

# REPORT – TYPST PROJECT

## INTRODUCTION

Our project aims to develop an innovative tool dedicated to composing personalized documents from various sub-documents. This software, based on the document composition technology of Typst, will be specially designed to meet the diverse needs of educators, particularly in the field of exercise sheet editing.

## CONTEXT AND PROJECT OBJECTIVE

Imagine a teacher with a rich set of exercises including solutions and various levels of hints. Our goal is to simplify the process of creating customized exercise sheets, tailored to a specific audience and intended to be converted into PDF documents and/or integrated into a website.

We will use Typst, an advanced document composition software, to accomplish this project, along with the Python programming language.

The objective of the project is thus to be able to create fully customizable exercise sheets from different exercise files contained in a database. The user can then select the desired exercises, supplement the database with their own files, and create their exercise sheets with the option to modify the headers if desired.

## APPROACH

As the first group assigned to this project, we decided to provide a robust codebase and architecture to facilitate the handover of the project to subsequent years.

Therefore, we started by implementing a database consisting of exercises in Typst and LaTeX formats, formatted simply to approximate the syntax of examples provided by a teacher and to be parsed correctly by our program. We then implemented Exercise and Sheet classes, as well as an automaton, which we will discuss in more detail in the following section.

One possible outcome for creating an exercise sheet (based on the provided data) is as follows:

## Ma Feuille d'exercices



**Author : Moi-même - Date : 25 Mars 2024 - Duration : 3 heures**

Sont interdits : les documents, les ordinateurs, les téléphones (incluant smartphone, tablettes,... tout ce qui contient un dispositif électronique). Seuls les dictionnaires papier pour les personnes de langue étrangère sont autorisés. En cas de doutes sur l'énoncé, préciser les choix que vous faites sur votre copie et continuer. Il sera tenu compte de la qualité de la rédaction et de la clarté de la présentation (2 pts). Le barème indicatif : Exercice I : 5 points ; Exercice II : 5 points ; Problème III : 8 points Les exercices et le problème peuvent être traités indépendamment. Les durées sont indicatives, penser à se relire.

### Exercise : Integration Exercise

(~ 30min) | 20pts

Write an algorithm to find the maximum element in an array of integers.

**Solution :**

```
def find_max_element(arr):
    # Initialize max_element with the first element of the array
    max_element = arr[0]

    # Iterate through the array starting from the second element
    for num in arr[1:]:
        # Update max_element if the current element is greater
        if num > max_element:
            max_element = num

    # Return the maximum element
    return max_element

# Example usage:
array = [3, 5, 2, 9, 10, 7, 1]
print("Maximum element in the array:", find_max_element(array))
```

### Exercise : Addition Exercise (Bonus)

(~ 30min) | 10pts

Calculate the sum of the following numbers :

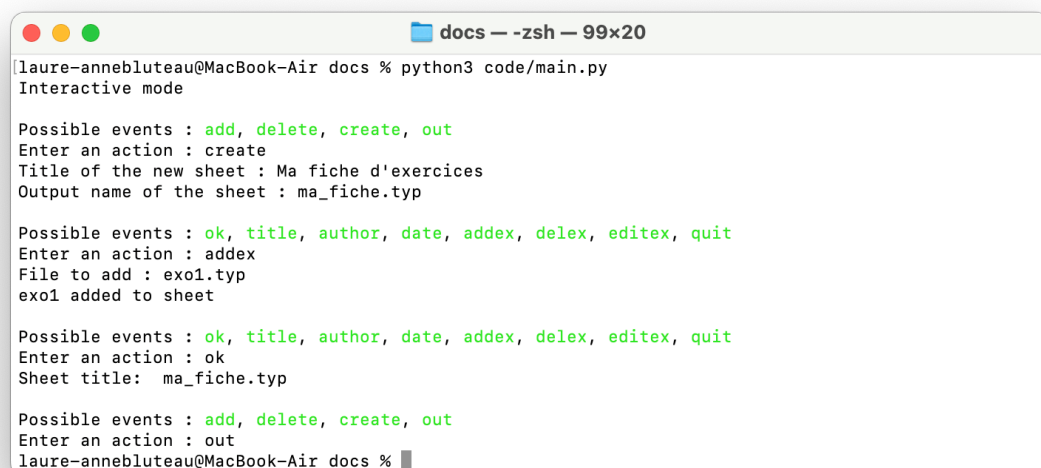
```
let numbers = [5, 8, 12, 3]
let sum = 0
for n in numbers {
```

## CHOICES MADE

To achieve the above result, we made several design choices:

- All exercices are formatted in the same way (whether in Typst or LaTeX) to ensure correct parsing and to enhance modularity for future needs.

