

# Alignement pédagogique

---

Session de Restitution de l'Appel à Candidatures NewDEAL 2024

Laurent Réveillère & Christophe Casseau

# Caractéristiques de l'UE

---

- *UE Programmation orientée objet*
  - UE portée par l'UF Informatique
- **Public concerné**
  - L2 parcours Informatique (Licence Informatique)
  - L3 parcours Math-Info (Licence Informatique & Licence Mathématiques)
  - L3 parcours CMI ISI (Licence Informatique & Licence Mathématiques)
  - L3 parcours OPTIM (Licence Informatique & Licence Mathématiques)
- **Effectif**
  - Environ 190 étudiants
  - 6 groupes : équipe pédagogique constituée de 5 enseignants permanents

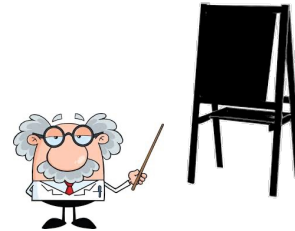
# Caractéristiques de l'UE

- Organisation

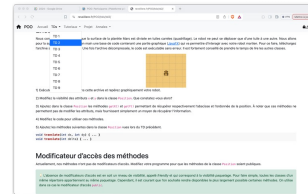
- Semestre de printemps
- Alternance sur 12 semaines de 1h20 de CM et 2h40 de TD sur machine
- Groupes d'environ 30 étudiants

- Modalités

- Cours à la craie en amphithéâtre
  - Pas de support de cours
  - Références vers des ressources en lignes



- Énoncés de TD disponibles en ligne



# État initial avant alignement pédagogique

---

- TD sur machine
  - 9 séances sur deux grandes thématiques
  - Exercices progressifs que l'étudiant travaille en autonomie
- Projet
  - Encadrement sur 3 séances
  - Travail en binôme
- Évaluations (CCI)
  - QCM en ligne sur Moodle (semaine 3) Coef. 10%
  - TP noté (semaine 6) Coef. 30%
  - Projet (semaines 10 à 12) Coef. 30%
  - DST (QCM + questions ouvertes) Coef. 30%

# Ressource existante

The image displays three overlapping screenshots of a Moodle course page and a tutorial page. The top-left screenshot shows the Moodle course page for 'Programmation Orientée Objet (2023-2024)' with navigation tabs for 'Cours', 'Paramètres', 'Participants', 'Notes', 'Rapports', and 'Plus'. The top-right screenshot shows a table of contents for 'POO' with a dropdown menu for 'TD 4' selected. The bottom-left screenshot shows the course page for 'PROGRAMMATION ORIENTÉE OBJET EN JAVA' with a table of evaluation modalities and a 'Cours magistral' section. The bottom-right screenshot shows a tutorial page for 'RATON MARTIENNE' with instructions on abstract classes and code organization.

**PROGRAMMATION ORIENTÉE OBJET EN JAVA**

L'objectif principal de ce cours porte sur la mise en oeuvre de programmes sous le paradigme objet dans le langage JAVA.

Cet enseignement abordera les notions centrales d'objets, de classes, de méthodes, d'instanciation et d'exception. Les bases de l'approche objet que sont l'encapsulation, la modularité, la délégation et l'héritage. L'étudiant sera amené à utiliser des bibliothèques logicielles existantes.

### Modalité de contrôle des connaissances et des compétences (MCCC)

Contrôle continu intégral, pas de seconde session.

Évaluation	Coefficient
Micro-évaluations	10%
TP noté	30%
Projet	30%
DST	30%

### Cours magistral

Les cours magistraux ont lieu (en principe) le lundi à 11h (A22/Amphithéâtre Alfred WEGENER). Consulter l'emploi du temps en ligne sur Celcat pour les modifications de lieux et d'horaires.

Jour	Créneau	Enseignant
Lundi	09:30 - 12:20	Laurent Réveillère

**POO** Accueil TDs Tutoriaux Projet Annales

**Table des matières**

- TD 1
- TD 2
- TD 3
- TD 4
- TD 5
- TD 6
- TD 7
- TD 8
- TD 9

**TD 4** RATION MARTIENNE

⚠ Vous devez terminer le TD précédent avant de commencer celui-ci.

