Bordeaux INP
ENSC

Rapport
Programmation Avancée

Pokédojo

Cavaillole Célia, Gorbatoff Alexane - 1A ENSC
24/05/2019

Table des matières

Introduction

L'objectif de ce projet est de créer un jeu en C# : Pokédojo, permettant de simuler un tournoi où s'affrontent des pokémons. Pour cela nous avions des règles à respecter et l'obligation de s'organiser avec GitHub. Nous allons expliquer dans ce document tout d'abord notre organisation, ensuite nos choix de modélisation et de conception et enfin nos manières de représenter et de construire les différentes fonctionnalités du jeu.

Gestion de projet

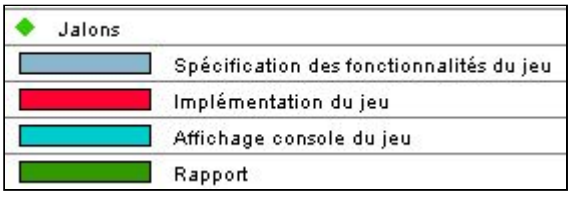

Figure 1. Légende des plannings

Planning prévisionnel

Le sujet nous a été distribué le vendredi 19 avril 2019. Nous avions prévu de commencer par se familiariser avec le jeu Pokémon en réalisant quelques recherches. Ainsi nous avions prévu d'identifier ses fonctionnalités puis de réfléchir à la définition des différentes classes, champs et méthodes. Ensuite s'enchaîne la création des classes, constructeurs et accesseurs suivi de l'implémentation des méthodes nécessaires aux fonctionnalités de base. Après avoir réalisé ce travail, nous allons donc procédé à l'affichage puis aux fonctionnalités avancées de notre jeu Pokédojo. Sans oublier bien évidemment la rédaction du rapport.

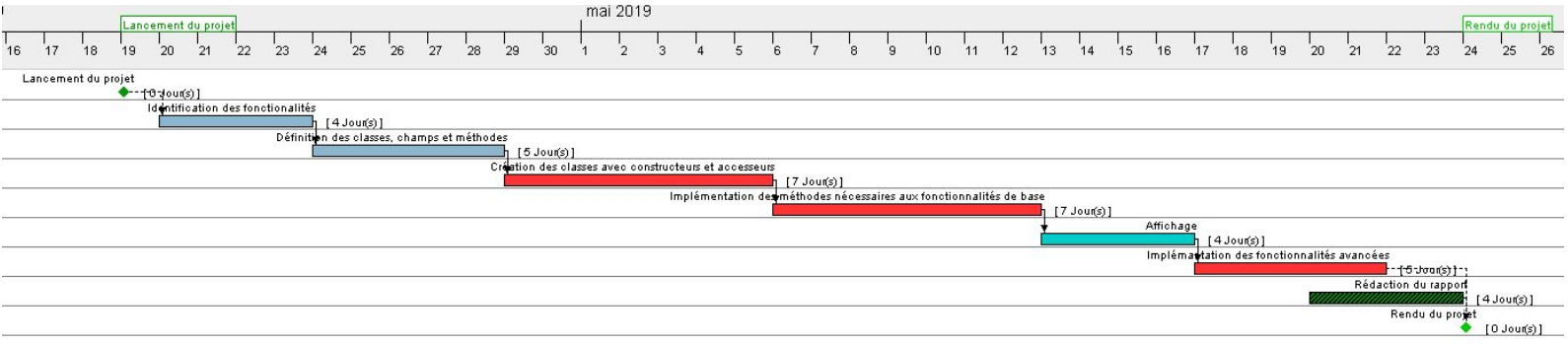

Figure 2. Planning prévisionnel

Planning final

Pendant la semaine de vacances nous avons identifié les fonctionnalités dont le jeu dispose et réfléchi sur feuille aux potentielles classes à construire chacune de notre côté. Nous avons également pris en main GitHub.
Une fois cela réalisé, nous avons commencé à créer les classes, les champs et à manipuler les constructeurs et les accesseurs. Ensuite nous avons implémenté les méthodes et avons réfléchi en parallèle à l'affichage de notre jeu. Ainsi nous sommes passées à l'implémentation des fonctionnalités avancées du jeu.
Le rapport a été commencé le week end du 11 mai.

