



**HAL**  
open science

# Analyser et représenter la progression de la difficulté d'un jeu sérieux du point de vue ludique et pédagogique

Thibault Carron, Mathieu Muratet, Bertrand Marne, Amel Yessad

## ► To cite this version:

Thibault Carron, Mathieu Muratet, Bertrand Marne, Amel Yessad. Analyser et représenter la progression de la difficulté d'un jeu sérieux du point de vue ludique et pédagogique. EIAH 2017, Jun 2017, Strasbourg, France. hal-01515753

**HAL Id: hal-01515753**

**<https://hal.science/hal-01515753>**

Submitted on 1 Oct 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Analyser et représenter la progression de la difficulté d'un jeu sérieux du point de vue ludique et pédagogique

Thibault Carron<sup>1,3</sup>, Mathieu Muratet<sup>1,2</sup>, Bertrand Marne<sup>4</sup> et Amel Yessad<sup>1</sup>

<sup>1</sup> Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris, France

<sup>2</sup> INS HEA, 58-60 Avenue des Landes, 92150 Suresnes, France

<sup>3</sup> Université Savoie Mont Blanc, 73376 Le Bourget du Lac, France

<sup>4</sup> CNRS, Université Lyon 2, ICAR UMR 5191, Lyon, France

{thibault.carron, mathieu.muratet, amel.yessad}@lip6.fr, bertrand.marne@univ-lyon2.fr

**Résumé.** Avec le développement de l'apprentissage mobile, des jeux sérieux suivent cette tendance en proposant des jeux conçus autour d'un ensemble de nombreux niveaux dédiés à la pratique des compétences spécifiques. Ils ont la particularité de présenter des niveaux courts sous forme de casse-têtes adaptés aux situations mobiles. Comme dans les jeux vidéo, il est très important d'offrir un défi progressif ou encore mieux un défi adapté pour l'apprenant. La conception de niveaux est la plupart du temps empirique : certains sont évalués trop faciles, d'autres trop difficiles, ou encore mal placés dans la progression générale et certains ajustements sont faits progressivement. Dans cet article, nous nous concentrons donc sur une méthodologie pour décrire les niveaux d'un jeu d'apprentissage tant du point de vue ludique que du point de vue pédagogique afin d'être en mesure d'évaluer la progression (difficulté, apprentissage, challenge/plaisir). Nous avons appliqué cette approche à des jeux d'apprentissage et avons été en mesure de suggérer des améliorations ou plus simplement d'en représenter une vision globale, qui est une condition *sine qua non* pour une adaptation plus fine du défi offert à chaque apprenant.

**Mots-clés.** Jeu sérieux, évaluer la progression de l'apprentissage, apprentissage mobile, adaptation de contenu d'apprentissage

**Abstract.** With the development of mobile learning, some learning games follow this trend by proposing games designed around a lot of short levels dedicated to the practice of specific skills. As in videogames, it is very important to offer a progressive challenge or even better a challenge adapted for the learner. The design of levels is most of the time empirical: some are evaluated too easy or other too difficult, misplaced and some adjustments are made progressively. In this paper, we focus thus on a methodology to describe the levels of a learning game both from a playful and a pedagogical point of view in order to be able to evaluate the progression (difficulty, learning, gaming/fun). We applied this approach to existing learning games and are able to suggest some improvements or simply draw a global view, which is a prerequisite for a more accurate challenge adaptation for a specific learner.

**Keywords.** learning game, evaluate learning progression, mobile learning, learning content adaptation

## 1 Introduction

Les jeux réalisés à des fins éducatives (*Learning Games*) peuvent être définis comme « [...] à la fois un environnement virtuel et une expérience de jeu dans lesquels le contenu qui souhaite être enseigné peut être incorporé naturellement avec une certaine pertinence contextuelle en termes de *gameplay* [...] » [1]. Il est important de proposer suffisamment de niveaux pour maintenir un défi suffisant (c'est-à-dire incluant de la variété et une progression régulière) et, éventuellement de mettre en œuvre une réelle adaptation des scénarios pédagogiques de ces jeux aux profils des joueurs/apprenants. En outre, la tendance de l'apprentissage mobile accentue ce problème avec une démarche fondée sur la définition d'un grand nombre de niveaux courts mais indépendants d'un point de vue « scénaristique » et où le processus de production de niveaux de jeu demeure très complexe et coûteux pour mettre en œuvre et maintenir le concept de *flow* (flux ou flot) tel que décrit par Csikszentmihalyi [2]. En outre, ce processus doit impliquer des concepteurs pédagogiques, des concepteurs de jeux et des experts du domaine qui ne parlent pas forcément le même langage métier. En raison de ces difficultés, la conception de la progression dans ce type de jeu est une question complexe.

