Rapport de Projet

Binary Deployment Squadron Manager

Loïc FABRE

Introduction

J'ai réalisé **Binary Deployment Squadron Manager** pour fournir aux joueurs du jeu de plateau X-Wing Miniatures, un outil permettant de gérer leurs escadrons et listes d'armée. Je joue à ce jeu avec mes deux frères depuis 5 ans, et comme il n'est plus maintenu par ses éditeurs (**Atomic Mass Games**), j'ai décidé de recréer leur Squad Builder afin de manier les règles à notre sauce et faciliter le démarrage d'une partie.

Conception et Architecture

Pour le backend j'ai utilisé Symfony et PHP avec une base de données gérée par Doctrine en MySQL. La partie frontend est gérée avec Twig en utilisant la bibliothèque Bootstrap. Pour la connexion entre le front et le back, j'ai utilisé Axios.

J'ai utilisé Turbo, géré par Symfony afin de fluidifier la navigation (sans chargement). J'ai principalement stocké les données des factions, des vaisseaux et des pilotes en session pour limiter les requêtes à la BDD et l'authentification est gérée par Symfony (au niveau de la sécurité, du hachage de mots de passe)

Fonctionnalités

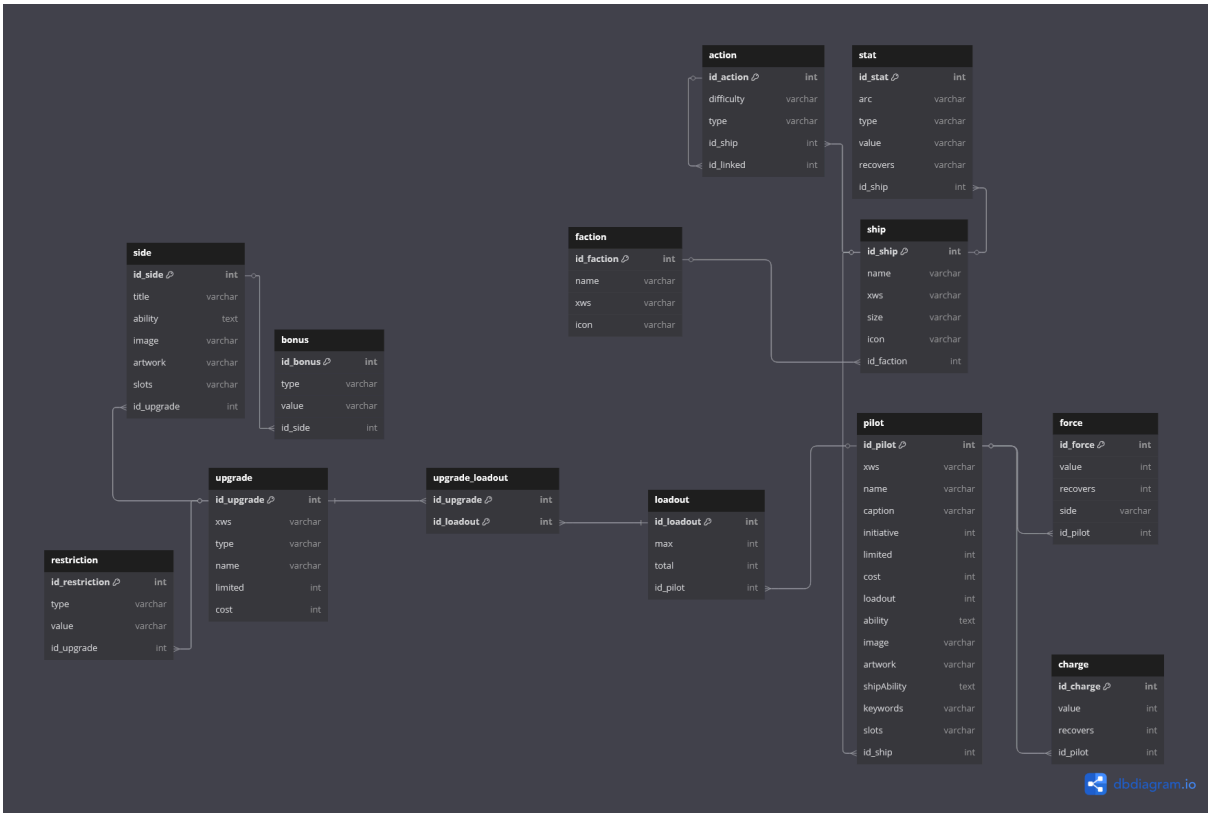
L'application permet aux utilisateurs de :

- Explorer les factions, vaisseaux et pilotes via un système de contenu collapsible.
- Visualiser les statistiques d'un pilote, ses capacités et une image représentant sa carte.
- Visualiser les manœuvres des pilotes avec la grille d'affichage organisée en fonction des vitesses et types de manœuvres.
- S'authentifier rapidement grâce à un système minimaliste de connexion.
- Gérer leurs escadrons : pour l'instant juste l'affichage des escadrons et possibilité d'en créer pour les utilisateurs connectés.

Développement et Implémentation

La partie la plus importante de l'application dans son état actuel (et celle qui m'a demandé le plus de temps) est la traduction des données Json en Base de Données MySql. Vous trouverez le contrôleur ImportController qui fait appel au service ImportService qui de sont côté parse les fichiers de données Json pour en faire des tableaux associatifs et appeler doctrine pour insérer les données en Base.

L'architecture de la base de données est organisée autour de plusieurs entités interconnectées : Faction, Ship, Pilot, Squadron, ainsi que des entités auxiliaires telles que Maneuver, Stat, Action (et Upgrade). Chaque Squadron appartient à un utilisateur unique et peut contenir plusieurs Pilots, chacun étant associé à un Ship spécifique. Les manœuvres sont définies par une table dédiée contenant la vitesse, la direction et la difficulté, facilitant leur affichage sous forme de grille. L'utilisation de Doctrine ORM permet une gestion efficace des relations et des performances optimisées grâce au préchargement des données statiques.



Concernant la partie de la BDD du côté des Upgrades, je n'ai pas importé les données par manque de temps. Je me suis plutôt concentré sur l'affichage des données des pilotes et des vaisseaux.

Conclusion et Perspectives

Je ne suis pas tout à fait satisfait de ce que j'ai pu implémenter dans le projet pour le rendu, mais j'ai pu au moins appliquer une base solide pour la suite. Je compte poursuivre son développement afin d'avoir une application utilisable pour moi et mes frères, et pourquoi pas plus tard la publier à la communauté qui continue le jeu.

Pour ce qui est des axes d'amélioration:

- Ajouter les données des Upgrades pour l'affichage
- Ajouter les données des autres faction (ici j'en ai mis 3 sur les 7 disponibles)
- Étoffer les possibilités pour les utilisateurs connecté (système de score pour les escadrons créés par exemple)
- Finir les fonctionnalités de la page Squad Builder, c'est-à-dire permettre l'ajout de vaisseaux / pilotes dans un escadron et pouvoir équiper les pilotes d'upgrades. Et mettre à jour l'affichage des escadrons avec les vaisseaux ajoutés.

En outre, je me suis plutôt amusé durant ce projet en découvrant Symfony malgré la quantité de travail qui nous est tombé dessus ces dernières semaines.