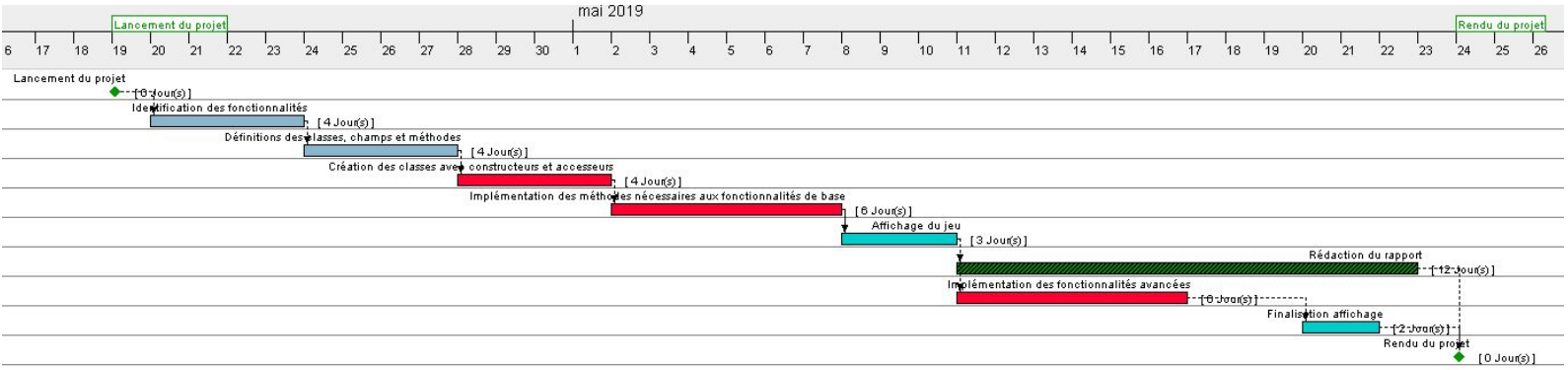

Figure 3. Planning final

Répartition des tâches

Dans un premier temps nous avons privilégié le travail de groupe pour bien comprendre les fonctionnalités du jeu et se familiariser avec les Pokémons. Nous avons alors défini les classes, champs et méthodes.

Dans une deuxième partie, Alexane a créé les classes avec les constructeurs et les accesseurs pendant que Célia créait la base de données des Pokémons. Une fois cette étape réalisée, nous avons procédé ensemble à la création des méthodes nécessaires aux fonctionnalités de base et à l'affichage du jeu.

Dans un troisième temps, Alexane a effectué l'implémentation des fonctionnalités avancées pendant que Célia réalisait la modélisation UML.

Enfin, nous avons finalisé l'affichage et avons rédigé le rapport ensemble.