### Classes abstraites

- Dans la classe `Vehicule`, les méthodes `distance` et `getPathTo` retournent des valeurs fictives. En réalité, cela ne fait pas de sens d'instancier un objet de la classe `Vehicule`. Nous allons empêcher cela en définissant la classe `Vehicule` comme **abstraite**. Modifiez le code en conséquence.
- De même, faites en sorte que la classe `Sprite` soit abstraite.

### Organisation du code

Nous commençons à avoir beaucoup de classes et cela devient difficile de s'y retrouver.

- Créez un package `view` et déplacez les classes `ImageResource`, `View` et toutes les classes `Sprite` dans le package `view`.
- Créez un package `model` et déplacez les classes en relation avec les véhicules dedans.

Un package en Java correspond à un répertoire. Utilisez les outils de refactoring pour déplacer correctement vos classes dans de nouveaux dossiers.

# Manque d'alignement

---

- Expression initiale du besoin d'accompagnement
  - Définir les acquis d'apprentissage et identifier ceux qui sont travaillés dans les différentes activités pédagogiques
  - Faire évoluer les évaluations

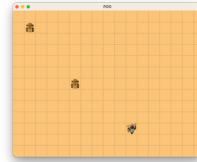


- Cerise sur le gâteau
  - Mieux structurer et présenter les activités pédagogiques
  - Faire le lien avec les compétences portées par le BCC

# Structuration en séquences

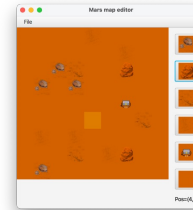
- Séquence 1 : Exploration de Mars

- Afficher des robots d'exploration
- Gérer les déplacements



- Séquence 2 : Modélisons le monde martien

- Modéliser une carte
- Manipulation de fichiers et gestion des erreurs



- Séquence 3 : Projet

- Un hérisson est perdu et désespéré. Votre mission, si vous l'acceptez, est d'aller le secourir. Pour cela, vous devrez traverser plusieurs jardins, plus fantastiques les uns que les autres.



# Définition des acquis d'apprentissages

---

- **AA1** : Identifier les principes de base de la programmation orientée objet (héritage, objet, polymorphisme etc.)
- **AA2** : Expliquer les notions principales en programmation orientée objet en précisant leur intérêt (objet, héritage, polymorphisme, gestion d'erreurs)
- **AA3** : Utiliser le langage Java pour mettre en œuvre les notions principales de programmation orientée objet
- **AA4** : Décrire le fonctionnement d'un programme Java lors de son exécution en se basant sur un extrait de code donné
- **AA5** : Utiliser l'API Java pour résoudre des problèmes courants dans des applications logicielles
- **AA6** : Analyser la qualité d'un code Java donné
- **AA7** : Produire un code Java de qualité



# UE dans le contexte des BCC

---

- BCC Programmation
- Compétences associées
  - C1 : Travailler en en autonomie et collaborer en équipe au service d'un projet
  - C2 : Appréhender le fonctionnement d'un ordinateur et d'un réseau
  - C3 : Respecter les bonnes pratiques et maîtriser les outils adaptés pour le développement logiciel
  - C4 : Mettre en œuvre une solution technique efficace en utilisant différents paradigmes de programmation

# Compétences associées aux AA

---

- **AA1** : Identifier les principes de base de la programmation orientée objet (héritage, objet, polymorphisme etc.)
  - **Compétences** : C2, C3
- **AA2** : Expliquer les notions principales en programmation orientée objet en précisant leur intérêt (objet, héritage, polymorphisme, gestion d'erreurs)
  - **Compétences** : C2, C3
- **AA3** : Utiliser le langage Java pour mettre en œuvre les notions principales de programmation orientée objet
  - **Compétences** : C2, C3, C4
- **AA4** : Décrire le fonctionnement d'un programme Java lors de son exécution en se basant sur un extrait de code donné
  - **Compétences** : C2
- **AA5** : Utiliser l'API Java pour résoudre des problèmes courants dans des applications logicielles
  - **Compétences** : C1, C3, C4
- **AA6** : Analyser la qualité d'un code Java donné
  - **Compétences** : C2, C3
- **AA7** : Produire un code Java de qualité
  - **Compétences** : C1, C2, C3, C4

