WORLD, TRAINERS AND INTERACTIONS LOGIC

POKEMONS AND BATTLE LOGIC

Sign
- text: String

+ onInteract() override

<<Interface>>
IInteractable

+ onInteract()

<>
Trainer
- name: String
team: List<Pokemon>

+ onInteract() override
+ chooseAction(): Move*

NPC
- dialogue: String

+ talk()
+ onInteract() override
+ chooseAction() override

Player
- inventory: Inventory
- money: Int

+ useItem()
+ onInteract() override
+ chooseAction() override

Inventory
- items: Map<Item, Int>

+ addItem(item: Item)
+ useItem(item: Item, target: Pokemon)

<>
Item
- name: String
- price: Int

+ apply(target: Pokemon)*

Potion
- healAmount

+ apply(target: Pokemon) override

Pokeball
- catchRate()

+ apply(target: Pokemon) override

<<Enumeration>>
PokemonType
- NORMAL
- FIRE
- WATER
- GRASS

+ typeEffectiveness(targetType: PokemonType): Double

<<object>>
PokemonRegistry
+ charmander(level: Int) Pokemon
+ bulbasaur(level: Int) Pokemon
+ squirtle(level: Int) Pokemon
- generateRandomIV() Stats

Pokemon
+ name: String
+ type: PokemonType {readOnly}
+ baseStats: Stats {readOnly}
+ IVStats: Stats {readOnly}
+ stats: Stats
- exp: Int
- level: Int
+ moves: List<Move> {readOnly}
- _moves: MutableList<Move>
+ learnableMoves: List<Pair<Int, () -> Move>> {readOnly}
+ gainExp()
+ recalculateStats()
+ levelUp()
- learnMove(move: Move)
+ evolution()

Stats
- maxHp: Int
- attack: Int
- defence: Int
- speed: Int
- currentHp: Int

+ takeDamage(damage: Int)
+ heal(healAmount: Int)
+ modifyStat(statName: String, amount: Int)
+ setStat(statName: String, amount: Int)
+ isAlive() Boolean

Move
+ name: String {readOnly}
+ type: PokemonType {readOnly}
+ maxPp: Int {readOnly}
+ accuracy: Int {readOnly}
- pp: Int
- effects: List<IMoveEffect> {readOnly}

+ execute(attacker: Pokemon, target: Pokemon, type: PokemonType)*

<<object>>
MoveRegistry
+ tackle() Move
+ ember() Move
+ growl() Move
+ tailWhip() Move
+ recover() Move

<<interface>>
IMoveEffect

+ apply(attacker: Pokemon, target: Pokemon, moveType: PokemonType)*

DamageEffect
- power: Int

+ apply(attacker: Pokemon, target: Pokemon, moveType: PokemonType)

HealEffect
- healAmount: Int

+ apply(attacker: Pokemon, target: Pokemon, moveType: PokemonType)

RecoilEffect
- power: Int
- recoilPercent: Int

+ apply(attacker: Pokemon, target: Pokemon, moveType: PokemonType)

StatChangeEffect
- statName: String
- amount: Int
- onTarget: Boolean

+ apply(attacker: Pokemon, target: Pokemon, moveType: PokemonType)