

# Projet: Pokémons

Conception et développement d'une application  
informatique

Azelle Courtial, Melvin Hersent, Marie-Dominique Van Damme

Vendredi 03 mai 2024

# Programme première journée

1/ Présentation du projet et des consignes

2/ Choix des groupes d'étudiants

3/ A vous de travailler: analyse du projet, modélisation

# Grands principes du jeu

Les pokémons sont de gentils animaux qu'il faut "capturer".

Un utilisateur va jouer en cherchant à en capturer le plus possible tant que c'est encore possible



Gotta catch 'em all!™

“Attrapez les tous”

# Consignes

# Objectifs pédagogiques

- Mettre en œuvre plusieurs matières vues cette année:
  - l'algorithmie,
  - les outils de gestion de projet,
  - l'analyse informatique
  - la programmation orientée objet en Python
- Vous devrez en partie (dans le désordre):
  - formaliser une analyse du sujet avec les outils UML ;
  - Écrire des algorithmes ;
  - Mettre en œuvre les concepts de l'orienté objet ;
  - Charger, manipuler, visualiser des données ;
  - Produire une documentation de votre code ;
  - Rédiger un rapport d'analyse.

# Autres compétences attendues

- Travailler en équipe. Attention en 5 jours, bien s'organiser !

# Organisation générale

- 12 séances de TP de 3 heures, présence obligatoire à chaque séance dans les salles prévues :
  - Vendredi 3 mai (2 encadrants)
  - Lundi 6 mai (2 encadrants matin, 1 après-midi)
  - Mardi 7 mai (1 encadrant)
  - Lundi 13 mai (2 encadrants)
  - Vendredi 17 mai (2 encadrants)
  - Mardi 21 mai
- Groupe de **3** étudiants

# Encadrement

- Vous serez réparti dans 2 salles informatiques
- Certaines séances seront encadrées, d'autres en autonomie
- 3 enseignants: Azelle Courtial, Melvin Hersent et Marie-Dominique Van Damme



# 2 Livrables (+ Présentation)

- code source
- rapport d'analyse de maximum 5 pages (avec du texte !)

# Deadlines

1/ Dernier commit sur le serveur git : **vendredi 17 mai**  
23h59

2/ Rapport d'analyse à envoyer avant le **mardi 21 mai**  
18h

3/ « Présentation » lors de la journée du **mardi 21 mai**  
9h-17h

# Livrable 1. Code source

- A déposer dans un nouveau dépôt GIT (que vous allez créer):  
<https://github.com/ENSG-POO-2024>
- Le code doit être documenté
- Rédiger un manuel d'utilisation de votre application dans le fichier README de votre repository sur Github
- Toutes les informations seront disponibles dans un dépôt GIT. La première étape consiste à "forker" le projet dans votre dépôt  
=> 1er travail à réaliser vendredi 03 mai

# Livrable 2 : Rapport d'analyse

Le rapport d'analyse permet de formaliser une analyse du cas d'étude, mieux comprendre le fonctionnement demandé et modéliser l'application avant sa réalisation.

Les modèles seront représentés par des diagrammes UML.  
2 à 3 diagrammes maximum par rapport d'analyse.

# Livrable 2 : Rapport d'analyse

Vous pouvez réaliser:

- un diagramme d'utilisations afin d'exprimer les principales fonctionnalités de votre application ;
- un ou plusieurs diagrammes de classes suivant les composants identifiés ;
- un diagramme d'activités pour modéliser votre processus de décision ;
- un diagramme de composants

# Présentation (dernière journée)

Ce n'est **PAS** une soutenance

Nous passerons auprès des groupes pour :

- observer l'application tourner sur vos machines
- vous poser une série de questions

# Matériel de départ

- Cette présentation
- Données: liste de pokémons avec des caractéristiques
- Règles du jeu en détail

# Évaluation (titre indicatif)

Objets	Critères	Barème
Rapport d'analyse	<ul style="list-style-type: none"><li>- Reformulation sujet, comment les règles de l'exercice ont été adaptées, elles doivent être spécifiées.</li><li>- Modélisation UML: diagramme de cas d'utilisations,</li><li>- Modélisation UML: diagramme de classes,</li><li>- Modélisation UML: autre diagramme,</li><li>- Forme du rapport (facile à lire, bien découpé, claire, liens entre le texte et les diagrammes),</li><li>- Explication des fonctionnalités, des algorithmes,</li></ul>	8
Code informatique	<ul style="list-style-type: none"><li>- Application de la POO: classes, héritage, instanciations des objets, attributs, surcharge, conteneur</li><li>- Utilisation de git</li><li>- Clarté des commentaires, respect des conventions de codage</li><li>- Pas d'abus du statique</li><li>- Interface graphique qui se lance bien, interactions avec le joueur, tourne pour 1 combat</li><li>- Architecture du code (ex séparation gui/métier)</li><li>- Bonne répartition du code dans les méthodes,</li><li>- Ajout de tests</li></ul>	12



