



Étude de l'intégration d'un jeu sérieux pour l'enseignement de la programmation dans différents contextes universitaires

► **Mathieu MURATET** (Grhapes, Suresnes), **Élisabeth DELOZANNE** (LIP6, Paris), **Fabienne VIALLET** (UMR EFTS, Toulouse), **Patrice TORGUET** (IRIT, Toulouse)

■ **RÉSUMÉ** • Cet article étudie l'utilisation sur une large échelle (388 étudiants, 22 enseignants), dans 7 différents contextes et sur plusieurs sessions, du jeu sérieux *Prog&Play*, logiciel développé à l'IRIT et libre d'accès. Après une analyse a priori du domaine, une enquête auprès d'étudiants sur leur pratique des jeux vidéo et une enquête auprès des enseignants pour prendre en compte leurs contraintes, *Prog&Play* a été conçu pour accompagner un enseignement de la programmation à des étudiants débutants. Une première étude s'appuyant sur un questionnaire auprès des étudiants après utilisation du jeu, a montré que l'intérêt des étudiants pour *Prog&Play* ne dépend pas uniquement de la qualité intrinsèque du jeu mais est corrélé avec le dispositif d'enseignement dans lequel le jeu a été utilisé. Dans cet article, après une présentation de l'ensemble du projet, nous analysons les commentaires de 56 étudiants et 17 enseignants, ayant utilisé le jeu, pour explorer les conditions de l'adoption de ce jeu sur la durée dans un dispositif d'enseignement universitaire.

■ **MOTS-CLÉS** • jeux sérieux, enseignement de la programmation, usages d'un jeu sérieux, appropriation par les enseignants.

■ **ABSTRACT** • *This paper studies a large-scale use (388 students and 22 teachers), in various contexts and during several sessions of Prog&Play, a free serious game specially designed to teach computer programming fundamentals. In earlier quantitative studies, we found, through a students' motivation survey, that the students' interest for Prog&Play was not only related to the intrinsic game quality, it was also related to the teaching context. In this paper, after presenting the project, we investigate conditions of the game adoption in different university teaching settings. Results from a qualitative study of 56 students' and 17 teachers' comments, suggest guidelines for a successful implementation of Prog&Play.*

■ **KEYWORDS** • *serious games, programming teaching, serious games use, teachers' appropriation*

1. Introduction

Zyda (2005) définit un jeu sérieux comme un défi mental joué avec un ordinateur, qui respecte des règles spécifiques et qui s'appuie sur le plaisir pour atteindre des objectifs de formation, d'éducation, de santé ou de communication. De nombreux travaux ont étudié l'effet des jeux sérieux sur la motivation (Bizzocchiet Paras, 2005). D'autres travaux portent sur la mise au point de modèles et d'outils pour faciliter leur conception (Marfi-si-Schottman *et al.*, 2010 ; Yessad *et al.*, 2010), en particulier par des enseignants (Marchiori *et al.*, 2012 ; Marne *et al.*, 2013), ou pour suivre l'apprenant-joueur et évaluer l'évolution de ses compétences (Thomas *et al.*, 2011). D'autres recherches se penchent sur leur efficacité pour l'apprentissage en comparaison avec d'autres méthodes. Par exemple Stizmann (2011), dans une méta-analyse récente des publications sur ce sujet, met en évidence des caractéristiques des jeux et des caractéristiques du contexte pédagogique qui influencent l'effet sur l'apprentissage. Les jeux doivent être ludiques et l'apprenant doit agir et pas seulement lire ou écouter. Quand au contexte, il doit permettre un accès fréquent au jeu et les jeux sont plus efficaces comme complément d'une autre méthode pédagogique qu'utilisé comme seule méthode d'apprentissage. Cette auteure appelle de ses vœux des études détaillées et précises sur les caractéristiques des jeux et les contextes pédagogiques où ils s'avèrent favoriser les apprentissages.

Dans cet article, nous étudions comment un même jeu sérieux peut être utilisé avec profit dans certains contextes pédagogiques et rejeté ou tout au moins abandonné dans d'autres contextes. Le projet *Prog&Play* a pour objectif de motiver les étudiants dans leur apprentissage de la programmation en leur proposant de mobiliser leurs compétences *via* un jeu sérieux. Ce projet se fonde sur l'hypothèse que les jeux vidéo (i) attirent un grand nombre d'étudiants et (ii) constituent un contexte d'application des concepts et savoir-faire de base de programmation aussi riche mais plus motivant que les exemples mathématiques souvent proposés aux étudiants.

Une précédente étude (Muratet *et al.*, 2012) a mis en évidence que la motivation des étudiants pour apprendre la programmation avec *Prog&Play* ne dépend pas seulement des qualités intrinsèques du jeu, mais dépend aussi du contexte d'enseignement dans lequel le jeu est utilisé. Notre contribution dans cet article ne cherche pas à montrer la supériorité d'un jeu sérieux par rapport à un enseignement traditionnel, mais à étudier les rapports entre les caractéristiques du jeu et celles des différents

contextes d'enseignement dans lesquels il a été utilisé pour comprendre ce qui a fait obstacle à son intégration ou, au contraire, ce qui a facilité son appropriation par certains enseignants et conduit à son adoption plusieurs années de suite. Après une exploration de travaux connexes pour positionner notre travail, nous présentons l'environnement *Prog&Play* et les différentes mises à l'épreuve du jeu dans différents contextes universitaires. Nous analysons ensuite les commentaires recueillis par questionnaires auprès de 56 étudiants et 17 enseignants et concluons sur des recommandations pour une utilisation fructueuse de *Prog&Play*.

2. Positionnement scientifique du projet *Prog&Play*

Notre travail s'appuie, d'une part, sur les recherches concernant l'apprentissage de la programmation et, d'autre part, sur celles concernant les jeux sérieux pour faciliter cet apprentissage.

2.1. L'apprentissage des fondamentaux de la programmation

Un document mis au point par les principales associations scientifiques regroupant les informaticiens au niveau international (ACM et IEEE-CS, 2008, p. 38) précise les concepts à enseigner aux débutants en programmation. Le même document précise les compétences visées. Les étudiants doivent être capables : (i) d'analyser et expliquer le comportement de programmes simples ; (ii) de modifier et compléter des programmes simples ; (iii) de concevoir, de mettre en œuvre, de tester et de corriger des programmes comportant les constructions de base ; (iv) de choisir les constructions appropriées pour un problème donné ; (v) de décomposer un problème en sous-problèmes et de mettre en œuvre des fonctions pour les résoudre ; (vi) de décrire les mécanismes de passages des paramètres. Ces notions sont très généralement enseignées par le biais de cours magistraux, suivis par la résolution de séries d'exercices d'application sans lien entre eux, lors de séances de travaux dirigés papier crayon (TD) et de travaux pratiques sur machine (TP). Parfois, l'enseignement est complété par un projet personnel ou de groupe qui demande à l'étudiant de résoudre un problème un peu plus complexe. L'apprentissage est évalué par un examen écrit et par des devoirs rendus durant la session (McCracken *et al.*, 2001).

De nombreuses recherches ont mis en évidence les dysfonctionnements de ce type d'enseignement et les raisons des taux d'abandon et d'échec très élevés observés dans ce type de formation dans le monde entier. Elles mettent en cause, en particulier, la méconnaissance de la part des enseignants des difficultés cognitives auxquelles doivent faire face les

étudiants dans ce domaine et des stratégies d'enseignement inadaptées (Du Boulay, 1986 ; Jenkins, 2002 ; Kinnunen & Malmi, 2006).

En ce qui concerne l'enseignement, Pears *et al.* (2007) considèrent trois facteurs qui influencent le succès des étudiants : (i) le choix de l'enseignant de privilégier soit la résolution de problème, soit l'apprentissage d'un langage particulier, soit un environnement de développement particulier ; (ii) le choix du langage et du paradigme de programmation (fonctionnel, impératif, orienté objet) ; (iii) les différents outils utilisés pour accompagner l'enseignement, l'apprentissage et l'évaluation (outils de visualisation, d'évaluation automatique, environnements de programmation, micro-mondes ou tuteurs intelligents).

Du côté étudiant, peu d'étudiants considèrent que l'apprentissage de la programmation est facile. Jenkins (2002) a identifié plusieurs difficultés concernant la nature même de l'activité de programmation, mais aussi la façon dont elle est enseignée. En particulier, il met en évidence l'inefficacité de cours magistraux centrés sur les détails de syntaxe et le manque d'intérêt, pour les étudiants, d'exercices qui demandent des manipulations mathématiques simples d'ensembles de notes, de stocks ou de comptes bancaires. De nombreuses recherches ont montré qu'une approche efficace consiste à encourager les apprenants à travailler très tôt sur des tâches qui font sens pour eux (Greitzer *et al.*, 2007). Des outils comme *Scratch* (Maloney *et al.*, 2004) ou *Alice2* (Kelleher *et al.*, 2002) ont été mis au point et sont utilisés afin de rendre les premières expériences de programmation plus attractives pour les jeunes. Nos travaux ont le même objectif. Plus précisément, *Prog&Play* a un double objectif : (i) Ancrer l'apprentissage de la programmation dans un contexte qui intéresse et fait sens pour les étudiants (au moins pour une majorité d'entre eux) ; (ii) Proposer un outil aux enseignants qui les laisse libres de choisir la stratégie d'enseignement, le langage, le paradigme et l'environnement de programmation les plus adaptés à leur contexte.

2.2. Les jeux sérieux pour l'apprentissage de la programmation

La motivation pour poursuivre une tâche est une des principales caractéristiques mises en avant pour promouvoir les jeux comme outil d'apprentissage. Girard *et al.* (2012) étudient les résultats publiés concernant l'efficacité des jeux sérieux pour favoriser l'engagement des étudiants et les apprentissages. En ce qui concerne l'initiation à la programmation s'appuyant sur les jeux vidéo, deux approches sont mises en œuvre.

La première approche consiste à demander aux apprenants de programmer eux-mêmes un jeu vidéo. Chen et Cheng (2007) proposent aux étudiants un projet collaboratif pour programmer en C++, sur un semestre, un jeu de moyenne taille à l'aide d'un outil professionnel de développement de jeux vidéo. Gestwicki et Sun (2008) demandent aux étudiants de développer, en java, un jeu d'arcade en utilisant des patrons de conception dans l'environnement EEclone (2011).

