

# **Portfolio Corentin Borde**

Version 1 - Octobre 2020

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Cursus universitaire</b>	<b>3</b>
2.1	Licence 1 . . . . .	3
2.2	Licence 2 . . . . .	3
2.3	Licence 3 . . . . .	3
2.4	Master 1 . . . . .	3
2.5	Master 2 . . . . .	3
<b>3</b>	<b>Stage de Licence 1</b>	<b>4</b>
<b>4</b>	<b>Stage de Licence 3</b>	<b>5</b>
<b>5</b>	<b>Projet de recherche documentaire de Licence 2</b>	<b>6</b>
<b>6</b>	<b>Projet de web en Licence 2</b>	<b>7</b>
<b>7</b>	<b>Projet tutoré de Licence 3 « Application web de prise de rendez-vous médicaux par géolocalisation »</b>	<b>14</b>
<b>8</b>	<b>Projet web en Licence 3</b>	<b>21</b>
<b>9</b>	<b>Projet de développement agile de machines virtuelle en Master 1</b>	<b>25</b>
<b>10</b>	<b>Développement d'une application mobile Androïd en Master 1</b>	<b>27</b>
<b>11</b>	<b>Donkey Kong</b>	<b>33</b>

# 1 Introduction

Bienvenue sur mon portfolio personnel.

Je suis Corentin Borde, actuellement en Master 2 ISL (Ingénierie Systèmes et Logiciels) Informatique. Je suis actuellement en recherche de stage d'une durée de 16 semaines minimum à partir de Février 2021.

Ce portfolio a pour but de regrouper mes connaissances et compétences obtenues durant ma formation à l'université de Franche-Comté à Besançon. Je montre également en détail quelques projets ainsi que mes stages déjà réalisés.

## 2 Cursus universitaire

Durant ma formation, j'ai pu découvrir et apprendre différentes notions dans différents domaines de l'informatique. Je fait ici un petit récapitulatif de tout ce qui a été vu, avant de présenter des projets et les stages réalisés.

### 2.1 Licence 1

- Bases de la programmation : Java et réalisation d'un jeu : le casse brique ;
- Base de données : SQL, phpMyAdmin, Access, et mini projet en Java ;
- L<sup>A</sup>T<sub>E</sub>X.

### 2.2 Licence 2

- Architecture des ordinateurs : schéma logiques et assemblleur ;
- Diagrammes UML ;
- Langage c ;
- Programmation système sous UNIX : shell, makefile, c ;
- Programmation web (voir partie 6).

### 2.3 Licence 3

- Outils pour la programmation : IDE, Subversion et Git, cmake, tests unitaires, tests fonctionnels, tests d'acceptation ;
- Réseau : c et sockets c ;
- Théorie des Langages : automates et mini-projet en c++ ;
- Analyse Syntaxique : bison/flex et c ;
- Web Avancé : voir partie 8 ;
- XML.

### 2.4 Master 1

- Développement mobile : Androïd Studio, Java, Kotlin (voir partie 10) ;
- Compilation et Interprétations : voir partie 9 ;
- Théorie des graphes : Java ;
- Informatique Graphique : c++, GLUT ;
- Bases de la programmation IA : prolog, et mini projet : java, c, sockets, prolog.

### 2.5 Master 2

Année en cours, des détails arriveront bientôt !

### **3 Stage de Licence 1**

## **4 Stage de Licence 3**

## **5 Projet de recherche documentaire de Licence 2**

Un projet de recherche documentaire a été réalisée sur tous le long de l'année de Licence 2. Il s'est effectué par groupe de 2 étudiants et s'effectuait en présence de deux tuteurs universitaire.

Notre projet s'intitulait « État de l'art des ontologies médicales de l'abdomen ». Lors de notre recherche, nous avons découvert la segmentation d'images médicales, le raisonnement à partir de cas (RàPC) et ce qu'est une ontologie.

Le RàPC est une méthode permettant de résoudre un problème grâce à l'expérience, l'analogie et la réutilisation de problèmes déjà résolus.

Une ontologie (en informatique) est est l'ensemble structuré des termes et concepts représentant le sens d'un champ d'informations.

Dans le cadre de ce projet de recherche documentaire, nous avons développé notre propre ontologie pour situer les structures anatomiques de l'abdomen les unes par rapport aux autres. Pour celà, nous avons utilisé le logiciel Protege<sup>1</sup>.

Le rapport ainsi que le diaporama pour la soutenance réalisés dans le cadre du projet se trouve dans le dossier **ProjetTutore**.

---

1. <https://protege.stanford.edu/>

## **6 Projet de web en Licence 2**

### **Description**

Ce projet a été développé comme mini-projet pour le module « Langages du web » de Licence 2. Le but était de développer un site de e-commerce de livres nommé « BookShop ». Ce projet a été réalisé par groupe de 2 étudiants.

Les langages du web HTML, CSS et PHP ont été utilisés pour développer ce projet. Une base de données avec phpMyAdmin était utilisée. La charte graphique nous était imposée.

Les sources de ce projet se trouvent dans le dossier `Bookshop`.

