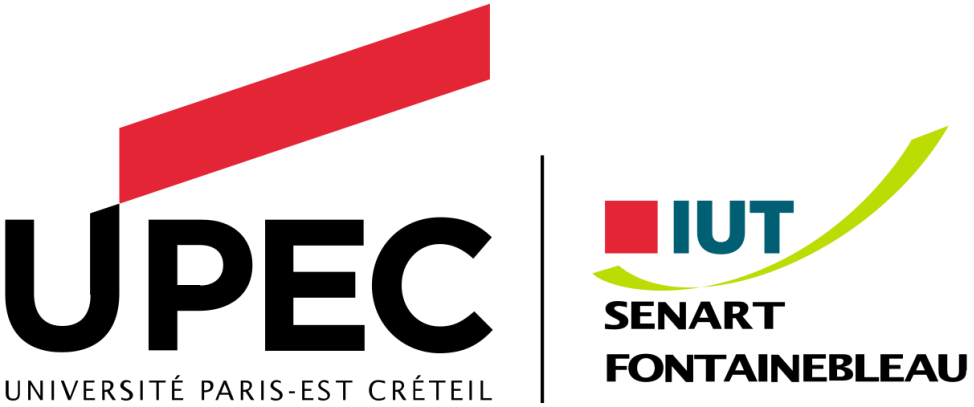
08/12/2022

Projet IHM



Département informatique

Formation : BUT2 FI

Rapport de projet

Réalisé par :
Youcef SAYAD
Yanis BOUARROUDJ
Adam MEDDAHI

Responsable de projet :
Florent MADELAINE
Luc HERNANDEZ

Sommaire

Introduction

Ce projet consistait en la réalisation d'un logiciel de gestion de groupes et de deux logiciels pour les utilisateurs. Pour ce faire, nous avons structuré notre code en respectant le modèle MVC. Nous avons aussi mis en place la persistance des données via une base de données. Tout au long de ce projet, nous avons essayé de comprendre et de respecter les concepts de programmation orientée objets rencontrés.

Les commandes afin de lancer les logiciels sont :

- make admin
- make eleve
- make prof

1. Interfaces utilisateur

Nous avons divisé les interfaces en deux parties :

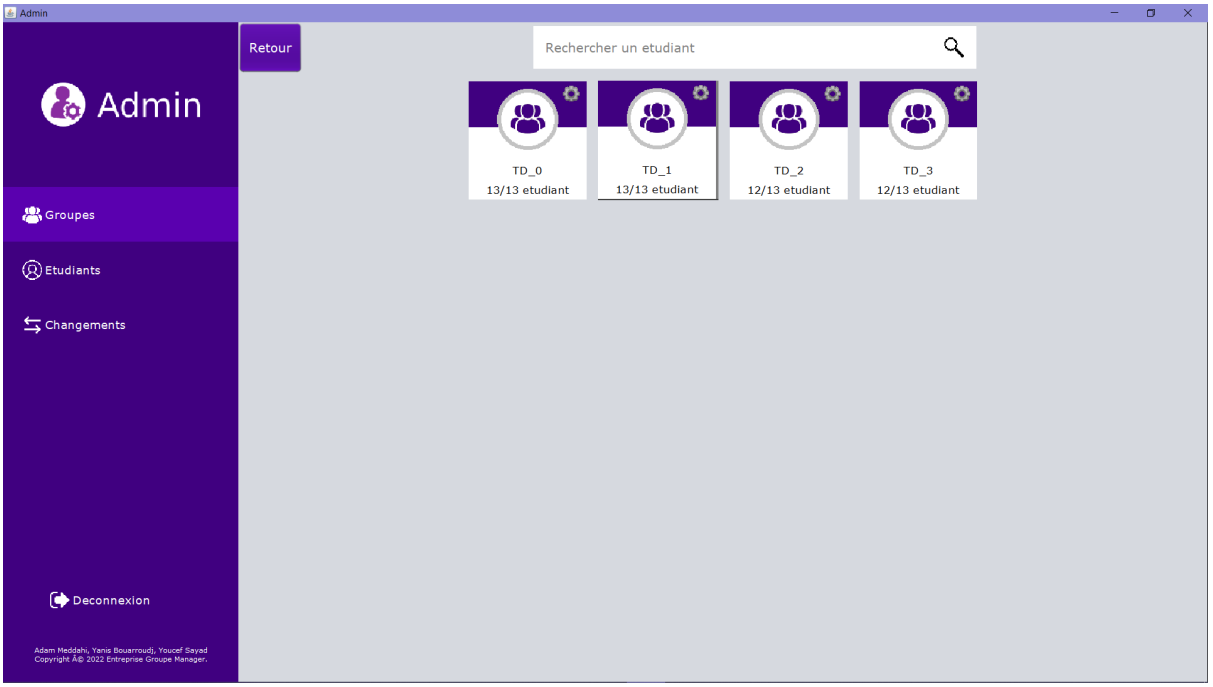
- Un dock de navigation
- Un panel d'interaction
-

