

Rapport de projet

Juste Prix

Sommaire

1 Description du projet.....	4
2 Prérequis pour lancer le projet.....	6
2.1 Environnement Logiciel.....	6
2.2 Exécution.....	6
2.2.1. Compilation :.....	6
3 Diagrammes UML.....	6
4 Fonctionnalités implémentées.....	9
3.1 Écran d'accueil (WelcomeScreen).....	9
3.2 Logique du Jeu (GuessingGame).....	9
3.3 Interface du Jeu (GameScreen).....	10
3.4 Gestion des Joueurs (PlayerUser).....	11
5 Difficultés et fonctionnalités non implémentées.....	11
5.1 Difficultés rencontrée.....	11
5.2 Fonctionnalités non implémentées.....	12
Conclusion.....	13

1 Description du projet

Le projet est une application interactive conçue en Java, utilisant le framework **Java Swing** pour offrir une expérience utilisateur intéressante, enrichie par différents éléments graphiques et sonores personnalisés.

Ce jeu de devinettes repose sur un principe simple : le joueur doit deviner un nombre mystère généré aléatoirement, en s'appuyant sur des indices qui l'aident à se rapprocher de la bonne réponse. L'objectif est de trouver ce nombre avec le moins d'essais possible. Parmi ses fonctionnalités principales, le jeu permet de :

- Démarrer une nouvelle partie avec le choix de différents niveaux de difficulté,
- Sauvegarder et recharger une partie en cours,
- Afficher le score et le temps restant en temps réel,
- Intégrer un mode hexadécimal, pour proposer un défi supplémentaire aux joueurs expérimentés.

Le projet est structuré selon les principes de la programmation orientée objet (POO), garantissant clarté dans la programmation. Il s'appuie sur plusieurs classes principales :

- `WelcomeScreen`, qui gère l'écran d'accueil et les options de navigation,
- `GuessingGame`, où se trouve la logique centrale du jeu,
- `GameScreen`, qui représente l'interface principale et les interactions avec le joueur,
- `PlayerUser`, qui stocke et gère les données du joueur, telles que le pseudonyme et le score.

Enfin, la gestion des fichiers joue un rôle clé en permettant de sauvegarder et de restaurer les données des joueurs ainsi que l'état des parties, assurant ainsi une continuité dans l'expérience de jeu.

2 Prérequis pour lancer le projet

Avant de lancer ce projet, assurez-vous de disposer des éléments suivants :

2.1 Environnement Logiciel

- **Java Development Kit (JDK)** : Version 21.
 - Vérifiez que `java` et `javac` sont configurés dans votre `PATH`.
- **Java Swing** : Inclus par défaut dans le JDK.

2.2 Exécution

2.2.1. Compilation :

Depuis la ligne de commande, placez-vous dans le répertoire contenant les fichiers `.java`.

Compilez le projet avec la commande suivante : `javac Main.java`

Exécutez le projet avec la commande suivante : `java Main`

3 Diagrammes UML

Le diagramme présenté ci-dessous illustre les relations entre les principales classes du projet. Il met en avant les interactions entre les différents composants de l'application, tout en respectant les principes de programmation orientée objet.