Ainsi, l'objectif de notre recherche est de proposer une description des niveaux de jeux sérieux (JS) en nous appuyant sur le processus de résolution de ces niveaux. Cette description a pour but d'améliorer la vision globale de la conception du jeu d'apprentissage et de distinguer le plus possible les niveaux du jeu. Dans cet article, nous nous concentrons principalement sur des jeux à niveaux ou jeux de type « puzzle » (casse-têtes) mais nous pourrions également considérer le contenu des activités ou des quêtes provenant d'autres types de JS. Le document est structuré comme suit : dans la première partie qui correspond à la section 2, nous discutons des travaux similaires et antérieurs. À la fin de cette section, nous présentons la question de recherche qui est abordée dans nos travaux : Comment décrire et discriminer des niveaux de JS ? Dans la section 3, nous présentons la méthodologie que nous proposons pour décrire les niveaux selon plusieurs points de vue ainsi qu'un exemple pour illustrer cette approche descriptive. Enfin, la section 4 présente les résultats obtenus grâce à cette approche.

## 2 Travaux similaires

De nombreux chercheurs ont déjà traité le problème de la difficulté dans les jeux et ont déjà proposé des cadres pour l'analyse des jeux vidéo, afin d'aider les concepteurs. On peut classer ces travaux de recherche en trois groupes, associés aux trois principales approches. Tout d'abord, les approches fondées sur les *design patterns* [3, 4]. Le but de ces approches est de formaliser certaines bonnes pratiques de conception (1) pour aider les concepteurs au cours du processus de création de nouveaux jeux vidéo et (2) pour améliorer la communication entre les différents acteurs impliqués dans ce processus. Une deuxième approche consiste à exploiter des ontologies comme dans les travaux de Zagal et al. [5]. Le but de ces approches est de fournir des concepts explicites et leurs relations pour décrire l'espace de conception

de jeux vidéo. Leur but final est de fournir un vocabulaire formel et partagé permettant de décrire les jeux vidéo de manière standardisée. La troisième approche dite « directive » [1] fournit aux concepteurs une liste de règles de conception et des lignes directrices afin de répondre aux problèmes de conception spécifiques.

D'autres cadres pour l'analyse de JS sont basés sur des méthodes issues des sciences humaines et sociales. Par exemple, les jeux vidéo ont été analysés comme des systèmes de narration [6, 7], des systèmes sémiotiques [8], ou comme des systèmes dynamiques intégrant des relations temporelles entre actions et événements [9].

Dans notre contexte de travail, nous modélisons les niveaux de JS et leur difficulté en nous appuyant sur les procédures de résolution menées par les apprenants qui appliquent des stratégies comme l'essai/erreur, ou d'autres plus complexes comme la planification dans les jeux d'échecs. Dans ce type de résolution de problèmes, l'apprenant cherche des chemins dans un espace d'états représentant l'ensemble des chemins des résolutions possibles. Plus l'espace d'état est grand, plus complexe et plus difficile est la résolution du niveau de jeu. Notre hypothèse consiste à dire que la manière de résoudre un niveau constitue un bon moyen pour le décrire.

Notre modèle est composé de descripteurs qui ont pour objectif de distinguer autant que possible les niveaux de jeux d'apprentissage. Ces descripteurs ont été directement extraits de la procédure de résolution des niveaux en analysant à la fois le contenu de chaque niveau et les actions à réaliser par les joueurs pour résoudre les niveaux.

L'analyse des JS que nous proposons diffère donc des travaux sur le même thème par les points suivants :

- plus axée sur les compétences procédurales ou pratiques et donc plus adaptée aux JS,
- une approche *bottom-up* puisque la description globale du JS s'appuie sur la description des niveaux et de leurs stratégies de résolution,
- axée sur l'évaluation du degré de difficulté des activités et des niveaux.

### 3 Descripteurs pour l'analyse des niveaux de jeu d'apprentissage

Pour la description des niveaux de jeux d'apprentissage, nous proposons un modèle basé sur la résolution de ces niveaux. Notre méthodologie propose un ensemble de descripteurs pédagogiques pour coder la partie « défi à résoudre » des niveaux. Ces descripteurs sont basés sur le processus de résolution de ces niveaux et ont été élaborés à travers l'analyse de plusieurs JS de type « puzzle » : « Refraction » [10], « BlocklyMaze », « Learn2Code » et « DragonBox5 ». Dans le cadre de cet article, nous illustrons notre propos en prenant comme exemple le jeu sérieux « Refraction ».