- All exercises are parsed and converted into Exercise objects, allowing us to retrieve their metadata (such as visible headers) and content. Creating an Exercise object has several advantages, including modularity - we can choose to make certain elements of the header visible or not, and it allows for a single processing step regardless of the original extension (.typ or .tex).
- Upon the creation of each sheet, a Sheet object is created. This object allows us to assign a title and headers to the sheet, convert exercise content into Typst format, format its display (in the ToTypst function, and with the utilities.typ file), modify its list of exercises, etc.
- An automaton has been implemented to make the creation process more user-friendly. The user has the option to create their sheet in user mode :



```
docs — -zsh — 99x20
[laure-annebluteau@MacBook-Air docs % python3 code/main.py
Interactive mode

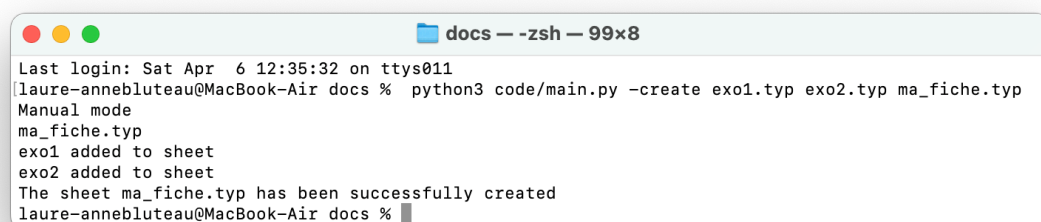
Possible events : add, delete, create, out
Enter an action : create
Title of the new sheet : Ma fiche d'exercices
Output name of the sheet : ma_fiche.typ

Possible events : ok, title, author, date, addex, delex, editex, quit
Enter an action : addex
File to add : exo1.typ
exo1 added to sheet

Possible events : ok, title, author, date, addex, delex, editex, quit
Enter an action : ok
Sheet title: ma_fiche.typ

Possible events : add, delete, create, out
Enter an action : out
laure-annebluteau@MacBook-Air docs %
```

or via command line interface (CLI):

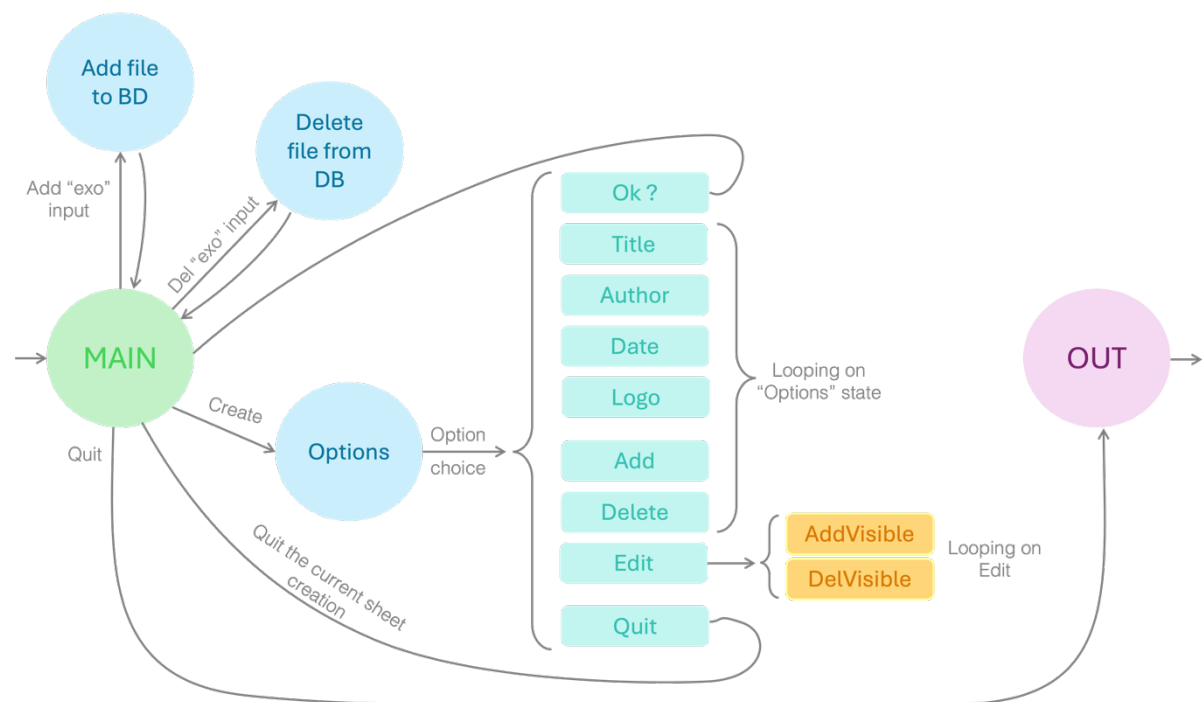


```
docs — -zsh — 99x8
Last login: Sat Apr  6 12:35:32 on ttys011
[laure-annebluteau@MacBook-Air docs % python3 code/main.py -create exo1.typ exo2.typ ma_fiche.typ ]
Manual mode
ma_fiche.typ
exo1 added to sheet
exo2 added to sheet
The sheet ma_fiche.typ has been successfully created
laure-annebluteau@MacBook-Air docs %
```

