# Architecture et Modèle logique

Groupe 11 : YILDIZ Marcel VUKASINOVIC Stefan NERONDAT Clément

### <u>Interprétation du sujet :</u>

Il s'agit d'implémenter un jeu en ligne multijoueur dans lequel les joueurs pourront interagir à la fois avec des joueurs et avec des objets que l'on pourra éventuellement visualiser en réalité augmentée. La principale difficulté se trouve dans l'utilisation de la géolocalisation, a défaut nous utiliserons une carte de jeu interactive. Une autre difficulté que l'on a mis en évidence est l'apparition d'objets.

Pour ce qui est des enjeux, avant même de commencer à coder, il faut prévoir les éventuelles extensions futures afin qu'elles soient le plus simple possible a implémenter. C'est pourquoi il faut soigner l'architecture afin d'avoir un code malléable dans le temps.

### **Concepts**:

Afin de structurer notre travail, de faciliter le travail en groupe et s'agissant d'un jeu interactif, nous avons fait le choix d'utiliser le patron d'architecture logicielle MVC (model-view-controller).

Le modèle de données (model) sera constitué de Joueurs, de Pokédex comportant des Pokémons, d'Inventaires comportant des Objets et de la Carte de jeu.

L'interface utilisateur (vue) sera l'interface graphique du jeu, la question de la réalité augmentée se posera ici.

La gestion des évènements (controller) sera faite par rapport aux modifications exhaustives possibles (changement de pseudo d'un Joueur, capture d'un Pokémon par un joueur, changement de direction lors du déplacement d'un Joueur, ...)

# Description de l'architecture :

#### class Joueur

pseudo : String argent : Integer positionX : Integer positionY : Integer caseActuelle : Case inventaire : Inventaire pokédex : Pokedex

monter() descendre() gauche() droite()

lancerPokeball(Pokemon p) capturerPokemon(Pokemon p)

### class Inventaire

listeObjets : List[Objet]

retirer(Objet o) ajouter(Objet o)

## abstract class Objet

type: String

positionX : Integer positionY : Integer

# class Pokeball extends Objet

type : String prix : Integer

### class Pokemon

difficulte: Integer nom: String type: String niveau: Integer experience: Integer pointsDeVieMax: Integer pointDeVie: Integer

progresserCombat(Pokemon ennemi)
monterDeNiveau()

evoluer()

### class Pokedex

listePokemons : List[Pokemon]

## class Case

contientPokemon : Boolean contientJoueur : Boolean

## class Map

carteDeJeu : Array[Array[Case]]

Figure 1 : Diagramme des classes

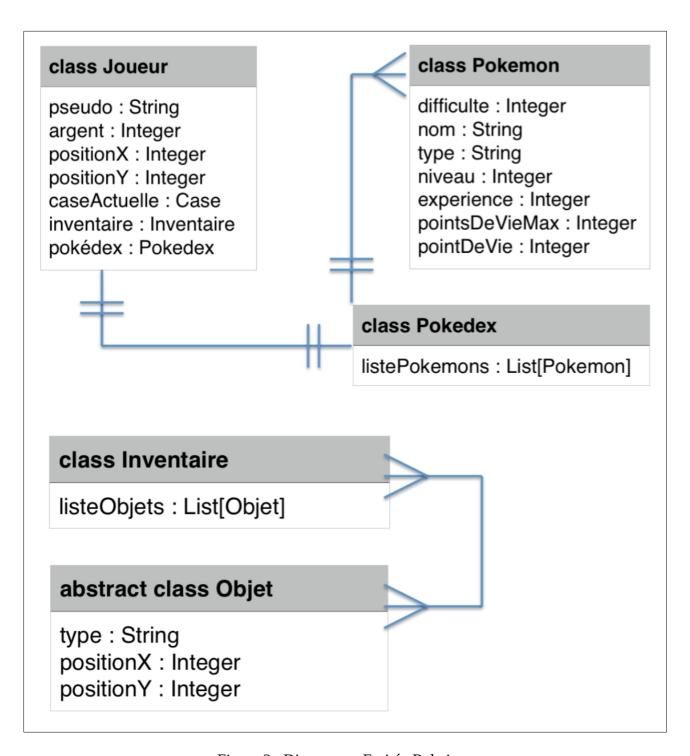


Figure 2 : Diagramme Entités-Relations

# Extensions envisagées :

Implémenter la réalité augmentée, géolocaliser les joueurs, faire déplacer le Joueur sur la carte en fonction de sa géolocalisation, implémenter les interactions possibles entre joueurs, donner un visuel aux différentes attaques de Pokémons, ....