

Un projet Jeu sérieux pour approfondir l'apprentissage de la programmation en première année à l'université

Élisabeth Delozanne¹, Pierre Jarraud¹, Mathieu Muratet²
elisabeth.delozanne@upmc.fr, pierre.jarraud@upmc.fr, mathieu.muratet@irit.fr

¹ L'UTES-UPMC-Sorbonne Université (Paris, France)

²IRIT (Toulouse, France)

Résumé. Cet article décrit l'utilisation du jeu sérieux « ProgAndPlay » comme support pour un projet de programmation en C dans le cadre d'une UE transversale en première année à l'Université Pierre et Marie Curie. Il s'agit du compte-rendu d'une pratique que nous estimons positive et qui montre comment nous avons intégré un outil issu de la recherche comme support à une pédagogie de projet sur un thème plébiscité par les jeunes : un jeu vidéo.

Mots-clés: Jeux sérieux, Programmation, Instructions de base, Pédagogie de projet, C2I

Introduction

Le projet de programmation présenté ici s'appuie sur l'environnement Prog&Play (Muratet, Torguet, Viallet, & Jessel, 2011) qui permet à des étudiants, débutants en programmation, de programmer des unités d'un jeu de stratégie temps réel. Cet environnement, du point de vue de la pédagogie de la programmation en L1 (première année de Licence), nous a intéressés car il permet aux étudiants de mettre en pratique les connaissances de base en algorithmique (affectation, fonctions, alternatives, boucles) sur un problème ouvert et, a priori, motivant pour eux : remplir une mission dans un jeu. Notre préoccupation d'enseignant en Informatique était de fournir aux étudiants un cadre dans lequel ils puissent analyser un problème par eux-mêmes et programmer une solution sans que le problème soit par avance décomposé en sous-problèmes dans l'énoncé comme c'est le cas dans les sujets de TP prévus pour être résolus en moins de deux heures par une majorité d'étudiants. Notre préoccupation en tant que membres d'une équipe chargée de développer l'utilisation des TICE à l'UPMC (Université Pierre et Marie Curie-Paris 6) était d'étudier en quoi un jeu sérieux pouvait servir la pédagogie universitaire et pour quel type d'étudiants.

Le cadre du projet

En première année de licence à l'UPMC (Université Pierre et Marie Curie-Paris 6) les étudiants doivent choisir des UE dites transversales. L'une d'elles « Utiliser les TICE dans sa formation » a pour objectif d'améliorer la productivité des étudiants par l'utilisation des TICE, en particulier en les préparant au C2i niveau 1 (Certificat Informatique et Internet). Elle comporte une partie de travaux pratiques sur des fonctionnalités avancées des outils bureautiques et sur l'utilisation de l'ENT de l'université, et une partie projet. Depuis deux ans, nous avons proposé un projet « Jeu sérieux » qui s'appuie sur Prog&Play, à des étudiants de la filière MIME (Mathématiques-Informatique-Mécanique-Électronique) qui suivent par ailleurs une initiation à la programmation impérative en C. À l'UPMC l'enseignement de la programmation débute au premier semestre par une initiation à la programmation fonctionnelle (SCHEME) et, au second semestre, par une initiation à la programmation impérative où les étudiants ont le choix entre une UE en langage C ou une autre UE en VB. Dans ce cadre, le projet Prog&Play comporte 4 objectifs. L'objectif concernant les TICE consiste à faire réfléchir les étudiants sur les apports et les limites d'un jeu sérieux. L'objectif concernant la programmation est celui d'une pédagogie de projet : mettre en œuvre les concepts de base de la programmation sur une application d'une certaine importance (ici un jeu de stratégie). L'objectif concernant la communication vise à apprendre à exposer le résultat d'un projet en

s'appuyant sur un diaporama et à rédiger un rapport de fin de projet. Enfin, le dernier objectif concerne la préparation et l'obtention de la certification C2i niveau 1.

Cette UE a concerné huit étudiants la première année et treize la seconde année. Il ne s'agit donc pas ici, d'une étude statistique, mais d'un compte-rendu d'une expérience qui nous semble positive. Tous les documents sont en ligne sur le projet de recherche et l'environnement Pro&Play (Muratet, 2011) et sur le projet de programmation à l'UPMC (Delozanne, 2011).

La mise en place du projet

Les 4 premières séances de l'UE « TICE et Jeu Sérieux » sont consacrées uniquement à la préparation du C2i, le temps que les étudiants se soient familiarisés avec les notions de base de la programmation impérative (affectation, conditionnelles, répétition et fonctions) dans l'UE d'initiation à la programmation impérative en C qui débute en parallèle au second semestre. Le projet Prog&Play débute ainsi deux semaines avant le partiel de mi-semestre de C.