## Démonstration



FIGURE 1 – Page d'accueil de Bookshop

**Page d'accueil** La page d'accueil (figure 1) du site montre quelques livres tirés de la catégorie « Nouveautés » et « Top des ventes ». Cette page est la même pour des utilisateurs connectés ou non.

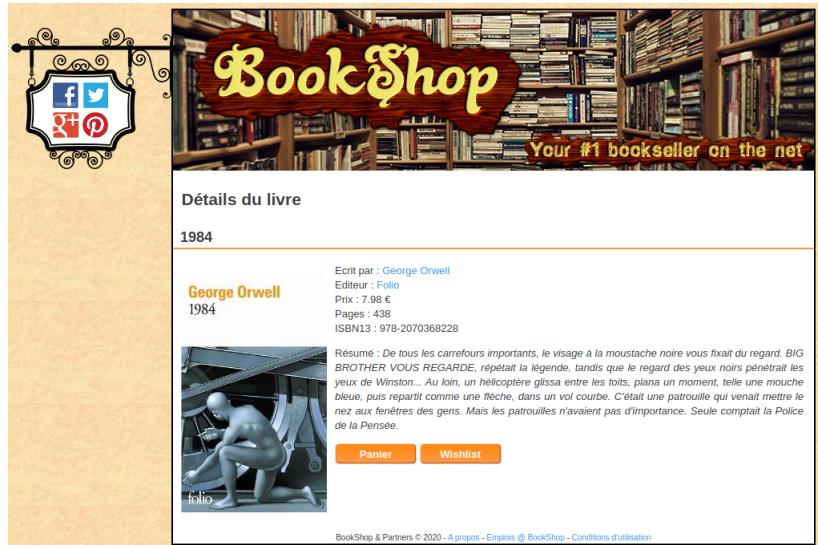


FIGURE 2 – Détail livre

**Détails sur un livre** Cette page permet de visualiser les détails sur un livre. Des boutons permettent de l'ajouter au panier ou sur la liste de vœux seulement si l'utilisateur est connecté.



FIGURE 3 – Page de connexion

**Connexion et inscription** La page de connexion (figure 3) permet à un utilisateur de se connecter. Si il n'a pas de compte, il peut en créer un :

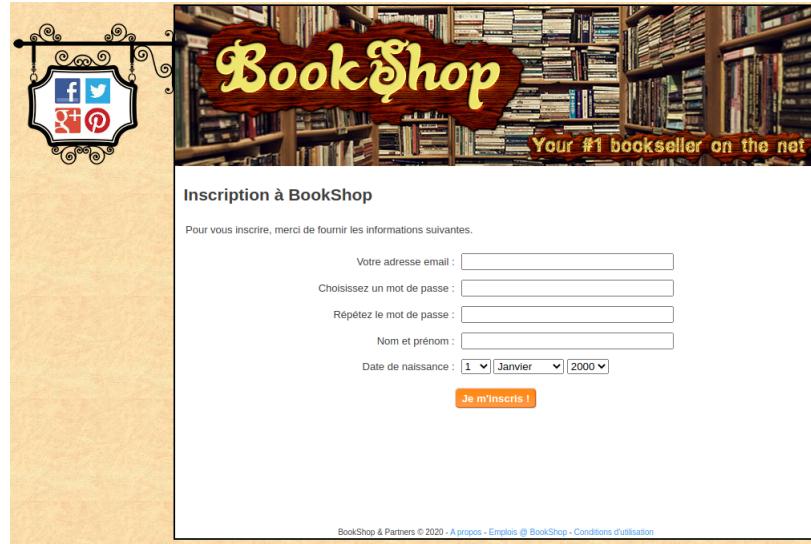


FIGURE 4 – Inscription

The screenshot shows a user profile editing interface for 'BookShop'. At the top, there's a decorative banner with the text 'Your #1 bookseller on the net' and a background image of bookshelves. On the left, a sidebar features a sign with social media icons (Facebook, Twitter, Google+, Pinterest) and a parchment-style background. The main content area has a white background and contains two sections: 'Mes informations personnelles' and 'Informations du compte'. The 'Informations du compte' section includes fields for email (test@test.fr), name (Test Corentin), birth date (8 Août 1998), password creation, and password repetition. The 'Informations de livraison' section includes fields for address (16 route de Gray), postal code (25000), city (Besançon), and country (France). A 'Modifier !' button is located at the bottom right of this section.

Mes informations personnelles

Accédez au récapitulatif des commandes !

Informations du compte

Votre adresse email :

Nom et prénom :

Date de naissance :

Choisissez un mot de passe :

Répétez le mot de passe :

Informations de livraison

Adresse :

Code postal :

Ville :

Pays :

**Modifier !**

FIGURE 5 – Informations Personnelles

**Informations personnelles** La page des informations personnelles (figure 5) permet à un utilisateur de visualiser et modifier ses informations personnelles.

