



Projet Pokémon

Document de Spécification : Fonctionnelles



Mohamed CHINE
Nicolas CLEMENT
Louis DUBOIS
Aurélia HUISSE-DALBOUSSIÈRE
Sirine NADIFI

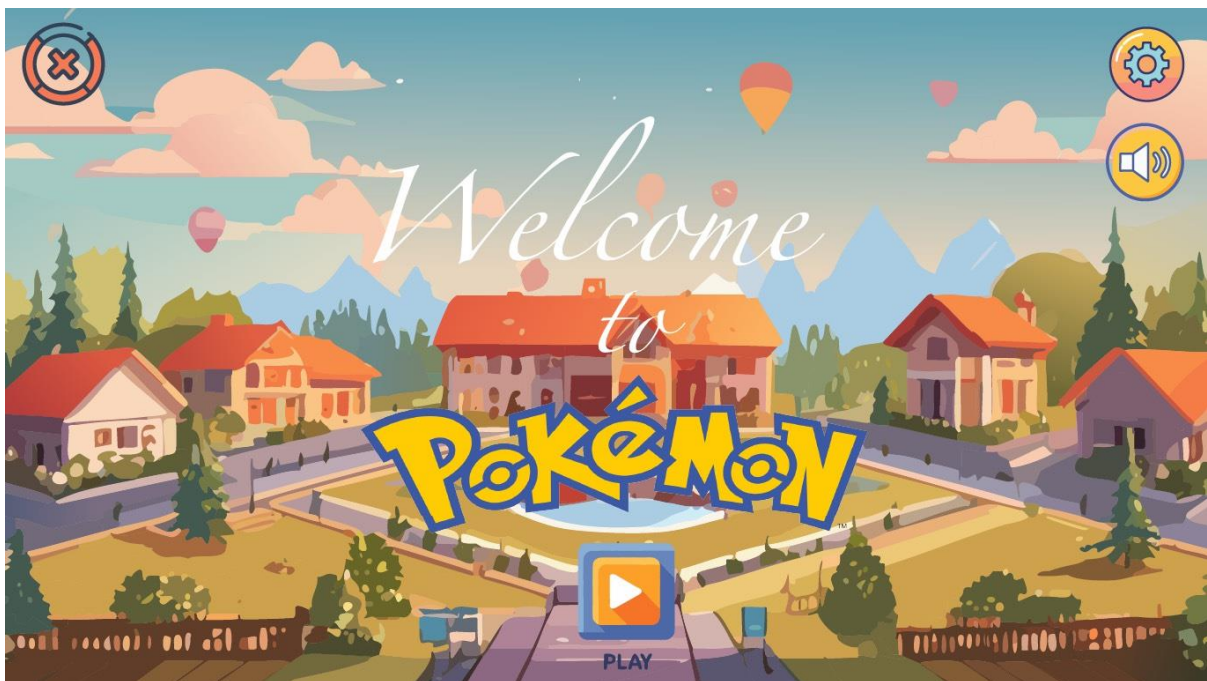
Table des Matières

Description de l'application	2
Fonctionnement de l'application.....	3
Fonctionnalités obligatoires	3
Fonctionnalités utiles	3
Fonctionnalités optionnelles	3
Modélisation.....	5

Description de l'application

Notre projet est un jeu vidéo dans le style pokémon. Pokémon est une franchise japonaise créée par Satoshi Tajiri en 1995 qui présente sous la forme d'un jeu vidéo un univers fantastique dans lequel le joueur incarne un dresseur qui doit apprivoiser des créatures sous le nom de Pokémons. Dans notre jeu, nous utiliserons donc l'univers mais pas l'histoire.

Dans notre jeu, nous incarnerons un jeune citoyen qui un jour, rencontre un pokémon mal en point dans la nature. Une fois le pokémon sauvé, le but va être de l'entraîner, pour cela, il nous faudra nous balader sur la carte à la recherche d'autres pokémons pour faire apprendre leurs capacités à notre pokémon et surtout récupérer des objets. Ce système pourra servir de tutoriel pour les nouveaux joueurs. Pour réaliser ce jeu, nous allons utiliser une interface graphique en deux dimensions comme le premier jeu pokémon paru en 1996 sur GameBoy. Ci-dessous, un exemple de l'écran de démarrage du jeu :



Fonctionnement de l'application

Notre application est une application modulaire qui utilise la programmation orientée objet pour fonctionner. Dans cette section, nous allons présenter les fonctionnalités théoriques de notre jeu.

Fonctionnalités obligatoires

Dans cette sous-section, nous allons présenter les fonctionnalités indispensables pour le bon fonctionnement du jeu.

