Gestion de projet informatique

Cléo BARAS, Rémy CHOLLET, Jérôme MARTIN
UGA - IUT1 - Département RT
Domaine universitaire, 151 rue de la Papeterie, 38400 Saint-Martin d'Hères
Email : {cleo.baras, remy.chollet, jerome.martin}@univ-grenoble-alpes.fr

***Index Terms*—Gestion de projet, SVN, Gantt, Python, Sphinx, Tests unitaires**

I. Introduction

Savoir *"appréhender un projet dans sa globalité"*[5] est une compétence phare du PPN du DUT Réseaux et Télécommunications (RT). Quatre modules répartis sur deux années doivent former 1) à la gestion de projet avec ses contraintes économiques et temporelles, 2) à la mise en place d'un cahier des charges, 3) à sa réalisation concrète et 4) à sa restitution. Cet objectif est d'autant plus ambitieux qu'il attend une transdisciplinarité et un recul difficiles à obtenir de beaucoup d'étudiants pour le module de gestion de projet M2109 du semestre 2 : *"Mettre en oeuvre des méthodes de conduite de projets"*.

Les compétences en programmation, attendues en fin de semestre 2 (modules M1207/M2207), sont elles aussi ambitieuses : à la maitrise de l'algorithme et de la programmation structurée s'ajoute la notion d'*objets* et d'*architectures client-serveur avec un client interrogeant un service serveur* dans un volume d'heures restreint.

Notre choix pédagogique s'est donc dirigé vers un projet de développement informatique réalisé en binôme mis en oeuvre dans le cadre du module M2109. Ses avantages sont multiples :

- prolonger l'acquisition des savoirs en programmation et offrir un cadre propice à une pratique intensive de la programmation que l'on sait être nécessaire pour y acquérir de réelles compétences ;
- offrir une première réalisation technique fonctionnelle guidée par un cahier des charges imposé et détaillé : 1) elle confronte les étudiants à la prise en main d'un document technique conséquent et 2) elle est adaptable aux niveaux des étudiants en programmation (car de difficulté incrémentale) ;
- faire l'expérience d'un projet en équipe sur 12 semaines : répartir le travail dans le temps et entre les éléments du binôme, respecter un cahier des charges avec des échéances régulières et utiliser des outils de travail collaboratif (serveur de version) et de gestion de projet (diagramme de gantt).

II. Le projet de développement informatique

Pour faciliter le travail d'encadrement tout en restant dans un cadre ludique, nous proposons aux étudiants de développer un jeu dont nous imposons l'architecture et le langage de programmation (ici Python abordé en cours). 3 jeux sont possibles : la *bataille navale* où deux joueurs manipulent une grille de navires à torpiller ; le *sokoban* où un joueur pousse des caisses sur les cibles d'un entrepot fermé jusqu'à résoudre le puzzle ; le *puissance 4* où deux joueurs alignent des séries de pions. Tous sont des jeux de plateau dans lequel les éléments du jeu (navires, pions, caisses) se positionnent sur des cases d'une grille 2D.

Les étudiants ont tous le même jeu à réaliser, avec un roulement d'une année sur l'autre.

A. Version basique du jeu proposé

Ces jeux sont particulièrement propices à réviser les concepts de programmation présentés en cours :

- des structures de données pour modéliser les informations des joueurs (pseudo, couleurs de pion pour le *puissance 4* ou nom du puzzle pour le sokoban) ici dans des *dictionnaires mutables*, ou pour se repérer dans la grille 2D (fameuses actions *"je tire en C1"* de la *bataille navale*) permettant de travailler sur les *tableaux* et les *chaines de caractères* ;
- des *fonctions* dont certaines permettent de revoir le concept de *passage de paramètre par réference* ;
- des modules dédiés à des tâches spécifiques et relativement autonomes pour structurer le code : tâches d'affichage, traitement du plateau de jeu, interaction avec l'utilisateur et validité des données saisies.

Un premier développement simple (environ 700 lignes de code), basé sur les seuls acquis des modules de programmation, permet d'aboutir à un jeu fonctionnel, avec affichage textuel dans une console similaire à celui de la figure 1.

B. Des variantes de difficultés diverses

Ces jeux se prêtent bien à developper plusieurs variantes prolongeant l'acquisition de compétences à des concepts non abordés en cours, parmi lesquels :

- la programmation objet et évenementielle avec une version graphique utilisant des modules spécifiques (TkInter ou Pygame) pour un rendu tel que présenté figure 2 ;
- des communications client-serveur permettant à deux joueurs de se connecter à un serveur (fourni et développé par nos soins) pour s'échanger les informations de jeu (par ex. colonne jouée dans le *puissance 4*) ;
- des concepts d'algorithmique plus ou moins élaborés pour jouer contre une intelligence artificielle.

