

# Rapport de développement

## Auteurs:

Vincent BUISSET (TDA – alpha - alternant)

Julien FAIDIDE (TDA – alpha - alternant)

## Dépendances du projet:

- JavaFX

## Architecture du projet (dossier src/)

- base Contient quelques classes basiques du projet.
- graphics/ Contient les objets JavaFX.
  - pages/ Contient les pages .fxml et les controllers pour chaque page.
    - capacityBuilder/ Page de choix des capacités.
    - fightPage/ Page du combat.
    - ligue/ Page de lancement de combats de ligue.
    - loadSave/ Page de chargement d'une sauvegarde.
    - mainPage/ Page d'accueil.
    - page2/ Page du menu principal.
    - TeamBuilder/ Page de choix des pokemons.
  - utilities/ Contient d'autres objets en rapport avec JavaFX.
    - dialogArea/ Page qui représente l'emplacement de la popu de dialogue.
- managers/ Contient tous les managers du jeu.
  - BattleEvent/ Gère les évènements du combat.
  - DamageManager Calcul et applique les effets des capacités sur les pokemons.
  - GameManager/ Controle la globalité du programme. Les actions principales.
  - PageManager/ Permet de changer de page dans la même fenêtre.
  - SoundManager Joue du son.
- Pokemons/ Contient la plus grosse partie des objets du modèle du jeu (les pokémons, les capacités, la ligue, les stats, les types etc.)

- resources/ Contient les ressources externes (css, polices d'écritures, sons images, documents .csv).
- saves/ Répertoire où sont placés les sauvegardes.
- test/ Programme Main de test (notamment utile pour tester la créations des objets à partir des fichiers .csv).
- utils/ Contient des exceptions et des méthodes utilitaires personnalisées.

## Répartition des tâches

Vincent BUISSET:

- Classes du modèle (PokemonDescriptor, PokemonCreature, BaseStat, Capacity, PokemonObject etc.)
- Classes usines (Stats, Capacities, Pokedex, Types, DamageManager)
- Managers (DamageManager, SoundManager)
- Pages (TeamBuilder, capacityBuilder, ligue)
- Utils
- Rédaction des documents

Julien FAIDIDE:

- Managers (BattleEvent, GameManager, PageManager)
- Pages (fightPage, loadSave, mainPage, page2, utilities)
- Base (Player, PlayerBot)
- Utils
- Mode multijoueur

## Difficultés rencontrées

- Mise en place de l'environnement de travail avec JavaFX: Nous avons passé beaucoup de temps pour réussir à installer JavaFX, à bien configurer les arguments de VM et le .gitignore. Beaucoup de tests et de recherches pour mieux comprendre l'environnement ont été nécessaires.
- Concevoir une architecture de pages avec javaFx. Après une phase de recherches, nous avons choisi une arborescence de dossiers et de classes générique. Celle-ci nous à permis de développer et ajouter au jeu diverses composants au ferez à mesure du temps, sans interférer dans nos taches respectives.
- Échange de données entre différentes fenêtres (CapacityBuilder et TeamBuilder): Une interface à due être conçue et utilisée dans le GameManager, elle permet à la fenêtre fille de récupéré le stage de l'application pour pouvoir utiliser un setter avec le choix de l'utilisateur.

- Prise en main de Scene Builder.
- Trouver des créneaux pour travailler, face à un grand nombre de projets et de partiels à préparer couplés avec des semaines de travail en entreprise chargés pendant les vacances de Noël.

## Implémentations / Evolution.

- I. Avant-propos.
- II. Les types, le pokédex, les capacités et les pokémons ont été implémentés.
- III. Le système de sauvegarde a été implémenté.
- IV. Un mode de combat classique a été implémenté (seul contre IA).
- V. Une ligue classique a été implémenté.
- VI. La gestion des types dans le calcul des dégâts a été implémenté.
- VII. Le mode Multijoueur.
- VIII. Les objets.

Les éléments suivants **n'ont pas été implémentés.**

- IX. Les états.
- X. L'acquisition de capacités.
- XI. Les objets tenables de boosts.
- XII. Mode Duo et trio.

Autres propositions d'évolutions:

- Un mode debug qui rajouterait des pokemons, objets et capacités surpuissantes dans la liste. Cela permettrait de tester le jeu rapidement.
- Prise en comptes des effets spéciaux de capacités (exemple: la capacité autodestruction tue son lanceur, torgnole se lance plusieurs fois, guillotine met forcément le pokemon ennemi K.O.  
[https://www.pokepedia.fr/Calcul\\_des\\_d%C3%A9g%C3%A2ts](https://www.pokepedia.fr/Calcul_des_d%C3%A9g%C3%A2ts) )