

N° d'ordre 37 /2020

THESE DE DOCTORAT

Présentée par

Alaeeddine YASSINE

Spécialité : Informatique

Sujet de la thèse : ÉLABORATION D'UNE NOUVELLE APPROCHE DE E-LEARNING ORIENTÉE MOBILE LEARNING EN ADAPTANT LES PRINCIPES DES JEUX PÉDAGOGIQUES

Thèse présentée et soutenue le 24/09/2020 devant le jury composé de :

Nom Prénom	Titre	Etablissement	
Pr. MRABTI Mostafa	PES	Ecole Nationale des Sciences Appliquées, Fès	Président
Pr. ESSAOUDI Ismail	PES	Faculté des Sciences, Meknès	Rapporteur
Pr. CHAOUI Habiba	PES	Ecole Nationale des Sciences Appliquées, Kénitra	Rapporteur
Pr. CHIKHAOUI Ahmida	PES	Ecole Normale Supérieure, Fès	Rapporteur
Pr. KHAISSIDI Ghizlane	PH	Ecole Nationale des Sciences Appliquées, Fès	Examineur
Pr. TAHIRI Ahmed	PH	Ecole Normale Supérieure, Fès	Examineur
Pr. CHENOUNI Driss	PES	Ecole Normale Supérieure, Fès	Directeur de thèse
Pr. BERRADA Mohammed	PH	Ecole Nationale des Sciences Appliquées, Fès	Directeur de thèse

Laboratoire d'accueil : Laboratoire d'informatique et de physique interdisciplinaire (LIPI)

Etablissement : Faculté des Sciences et Techniques de Fès

Centre d'Etudes Doctorales : Sciences et Techniques de l'Ingénieur

Resumé

Les technologies éducatives sont des technologies en perpétuelle évolution permettant ainsi de repenser constamment les modalités d'apprentissage en ligne. L'un des défis actuels dans la littérature est de fournir aux apprenants un environnement approprié afin de stimuler leur motivation et leur intérêt à l'apprentissage en utilisant des technologies de pointes.

L'une des tendances émergentes dans le domaine de l'apprentissage en ligne est les jeux sérieux: des jeux conçus pour attirer l'attention des apprenants, les aider à prendre activement part à leur univers d'apprentissage et acquérir de nouvelles connaissances tout en s'amusant. Les jeux se basent sur le divertissement et le challenge. Les apprenants prennent du plaisir à jouer pendant plusieurs heures ou plusieurs jours, perdant ainsi la notion du temps en poursuivant leurs objectifs.

Apprendre à programmer est une discipline qui attire l'attention de nombreux apprenants. Tout comme le jeu, la programmation exige de nombreuses heures de pratique afin de se familiariser avec les instructions d'un langage de programmation. Elle requiert également de nombreuses compétences telles que l'analyse critique et la résolution de problèmes. Bien que la programmation se base sur des challenges et défis, de nombreux problèmes sont soulevés, comme le haut niveau d'abstraction et des instructions considérées souvent comme complexes.

La première contribution de cette thèse, est de présenter et discuter des difficultés rencontrées par les enseignants et les apprenants pour enseigner et apprendre les langages de programmation. L'accent est mis sur l'approche pédagogique ainsi que sur l'utilisation de l'éducation en ligne pour faciliter le processus d'apprentissage. La deuxième contribution porte sur la conception et le développement d'une nouvelle application mobile multiplateforme pour apprendre les bases de la programmation. L'application se base sur un système multi-agents et une ontologie de domaine. Un prototype est présenté pour expliquer les principales fonctionnalités de l'application.

Mots-clés

E-Learning, jeux sérieux, langage de programmation, Application Mobile multi-plateformes, système multi-agents, ontologie de domaine, instructions personnalisées.

Remerciements

Je tiens à remercier et à exprimer ma profonde gratitude à mes directeurs de thèse **Pr. CHENOUNI Driss** et **Pr. BERRADA Mohammed**. Je vous remercie tout d'abord de m'avoir accordé l'opportunité de préparer une thèse dans un domaine qui me passionne et d'intégrer une équipe de chercheurs dynamiques. Vous m'avez parfaitement initié à l'esprit de recherche et à la rédaction scientifique. Je vous remercie pour vos conseils précieux et votre encadrement tout au long de la période de la préparation de cette thèse, vos interventions et les discussions fructueuses ont été d'une grande aide. Votre encadrement représente le pilier principal de l'aboutissement de ce travail.

Je tiens à exprimer ma profonde reconnaissance à **Pr. TAHIRI Ahmed** pour ses encouragements et conseils avisés durant toute la période de la préparation de la thèse.

J'adresse mes sincères remerciements à tous les membres du Jury qui m'ont fait l'honneur de bien vouloir accepter d'examiner mon mémoire et évaluer mes travaux de recherche.

Je remercie **Pr. MRABTI Mostafa** d'avoir accepté de présider le jury. **Pr. ESSAOUDI Ismail**, **Pr. CHAOUI Habiba** et **Pr. CHIKHAOUI Ahmida** d'avoir accepté d'être rapporteurs; **Pr. KHAISSIDI Ghizlane** et **Pr. TAHIRI Ahmed** d'avoir accepté d'examiner cette thèse.

J'ai été ravi de mener mes activités de recherche dans une ambiance harmonieuse dans le Laboratoire d'informatique et de physique interdisciplinaire (LIPI). Je remercie mes collègues avec qui j'ai pu discuter, échanger et collaborer.

Abstract

Educational technologies changed drastically the teaching learning process. For instance, one of the current challenges in the literature is to provide learners with suitable environment in order to foster their motivation and interest using technologies they are the most familiar with.

One of the emerging trends in the e-learning area is serious games: games designed to attract student attention and help them acquire new knowledge while having fun. Serious Games are based on an entertaining side for the learners to enjoy challenges. Learners usually play for many hours or days -losing track of time- chasing objectives. This has the potential to keep them engaged and academically on the track.

Learning to program is a discipline attracting many learners. Like playing game, programming required many hours of practice in order to get familiar with a language's instructions and many skills such as critical thinking, problem solving, and analysis. Although programming is a challenging task, many issues are reported such as the high-level abstraction, complex instructions. It's also difficult for some learners to get the logic behind the code since it requires imagination and prerequisites in the field of mathematics or logic.

The first contribution of this thesis is to discuss difficulties facing both teachers and learner to teach and a learning programming language. The focus is on pedagogical approach as well as the use of educational technology to leverage the learning process. The second contribution is about the design and development of a new cross-platform mobile application to learn programming basics. The core application is based on a multi-agent system and a domain ontology to make the application interactive, flexible and dynamic and provide students with tailored instructions. A prototype is presented showing the main features of the application.

Keywords

E-Learning, Serious games, Programming Languages Cross-Platform, Multi-agent System, Domain Ontology, Tailored Instructions.

Résumé

Les technologies éducatives sont des technologies en perpétuelle évolution permettant ainsi de repenser constamment les modalités d'apprentissage en ligne. L'un des défis actuels dans la littérature est de fournir aux apprenants un environnement approprié afin de stimuler leur motivation et leur intérêt à l'apprentissage en utilisant des technologies de pointes.

L'une des tendances émergentes dans le domaine de l'apprentissage en ligne est les jeux sérieux: des jeux conçus pour attirer l'attention des apprenants, les aider à prendre activement part à leur univers d'apprentissage et acquérir de nouvelles connaissances tout en s'amusant. Les jeux se basent sur le divertissement et le challenge. Les apprenants prennent du plaisir à jouer pendant plusieurs heures ou plusieurs jours, perdant ainsi la notion du temps en poursuivant leurs objectifs.

Apprendre à programmer est une discipline qui attire l'attention de nombreux apprenants. Tout comme le jeu, la programmation exige de nombreuses heures de pratique afin de se familiariser avec les instructions d'un langage de programmation. Elle requiert également de nombreuses compétences telles que l'analyse critique et la résolution de problèmes. Bien que la programmation se base sur des challenges et défis, de nombreux problèmes sont soulevés, comme le haut niveau d'abstraction et des instructions considérées souvent comme complexes.

La première contribution de cette thèse, est de présenter et discuter des difficultés rencontrées par les enseignants et les apprenants pour enseigner et apprendre les langages de programmation. L'accent est mis sur l'approche pédagogique ainsi que sur l'utilisation de l'éducation en ligne pour faciliter le processus d'apprentissage. La deuxième contribution porte sur la conception et le développement d'une nouvelle application mobile multiplateforme pour apprendre les bases de la programmation. L'application se base sur un système multi-agents et une ontologie de domaine .un prototype est présenté pour expliquer les principales fonctionnalités de l'application.

Mots-clés

E-Learning, jeux sérieux, langage de programmation, Application Mobile multi-plateformes, système multi-agents, ontologie de domaine, instructions personnalisées.

Table des matières

Remerciements	i
Abstract	ii
Keywords	ii
Résumé	III
Mots-clés	III
Table des matières	IV
Liste des figures	xiv
Liste des tableaux	xv
Liste des publications et des communications	xvi
Introduction générale	1
1.1. Motivations.....	2
1.2. Problématique de recherche	3
1.3. Objectifs de recherche.....	4
1.4. Organisation du mémoire.....	5
Chapitre 1: Etat de l'art	7
1.1. E-learning.....	7
1.2. Types de formation en ligne	8
1.3. Historique des jeux vidéo	9
1.3.1. Des débuts expérimentaux	9
1.3.2. Facteurs sociaux pour le changement et l'évolution.....	10
1.3.3. L'ère des bornes arcade	10
1.3.4. L'avènement des consoles de salon :	11
1.3.5. La 3D dans le jeu vidéo.....	12
1.3.6. L'introduction des consoles portables	13
1.3.7. Les consoles NEXT-GEN	13
1.4. Les jeux vidéo comme outil potentiel pour l'apprentissage	15
1.5. Les jeux sérieux comme outil d'enseignement et d'apprentissage de la programmation	17
1.5.1. Concrétisation des concepts d'apprentissage abstraits	18
1.5.2. Motivation des apprenants	19
1.5.3. Des outils avec lesquels les apprenants sont familiers	19
1.5.4. Environnement d'apprentissage réaliste avec des objectifs significatifs	19
1.6. Analyse de jeux sérieux existants	21
1.7. Synthèse	31
Chapitre 2: Techniques de jeu et pédagogies pour l'apprentissage de la programmation	32
2.1. Techniques de jeu	32
2.1.1. Drag&Drop (D&D)	32

2.1.2 Point&Click	34
2.1.3 Le code writing	35
2.1.4 Le Storytelling	36
2.2. Vers une Approche pédagogique adaptée aux jeux sérieux	38
2.2.2. Approches pédagogiques.....	42
2.3 Les taxonomies.....	44
2.3.2. La taxonomie de bloom	46
2.3.3. La taxonomie de BLOOM révisée	48
2.4 Les principes d'apprentissage dans les jeux sérieux.....	49
2.5 Synthèse.....	51
Chapitre 3: Approches de développement mobiles	54
3.1. Approche de développement native	54
3.2. Approche de développement hybride	55
3.3. Approche de développement cross-platform.....	56
3.4. Frameworks de développement mobile	56
3.4.1. Titanium	56
3.4.2. Xamarin	57
3.4.3. Unity.....	57
3.4.4. React Native	58
3.4.5. Phonegap/ Cordova.....	59
3.4.6. Ionic.....	59
3.4.7. Sencha touch.....	59
3.4.8. Rhodes	60
3.5. Comparaison des frameworks de développement mobile	60
3.6. Synthèse.....	64
Chapitre 4 : conception du jeu <i>Globrain's Quest</i>	66
4.1. Arborescence des scènes du jeu	66
4.1.1. Scène d'introduction	66
4.1.2. Scène pivot	66
4.1.3. Scène Cours	66
4.1.4. Scène Épreuve	67
4.2. La conception graphique	67
4.2.1. Quel style de dessin adopter selon le public cible?	67
4.2.2. A quoi ressembleront les personnages du jeu?	68
4.2.3. Quel logiciel utiliser pour créer les visuels ?	68
4.3. Développement du prototype	69
4.3.1. Description de l'interface utilisateur	70
4.3.2. Exemple de traitement simple pris en charge par Unity	71
4.3.3. Exemple de traitement codé en C#: le questionnaire de cours	72

4.4.	Synthèse	74
Chapitre 5: Prototype du jeu <i>Globrain's Quest</i>		75
5.1.	Description générale du jeu.....	75
5.2.	Description des scènes	75
5.2.1.	Scène #1 : texte introductif	75
5.2.2.	Scène #2 : l'écran titre.....	76
5.2.3.	Scène #3: Première partie du scénario	77
5.2.4.	Scène #3: Menu des chapitres	80
5.3.	Description du premier chapitre.....	83
5.3.1.	Menu principal du premier chapitre	83
5.3.2.	GLOBRAIN'S QUEST ET LA TAXONOMIE DE SOLO	85
5.3.3.	<i>Globrain's quest</i> et la taxonomie de Bloom, exemple du premier cours sur les opérations en Python87	
5.4.	Synthèse.....	96
Chapitre 6: Vers un jeu multi-joueurs grâce au SMA		97
6.1.	L'architecture multi-agent dans notre contexte.....	97
6.2.	Base de connaissances.....	101
6.3.	Synthèse	104
Conclusion générale.....		106
Bibliographie		110

Liste des figures

Fig 1. capture du jeu Spacewar ! (1962)	10
Fig 2.capture du jeu Pac-man (1980)	11
Fig 3. capture du jeu Space Invaders (1980)	11
Fig 4. capture du jeu Pong (1972)	11
Fig 5. la console Megadrive.....	12
Fig 6. la console Gameboy de Nintendo	12
Fig 7. capture du jeu Tomb raider (1997)	12
Fig 8.la console Playstation 4 de Sony	14
Fig 9. World Of Warcraft.....	16
Fig 10. Plateforme de vente en ligne du jeu World Of Warcraft.....	17
Fig 11. François 1er à la cathédrale de Reims	23
Fig 12.La Salamandre du roi François 1er	23
Fig 13. Le choix d'un(e) villageois(e)	24
Fig 14. campement dans la région du Darfour	24
Fig 15. Trajet de ravitaillement en eau	24
Fig 16. Interface of Light-bot 2.0.....	26
Fig 17.Interface of Hour of code	27
Fig 18.Code in Javascript corresponding to a level of Hour of code	27
Fig 19.Interface of Code Combat	27
Fig 20.Interface of World of variables.....	28
Fig 21. Interface de Robot ON!	29
Fig 22. Palette des blocs de code à glisser	30
Fig 23. Ecran d'exécution du code dans scratch	30
Fig 24. Zone de script affichant des blocs de code à exécuter	30
Fig 25. Level 1 of the game 'World of variables	33
Fig 26. Level 1 of the game 'Tynker, lost in space	33
Fig 27. Phase de recherche d'objets	35
Fig 28. Phase de résolution d'énigmes.....	35
Fig 29. Niveau 4 de Code Combat	36
Fig 30. Storytelling Alice.....	37
Fig 31. Schéma d'arborescence des scènes du jeu	67
Fig 32. Création de personnage avec Adobe Photoshop	69
Fig 33. Le rendu visuel de la scène pivot sous unity3D	71
Fig 34. Exemple de traitement simple pris en charge par Unity	71
Fig 35. Extrait des scripts c# pour développer le quizz sur Unity3D.....	72
Fig 36. Extrait du code du Script Quizz.....	73

Fig 37. Ecran d'introduction, Globrain's Quest	76
Fig 38. Ecran titre, Globrain's Quest	77
Fig 39. Scénario partie 1, Globrain's Quest	78
Fig 40. Antagonistes figurant dans Globrain's Quest	79
Fig 41. Pr. Globrain (à gauche) et le Capitaine Globrave (à droite)	80
Fig 42. Menu des chapitres, Globrain's Quest	81
Fig 43. Fenêtre popup du premier chapitre, Globrain's Quest	83
Fig 44. Chapitre 1, Globrain's Quest	84
Fig 45. Niveau 1, les opérations dans Python, Globrain's Quest.....	88
Fig 46. Compilateur Python, Globrain's Quest	90
Fig 47. Dummy test 1, Globrain's Quest	91
Fig 48. Dumy test 1, Globrain's Quest.....	91
Fig 49. Dummy test 1 – popup victoire, Globrain's Quest	92
Fig 50. – Dummy test 1 – popup défaite, The Stronglobes	93
Fig 51. – niveau 2 chapitre 1, Globrain's quest	94
Fig 52. – niveau 2 chapitre 1, Globrain's quest	95
Fig 53. Architecture Multiagent	100
Fig 54. Extrait du modèle pédagogique	104

Liste des tableaux

Tableau 1: caractéristiques des consoles sony PS4 et PS5.....	14
Tableau 2. Liste des jeux serieux étudiés (Yassine et al., 2020).	37
Tableau 3: Les niveaux de la taxonomie de SOLO.....	46
Tableau 4: La taxonomie de bloom révisée	48
Tableau 5. Caractéristiques de l'approche de développement native	55
Tableau 6. Comparaison des approches de développement mobile.....	56
Tableau 7. Langage(s) de programmation par framework de développement mobile	61
Tableau 8. Comparaison des framework de développement mobile	63
Tableau 9. Structuration du cours selon la taxonomie de Solo.....	86

Liste des publications et des communications

Publications:

- Yassine Alaeeddine, Mohammed Berrada, Ahmed Tahiri, et Driss Chenouni. “ *A Serious Game for Teaching Python Programming Language*”, In *Advances in Intelligent Systems and Computing* (Vol. 1076, p. 389- 397), Springer, 2020. https://doi.org/10.1007/978-981-15-0947-6_37
- Yassine Alaeeddine, Mohammed Berrada, Ahmed Tahiri, et Driss Chenouni.. “ A cross-platform mobile application for learning programming basics ”. *International Journal of Interactive Mobile Technologies* 12 (7), 139- 51, 2018 <https://doi.org/10.3991/ijim.v12i7.9442>
- Yassine Alaeeddine, Driss Chenouni, Mohammed Berrada, et Ahmed Tahiri. “ *A serious game for learning C programming language concepts using solo taxonomy*”. *International Journal of Emerging Technologies in Learning* 12 (3), 0- 27, 2017. <https://doi.org/10.3991/ijet.v12i03.6476>.

Communications :

- Yassine Alaeeddine, Mohammed Berrada, Ahmed Tahiri, et Driss Chenouni, “A serious game for teaching Python programming language”, The 1st international conference of embedded systems and artificial intelligence(ESAI 2019), May 02-03, 2019, FSDM, USMBA, FEZ, MOROCCO.
- Yassine Alaeeddine, Mohammed Berrada, Ahmed Tahiri, et Driss Chenouni, “Description de la transmission et de l'évaluation du savoir d'une application mobile ludique ”, La 3ème édition de la Journée Doctorale et PostDoctorale des Systèmes d'Information et Télécommunications JDPD SIT'2018 27 Décembre 2018, ENSA de Fès.
- Yassine Alaeeddine, Mohammed Berrada, Ahmed Tahiri, et Driss Chenouni, « *Vers une expérience de jeu personnalisée et collaborative* »,La 2ème édition de la Journée Doctorale et PostDoctorale des Systèmes d'Information et Télécommunications (JDPD SIT'2017), 22 Décembre 2017, ENSA de Fès.
- Yassine Alaeeddine, Mohammed Berrada, Ahmed Tahiri, et Driss Chenouni, “*Role play game for teaching pointers using educational technology*”, 10th international conference of

education, research and innovation (ICERI 2018) 16, 17 et 18 Novembre 2017- Séville, Espagne <https://library.iated.org/view/YASSINE2017ROL>

- Yassine Alaeeddine, Mohammed Berrada, Ahmed Tahiri, et Driss Chenouni, “*A serious game for learning C programming language concepts using solo taxonomy*”, La 1ère édition de la Journée Doctorale et PostDoctorale des Systèmes d’Information et Télécommunications (JDPD-SIT 2016), 30 Décembre 2016, ENSA de Fès.
- Yassine Alaeeddine, Mohammed Berrada, Ahmed Tahiri, et Driss Chenouni, “*Vers une nouvelle approche pédagogique d’enseignement de la programmation en se basant sur les techniques des jeux sérieux*”, La 1ère édition du workshop international sur les approches pédagogiques et e-learning (APEL 2015), EST de Fès, les 25 et 26 Novembre 2015.

Introduction générale

L'évolution perpétuelle et la démocratisation des moyens technologiques de notre époque créent des opportunités d'exploitation encore plus vastes et dans des disciplines plus étendues. Plus précisément, l'emploi de la technologie dans le domaine de l'éducation fait l'objet de nombreux travaux de recherche. Ces derniers mettent l'accent sur les pistes d'évolution de l'enseignement de différentes matières, en munissant le professeur d'une palette d'outils multimédia capable d'améliorer sa communication et sa pédagogie.

Notre travail de recherche s'articule autour d'une problématique liée à l'enseignement classique de la programmation. Cette problématique est formulée en considérant trois axes auxquels nous tentons d'apporter de nouvelles réponses. Premièrement, l'environnement d'une classe qui peut créer chez certains apprenants un sentiment d'isolement car exprimer une incompréhension ou formuler une question au tuteur n'est pas toujours évident. Deuxièmement, le manque de perception de l'utilité de l'apprentissage qui empêche les apprenants d'être motivés du fait du manque de mise en relation avec le monde du travail ou avec des situations réalistes où ils peuvent mettre en œuvre leurs connaissances. Troisièmement, le contenu pédagogique enseigné en classe comporte un nombre important de notions abstraites difficilement assimilables par l'esprit de l'apprenant car ce dernier n'a pas de représentation visuelle de la nature de ces notions.

Ces trois axes constituent des obstacles qui empêchent les apprenants de bénéficier totalement de leurs apprentissages. Il est donc question d'étudier l'impact du cours classique de programmation sur l'apprenant pour relever les faiblesses et les failles à exploiter et proposer de meilleures solutions. Il faut ensuite dégager les possibles pistes d'évolution qui offrent aux apprenants les moyens appropriés pour apprendre et surmonter leurs difficultés. Envisager des outils de suivi pour mesurer l'efficacité de nos solutions sur les moyens et longs termes.

Notre travail de recherche s'appuie sur le principe des jeux sérieux. Les jeux dits « sérieux », en plus de leur vocation ludique, remplissent des objectifs éducatifs pensés. C'est justement cet aspect divertissant qui attise l'intérêt des étudiants et les engage dans

le processus d'apprentissage. Nous nous intéressons particulièrement à l'intérêt de l'emploi des jeux sérieux pour enseigner des langages de programmation aux apprenants débutants. En effet, les nombreuses notions abstraites existantes dans la programmation informatique rendent sa compréhension délicate et hors de portée de certains apprenants. Ainsi, nous répondrons à cette problématique en proposant une solution technique et pédagogique capable d'améliorer l'enseignement de la programmation pour la rendre plus accessible et plus concrète, à travers un contexte ludique.

1.1. Motivations

La motivation principale de notre travail de recherche est d'explorer les différentes pistes d'amélioration des méthodes d'enseignement de la programmation informatique, par le biais de nouvelles technologies et d'approches pédagogiques adaptées. Nous souhaitons contribuer à faire évoluer la perception du public de la programmation informatique, qui la considère comme une discipline complexe, abstraite et difficile à maîtriser.

Le choix de ce sujet de thèse est motivé par les éléments suivants:

- **Utilisation en croissance des technologies de l'information de la communication:** l'importance croissante que tiennent les Technologies de L'information et Communication dans différents champs académiques et professionnels est un atout à exploiter, principalement grâce au fait que cette tendance au numérique est une voie d'avenir qui promet le développement et la disponibilité de l'information. C'est dans cette optique que nous voulons proposer une nouvelle contribution dans le domaine éducatif en faveur d'un large public déjà familier avec les supports médiatiques actuels(Yassine et al., 2015).
- **Introduction des jeux sérieux comme une nouvelle approche d'enseignement:** Devant la vastitude et la richesse des supports, nous désirons employer un moyen ludique pour construire le savoir et intéresser des apprenants le plus souvent exigeants lorsqu'il s'agit de leur formation. Dans ce sens, les jeux sérieux se présentent comme une piste à explorer pour transmettre les connaissances et compétences de façon ludique tout en s'appuyant sur un environnement familier à la generation Y (Yassine et al., 2015).
- **Répondre aux attentes des apprenants débutant en programmation qui trouvent de nombreuses difficultés face aux contenus à apprendre:** débuter en pro-

grammation informatique est l'étape la plus souvent délicate car elle demande un temps d'adaptation à une réflexion logique et mathématique que l'apprenant ne possède forcément pas. De nombreux apprenants éprouvent ainsi des difficultés à comprendre et à maîtriser la programmation dès les premières séances. Plusieurs facteurs seront discutés plus loin dans la section problématique. Notre motivation est d'enseigner un langage de programmation à travers un jeu sérieux pour surmonter les nombreuses difficultés que rencontrent notamment les débutants dans ce domaine (Yassine et al., 2017b)(Yassine et al., 2017a).