La seconde approche consiste à demander aux étudiants de programmer les entités virtuelles d'un jeu existant soit dans une approche *storytelling* pour réussir une ou plusieurs missions, soit dans une approche compétitive pour que les entités virtuelles contrôlées par le programme du joueur-apprenant soient meilleures que celles programmées par ses adversaires (ordinateurs ou co-apprenants). Colobot (2001) est un exemple de l'approche *storytelling* où le joueur incarne le rôle d'un explorateur qui part à la conquête de planètes extraterrestres. Le joueur est plongé dans une scénarisation qui l'invite à programmer de petits robots (avec un langage orienté objet proche de C++) afin d'atteindre les objectifs fixés par chaque mission. Dans *Robocode*, Hartness (2004) propose une approche compétitive ; le joueur programme en Java (ou, avec les versions récentes, dans un langage compatible .NET) un tank pour combattre d'autres tanks programmés par les autres joueurs. Ce jeu est accessible aussi bien aux débutants (un comportement simple peut être programmé en quelques minutes) qu'aux programmeurs experts (une véritable intelligence artificielle prendra plusieurs mois à être développée). D'autres jeux adoptent cette approche compétitive, mais tous proposent des langages de programmation peu répandus dans le domaine de l'enseignement de la programmation : par exemple *Guntactyx* (Gun-Tactyx, 2008) utilise le langage SMALL, et *Robot Battle* (Robot Battle, 2009) un langage de script propriétaire.

Dans le projet *Prog&Play*, nous avons adopté une double approche. Le premier mode de jeu propose une approche *storytelling* où les étudiants ont à réussir des missions de difficultés progressives comme dans Colobot ; le second mode de jeu prépare les étudiants à réaliser un programme dans un contexte de duel (contre l'ordinateur ou contre leurs condisciples). Un tournoi peut alors être organisé pour déterminer le programme le plus performant dans une situation de jeu donnée. De plus, afin de faciliter son intégration, *Prog&Play* a été implémenté pour enseigner la programmation dans la plupart des langages utilisés actuellement dans les établissements français. Dans cet article, nous nous centrons sur

l'approche par missions dont les scénarios sont conçus en fonction d'objectifs pédagogiques précis. Le scénario d'utilisation de *Prog&Play* est le suivant : (i) Le joueur prend connaissance de l'objectif à atteindre pour une mission donnée ; (ii) il met au point une stratégie pour atteindre l'objectif ; (iii) il écrit des programmes dans un des langages pour lesquels la bibliothèque *Prog&Play* a été développée ; (iv) il observe les effets de son programme sur le jeu et très souvent il est amené à modifier itérativement son programme pour atteindre l'objectif de la mission.

3. Conception et implémentation de *Prog&Play*

Au début du projet, un questionnaire, diffusé auprès de 950 étudiants en informatique en première année à l'université (780 garçons et 170 filles), nous a appris que les jeux de stratégie temps réel (STR) étaient un type de jeu très populaire dans notre public cible (Muratet *et al.*, 2009). Nous avons donc choisi de fonder *Prog&Play* sur un STR au code source libre appelé Kernel Panic (2012). Ce jeu utilise une métaphore informatique : les unités du jeu représentent des bits ; des octets ; des assembleurs ; des pointeurs ; des noyaux... C'est un jeu en 3D, multi-joueurs ; la qualité du moteur de jeu sous-jacent permet de faire tourner le jeu sur des ordinateurs de salle de TP ordinaire tout en restant attrayant.

Dans un STR, le joueur ordonne à ses unités d'exécuter des actions (par exemple avancer, construire ou attaquer) en cliquant avec la souris. Nous avons modifié le jeu afin que le joueur donne ses ordres par programme. Les étudiants accèdent aux données du jeu par le biais d'une *Applicative Programming Interface* (API) que nous avons développée afin de les dispenser de gérer les problèmes complexes de synchronisation. Dans leurs programmes, ils utilisent ainsi les fonctions de haut niveau de la bibliothèque *Prog&Play* pour modifier l'état du jeu *via* des constructions simples (conditionnelles, itérations ou appel de fonctions).

Dans notre approche, il est important de laisser l'enseignant libre de choisir l'environnement de programmation et le langage. À l'heure actuelle, nous avons implémenté six versions de l'API pour les langages : ADA ; C ; Java ; Ocaml ; Scratch ; CompAlgo. Le logiciel est téléchargeable en libre accès (*Prog&Play*, 2012)

Dans le mode mission, le scénario proposé aux étudiants par *Prog&Play* est le suivant : « *Depuis un certain nombre d'années, une guerre secrète fait rage au sein même des ordinateurs. Des attaques ont régulièrement lieu contre d'innocentes victimes. Aujourd'hui c'est votre tour. Votre agresseur a capturé le contrôleur de votre souris. Vous devez le récupérer. Votre seule solution : la pro-*

grammation ». Pour atteindre l'objectif final, nous avons défini plusieurs missions de difficulté croissante. Les premières missions se centrent sur les instructions de base et servent à la prise en main progressive de *Prog&Play* et de sa bibliothèque.

Par exemple, la mission 5 (figure 1) consiste en une boucle pour rechercher une unité particulière dans l'ensemble des unités. La mission est ainsi proposée : « *L'assembleur vient d'apparaître sur la carte, aidez le à rejoindre le reste de votre armée. Déplacez l'assembleur, et uniquement lui, aux coordonnées (256, 811).* » Dans la dernière mission (mission 8), le problème est ouvert et le joueur doit définir et implémenter une stratégie pour remporter la victoire contre l'ordinateur.

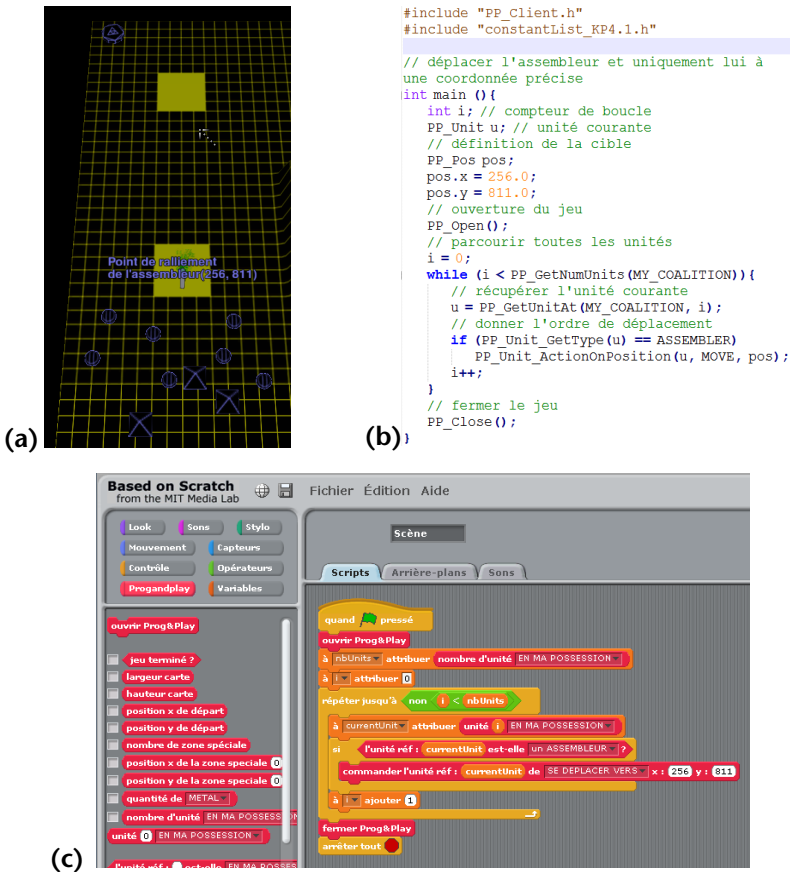


Figure 1 • Mission 5 : recherche d'une unité.

(a) Vue du jeu ; (b) Solution en langage C ; (c) une solution en SCRATCH

4. Questions de recherche et méthodologie

Une première étude quantitative à partir de questionnaires de satisfaction (Muratet et al., 2012) a suggéré que *Prog&Play* était plus apprécié des étudiants dans des contextes d'ateliers ou de projets que comme substitut à des séances de TP dans le cadre d'une formation classique. En effet, les données collectées suggèrent que l'utilisation d'un jeu sérieux comme simple outil d'illustration dans un enseignement classique ne semble pas avoir d'influence sur la motivation. En revanche, lorsque les contraintes temporelles sont moindres et que la continuité du jeu peut être respectée, les étudiants perçoivent mieux l'avantage inhérent du jeu sérieux : apprendre par l'expérience. Cette première analyse quantitative a permis de formuler ces suggestions qui méritent d'être approfondies par une analyse qualitative des commentaires des étudiants et des enseignants ayant utilisés le jeu. Dans cette seconde étude présentée ici, nous cherchons à comprendre les caractéristiques des situations d'enseignement s'appuyant sur *Prog&Play* pour favoriser des apprentissages et celles pour lesquelles l'utilisation du jeu n'apporte pas d'amélioration voir introduit une complexité ressentie comme néfaste. Nos questions de recherche sont les suivantes :

- Quelles sont les caractéristiques de *Prog&Play* et des situations d'usage mises en avant par les étudiants et les enseignants pour expliquer l'adoption de *Prog&Play* sur la durée ou son abandon après une première utilisation ?
- Quelles recommandations établir pour conduire à une utilisation favorable de *Prog&Play*.

4.1. Méthodologie d'observation

Les différentes expérimentations de *Prog&Play* ont été réalisées dans le cadre d'un dispositif expérimental (Cobb *et al.*, 2002) très souple qui consiste à laisser l'enseignant ou l'institution totalement libre dans la mise en place du dispositif. Notre objectif est en effet de pouvoir découvrir les différentes potentialités de l'outil en termes de pratiques enseignantes. L'idée a été de proposer à des enseignants ou à des institutions, d'intégrer *Prog&Play* dans leur enseignement ordinaire. Au départ, nous avons sollicité des enseignants à titre individuel, puis nous avons répondu à des demandes d'institutions ou d'enseignants désirant tester le jeu sérieux.

Certains enseignants ont utilisé *Prog&Play* en toute autonomie, cependant, les chercheurs sont parfois intervenus pour aider à la mise en place des séances d'enseignement, voire participer aux enseignements. L'aide des chercheurs a porté autant sur les parties techniques (installation de

Prog&Play) que sur les parties pédagogiques (élaboration de sujets de TP et leur animation).

Les seules contraintes imposées aux différentes personnes qui ont expérimenté *Prog&Play*, ont été (i) de nous préciser les caractéristiques de la formation où ils sont intervenus (niveau universitaire, institution, formation continue ou professionnelle, etc.) et de décrire la situation d'enseignement dans laquelle ils ont intégré le jeu (TD et TP d'algorithmique, cours de soutien, projet, etc.) ; (ii) de nous faire un compte rendu de leur expérience en répondant à un questionnaire très ouvert ; (iii) de faire passer auprès des étudiants, deux questionnaires en début et en fin d'expérience.

Le questionnaire passé en fin d'expérience comprend actuellement trois parties : une première concerne l'évaluation de leur degré de satisfaction vis-à-vis de *Prog&Play*, une seconde leur rapport à la programmation et la dernière leurs commentaires et suggestions. Dans les deux premières parties les réponses sont fournies sur une échelle de Likert en 5 points. Au fur et à mesure des expérimentations, nous avons fait évoluer les questionnaires (Questionnaire, 2008, 2010, 2013) afin de clarifier et mieux cibler les questions sur la motivation mais aussi sur les connaissances et compétences acquises.

