Pokemon

Un jour je serai le meilleur dresseur,

je me battrai sans repit !

Documentation Technique du Projet

Redigee par l'equipe de developpement

Notre equipe developpe un jeu Pokemon complet avec interface graphique. Ce document presente notre architecture technique, les 18 types de Pokemon, le systeme de combat, et notre planning de developpement.

Ce que nous developpons :

- Systeme complet avec les 18 types officiels de Pokemon
- Tableau d'efficacite des types 18x18 (Gen 6+)
- Base de donnees de Pokemon issue du Pokedex National
- Systeme de combat tour par tour avec calcul de degats
- Pokedex persistant (JSON) avec gestion des doublons
- Interface graphique soignee (Tkinter/Pygame)
- Architecture POO : heritage, encapsulation, polymorphisme

Table des matieres

1. Le projet et notre equipe
2. Regles du jeu
3. Les 18 types de Pokemon
4. Tableau d'efficacite des types (18x18)
5. Systeme de combat detaille
6. Systeme d'evolution
7. Systeme de Pokedex
8. Liste des Pokemon (Pokedex National)
9. Interface utilisateur
10. Feedbacks visuels
11. Architecture technique
12. Diagramme de classes
13. Principes de POO
14. Gestion des erreurs
15. Perimetre et fonctionnalites (MVP)
16. Criteres d'acceptation
17. Planning et repartition
18. Direction artistique
19. Rendu et competences
20. Ressources

1. Le projet et notre equipe

Contexte

Nous sommes une equipe de developpeurs travaillant dans le cadre d'un projet a La Plateforme. Notre objectif est de developper un jeu Pokemon complet avec interface graphique, integrant les 18 types officiels de Pokemon.

Ce projet nous permet de mettre en pratique les principes de la programmation orientee objet (POO) en Python : heritage, encapsulation, polymorphisme et abstraction.

Notre vision

> *"Un jour je serai le meilleur dresseur, je me battrai sans repit !"*

Ce que nous implementons :

- Systeme de combat avec les 18 types officiels et leur tableau d'efficacite
- Base de donnees de 151 Pokemon (1ere generation) issue du Pokedex National
- Pokedex persistant en JSON avec gestion intelligente des doublons
- Interface graphique soignee avec feedbacks visuels
- Architecture modulaire respectant les principes SOLID

Nos choix techniques

- Langage : Python 3.10+
- GUI : Tkinter (inclus dans Python) ou Pygame
- Persistance : Fichiers JSON (pokemon.json, pokedex.json)
- Architecture : POO avec heritage par type
- Versionning : Git + GitHub
- Source des donnees : Pokepedia (Pokedex National)

2. Regles du jeu

Regles du combat

Regle	Description	Detail
Tour par tour	Les Pokemon attaquent chacun leur tour	Le joueur attaque en premier
Points de vie	Chaque Pokemon a des PV	Quand PV = 0, le Pokemon est KO
Defense	La defense reduit les degats subis	Degats = Attaque * Mult. - Defense
Attaque ratee	Un Pokemon peut louper son attaque	Probabilite aleatoire d'echec
18 types	Le type modifie les degats	Voir tableau complet des types
Victoire	Le dernier Pokemon debout gagne	Message affiche le vainqueur

Formule de calcul des degats

Nous implementons la formule suivante :

```
degats_bruts = attaque_attaquant * multiplicateur_type
degats_finaux = max(0, degats_bruts - defense_defenseur)
pv_defenseur = max(0, pv_defenseur - degats_finaux)
```

> *Exemple : Dracaufeu (Feu, ATK=84) attaque Florizarre (Plante/Poison). Feu vs Plante = x2, donc degats_bruts = 84 * 2 = 168. Si Florizarre a 83 en DEF : degats_finaux = 168 - 83 = 85 PV retires.*

3. Les 18 types de Pokemon

Contrairement a la version simplifiee a 4 types, nous implementons les 18 types officiels de l'univers Pokemon (depuis la 6eme generation). Chaque type a ses forces et faiblesses, creant un systeme strategique profond.