### 3.1 Un exemple de jeu d'apprentissage organisé en défis : Refraction<sup>1</sup>

L'objectif éducatif de Refraction est de faire travailler des élèves sur l'arithmétique des fractions. Le *gameplay* (ou ressorts ludiques) de ce jeu consiste à orienter un ou plusieurs lasers pour alimenter des cibles tout en évitant des obstacles. Chaque laser et chaque cible ont leur propre niveau de puissance. Ainsi, le joueur doit organiser plusieurs éléments de « plomberie pour les lasers » afin de les rediriger, les diviser ou au contraire les combiner. L'objectif est d'atteindre la puissance attendue pour alimenter les cibles. Les valeurs de puissance utilisées ici sont des entiers ou des fractions. Ainsi, diviser les lasers revient à multiplier leur puissance par une fraction et combiner des lasers revient à additionner leurs puissances (si elles ont un dénominateur commun). Refraction aborde trois compétences relatives aux fractions : la multiplication de fractions, l'addition des fractions et le changement de dénominateur. Des défis additionnels sont disponibles pour certains niveaux : par exemple, récolter une pièce de monnaie ou une carte bonus. Le jeu Refraction propose 61 niveaux répartis dans 7 mondes.

Afin de repérer les différents descripteurs utiles pour décrire les niveaux du jeu, nous avons commencé par les analyser en identifiant la manière dont les compétences sont travaillées dans les niveaux.

### 3.2 Réification du processus de résolution de problèmes

L'étude du processus de résolution d'un niveau nous a donné un premier modèle permettant de mettre l'accent sur les compétences attendues et les séquences d'actions nécessaires pour résoudre le niveau. Sur l'exemple donné sur le tableau 1 : le niveau 7.1 nécessite les compétences *addition* et *multiplication* sur les fractions (tout comme le niveau 7.7).

Un processus de réification a ensuite été appliqué pour transformer chaque processus de résolution en un vecteur unique de compétences qui précise la manière dont la compétence est utilisée. En effet, à chaque compétence abordée dans un niveau sont associés les trois descripteurs pédagogiques suivants (P1 à P3<sup>2</sup>) décrivant la manière dont la compétence est travaillée dans le niveau :

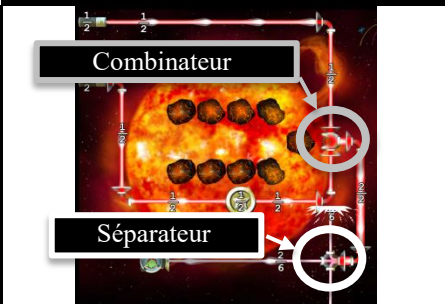
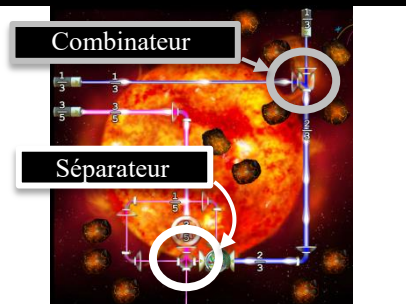
- *Apparaît* (P1) : la compétence est travaillée au moins une fois dans le niveau. Nous attribuons la valeur 1 au descripteur lorsque la compétence est travaillée et 0 sinon,
- *Parallèle* (P2) : la compétence est travaillée au moins deux fois indépendamment, c'est-à-dire, qu'elle apparaît sur des branches parallèles et séparées du graphe de résolution. Nous attribuons au descripteur le nombre de branches dans lesquelles apparaît la compétence moins 1,
- *Série* (P3) : la compétence est travaillée au moins à deux reprises et il existe un lien entre ces occurrences. Ce descripteur illustre concrètement des branches, associées à la compétence, qui sont reliées en séquence par un nœud commun dans le graphe de résolution du niveau. Nous attribuons au

<sup>1</sup> <http://play.centerforgamescience.org/refraction/site/> consulté le 23 janvier 2017

<sup>2</sup> P pour pédagogique

descripteur le nombre de branches en série dans lesquelles apparaît la compétence moins 1.

**Tableau 1.** Description pédagogique de deux niveaux similaires de Refraction, seules les compétences « Multiplier » et « Additionner » sont abordées dans ces niveaux.

