

# Projet: Pokémons

Conception et développement d'une application informatique

Malika Grim-Yefsah, Melvin Hersent, Marie-Dominique Van Damme

Mardi 11 avril 2023

# Programme après-midi

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
- Pourquoi ce projet ? En 5 jours, on peut mettre en pratique la POO de façon ludique.

# Consignes

# Objectifs pédagogiques

- Mettre en œuvre plusieurs matières vues cette année:
  - outils de gestion de projet,
  - l'analyse informatique
  - la programmation orientée objet en JAVA
- 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 ;
  - Présenter les algorithmes utilisés et le modèle de données devant un auditoire
  - Rédiger un rapport technique, expliquer clairement un algorithme, son contexte

# Autres compétences attendues

- Travailler en équipe. Attention en 5 jours, bien s'organiser
- Synthétiser et présenter une partie du travail réalisé à l'oral

# Organisation générale

- Planning court: aujourd'hui & vendredi 14 avril au vendredi 21 avril
- 11 séances de TP de 3 heures, présence obligatoire à chaque séance
- Groupe de 3 étudiants
- Deadline : rendre projet le vendredi 21 avril 2023 - au soir, minuit
- Soutenances: journée du mardi 9 mai 2023

# Encadrement

- Vous serez réparti dans 3 salles
- Certaines séances seront encadrées, d'autres en autonomie
- 3 enseignants: Malika Grim-Yefsah, Melvin Hersent et Marie-Dominique Van Damme



# 3 Livrables

1. La soutenance du projet comprend une présentation ainsi qu'une démonstration (plus de détail bientôt)
- A livrer avant vos vacances:
    - 2. code source
    - 3. rapport d'analyse de maximum 6 pages (avec du texte !)

# Livrable 1 : 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. Vous pouvez réaliser:

- un diagramme d'utilisations afin d'exprimer les grandes 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

# Livrable 2. Soutenance

plus de détail bientôt

# Livrable 3. Code source

- A déposer dans un nouveau dépôt GIT (que vous allez créer):  
<https://github.com/ENSG-POO>
- 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 14 avril

# Matériel de départ

- Cette présentation
- Complément pour la soutenance
- Données: liste de pokémons avec des caractéristiques
- Règles du jeu en détail
- Squelette d'une interface graphique pour démarrer

# Évaluation (titre indicatif)

Objets	Critères	Barème
Rapports analyse	Reformulation du sujet, Hypothèses de travail, Modélisation UML, Nombre de fonctionnalités, Forme du rapport.	7
Soutenance orale	Qualité de l'expression orale, Qualité du support, Respect des contraintes de temps, Réponses aux questions, Démonstration dynamisme, clarté.	5
Code informatique	Mise en œuvre des concepts objets, Clarté des notations, respect des conventions, Organisation modulaire, Tests, Manuel d'utilisation et d'installation.	8

Sujet du projet

- Au départ, le joueur possède déjà quelques pokémons.
- Le but du jeu est de chercher à en capturer d'autres en faisant combattre un des pokémons du joueur contre un autre "sauvage".
- Au départ, des pokémons en liberté sont placés sur une "carte" grâce à leurs coordonnées. Ils ne bougeront pas au cours du jeu. Au démarrage, les pokémons sont tous invisibles. Ils vont être découverts par l'utilisateur (choix d'un lieu par un clic), les pokémons "proches" sont alors "à capturer".
- L'utilisateur choisit les 2 pokémons dont il veut organiser le combat. Un pokémon est à choisir parmi les siens et un autre visible sur la carte.
- Si le pokémon de l'utilisateur perd son combat, le combat est fini. S'il gagne, le pokémon en liberté est capturé et appartient au joueur.



- Chaque pokémon appartient à une famille, ce qui lui donne des caractéristiques.  
Par exemple: feu, eau, etc.

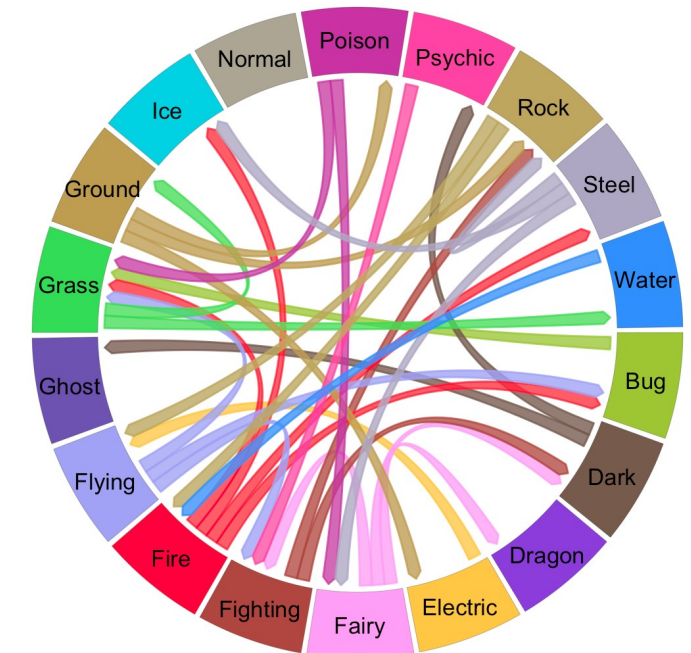


- Les pokémons possèdent une "barre de vie": un nombre de point de vie fixé au départ (ici représenté par une barre) et dépendant de sa famille.

- 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.

**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

- 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, ...).

- 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.