Conception du jeu

Modélisation UML

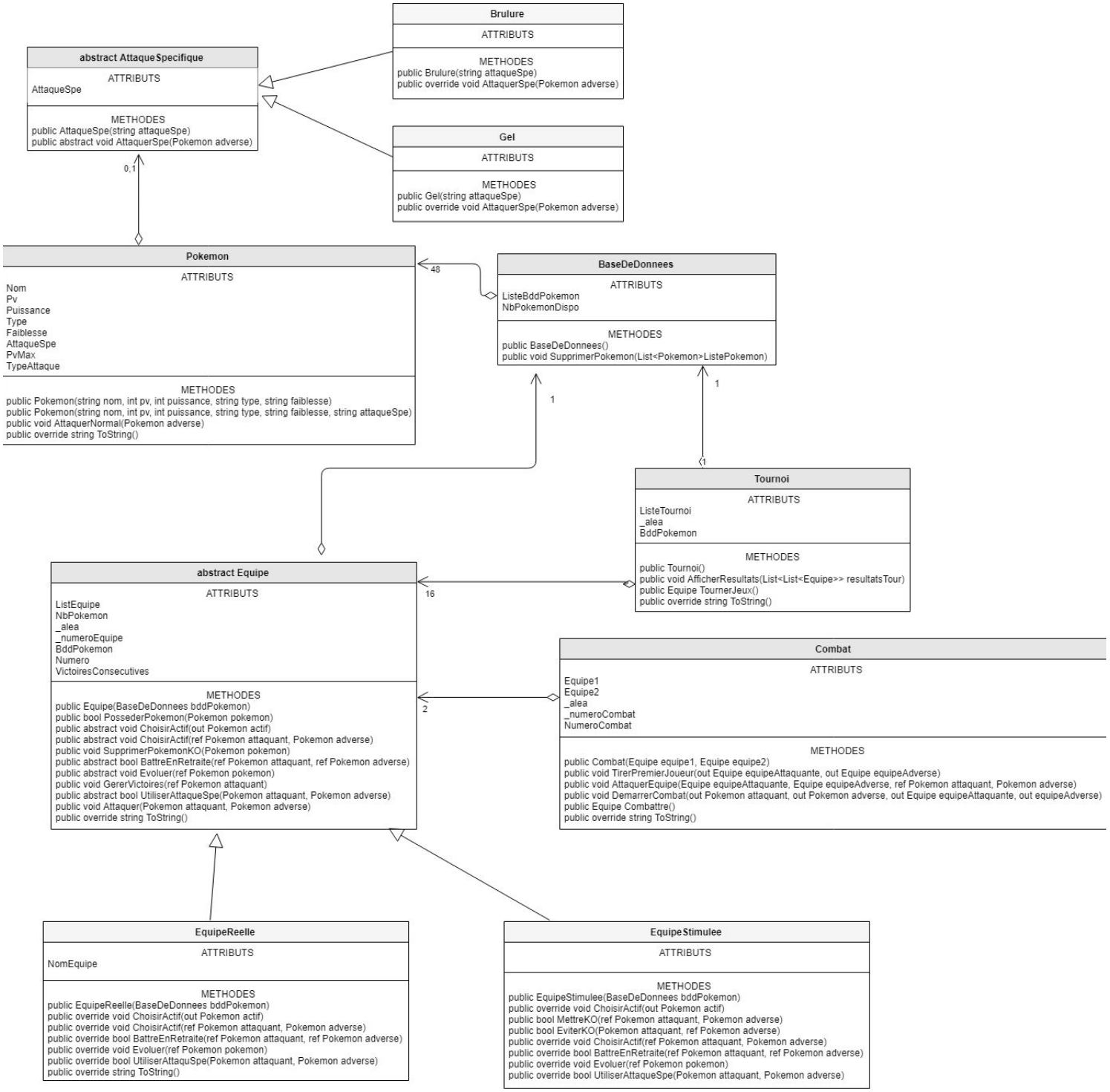

Figure 4. Modélisation UML

Voici ci-dessus le diagramme UML représentant l'organisation de notre code. Sa grande taille ne le rend pas vraiment visible sur ce document, ce pourquoi nous vous l'avons mis en fichier png nommée "DiagrammeUML" avec le rendu.

Choix de conception

Pour la conception de ce jeu, nous avons été amenée à effectuer des choix de modélisation. Nous avons tenté de faire ces choix de façon à ce que le jeu puisse être amélioré, par exemple par l'ajout de Pokémons ou encore de fonctionnalités complémentaires. Ces choix sont décrits dans cette partie.

Les classes et leurs relations

Notre code est composé de 6 classes principales dont 2 classes abstraites (*Equipe* et *AttaqueSpecifique*) desquelles découlent pour chacune 2 classes héritées.
Les classes *EquipeReelle* et *EquipeSimulee* héritent de la classe *Equipe*. Ce sont toutes deux des types d'équipes différentes qui n'ont pas exactement le même comportement et ne définissent donc pas de la même manière l'ensemble de leurs méthodes. En effet, la classe *EquipeReelle* modélise une équipe appartenant à un joueur réel, tandis que la classe *EquipeSimulee* modélise une équipe appartenant à un joueur simulé par l'ordinateur.
Les classes *Brulure* et *Gel* quant à elles, héritent de la classe *AttaqueSpecifique*. Elles permettent chacune de définir un type d'attaque spécifique particulière.

Il existe différentes relations entre ces classes :

- La classe *Pokemon* est en relation d'association avec la classe *AttaqueSpecifique* : un Pokémon peut posséder un type d'attaque spécifique.

- La classe *BaseDeDonnees* est en relation d'association avec la classe *Pokemon* : une base de données est composée de 48 Pokémons.

- La classe *Equipe* est en relation d'association avec la classe *BaseDeDonnees* : une équipe possède 3 Pokémons présents dans la base de données.

- La classe *Combat* est en relation d'association avec la classe *Equipe* : un combat implique 2 équipes différentes.

- La classe *Tournoi* est en relation d'association avec la classe *Equipe* et la classe *BaseDeDonnees* : un tournoi se déroule en présence de 16 équipes différentes, chacune étant composée de 3 Pokémons distincts faisant parties de la base de données (2 équipes ne peuvent avoir un Pokémon en commun).

Modélisation d'un Pokémon

Un Pokémon est caractérisé par un nom, un nombre de point de vie (Pv), une puissance d'attaque, un type, une faiblesse et éventuellement une attaque spécifique.

Modélisation de la base de données

Une classe BaseDeDonnees a été mise en place afin d'implémenter notre base de données. Elle permet la création d'une liste, elle-même composée de 48 listes. Chacune de ces 48 listes étant constituée de 3 Pokémons correspondant à une lignée, c'est-à-dire une Pokémon et ses deux Pokémons évolués.

Modélisation d'une équipe

Une équipe est composée de 3 Pokémons. Chaque Pokémon a deux évolutions possibles. Ainsi, pour modéliser une équipe, nous créons une liste de trois listes de Pokémon et de leurs évolutions associées.

```
{
        {Pokémon1,Pokémon1Evolution1,Pokémon2Evolution2},
        {Pokémon2,Pokémon2Evolution1,Pokémon2Evolution2},
        {Pokémon3,Pokémon3Evolution1,Pokémon3Evolution3}
}
```

Figure 5. Liste modélisant une équipe

Le Pokémon d'indice 0 de la liste d'un Pokémon et de ses évolutions, représente le Pokémon le moins évolué de la lignée. A chaque évolution, le Pokémon qui subit une évolution est supprimé de la liste et son Pokémon évolué devient celui d'indice 0.