Niveau 7-1		Niveau 7-7																																					
																																							
Analyse du processus de résolution																																							
<p>La compétence « Addition » de fractions est présente dans ce niveau en raison de l'utilisation d'un <i>combinateur</i> qui ajoute deux faisceaux laser. Ainsi, le descripteur P1 de cette compétence est à 1 (cette compétence apparaît au moins une fois).</p> <p>La compétence « Multiplication » de fractions est présente dans ce niveau en raison de l'utilisation d'un <i>séparateur</i> qui divise le rayon précédemment obtenu en trois faisceaux plus petits. Ainsi, son descripteur P1 est à 1 (idem).</p>		<p>La compétence « Addition » de fractions est présente dans ce niveau en raison de l'utilisation d'un <i>combinateur</i> qui ajoute deux lasers de puissance <math>1/3</math>. Par conséquent, le descripteur P1 de cette compétence est à 1.</p> <p>La compétence « Multiplication » de fraction est présente dans ce niveau en raison de l'utilisation d'un <i>séparateur</i> qui divise le faisceau laser de puissance <math>3/5</math> en trois faisceaux plus petits. Ainsi, son descripteur P1 est à 1.</p>																																					
Descripteurs pédagogiques déduits et affichés en vecteur																																							
<table border="1" style="display: inline-table;"> <thead> <tr> <th colspan="3">Multiplication</th> <th colspan="3">Addition</th> </tr> <tr> <th>P1</th> <th>P2</th> <th>P3</th> <th>P1</th> <th>P2</th> <th>P3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		Multiplication			Addition			P1	P2	P3	P1	P2	P3	1	0	0	1	0	0	<table border="1" style="display: inline-table;"> <thead> <tr> <th colspan="3">Multiplication</th> <th colspan="3">Addition</th> </tr> <tr> <th>P1</th> <th>P2</th> <th>P3</th> <th>P1</th> <th>P2</th> <th>P3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> </tbody> </table>		Multiplication			Addition			P1	P2	P3	P1	P2	P3	1	0	0	1	0	0
Multiplication			Addition																																				
P1	P2	P3	P1	P2	P3																																		
1	0	0	1	0	0																																		
Multiplication			Addition																																				
P1	P2	P3	P1	P2	P3																																		
1	0	0	1	0	0																																		

Le processus d'encodage d'un niveau avec les descripteurs pédagogiques est détaillé comme suit : (1) identifier dans le processus de résolution du niveau les compétences mises en jeu (et la manière dont elles sont utilisées) et (2) réifier cette résolution en un vecteur de compétences. Chaque compétence est donc décrite selon les trois descripteurs pédagogiques (P1 à P3). Ce processus d'encodage est réalisé pour chaque niveau du jeu. Pour les niveaux 7-1 et 7-7, les descripteurs produisent le même vecteur et nous invitent à interroger leur potentiel discriminatoire.

### 3.3 Quel est le niveau de discrimination offert par les descripteurs pédagogiques ?

Même si nos descripteurs pédagogiques nous permettent de discriminer des niveaux similaires (qui se ressemblent beaucoup dans leur construction), ils ne sont pas toujours assez fins pour distinguer tous les niveaux. C'est justement le cas, dans le tableau 1 : nous constatons que le niveau 7-1 est décrit comme le niveau 7-7. Bien que ces 2 niveaux aient des processus de résolution différents, ils sont réifiés avec le même vecteur de descripteurs pédagogiques (en l'occurrence : 100100). Cependant, ces niveaux diffèrent par les défis qu'ils offrent aux joueurs : nous avons donc complété les descripteurs pédagogiques en introduisant de nouveaux descripteurs basés sur les aspects non-pédagogiques des challenges proposés (des ressorts ludiques), là aussi extraits d'une analyse des processus de résolution de chacun des niveaux.

### 3.4 Descripteurs pour les défis non-pédagogiques

Dans les JS que nous avons étudiés, alors que les joueurs sont déjà bien sollicités dans le but d'atteindre des objectifs pédagogiques, la difficulté est souvent renforcée avec d'autres types de défis. Ces derniers sont importants, car ils augmentent la complexité de résolution des niveaux correspondants et apportent de la nouveauté, évitent la lassitude et/ou permettent de vérifier d'une autre manière la solidité de l'acquisition de connaissances (par exemple en re-testant indirectement des concepts déjà vus sous un autre angle).

Nous avons donc décidé d'inclure ces défis non-pédagogiques à nos descripteurs afin d'améliorer le caractère discriminant des motifs de description des niveaux. Nous avons identifié six classes distinctes de challenges non-pédagogiques issues de notre analyse de plusieurs JS (C1 à C6<sup>3</sup>) :

- *Induire en erreur* (C1) : augmente la difficulté en construisant le niveau de manière à induire en erreur les joueurs avec de mauvais chemins et de fausses pistes.
- *Empêcher* (C2) : bloquer le joueur dans l'accomplissement de la résolution de leur niveau par des obstacles. Ainsi, certaines branches du graphe de résolution ne sont pas/plus disponibles.
- *Élargir* (C3) : élargit les possibilités de résolution offertes au joueur. Par exemple, les joueurs pourraient avoir à combiner des objets ou des paramètres pour résoudre le niveau de jeu. Cela peut augmenter le nombre de branches dans le graphe de résolution.
- *Approfondir* (C4) : augmente le nombre d'étapes à accomplir pour résoudre le puzzle. Ainsi, on augmente la difficulté du processus de résolution.
- *Complexifier* (C5) : augmente la difficulté pour construire la solution pour les joueurs, en les exposant à des sauts cognitifs, ou des niveaux élevés d'abstraction.

