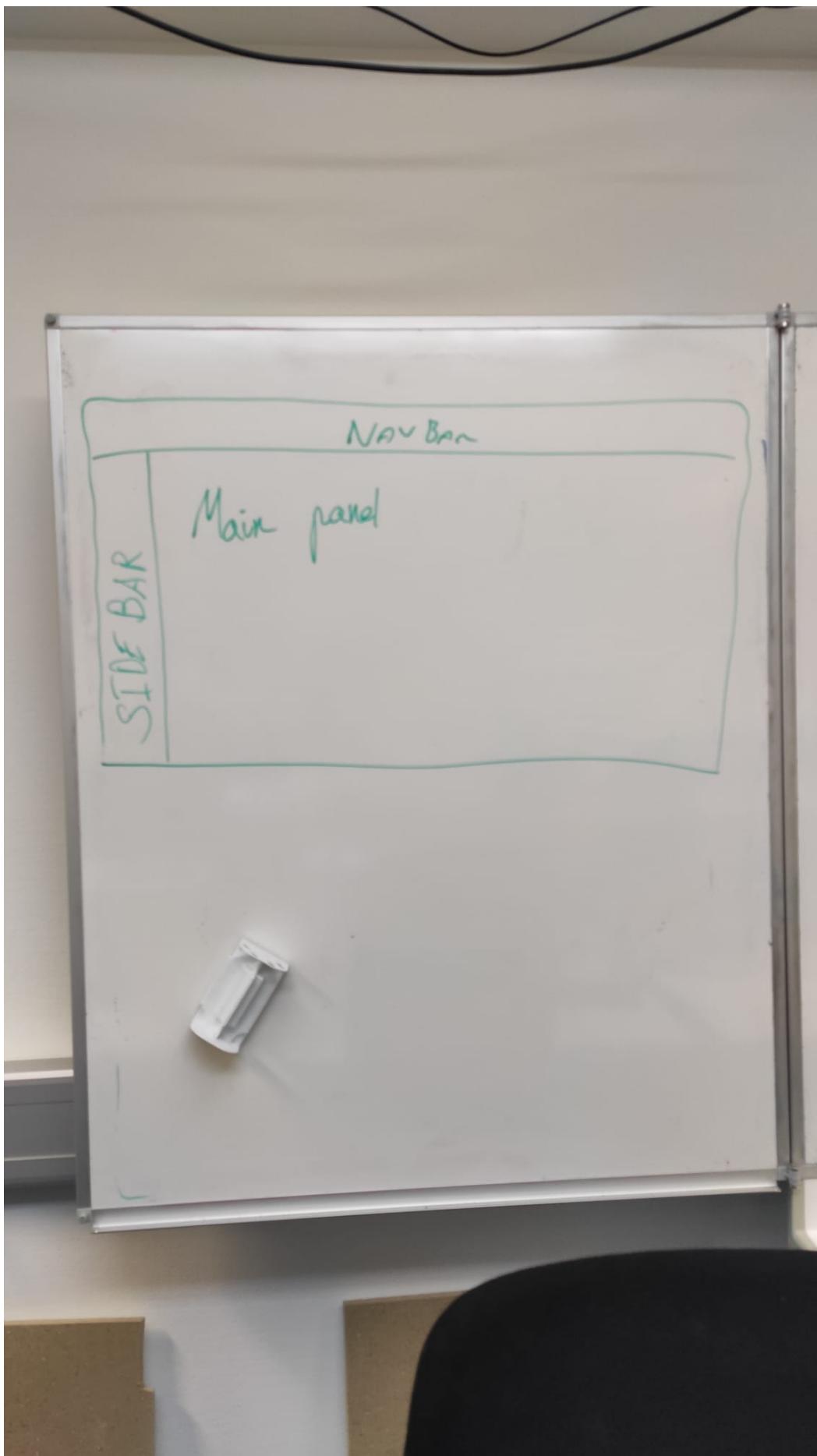
---

## **15.14 05.05.2021**

### **15.14.1 8h05**

Je continue à améliorer le fonctionnement de l'interface.

J'ai décidé de diviser l'interface en 3 "partie" la navbar la sidebar le main contenu



### **15.14.2 10h50**

J'ai créé une nouvelle classe ButtonContext.cs, elle contient les paramètres qui doivent être passer a la forme pour quel sache l'action à exécuter.

