Rapport de Projet : Paradigmes de Programmation

Application : Speed Math (Calcul Mental Express)

Judith HEDIBLE

Février 2026

Résumé

Ce projet consiste en la création d'une application ludo-éducative de calcul mental. L'objectif est de démontrer la mise en œuvre de différents paradigmes de programmation (POO, procédural, événementiel) à travers deux interfaces (console et graphique distinctes partageant une logique commune et une persistance de données en JSON.

Table des matières

1 Introduction

Dans le cadre de la fin du module "Paradigmes de Programmation", j'ai développé l'application **Speed Math**. Ce projet permettra de mettre en lien les différents paradigmes autour d'une application centrale. Le jeu propose des défis mathématiques de difficulté progressive. Ce rapport détaille l'architecture logicielle, les fonctionnalités de sécurité des données et les modes de déploiement.

2 Architecture et Paradigmes

2.1 Programmation Orientée Objet (POO)

La structure centrale du projet repose sur la POO, facilitant la maintenance :

- **GameEngine** : Gère la génération de calculs aléatoires et la validation des réponses.
- **ScoreManager** : Encapsule les opérations de lecture/écriture du fichier `scores.json`.
- **MusicPlayer** : Gère l'ambiance sonore via des threads pour ne pas bloquer l'interface.

Vous pouvez également visualiser cette logique sur le Diagramme UML via le fichier `.puml`.

2.2 Programmation Procédurale

La version console (`speed_math_console.py`) suit un flux linéaire et séquentiel, idéal pour comprendre l'exécution pas à pas des instructions et la gestion des entrées utilisateur basiques.

2.3 Interface Homme-Machine (Événementiel)

La version graphique utilise la bibliothèque `Tkinter`. Ici, le programme attend des "événements" (clics de souris, pressions de touches) pour déclencher des fonctions spécifiques, illustrant la programmation pilotée par les événements. Dans cette version j'ai penser à deux mode **simple** et un mode **challenge** avec un timer de 15 secondes par réponse, pour rendre le jeu un peu plus compétitive.

3 Fonctionnalités et Sécurité

3.1 Vérification d'unicité des pseudos

Une fonctionnalité clé a été ajoutée pour garantir l'intégrité du classement :

1. Au démarrage, le programme interroge le `ScoreManager` pour pouvoir récupérer la liste des scores précédents. Cela permettra d'avoir un Top 10 à jour et également la liste de tous les Joueurs.
2. Si àprès vérification le pseudo saisi existe déjà dans la base de données JSON, le jeu bloque l'accès.
3. Un message d'erreur invite l'utilisateur à choisir un identifiant unique.

3.2 Algorithme d'assistance (Indices)

Pour aider le joueur sans rendre le jeu trop facile j'ai rajouter le bouton indice. Pour les multiplications complexes (ex: $a \times b$ où $a, b \geq 10$), le moteur de jeu génère automatiquement un indice basé sur la distributivité :

$$a \times (dizaines\_b + unites\_b)$$

4 Guide d'Utilisation et Déploiement

4.1 Scripts de lancement (Batch)

Pour simplifier l'expérience utilisateur sur Windows, deux lanceurs automatisés ont été créés :

- **Lancer_Console.bat** : Exécute le script en mode terminal. Ce fichier lance la version texte dans l'invite de commande.
- **Lancer_Graphique.bat** : Lance l'interface IHM de manière autonome. Ce fichier lance la version avec fenêtre. **"pythonw"** est utilisé au lieu de **"python"** pour éviter qu'une fenêtre noire de terminal ne s'affiche derrière le jeu.

Assurez-vous que Python soit bien ajouté au "PATH" de votre ordinateur pour que la commande `python` fonctionne dans le fichier `.bat`.

4.2 Visualisation des scores (Web)

Le dossier **Leaderboard** contient un site web permettant de visualiser le Top 10 des scores de manière externe. Le fichier `index.html` permet de consulter le classement via une page HTML . En chargeant le fichier `scores.json`, une page web dynamique affiche les meilleurs joueurs.

4.3 Lancement manuel

- Les jeux peuvent être lancer dans un éditeur de fichier (vscode) sur le terminal avec `py nom_du_fichier_a_lancer.py`. Pour que la version graphique fonctionne avec la music et le timer , penser à installer la bibliothèque **Pygame** avec `py -m pip install pygame`.
- Vous pouvez aussi importer les fichier dans un idle Python.

5 Evolution futur

Pour l'avenir je pense rendre le jeu un peu plus compétitif avec le système multijoueurs. Aussi la mise en plce d'une interface (site) et d'une BDD pour pouvoir le top 10 à jour chez tous les joueurs.

6 Conclusion

Ce projet a permis d'explorer la complémentarité des paradigmes de programmation. La séparation entre la logique métier et les interfaces permet d'offrir une expérience utilisateur variée tout en conservant une base de code unique et solide.