

ÉCOLE NATIONALE DE LA STATISTIQUE
ET DE L'ANALYSE DE L'INFORMATION



RAPPORT D'ANALYSE

Jeu **Pokémon**

Projet informatique



Étudiant :

HAMZA EL YOUNI

LUCIE MARTIN

SHIRAZ ADAMALY

MOUSSA KAFANDO

WISSEM ABDUL-AKHIR BABA MOUSSA

Encadrant :

Sylvain LEVEUGLE

Table des matières

Introduction	3
1 Fonctionnement du jeu Pokémon	4
1.1 Diagramme des cas d'utilisation	4
1.2 Description des menus et des fonctionnalités	5
1.2.1 Démarrage du jeu	5
1.2.2 Menu principal	7
1.2.3 Menu Combat ou interface Combat	10
1.2.4 Fonctionnalités avancées	10
1.3 Diagramme d'activité	11
2 Architecture du programme	14
2.1 Diagramme de classes	14
2.2 Gestion des fichiers	16
2.3 Diagramme de séquence	16
3 Accès et persistance des données	18
3.1 Fonctionnement et utilisation de l'API	18
3.2 Modèles logique et physique de données	19
4 Organisation du projet	21
4.1 Organisation prévisionnelle	21
4.2 Retour d'expérience	23
Conclusion	24
Annexes	25
A Lien vers l'API	25
B Notes individuelles	25
B.1 Shiraz Adamaly	25
B.2 Wissem Baba-Moussa	27
B.3 Hamza El Youmni	29
B.4 Moussa Kafando	31
B.5 Lucie Martin	33

Table des figures

1.1	Diagramme de cas d'utilisation	5
1.2	Exemple d'inscription au jeu	6
1.3	Exemple d'inventaire	7
1.4	Exemple d'achat de Pokéball	8
1.5	Exemple de combat contre un Pokémon sauvage	9
1.6	Exemple de rencontre de dresseur	10
1.7	Tables des types	11
1.8	Diagramme d'activité de la connexion et inscription	12
1.9	Diagramme d'activité des menus principal et combat	13
2.1	Diagramme de Classes	15
2.2	Diagramme de Séquence sur l'interface de connexion	17
3.1	Modèle logique de données	19
3.2	Modèle physique de données	20
4.1	Diagramme de Gantt	22

Introduction

Pokémon a été créé par le développeur japonais Satoshi Tajiri. Ce nom Pokémon provient de la contraction de Poketto Monsutä, qui signifie Pocket Monsters (en français montre de poche). Il existe plus de 800 espèces de Pokémon qui sont réparties en huit générations.

Chacune de ces créatures (Pokémons) possède un nom, des caractéristiques et des aptitudes particulières liées à des éléments comme le feu, l'eau, l'air, mais aussi l'électricité, la foudre, la force psychologique, les plantes, etc. Dans cet univers, certains humains sont des dresseurs de Pokémon. Pour cela, il faut d'abord capturer les Pokémons à l'aide d'une Pokéball.

Les dresseurs peuvent s'affronter entre eux en organisant des combats de Pokémon. Les combats prennent fin lorsque tout les Pokémons d'un dresseur sont K.O.

Notre projet a consisté à coder ce jeu Pokémon simplifié. Nous avons utilisé l'API PokéAPI afin d'obtenir les informations sur les Pokémons telles que les statistiques et le type. Nous avons utilisé une base de données, celle-ci contiendra l'ensemble des dresseurs que le joueur affrontera, l'ensemble des Pokémons du joueur et enfin le Pokédex.

Le joueur a le choix entre aller attraper ou s'entraîner contre des Pokémons sauvages, affronter des dresseurs, acheter des Pokéballs ou changer son Pokémon actif. De plus nous simplifierons la gestion des statistiques, et les dégâts des attaques, tout en restant le plus proche possible du jeu originel. En plus de cela, nous avons géré les faiblesses des Pokémons dans notre application.

Notre travail s'articulera autour de quatre chapitres :

Chapitre 1. Ce chapitre s'attache au fonctionnement du jeu Pokémon.

Chapitre 2. Ce chapitre abordera l'architecture du programme.

Chapitre 3. Dans ce chapitre, nous allons présenter l'accès et persistance des données.

Chapitre 4. Nous allons présenter l'organisation du projet, c'est à dire l'organisation prévisionnelle et notre retour d'expérience.

Ce travail se termine par une conclusion générale.

Chapitre 1: Fonctionnement du jeu Pokémon

1.1 Diagramme des cas d'utilisation

Dans cette section nous présenterons le diagramme des cas d'utilisation de notre Jeu de "Pokémon". Ce diagramme permet d'illustrer les principales fonctionnalités ou lots d'actions que pourront réaliser les utilisateurs de notre jeu .

Étant donné qu'avant d'accéder au menu principal du jeu, l'utilisateur devra d'abord s'authentifier, nous avons choisi de distinguer deux acteurs principaux dans notre diagramme de cas d'utilisation : un acteur "joueur" et un acteur "joueur connecté".

A l'ouverture du jeu, l'utilisateur qui à ce stade représente un "joueur", aura la possibilité sur l'écran d'accueil de "se connecter", s'il avait déjà un compte, ou de "s'inscrire" afin de créer un nouveau compte. Lorsqu'il choisit de "se connecter", l'utilisateur devra alors "entrer son pseudo" et "entrer son mot de passe" pour pouvoir accéder au menu principal. Par ailleurs, dans le cas où l'utilisateur choisit de "s'inscrire", il devra "entrer son pseudo", "entrer son mot de passe" qu'il devra "confirmer" et devra "choisir un Pokémon de départ" avec lequel il débutera le jeu.

Une fois l'authentification effectuée, l'utilisateur devient un "joueur connecté" et accède donc au menu principal du jeu où il aura la possibilité "d'accéder à ses informations" : en occurrence "voir son argent", "accéder à son Pokédex", "voir ses Pokéballs" et "voir ses Pokémon". Il pourra également dans le menu principal "acheter des Pokéballs", "chasser des Pokémon sauvages" et "affronter des dresseurs" afin de finir le jeu.

La figure ci-dessous présente le diagramme d'utilisation.

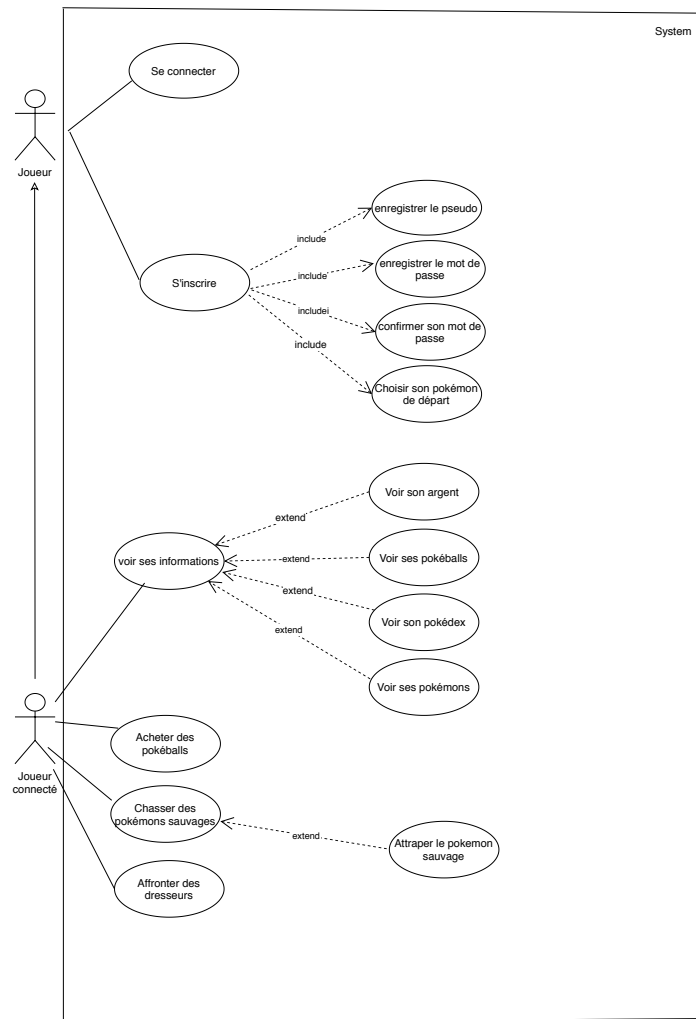


Figure 1.1 – Diagramme de cas d'utilisation

1.2 Description des menus et des fonctionnalités

L'objectif de ce projet est de créer un mini-jeu Pokémon jouable au format texte et permettant principalement de : gérer une équipe de Pokémon, rencontrer ou attraper des Pokémon sauvages ou rencontrer d'autres dresseurs, acheter des Pokéballs. Ainsi, dans cette section, nous présenterons le menu principal ainsi que les principales fonctionnalités de notre jeu.

1.2.1 Démarrage du jeu

Au démarrage du jeu, l'utilisateur ou le joueur verra s'afficher un écran d'accueil qui servira à l'authentification des joueurs. Cet écran d'accueil permettra ainsi aux utilisateurs de "S'inscrire" pour créer un nouveau compte et démarrer une nouvelle partie ou de "Se connecter" pour accéder à un compte déjà existant.

Option "S'inscrire"

Cette option permettra à un joueur de s'inscrire comme nouveau joueur et ainsi créer un nouveau compte.

À l'inscription, le joueur devra renseigner un nom d'utilisateur (pseudo), un mot de passe qu'il devra confirmer afin de valider l'enregistrement dans la base de données. Une fois les informations renseignées, le joueur devra choisir un Pokémon de départ avec lequel il débutera le jeu parmi 3 Pokémon standards qui sont Bulbizarre, Salamèche et Carapuce.

Le nouveau joueur démarrera ainsi avec un pack de départ constitué :

- D'un Pokémon de départ qu'il aura choisi de niveau 1,
- De 5 pokéballs,
- De 60 euros

Une fois l'inscription terminée, le joueur est reconduit à l'écran d'accueil afin de se connecter et de démarrer la partie.



Figure 1.2 – Exemple d’inscription au jeu

Option "Se connecter"

Cette option est réservée aux joueurs possédant déjà un compte de joueur. Elle leur permettra d'accéder à une partie en cours.