Il est maintenant possible de cliquer sur un bouton et la manette va reprendre la ou l'utilisateur a cliqué.

### **15.14.3 12h40**

Je permet le déplacement grâce au joystick gauche

### **15.14.4 13h00**

J'essaie de faire en sorte que je puisse revenir en arrière dans les fenêtre

### **15.14.5 15h00**

J'ai essayé de sauver dans une liste les choses précédemment affichée mais j'ai des soucis avec les liens entre les différents panel.

Je vais essayer de recréer les anciens panel a chaque fois au lieu de les recharger.

---

## **15.15 06.05.2021**

### **15.15.1 8h05**

Je continue a faire en sorte que je puisse retourner en arrière grâce à la touche "B"

### **15.15.2 9h00**

Il est maintenant possible de revenir en arrière dans la navigation

### **15.15.3 10h00**

si il y a des “trou” dans la navigation le curseur le contourne

### **15.15.4 10h40**

J'essaie d'afficher des images dans l'application

### **15.15.5 13h00**

J'ai discuté avec Mr Maréchal de mon git. Pour pouvoir appliquer correctement mon gitignore j'ai dus supprimer les fichier du git

### **15.15.6 14h00**

J'ai fais des recherche sur la publication de projet et j'ai corrigé des bugs

---

## **15.16 07.05.2021**

### **15.16.1 8h05**

J'ai essayé le paquet “Microsoft Visual Studio Installer Projects” que m'a conseillé M. Schmid. L'installation marche mais certains dossiers de PCSX2 et de Dolphin ne sont pas inclus dans l'installation.

### **15.16.2 10h00**

Je commente les classe que j'ai créé et je supprimer les fonctions qui ne sont pas utilisé

### **15.16.3 12h40**

documentation + Création d'une release pour pouvoir essayer l'interface

Je déplace la documentation du projet web au projet desktop la seul documentation qui reste dans le projet web est celle qui concerne son propre code (doxygen)

#### **15.16.4 notes personnelles**

Je dois modifier la connexion a la base de données pour passer des fonctions de cryptographie de PHP au mds. Le problème ce que je ne peux pas me connecter depuis le c# avec les fonctions de cryptographie de php.

#### **15.16.5 15h30**

J'ai mis à jour ma documentation du projet.

J'ai essayé d'ajouter des tâches à mon git mais j'ai eu des soucis pour la signature du projet donc je remet ça à plus tard.

---

### **15.17 10.05.2021**

#### **15.17.1 8h05**

Je modifie la création de mot de passe et la connexion pour qu'un utilisateur puisse se connecter depuis le site web et Caiman.

J'utilisais les fonctions de php (password\_hash et password\_verify) mais je passe a du md5 + salt. La raison est que je ne pouvais pas utiliser ces fonctions pour me connecter depuis l'application c#.

#### **15.17.2 10h10**

je vais essayer de créer un login en C#

Pour ce faire, je commence par créer une classe “AccessDatabase.cs” pour communiquer avec la base. Pour stocker la la connexionString je l'ai mise dans les settings de l'application.

#### **15.17.3 10h45**

Pour pouvoir me connecter a la base de donnée je dois passer par le port 1433 mais il est fermé dans le firewall du coup je dois essayer de me connecter par le port 433

### **15.17.4 13h30**

Je commence à créer la “vrai” interface je commence par les menu de configurations et le menu pour quitter l’application.

### **15.17.5 14h00**

Après une discussion avec M. Scmid j’ai décidé de faire une api.

### **15.17.6 14h30**

J’ai demandé à M.Borel de l’aide pour la structure de mon api, il a pu m’indiquer une structure correcte mais elle est très verbeuse donc je vais sûrement avoir pour minimum 3 jours à faire mon API.

---

## **15.18 11.05.2021**

### **15.18.1 8h00**

Je continue mon api, je commence par essayer la structure de M.Borel sur une seule table pour essayer puis je vais faire les autres.