Les 18 types officiels

Normal	Feu	Eau
Plante	Electrik	Glace
Combat	Poison	Sol
Vol	Psy	Insecte
Roche	Spectre	Dragon
Tenebres	Acier	Fee

Correspondances type / couleur (implementation)

Type	RGB	Hex
Normal	(168, 168, 120)	#A8A878
Feu	(240, 128, 48)	#F08030
Eau	(104, 144, 240)	#6890F0
Plante	(120, 200, 80)	#78C850
Electrik	(248, 208, 48)	#F8D030
Glace	(152, 216, 216)	#98D8D8
Combat	(192, 48, 40)	#C03028
Poison	(160, 64, 160)	#A040A0
Sol	(224, 192, 104)	#E0C068

Type	RGB	Hex
Vol	(168, 144, 240)	#A890F0
Psy	(248, 88, 136)	#F85888
Insecte	(168, 184, 32)	#A8B820
Roche	(184, 160, 56)	#B8A038
Spectre	(112, 88, 152)	#705898
Dragon	(112, 56, 248)	#7038F8
Tenebres	(112, 88, 72)	#705848
Acier	(184, 184, 208)	#B8B8D0
Fee	(238, 153, 172)	#EE99AC

4. Tableau d'efficacite des types (18x18)

Ce tableau est au coeur de notre systeme de combat. Ligne = type attaquant, Colonne = type defenseur. Vert = super efficace (x2), Rouge = peu efficace (x0.5), Noir = aucun effet (x0), Blanc = normal (x1).

ATK \ DEF	NOR	FEU	EAU	PLA	ELE	GLA	COM	POI	SOL	VOL	PSY	INS	ROC	SPE	DRA	TEN	ACI	FEE
Normal	1	1	1	1	1	1	1	1	1	1	1	1	0.5	0	1	1	0.5	1
Feu	1	0.5	0.5	2	1	2	1	1	1	1	1	2	0.5	1	0.5	1	2	1
Eau	1	2	0.5	0.5	1	1	1	1	2	1	1	1	2	1	0.5	1	1	1
Plante	1	0.5	2	0.5	1	1	1	0.5	2	0.5	1	0.5	2	1	0.5	1	0.5	1
Electrik	1	1	2	0.5	0.5	1	1	1	0	2	1	1	1	1	0.5	1	1	1
Glace	1	0.5	0.5	2	1	0.5	1	1	2	2	1	1	1	1	2	1	0.5	1
Combat	2	1	1	1	1	2	1	0.5	1	0.5	0.5	0.5	2	0	1	2	2	0.5
Poison	1	1	1	2	1	1	1	0.5	0.5	1	1	1	0.5	0.5	1	1	0	2
Sol	1	2	1	0.5	2	1	1	2	1	0	1	0.5	2	1	1	1	2	1
Vol	1	1	1	2	0.5	1	2	1	1	1	1	2	0.5	1	1	1	0.5	1
Psy	1	1	1	1	1	1	2	2	1	1	0.5	1	1	1	1	0	0.5	1
Insecte	1	0.5	1	2	1	1	0.5	0.5	1	0.5	2	1	1	0.5	1	2	0.5	0.5
Roche	1	2	1	1	1	2	0.5	1	0.5	2	1	2	1	1	1	1	0.5	1
Spectre	0	1	1	1	1	1	1	1	1	1	2	1	1	2	1	0.5	1	1
Dragon	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	0.5	0
Tenebres	1	1	1	1	1	1	0.5	1	1	1	2	1	1	2	1	0.5	0.5	0.5
Acier	1	0.5	0.5	1	0.5	2	1	1	1	1	1	1	2	1	1	1	0.5	2

Fee	1	0.5	1	1	1	1	2	0.5	1	1	1	1	1	1	2	2	0.5	1

Legende : Vert = x2 (super efficace) | Rouge = x0.5 (peu efficace) | Noir = x0 (aucun effet) | Blanc = x1 (normal)

5. Systeme de combat detaille

Notre implementation du calcul de degats

Voici comment nous implementons le calcul des degats :

- 1. Recuperer la puissance d'attaque de l'attaquant
- 2. Determiner le type de l'attaquant et le(s) type(s) du defenseur
- 3. Consulter le tableau 18x18 pour obtenir le multiplicateur
- 4. En cas de double type du defenseur : multiplier les deux multiplicateurs
- 5. Calculer degats_bruts = attaque * multiplicateur
- 6. Calculer degats_finaux = max(0, degats_bruts - defense)
- 7. Appliquer : pv_defenseur = max(0, pv_defenseur - degats_finaux)

Implementation du double type

Si le defenseur possede deux types, nous multiplions les multiplicateurs. Exemple : Leviator est Eau/Vol. Une attaque Electrik fait x2 contre Eau ET x2 contre Vol, donc x4 au total.

```
def get_multiplicateur(self, type_attaquant, types_defenseur):
    mult = 1.0
    idx_att = TYPE_NAMES.index(type_attaquant)
    for type_def in types_defenseur:
        idx_def = TYPE_NAMES.index(type_def)
        mult *= TYPE_CHART[idx_att][idx_def]
    return mult
```

Attaque ratee

Nous implementons un systeme de precision : a chaque tour, il existe une probabilite (configurable, par defaut 10%) qu'un Pokemon rate son attaque. Si l'attaque est ratee, aucun degat n'est inflige.