Lorsqu'un Pokémon est mis KO c'est l'ensemble de la liste du Pokémon en question et de ses Pokémons évolués qui est supprimée.

Modélisation d'une attaque spécifique

Une classe, en relation avec la classe *Equipe*, a été créée pour ce type d'attaque. Il s'agit d'une classe abstraite dont héritent les classes qui décrivent les fonctions relatives aux différentes attaques spécifiques.

On associe un objet AttaqueSpecifique à un Pokémon en fonction de l'attribut AttaqueSpe passé en paramètre dans le constructeur du Pokémon : AttaqueSpe "gel" donne lieu à la création d'un objet de type Gel, AttaqueSpe "brulure" donne lieu à la création d'un objet de type Brulure. Si le Pokémon est créé sans AttaqueSpe (à l'aide du constructeur ne prenant pas d'AttaqueSpe en argument), alors la création de l'objet AttaqueSpecifique n'a pas lieu pour ce Pokémon.

Modélisation des victoires consécutives

On attribue à chaque objet *Equipe* un paramètre *VictoiresConsecutives* valant 0 par défaut. Il prend la valeur 1 dès que le Pokémon actif réalise un premier KO de son

adversaire. Si à l'attaque suivante l'équipe n'a pas changé de Pokémon actif et que ce dernier réalise un KO, *VictoiresConsecutives* est incrémenté, sa valeur passe à deux, ce qui permet de modéliser deux victoires consécutives et donc la première évolution du Pokémon actif. Sinon, la valeur repasse à 0. De la même manière, à l'attaque suivante de l'équipe, la valeur peut passer à 3, ce qui correspondra à 3 victoires consécutives et donc à la deuxième évolution possible du Pokémon.

Modélisation d'un tournoi

Un tournoi est composé de 16 équipes. Dans la classe Tournoi, ces équipes sont contenues dans une liste d'objets *Equipes*. Dès qu'une équipe est éliminée d'un tournoi elle est supprimée de cette liste. A la fin du tournoi, la liste ne contient plus que l'équipe victorieuse.

Choix des fonctionnalités

Un tournoi Pokédojo implique une équipe appartenant à un joueur réel et 15 équipes appartenant à 15 joueurs simulées par ordinateur. Chaque équipe est composée de 3 Pokémons choisis aléatoirement parmi une base de données de 48 Pokémons. Un tournoi se déroule en 3 tours durant lesquels chaque équipe combat une équipe adverse. La fin d'un combat a lieu lorsque tous les Pokémons d'une équipe ont été mis KO. L'équipe en question est alors éliminée et ne participe pas au tour suivant.

Les équipes simulées ne jouent pas de manière aléatoire, elles possèdent une certaine intelligence définie dans les fonctions spécifiques à la classe *EquipeSimulee*. La façon dont ce type d'équipe fait ses choix sont expliquées ci-dessous.

Choix du Pokémon actif

Au cours d'un combat, on distingue deux types de choix du Pokémon actif d'une équipe : choix d'un Pokémon actif au début du combat et choix d'un Pokémon actif en cours de combat.

- Choix d'un Pokémon actif au début du combat :

Dans ce cas, on demande au joueur de choisir son Pokémon actif avant d'annoncer si c'est lui ou son adversaire qui attaque en premier. Son choix est donc fait sans connaître le rôle qu'il aura lors de la première attaque (attaquant ou adverse).

Pour les équipes réelles, on demande au joueur de choisir parmi les Pokémons qu'il possède encore dans son équipe, c'est-à-dire ceux qui n'ont pas été mis KO au cours des combats précédents.

Pour les équipes simulées, le choix est fait de façon aléatoire parmi les Pokémons toujours présents dans l'équipe.

- Choix d'un Pokémon actif en cours de combat :

Au cours d'un combat, lorsque le Pokémon actif d'une équipe est mis KO l'équipe doit choisir un nouveau Pokémon actif qui aura le rôle d'attaquant lors de la prochaine attaque.

Pour les équipes réelles, on demande au joueur de choisir parmi le Pokémons qu'il possède dans son équipe (même principe que le choix fait en début de combat).