### **15.18.2 11h40**

Je peux maintenant faire des requêtes mais j’ai un souci avec les headers. Le header Authorization n’est pas correctement reçu Je pense que vu le temps que faire une API prend je fais faire en sorte que l’API ne soit que faite pour Caiman. Je vais donc seulement créer les requêtes nécessaires

### **15.18.3 13h00**

Je vais lister les requêtes qui me seront potentiellement utile pour Caiman

liste des jeux recherche de jeu affichage des informations d’un jeu recherche des informations d’un utilisateur recherche des jeux avec un nombre qui ont été joué par un utilisateur particulier recherche d’un jeu selon sa catégorie jeux favoris d’un utilisateur recherche des jeux qui ont été joué connexion d’un utilisateur Création de compte réception d’un fichier de sauvegarde réception d’un fichier de configuration

### **15.18.4 13h40**

Je fais un test d'appel a l'API depuis Caiman

### **15.18.5 14h20**

Pour pouvoir utiliser correctement les appels à l'api je dois utiliser des objets que je rempli avec chaque appel.

Par exemple, si je veux recevoir les informations d'un jeu je créer un objet jeu a qui je vais attribuer les données que je viens de recevoir.

Il me faut donc créer une classe pour l'utilisateur et une classe pour les jeux.

### **15.18.6 15h50**

je dois aussi pouvoir connaître la liste des catégories grâce à l'API

### **15.18.7 16h00**

Il est maintenant possible de récupérer les catégories

---

## **15.19 12.05.2021**

### **15.19.1 8h05**

Documentation et suppression de code inutile

### **15.19.2 13h00**

J'ai discuté avec M.Smid il m'a donner des conseil sur pour mon api

### **15.19.3 14h30**

J'ai changé les routes de mon api exemple : /games/userFavorites/8 => /games/?byUserFavorites=8

## **15.19.4 15h30**

J'ai essayé de mettre l'api en ligne mais quand j'arrive sur une page j'ai une erreur 500.

---

## **15.20 13.05.2021**

### **15.20.1 11h00**

L'erreur 500 que j'avais était liée à une version de PHP, le serveur avait une version de php 7.3 alors que mon API a besoin d'une version PHP minimum en 7.4. J'ai donc mis à jour la version présente sur le serveur.

### **15.20.2 12h00**

J'ai corrigé différentes erreurs liées à l'api par exemple quand on envoyait que le username et pas de password pour se connecter une erreur apparaissait.

### **15.20.3 13h00**

J'ai une erreur sur le serveur quand je veux faire une requête d'utilisateur avec sa clé d'API j'ai une erreur 404.

Je vais peut être passer la recherche d'api en POST et non en GET

## **15.21 14.05.2021**

### **15.21.1 8h00**

J'ai toujours une erreur quand je veux récupérer les informations d'un utilisateur grâce à son apitoken, l'erreur n'est présente que sur le caiman.cfpt.info. Pour corriger l'erreur je vais passer par du post pour éviter de perdre trop de temps.

### **15.21.2 09h22**

j'ai push sur le serveur une version de l'API qui fonctionne bien, je vais maintenant continuer ma documentation

### **15.21.3 09h30**

je me rend compte qu j'ai une requête qui ne marche pas

### **15.21.4 11h30**

J'ai dû modifier le .htaccess pour pouvoir appliquer les rewriteRules. Le fichier n'était pas pris en compte sur le serveur je l'ai donc mis dans le dossier ou point le virtual host.

### **15.21.5 12h00**

documentation

---

## **15.22 17.05.2021**

### **15.22.1 8h05**

Création d'une ébauche de diagram de classe sur le tableau.

J'ai repris en grande partie le diagramme de mon POC en y ajoutant une gestion des sauvegardes et des téléchargement.

### **15.22.2 9h00**

Création du schéma sous UMLetino

Première version du diagramme de la logique de Caiman.

### **15.22.3 13h20**

J'essaie de réfléchir à la manière de télécharger les sauvegardes et la façon dont je dois stocker les jeux que l'utilisateur a téléchargé.

Je pense que je vais faire en sorte de vérifier si les jeux qui devraient être présents sur le disque de l'utilisateur le sont réellement.

Je vais faire en sorte que l'utilisateur ait la possibilité de pouvoir télécharger les jeux dans un dossier spécifique. Il va devoir au premier lancement de l'application spécifier où l'installation doit se faire. (c'est peut être contraire à mon but de faire une application simple d'utilisation).

