

Table des matières

1

Introduction

1

1.1

A quoi sert ce logiciel

1

1.2

Comment l'utiliser

1

1.3

Architecture

1

1.3.1

Modules principaux

2

2

Fonctionnalités

3

2.1

Vue d'ensemble de l'interface

3

2.2

Création d'une nouvelle série

3

2.2.1

Saisie manuelle

3

2.2.2

Génération assistée par IA

3

2.3

Chargement de questions existantes

4

2.3.1

Charger un modèle

4

2.4

Sauvegarde et compilation

4

3

Utilisation personnalisée

5

3.1

Modifier le modèle

5

3.2

Configuration du chemin des images

5

3.3

Modifier les prompts

5

3.4

Configuration de l'assistant IA

6

3.4.1

Choix du provider

6

3.4.2

Configuration des clés API

6

3.4.3

Avantages de chaque provider

6

2.4.1

Sauvegarde dans la base de données

4

2.4.2

Génération LaTeX et compilation

5

2.5

Création d'évaluations

5

1. Introduction

1.1 A quoi sert ce logiciel

Ce logiciel est un **gestionnaire de questions Flash**.
Il permet de :

1. Créer des fichiers « \LaTeX » basés sur un **modèle** adapté pour la **présentation en classe**.
2. Organiser ces fichiers dans un répertoire, et permettre un accès facile à l'enseignant.
3. Charger des questions prédéfinies par l'application « `implémenter_enonce` ».
4. Compiler les documents et les organiser par année - mois - date prévue de projection.
5. Construire des évaluations par choix aléatoire de questions durant une période donnée.

Cela permet une gestion fine des projections d'automatismes en classe.
L'enseignant peut simplement **choisir** dans un corpus de modèles, les questions qu'il souhaite projeter.
Cela permet d'**adapter** les questions projetées au niveau de la classe et aux **objectifs** de l'enseignant.

1.2 Comment l'utiliser

Dans le répertoire courant de l'application, toutes les manipulations à effectuer sont données dans le fichier **README.md**
Des **scripts d'installation et de lancement** ont été prévues, et nécessitent une distribution **Python** comportant la librairie **tcl** binaire.
Tester avec votre distribution, et si ça ne fonctionne pas, demander à un agent IA de vous dire quelle version télécharger pour être compatible.

Le projet s'organise selon l'arborescence suivante :

```

NEW_QF_generator/
├── databases/
│   ├── database_6eme/
│   │   ├── database_2025_1.json
│   │   ├── database_2025_2.json
│   │   └── ...
│   ├── database_5eme/
│   ├── database_4eme/
│   └── ... (un dossier par niveau)
├── generated/
│   ├── tex/ (fichiers  $\LaTeX$  générés)
│   │   ├── 6ème/
│   │   │   ├── 2025/
│   │   │   │   ├── 09/
│   │   │   │   └── 10/
│   ├── pdf/ (PDF compilés)
│   ├── eval_tex/ (évaluations  $\LaTeX$ )
│   └── eval_pdf/ (évaluations PDF)
├── qf_gen_config/
│   ├── api_key.txt (clés API)
│   ├── ai_provider.txt (choix IA)
│   ├── prompts.txt (prompts IA)
│   └── cfg.py (configuration)
├── latexhighlighter/ (coloration syntaxique)
├── UI_question_loader/ (interface de chargement)
├── main.py (interface principale)
├── auto_question_maker.py (génération IA)
├── modele_QF.tex (modèle questions)
└── Modele_QF_eval.tex (modèle évaluations)

```

1.3.1 Modules principaux

1. **main.py** : Interface graphique principale (CustomTkinter). Gère les 3 questions par série, la sélection de classe/date, la compilation LaTeX.
2. **auto_question_maker.py** : Module de génération automatique via IA (OpenAI ou Claude). Permet de générer question, réponse détaillée, réponse courte et thème.
3. **question_manager.py** : Gestion de la base de données JSON des questions par niveau et par période.
4. **latexcompiler.py** : Compilation LaTeX avec LuaLaTeX, gestion des fichiers auxiliaires et ouverture automatique des PDF
5. **latexhighlighter/** : Coloration syntaxique en temps réel, vérification de syntaxe LaTeX, et visualisation PDF intégrée.
6. **UI_question_loader/** : Interface de navigation dans les bases de données pour charger des questions existantes comme modèles.
7. **tex_eval_maker.py** : Génération d'évaluations par sélection aléatoire de questions sur une période donnée.

2. Fonctionnalités

2.1 Vue d'ensemble de l'interface

L'interface principale permet de gérer jusqu'à 3 questions par série de questions flash.