---

<sup>3</sup> C pour challenge

- *Randomiser (C6)* : réduit la prévisibilité du processus de résolution pour le joueur en introduisant le facteur chance (la résolution est aléatoire).

Nous analysons chaque niveau de jeu afin de rechercher des instances de chacune de ces 6 classes. Pour illustrer, nous avons identifié les cas suivants pour le jeu Refraction :

- Induire en erreur (C1) : des dispositifs ou des ressources inutiles (par ex. des éléments de plomberie restent non utilisés dans l’inventaire ou un faisceau laser inutile pour la résolution du niveau),
- Empêcher (C2) : apparition d’obstacles spécifiques dans certains niveaux (par exemple pour les lasers, les roches sur le terrain de jeu),
- Élargir (C3) : objectifs optionnels (par exemple des pièces Bonus (C3a), des cartes Bonus (C3b), des sources multiples de laser (C3c), plusieurs cibles à alimenter (C3d) et plusieurs entrées sur une cible(C3e),
- Complexifier (C5) : par exemple la puissance initiale d’un laser est fractionnaire.

Le tableau 2 résume la comparaison détaillée des challenges non-pédagogiques inclus dans les niveaux 7-1 et 7-7<sup>4</sup>. Nous notons que trois des huit descripteurs pour les défis non-pédagogiques pour le niveau 7-1 sont différents de ceux du niveau 7-7 : C3b, C3c et C3e. En conséquence, nous avons obtenu les critères additionnels de discrimination souhaités.

**Tableau 2.** Discrimination de deux niveaux très semblables de Refraction grâce aux challenges non-pédagogiques.

Niveau 7-1								Niveau 7-7							
Défis non-pédagogiques								Défis non-pédagogiques							
C1	C2	C3a	C3b	C3c	C3d	C3e	C5	C1	C2	C3a	C3b	C3c	C3d	C3e	C5
1	1	1	1	2	1	0	1	1	1	1	0	3	1	1	1

La figure 1 ci-dessous présente le vecteur complet pour le niveau 7-1.

Level	Multiplication			Addition			Dénominateur			Défis non-pédagogiques							
	P1	P2	P3	P1	P2	P3	P1	P2	P3	C1	C2	C3a	C3b	C3c	C3d	C3e	C5
7-1	1	0	0	1	0	0	0	0	0	1	1	1	1	2	1	0	1

**Figure 1.** Description complète du niveau 7-1 de Refraction

Les mêmes vecteurs, faits avec les descripteurs pédagogiques et non pédagogiques, ont été produits pour tous les niveaux de Refraction afin d’en extraire une représentation de la progression de la difficulté de ces jeux de type « casse-têtes » (puzzle game)<sup>5</sup>.

Pour rendre ces vecteurs plus intelligibles, nous nous sommes penchés sur d’autres styles de représentation.

<sup>4</sup> Pour mémoire, nous rappelons que ces 2 niveaux avaient la même description si on se limitait aux descripteurs purement pédagogiques.

<sup>5</sup> Nous avons mis en ligne des exemples de résultats avec d’autres jeux à l’adresse suivante : <http://mmi-interne.univ-smb.fr/thibault.carron/eiah2017/>



## 4 Représentations de ces indicateurs

Quand nous avons développé les descripteurs de niveau, nous avons gardé à l'esprit les objectifs suivants :

1. Discrimination : le modèle permet de distinguer autant que possible les patrons de niveaux qui apparaissent comme similaires ;
2. Généricité et indépendance du domaine : le modèle doit pouvoir être utilisé pour n'importe quel JS quel que soit le domaine des connaissances visé ;
3. Simplicité et lisibilité : les créateurs du jeu et les concepteurs pédagogiques devraient être en mesure d'utiliser le modèle pour décrire ou comprendre n'importe quel enchaînement de niveaux.

En ce qui concerne le dernier point, nous avons essayé d'améliorer la présentation du résultat afin d'atteindre cet objectif. Les descripteurs susmentionnés sont volontairement assez simples, mais pour affiner la description, il est envisageable de sélectionner certains d'entre eux, de leur affecter un poids par un coefficient, d'en calculer une représentation partielle, sélective afin de présenter une évaluation plus ciblée (selon un objectif précis) ou une synthèse plus lisible.

Nous avons par exemple créé 3 visions globales du jeu qui font apparaître les niveaux similaires en fonction du degré de description choisi : la première en se limitant aux compétences, la seconde en précisant la manière dont elles sont travaillées puis la dernière en incluant les challenges (descripteurs non pédagogiques) comme indiqué dans le tableau 3. Nous pouvons ainsi voir dans la colonne de droite (discrimination en utilisant tous les descripteurs) que la discrimination est plus efficace que dans les autres colonnes : seuls 3 mondes présentent des niveaux similaires (1,3 et 6) et seulement 2 niveaux sont concernés à chaque fois (1-1 et 1-3 ; 3-1 et 3-3 ; 6-1 et 6-2). Au-delà de ces considérations, il peut s'agir d'une visualisation intéressante pour l'ingénieur pédagogique ou l'enseignant pour choisir s'il est important de sélectionner un niveau spécifique et ainsi mettre en place une adaptation au profil d'un apprenant (sélectionner des niveaux avec la compétence que l'apprenant doit / souhaite travailler).