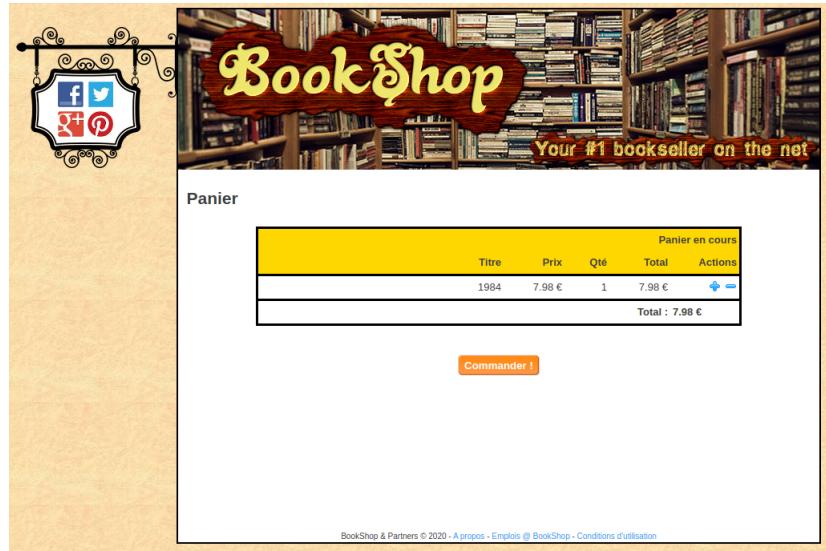


FIGURE 6 – Panier

**Panier** Cette page (figure 6) permet à l'utilisateur de visualiser les articles qu'il aurait pu ajouter à son panier. Le bouton « Commander » lui permettrait de passer une commande, mais l'implémentation du paiement n'a pas été réalisée sur ce projet.



FIGURE 7 – Liste de vœux

**Liste de vœux** Cette page (figure 7) permet à l'utilisateur de voir tous les articles qu'il aurait pu mettre dans sa liste depuis qu'il a créé son compte. Il peut également visualiser la liste de vœux d'un autre utilisateur en faisant une recherche avec l'email de la personne.

## **7 Projet tutoré de Licence 3 « Application web de prise de rendez-vous médicaux par géolocalisation »**

### **Description**

Ce projet a été développé dans le cadre du projet tutoré de Licence 3, sur toute l'année par groupe de 3 étudiants. Le but était de développer une application web où un utilisateur peut rechercher des médecins selon leur spécialités autour de sa position. Ce même utilisateur doit avoir la possibilité de prendre un rendez-vous sur un créneau qu'un médecin aurait préalablement ajouté dans son emploi du temps.

Nous avons appelé notre application « Géo'doc ». Ce projet a été réalisé avec les langages du web HTML, CSS, PHP, JavaScript. Nous avons utilisé phpMyAdmin pour notre base de données. Pour afficher un fond de carte et y placer des repères, nous avons utilisé une API Open Source qui s'appelle Leaflet<sup>2</sup>. Cette API s'utilise avec JavaScript, donc pour récupérer les données de la base de donnée, nous avons fait des appels AJAX.

Les sources de ce projet, ainsi que le rapport et le diaporama de la soutenance se trouvent dans le dossier **GeoDoc**. À la source du dossier se trouve le code source de la page **Index.php** ainsi que le script SQL pour créer la base de données. Tous les autres fichiers source sont rangés dans les dossiers correspondants. Les images utilisées pour l'identité visuelle sont rangées dans le dossier **images**.

