

Rapport de stage

Génération de questionnaires Moodle et Auto-multiple choice



Institut de Recherche
en Informatique de Toulouse
CNRS - INP - UT3 - UT1 - UT2J

Entreprise : IRIT 2 rue Charles Camichel 7122 31071 Toulouse

Maître de stage : Xavier Crégut

Période de stage : 2/05/2022 au 10/06/2022

Antonin Tarrade



Sommaire

Sommaire	2
Remerciements	3
Résumé	4
Abstract	4
Introduction	5
Présentation générale du laboratoire	6
Situation géographique	6
Secteur d'activité	6
Equipe ACADIE	8
Présentation du stage	9
Espace de travail	9
Objectifs	9
Présentation des plateformes	10
Moodle	10
AMC	10
LaTeX	10
Méthode de travail	11
Présentations des activités	12
Premières réflexions	12
Approfondissement sur les plateformes Moodle et AMC	13
Moodle	13
AMC	14
Explication du programme	16
Visitor et Design pattern	17
Principe de Templates	18
Voies d'amélioration	21
Prise en compte de Wooclap	21
Amélioration de la description	21
Nouveau type de question : question portée sur un programme	22
Conclusion	24
Références bibliographiques	25
Annexes	26
Description JSON	28
Formats intermédiaires	28
Résultats	31

Remerciements

Pour débiter ce rapport, je souhaite remercier toutes les personnes importantes à la bonne réalisation de ce stage et qui m'ont apporté une aide fortement appréciable.

Un grand merci à **Xavier Crégut**, qui m'a offert la possibilité de réaliser un stage enrichissant et divertissant, mais aussi qui m'a encadré avec bienveillance tout au long de mon stage et m'a apporté une aide précieuse quant à l'orientation de mes choix techniques ainsi que pistes de travail. Mes remerciements se tournent également à toute l'**équipe ACADIE**, qui m'a remarquablement bien intégré et offert l'accès aux technologies et espaces de travail dont j'avais besoin pour la réalisation de mon travail.

Je souhaite d'autre part remercier **Valérie Bellecave**, pour m'avoir suivi tout au long du stage et avoir été là pour me guider ou répondre à d'éventuelles questions.

Enfin, je souhaite remercier **Francis Bony** ainsi que toute **La Prépa des INP de Toulouse** pour avoir intégré ce stage dans leur cursus de classe préparatoire et m'avoir ainsi permis de réaliser un tel projet.

Résumé

Ce rapport présente un compte rendu de mon stage réalisé dans le cadre de deuxième année de La Prépa des INP. Ce stage a pour but de nous préparer à une insertion professionnelle future, et pour ma part de gagner de l'expérience et des connaissances dans le domaine de l'informatique. Il s'est déroulé au sein du laboratoire de L'IRIT à Toulouse, pendant six semaines. Durant cette période, ma mission fut de créer un programme Python permettant de générer des questionnaires à choix multiples automatiquement. Plus précisément, je devais imaginer un moyen de générer plusieurs questionnaires relatifs à plusieurs plateformes d'enseignements, tel que Moodle et AMC à partir d'une description commune.

J'ai pu évoluer dans un cadre de travail agréable et nouveau, tout en étant en autonomie presque totale. Nous allons voir dans ce rapport comment j'ai pu mettre à profit mes connaissances et compétences dans la réalisation de ce projet, quelle fut mon organisation, ainsi que les grandes lignes de comment fonctionne mon programme.

Abstract

This report is about my 6 week internship in the IRIT laboratory. The purpose of this internship is to grow in a professional environment, but also to gain experience and knowledge in the computer science field. During that time, my goal was to create a Python program to generate multiple choice quizzes automatically. I had to create a way to generate multiple quizzes related to several teaching platforms, such as Moodle and AMC, from a single description. I was able to evolve in a pleasant and new work environment, while being almost totally independent. In this report , we will discuss how I was able to use my knowledge and skills in the realization of this project, what my organization was, and the main features of my program.

Introduction

La crise de la covid 19 a totalement bouleversé la manière d'enseigner. En effet, les cours en distanciel ont participé à l'émergence de plateformes d'enseignement en ligne, tel que Zoom par exemple. Ajouté à cela, d'autres plateformes déjà existantes se sont révélées des plus utiles lors des différents confinements, tel que Moodle pour pouvoir réaliser des tests en ligne. Néanmoins, à l'approche de la fin de la crise sanitaire, ces plateformes restent régulièrement utilisées, et apportent une nouvelle approche à l'enseignement. De nombreux outils permettent aujourd'hui aux éducateurs et enseignants de pouvoir mettre en place des questionnaires, que ce soit en ligne ou destinés à être imprimés. Cependant, chacun de ces outils demandent différents pré-requis, un travail particulier sur la rédaction, et ne sont pas toujours faciles à prendre en main.

On peut dès lors se demander si on peut améliorer l'accès à ces divers outils, dans l'optique de faciliter l'apprentissage tout en apportant de nouvelles ressources aux enseignants.

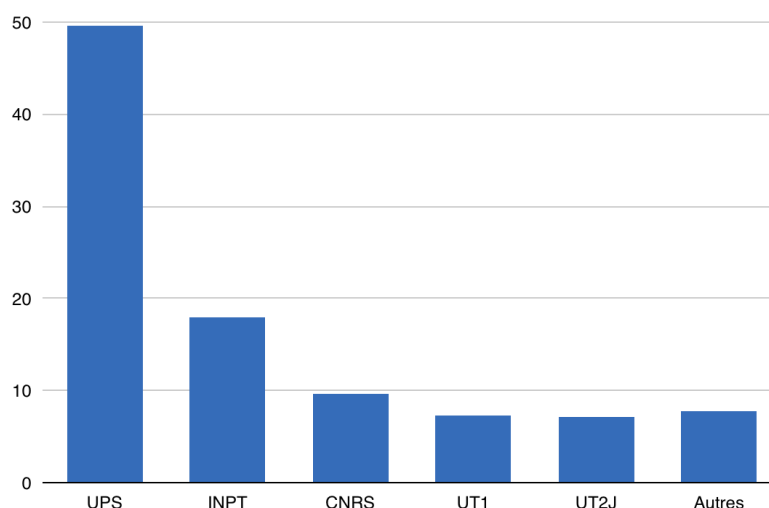
Le but de ce stage est de créer un programme pouvant répondre à cette problématique, en engendrant des questionnaires Moodle, mais aussi en utilisant d'autres logiciels tels que AMC à partir d'une description commune. Concrètement, les objectifs du stage sont de réfléchir à la forme de la description, produire un programme en langage python, lisible et compréhensible permettant de répondre à la problématique - génération de QCM - tout en prenant en compte l'ajout possible d'autres plateformes.