- **Créer une application référence dans le domaine de l'enseignement de la programmation informatique:** il s'agit de concrétiser notre travail de recherche à travers une application ludique qui englobe notre vision et notre ambition de munir l'enseignement de la programmation d'un dispositif technique et instructif convaincant et solide. Dans la suite de ce mémoire, nous décrivons en détail les aspects techniques et les courants pédagogiques que nous adopterons(Yassine et al.,2018a).

Ces motivations représentent une feuille de route pour avancer en concordance avec les exigences de l'enseignement et l'apprentissage à l'ère actuelle.

1.2. Problématique de recherche

Avant d'apprendre à écrire du code, l'apprenant doit assimiler les différentes notions, souvent abstraites, qu'il est amené à manipuler dans un programme informatique. Selon Gomes et Mendes (Gomes & Mendes, 2008), l'une des raisons de la difficulté de la programmation est qu'elle nécessite un haut niveau d'abstraction et de nombreuses heures d'étude et de pratique. Ceci souligne l'effort cognitif considérable que l'apprenant doit fournir pour comprendre la nature des éléments de base d'un langage. (Muratet, Torguet, Viallet, & Jessel, 2011) ajoute que les apprenants débutants postulent qu'étudier la programmation de façon théorique, et sans pratique, génère un manque de motivation et d'intérêt. L'apprenant, désintéressé du contenu théorique, se concentre davantage sur la pratique sans pour autant accorder le temps nécessaire à la compréhension préalable des notions de base. Effectivement, Sandra Milena Merchán Rubiano (Rubiano, López-Cruz, & Soto, 2015) soulignent le fait que les apprenants débutants se

concentrent davantage sur la grammaire du langage informatique que sur la sémantique derrière la syntaxe. En résulte un apprentissage par cœur de la syntaxe d'un langage au lieu de la vraie compréhension et l'emploi dynamique des éléments qui le composent dans le cadre d'un algorithme. En somme, L'apprenant restreint considérablement son champ de solutions car il ne peut employer que ce qu'il a appris par cœur.

Les enseignants rencontrent également des difficultés lors de la transmission du savoir théorique. L'obstacle majeur qu'ils rencontrent est celui des compétences requises chez l'apprenant et qui sont liées à certaines logiques du domaine mathématique. Le code informatique utilise des algorithmes logiques dont l'architecture est inspirée d'éléments mathématiques considérés comme acquis par l'apprenant. Ainsi, le professeur se voit obligé de remédier à certaines lacunes n'appartenant pas directement au domaine de la programmation informatique, mais liées à des leçons antérieures de mathématiques et logique.

La pratique de la programmation informatique fait également partie de la problématique de notre travail de recherche. Perçue par les apprenants comme une tâche répétitive et dénuée de sens, l'écriture du code doit évoluer vers une activité ayant des objectifs réalistes et motivants. Ceci est un volet important sur lequel nous nous concentrons et auquel nous essayons d'apporter une réponse. En effet, investir des heures de codage ne peut être efficace que si l'apprenant y perçoit un réel intérêt. Ainsi il faut adopter une pédagogie qui reprend des objectifs réalistes et qui mettent l'apprenant dans des situations où les solutions qu'il propose résolvent des problématiques réelles.

1.3. Objectifs de recherche

Nous citons ainsi les objectifs suivants pour tenter de remédier à la problématique principale :

- Analyser les besoins techniques et pédagogiques des apprenants en programmation informatique pour les inclure et les exploiter dans notre solution.
- Comprendre le potentiel et l'apport pédagogique des jeux sérieux dans le domaine éducatif de façon globale et dans l'enseignement/apprentissage de la programmation plus particulièrement. Une étude des jeux existants est donc nécessaire.

- A travers le recensement des jeux sérieux les plus populaires, relever et étudier les différentes techniques de jeu qui y figurent et qui peuvent au mieux servir notre approche ludique de l'enseignement de la programmation.
- De ce recensement, nous devons également lister les différentes approches pédagogiques les plus employées et que nous pourrions améliorer et intégrer dans le processus de création de notre jeu sérieux.
- Proposer un jeu sérieux pouvant constituer un support d'enseignement et d'évaluation pour le professeur et qui soit capable de répondre aux besoins relevés en programmation informatique.

1.4. Organisation du mémoire

Ce rapport de thèse se divise en 5 chapitres organisés comme suit:

- Le chapitre 1 présente l'état de l'art. Plusieurs aspects sont abordés. Dans un premier temps des principes généraux en relation avec la formation en ligne sont présentés de façon à faciliter la lecture des sections suivantes. S'en suit, une description de l'historique des jeux vidéos et des consoles ayant marqué la génération Y. Nous justifions également notre choix des jeux sérieux comme un outil d'enseignement et d'apprentissage de la programmation.
- Le chapitre 2 aborde les techniques de jeux sérieux et les pédagogies pouvant être employées à cette issue. Plus particulièrement, l'accent est mis sur les techniques « Drag&Drop », « Point&Click », « Le code writing », « Le storytelling ». Cette distinction nous permet de choisir les techniques à mettre en œuvre dans la conception et le développement de notre jeu sérieux. Dans ce chapitre, nous aborderons le volet pédagogique à travers l'analyse de plusieurs approches telles que l'Approche Par Compétences, l'Approche Par Résolution de Problèmes, etc. L'objectif étant de pouvoir identifier les techniques et les pédagogies les mieux appropriées à notre contexte.
- Le chapitre 3 met l'accent sur les principales approches de développement mobile. Nous nous intéressons particulièrement à l'approche cross-plateform qui a pour objectif de cibler un large public à travers une application mobile pouvant fonctionner sur les différents systèmes (Android, iOS, etc). Cette comparaison nous a inspiré pour créer un jeu sérieux bénéficiant des avantages du développement mobile Crossplatforme

tout en identifiant le langage de programmation le plus approprié à notre contexte de recherche.

- Le chapitre 4 a pour objectif de présenter notre conception d'un jeu sérieux pour l'apprentissage de la programmation. Il est question de présenter l'architecture multi-agent et son utilisation dans notre contexte. Les différents composants de l'application sont décrits et sont détaillés. Nous décrivons dans ce chapitre le formalisme utilisé pour la description du contenu pédagogique et les taxonomies qui sont employées pour organiser les activités d'apprentissage.
- Le chapitre 5 constitue le cœur de notre contribution, il a pour vocation de présenter la l'application mobile « Globrain's Quest » qui a pour objectif l'enseignement et l'apprentissage de la programmation en langage Python.
-
- Le chapitre 6 se focalise sur l'intégration d'un système multi-agents (MAS) au sein de l'architecture technique de notre application pour dynamiser l'expérience ludique et communiquer avec l'apprenant selon sa manière de jouer et son niveau de maîtrise du langage de programmation étudié. Nous discutons dans ce chapitre les différents agents qui communiquent entre eux différentes informations issues de l'interaction de l'apprenant et renvoient des données pour personnaliser l'expérience ludique.

Finalement, nous synthétisons le travail proposé dans la conclusion générale. Plusieurs perspectives sont présentées afin de poursuivre les activités de recherche sur la solution proposée ou l'une de ses composantes.

Chapitre 1: Etat de l'art

La littérature scientifique est prolifique en matière de e-learning. Il en découle plusieurs axes de recherche qui varient selon les techniques pédagogiques, le dispositif d'apprentissage, ou les caractéristiques du public cible de la formation. A ce stade, il est indispensable de présenter notre domaine de recherche dans sa globalité avant d'entamer les détails spécifiques à notre sujet. Ce premier chapitre est dédié à une présentation du domaine global auquel appartient la notion de jeu sérieux. Nous commençons par présenter le contexte du E-learning. Ensuite, l'accent est mis sur les différentes formes d'apprentissage en ligne. S'en suit l'évocation de l'historique de l'évolution des jeux vidéo dans le temps et l'étude de leur potentiel à transmettre des compétences et des connaissances.

1.1. E-learning

Le E-learning est défini selon Ruth Colvin Clark et Richard E.Mayer (Clark & Mayer, 2012) comme étant un ensemble d'instructions livrées sur un dispositif numérique destiné à soutenir l'apprentissage. Ils stipulent également que le e-learning possède certaines caractéristiques comme par exemple la transmission de l'information à travers les mots et les images, son stockage sur support électronique, sur le Cloud ou sur des serveurs connectés à internet.

L'idée fondatrice du e-learning part de la volonté de faciliter l'interaction avec les supports d'apprentissage. Pour mieux définir le e-learning, il serait judicieux de suivre son évolution au cours du temps. La formation à distance a commencé avec l'invention du papier et de la poste. Le support étant le papier et le canal de transmission étant la poste, ceci a permis la transmission des connaissances d'une personne à une autre, permettant ainsi de véhiculer de l'information entre deux entités distantes.

L'introduction de la technologie a permis à son tour de faciliter la communication entre les différents acteurs. Le milieu éducatif a alors investi dans des moyens informatisés pour transmettre le savoir, au lieu de se contenter des manuels en papier. Les cours par correspondance, par radio, sur support audio et vidéo représentaient alors de nouveaux supports multimédia qui rapprochent le savoir de l'apprenant.

Un autre facteur déterminant dans l'évolution du e-learning est l'explosion d'Internet et du large public qui y a recours. Le e-Learning offre aux enseignants et aux apprenants une expérience d'apprentissage nettement améliorée par l'utilisation de la technologie

web et/ou mobile en consultant des ressources sur le web. Il favorise l'apprentissage en créant des conditions appropriées, et rend la formation possible tout au long de la vie, n'importe où et n'importe quand, d'où une nouvelle considération des concepts : présence/distance, enseigner/apprendre, espace/temps. Cependant, il est tout de même opportun de se demander à quelles conditions une expérience e-Learning peut-elle réussir? Le succès d'une expérience d'apprentissage est lié fortement au degré auquel cette expérience est perçue comme interactive ou réactive et dépend, en grande partie, de l'apprenant, y occupant une place centrale: ses capacités d'acquisition, perceptions, préférences pédagogiques, maturité, motivation, engagement etc. Autant de facteurs qui influent sur le processus d'apprentissage.

1.2. Types de formation en ligne

Plusieurs formes de formation en ligne ont apparues pour répondre à des problématiques diverses en relation avec l'apprentissage en ligne telles que le M-learning qui se réfère à l'utilisation des dispositifs mobiles pour un usage éducatif ou les simulations virtuelles dans des environnements complexes pour dispenser un apprentissage difficile à réaliser dans un environnement réel. Une autre tendance émergente est les Massive Open Online Courses (MOOCs) : plusieurs prestigieuses universités présentent leurs cours à un très large public d'où l'appellation "Massive" dans une perspective de démocratiser l'accès à la formation.

Dans certains cas, le e-learning permet de pallier au manque d'experts qualifiés dans certains domaines, ou lorsque les besoins en enseignement dépassent largement les ressources professorales disponibles. Il facilite également le transfert de connaissances et de savoirs faire à des apprenants distants ou isolés géographiquement (par exemple les systèmes de télé-médecine permettant à un chirurgien d'assister un collègue dans une opération). Le e-learning permet alors de se former sans se déplacer sur le lieu de formation et sans la présence physique d'un formateur.

Une autre forme de e-learning attire de plus en plus de chercheurs: exploiter les jeux vidéo à des fins éducatives. Les jeux sérieux occupent une place importante dans le domaine éducatif car ces derniers incluent une palette d'éléments interactifs pour rapprocher le sujet de l'apprenant et l'aider à percevoir l'utilité de son apprentissage. Un avantage crucial est celui de l'aspect multi-joueurs qui permet aux apprenants d'échanger, de collaborer et de se comparer. L'approche ludique des jeux sérieux vise à exploiter

l'attractivité et l'engagement inhérents au jeu pour attirer l'attention et engager les apprenants dans leurs propres apprentissages et ainsi développer leurs connaissances et leurs compétences dans des domaines variés.

Les jeux sérieux reposent dans certains cas, par exemple les simulateurs, sur des environnements virtuels proches de la réalité ou du moins qui tentent de reproduire des mécanismes réalistes. Dans un contexte virtuel, plusieurs obstacles peuvent être dépassés comme la crainte de commettre des erreurs. Disposant dans son parcours d'un nombre d'essais souvent illimité, l'apprenant peut alors se permettre d'apprendre à son propre rythme en exploitant les outils interactifs dont il dispose pour avancer dans ses apprentissages. Ainsi, la prise d'initiatives, l'échec et l'essai à volonté (là où des environnements réels ne le permettent forcément pas) deviennent autant d'atouts qu'il serait judicieux d'exploiter dans une formation en ligne.

Le choix des jeux sérieux comme support de formation est motivé par le fait que de plus en plus de personnes utilisent plus fréquemment leur appareils mobiles que ce soit dans un cadre formel, comme pour étudier ou travailler par exemple ou informel, pour des loisirs divers. Il serait intéressant d'adapter l'offre de formation à une nouvelle génération qui s'est familiarisée avec une technologie mobile qui se démocratise au fil du temps. C'est dans cette optique qu'il faut modeler l'apprentissage pour le rendre plus interactif et plus engageant en s'appuyant sur une approche ludique. Ceci a la spécificité de rendre le joueur plus réceptif et plus enclin à réaliser des tâches complexes sans avoir peur de l'échec.

1.3. Historique des jeux vidéo

Il est certain qu'avec l'avènement des outils technologiques dans différents domaines, il existe un grand nombre d'alternatives à l'enseignement classique pour apporter plus de motivation et de performance dans l'apprentissage. Nous avons choisi de focaliser notre effort de recherche sur le potentiel ludique du jeu vidéo dans l'amélioration de la qualité de l'enseignement, et ceci pour diverses raisons.

Avant de citer les arguments appuyant notre choix, il est important de décrire le jeu vidéo dans son ensemble à travers l'évocation de son évolution au fil des ans.

1.3.1. Des débuts expérimentaux

Vers les années 1950 et 1960, le jeu vidéo fait ses grands débuts avec Tennis for Two de William Higinbotham et Spacewar! de Steve Russell (Russell, Graetz, software, &

1962, s. d.). Spacewar! est considéré à ce jour comme le véritable premier jeu vidéo fonctionnant sur ordinateur. Son concept repose sur l'idée de contrôler un vaisseau spatial pour détruire le vaisseau adverse. Ces premiers jeux vidéo poseront les bases de la culture vidéo ludique basée sur l'exploitation d'un poste de télévision et d'une manette de jeu pour créer l'interaction.



Fig 1. capture du jeu Spacewar ! (1962)

1.3.2. Facteurs sociaux pour le changement et l'évolution

C'est au début des années 1970 que les ménages commencent à s'équiper de toutes sortes de biens électroménagers, notamment la télévision. L'accès à la technologie change les mentalités de la société qui y perçoit un potentiel de divertissement certain. Les grands industriels commencent alors à songer à une application pour le grand public.

1.3.3. L'ère des bornes arcade

C'est en 1972 que Nolan Bushnell, fondateur de la société Atari, commercialise le jeu vidéo Pong. Ce dernier est un jeu s'apparentant à du tennis de table opposant deux joueurs se renvoyant une balle pour tenter de marquer des points. Pong marque le début du succès de la borne arcade disponible dans des lieux publics comme les cafés et même des salles dédiées aux loisirs vidéo ludiques. Ce mode deux joueurs souligne le caractère multi-joueurs du jeu vidéo. Parmi les arcades les plus connues : Pac-Man (« Pac-Man — Wikipédia », s. d.) et Space Invaders (« Space invaders - Jeu flash d'arcade en ligne - Jeux gratuit online sur 1980 games », s. d.).



Fig 2. capture du jeu Pac-man (1980)



Fig 3. capture du jeu Space Invaders (1980)

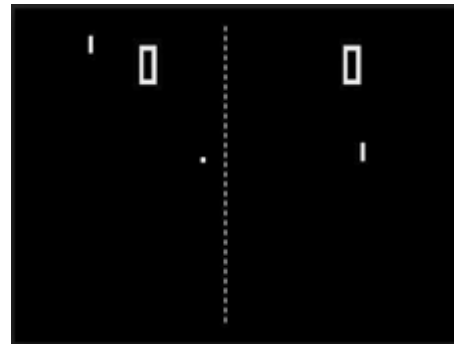


Fig 4. capture du jeu Pong (1972)

1.3.4. L'avènement des consoles de salon :

Malgré des débuts mitigés en raison de leurs prix et surtout au fait qu'une console n'est vendue qu'avec un unique jeu, les consoles de salon ont su s'imposer à partir des années 1980. La Megadrive de Sega et la GameBoy de Nintendo, proposent des jeux aux graphismes plus travaillés et avec des scénarios plus complexes permettant aux joueurs de s'identifier à des héros hauts en couleur. C'est également l'époque où la 3D fait ses débuts dans le jeu vidéo grâce au principe de la 3D isométrique. Ce n'est que plus tard, dans les années 1990, que la 3D permettra au jeu de faire un pas de géant dans l'industrie vidéo ludique.



Fig 5. la console Megadrive



Fig 6. la console Gameboy de Nintendo

1.3.5. La 3D dans le jeu vidéo

En 1994, apparaît au Japon la console de jeux qui va bouleverser l'industrie du jeu vidéo : La Playstation. Initialement, cette console est le fruit de la collaboration entre Sony et Nintendo. Suite à l'abandon de Nintendo en cours de route, Sony poursuit seul le projet de développement de cette nouvelle console. La révolution apportée par la Playstation est surtout technique, grâce à des jeux vidéo proposant des environnements en 3D où le joueur possède une liberté d'exploration inédite pour l'époque.

Le succès la console de Sony introduit sur le marché du jeu vidéo un nombre important de consoles comme la Gamecube et plus tard la Playstation 2 en 2002. Parmi les personnages marquants de cette époque, Lara Croft dans Tomb Raider et Crash dans Crash Bandicoot.



Fig 7. capture du jeu Tomb raider (1997)

1.3.6. L'introduction des consoles portables

A partir de fin de l'année 2004, l'univers de la console portable est en pleine révolution. L'apparition de la Sony Playstation Portable, communément appelée la PSP, et de la Nintendo DS de la société japonaise Nintendo, marquent un tournant dans l'industrie en proposant une plateforme mobile capable de faire tourner des jeux plus ou moins exigeants en ressources. La PSP propose des jeux vidéo avec des graphismes réalistes et des performances inégalées, tandis que la Nintendo DS se voit dotée de deux écrans séparés, dont un tactile qui modifie la manière de jouer. Ainsi, la consommation du jeu vidéo entre dans une nouvelle ère.

1.3.7. Les consoles NEXT-GEN

Vers la fin de l'année 2013, une nouvelle génération de consoles voit le jour, les consoles dites Next-gen. La Playstation 4 en est l'exemple le plus vendu avec plus de 70 millions d'exemplaires écoulées entre 2013 et 2018. Proposant des jeux d'un réalisme immersif, cette nouvelle génération est également dotée de plateformes de téléchargement de jeux gratuits et payants. C'est également l'ère des jeux massivement multi-joueurs qui voient leur succès atteindre des limites inégalées.

Sony Interactive Entertainment ¹ a annoncé la création d'une nouvelle console Playstation, il s'agit de la PS5. Cette version sera disponible vers la fin de l'année 2020 (quatrième trimestre). Les nouveautés apportées par la PS5 ont pour objectif de proposer une meilleure expérience utilisateur. L'aspect visuel étant un facteur déterminant dans les jeux, la console se basera sur un processeur graphique AMD de génération Navi et utilisera la technologie inédite de ray tracing pour produire des scènes qui résonnent avec le réalisme attendu par les jeux. Une autre amélioration notable de la PS5 est la rapidité de chargement des images puisque le stockage qui sera utilisé est de type SSD pour permettre des temps de chargement beaucoup plus courts. La console sera dotée également d'une Puce AMD pour l'audio 3D pour produire un son spatial permettant aux joueurs d'identifier la source du son dans une scène de jeu.

¹ www.playstation.com



Fig 8.la console Playstation 4 de Sony

Le tableau ci-dessous liste quelques propriétés des consoles PS4 et PS5.

Tableau 1: caractéristiques des consoles sony PS4 et PS5

Caractéristiques	PS4	PS5
CPU	Jaguar / 8 coeurs à 1,6 GHz	Zen 2 / 8 coeurs à 3,5 GHz
GPU	GNC	RDNA 2
Puissance GPU	18 unités de calcul à 800 MHz	36 unités de calcul à 2,23 GHz
Mémoire vive	8 Go GDDR5	16 Go GDDR6
Bande passante de la RAM	176 Go/s	448 Go/s
Stockage interne	HDD 500 Go / 1 To	SSD 825 Go
Lecteur	Lecteur Blu-ray	Lecteur Blu-ray 4K

Pour conclure cette section, il est judicieux de mettre l'accent sur l'évolution des jeux vidéo. Cette dernière suit la logique d'un marché mondial exigeant en termes de divertissement, dont les idéaux ont évolué, donnant lieu à une perception plus mature du jeu vidéo. Désormais, la passion vidéo ludique n'est plus considérée comme une simple forme de divertissement, mais comme une expression technique et artistique au potentiel qui dépasse les limites du simple passe-temps.

En vue de l'évolution fulgurante des moyens technologiques employés dans les jeux vidéo, il est désormais permis de considérer ce support comme étant une culture à part entière. L'industrie vidéo ludique, évolue rapidement et génère des revenus importants. On comprend dès lors tout l'intérêt suscité par l'activité de jeu et le potentiel que cette technologie représente pour le domaine de l'enseignement et de l'apprentissage.

Notre réflexion est toutefois portée vers un type de jeux vidéo très particulier qui ne se limite pas au simple divertissement mais qui propose au joueur une expérience derrière

laquelle prend forme un contenu pédagogique structuré. Considérés comme une discipline du e-learning, les jeux sérieux ou serious games représentent l'axe central de notre sujet de recherche.

1.4. Les jeux vidéo comme outil potentiel pour l'apprentissage

Les jeux vidéo sont depuis plusieurs années un média ludique hautement populaire qui comprend des millions d'adeptes dans le monde entier. Plusieurs études ont été depuis menées dont la volonté est de mesurer à quel degré le jeu vidéo peut être un média d'apprentissage et d'enseignement. Parmi les points essentiels relevés nous citons :

- **Postulat 1: le jeu vidéo est capable de mettre en place de nouvelles approches d'apprentissage**

Les apprenants d'aujourd'hui sont entourés de moyens technologiques qui leur permettent d'exploiter une palette d'applications aux multiples vertus pédagogiques. C'est dans cette optique qu'il faut modeler l'apprentissage pour le rendre plus interactif et plus motivant. Ainsi, en se séparant du modèle de consommation du savoir traditionnel qui fait dans certains cas de l'apprenant un simple spectateur, on a tendance à se diriger vers une vision plus engageante qui fait participer ce dernier pour développer son apprentissage.

Selon James Paul Gee (Gee, 2007) les écoles et les milieux professionnels ont beaucoup à apprendre des jeux vidéo. En prenant en considération les conditions stressantes de l'apprentissage et du travail, il est justifié de considérer la place prépondérante que pourrait occuper l'activité ludique. Le même auteur affirme que les jeux sont des plateformes de production et non pas seulement une forme de consommation brute, car le joueur par son action, façonne le jeu à l'image de l'expérience qu'il veut en acquérir. En effet, les activités entreprises dans un jeu vidéo demandent un effort physique et cognitif de la part du joueur. Cet effort, loin d'être contraignant, émane d'une volonté profonde de divertissement et d'une détermination à produire un résultat dans un monde virtuel où la transmission du savoir est en grande partie sous le contrôle de l'apprenant.

En somme, élaborer de nouveaux automatismes de transmission du savoir contribuerait à rendre l'apprenant actif, attentionné et motivé.

- **Postulat 2: le jeu vidéo est un format de divertissement à succès**

Comme cité dans la section consacrée à l'histoire du jeu vidéo, le succès des plateformes vidéo ludiques est un avantage certain à exploiter dans le domaine de l'apprentissage. Le bénéfice des jeux réside dans leur capacité à susciter l'intérêt de l'apprenant et le rend plus réceptif à l'information qu'il reçoit et plus enclin à réaliser des tâches complexes pour obtenir un résultat.

Le joueur est souvent capable d'investir des heures de jeu sans ressentir de lassitude, ceci notamment grâce à de vastes perspectives de découverte et à l'exploitation efficace de sa curiosité et sa recherche permanente de performance et de réussite.