Nous avons fait en sorte que le logiciel soit le plus intuitif possible. Nous avons utilisé des concepts connus afin de ne pas dépayser l'utilisateur (par exemple l'affichage sous forme de cartes). Nous avons aussi respecté des heuristiques et des règles d'IHM (gnome et Jakob Nielsen) tels que le sens de lecture habituel d'une page en occident, un dock permettant de naviguer, des boutons cliquables, des dessins, un haut contraste et une police facilement lisible, des spin-buttons etc.

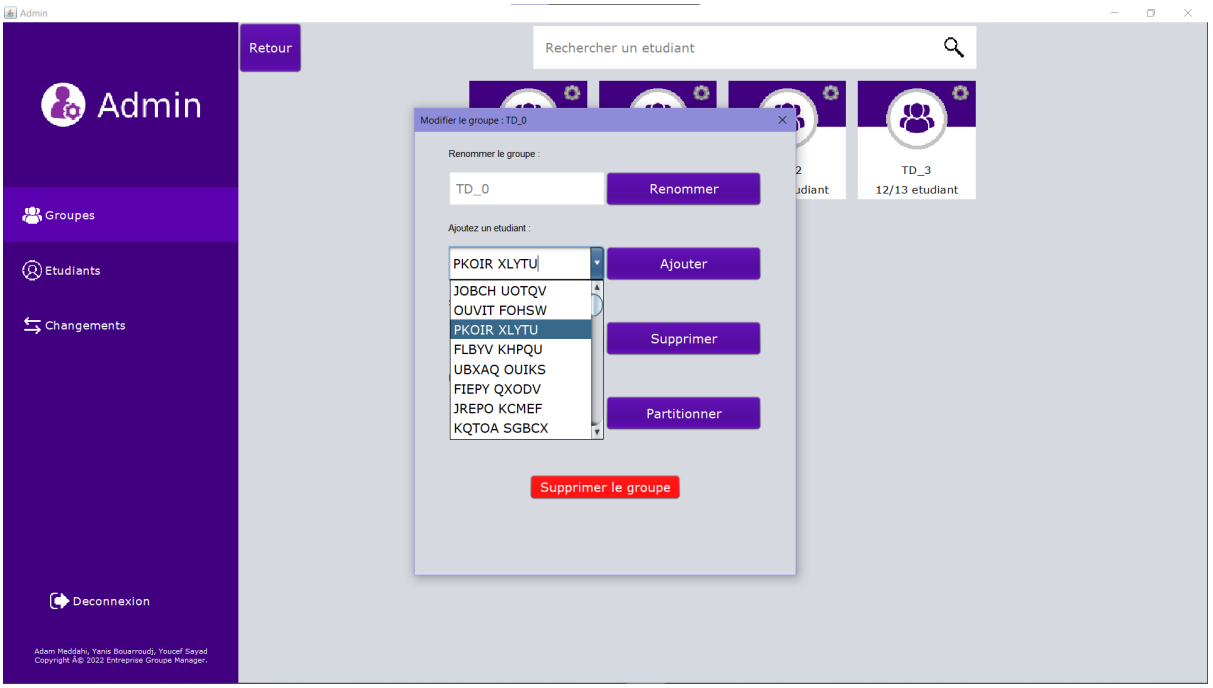
Nous nous sommes appliqués à rendre la navigation la plus fluide et naturelle possible. Par exemple, dans le logiciel Administrateur, nous avons permis une navigation récursive à travers les groupes, en effet, cliquer sur un groupe nous amène à ses sous-groupes. Cela rend l'expérience utilisateur légère et agréable.