Dans ce rapport, nous allons dans un premier temps présenter le laboratoire de l'IRIT, ainsi que l'équipe ACADIE. En second temps, nous ferons une présentation du stage et des objectifs à remplir ainsi que les connaissances requises pour avancer. Pour finir, nous verrons plus précisément comment construire un programme structuré, remplissant les objectifs.

Présentation générale du laboratoire

1) Situation géographique

L'IRIT (Institut de Recherche en Informatique de Toulouse) est une des plus larges Unité Mixte de Recherches (UMR) du pays. Son site principal est situé à l'Université Paul Sabatier, mais l'institut s'étend sur de nombreux sites de Toulouse, tel que au CNRS par exemple.

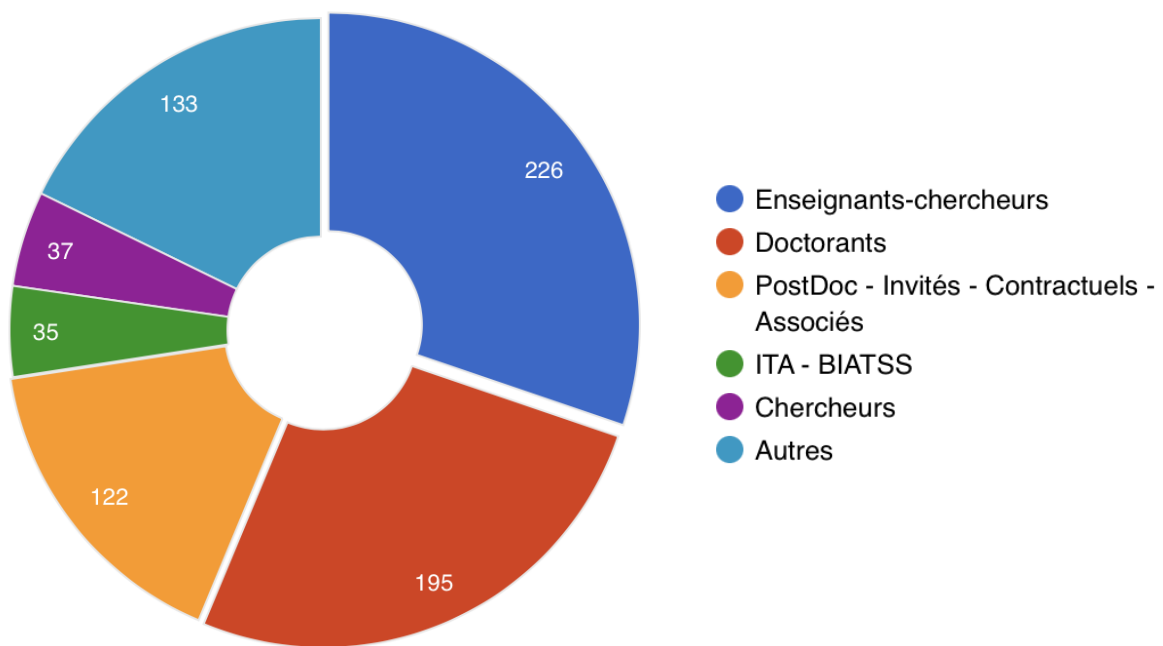


Pourcentage de membres par établissement de tutelles au sein de l'IRIT

Dans mon cas, j'ai intégré le site présent dans l'établissement de l'ENSEEIH (École nationale supérieure d'électrotechnique, d'électronique, d'informatique, d'hydraulique et des télécommunications) situé au centre ville de Toulouse.

2) Secteur d'activité

Créé en 1990, l'IRIT représente un des plus forts potentiels de la recherche en informatique en France avec un effectif global d'environ 800 personnes dont 270 chercheurs et enseignants chercheurs ainsi que 200 doctorants.



Nombres de membres de l'IRIT par catégorie

L'institut se structure autour de cinq grands sujets scientifiques :

- Conception et construction de systèmes (fiables, sûrs, adaptatifs, distribués, communicants, dynamiques...)
- Modélisation numérique du monde réel
- Concepts pour la cognition et l'interaction
- Etude des systèmes autonomes adaptatifs à leur environnement
- Passage de la donnée brute à l'information intelligible

Ces recherches, théoriques et appliquées, en Science des Données et du Calcul, trouvent leur application dans la vie courante.

Ainsi, celles-ci sont matérialisées dans six domaines d'application :

- Santé, Autonomie, Bien-être
- Ville intelligente
- Aéronautique, Espace, Transports
- Média sociaux numériques et diffusion de l'information
- Cybersécurité, Sécurité des biens et des personnes
- Education en ligne (e-education)

3) Equipe ACADIE



L'IRIT est divisé en 7 départements, chacun composé de plusieurs équipes, pour 24 équipes au total. Pendant le stage, j'ai pu travailler au sein de l'équipe ACADIE. L'équipe ACADIE (Assistance à la Certification d'Applications DIstribuées et Embarquées) étudie le développement et la vérification de systèmes logiciels critiques pour lesquels il est nécessaire d'exhiber un certificat de correction.

L'équipe est structurée en trois groupes :

- Théorie des types et théorie des catégories
- Développement de logiciels critiques
- Vérification de systèmes distribués critiques

C'est donc au sein de ce laboratoire que j'ai pu, pendant 6 semaines, réfléchir et travailler sur le projet.

Présentation du stage

1) Espace de travail

Tout au long du stage, j'ai pu travailler dans un espace dédié. En effet, l'équipe ACADIE nous a accueilli , d'autres stagiaires et moi, dans une de leur salle informatique. Celle-ci se situe dans le bâtiment E de l'ENSEEIH. Avoir un espace de travail adapté et agréable est primordial pour stimuler sa productivité et sa créativité. De plus, suite à certains imprévus, je me suis vu dans l'obligation de travailler quelques jours en télétravail. Après ces périodes de confinement, de nombreuses personnes ont choisi d'agréer ce mode de travail dans leur quotidien. En effet il présente plusieurs avantages dont notamment le gain de temps passé dans les transports en commun ou divers trajets au quotidien. Néanmoins, après avoir expérimenté ce mode de travail, je me suis trouvé moins productif, et plus facilement distrait que lorsque j'étais sur site.