---

2. Leaflet. Leaflet, a javascript library, <https://leafletjs.com/>

## Démonstration

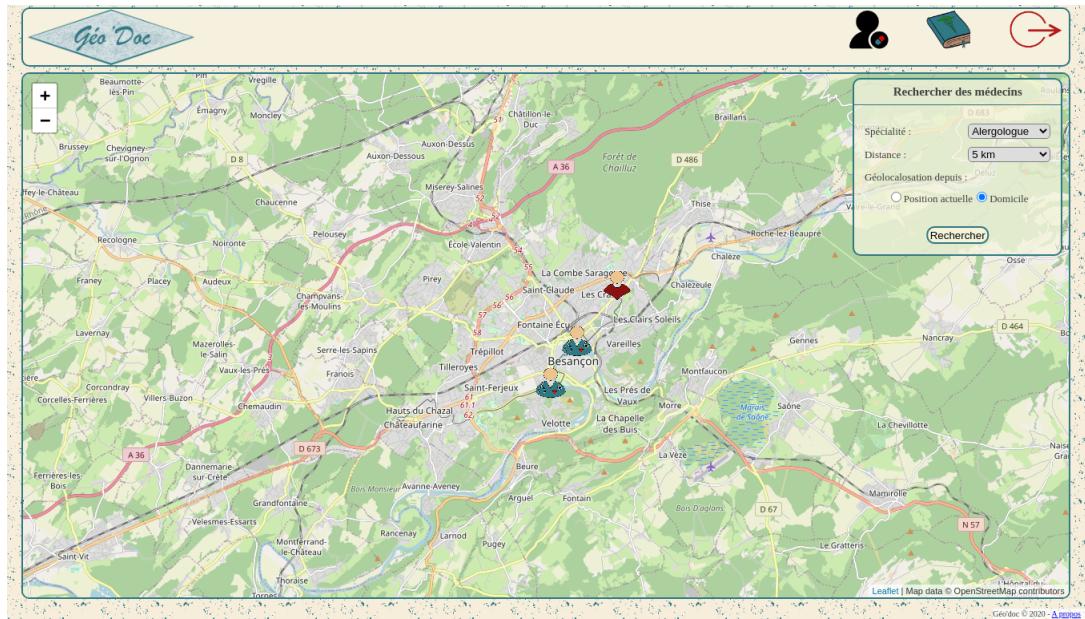


FIGURE 8 – Page d'accueil de Géo'doc

**Page d'accueil** La page d'accueil (figure 8) est la page qui affiche un fond de carte, avec la position courante de l'utilisateur. En haut à droite, se trouve un petit formulaire de recherche. L'utilisateur peut choisir une spécialité de médecin, ainsi qu'une distance maximale de recherche. Si il est connecté et qu'il a déjà rempli une adresse postale, alors il peut effectuer la recherche autour de son domicile et non de sa position courante.

La figure 9 montre les informations d'un médecin lors d'un clic sur son repère.

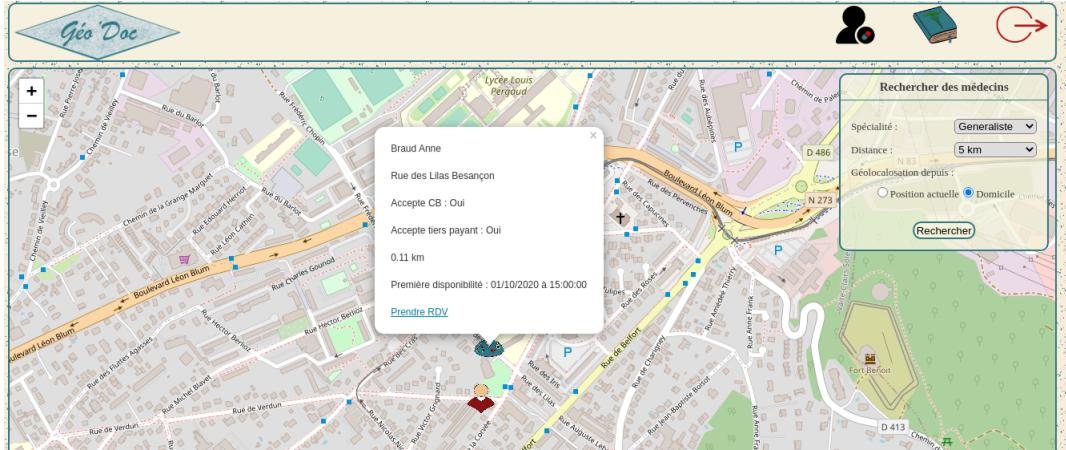


FIGURE 9 – Repère avec information sur un médecin

FIGURE 10 – Page de profil d'un médecin

**Profil Utilisateur** La page du profil utilisateur (figure 10) permet à un utilisateur de voir et modifier ses informations personnelles. Cette page est la même pour les patients et les médecins. La seule différence est que les patients n'ont pas la cadre « Profil Médical »

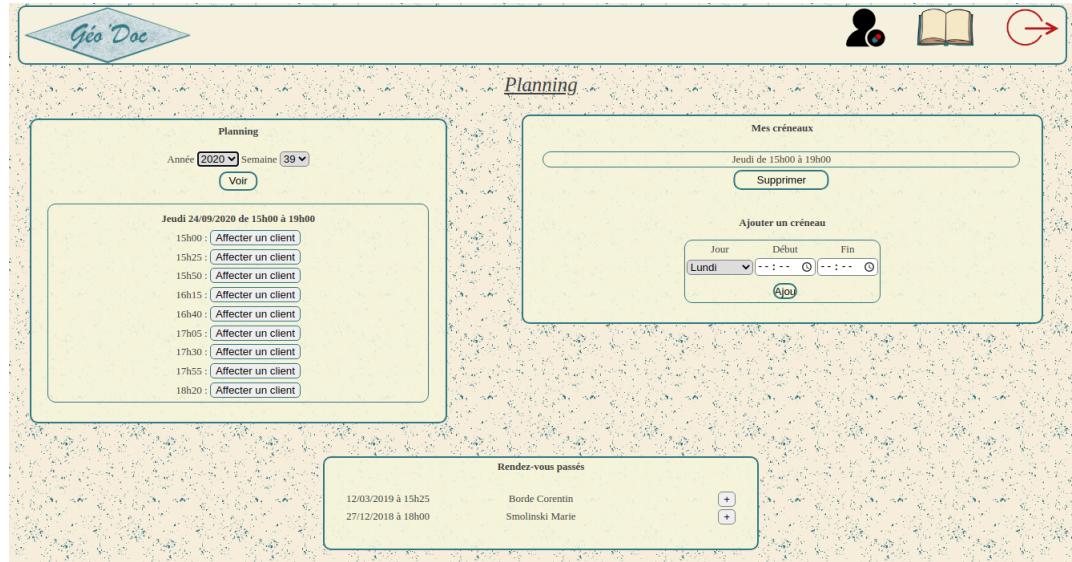


FIGURE 11 – Planning d'un médecin

**Planning Médecin** La page du planning (figure 11) est une page réservée aux utilisateurs médecins. Elle permet à ceux-ci d'ajouter des créneaux à leur emploi du temps. Ces créneaux sont répétés chaque semaine.