Note: The user does not need to add the full path to the file, but simply its name, ensuring in advance that it is present in the database and in the correct directory according to its extension.

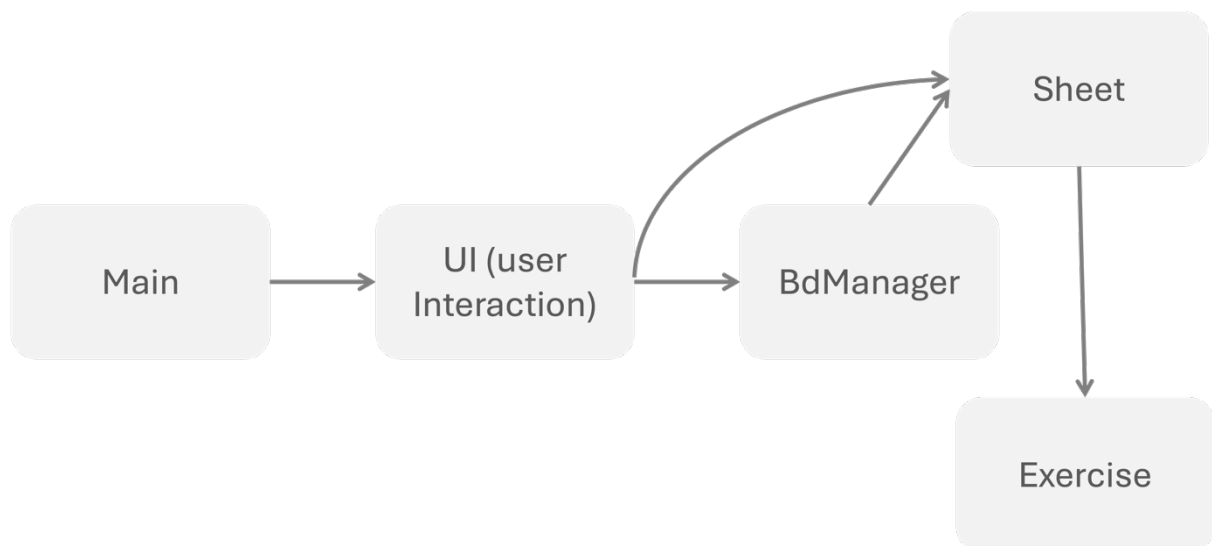
The advantage of the automaton is to make the program usage simpler and more understandable for the user, but also to make the project more modular for future years by easily adding new options. This automaton can also serve as a good foundation for implementing a graphical user interface.

The automaton is as follows:



## ARCHITECTURE AND OTHER DIAGRAMS

The code is structured as follows:



## FUTURE IMPROVEMENTS

At this stage, the program does not yet allow us to convert the content of a LaTeX exercise into Typst, only its metadata is parsed. Implementing a LaTeX to Typst converter would therefore be considered for the project's evolution.

Additionally, we have not yet managed the inclusion of figures in our code for Typst files. If an image is present in an exercise, the complete path must be provided for it to be displayed.

Finally, implementing a user interface could also be part of future improvements, making the program more user-friendly.

---

[ FRENCH VERSION BELOW ]

# RAPPORT – PROJET TYPST

## INTRODUCTION

Notre projet vise à développer un outil innovant dédié à la composition de documents personnalisés à partir de différents sous-documents. Ce logiciel, reposant sur la technologie de composition de documents Typst sera spécialement conçu pour répondre aux besoins variés des enseignants, notamment dans le domaine de l'édition de fiches d'exercices.