Afin d'obtenir une vision globale de la progression de toutes les difficultés ou apprentissages au cours du jeu, nous sommes capables d'élaborer d'autres visualisations plus graphiques.

Par exemple, à partir de descripteurs complets de chaque niveau, nous générons un graphique où les valeurs sont (en ordonnée) les sommes des descripteurs de chaque vecteur (c'est-à-dire le nombre de descripteurs identifiés et utilisés dans ce niveau) par niveau de jeu (en abscisse)<sup>6</sup>. La figure 2 est un exemple qui montre deux histogrammes associés avec trois versions différentes de Refraction : Refraction 1, Refraction 2 et « Infinite Refraction ». Refraction 2 contient uniquement deux des trois compétences travaillées dans Refraction 1 et n'utilise pas les descripteurs de C3a et C3b. « Infinite Refraction » est une variante où la progression est ajustée automatiquement [11] et où seule la compétence « Multiplier » est travaillée.

---

<sup>6</sup> Les chiffres en abscisse de 1 à 7 correspondent aux « mondes » (système de regroupement de niveaux propre à Refraction).

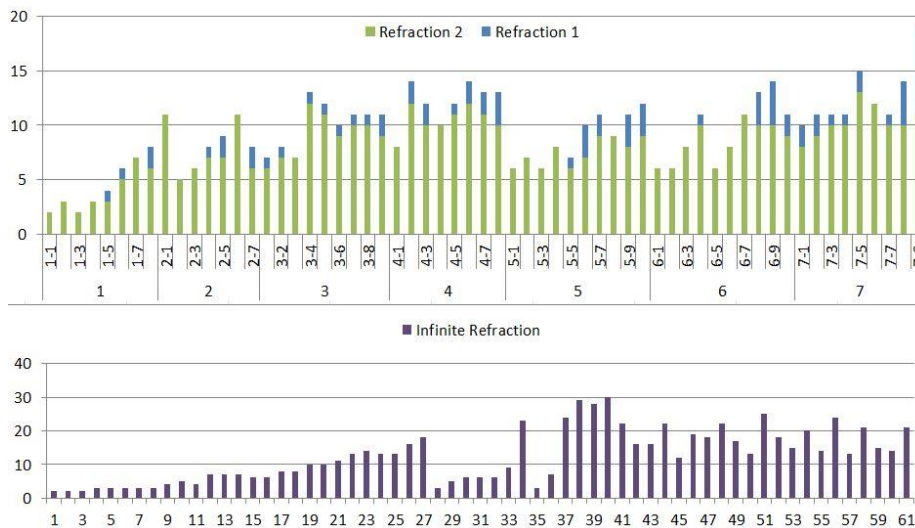
**Tableau 3.** Discrimination selon les types de descripteurs. Chaque jeu présente des niveaux similaires en fonction des critères indexés.

Mondes	Indexé par C*	Indexé par C* et DP**	Indexé par C*, DP** et DNP***
Monde 1	(1-1, 1-2, 1-3, 1-4, 1-5) et (1-6, 1-7, 1-8)	(1-1, 1-2, 1-3, 1-4, 1-5) et (1-6, 1-7, 1-8)	(1-1, 1-3)
Monde 2	(2-1, 2-2, 2-3, 2-4, 2-5, 2-6, 2-7)	(2-3, 2-5) et (2-4, 2-7) et (2-1, 2-6)	
Monde 3	(3-1, 3-2, 3-3, 3-4, 3-5, 3-6, 3-7, 3-8, 3-9)	(3-1, 3-2, 3-3) et (3-4,3-8) et (3-5, 3-6, 3-9)	(3-1, 3-3)
Monde 4	(4-2, 4-3, 4-4, 4-5, 4-6, 4-7, 4-8)	(4-3, 4-8) et (4-4, 4-5,4-7)	
Monde 5	(5-1, 5-2, 5-4) et (5-5, 5-6, 5-9) et (5-7, 5-8,5-10)	(5-1, 5-2, 5-4)	
Monde 6	(6-1, 6 - 2) et (6-3, 6-4, 6-5, 6-6, 6-7, 6-10)	(6-1, 6 - 2) et (6-5, 6-6,6-10)	(6-1, 6 - 2)
Monde 7	(7-1, 7-3, 7-4, 7-7) et (7-2, 7-5, 7-8, 7-9)	(7-1, 7-7)	

\* C : Compétence (P1)

\*\* DP : descripteur pédagogique (P2-P3)

\*\*\* DNP : Défis non-pédagogiques (C1-C6)



**Figure 2.** Représentation graphique de l'évolution de la complexité à travers les niveaux de Refraction 1, Refraction 2 et « Infinite Refraction ».

