

Application des techniques de développement piloté par les tests à l'apprentissage de l'algorithmique

Jérôme MARTIN, Cléo BARAS, Rémy CHOLLET

UGA - IUT1 - Département RT

Domaine universitaire, 151 rue de la Papeterie, 38400 Saint-Martin d'Hères

Email : {jerome.martin, cleo.baras, remy.chollet}@univ-grenoble-alpes.fr

Index Terms—Développement piloté par les tests, Tests unitaires, Intégration continue, Python, pytest

I. INTRODUCTION

Appréhender les bases de l'algorithmique et découvrir un langage informatique sont deux des briques de bases de nombreuses formations techniques post-bac. Même avec les efforts engagés pour introduire la programmation au niveau de l'enseignement secondaire, l'informatique reste une matière difficile à prendre en main pour les étudiants du DUT Réseaux et Télécommunications ; l'enjeu est d'autant plus grand que le PPN [4] de la formation la considère même comme une compétence transverse utile à tous les savoirs technologiques R&T.

Ces bases de l'algorithmique sont envisagées en première année à travers deux modules du PPN (M1207 et M2207) de 60 heures avec des notions simples appliquées à un langage de programmation :

- Connaissance des variables et des structures de données (tableaux, tableaux associatifs) manipulant des types classiques (nombre et chaînes de caractères) avec une ouverture vers la notion d'objet ;
- Programmation procédurale et structures de contrôle fondamentales (tests, boucles) ;
- Structuration du code en fonctions et sous-programmes.

Pour asseoir leurs compétences informatiques par une pratique plus intensive de la programmation, nous avons également orienté le module de gestion de projet (M2109) de fin de première année vers la réalisation d'un projet informatique [1].

Les notions visées par ces trois modules cherchent non pas à faire de nos étudiants des développeurs informatique mais bien :

- à leur donner les outils pour automatiser les tâches récurrentes d'un système informatique (par exemple, création de comptes utilisateurs, traitement des fichiers de logs, déploiement automatique, ...);
- à les former au respect d'un cahier des charges avec des contraintes en entrée et en sortie (par exemple insérer les comptes utilisateurs dans une base de données à partir d'un tableur).

Nous faisons principalement face à deux profils d'étudiants ayant des difficultés :

- les réfractaires dont la motivation est uniquement centrée sur les aspects réseau de notre DUT les condui-

sant à dire : *“la programmation n'est pas faite pour moi”* ou *“à quoi ça va me servir, je veux faire du réseau”* ;

- les enthousiastes qui pensaient facilement réussir car ayant déjà fait du “Scratch”, développé une page web ou étant fan de jeux vidéo et dont certains se retrouvent vite face à des désillusions.

Pour l'enseignant, ces difficultés viennent de plusieurs raisons :

- la lecture et la compréhension de sujet, un problème récurrent quelque soit le niveau, y compris à l'ENA [2] ;
- l'apprentissage du langage de programmation avec, comme pour les langues étrangères, ses mots clés, sa grammaire et ses difficultés de traduction ;
- le respect des consignes et du cahier des charges, assurément lié à la compréhension du sujet, dans lequel une réalisation approximative n'a pas sa place.

Ces raisons peuvent s'illustrer à travers un exemple d'exercice simple : *écrire une fonction ayant pour paramètre un entier positif et qui retourne sa représentation binaire sous la forme d'une chaîne de caractères*. Une grande partie des étudiants viendra afficher le résultat au lieu de le renvoyer. Une autre ne saura pas comment écrire les itérations nécessaires au calcul du code binaire en utilisant une boucle. Une dernière considérera qu'un programme principal qui demande une valeur à l'utilisateur et affiche le résultat suffira à répondre à la question.

Pour répondre à ces problèmes, nous avons décidé d'adapter notre pédagogie en nous inspirant des méthodes de développement de l'industrie. Elle a mis en place un processus efficace pour développer un produit informatique qui consiste à :

- 1) coder les tests qui traduisent les contraintes du cahier des charges avant même de développer le produit ;
- 2) itérer des étapes de développement et de validation consistant à coder progressivement le produit et à le valider avec les tests, dans l'idée que l'ensemble des ces itérations convergent vers le produit final.

En entreprise, ces méthodes reposent également sur des outils informatiques permettant le partage du code entre développeurs et l'automatisation des tests sur chaque nouvelle version développée. On parle d'**intégration continue**.

Notre approche pédagogique vise à initier progressivement les étudiants à ces méthodes de développement tout en les formant à l’algorithmique et la programmation. Trois objectifs pédagogiques incrémentaux en découlent :

- 1) apprendre à valider de manière systématique le code avec des tests que nous avons développé pour nos étudiants ;
- 2) les amener à prendre conscience de l’importance/intérêt des tests pour valider leur travail ;
- 3) initier leur connaissance sur l’écriture de tests pour, dans un horizon plus lointain, les rendre autonomes.

II. OBJECTIFS DE LA PUBLICATION

Dans cet article, nous décrivons plus concrètement l’approche pédagogique envisagée en suivant la chronologie des trois modules de première année et en les illustrant d’exemples, qui dans notre cas, se basent sur le langage de programmation Python. Nous aborderons donc :

- le premier module (M1207 - Bases de la programmation), visant les rudiments de l’algorithmique (structures alternatives, boucles et fonctions) : nous avons choisi de l’orienter vers des exercices graphiques et nous avons développé pour l’occasion des tests permettant à un étudiant d’exécuter son code avec une sous-impression du résultat attendu (cf. figure 1) ;
- le second module (M2207 - Programmation avancée), visant la connaissance des structures de données avancées (listes, dictionnaires), pour lequel les étudiants ayant déjà acquis la notion de fonction, nous introduisons des tests basés sur l’outil existant pytest [3], couramment utilisé par la communauté Python ;
- le dernier module (M2109 - Gestion de projet), dans lequel les étudiants développent un programme informatique [1] conséquent (jeu de sokoban) avec des outils d’intégration continue (serveur de version, tests automatisés). Ces outils permettent aux étudiants de vérifier leur progression par rapport aux attendus des échéances fixées pour le projet et aux enseignants de valider l’avancement.

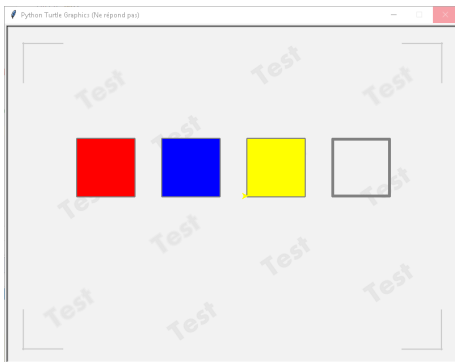


FIGURE 1. Exemple d’exercice graphique du module de programmation 1 : les carrés de couleur sont dessinés par le programme de l’étudiant au dessus du modèle représenté par un cadre gris.

Nous concluons en présentant un retour d’expérience sur nos dernières années de pratique.

RÉFÉRENCES

- [1] Cléo Baras, Rémy Chollet, and Jérôme Martin. Gestion de projet informatique (en cours de soumission). In *Workshop pédagogique R&T*, 2018.
- [2] Thierry Bert. Rapport sur les concours d’entrée à l’École nationale d’administration. Technical report, École Nationale d’Administration, 2016.
- [3] Holger Krekel and pytest-dev team. pytest : helps you write better programs. <https://docs.pytest.org/en/latest/>.
- [4] Ministère de l’enseignement supérieur et de la recherche. *Programme pédagogique national du DUT Réseaux et Télécommunications*, 2013.