## CONTEXTE ET BUT DU PROJET

Imaginez un enseignant disposant d'un riche ensemble d'exercices comprenant solutions et niveaux d'indices variés. Notre objectif est de simplifier le processus d'élaboration de fiches d'exercices sur mesure, adaptées à un public spécifique et destinées à être converties en document PDF et/ou intégrées à un site web.

Nous utiliseront Typst, un logiciel de composition de documents avancés pour réaliser ce projet, ainsi que le langage de programmation Python.

Le but du projet est donc de pouvoir créer des fiches d'exercices entièrement modulables à partir de différents fichiers exercices contenus dans une base de données. L'utilisateur peut alors sélectionner les exercices souhaités, intégrer la base de données avec ses propres fichiers, et confectionner ses feuilles d'exercices en ayant la possibilité de modifier les entêtes s'il le souhaite.

## ANGLE D'ATTAQUE

En tant que premier groupe affecté à ce projet, nous avons décidé de fournir une base de code et d'architecture robuste pour faciliter la transmission du projet aux années suivantes.

Nous avons donc commencé par implémenter une base de données constituées d'exercices au format Typst et LaTeX, formatés simplement de façon à s'approcher de la syntaxe d'exemples ayant été fournis par un professeur, et à être parsés

correctement par notre programme. Nous avons ensuite implémenté des classes Exercices et Fiches, ainsi qu'un automate, que nous verrons plus en détails dans la partie suivante.

Un résultat pouvant être obtenu pour la création d'une fiche d'exercices (en fonction des données fournies) est le suivant :

## Ma Feuille d'exercices



**Author : Moi-même - Date : 25 Mars 2024 - Duration : 3 heures**

Sont interdits : les documents, les ordinateurs, les téléphones (incluant smartphone, tablettes,... tout ce qui contient un dispositif électronique). Seuls les dictionnaires papier pour les personnes de langue étrangère sont autorisés. En cas de doutes sur l'énoncé, préciser les choix que vous faites sur votre copie et continuer. Il sera tenu compte de la qualité de la rédaction et de la clarté de la présentation (2 pts). Le barème indicatif : Exercice I : 5 points ; Exercice II : 5 points ; Problème III : 8 points Les exercices et le problème peuvent être traités indépendamment. Les durées sont indicatives, penser à se relire.

### Exercise : Integration Exercise

(~ 30min) | 20pts

Write an algorithm to find the maximum element in an array of integers.

#### Solution :

```
def find_max_element(arr):
    # Initialize max_element with the first element of the array
    max_element = arr[0]

    # Iterate through the array starting from the second element
    for num in arr[1:]:
        # Update max_element if the current element is greater
        if num > max_element:
            max_element = num

    # Return the maximum element
    return max_element

# Example usage:
array = [3, 5, 2, 9, 10, 7, 1]
print("Maximum element in the array:", find_max_element(array))
```

### Exercise : Addition Exercise (Bonus)

(~ 30min) | 10pts

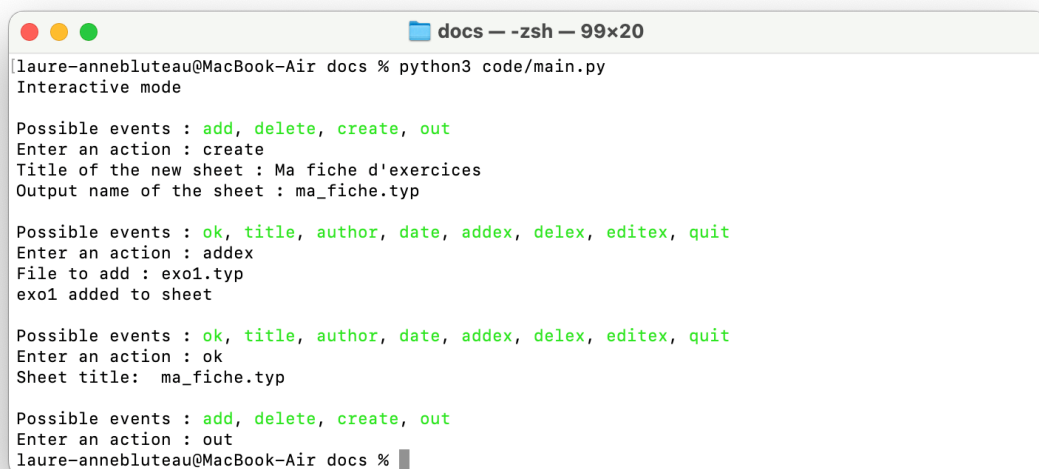
Calculate the sum of the following numbers :

```
let numbers = [5, 8, 12, 3]
let sum = 0
for n in numbers {
```

## CHOIX EFFECTUÉS

Pour parvenir au résultat ci-dessus, nous avons effectués plusieurs choix de conception.