En outre, les graphiques montrent que Refraction 2 a été simplifiée par rapport à Refraction 1. Ce résultat était attendu, car Refraction 1 contient plus de défis non-

pédagogiques que Refraction 2 (des étoiles, des cartes bonus, des éléments de « plomberie » pour changer les dénominateurs des puissances de laser, *etc.*).

Concernant « Infinite Refraction », Butler *et al.* [11] présentent des travaux qui visent à automatiser la conception de la progression d'un jeu en analysant les caractéristiques des solutions. Nos résultats montrent que leurs travaux appliqués au jeu « Infinite Refraction » permettent de définir automatiquement une progression graduelle et régulière sur 27 niveaux ne traitant tous que d'une seule compétence. Au-delà, l'évolution de la difficulté devient plus chaotique selon le point de vue que nous proposons.

Finalement, les représentations graphiques, représentées sur la figure 2 mettent bien en évidence l'évolution de la complexité entre les niveaux comme nous le souhaitions. La section suivante présente ces travaux de manière plus générale ainsi qu'une discussion sur les résultats que nous avons obtenus.

## 5 Résultats et discussion

Si nous prenons un peu de recul par rapport à notre contribution, nous pouvons identifier un modèle pour décrire les niveaux d'un jeu au moyen de descripteurs (cf. Figure 3).

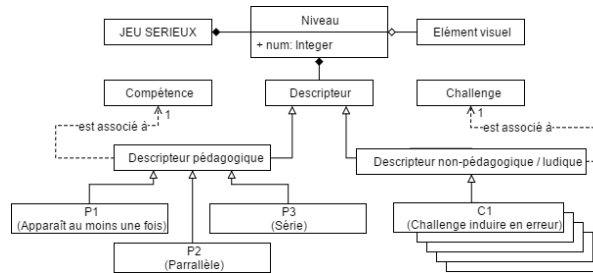


Figure 3. Le modèle UML simplifié d'un niveau vu selon l'angle de nos descripteurs

Un niveau de JS est composé de descripteurs qui peuvent être pédagogiques (et associés à des compétences) et/ou non-pédagogiques/ludiques (et dans ce cas, associés à des challenges ou défis). En outre, un niveau peut être éventuellement constitué d'éléments visuels (comme une image de fond, des décors qui n'ont pas d'impact sur la difficulté).

Ensuite, nous pouvons identifier une méthodologie (cf. Figure 4) qui nous a permis à partir de ce modèle appliqué à chacun des niveaux de répondre à 2 objectifs : (a) réingénierie : améliorer un JS en analysant la progression de la difficulté dans l'enchaînement des niveaux (ex. suggérer une refonte de certains niveaux en vue d'en augmenter (ou réduire) la difficulté et optimiser ainsi la progression du jeu) et (b) adaptation : utiliser cette vision globale du jeu pour identifier des parcours (sélectionner des niveaux qui permettent de travailler telle et/ou telle compétence), extraire/créer des exercices de remédiation, choisir le prochain exercice adapté à tel apprenant, *etc.*

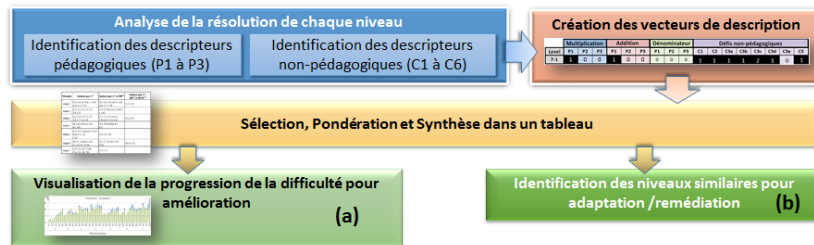


Figure 4. La méthodologie d'analyse de la progression de la difficulté d'un JS.

De plus, nous avons pu observer sur Refraction 1 et 2 qu'après les niveaux difficiles, le jeu offre au joueur des périodes de relâchement et ce, systématiquement, tout au long du jeu. Cette réduction de la complexité peut être assimilée à une période de détente qui récompense le joueur après son effort pour avoir achevé des niveaux compliqués (semblables à un classique « boss » de fin de niveau dans les jeux vidéo) et le prépare à de nouveaux défis. Utiliser des temps de relaxation après un « boss » est en effet un modèle de conception (*design pattern*) commun dans les jeux vidéo correspondant à : « une progression de la difficulté en dents de scie à travers plusieurs niveaux de jeu » [12]. Similairement, nous pouvons imaginer créer des niveaux identifiés comme plaisants (en alternant et combinant P1, P2, P3 et C1..C6) et ainsi reconnaître cela comme des schémas de conception qui fonctionnent bien et que le concepteur peut réutiliser.