4.2. Méthodologie de constitution du corpus de données

4.2.1. Nature et sélection des données

La multiplicité et l'hétérogénéité des expérimentations menées ont conduit à la constitution d'un corpus de données hétérogène. Tous les enseignants n'ont pas systématiquement transmis les questionnaires aux étudiants ou ne nous les ont pas fait remonter. Ainsi, sur les 11 contextes d'utilisation identifiés, nous en avons retenu 7, impliquant 388 étudiants et 22 enseignants.

Au cours d'une des expérimentations, l'enseignant a demandé aux étudiants de produire un rapport sur leur projet dans lequel ils décrivent les stratégies mises en œuvre pour remporter les missions, les difficultés rencontrées et, en conclusion, ce qu'ils pensaient avoir appris. Dans certains cas, nous avons pu conduire des entretiens avec les enseignants en fin d'expérimentation, dans d'autres cas ce sont des entretiens post séances informels que nous avons menés.

Dans cet article, pour comprendre les raisons d'adoption/abandon de *Prog&Play* par les enseignants et de la satisfaction/insatisfaction variable des étudiants, nous centrons notre analyse sur les parties ouvertes des

questionnaires enseignants et étudiants et sur les conclusions des rapports de projet étudiants.

4.2.2. Définition des variables indépendantes et des indicateurs

L'étude portant sur les conditions d'appropriation de *Prog&Play* par les enseignants, nous avons choisi comme variables indépendantes les éléments permettant de caractériser chaque situation, à savoir : (i) l'institution (IUT ou Université) ; (ii) le curriculum (IUT SRC, IUT info, IUT GII, L1) ; (iii) le langage support (Compalgo, C, Java, Ocaml) et le fait qu'il soit enseigné en premier langage ou en second ; (iv) le degré d'intégration des séances *Prog&Play* dans le cursus (fusionné aux TP ou en complément de la formation initiale), leur caractère d'obligation ou non et l'évaluation des étudiants ; (v) le temps consacré par les étudiants aux séances *Prog&Play* ; (vi) le degré d'implication des enseignants dans la mise en place des séances.

À partir d'une première lecture de l'ensemble du corpus, nous avons déterminé les indicateurs pertinents pour cette étude. Nous avons retenu trois dimensions dans les propos des étudiants et des enseignants : (1) la motivation et l'intérêt manifesté par les étudiants ou perçu par les enseignants ; (2) l'organisation des séances avec *Prog&Play* ; (3) le contenu enseigné/appris. Puis, pour chaque dimension nous avons repéré plusieurs catégories résumées dans le Tableau 1 et détaillées dans la section 5.

4.2.3. Méthodologie de traitement des données

Pour analyser ce corpus, nous avons procédé à une analyse de contenu manuelle de type catégorielle (Bardin, 1998). Les propos des enseignants et des étudiants ont été classés suivant la situation et la personne. Le texte a été segmenté au regard des catégories retenues. Par exemple, dans la phrase de l'étudiant 6 de la situation 5 « *le jeu m'a bien plu mais je n'aime pas du tout le langage de programmation qui est utilisé* », les auteurs ont codé la catégorie « engagement dans le jeu » (cf. catégorie 1.1) pour la première partie de la phrase et la catégorie « langage de programmation » (cf. catégorie 3.1) pour la seconde partie. Lors d'une première réunion, les chercheurs ont confronté leur segmentation des propos par situation et personne interrogée.

Tableau 1 • Nombre de commentaires des étudiants et des professeurs par dimension, par catégorie et par situation

		Situation		S1		S2		S3		S4		S5		S6		S7		Total	Total
		Étudiant	Professeur	É	Pr	É	Pr	É	Pr	É	Pr	É	Pr	É	Pr	É	Pr		
Catégories		Nb commentaires		4	3	2	2	1	23	12	9	1	1	Total		Total			
1- Motivation / Intérêt	1- Engagement dans le jeu (jeu, flow, investissement, jouer à programmer)	2+	1+	1+					3+	1= 1-	11+	1+	2+	28		173			
	2- Application de la programmation (programmer un jeu, concret, réel, utilité de l'informatique)	4+			1+ 1-			6+	6+ 1-	19+	1+			39					
	3- Créativité, réflexion, collaboration, autonomie	2+						1+		18+ 1-	2+	1+		25					
	4- Innovation pédagogique (apprendre en jouant, nouveauté, expérience à étendre)	5+		2=	1+	1+	10+	6+ 1-	8+	1+	3+			39					
	5- Motivation, intérêt		1+ 1-				6+ 5-	6+ 3= 3-	8+	1-				34					
	6- Orientation professionnelle ou universitaire						2+ 1-		5+					8					
Total		13	3	3	3	1	34	34	71	5	6								
2- Organisation	1- Préparation installation et prise en main de l'environnement (appropriation du jeu, charge de travail enseignant)				2-	1-	1+ 1-	1+ 3= 4-	1-	1-	2+		17		72				
	2- Gestion du temps		1-	1=			1-	4-	1+ 1-				9						
	3- Organisation des séances (rédaction des sujets, intégration dans le cursus, évaluation, semestre)			2=		1-	2= 5-	1+ 8= 7-	2+ 2= 1-	1-	1+		33						
	4- Gestion des différences et aide		1-				1+ 2= 1-	2= 2-	1=	1= 1-	1+		13						
Total			2	3	2	2	14	32	9	4	4								
3- Contenu enseigné	1- Langage (C/Caml)						3= 4-	1= 2-		1+			11		84				
	2- Facilité/Difficulté			1=			4-	12-	3=	1-			21						
	3- Notions de programmation (utilisation d'une API, abstraction des données, décomposition d'un programme, boucles, commentaires, tests)	1+			2-	1-		1= 4-	15+	1+			25						
	4- Influence sur l'apprentissage	2+				1=	2+ 1=	1+ 3= 4-	9+ 2-	1+	1+		27						
Total		3		1	2	2	14	28	94	29	4	1							
Total		16	5	7	7	5	62	94	109	13	11			329					

Remarques principalement positives
 Remarques principalement neutres

Remarques principalement négatives
 Aucune remarque recensée

Tableau 2 • Les différentes situations d'usage de *Prog&Play*

	Institution et curriculum	An- née	Nb participants	Langage, intégration des séances dans le cursus et implication des ensei- gnants	Temps de jeu	
S1	IUT A (Dpt Info, Semestre1)	2008	ens. : 2 ; étu. : 15	Compalgo, étudiants volontaires, atelier en complément de la formation initiale, aucun enseignant externe au projet	5 * 1h30	
S2	IUT B (Dpt Info, Semestre1)	2009	ens. : 2 étu. : 35	C, soutien pour les étudiants en difficulté en complément de la formation initiale, 1 enseignant externe au projet	3 * 1h30	
S3	IUT B (Dpt Info, S3)	2009	ens. : 2 étu. : 16	Java, étudiants volontaires, atelier en complément de la formation initiale, 1 enseignant externe au projet	3 * 1h30	
S4	IUT C (Dpt SRC, Semestre1)	2009	ens. : 2 étu. : 60	C, travaux pratiques obligatoires pour tous les étudiants, 2 enseignants externes au projet	5 * 1h30	
		2011	ens. : 1 étu. : 10	C, sessions de remédiation pour les étudiants en difficulté, 1 enseignant externe au projet	8* 1h	
S5	Université A (L1 Semestre1)	2009	ens. : 15 étu. : 99	OCaml, travaux pratiques obligatoires pour tous les étudiants, 14 enseignants externes au projet	2 * 2h	
S6	Université B (L1 Semestre 2)	2009	ens. : 1 étu. : 9 par année	C, étudiants volontaires, UE projet facultative en complément d'une UE initiation au langage C, 1 enseignant externe au projet	6 * 2h + travail personnel	
		2010				
		2011				
		2012				
		2013	ens. : 1 étu. : 19			
S7	IUT D (Dpt GEII, Semestre 1)	2009	Nb ens. : 1 étu. : 15 par année	C, projet intégré à une formation spécialement conçue pour un public désigné d'étudiants en difficulté, 1 enseignant externe au projet	5 * 2h	
		2010				
		2011				ens. : 1 étu. : 30 par année
		2012				

Chaque chercheur a ensuite attribué une valence positive, négative ou nulle aux segments selon que le commentaire est jugé favorable, critique ou neutre. Une confrontation entre les différentes analyses a été effectuée pour obtenir un consensus. En effet, étant donné le grand nombre de variables et la faiblesse des effectifs de données recueillis dans certaines situations, nous n'avons pas voulu effectuer de comparaison statistique. Le consensus étant obtenu entre les différents chercheurs, un tableau de synthèse a été établi permettant de rendre compte des valences des différents thèmes sur tout le corpus, classé par situation et personne interrogée. Le corpus annoté, résultat de ce travail, est consultable en ligne (Corpus, 2013).

4.3. Description du corpus de données : situations d'utilisation et participants retenus

Nous avons ainsi retenu sept situations différentes (S1 à S7) impliquant 388 étudiants et 22 enseignants (Tableau 2). Les sujets étaient des étudiants en informatique de première année d'université, ou bien, de première ou deuxième année d'IUT. Dans les situations 1, 2, 3 et 5, un membre de l'équipe de conception de *Prog&Play* faisait partie de l'équipe enseignante ce qui n'était pas le cas dans les autres situations. Dans les situations 1, 3 et 6, les étudiants étaient volontaires alors que dans les autres situations l'utilisation de *Prog&Play* était obligatoire, soit en substitut de TP classiques (situation 4, 5), soit pour un public désigné nécessitant un enseignement de remédiation (situation 7), soit à titre d'exercices complémentaires (situation 2), soit dans le cadre d'un projet (situation 6). Les situations 6 et 7 sont particulières dans la mesure où, dans deux contextes différents, les deux enseignants ont monté un enseignement spécifique fondé sur une pédagogie de projet autour de *Prog&Play*. Dans ces deux contextes, l'évaluation des étudiants tenait compte de la qualité des programmes produits dans l'environnement *Prog&Play* et d'une réflexion sur les apports du projet en termes d'apprentissage de la programmation. Ce n'était pas le cas dans les autres situations. Ces enseignants continuent d'utiliser *Prog&Play* chaque année depuis quatre ans.

Dans chaque situation, les étudiants jouaient et travaillaient avec *Prog&Play* dans les salles de TP où le jeu était déjà installé sur les machines et où ils bénéficiaient de l'assistance d'un enseignant. Dans la situation 6 cependant, les étudiants pouvaient terminer les missions en dehors des séances avec une aide, de l'enseignant ou de leurs pairs, par courrier électronique pour installer l'environnement sur leur ordinateur personnel ou pour mettre au point leurs programmes.