Les équipes simulées choisissent un Pokémon ayant une puissance d'attaque minimale supérieure au nombre de PV de l'adversaire si possible. Sinon, elles choisiront un Pokémon ayant un nombre le point de vie minimal supérieur à la puissance d'attaque de l'adversaire (pour éviter qu'il soit mis KO au tour suivant). Si aucun des deux choix n'est possible, il se fera de manière aléatoire (de la même façon que le choix fait en début de combat).

```
Veuillez saisir un nom d'équipe :
EquipeCel
Tournoi de 16 équipes en trois tours


################################################################################
PREMIER TOUR

Vous combattez contre l'équipe 11

Equipe numéro 11
---------------------------------------
Nom : grainipiot
Nombre de point de vie : 40
Puissance d'attaque : 40
Type : plante
Faiblesse :insecte
---------------------------------------
Nom : tortipouss
Nombre de point de vie : 55
Puissance d'attaque : 68
Type : plante
Faiblesse :insecte
---------------------------------------
Nom : terhal
Nombre de point de vie : 40
Puissance d'attaque : 55
Type : acier
Faiblesse :feu
Attaque spécifique : brulure
---------------------------------------

Equipe EquipeCel : Equipe 1
---------------------------------------
Nom : ptitard
Nombre de point de vie : 40
Puissance d'attaque : 50
Type : eau
Faiblesse :electrik
---------------------------------------
Nom : obalie
Nombre de point de vie : 70
Puissance d'attaque : 40
Type : glace
Faiblesse :plante
Attaque spécifique : gel
---------------------------------------
Nom : galekid
Nombre de point de vie : 50
Puissance d'attaque : 70
Type : acier
Faiblesse :combat
---------------------------------------

Quel Pokémon voulez-vous faire combattre ?
```

Figure 6. Choix du pokémon actif

Battre en retraite un Pokémon actif

A la fin de chaque attaque, chaque équipe (attaquante ou adverse) peut décider de battre en retraite son Pokémon actif s'il lui reste d'autres Pokémons dans son équipe.

Pour les équipes réelles, une question leur est posé à chaque fin de combat afin qu'elles puissent décider ou non de faire battre en retraite leur Pokémon actif.

Une équipe simulée fait battre en retraite son Pokémon actif en fin de combat si :

- ➔ il est attaquant et que sa puissance d'attaque est inférieure au nombre de PV de son adversaire mais qu'il possède un Pokémon dont la puissance est supérieure à ce nombre de PV.
- ➔ il est adverse et que son nombre de PV est inférieur ou égal à la puissance d'attaque de son adversaire mais qu'il possède un Pokémon dont le nombre de PV est supérieur à cette puissance.

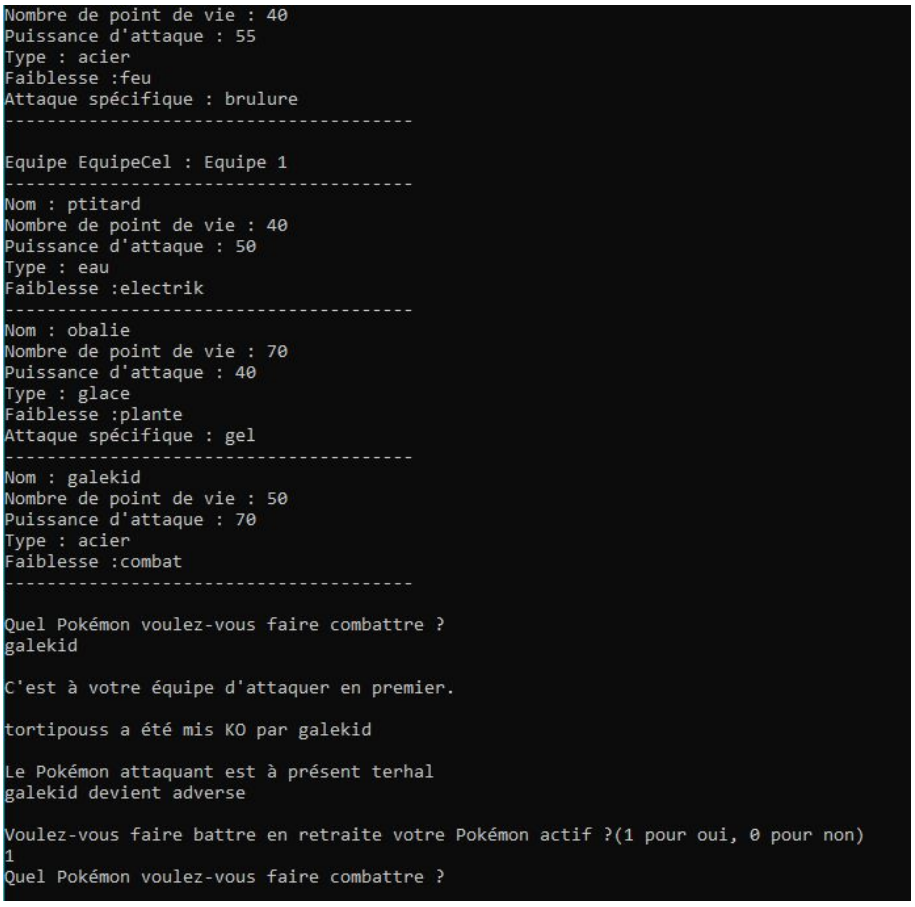

Figure 7. Demande au joueur réel, la fonction battre en retraite de son pokémon actif