Les jeux massivement multi-joueurs, tendance depuis le début des années 2000, proposent des plateformes de jeux de transaction en ligne d'éléments permettant au joueur d'évoluer en niveau et acquérir plus de puissance en comparaison avec ses amis ou ses adversaires dans le jeu. Par exemple, le jeu World Of Warcraft, sorti en 2005, permet à ses joueurs en contrepartie d'un abonnement mensuel d'incarner un personnage imaginaire dans un vaste territoire peuplé d'êtres humains, de créatures diverses et de héros étincelants. Blizzard Entertainment, créateur de World Of Warcraft, face au succès mondial de leur titre, ont mis en place des transactions monétaires pour accorder au joueur des bonus tels que des armes, des armures, des montures ou une croissance en niveau. Ce vaste succès commercial se traduit par des millions de joueurs connectés, un fait qui lui-même se traduit par plus de compétitivité. Un joueur dans cet univers voudra toujours avoir l'avantage même si cela lui coûte de l'argent réel.



Fig 9. World Of Warcraft

Ceci

démontre de ma-

nière générale la capacité des jeux vidéo à se servir de certains traits de caractère du joueur pour maximiser son temps de jeu et rentabiliser financièrement son expérience.

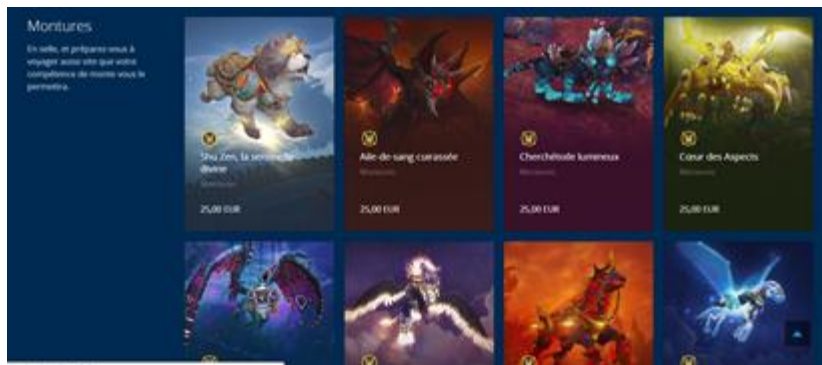


Fig 10. Plateforme de vente en ligne du jeu World Of Warcraft

Notre projet d'application ludique, n'ayant pas un but lucratif, cherche avant tout à exploiter la curiosité du joueur qui découvre une nouvelle méthode d'apprendre la programmation. On tente ainsi d'améliorer la perception de l'apprenant de la programmation informatique en ajoutant un attrait ludique à des activités sobres et linéaires dans le contexte classique d'apprentissage. Notre proposition émane de notre volonté à engager l'apprenant dans l'activité de programmation.

1.5. Les jeux sérieux comme outil d'enseignement et d'apprentissage de la programmation

Cette section se focalise sur les arguments justifiant notre choix des jeux sérieux comme support d'enseignement et d'apprentissage. Ainsi, l'accent est mis sur le potentiel technique et pédagogique de ces plateformes ludiques.

Les jeux sérieux constituent une nouvelle approche pour repenser le processus d'apprentissage en utilisant les avancées des nouvelles technologies de la communication et de l'information. Le terme «jeu sérieux» fait référence aux jeux utilisés à des fins éducatives. En plus de leur but de divertissement, les jeux sérieux ont une vocation pédagogique et se focalisent sur la transmission des connaissances de façon amusante et l'évaluation des résultats de l'apprentissage.

En utilisant des jeux sérieux, les apprenants incarnent des personnages et ressentent des émotions. Cela active leur intérêt car, pour être un bon joueur, un apprenant doit avoir le contrôle de la situation. Les élèves apprennent alors à élaborer des stratégies pour résoudre un problème ou à accomplir une tâche. C'est ce qui est nécessaire pour

créer un algorithme et le traduire en programme informatique: des bonnes stratégies traduites en instructions logiques.

Un autre avantage de l'utilisation des jeux sérieux est qu'ils s'adaptent au rythme de chacun. L'apprenant peut échouer, revenir en arrière, revoir sa leçon et essayer à nouveau de mettre son savoir en application pour réussir. Selon Clark Abt (Abt, 1974), les jeux «sérieux» ne sont pas destinés uniquement à l'amusement, ils servent également des objectifs éducatifs réfléchis et c'est le côté divertissant qui attire l'attention des élèves et les motive à participer au processus d'apprentissage.

Deux paramètres doivent être pris en compte dans les jeux sérieux: (1) le divertissement, représenté par le jeu ou les scénarios ludiques, et (2) le côté éducatif, qui fait référence au contenu implicitement transmis lors de l'expérience de jeu. L'utilisation de ces dimensions permet de repenser l'enseignement de la programmation en abordant quatre problèmes explicités ci-dessous:

1.5.1. Concrétisation des concepts d'apprentissage abstraits

L'étude menée par Essi Lahtinen (Lahtinen, Ala-Mutka, & Järvinen, 2005) a montré que les apprenants éprouvent des difficultés à assimiler des concepts abstraits tels que les pointeurs et la gestion de la mémoire, ce qui nécessite parfois une bonne compréhension de l'architecture de l'ordinateur. Une autre étude menée par Milne et Rowe (Milne & Rowe, 2002) valide l'étude précédente: les concepts les plus difficiles à comprendre en langage de programmation C sont les pointeurs, la récursivité et les structures de données (Yassine et al., 2017b).

Afin de transformer ces anomalies en problème à résoudre et de faciliter l'enseignement de la programmation, le jeu sérieux 'World of Variables' a été mis au point pour enseigner le concept de variables en langage C. Ce jeu est basé sur l'histoire d'une race extraterrestre qui veut quitter sa planète et chercher refuge ailleurs. L'objectif principal du jeu est de comprendre les caractéristiques et le fonctionnement de la variable pour les débutants (valeur, libellé, mémoire, types, etc.).

Tout au long du jeu, chaque niveau comporte une mission, un jeu et aborde un ensemble de résultats d'apprentissage, l'apprenant découvre en jouant comment déclarer une variable, distinguer les types de variables, initialiser et stocker une valeur dans une variable ou attribuer implicitement une valeur à une variable.

Cette approche aide les joueurs à apprendre des concepts abstraits à travers des situations concrètes, augmentant leur motivation et leur intérêt pour la programmation.

1.5.2. Motivation des apprenants

La programmation fait appel à des opérations répétitives et complexes. Pour devenir un expert dans un langage de programmation, il faut beaucoup d'heures de pratique et un intérêt pour cette discipline. Judith Bishop & al soutiennent cette idée, affirmant que la motivation de l'apprenant pourrait être encouragée par l'intégration d'un aspect divertissant à la programmation en utilisant des éléments audiovisuels, des réseaux sociaux et un apprentissage mobile. Ils ont proposé un jeu sérieux Code Hunt (« Code Hunt - Microsoft Research », s. d.). Le joueur doit écrire du code en Java ou en C # pour progresser dans les niveaux du jeu. Plus spécifiquement, il est demandé aux joueurs d'écrire un code afin de se déplacer ou de faire une action. Donc, pour rendre le jeu plus ardu, toutes les activités sont réalisées dans un contexte de compétition où plusieurs joueurs peuvent comparer leurs résultats.

1.5.3. Des outils avec lesquels les apprenants sont familiers

L'émergence de nouvelles technologies a permis au secteur de l'éducation de renouveler et d'adapter son contenu aux apprenants qui utilisent fréquemment des ordinateurs, des smartphones et des tablettes. Selon Houda Mouaheb, un jeu sérieux est un outil d'information et de communication. Issu de l'industrie du jeu vidéo, le jeu sérieux a pour objectif d'enseigner, d'informer, d'éduquer et de sensibiliser (Mouaheb, Fahli, Moussetad, & Eljamali, 2012).

1.5.4. Environnement d'apprentissage réaliste avec des objectifs significatifs

Dans un environnement éducatif standard, les étudiants sont souvent désorientés sans exemples concrets. Les apprenants ont besoin d'un apprentissage situé pour développer des compétences significatives en s'inspirant d'un contexte réel. Donc, pour donner un sens à l'apprentissage, les jeux sérieux offrent une ou plusieurs situations réalistes dans lesquelles le joueur a des objectifs clairs et significatifs. Il s'agit d'un outil permettant aux apprenants de revenir en arrière et de revoir les informations. Par exemple, " darfur

is dying serious game " (interFUEL, 2006) est un jeu sérieux dans lequel l'apprenant joue le rôle d'un villageois du Darfour qui doit assurer la survie de son village régulièrement attaqué par des milices. Le joueur est donc engagé dans une mission humanitaire et se sent motivé pour réussir la mission.

Ainsi, la motivation de l'apprenant est assurée par l'intégration d'un scénario réaliste avec des objectifs éducatifs bien définis, lui permettant de percevoir l'intérêt de son apprentissage.

En résumé, les jeux sérieux se basent sur des principes techniques et pédagogiques nécessaires pour surmonter les problèmes de l'apprentissage classique. Reste à définir trois paramètres par lesquels le jeu sérieux transmet le savoir:

1- Les techniques de jeu qui définissent l'interaction de l'apprenant avec le jeu sérieux et qui servent à :

- Réduire l'aspect abstrait de certains concepts en se basant sur des situations concrètes poussant l'apprenant à interagir avec les objets qui lui sont présentés pour un apprentissage contextualisé.
- Mettre le joueur dans un environnement de jeu afin qu'il puisse faire l'expérience d'une nouvelle façon d'apprendre.
- Exploiter les nouvelles technologies familières aux apprenants et aux joueurs.

2- L'acquisition des connaissances à travers une structure pédagogique appropriée censée :

- Rendre la pratique de la programmation plus divertissante en favorisant l'épanouissement et l'esprit de défi.
- Mettre l'apprenant dans une atmosphère de jeu avec des objectifs éducatifs clairs et significatifs.
- Doter l'apprenant d'un contenu pédagogique structuré, taxonomie et cours.

3- L'architecture technique globale qui décrit la transmission de l'information et assure le dynamisme de l'apprentissage et qui doit :

- Adapter le contenu du jeu aux besoins de l'apprenant, analyser ses faiblesses à travers l'évaluation et le diriger vers les points à améliorer.
- Intégrer un aspect multi-joueurs capable de pousser vers la collaboration ou la compétition selon les situations.
- Offrir une expérience ludique dynamique et non un parcours figé au risque d'ennuyer l'apprenant.

La prochaine étape est celle de l'analyse de l'existant dans le domaine des jeux sérieux, concernant la programmation ou d'autres sujets. Il en existe une multitude à ce jour et présentent des interactions variées et plusieurs visions de l'apprentissage.

1.6. Analyse de jeux sérieux existants

1.6.1. Définition de l'apprentissage par le jeu

Avant de mettre en lumière ce qu'est l'apprentissage par le jeu, Prensky affirme que la condition nécessaire au bon apprentissage est la motivation. Un apprenant motivé n'est pas sujet à la lassitude ou à l'ennui car il apprend en percevant l'intérêt de ses efforts et en poursuivant des objectifs précis (Prensky, 2003).

L'apprentissage par le jeu est défini par Prensky comme étant un processus d'apprentissage exploitant des ressources ludiques comme les jeux vidéo. Cette approche éducative se distingue par un atout majeur absent dans les autres approches: elle propose un environnement divertissant contenant un savoir accessible à volonté, laissant une grande liberté au joueur de décider du moment d'apprendre. Contrairement à une salle de classe où le tuteur impose souvent un rythme d'apprentissage, un jeu vidéo est à l'entière disposition de l'apprenant qui contrôle la quantité du savoir qu'il acquiert et la manière dont il interagit avec ce savoir.

D'apparence parfois superficielle, l'activité pédagogique proposée dans certains jeux peut paraître sans intérêt et manquant d'un but pédagogique concret. Par exemple, un joueur qui construit des bâtiments dans un jeu de construction choisit des matériaux spécifiques, associe des éléments et gère son espace de construction. Au-delà du simple bâtiment qu'il édifie, le joueur apprend l'autonomie en prenant seul des décisions selon les ressources qu'il possède, apprend également à analyser les problématiques pour apporter des alternatives et perfectionne ses méthodes en procédant à plusieurs expérimentations. A ce stade, il est important de mentionner que les jeux sérieux aident les appre-

nants à acquérir non seulement des compétences disciplinaires mais aussi des compétences transversales telles que l'organisation, la collaboration, etc.

Cette activité riche en interaction est possible grâce aux jeux sérieux, qui sont définis comme étant des jeux qui, en plus de proposer du divertissement, intègrent un contenu pédagogique et ayant pour but l'acquisition de connaissances. L'approche ludique des jeux sérieux vise à exploiter l'attractivité et l'engagement inhérents au jeu pour attirer l'attention et engager les apprenants dans leurs apprentissages et développer leurs connaissances et compétences dans des domaines variés.

1.6.2. Domaines d'application des jeux sérieux

L'enseignement interactif par le jeu devient de plus en plus recherché et adopté non pas seulement dans le champ éducatif, mais également dans plusieurs milieux professionnels et notamment pour la formation des cadres d'entreprises. Dans ce qui suit, nous allons citer des exemples de jeux sérieux appliqués à une variété de domaines.

1.6.2.1. Domaine de l'éducation : Exemple de l'enseignement de l'Histoire

Devant l'importance de sensibiliser la jeunesse à l'histoire et au patrimoine mondial, des solutions interactives voient le jour et dont l'objectif est de faire évoluer l'apprentissage de l'histoire en s'éloignant de l'image des longs discours et des textes à apprendre par cœur que souvent l'apprenant doit assimiler par cœur.

Parmi les jeux sérieux à succès, *Le Roi & La Salamandre* de la société française Pinpin Team, est un jeu sérieux réalisé pour le compte du Centre des monuments nationaux dans lequel le joueur incarne le roi François 1er de France pendant la période de la renaissance, vers 1515. Le jeu se présente sous forme d'une suite de chapitres scénarisés et qui reprennent la vie et le règne de ce roi illustre. Toutefois, le joueur n'incarne pas uniquement le roi François 1er, mais également sa fidèle Salamandre qui figure sur son blason. En effet, certaines situations du jeu nécessitent l'incarnation de la Salamandre pour arriver aux objectifs, notamment grâce à sa petite taille.



Fig 11. François 1er à la cathédrale de Reims



Fig 12. La Salamandre du roi François 1er

Le joueur explore plusieurs événements de l'histoire de France à l'époque de François 1er et rencontre une panoplie de personnages qui ont marqué son règne. Les actions entreprises par le joueur sont une sorte de biographie interactive d'un des plus grands souverains d'Europe.

En analysant davantage ce titre, nous avons constaté que son gameplay repose sur le contrôle direct des mouvements des deux personnages et, sur leurs actions sur les éléments interactifs composant les décors du jeu. Des phases de lecture automatique servent d'introduction aux chapitres du jeu pour aider l'apprenant à se situer dans le temps et l'espace. Le jeu est disponible sur les plateformes de téléchargement d'application mobile mais n'est toutefois pas gratuit. Cependant, une version de démonstration est disponible pour donner un aperçu du jeu et pour encourager un public qui se familiarise doucement avec ce nouveau genre d'expériences ludiques.

1.6.2.2. Domaine de la sensibilisation : Exemple de la précarité dans le monde

Les campagnes publicitaires visant à sensibiliser la société sur la pauvreté et la précarité de certains peuples se multiplient et touchent davantage de public grâce à l'avènement des réseaux sociaux et des médias. Ces campagnes visent à donner un aperçu des difficultés vécues par des populations à travers le monde et encourager aux dons.

Darfur is dying est un jeu sérieux qui propose un support de sensibilisation interactif qui permet au joueur d'incarner un(e) villageois(e) de la région du Darfour dans son combat contre les milices armées qui contrôlent sa région.



Fig 13. Le choix d'un(e) villageois(e)

Le choix du personnage, comme montré sur la figure (Fig 13), n'impacte en aucun cas l'expérience de jeu. Le personnage incarné par le joueur appartient à un camp et doit le ravitailler en eau et en nourriture. Le trajet vers le point d'eau par exemple, est constamment surveillé par les milices, incitant le joueur à se cacher en utilisant des éléments du décor.



Fig 14. campement dans la région du Darfour



Fig 15. Trajet de ravitaillement en eau

Par son action, le joueur est investi d'une mission et ressent une responsabilité face à une population de villageois en danger. L'immersion dans l'atmosphère du jeu est une forme directe de sensibilisation qui fait vivre au joueur les conditions délicates au Darfour et participe à la mise au courant du public sur cette grave crise humanitaire.

Le jeu est disponible sur le lien www.darfurisdying.com et est gratuit.

1.6.2.1. Domaine de la formation professionnelle : Exemple de la simulation

La simulation est une approche fortement répandue dans le milieu éducatif. L'interaction entre le système et l'apprenant s'effectue au sein d'un environnement virtuel reproduisant des situations d'apprentissage réelles. L'exploitation de la simulation dans le processus d'apprentissage émane d'une approche de découverte et de construction du savoir (Guéraud & al. 1999). Un simulateur est en quelque sorte un bac à sable

où l'expérimentation est permise, et où l'analyse et la prise de décision sont fortement encouragées. Le milieu de la médecine utilise la simulation virtuelle lorsqu'il s'agit de former les médecins ou les infirmiers à des interventions délicates où l'échec peut représenter un danger pour la santé du patient.

A titre d'exemple, LabQuest(Denami, 2018)est une simulation virtuelle photoréaliste conçue pour la formation et l'évaluation des travailleurs de l'industrie pharmaceutique, qui entrent et opèrent dans une zone aseptique ou salle blanche. Les participants ont été invités à réaliser les procédures en vigueur en salle stérile. Par exemple, le nettoyage biologique, la désinfection des mains ou encore le processus de substitution de la boîte de Pétri. Pour mener à bien ces procédures, les participants disposaient d'objets différents comme un gel désinfectant pour les mains, une solution hydroalcoolique pour le nettoyage et la désinfection des objets et des surfaces, des lingettes, etc.

Cette simulation est utilisée par un certain nombre d'organisations dont des hôpitaux pour former leurs équipes aux métiers de la santé. Cette alternative présente des avantages certains comme en témoigne le site internet du simulateur www.labquest.fr.

1.6.2.2. Jeux sérieux pour l'enseignement/apprentissage de la programmation

La section qui suit est une description brève de certains jeux vidéo destinés à l'apprentissage de la programmation informatique. Notre description de ces applications est faite à partir de notre propre expérience de jeu.

A. Light-Bot

Light-Bot(« LightBot », s. d.) est un jeu sérieux enseignant la programmation informatique et qui met l'accent sur certains concepts fondamentaux tels que les boucles et les conditions. Non relié à un langage de programmation en particulier, le jeu met en scène un robot que le joueur contrôle en glissant des blocs de code préprogrammé dans des cases bien définis pour produire un mouvement. De cette manière, le robot avance selon un parcours, qui augmente en difficulté au fil des niveaux. La difficulté croissante demande au joueur de proposer des solutions plus ingénieuses et enseigne à adopter une logique de résolution de problèmes plus générique.

Adapté aux débutants en programmation, le jeu dispose d'une interface simple permettant au joueur de déduire facilement les commandes à utiliser. Par exemple, un bloc supérieur aux autres requiert de la part du joueur de déposer le bloc de saut contenant

l'instruction qui produit un mouvement de saut chez le robot. Cependant, selon l'étude de (Gouws, Bradshaw, & Wentworth, 2013), l'interface du jeu manque d'une zone de saisie de code qui permettrait une manipulation plus directe.

Le jeu est disponible sur le lien ²

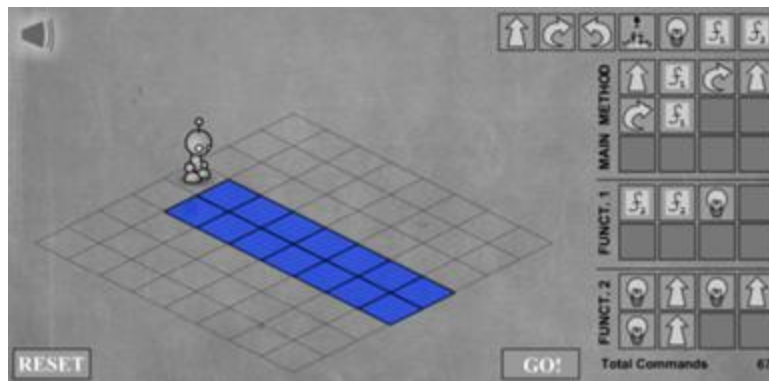


Fig 16. Interface of Light-bot 2.0

B. Hour of code

Hour of Code est une introduction à l'informatique et à la programmation. Le but de ce jeu est de rendre accessible la programmation à un public jeune de manière plus large. Selon l'étude (Du, Wimmer, & Rada, 2016), les joueurs dans Hour of Code peuvent reconnaître une situation problématique et la traduire en un algorithme. L'objectif est d'engager les étudiants dans un processus d'apprentissage leur fournissant des solutions à chaque problème, des solutions sous forme d'algorithmes pouvant être traduits en code. L'interaction dans le jeu se fait selon la technique du Drag& Drop en glissant et déposant des blocs de code pour guider le personnage vers son objectif comme représenté sur la figure (Fig 17). Pour ce faire, les blocs de code doivent être placés dans un certain ordre pour générer un programme écrit en langage Javascript.

Le jeu est disponible sur le lien ³

² www.armorgames.com/play/6061/light-bot-20

³ <https://studio.code.org/hoc/1>

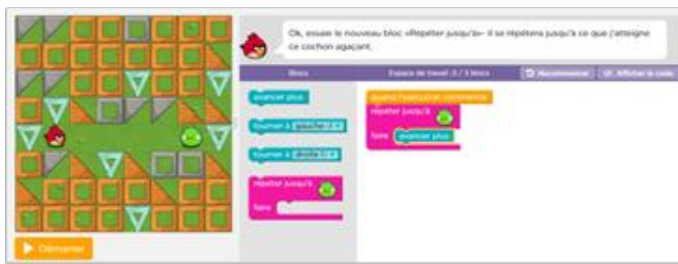


Fig 17.Interface of Hour of code



Fig 18.Code in Javascript corresponding to a level of Hour of code

C. Code combat

Code Combat est un jeu sérieux qui repose sur une mécanique différente de celles des deux jeux cités précédemment. Dans ce cas précis, le joueur doit directement rédiger du code dans un éditeur pour engendrer le mouvement de son personnage dans un labyrinthe. (accessible sur le lien <https://codecombat.com>)

L'interface du jeu, illustrée dans la figure 19, montre une large zone à droite où l'apprenant insère son code. Chaque labyrinthe du jeu possède ses propres obstacles, épreuves et récompenses.

Code Combat intègre un aspect de personnalisation du héros en le dotant d'équipements perfectibles que le joueur peut améliorer en remportant des épreuves. Cet élément est un moteur de motivation qui encourage le joueur à réussir des niveaux et à perfectionner son personnage tout en évoluant dans la maîtrise du langage enseigné dans le jeu. En ce qui concerne le niveau de difficulté, plus vous avancez dans le jeu, plus la difficulté augmente et plus les notions de programmation impliquées dans le processus d'apprentissage deviennent complexes.



Fig 19.Interface of Code Combat

D. World of variables

Selon les créateurs du jeu World of variables, Matej Zapušek and Jože Rugelj (Sillaots & Maadvere, 2013), parmi les objectifs du jeu est de comprendre la notion de variable dans le langage de programmation C comme étant une valeur enregistrée dans la mémoire vive d'un ordinateur. Également, prendre connaissance des différents types de variables ainsi que l'allocation dynamique de la mémoire. C'est donc un jeu sérieux qui se concentre principalement sur une notion de base appartenant au langage de programmation C.

La figure (Fig 20) illustre un niveau du jeu où l'apprenant doit interagir avec des éléments représentant des valeurs de variables de types différents. Par exemple, l'étiquette « true » doit être déposée dans le vaisseau nommé « boolean ». Le jeu est divisé en plusieurs niveaux, chacun d'eux étant dédié à un aspect particulier lié aux variables. Le tout mis en scène dans un scénario original.

Le jeu est accessible à partir du lien http://hrast.pef.uni-lj.si/~svet_spremenljivk



Fig 20. Interface of World of variables

E. Robot ON !

Miljanovic et Bradbury (Miljanovic & Bradbury, 2016), créateur de Robot ON! ont pour projet de créer un jeu sérieux de type puzzle pour apprendre les bases du langage de programmation C++. Destiné à des débutants en programmation, le jeu est divisé en cinq niveaux pour enseigner plusieurs concepts appartenant au langage C++, à partir de la plus simple notion aux plus complexes telles que les boucles et les conditions. L'interface du jeu est constituée de deux zones, la zone droite affiche au joueur les tâches à accomplir, les outils disponibles et le temps restant pour terminer chaque ni-

veau. La zone gauche, quant à elle est la zone où le joueur contrôle l'avatar pour naviguer dans le code source et ainsi compléter les objectifs.

