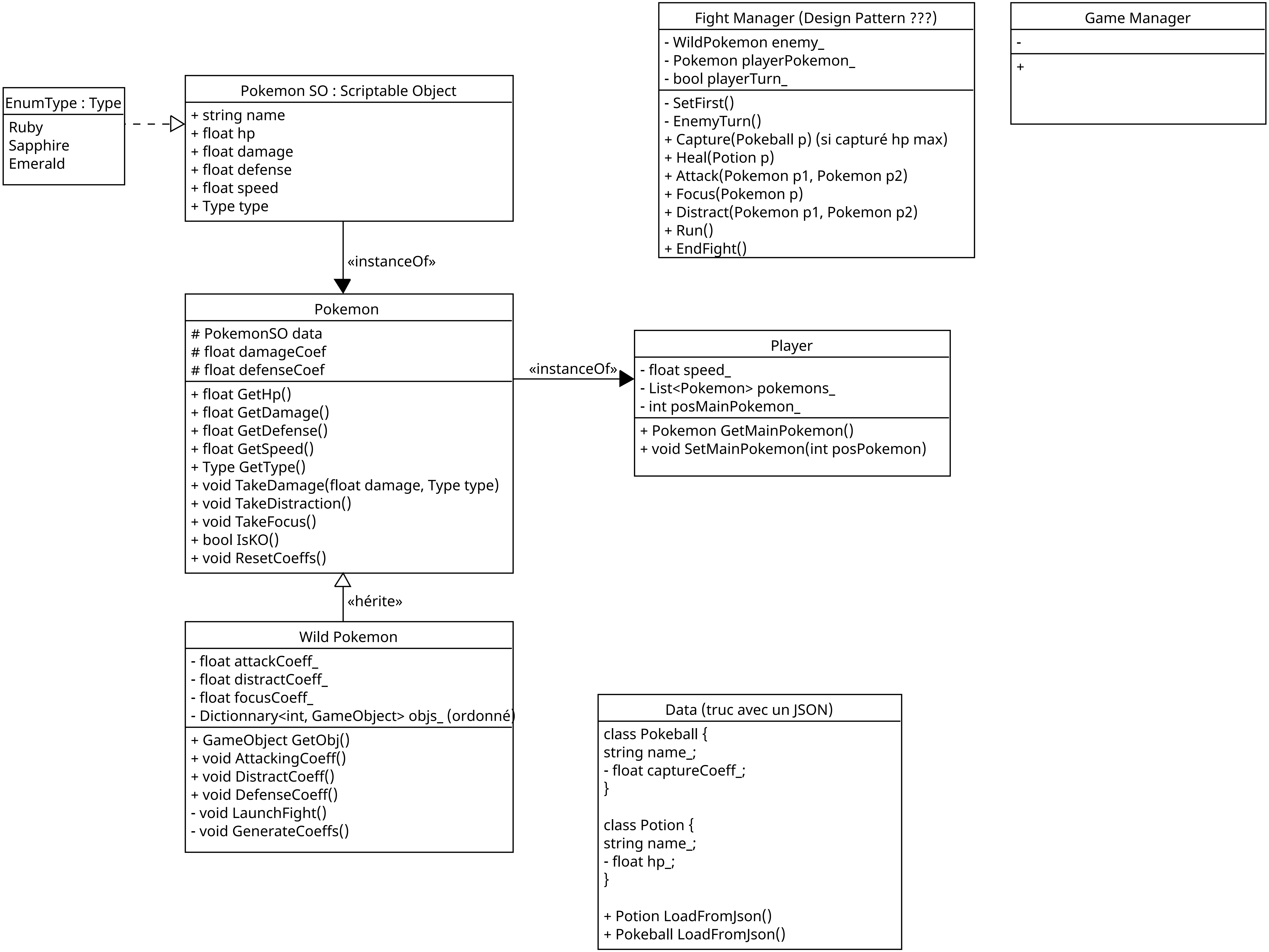

EnumType : Type

Ruby
Sapphire
Emerald

Pokemon SO : Scriptable Object

+ string name
+ float hp
+ float damage
+ float defense
+ float speed
+ Type type

Pokemon

# PokemonSO data
# float damageCoef
# float defenseCoef

+ float GetHp()
+ float GetDamage()
+ float GetDefense()
+ float GetSpeed()
+ Type GetType()
+ void TakeDamage(float damage, Type type)
+ void TakeDistraction()
+ void TakeFocus()
+ bool IsKO()
+ void ResetCoeffs()

Wild Pokemon

- float attackCoeff_
- float distractCoeff_
- float focusCoeff_
- Dictionnary<int, GameObject> objs_ (ordonné)

+ GameObject GetObj()
+ void AttackingCoeff()
+ void DistractCoeff()
+ void DefenseCoeff()
- void LaunchFight()
- void GenerateCoeffs()

Fight Manager (Design Pattern ???)

- WildPokemon enemy_
- Pokemon playerPokemon_
- bool playerTurn_

- SetFirst()
- EnemyTurn()
+ Capture(Pokeball p) (si capturé hp max)
+ Heal(Potion p)
+ Attack(Pokemon p1, Pokemon p2)
+ Focus(Pokemon p)
+ Distract(Pokemon p1, Pokemon p2)
+ Run()
+ EndFight()

Game Manager

-

+

Player

- float speed_
- List<Pokemon> pokemons_
- int posMainPokemon_

+ Pokemon GetMainPokemon()
+ void SetMainPokemon(int posPokemon)

Data (truc avec un JSON)

```
class Pokeball {
string name_;
- float captureCoeff_;
}

class Potion {
string name_;
- float hp_;
}

+ Potion LoadFromJson()
+ Pokeball LoadFromJson()
```