Admin (images disponible sur le Git) :

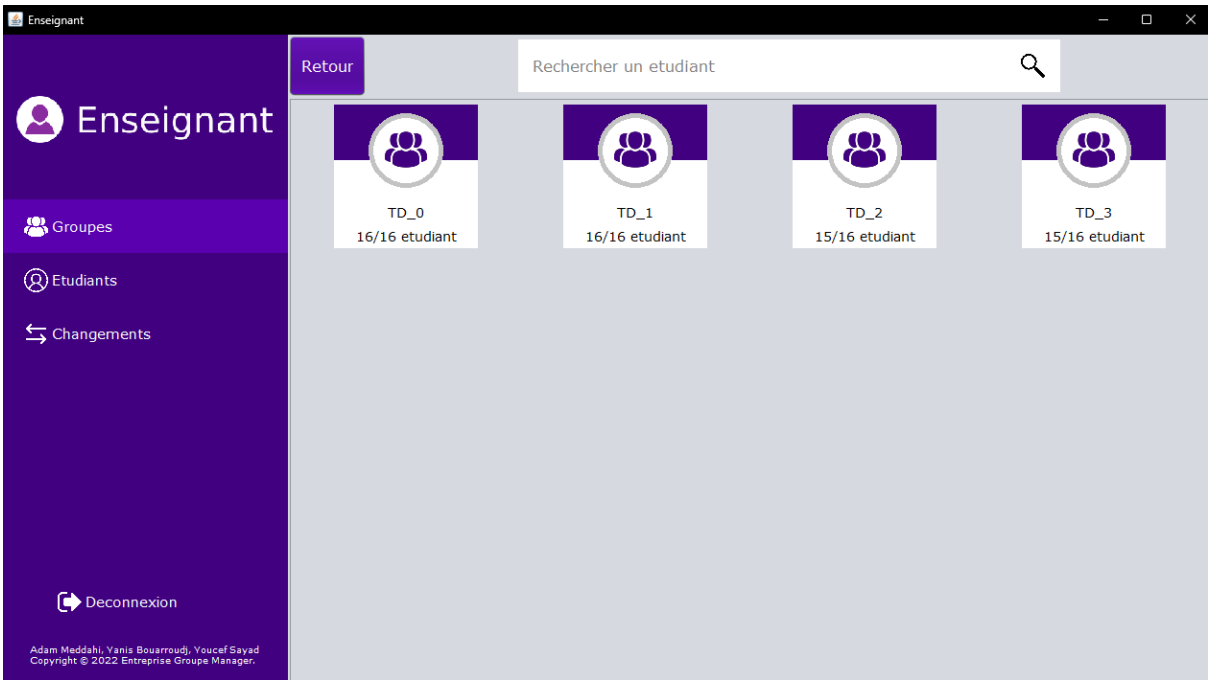
L'administrateur navigue dans l'arborescence des groupes en cliquant sur les cartes