Une fois l'option sélectionnée, le joueur devra juste renseigner son nom d'utilisateur (pseudo) et son mot de passe afin d'accéder à sa partie.

- Superball : Ces Pokéballs sont d'un prix et d'une efficacité modérés, de probabilité de capture à 60%
- Hyperball : Ce sont les Pokéballs les plus chers et donc les plus efficaces, de probabilité de capture à 80%

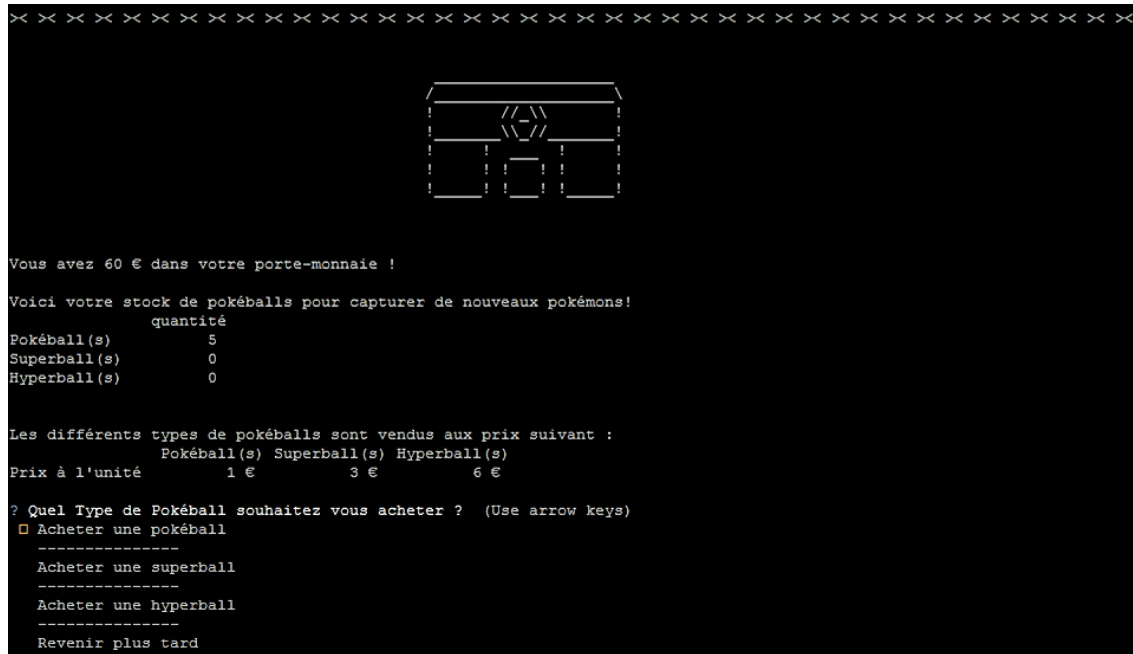


Figure 1.4 – Exemple d’achat de Pokéball

Onglet "Explorer les hautes herbes"

Cet onglet permettra au joueur d'aller à la chasse aux Pokémon sauvages cachés dans les hautes herbes. Le joueur pourra ainsi affronter des Pokémon sauvages qui lui seront présentés de façon aléatoires en fonction du niveau de son Pokémon actif. Un combat dans les hautes herbes présentera trois issues. Le joueur pourra donc soit :

- Mettre le Pokémon sauvage K.O., afin de faire gagner à son Pokémon actif de l'expérience
- Attraper le Pokémon sauvage et ainsi l'intégrer dans son PC. Le joueur pourra donc ensuite l'utiliser afin d'affronter des dresseurs ou chasser dans les hautes herbes
- Fuire le Pokémon sauvage, le gestion de la fuite sera liée au niveau du Pokémon adverse en fonction du niveau de notre Pokémon.

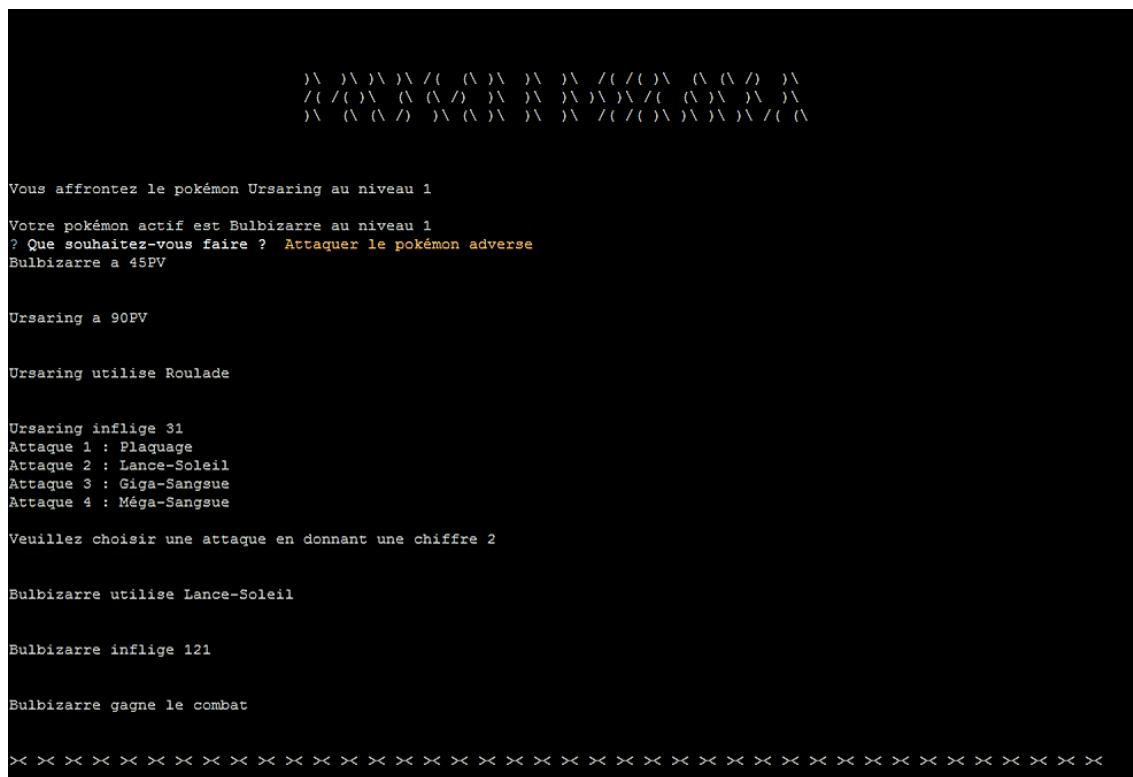


Figure 1.5 – Exemple de combat contre un Pokémon sauvage

Onglet "Affronter des dresseurs"

Comme tout jeu, notre jeu a un but et c'est de réussir à battre un ensemble de dresseurs et leurs Pokémon préalablement définis. Nous avons choisi de créer une liste de neuf dresseurs représentant chacun un niveau d'avancement du jeu. L'objectif du jouer est donc de réussir à battre ces neuf dresseurs. Chaque dresseur possède un Pokémon et le niveau de ces Pokémon varie en fonction de la position du dresseur dans la liste. Le Pokémon du premier dresseur sera de niveau 15 et les niveaux des Pokémon des dresseurs suivant évolueront par palier de 10. Pour débloquent un dresseur il faudra avoir un Pokémon de niveau au moins égal au Pokémon du dresseur. Le dresseur final aura un Pokémon de niveau 100. Une fois le dernier dresseur battu on considère que le joueur a réussi à terminer le jeu. A chaque fois que le joueur gagne contre un dresseur, il reçoit une grosse somme d'argent.



Figure 1.6 – Exemple de rencontre de dresseur

1.2.3 Menu Combat ou interface Combat

Ce menu est celui qui permettra la gestion du déroulement des combats. Il sera sollicité à chaque fois que le joueur affrontera un Pokémon. Un combat de Pokémon est un combat tour par tour. A chaque tour, le joueur aura la possibilité :

- D’attaquer avec son Pokémon, avec une attaque choisie
- De fuir le combat
- D’attraper le Pokémon adverse.

Si le Pokémon du joueur est battu lors d’un combat, un texte affichera que le joueur a perdu et celui-ci retournera au menu initial avec le Pokémon soigné automatiquement.

1.2.4 Fonctionnalités avancées

Afin d’améliorer notre application, nous avons choisi d’implémenter deux fonctionnalités avancées dans le jeu. Nous avons ainsi choisi :

- D’afficher l’image du Pokémon pendant le combat. Néanmoins, par manque de temps, nous n’avons pas réussi à implémenter cette fonctionnalité dans le code.
- De gérer les faiblesses des Pokémon en combat. cette gestion se fera essentiellement à travers l’utilisation du tableau des types. Pour y arriver, nous avons utilisé l’API. Dans la section de chaque type, il y a un onglet appelé "damage relations". Dans cet onglet, les effets des autres types sur la section du type associé sont renseignés. Ainsi un simple appel à l’API permet de connaître les relations entre les types.

Si le Pokémon du joueur gagne contre le pPokémon d’un dresseur, il obtiendra un gain d’expérience, et si celui-ci monte de niveau, un message sera affiché à l’utilisateur, et de plus il gagnera une somme d’argent. De la même manière si le Pokémon du joueur gagne contre un Pokémon sauvage, il obtiendra seulement de l’expérience avec un message si celui-ci monte de niveau.

Lors de la montée de niveau, le Pokémon gagne des statistiques de vitesse, d’attaque, de défense et enfin de pv. Cette augmentation des statistiques est enregistré sur l’objet Pokémon, et sur la base de données.

Le niveau du Pokémon sauvage est égal au niveau du Pokémon de l’utilisateur. Lors des combats, les Pokémon les plus rapides, commencent à attaquer. Les dégâts des attaques des Pokémon sont calculés

en fonction du niveau et des types des Pokémons. Par exemple une attaque feu par un Pokémon de type feu fera plus de dégâts, et puisque nous gérons la gestion des différents types, si le type de l'attaque a un effet sur le type du Pokémon adverse, un message d'efficacité sera affiché avec la quantité de dégâts affichée. Ci-dessous nous avons affiché un tableau des types.