Conditions de victoire

- Un Pokemon est KO quand ses PV atteignent 0
- Le combat se termine immediatement
- Le nom du Pokemon vainqueur est affiche
- Le Pokemon adverse est enregistre dans le Pokedex du joueur

6. Systeme d'evolution

Notre implementation

Nous implementons l'evolution des Pokemon selon les chaines officielles :

- L'evolution modifie les statistiques du Pokemon (PV, attaque, defense)
- Le Pokemon change de nom apres evolution (ex: Salameche -> Reptincel)
- Le niveau conditionne l'evolution
- L'evolution peut modifier ou ajouter un type (ex: Salameche->Dracaufeu gagne Vol)

Exemples de chaines d'evolution

Stade 1	Type	Stade 2	Type	Stade 3	Type
Bulbizarre	Plante/Poison	Herbizarre	Plante/Poison	Florizarre	Plante/Poison
Salameche	Feu	Reptincel	Feu	Dracaufeu	Feu/Vol
Carapuce	Eau	Carabaffe	Eau	Tortank	Eau
Pikachu	Electrik	Raichu	Electrik	-	-
Abra	Psy	Kadabra	Psy	Alakazam	Psy
Minidraco	Dragon	Draco	Dragon	Dracolosse	Dragon/Vol

Impact sur les stats

Statistique	Effet de l'evolution
Points de vie	Augmentation significative (+20 a +40%)
Attaque	Augmentation (+15 a +30%)
Defense	Augmentation (+15 a +30%)
Type	Peut changer ou s'ajouter

7. Systeme de Pokedex

Notre implementation

Le Pokedex est un composant cle de notre application. Il enregistre automatiquement les Pokemon rencontres lors des combats et persiste les donnees en JSON.

Donnees enregistrees par Pokemon

- Numero du Pokedex National
- Nom du Pokemon
- Type(s)
- Points de vie
- Puissance d'attaque
- Defense

Gestion des doublons

Avant d'ajouter un Pokemon au Pokedex, nous verifions par son numero si il est deja enregistre. Pas de doublon possible.

Structure du fichier pokedex.json

```
{
    "pokedex": [
        {
            "numero": "025",
            "nom": "Pikachu",
            "types": ["Electrik"],
            "pv": 35,
            "attaque": 55,
            "defense": 40
        },
        {
            "numero": "006",
            "nom": "Dracaufeu",
            "types": ["Feu", "Vol"],
            "pv": 78,
            "attaque": 84,
            "defense": 78
        }
    ],
    "nombre_total": 2
}
```

Structure du fichier pokemon.json

```
{
    "pokemons": [
        {
            "numero": "007",
            "nom": "Carapuce",
            "types": ["Eau"],
            "pv": 44,
            "niveau": 5,
            "attaque": 48,
            "defense": 65,
            "evolution": "Carabaffe"
        },
        {
            "numero": "004",
            "nom": "Salameche",
            "types": ["Feu"],
            "pv": 39,
            "niveau": 5,
            "attaque": 52,
            "defense": 43,
            "evolution": "Reptincel"
        }
    ]
}
```

8. Liste des Pokemon (Pokedex National - Gen 1)

Source : Pokepedia (pokepedia.fr). Nous integrons les 151 Pokemon de la 1ere generation dans notre fichier pokemon.json. Voici la liste complete :

#	Nom	Type 1	Type 2
001	Bulbizarre	Plante	Poison
002	Herbizarre	Plante	Poison
003	Florizarre	Plante	Poison
004	Salameche	Feu	-
005	Reptincel	Feu	-
006	Dracaufeu	Feu	Vol
007	Carapuce	Eau	-
008	Carabaffe	Eau	-
009	Tortank	Eau	-
010	Chenipan	Insecte	-
011	Chrysacier	Insecte	-
012	Papilusion	Insecte	Vol
013	Aspicot	Insecte	Poison
014	Coconfort	Insecte	Poison
015	Dardargnan	Insecte	Poison
016	Roucool	Normal	Vol
017	Roucoups	Normal	Vol
018	Roucarnage	Normal	Vol
019	Rattata	Normal	-
020	Rattatac	Normal	-
021	Piafabec	Normal	Vol
022	Rapasdepic	Normal	Vol
023	Abo	Poison	-
024	Arbok	Poison	-