5. Résultats

Le Tableau 1 résume les résultats de notre analyse. Sur les 75 commentaires analysés (56 d'étudiants et 17 d'enseignants), nous avons retenu 329 remarques (199 d'étudiants et 130 d'enseignants). En ce qui concerne les valences, sur l'ensemble de ces remarques, 88 sont négatives et expriment des critiques ou des difficultés rencontrées, 46 sont neutres ou sont de l'ordre du conseil ou de la proposition et 195 sont positives. En ce qui concerne le contenu des remarques, nous les étudions selon les dimensions et catégories que nous avons retenues. Dans la suite de l'article, nous référençons les citations des étudiants et des enseignants de la manière suivante : SxEy référence une citation de l'étudiant « y » de la situation « x » ; SxPy référence une citation de l'enseignant « y » de la situation « x ».

5.1. Motivation

Les remarques les plus nombreuses concernent la dimension de la motivation pour laquelle nous avons distingué six catégories de remarques : 1) celles qui concernent l'engagement des étudiants par le jeu ; 2) celles qui considèrent *Prog&Play* comme un domaine d'application particulier de la programmation ; 3) celles qui se focalisent sur les compétences transversales mises en œuvre ; 4) celles qui font référence au caractère innovant de ce type de pédagogie ; 5) celles qui mentionnent l'intérêt et la motivation sans précision ; 6) celles qui font référence à une orientation professionnelle ou universitaire future.

5.1.1. Engagement dans le jeu

Nous avons classé dans la catégorie engagement des étudiants dans le jeu les remarques qui mentionnent l'aspect ludique.

« Très bon accueil des étudiants qui sont [...] pour certains démotivés pas la programmation. Les étudiants ont montré leur implication (notamment en travaillant chez eux alors que cela n'était pas demandé) et c'est déjà un gain important » (S7P1)

« Chaque mission réussie, est ressentie comme une victoire. On a envie de programmer, on a envie de réfléchir pour trouver une stratégie et gagner le jeu. » (S6E17).

« Pour le mot de la fin, nous pouvons dire que l'approche du jeu sérieux est très efficace, nous ne nous attendions pas à quelque chose pouvant susciter tant d'intérêt, mais plutôt quelque chose de très scolaire maquillé sous le thème du jeu. Nous avons pris vraiment plaisir, à faire le travail qui s'apparentait à un jeu et là est le tour de force. Nous ne devons plus faire un programme pour tra-

vailler la syntaxe sur un énoncé imposé, mais pour faire marcher et donner vie à un jeu avec nos tactiques et donc nos propres énoncés, que nous pouvions rendre plus en plus complexes. Il nous a fallu nous armer de patience mais lorsque le résultat était là il était très gratifiant de voir le fruit de l'élaboration de nos tactiques. » (S6E21)

« Le code de la 7^e mission a été réellement très long à établir. Beaucoup, d'idées ont dû être abandonnées 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. » (S6E4)

Cette dernière citation illustre bien la notion de *flow*, définie comme un état où le joueur perd la notion du temps et se retrouve absorbé par l'expérience du jeu (Csikszentmihalyi, 1991).

5.1.2. Application concrète de la programmation

Un des points qui revient très souvent est l'intérêt pour une application souvent qualifiée de « concrète » de l'informatique. Certaines remarques relèvent que *Prog&Play* montre l'utilité de la programmation.

« L'expérience est positive dans le sens où ça peut montrer aux étudiants que programmer est nécessaire (ne serait-ce que pour écrire un jeu). » (S5P4)

« À notre surprise, nous qui n'aimons pas les jeux vidéo, les étudiants ont trouvé qu'ils programmaient quelque chose de réel. » (S6P1)

« C'est un exemple concret où on comprend à quoi servent nos algorithmes. » (S1E2)

« 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. » (S6E3)

« Pour les néophytes dans la programmation, ça permet d'avoir une application directe autre que mathématique qui, je pense, nous permet une meilleure compréhension de la matière. » (S6E23)

Dans cette catégorie, quelques remarques soulignent l'intérêt de démystifier les jeux vidéo.

« C'est un peu comme quand on explique un tour de magie, on a l'impression de maîtriser quelque chose de plus. » (S5P11)

« Pour moi les jeux, l'informatique, c'était un peu magique. Là je vois comment ça marche. » (S6E8)

« On est dans l'envers du décor de notre enfance, on programme un jeu alors que d'habitude on y joue et pour une fois on y voit la complexité (avec tous les

cas qu'on doit étudier) mais aussi la possibilité de le faire, cela devenant accessible. » (S6E10)

Enfin, dans la même catégorie, nous avons aussi placé les remarques qui insistent sur la visualisation des résultats des programmes.

« Ils ont apprécié de jouer et de voir que leurs programmes avaient un effet sur le déroulement du jeu. » (S5P6)

« La visualisation du résultat des actions amène un côté didactique motivant pour les élèves. » (S5P9)

« On voit le résultat que l'on produit et ça change de la théorie du cours. » (S1E3)

5.1.3. Créativité, Réflexion, Travail collaboratif

Principalement dans les contextes de projets (S6 et S7) mais pas seulement, les enseignants et certains étudiants, mettent en avant que *Prog&Play* favorise l'imagination, la créativité, l'autonomie et aussi la collaboration.

« Nous avons été impressionnées par la diversité des stratégies envisagées par les étudiants. » (S6P1)

« On se sent dans un projet collectif et on n'a pas besoin de nous dire faites cette fonction de cette procédure. Car cela vient tout seul. [...] Alors qu'en TD on ne réfléchit pas, on suit simplement les questions. » (S1E3)

« La programmation est propice à l'imagination comme dans ce que nous avons à faire dans ce module qui était de faire sa stratégie pour battre l'adversaire et c'est ça qui était le plus intéressant à mon goût. » (S6E5)

« 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. » (S6E6)

« Ce projet a la particularité, contrairement aux TP machines d'informatique, de faire suivre un exercice sur plusieurs séances avec un binôme, nous avons dû apprendre à nous organiser. » (S6E21)

*« Certes, il y a eu des difficultés liés au fonctionnement pas très intuitif de *Prog&Play*, mais cela nous a appris à persévérer pour trouver la solution et à prendre de recul par rapport à notre façon de coder : il faut toujours se demander comment le programme marche, et remettre en question ce qui semble à nous logique. »* (S6E23)

5.1.4. Innovation pédagogique à étendre

De nombreuses remarques relèvent le côté innovant de ce type de pédagogie.

« Prog&Play a été bien accueilli par sa dimension innovante. » (S4P1)

« Bonne initiative mais c'est dommage que ce soit si long et si compliqué. Mais dans l'ensemble, les étudiants étaient quand même satisfaits car cela sortait de l'ordinaire. » (S5P8)

« Ce module devrait être intégré totalement à l'IUT car on s'améliore à une vitesse surprenante. » (S1E3)

« Suggestions ? Oula, oui, et plein !!! D'abord, étendre l'enseignement par jeux vidéo à tous les élèves, c'est une expérience remarquable d'audace et d'ingéniosité. Et apparemment, elle porte ses fruits... [clin d'œil] Pourquoi ne pas essayer le même système dans les autres matières ? Ce serait visionnaire [cool]. » (S5E18)

« Tout cela a bien sûr éveillé notre goût de l'informatique, nous espérons avoir une UE de la sorte dans notre parcours et pouvoir un jour aller plus loin dans la programmation ludique. » (S6E21)

5.1.5. Intérêt et motivation

Certains enseignants de la situation 5 et un étudiant relativisent l'intérêt d'un jeu (ou de ce type de jeu) pour étudier la programmation.

« Je suis malheureusement incapable de dire si ça les a réellement motivés. [...] Un jeu où l'on se tire dessus n'est pas non plus forcément un bon choix. [...] les filles n'étaient pas fana de ça » (S5P4)

« Le sujet a surtout eu l'air de motiver les plus joueurs d'entre eux (trop complexe pour les autres !). » (S5P9)

« Apprendre la programmation en se basant sur un jeu peut motiver certaines personnes, mais cependant, cela peut également avoir l'effet inverse sur les personnes qui ne sont pas du tout intéressées par les jeux. » (S2E3)

Dans la situation 5 où, à l'inverse des situations 6 et 7, les énoncés des missions étaient très fermés et directifs, des enseignants et des étudiants estiment les missions trop simples et ennuyeuses.

« En fait, je pense qu'ils ont tous été frustrés : soit ils n'arrivent à rien (car la moindre fonction à écrire repose sur l'utilisation de fonctions prédéfinies), soit ils trouvent qu'ils n'ont pas assez programmé. » (S5P3)

« Les étudiants étaient plus motivés au départ car il s'agit d'un jeu, ils ont bien aimé jouer à la première séance. Après ils sont tombés de haut, car même si

ça leur faisait plaisir d'arriver au bout d'une mission et qu'ils étaient réellement motivés à terminer, ils étaient tout de même frustrés car les missions avaient un but précis sans aucune marge de manœuvre. » (S5P8)

« Au final, on n'a pas réellement joué au jeu, on programmait et puis, 1 fois sur 10, un pion bougeait... »(S5E15)

« On ne crée pas vraiment le jeu, on fait juste des fonctions pour le déplacement et autre. » (S5E21)

Pendant, même dans cette situation, certains étudiants estiment que le jeu les a intéressés.

« J'ai adoré ces TP et je suis impatient de continuer au semestre 2. » (S5E8)

« Je pense que le fait d'apprendre à travers un jeu est intéressant : on voit, de suite les résultats de ce que l'on fait. Pour moi, cela a suscité un plus grand intérêt pour les TP. » (S5E16)

« Étant donné que la nouvelle génération est tournée vers les nouvelles technologies et en particulier vers l'informatique et les jeux vidéo, je trouve l'idée bonne pour motiver les étudiants de cette filière. » (S5E17)

5.1.6. Orientation professionnelle ou universitaire

Quelques remarques d'étudiants font référence à l'influence de Prog&Play sur leur orientation future.

« Très bonne idée d'utiliser un jeu vidéo pour la programmation. Ces TP ont renforcé mon envie de continuer en informatique. » (S5E10)

« Je me suis aperçu lors de ce premier semestre que l'informatique était un domaine qui ne m'intéressait pas du tout. » (S5E11)

« Au bout de 4 mois, notre vision en tant qu'étudiants a évolué. Dans le sens où notre connaissance en C s'est approfondie à travers la découverte de nouvelles bibliothèques, le travail de groupe a pris un rôle important, de la partie codage, des idées se sont échangées et on a découvert d'autres façons de concevoir l'informatique, ainsi que la préparation de notre avenir en entreprise où le travail d'équipe a une place importante. » (S6E10)

« Une des autres choses que m'a apportées cette UE mais cela est à titre personnel c'est qu'elle m'a donné envie d'en savoir plus sur un sujet qui avant ne me serait jamais venu à l'esprit : l'Intelligence Artificielle. » (S6E14)

5.2. Organisation du travail

L'organisation des séances était très variable selon les situations. Nous avons retenu quatre catégories de remarques : 1) celles qui abordent la préparation, l'installation et la prise en main de l'environnement ; 2)

celles qui traitent de la gestion du temps ; 3) celles qui portent sur l'organisation même des séances et leur intégration dans le cursus ; 4) celles qui concernent la gestion des différences entre étudiants et l'aide à leur apporter.