2) Objectifs

L'objectif principal du stage est de produire un programme Python, permettant à l'utilisateur d'engendrer automatiquement des questionnaires compatibles avec différentes plateformes à partir d'une description commune. Pour commencer, les premières plateformes devant être supportées, et servant donc d'exemples fondamentaux sont Moodle et AMC :

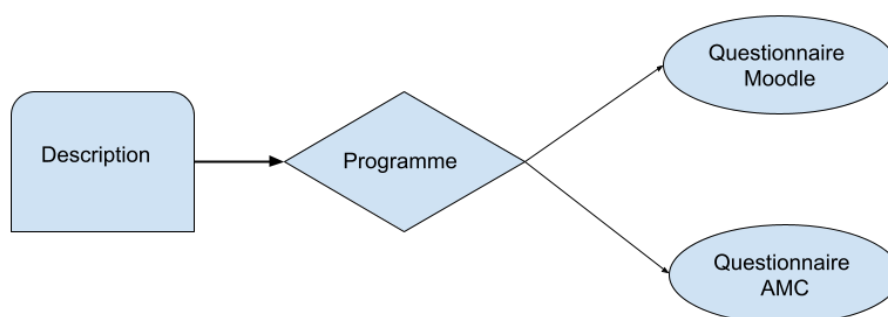


Diagramme de fonctionnement du programme

3) Présentation des plateformes

a. Moodle

Moodle, de l'acronyme *Modular Object Oriented Dynamic Learning Environment*, est un système de gestion de l'éducation qui fournit une plate-forme pour l'apprentissage en ligne. Il aide considérablement les différents enseignants à conceptualiser les cours et leurs structures ainsi que les programmes d'études, facilitant ainsi l'interaction en ligne avec les étudiants. Moodle a été conçu par Martin Dougiamas et depuis sa création en août 2002, son objectif principal a été de contribuer de manière appropriée au système d'apprentissage en ligne et de faciliter l'éducation. Moodle permet aux enseignants de créer des espaces virtuels de cours, dans lesquels il est possible de déposer des fichiers, ou encore y incorporer des activités, tel que des sondages, ou des quiz. L'activité "quiz" permet de pouvoir tester les étudiants à travers multiples questions, que ce soit sous forme d'examen ou encore d'entraînement. Moodle est donc une plateforme très diversifiée, utile pour l'apprentissage en ligne, notamment pour générer des questionnaires variés. Néanmoins, il existe aussi des logiciels profitables aux enseignants pour la mise en page de questionnaires papiers, tel que Amc.

b. AMC

AMC, ou *Auto Multiple Choice*, est un logiciel destiné à créer et à gérer des questionnaires papiers à choix multiples, avec une notation automatisée. Dès lors, il est donc possible de corriger automatiquement les copies une fois scannées. De plus, AMC permet à l'utilisateur de piocher aléatoirement dans un groupe de questions, ainsi que de mélanger l'ordre d'apparition de celles-ci. Il est donc possible de générer une copie différente par étudiant, tout en gardant un barème identique.

Ces deux plateformes présentent un grand intérêt pour la réalisation de ce projet. En effet, celles-ci présentent la possibilité d'importer des questionnaires, via un fichier LaTeX.

c. LaTeX

LaTeX est un système d'écriture de haute qualité: il comprend des fonctionnalités conçues pour la production de documents techniques et scientifiques, tel que notamment l'insertion de formules mathématiques. Lorsqu'il écrit, le rédacteur utilise du texte brut par opposition au texte formaté que l'on trouve dans les traitements de texte comme Microsoft Word ou LibreOffice Writer. Le rédacteur utilise des conventions de balisage pour définir la

structure générale d'un document ainsi que pour styliser le texte dans un document (tel que le gras et l'italique par exemple). Une compilation du document via une distribution TeX est utilisée pour produire un fichier de sortie (tel que PDF ou DVI) adapté à l'impression ou à la distribution numérique.

En outre, même si les deux logiciels permettent tous deux d'importer des questionnaires via LaTeX, ceux-ci sont très différents, que ce soit dans la construction de ce document LaTeX, les types de questions supportées ou plus simplement leur implémentation.

Pour conclure, il est totalement plausible de pouvoir mettre en place un programme permettant de générer deux fichiers LaTeX (un pour chaque plateforme), que l'utilisateur n'aura plus qu'à importer sur Moodle ou AMC. De plus, les deux logiciels sont totalement différents, permettant alors de réfléchir à plusieurs approches possibles. Ces plateformes constituent alors un exemple pertinent pour la conception du programme.

4) Méthode de travail

Durant cette période de stage, j'ai pu suivre une méthode de travail particulière. En effet, comme expliqué auparavant, ce sujet m'a nécessité d'acquérir de nombreuses connaissances.

J'ai alors suivi une méthode de travail particulière, dite méthode de développement itérative :

Tout d'abord, je met en évidence les points importants du sujet ainsi que les objectifs à remplir. Je commence alors par travailler avec un exemple simple et réfléchis à une stratégie pour le mettre en place. Après celle-ci mise en place, je dois pouvoir élargir cet exemple, c'est-à-dire pouvoir répondre aux plus de cas possibles. Je retourne alors à la phase de réflexion et de conception, et ainsi de suite, jusqu'à couvrir tous les cas possibles.

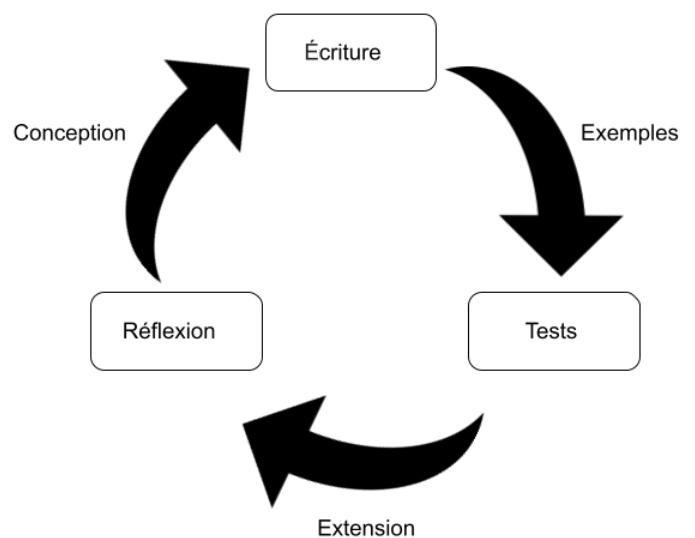


Schéma de méthode de travail

Grâce à cette méthode, j'ai pu découper le programme en sous programme qui une fois testé sont complètement utilisables et indépendants.