Type de la cible Attaque	ACIER	COMBAT	DRAGON	EAU	ELECTRIQUE	FEE	FEU	GLACE	INSECTE	NORMAL	PLANTE	POISON	PSY	ROCHE	SOL	SPECTRE	TENEbres	VOL
ACIER	x½			x½	x½	x2	x½	x2						x2				
COMBAT	x2					x½		x2	x½	x2		x½	x½	x2			x2	x½
DRAGON	x½		x2			0												
EAU			x½	x½			x2				x½			x2	x2			
ELECTRIQUE			x½	x2	x½						x½				0			x2
FEE	x½	x2	x2				x½					x½					x2	
FEU	x2		x½	x½			x½	x2	x2		x2			x½				
GLACE	x½		x2	x½			x½	x½			x2				x2			x2
INSECTE	x½	x½				x½	x½				x2	x½	x2			x½	x2	x½
NORMAL	x½													x½		0		
PLANTE	x½		x½	x2			x½				x½	x½		x2				x½
POISON	0					x2					x2	x½		x½	x½	x½		
PSY	x½	x2										x2	x½				0	
ROCHE	x½	x½					x2	x2	x2						x½			x2
SOL	x2				x2		x2		x½		x½	x2		x2				0
SPECTRE										0			x2			x2	x½	
TENEbres		x½				x½							x2			x2	x½	
VOL	x½	x2			x½				x2		x2			x½				

Figure 1.7 – Tables des types

1.3 Diagramme d'activité

Ce premier diagramme a pour but de décrire le fonctionnement du menu principal. Tout d'abord, le jeu commence par l'accès à un écran d'accueil. Ensuite, deux choix sont proposés au joueur, de *se connecter* ou de *s'inscrire*.

Si le joueur décide de *s'inscrire*, il doit entrer un pseudo, et le programme vérifie qu'il ne s'agit pas d'un pseudo déjà utilisé pour un autre joueur. Ensuite, il doit entrer un mot de passe, le confirmer, et enfin, il doit choisir un Pokémon avec lequel il va commencer l'aventure. Un message de validation de la création du compte est affiché, puis le joueur est directement envoyé dans la partie *se connecter*.

Dans la partie *se connecter*, le joueur doit entrer son pseudo et son mot de passe. Puis le programme vérifie que le compte appartient à la base de donnée, et si cela n'est pas le cas, le joueur est invité à réécrire son pseudo et son mot de passe. Le joueur a trois essais. Au bout du troisième, si le pseudo ou le mot de passe n'est pas bon, le joueur retourne à l'écran d'accueil. Néanmoins, si le compte appartient à la base de données, la connexion est réussie, et le joueur est envoyé dans le *menu principal* du jeu.

Ce diagramme a pour but de décrire le menu principal, et le menu combat. Tout d'abord il y a trois choix :

- Acheter des Pokéballs et si l'on ne possède plus assez d'argent, le joueur a la possibilité de retourner dans le menu principal.
- Accéder aux infos. Dans cette partie, le joueur peut avoir accès à l'ensemble des Pokémons que le joueur a capturés et de plus de changer le Pokémon actif (celui qui est prêt à combattre), à son

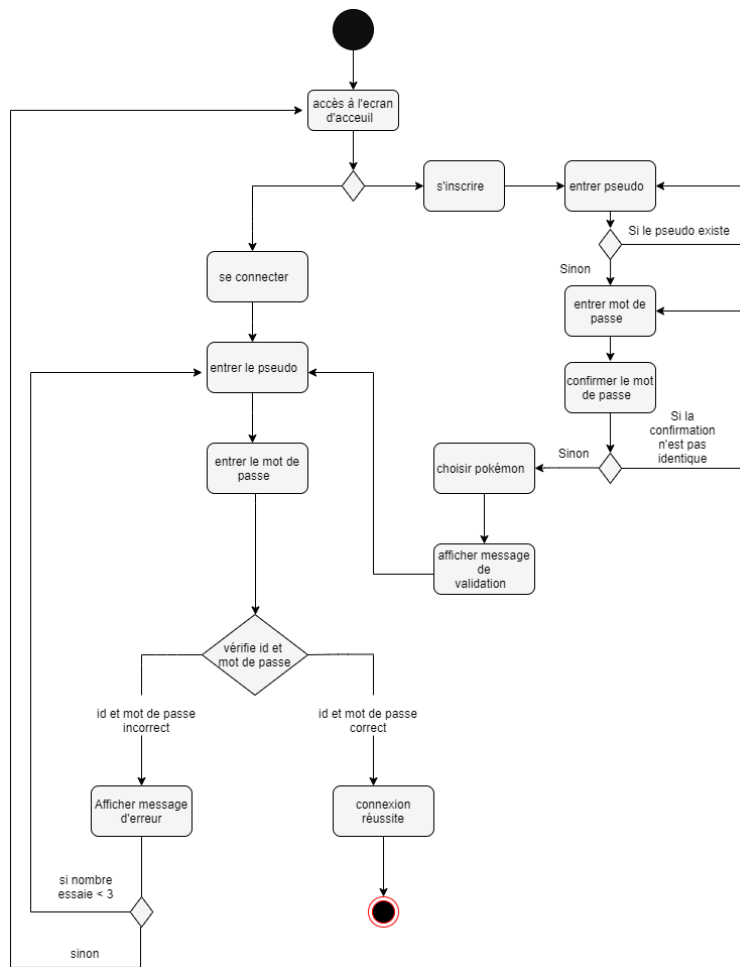


Figure 1.8 – Diagramme d’activité de la connexion et inscription

stock de Pokéballq, et enfin, au Pokédex, et plus particulièrement, le joueur peut avoir accès au information d’un Pokémon à travers le Pokédex.

- Accéder au menu combat. Dans ce menu, le joueur a le choix entre chasser un Pokémon, ou combattre un dresseur. Si le joueur choisi d’affronter un dresseur, il doit mettre le Pokémon adverse K.O. et dans ce cas le joueur gagne, sinon le joueur perd.

Si le jouer choisi de combattre des Pokémon sauvages, il peut fuir, attraper le Pokémon, et tout comme le dresseur mettre K.O. le Pokémon sauvage.

Et enfin, depuis le menu principal, le joueur peut choisir de quitter la partie en cours.

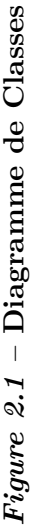
Chapitre 2: Architecture du programme

2.1 Diagramme de classes

Pour décrire et formaliser l'architecture de notre code, nous avons conçu le diagramme de classes suivant. Celui-ci décrit les différents objets qui vont être appelés dans notre programme et les relations entre ceux-ci. On peut y lire les attributs (ce qui les décrit) et les méthodes (ce qu'ils peuvent faire) de ces objets.

Notre code sera subdivisé en couches. En bleu, nous avons la couche contrôleur qui permet l'interaction avec l'utilisateur, en jaune la couche Service qui exécute les actions demandées par l'utilisateur et en gris, la couche Métier avec des objets spécifiques à notre application.

Afin d'alimenter notre application en données, nous avons créé deux couches spécifiques : en orange, une couche DAO et en vert, une couche Webservice. DAO signifie Data Access Object. Les objets DAO servent d'interface entre la base de données que nous avons créée et l'application. La couche Webservice permet de récupérer des données en libre accès sur une API.



2.2 Gestion des fichiers

Pour avoir une architecture de notre application cohérente, nous avons décidé de découper notre programme en plusieurs dossiers. Le fichier main est un exécutable directement au lancement de l'application. Au même niveau, des dossiers correspondront aux différentes couches de notre programme. Ces dossiers seront eux-mêmes décomposés en fichiers, un pour chaque classe du même nom. Nous aurons donc les dossiers :

- Contrôleur : permet l'interaction entre le joueur et l'application qui comprend le fichier abstract-View, accueilView, achatView,..., session.
- Service : partie du code qui manipule les objets métiers pour créer de l'information et lancer les fonctionnalités de notre application. Cette partie est composée des classes suivantes :
 - la classe **Achat** qui permet au joueur d'acheter la quantité de Pokéballs voulue.
 - la classe **Combat** qui permet au joueur de combattre d'autres dresseurs. Ainsi, le joueur peut attaquer d'autres Pokémon, il peut aussi attrapper d'autres Pokémon ou fuir le combat.
 - la classe **Connexion** qui permet au joueur de se connecter à travers son identifiant et son mot de passe. Elle utilise aussi la fonction "recuperer" de JoueurDAO qui à partir du pseudo devrait récupérer toutes les informations sur le joueur.
 - la classe **Inscription** qui permet à un nouveau joueur de s'inscrire avant de se connecter pour commencer le jeu.
 - la classe **Inventaire** qui nous permet d'afficher l'argent disponible du joueur, le nombre total de Pokéballs du joueur et permet aussi de changer de Pokémon actif.
 - la classe **Manager**, comme son nom l'indique permet de manager le jeu (préparer le jeu, l'affichage du message welcome et goodbye)
- Métier : objets du code propres à notre application. Cette partie comprend les classes suivantes :
 - la classe **Attaques**, cette classe gère les attaques des Pokémon en fonction du nom, la description, le type et les dégâts de chaque attaque.
 - la classe **Pokedex**, elle permet d'afficher le Pokédex et les Pokémon du joueur selon leurs caractéristiques.
 - Il faut noter que la partie "Metier" comprend aussi la "classe Pokemon", la classe "Statistique", "Joueur", "Type", et "Dresseur". Elle comprend aussi le test unitaire sur la classe "Dresseur".
- DAO : partie du code qui gère la communication avec notre base de données
- Webservice : partie du code qui gère la communication avec l'API PokéAPI

A ces dossiers, s'ajouteront un dossier Installation qui regroupera les informations et requêtes de création de notre base de données et un dossier Assets avec les fichiers qui serviront à rendre le design des sorties consoles un peu plus ludique.

On retrouve aussi un fichier README à la racine de notre projet qui met en évidence le mode de fonctionnement de notre application.

2.3 Diagramme de séquence

Avant d'entrer au menu du jeu et faire l'ensemble des autres scénarios, l'utilisateur doit créer un compte ou se connecter en utilisant son login et le mot de passe, s'il a déjà un compte.