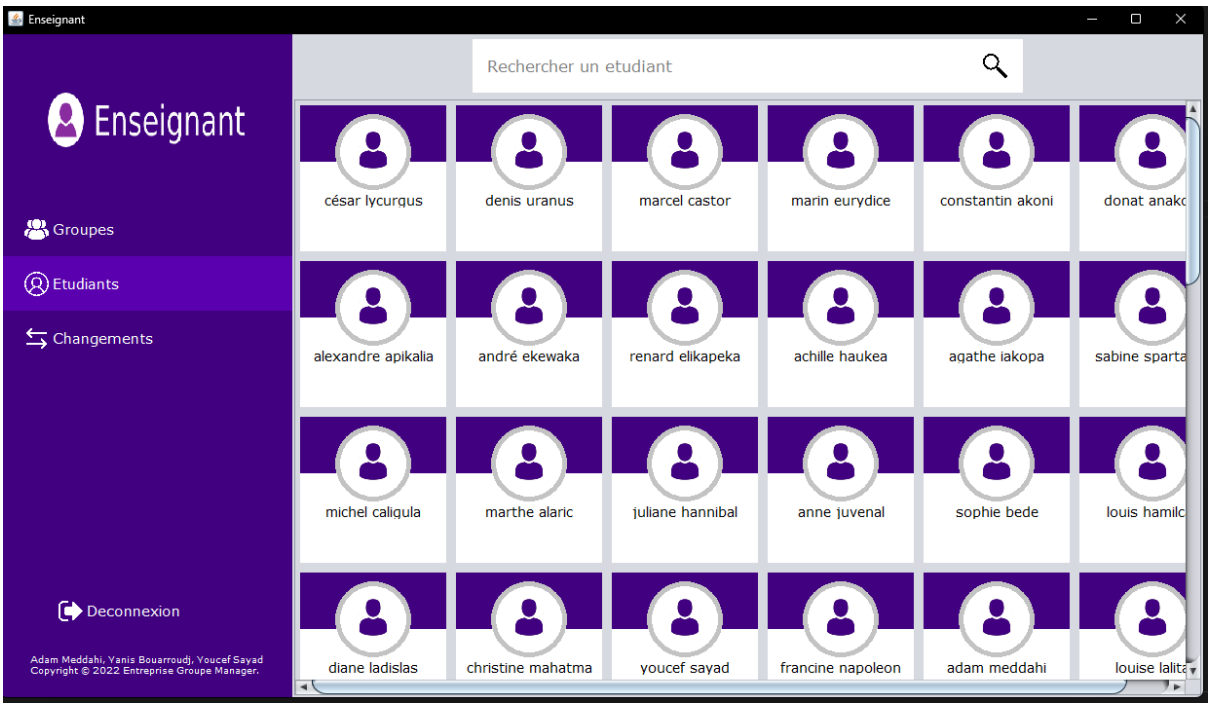


L’administrateur peut modifier les groupes

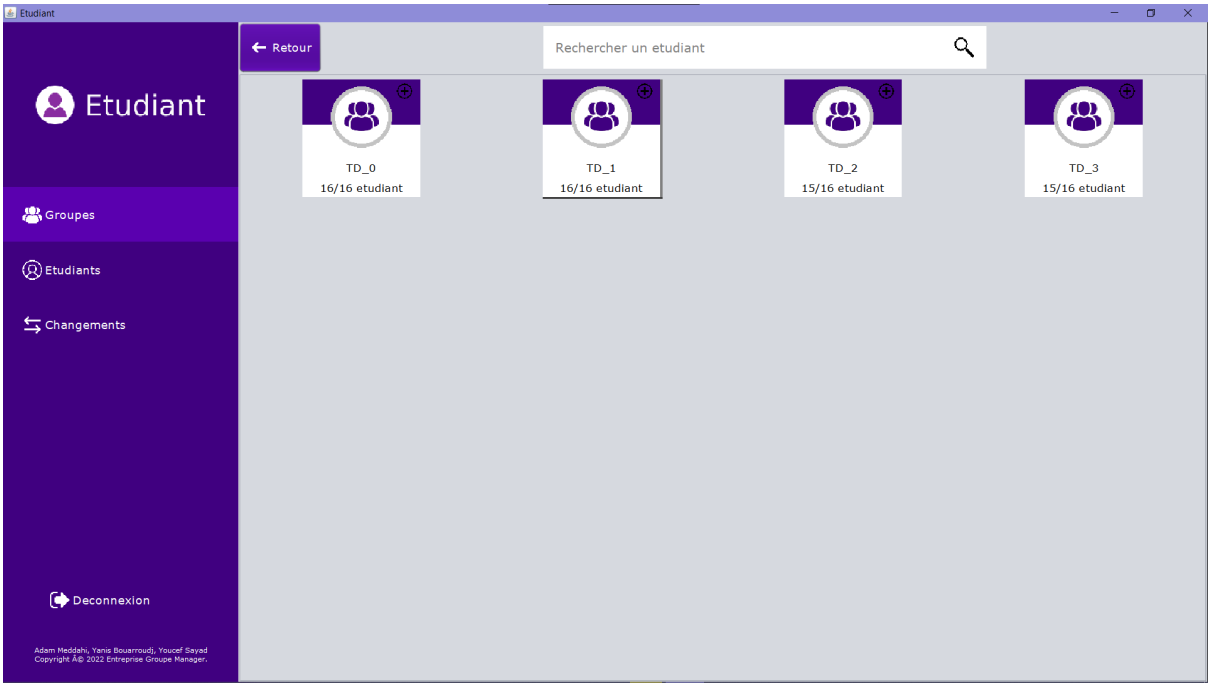


Enseignant :