Présentations des activités

1) Premières réflexions

La première réflexion de conception repose sur le format de la description. En effet, celle-ci est à la fondation même du programme : Une des meilleures façons de stocker des données lisibles et compréhensibles par l'homme est d'utiliser un fichier JSON. JSON (JavaScript Object Notation) est un format léger d'échange de données. Il est facile à lire et à écrire manuellement mais aussi facile à analyser et à générer par les machines. Il est basé sur un sous-ensemble de la norme de langage de programmation JavaScript. JSON est donc un format texte totalement indépendant du langage, mais qui utilise des conventions familières à certains, dont JavaScript et Python. Par conséquent, ces propriétés font de JSON un langage d'échange de données idéal pour la réalisation du projet.

```
{
  "firstName": "John",
  "lastName": "Smith",
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  }
}
```

Exemple de fichier JSON

2) Approfondissement sur les plateformes Moodle et AMC

a. Moodle

Moodle propose aux enseignants ou éducateurs de créer leur propre questions, afin de les ajouter à la “banque de questions”. Néanmoins, ceci n’est pas toujours facile, cela peut parfois prendre du temps et ce n’est pas idéal lorsque l’on veut implémenter des formules mathématiques. Ainsi, Moodle propose une autre manière de créer des questionnaires, en les important en format XML (Extensible Markup Language). Pour pouvoir les importer de la sorte il est possible de rédiger tout d’abord un fichier LaTeX, en respectant des structures et environnements requis par la plateforme. En effet, il existe un paquet LaTeX qui permet d’engendrer un tel fichier XML, permettant alors à l’enseignant d’utiliser un outil qui lui est familier pour engendrer sa banque de questions.

```
\begin{<question type>}[<question options>]{<question name>}
<question text>
    \item[<item options>] <item>
    \item[<item options>] <item>
\end{<type de question>}
```

Exemple simple de structure LaTeX à respecter pour une question Moodle

Comme on peut le voir sur cet exemple, la structure à respecter est bien particulière est surtout celle ci dépend des différents types de questions possibles.

En effet, Moodle présente plusieurs types de questions :

- truefalse* : vrai ou faux.
- multi* : question à choix multiple classique.
- shortanswer* : question ouverte.
- numerical* : idem que *shortanswer* mais impose que la réponse soit un nombre.
- matching* : comprend plusieurs questions, il faut donc associer chaque réponse à la question correspondante.
- essay* : permet à l’étudiant de rendre un projet, sous format texte ou d’un fichier.
- description* : utiliser pour donner des feedback aux étudiants.
- cloze* : type de question pouvant regrouper plusieurs autres sous questions

Par conséquent, il est primordial de savoir avec quel type de question on veut travailler pour pouvoir importer des questionnaires depuis Latex. Pour finir, lorsque le document Latex est construit, grâce à des packages spécialisés, la compilation de celui-ci va automatiquement engendrer un document .xml, c'est donc ce dernier qu'il faudra importer.

b. AMC

Comme vu précédemment, le grand intérêt d'utiliser AMC est que ce logiciel permet d'avoir une correction automatique des copies. De plus, celui-ci permet également de mélanger les questions aléatoirement mais aussi de "piocher" dans un groupe de question donné pour chaque copie, rendant celle-ci complètement distinctes. De plus, pour pouvoir travailler sur la mise en page, AMC fonctionne via un fichier source, qui peut être en .txt ou format LaTeX. Néanmoins, pour une meilleure mise en page et notamment pour l'utilisation de formules mathématiques, il est nettement conseillé d'utiliser le format LaTeX. Dans les deux cas, le fichier source doit, à l'image de Moodle, être structuré de manière particulière en fonction des types de questions.

```
\begin{question}{identifiant}
  Texte de la question...
  \begin{reponses}
    \bonne{La bonne réponse}
    \mauvaise{Une mauvaise réponse}
    \mauvaise{Une autre mauvaise réponse}
  \end{reponses}
\end{question}
```

Exemple simple de structure LaTeX à respecter pour une question AMC

En effet, comme pour Moodle, AMC prend en charge plusieurs types de questions et par conséquent la rédaction du fichier latex sera donc plus ou moins différente pour chaque type.

Comme vu précédemment, l'objectif principal est de construire un programme pouvant engendrer automatiquement des questionnaires Moodle et AMC. Cependant, les types de questions d'AMC ne correspondent pas forcément avec ceux de Moodle. La

première chose à faire est alors de comparer les différents types de questions pour chaque logiciel. (Voir table 1)

Moodle	AMC
truefalse	question (doit être fait à la main)
multi (mode = single)	question
multi (mode = multiple)	questionmult
numerical	\Amc open
shortanswer	\Amc open
essay	\Amc open
matching	∅
cloze	∅
description	\QuestionIndicative

	Peut être corrigé automatiquement		Ne peut pas être corrigé automatiquement
	N'est pas noté		Non supporté

Table 1 : *Tableau des différents types de question Moodle et leur équivalent en AMC*

On remarque alors que certains types de questions Moodle ne sont pas pris en charge ou alors ne peuvent pas être corrigés automatiquement via AMC. Néanmoins cela va à l'encontre du projet, le principe même de l'utilisation d'AMC est de pouvoir corriger automatiquement les copies. Ces types de question Moodle vont donc demander un travail supplémentaire sur le programme, pour les transformer en d'autres types de question AMC. (Voir table 2)

Moodle	AMC
truefalse	Conversion en QCM classique, avec 2 réponses : Vrai et faux
multi (mode = single)	Pris en charge
multi(mode = multiple)	Pris en charge
shortanswer/numerical	Conversion en QCM classique, sous réserve que l'utilisateur rentre des "mauvaises réponses"
matching	Conversion en un groupe de questions. Chaque question utilisera les réponses des autres questions comme "mauvaises réponses"
cloze	Conversion en un groupe de questions. Chaque question sera transformée selon son type comme précédemment.

	Supporté		Nécessite une conversion		Nécessite une conversion + division en plusieurs questions
--	----------	--	--------------------------	--	--

Table 2 : *Tableau de types de question Moodle et leur conversion en AMC*

Par exemple, on peut voir le *matching* de Moodle comme plusieurs questions simples AMC. C'est donc en ayant ces modifications en tête que l'on peut construire un programme robuste et efficace.

3) Explication du programme

Avant de commencer à écrire un programme, il est essentiel de réfléchir à toutes les éventualités. En effet, on ne voudrait pas devoir tout modifier car on n'avait pas pensé à certains cas particuliers. Premièrement, comme vu précédemment, il faudra forcément agir de manière différente pour chaque type de question. En effet, si le type de question demande est un 'vrai ou faux' cela devra être pris en charge complètement différemment qu'une question ouverte par exemple. De plus, en fonction de si l'on veut un questionnaire Moodle ou AMC, la forme des questionnaires varie aussi. En outre, un des objectifs de ce projet est de pouvoir viser et donc d'implémenter d'autres plateformes, le plus facilement possible. Un problème se pose alors, comment concevoir un tel programme ?