Le jeu est disponible sur le lien <https://github.com/sqlr/RobotON>



Fig 21. Interface de Robot ON!

F. Scratch

Scratch est le nom d'un langage de programmation et aussi le nom d'un logiciel et site web (<https://scratch.mit.edu/>) dont l'objectif est d'initier les élèves à la programmation en générale et à l'algorithmique en particulier. L'environnement scratch se base sur la technique de jeu appelée "Glisser-Déposer", les utilisateurs glissent et déposent des blocs d'instructions pour réaliser une action spécifique sur un personnage ou objet de la scène. Ainsi, l'exécution d'un code sur l'environnement du jeu suite à l'action de l'apprenant permet à ce dernier de comprendre la notion d'instruction utilisée dans la programmation. Il est également possible de se familiariser avec des notions informatiques telles que les structures répétitives quand l'apprenant essaie de réaliser une opération qui doit se répéter à plusieurs reprises ou encore les structures conditionnelles quand l'apprenant essaie de vérifier une condition spécifique avant l'exécution d'un bloc d'instruction.

Scratch permet d'apprendre d'une façon ludique la philosophie de programmation ou de résolution de problèmes en informatique. L'élève apprend à réfléchir aux différents scénarios possibles pour résoudre un problème, de décomposer ce dernier en une série d'étapes successives organisées selon un schéma bien organisé. En plus, scratch permet de générer un prototype fonctionnel que d'autres utilisateurs peuvent tester.

Scratch est un jeu très populaire, il est souvent recommandé pour l'enseignement de la programmation dans le collège et le lycée. Il permet aux apprenants de créer une diver-

sité de jeux et de scénarios ce qui favorise le développement de la compétence d'analyse critique aussi bien que l'imagination et la créativité.

Les interfaces du logiciel Scratch sont intuitives et faciles à utiliser par des apprenants novices, elles se présentent comme suit:

- **L'écran de contrôle ou écran d'exécution:** cet espace permet à l'apprenant de visualiser le résultat d'exécution de son bloc de code comme mentionné sur la figure (Fig 23) .

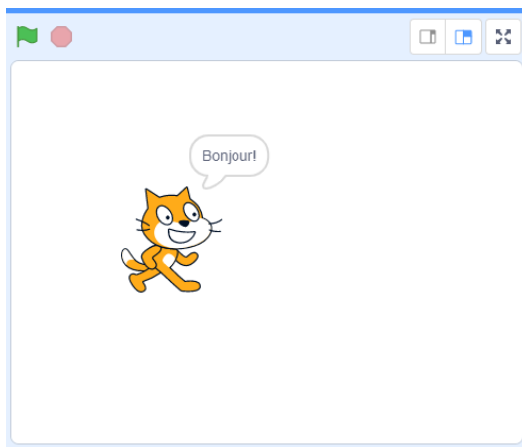


Fig 23. Ecran d'exécution du code dans scratch

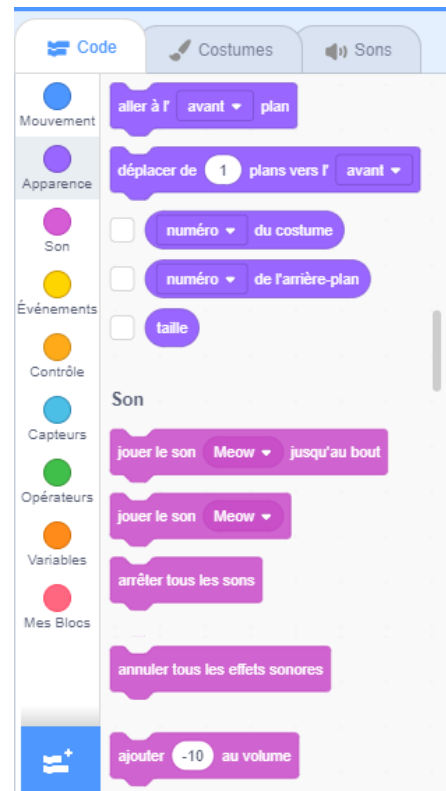


Fig 22. Palette des blocs de code à glisser

- **Blocs de code**

La barre latérale affiche une palette de code (Fig 22) que l'apprenant peut utiliser en glissant le bloc et en le déposant sur la zone de script. Il est également possible d'utiliser l'onglet costume pour personnaliser les personnages et objets utilisés dans les scènes et l'onglet son pour ajouter des effets sonores.

- **Zone de Script**

C'est un espace de travail dans lequel l'utilisateur dépose les blocs de codes pour construire le traitement souhaité.

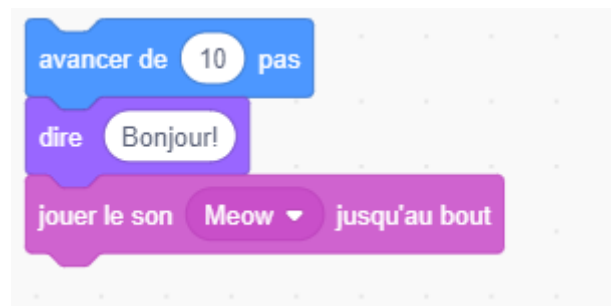


Fig 24. Zone de script affichant des blocs de code à exécuter

1.7. Synthèse

Le succès des jeux vidéo ne cesse de croître. L'interaction ludique devient de plus en plus importante et les joueurs font preuve d'un intérêt et d'une motivation croissants. Les univers qui leur sont proposés créent chez eux un sentiment de curiosité qui les pousse à exploiter pleinement leurs capacités physiques et cognitives.

Plusieurs travaux de recherche ont été menés dans l'objectif de prouver l'efficacité des jeux vidéo dans la formation et l'apprentissage. La performance des « serious games » est constamment comparée à la performance de l'enseignement classique. Cette comparaison a pour but de dégager les avantages de l'apprentissage ludique et étudier les façons de rendre son utilisation plus large.

Cette section a pour objectif de répondre à la question qui représente l'axe autour duquel s'articule notre projet de recherche: pourquoi opter pour le jeu vidéo comme outil d'enseignement et d'apprentissage ?

Il est certain qu'avec l'avènement des outils technologiques dans différents domaines, il existe un grand nombre d'alternatives à l'enseignement classique pour apporter plus de motivation et de performance dans l'apprentissage. Nous avons choisi de focaliser notre effort de recherche sur le potentiel ludique du jeu vidéo dans l'amélioration de la qualité de l'enseignement, et ceci pour diverses raisons.

Chapitre 2: Techniques de jeu et pédagogies pour l'apprentissage de la programmation

Ce chapitre vise à identifier les techniques de jeu et les approches pédagogiques pour enseigner et apprendre un langage de programmation à travers un jeu sérieux. Le terme anglais 'Gameplay' représente la manière avec laquelle les joueurs interagissent avec un jeu et quels mécanismes entrent en vigueur pour définir la façon de jouer. De nombreux facteurs doivent être pris en compte lors du choix d'un Gameplay, à savoir: l'âge du joueur, le support physique, la stratégie de jeu, etc. Dans cette section, nous soulignons les techniques de gameplay les plus courantes relevées dans la littérature, chaque technique est décrite et illustrée à travers un exemple de jeu sérieux dans lequel elle figure. Nous abordons ce chapitre par l'évocation des différentes techniques de jeu décrivant l'interaction des joueurs avec les environnements ludiques qui leur sont proposés. Ensuite, nous décrirons les approches pédagogiques capables de faire progresser l'apprenant investi dans une ou plusieurs activités ludiques. Nous évoquerons par la suite certaines taxonomies figurant dans la littérature et terminerons par l'argumentation nécessaire pour expliquer nos choix techniques et pédagogiques pour la création de notre propre solution.

2.1. Techniques de jeu

2.1.1. Drag&Drop (D&D)

La technique du Drag & Drop est une technique de jeu adapté aux jeux sérieux d'un point de vue pédagogique. En effet, l'intérêt de cette technique réside dans le fait qu'elle convient aux apprenants débutants dans le cadre de leur initiation à une discipline particulière. En utilisant une souris d'ordinateur ou simplement une technologie tactile, le joueur peut glisser et déposer des éléments interactifs du jeu pour générer un résultat. Dans le cas de certains jeux sérieux pour l'enseignement de la programmation, le joueur glisse des morceaux de code et le place correctement dans un programme informatique incomplet pour créer une interaction avec le jeu.

Par exemple, le jeu sérieux Monde de variables emploie le Drag & Drop pour apprendre la notion de variable du langage C de façon concrète en utilisant des objets virtuels interactifs. Comme l'indique la figure (Fig 25), le joueur doit combiner correctement les ob-

jets (bouteilles, liquides, étiquettes, etc.) en les glissant l'un dans l'autre, comme lorsqu'il doit mettre le liquide et la bonne étiquette sur la bouteille. Les combinaisons correctes finissent par être rangées dans les rayons.

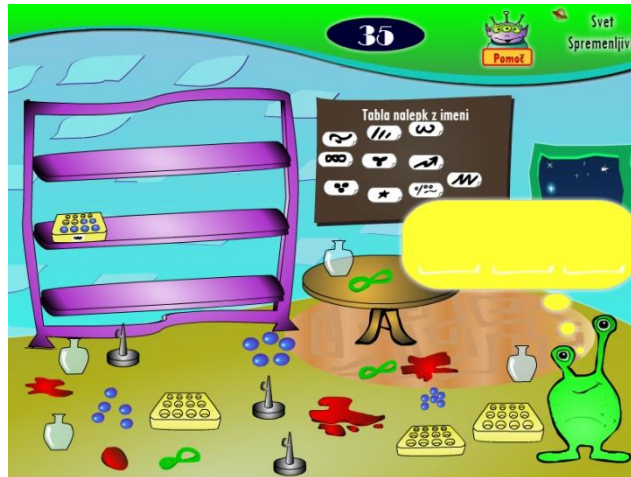


Fig 25. Level 1 of the game 'World of variables'

Le jeu sérieux Tynker, lost in space est également un exemple de l'utilisation du Drag&Drop. Le joueur dispose de différents blocs (Run, Jump..) contenant du code en deux langages : Javascript et Python. Il lui est demandé d'assembler les blocs de façon à faire avancer le héros dans un parcours pour atteindre différents objectifs. L'assemblage final est retranscrit en langage de programmation pour que l'apprenant puisse faire le parallèle entre les blocs qu'il a utilisé et le code résultant. Par conséquent, l'apprenant arrive à mieux décortiquer le code en Javascript ou en Python.

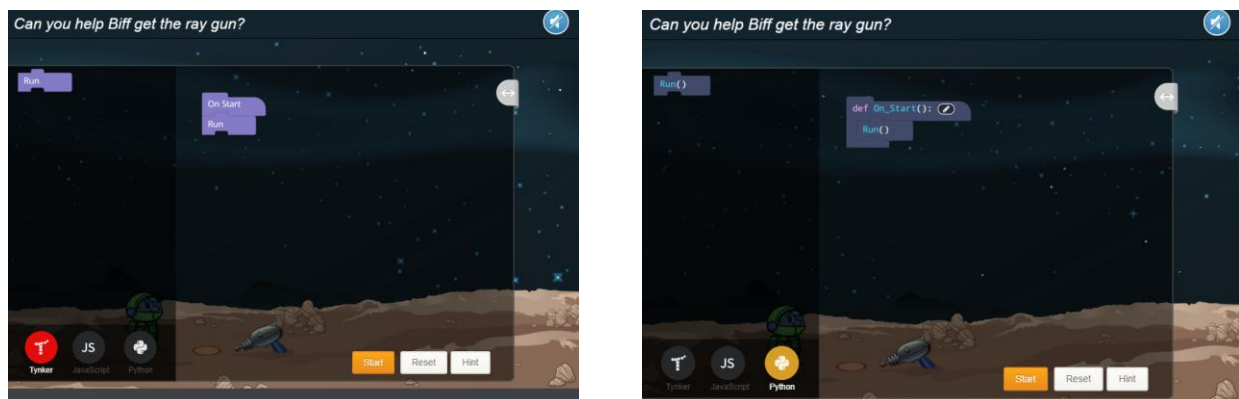


Fig 26. Level 1 of the game 'Tynker, lost in space'

L'avantage d'utiliser une telle technique est de fournir à l'apprenant un environnement plus ou moins riche en interaction, où les éléments à exploiter sont visibles et dont les objectifs sont clairs. Cette forme d'interaction peut également être interprétée comme une forme de concrétisation d'une notion abstraite telle que la variable dans le langage de programmation C. L'apprenant construit par son effort ses propres connaissances et facilite sa compréhension des concepts abstraits. Par exemple, dans "Monde de variables", le Drag & Drop reproduit implicitement le fonctionnement d'une variable, permettant à l'apprenant de comprendre qu'une variable a une valeur (liquide dans la bouteille) et un libellé (étiquette sur la bouteille).

2.1.2 Point&Click

La technique du Point & Click est considérée comme étant la plus simple forme d'interaction dans le domaine du jeu vidéo. Elle ne requiert qu'une action d'appui sur l'élément interactif en question pour produire un résultat. Adapté aux débutants, elle encourage les apprenants à utiliser leur capacité à distinguer les éléments interactifs dissimulés dans un décor visuel riche et labyrinthique, comme en témoigne les nombreux jeux d'enquête dans lequel cette technique est largement employée.

Selon Marine BENEZECH, le choix du Point & click permet de privilégier l'aptitude à la réflexion et à l'exploration au lieu de limiter le joueur à de simples activités ludiques mettant à l'épreuve sa rapidité d'action. D'une part, les jeux impliquant la vitesse sont connus pour leur montée d'adrénaline, où l'attention est absorbée par l'activité du jeu plutôt que par le contenu lui-même. D'un autre côté, les jeux sérieux basés sur le Point & Click sont connus pour entreprendre le suspense autour d'un scénario bien construit et cherche à exploiter la curiosité du joueur et son intérêt vis-à-vis d'une trame scénaristique riches en personnages et en évènements.

Le meilleur exemple de l'utilisation du Point & click est indiscutablement les jeux d'enquête bénéficiant d'une grande popularité et qui sont disponibles à l'achat en ligne sur des sites web comme celui de Big Fish Games. Ce type de jeux vidéo propose une expérience ludique simple reposant sur deux activités principales : la recherche d'objets clés et la résolution de différentes énigmes. Les étapes du jeu suivent l'enchaînement du scénario pour entretenir le suspense et éveiller la curiosité du joueur. Les illustrations ci-dessous proviennent du jeu Hidden Expedition et donnent un aperçu sur les deux phases de jeu citées précédemment.



Fig 27. Phase de recherche d'objets



Fig 28. Phase de résolution d'énigmes

2.1.3 Le code writing

La technique du Code writing consiste, au lieu de travailler avec du code déjà établi, d'en écrire soi-même pour créer une interaction. Adapté à la fois aux débutants et aux apprenants confirmés, cette technique encourage l'apprenant à directement manipuler le code.

Le jeu sérieux Code Combat, disponible sur la plateforme web à travers le lien www.codecombat.com, est un jeu de rôle où l'apprenant mène son héros, choisi parmi une grande sélection de personnages, à travers différents labyrinthes et donjons. Manipuler le héros requiert l'écriture de code en langage Python, Javascript ou CoffeeScript. Par exemple, pour faire avancer le héros à droite, le joueur doit écrire dans une zone dédiée au code la ligne suivante : `hero.moveRight()`, `moveRight()` étant la méthode permettant de réaliser le mouvement à droite.

La forme et les objectifs du parcours imposent au joueur les méthodes à exploiter ainsi que leurs valeurs. La bibliothèque des méthodes à exploiter s'enrichit au fil des niveaux car les parcours deviennent plus difficiles, comportant plus des pièges et plus d'ennemis. Ainsi, au code réalisant un simple mouvement s'ajoute celui permettant d'attaquer et ainsi de suite. A la fin de chaque parcours, l'apprenant reçoit plusieurs récompenses sous forme de gemmes et de pièces d'or à dépenser dans le jeu pour acquérir de l'équipement pour le héros.

La figure (Fig 29) illustre un des niveaux du premier donjon dans lequel le héros doit avancer. On distingue le parcours à gauche, et la console contenant le code à droite. Le jeu bénéficie d'un rendu visuel soigné et détaillé, ainsi qu'une bande sonore variée pour assurer l'immersion dans l'ambiance du jeu.

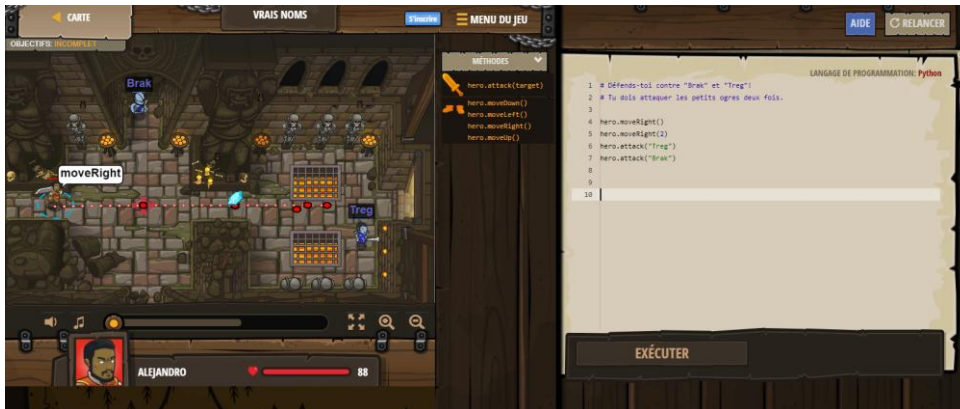


Fig 29. Niveau 4 de Code Combat

Le code writing est donc une méthode centrée sur la maîtrise de la syntaxe d'un langage de programmation et familiarise le joueur avec la structure du code qu'il apprend.

2.1.4 Le Storytelling

Selon CAITLIN KELLEHER et RANDY PAUSCH (Kelleher & Pausch, 2005), Le 'Storytelling' est une méthode naturelle et implicite pour apprendre les concepts basiques de la programmation orientée objet.

Les mêmes auteurs ont affirmé suite à une baisse d'intérêt vis-à-vis de l'apprentissage de l'informatique qu'utiliser l'environnement de programmation 'Storytelling Alice' pour créer des films animés par ordinateur aide à motiver les écolières du premier cycle secondaire pour la programmation, car ils constatent que créer un film animé et apprendre à développer sur un ordinateur peuvent être deux activités fondamentalement similaires.

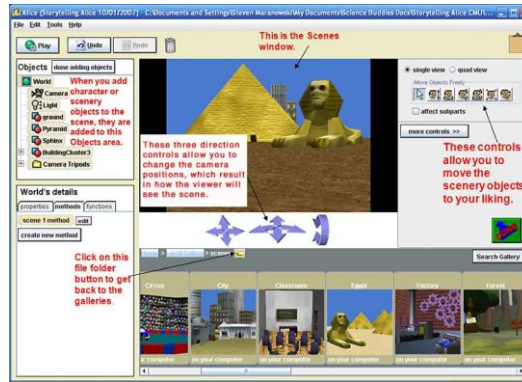


Fig 30. Storytelling Alice

Cette approche technique, bien qu'intéressante, ne nous semble pas adéquate à notre problématique. En effet, nous ne cherchons pas à concevoir une application pour créer des histoires, mais à partir d'une histoire au scénario bien imaginé, intégrer des aspects techniques et pédagogiques pour enseigner un langage de programmation. Ainsi nous, désirons créer une expérience ludique avec un scénario et des personnages établis au préalable.

Tableau 2. Liste des jeux sérieux étudiés (Yassine et al., 2020).

Serious Game	Goal	Target audience	Platform	Language	Game play
Light-Bot (Gouws et al., 2013)	Teaching programming concepts such as loops and conditions.	Novice	Multicross-platform	- ⁴	Drag & drop
Tynker: lost in space (Tynker, s. d.)	Creating algorithms with an increasing complexity to help an astronaut named Biff achieve a set of desired goals.	Novice	Web-based	JavaScript and Python language	Drag & drop
Hour of code (Wilson & Cameron, 2014)	Make programming accessible to a larger audience. Building solutions in the form of algorithms that can be translated into code. The player must guide a bird towards an enemy, represented by a green pig.	Novice	Web-based	JavaScript	Drag & drop
Code combat (Code Combat, s. d.)	The player must write code directly in a dedicated area to make the hero move and advance through a maze. Each maze in the game has its own traps and rewards.	Novice	web-based	JavaScript and Python language	Writing codes
Ruby warrior (« Ruby Warrior - Popular Free Ruby Programming Tutorial Game », s. d.)	Teach open-source programming language Ruby. The learner must write lines of code to make the character evolve in various courses.	Novice	web-based	Ruby	Writing codes

⁴ Not related to a particular programming language

World of variables (Zapušek & Rugelj, 2013)	Understanding of variables in C programming language as a stored value in the memory of the computer. The learner must drag and drop many variables in their proper containers.	Novice	web-based	C	Drag & drop
Robot ON! (Miljanovic & Bradbury, 2016)	A puzzle-like serious game to teach the basics of C++ programming language. Players should demonstrate their understanding of a program's behavior. To progress in the game, the player is asked to complete a set of comprehension tasks.	Novice	Desktop application	C++	Puzzle
Code Spells (Esper, Foster, Griswold, Herrera, & Snyder, 2014) (CodeSpells, s. d.)	It's a 3D game designed to teach Java programming language. The game uses the metaphor of wizardry: the learner is a magician who needs to learn how to remember spells (java programs) and write and execute new ones.	Novice	Windows	Java	Role play game
CMX (Malliarakis, Satratzemi, & Xinogalos, 2013)	In team mode, the learners play the role of activists or hackers. Activists are players who should invade a factory which pollutes the ecosystem and hackers are paid employees of the factory. Both groups are required to collect a series of weapons by solving programming tasks.	Novice	web-based	Java	Drag & drop
Kodable (Fokides & Atsikpasi, 2017) (Kodable, s. d.)	A serious game teaching logic and fundamental concepts of programming. The scenario is about a family who needs to explore many mazes and collect coins. The player must drag and drop commands to advance in the game.	Novice	Android, IOS	-	Drag & drop
Move the turtle (Movetheturtle, s. d.)	Move the turtle is a serious game that teaches how to plan complex operations based on simple steps. The players use predefined commands to build a program and help the turtle get diamonds.	Novice	IOS	-	Point & click
Robot Logic (Robo Logic, s. d.)	The purpose of RoboLogic is to teach Logic. Players must program the robot movement by dragging and dropping commands to the memory of the bot. The goal is to activate all the marked boxes.	Novice	IOS	-	Drag & drop
Robozzle (RoboZZle, s. d.)	Robozzle is a robot-programming game in which players are asked to construct a program using a set of predefined commands. When executed, a program is performing and picking up stars in a grid environment.	Novice	Web-based, IOS, Android	-	Drag & drop

2.2. Vers une Approche pédagogique adaptée aux jeux sérieux

Cette section a pour vocation de décrire les principales approches pédagogiques dans l'enseignement en vue de choisir les plus adaptées à l'enseignement de la programmation. A ce stade, il nous est opportun de faire la différence entre approche et courant pédagogique. Tout d'abord, une théorie d'apprentissage ou courant pédagogique repré-

sente une philosophie ou une perception de l'apprentissage, alors qu'une approche pédagogique décrit la façon par laquelle une situation pédagogique est abordée et présentée à l'apprenant.

2.2.1. Courants pédagogiques

Les principaux courants pédagogiques cités dans littérature sont : Le cognitivisme, le behaviorisme, le constructivisme et le socioconstructivisme, et seront abordés dans les sections ci-dessous.

- **LE COGNITIVISME:**

Le cognitivisme repose sur le principe que l'apprentissage est un processus de traitement de l'information qui accorde de l'importance à la façon par laquelle l'apprenant reçoit, organise et mémorise l'information.

Plus précisément, les cognitivistes s'intéressent principalement à l'activité cérébrale de l'apprenant quand ce dernier investit son effort pour la résolution de problèmes ou pour réaliser différentes tâches liées à l'apprentissage. Ils assimilent l'activité d'apprentissage à un traitement de l'information s'inspirant du modèle de l'ordinateur et du pouvoir de l'intelligence artificielle. Dans ce qui suit, nous allons décrire brièvement ce processus :

Le processus débute avec un stimulus pouvant être visuel, tactile, auditif, etc. Le stimulus est reconnu par l'apprenant et est filtré par la mémoire sensorielle qui peut le contenir pendant une fraction de seconde. L'information est ensuite relayée vers la mémoire à court terme (MCT). Cette même information est ensuite analysée et stockée dans la mémoire à long terme (MLT), reconnue comme mémoire permanente et d'une capacité de stockage illimitée. Pour répondre à une situation problématique, l'apprenant doit chercher dans sa MLT l'information correspondant à la situation en question pour la ramener dans sa MCT. C'est donc un processus qui définit l'apprenant comme un processeur traitant et relayant de l'information comme le ferait un ordinateur. (Basque, 1999)

En somme, pour conclure cette brève description du cognitivisme, ce courant tend à lier la perception de l'apprenant des éléments extérieurs (médias, technologies, etc) qui sont considérés comme des sources d'apprentissage. Par conséquent, les cognitivistes affirment qu'il faut diversifier ces sources pour satisfaire différents profils cognitifs. Nous pouvons donc déduire que la variété des ressources technologiques de notre ère à savoir

les ordinateurs, les tablettes ou les smartphones sont autant de ressources au profit de ce courant.