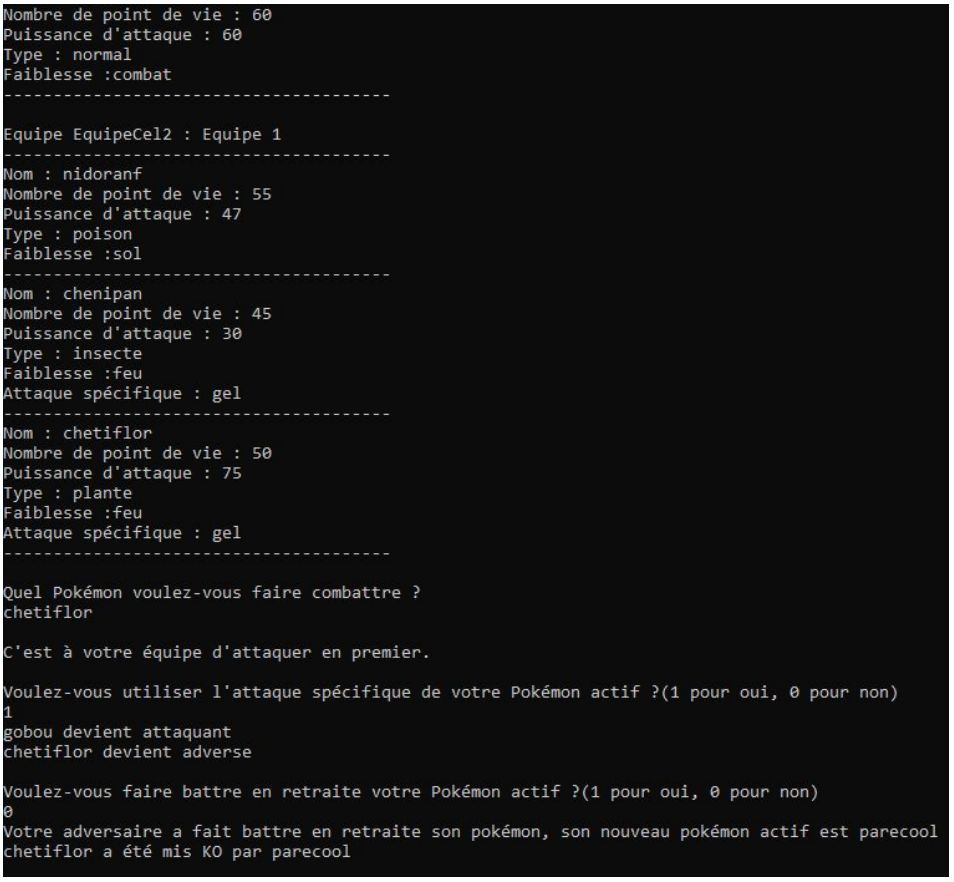

Figure 8. Fonction battre en retraite d'une équipe simulée par l'ordinateur

Évolution d'un Pokémon

Un Pokémon a deux évolutions possibles. Il évolue s'il fait deux victoires consécutives lors d'un combat, c'est-à-dire qu'il met deux fois d'affilée son adversaire KO et qu'il a survécu à une attaque de la part de son adversaire entre temps, donc qu'il n'a pas été battu en retraite pour l'attaque séparant ses deux victoires consécutives. A la troisième victoire consécutive, le Pokémon atteint sa deuxième évolution (évolution maximale). Le nombre de victoires consécutives d'un Pokémon est relatif à un combat, à chaque nouveau combat le nombre de victoires consécutives est remis à 0.

```
Puissance d'attaque : 57
Type : poison
Faiblesse :sol
Attaque spécifique : brulure
-------------------------------------
Nom : magby
Nombre de point de vie : 45
Puissance d'attaque : 75
Type : feu
Faiblesse :sol
Attaque spécifique : brulure
-------------------------------------
Nom : machoc
Nombre de point de vie : 70
Puissance d'attaque : 80
Type : combat
Faiblesse :psy
-------------------------------------

Quel Pokémon voulez-vous faire combattre ?
machoc

C'est à votre équipe d'attaquer en premier.

parecool a été mis KO par machoc

Le Pokémon attaquant est à présent tortipouss
machoc devient adverse

Voulez-vous faire battre en retraite votre Pokémon actif ?(1 pour oui, 0 pour non)
0
machoc devient attaquant
tortipouss devient adverse

Voulez-vous faire battre en retraite votre Pokémon actif ?(1 pour oui, 0 pour non)
0
tortipouss a été mis KO par machoc

machoc a évolué en machopeur
Le Pokémon attaquant est à présent granivol
```

Figure 9. Exemple d'évolution du Pokémon machoc (évolution 0) en machopeur (évolution 1) après 2 KO.