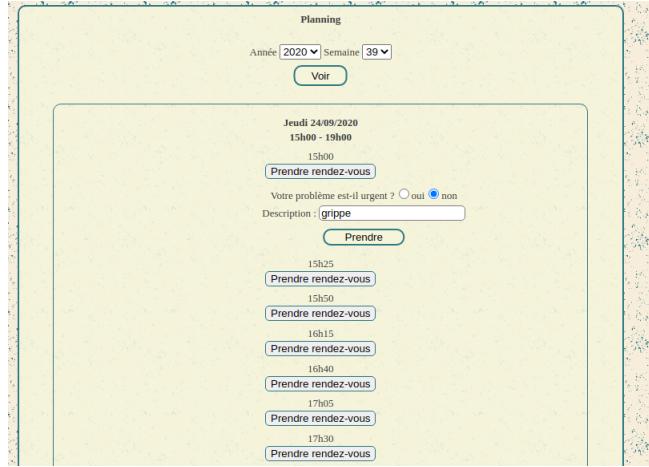


FIGURE 12 – Prendre un Rendez-vous

**Prendre Rendez-vous Patient** La page de prise de Rendez-vous (figure 12) permet à un patient de prendre un rendez-vous auprès d'un médecin sur un créneau que le médecin a ajouté.

La figure 13 montre cette même page après qu'un patient ait pris un rendez-vous.

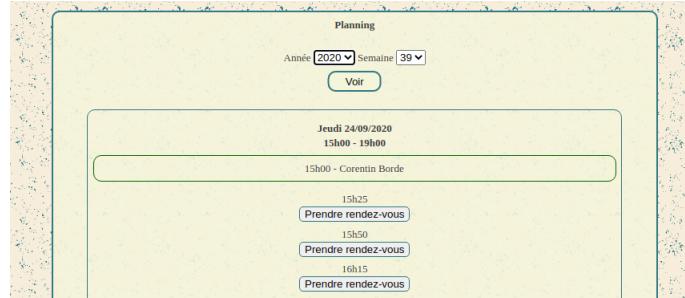


FIGURE 13 – Page Rendez-vous après en avoir pris un

**Mes Rendez-vous** La page de visualisation des rendez-vous (figure 14) est réservée aux utilisateurs patient. Elle permet de voir les rendez-vous à venir ainsi que les rendez-vous passés.



FIGURE 14 – Mes Rendez-vous

**Page de connexion, d'inscription** Cette page permet à un utilisateur de se connecter, ou de créer un compte.

The screenshot shows a split-screen interface for the Géo Doc application. On the left side, there is a 'Connexion' (Login) form with fields for 'Login:' and 'Mot de passe:' (Password), and a 'Se connecter' (Connect) button. On the right side, there is an 'Inscription' (Registration) form. A red box highlights the top section of the registration form, which contains the text 'Pour un médecin l'inscription se fait ici.' (For a doctor, registration is done here). The registration form includes fields for 'Login:', 'Mot de passe:', 'Confirmer le mot de passe:' (Confirm password), 'Nom:' (Name), 'Prénom:' (First name), 'Email:', 'Téléphone:' (Phone), and an 'Adresse' (Address) section with fields for 'Numéro de rue:', 'Rue:', 'Complément (bis, A, ...)', 'Code postal:', 'Ville:', and 'Pays:'. At the bottom of the registration form is an 'Inscription' (Registration) button.

FIGURE 15 – Connexion, Inscription

## **8 Projet web en Licence 3**

### **Description**

Ce projet a été développé comme mini-projet pour le module « Web avancé » de Licence 3. Le but était de développer une application web de dessin, chat, du type Pictionary en ligne. Nous avons appelé ce projet « Netwok ». Ce projet a été réalisé par groupe de 2 étudiants.

Les langages du web HTML, CSS, JavaScript ainsi que NodeJS ont été utilisés pour développer ce projet.

Les sources de ce projet se trouvent dans le dossier `netwok`.

Pour notre projet, le but n'est pas de faire deviner n'importe quel dessin mais des syllabes japonaises.