- **LE BEHAVIORISME:**

Dans le paradigme béhavioriste, l'apprentissage est décrit en termes de modification de comportement excluant les "activités psychologiques". Ainsi, contrairement au courant cognitif, le béhaviorisme met de côté toute l'analyse cognitive liée à l'activité cérébrale de l'apprenant et se concentre sur le comportement de ce dernier vis-à-vis d'une situation donnée.

- **LE CONSTRUCTIVISME:**

Le constructivisme est une théorie d'apprentissage qui fut initiée par Jean Piaget (psychologue et biologiste suisse) en 1923. C'est un courant pédagogique qui a pour objectif de mettre fin aux limitations du courant Behavioriste qui réduit la notion d'apprentissage à stimulus-réponse. En effet, dans ce genre de situations et selon la théorie behavioriste, le sujet est passif et n'intervient pas dans le processus d'apprentissage. Cette hypothèse a été sujette à plusieurs critiques dont l'omission de l'activité mentale ou facteurs internes qui influent considérablement sur l'apprentissage. Dès lors, le courant behavioriste a été remis en cause.

De là s'annonce l'apport du courant constructiviste qui stipule deux énoncés fondamentaux :

L'apprentissage est un processus actif de construction des connaissances (Johnston & Alexander-James, 2015). C'est un cumul d'expériences et d'informations qui surgissent à chaque nouvel apprentissage. Plus particulièrement, le constructivisme met l'accent sur le processus mis en œuvre par le sujet dans une situation d'apprentissage et considère l'apprenant comme un élément central de sa formation.

Concernant le rôle de l'enseignement, le constructivisme stipule que ce dernier n'est plus considéré comme étant un transmetteur du savoir mais plutôt comme un soutien pour l'apprenant dans sa construction de ses propres connaissances. (Muller, Buchheister, & Boutte, 2017)

Du premier énoncé émane l'aspect de recherche de signification du savoir. En effet, au lieu de simplement répondre à un stimulus, l'apprenant cherche un sens à ce qu'il apprend. L'enseignement va dans ce sens en mettant à l'épreuve les significations de l'apprenant, le poussant à perfectionner sa réflexion.

Parmi les idées fondatrices de ce courant est que chaque individu crée et alimente un schéma d'apprentissage en interagissant avec son environnement. Ce schéma est constamment mis à jour et se développe durant la vie d'un individu.

- **SOCIOCONSTRUCTIVISME:**

Le socioconstructivisme est un courant pédagogique qui se focalise sur la dimension relationnelle de l'apprentissage. Pour acquérir des connaissances, l'apprenant doit communiquer et échanger. Malgré le fait que l'apprentissage fasse appel à des ressources internes de l'apprenant, il a toujours lieu dans un cadre social d'où la nécessité de prendre en considération cet aspect. Le courant socioconstructiviste considère l'apprent comme étant le premier responsable de ses apprentissages; dans un environnement dans le quel les apprentissages sont contextualisées et signifiants (Bocquillon et all, 2020). Le socioconstructivisme considère que chaque étudiant possède un ensemble d'acquis en termes de connaissances qui lui permettront d'en construire de nouvelles après résolution d'un problème. Plus concrètement, plusieurs facteurs favorisent le développement d'une nouvelle connaissance selon l'approche constructiviste, nous citons entre autres :

- Les connaissances et compétences acquises.
- Le contexte d'apprentissage.
- Les expériences antérieures.
- La collaboration et coopération entre apprenants, et entre tuteur et apprenants.
- Difficultés rencontrées favorisant la réflexion.

- **CONNECTIVISME:**

Le connectivisme est une théorie d'apprentissage récente qui fut proposée par George Siemens et Stephen Downes qui se sont inspirés du néo-constructivisme de Lev Vygotski. Le connectivisme est née suite à l'utilisation d'internet et des nouvelles technologies à des fins éducatives et repose sur le postulat suivant: l'activité d'apprentissage n'est pas individualiste, elle doit tenir compte des outils numériques utilisés par les apprenants. Tout d'abord les outils informatiques tels que l'ordinateur, smartphone ou tablette rendent accessible l'information n'importe quand et n'importe où. En plus, grâce aux réseaux sociaux et forums de discussion l'apprenant n'est plus isolé mais fait partie d'une communauté en perpétuelle évolution. Selon la métaphore de Stephen Downes «le savoir est distribué à travers un réseau composé de nœuds », chaque nœud

présente une information, donnée, image et est connecté à un autre nœud et c'est l'interaction de l'apprenant avec ce réseau de nœuds (ou liens) qui lui permet de développer des connaissances et compétences.

Le courant connectiviste est lié au concept de MOOC (Massive Open Online Courses) qui ont changé la façon d'apprendre. Les cours des universités les plus prestigieuses sont ouverts au grand public et ce grâce à internet. Dans ce sens, les apprenants peuvent s'inscrire à n'importe quel cours, naviguer dans le cours, communiquer et réaliser des activités pédagogiques en ligne. Le connectivisme est un courant pédagogique tenant en considération les spécificités d'une nouvelle génération d'apprenants de plus en plus connectés.

2.2.2. Approches pédagogiques

Dans cette section, nous allons discuter les principales approches pédagogiques à savoir l'Approche Par Compétences, Approche par Résolution de Problèmes, Approche par Projet, et finalement l'Approche Ludique.

- **APPROCHE PAR COMPETENCES:**

Approche Par compétences (APC), appelée également pédagogie centrée sur les compétences est une approche pédagogique qui se focalise sur le processus d'enseignement-apprentissage et met l'accent sur l'apprenant et son avancement pour développer des compétences. L'un des principes fondateurs de l'APC est la différenciation pédagogique, une pratique qui permet à chaque apprenant d'avancer selon son propre rythme pour atteindre par la voie la plus optimale la(les) compétence(s) souhaitée(s).

Selon le ministère de l'éducation nationale, de la formation professionnelle, de l'enseignement supérieur et de la recherche scientifique Marocain⁵, l'APC est une approche indispensable pour l'analyse des compétences requises pour accomplir des tâches assignées lors de l'exercice d'un métier donné. L'objectif est de former les étudiants et aussi faciliter leur insertion socio-professionnelle. Parmi les éventuelles ébauches de l'application de l'APC proposées par le ministère dans le cadre de la stratégie nationale de la formation professionnelle 2021: la mise en place d'un système intégré d'identification des besoins et de construction de l'offre de formation profes-

⁵ <http://dfp.gov.ma/ingenierie-de-formation.html>

sionnelle. Il s'agit d'une offre de formation dont l'objectif est d'harmoniser la formation et les besoins de la société.

L'approche par compétences nécessite également un enseignement contextualisé. Selon Philippe Pernoud, les enseignements basés sur l'APC donnent un sens aux savoirs enseignés. Pour un meilleur développement des compétences, il est nécessaire de mettre les étudiants dans un contexte d'apprentissage renvoyant à une situation réelle.

En ce qui concerne l'étape de l'évaluation, Paquette Gilbert (Paquette, 2007) propose d'évaluer constamment l'écart de compétence entre le niveau actuel et le niveau visé en faisant appel à une échelle de performance. La performance traduit le degré de maîtrise d'une compétence. Cette échelle permet d'associer à chaque compétence (habilité+ connaissances) une valeur entre 0 et 10. Le niveau de sensibilisation entre (0 et 2.5), le niveau de familiarisation (2.5 à 5), le niveau de maîtrise (5 à 7.5) et le niveau d'expertise (7.5 à 10). Alors que (Benner, 1982) a mis une échelle qui classe les apprenants selon les niveaux : (1) Novice, (2) Débutant, (3) Compétent, (4) Performant, (5) Expert. Ces échelles de performance permettent d'accorder une valeur à la compétence et de situer l'apprenant sur celle-ci.

- **PÉDAGOGIE PAR PROJET:**

L'apprentissage par projet est une pratique pédagogique qui permet l'acquisition des connaissances et des compétences à travers la réalisation d'un projet (seul ou en groupe). Le projet est l'élément crucial autour duquel se réunissent plusieurs individus, leur permettant de contextualiser leur apprentissage et de réaliser une production concrète. La réalisation du projet est supervisée par un enseignant ou tuteur mais les étudiants s'entraident pour la réalisation du travail ce qui favorise le transfère des compétences entre les membres du groupes, le développement de soft-skills puisque le projet est souvent présenté devant le professeur mais aussi devant les autres étudiants (parfois même un large public). En travaillant sur un projet, l'apprenant développe également de l'autonomie.

La pédagogie par projet s'enregistre dans le courant pédagogique constructiviste et socio-constructiviste puisque l'acquisition des connaissances se fait de façon progressive avec l'exposition des idées aux pairs. L'objectif est d'armer l'apprenant d'outils, de le situer dans une démarche collaborative pour qu'il soit capable de mener à bien le projet.

- **APPROCHE PAR RÉOLUTION DE PROBLÈME:**

Approche Par Résolution de Problème, appelée également APP (Approche Par Problème) est une approche pédagogique qui encourage la participation active des apprenants dans le processus d'enseignement-apprentissage. Cette approche s'inscrit dans le courant constructiviste et considère l'habileté à résoudre des problèmes comme complexe et qui exige, l'acquisition des connaissances et des compétences ainsi que beaucoup de pratique. L'apprenant participe à son apprentissage en jouant le rôle de résolveur de problème. Cette approche favorise l'autonomie mais aussi le développement des compétences transversales telles que la communication, l'argumentation ou encore la recherche d'information dans le sens où l'apprenant doit suivre une démarche réfléchie soit seul ou en équipe pour résoudre un problème.

Il s'agit de rendre les apprenants autonomes tout en augmentant leur intérêt et motivation à apprendre. Les problèmes à résoudre sont souvent complexes et semblables à ceux que l'apprenant pourrait rencontrer dans la vie réelle ou même professionnelle.

L'enseignant joue alors le rôle de facilitateur qui a pour objectif de présenter l'activité pédagogique (le problème à résoudre), et de donner des rétroactions pour guider l'apprenant à bien mener l'activité de réflexion. Il a également pour mission de susciter l'intérêt des apprenants et les sensibiliser quant à l'importance du problème à résoudre et son impact sur leur apprentissage.

- **APPROCHE LUDIQUE:**

L'approche ludique repose sur l'exploitation du jeu dans l'éducation pour transmettre un ensemble de savoirs à l'apprenant d'une manière dite ludique. Cette approche vise à immerger l'apprenant dans un environnement proche à la réalité. La particularité d'une telle approche est que l'apprenant apprend sans se rendre compte qu'il est en train d'apprendre, il est actif et engagé, apprend à son rythme. Cette approche permet de susciter chez l'apprenant le sentiment de réussite parce qu'il peut refaire l'expérience autant de fois jusqu'à la maîtrise du sujet.

2.3 Les taxonomies

Une taxonomie est une classification qui organise l'acquisition des compétences de façon hiérarchique en plusieurs niveaux. Cette modélisation met l'accent sur le développement des compétences façon graduée, allant de l'acquisition des connaissances à la manipulation

complexe des concepts pour résoudre un problème. En utilisant une taxonomie, les pédagogues peuvent décrire en un langage humain et de façon structurée, le processus de développement d'une compétence.

2.3.1 TAXONOMIE DE SOLO (Structure of observed learning outcome)




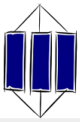
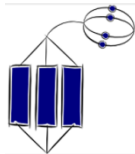
La taxonomie de Solo fut proposée par (BIGGS & COLLIS, 1982). C'est un modèle inspiré de la théorie de Piaget relative aux stades de développement cognitif. Selon les fondateurs de la taxonomie, pour acquérir une compétence, l'apprenant passe par plusieurs stades. Ces mêmes stades sont utilisés en vue de modéliser de façon graduée, l'acquisition des compétences (Learning Outcomes) en accordance avec l'approche pédagogique de l'alignement constructiviste proposé par le même auteur. La taxonomie peut être appréhendée comme modèle d'acquisition de compétences mais aussi comme une échelle pour évaluer, de façon quantitative et qualitative, le niveau de compétence de l'apprenant.

Solo taxonomy organise l'acquisition des connaissances et des compétences en utilisant les niveaux suivants: (1) pre-structurel, (2) uni-structurel, (3) multi-structurel, (4) relationnel, (5) abstrait étendu. Les deux premiers niveaux sont qualifiés de quantitatifs puisque durant les premiers stades de développement des compétences, l'étudiant acquiert des connaissances fondamentales qu'il doit découvrir, lister, identifier, nommer, etc. Les niveaux 3 et 4 sont qualifiés de qualitatifs puisqu'à ce moment l'étudiant peut traiter plusieurs aspects du sujet étudié et commence à construire une relation entre les aspects, analyser, comparer. Le dernier niveau abstrait étendu dénote la capacité de l'étudiant à innover.

Par exemple, (Soobard, Rannikmäe, & Reiska, 2015) utilisent la taxonomie de Solo en vue de concevoir des sujets d'évaluations pour les étudiants du lycée. La taxonomie est utilisée pour évaluer les compétences scientifiques, en particulier la capacité des étudiants à résoudre des problèmes. Ce teste se divise en deux grandes parties : une évaluation quantitative et qualitative selon les niveaux proposés par la taxonomie.

(Yassine et all, 2017) utilisent la taxonomie de Solo pour modéliser et décrire la progression dans l'acquisition des compétences en programmation. Plus spécifiquement, les auteurs proposent un jeu sérieux pour apprendre à programmer. La conception des niveaux des jeux (Level design) est réalisée selon la taxonomie de solo pour décrire les étapes par lesquelles doit passer l'apprenant en vue d'apprendre à programmer (développer une compétence en programmation).

Tableau 3: Les niveaux de la taxonomie de SOLO

Niveau	Stade	Objectif	Compétences
 PRE-STURCTUREL	Stade quantitatif	Aucune ou très peu de compréhension	<ul style="list-style-type: none"> - Découvrir
 UNI-STRUCTUREL	Stade quantitatif	L'apprenant ne peut traiter qu'un seul aspect du sujet.	<ul style="list-style-type: none"> - Identifier - Nommer - Réciter - Suivre les instructions
 MULTI-STURCTUREL	Stade qualitatif	Traiter indépendamment plusieurs aspects	<ul style="list-style-type: none"> - Définir - Décrire - Lister - Combiner
 RELATIONNEL	Stade qualitatif	Construire une relation entre les aspects	<ul style="list-style-type: none"> - Comparer - Expliquer - Classifier - Analyser - Appliquer - Formuler une question - Organiser
 ABSTRAIT ÉTENDU	Stade innovation	Construire de nouvelles connaissances et compétences	<ul style="list-style-type: none"> - Justifier - juxtaposer - Composer - Généraliser - Créer - Faire des hypothèses. - Evaluer

2.3.2. La taxonomie de bloom

La taxonomie de BLOOM a été proposée par Benjamin Bloom dans l'intérêt de d'organiser les niveaux d'acquisition des connaissances allant du niveau simple de mémorisation des connaissances et leur restitution au niveau complexe faisant appel à des capacités cognitives de de niveau supérieur. La taxonomie s'articule autour de 6 niveaux qui sont : (1) Acquisition des connaissances, (2) compréhension, (3) application, (4) analyse, (5) évaluation et (6) synthèse. Chaque niveau est ensuite décrit en utilisant des verbes d'action permettant de

définir les actions de l'apprenant. Ces verbes sont également utilisés par les concepteurs pédagogiques pour décrire les objectifs pédagogiques et par les évaluateurs pour exprimer les capacités de l'apprenant (Anderson & Krathwohl, 2001).

1. Phase acquisition des connaissances :

Dans cette phase l'apprenant doit mémoriser les idées et les concepts. Il s'agit d'acquérir les connaissances et de pouvoir les reproduire/restituer au moment opportun.

2. Phase de compréhension :

Le deuxième niveau de la taxonomie de Bloom est le niveau appelé compréhension. A ce stade, l'apprenant saisi la signification des concepts enseignés. Cela se traduit par la capacité de ce dernier à tirer des interprétations, transposer ou reproduire les concepts étudiés en les communiquant et en formulant des idées. L'apprenant commence également à tisser des liens avec les concepts précédemment mémorisés dans la phase d'acquisition des connaissances.

3. Phase application :

Une fois la phase de compréhension des concepts accomplie, l'apprenant passe à l'étape d'application des connaissances acquises dans un contexte concret. L'application est une étape permettant de confronter l'apprenant à de nouvelles situations, défis, challenges ou problème. Par exemple, à ce stade l'apprenant est capable de résoudre des problèmes en relation avec les sujets étudiés, mettre en oeuvre des théories, etc.

4. Phase d'analyse :

Dans cette phase, l'accent est mis sur la capacité de l'apprenant à décomposer l'information en plusieurs éléments conséquents, les organiser et les examiner pour en tirer une vue d'ensemble. la phase d'analyse permet à l'apprenant d'approfondir ses connaissances mais aussi d'en inférer de nouvelles grâce aux liens établis entre les concepts précédemment appris et aussi aux cas étudiés dans la phase d'application.

5. Phase de synthèse :

La phase de synthèse comme son nom l'indique exprime la capacité de l'apprenant à exploiter l'ensemble des connaissances et compétences acquises comme un tout. L'apprenant est alors capable de construire sa propre perception, créer des solutions, élaborer des projets, imaginer de nouvelles idées, etc.

6. Phase d'évaluation :

La phase d'évaluation est une étape dans laquelle l'apprenant est en mesure de porter un jugement. Cette phase implique l'acquisition des phases qui la précèdent. Pour produire une évaluation, l'apprenant doit maîtriser tous les aspects du sujet à évaluer. Par exemple, l'apprenant peut vérifier des hypothèses, tirer des conclusions, débattre, faire de nouvelles propositions, etc.

Le tableau ci-dessous résume quelques verbes d'action associés à chaque phase d'apprentissage de la taxonomie de BLOOM. Ces verbes permettent au concepteur pédagogique de formuler des objectifs d'apprentissage en adéquation avec cette taxonomie.

2.3.3. La taxonomie de BLOOM révisée

Lorin anderson propose une version révisée de la taxonomie de bloom (Anderson & Krathwohl, 2001) En tenant en considération deux aspects majeurs: tout d'abord, les noms ont été remplacés par des verbes d'action pour mieux formuler les objectifs pédagogiques et pour exprimer la notion de compétences ou encore des capacités que l'apprenant acquiert au cours du processus d'apprentissage. Le deuxième aspect est relatif aux deux derniers stades de la taxonomie qui, sont permutés dans la nouvelle version pour que la taxonomie soit pertinente pour les compétences du 21e siècle (Adesoji, 2018). Le tableau suivant, illustre les différentes phases de la taxonomie de Bloom révisée associées à des verbes d'action.

Tableau 4: La taxonomie de bloom révisée

Connaissances	Compréhension	Application	Analyse	Synthèse	Evaluation
----------------------	----------------------	--------------------	----------------	-----------------	-------------------

Mémoriser	Traduire	Appliquer	Analyser	Créer	Débattre
Définir	Exemplifier	Choisir	Examiner	Concevoir	Juger
Reconnaître	Classifier	Pratiquer	Contraster	Inventer	Vérifier
Nommer	Comparer	Utiliser	Discriminer	Fusionner	Conclure
Lister	Décrire	Planifier	Découvrir une règle	Synthétiser	Valider
Ordonner	Illustrer	Calculer	Extrapoler	Fabriquer	Reformuler
Répéter	Grouper	Démontrer	Schématiser	Mettre en œuvre	Convaincre
Reproduire	Associer	Exécuter	Organiser	Imaginer	Prédire
Énumérer	Interpréter	Développer	Regrouper	Construire	Discriminer
Identifier	Situer	Combiner	Corréler		Estimer
Décrire	Paraphraser	Simuler	Catégoriser		Argumenter
Distinguer	Interpréter	Adapter	Organiser		Persuader
Citer	Reformuler	Implanter	Structurer		
Ordonner		Prendre en charge			
Copier					

2.4 Les principes d'apprentissage dans les jeux sérieux

Pour engager et motiver le joueur, les jeux vidéo se basent sur des critères d'apprentissage. Ces derniers reposent essentiellement sur la capacité d'un jeu à exploiter son univers (personnages, environnements, épreuves...) pour transmettre les connaissances. Ces critères, énoncés par James Paul Gee (Gee & Morgridge, s. d.), sont inclus au sein de trois principes : *Engagement de l'apprenant*, *Apprentissage par résolution de problème* et la *compréhension approfondie*. Ces trois principes mettent l'apprenant au centre du processus d'enseignement.

2.4.1 Engagement de l'apprenant

Ce principe inclut 4 critères:

- **Co-design:** L'apprenant doit ressentir qu'il a un rôle à jouer. Les actions et les décisions qu'il opère ont un effet sur le jeu, il a donc le pouvoir de modeler le jeu selon ses choix et ses préférences. Le concepteur du jeu propose des choix et des possibilités au joueur qui font de lui un co-concepteur.
- **Personnalisation:** Il existe plusieurs types d'apprenants et de joueurs, il faut donc proposer des façons diverses d'aborder le jeu, et qui s'adaptent aux habitudes des différents joueurs.
- **Identification:** le joueur doit pouvoir s'identifier aux éléments du jeu (personnage, situation, scénario...) pour être motivé et engagé, car les jeux ont l'avantage de proposer des personnages attachants et des buts clairs et précis. Par exemple, pour apprendre la géométrie, l'apprenant incarne le rôle d'un architecte qui apprend les bases de la construction des bâtiments.

- **Manipulation:** L'apprenant utilise ses capacités physiques et cognitives pour manipuler les personnages et les environnements.

2.4.2 Apprentissage basé sur Situation problème

James Paul Gee affirme que de nos jours, l'apprentissage ne peut être efficace que s'il est basé sur une philosophie de résolution de problèmes, ainsi ce principe inclut 7 critères :

- **Problèmes bien ordonnés :** Les problèmes proposés aux apprenants doivent s'enchaîner de manière à ce que les solutions proposées par ces derniers puissent être génériques. L'enchaînement doit aller du plus simple problème au plus complexe. Ainsi, la méthode de résolution des problèmes s'améliore progressivement chez les joueurs.
- **Agréablement frustrant:** L'apprenant doit avoir le sentiment qu'il relève un défi dans le jeu, et que ce défi est assez stimulant et que son effort apporte des résultats, même dans les situations d'échec.
- **Cycle d'expertise:** Le défi -> La pratique -> le savoir -> La maîtrise. Chaque niveau du jeu doit suivre ce cycle et plus l'apprenant avance dans le jeu et plus ce dernier propose un nouveau défi, et ainsi aide l'apprenant à s'améliorer de cycle en cycle.
- **Juste à temps & à la demande:** l'apprenant est souvent confronté à une quantité importante 'information, créant chez lui une confusion. Ce critère atteste que l'information 'Juste à temps' ne doit parvenir à l'apprenant que quand il en a besoin et qu'il doit l'utiliser dans son contexte. L'information 'à la demande' quant à elle est une information dont l'apprenant a besoin. C'est en quelque sorte une aide que le jeu doit mettre à la disposition du joueur pour l'aider à réussir.
- **Aquarium :** Lorsqu'il s'agit d'apprendre un sujet particulier, ce dernier peut se révéler plus complexe et devient plus difficile pour l'apprenant de le cerner et le maîtriser. Il faut ainsi prendre un échantillon de chaque élément du sujet pour mieux le comprendre. Dans certains cas, les jeux sérieux divisent le sujet en niveaux d'apprentissage au contenu restreint et plus facilement assimilable. l'apprenant peut ainsi étudier la globalité du sujet en interagissant avec une quantité réduite d'informations à la fois.
- **Sandbox (le bac à sable) :** Le jeu est un univers sûr. Néanmoins, la possibilité d'échec est toujours présente (dans le sens où un risque d'erreur est possible). Cet aspect permet au joueur de prendre des risques tout en sachant qu'il ne court aucun réel

danger et ne peut être reproché pour les erreurs commises. Avant que le joueur puisse apprendre et évoluer, il a besoin de temps pour explorer le jeu, le comprendre et se familiariser avec ses mécanismes.