Attaques spécifiques

Certains Pokémons possèdent des attaques spécifiques qui peuvent être utilisées une unique fois par tournoi. Ces attaques causes des dégâts particuliers sur les Pokémons adverses.

Les attaques spécifiques modélisées dans notre jeu sont :

- Brûlure : Une brûlure réduit l'attaque adverse de moitié. Le Pokémon adverse perd 1/6 de son PvMax. Un Pokémon de type "feu" ne peut pas être brûlé.

- Gel : Le Pokémon adverse ne peut plus attaquer : sa puissance d'attaque vaut 0 durant toute la suite du tournoi. Un Pokémon de type glace ne peut pas être gelé.

A chaque attaque, si le Pokémon actif possède une attaque spécifique, l'équipe peut choisir ou non de l'utiliser, si elle n'a pas déjà été utilisée lors du tournoi. Les choix sont fait de la manière suivante :

Pour les équipes réelles, on demande au joueur s'il veut utiliser l'attaque spécifique de son Pokémon actif.

L'équipe simulée utilise l'attaque spécifique de son Pokémon actif, s'il en possède une, dans le cas où sa puissance d'attaque n'est pas assez élevée pour mettre KO son adversaire et qu'il sait qu'il est susceptible de se faire tuer par son adversaire au tour suivant (puissance d'attaque de l'adversaire supérieure au nombre de Pv du Pokémon actif de l'équipe).

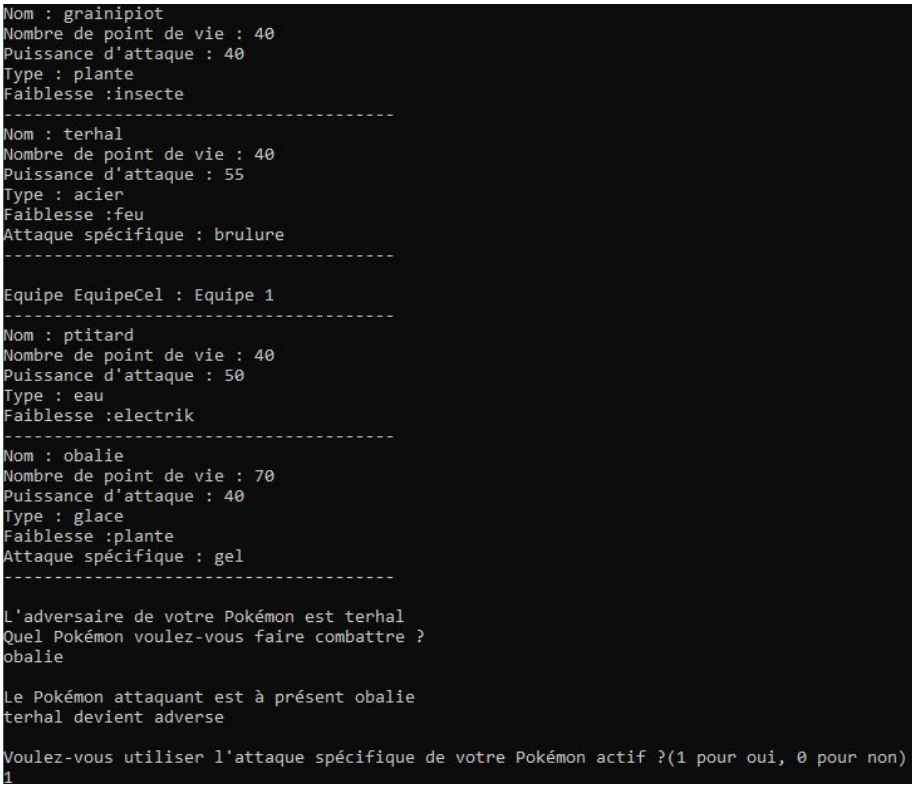

Figure 10. Exemple d'utilisation de l'attaque spécifique du pokémon actif d'un joueur réel