The interface shows a top navigation bar with filters for 'Classe ciblée' (2nde), 'Date' (03_10_2025), and 'ID Question' (QF_04_10_2025). Below are buttons for 'Générer LaTeX', 'Sauvegarder les modifications', 'Créer une nouvelle série', 'Charger un contenu précédent', 'Créer une évaluation', and 'Ouvrir le répertoire des questions Flash'. The main area contains three rows for question creation. Each row has a 'Prompt' field with a text area and 'Appeler l'assistant'/'Charger un modèle' buttons, followed by 'Question', 'Réponse', 'Détails', and 'Thème' fields. The first row is populated with LaTeX code and text, while the others are empty.

FIGURE 1 : Interface principale du logiciel

Pour chaque question, l'utilisateur dispose de :

1. Un champ pour l'**énoncé** de la question (LaTeX)
2. Un champ pour la **réponse courte** (affichée en bas de document)
3. Un champ pour la **réponse détaillée** (correction complète)
4. Un champ pour le **thème** de la question
5. Un champ **prompt** pour guider l'assistant IA

2.2 Création d'une nouvelle série

2.2.1 Saisie manuelle

L'utilisateur peut saisir directement le code LaTeX dans les champs prévus. La coloration syntaxique aide à repérer les erreurs de syntaxe.

This screenshot shows the same interface as Figure 1, but with the first row filled with content. The 'Prompt' field contains 'Un exercice sur la résolution d'équation de premier degré $ax+bx=cx+d$ '. The 'Question' field contains LaTeX code for a problem statement. The 'Réponse' field contains a LaTeX fraction. The 'Détails' field contains a detailed explanation in French. The 'Thème' field contains the word 'Équations'.

FIGURE 2 : Exemple de saisie d'une question sur les équations

2.2.2 Génération assistée par IA

En cliquant sur « Appeler l'assistant », le logiciel utilise l'IA configurée (OpenAI ou Claude) pour :

1. Générer l'énoncé à partir du prompt
2. Générer la réponse détaillée
3. Générer la réponse courte
4. Proposer un thème pour la question

Le résultat s'affiche en **temps réel** (streaming) dans les champs correspondants.

2.3 Chargement de questions existantes

2.3.1 Charger un modèle

Le bouton « Charger un modèle » permet de naviguer dans la base de données des questions déjà créées.

