**Documentation Technique**

**Titre :** *Laying Glass - Documentation technique*

**1. Introduction**

Le projet "Laying Glass" est un jeu de stratégie basé sur des mécanismes de placement de tuiles, inspiré par *The Devil's Plan* (Netflix). Il simule un environnement compétitif où les joueurs doivent placer des tuiles sur un plateau afin de maximiser leur territoire.

Ce document décrit l'architecture technique, les choix de conception, les méthodes clés et les aspects d'optimisation implémentés dans ce projet.

**2. Objectifs Techniques**

1. **Concevoir un jeu robuste et flexible :**
   * Permettre à 2 à 9 joueurs de participer.
   * Intégrer une logique claire pour le placement des tuiles et le calcul des scores.
2. **Assurer une modularité :**
   * Faciliter l'ajout de nouvelles règles ou modifications dans le futur.
3. **Garantir l'intégrité des règles de jeu :**
   * Les tuiles doivent respecter les contraintes de placement.
   * Le système doit correctement identifier le gagnant.

**3. Architecture du Projet**

L'architecture du jeu est centrée sur quatre classes principales : Board, Joueur, Tuile, et LayingGlass.

**Description des Classes**

**a. Board**

* **Rôle :** Modélise le plateau de jeu.
* **Responsabilités :**
  + Gérer la grille de 20x20 ou 30x30.
  + Vérifier les contraintes de placement des tuiles.
  + Calculer les scores (taille des carrés et cases occupées).
* **Attributs :**
  + grille : Représente le plateau sous forme de matrice 2D.
  + largeur, hauteur : Dimensions du plateau.
* **Méthodes principales :**
  + peutPlacerTuile : Vérifie si une tuile peut être placée à une position donnée.
  + PlacerTuile : Place une tuile sur le plateau si elle respecte les règles.
  + calculerTaillePlusGrandCarre : Calcule la taille maximale d’un carré.

**b. Joueur**

* **Rôle :** Modélise les joueurs.
* **Responsabilités :**
  + Gérer les informations des joueurs (nom, couleur, score).
  + Traiter les interactions avec les tuiles.
* **Attributs :**
  + name, color, score : Données essentielles du joueur.
  + couponsEchange : Nombre de coupons disponibles.
  + cases : Liste des positions occupées sur le plateau.
* **Méthodes principales :**
  + utiliserCoupons : Décrémente le nombre de coupons si disponible.
  + addCases : Ajoute une case au territoire du joueur.

**c. Tuile**

* **Rôle :** Modélise une tuile du jeu.
* **Responsabilités :**
  + Représenter les formes des tuiles.
  + Offrir des fonctionnalités de manipulation (rotation, retournement).
* **Attributs :**
  + shape : Une matrice 2D représentant la forme.
  + orientation : Indique l’état actuel de la tuile (normal ou retournée).
* **Méthodes principales :**
  + rotation : Tourne la tuile de 90°.
  + retourner : Retourne la tuile horizontalement.
  + display : Affiche la tuile en mode standard ou compact.

**d. LayingGlass**

* **Rôle :** Gère l'ensemble du jeu.
* **Responsabilités :**
  + Configurer les joueurs et le plateau.
  + Gérer les tours et les interactions.
  + Déterminer le gagnant à la fin.
* **Méthodes principales :**
  + setupGame : Configure les joueurs et leurs couleurs.
  + game : Exécute les 9 tours du jeu.
  + determinerGagnant : Identifie le gagnant selon les règles.

**5. Décisions Techniques**

1. **Utilisation de std::vector :**
   * Facilite la manipulation dynamique des grilles et des tuiles.
2. **Structuration Modulaire :**
   * Séparation des responsabilités entre les classes pour une meilleure maintenabilité.