Affichage des résultats finaux

```
lixy a été mis KO par hypotrempe

Votre équipe a perdu le combat, vous êtes éliminé...

 ---------------------------------------------------------
| Combat 1 | Equipe 7  / Equipe 11 | Vainqueur : Equipe 7  |
 ---------------------------------------------------------
| Combat 2 | Equipe 13 / Equipe 14 | Vainqueur : Equipe 13 |
 ---------------------------------------------------------
| Combat 3 | Equipe 4  / Equipe 3  | Vainqueur : Equipe 4  |
 ---------------------------------------------------------
| Combat 4 | Equipe 6  / Equipe 12 | Vainqueur : Equipe 12 |
 ---------------------------------------------------------
| Combat 5 | Equipe 15 / Equipe 2  | Vainqueur : Equipe 2  |
 ---------------------------------------------------------
| Combat 6 | Equipe 5  / Equipe 9  | Vainqueur : Equipe 9  |
 ---------------------------------------------------------
| Combat 7 | Equipe 16 / Equipe 1  | Vainqueur : Equipe 16 |
 ---------------------------------------------------------
| Combat 8 | Equipe 10 / Equipe 8  | Vainqueur : Equipe 10 |
 ---------------------------------------------------------

################################################################################
SECOND TOUR

 ---------------------------------------------------------
| Combat 1 | Equipe 2  / Equipe 4  | Vainqueur : Equipe 4  |
 ---------------------------------------------------------
| Combat 2 | Equipe 16 / Equipe 12 | Vainqueur : Equipe 12 |
 ---------------------------------------------------------
| Combat 3 | Equipe 9  / Equipe 10 | Vainqueur : Equipe 9  |
 ---------------------------------------------------------
| Combat 4 | Equipe 7  / Equipe 13 | Vainqueur : Equipe 13 |
 ---------------------------------------------------------

################################################################################
TROISIEME TOUR

 ---------------------------------------------------------
| Combat 1 | Equipe 4  / Equipe 12 | Vainqueur : Equipe 12 |
 ---------------------------------------------------------
| Combat 2 | Equipe 9  / Equipe 13 | Vainqueur : Equipe 9  |
 ---------------------------------------------------------

################################################################################
FINALE

 ---------------------------------------------------------
| Combat 1 | Equipe 9  / Equipe 12 | Vainqueur : Equipe 9  |
 ---------------------------------------------------------

L'équipe vainqueur est l'équipe numéro 9


Voulez-vous recommencer une nouvelle partie ? (1 pour oui, 0 pour non)
```

Figure 11. Affichage des résultats finaux après une défaite de son équipe au premier tour

Tests réalisés

Nous avons réalisé deux types de tests différents :

- Tests spécifiques aux équipes simulées :

Le premier type de test a été mis en place dans le but de tester en particulier les fonctionnalités spécifiques aux équipes simulées par ordinateur. Ces fonctionnalités reposent sur une certaine "intelligence" que nous avons défini. Les utilisateurs n'étant pas informés de la façon dont une équipe réelle fait ses choix, nous nous sommes chargées de réaliser ces tests nous même, sans faire intervenir d'utilisateurs externes au projet. Ceci nous a permis de voir si les équipes simulées se comportaient bien de la manière dont nous l'avions prévu et donc de vérifier si l'ensemble de nos méthodes étaient bien définies.

- Tests spécifiques aux équipes réelles :

Le second test réalisé a consisté à expérimenter les fonctionnalités de jeu relatives à une équipe réelle. Ces tests ont été réalisés par des utilisateurs externes au groupe de notre projet. Ainsi, nous avons pu vérifier si la façon dont nous avions conçu notre jeu paraissait intuitive pour un utilisateur et si les règles présentées étaient assez claires. Ceci nous a amené à effectuer quelques modifications notamment concernant les affichages au cours d'un tournoi. Ces tests ont également été l'occasion d'observer la façon dont jouent les utilisateurs et les stratégies qu'ils mettent généralement en place. La prise en compte de ces observations nous a permis d'améliorer les fonctions dites "intelligentes" utilisées par les équipes simulées..

Bilan

Les difficultés que nous avons rencontré lors de la réalisation de ce projet ont été tout d'abord dans la prise en main de GitHub. C'est un logiciel dont la manipulation n'est pas simple, il nous semblait donc important de se familiariser avec cet outil dès le début de notre projet. Nous avons procédé à l'implémentation de l'ensemble du code initial une fois GitHub installé et son fonctionnement compris par les membres du groupe.

Une deuxième phase a été de comprendre les règles du jeu autour des Pokémons, n'étant pas forcément passionnées par ce genre de jeu, une seconde difficulté s'est mise en place. Pour y remédier nous nous sommes donc mis d'accord ensemble sur le rôle des fonctionnalités à mettre en place, pour ainsi les développer selon nos interprétations communes.

La modélisation des attaques spécifiques nous a demandé plus d'attention et de rigueur car elle a soulevé un grand nombre de questions. Pour résoudre le problème, nous avons donc décidé de développer seulement deux attaques spécifiques (brûlure et gel). Par conséquent, nous avons créé une classe spécialement dédiée à ce type d'attaque, ce qui permet d'ajouter facilement d'autres attaques spécifiques, rendant ainsi notre jeu modulable.