

QCM Maker

This is a temporary version of the Readme.

More details about the utilisation of the software can be found in the folder `Instructions`

Introduction

The project is a tool to generate automatically MCQs, in `txt` , `docx` , `xml` and `h5p` formats.

Architecture of the project

The project is divided into several packages:

- `Classes`
- `data` , containing the file `settings.json` where the language of the interface is specified as well as the default template. It also contains the *json* files to get the translation of the whole interface in several languages. You may add other files, if you want to translate the interface in another language.
- `data_kivy` , containing the styling files of the interface, using the *Python* graphic librairy *Kivy*.
- `Question Database`
- `resources`
- `Templates` , containing the templates you want to use for the generation in `docx` format.
- `test`

It also contains the following modules:

- `MCQMaker.kv` , *Kivy* file containing the graphics of the interface, which is linked to the module `MCQMaker.py` .
- `MCQMaker.py`

Installation

Creation of a virtual environnement

If you choose to clone this repository, you might want to use a virtual environnement.

You can create one with the following command :

```
python -m virtualenv mcqmaker_venv
```

To start it, use the command on *Windows* :

```
mcqmaker_venv\Scripts\Activate.ps1
```

Or for *MacOS* and *Linux* :

```
mcqmaker_venv\Scripts\activate
```

Installation of the necessary packages

To use this software, you need several *Python* libraries specified in the `requirements.txt` file.

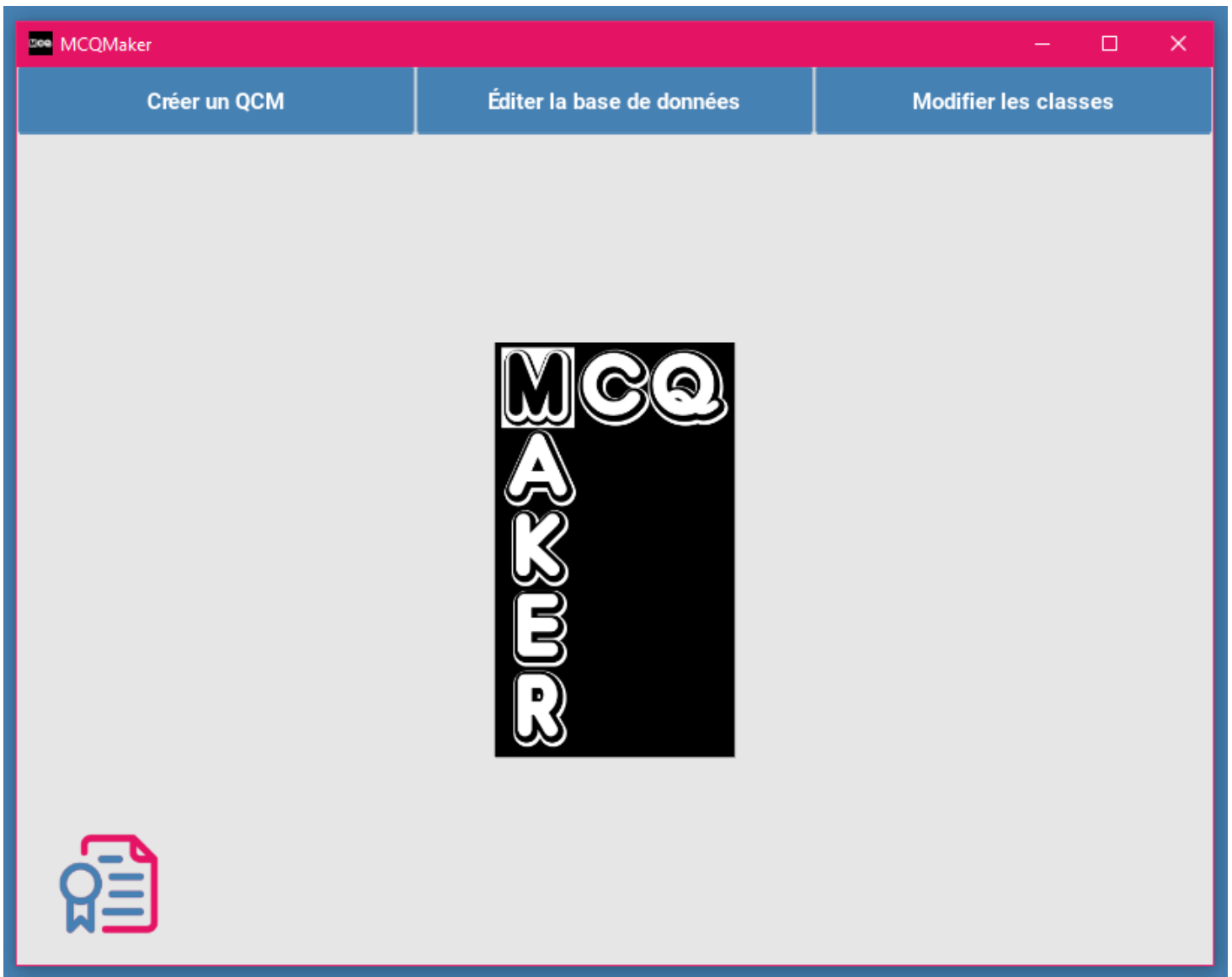
To install them, use the command :

```
pip install -r requirements.txt
```

Utilisation

After having executed the module `MCQMaker.py` , the following window is displayed. The user may then choose between three menus, thanks to the three buttons on the top of the screen:

- Create a MCQ
- Edit the database
- Modify classes



Create a MCQ

The main goal of this window is to generate a MCQ in *txt* and *docx* format

Edit the database

Modify classes

In this menu, the user may handle the data of each class and create new ones.

On the left part, the user may select a class among those contained in the folder `Classes`.

The button `Reset data of the class` allows the user to reset the questions used by a class, after having selected it. It means, that for a future MCQ generation, the corresponding class will have access the questions of the whole database.

After having selected a class, the number of left questions for each file of the database is displayed in the scroll view on the bottom of the screen.

For the creation of a new class, the user may enter its name and click on the button
Create the new class .

Development

Data structures

```
content = [
    {
        "question": str,
        "options":
            [
                "string1",
                "string2"
            ],
        "answer": int
    }
]

config = {
    "QCM_name": str,
    "questions":
        [
            {"folder_name": str, "file_name": str, "nb_questions": int},
        ],
    "template": str,
    "mix_all_questions": bool,
    "mix_among_databases": bool
}

class_data = [
    {"name_folder": str, "name_file": str,
     "used_questions": int, "total_questions": int, "list_questions_used": list}
]

QCM_data = {
    "QCM_name": str,
    "questions": [
        {"question": str, "options": list, "answer": int}
    ]
}
```