Ainsi, l'interface d'authentification s'affiche avec deux options : une option pour créer un compte si le joueur n'a pas de compte et une autre option pour se connecter si l'utilisateur a déjà un compte. Concernant la création du compte, un formulaire de saisie des données est ouvert. Le joueur saisit un identifiant et le mot de passe puis valide la saisie.

Un joueur possédant un compte saisit uniquement son login et le mot de passe puis après vérification il est redirigé vers l'écran d'accueil.

Le diagramme qui suit présente l'enchaînement de la phase d'authentification et fait le lien entre le diagramme d'activité et le diagramme de classes.

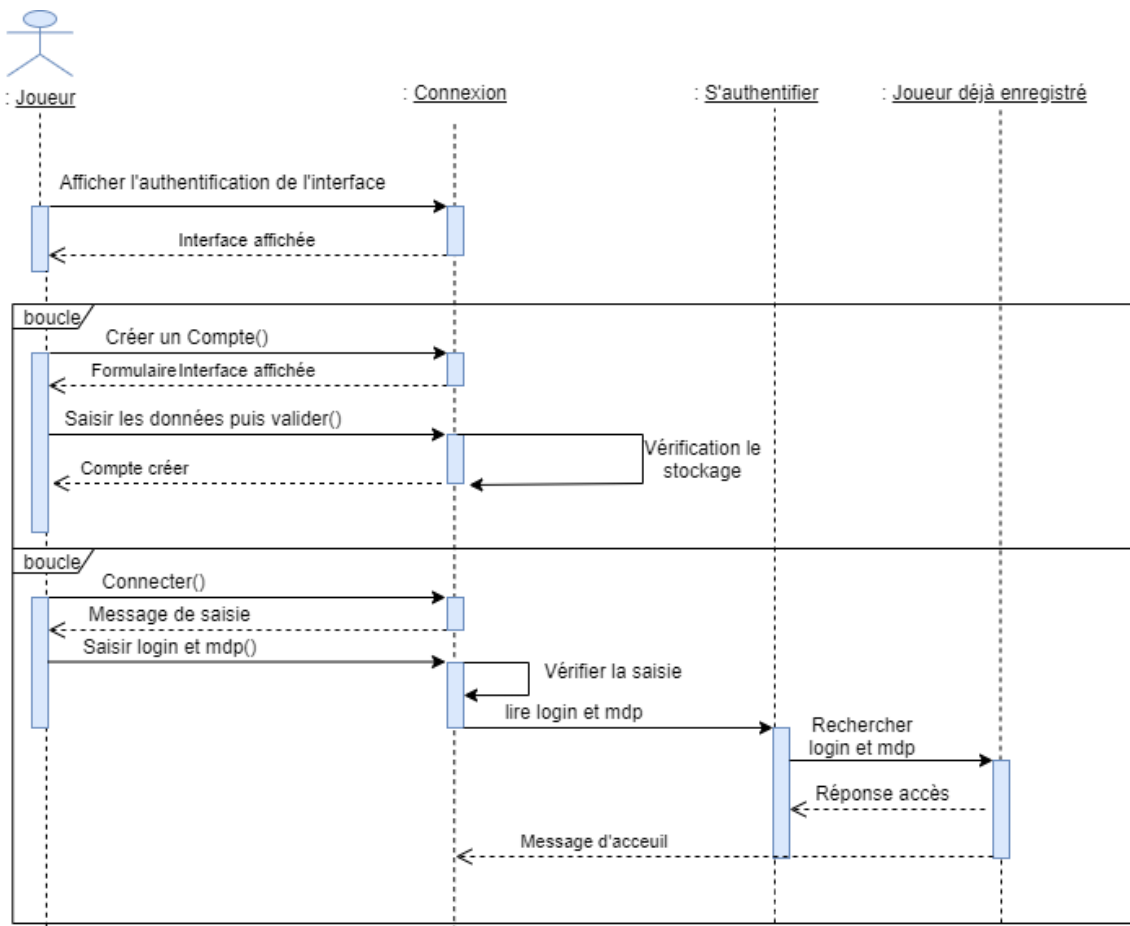


Figure 2.2 – Diagramme de Séquence sur l'interface de connexion

Chapitre 3: Accès et persistance des données

3.1 Fonctionnement et utilisation de l'API

L'API que nous allons utiliser dans ce projet est l'API PokéAPI fournie par la communauté Pokémon et The Pokemon Company . Le type de données fourni par l'API est un Json. Les Pokémon sont classés par un numéro, par exemple Pikachu est le Pokémon numéro 25, ou Mr-mime est le pokemon numéro 122.

Dans l'API, chaque Pokémon possède :

- *base experience* :

Nous ne savons pas exactement ce que veut dire "base" experience". Néanmoins pour la partie code, nous l'avons utilisé comme le premier terme d'une suite pour modéliser l'expérience d'un Pokémon. Ainsi, pour passer du niveau k à $k+1$, il faut obtenir $\text{base experience} + k$ points d'expérience. De plus, "base experience" est la quantité d'expérience obtenu si le Pokémon est battu

- *id* :

Il s'agit de l'identifiant du pokemon, comme nous l'avons expliqué au début, l'identifiant de Pikachu est 25 par exemple. Cette id sert surtout pour générer un Pokémon aléatoire. En effet, nous avons géré uniquement les 3 premières générations de Pokémon qui correspondent au 365 premiers Pokémon. Ainsi pour générer un Pokémon aléatoire, il suffit simplement de générer un nombre entre 1 et 365, et ainsi on obtient l'id du Pokémon.

- *move* :

Il s'agit de l'ensemble des attaques que le Pokémon peut apprendre, chaque attaque dans l'API possède un nom (en différentes langues), une précision (en nombre entier), un type, pp (en nombre entier), et une puissance.

- *nom* :

Il s'agit du nom du Pokémon, ici en anglais. Pour obtenir les noms en français des Pokémon. Il faut accéder à la partie "pokémon-spécies", et non à l'onglet "pokémon"

- *sprite* :

Il s'agit de l'image du Pokémon. Comme nous l'avons dit précédemment, nous avons prévu d'implémenter l'image du Pokémon pendant les combats. Celui-ci représente un lien, qui charge l'image du Pokémon sous différents angles.

- *stats* :

Il s'agit de l'ensemble des statistiques du Pokémon. Une statistique contient un nombre entier, qui correspond à la puissance de la statistique, et un nom. Par exemple (45, Vitesse).

- *types* :

Il s'agit du type du Pokémon, un Pokémon peut avoir jusqu'à deux types, et chaque type contient simplement un nom dans l'API.

Nous n'avons pas décrit exhaustivement les paramètres du Pokémon dans l'API. Nous avons essayé de décrire les principaux, et ceux que nous avons dans le code du jeu. Pour la partie gestion des types, nous avons utilisé l'API. En effet dans la section de chaque type, il y a une section appelé "damage relation". Cette section fournit les types qui sont plus ou moins efficace. Ainsi la gestion des types a été fortement facilitée grâce à l'API.

3.2 Modèles logique et physique de données

Afin de garantir la persistance de nos données, nous avons considéré les tables suivantes pour construire notre système de gestion de base de données :

- Pokemon : Nous allons stocker toutes les informations recueillies sur les Pokémon dans cette table.
- Type : Dans cette table nous stockons les types des différents Pokémon.
- Dresseur : Cette table contient les données des dresseurs que le joueur va affronter.
- Attaque : Il s'agit des attaques des Pokémon.
- Joueur : Cette table stocke l'ensemble des joueurs de notre application.
- Statistique : Cette table contient les statistiques d'un Pokémon fournies par l'API

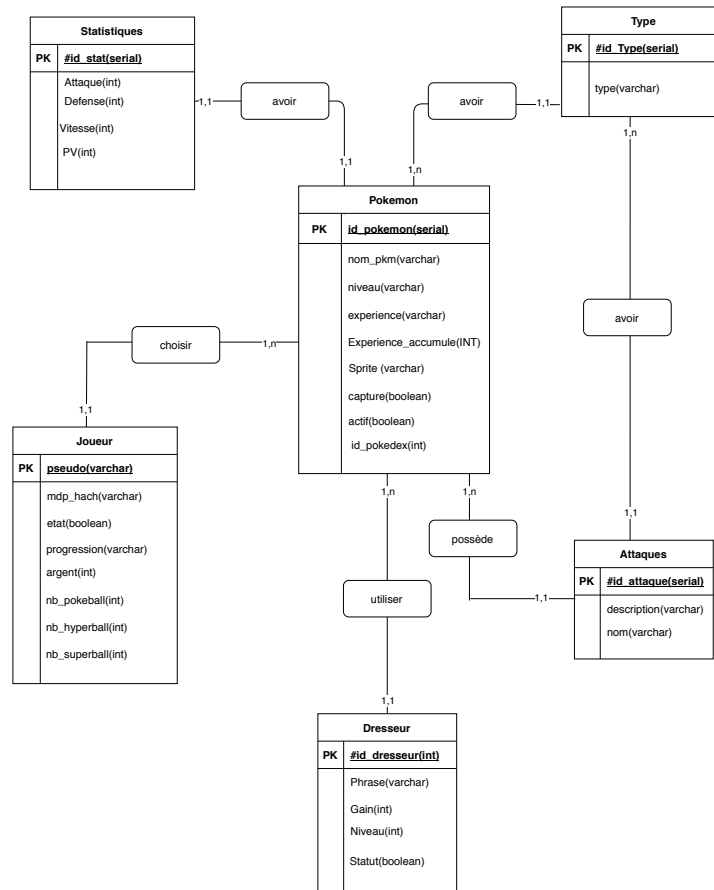


Figure 3.1 – Modèle logique de données

Par conséquent, il en découle le modèle physique suivant :

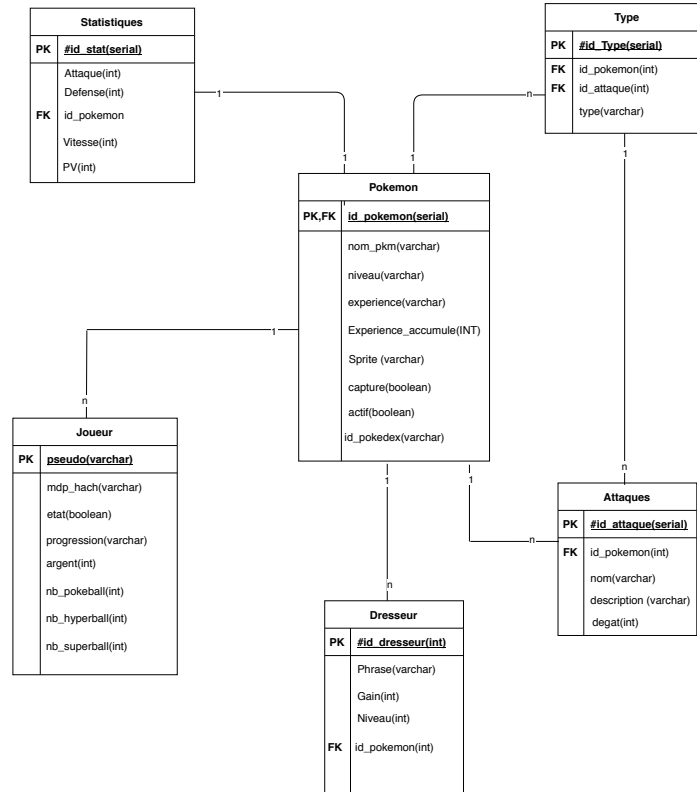


Figure 3.2 – Modèle physique de données

Chapitre 4: Organisation du projet

4.1 Organisation prévisionnelle

Pour mener à bien ce projet informatique, nous avons décidé de nous organiser de la manière suivante.