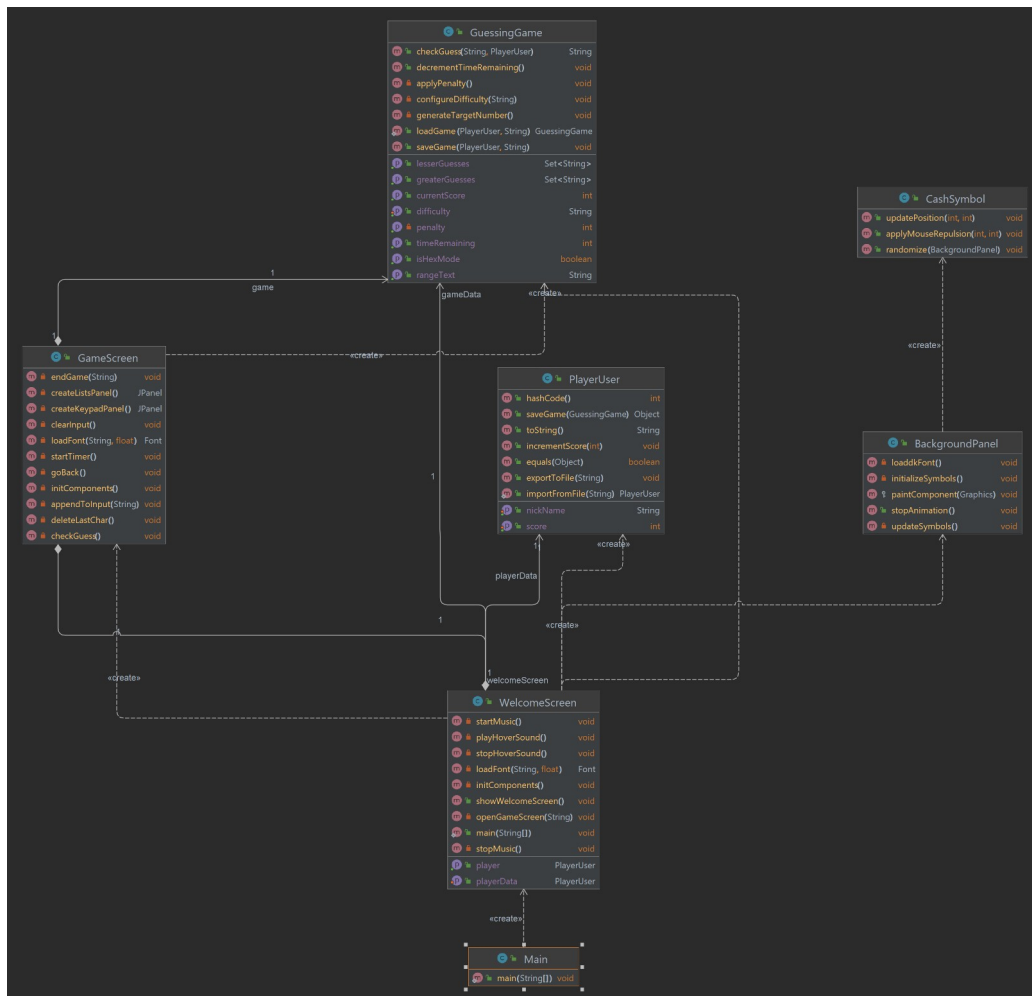


Figure 1: Diagramme UML de Juste Prix

- **Main** : C'est le point d'entrée de l'application. Cette classe initialise le jeu en lançant l'écran d'accueil, représenté par la classe `WelcomeScreen`.
- **WelcomeScreen** : Elle gère l'interface d'accueil du jeu, où l'utilisateur peut choisir de démarrer une nouvelle partie, charger une sauvegarde ou quitter l'application.
- **GameScreen** : Cette classe représente l'interface principale utilisée par le joueur pendant une partie. Elle affiche des informations essentielles comme le score, les devinettes déjà effectuées et le temps restant.

- **GuessingGame** : Elle contient la logique centrale du jeu, notamment la génération du nombre mystère, la gestion des suppositions du joueur et le calcul du score.
- **PlayerUser** : Cette classe stocke et gère les données du joueur, telles que son pseudonyme et son score. Elle offre également la possibilité de sauvegarder ou de restaurer ces informations.
- **CashSymbol** et **BackgroundPanel** : Ces deux classes sont dédiées aux animations graphiques.

La classe `WelcomeScreen` est responsable de créer les instances de `PlayerUser` et `GuessingGame` nécessaires pour initialiser une nouvelle partie.

La classe `GameScreen` interagit directement avec `GuessingGame` pour récupérer et afficher des informations comme le score actuel, les suppositions et le temps restant.

Les classes `BackgroundPanel` et `CashSymbol` enrichissent l'apparence visuelle de l'application en ajoutant des animations et des éléments graphiques dynamiques.

4 Fonctionnalités implémentées

Le projet intègre plusieurs fonctionnalités, chacune réparties entre les différentes classes, pour offrir une expérience de jeu ludique. Voici un aperçu des éléments principaux :

3.1 Écran d'accueil (WelcomeScreen)

L'écran d'accueil propose une interface qui permet au joueur de configurer et démarrer une partie selon ses préférences. Les options disponibles incluent un choix parmi 4 niveaux de difficulté :

- **Facile** : Deviner un nombre entre 1 et 10
- **Moyen** : Deviner un nombre entre 1 et 100
- **Difficile** : Deviner un nombre entre -100 et 100
- **Extrême** : Deviner un nombre entre 1 et 255, en mode hexadécimal, pour les joueurs expérimentés
- Affichage des règles du jeu,
- Options de personnalisation comme la gestion du pseudonyme et du son de fond.
- Boutons interactifs pour sauvegarder ou charger une partie, ainsi que quitter le jeu

