Projet : L'Île Interdite

- Auteurs : Alexis VO et Théo GERVINET
- Groupe : LDD2 - IM1
- Date : Avril 2025
- Université Paris-Saclay

1. Parties du sujet traitées

Dans ce projet, nous avons développé une version numérique du jeu **L'Île Interdite**.
Les principales fonctionnalités implémentées sont :

- Création et affichage de la **grille 6x6** de l'île (zones normales, inondées, submergées) qui reproduit la même disposition que dans le jeu réel.
- **Déplacement** des pions avec boutons directionnels et clavier (ZQSD).
- Gestion de **l'inondation** progressive de l'île.
- **Ramassage d'artefacts** et **gestion des clés**.
- Système **tour par tour** avec actions limitées.
- Gestion de la vue **victoire** ou **défaite** selon l'état du jeu.
- **Rôles** : Intégration de capacités spéciales : *Ingénieur* et *Pilote*.
- **Interface graphique avancée** :
    - Arrière-plan personnalisé,
    - Boutons stylisés et colorés,
    - Affichage du joueur actif et de son rôle,
    - Cartouche "Actions restantes" et panneau "Joueurs et clés",
    - Bouton **Mute** pour la musique de fond.
    - Images pour chaque zone qui change selon son état.
    - Joueurs représentés par de vrais pions.

---

2. Choix d'architectures

Le projet suit une architecture **MVC** claire :

Composant	Rôle
Modèle (`ile.modele`)	Gère la logique du jeu (zones, pions, inondations, artefacts)
Vue (`ile.vue`)	Affiche l'interface graphique (grille, boutons, infos)
Contrôleur (`ile.controleur`)	Gère les interactions clavier/souris/boutons

Principales classes :

- `Ile` : Plateau de jeu.

- `Zone`, `Temple`, `Heliport` : Différents types de cases. **Temple** et **Heliport** héritent de la classe `Zone`, ce qui permet de gérer facilement leur comportement particulier tout en manipulant des références communes dans la grille.
- `Pion` : Joueur avec clés, artefacts et rôle.
- `VueIle` : Interface graphique Swing.
- `ControleurIle` : Logique de déplacement, ramassage et assèchement.
- `Role` (enum) : Gestion des rôles spéciaux (à venir).

Détails d'implémentation graphique :

La grille de l'île est modélisée par un `JPanel` principal utilisant un `GridLayout 6x6`, subdivisé en 36 `JLabel`, chaque JLabel représentant une Zone graphique de la carte. Les pions sont représentés par des images ajoutées dynamiquement dans ces `JLabel`.

Technologies utilisées :

- **Swing** pour l'affichage graphique.
- **Gestion du son** via `javax.sound.sampled`.
- **Layout managers** (`BorderLayout`, `GridLayout`, `BoxLayout`) pour l'organisation.
- **Listeners** pour gérer actions clavier, souris, boutons.
- **Design patterns** appris en cours (MVC).

---

3. Problèmes présents non éliminés

- Quand plusieurs pions sont sur la même case, les images peuvent légèrement se chevaucher.

---

4. Morceaux de code écrits à plusieurs ou empruntés

- **Lecture audio** : modèle de code trouvé pour lire un fichier .wav en Java.
- **Fond d'image** : Inspiration pour utiliser `paintComponent` dans un `JPanel`.
- Aucune autre partie du code n'a été copiée : l'ensemble de l'architecture, des classes, de la logique du jeu et de l'interface a été développé par nos soins.

5. Améliorations possibles

- Automatisation de la "Fin de tour" dès que le nombre d'actions est nul.
- Implémentation des autres rôles
- Jouer contre l'ordinateur

6. Diagramme de classes