La communication interne se fait à partir de l'application Microsoft Teams. Celle-ci nous permet de nous transmettre des documents pdf et des images mais aussi de réaliser des réunions en distanciel. La rédaction du rapport et des différents rendus est fait sur Latex afin de travailler avec un Wiki. C'est à dire être à plusieurs sur un même document. Pour la partie programmation, nous utiliserons Gitlab, cette plateforme permet la rédaction du code de notre application avec un principe de versionnage. Il sera possible de garder plusieurs versions de ce que nous aurons développé et d'éviter les conflits.

Afin de gérer au mieux les échéances, nous nous baserons sur le diagramme de Gantt suivant. Celui-ci représente en ligne temporelle, le volume horaire que nous pensons nécessaire à la réalisation des différentes tâches qui composent ce projet.

Pour finir, nous avons défini un chef de projet qui se chargera de l'organisation et de la centralisation du projet ainsi que de la communication avec le tuteur. Il s'agit de Lucie.

Pour la phase de conception, nous nous sommes répartis le travail de la façon suivante :

- Shiraz : analyse de l'API, détails des menus et fonctionnalités, réalisation des diagramme d'activité et du modèle logique de données.
- Wissem : détails des menus et fonctionnalités, réalisation des diagramme des cas d'utilisation et d'activité.
- Moussa : réalisation du diagramme de séquence et de la premier version du diagramme de Gantt.
- Hamza : réalisation de la première version du diagramme de Gantt, et des modèles physique et logique de données.
- Lucie : réalisation du diagramme de classes et de Gantt final, gestion du projet.

Pour la suite du projet, nous nous sommes répartis les différentes taches.

- Shiraz : construction de la base de données, développement des classes spécifiques aux Pokémons et des actions de jeux et de combats.
- Wissem : construction de la base de données, des classes spécifiques aux joueurs et dresseurs et des classes de connexions et inscriptions.
- Moussa : développement de la classe manager, des classes spécifiques aux Pokémons et des actions de jeux et de combats.
- Hamza : développement de la classe manager, des classes spécifiques aux joueurs et dresseurs et les achat et l'accès à l'inventaire.
- Lucie : développement de la couche contrôleur, des classes de DAO et de Webservice ainsi que les classes de connexions et inscriptions.

La rédaction du rapport final et la préparation de la soutenance ce fera en commun.

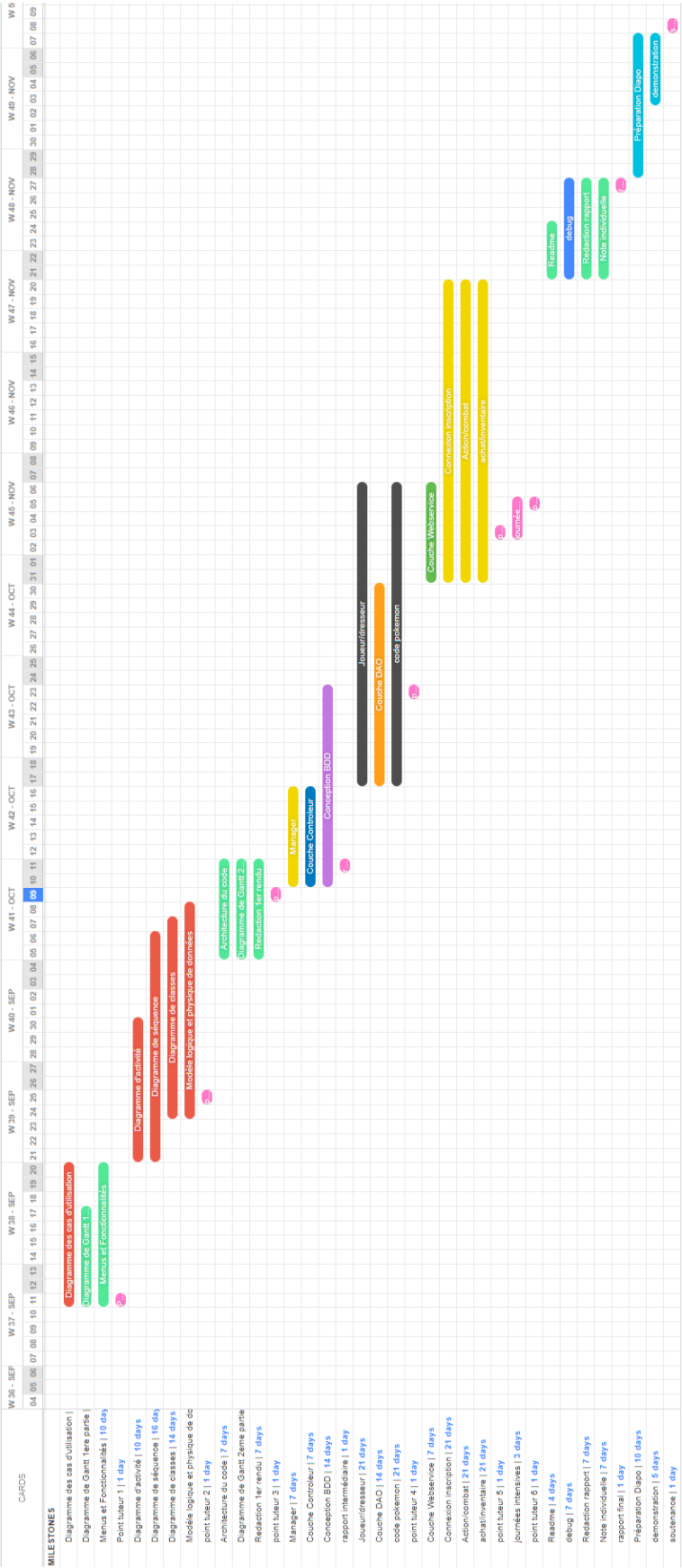


Figure 4.1 – Diagramme de Gantt

4.2 Retour d'expérience

Lors de la phase de développement, nous avons, tout d'abord, commencé à suivre notre diagramme de Gantt ci-dessus. Malheureusement, nous avons très vite pris un peu de retard sur l'organisation prévue car certains d'entre nous ont rencontré des problèmes techniques avec la prise en main de Pycharm, de notre IDE, ou de git tels que des problèmes de push des fichiers liés au choix de l'interpréteur Python.

Par ailleurs, nous nous sommes rendus compte que certains comprenaient un peu plus finement que d'autres certaines subtilités du code. Ceci a conduit à une réaffectation des tâches. La répartition effective du travail s'est donc faite de la façon suivante :

- Shiraz : construction de la base de données, développement des classes spécifiques aux Pokémon et des actions de jeux et de combats. De plus, Shiraz s'est occupé d'une partie de la DAO, de la gestion complète de la couche de Webservice et d'une partie du débogage.
- Wissem : construction de la base de données, des classes spécifiques aux joueurs et dresseurs et des classes de connexions et inscriptions. De plus, Wissem s'est chargé d'une partie du débogage et enfin de commenter la classe dresseur dans le cadre de l'évaluation.
- Moussa : développement des classes Pokédex, Achat et Inventaire du jeu. De plus Moussa a développé, dans le cadre de l'évaluation, la classe de test unitaires associée à la classe Dresseur.
- Hamza : développement des classes spécifiques aux joueurs et dresseurs et les achats et l'accès à l'inventaire. De plus, Hamza a développé quelques fonctions dans les classes de Combat et d'Inscription.
- Lucie : développement de la classe Manager, la couche contrôleur, des classes de DAO ainsi que les classes de connexions et inscriptions. De plus, Lucie s'est chargée avec Wissem et Shiraz de la phase de débogage de l'application et de la création d'un dossier pour l'installation, accompagné d'un fichier README.

Ces changements d'organisation nous ont permis de rattraper le retard du début de projet. Cependant, de nouveaux contre-temps sont apparus dans les dernières, notamment des soucis d'accès à la base de données SQL en ligne de l'école, des ralentissements de nos Virtual Machines ou encore Pycharm qui ne s'ouvrait plus. L'une des solutions techniques pour résoudre ces problèmes a été d'installer en local une base de données SQL sur nos propres machines. Ces retards, nous ont conduit à revoir le choix de notre fonctionnalité avancée comme il a été décrit plus haut, dans la partie de description des menus et fonctionnalité de notre application Pokémon.

De plus, l'évolution de la crise sanitaire actuelle n'a pas facilité notre travail. Cela a limité les interactions puisque nous avons été un mois en semi-distanciel puis au retour des vacances de la Toussaint, nous avons basculé en 100% distanciel. De ce fait, nous nous sommes beaucoup reposés sur les outils tels que Microsoft Teams et Gitlab. Le fait de ne pas être en présentiel a rendu plus difficile la communication interne de notre groupe mais aussi a réduit les interactions avec les professeurs lors de séances de suivi.

Nous sommes, malgré tout, heureux de pouvoir présenter une version fonctionnelle de notre application et sommes fiers du travail accompli.