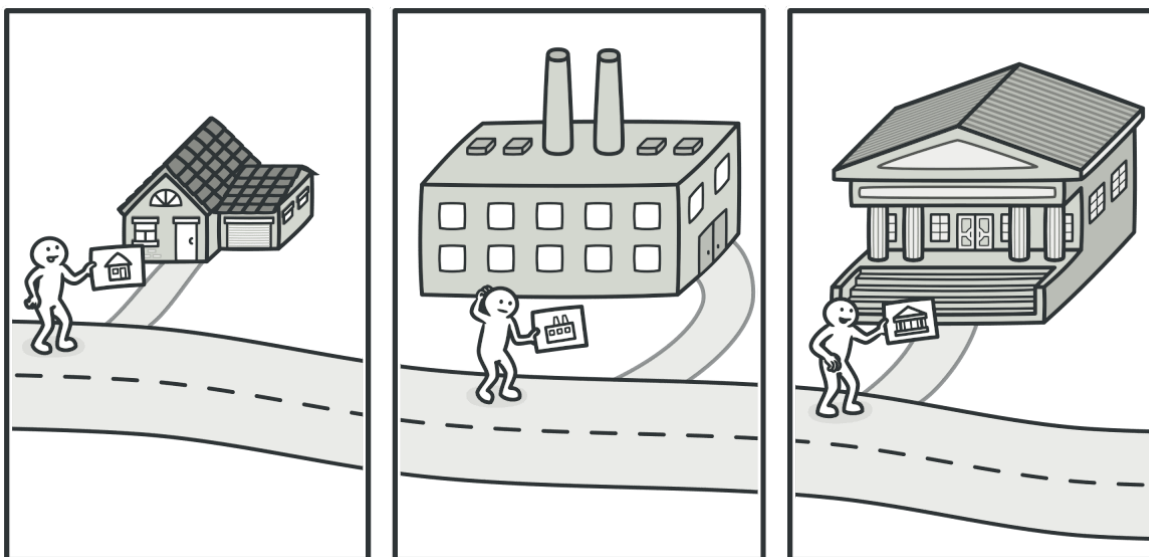
4) Visitor et Design pattern

Depuis le début de l'informatique, les développeurs ont remarqué que les problèmes de conception sont souvent récurrents. Ils ont alors mis en place des modèles de conception, aussi appelés “*design pattern*”.

En ingénierie informatique, un “*design pattern*” est une solution générale et réutilisable à un problème courant dans un contexte donné de conception logicielle. Cette conceptualisation ne se traduit pas directement en code mais plutôt en une idée abstraite de la manière de résoudre le problème récurrent.

Ces patrons de conception sont des meilleures pratiques formalisées que le programmeur peut utiliser pour résoudre des problèmes courants lors de la conception d'une application ou d'un système. Ils peuvent être considérés comme une approche structurée de la programmation informatique, un intermédiaire entre les niveaux d'un paradigme de programmation et d'un algorithme concret.

Parmi ces modèles de conceptions, le modèle “Visiteur” est le plus adapté pour résoudre notre problème. Voyons ainsi le fonctionnement du “*Visitor pattern*” à travers un exemple concret. Imaginons un agent d'assurance qui est impatient de trouver de nouveaux clients. Ce dernier peut visiter tous les immeubles d'un quartier, en essayant de vendre des assurances à tous ceux qu'il rencontre. Selon le type d'organisation qui occupe l'immeuble, il peut proposer des polices d'assurance spécialisées :



- Si c'est un immeuble résidentiel, il vend des assurances médicales.
- Si c'est une banque, il vend une assurance contre le vol.
- Si c'est un café, il vend une assurance incendie et inondation.

En programmation, l'agent d'assurance sera appelé visiteur et les différents bâtiments seront donc des objets visitables par ce dernier. Le visiteur réagira alors différemment pour chaque objet. On peut de plus facilement créer un autre visiteur, agissant différemment pour chaque objet lui aussi, mais pas de la même façon que le précédent. C'est le *Visitor Pattern*. (Voir annexe 2 pour le diagramme UML)

Pour retourner à notre programme, les objets visitables sont les différents types de question, et les visiteurs seront donc chacun assimilé à une plateforme quelconque. On crée alors dans notre cas un visiteur Moodle et un visiteur AMC. Chaque visiteur correspondant à une plateforme va alors opérer différemment pour chaque objet visité, correspondant ici à un type de question. Pour finir, si on veut implémenter une nouvelle plateforme, il suffit alors de créer un nouveau visiteur, sans avoir à modifier le code déjà existant.

5) Principe de Templates

Comme vu précédemment, la rédaction des différents documents LaTeX est spécifique à chaque type de question. Cependant, cette rédaction type doit être réécrite pour chaque même type de question, en changeant seulement le contenu. (Voir exemples page 12

et 13) Il est alors facile d' imaginer que réécrire la même structure un grand nombre de fois est redondant. Pour cela on utilise ce qu'on appelle des “templates”. Concrètement, en informatique une template est un moyen d'écrire plusieurs fois la même chose, à quelques variables près. Par exemple, si on veut pour une liste d'étudiants donnée, renvoyer leur moyenne générale, on peut écrire :

```
L'étudiant ${nom} a ${note} de moyenne.
```

Exemple d'utilisation de templates Mako

Dans cet exemple (rédigé en Mako), le texte en violet représente les variables, et le symbole “\$” en rose indique à l'interpréteur que le texte entre accolades est en fait une variable. Si l'on exécute cet exemple via une méthode Mako en imposant : nom = Dupond, note = 14.5 , on obtient alors la phrase : “ *L'étudiant Dupond a 14,5 de moyenne*”. On peut donc via un code python et en utilisant des templates Mako arriver aux fins de notre exemple en remplaçant “nom” et “note” par leur valeur pour chaque étudiant de la liste :

```
from mako.template import Template
Liste_Etudiants = [('Paul',14), ('Eric',12.4), ('Marc',19.5)]
Phrase =r'''L'étudiant ${nom} a ${note} de moyenne.'''
for etudiant,note in Liste_Etudiants :
    print(Template(Phrase).render(nom = etudiant,note = note))
```

Exemple de programme python utilisant Mako

Ce programme renvoie alors :

```
L'étudiant Paul a 14 de moyenne.
L'étudiant Eric a 12.4 de moyenne.
L'étudiant Marc a 19.5 de moyenne.
```

Il existe plusieurs librairies python pour pouvoir travailler avec des templates. J'ai choisi de travailler avec Mako, car celle-ci présente plusieurs avantages. Notamment, Mako permet de mettre du code python à l'intérieur même de la template. Cela me permet donc d'utiliser les structures de contrôle de python pour pouvoir réitérer une écriture (pour chaque bonne réponse par exemple), ou encore pour faire plusieurs cas (par exemple le cas où la bonne réponse est “Vrai” par rapport à celui où la bonne réponse est “Faux”)

```
% if question.correct :
    \bonne{Vrai}
    \mauvaise{Faux}
% else :
    \mauvaise{Vrai}
    \bonne{Faux}
% endif
```

Exemple d'utilisation de structure de contrôle dans une template à l'aide de Mako

Le programme repose ainsi sur ces trois principes :

- la description, formatée en JSON
- les visiteurs
- l'utilisation de templates.