- **Skills under strategies** : Pour qu'un joueur puisse progresser dans une activité, il doit d'abord maîtriser des bases, ce qui implique beaucoup de pratique. Cette pratique, pour ne pas devenir une contrainte pour l'apprenant, doit être accompagnée d'une stratégie, autrement dit un but.

2.4.3 Compréhension approfondie

A travers le principe de la *Compréhension approfondie*, James Paul Gee essaye d'étudier comment réaliser une compréhension en profondeur et non superficielle. C'est-à-dire quels moyens adopter pour que les informations apprises puissent rester en mémoire pendant longtemps.

- **La logique du système**: James Gee définit un jeu comme étant un ensemble de règles qui interagissent de manière à donner effet aux décisions du joueur (Gee, 2007). Pour résoudre les problèmes, l'apprenant doit d'abord comprendre comment les variables du jeu interagissent pour avoir une idée sur la logique du jeu.
- **Sens à travers actions et images**: Un jeu véhicule son contenu grâce aux textes, mais surtout grâce aux images et aux sons. L'apprenant déduit les choses à faire à travers l'environnement du jeu, c'est une communication graphique et contextuelle.

2.5 Synthèse

Dans leur apprentissage de la programmation, les apprenants sont confrontés à des concepts abstraits et des instructions complexes qui rendent le code difficile à comprendre et à maîtriser. Selon Matej Zapusek et Joze Rugelj (Zapušek & Rugelj, 2013), l'enseignement de la programmation est principalement orienté vers la syntaxe et l'écriture du code informatique, plutôt que la compréhension approfondie des structures de programmation et des concepts abstraits.

Pour répondre à cette problématique, les techniques et les pédagogies relevées présentent autant d'avantages et de bénéfices vis-à-vis de notre conception de l'application à développer

dans le cadre de ce projet de recherche. Dans ce sens, nous estimons que les techniques Drag&Drop et Point&Click sont des techniques familières auprès des apprenants dans une plateforme mobile, et s'alignent avec notre approche orientée mobile d'enseignement de la programmation.

Concernant les approches pédagogiques, notre choix se porte sur une combinaison des approches suivantes: «Approche Par Compétences », « Approche Ludique » et « Approche Par résolution de Problème ». Ce choix encourage les apprenants à découvrir des concepts abstraits et de faciliter leur compréhension grâce à un processus d'enseignement moins frustrant. D'une part, l'approche ludique a pour objectif de rendre l'activité d'apprentissage de la programmation motivante, en encourageant l'apprenant à développer un intérêt pour la programmation. D'autre part, l'approche par compétences organise l'acquisition des compétences de façon hiérarchique en plusieurs niveaux. Cette modélisation met l'accent sur le développement des compétences de façon graduée, allant de l'acquisition des bases à la manipulation de concepts complexes pour résoudre un problème. Finalement, l'Approche Par résolution de Problème est utilisée pour assister l'apprenant progressivement à la résolution d'un problème complexe.

En ce qui concerne les taxonomies de Solo et celle de Bloom, la première servira à décomposer le cours de Python en unités brèves et orientées vers l'explication des notions de base du langage. Les liens entre les notions apprises séparément fera également l'objet de certaines unités de cours. De ce fait, l'apprenant comprend non seulement la nature et le rôle des éléments du langage qu'il apprend, mais également les liens entre eux pour ainsi pouvoir comprendre la syntaxe et en produire une nouvelle.

La taxonomie de Bloom aura un rôle majeur dans la conception des cours et des épreuves de notre future application. Pouvoir inclure les niveaux d'apprentissage évoqués par Benjamin Bloom incite à une réflexion orientée vers la conception d'un cours complet, riche en interaction et d'épreuves permettant au joueur non seulement d'appliquer le savoir qu'il acquiert, mais aussi améliorer ses méthodes en produisant de nouvelles solutions.

Parmi les critères d'apprentissage cité dans la section 2.4, quatre se distinguent par leur rapprochement vis-à-vis des approches pédagogiques choisies.

Le 'Co-design' vise à faire du joueur un acteur engagé à travers ses actions. Il reflète l'approche ludique puisqu'il prend tout son effet dans l'interaction de l'apprenant dans son

action de jouer. Les différentes interactions produisent en même temps du divertissement et des effets dans le jeu.

Le critère des 'Problèmes bien ordonnés' a pour but de proposer un contenu progressif permettant au joueur d'améliorer son esprit d'analyse, pour résoudre des épreuves de programmation proposant un défi croissant dont la difficulté augmente constamment.

Le 'Cycle d'expertise' aide l'apprenant à progresser au fil des niveaux du jeu en suivant une logique alignant défi, pratique, savoir puis maîtrise.

Le 'Juste à temps' et 'à la demande' est également un critère à prendre en considération car il permet au joueur d'accéder à l'information dans le jeu d'une manière simple, pédagogique en évitant le sentiment de confusion.

Les trois derniers critères concordent avec les principes de l'approche par résolution de problèmes que nous avons choisi pour notre future approche pédagogique ont pour but de mettre l'apprenant face à des obstacles ludiques et pédagogiques pour améliorer ses compétences.

Les critères sélectionnés ont un impact sur l'enseignement de la programmation car ils permettent de vérifier, au sein du jeu sérieux, des conditions d'apprentissage favorisant une interaction ludique, un enseignement évolutif et une information accessible à l'apprenant selon ses besoins.

Après avoir étudié les différentes techniques, pédagogies, taxonomies et principes d'apprentissage dans les jeux sérieux, nous passons à la description des plateformes de développement mobiles existantes, pour choisir celle qui convient le mieux aux exigences techniques telles que le langage de programmation à utiliser pour développer l'application, la plateforme mobile cible, et les fonctionnalités propres à chaque plateforme.

Chapitre 3: Approches de développement mobiles

Pour cibler plusieurs utilisateurs, les développeurs d'applications mobiles doivent choisir entre plusieurs stratégies. Ces stratégies diffèrent selon la technologie utilisée, le langage de programmation, le système d'exploitation visé, les performances souhaitées et le type d'application, etc. Chacune de ces stratégies présente des avantages et des inconvénients. Ainsi, avant de développer notre jeu sérieux, il est indispensable d'étudier les différentes approches de développement mobile afin de choisir la plus appropriée à notre contexte. Dans cette section, nous rapportons les approches les plus répandues. Elles sont classées selon les catégories suivantes: Approche de développement native, approche de développement hybride et approche de développement cross-platform.

3.1. Approche de développement native

Les applications mobiles appelée « Platform-Based Native » désigne une catégorie d'applications mobiles conçues et développées pour fonctionner sur une plateforme spécifique (Android, iOS, Windows phone, etc). Pour réaliser de telles applications, les développeurs doivent utiliser un langage de programmation propre à la plateforme cible. Par exemple, l'IDE Xcode et le langage de programmation Objective-C servent à créer une application mobile qui fonctionnera uniquement sur les terminaux munis du système iOS. De même, l'IDE Eclipse et le langage de programmation Java servent à créer une application mobile qui fonctionnera uniquement sur les terminaux munis du système Android. L'intérêt d'utiliser un langage de programmation et des outils dédiés à chaque plateforme réside dans le fait d'offrir une expérience utilisateur qui ressemble aux applications installées dans le smartphone avec une interface homogène avec un aspect visuel familier à ce que l'utilisateur a l'habitude de visualiser. En plus, en utilisant une approche de développement native, il est très facile d'accéder aux fonctionnalités natives offertes par chaque smartphone telles que la caméra, microphone, GPS, Bluetooth, etc. Parce que le langage de programmation utilisé est proche et compatible avec le langage que les constructeurs ont utilisés pour créer ses systèmes. Par exemple, le système Android a été développé en Java. Par conséquent, pour accéder aux fonctionnalités du système, il est très commun d'utiliser le langage Java à cet effet. Finalement, un autre aspect qui caractérise cette approche de développement mobile est l'étape de publication. En utilisant l'approche native, chaque plateforme propose un endroit commun

appelé «Market Place où tous les développeurs vont publier leurs applications». Le tableau ci-dessous illustre les caractéristiques du développement mobile natif.

Tableau 5. Caractéristiques de l'approche de développement native

	Programming language	IDE	Market place	GUI
iOS	Objective-C	XCode	iTunes Apps Store	Cocoa Touch
Windows Phone	C#	Visual Studio	Windows Phone Store	XAML
Android	Java	Eclipse	Google Play Store	XML files
basic for android	Langage spécifique similaire à Visual Basic	Basic4Android	Google Play, Samsung Apps and Amazon Appstore	WYSIWYG ⁶

3.2. Approche de développement hybride

L'idée fondatrice de l'approche de développement mobile hybride est de pouvoir combiner les avantages du développement natif et le web. Une application mobile hybride prend souvent la forme d'une application web empaquetée dans un composant natif appelé «Web View». Plus concrètement, l'utilisateur a la possibilité d'installer l'application et l'exécuter sans pour autant s'apercevoir de la différence entre l'application hybride et celle native: pour l'utilisateur, le produit final est le même. De même, La mise à jour de l'application est transparente à l'utilisateur. Un autre avantage majeur de cette approche est la réduction du coût de développement. Une même application servira à tous les systèmes d'exploitation mobiles (Android, iOS et Windows Phone). Pour ce qui est de langage de programmation utilisé, la majorité des applications hybrides repose sur HTML5, CSS3 et Javascript, ce qui rend l'expérience utilisateur est moins agréable qu'une application native. Un autre inconvénient majeur est l'accès limité aux fonctionnalités avancées du dispositif telles que le GPS, Contacts, Bluetooth. De même, que l'interfaçage avec des applications existantes qui nécessite un développement natif pour pouvoir assurer la communication et l'échange entre deux applications mobiles.

⁶ Acronyme de « what you see is what you get », il s'agit de glisser les composants depuis la barre d'outils et les déposer pour construire l'interface graphique de l'application

3.3. Approche de développement cross-platform

L'approche de développement cross-platform est née d'un besoin: combiner les avantages des deux approches précédemment citées. Plus spécifique, il s'agit de réduire le temps et le coût de développement, mise à jour et maintenance en créant une seule application qui fonctionnera sur tous les systèmes d'exploitation en respectant le principe «write once, run anywhere». Un autre aspect important et qui a été résolu est la performance de l'application: il est possible d'accéder aux fonctionnalités du dispositif mobile et d'offrir une expérience utilisateur similaire à une application native. Les composants graphiques utilisés ressemblent aux composants du smartphone ce qui facilite la navigation dans l'application. Pour ce qui est du langage de programmation, les applications cross-platform se basent sur JavaScript, C# (selon le framework utilisé), etc. Le principal avantage de cette approche est que le produit final est une application mobile native générée à partir du code source de l'application.

Le tableau suivant résume les avantages et les inconvénients de chaque approche de développement :

Tableau 6. Comparaison des approches de développement mobile

	Accès aux fonctionnalités	Maintenance	Performance	Coût	Ergonomie
Native	+	-	+	-	+
Hybride / Mobile	-	+	-	+	+
Cross-Plateforme	+	+	+	+	-

3.4. Frameworks de développement mobile

3.4.1. Titanium

Titanium est un kit de développement d'application cross-plateform utilisant les technologies du web telles que Javascript, CSS et HTML. C'est un projet open source développé par Appcelerator Inc et enregistré sous la License Apache Public License. Les applications mobiles développées par Titanium fonctionnent sur Androis, iOS et Windows phone.

Titanium utilise javascript comme langage pour écrire le code métier de l'application. Ensuite, le SDK se charge de traduire le code javascript en code natif. Par exemple,

lorsque Titanium.UI.Label est utilisée, le SDK procède à sa traduction en UILabel pour Objective-C et en android.widget.TextView pour le langage JAVA. Il permet également d'accéder à plusieurs fonctionnalités du smartphone, dépassant ainsi les limites de l'approche de développement hybride parce que le code généré n'est pas intégrée dans le composant natif « Web View » mais il est traduit dans un langage utilisé pour le développement natif.

Pour ce qui est de l'IHM, elle est développée en utilisant le langage XML, le style en utilisant Titanium Style Sheets (TSS) qui ressemble à la notation JSON et le code métier en JavaScript.

3.4.2. Xamarin

Xamarin.Forms est un framework de développement mobile se basant sur la technologie .NET pour réaliser des applications 100% natives pour iOS, Android et Windows Phone. Xamarin se base sur le principe «Write Once Run Anywhere » c'est-à-dire un seul code est écrit pour produire une application mobile qui a le potentiel de fonctionner sur toutes les plateformes. Par un seul code, nous entendons un seul code pour l'interface graphique et un seul code pour le code métier en C #.

Pour créer des IHMs, il est indispensable de passer par le langage XAML. C'est une façon déclarative⁷ pour créer des IHMs et les partager entre plateformes mobiles et bureau. Tout le code de l'IHM est écrit une seule fois dans ce qu'on appelle bibliothèque de classes portable (PCL), le Framework traduit automatiquement les composants de l'IHM en des composants natif pour les plateformes iOS, Android et Windows.

La particularité de Xamarin est que le code de l'interface graphique et le code métier peuvent être personnalisés pour créer une interface spécifique à une plateforme cible ou pour accéder à des fonctionnalités propres à une plateforme donnée.

3.4.3. Unity

Unity est un moteur de jeu multi-plateforme (smartphone, ordinateur, consoles de jeux vidéo et Web) et un environnement de développement intégré (IDE) pour la création de médias interactifs, généralement des jeux vidéo. La première version fut proposée le 08

⁷ Pour créer une interface graphique, il est essentiel d'écrire le code des composants en utilisant un langage spécifique.

Juin 2005. Unity est développé en C++ et a pour objectif de produire des jeux immersifs en 2D et 3D.

Unity se base sur le principe de programmation par les composants appelés «GameObject» et un ensemble d'Asset. Un asset est un élément contenant des objets, animations et code qui est prêt à l'emploi faisant économiser du temps et des efforts considérables. Les assets sont disponibles sur l'Asset Store («Unity Asset Store», s. d.). En ce qui concerne les langages de programmation utilisés, Unity se base sur le langage C# et le langage JavaScript.

Unity fonctionne sous Windows et Mac OS X. Les applications développées fonctionnent sous Windows, Mac, navigateurs Web, Wii, iOS (iPhone, iPod, Touch, et iPad), Android, Xbox 360 et Playstation 3. Puisque Unity peut exporter le même jeu sur plusieurs plates-formes, le développeur gagne également du temps dans cette phase. L'éditeur se base également sur le principe de "What You See Is What You Get", très utile pour le prototypage rapide des applications.

Unity intègre aussi bien l'IDE et le compilateur libre MonoDevelop. Il est également possible de l'utiliser Visual Studio.

L'auteur (Dickson, 2015) propose d'intégrer le développement des jeux vidéos dans les programmes de formation pour attirer l'attention des étudiants et susciter leur envie d'apprendre et cite Unity comme un moteur de jeu, gratuit, multiplateforme.

3.4.4. React Native

React Native est un framework d'applications mobiles open source créé par Facebook. L'objectif étant de produire des applications mobiles natives et multiplateformes.

Le framework a été développé à l'origine par Facebook et publié en 2015 pour iOS seulement, mais une communauté active a depuis ajouté le support d'Android.

Les applications React Native se base sur langage de script «JavaScript», plus précisément en EcmaScript.

La plupart des fonctionnalités natives communes sont déjà implémentées dans React Native et peuvent être appelées directement depuis le code JavaScript (Kuitunen, 2019). Il n'existe pas un environnement intégré de développement spécifique au framework React Native, mais plusieurs éditeurs peuvent être utilisés à cet effet tels que atom.io, Visual Studio Code, Eclipse, IntelliJ, ou d'autres).

3.4.5. Phonegap/ Cordova

Phonegap est un framework open source pour le développement d'applications mobile cross-platform avec les technologies du web telles que HTML, CSS et JavaScript.

Le framework a été détenu par Adobe puis a été délivré à la communauté d'Apache pour qu'il soit renommé en Cordova. Les technologies utilisées sont: le html pour la création des interfaces graphiques et le CSS pour le style. Javascript assure les différentes interactions et l'ajax pour consommer des services web. Les applications développées avec Phonegap fonctionnent sur Phone, iTouch, iPad, Android, Palm, Symbian et BlackBerry. Il est également possible de combiner le framework Cordova avec d'autres technologies comme JQuery Mobile, SenchaTouch.

3.4.6. Ionic

Ionic est une framework Open Source qui fut proposé par Max Lynch, Ben Sperry en 2013. Il permet de créer des applications natives multiplateformes en utilisant les technologies du WEB telles que HTML, CSS, JavaScript avec la possibilité d'utiliser le langage TypeScript également. Le framework se base sur AngularJS pour la partie web et sur Cordova pour la partie construction des applications natives.

En ce qui concernent les outils de développement, les développeurs peuvent utiliser Ionic Studio, Atom, WebStorm, ALM, Angular IDE by Webclipse(Ionic, 2017).

3.4.7. Sencha touch

Sencha Touch est un framework javascript pour la création des interfaces graphiques pour le web Mobile. Il s'agit de la version mobile du framework JavaScript ExtJS. L'avantage offert par le framework Sencha Touch est que les interfaces produites ressemblent aux interfaces des applications mobiles natives. La dernière version stable 2.4.2 est sortie le 15 juin 2015. Sencha Touch fournit également des plugins pour faciliter les tâches de développement. Les avantages de ces plugins incluent la génération et l'auto-complétion de code, le refactoring et la facilité de navigation vers les classes. Le framework Sencha Touch peut être intégrées dans les IDEs tels que JetBrains, Visual Studio et Eclipse. Il existe également un éditeur d'applications visuelles appelé Sencha Architect permettant de construire des applications HTML5 cross-platforms pouvant fonctionner sur iOS, Android, BlackBerry et Windows Phone 8. Ce framework se base sur les technologies du web tels que HTML5, CSS3 et javascript. Le principal inconvé-

nient est que les applications fonctionnent en local ce qui rend le temps de chargement de l'application assez longs.

3.4.8. Rhodes

Rhodes est un framework open source permettant de créer des applications mobiles cross-platform en se basant sur le langage de programmation Ruby. RhoMobile est un ensemble d'outils pour la création d'applications mobiles natives et multiplateformes. Il se base sur le design Pattern MVC (Model-View-Controller). Pour accéder aux fonctionnalités natives du dispositif le framework se base principalement sur des API Ruby. Rhodes permet aux développeurs de créer des applications mobiles natives à l'aide de technologies du Web telles que CSS3, HTML5, JavaScript et Ruby. L'application résulte sur OS, Android, Windows Phone, Windows Mobile, Windows CE, Windows 10 Mobile et Windows Desktop. La dernière version RhoMobile Suite 6.0 fut proposée en juillet 2017.

3.5. Comparaison des frameworks de développement mobile

En ce qui concerne le critère d'environnement de développement intégré (EDI). Il s'agit d'un logiciel fournissant un cadre de travail, un ensemble d'outils et des plugins pour faciliter le développement et le débogage.

En ce qui concerne le framework Unity, il existe un environnement de travail spécifique permettant de créer des jeux interactifs en assurant la collaboration entre les concepteurs, les développeurs et les designers appelé «Unity». Des jeux peuvent être également développés en utilisant le framework unity au sein de l'IDE Visual Studio et JetBrains Rider (Windows, macOS, Linux).

Pour le framework Cordova, les développeurs peuvent utiliser une édition spéciale de l'IDE Eclipse appelée THym (The Hybrid Mobile project) permettant de développer des applications mobiles hybrides. D'autres IDEs peuvent être également utilisés tels que Netbeans, à partir de la version 8.0, et l'IDE JetBrains, sauf dans la version ultime comme cité dans la documentation, et l'IDE Visual Studio.

Pour développer des applications en utilisant le framework Xamarin, il est important d'utiliser l'IDE Visual Studio ou Visual Studio pour Mac. Pour rendre les jeux attractifs plusieurs bibliothèques sont mises à la disposition des développeurs appelé «Nuggets» tels que *UrhoSharp* qui est un moteur 3D et 2D pour la création des scènes animées.

Des éditeurs spécifiques sont utilisés dans le cas du framework ionic tels que Visual Studio Code, Atom, WebStorm, ALM, Angular IDE by Webclipse.

En ce qui concerne le critère langage de programmation utilisé par les différents frameworks de développement mobile, les approches hybrides privilégient l'utilisation des langages du web tels que HTML et CSS pour la mise en page et le javascript pour le codage des événements. Pour les approches de développement natives, les langages utilisés sont le langage Java, C# et Objective-C. Finalement, pour l'approche de développement cross-platform, les langages utilisés sont le langage C#. Le tableau suivant résume les langages de programmation utilisés par les différents frameworks sujets à l'étude.

Tableau 7. Langage(s) de programmation par framework de développement mobile

Framework	Langage de programmation
Phonegap/ Cordova	Javascript, HTML, CSS
Ionic	Javascript, HTML, CSS
Xamarin	C#
Titanium	Javascript
React Native	Javascript, HTML, CSS
Unity	C#, UnityScript
Flutter	Dart
Rodhes	CSS3, HTML5, JavaScript et Ruby
Unreal Engine	C++
Adventure Game Studio	Langage C

Un autre critère qui nous intéresse est la possibilité de publier l'application mobile sur le *Market Place*. Le *Market place* désigne l'endroit spécifique dans lequel tout utilisateur cherchera à télécharger et à installer l'application. La majorité des frameworks permettent de publier l'application mobile sur iTunes Apps Store et Google Play Store. Alors que Windows Phone n'est pas pris en charge pour Cordova, Sencha, Ionic et React Native Frameworks ce qui limite leur portée à deux types de dispositifs mobiles (Android et iOS)

En ce qui concerne la licence d'utilisations, tous les frameworks sont disponibles en licence libre (Open Source) sauf pour Unity qui est un logiciel propriétaire avec la possibilité de l'utiliser et tester ses fonctionnalités.

Un autre facteur déterminant est la possibilité pour chacun des frameworks d'accéder aux fonctionnalités natives du dispositif mobile. Dans ce sens, les frameworks Unity, Xamarin, Ionic et ReactNative permettent d'accéder aux principales fonctionnalités telles que la caméra, l'accéléromètre, la géolocalisation (GPS), le stockage de l'appareil, les gestes Multi-touches et l'orientation

L'étude proposée par (Ribeiro & da Silva, 2012) prouvent que « Xamarin » et « Titanium », lors de l'exécution, les performances de chargement sont plus faibles, car l'interprétation du code source sur l'appareil doit être effectuée à chaque fois que l'application s'exécute.

Selon (Ciman & Gaggi, 2017) la consommation de batterie est augmentée de façon significative lors de l'utilisation des frameworks de développement mobile hybrides, dépassant dans certains cas 250% par rapport à une application native équivalente. Également, les applications mobiles dites hybrides consomment généralement une quantité plus importante de RAM, elles gèrent les animations avec moins de fluidité et semblent souvent lentes lors des transitions entre les pages de l'application.

Une des particularités du développement cross-platform est qu'il va au-delà des dispositifs standards tels que les smartphones et les tablettes pour couvrir une nouvelle génération de dispositifs et gadget tels que les montres intelligentes, les smartTV ou encore les voitures connectés, les lunettes intelligentes. (Rieger & Majchrzak, 2019).

La majorité des frameworks se concentrent sur la génération du code source de l'application pour une meilleure réutilisation et ne prennent pas souvent en charge la génération de l'interface utilisateur. D'un autre point de vue, et même si les frameworks de développement mobile hybrides ont cette possibilité de réutilisation du code de l'interface graphique puisqu'elles se basent principalement sur les technologies du web. Elles ont une limitation principale: les applications générées ne ressemblent pas aux applications natives. C'est un facteur important puisque la convivialité de l'application joue un rôle important dans l'expérience utilisateur.(El-Kassas, Abdullah, Yousef, & Wahba, 2017) .