Sujet du projet

# But de l'application

*Gotta catch 'em all!*™

“Attrapez les tous”

- capturer un exemplaire de chaque pokemon

# État de départ

- le joueur possède déjà quelques pokémons
- des pokémons en liberté sont placés sur une "carte" grâce à leurs coordonnées. Ils ne bougeront pas au cours du jeu.
- les pokémons sauvages sont invisibles. Ils seront découverts par l'utilisateur lors de ses déplacements.

# Fonctionnalités attendues

- Déplacement du joueur
- Découverte d'un pokemon lorsque le joueur est proche de lui
- Combat entre le pokemon d'un joueur et le pokemon sauvage
- S'il gagne, le pokémon en liberté est capturé et appartient au joueur.

# Détail du projet

# Les pokemons

- Chaque pokémon appartient à une famille, ce qui lui donne des caractéristiques. Par exemple: feu, eau, etc.
- Les pokémons possèdent une "barre de vie": un nombre de point de vie fixé au départ (ici représenté par une barre) et dépendant de sa famille
- Les attaques infligent un nombre de dégât déterminé par les règles du jeu
- Un pokemon est hors combat (k.o.) si sa quantité de point de vie passe à zéro ou en dessous.

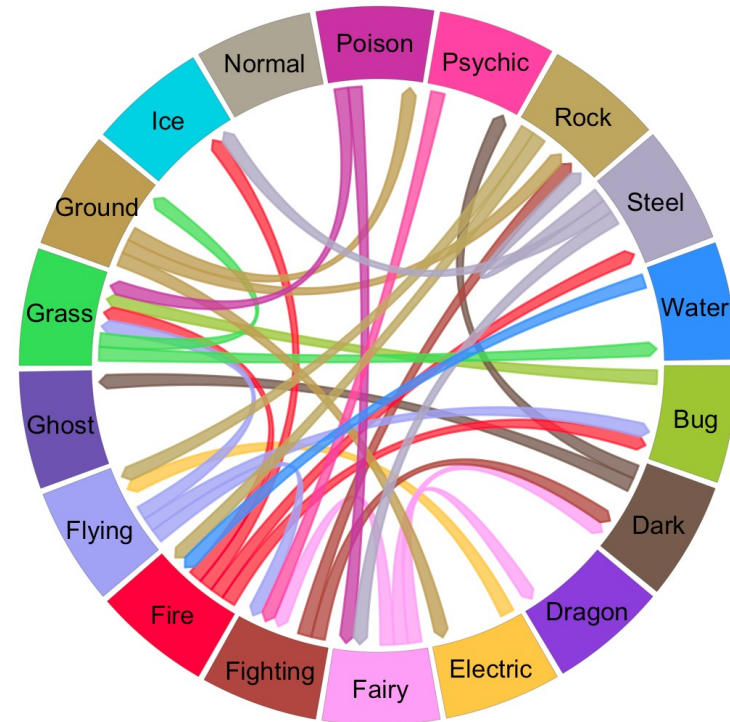
# Les pokemons

- Un combat se joue au "tour par tour" : un pokémon effectue une action, puis c'est au pokémon adverse d'effectuer une action et ainsi de suite.
- Lors de leur tour, l'utilisateur choisit parmi plusieurs actions:
  - attaquer,
  - changer de pokemon,
  - fuir
- Les attaques peuvent être choisies parmi une liste d'attaques que connaît le pokémon:
  - une attaque "neutre"
  - une attaque de l'élément du pokemon (eau, feu, ...).

# Les familles (ou types) de pokemons



Pokemon types dominance



Type A dominates type B if type A attacks are supereffective in type B and type B attacks are not very effective in type A

Version 1.0



- A vous de travailler !

Avez-vous des questions ?

# 3 packages à programmer

- Jeu: classes pour gérer le jeu. Session de l'utilisateur, une classe qui pilote le jeu, etc.
- GUI : interface graphique. Interactions entre l'utilisateur et le jeu. Viser simple pour commencer !
- Gestion des pokémons: chargement, modélisation, combat, état, etc.