- Les exercices sont tous formatés de la même façon (selon qu'ils soient en Typst ou en LaTeX) pour pouvoir être parsés correctement, et pouvoir être plus modulables pour les besoins futurs.
- Les exercices sont tous parsés et convertis en objets Exercice, afin de pouvoir récupérer leurs métadonnées (entêtes visibles ou non) et leur contenu. La création d'un objet Exercice a plusieurs avantages, notamment la modularité – on peut choisir de mettre en visible ou non des éléments de l'entête par exemple, mais permet également de n'avoir qu'un seul traitement, qu'elle que soit l'extension d'origine (.typ ou .tex).
- À chaque création de feuille, un objet Fiche est créé. Cet objet nous permet de donner à la fiche son propre titre et ses propres entêtes, de convertir le contenu des exercices en Typst, de formater son affichage (dans la fonction ToTypst, et avec le fichier utilities.typ), de modifier sa liste d'exercices, etc...
- Un automate a été implémenté pour rendre le processus de création plus accessible pour l'utilisateur. L'utilisateur a alors la possibilité de créer sa fiche en mode utilisateur :



```
docs — zsh — 99x20
[laure-annebluteau@MacBook-Air docs % python3 code/main.py]
Interactive mode

Possible events : add, delete, create, out
Enter an action : create
Title of the new sheet : Ma fiche d'exercices
Output name of the sheet : ma_fiche.typ

Possible events : ok, title, author, date, addex, delex, editex, quit
Enter an action : addex
File to add : exo1.typ
exo1 added to sheet

Possible events : ok, title, author, date, addex, delex, editex, quit
Enter an action : ok
Sheet title: ma_fiche.typ

Possible events : add, delete, create, out
Enter an action : out
laure-annebluteau@MacBook-Air docs %
```



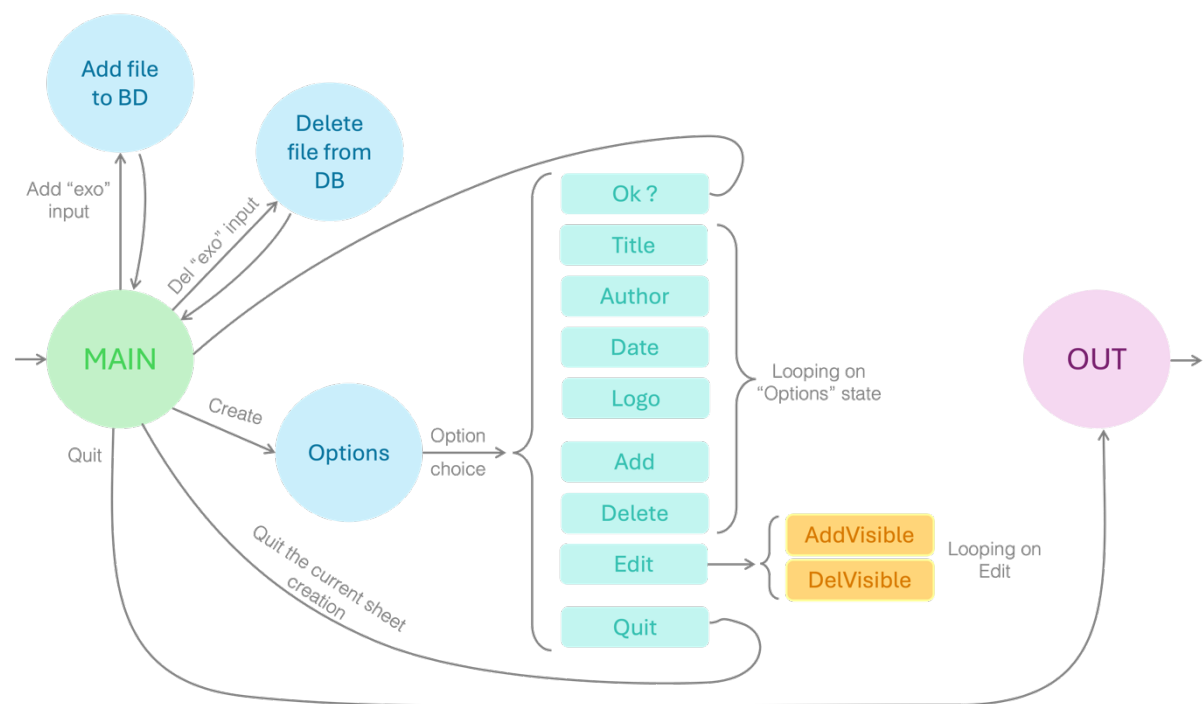
ou en ligne de commande :

```
docs — zsh — 99x8
Last login: Sat Apr 6 12:35:32 on ttys011
[laure-annebluteau@MacBook-Air docs % python3 code/main.py -create exo1.typ exo2.typ ma_fiche.typ ]
Manual mode
ma_fiche.typ
exo1 added to sheet
exo2 added to sheet
The sheet ma_fiche.typ has been successfully created
laure-annebluteau@MacBook-Air docs %
```