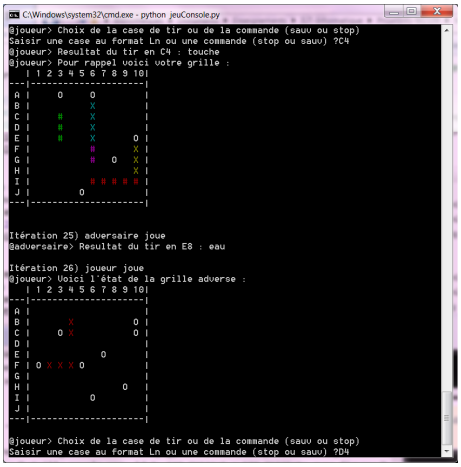

FIGURE 1. Une vue du jeu de bataille navale basique avec affichage dans une console Windows

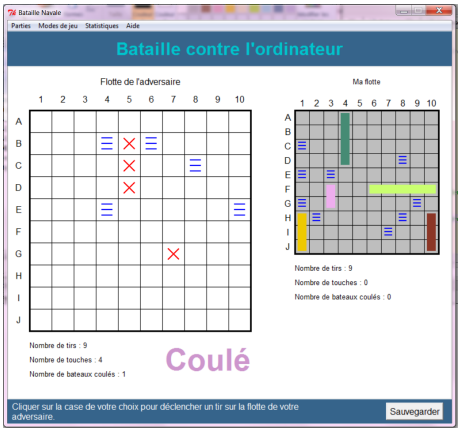

FIGURE 2. Version graphique du jeu de bataille navale en TkInter

C. Attendus et avantages pédagogiques

Le modèle d'implémentation du jeu a été soigneusement rédigé sous la forme d'un cahier des charges approfondi [2]) auquel les étudiants doivent se conformer. Ce cahier des charges impose à la fois :

— l'ensemble de la structure du programme (modèles des données, fonctions et modules à implémenter, protocole de communication entre les clients et le serveur, ...) ;

— les échéances datées auxquelles les éléments du programme doivent être livrés : la version basique du jeu est notamment segmentée en 8 échéances (contenant de 2 et 8 fonctions), à raison d'une par semaine ;

— des tests unitaires (développés par nos soins et fournis) pour permettre aux étudiants de vérifier de manière autonome la conformité de leur travail avec les attendus du cahier des charges.

La production des binômes doit être publiée sur un serveur de gestion de version SVN [1] mis à leur disposition. Ce serveur :

— facilite le travail collaboratif au sein du binôme : la structure modulaire (sur plusieurs fichiers) du jeu est particulièrement propice à la répartition du travail au sein du binôme et à des dépôts SVN concurrents sans trop de conflits.

— permet la supervision du travail par les enseignants avec un accès permanent aux codes des étudiants,

— automatise l'évaluation du travail des étudiants : à chaque échéance, les tests unitaires sont exécutés directement sur le serveur, en utilisant la dernière version du code soumise par les étudiants ; un score de conformité est automatiquement calculé pour vérifier si le binôme a rendu le travail attendu dans le temps imparti.

III. OBJECTIFS DE LA PUBLICATION

Dans cet article, nous nous proposerons de décrire la mise en oeuvre de ce projet informatique dans un contexte de gestion de projet. Nous aborderons les outils techniques conçus par les enseignants et mis à disposition des étudiants :

1) l'outil de documentation automatique de code Sphinx [3] pour générer un cahier des charges avec les avantages de l'hypertexte : une lecture non linéaire de son contenu, un outil de recherche inclus et un interfaçage aisé avec les traducteurs automatiques du web pour les étudiants étrangers ;

2) le serveur de gestion de versions (ici l'outil Apache Subversion [1], [4] et son client TortoiseSVN [6]) qu'utilisent souvent les entreprises de développement informatique.

Nous décrirons également les modalités de mise en oeuvre du projet :

1) des TD de gestion de projet pour initier les étudiants aux impératifs organisationnels ;

2) des TD machine pour prendre en main les éléments techniques du projet ;

3) le suivi de l'avancement avec les tests unitaires et l'évaluation finale consistant à faire recoder aux étudiants quelques fonctions qu'ils ont soumis sur le serveur SVN.

Nous conclurons par un retour d'expérience basé sur nos 4 années de mise en pratique.

RÉFÉRENCES

[1] Apache Subversion, enterprise-class centralized version control for the masses. https ://subversion.apache.org/.

[2] Cléo Baras, Rémy Chollet, Yves Delnondedieu, and Jérôme Martin. Projet informatique - documentation technique du jeu battleship. http ://chamilo1.grenet.fr/ujf/courses/IUT1RT1M2109DOC/document/index.html.

[3] Georg Brandl. Sphinx, Python documentation generator. http ://www.sphinx-doc.org/en/stable/.

[4] Ben Collins-Sussman, Brian W. Fitzpatrick, and C. Michael Pilato. *Gestion de versions Subversion*.

[5] Ministère de l'enseignement supérieur et de la recherche. *Programme pédagogique national du DUT Réseaux et Télécommunications*, 2013.

[6] The tortoiseSVN team. TortoiseSVN, the coolest interface to (Sub)version control. https ://tortoisesvn.net/.