	Cordova	Unity	Xamarin	Ionic	React Native	Rhodes	Sencha	Titanium
Approche	Hybride	Cross-Platform	Cross-Platform	Cross-Platform	Cross-Platform	Cross-Platform	Hybride	Hybride
Langage	Javascript, HTML, CSS	C#, UnityScript	C#	TypeScript	Javascript, HTML, CSS	Ruby	Javascript, HTML, CSS	Javascript
Intégration dans un IDE	Oui	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Environnement	VS/ App Builder	Unity3D / Visual Studio	Visual studio (VS)	VS / ATOM/ Angular IDE/ WebStorm/ALM	VS/ ATOM/ WebStorm/ Eclipse/	RhoStudio (Eclipse-based) / RhoHub)	JetBrains, Visual Studio/ Eclipse	Titanium Studio (Eclipse-based)
Support de Market Place	iTunes Apps Store	Oui	Oui	Oui	Oui	Oui	Oui	Oui
	Windows Phone Store	Non	Oui	Oui	Non	Non	Oui	Non
	Google Play Store	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Plateforme cible	Android	Oui	Oui	Oui	Oui	Oui	Oui	Oui
	iOS	Oui	Oui	Oui	Oui	Oui	Oui	Oui
	Windows Phone	Oui	Oui	Oui	Non	Non	Oui	Non
	BlackBerry OS	Oui	Oui	Non	Non	Non	Oui	Oui
Accès aux fonctionnalités natives	Caméra	Oui	Oui	Oui	Oui	Oui	Oui	Non
	Accéléromètre	Oui	Oui	Oui	Oui	Oui	Oui	Non
	Géolocalisation	Oui	Oui	Oui	Oui	Oui	Oui	Oui
	Stockage	Oui	Oui	Oui	Oui	Oui	Oui	Oui
	Gestes Multi-Touch	Non	Oui	Oui	Oui	Oui	Oui	Oui
	Orientation	Oui	Oui	Oui	Oui	Oui	Oui	Oui
Créateur	Apache project community	Unity Technologies	Microsoft	Drifty Co	Facebook	Motorola	Sencha	Appcelerator, Inc.
Licence	Open Source (Licence Apache v2)	Licence propriétaire	Open Source (MIT License)	Open Source (MIT License)	Open Source (BSD 3-Clause)	Open Source (GNU General Public License v3.0)	Open Source (GPLv3 or commercial)	Open Source (Apache 2.0) et commerciale
Documentation	http://cordova.apache.org/ (Jetbrains, s. d.)	https://unity.com/ (Technologies, 2019)	http://www.xamarin.com/ (« Xamarin Documentation - Microsoft Docs », s. d.)	https://ionicframework.com/ (Ionic, 2017)	http://facebook.github.io/react-native/	http://tautologies.com/products/rhobile/	https://www.sencha.com/	http://www.appcelerator.com/titanium/

Tableau 8. Comparaison des framework de développement mobile

3.6. Synhèse

Devant notre besoin de proposer une solution multi-plateforme, Unity, Xamarin, React natives, Rhodes et Ionic se distinguent par leur capacité à produire une application pouvant fonctionner sur les plateformes IOS, Android, Windows Phone et autres. Cependant, les applications créées avec React Native et Ionic ne peuvent être déployées que sur Windows Phone et IOS, et donc ne peuvent pas être retenues comme solutions optimales dans notre contexte.

Ces frameworks se basent sur des technologies web pour une création d'application standards et non dédiées aux jeux, alors que les jeux sérieux demandant des outils de conception graphique pour créer un rendu visuel extrêmement réaliste et sont connus par un environnement riche attirant l'attention du joueur.

Rhodes est un framework qui a particulièrement attiré notre attention car il présente tous les critères pour créer un jeu sérieux. Cependant, il se base sur le langage Ruby et l'inconvénient du framework est qu'il n'est pas très bien documenté par rapport aux autres avec une faible communauté qui soutient le langage ruby par rapport à Java ou C# (Kim, Karunaratne, Regenbrecht, Warren, & Wunsche, 2015)

Concernant Xamarin, nous avons créé un premier prototype en se basant sur le Yassine et al., 2018b) langage C#. Cette première tentative a révélé certaines limites vis-à-vis de la création d'un jeu à l'interaction et à l'environnement voulues riches et ludiques. Ceci est dû au fait que Xamarin se base sur le langage XAML (langage XML) pour définir les composants de l'interface graphique. Pour le joueur, le rendu visuel aussi bien que les interactions ne sont pas intuitives. Par exemple, l'utilisateur peut sentir un ralentissement ou un manque de fluidité ce qui rend l'expérience utilisateur moins attirante.

Finalement, devant les limites des plateformes citées, nous estimons que Unity présente es avantages :

- Présence d'une grande communauté soutenant ce framework.
- Plateforme de conception graphique intuitive et facilement manipulable, offrant une liberté d'intégration d'éléments visuels.
- Gestion automatique des différents formats d'écrans, même pour les nouveaux dispositifs tels que smart Tv ou Smart watch.

-
- Version gratuite puissante et aux fonctionnalités avancées.
 - Possibilité d'utiliser l'interface en drag&drop pour générer les scripts de gestion des scènes du jeu.
 - Taille optimisée de la solution générée par rapport à Xamarin.

Le prochain chapitre a pour objectif de présenter le prototype du jeu Globrain's Quest. Un jeu développé pour l'acquisition des connaissances et compétences en programmation en langage Python. L'accent est mis sur les scènes du jeu aussi bien que l'aspect pédagogique en présentant notre exploitation des taxonomies pour l'organisation des activités d'apprentissage.

Chapitre 4 : conception du jeu *Globrain's Quest*

Dans ce chapitre, nous allons évoquer la démarche suivie pour la conception graphique et technique des scènes de notre jeu sérieux. Dans un premier temps, nous allons aborder l'arborescence des scènes. Par arborescence, nous entendons un schéma de scènes que nous avons conçu et qui est utilisé comme template pour les différents niveaux du jeu. Ensuite, nous décrirons la conception graphique du jeu et quelques questionnements qui nous ont inspiré à ce stade pour créer l'aspect visuel. Finalement, nous présentons les outils, techniques utilisées pour le développement d'un prototype de notre jeu *Globrain's Quest*.

4.1. Arborescence des scènes du jeu

Le développement des différentes vues du jeu requiert une étape conceptuelle dont le rôle est de définir la succession des éléments à afficher au joueur. Dans ce sens, différents types de scènes ont été conçus pour constituer le schéma d'arborescence des scènes du jeu. L'objectif de ce dernier est de présenter une vue globale de l'ordre de présentation du contenu de l'application. Avant de présenter ce schéma, il est essentiel de distinguer entre les différents types de scènes.

4.1.1. Scène d'introduction

Cette scène est présentée au démarrage de l'application et comporte des éléments visuels pour introduire l'environnement du jeu. Elle a pour objectif de poser premiers repères visuels qui familiarisent le joueur avec le style graphique employé.

4.1.2. Scène pivot

La scène pivot est la scène centrale du jeu. Elle représente l'axe central autour duquel s'articule, les éléments accessibles dans le jeu. Dans notre cas, elle peut être la scène des chapitres du jeu qui permettra l'accès aux différentes enseignements et épreuves ludiques du jeu. Elle comporte également la gestion des options du jeu.

4.1.3. Scène Cours

Ces scènes sont d'une importance primordiale dans notre jeu sérieux dans la mesure où ce sont les scènes qui construisent le savoir acquis par le joueur. Au delà de la présentation du contenu à apprendre, une scène de cours doit contenir des éléments ludiques capables d'assurer le sentiment d'immersion dans un environnement d'apprentissage.

4.1.4. Scène Épreuve

Les scènes d'épreuves sont des phases d'évaluation du savoir qui exploitent des techniques ludiques et ont pour objectif de tester l'acquisition des connaissances. Par exemple, le drag and drop est une technique de jeu convenable aux activités contenues dans ce type de scène. L'apprenant est amené par exemple à compléter par exemple un code ou encore à résoudre une situation problématique.

A ce stade de la conception du jeu, ces scènes suffisent pour proposer un prototype complet et cohérent et qui propose une expérience variée et amusante. La figure ci-dessous illustre le schéma d'arborescence des scènes du jeu.

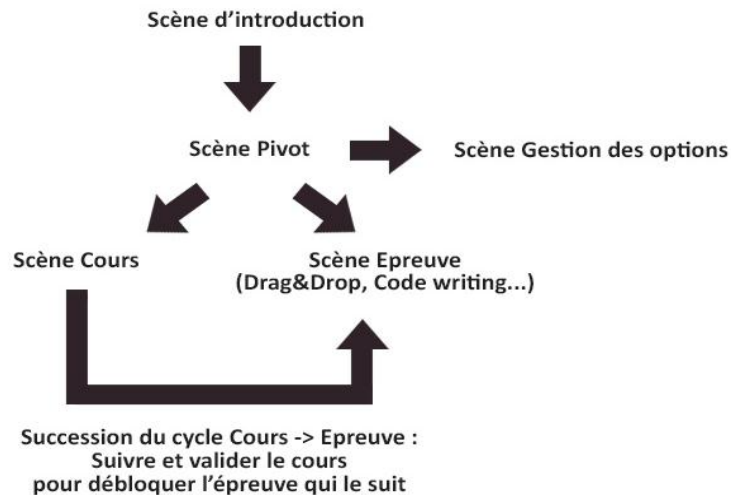


Fig 31. Schéma d'arborescence des scènes du jeu

4.2. La conception graphique

La conception graphique est un processus qui démarre par l'imagination des éléments visuels d'un jeu et qui s'achève par la création des éléments imaginés grâce aux logiciels de dessin à disposition des créateurs. Dès lors, pour bien entamer ce processus, il faut se poser les questions suivantes :

4.2.1. Quel style de dessin adopter selon le public cible?

Différents styles peuvent être imaginés et adoptés dans notre solution car chaque style de dessin possède son propre public. Là où certains jeux proposent des univers 3D réalistes, d'autres optent pour des environnements en 2D plus fantaisistes et colorés. Néanmoins, en considérant que notre objectif est d'intéresser le public le plus large pos-

sible, le style que nous devons choisir doit fédérer des joueurs d'âges et de sexes différents.

Des jeux comme Angry Birds 2 (Entertainment, s. d.) ou Plants Vs Zombies Heroes (Majgaard, 2015) sont appréciés par une grande communauté de joueur dans le monde. Une communauté constituée de jeunes joueurs comme de joueurs adultes, tous intéressés par les perspectives ludiques proposées par ces deux jeux.

Opter pour une direction artistique⁸⁸ réaliste risquerait de détourner le jeune public qui risquerait de juger le visuel comme manquant de fantaisie. Ainsi, nous suivrons l'exemple d'Angry Birds 2 et Plants Vs Zombies Heroes pour imaginer nos propres personnages et les environnements dans lesquels ils vont évoluer.

4.2.2. A quoi ressembleront les personnages du jeu?

Concevoir l'aspect visuel des personnages, avant même de définir leurs rôles, est la première étape dans la création de ceux-ci. La relation entre le joueur et son personnage passe tout d'abord par la perception de l'aspect physique du personnage incarné. Les joueurs s'identifient plus facilement à des personnages à l'aspect humain car la similitude physique entre le joueur et le personnage est l'élément fondamental du sentiment d'identification.

Nous devons dans ce sens, imaginer une série de personnages à l'allure humaine mais tout en gardant l'aspect cartoon inspiré des deux jeux cités plus haut.

4.2.3. Quel logiciel utiliser pour créer les visuels ?

Parmi un large choix de logiciels de dessin assisté par ordinateur, notre choix s'est tourné vers le logiciel d'Adobe Photoshop CC. Ceci principalement en raison de la multitude d'outils proposés dans le logiciel et qui sont orientés vers la pratique du dessin.

Photoshop fonctionne avec le principe des calques, qui sont des couches superposées permettant de travailler sur chaque élément de manière séparée. Cette flexibilité permet de concentrer l'attention de l'artiste sur les détails de chaque dessin pour produire des illustrations de qualité. Comme le montre la figure (Fig 32), le dessin illustre un des personnages du jeu dans un environnement végétal dont l'atmosphère particulière est illustrée par un ciel sombre et une lune brillante et spectrale.

⁸⁸ Direction artistique représente l'ensemble des choix artistiques qui relèvent du jeu.

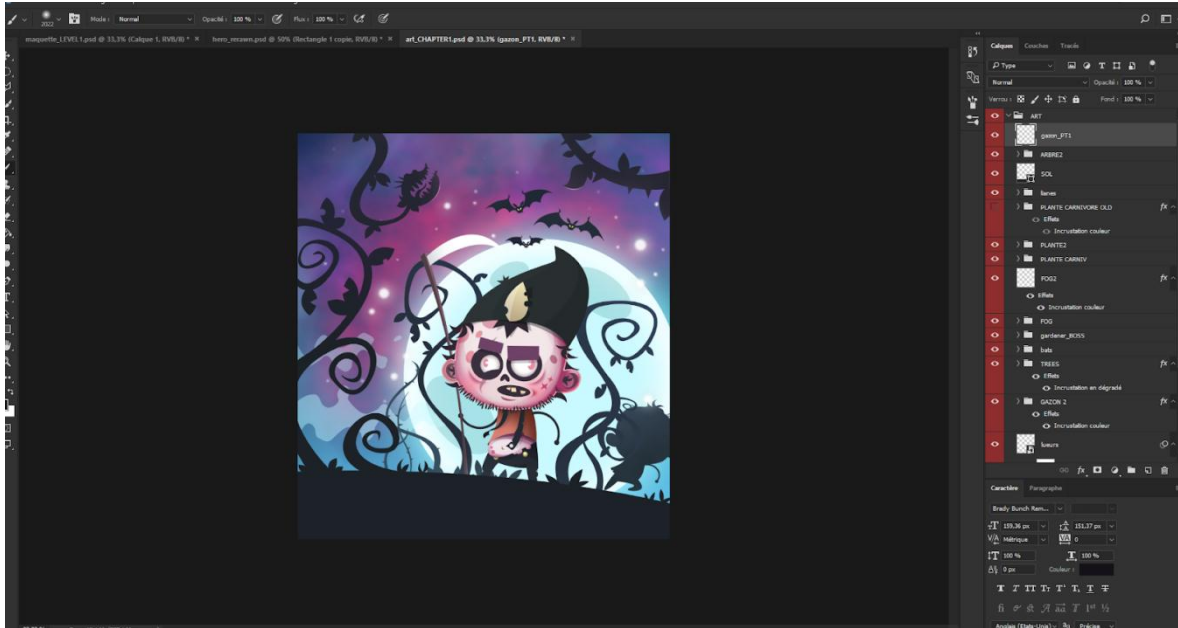


Fig 32. Création de personnage avec Adobe Photoshop

4.3. Développement du prototype

Pour le développement du prototype du jeu. Nous avons utilisé Unity3D qui est un environnement de développement pour la production de jeux vidéo 2D et 3D; Unity 3D permet de développer un jeu pouvant fonctionner sur tous les dispositifs mobiles aussi bien que sur les ordinateurs.

Le choix du moteur de jeu Unity3D est justifié par sa capacité à créer rapidement des prototypes de jeux disponibles sur toutes les plateformes et pour plusieurs résolutions d'écran. En plus, il supporte le langage de programmation C# ce qui nous permettra de réaliser du code derrière les interfaces graphiques. Un autre avantage d'Unity, c'est qu'il est supporté par une large communauté ce qui rend facile l'étape d'apprentissage et de familiarisation avec l'outil.

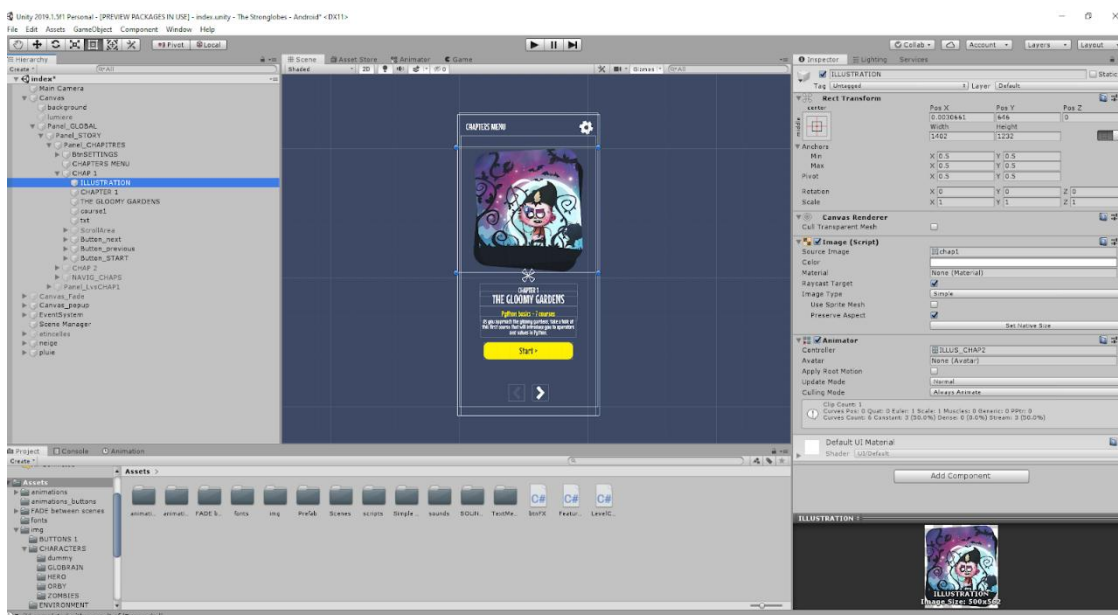
Pour créer le jeu sur Unity3D, nous nous sommes basés sur la programmation par composants. En effet, la première étape est de créer une scène contenant des objets de différents types appelés "GameObject". Chaque GameObject est une association d'éléments prévus pour opérer un traitement particulier. Les éléments disposés sur une scène affichent des "Assets", ces derniers étant des ressources importées dans l'application et peuvent être des images, des sons, des animations, des polices de textes, des scripts en langage C#, etc.

4.3.1. Description de l'interface utilisateur

Sur la figure ci-dessous (Fig33) le rendu visuel de la scène pivot citée plus haut. La scène figure au centre du plan de travail. Sur le côté gauche de l'interface d'utilisateur de Unity figure une arborescence intitulée "Index". Elle présente la succession des couches d'éléments qui composent la scène, à savoir le titre, le bouton de réglages, l'image centrale animée, le texte descriptif du chapitre, le bouton de démarrage et les boutons de navigation tout en bas.

Sur le côté droit s'affiche les propriétés de l'élément sélectionné dans l'index, comme par exemple, la hauteur et la largeur de l'image du centre. En bas se présente la bibliothèque des ressources ou les "Assets". Ils peuvent être rangés dans des dossiers spécifiques pour faciliter le repérage des éléments recherchés.

Grâce au bouton play en haut au centre, il est possible de visualiser l'exécution de l'application tout en opérant des modifications en temps réel.



4.3.2. Exemple de traitement simple pris en charge par Unity

Le bouton jaune Start dans la scène permet d'ouvrir la liste des niveaux du premier chapitre du jeu. En terme de développement, la fonction contenu dans le bouton jaune Start ne nous demande pas de développer du code manuellement, car Unity permet automatise ce traitement à travers le panel illustré dans la figure suivante (Fig 34) :

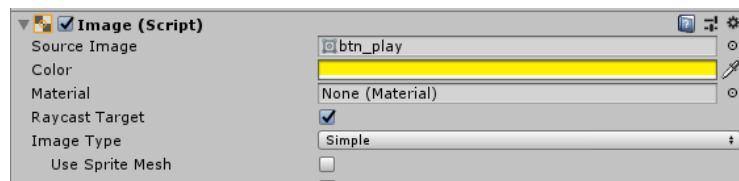


Fig 33. Le rendu visuel de la scène pivot sous unity3D

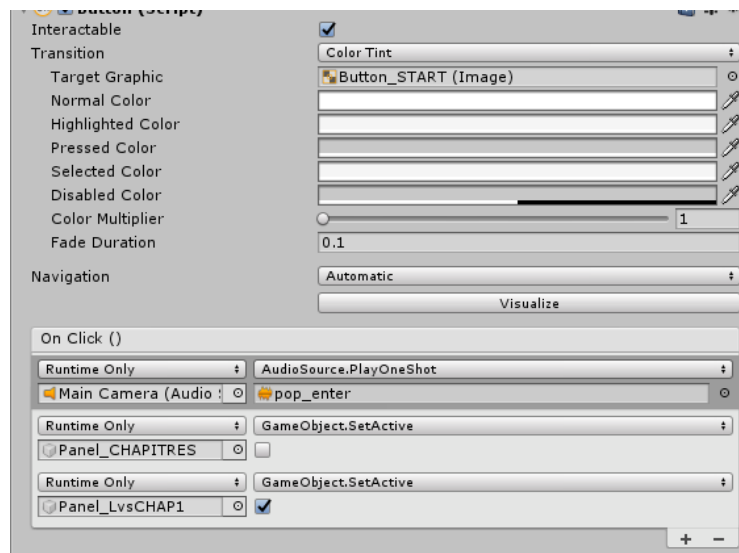


Fig 34. Exemple de traitement simple pris en charge par Unity

La fenêtre On Click() permet de choisir dans une liste de différents types de traitements, celui qui sera dans la fonction On Click () sur le script en C# qui concerne cette scène. Ainsi, trois traitements se succèdent: Premièrement, le déclenchement d'une bande sonore nommée '*pop_enter*', ensuite la disparition du panel contenant la liste des chapitres et enfin l'apparition du panel qui contient les niveaux du chapitre 1. Un processus simple et facile à réaliser grâce aux automatismes présents dans Unity.

Ce genre de traitement se répète dans plusieurs scènes du jeu que nous présenterons dans la section suivante. L'important est de signaler que les mécaniques les plus simples ne requièrent pas d'écriture de code et représentent ainsi un gain de temps considérable pour éventuellement développer des composants plus complexes et personnalisés.

4.3.3. Exemple de traitement codé en C#: le questionnaire de cours

Dans notre jeu sérieux, nous avons prévu une étape incontournable liée au cours enseignés, les questionnaires. En effet, après avoir consulté un cours, l'apprenant est amené à répondre à différentes questions pour vérifier ses connaissances et valider sa maîtrise du cours appris.

Créer les règles techniques de ce questionnaire requiert le développement de fonctions capables de proposer des questions au joueur et de vérifier si les réponses sont justes ou fausses. En plus de cela, nous avons créé un aspect ludique lié à cette activité et qui sera expliqué en détail dans la section suivante.

Unity ne propose pas de schéma de questionnaire prêt à l'utilisation tant les cas de figure pour une telle activité peuvent varier selon le besoin. Nous avons donc développé en C# différents scripts qui gèrent chacun un aspect différent :

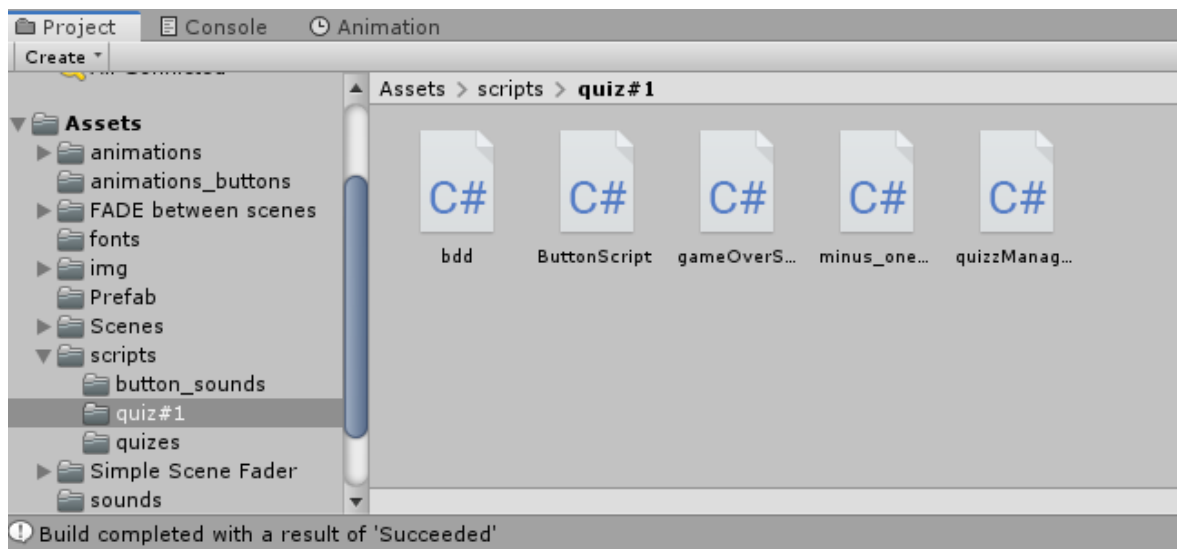


Fig 35. Extrait des scripts c# pour développer le quizz sur Unity3D

Dans la figure ci-dessus, les scripts contenus dans le dossier « quiz#1 » gèrent le déroulement du questionnaire du premier cours du jeu. Par exemple, le script `bdd` est une base de données de questions appelées au moment de l'exécution du script `quizzManager`. Ce dernier contient le code nécessaire pour générer les questions, les afficher au joueur et vérifier si ses réponses sont justes ou erronées. Le script `ButtonScript` quant à lui, gère les autorisations concernant les moments où le joueur est autorisé à appuyer sur les réponses pour ainsi éviter de répondre plusieurs fois en appuyant sur la réponse plusieurs fois. Les questions sont générées de manière aléatoire et la base de données contient une cinquantaine de questions différentes, et il n'est demandé au joueur que de réussir quatre bonnes réponses successives.