Je dois aussi savoir si le disque de l'utilisateur n'a pas la place requise pour télécharger le jeu demandé.

## **15.22.4 14h35**

Discussion avec M.Maréchal

## **15.23 18.05.2021**

### **15.23.1 8h05**

Je vais créer une page de connexion pour l'utilisateur

### **15.23.2 10h00**

La connection marche, je vais maintenant télécharger les images des jeux

### **15.23.3 12h00**

Je commence à faire en sorte que quand un nouveau jeu est reçu depuis l'API son image est automatiquement téléchargé et mise dans le dossier %appdata% de l'application

### **15.23.4 13h30**

J'affiche dans la sidebar le nom des catégories

### **15.23.5 15h00**

Quand je clique sur une catégorie les jeux de la catégorie choisie sont afficher.

---

## **15.24 19.05.2021**

### **15.24.1 8h05**

je fais en sorte d'afficher tous les jeux quand je lance l'application.

### **15.24.2 9h00**

je vais maintenant faire en sorte de pouvoir voir les détails d'un jeu.

### **15.24.3 9h30**

J'ai un problème avec les image bouton le clic n'est pas pris en compte

### **15.24.4 11h00**

j'ai corriger le soucis des boutons et je modifie un petit peu l'interface

### **15.24.5 13h00**

je commence l'affichage des détails d'un jeu

### **15.24.6 14h00**

je fais des recherche sur la façon de télécharger un jeu

---

## **15.25 20.05.2021**

### **15.25.1 8h05**

Je crée une route pour pouvoir recevoir le lien de téléchargement d'un jeu.

### **15.25.2 09h15**

Je me rend compte que pour créer la route de téléchargement je dois :

Avoir le lien du site Avoir le lien pour la console avoir le nom du fichier

Pour avec ces différents éléments je dois modifier mon API pour pouvoir faire tout ça.

### **15.25.3 11h00**

J'arrive à créer l'URL mais le fichier n'est pas accessible en dehors du serveur donc je ne sais pas vraiment quoi faire

### **15.25.4 12h50**

Correction d'un problème où l'on pouvait se déplacer dans une case qui n'existe pas donc l'application plantait.

### **15.25.5 14h00**

Je commence à faire en sorte que je puisse créer la route pour télécharger le fichier des jeux

---

## **15.26 21.05.2021**

### **15.26.1 9h00**

Je voulais télécharger un jeu en passant par le WebClient en C# mais ma route est en POST dans mon API. Malheureusement je ne peux pas fournir de paramètres en POST avec la fonction downloadFileAsync donc je dois passer ma route en GET

### **15.26.2 11h00**

j'ai changé ma route et j'en ai ajouté pour pouvoir connaître les dossiers où je dois stocker les jeux et le nom du fichier d'un jeu.

### **15.26.3 16h00**

J'ai réussi à afficher la liste des téléchargements et maintenant je suis en train de faire en sorte de pouvoir ajouter un jeu au favoris. j'ai réglé des soucis de droit d'accès au fichier qui sont en cours de lecture.

---

## **15.27 24.05.2021**

### **15.27.1 15h15**

les téléchargement se font maintenant l'un après l'autre

### **15.27.2 16h35**

Affichage des jeux téléchargé

---

## **15.28 25.05.2021**

### **15.28.1 8h10**

je dois modifier l'API pour pouvoir manipuler les jeux favoris des utilisateur

### **15.28.2 10h25**

il est maintenant possible d'ajouter et de supprimer un jeu des favoris. je vais maintenant faire en sorte de pouvoir exécuter des jeux depuis caiman.

### **15.28.3 10h30**

Je regarde le code que j'ai fais pour le poc pour voir si je peux reprendre des parties de codes.

### **15.28.4 14h00**

J'ai un soucis avec le lancement des jeux de PS2

### **15.28.5 14h25**

Le problème venait des paramètres j'avais oublié une espace entre le nom du fichier à exécuter et les paramètres

## **15.28.6 15h50**

J'ai pu ajouter les émulateurs et appliquer certains paramètres graphique.

---

## **15.29 26.05.2021**

### **15.29.1 8h05**

Je dois modifier mon api pour pouvoir générer un token à chaque connexion pour ne pas avoir à se connecter à chaque lancement de l'application.

### **15.29.2 10h30**

J'ai modifié l'API, je vais maintenant passer a Caiman.