Conclusion

En conclusion, les différents diagrammes nous ont permis d'avoir une vision plus réaliste de notre application. Grâce au diagramme de Classe nous avons défini l'ensemble des classes que nous avons utilisé. Le Modèle physique et logique de données nous a permis de définir clairement notre base de donnée. Tandis que les diagramme d'activités et de cas d'utilisation nous a permis de définir clairement les fonctionnalités de notre jeu.

Construire une application du cahier des charges à la programmation en plus du rapport et en un laps de temps très réduit a été un défi à relever. Nos connaissances en la matière étaient très récentes et fragmentaires sur bien des aspects. La programmation objet, et tous les avantages qui s'y rattachent, requiert une organisation importante en amont ; et donc du temps, ressource très précieuse dans tout projet et qui manque forcément à la fin. Avec la crise sanitaire, l'évolution des cours et la mauvaise connexion, nous avons pu ressortir le meilleur de nous même. Nous avons su bien gérer l'organisation du projet, la répartition des tâches ainsi que l'implication de tout les éléments du groupe. Nous avons eu pas mal de réunion dans la matinée tout comme dans la soirée.

De la conception à la programmation, nous avons souhaité suivre le plus possible le paradigme Orienté Objet. C'est pourquoi, le temps de conception a été assez long. La quasi totalité de ce que nous avons imaginé au début est présente dans notre application bien que des pistes d'amélioration existent.



Annexes

A Lien vers l'API

Le lien suivant conduit vers l'API PokéAPI qui nous a servi à la réalisation de ce projet : <https://pokeapi.co/>

B Notes individuelles

B.1 Shiraz Adamaly

Ce projet informatique est le premier projet informatique que je considère concluant. Nous avons été réparti par groupes de 5 personnes. Néanmoins, j'ai eu un peu d'expérience dans le domaine informatique par des petits projets personnels, et d'un certain goût pour l'informatique. Lors de l'attribution des groupes, je vous avoue avoir eu de l'appréhension, en effet, je ne connaissais pas très bien les camarades de mon groupe. Ainsi, j'ai eu de l'appréhension concernant l'implication de certains mes camarades pour le projet informatique.

Pour ce projet, sur la partie UML, je me suis occupé d'une partie des diagrammes d'activité, avec Wissem. De plus, encore avec Wissem, on s'est occupé de décrire l'ensemble des fonctionnalités de notre application, et des différents menus. De plus lors des différentes réunions hebdomadaire j'ai dû expliquer plusieurs fois les spécificités du jeu pokémon. En effet, du groupe, je pense être celui qui connaît le mieux le jeu pokémon. C'est pourquoi je me suis occupé de la description de notre application et des différents menus avec Wissem. Car cela permet de bien comprendre les enjeux et les limites de notre application, la gestion des statistiques, la gestion de l'expérience, les différentes pokeballs, les différentes arènes, etc... Pour la partie code, je me suis occupé de la partie Webservice, tout ce qui concerne l'API, récupérer les données des pokemons, la gestion des types avec l'API. Ensuite je me suis occupé de quelques classes de la couche DAO. Et enfin, je me suis occupé de la classe Pokemon, les statistiques, les types des différents pokemons, pour ensuite m'occuper de la partie combat, qui fut pour moi la plus difficile en terme de temps. En effet les combats pokemon font intervenir l'ensemble des couches de l'application. La fuite de pokemon qui fait intervenir les différentes View, la couche Webservice lors de la création d'un pokemon sauvages, avec les différentes attaques et les classes DAO, lors de l'enregistrement des différentes augmentation de statistique lors des montées de niveau des pokemon. Je tiens à préciser que j'ai été aidé par Lucie sur cette partie. Et enfin sur la fin du projet, je me suis occupé avec Wissem et Lucie de la partie debuggage de l'application.

Concernant les différentes difficultés du projet. Tout d'abord, n'ayant pas été assidu au cours d'UML de l'année dernière, j'ai dû réapprendre les différents diagrammes UML pour comprendre ce qu'on fait mes camarades. Pour la partie code, j'ai dû comprendre comment utiliser l'API, savoir ce que c'est, les cours en autonomie de l'UE Complément informatique fut d'une grande aide. Ensuite, j'ai eu quelques problèmes d'importation de module sur pycharm, ce qui m'a ralenti sur le milieu de projet. En effet je ne codais pas sur la VM, mais le pycharm de mon ordinateur, ainsi, j'ai dû installer un interpréteur pour remédier à cela. Enfin pour la gestion des pokemon, j'ai eu de multiples problèmes, en effet, j'ai dû rajouter des attributs dans la classe pokemon pour établir une gestion de l'expérience similaire au jeu pokemon. Or je n'avais pas réalisé qu'il fallait que je fasse de même dans la base de donnée et ainsi j'ai

dû changer une grande partie du programme. De plus, je ne codais pas sur la VM, j'ai dû installer une base de donnée postgres depuis mon ordinateur en local. Enfin pour la partie combat, avec lequel j'ai eu le plus de difficulté, j'ai du apprendre à repérer une erreur rapidement, et à cibler le problème à l'aide de "print" bien placé. J'ai eu plusieurs problèmes, le niveau qui ne changeait pas après un combat, les points de vie qui ne s'enregistraient pas à la fin d'un tour de combat, et j'en passe...

Pour conclure, j'ai trouvé ce projet très formateur, mais néanmoins très frustrant, parce que je considère que le code que j'ai produit peut être amélioré sur plusieurs points, mais par manque de temps j'ai préféré éviter de changer. Avec notre groupe nous avons fait une réunion hebdomadaire qui je trouve était une excellente idée et que je referai. Néanmoins, j'ai réellement apprécié ce projet informatique, et je pense que l'informatique sera certainement essentiel lors de mon futur travail. Je tiens à remercier également notre tuteur Sylvain Leveugle qui fut d'une grande aide pendant tout le long du projet.

B.2 Wissem Baba-Moussa

Appréhension et organisation du projet

Ce projet informatique a démarré dès ma deuxième semaine à l'ENSAI et est le tout premier de cette envergure sur lequel je travaille. Étant donné que nous n'étions pas avancés dans le cours de remise à niveau et que les notions nécessaires au projet m'étaient pour la plupart étrangères j'appréhendais vraiment ce projet et pensais ne pas pouvoir être d'une grande utilité dans le groupe. L'appréhension était double du fait que étant admis sur titres je ne connaissais pas du tout les autres membres du groupe.

Mais après présentation et appropriation du sujet, j'ai été intéressé par la thématique et j'ai eu la chance de me retrouver dans un groupe de personnes motivées avec une chef d'équipe ayant un très bon sens de l'organisation et ne laissant aucun membre à la marge.

Dès l'entame du projet nous avons défini une stratégie de travail et nous avons des réunions hebdomadaires afin de faire un point sur l'avancée, de pouvoir se répartir les tâches et de pouvoir discuter sur les zones ombres. Nous avons d'ailleurs eu plusieurs rencontres lors de la partie conception afin de nous entendre sur les bases du projet et les classes à utiliser.

À la suite du confinement les réunions se sont poursuivies via Teams, les tâches étaient réparties entre membre et chaque semaine des objectifs et des attendus étaient fixés. Chacun des membres a pu travailler sur les parties sur lesquelles il se sentait un peu plus à l'aise et le travail a vraiment été participatif et collaboratif.

Mes réalisations

La partie de conception a été très importante et intéressante car elle m'a permis de m'approprier le sujet et a permis de visualiser plus facilement les attendus et le squelette du code que nous devons réaliser. Dans cette partie j'ai essentiellement travaillé avec Shiraz sur la rédaction de la description de l'application, du menu et des fonctionnalités et la construction des diagrammes. Grâce à des cours suivis sur la plateforme Open Classroom j'ai pu réaliser le diagramme des cas d'utilisation et le diagramme d'activité de l'écran d'accueil.

Dans la réalisation du code j'ai essentiellement travaillé sur les classes métiers en rapport avec les dresseurs (AbstractDresseur, Dresseur et Joueur) et sur les classes de connexion et d'inscription où j'ai pu découvrir les procédures de hachage pour la sécurité informatique. J'ai ensuite travaillé sur la liaison des classes d'inscription et de connexion avec les views associées avec Lucie mais également sur le débogage du code avec Lucie et Shiraz. Cette étape de débogage m'a permis de parcourir l'ensemble des couches de l'application (DAO, Service, Métier, View, API...). Pour finir j'ai fait les commentaires des classes AbstractDresseur, Joueur et Dresseur et corrigé les diagrammes d'activité.

Difficultés rencontrées

Comme tout travail humain, ce projet n'a pas été exempt de difficultés. Au début du projet, j'avais principalement du mal à suivre les TP qui devaient nous préparer à la réalisation du projet et me sentais complètement largué. Mais avec les recherches et l'aide de notre enseignant Rémi Pépin j'ai su prendre en main les différents outils.

Par la suite, les difficultés étaient principalement liées à l'utilisation de Git (la gestion des conflits) et surtout au débogage du code. Cette partie de débogage a d'ailleurs nécessité plusieurs fois l'intervention de notre tuteur Sylvain LEVEUGLE et de notre enseignant Rémi Pépin. Par ailleurs les révisions pour les examens, le suivi des cours et des TP, et le confinement ont à un moment donné ralenti l'avancée du projet et il était assez difficile de cumuler toutes ces tâches et de pouvoir faire un bon arbitrage entre elles et le projet. D'autres difficultés afférentes à l'utilisation du cluster de l'école et de pycharm sur la

machine virtuelle ont également pu se manifester dans la réalisation du projet.

Mais malgré ses diverses difficultés nous avons pu mener à bien la majorité des objectifs que nous nous étions fixés en début de projet.

Apport du projet

Ce projet a vraiment été bénéfique et m'a permis d'acquérir de nombreuses connaissances. Entre autres, il m'a permis de découvrir l'élaboration d'un projet informatique de sa conception jusqu'à sa réalisation. J'ai de ce fait pu découvrir la modélisation UML qui est un peu différente de la modélisation MERISE qui était celle que je connaissais, approfondir mes connaissances en SQL et découvrir le découpage en couche qui facilite la répartition des tâches et la résolution des problèmes dans le code.

Par ailleurs n'ayant jamais fait de programmation orienté objet auparavant ce projet m'a permis de m'y initier et d'approfondir ma connaissance du langage python qui est un langage très utilisé dans le monde de la Data Science et dont je n'avais qu'une connaissance sommaire.