5.2.1. Préparation, installation et prise en main de l'environnement

Les enseignants estiment n'avoir pas passé plus de temps à préparer les séances de TP que pour des TP ordinaires.

« La charge de travail vient essentiellement du fait que c'est différent des années précédentes. Un sujet classique mais nouveau aussi, aurait également demandé une charge importante. Le surplus dû au fait que c'est du serious gaming n'est pas très important à mon avis et est essentiellement dû au fait qu'il faut un peu de temps pour maîtriser le nouvel environnement. » (S5P1)

« Ce genre de TP n'impose pas vraiment de charge supplémentaire quant à la préparation. Le surplus subsiste dans la prise en main de l'environnement Kernel Panic, mais pour le reste, c'est équivalent. » (S5P8)

Seule l'enseignante de la situation 6 mentionne des difficultés à installer le jeu sur des machines pour faire jouer les étudiants en réseau.

« En fait nous avons passé beaucoup de temps à préparer (3 jours entiers à deux) surtout pour surmonter les difficultés techniques (trouver des portables, installer, copier, tester, redéfinir les adresses IP pour créer un réseau local etc.) » (S6P1)

Par contre, dans la situation 5, certains commentaires indiquent que la prise en main de l'environnement *Prog&Play* par les enseignants était insuffisante.

« On peut la [l'expérience] répéter l'année prochaine, en espérant qu'au moins les enseignants se soient mieux habitués à l'environnement du jeu. » (S5P6)

Certains enseignants signalent aussi des difficultés pour identifier des stratégies de jeu et les mettre en œuvre.

« Difficultés rencontrées : Principalement, sans bien connaître le jeu, identifier des stratégies de solution à mettre en œuvre, expliquer aux étudiants comment mettre en œuvre une stratégie qu'ils avaient décidée (passer de la stratégie aux actions). » (S3P1)

5.2.2. Gestion du temps

Le temps de travail avec *Prog&Play* est évoqué. Pour des enseignants de la situation 5, les sujets ont paru trop longs et mal adaptés au temps alloué aux séances *Pro&Play*.

« Les TP étaient bien longs et par conséquent on ne couvrait que très peu d'exercices en séances. » (S5P5)

« Les TP étaient vraiment plus longs et plus difficiles que ceux de l'enseignement classique. À la limite le TP3 P&P aurait pu faire séance 3 et 4 (et zapper le TP4 ou le mettre en annexe...) pour que les étudiants puissent vraiment à la fois s'imprégner du jeu et comprendre ce qu'on attendait d'eux et arriver à quelque chose. » (S5P8)

Pour d'autres, au contraire, le temps a passé trop vite et les étudiants auraient voulu davantage de TP de ce genre

« Pour une fois c'est passé vite en programmant le jeu ! » (S2E2)

« Il faudrait qu'il y ait plus de missions, que la durée de l'atelier soit plus longue, afin de rentrer encore plus dans les détails de la programmation. Et que les dernières missions se fassent à plusieurs (en réseau). :) » (S3E2)

5.2.3. Organisation des séances

L'organisation des séances est abordée presque exclusivement par les étudiants et les enseignants de la situation 5 pour déplorer la longueur des sujets, pour mentionner le décalage entre les TP *Prog&Play* et les TP traditionnels et l'évaluation de la formation. Certains proposent des améliorations tant sur les sujets que sur l'organisation.

« Source du problème : pas le jeu en lui-même, mais le sujet. Il faut continuer à essayer le *Prog&Play*, mais il faut modifier les sujets de TP : les simplifier ou les rendre moins longs, moins compliqués. » (S5P8)

« Ce qui est peut-être dommage c'est que toutes les séances ne soient pas toutes basées sur le serious game, le déroulement des séances a l'air de manquer d'un peu de lisibilité pour les étudiants. » (S5P9)

« Je pense que *Prog&Play* serait plus adapté pour des étudiants plus avancés dans le cursus. En début de L2 par exemple, sous réserve de le faire dans un langage que les étudiants maîtrisent déjà, l'apprentissage de la bibliothèque et des outils associés à *Prog&Play* pourrait être plus facile. Dans ce cadre *Prog&Play* permettrait d'aborder des notions simples (d'IA par exemple) en offrant un cadre difficile à mettre en place autrement. » (S5P12)

« Un point négatif pour les TP : dans les fiches *Prog&Play*, il y a plus à lire qu'à écrire, je ne pense pas que ce soit le but premier d'un TP. » (S5E1)

« Je pense qu'on aurait dû varier les types de TP informatiques pour être plus complet, mais peut être avec des TP de *Prog&Play* plus complexes, avec une sorte d'intelligence artificielle, puisque utiliser la fonction se déplacer était assez simple. » (S5E22)

Dans les autres situations, l'organisation des séances ne semble pas avoir posé de problème et n'est pratiquement pas évoquée. Enfin certaines remarques évoquent l'intégration dans le cursus.

« *Nous aurions dû, par plus de préparation, beaucoup plus l'adapter (simplification des bibliothèques) au niveau des étudiants et l'intégrer plus étroitement aux séances et à l'évaluation du module dans lequel nous l'avons utilisé* » (S4P1)

« *L'UE nous a donné une idée de comment sont liées les jeux et la programmation. C'était une UE intéressante et plutôt relaxante comparée aux autres UEs. On a aussi appris comment un jeu sérieux peut être utilisé comme instrument didactique. On a pu améliorer nos capacités en C, et du coup les cours de LI115ⁱ étaient plus faciles.* » (S6E15)

Notons qu'un étudiant de la même situation est d'un avis contraire. C'est le seul à exprimer cette idée, qui n'en est pas moins légitime. En effet, il estime que le changement de contexte d'application n'est pas profitable car l'évaluation de l'UE de programmation en C (appelée LI115), qui se déroule en parallèle de l'UE *Prog&Play*, porte sur le contexte traditionnel.

« *Pour nous qui venons de découvrir le langage C ce semestre et qui débutons en informatique, ce jeu nous a permis de connaître une autre application de la programmation. Nous n'avions utilisé le langage C que dans le but de résoudre de petits problèmes mathématiques. [...] Cependant, un inconvénient que nous pourrions trouver à ce jeu est qu'il n'est pas vraiment pédagogique. Il ne nous apprend pas le langage C, il permet juste de l'appliquer. De plus, il n'aide pas à la préparation de l'examen qui nous attend à la fin du semestre en C, les exercices demandés étant totalement différents. Enfin, la bibliothèque que l'on utilise dans ce jeu est forcément différente de celle utilisée en classe, ce qui du coup ne nous aide pas à nous familiariser avec.* » (S6E11)

5.2.4. Gestion des différences et aide

Certains enseignants notent des différences interindividuelles et des difficultés à gérer les aides à apporter tout en maintenant la motivation.

« *Malgré tous mes efforts, certains étudiants n'ont rien compris : j'ai l'impression que ce genre d'exercice creuse encore plus le fossé entre ceux qui savent lire et les autres (qui sont malheureusement de + en + nombreux).* » (S5P3)

« *J'ai eu l'impression d'une influence différente suivant les étudiants : pour les étudiants déjà motivés par l'informatique et de niveau au moins correct, cela a été visiblement motivant ; pour les étudiants de niveau plus faible, l'utilisa-*

tion des bibliothèques a constitué une barrière supplémentaire, que j'ai dû leur aider à passer. » (S5P12)

« Un binôme (celui qui n'a pas réussi à programmer la mission 7) aurait souhaité plus de support, les autres ont au contraire apprécié devoir trouver par eux-mêmes. » (S6P1)

L'enseignant de la situation 7, quant à lui, pointe l'intérêt du dispositif dans un contexte hétérogène.

« [...] cela permet pour les plus faibles (souvent issus d'un "redoublement") de se motiver, et pour les plus forts (ceux qui par exemple arrivent de classes préparatoires en février) de développer de véritables stratégies dans Prog&Play. » (S7P1)

Les étudiants mentionnent à plusieurs reprises, l'importance de la médiation des enseignants (ou des pairs).

« Le seul point sur lequel je suis déçu c'est le fait que l'on ne soit pas assez "entouré". J'aurais préféré aller moins vite mais avec plus d'attention au cas par cas de la part du professeur car les difficultés ne sont pas les mêmes pour tout le monde. » (S2E1)

« Plus à lire qu'à faire en particulier dans les TP Prog&Play. Éventuellement ajouter des fiches de rappels / didacticiels sur le site de moodle pour ceux qui ne comprennent pas trop. » (S5E3)

« Les enseignants étaient là pour nous expliquer et nous pousser à tenter de faire les TP par nous-mêmes. » (S5E10)

« Pour la mission 7, au début on n'arrivait à réparer qu'une unité, il a donc fallu trouver le moyen de toutes les réparer, et c'est avec notre travail d'équipe et l'aide du professeur qui nous explique où cela ne fonctionne pas qu'on a réussi à gagner la mission. » (S6E10)

5.3. Contenu enseigné

Sur la dimension de la programmation, contenu d'apprentissage visé par *Prog&Play*, nous avons retenu quatre catégories de remarques : 1) celles qui concernent le langage étudié ; 2) celles qui évoquent la difficulté du travail (ou sa trop grande facilité) ; 3) celles qui mentionnent explicitement des notions de programmation ; 4) celles qui évaluent l'influence sur l'apprentissage.

5.3.1. Le langage de programmation étudié

Concernant le langage de programmation, les remarques concernent uniquement la situation 5. Au-delà des plaintes bien connues des

étudiants sur la lourdeur ou l'inutilité de Caml, des enseignants s'interrogent sur la compatibilité de *Pro&Play* avec l'enseignement d'un langage fonctionnel.

« Je pense que le *Prog&Play* est une bonne idée. Mais pas en l'état. Il arrive trop tôt, et le Caml ne me semble pas le meilleur langage pour utiliser cet outil. Peut-être que l'introduire au second semestre, pour les cours de C, serait plus intéressant. » (S5P2)

« Je crois que les concepts abordés qui ressemblent un peu à la programmation objet (pour agir sur une entité j'utilise des méthodes) leur échappe complètement et même en expliquant et en réexpliquant (..), certains ont du mal à comprendre et je ne suis pas certaine qu'ils aient assimilé le truc. Ça risque donc, pour certains, de les entraver plus qu'autre chose. Mais, pour la majorité, ça a marché... » (S5P4)

5.3.2. Difficulté/Facilité

Deux étudiants et six enseignants de la situation 5 estiment que le jeu est contre-productif, trop difficile, trop complexe et constitue une perte de temps.