A partir de la description, un sous programme va récupérer les informations, déterminer le type de la question et générer un objet “type de question”. Ensuite pour chaque plateforme, celui-ci va appeler le visiteur correspond pour qu’il ‘visite’ cette question a sa manière. Grâce aux templates, celui-ci n’a plus qu’à remplir les “trous” avec les informations de la description (Énoncé de la question, bonnes et mauvaises réponses, etc..). Dans les deux cas, pour Moodle ou AMC, ce template est alors écrit sur un fichier Latex. Néanmoins on note une différence importante pour les deux plateformes. En effet, AMC propose une fonctionnalité de groupe de questions et ne comprend qu’un seul fichier source. Il est donc indispensable d’avoir tous les groupes sur le même fichier Latex, puis avec un traitement AMC, on pourra ainsi récupérer une à plusieurs questions de chaque groupe pour les incorporer sur chaque copies.

Néanmoins, sur Moodle il pourrait être plus pertinent de créer un fichier Latex par question ou groupes de questions . L’utilisateur saura alors exactement quelle(s) question(s) se trouve sur quel fichier, rendant alors l’importation plus facile ainsi que s’il désire en modifier seulement quelques-unes. C’est donc pour cela que le programme, va pour Moodle générer plusieurs fichiers Latex, alors que pour AMC celui-ci n’en génère qu’un unique. Le programme prend alors totalement en charge Moodle et AMC, l’objectif principal est alors atteint.

Cependant on peut déjà réfléchir à de futures améliorations...

6) Voies d'amélioration

a. Prise en compte de Wooclap

Premièrement il est facile de deviner comment renforcer la portée du programme. En effet, nous avons favorisé celui-ci pour qu'il puisse supporter d'autres plateformes aisément. Dès lors on peut donc ajouter d'autres plateformes pour les rendre compatibles avec ce dernier. Prenons alors l'exemple de Wooclap. Wooclap est une plateforme électronique interactive utilisée pour créer des sondages et des questionnaires. Les utilisateurs répondent aux questions de manière anonyme par le biais de leurs smartphones ou ordinateurs portables. Le principe de Wooclap est très proche de l'activité 'quiz' de Moodle, cependant Wooclap est plutôt destiné à être utilisé en temps réel, lors d'un cours magistral ou une présentation par exemple. Pour importer des questionnaires, Wooclap permet plusieurs formats, dont particulièrement intéressants. Premièrement, Wooclap permet de les importer directement via le format .xml de Moodle. On pourrait alors se dire que le travail est terminé et qu'il suffirait de travailler exactement comme pour Moodle, cependant cela est loin d'être vrai. En effet, malgré cela, Wooclap ne possède pas exactement les mêmes types de questions que Moodle, rendant alors les importations de certains types compliqués voir impossible.

Le deuxième format supporté par Wooclap est le format .xls, ou format Excel. Celui-ci comme pour Moodle et AMC demande alors une rédaction particulière mais demande donc un traitement totalement différent. Selon moi ajouter une plateforme comme Wooclap serait très pertinent ainsi que bénéfique pour le programme, démontrant alors la variété de paramètres pris en compte, tel que le type de fichier en sortie.

b. Amélioration de la description

Une autre piste majeure d'amélioration serait au niveau de la description. En effet, on a vu que celle-ci est en format JSON, pour pouvoir stocker et manipuler plus facilement les données. Néanmoins, cela demande alors à l'utilisateur de connaître et savoir remplir un fichier JSON correctement. Dès lors, on peut imaginer un système permettant à l'utilisateur de simplement rentrer des informations, puis les transcrire automatiquement en un fichier JSON adapté. Il serait alors possible de mettre en place une IHM (Interface Homme-Machine) à travers laquelle l'utilisateur pourrait joindre un fichier texte, ainsi que cocher des cases pour indiquer le type de chaque question. Cela rendrait alors le programme plus simple à utiliser et donc beaucoup plus accessible.

c. Nouveau type de question : question portée sur un programme

Dans la conception du programme, les types de questions utilisés sont ceux qui existent déjà dans certaines plateformes, qui ont souvent dû être convertis pour les rendre accessibles à d'autres. Néanmoins, il est totalement imaginable de créer de nouveaux types de questions, propres au projet. Prenons l'exemple de questions d'informatique, portant sur un programme. Imaginons alors que le but soit de savoir ce que renvoie ce programme avec une certaine initialisation, que l'on veut différente pour chaque étudiant. Si ce genre de question devient récurrent, il est alors pénible pour l'utilisateur de chercher lui-même ce que renvoie le programme à chaque fois.

On pourrait donc implémenter un nouveau type de question, appelé "*code*" par exemple. Cependant, ce type de question n'est évidemment pas pris en charge par les différentes plateformes, et un travail supplémentaire est alors nécessaire pour pouvoir supporter ce nouveau type. Dans cet exemple, l'utilisateur rentrera non seulement le programme en question mais aussi les différentes variations de celui-ci. Cela serait largement trop différent des autres types, et serait compliqué à généraliser. Ce qui est important de remarquer ici, c'est que précédemment, la porte d'entrée du programme était la description, formatée en JSON. Le programme fonctionne alors à partir de ce JSON particulier. En voulant implémenter ce nouveau type de question, il faudrait alors normalement complètement changer le fonctionnement de cette description et donc du programme.

On peut ici passer par une autre méthode plus adaptée. Comme on a déjà un programme fonctionnant à partir d'une certaine description JSON, il suffit alors de générer cette description à partir des informations données par l'utilisateur. C'est-à-dire que l'on va à partir d'un JSON spécial, générer plusieurs JSON pouvant être pris en charge par le programme.

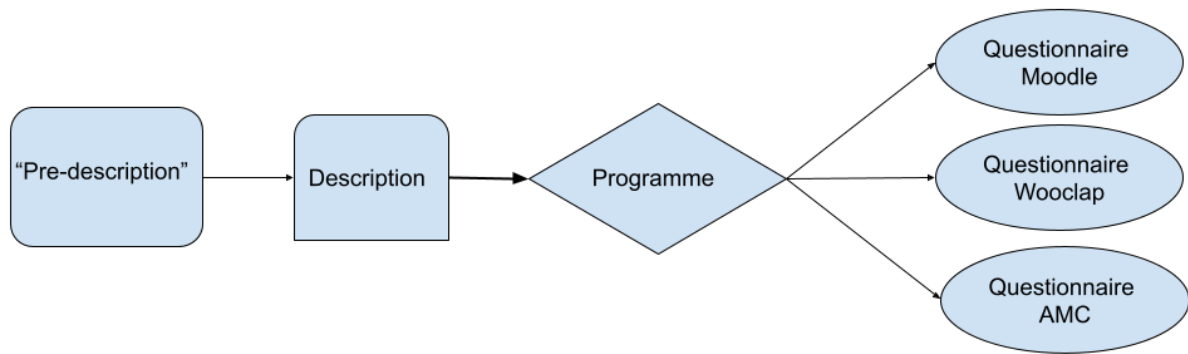


Schéma de fonctionnement du programme

À travers cet exemple, on imagine d'autres visions d'améliorations. Ici, on utilise les sous-programmes déjà existant à la place de créer une autre alternative pour pouvoir supporter un nouveau type de questions. Cela démontre alors la flexibilité du programme.