La première séance en présentiel débute par une présentation du projet ; puis les étudiants prennent en main le jeu pendant une demi-heure ou trois quarts d'heure sur des portables où il est installé. En effet, avant de leur demander de programmer les entités du jeu, nous estimons indispensables de leur faire découvrir les mécanismes du jeu. Dans une pré-expérimentation, nous avons présenté les entités, et les sujets nous avaient indiqué que la découverte des entités et de leurs possibilités faisait partie de l'intérêt du jeu. Le projet Prog&Play s'appuie sur le jeu Kernel Panic dont l'univers s'inscrit dans la thématique informatique. Les joueurs ont pour objectif de prendre le contrôle de diverses unités telles qu'un Kernel, des Bits, des Bytes, etc. Chacune de ces unités possède plus ou moins de résistance, de force de frappe et de vitesse de déplacement. Chaque joueur doit exploiter au mieux ces différentes unités et leurs caractéristiques pour mettre en œuvre des stratégies défensives ou offensives selon leur imagination ou leur habileté. Nous avons ainsi constaté qu'après quelques tâtonnements, tous les étudiants de L1 se prenaient au jeu (contrairement à des stagiaires de formation continue plus âgés). Dans une deuxième partie, l'enseignante présente les missions. Les six premières missions sont très simples : déplacer une unité vers un point donné (appeler une fonction avec les bons paramètres), trier des entités (conditionnelle), regrouper les unités (répétition), réparer des unités endommagées (répétition et conditionnelle). Elles ont pour but d'amener les étudiants à explorer les diverses fonctions de la bibliothèque qui leur seront utiles pour la mission finale suivante. Les étudiants testent d'abord des solutions dans un langage algorithmique simple. Ce langage algorithmique permet de voir comment commander les unités sans entrer dans les détails des structures de données en C et de la bibliothèque des fonctions du jeu (Tableau 1).

Dans la deuxième séance du projet, toujours en présentiel, l'enseignante apporte quelques compléments de C (type utilisateur, structures, enum, compilation séparée) et, sur des missions simples, présente les principales fonctions de la bibliothèque Prog&Play. L'enseignante force les étudiants à se mettre en binôme. Cette étape est difficile à négocier car les étudiants ne se connaissent pas et considèrent souvent, a priori, le travail en groupe comme un handicap. Les trois séances suivantes sont réservées au travail en binôme pour finir les six premières missions qui sont très faciles et s'attaquer à la mission 7 qui est le véritable sujet du projet. Ces séances ne sont pas obligatoires : ne viennent que les étudiants qui le souhaitent. En pratique, à deux exceptions près, les étudiants qui ne viennent pas sont ceux qui abandonnent. Les autres viennent et échangent entre eux et avec l'enseignante. Entre les séances, seuls deux binômes d'étudiants ont envoyé des courriels pour demander de l'aide. L'avant dernière séance est prévue pour une mise en compétition des programmes des étudiants qui jouent en réseau. En fait, pour des raisons techniques, la première année les étudiants n'ont pas pu jouer en réseau, et cette année, leurs programmes n'étaient pas encore au point ou bien n'étaient pas conçus pour. Les étudiants avaient testé leur programme en jouant contre l'ordinateur et en fixant les positions de l'ennemi au lieu de les passer en paramètres. Lors de cette séance, les étudiants doivent rendre leur programme afin d'avoir un retour de l'enseignante lors de la dernière séance. Seul un binôme a respecté cette consigne. Lors de la dernière séance, chaque binôme fait un court exposé de 15 minutes pour présenter les stratégies utilisées, les difficultés rencontrées et les apports du projet.

Tableau 1. Programmes de la mission 1 (déplacer une unité à une position précise)

Langage algorithmique	Langage C
<pre> // Mission 1 // déplacer une unité à une position précise #include "PP_ALGO.h" Debut OUVRIR_JEU; PREMIERE_UNITE; DEPLACER_VERS(1983, 1279); FERMER_JEU; Fin </pre>	<pre> #include "PP_Client.h" #include "constantList_KP4.1.h" int main (){ // définition de la position cible PP_Pos p; p.x = 1983.0; p.y = 1279.0; // ouverture du jeu PP_Open(); // récupérer l'état courant du jeu PP_Refresh(); // récupérer la première unité PP_Unit u = PP_GetUnitAt(MY_COALITION, 0); // ordonner à l'unité courante de se déplacer PP_Unit_ActionOnPosition(u, MOVE, p); // fermer le jeu PP_Close(); } </pre>

Avis des étudiants

Un questionnaire de satisfaction (Delozanne, 2011) montre que les étudiants sont demandeurs de ce type de pédagogie, et majoritairement, conseilleraient l'UE à un camarade. Plutôt que de donner des moyennes sur de petits effectifs, voici un florilège des conclusions de leur rapport écrit. Les étudiants mettent en avant l'aspect « application concrète », l'aspect « motivation », l'aspect projet et quelques fois des changements de représentation sur la programmation.