« *Ce qui a handicapé le groupe Prog&Play : gestion de l'environnement. L'environnement n'est pas mal fait, je ne saurais pas comment l'améliorer. C'est juste un overhead inhérent au contexte du jeu.* » (S5P6)

« *L'apprentissage de l'environnement se fait au détriment de la programmation [...] avec le jeu on rajoute un niveau d'abstraction supplémentaire par rapport à un TP traditionnel.* » (S5P10)

« *Le jeu est quand même relativement compliqué pour un (premier) semestre de programmation.* » (S5E21)

D'autres, au contraire, mentionnent une absence de difficultés.

« *Aucune difficulté rencontrée sur l'utilisation de l'environnement de développement.* » (S7P1)

« *Ce fût plus facile pour moi de construire les programmes à travers un jeu.* » (S1E1)

Enfin, les étudiants de la situation 6 jugent les dernières missions plutôt difficiles et, ce qui est cohérent, dans la situation 5 où seules les premières missions ont été proposées, quatre étudiants et un enseignant estiment les missions trop faciles.

« *Les exercices de programmation avec Kernel Panic pourraient être plus avancés, dans les TP c'est juste réaliser une action, c'est plus du type "entrer une commande sous DOS" que des vrais lignes de code.* » (S5E4)

« Les premières missions furent faciles. Puis en fonction des missions, le professeur nous demande d'approfondir le code (par exemple lors de la première attaque, on attend que l'ennemi apparaisse puis on attaque). Les missions 7 et 8 furent assez difficiles. » (S6E10)

5.3.3. Notions de programmation

Certaines remarques font référence aux concepts informatiques mis en jeu par l'utilisation de *Prog&Play*. Dans la situation 5, ce sont surtout les enseignants qui signalent des difficultés vis-à-vis de points qui ne sont pas enseignés en cours (utilisation d'une bibliothèque de fonctions, abstraction de données par exemple).

« L'introduction du *Prog&Play* demandait aux étudiants de faire appel à des concepts qu'ils ne maîtrisent pas encore (notamment en ce qui concerne l'utilisation de fonctions ou de variables prédéfinies) qui en a bloqué certains pendant de longs moments. » (S5P2)

« Il n'est peut-être pas possible de traiter toutes les notions du cours/TD à l'aide du jeu. » (S5P9)

« Pour les étudiants de niveau plus faible, l'utilisation des bibliothèques a constitué une barrière supplémentaire [... en particulier] l'utilisation des fonctions associées et l'abstraction des données (entités). » (S5P12)

D'autres signalent un apprentissage (cas des boucles, de la décomposition d'un programme en fonctions, de l'intérêt des commentaires ou des tests).

« Du point de vue apprentissage du C, je pense qu'ils ont approfondi leur connaissances sur les boucles, appris à chercher des fonctions dans une bibliothèque en se contentant des commentaires et surtout qu'ils ont été obligés, par eux-mêmes, de décomposer un problème en sous problèmes (alors qu'en TD/TP/examen ils doivent résoudre les sous problèmes ponctuels qui sont donnés et plus faciles à noter). Par contre, je ne suis pas sûre que ce travail les avantage pour l'UE de C qui n'évalue pas ces compétences. » (S6P1)

Les remarques des étudiants proviennent essentiellement de la situation 6 où les étudiants ont rédigé un rapport pour lequel la consigne était de préciser les difficultés rencontrées et ce qu'ils avaient appris durant le projet. Ces étudiants comparent l'apprentissage avec *Prog&Play* à celui de l'UE de programmation de C qu'ils suivent en parallèle.

« Pour pouvoir créer une IA on doit réfléchir par nous-mêmes aux fonctions nécessaires qu'on va devoir créer. Alors qu'en TD on ne réfléchit pas, on suit simplement les questions. » (S1E3)

« Il y a dans ce jeu une notion qu'il n'y a pas en UE de LI115, qui est la gestion du temps. En effet, il faut attendre que certains ordres soient effectués avant d'en donner d'autres (par exemple, il faut attendre que l'unité qui nous intéresse soit créée avant de lui envoyer une commande). » (S6E11)

« Le fait de devoir écrire un programme entier complet à l'aide de bibliothèques extérieures qui nous étaient jusqu'ici inconnues constitue une nouveauté (plutôt plaisante) dans notre cursus informatique. [...] Si nous avons déjà pu être amenés à nous concentrer sur des problèmes plus conséquents que ceux présentés en TD, celui de terminer la mission huit aura sans doute constitué le plus difficile de ces derniers. Ainsi, nous pensons en avoir retenu de bonnes habitudes de programmation, car à plusieurs reprises, par manque de tests, nous nous sommes retrouvés dans une position difficile où la correction de nos fonctions devenait assez complexe. » (S6E16)

« Nous avons également pu mettre en pratique ce que nous avons appris en LI115, notamment la création et l'utilisation de fonctions. Nous nous sommes également améliorés sur l'utilisation de boucles et d'alternatives, dans le choix des priorités des unes par rapport aux autres. » (S6E18)

« Ce jeu nous a amené à faire un long programme (pour la mission 8), ce que nous ne faisons pas en TP de C. Ainsi nous avons appris plusieurs choses :

- L'utilité de faire des fonctions : Nous avons découvert l'intérêt de découper le code en plusieurs fonctions, ce qui nous a aidé à détecter les erreurs, et à alléger le programme pour le rendre plus compréhensible.
- L'utilité de mettre des commentaires : Au bout de la 2^e séance sur la mission 8, nous nous sommes rendu compte que, en l'absence de commentaires, nous mettrions beaucoup de temps à déchiffrer le code et à nous rappeler où nous en sommes.
- Une technique pour détecter les erreurs : Lorsque nous ne savions pas pourquoi la fonction ne faisait pas ce qu'elle devait, nous avons découvert qu'il était utile de demander à la fonction d'afficher quelque chose. » (S6E20)

« Jusqu'à maintenant nous étions habitués aux TP rapides où les commentaires et les variables explicités n'étaient pas indispensables car nous n'avions pas à revenir sur le travail plus tard. La lisibilité du code n'était pas importante, il suffisait que le code fonctionne comme annoncé. [...] De même, on retrouve une utilisation systématique des fonctions qui s'avèrent indispensables pour décomposer le travail. » (S6E21)

« Ça permet aussi de se forger une bonne base, surtout dans l'écriture de C et ça nous apprend aussi à bien lire et utiliser un document de programmation. » (S6E23)

5.3.4. Influences sur l'apprentissage

Des remarques sont partagées concernant une influence sur l'apprentissage.

Certains enseignants de la situation 5 estiment l'influence négative, au moins dans l'état du jeu ou dans le dispositif qui avaient été mis en place.

« Ces étudiants ont trouvé que le temps de lecture est beaucoup trop important par rapport au temps nécessaire à la programmation : en résumé, "tout ça pour ça". [...] Au final, j'ai choisi à l'issue des TP 3 et 4 de Prog&Play de leur faire faire le TP4 traditionnel sur les listes avant de passer au TP 5. » (SSP3)

« Je pense que l'influence sur leur motivation était positive. [...] Par contre, l'impact sur leurs performances effectives était clairement négatif. [...] Je pense qu'il serait prématuré de trancher après une première expérience. » (SSP6)

D'autres enseignants ont un avis plus nuancé et lié au profil des étudiants.

« J'ai l'impression (à confirmer) que les étudiants en grande difficulté n'améliorent pas vraiment leur connaissance en C mais sont débordés par la complexité ajoutée par l'environnement d'exécution du jeu. Peut-être par ce que je ne les ai pas assez aidés. » (S6P1)

Enfin de nombreuses remarques mentionnent une amélioration des connaissances et un apprentissage facilité sans faire toujours référence à un contenu de programmation ou en faisant allusion à des compétences « transversales » de réflexion, d'autonomie comme nous en avons donné des exemples dans la section 5.1.

« En ce qui concerne les étudiants, très forte motivation au départ, et réel engouement pour certains. Sur la fin certains sont arrivés à leurs limites mais sur une ou deux séances uniquement. Tous les étudiants ont atteint les objectifs initiaux. » (S7P1)

« 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. » (S6E6)

« Cette UE m'a énormément apporté aussi en ce qui concerne la programmation en C et m'a aidé en LI115. Le fait aussi de travailler en binôme fut très intéressant. » (S6E12)

« Ce projet nous a permis d'approfondir nos connaissances en programmation. En effet étant donné que nous avons commencé à programmer il y a peu de temps, nous n'avions qu'une approche un peu trop théorique de la programmation. Ce projet nous a donc apporté un côté pratique non négligeable. » (S6E13)

6. Discussion

À la lumière de cette analyse, quatre facteurs nous apparaissent déterminants dans les situations étudiées : 1) la prise en compte des subtiles différences dans les connaissances travaillées dans le contexte du jeu et celles qui font partie du programme des enseignements traditionnels ; 2) le temps d'exposition au jeu ; 3) le degré d'implication des enseignants dans la mise en place de la formation ; 4) la médiation mise en œuvre par l'enseignant. Nous déduisons de ces observations des recommandations pour faciliter l'utilisation de *Prog&Play*.

6.1. Contenu enseigné

Les missions de *Prog&Play* visent à intégrer des concepts de programmation dans un contexte que les étudiants trouvent réaliste ou, du moins, qui fait sens pour eux. On peut se demander si *Prog&Play* ne renforce pas chez les étudiants le cliché « informatique égal jeux vidéo », mais il peut aussi démystifier et ouvrir l'intérêt vers de nouvelles applications (cf. remarques de la catégorie 1.2 – voir section 5.1.2).

Les concepts travaillés sont les mêmes que ceux de l'enseignement traditionnel (affectation, structures de contrôle conditionnelles et itératives, fonctions) mais ils sont appliqués dans un contexte différent : d'une part l'enseignement traditionnel propose des exercices mathématiques simples et, d'autre part, l'environnement *Prog&Play* permet de commander des entités virtuelles dans un monde qui évolue avec le temps. Dans les deux cas, des prérequis sont nécessaires : des concepts mathématiques pour les enseignements traditionnels et la manipulation de l'environnement et l'interaction avec le jeu dans le contexte de *Prog&Play*. Lorsque ces prérequis restent implicites et à la charge des étudiants, cela crée des obstacles pour de nombreux étudiants. Nous recommandons donc un enseignement explicite des manipulations de l'environnement du jeu.

Si l'on accepte l'hypothèse que l'apprentissage est contextuel, le changement de contexte n'est pas indifférent sur l'apprentissage comme le relèvent certains étudiants de la situation 6 (par exemple à propos de la gestion du temps, catégorie 3.3 – voir section 5.3.3). De plus, certains enseignants de la situation 5 remarquent que le contexte du jeu (donner des ordres à des objets du jeu) est plus proche du paradigme de la programmation impérative et objet que de celui de la programmation fonctionnelle qui est par contre très adapté au contexte mathématique.