# Utilisation des Acquis d'apprentissage

- **AA2** : Expliquer les notions principales en programmation orientée objet en précisant leur intérêt (objet, héritage, polymorphisme, gestion d'erreurs)
  - **Compétences** : C2, C3
- **AA3** : Utiliser le langage Java pour mettre en œuvre les notions principales de programmation orientée objet
  - **Compétences** : C2, C3, C4
- **AA7** : Produire un code Java de qualité
  - **Compétences** : C1, C2, C3, C4

## ACQUIS D'APPRENTISSAGE

- Compiler et exécuter un programme rudimentaire en Java
- Implémenter une classe simple
- Manipuler le concept d'objets

## TD 1 : PREMIERS PAS

Dans cette première séance nous allons voir comment compiler et exécuter un fichier de code source Java, et effectuer des manipulations basiques sur les objets.

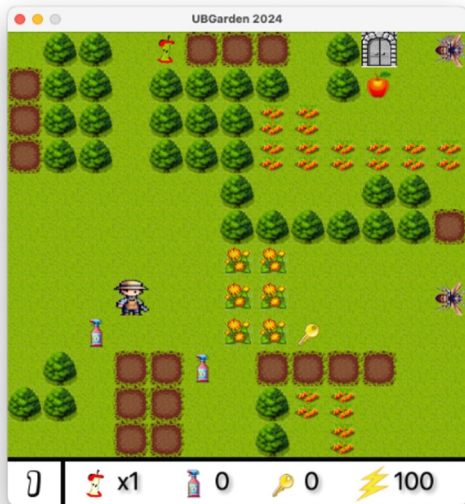


À insérer dans chaque TD

# Présentation du projet aux étudiants

## PROJET DE POO

Réalisation d'un jeu vidéo 2D : **UBGarden**.



### Principes du jeu

Un hérisson est perdu et désespéré. Votre mission, si vous l'acceptez, est d'aller le secourir. Pour cela, vous devrez traverser plusieurs jardins, plus fantastiques les uns que les autres.

Chaque jardin est représenté par une carte (rectangulaire) et nécessiteront d'avoir une clé pour les ouvrir. Attention, chaque cellule en cellule, dans un espace géométrique discret. Le jardinier ne peut pas s'y aventurer. Chaque déplacement du jardinier est par `diseaseLevel`, initialisé à 1. Les frelons peuvent voler `energyRecoveryDuration` secondes sans bouger, dans la limite de leur énergie.

En cas de piqure d'un frelon, vous perdez 20 points d'énergie. Les insecticides qui tuent les frelons. Chaque bombe peut tuer un frelon et se fait piquer, la partie est perdue. Les frelons peuvent voler.

Le jardinier ramasse automatiquement les bonus qui se trouvent dans les insecticides qui tuent les frelons. Chaque bombe peut tuer un frelon et se fait piquer, la partie est perdue. Les frelons peuvent voler.

Image	Description	Effet
	Herbe	Le jardinier consomme 1 * <code>diseaseLevel</code>
	Terre	Le jardinier consomme 2 * <code>diseaseLevel</code>
	Carottes	Le jardinier consomme 3 * <code>diseaseLevel</code>
	Pomme empoisonnée	Augmente <code>diseaseLevel</code> de 1
	Pomme	Réinitialise <code>diseaseLevel</code> à 1
	Bosquet de fleurs	Infranchissable pour le jardinier
	Bombe insecticide	Permet de tuer un frelon, le jardinier se fait piquer
	Nid de frelons	Génère un nouveau frelon et tuer le jardinier
	Frelon	Frelon pas sympa. Le frelon meurt si le jardinier se fait piquer
	Jardinier	Jardinier super cool
	Hérisson	Hérisson apeuré
	Porte fermée	Permet de tuer un frelon, le jardinier se fait piquer
	Porte ouverte	Permet de tuer un frelon, le jardinier se fait piquer
	Clé	Permet d'ouvrir une porte fermée

### TRAVAIL À FOURNIR

Archive du jeu de base : [UBGarden-student-2024.zip](#)

### Premiers pas

Dans cette première version, les frelons sont immobiles. Nous utiliserons pour le moment la carte par défaut.