025	Pikachu	**Electrik**	-

026	Raichu	**Electrik**	-

#	Nom	Type 1	Type 2
027	Sabelette	**Sol**	-
028	Sablaireau	**Sol**	-
029	Nidoran F	**Poison**	-
030	Nidorina	**Poison**	-
031	Nidoqueen	**Poison**	**Sol**
032	Nidoran M	**Poison**	-
033	Nidorino	**Poison**	-
034	Nidoking	**Poison**	**Sol**
035	Melofee	**Fee**	-
036	Melodelfe	**Fee**	-
037	Goupix	**Feu**	-
038	Feunard	**Feu**	-
039	Rondoudou	**Normal**	**Fee**
040	Grodoudou	**Normal**	**Fee**
041	Nosferapti	**Poison**	**Vol**
042	Nosferalto	**Poison**	**Vol**
043	Mystherbe	**Plante**	**Poison**
044	Ortide	**Plante**	**Poison**
045	Rafflesia	**Plante**	**Poison**
046	Paras	**Insecte**	**Plante**
047	Parasect	**Insecte**	**Plante**
048	Mimitoss	**Insecte**	**Poison**
049	Aeromite	**Insecte**	**Poison**
050	Taupiqueur	**Sol**	-

051	Triopikeur	**Sol**	-

052	Miaouss	**Normal**	-

#	Nom
053	Persian
054	Psykokwak
055	Akwakwak
056	Ferosinge
057	Colossinge
058	Caninos
059	Arcanin
060	Ptitard
061	Tetarte
062	Tartard
063	Abra
064	Kadabra
065	Alakazam
066	Machoc
067	Machopeur
068	Mackogneur
069	Chetiflor
070	Boustiflor
071	Empiflor
072	Tentacool
073	Tentacruel
074	Racaillou
075	Gravalanch
076	Grolem

077	Ponyta

078	Galopa

#	Nom	Type 1	Type 2
079	Ramoloss	Eau	Psy
080	Flagadoss	Eau	Psy
081	Magneti	Electrik	Acier
082	Magneton	Electrik	Acier
083	Canarticho	Normal	Vol
084	Doduo	Normal	Vol
085	Dodrio	Normal	Vol
086	Otaria	Eau	-
087	Lamantine	Eau	Glace
088	Tadmorv	Poison	-
089	Grotadmorv	Poison	-
090	Kokiyas	Eau	-
091	Crustabri	Eau	Glace
092	Fantominus	Spectre	Poison
093	Spectrum	Spectre	Poison
094	Ectoplasma	Spectre	Poison
095	Onix	Roche	Sol
096	Soporifik	Psy	-
097	Hypnomade	Psy	-
098	Krabby	Eau	-
099	Krabboss	Eau	-
100	Voltorbe	Electrik	-
101	Electrode	Electrik	-
102	Noeunoeuf	Plante	Psy
103	Noadkoko	Plante	Psy
104	Osselait	Sol	-

#	Nom	Type 1	Type 2
105	Ossatueur	Sol	-
106	Kicklee	Combat	-
107	Tygnon	Combat	-
108	Excelangue	Normal	-
109	Smogo	Poison	-
110	Smogogo	Poison	-
111	Rhinocorne	Sol	Roche
112	Rhinoferos	Sol	Roche
113	Leveinard	Normal	-
114	Saquedeneu	Plante	-
115	Kangourex	Normal	-
116	Hypotrempe	Eau	-
117	Hypocean	Eau	-
118	Poissirene	Eau	-
119	Poissoroy	Eau	-
120	Stari	Eau	-
121	Staross	Eau	Psy
122	M. Mime	Psy	Fee
123	Insecateur	Insecte	Vol
124	Lippoutou	Glace	Psy
125	Elektek	Electrik	-
126	Magmar	Feu	-
127	Scarabrute	Insecte	-
128	Tauros	Normal	-
129	Magicarpe	Eau	-
130	Leviator	Eau	Vol

#	Nom
131	Lokhlass
132	Metamorph
133	Evoli
134	Aquali
135	Voltali
136	Pyroli
137	Porygon
138	Amonita
139	Amonistar
140	Kabuto
141	Kabutops
142	Ptera
143	Ronflex
144	Artikodin
145	Electhor
146	Sulfura
147	Minidraco
148	Draco
149	Dracolosse
150	Mewtwo
151	Mew

9. Interface utilisateur

Menu principal

Nous implementons un menu avec les options suivantes :

Option	Description
Lancer une partie	Demarre un combat Pokemon
Ajouter un Pokemon	Ajoute un Pokemon dans pokemon.json
Acceder au Pokedex	Consulte la liste des Pokemon rencontres
Quitter	Ferme le jeu