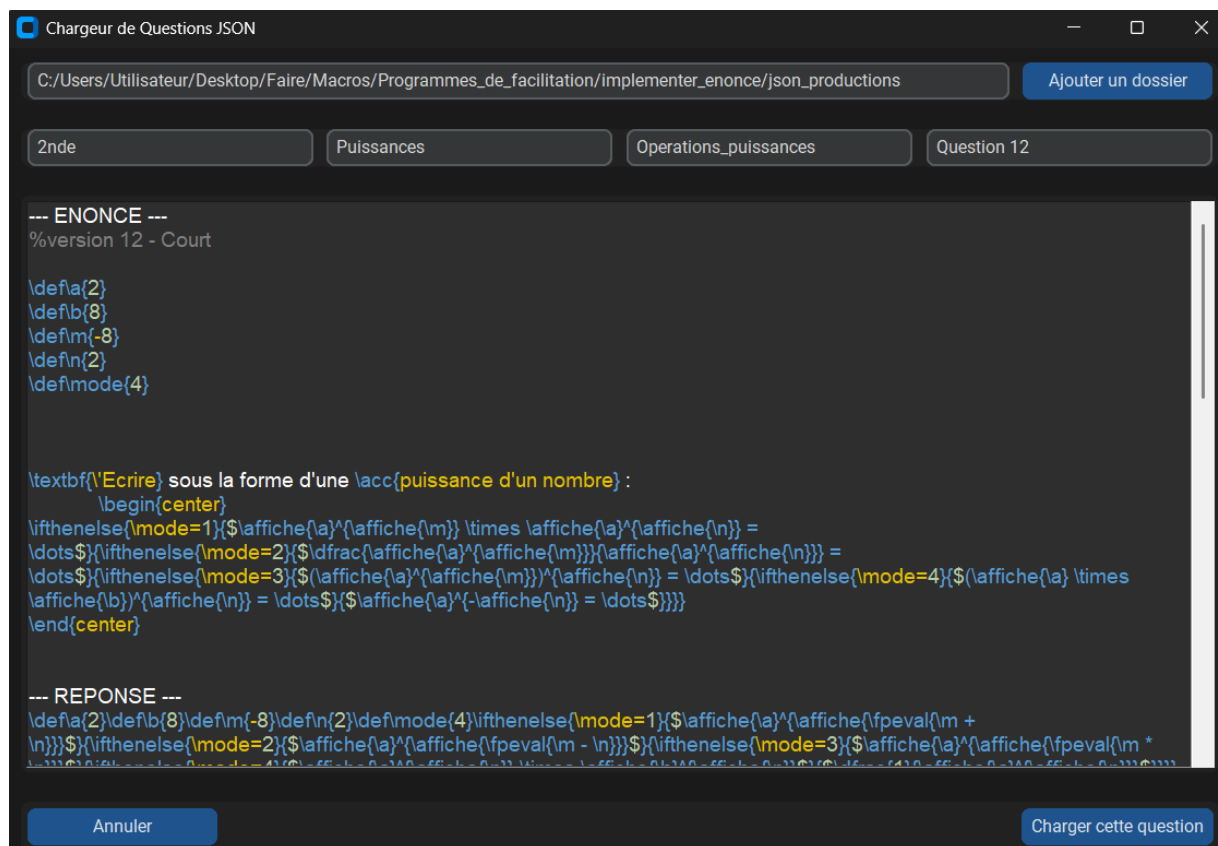


FIGURE 3 : Interface de chargement de questions existantes

L'utilisateur peut :

1. Parcourir les questions par **niveau** (6ème, 5ème, etc.)
2. Filtrer par **thème**
3. Prévisualiser les questions avant de les charger
4. Charger une question comme base pour une nouvelle

2.4 Sauvegarde et compilation

2.4.1 Sauvegarde dans la base de données

Le bouton « Sauvegarder les modifications » enregistre les 3 questions dans la base de données JSON correspondant au niveau et à la date sélectionnés.

Format : databases/database_{classe}/database_{année}_{mois}.json

2.4.2 Génération LaTeX et compilation

Le bouton « Générer LaTeX » :

1. Crée le fichier `.tex` à partir du modèle `modele_QF.tex`
2. Le place dans `generated/tex/{classe}/{année}/{mois}/`
3. Lance la compilation avec LuaLaTeX
4. Place le PDF dans `generated/pdf/{classe}/{année}/{mois}/`
5. Ouvre automatiquement le PDF généré

2.5 Création d'évaluations

Le bouton « Créer une évaluation » permet de générer une évaluation par sélection aléatoire :

1. Sélectionner le **niveau** concerné
2. Définir une **période** (date min - date max)
3. Choisir le **nombre de questions** à inclure
4. Le logiciel sélectionne aléatoirement des questions dans cette période
5. Génère un document d'évaluation avec le barème de points

Les évaluations sont enregistrées dans `generated/eval_tex/` et `generated/eval_pdf/`.

3. Utilisation personnalisée

3.1 Modifier le modèle

Il est possible de modifier le fichier « `modele_QF.tex` » ou « `Modele_QF_eval.tex` » pour adapter à votre présentation ou utiliser vos propres packages pour la génération des documents.

3.2 Configuration du chemin des images

Pour que les images s'affichent correctement dans vos documents, vous devez configurer le chemin vers le dossier `images/` dans votre entête de modèles.

Vous pouvez : Utiliser le paramètre `bfcours dbIconPath` :

```
\dbIconPath {C:/chemin/vers/vos/images/}
```

Remarque : Utilisez des barres obliques / (slash) plutôt que des antislash \ pour la compatibilité LaTeX, même sous Windows.

Vérification :

Pour vérifier que le chemin est correctement configuré, compilez un document contenant une image de test. Si l'image ne s'affiche pas, vérifiez :

1. Que le chemin est absolu (commence par `C:/` sous Windows)
2. Que les barres obliques sont utilisées
3. Que le dossier `images/` existe bien à l'emplacement indiqué

3.3 Modifier les prompts

Les prompts commencent à être un peu anciens, mais comme ils fonctionnent, je ne les modifie pas. Vous pouvez tout à fait, à vos risques et périls, les modifier pour personnaliser l'affichage.

Le logiciel supporte deux fournisseurs d'IA : **OpenAI** et **Claude (Anthropic)**.

3.4.1 Choix du provider

Pour changer de provider IA, modifiez le fichier `qf_gen_config/ai_provider.txt` :

`claude`

ou

`openai`

Par défaut, le logiciel utilise **Claude** avec le modèle `claude-sonnet-4-5-20250929`.

La configuration pour **openai** utilise le modèle `gpt-5-2025-08-07`

3.4.2 Configuration des clés API

Les clés API sont stockées dans `qf_gen_config/api_key.txt` au format :

`OPENAI_API_KEY="sk-proj-xxxxx ..."`

`ANTHROPIC_API_KEY="sk-ant-api03-xxxxx ..."`

Conservez les guillemets autour des clés.

3.4.3 Avantages de chaque provider

Claude :

1. Excellent pour les mathématiques.
2. Meilleure compréhension du LaTeX.
3. Réponses plus structurées.

OpenAI :

1. Intelligence améliorée (modèle frontière).
2. Génération de qualité.