

# LinguaCrypt Roadmap

## Jour 1 : Lundi 06/01 - Initialisation et Planification

### Objectifs

- Lire attentivement le sujet et les règles complètes du jeu CodeNames™ pour en comprendre tous les détails.
- Répartir les rôles au sein de l'équipe : responsable de l'interface utilisateur, du backend, des tests, etc.
- Installer tous les outils nécessaires : Java (avec un JDK récent), JavaFX, un IDE (comme IntelliJ IDEA ou Eclipse), Git, Gradle pour l'automatisation des builds.
- Configurer le dépôt Git : conventions de commits, branches, utilisation des tags pour les livraisons quotidiennes.
- Prototyper une interface utilisateur avec une grille simple affichant les 25 cases.
- **Établir les principales classes et packages (architecture initiale) :**
  - Classe principale pour lancer l'application.
  - Modèle pour représenter les cartes (nom, équipe, etc.).
  - Contrôleurs pour gérer la logique du jeu.
  - Interface pour la grille de jeu.
- Planifier les activités de la semaine avec un diagramme Gantt ou un tableau Kanban.
- Installation de Gradle et tests de configuration (builds et dépendances).
- Première application :
  - Créer un projet gradle pour avoir une base commune
  - Créer le layout de l'application
  - Créer la bd des mots
  - Créer une première grille du jeu, ça sera pour après de pouvoir fixer la grille
  - Créer deux fenêtres pour les deux grilles

### Livrables

- README initial contenant une présentation du projet, les instructions pour configurer l'environnement et les rôles de l'équipe.
- Grille statique démontrant la faisabilité de l'affichage.
- Diagramme d'architecture initial.

## Jour 2 : Mardi 07/01 - Fonctionnalités de Base

### Objectifs

- Permettre la création d'une partie avec les paramètres suivants :
  - Taille de la grille (par défaut : 5x5).
  - Liste de mots thématiques.
  - Nombre de joueurs et rôles (maître-espion, équipe).
  - Limitation de temps (on/off).
- Implémenter la logique de sélection des cartes "Nom de Code" :
  - Assignment aléatoire des équipes (rouge, bleu, neutre, assassin).
  - Affichage masqué pour les joueurs sauf pour le maître-espion.
- Implémenter les règles principales :
  - Validation des indices proposés par le maître-espion.
  - Interaction des joueurs pour deviner les mots.
  - Gérer les conséquences des choix : points gagnés, fin de partie si l'assassin est trouvé.

- Tester l'application pour s'assurer que les règles de base fonctionnent correctement.
- Création d'une fenêtre de menu pour paramétrer la partie (ex. nombre de joueurs, thématique).
- Gestion de l'affichage des rôles (fenêtre des maîtres et fenêtre des espions distinctes).

### **Livrables**

- Une interface fonctionnelle permettant de créer une partie avec tous les paramètres.
- Gestion dynamique de la grille et de l'attribution des cartes.
- Documentation des règles implémentées avec exemples.

## **Jour 3 : Mercredi 08/01 - Fonctionnalités Avancées**

### **Objectifs**

- Ajouter la possibilité de sauvegarder une partie en cours :
  - Création d'un système de sauvegarde (sérialisation des données de jeu).
  - Recharger une partie à partir d'un fichier sauvegardé.
- Intégrer un sablier pour limiter le temps de réflexion :
  - Ajout d'un minuteur visuel pour les joueurs.
  - Mode "blitz" avec des durées limitées pour chaque phase.
- Permettre l'édition et le choix des cartes thématiques :
  - Interface pour ajouter de nouveaux mots ou importer une liste prédéfinie.
  - Gestion des erreurs (mots en double, entrées invalides).
- Utilisation d'une base de données pour générer des mots aléatoires ou des images (cartes "Nom de Code").

### **Livrables**

- Sauvegarde et rechargement fonctionnels.
- Mode blitz testé avec des scénarios variés.
- Outil d'édition et gestion des thématiques de cartes.

## **Jour 4 : Jeudi 09/01 - Mode Solo et Extensions**

### **Objectifs**

- Développer un mode solo avec des indices préprogrammés :
  - Ajuster la difficulté (niveau facile, moyen, difficile).
- Ajouter un mode "images" :
  - Conversion des cartes textuelles en cartes visuelles (illustrations).
  - Interface pour créer et importer des cartes images.
- Intégrer une gestion des statistiques :
  - Stocker les données de chaque partie (temps, scores, victoires).
  - Afficher les performances des joueurs sous forme de tableau.
- Effectuer des tests approfondis pour assurer la stabilité et corriger les bugs.

### **Livrables**

- Mode solo pour jouer l'agent espion.
- Support des cartes avec images, incluant un exemple prédéfini.
- Tableau des statistiques accessible depuis le menu principal.

## **Jour 5 : Vendredi 10/01 - Finalisation et Livraison**

### **Objectifs**

- Intégrer toutes les fonctionnalités dans une version stable et homogène.
- Réaliser des tests complets couvrant tous les scénarios possibles.
- Créer une vidéo démonstrative de 10 minutes :
  - Montrer toutes les fonctionnalités principales et avancées.
  - Ajouter des commentaires audio pour expliquer les étapes.
- Préparer la livraison finale avec toutes les instructions (RELEASE\_FINAL).

### **Livrables**

- Application finale déployable avec un manuel d'installation clair.
- Vidéo de démonstration publiée en ligne (lien ajouté au README).
- Dépôt Git complet avec toute la documentation (README, architecture, instructions).

### **Évolutions Futures**

- Intégrer un mode réseau pour jouer en ligne :
  - Utilisation de sockets ou d'une bibliothèque tierce pour la communication réseau.
  - Création de lobbies pour accueillir plusieurs joueurs.
- Ajouter une intelligence artificielle avancée :
  - Implémenter un système adaptatif capable d'apprendre les stratégies des joueurs.
- Proposer un mode coopératif inspiré de la version Duo :
  - Mécanique adaptée pour deux joueurs travaillant ensemble contre un score cible.