Problème concret :

<p>« un projet très intéressant :</p> <ul style="list-style-type: none"> • Premier contact avec de la programmation « appliquée » • Niveau et rythme adapté pour les débutants • Nécessité de se débrouiller • Le cadre ludique permet de maintenir la motivation 	<p>« Cet enseignement nous a tout d'abord montré une application de la programmation. Bien que nous ne connaissions qu'une infime partie de l'informatique. Le principe des serious game est donc particulièrement novateur, car il permet non seulement de s'amuser à un jeu vidéo, mais en même temps de développer des programmes permettant de devenir plus forts à ce même jeu. »</p>	<p>« Nous avons pu nous exercer à une pratique qu'on effectue depuis un an : la programmation. Cependant cette fois nous avons une application concrète à observer. Grâce à nos programmes nous sommes parvenues à faire se déplacer des unités, à en construire d'autres. En plus de cet aspect technique nous avons dû réfléchir à différentes stratégies, les comparer afin de savoir laquelle choisir. Et au final même si nous avons trouvé une solution qui marchait il y avait toujours un moyen ou un autre pour l'améliorer. »</p>
---	--	---

Motivation :

« Le code de la 7 ^{ème} mission a été réellement très long à établir. Beaucoup, d'idées ont du être abandonnés et revues dès le début. Au total, des centaines de lignes de code ont été écrites. De nombreuses fois j'ai vu mon armée se faire détruire entièrement. Cependant le fait d'approcher de plus en plus de la victoire à chaque tentative m'a tenu en haleine. Et c'est grâce à cette tension que j'ai fini par la réussir. C'est un <i>serious game</i> réussi, dans le sens, où j'ai appris beaucoup de choses, notamment, de nouvelles manières de penser les boucles, qui me seront très utiles pour le futur. C'est donc une UE que je recommanderais aux autres étudiants. »	« Ce projet m'a permis de découvrir la programmation dans un domaine que j'apprécie beaucoup c'est-à-dire les jeux vidéo et ça a été une bonne surprise pour moi de découvrir que dans ce module nous allions en faire. Cela m'a motivé encore plus pour la programmation car même si ça peut paraître ennuyeux pour certain moi je trouve ça très passionnant. La programmation est propice à l'imagination comme dans ce que nous avons à faire dans ce module qui a été de faire sa stratégie pour battre l'adversaire et c'est ça qui a été le plus intéressant à mon goût. »
---	---

Projet, initiative, Travail en équipe :

« La « Force » et la « Faiblesse » de ce projet est, nous pensons, le fait qu'il demande beaucoup d'initiatives et de patience. « Force » car l'étudiant qui n'abandonne pas et fait de nombreux essais, en sort enrichi de cette expérience. « Faiblesse » car l'étudiant qui n'aurait pas su prendre des initiatives, qui n'aurait pas su lever le blocage, n'apprendrait rien. Il n'y a en effet aucune connaissance théorique dans ce qui a été enseigné. C'est pour cela que nous pensons que cette UE est très formateur et est orienté « pratique ». Nous pensons qu'il devrait y avoir beaucoup plus d'UEs qui nous apprenne à développer cet esprit d'analyse, d'investigateur que nous avons pu développer dans cette UE de jeux sérieux. »	« Au fil de cette aventure, nous avons repensé notre projet, nous nous sommes remis en question. Et tout cela, dans le seul but de parvenir à un projet concret, développé, concis, simple et limpide. Malgré les difficultés, les problèmes rencontrés, notre entente constitua l'une des plus grandes solutions, et édifia notre avancé. Achevé, notre projet bâtit un solide édifice quant à l'apport en stratégie et en programmation que nous étions loin de nous imaginer. Ce projet fut donc un régal malgré quelques petits points noirs. » (ne sont pas précisés)
---	--

Intérêt/Limites de la programmation :

« Pour moi les jeux, l'informatique, c'était un peu magique. Là je vois comment ça marche. »	« Au niveau des apports, ce projet nous a permis de renforcer notre capacité à programmer en langage C, et d'apprendre quelques astuces comme comment créer une bibliothèque. Ce projet permet de réfléchir à la création d'une intelligence artificielle, et aux limites de la programmation. C'est impossible de recréer complètement la manière dont un joueur humain peut s'adapter à toutes les situations. »
--	--

Quant à ceux qui ont abandonné, ils n'ont pas réussi à installer l'environnement de programmation associé et n'ont pas demandé d'aide ou bien ils ont aussi abandonné l'UE de C.