Notre jeu doit contenir obligatoirement :

- Un écran d'accueil dans lequel nous pourrions accéder à une section des paramètres pour modifier la taille d'affichage de l'écran.
- Un Tutoriel au premier lancement du jeu
- Une carte dans laquelle on peut contrôler le joueur.
- Des pokémons que l'on peut capturer.
- Des maisons dans lesquels nous pourrions entrer.
- Un menu échap qui nous permet de quitter le jeu.
- Un système d'inventaire dans lequel on peut stocker les pokémons

Fonctionnalités utiles

Les fonctionnalités que nous aimerions implémenter sont :

- Créer un système de sauvegarde pour sauvegarder et charger une sauvegarde en cours.
- Un système de changement des touches
- Créer un système de combat entre pokémons
- Améliorer la gestion des collisions et les hitboxes entre le joueur et les pokémons
- Ajouter des animations à chaque changement de scènes

La liste de ces fonctionnalités est une liste qui aurait pu être utile et qui pourrait améliorer les projets mais qui n'ont pas forcément toutes été mises en place.

Fonctionnalités optionnelles

Dans cette sous-section, nous allons aborder les fonctionnalités totalement optionnelles et surtout fictive qui aurait pu être mises en place avec une peu plus de temps. On pourrait donc imaginer avoir :

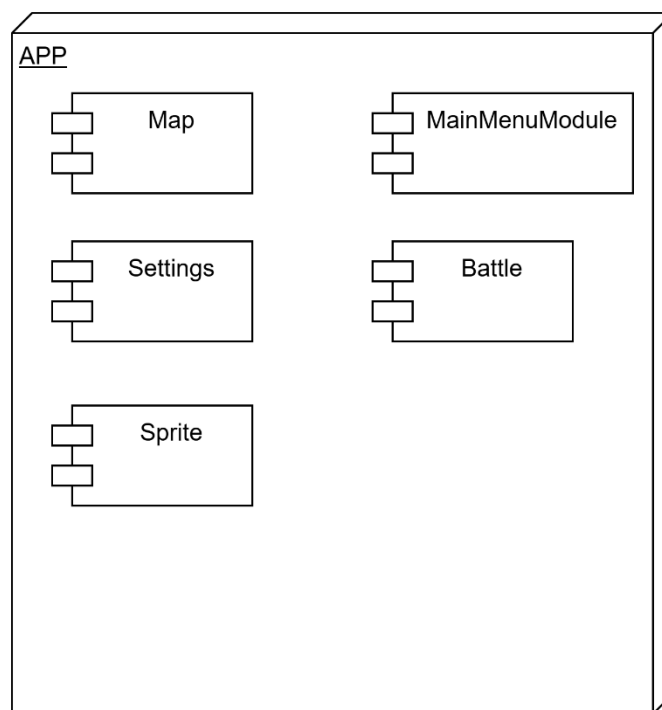
- Un traducteur qui nous permettrait d'implémenter plusieurs langues pour l'interface et pour les dialogues. Cela aurait inclus plus de personnes sur le jeu.
- Une intelligence artificielle pour les pokémons qui aurait pu réagir en fonction des déplacements du joueur ou alors avec le décor.
- Un multijoueur dans lequel plusieurs joueurs s'affronter et que celui qui est le plus rapide dans la capture des pokémons remporte la partie
- Une API qui nous permettrait de créer une application mobile et de lier les données entre le logiciel et le téléphone. Cela permettrait à un joueur qui jouer sur ordinateur de récupérer son compte sur téléphone.
- Imaginer plusieurs mini-jeux en plus de la capture des pokémons.
- Inclure un système de quête qui permettrait de donner une histoire au jeu : cela Permet d'avoir des interactions et une ligne directrice lorsqu'on joue au jeu en donnant des récompenses tel que des objets et de l'expérience pour le joueur et le pokémon. Nous pourrions accepter ou refuser des quêtes lorsque nous parlons avec les PNJ (personnage non jouable) et pourront se manifester par des combats ou par des échanges avec plusieurs PNJ.
- Inclure d'autre personnages en plus des pokémons
- Inclure différentes statistiques aux pokémons : Les pokémons auraient plusieurs statistiques qui influeront sur les combats tel que la force pour la puissance d'attaque, les PV (Point de vue) pour la défense du pokémon et la vitesse pour la vitesse d'attaque des pokémons.
- Plusieurs accessoires pour le joueur comme des pokéballs permettant de capturer les pokémons ou alors des accessoires de soins.

Ces fonctionnalités sont des fonctionnalités optionnelles, on pourrait donc en imaginer des centaines qui pourrait d'ailleurs être implémenter ou non dans des versions futures du logiciel. C'est en quelques sorte notre carte de route.

La section suivante abordera la modélisation, c'est-à-dire comment notre projet a été pensé en termes de code.