3.2 Logique du Jeu (GuessingGame)

La mécanique du jeu repose sur une logique bien définie et adaptée à chaque niveau de difficulté :

- Génération aléatoire du nombre mystère en fonction de la plage correspondant à la difficulté choisie
- **Gestion des suppositions** :
 - Le joueur reçoit des retours immédiats après chaque tentative :
 - *"Trop petit"*,

- *Trop grand*,
- ou *"Correct !"* si la réponse est trouvée.
- Les suppositions incorrectes sont classées dans des listes distinctes :
 - Les nombres plus petits que la solution
 - Et ceux plus grands
- **Système de score :**
 - Un score initial est attribué selon le niveau de difficulté (entre 100 et 500 points).
 - Chaque erreur entraîne une réduction du score, avec des pénalités adaptées.

3.3 Interface du Jeu (GameScreen)

L'interface principale permet au joueur d'interagir directement avec le jeu et d'accéder à des informations en temps réel :

- **Affichage des données :**
 - Le nombre de suppositions déjà effectuées,
 - Le score actuel,
 - Et un chronomètre indiquant le temps restant pour terminer la partie (90 secondes par défaut).
- **Saisie des réponses :**
 - Via un pavé numérique graphique,
 - Ou directement au clavier pour plus de rapidité.
- **Gestion des parties :**
 - Option pour sauvegarder une partie en cours,
 - Possibilité de retourner à l'écran d'accueil.

3.4 Gestion des Joueurs (PlayerUser)

Le jeu inclut un système de gestion des joueurs pour personnaliser l'expérience :

- **Exportation et importation des données :**
 - Les pseudonymes et scores sont sauvegardés dans des fichiers, permettant de reprendre une partie à tout moment.
- **Personnalisation individuelle :**
 - Chaque joueur peut configurer son pseudonyme et suivre ses performances de manière unique.

5 Difficultés et fonctionnalités non implémentées

Au cours du développement du projet, plusieurs défis ont été rencontrés et certaines fonctionnalités prévues initialement n'ont pas pu être intégrées dans cette version. Voici un résumé des points importants :

5.1 Difficultés rencontrée

- **Gestion des sons:**
 - L'intégration des effets sonores a été plus complexe que prévu, notamment pour éviter que plusieurs sons se superposent lors d'interactions rapides. Il a fallu ajuster et synchroniser leur déclenchement.
- **Gestion des fichiers:**
 - Assurer la compatibilité entre les fichiers de sauvegarde et l'état actuel du jeu a demandé une réflexion approfondie. Il a fallu concevoir un format structuré pour éviter des erreurs lors du chargement des parties ou des données des joueurs.

- **Équilibre des pénalités :**
 - Trouver le juste équilibre dans les pénalités infligées après une mauvaise supposition s'est révélé être un défi. L'objectif était de maintenir un niveau de difficulté stimulant tout en évitant de rendre le jeu frustrant ou trop indulgent.

5.2 Fonctionnalités non implémentées

- **Classement global des scores :**
 - Bien que les scores des joueurs soient enregistrés et exportables, un système permettant d'afficher un classement global ou un tableau des meilleurs scores n'a pas été intégré dans cette version. Cela pourrait être une amélioration intéressante pour renforcer l'aspect compétitif du jeu.
- **Mode multijoueur :**
 - Le jeu est actuellement conçu pour un seul joueur par partie. Un mode multijoueur permettant à plusieurs utilisateurs de jouer à tour de rôle ou de s'affronter pourrait enrichir l'expérience.

Conclusion

Le projet a abouti à une application fonctionnelle combinant une logique de jeu, une interface graphique et une gestion des données des joueurs.

Les fonctionnalités principales répondent aux objectifs initiaux, offrant une expérience ludique. Le système de score incite les joueurs à se surpasser tandis que le chronomètre ajoute une touche de défi en renforçant l'aspect compétitif.

Dans le futur, la mise en place de nouvelles fonctionnalités, telles qu'un classement global ou un mode multijoueur pourrait encore améliorer l'expérience de jeu. Le projet souligne l'avantage de Java Swing pour créer des applications interactives et organisées, tout en fournissant une fondation pour des améliorations ultérieures.

Table des figures

Figure 1: Diagramme UML de Juste Prix.....	6
--	---