Conclusion

Ce stage fut très enrichissant pour moi, que ce soit sur le plan professionnel ou personnel. D'un point de vue professionnel, j'ai pu m'épanouir pendant 6 semaines dans un milieu de travail très différent. De plus, j'ai appris l'importance d'une organisation efficace, et découvert de nouvelles méthodes de travail. En outre, j'ai pu découvrir à travers mes divers échanges au sein de l'équipe, le fonctionnement d'un laboratoire de recherche et ce que représente concrètement le métier de chercheur en informatique. D'un point de vue personnel, ce stage m'a permis de gagner en autonomie, tout en acquérant de nombreuses connaissances en informatique. J'ai pu, par exemple, me familiariser avec une des notions les plus importantes en programmation, la Programmation Orientée Objet. Pour aller plus loin, ce stage a renforcé mon engouement pour l'informatique et le développement, et donc solidifie mon projet professionnel.

L'objectif principal de ce stage était de faciliter l'accès aux enseignants à la mise en place de questionnaires de différentes plateformes. Pour ce faire, j'ai construit un programme permettant de générer des fichiers pouvant être directement importés sur chaque plateforme, à partir d'une description commune. J'ai alors appris l'importance des modèles de conceptions, tel que le "visitor pattern" mis en place dans ce programme, mais aussi d'autres modèles intéressants que j'utiliserais très certainement dans d'autres projets. De plus, j'ai découvert divers outils tels que les fichiers JSON et les templates Mako faisant alors passer mes compétences en programmation au niveau supérieur. N'ayant que 6 semaines pour réaliser ce projet, j'ai dû réfléchir à de futures améliorations, ainsi que rendre mon code le plus lisible et compréhensible possible, dans l'optique de pouvoir être facilement agrémenté par d'autres personnes.

Grâce à ce projet, j'ai pu mettre en avant l'enseignement en ligne, qui est une pratique en plein développement. En effet, il s'agit d'une nouvelle forme d'apprentissage, pouvant profiter à tous, et rendant alors l'éducation plus accessible.

Références bibliographiques

Irit :

<https://www.irit.fr>

Moodle :

<https://moodle.org/demo>

<https://ctan.org/pkg/moodle?lang=en>

Auto Multiple Choice :

<https://www.auto-multiple-choice.net/>

Wooclap :

<https://docs.wooclap.com/en/collections/2357575-integrations-imports-exports>

Mako :

<https://www.makotemplates.org/>

JSON :

<https://docs.python.org/3/library/json.html>

Latex :

<https://www.latex-project.org/>

XlsxWriter :

<https://xlsxwriter.readthedocs.io/>

Visitor Pattern :

<https://python-3-patterns-idioms-test.readthedocs.io/en/latest/Visitor.html>

<https://refactoring.guru/design-patterns/visitor/python/example>

Composite pattern :

<https://www.youtube.com/watch?v=EWDmWbJ4wRA&list=PLrhzyIcii6GNjpARdnO4ueTUAVR9eMBpc&index=14>

SubProcess :

<https://docs.python.org/3/library/subprocess.html>

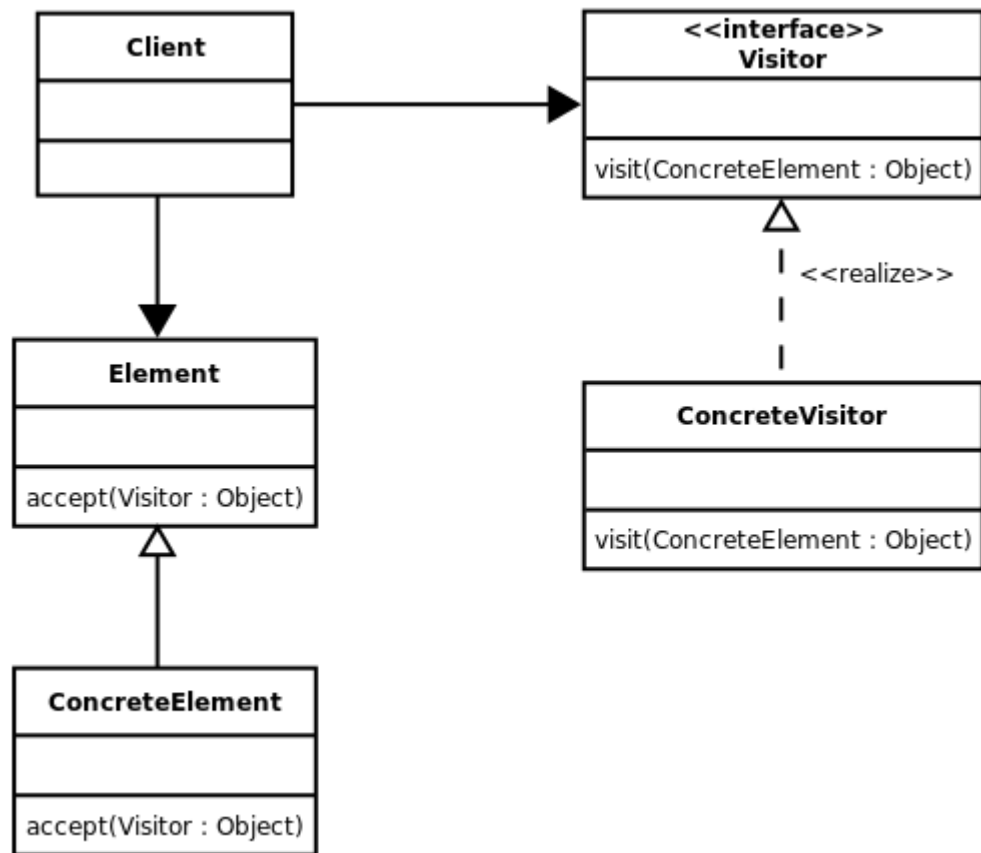
Annexes

- Annexe 1 : Tableau des différents types de questions et leurs équivalents dans chaque plateforme

Moodle	AMC	Wooclap
truefalse	question (has to be formatted manually)	MCQ (has to be formatted manually)
multi (mode = single)	question	MCQ
multi (mode = multiple)	questionmult	MCQ
numerical	\Amc open	GuessNumber
shortanswer	\Amc open	OpenQuestion
essay	\Amc open	∅
matching	∅	Matching
cloze	∅	∅
description	\QuestionIndicative	Diapositive
∅	∅	Sorting
Cloze + shortanswer	∅	FillinTheBlanks

	Peut être corrigé automatiquement		Ne peut pas être corrigé automatiquement
	N'est pas noté		Non supporté