Modélisation

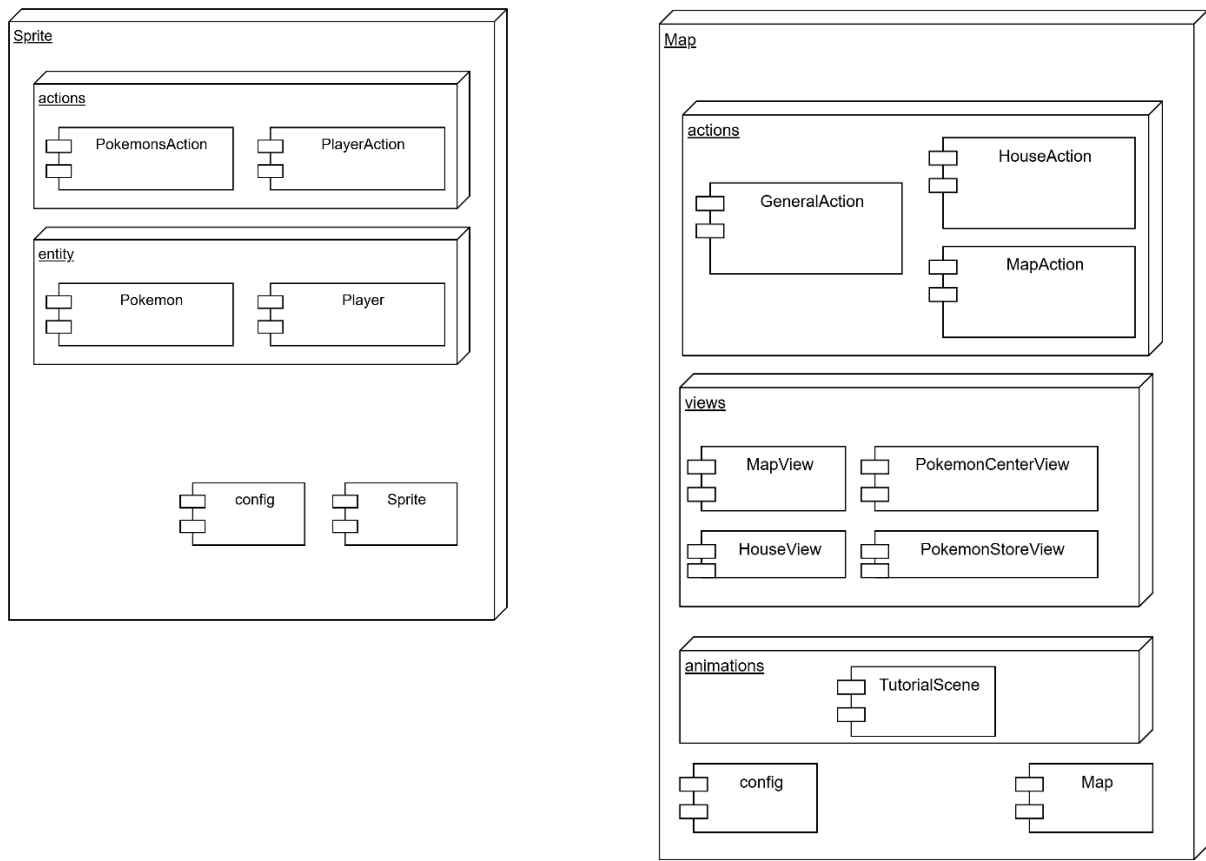
Notre projet a été pensé avec l'aide du pattern Model View Controller et un système de module. Nous avons donc décidé, étant donné l'envergure de notre projet, de présenter une version simplifiée du diagramme UML, puis un exemple de module détaillé car chaque module est basé sur le même schéma. De plus, la librairie ne sera abordée car selon moi, les fonctionnalités ajoutées sont des fonctionnalités qui devraient exister de base dans le moteur de jeu pygame. Voici donc ci-dessous, le diagramme simplifié de notre application :



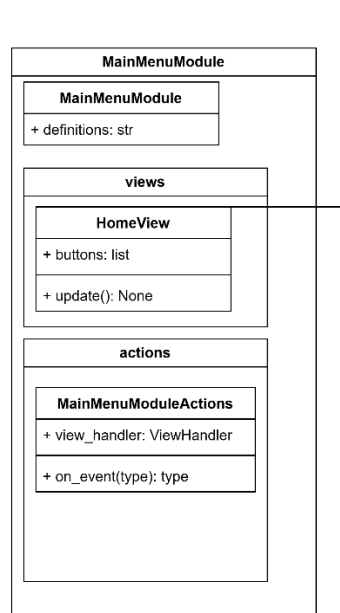
Chaque module à l'intérieur de l'application contient globalement la même chose :

- Un package views (dans lequel on intègre les vues)
- Un package entity (dans lequel on gère les modèles s'il y en a)
- Un package actions pour les controllers
- Un package animations pour les cinématiques
- Un fichier config pour la configuration et un fichier qui initialise le module

Cela nous donne le résultat suivant en pratique :



On obtient donc par exemple pour le module de l'écran de chargement le diagramme UML suivant :



Nous avons donc raisonné de la même manière pour chaque module du projet.