Une autre caractéristique de l'usage de *Prog&Play* est l'utilisation d'une bibliothèque de fonctions et son exploration. Les enseignants de la situa-

tion 5 notent cette caractéristique comme étant un élément négatif puisque cette compétence n'est pas explicitement enseignée (ni évaluée), alors que les enseignants et des étudiants des contextes 6 et 7 le soulignent comme étant une compétence très intéressante travaillée dans l'environnement *Prog&Play*.

Enfin, dans les situations de projet (situations 6 et 7), les enseignants ont insisté sur l'autonomie des étudiants qui apprécient d'être dans un environnement qui leur permet d'être créatifs. Les étudiants cherchent par eux-mêmes les stratégies gagnantes et créent les programmes pour les mettre en œuvre. Ils exercent ainsi des compétences peu travaillées en première année (par exemple, d'analyse descendante) qui sont cependant très importantes en programmation comme le souligne le curriculum ACM-IEEE déjà cité. Cependant, cette autonomie n'est profitable qu'à condition de connaître un minimum de syntaxe ce qui est l'objectif des séances de TD et de TP habituels. Notons que dans la situation 6, l'utilisation de *Prog&Play* ne débute qu'en semaine 5 au second semestre, c'est-à-dire une fois que les étudiants ont bénéficié de l'enseignement traditionnel sur les notions de base qui peuvent alors être travaillées en profondeur dans le jeu.

Ces différences dans les notions travaillées posent bien évidemment le problème de l'évaluation des étudiants. Les commentaires des étudiants ont été recueillis avant l'évaluation finale des étudiants. Sur ce point, un seul juge négatif le transfert des compétences travaillées avec *Prog&Play* dans l'UE de programmation classique (*cf.* catégorie 2.3 – voir section 5.2.3, remarque de l'étudiant S6E11). Cette question du transfert transparaît aussi dans certains commentaires d'enseignants de la situation 5 qui s'inquiètent de défavoriser les étudiants en utilisant *Prog&Play* dans certains TP. Dans les situations 6 et 7, dans le cadre d'une pédagogie de projet, les enseignants ont mis au point une évaluation spécifique tenant compte des compétences travaillées dans le jeu, ce qui n'était pas le cas dans les autres situations où l'évaluation classique porte généralement sur des habiletés plus spécifiques.

L'environnement *Prog&Play* apparaît donc plus adapté pour approfondir les notions de base et pour développer des compétences que pour introduire ces notions de base. Il peut cependant être utilisé très tôt dans un cours d'introduction à la programmation comme le montre la situation 6 où il est utilisé à partir de la semaine 5 (sur 12 semaines du cours d'introduction).

6.2. Temps d'exposition au jeu

Le facteur temps est lié à un des aspects constituants d'un jeu : les joueurs apprécient de découvrir par eux-mêmes l'univers du jeu, les différentes unités et leurs propriétés. Ainsi, dans *Prog&Play*, le temps d'exposition au jeu est important pour permettre aux étudiants de comprendre comment l'univers du jeu réagit à leurs programmes ce qui est un des ressorts principaux de la motivation et de l'apprentissage dans cet environnement (cf. remarques de la dimension 1 et des catégories 2.2, 2.3, 3.3 et 3.4 – voir sections 5.1, 5.2.2, 5.2.3, 5.3.3 et 5.3.4).

Dans un TP traditionnel d'introduction à la programmation, chaque séance cible un concept unique à enseigner (par exemple, les conditionnelles ou les boucles). Lorsque *Prog&Play* a été intégré dans une séquence pédagogique existante (S4 et S5), les enseignants ont éclaté les différentes missions sur plusieurs séances afin de les mettre en correspondance avec les objectifs de chaque séance. Les étudiants en difficulté ont été aidés par leur enseignant pour terminer la mission dans le temps imparti à la séance. Ces étudiants se sont retrouvés ainsi dans une situation d'exécution au lieu d'être dans une démarche exploratoire. Cette parcellisation des missions va à l'encontre de l'immersion des étudiants dans le jeu, alors même que les missions ont été conçues dans une logique de *flow* qui est un des éléments clés de la persistance à rester dans l'action (Csikszentmihalyi, 1991) comme en témoignent les nombreuses remarques de la dimension 1.

La mise en place de TP *Prog&Play* en substitution d'un TP traditionnel, met les enseignants mal à l'aise car ils ont l'impression de ne pas traiter assez d'exercices et de perdre du temps en laissant les étudiants explorer. Les situations d'ateliers ou projets permettent une organisation plus flexible en cohérence avec la notion de *flow* où les étudiants peuvent découvrir le jeu par eux-mêmes et apprendre à partir du *feedback* que l'environnement produit sur leurs actions.

6.3. Implication des enseignants

Le critère de succès que nous avons retenu (réutilisation du jeu après une première expérimentation) montre l'importance de l'investissement de l'enseignant dans la mise en place de la situation d'apprentissage avec *Prog&Play*. Dans les situations 4, 6 et 7, les enseignants persévèrent à utiliser le jeu. Ils ont été à l'initiative de la mise en œuvre du jeu qui leur semblait pouvoir servir leurs objectifs pédagogiques. Ils ont ainsi mis en place un dispositif autour du jeu : spécification des objectifs, préparation des

sujets, consignes données aux étudiants, planification des séances, évaluation des étudiants dans les contextes 6 et 7. Ils ont aussi assumé l'installation du jeu sur les ordinateurs de leur institution. Dans les autres situations, l'équipe de conception de *Prog&Play* avait sollicité des enseignants pour tester le jeu dans leur contexte d'enseignement. Ce sont les membres de cette équipe qui ont installé *Prog&Play* sur les ordinateurs des différentes institutions. Peu d'enseignants ont participé à cette phase, les autres ont soutenu la démarche mais ne se sont pas réellement investis. En conséquence, lorsque les expérimentations se sont terminées, aucun enseignant n'était réellement capable de poursuivre l'initiative. De plus, comme le révèlent certains commentaires de la situation 5 (dimension 2 – voir section 5.2), les sujets de TP ont semblé inadaptés bien qu'ils aient été validés en amont par l'équipe pédagogique de la formation. Un des enseignants souligne que l'expérience manquait de lisibilité pour les étudiants, mais sans doute aussi pour les enseignants qui animaient les séances.

Le rôle de l'implication des enseignants dans la réussite d'une innovation pédagogique est très bien documentée (Egenfeldt-Nielsen, 2006). Pour les jeux sérieux, comme pour une autre technologie éducative, elle est cruciale pour qu'une innovation se scolarise (Bruillard & Baron, 2006). En ce qui concerne *Prog&Play*, pour dépasser le seul attrait de la nouveauté, l'enseignant doit tout d'abord bien évidemment s'assurer que le jeu sert ses objectifs pédagogiques et ensuite se familiariser assez avec l'environnement pour être à l'aise avec les étudiants. Au regard de la situation 6, nous constatons qu'être joueur n'est pas une condition *sine qua none* pour enseigner avec *Prog&Play*. L'important réside dans la clarté des objectifs d'apprentissage avec le jeu.

6.4. Médiation de l'enseignant

Comme dans toute pédagogie active, l'exposition au jeu et l'autonomie de travail doivent être accompagnées d'une médiation de l'enseignant. L'enseignant clarifie les concepts de programmation mobilisés dans le jeu, incite à faire le lien avec les concepts étudiés en cours et facilite le travail de contextualisation/décontextualisation des étudiants. Ces interventions sont modulées en fonction des profils des étudiants. Les étudiants avancés aiment trouver les solutions par eux-mêmes alors que les étudiants moins avancés ont besoin d'être davantage sollicités et guidés. L'équilibre entre laisser suffisamment de défi pour que le jeu soit motivant et le guidage pour éviter que certains étudiants ne soient noyés,

n'est pas facile à mettre en place, comme en témoignent les remarques de la catégorie 2.4 (voir section 5.2.4).

Étant donnée la diversité des stratégies mises en œuvre et des rythmes de progression de chacun, un point délicat à gérer est de ménager des temps de réflexion en commun et d'institutionnalisation de ce qui a été appris. Dans la situation 6, à la dernière séance, chaque binôme exposait devant les autres ses stratégies, les difficultés rencontrées et ce qu'il avait appris durant les séances. Cette dernière séance est l'occasion pour l'enseignant de susciter cette réflexion collective.

6.5. Recommandations

L'étude présentée ici suggère des recommandations pour faciliter l'utilisation de *Prog&Play* :

- prévoir du temps pour la prise en main par les enseignants de l'environnement du jeu, pour prendre la mesure des changements pédagogiques induits par l'approche ludique et pour expliciter les connaissances à mettre en œuvre pour réussir les missions ;
- mettre en place une évaluation qui prenne en compte les connaissances travaillées dans le jeu ;
- laisser du temps aux étudiants pour découvrir le jeu et ses règles ;
- indiquer clairement aux étudiants les objectifs pédagogiques des sessions de travail avec *Prog&Play* et les apprentissages qui seront évalués ;
- trouver un équilibre pour soutenir aussi bien les étudiants autonomes que les étudiants qui le sont moins ;
- encourager la collaboration entre étudiants ;
- prévoir une phase de réflexion collective et d'institutionnalisation sur ce qui a été appris pendant les sessions *Prog&Play* ;
- enseigner explicitement les connaissances implicites : pourquoi il est important en programmation d'utiliser des bibliothèques prédéfinies et comment les utiliser, comment utiliser un nouvel environnement de programmation et comment passer de cet environnement au jeu.

Certaines de ces recommandations sont proches des bonnes pratiques relevées par (Stizman 2011). Cependant, l'étude précise que nous avons menée met l'accent sur les difficultés potentielles d'intégration d'un dispositif d'enseignement fondé sur un jeu dans un dispositif d'enseignement plus traditionnel. Elle met aussi l'accent sur la nature des connaissances dont l'apprentissage est visé par le jeu.

7. Conclusion

Dans cet article, nous avons présenté *Prog&Play* un environnement d'apprentissage de la programmation s'appuyant sur un jeu vidéo et le projet de recherche qui lui a donné naissance. Nous avons analysé des données recueillies auprès de 56 étudiants et 17 enseignants ayant utilisé *Prog&Play* lors de projets ou de TP dans sept contextes universitaires différents. Pour trois des situations étudiées, la formation avec *Prog&Play* est reconduite depuis plusieurs années.

De nombreuses recherches visent à étudier l'efficacité des jeux sérieux. Notre expérience avec *Prog&Play* confirme que la question ne peut être tranchée indépendamment du dispositif d'enseignement que le jeu sérieux instrumente. Nous avons retenu un critère d'évaluation manichéen : *Prog&Play* est-il réutilisé après une première tentative ? Ce critère nous permet d'étudier les deux questions suivantes :

Quelles sont les caractéristiques de *Prog&Play* mises en avant par les étudiants et les enseignants pour expliquer l'adoption de *Prog&Play* sur la durée ou son abandon après une première utilisation ?

Quelles recommandations établir pour conduire à une utilisation favorable de *Prog&Play* ?