## Démonstration

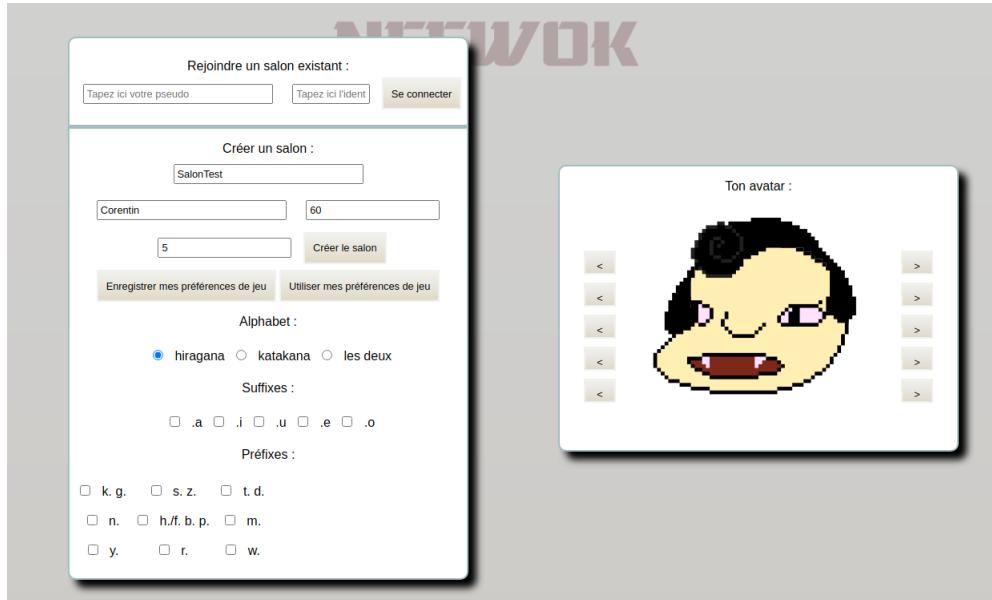


FIGURE 16 – Page d'accueil

**La page d'accueil** La page d'accueil (figure 16) permet à un utilisateur de créer un salon, ou de se connecter à un salon déjà existant qu'un autre joueur aurait déjà créé et ayant récupéré l'ID de celui-ci.

Lors de la création d'un salon, l'utilisateur à la possibilité d'enregistrer ces préférences dans le navigateur pour pouvoir les réutiliser plus tard.

L'utilisateur a également la possibilité de personnaliser son avatar.



FIGURE 17 – Salon de Jeu

**Le salon de jeu** Le salon de jeu (figure 17) est composée de 3 grandes parties :

- La partie de gauche montre tous les joueurs connectés ainsi que leurs scores respectifs;
- La partie du milieu est le canevas de dessin ;
- La partie de droite est le chat, où les joueurs peuvent s'envoyer des messages et aussi faire leurs propositions. Si un joueur fait une bonne proposition, le serveur la détecte et lui attribue les points.

Le nom, l'Id du salon ainsi que le temps restant de la manche sont notés en haut.

Sur cette image, on voit également que le joueur qui doit dessiner peut choisir la syllabe à faire deviner.

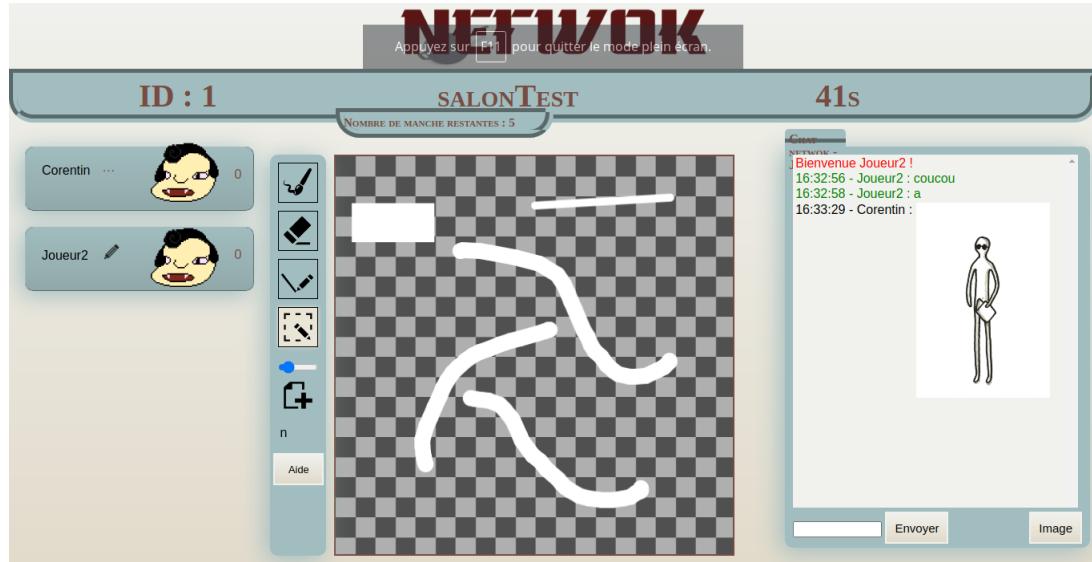


FIGURE 18 – Salon de jeu : canvas et chat

La figure 18 est une capture d'écran prise en pleine partie. Le joueur actuel est en train de dessiner sur le canvas. Il a la possibilité de dessiner avec le crayon, de faire des lignes droites et des rectangles. Il peut utiliser la gomme ou tout effacer. Il est aussi possible de choisir la taille du trait (stylo, gomme et ligne droite). Un bouton d'aide lui permet de voir le symbole japonais correspondant à la syllabe choisie (avec un malus de point). Les joueurs ont la possibilité de communiquer au moyen de la fenêtre de chat. Ils peuvent envoyer des messages, faire des propositions et ils peuvent même mettre des gifs.