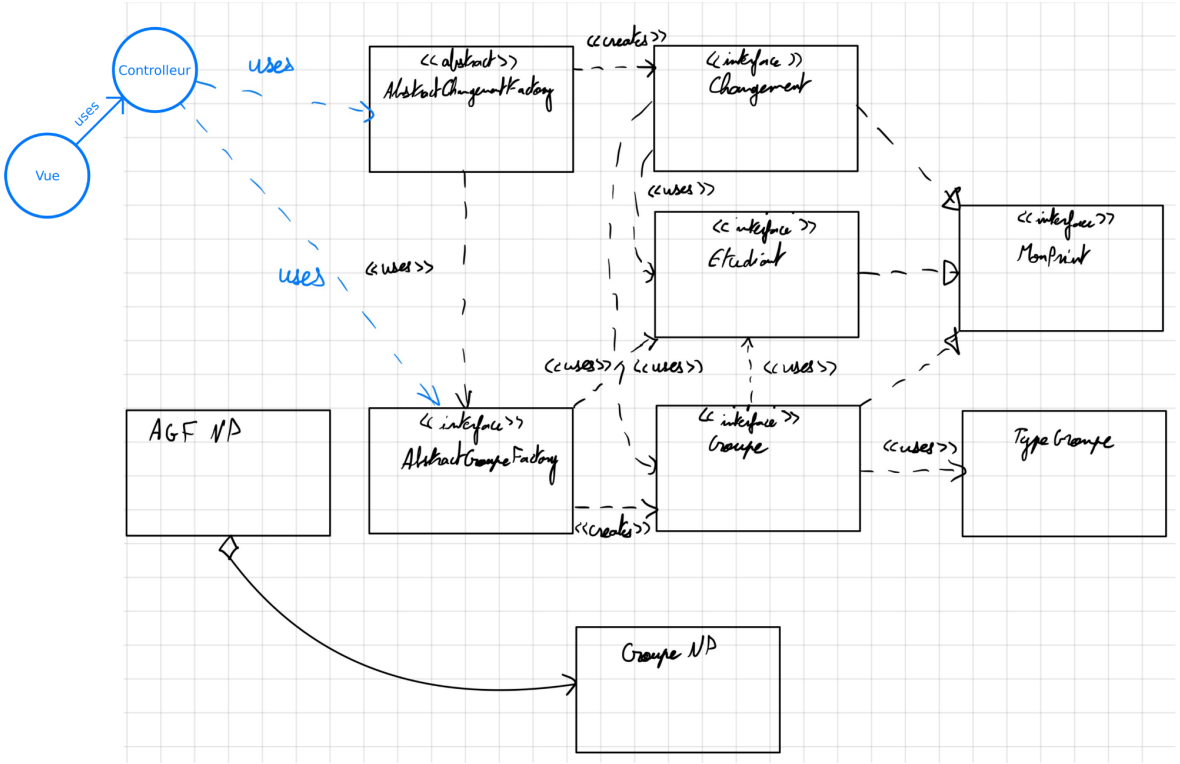

Enseignant
Enseignant
Rechercher un etudiant
Groupes
Etudiants
Changements
Deconnexion
Adam Meddahi, Yanis Bouarroudj, Youcef Sayad
Copyright © 2022 Entreprise Groupe Manager.
césar lycurgus
denis uranus
marcel castor
marin eurydice
constantin akoni
alexandre apikalia
andré ekewaka
renard elikapeka
achille haukea
agathe iakopa
michel caligula
marthe alaric
juliane hannibal
anne juvenal
sophie bede
diane ladislas
christine mahatma
youcef sayad
francine napoleon
adam meddahi

Etudiant :


Etudiant
Etudiant
Retour
Rechercher un etudiant
Groupes
Etudiants
Changements
Deconnexion
TD_0
16/16 etudiant
TD_1
16/16 etudiant
TD_2
15/16 etudiant
TD_3
15/16 etudiant

2. Diagrammes

API: (Image disponible sur le dépôt Git)



Nous avons fait en sorte de ne pas appeler les implémentations, mais toujours les interfaces afin de s'assurer que nous utilisions correctement l'API et que notre code soit compatible avec les implémentations d'autres groupes. Nous nous sommes aussi assuré que notre code soit correctement agencé en respectant le MVC. Notre vue n'est pas directement liée à notre modèle (notion de couplage faible) mais communique avec un contrôleur qui sert d'intermédiaire.

3. Réalisations et tests

Nous avons réalisé le logiciel de l'administrateur, de l'enseignant et de l'étudiant. Nous avons donc priorisé les tâches à réaliser grâce à la notation Moscow. Nous avons réalisé :