*Remarque : l'utilisateur n'a pas besoin d'ajouter le chemin complet jusqu'au fichier, mais simplement son nom, en s'assurant à l'avance que celui-ci soit bien présent dans la base de données, et dans le bon répertoire en fonction de son extension.*

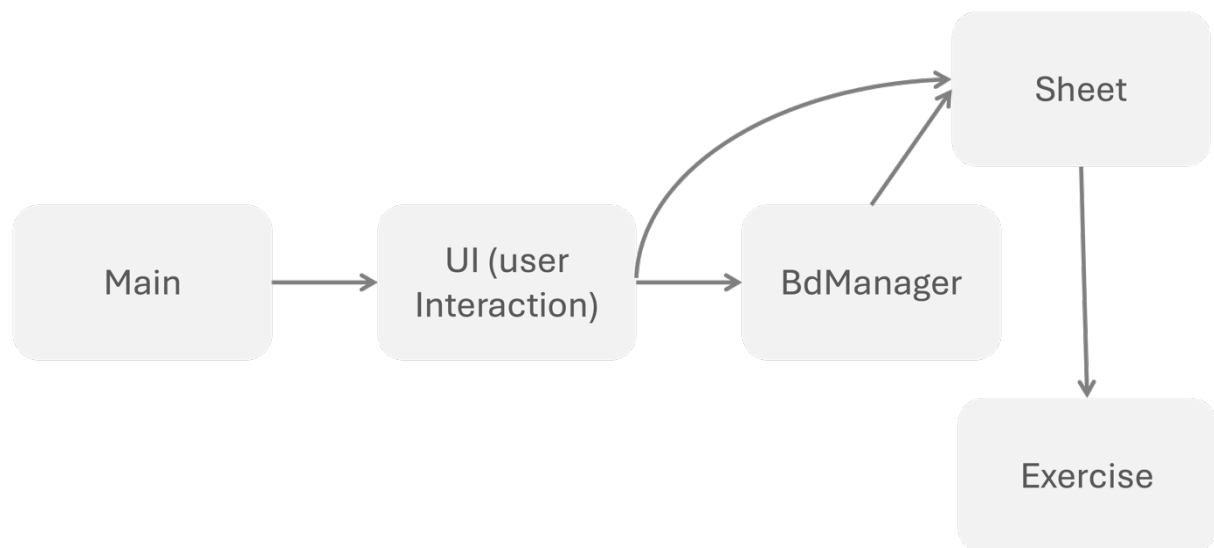
L'avantage de l'automate est de rendre l'utilisation du programme plus simple et compréhensible pour l'utilisateur, mais également de rendre le projet plus modulable pour les années suivantes en ayant la possibilité d'ajouter facilement de nouvelles options. Cet automate peut aussi représenter une bonne base pour l'implémentation d'une interface graphique.

L'automate est le suivant :



## ARCHITECTURE ET AUTRES DIAGRAMMES

Le code se structure de la façon suivante :



## AMÉLIORATIONS FUTURES

À ce stade, le programme ne nous permet pas encore de convertir en Typst le contenu d'un exercice en LaTeX, seules ses métadonnées sont parsées. L'implémentation d'un convertisseur LaTeX vers Typst serait donc à envisager pour l'évolution du projet.

Nous n'avons également pas encore géré la prise en compte des figures dans notre code pour les fichiers Typst. Si une image est présente dans un exercice, il faudra donner son chemin complet pour que celle-ci soit affichée.

Enfin, l'implémentation d'une interface utilisateur pourrait aussi faire partie des améliorations futures, rendant l'utilisation du programme plus agréable.