L'analyse présentée indique que *Prog&Play* est un bon support pour une pédagogie de projet qui laisse du temps aux étudiants pour s'approprier le contexte du jeu et pour expérimenter par eux-mêmes des stratégies avant d'arriver à programmer une stratégie optimale. Par contre, utiliser *Prog&Play* ponctuellement comme outil pour introduire des notions dans une pédagogie plus transmissive apparaît limiter à la fois la motivation et l'apprentissage. L'environnement *Prog&Play* apparaît plus adapté pour approfondir les notions de base et pour développer des compétences que pour introduire les notions de base.

Dans cet article, nous nous sommes intéressés au critère d'évaluation « adoption sur la durée » de l'environnement. D'autres critères d'évaluation nous semblent intéressants à étudier. Le premier est celui de l'adaptabilité de l'environnement par les enseignants. Nous pouvons citer plusieurs exemples. Nous avons vu que des enseignants ont organisé des dispositifs de soutien (situation 4), de projets (situation 6 et 7) autour de *Prog&Play*. L'enseignant de la situation 7, pour l'évaluation, a imaginé et intégré au jeu une nouvelle mission. D'autres enseignants ont créé une situation de jeu originale que nous n'avons pas présentée ici, faute de données recueillies. Ils ont organisé un concours de programmation non

obligatoire ouvert à toutes les formations en informatique de leur université (DUT, Licence, Master soit 42 participants). Enfin, l'utilisation de *Prog&Play* pour initier à l'informatique des publics scolaires ou non informaticiens est une perspective qui nous intéresse et pour laquelle nous avons développé l'interface en Scratch. Le deuxième critère consisterait à étudier l'influence sur la réussite aux examens de l'UE traditionnelle des étudiants ayant suivi l'UE *Prog&Play*. Nous avons déjà exploré ce critère d'évaluation dans le cadre de la situation 5 (Muratet *et al.*, 2011) et il mériterait d'être approfondi dans de nouveaux contextes. Le troisième porterait sur l'évaluation du détournement de l'environnement comme un outil pédagogique pour former les étudiants en informatique plus avancés à travers des projets de développement du jeu (par exemple pour de nouvelles interfaces pour d'autres langages de programmation) ou des projets de *game design* (par exemple pour concevoir de nouvelles missions et de nouvelles situations de jeu).

Ces premières étapes du projet montrent qu'il est possible d'introduire (et de maintenir) un jeu sérieux dans des enseignements universitaires fondamentaux. Nos recherches actuelles portent sur trois points : (i) diversifier les jeux pour faciliter l'organisation de tournois ; (ii) créer un outil auteur pour faciliter la création de missions par les enseignants ; (iii) proposer des jeux sérieux pour des publics à besoins éducatifs particuliers.

Remerciements

Nous remercions chaleureusement les étudiants et les enseignants qui ont participé à ces mises à l'essai et tout particulièrement Pierre Jarraud, Françoise Le Calvez, Véronique Gaildrat, André Péninou, Sylvain Barreau et Jérémie Guiochet.

1 LI115 est une UE d'initiation au langage C que les étudiants de la situation 6 suivaient en parallèle de l'UE transversale *Prog&Play*.

BIBLIOGRAPHIE

- BARDIN L. (1998). *L'analyse de contenu*, Presses Universitaires de France, 9^e édition, Paris.
- BIZZOCHI J., PARAS B. (2005). Game, motivation, and effective learning: an integrated model for educational game design. *International DiGRAConference*.
- BRUILLARD E., BARON G.-L. (2006). Usages en milieu scolaire : caractérisation, observation et évaluation. In Monique Grandbastien et Jean-Marc Labat (dir.), *Environnements informatiques pour l'apprentissage humain*, Traité IC2, Lavoisier, Paris, p. 269-284.
- CHEN W.-K., CHENG Y. C. (2007). Teaching Object-Oriented Programming Laboratory With Computer Game Programming. *IEEE Transactions on Education*, Vol. 50 n° 3, p. 197-203.
- COOB P., CONFREY J., DISESSA A., LEHRER R., SCHAUBLE L. (2002). Design experiments in educational research. *IEEE Transaction on Education*, Vol. 40 n° 1, 412-416.
- CSIKSZENTMIHALYI M. (1991). *Flow - The Psychology of optimal Experience*. Harper Perennial.
- Du BOULAY B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, Vol. 2 n° 1, 57-73.
- EGENFELDT-NIELSEN S. (2006). Overview of research on the educational use of video games. *Digital Kompetanse*, Vol. 1 n° 3, 184-213.
- GESTWICKI P., SUN F.-S. (2008). Teaching design patterns through computer gamedevelopment. *ACM Journal on Educational Resources in Computing*, Vol. 8 n° 1, 1-22.
- GIRARD C., ESCALLE J., MAGNAN A. (2012). Serious games as new educational tools: how effective are they? A meta-analysis of recent studies. *J. of Computer Assisted Learning*, Vol. 29 n° 3, 207-219.
- GREITZER F. L., KUCHAR O. A., HUSTON K. (2007). Cognitive science implications for enhancing training effectiveness in a serious gaming context. *J. Educ. Resour. Comput.*, Vol. 7 n° 3, art. 2.
- HARTNESS K. (2004). Robocode: using games to teach artificial intelligence. *J. of Comput. Sciences in Colleges*, Vol. 19 n° 4, 287-291.
- JENKINS T. (2002). On the difficulty of learning to program. *3rd Annual Conference of LTSN-JCS*, p.53-58.
- KELLEHER C., COSGROVE D., CULYBA D., FORLINES C., PRATT J., PAUSCH R. (2002). Alice2: Programming without Syntax Errors. *15th annual symposium on the User Interface Software and Technology*.
- KINNUMEN P., MALMI L. (2006). Why students drop out CS1 course? *Proceedings of the Second International Workshop on Computing Education Research*, p. 97-108.
- MALONEY J., BURD L., KAFAI Y., RUSK N., SILVERMAN B., RESNICK M. (2004). Scratch: A Sneak Preview. *2nd International Conference on Creating Connecting, and Collaborating through Computing*, p. 104-109.

MARCHIORI E. J., TORRENTE J., DEL BLANCO A., MORENO-GER P., SANCHO P., FERNANDEZ-MANJON B. (2012). A narrative metaphor to facilitate educational game authoring. *Computers & Education*, Vol. 58 n° 1, 590-599.

MARFISI-SCHOTTMAN I., GEORGE S., TARPIN-BERNARD F. (2010). Tools and methods for efficiently designing serious games. *4th European Conference on Games Based Learning ECGBL2010*, Copenhagen, Denmark, p. 226-234.

MARNE B., CARRON T., LABAT, J.-M. (2013). Modélisation des parcours pédago-ludiques pour l'adaptation des jeux sérieux. *6^e conférence sur les Environnements Informatiques pour l'Apprentissage Humain EIAH2013*, Toulouse, France, p. 55-66.

MCCRACKEN M., ALMSTRUM V., DIAZ D., GUZDIAL M., HAGAN D., KOLIKANT Y.B-D., LAXER C., THOMAS L., UTTING I., WILUSZ T. (2001). A Multi-national, Multi-institutional Study of Assessment of Programming Skills of First-year CS Students. *Working group reports from ITiCSE on Innovation and Technology in Computer Science Education*, p. 125-180.

MURATET M., TORGUET P., JESSEL J.-P., VIALLET F (2009). Towards a Serious Game to Help Students Learn Computer Programming. *International Journal of Computer Games Technology*, Hindawi Publishing Corporation, Vol. 2009.

MURATET M., TORGUET P., VIALLET F., JESSEL J.-P. (2011). Évaluation d'un jeu sérieux pour l'apprentissage de la programmation. *Revue d'Intelligence Artificielle*, Hermès Science Publications, Vol. 25 n° 2, 175-202.

MURATET M., DELOZANNE E., TORGUET P., VIALLET F. (2012). Jeu sérieux et motivation des étudiants pour apprendre : influence du contexte avec *Prog&Play*. *Colloque Technologies de l'Information et de la Communication pour l'Enseignement (TICE 2012)*, Lyon, France, p. 91-97.

PEARS A., SEIDMAN S., MALMI L., MANNILA L., ADAMS E., BENNEDSEN J., DEVLIN M., PATERSON J. (2007). A survey of literature on the teaching of introductory programming. *SIGCSE Bull.*, Vol. 39 n° 4, 204–223.

STIZMANN T. (2011). A meta-analytic examination of the instructional effectiveness of computer-based simulation games. *Personnel Psychology*, Vol. 64 n° 2, 489-528.

THOMAS P., YESSAD A., LABAT J.-M. (2011). Petri nets and ontologies: tools for the “learning player” assessment in serious games. *Advanced Learning Technologies*, 415-419.

YESSAD A., LABAT J.-M., KERMORVANT F. (2010). SeGAE: a serious game authoring environment. *10th IEEE International Conference on Advanced Learning Technologies (ICALT)*, p. 538–540.

ZYDA M. (2005). From visual simulation to virtual reality to games. *IEEE Computer*, Vol. 38 n° 9, 25-32.

Référencement des documents électroniques et des sites web

ACM et IEEE-CS (2008). *Computer Science Curriculum 2008: An Interim Revision of CS2001*. ACM Press and IEEE Computer Society Press, New York.

CORPUS (2013). Corpus des commentaires annotés (consulté le 1^{er} juin 2013) : http://www.irit.fr/ProgAndPlay/ressources/codageProgAndPlay_Sticef.pdf

COLOBOT (2001). *Colobot*, <http://www.ceebot.com/colobot/index-f.php>, (consulté le 21 mai 2013).

EECLONE (2011). *EEClone*, <http://www.cs.bsu.edu/homepages/pvg/games/eclone/>, (consulté le 21 mai 2013).

GUN-TACTYX (2008). *Gun-Tactyx*, <http://apocalyx.sourceforge.net/guntactyx/>, (consulté le 21 mai 2013).

KERNEL PANIC (2012). *Kernel Panic*, http://springrts.com/wiki/Kernel_Panic, (consulté le 21 mai 2013).

PROG&PLAY (2012). *Prog&Play*, <http://www.irit.fr/ProgAndPlay/>, (consulté le 21 mai 2013).

QUESTIONNAIRE (2008). Version du questionnaire de motivation et de satisfaction utilisé en 2008 : http://www.irit.fr/ProgAndPlay/ressources/Questionnaire_2008.pdf, (consulté le 1^{er} juin 2013).

QUESTIONNAIRE (2010). Version du questionnaire de motivation et de satisfaction utilisé en 2010 : http://www.irit.fr/ProgAndPlay/ressources/Questionnaire_2010.pdf, (consulté le 1^{er} juin 2013).

QUESTIONNAIRE (2013). La version actuelle du questionnaire de motivation et de satisfaction : <http://goo.gl/qj3fE>, (consulté le 1^{er} juin 2013)

ROBOT BATTLE (2009). *Robot Battle*, <http://www.robotbattle.com/>, (consulté le 21 mai 2013).