Administrateur:

- M - Créer, supprimer et renommer un groupe
- M - Ajouter un individu dans un groupe
- M - Déplacer un individu en validant sa demande de type 1
- M - Refus d'une demande de type 1
- S - Fabriquer automatiquement une partition des étudiants en plusieurs groupes
- C : L'arborescence des groupes
-

Enseignant:

- M - Afficher la liste des groupes
- M - Afficher la liste des étudiants d'un groupe donné
- S - Chercher le groupe d'un étudiant à partir des premières lettres de son nom

Etudiant:

- M - Afficher la liste des groupes
- M - Afficher la liste des étudiants d'un groupe donné
- M - Demander à passer dans un groupe qui est moin plein que le sien en ajoutant une explication
- S - Demander à passer dans un groupe qui est de même taille ou plus grand en ajoutant une explication

Suite à cela nous avons réalisé des wireframes afin d'avoir une idée de l'interface de nos logiciels. L'API permettant de définir les interactions et donc de ne pas être dépendant des implémentations, le groupe s'est scindé afin de réaliser d'une part le Modèle (l'implémentation persistante de l'API) et de l'autre la Vue/Controller. Toutefois, l'équipe s'est concertée tout au long du projet et a pu s'entraider. La vue est fonctionnelle dans le sens où tout fonctionne correctement avec un modèle non-persistant, la liaison avec le modèle persistant a été problématique à cause d'un problème explicité plus bas.