L'avis des enseignants

Les deux années, nous avons été surpris par la diversité des stratégies envisagées par les étudiants. À noter : seul un groupe de filles s'est préoccupé de préserver la vie des unités. Les filles sont peu nombreuses dans cette UE (4 sur 21) mais, elles sont aussi moins nombreuses dans les UE informatiques. Elles jouent autant que les garçons (contrairement à ce que les enseignants pensaient a priori). Les commentaires relevés dans les rapports et l'attitude de tous les étudiants pendant les séances de travail montrent qu'à part les étudiants qui ont abandonné, les étudiants présents se sont investis dans le projet. Ce qu'ils déclarent apprécier est généralement ce que les étudiants apprécient dans un projet (autonomie, travail en équipe, un « vrai problème à résoudre », la diversité des solutions envisagées) et le fait que le domaine d'application soit un jeu (Cf. le florilège).

Concernant les apprentissages, dans cette UE transversale, les étudiants ont appris à mettre en œuvre les compétences du C2I, à travailler en binôme et ont développé leurs compétences en programmation. Sur ce dernier aspect, notre idée était d'étudier s'ils avaient progressé entre le partiel de l'UE de C qui a lieu au tout début du projet et l'examen de C qui a lieu deux semaines après la fin du projet. Le faible nombre d'étudiants nous empêchent de tirer des conclusions et les tableaux de leurs résultats (Cf. Annexe) montrent que, s'il n'y a pas d'effet négatif manifeste, les effets positifs sur les notes ne sont pas, non plus, manifestes. Cependant, si le projet ne met pas en œuvre de connaissances vraiment nouvelles, il confronte les étudiants à des situations différentes de celles proposées en TD et en TP. Dans le projet Prog&Play, ils ont à leur charge de décomposer par eux-mêmes le programme en sous-programmes, d'écrire des boucles imbriquées, des boucles d'attente, de découvrir seuls une bibliothèque de fonctions et d'y trouver les fonctions qui peuvent servir leur dessein. Certains ont découvert qu'un tableau déclaré localement à une fonction n'était pas utilisable à l'extérieur de la fonction. Ils ont été confrontés à l'intérêt de bien documenter des fonctions et, aussi, à définir des paramètres plutôt que de donner des valeurs fixées dans le marbre. Toute chose qu'on leur enseigne bien sûr dans l'UE d'initiation au langage C, mais dont ils ne réalisent pas l'importance tant qu'ils n'ont pas été confrontés au problème. L'intérêt majeur de ce projet par rapport à d'autres projets à ce niveau d'études, c'est l'aspect motivation : le sujet les tient en haleine et ils s'investissent, et donc, à notre sens, ils apprennent ; en tout cas, ils pratiquent pendant une vingtaine d'heures supplémentaires (d'après leur déclaration) ce qu'on leur enseigne par ailleurs. Un problème lié à l'autonomie laissée aux étudiants doit cependant être relevé : certains utilisent des procédures erronées que l'enseignante détecte trop tard, c'est-à-dire, au moment de la remise du programme car ils sont peu nombreux à respecter planning de remise des documents au fur et à mesure.

En comparant la qualité des programmes obtenus et les notes des étudiants à l'UE de C (en Annexe), bien évidemment, les bons étudiants en C (ceux qui se situent dans le premier tiers de la promotion) produisent les meilleurs programmes. Des étudiants moyens et même faibles réussissent à écrire un programme acceptable avec l'aide des enseignants et d'autres étudiants. Cependant, le projet ne fait pas de miracle, en particulier pour un étudiant très faible, d'autant que les compétences mises en œuvre dans le projet (par exemple décomposer un programme en sous-programmes) ne sont pas celles qui sont testées à l'examen qui est constitué de petites questions à réponse courte. De plus, dans certains binômes, la répartition du travail est, comme souvent, l'un programme et l'autre rédige les documents. Mais dans notre expérience, celui qui ne programmait pas, était quand même capable de discuter de la stratégie de jeu et avait apparemment testé le programme de son co-équipier.