Ecran de selection du Pokemon

- Le joueur choisit son Pokemon parmi les 151 disponibles
- L'adversaire est choisi aleatoirement dans pokemon.json
- L'adversaire ne peut pas etre le meme Pokemon que celui du joueur
- Les types du Pokemon sont affiches avec leurs couleurs

Ecran de combat

Notre ecran de combat affiche :

- Les deux Pokemon avec leurs sprites/images
- Les barres de vie avec gradient (vert > jaune > rouge)
- Les types de chaque Pokemon avec badges colores
- Les options : Attaquer / Changer de Pokemon / Abandonner
- Le journal de combat avec les messages d'efficacite de type

Ecran du Pokedex

- Liste scrollable des Pokemon rencontres
- Filtrage par type possible
- Compteur : X / 151 Pokemon decouverts
- Details de chaque Pokemon au clic

10. Feedbacks visuels

Messages d'efficacite de type

Multiplicateur	Message	Couleur
x4	C'est hyper efficace !!	Vert vif
x2	C'est super efficace !	Vert
x1	(aucun message special)	Neutre
x0.5	Ce n'est pas tres efficace...	Orange
x0.25	Ce n'est vraiment pas efficace...	Rouge
x0	Ca n'affecte pas le Pokemon...	Gris

Autres feedbacks

- Attaque reussie : animation + affichage des degats + mise a jour barre de vie
- Attaque ratee : message "L'attaque a echoue !" + animation d'echec
- Pokemon KO : barre a 0 + message "[Nom] est KO !" + animation
- Victoire : ecran de victoire avec nom du gagnant + ajout au Pokedex

11. Architecture technique

Vue d'ensemble

Notre architecture suit les principes de la POO. Chaque classe est dans un fichier separe. Le systeme de types utilise un dictionnaire/matrice plutot que 18 classes separees, pour plus de maintenabilite.

Structure des fichiers

```
pokemon/
|-- main.py                # Point d'entree du programme
|-- combat.py              # Classe Combat (gestion des combats)
|-- pokemon.py             # Classe Pokemon (classe de base)
|-- type_chart.py          # Tableau d'efficacite des 18 types (matrice)
|-- pokedex.py             # Classe Pokedex (gestion du Pokedex)
|-- menu.py                # Classe Menu (interface principale)
|-- interface.py           # Classe Interface (GUI Tkinter/Pygame)
|-- utils.py               # Fonctions utilitaires
|-- data/
|   |-- pokemon.json       # Base de donnees des 151 Pokemon
|   |-- pokedex.json       # Pokemon rencontres par le joueur
|   |-- sauvegarde.json    # Sauvegarde de partie (bonus)
|-- assets/
|   |-- sprites/           # Images des Pokemon
|   |-- sounds/            # Effets sonores (bonus)
|-- README.md              # Documentation du projet
```

Classe Pokemon (classe de base)

Attribut	Type	Description
numero	str	Numero du Pokedex National
nom	str	Nom du Pokemon
types	list[str]	Type(s) du Pokemon (1 ou 2)
pv	int	Points de vie actuels
pv_max	int	Points de vie maximum
niveau	int	Niveau du Pokemon
attaque	int	Puissance d'attaque
defense	int	Valeur de defense
evolution	str | None	Nom de l'evolution suivante

Methode	Description
attaquer(adversaire)	Inflige des degats (utilise le tableau des types)
subir_degats(degats)	Reduit les PV en fonction de la defense
est_ko()	Retourne True si PV <= 0
evoluer()	Fait evoluer le Pokemon (modifie stats et nom)
get_types()	Retourne la liste des types
__str__()	Affichage formate des informations

Classe TypeChart

Plutot que de creer 18 classes filles (une par type), nous centralisons le tableau d'efficacite dans une classe TypeChart. Cela rend le code plus maintenable et extensible.