## 9 Projet de développement agile de machines virtuelle en Master 1

Ce projet a été développé dans le cadre des modules Compilation et Génie Logiciel en Master 1. Le but était de développer un IDE (Integrated Development Environment) simplifié qui serait capable de :

- reconnaître un langage simplifié ;
- le compiler en un langage de bas niveau type assembleur ;
- d'interpréter le code résultant de la compilation ;
- de gérer l'état de la mémoire pendant l'interprétation ;
- de permettre à l'utilisateur d'exécuter son programme en mode débug pas à pas.

Ce projet a été développé sur un semestre par groupe de 6 étudiants.

Nous avons développé ce projet en Java avec Maven. Pour le lexer/parser, nous avons utilisé Antlr4<sup>3</sup>.

Nous avons utilisé les techniques agiles :

- découpage du projet en sprint de 2 semaines ;
- réunions de début et fin de sprint ;
- une première release en milieu de semestre ;
- utilisation de Trello, pour décrire et se répartir les tâches, ainsi que pour décrire le backlog du produit pour chacune des releases ;
- utilisation de Subversion comme gestionnaire de version
- utilisation de Jenkins comme serveur d'intégration continue.

Un exécutable *.jar* ainsi qu'un exemple de code source compilable et interprétable par notre IDE se trouve dans le dossier IDE, le rapport et le diaporama de la soutenance s'y trouvent également. Il suffit de taper dans un terminal la commande suivante pour l'exécuter : `java -jar IDE\ MiniJaja.jar`. Le code source écrit est un fichier avec l'extension *.mjj*, il doit être sauvegardé sur le disque, la compilation va produire un fichier avec l'extension *.jjc*.

---

3. <https://www.antlr.org/>

The screenshot shows the IDE Minjaja window. The left pane contains C-like source code:

```

1 class exemple2{
2     int x = 0;
3     int t[4];
4
5     int fct(int max){
6         int y = 5;
7         while (max > 0){
8             y+=max;
9             max = max -1;
10        }
11        return y;
12    }
13
14    main{
15        while (4 > x){
16            t[x] = fct(1);
17            writeln(t[x]);
18            x++;
19        }
20    }
21 }

```

The right pane shows the generated assembly code:

```

1 init
2 push(0)
3 new([.exemple2.]x,entier,var,0)
4 push(4)
5 newarray([.exemple2.]t,entier)
6 push(9)
7 new([.exemple2.]fct_entier,entier,meth,0)
8 goto(29)
9 new([.exemple2.fct_entier.]max,entier,var,1)
10 push(0)
11 new([.exemple2.fct_entier.]y,entier,var,0)
12 load([.exemple2.fct_entier.]max)
13 push(0)
14 sup
15 not
16 if(24)
17 load([.exemple2.fct_entier.]max)
18 inc([.exemple2.fct_entier.]y)
19 load([.exemple2.fct_entier.]max)
20 push(1)
21 sub
22 store([.exemple2.fct_entier.]max)
23 nnto(12)

```

The bottom console area shows the output of the program execution:

```

6
6
6
6

```

FIGURE 19 – Capture d'écran de l'IDE

La figure 19 montre une capture d'écran de notre IDE. Il est possible pour l'utilisateur d'écrire son code source dans la partie de gauche. Si il compile ce code, le code source de bas niveau généré sera affiché sur la partie de droite. La partie du bas est la console, elle affiche si la compilation s'est bien passée ou signale les erreurs. Elle affiche également le résultat d'exécution du programme. Tout en haut, il y a un menu, il est possible de :

- sauvegarder un fichier ;
- ouvrir un fichier ;
- de mettre l'IDE dans un thème *dark* ;
- de fermer l'IDE ;
- de lancer l'exécution du programme pas à pas ou par *breakpoints* que l'utilisateur aurait ajouté en cliquant sur le numéro de ligne dans le programme.

## **10 Développement d'une application mobile Androïd en Master 1**

### **Description**

Ce projet a été développé dans le cadre du module de programmation Androïd en Master1. Le but était de réaliser une application pour pouvoir jouer à des quiz. L'utilisateur à la possibilité de :

- jouer à des quiz en ligne ;
- créer des quiz locaux ;
- gérer des quiz locaux déjà existant ;
- jouer à des quiz locaux ;
- sauvegarder ses scores ;
- voir les scores.

Ce projet a été développé en langage Kotlin avec l'IDE Androïd Studio et seul. Le code source ainsi qu'un rapport se trouve dans le dossier **quizAndroid**.