Si, par rapport à nos préoccupations pédagogiques initiales, le dispositif nous semble intéressant, les améliorations suivantes pourront être apportées. Pour lutter contre l'abandon, il faudrait d'abord simplifier la procédure d'installation de l'environnement de programmation (problème de l'UE de C) et aussi obliger les étudiants à passer par le forum plutôt que par des courriels individuels aux enseignants afin de faire bénéficier tous les étudiants des conseils et afin de les encourager à poser des questions ou à faire corriger des morceaux de code. Enfin, il faudrait être plus strict sur le rendu des premières missions (et pas simplement la dernière) pour détecter des erreurs graves (par exemple des boucles qui ne s'exécutent qu'une seule fois) et faire un retour aux étudiants sur leur programme afin qu'ils puissent corriger ces erreurs dans la mission finale. Seuls, les étudiants qui ont respecté les délais de livraison (un seul binôme) ou dont le rapport était insuffisant (un autre étudiant) ont bénéficié d'un tel retour.

Conclusion

L'utilisation du jeu sérieux Prog&Play nous semble difficile à intégrer dans un enseignement classique (cours-TD-TP) d'introduction à la programmation en raison du temps nécessaire à la prise en main du jeu et de la bibliothèque pour programmer les unités ; par contre, comme support à une pédagogie de projet, c'est, à notre sens un environnement tout à fait adapté aux étudiants qui suivent un cours d'initiation à la programmation. Il leur fournit un terrain d'application et d'approfondissement de leurs connaissances sur un sujet qui les intéresse : un jeu de stratégie.

Remerciements

Merci aux ingénieurs de LUTES pour le support technique.

Bibliographie

- Delozanne, É. (2011, avril). Projet Prog&Play : jeux sérieux pour programmer en langage C. Consulté le 21 juillet, 2011, de <http://lutes.upmc.fr/delozanne/2010-2011/PP-2011.htm>
- Muratet, M. (2011). Présentation de Prog&Play. Consulté le 19 juillet, 2011, de <http://www.irit.fr/~Mathieu.Muratet/progAndPlay.php>
- Muratet, M., Torguet, P., Viallet, F., & Jessel, J.-P. (2011). Evaluation d'un jeu sérieux pour l'apprentissage de la programmation. *Revue d'intelligence artificielle*, 25(2), 175-202.

Annexe : Évaluation par binôme des missions Prog&Play des participants et leur résultats dans l'UE d'initiation à la programmation en C (LI115)

Table 2. Projet de 2010 (début 17/3/2010)

Binômes	Missions Prog&Play rendues	Évaluation du Programme Mission 7 12/5/2010	Partiel LI115 (31/3/2010) Note sur 20 / Rang sur 294	Examen LI115 (7/6/2010) Note sur 70 Rang examen sur 296
1-Garçon	1-6	Non rendu	19/16	62.5/12
1-Garçon	1-6		12.5/58	65.63/4
2-Garçon	7	Très bien	11/88	56.33/40
2-Garçon	7		17/5	61.7/17
3-Fille	1-6 malade 2 mois	Non rendu	10.5/98	46.63/102
3-Garçon	7	Très bienB	16.75/9	63.39/8
4-Garçon	7	Bien	9.66/126	51.5/67
4-Garçon	abandon		10/112	52,63/59

Table 2. Projet de 2011 (début le 16/3/2011)

Binômes	Évaluation Programme Mission 7	Partiel LI115 30/3/2011 Note sur 40 rang sur 394	TP solo LI115 4 au 8 avril 2011 Note sur 20 / rang sur 353	Examen (6 /6/2011) Note sur 70 Rang sur 370	Examen session Note sur 70 Rang sur 163
1-Garçon	Mauvais	Non inscrit	Non inscrit	Non inscrit	Non inscrit
1-Garçon		10.5/353	abs	12.75/289	22.25/163
2-Garçon	Bien	20.25/226	11.5/221	43.25/91	
2-Fille		20.75/216	10/252	9.25/307	31.07/72
3-Garçon	abs	19.25/246	7.5/297	4/340	18.5/118
3-Garçon	Passable	20.75/216	14.5/143	20/233	54.25/5
4-Garçon	Bien	30/71	17.5/58	45.25/72	
4-Garçon		32.4/35	19.6/18	58.75/20	
5-Fille	Assez Bien	22.75/182	20/1	19.75/236	
5-Fille		30.2/70	14.7/142	41/103	
6-Garçon	Très Bien	38/5	20/1	51,25/43	
6-Garçon		21.25/205	19/22	54/34	
7-Garçon	Très bien	28.4/95	14.2/163	29.5/173	