Methode	Description
get_multiplicateur(type_att, types_def)	Retourne le multiplicateur (gere le double type)
get_efficacite(type_att, type_def)	Retourne le multiplicateur pour un type unique
get_faiblesses(type)	Liste les types super efficaces contre ce type
get_resistances(type)	Liste les types peu efficaces contre ce type
get_immunites(type)	Liste les types sans effet contre ce type

Classe Combat

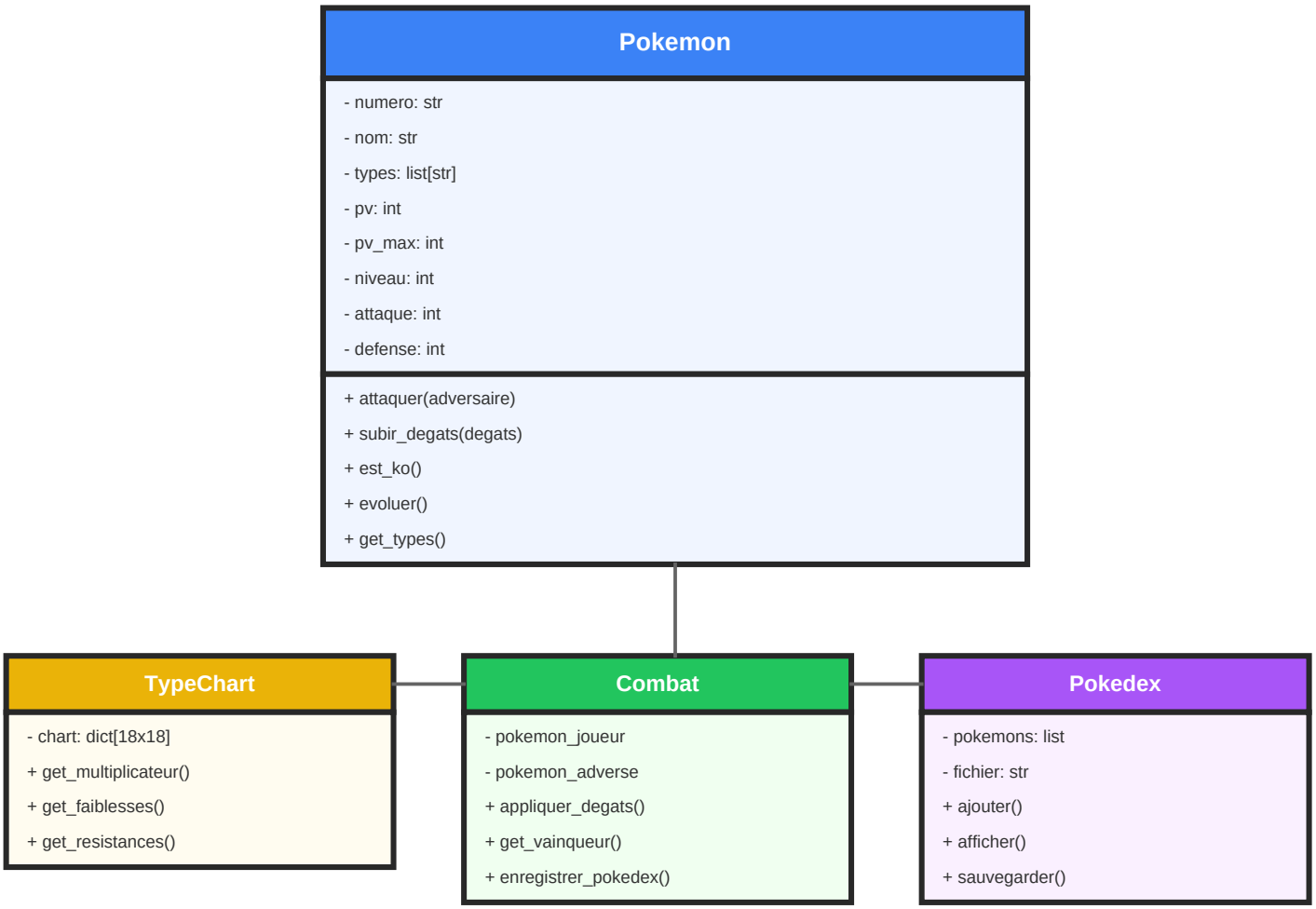
Methode	Description
get_multiplicateur(att, def)	Utilise TypeChart pour calculer le multiplicateur
appliquer_degats(att, def)	Calcule et applique les degats (avec precision)
tour_joueur()	Gere le tour du joueur
tour_adversaire()	Gere le tour de l'adversaire (IA)
get_vainqueur()	Renvoie le Pokemon vainqueur
get_resultat()	Renvoie le resultat complet du combat
enregistrer_pokedex(pokemon)	Enregistre le Pokemon dans le Pokedex

Classe Pokedex

Methode	Description
ajouter(pokemon)	Ajoute un Pokemon (verification par numero)
afficher()	Affiche tous les Pokemon rencontres
get_nombre()	Retourne le compteur (X / 151)
filtrer_par_type(type)	Filtre les Pokemon par type
sauvegarder()	Sauvegarde dans pokedex.json
charger()	Charge depuis pokedex.json