## Démonstration



FIGURE 20 – Menu

**Menu principal** La figure 20 montre le menu principal de l'application.



FIGURE 21 – Interface de jeu

**Jouer** La figure 21 montre l’interface de jeu. Elle est la même que ce soit pour jouer avec un quiz en ligne ou un quiz en local. Les quiz en ligne sont stockés sur un serveur externe (celui de l’Université pour se projet) et sous forme XML. Les quiz en local sont stockés dans une base de données SQLiteDataBase.

**Score** L'utilisateur à la possibilité de sauvegarder son score (figure 22).



FIGURE 22 – Sauvegarde du score

L'utilisateur peut aussi consulter tous les scores sauvegardés (figure 23).

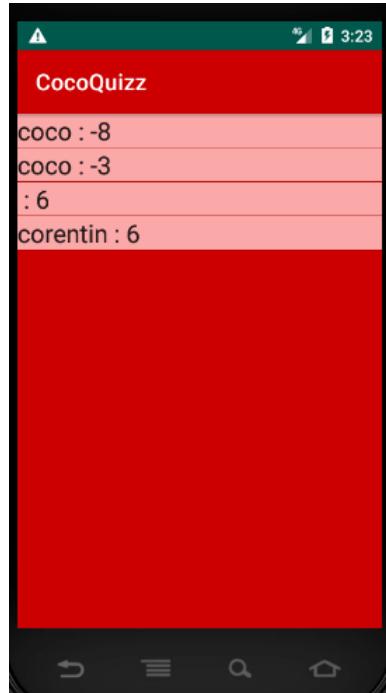


FIGURE 23 – Scores

**Gestion des quiz** L'utilisateur à la possibilité de créer des quiz qui seront stockés en local. Tout d'abord il doit remplir le thème et le nombre de question (figure 24).

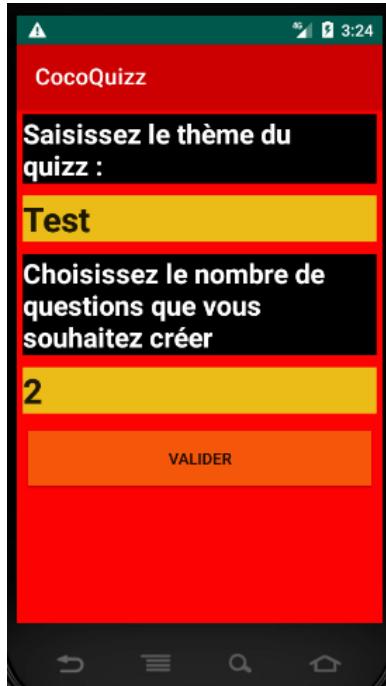


FIGURE 24 – Création d'un quiz - thème et nombre de questions

Il va pouvoir ensuite créer les questions. Par la suite, il aura la possibilité de modifier les questions ou d'en supprimer. La figure 25 montre une capture d'écran de la modification d'une question existante. Lors de la création d'un quiz, l'interface d'ajout de question est presque la même, il y a juste un champ de plus pour que l'utilisateur puisse choisir quelle proposition est la bonne.

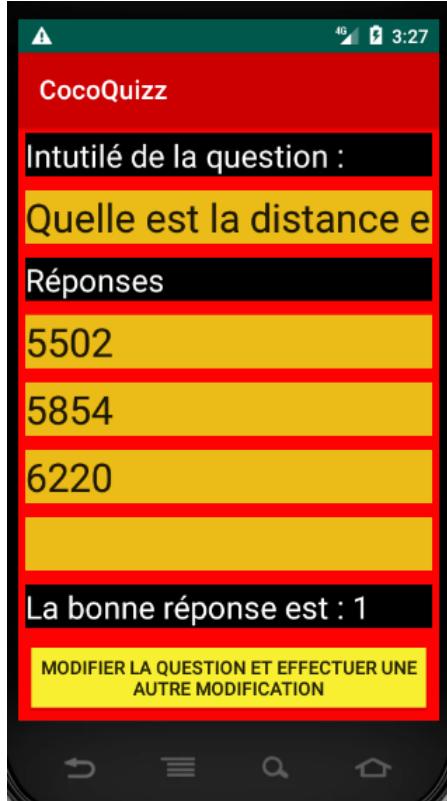


FIGURE 25 – Modification d'une question

## 11 Donkey Kong

## Résumé

- Programmation orientée objet (Java et C++) ;
- Gestion de base de données relationnelles (SQL, MySQL, Access) ;
- Programmation web : langages du web (HTML, CSS, PHP, JS, Node JS) et base de donnée phpMyAdmin ;
- Programmation système sous linux (Shell) ;
- Méthode agile, Génie Logiciel, tests unitaires, tests fonctionnels, tests d'acceptation ;
- Réseau : sockets en c et Java, gestion mini-réseaux rooteur, switch en console ;
- Compilation et interprétation, théorie des langages (lexer, parser, Antlr4, bison, flex) ;
- Programmation Androïd (Java, Kotlin, Androïd Studio) ;
- Bases de l'IA (prolog) ;
- **L<sup>A</sup>T<sub>E</sub>X**