- Annexe 2 : Exemple de diagramme UML de visitor pattern



- Annexe 3 : Différentes étapes du programme pour un exemple simple

1) Description JSON

```
{  "Category" : "Informatique 1A",
  "Statement" : null,
  "Code"      : null,
  "Quiz" : {
    "Options" : [],
    "Id" : "input",
    "Type" : "Mcq",
    "Core" : "Quel est le type de l'objet retourné par
float(input()) ?",
    "CorrectAnswers" : ["float"],
    "WrongAnswers" : ["bool", "int", "str"]
  }
}
```

Description commune en JSON

2) Formats intermédiaires

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage{moodle}
\begin{document}

\begin{quiz}{Informatique 1A}

\begin{multi}{input}
Quel est le type de l'objet retourné par float(input()) ?
  \item* float
  \item bool
  \item int
  \item str
\end{multi}
\end{quiz}
\end{document}
```

Document intermédiaire Latex pour Moodle

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- This file was generated on 2022-06-06 by LuaLaTeX -->
<!-- running on *NIX with the package moodle v0.9 -->

<quiz>

<question type="category">
  <category>
    <text>$course$/top/Informatique 1A</text>
  </category>
</question>

<question type="multichoice">
  <name format="html">
    <text><![CDATA[input]]></text>
  </name>
  <questiontext format="html">
    <text><![CDATA[<p>Quel est le type de l'objet retourné par float(input()) ? </p>]]></text>
  </questiontext>
  <defaultgrade>1.0</defaultgrade>
  <generalfeedback format="html"><text></generalfeedback>
  <penalty>0.10</penalty>
  <hidden>0</hidden>
  <single>true</single>
  <shuffleanswers>1</shuffleanswers>
  <answernumbering>abc</answernumbering>
  <answer fraction="100" format="html">
    <text><![CDATA[<p>float</p>]]></text>
  </answer>
  <answer fraction="0" format="html">
    <text><![CDATA[<p>bool</p>]]></text>
  </answer>
  <answer fraction="0" format="html">
    <text><![CDATA[<p>int</p>]]></text>
  </answer>
  <answer fraction="0" format="html">
    <text><![CDATA[<p>str</p>]]></text>
  </answer>
</question>

</quiz>

```

Document XML engendré par le fichier Latex pour Moodle

wooclap_input.xlsx							
	A	B	C	D	E	F	G
1	Type	Title	Correct	Choice	Choice	Choice	Choice
2	MCQ	Quel est le typ	1	float	bool	int	str

Document intermédiaire XLSX pour Wooclap

```

\documentclass[a4paper]{article}
\usepackage[utf8x]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[francais,bloc,completemulti]{automultiplechoice}
\begin{document}
\setdefaultgroupmode{withoutreplacement}
\element{Informatique 1A}{
\begin{question}{input}
Quel est le type de l'objet retourné par float(input()) ?
\begin{reponses}
\bonne{float}
\mauvaise{bool}
\mauvaise{int}
\mauvaise{str}
\end{reponses}
\end{question}
}
%% debut de l'en-tête des copies :
\noindent{\bf QCM \hfill TEST}
\vspace*{.5cm}
\begin{minipage}{.4\linewidth}
\centering\large\bf Test\ Examen du 01/01/2008\end{minipage}
\champnom{\fbox{
\begin{minipage}{.5\linewidth}
Nom et prénom :

\vspace*{.5cm}\namefielddots
\vspace*{1mm}
\end{minipage}
}}
\begin{center}\em
Durée : 10 minutes.

Aucun document n'est autorisé.
L'usage de la calculatrice est interdit.

Les questions faisant apparaître le symbole \multiSymbole{} peuvent
présenter zéro, une ou plusieurs bonnes réponses. Les autres ont
une unique bonne réponse.

Des points négatifs pourront être affectés à de \emph{très
mauvaises} réponses.
\end{center}
\vspace{1ex}

%% fin de l'en-tête
\begin{center}
\hrule\vspace{2mm}
\bf\Large Informatique 1A
\vspace{2mm}\hrule
\end{center}
\restituegroupe{Informatique 1A}
}
\end{document}

```

Document intermédiaire Latex pour AMC

3) Résultats

My first course / Informatique / Preview

QUIZ Informatique

Quiz Settings Questions Results Question bank More ▾

Back

Question 1

Not yet
answered

Marked out of
1.00

Flag
question

Edit
question

Quel est le type de l'objet retourné par float(input()) ?

Select one:

- ☐ a. int
- ☐ b. str
- ☐ c. bool
- ☐ d. float

Résultat après importation sur Moodle

www.wooclap.com/TZEYNZ

Quel est le type de l'objet retourné par float(input()) ?

← 1 float

2 bool

3 int

4 str →

wooclap Questions 1 / 1 Messages 100 % Exit 0 / 0

Résultat après importation sur Wooclap



+1/1/60+

QCM

TEST

Test
Examen du 01/01/2008

Nom et prénom :

.....

Durée : 10 minutes.

Aucun document n'est autorisé. L'usage de la calculatrice est interdit.

Les questions faisant apparaître le symbole ♣ peuvent présenter zéro, une ou plusieurs bonnes réponses. Les autres ont une unique bonne réponse.

Des points négatifs pourront être affectés à de très mauvaises réponses.

Informatique 1A

Question 1 Quel est le type de l'objet retourné par `float(input())` ?

- ☐ str
- ☐ bool
- ☐ int
- ☐ float

Résultat après importation sur AMC

- Annexe 4 : Liste des tâches effectuées

- Documentation, acquisition de nouvelles connaissances sur JSON et le sujet
- Début de raffinement et expérimentation (json)
- Documentation, acquisition de nouvelles connaissances sur Moodle et AMC
- Expérimentation sur Moodle et AMC avec un exemple simple
- Prise en main de Moodle sandbox
- Début de programme suivant l'exemple simple.
- Fin et amélioration du programme suivant l'exemple simple
- Remise en question des choix du programme
- Découverte et apprentissage : POO
- Mise en pratique, création d'une classe Moodle
- Continuation de l'acquisition de connaissances sur la POO
- Amélioration de la classe 'Moodle'
- Création d'une classe AMC
- Essai d'implémentation du package "minted"
- Mise en place des groupes pour amc
- Amélioration groupes amc et mise en page du document
- Documentation sur "le strategy pattern", "bridge pattern"
- Début de réflexion sur l'architecture du code
- Remise en question des patterns : les plus adaptés sont visitor et composite
- Documentation sur ces patterns
- Début de reconstruction du code
- Mise en place d'un Visitor pattern
- Documentation sur les templates : Mako
- Avancement du programme avec utilisation de templates
- Amélioration de l'architecture prenant en compte l'ajout d'autres plateformes
- Ajout d'une troisième plateforme : Wooclap