12. Diagramme de classes

Notre architecture de classes :



13. Principes de POO que nous appliquons

Encapsulation

- Les attributs des Pokemon sont proteges (prefixe _)
- L'acces se fait via des properties/getters
- Le tableau des types est encapsule dans TypeChart
- La logique de combat est isolee dans la classe Combat

Heritage

Bien que nous n'ayons pas 18 classes filles, nous utilisons l'heritage pour les specialisations necessaires :

- Pokemon est la classe de base
- Des classes filles peuvent specialiser certains comportements
- Le code commun est dans la classe mere

Polymorphisme

- La methode attaquer() peut etre surchargee dans les sous-classes
- Le calcul des multiplicateurs est generique grace a TypeChart
- L'interface commune permet de traiter tous les Pokemon uniformement

Abstraction

- La classe Pokemon definit l'interface commune
- Le systeme de combat manipule des Pokemon sans connaitre leur type
- L'interface graphique est abstraite de la logique metier

14. Gestion des erreurs et cas limites

Type inconnu

Si un type n'est pas dans notre tableau, le multiplicateur par defaut est 1.0.

Pokemon identique

L'adversaire ne peut pas etre le meme Pokemon. Nouveau tirage automatique.

Fichier JSON absent ou corrompu

Creation d'un fichier par defaut avec les starters (Bulbizarre, Salameche, Carapuce).

PV negatifs

Les PV sont toujours >= 0 grace a max(0, pv - degats).

Defense superieure aux degats

Degats finaux = 0. Message "L'attaque n'a eu aucun effet".

Double immunite

Si le defenseur a deux types et que l'un est immunise (x0), le multiplicateur total est 0.

15. Perimetre et fonctionnalites

MVP (Obligatoire)

Fonctionnalite	Priorite
Menu principal (4 options)	P0
Classe Pokemon avec tous les attributs	P0
Systeme des 18 types avec tableau d'efficacite	P0
Classe Combat avec methodes requises	P0
Systeme de combat tour par tour	P0
Possibilite de rater une attaque	P0
Verification PV et affichage du vainqueur	P0
Pokedex avec sauvegarde JSON (sans doublons)	P0
Ajout de Pokemon dans pokemon.json	P0
Selection du Pokemon (151 disponibles)	P0
Adversaire aleatoire	P0
Chaque classe dans un fichier separe	P0
Interface graphique soignee	P0
Gestion du double type	P0

MVP+ (Pour aller plus loin)

Fonctionnalite	Priorite
Sauvegarde / Reprise de partie	P1
Compteur Pokedex (X / 151)	P1
Filtrage du Pokedex par type	P1

MVP++ (Pour aller encore plus loin)

Fonctionnalite	Priorite
Combat multi-Pokemon (equipe de 6)	P2
Choix attaquer / changer / abandonner	P2
Systeme d'evolution en cours de jeu	P2

Sprites/images des Pokemon	P2
Effets sonores	P2

16. Criteres d'acceptation

Structure du code

CA-01 : Chaque classe est dans un fichier separe

CA-02 : Les principes de la POO sont correctement illustres

CA-03 : La classe Pokemon contient les attributs requis

CA-04 : Le systeme de types couvre les 18 types officiels

Combat

CA-05 : Le systeme de combat tour par tour fonctionne

CA-06 : Le tableau 18x18 est correctement implemente

CA-07 : Le double type est gere (multiplicateurs combines)

CA-08 : La defense reduit les degats subis

CA-09 : Un Pokemon peut rater son attaque

CA-10 : Quand PV = 0, le vainqueur est affiche

CA-11 : La classe Combat contient toutes les methodes requises

Pokedex

CA-12 : Les Pokemon combattus sont enregistres dans pokedex.json

CA-13 : Les doublons sont evites (verification par numero)

CA-14 : Le Pokedex affiche les Pokemon et le compteur X/151

Menu et interface

CA-15 : Le menu propose les 4 options

CA-16 : Le joueur choisit parmi 151 Pokemon

CA-17 : L'adversaire est aleatoire et different du joueur

CA-18 : L'interface affiche les types avec couleurs

CA-19 : Les messages d'efficacite sont affiches

Bonus

CA-20 : Sauvegarde / reprise de partie

CA-21 : Combat multi-Pokemon

CA-22 : Systeme d'evolution fonctionnel

17. Planning et repartition de notre equipe

Notre planning (2 semaines)