Enfin grâce à ce projet j'ai pu expérimenter divers outils de travail collaboratif notamment overleaf et surtout Gitlab qui est un outil également nouveau pour moi et que j'ai très apprécié du fait des différents avantages qu'il confère même si nous avons pu rencontrer quelques fois des problèmes avec son utilisation.

Conclusion

Au début de ce projet, je n'étais pas un fêru d'informatique, mais j'ai pu à travers découvrir tout ce qu'il était possible de réaliser grâce à cette discipline et son importance dans ma formation de Data Scientist. J'ai donc pu à travers ce projet développer un certain intérêt pour l'informatique et compte approfondir mes connaissances en la matière. Malgré les diverses difficultés rencontrées notamment dues au confinement qui n'ont pas rendus la réalisation de ce projet aisé, je ressors un bilan positif de cette expérience. J'ai appris beaucoup de choses à travers ce projet qui me seront très certainement utiles pour la suite de ma formation et dans ma future carrière professionnelle.

Je tiens à remercier l'ensemble des membres de mon groupe et en particulier Lucie notre chef d'équipe pour sa bonne organisation et pour son esprit participatif. J'aimerais également remercier notre tuteur pour son suivi et pour l'aide qu'il nous a apporté tout au long du projet ; et remercier notre enseignant Rémi Pépin qui nous a également aidé à résoudre nos problèmes durant le projet et qui m'a également aidé à comprendre un certain nombre de notions notamment en rapport avec la DAO et l'utilisation de Git.

Ce projet est vraiment très intéressant et important dans la formation offerte par l'ENSAI. Pour les années à venir il serait intéressant de permettre aux admis sur titre comme moi de pouvoir être avancé dans le cours de remise à niveau avant l'entame du projet pour éviter nous éviter un sentiment de déroute et d'inutilité dans les débuts du projet. Il serait peut-être aussi intéressant de permettre aux groupes de choisir le projet qui serait susceptibles de les intéresser (On apprend mieux en travaillant sur une thématique qu'on aime).

B.3 Hamza El Youmni

La répartition ayant lieu au début de l'année scolaire, m'a permis de joindre un groupe de 5 membres notamment « Shiraz - Adamaly - Wissem Baba Moussa - Moussa Kafando et Lucie Martin ». Dans le cadre d'une réunion de lancement de projet, afin de bien lire le sujet et de bien définir ses enjeux, nous avons pu déterminer comment va se dérouler le projet en abordant tous les aspects et les points clés, d'où l'importance de réussir cette étape cruciale. Nous avons créé par la suite un groupe sur Messenger nous facilitant la communication, la répartition des tâches et surtout la fixation des dates des prochains rendez-vous. Nous avons une bonne cheffe d'équipe, elle précisait les rôles et responsabilités de chacun et faisait approuver les règles de fonctionnement du groupe. Lucie était une véritable cheffe d'orchestre qui organisait et planifiait le travail du projet afin de s'assurer que le projet respectait les délais alloués.

Le premier rendez-vous s'est déroulé dans une salle de l'ENSAI, chaque membre du groupe a choisi un diagramme sur lequel il devrait travailler. De ma part j'ai commencé par me renseigner sur internet sur le fonctionnement et le principe du diagramme d'activité et du diagramme de Gant, puis je me suis lancé à les faire. Cette répartition de diagrammes a subi quelques modifications ce qui m'a poussé à approfondir mes recherches en regardant quelques vidéos à propos du modèle logique et physique permettant de mieux comprendre et de manipuler ce genre de diagramme. Entre temps, nous avons pu nous organiser à fixer un nombre considérable de réunions au sein de l'école dans le but de compléter le premier rapport. Après l'accomplissement de ce dernier nous avons eu un retour de la part de notre tuteur et grâce à lui et à sa compétence dans le domaine de l'informatique nous avons constaté les petites erreurs que nous avons commises et qu'on avait l'intention de corriger dans le 2ème rapport.

Personnellement, j'ai profité et j'ai beaucoup appris de cette expérience. D'un côté j'ai appris à sélectionner des informations et à valoriser toutes les sources disponibles. D'un autre côté le travail de groupe m'a permis de travailler en équipe avec des personnalités et des caractères différents que l'on trouve régulièrement en entreprise. Il faut donc sans cesse s'adapter. Ceci était en ma faveur sur le plan pratique notamment à travailler d'une manière structurée, à organiser mon temps et à collaborer avec les autres membres du groupe. Je trouve que la patience dans le monde de l'informatique est une clé primordiale afin de surmonter les problèmes et trouver les solutions pertinentes.

Les séances du complément informatique et les quiz que nous avons passé m'ont permis d'améliorer mes compétences, d'apprendre des nouvelles connaissances et en plus de créer mes propres idées à propos de notre sujet intitulé Pokemon. Sauf que le cours de Git était un peu difficile et en particulier à manipuler dès la première fois. J'étais alors censé de bien me documenter pour appliquer ces bonnes pratiques dans notre projet d'informatique.

Je ne nie pas que j'ai trouvé des difficultés au début mais cela ne m'a pas empêché de bien me programmer. La planification de projet est cruciale et l'élaboration d'un plan planning doit être réalisé avec autant de soin car elle décidera de façon étroite de son bon déroulement par la suite et déterminera la réussite ou l'échec du projet. Pour cette raison je me suis bien renseigné sur le fonctionnement de ma classe ou de ma fonction et de faire une petite conclusion de ce que j'aurai besoin pour coder, puis j'ai sollicité auprès de pas mal de vidéos et de documents qui peuvent m'aider à faire un bon code sans aucun souci. Et il ne faut surtout pas délaisser Git, cet éminent nouveau logiciel est imposant très important pour développer efficacement des projets professionnels dans le futur.

En revanche, j'étais merveilleusement satisfait de mon groupe car chaque membre du était très ponctuel et motivé à aider l'autre pour accomplir sa mission et avancer. Bien que le confinement et la distance qui nous sépare, la communication entre nous n'a jamais cessé, on parlait souvent pour surmonter les problèmes qui nous envisageaient et de comprendre ce que l'autre a fait. Nous étions motivé à travailler,

discuter et échanger les idées .Pour moi ce travail en équipe est un terrain de jeu parfait pour développer les soft-skills. Se montrer flexible, ouvert à de nouveaux défis, faire preuve d'optimisme et de résilience face aux changements, faire face à la pression de manière saine, parvenir à réguler et canaliser mon stress en étant capable de le reconnaître et d'adopter la bonne attitude pour retrouver mon calme. Nous avions une bonne ambiance au sein du groupe, nous étions ponctuels et déterminés. Chaque semaine, nous fixions une heure précise pour s'appeler sur Teams en faisant un partage d'écran dans le but de faire le point sur ce que nous avons fait durant toute la semaine. La communication entre nous a joué un grand rôle pour s'avancer et résoudre les problèmes. On utilisait des différents outils de communication soit écrits ou oraux. Ce projet m'a offert l'opportunité de travailler dans un groupe surdoué et engagé. Je conclus que pour bien gérer un projet avec un grand succès nous devons établir les priorités, puisque nous n'avons jamais le temps que nous souhaitons. Ainsi, nous devons nous concentrer sur les éléments qui auront le plus d'impact.

J'ai ainsi appris comment être capable de me remettre en question et avoir envie de progresser, être capable de mettre mon égo de côté, d'admettre mes erreurs, de reconnaître mes faiblesses, d'avouer lorsque je ne sais pas, être ouvert et à l'écoute de ce que les autres ont à m'apprendre, quelle que soit leur position, les mettre en avant lorsque c'est mérité, savoir demander de l'aide. Cette expérience à ajouter énormément de chose pour moi et même pour les autres membres de groupe.

Au terme de ce travail, je tiens à exprimer ma profonde gratitude à notre cher professeur et tuteur Monsieur LEVEUGLE pour son suivi et son énorme soutien, qu'il n'a cessé de nous prodiguer tout au long de la période du projet, pour le temps qu'il nous a consacré et pour les précieuses informations qu'il nous a communiqué avec intérêt et compréhension. Son aide est aussi importante sur le plan technique que sur le plan humain.

B.4 Moussa Kafando

Appréhension du projet

Généralement, les projets de classes constituent à la fois une source d'angoisse et un très bon cadre d'apprentissage où l'on acquiert beaucoup de connaissances à la fois techniques et humaines. Ainsi, avant de commencer le projet, chaque fois que je lisais le sujet j'étais très inquiet à l'idée que mes camarades de groupe et moi ne puissions le mener à bien malgré l'ensemble des connaissances que nous aurions mobilisé. Notre sujet était de concevoir le jeu Pokémon en Python. Pour cela, nous avions à mobiliser des connaissances en UML pour les différents graphiques, en SQL pour les bases de données et en python pour la programmation (conception).

Aussi, la répartition du groupe faisait que j'étais amené à travailler avec des personnes que je n'avais pas l'habitude de côtoyer. Je ne connaissais pas leur humeur et leur degré d'engagement pour les projets de classe.

Il faut noter que le management de Lucie sur le projet a été d'une immense utilité, sa coordination et sa manière de faire impliquer tous les membres dans le projet. Ensuite, l'implication de tous les membres du groupe a été très utile pour arriver au terme de ce projet.

Vécu de l'expérience : bel esprit d'équipe

Dès le début du projet, j'ai pu apprécier à quel point mes camarades de groupes étaient très motivés. Chacun donnait le meilleur de lui-même pour que nous puissions avoir un jeu fonctionnel avec toutes les fonctionnalités de base dans un premier temps, et dans un second temps que nous puissions aller au-delà de ces fonctionnalités basiques pour avoir un très beau jeu. La phase analyse conceptuelle a pris plus de temps qu'il ne fallait à mon avis, mais par la suite j'ai pu comprendre à quel point il était nécessaire d'être d'accord sur le fonctionnement général de notre jeu. Concernant la communication, dès l'entame du projet, les réunions se faisaient chaque semaine.

Le temps qui nous a été accordé par la conception, bien que considérable, n'a pas été suffisant au vu de nos ambitions pour le projet. Il nous a fallu trouver des horaires supplémentaires. Nous organisions des réunions deux fois par semaine et souvent ce sont des réunions imprévues (des réunions programmées une journée ou quelques heures en avant sa tenu).