- Modifiez le code pour afficher tous les éléments ainsi que les frelons et le hérisson. Vous pouvez maintenant utiliser la carte `MapLevelDefault` à la place de `MapLevelDefaultStart`.
- Modifiez le code pour que les mouvements du jardinier soient limités par le cadre de la carte et les éléments de décor.
- Faire en sorte que le jardinier perde de l'énergie lorsqu'il touche un frelon.
- Faire en sorte que la partie se termine par une défaite si le jardinier meurt.
- Faire en sorte que la partie se termine par une victoire si le jardinier retourne le hérisson.
- Mettre à jour le panneau d'affichage avec : niveau de fatigue, nombre de clés et niveau d'énergie
- Faire en sorte que le jardinier dépende de l'énergie quand il se déplace
- Permettre au jardinier de récupérer de l'énergie quand il reste immobile assez longtemps
- Faire en sorte que le jardinier puisse ramasser les bonus

### Donnons vie aux frelons

- La ruche produit un nouveau frelon toutes les 10 secondes.
- Faire en sorte que chaque frelon se déplace tout seul de manière aléatoire
- Un frelon doit mourir quand il pique le jardinier
- La vitesse de déplacement d'un frelon est définie par la variable de configuration `harnetMoveFrequency` qui indique le nombre de fois par seconde qu'un frelon peut bouger.

### Gestion des mondes

Dans la version de base, le jeu ne dispose que d'un seul niveau codé en dur dans le code. Nous allons maintenant charger une configuration complète de jeu depuis un fichier. Vous trouverez un répertoire `world` à la racine du projet avec deux fichiers d'extension `.properties` représentant un monde avec 2 niveaux (avec ou sans compression des niveaux). Nous utilisons le codage LRE vu en TD avec ou sans compression en fonction du paramètre `compression` dans le fichier de configuration.

- Permettre le chargement d'une carte depuis un fichier
- Gérez les portes ouvertes, le ramassage des clés et l'ouverture des portes fermées.

### Bombes insecticides

- À chaque fois qu'un frelon sort de la ruche, une bombe insecticide apparaît sur une case aléatoire (herbe) qui ne contenait pas déjà un bonus.
- Le jardinier peut ramasser des bombes insecticides (le compteur est mis à jour)
- Si le jardinier dispose d'une bombe insecticide quand il rencontre un frelon, il utilise sa bombe pour tuer le frelon et ne se fait pas piquer, mais il perd sa bombe.

# Évolution des évaluations

---

- Augmenter le nombre de QCM
  - Rappel : test sur Moodle, coef. 10%
  - Existant : un test semaine 3
  - Objectif : Définir 3 tests, un test sur chaque séquence
  
- Projet
  - Communiquer une version simplifiée de la grille d'évaluation aux étudiants
  - Faire un test Moodle (séquence 3) spécifique sur le projet

# Ajout d'évaluations formatives

- Bilan d'une séance avec une évaluation formative
  - Exemple de grille critériée potentielle
    - **AA1** : Identifier les principes de base de la programmation orientée objet (héritage, objet, polymorphisme etc.)
      - **Compétences** : C2, C3

Critères	Niveau 1 (Débutant)	Niveau 2 (Intermédiaire)	Niveau 3 (Avancé)
Identification des concepts de base (objets, classes)	Reconnait les termes mais ne peut pas les définir	Définit les termes avec quelques erreurs	Définit précisément les termes avec exemples
Compréhension de l'héritage	Confond l'héritage avec d'autres concepts	Explique l'héritage mais avec des confusions	Explique clairement avec des exemples
Polymorphisme	Incapable d'expliquer	Comprend partiellement avec des exemples limités	Explique clairement avec des exemples variés

# Conclusion

---

- Difficultés rencontrées
  - Expression et granularité des acquis d'apprentissage
  - Activités pédagogiques couvrant parfois un trop grand nombre d'acquis d'apprentissage
  - Expression des compétences trop générique
  - UE en cours sur le semestre de printemps, impossible de mettre en application les propositions de restructuration cette année

La route vers l'alignement pédagogique est un chemin sinueux