Semaine	Jour	Objectifs	Livrables
S1	Lundi	Analyse + architecture	Diagramme de classes, structure du projet, Git
S1	Mardi	Classes de base	Pokemon, TypeChart (18x18), pokemon.json
S1	Mercredi	Systeme de combat	Classe Combat, calcul degats, multiplicateurs
S1	Jeudi	Pokedex + double type	Classe Pokedex, gestion double type, JSON
S1	Vendredi	Tests + corrections	Tests unitaires, debug, revue de code
S2	Lundi	Interface graphique	Menu principal, navigation entre ecrans
S2	Mardi	Ecran de combat	Barres de vie, affichage types, animations
S2	Mercredi	Ecran Pokedex + feedbacks	Liste scrollable, filtres, messages efficacite
S2	Jeudi	Bonus + polish	Evolution, multi-Pokemon, sprites, sons
S2	Vendredi	Rendu final	README, tests finaux, presentation, push GitHub

Repartition de notre equipe

Dev 1

FRONTEND

Semaine 1 : Maquettes, layout general, composants UI

Semaine 2 : Ecrans (menu, combat, Pokedex), animations, feedbacks visuels

Dev 2

BACKEND

Semaine 1 : Classe Pokemon, TypeChart 18x18, pokemon.json (151 Pokemon)

Semaine 2 : Gestion double type, systeme d'evolution, tests unitaires

Dev 3

BACKEND

Semaine 1 : Classe Combat, calcul des degats, attaque ratee

Semaine 2 : Classe Pokedex, sauvegarde JSON, tests unitaires

Dev 4

BACKEND

Semaine 1 : Structure du projet, setup Git, utils, config

Semaine 2 : Bonus (multi-Pokemon, sons), README, presentation

Conventions de notre equipe

- Branches Git : feature/nom-feature, fix/nom-bug
- Commits : messages en francais, descriptifs
- Code : PEP 8, docstrings en francais
- Review : chaque merge request est revue par un autre dev
- Tests : tester les multiplicateurs de type en priorite

18. Direction artistique

Palette de couleurs des 18 types

Type	Code hex	Type	Code hex
Normal	#A8A878	Vol	#A890F0
Feu	#F08030	Psy	#F85888
Eau	#6890F0	Insecte	#A8B820
Plante	#78C850	Roche	#B8A038
Electrik	#F8D030	Spectre	#705898
Glace	#98D8D8	Dragon	#7038F8
Combat	#C03028	Tenebres	#705848
Poison	#A040A0	Acier	#B8B8D0
Sol	#E0C068	Fee	#EE99AC

Elements d'interface

Element	Couleur	Code
Barre PV elevee (>50%)	Vert	#16A34A
Barre PV moyenne (20-50%)	Jaune	#EAB308
Barre PV faible (<20%)	Rouge	#DC2626
Fond de jeu	Vert prairie	#22C55E
Texte principal	Noir	#1F2937
Texte secondaire	Gris	#6B7280
Boutons actifs	Bleu Pokemon	#3B82F6

19. Rendu et competences

Notre presentation

Nous presenterons notre travail avec un support contenant :

- Organisation de notre equipe et repartition des taches
- Notre diagramme de classes
- Les problemes rencontres et nos solutions
- Demonstration live du jeu
- Explication du systeme des 18 types

Notre depot GitHub

Repertoire public nomme "pokemon". Notre README contient :

- Description du projet
- Les 18 types et le tableau d'efficacite
- Instructions d'installation et de lancement
- Explication du systeme de combat
- Diagramme de classes
- Liste des fonctionnalites implementees
- Credits et sources (Pokepedia)

Competences visees

- Installer et configurer un environnement de developpement Python
- Developper des interfaces utilisateur (GUI)
- Developper des composants metier (logique de combat, types)
- Contribuer a la gestion d'un projet informatique (Git, planning)
- Definir l'architecture logicielle d'une application (POO, SOLID)

20. Ressources et references

Sources de donnees Pokemon

- Pokepedia - Liste des Pokemon : pokepedia.fr/Liste_des_Pokemon
- Pokepedia - Table des types : pokepedia.fr/Table_des_types
- Pokedex National complet avec types et stats

Documentation technique

- Python 3 : docs.python.org/fr/3/
- Tkinter : docs.python.org/fr/3/library/tkinter.html
- Pygame : pygame.org/docs/
- JSON en Python : docs.python.org/fr/3/library/json.html
- POO en Python : docs.python.org/fr/3/tutorial/classes.html

Ressources pedagogiques

- Les classes - Documentation officielle Python
- Programmation orientee objet avec Python
- Heritage et polymorphisme en Python
- Gestion de fichiers JSON en Python
- Design Patterns en Python (Factory, Strategy)