

Manuel utilisation projet

Maxime Huet

April 2024

1 Annexe : Mode d'emploi du logiciel

1.1 Pré-requis

Le logiciel utilise plusieurs outils (programmes et librairies Python) qu'il est nécessaire d'installer avant utilisation.

1.1.1 Programmes

Programme	Comment l'installer
Python	Rendez vous à l'adresse suivante : https://www.python.org/downloads Téléchargez la version de Python correspondant à votre système.
Docker	<i>Version interface</i> : Rendez vous à l'adresse suivante : https://www.docker.com/products/docker-desktop Téléchargez la version de Docker correspondant à votre système.

TABLE 1 – Programmes pré-requis à l'exécution du logiciel

1.1.2 Modules python

Modules python	Comment l'installer
Docker	Dans l'invite de commande Python, entrez la commande <code>pip install docker</code> .
Mako	Dans l'invite de commande Python, entrez la commande <code>pip install mako</code> .
Argparse	Dans l'invite de commande Python, entrez la commande <code>pip install argparse</code> .
Flask ¹	Dans l'invite de commande Python, entrez la commande <code>pip install flask</code> .

TABLE 2 – Modules python pré-requis à l'exécution du logiciel

0. Nécessaire pour la version avec interface uniquement.

1.1.3 Données sur les langages

Nous avons également créé un fichier JSON contenant les données relatives aux langages disponibles nécessaires à l'exécution du programme.

- L'extension des fichiers dans ce langage
- La chaîne de caractères correspondant à l'image Docker du langage
- La commande d'exécution du langage
- La chaîne de caractères de formatage Latex pour les insertions de code (*si existante*)

Il est possible de modifier ce fichier afin d'ajouter le support d'autres langages pour le programme.

```
"python3": {  
  "extension": ".py",  
  "docker_image": "python:3.8",  
  "execution_command": "python /app/{filename}",  
  "latex_format": "python"  
},  
"python2": {  
  "extension": ".py",  
  "docker_image": "python:2.7",  
  "execution_command": "python /app/{filename}",  
  "latex_format": "python"  
},  
"c": {  
  "extension": ".c",  
  "docker_image": "gcc:latest",  
  "execution_command": "/bin/bash -c 'gcc /app/{filename} -o /app/{filename}.out && /app/{filename}.out'",  
  "latex_format": "c"  
},
```

FIGURE 1 – Fichier de données des langages

1.1.4 Fichiers à fournir

Il est nécessaire de fournir plusieurs fichiers formatés spécifiquement pour le programme afin de générer un QCM.

Voici un exemple de fichier code à fournir :

```
def test(a):  
    b = $gen_int()$  
    print(a*b)  
_  
[CodeInsertion]  
_
```

FIGURE 2 – Fichier code

La partie "`_[CodeInsertion]_`" est l'endroit où sera inséré "l'appel du code" (voir ci-après) pour chaque question.

La partie "`$gen_int()$`" est une balise qui exécute une fonction de génération de variable (dans le but de créer des QCMs différents avec un même appel).

Voici un exemple de fichier d'appels de code à fournir :

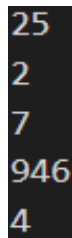
```
#Category1  
test(5)  
test(8)  
#Category2  
test(3)
```

FIGURE 3 – Fichier exécutions

La partie "`#Category1`" définit le début d'une catégorie de questions.

Chaque ligne ne commençant pas par un "`#`" représente un "appel de fonction" qui sera inséré dans le code.

Voici un exemple de fichier de réponses à fournir :



```
25
2
7
946
4
```

FIGURE 4 – Fichier exécutions

Chaque ligne représente une réponse qui sera inséré dans le code Latex du QCM.

1.2 Exécution

Nous avons développé deux versions du logiciel :

- Une version utilisant une interface en ligne de commandes
- Une version utilisant une interface graphique

Avant toute exécution, notez qu'il est nécessaire de démarrer l'application "Docker" installée plus tôt.

Il est nécessaire de fournir différentes informations au programme pour effectuer une génération de QCM, et ce, indépendamment de la version utilisée :

- Le nom de la question
- Le type de rendu (AMC ou Moodle)
- La clé du langage dans le fichier JSON d'informations sur les langages
- Le chemin d'accès du fichier contenant le code de la question
- Le chemin d'accès du fichier contenant les exécutions du code pour chaque question
- Le chemin d'accès du fichier contenant les mauvaises réponses à intégrer dans les questions
- Le type de question à générer (choix multiple ou réponse courte) (*uniquement requis pour les QCMs Moodle*)

Une fois exécuté, le programme générera les fichiers de rendu que vous pourrez télécharger directement depuis l'interface (voir figure).

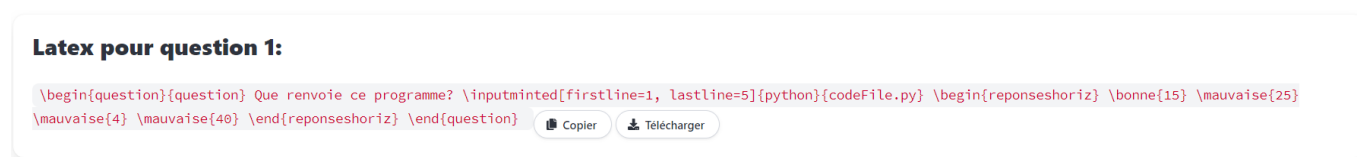


FIGURE 7 – Rendu de l'interface graphique

1.3 Fichiers de rendu

Une fois le code exécuté, comme explicité précédemment, les fichiers de rendu seront disponibles dans le dossier "Outputs" ou en téléchargement (selon la version du logiciel utilisée). Le programme génère un fichier LaTeX ainsi que des fichiers code (contenant le code à afficher pour les questions générées) formatés pour une utilisation sur les plateformes Moodle ou AMC selon les arguments donnés lors de l'exécution.

Les fichiers générés doivent être importés sur Overleaf et exportés au format `Moodle.xml` pour la version Moodle.

Pour la version AMC, il suffit d'importer les fichiers dans votre éditeur AMC.

2 Annexe : Lien du dépôt GitHub