J'ai beaucoup apprécié la qualité des discussions, les implications des uns et des autres pour la rédaction des divers rapports, la création des bases de données, les requêtes, la définition du diagramme de classe, l'intégration de l'API, l'exportation de données sous format json...

Connaissances acquises

Le projet informatique a été pour moi un cadre d'apprentissage technique et humain. Ainsi sur le plan technique, tout d'abord j'ai pu améliorer mes connaissances en programmation orientée objet avec python à travers les différentes classes que j'ai codé. Ensuite, j'ai appris comment utiliser une API à travers les différentes réunions, l'explication de Shiraz qui a codé PokéAPI ainsi que la lecture de son code, j'ai appris à mieux utiliser de nouveaux outils comme PyCharm et GitLab. Je trouve que ces outils sont très intéressants et je suis plutôt satisfait d'avoir pu les découvrir. Aussi, j'ai amélioré mes connaissances en modélisation UML avec les différents diagrammes de classe, d'activité, et de séquence (que j'ai codé). Enfin, J'ai appris à mieux structurer mon code en utilisant l'architecture, vue, Service, DAO. Sur le plan humain, le projet m'a permis de travailler, de discuter et de connaître plus amplement d'autres camarades de classe.

Difficultés rencontrées

Malgré le travail réalisé avec mes camarades, personnellement à l'instar des autres membres du groupe, j'ai rencontré beaucoup de difficultés avec pycharm. Au début de la programmation, je ne savais pas

exactement par où commencer. J'étais dans l'incapacité de « débbugger » les programmes que j'écrivais. Chaque fois que je rencontrais des problèmes de bug, j'appelais Lucie ou les autres membres du groupe pour qu'ils m'aident à corriger les erreurs. Pour pallier à ces problèmes, j'ai lu et j'ai repris tous les TD faites en classe, ça m'a beaucoup aidé à structurer mes codes et à être au même niveau. Ainsi, j'ai développé la classe Pokedex, Achat et Inventaire du jeu. De plus, j'ai aussi développé, dans le cadre de l'évaluation, la classe de tests unitaires associée à la classe Dresseur. Il faut noter que souvent j'avais du mal à pusher mes codes, il y avait des conflits que j'ai pu résoudre. Aussi, durant les deux dernière semaine avant le rendu final j'ai eu pas mal de difficultés pour me connecter à Pycharm, j'écrivais mon code sur spyder que j'envoyais à un membre du groupe pour pusher pour moi et là aussi je ne pouvais pas tester le code. Mais finalement j'ai pu résoudre le problème grâce à notre tuteur et aux script envoyé par par M. PEPIN.

Conclusion

En conclusion, à la fin du projet, je tire un bilan très positif. Dans un premier temps, je suis satisfait non seulement des membres du groupe que j'ai pu réaliser dans le cadre de ce projet. Dans un second temps, je suis très heureux de toutes les nouvelles connaissances que j'ai pu acquérir à travers ce projet. Enfin, je suis très heureux d'avoir travaillé avec Lucie Martin, Shiraz Adamaly , Wissem Baba-Moussa et Hamza El Youmni. Sans oublier aussi SylvainLEVEUGLE notre chargé de suivit de projet qui répondait toujours à nos mails et qui n'hésitait pas à nous aider chaque fois que nous rencontrions des problèmes techniques avec Pycharm, nous vous remercions pour votre disponibilité, vos remarques et vos conseils tout au long de ce projet.

B.5 Lucie Martin

Ce projet informatique vient à la suite d'un premier projet de programmation orientée objet et de mon stage de première année, au cours duquel nous avons réalisé, à trois, un programme de correction automatique d'exercices. J'ai donc commencé ce projet avec une pratique de Python importante malgré le fait de venir d'un cursus économique. Cela m'a permis de connaître par avance le système de versionnage Git, même si je n'avais pas encore utilisé les lignes de commandes, et d'avoir des acquis solides en diagrammes UML, à l'exception de la conception d'un diagramme de séquence. De plus, le fait d'avoir déjà joué au jeu Pokémon m'a aidé en comparaison à certains de camarades non initiés, à mieux visualiser le fonctionnement du jeu et ses spécificités. Je suis, par ailleurs, quelqu'un qui aime beaucoup travailler de manière organisée ce qui m'a conduit à endosser naturellement le rôle de chef de projet.

Je me suis donc investie dans la gestion du projet en me chargeant de la répartition des tâches entre les différents membres du groupe, de la planification des réunions et de la communication avec notre tuteur. J'ai ainsi réalisé notre diagramme de Gantt final et la rédaction de la partie du rapport faisant référence à la gestion interne du projet. En termes, de travail personnel, j'ai réalisé et mis à jour le diagramme de classes de notre application. J'ai rédigé la première version de notre système de référencement des fichiers et modules. Lors de la phase de développement, je me suis plus particulièrement consacrée au développement de la classe Manager et de son interaction avec la couche contrôleur, soit les menus de l'application. J'ai aussi codé une partie des classes faisant appel à la notion de Data Access Objects. Par la suite, en duo avec Wissem, nous avons réalisé l'inscription et la connexion de joueurs à notre application. Puis, nous avons tenté de déboguer les connexions entre les différentes classes de notre code. J'ai relu le code et implémenté le design de notre application pour la rendre plus ludique et visuelle pour le joueur. Pour finir, j'ai rédigé le fichier README et pris l'initiative de créer un dossier supplémentaire Installation qui permettront à une personne extérieure à notre groupe d'installer notre application sur sa propre machine.

Au cours de ce projet, j'ai appris à utiliser de manière plus complète Gitlab en utilisant les lignes de commandes et non plus le cliquer bouton. J'ai aussi développé de nouvelles bonnes pratiques de codages notamment pour déboguer l'application. En effet, je n'osais pas assez faire usage des "print" pour vérifier les sorties de fonctions ou les types de certains paramètres. De plus, le stockage dans une base de données reliée à une application Python était une nouveauté pour moi. Ce n'est pas quelque chose que j'avais géré durant mes précédents projets. Je n'avais jusqu'alors qu'une vision séparée de la base de données et du programme. La DAO m'a permis de mieux appréhender les liens possibles entre les langages SQL et Python. De même, j'ai pour la première fois codé seule les classes de menus qui permettent d'interagir avec l'application. Finalement, je me suis rendue compte que cette partie du code était plus abordable qu'il n'y paraît au premier abord. Enfin, bien que n'ayant pas codé directement cette partie de l'application, je comprends dorénavant mieux le fonctionnement d'une API.

Malheureusement, j'ai rencontré quelques difficultés techniques lors de la mise en place du projet. J'ai rencontré des problèmes de définition d'interpréteur que mon ordinateur ne reconnaissait pas et que Git pushait sur le dépôt en ligne. De plus, j'ai dû faire face à des problèmes avec la machine virtuelle et la base Postgres de l'école qui ont saturé puisqu'elles étaient utilisées par trop d'élèves en même temps. Pour ne pas rester bloquée, j'ai tenté d'installer une base de données en local sans grand succès. Cela m'a conduit à prendre plusieurs rendez-vous téléphoniques avec le professeur référent du projet pour régler ces problèmes. Ensuite, j'ai eu de petites difficultés liées aux merges mais j'ai réussi à les régler avec l'un de mes camarades. Il m'a montré les aides qu'offrait Pycharm pour ce genre de problème. J'ai également ressenti des difficultés à gérer la charge de travail accumulée avec la réalisation du projet, sa gestion et la quantité de cours à suivre en parallèle. En effet, la crise du Corona Virus a conduit à la mise en place de cours en ligne et de cours à faire en autonomie, ce qui a considérablement augmenté la quantité de travail à fournir. La gestion du projet n'a pas non plus été chose facile, malgré les appels en visio-conférences

et les suivis sur Teams. Bien qu'ayant demandé des avis sur la répartition des tâches et ayant rappelé plusieurs fois celle-ci, il semblerait qu'il y ait eu quelques fois des problèmes de compréhension de la part de quelques uns de mes camarades. Nous aurions donc pu avoir des tâches réalisées par deux camarades en même temps sans le savoir. Heureusement, cela a pu être évité. J'ai du rester vigilante pour éviter que cela ne se produise. Du plus, j'ai parfois eu l'impression d'avoir du mal à motiver une partie des membres du groupe. Par exemple, j'ai ressenti un manque de participation collective aux réunions internes et avec le tuteur. Cela a été pour moi une énorme source de stress et d'inquiétude quand à la réalisation du projet et le fait de réussir à suivre les cours en parallèle. Mes solutions ont été de suivre les conseils de deux des membres de mon groupe, notamment pour la planification des réunions afin de réduire le stress et de compter sur le soutien moral de la part de personnes extérieures au projet.

Suite à cette expérience, je pense que je pourrais encore m'améliorer sur la documentation du code que je ne pense pas souvent faire. Ceci pourrait permettre une meilleure compréhension de mon travail par les autres membres de mon groupe mais aussi par des personnes extérieures, comme le tuteur par exemple. J'aimerais aussi être plus à l'aise sur les options d'aides au débogage de Pycharm ou encore directement dans le terminal. Ayant aussi souvent des problèmes d'installation de logiciel ou de langage sur mon ordinateur, j'aimerais aussi être plus autonome de ce point de vue. Au delà du domaine technique, peut-être que ai-je manqué de communication et de pédagogie avec mes camarades ce qui a pu générer de l'incompréhension. Je pense que je devrais aussi gagner en assurance pour mieux gérer les compétences et attentes de mes camarades en tant que chef de projet. Enfin, je dois travailler sur moi même pour apprendre à mieux gérer mon stress.

En définitive, ce projet m'a permis de réaliser que l'intérêt que je porte à l'informatique est plutôt de l'ordre du ludique. En effet, pour moi, coder c'est un peu comme résoudre un casse-tête pas à pas. Ce domaine m'intéresse mais je ne compte cependant pas m'orienter vers la filière SID de l'école. Je cherche davantage à acquérir de la polyvalence dans les tâches qu'une entreprise pourrait me confier plutôt que de me spécialiser en informatique. J'ai, malgré tout, la volonté de garder une part significative d'informatique dans ma vie professionnelle future.