Ce cadre d'analyse a été développé et expérimenté grâce à d'autres jeux sérieux : nous pouvons citer « BlocklyMaze », « Learn2Code » et « DragonBox5 » qui nous ont donné des résultats similaires aux 183 niveaux des trois versions de Refraction sur lesquels nous avons mis le focus tout au long de cet article. L'un de nos jeux Prog&Play a d'ailleurs été retravaillé selon ce principe : nous avons notamment ajouté des niveaux et travaillé sur les défis pédagogiques et ludiques pour diversifier l'expérience de jeu (la description des niveaux selon les descripteurs ainsi que la représentation graphique de l'évolution de la complexité sont disponibles en ligne<sup>5</sup>).

## 6 Conclusion

Notre méthodologie d'analyse de jeux sérieux de type « casse-têtes » (suite de niveaux courts ciblant une pratique de compétences spécifiques) bien adaptés à l'apprentissage mobile est capable de décrire et distinguer les compétences/défis pédagogiques et non-pédagogiques mis en œuvre dans les niveaux et la manière dont ils interviennent dans la résolution d'un niveau.

Il ressort de notre étude (sur 7 jeux différents dont 3 versions de Refraction) que les descripteurs (pédagogiques et non pédagogiques) proposés se révèlent assez génériques pour décrire des niveaux dans de nombreux JS. Dans les travaux futurs, nous souhaitons justement vérifier le cas des jeux multi-joueurs et d'apprentissage collaboratif pour éventuellement étendre ce modèle avec de nouveaux descripteurs (travail collaboratif) et étendre cela aux approches à base de quêtes (*Role Playing Game - RPG*).

Dans les perspectives, nous pouvons également imaginer un outil auteur qui aide à saisir les vecteurs, génère automatiquement les visualisations voire génère également les parcours en cochant les niveaux que l'enseignant souhaite intégrer dans la prochaine session du jeu. La formalisation en vecteurs rend l'approche opérationnalisable (génération procédurale de niveaux et de parcours).

Notre contribution apporte principalement deux nouveautés : le modèle aide à décrire et discriminer des niveaux en identifiant les types de compétences et challenges dont ils sont constitués. Deuxièmement, en appliquant méthodiquement ce modèle à l'ensemble des niveaux, il donne des indications précieuses sur la progression de la complexité à travers les niveaux d'un JS ; en proposant éventuellement des possibilités de représentations graphiques (pédagogique, ludique ou les deux) de la totalité des niveaux du jeu et ainsi une vision globale pour de la réingénierie ou de l'adaptation de scénarios de JS.

## Références

1. Carlo Fabricatore, Miguel Nussbaum, and Ricardo Rosas. Playability in action videogames: A qualitative design model. *Hum-Comput. Interact.*, 17(4):311-368, December 2002.
2. Mihaly Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Harper Perennial, New York, NY, March 1991.
3. Staffan Bjork and Jussi Holopainen. *Patterns in Game Design (Game Development Series)*. Charles River Media, December 2004.
4. Bertrand Marne, John Wisdom, Benjamin Huynh-Kim-Bang, and Jean-Marc Labat. The six facets of serious game design: A methodology enhanced by our design pattern library. In *21st Century Learning for 21st Century Skills*, volume 7563 of *LNCS*, pages 208-221, Saarbrücken, Germany, September 2012. Springer Berlin / Heidelberg.
5. Jos P. Zagal, Michael Mateas, Clara Fernandez-vara, Brian Hochhalter, and Nolan Lichti. Towards an ontological language for game analysis. In *Proceedings of International DiGRA Conference*, pages 3-14, 2005.
6. Jonas Carlquist. Playing the story: Computer games as a narrative genre. *Human IT.*, 6(3):7-53, 2013.
7. Janet Horowitz Murray. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. The Free Press, New York, NY, USA, 1997.
8. Julian Kücklich. Perspectives of computer game philology. *Game Studies*, 3(1), 2003.
9. Markku Eskelinen. Towards computer game studies. *Digital Creativity*, 12(3):175-183, 2001.
10. Adam M Smith, Erik Andersen, Michael Mateas, and Zoran Popovic. A case study of expressively constrainable level design automation tools for a puzzle game. In *Proceedings of the International Conference on the Foundations of Digital Games*, pages 156-163. ACM, 2012.
11. Eric Butler, Erik Andersen, Adam M Smith, Sumit Gulwani, and Zoran Popovic. Automatic game progression design through analysis of solution features. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2407-2416. ACM, 2015.
12. Ernest Adams. *Fundamentals of game design*, volume 1. New Riders, 2010.