### **15.29.3 11h00**

Il est maintenant possible de se connecter automatiquement , je vais maintenant modifier un petit peu l'aspect graphique de Caiman.

### **15.29.4 13h00**

je veux afficher le jeu qui est actuellement utilisé par l'utilisateur

### **15.29.5 15h30**

Il est possible de voir le jeu actuellement lancé par l'utilisateur et peut importe comment l'utilisateur quite le jeu Caiman va être au courant

### **15.29.6 18h00**

Affichage du temps de jeu de la session

## **15.29.7 21h00**

Le nombre de minutes de jeu a comptabilisé

---

## **15.30 27.05.2021**

### **15.30.1 8h10**

J'ai remarqué que je ne pouvais pas upload de jeu actuellement je vais essayer de comprendre pourquoi.

### **15.30.2 09h00**

J'ai résolu le problème je devais ajouter les droit d'écriture dans mon debian Par contre hier soir j'ai oublié de push donc je ne peux pas continuer Caiman ce matin

### **15.30.3 10h05**

Je profite de ne pas pouvoir coder pour faire des tests, ajouter des jeux et faire de la doc

### **15.30.4 23h30**

J'ai réussi à scanner un dossier pour savoir si il a y eu une modification.

---

## **15.31 28.05.2021**

### **15.31.1 9h30**

j'ai réussi à faire en sorte que les sauvegardes de l'utilisateur qui se connecte sont déplacé dans le dossier des sauvegardes des émulateurs.

### **15.31.2 10h00**

Je réfléchi à comment faire pour synchroniser les sauvegardes des utilisateurs. J'ai eu l'idée de créer un sous répertoire pour chaque utilisateur sur le serveur

###10h30

J'ai réfléchi et finalement je pense envoyer des fichier zip contenant toutes les sauvegardes d'un utilisateur et cela pour chaque console. Cela a l'avantage de réduire la taille des fichier et de simplifier l'upload et le téléchargement

### **15.31.3 14h00**

J'ai eu un bogue ou les sauvegarde n'était pas envoyé dans le bon dossier

### **15.31.4 14h35**

Documentation

---

## **15.32 30.05.2021**

### **15.32.1 13h30**

Modification de l'API pour pouvoir télécharger un fichier de sauvegarde

### **15.32.2 16h12**

Je fais en sorte que les sauvegardes se synchronisent sur caiman

### **15.32.3 19h43**

j'ai ENFIN réussi à synchroniser les sauvegardes de Gamecube mais j'ai un soucis avec celle de ps2

---

## **15.33 31.05.2021**

### **15.33.1 8h05**

Je commente le code de Caiman

### **15.33.2 14h00**

Discussion avec M.Maréchal et ajout de tests pour la classe GameTimer

---

## **15.34 01.06.2021**

### **15.34.1 8h05**

Ajout de tests pour la classe TimeInGame

### **15.34.2 8h20**

Je veux faire en sorte que Caiman s'exécute sur l'écran principale du pc de l'utilisateur

### **15.34.3 8h30**

J'ai désactiver le .gitIgnore par ce que cela créait des soucis avec les dossier des émulateurs

---

## **15.35 02.06.2021**

### **15.35.1 8h05**

Commentaire du code de l'API

## **15.35.2 10h30**

documentation

## **15.35.3 15h30**

J'ai corrigé le fait que un compte pouvait être créé sans remplir tous les champ

---

# **15.36 03.06.2021**

## **15.36.1 8h10**

J'ai eu un problème de conflit avec mon git j'ai du re télécharger tout le projet mais maintenant c'est bon.

## **15.36.2 8h34**

Je continue la documentation

---

# **15.37 04.06.2021**

## **15.37.1 8h05**

Documentation

## **15.37.2 13h15**

La documentation avance bien je vais donc créer les diagrammes de classe pour le C#.

---

## **15.38 07.06.2021**

### **15.38.1 8h10**

Recherche sur comment exporter ma documentation en pdf et je continue ma doc.

### **15.38.2 11h20**

Je vais essayer d'installer un certificat ssl sur le serveur debian

### **15.38.3 13h30**

Je n'arrive pas à installer certbot donc je laisse ça en attente

### **15.38.4 14h00**

Documentation