Modèle:

Le modèle persistant est fonctionnel pour tout ce qui est relatif aux groupes. Notre réalisation de la persistance des données repose sur l'utilisation d'une base de données. Cette partie du modèle persistant permet donc :

- La création de la promotion dans le cas où la promo n'est pas encore créée
- La récupération de la promotion dans le cas où cette dernière est sur la BD (via récursivité)
- La création de sous-groupes
- La création de partitions
- La suppression de sous-groupes
- L'ajout d'un étudiant à un groupe
- La suppression d'un étudiant à un groupe
- La recherche d'un étudiant à partir de son nom exact
- La recherche d'étudiants avec un nom ayant pour début la chaîne de caractères recherchée
- La recherche des groupes possédant un étudiant

Pour ce qui est relatif au changement, nous n'avons pas pu les utiliser dans notre code. Notre implémentation de la factory relatives aux changements contient un constructeur qui récupère tous les groupes sur la BD et les ajoute à son this.brain. L'idée est la suivant :

- Récupérer tous les changements sur la base de données
- Récupérer l'objet Etudiant correspondant à l'étudiant du changement
- Récupérer les objets Groupe correspondant aux groupes de départ et d'arrivé
- Créer un objet Changement et l'ajouter au brain de la factory des changements

Nous avons rencontré un problème pour réaliser la 3eme étape. En effet, il est nécessaire de récupérer l'objet Groupe correspondant qui sont stocké dans AbstractGroupeFactoryP.brain mais aucune méthode dans l'interface AbstractGroupeFactory ne permettait de faire cela. Nous avons donc décidé de ne pas modifier l'API afin d'assurer la compatibilité de notre code mais cela c'est donc fait au détriment de l'intégration des changements à notre logiciel. Nous avons donc commenté tout le code dans la factory des changements afin de pouvoir compiler.

Tests Modele (groupes):

- Lancer le logiciel administrateur
- Aller dans l'onglet Groupe
- On peut voir l'ensemble des groupes
- cliquer sur un groupe

- On peut voir ses sous-groupes
- Cliquer sur le rouage d’un groupe
- On peut ajouter des étudiants au groupe
- Créer un groupe
- Créer une partition
- On quitte l’application
- On la relance
- On peut observer que les informations des groupes et que le groupe et la partition dernièrement créée sont présents

Conclusions

Adam :
J’ai grandement apprécié réaliser ce projet. J’ai pu approfondir mes connaissances en programmation orientée objets. Je me suis familiarisé avec une multitude de concepts tels que les Singleton, le MVC, les dictionnaires, les set, les interfaces iterator et iterable, la généricité et les arbres. J’ai aussi pu m’exercer à communiquer avec mes camarades afin de réaliser les choix les plus judicieux pour le bien de notre projet.

Yanis :
J'ai trouvé cette SAE très enrichissante car je n'ai jamais eu l'occasion de créer une application fonctionnelle, on a toujours eu pour habitude de faire de petits jeux et cette fois la difficulté a été plus élevée car il fallait utiliser un API et donc de faire attention à toutes les subtilités. Je remercie également les membres de mon groupe qui ont joué le jeu pour créer l'application la plus intuitive possible, je sais qu'il est toujours possible de faire mieux mais je suis déjà très ravi du résultat.

Youcef :
Cette SAÉ a été pour moi très challengeante mais le résultat à su récompenser les efforts fournis. C’est le premier vrai projet que j’ai dû réaliser en Java tout en essayant de suivre une méthodologie type MVC avec l’implémentation d’une base de données. Je suis assez content du résultat même s'il y a sans doute des choses à améliorer, nous avons bien travaillé et réparti les tâches au sein du groupe. J’ai pu rattraper mes lacunes en SQL ainsi qu’en Java et avec l’implémentation graphique Swing.