En appuyant sur l'icône d'un script, on peut visualiser son code sans devoir ouvrir le fichier, la figure suivante présente une partie du code du script quizzManager.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class quizzManager : MonoBehaviour
{
    public Text txtquestion, txtPvVirus;
    bdd bdd_quizz;
    public string p1, p2, p3, reponse;
    public bool canPlay;
    int pvVirus = 5;

    public GameObject oneDown1;
    public GameObject oneDown2;
    public GameObject oneDown3;
    public GameObject oneDown4;
    public GameObject PopupWin;

    public GameObject VirusNormal;
    public GameObject VirusInjured;
    public GameObject VirusDestroyed;
    public GameObject Voile;

    void Start()
    {
        bdd_quizz = GetComponent<bdd>();

        tirage();
    }

    public void tirage ()
    {
        canPlay = true;
        pvVirus--;
        txtPvVirus.text = pvVirus+"";
        string temp=bdd_quizz.ligneAleatoire();

        if(temp!=null && pvVirus!=0)
        {
            string[] tab_temp = temp.Split(',');

            txtquestion.text=tab_temp[0];
            reponse=tab_temp[1];
            melange(tab_temp);

            p1=tab_temp[1];
            p2=tab_temp[2];
            p3=tab_temp[3];

            //affichage des propositions dans les boutons
            GameObject.Find("B1").GetComponent<ButtonScript>().updateTxt(p1);
            GameObject.Find("B2").GetComponent<ButtonScript>().updateTxt(p2);
            GameObject.Find("B3").GetComponent<ButtonScript>().updateTxt(p3);
        }
    }
}
```

Fig 36. Extrait du code du Script Quizz

Pour créer de nouveaux scripts ou modifier ceux existants, Unity 3D intègre un compilateur libre appelé Monodevelop. Si l'utilisateur n'en dispose pas, il est possible d'ouvrir les scripts sur Microsoft Visual Studio. Une fois les modifications apportées, ces dernières sont automatiquement prises en compte dans Unity sans devoir importer à nouveau le fichier.

4.4. Synthèse

Pour conclure, ce chapitre a présenté la phase de création de notre prototype, tant sur le plan graphique que sur le plan technique. Les éléments graphiques conçus et exportés depuis Photoshop sont importés sur Unity et constituent les ressources nécessaires pour alimenter les différentes scènes pour construire un environnement interactive et ludique. Dans la section qui suit, nous allons présenter le produit de ce processus de création à travers un prototype opérationnel dont les écrans sont fonctionnels visuellement et techniquement.

Chapitre 5: Prototype du jeu *Globrain's Quest*

Avant de détailler le fonctionnement de notre jeu sérieux intitulé *Globrain's Quest*, il faut rappeler que la conception des scènes du jeu a été faite avec une réflexion centrée autour de l'état de l'art présenté auparavant. Les modules techniques et pédagogiques figurant dans le jeu sont sélectionnés à partir des différents éléments figurant dans l'état de l'art et qui s'adaptent le mieux à notre vision d'un jeu sérieux pour l'enseignement et l'apprentissage de la programmation informatique.

5.1. Description générale du jeu

Les scènes du jeu que nous allons présenter suivent une logique de présentation de l'information. En effet, pendant la conception du jeu, nous avons prévu une certaine quantité d'information que le joueur doit apprendre. Pour éviter ainsi de créer une confusion chez ce dernier, le contenu des scènes a été murement réfléchi, contextualisé et divisé de manière à le délivrer progressivement et simplifier son assimilation. C'est également dans ce sens que la trame scénaristique de notre jeu prend en considération un schéma de cours du langage Python pour réussir une association subtile entre le ludique et le pédagogique (Yassine, Berrada, Tahiri, & Chenouni, 2020).

Dans ce qui suit, nous décrivons les différentes scènes de *Globrain's Quest*. Nous nous limiterons dans cette description aux premières étapes du jeu qui suffisent à le décrire dans son entièreté.

5.2. Description des scènes

5.2.1. Scène #1 : texte introductif

A l'ouverture de l'application pour la première fois, se dévoile une scène contenant un court texte expliquant au joueur le contexte du jeu. Etant un jeu sérieux, il faut indiquer à l'apprenant la démarche pédagogique avant d'introduire les activités ludiques contenues dans l'application. Toutefois, le texte prévu à cet effet résume notre démarche dans sa globalité sans dévoiler de détails. En effet, le joueur aura toute la liberté de déduire, par son interaction, les différents mécanismes de l'application et l'approche pédagogique qui y figure.

L'importance cruciale de cette scène réside dans son rôle de rappel que l'expérience que l'apprenant est sur le point d'avoir n'est pas uniquement ludique, mais également à vo-

cation pédagogique dans le cadre de son apprentissage du langage de programmation Python.

Plus tard lors d'une seconde ouverture de l'application, cette scène ne se déclenchera plus car il ne sera plus nécessaire de la présenter au joueur, ce dernier ayant déjà pris conscience du message qu'elle délivre. La figure suivante illustre cette scène.

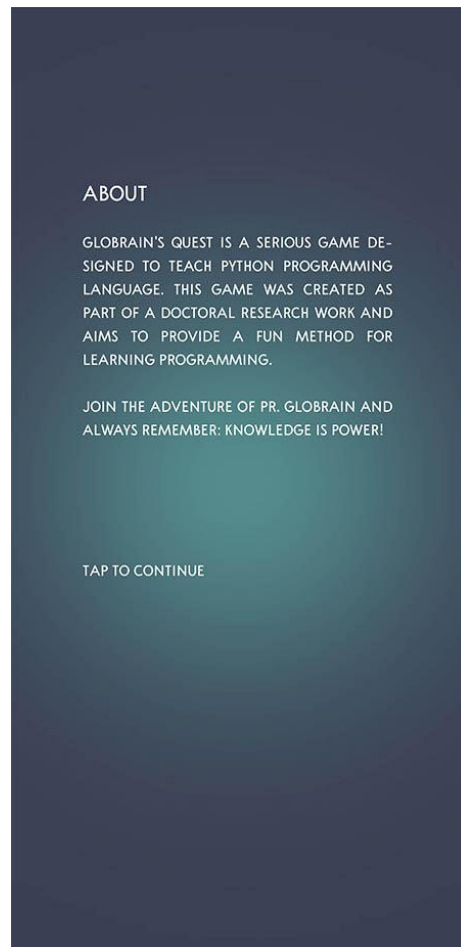


Fig 37. Ecran d'introduction, Globrain's Quest

5.2.2.Scène #2 : l'écran titre

L'écran titre, illustré dans la figure 30, représente la scène d'ouverture du jeu qui fait objet de premier contact à travers lequel le joueur se constitue des repères visuels. C'est une entrée en matière simple et colorée qui familiarise l'apprenant avec le style graphique de l'application. Elle comporte le logo du jeu ainsi qu'un des principaux personnages du jeu, le Pr.Globrain. Les éléments graphiques sont accompagnés du titre mis en évidence en haut de la scène. Le bouton « START » permet d'ouvrir la prochaine scène, celle contenant la première partie du scénario.

La figure suivante illustre l'écran titre.

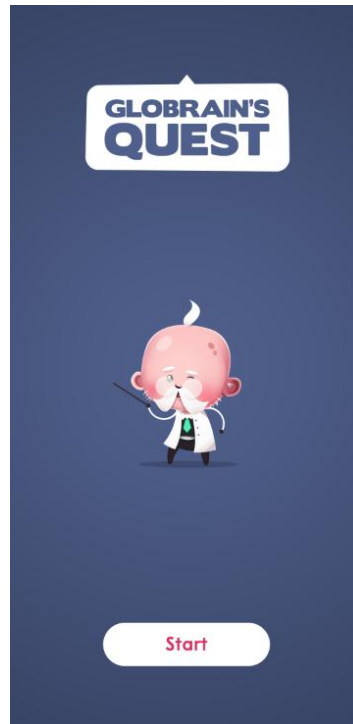


Fig 38. Ecran titre, Globrain's Quest

5.2.3.Scène #3: Première partie du scénario

La mise en place d'un scénario dans notre jeu sérieux représente l'étape fondatrice de notre vision de l'interaction que l'on propose au joueur. En effet, les niveaux du jeu s'enchaînent selon un scénario qui privilégie l'apprentissage pour créer une expérience à la fois ludique et éducative.

Avant de concevoir les personnages et les environnements du jeu, il a fallu se tenir à une règle primordiale : l'expérience de jeu ne doit pas empiéter sur l'apprentissage, ce dernier étant l'essence même de notre projet de recherche. Ainsi, nous suivons des axes directeurs qui définissent notre choix scénaristique :

- **Orienter l'attention du joueur vers le contenu à apprendre.** La concentration du joueur ne doit pas être entièrement absorbée par des éléments scénaristiques purement ludiques. Un jeu sérieux est une association équilibrée entre amusement et apprentissage. Le scénario doit donc être simple et orienté vers le savoir.
- **Allier théorie et pratique.** (Muratet et al., 2011) précise que la pratique de la programmation est un complément nécessaire à la théorie, sans elle l'apprentissage serait vain et sans intérêt ni motivation. Ainsi, il faut prévoir diverses activités dans l'application, dont celles d'acquisition du savoir et d'autres d'application de ce dernier tout en évaluant au fur et à

mesure la qualité de l'apprentissage du joueur. Cet aspect nous incite donc à puiser dans l'état de l'art technique et didactique détaillé auparavant pour exploiter autant de manières de jouer et de façons d'apprendre pour créer une expérience de jeu renouvelable et attrayante.

- **Inclure dans le scénario un but ultime qui constituerait la source de motivation du joueur.** Effectivement, un scénario sans réel finalité peut vite ennuyer le joueur qui n'y percevrait pas le sens de ces actions. Nous prévoyons donc de doter l'application d'un objectif global qui donne un sens à toutes les actions entreprises par l'apprenant durant son expérience

ludique.

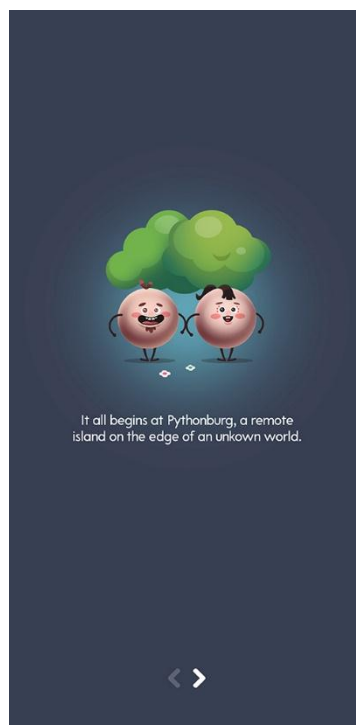


Fig 39. Scénario partie 1, Globrain's Quest

Ainsi, et pour concrétiser les idées citées auparavant

ravant, Le scénario suivant a été imaginé et son résumé est comme suit :

Dans une planète lointaine vivent des êtres imaginaires nommés les LOREM. Parmi eux se trouve un individu particulièrement intelligent qui s'appelle *Pr.Globrain*. Ce dernier possède une grande maîtrise du langage Python. Grâce à son savoir, il réalise des merveilles technologiques et son travail est admiré de tous.

A un moment précis, le *Pr.Globrain* décide de prendre un apprenti sous son aile pour lui transmettre son savoir et en assurer la pérennité. Son choix

s'arrête sur un jeune prodige du nom de LOREM. Mais au fil de son apprentissage, LOREM dévoile une personnalité sombre et avide de pouvoir. En secret, il construit une machine qu'il activera pour libérer une armée de puces téléguidés qui transformeront les cibles en des créatures malveillantes.

Le Pr.Globrain, incapable de braver les dangers pour décoder les puces lui-même, va devoir recruter le capitaine Globrave qui va devoir apprendre le langage Python pour libérer les nombreuses victimes du contrôle de LOREM.

Le scénario ainsi expliqué met au cœur de son intrigue le langage Python car le but est de garder l'attention du joueur fixée sur son objectif d'apprendre pour réussir sa mission. De nombreux personnages entrent en action et différentes épreuves sont prévues pour varier l'expérience de jeu.

Ci-dessous des illustrations du personnage principal et de certains antagonistes du jeu :



Fig 40. Antagonistes figurant dans Globrain's Quest

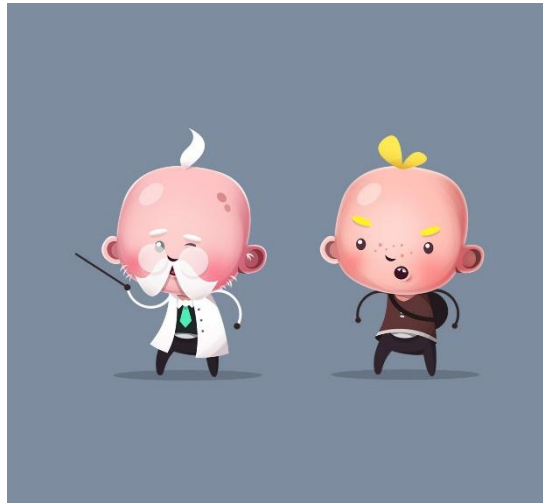


Fig 41.Pr.Globrain (à gauche) et le Capitaine Globrave (à droite)

Le joueur incarne un personnage tout au long de l'aventure –personnage de droite dans la figure 33, et dont la mission est de sauver son espèce de la destruction. Que ce soit du côté du protagoniste ou des antagonistes, chaque personnage est conçu avec un visuel différent car l'appréciation du jeu passe également par la variété esthétique et artistique qu'il présente. La différence entre les personnages n'est pas uniquement visuelle, mais aussi d'un point de vue des objectifs. En effet, deux antagonistes ne vous poseront pas les mêmes obstacles et leurs défis seront différents. Dans ce qui suit, nous allons préciser comment ce scénario a été adapté à un cours sur le langage Python.

5.2.4.Scène #3: Menu des chapitres

Une fois que le joueur a pris connaissance de la première partie du scénario du jeu, le menu des chapitres se dévoile comme une continuité du déroulement des événements, car le premier chapitre implique le héros dans sa quête du décodage des premières puces malveillantes.

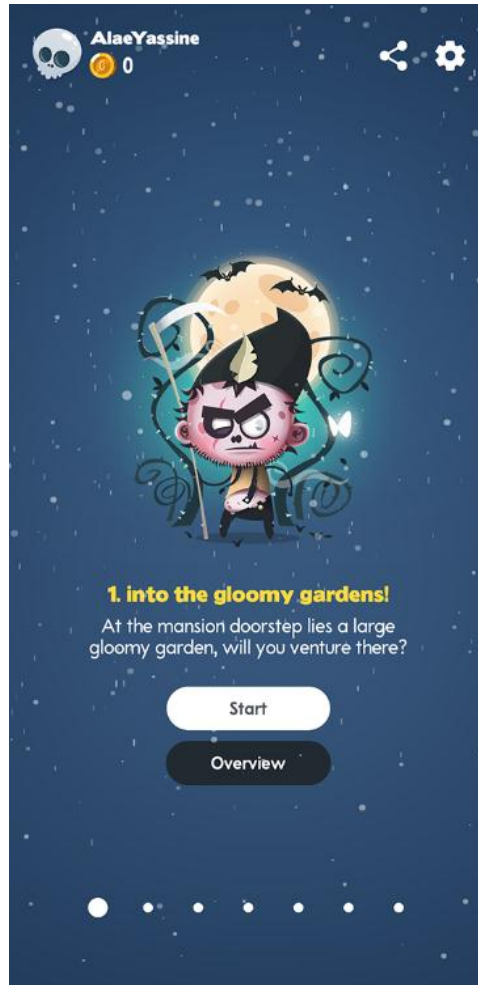


Fig 42. Menu des chapitres, Globrain's Quest

Illustrée sur la figure (Fig 42), cette scène représente l'axe central à partir duquel toutes les épreuves et les enseignements du jeu sont accessibles. Cette entrée en matière donne au joueur un aperçu du contenu pédagogique, mais également du contenu ludique. En effet, désireux d'associer un scénario ludique à un cours structuré, la succession des chapitres suit à la fois l'évolution de la trame scénaristique imaginée pour le jeu ainsi que la structure du cours concernant le langage Python. L'histoire débute donc à partir du premier chapitre et se termine logiquement avec la fin du dernier. Par conséquent, les héros, les ennemis et les épreuves varieront au cours du jeu. L'importance de cette scène réside dans le fait qu'elle joue un rôle de repère qui permet à l'apprenant d'accéder au contenu qu'il veut consulter. Elle représente également l'évolution des enseignements et des exercices par lesquels doit passer le joueur.

Tous les éléments du menu principal sont pensés et assemblés de manière à présenter au joueur un contenu facilement compréhensible et qui lie efficacement l'enseignement et

l'amusement. Pour ce qui est du droit d'accès, l'apprenant ne peut accéder au second chapitre qu'après avoir réussi les épreuves du premier et ainsi de suite. De cette manière, l'enseignement transmis suit des règles pédagogiques strictes. Par exemple, l'apprenant ne pourra apprendre la notion de chaîne de caractère qu'après avoir validé sa maîtrise des opérations dans le langage Python.

En plus du menu des chapitres, un ensemble d'éléments composent la scène principale. En haut de la scène se trouve le panel dédié au joueur et qui est composé de trois éléments : le pseudonyme du joueur qu'il entre dans une zone de texte dédiée à cet effet, l'avatar du joueur qu'il choisit parmi une liste dont les images sont à débloquent dans le jeu, et un compteur de pièces que le joueur peut gagner au fil des niveaux et qui permettent l'acquisition de certains éléments spéciaux comme les différents avatars. Le système de sauvegarde dans l'application est lié au profil du joueur. Ce dernier est libre de modifier ces informations sans risque d'altérer sa progression, l'image de l'avatar est également un bouton permettant d'accéder à un menu spécial prévu à cet effet.

Figure également dans cette scène un ensemble de boutons qui enrichissent l'interface utilisateur de fonctionnalités diverses que nous décrivons dans ce qui suit:

Le bouton Overview cité dans le tableau est essentiel dans la mesure où il donne un aperçu sur la liste des enseignements que l'apprenant va recevoir. Procéder aux niveaux du chapitre sans avoir au préalable une idée claire sur le cheminement à suivre constitue une lacune dans le schéma de transmission de l'information mis en place dans l'application.

La figure(Fig 43) illustre la fenêtre pop-up prévu dans ce sens et qui contient dans sa partie supérieure une illustration représentant le héros du jeu face à son ennemi principal du premier chapitre. Cette illustration vient ajouter une touche graphique pour éviter d'afficher uniquement un fond blanc avec du texte, et maintient également l'immersion dans le scénario du jeu.

Le texte figurant dans la fenêtre marque à la fois l'objectif scénaristique à atteindre et le détail du cours du chapitre, joignant ainsi savoir et divertissement.

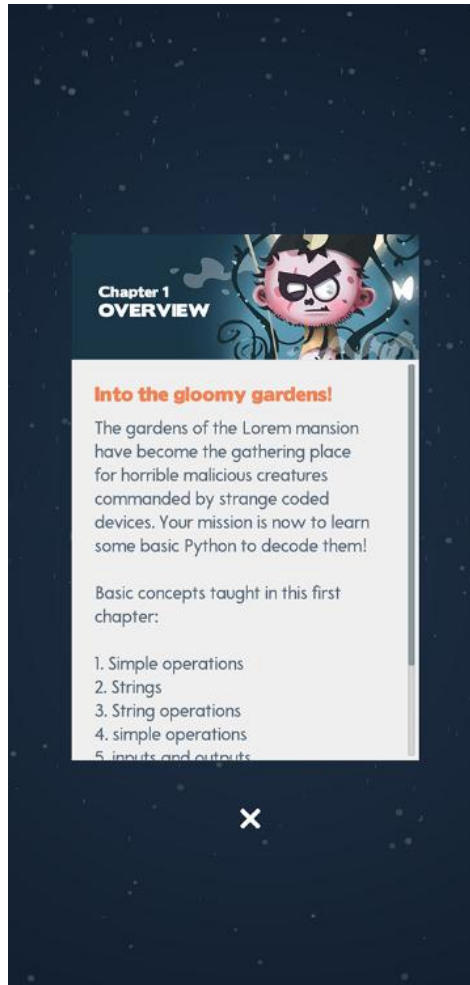


Fig 43. Fenêtre popup du premier chapitre, Globrain's Quest

En conclusion, la scène principale est un lien important entre les vocations ludique et pédagogique prévues dans ce projet. Elle relie les personnages, les événements et les activités et oriente le joueur vers son but. Dans ce qui suit, nous allons exposer les détails concernant le premier chapitre.

5.3. Description du premier chapitre

5.3.1. Menu principal du premier chapitre

Le premier chapitre du jeu est dédié entièrement aux notions de base du langage Python. Les activités comprises dans ce chapitre s'articulent autour d'un objectif précis : aider l'apprenant à comprendre les éléments de base du langage Python. C'est à ce niveau que débute la quête du joueur.

Il s'agit dans un premier temps de résoudre le code crypté de la première machine. Pour y parvenir, l'apprenant doit alterner des cours et des épreuves pour respectivement ap-

prendre et confirmer sa maîtrise du sujet. Dans ce qui suit, nous allons décrire ce premier chapitre et expliquer l'approche employée pour apprendre les notions de base de Python et évaluer la compréhension de l'apprenant et sa capacité à employer le savoir appris.



Fig 44.Chapitre 1, Globrain's Quest

La figure 36 illustre l'affichage des niveaux appartenant à un chapitre. Dans la figure, le premier niveau apparaît au centre de la scène et plusieurs éléments le décrivent. Premièrement, une illustration représentant le plus souvent le personnage principal et le lieu où se déroule l'action en termes de scénario. De cette manière, le joueur suit le scénario du jeu à la façon d'une bande dessinée. Ensuite, le titre et une description brève ajoutent de la matière pour communiquer au joueur le but du niveau, qu'il s'agisse d'un cours ou d'une épreuve.

La succession des cours et des évaluations suit une logique d'évaluation formative. En effet, progressant à travers les niveaux, le joueur accumule progressivement du

savoir. Toutefois, le contenu du première cours est requis pour la compréhension du contenu du deuxième. Ainsi, et pour s'assurer de la bonne maîtrise du savoir reçu, une évaluation vient s'ajouter entre deux leçons, évaluant ainsi les connaissances acquises lors de la précédente. La réussite de l'évaluation est nécessaire pour ouvrir l'accès aux prochains cours.

Cette évaluation doit ainsi trouver sa place dans le schéma scénaristique construit pour le jeu. Il est donc naturel d'associer les épreuves à des entités ennemies, sous la forme de défis de réflexion ou autres activités liées au codage en Python. A ce stade de la description de l'application, il est judicieux d'abord de se focaliser sur les mécanismes du cours et sur la manière scientifique d'allier ludisme et pédagogie pour apprendre Python sans risquer de briser l'immersion dans l'atmosphère de l'application. Le point de départ de cette description est l'évocation d'un aspect qui régit la succession de tous les enseignements du jeu : la taxonomie de Solo.

5.3.2. GLOBRAIN'S QUEST ET LA TAXONOMIE DE SOLO

Evoquée dans la section IV, la taxonomie de Solo répartie le savoir en unités structurales pour aider l'apprenant à bien entamer son apprentissage et le poursuivre de manière graduelle et contextualisée. Ainsi, et en prenant comme exemple le premier chapitre du jeu concernant les bases de Python, comment cette taxonomie subdivise le cours?

Avant de détailler l'adaptation de la taxonomie, il faut savoir que ce chapitre présente un certain nombre de notions de base, nécessaires à la maîtrise de la syntaxe du langage Python. L'apprenant est tenu de comprendre la nature et la fonction de ces notions, et est amené à les combiner entre elles pour pouvoir écrire des programmes complets et fonctionnels.

Il s'agit donc de déterminer comment la taxonomie de Solo facilite l'assimilation de tous ces éléments, ces derniers se présentant dans l'ordre qui suit :

1. Les opérations arithmétiques
2. Les chaînes de caractères
3. Les opérations sur les chaînes de caractères
4. Les entrées et les sorties

5. La conversion de type
6. Les variables

Cet enchaînement d'éléments de cours est fait à travers l'application de la taxonomie de Solo et le tableau suivant décrit cette logique que devra adopter l'apprenant pour sa formation ludique :

Tableau 9. Structuration du cours selon la taxonomie de Solo

Intitulé du cours	Degré d'acquisition du savoir	Description
Cours 1 : les opérations arithmétiques	Uni-structurel	Au commencement du chapitre, l'apprenant étudie le premier aspect de Python, les différents types d'opérations.
Cours 2 : les chaînes de caractères	De Uni-structurel à Multi-structurel	L'apprenant assimile la syntaxe relative aux chaînes de caractères et se prépare au cours suivant, les opérations sur les chaînes de caractères. A ce stade, l'apprenant maîtrise deux notions de Python, d'où le passage au degré multi-structurel d'acquisition du savoir.
Cours 3 : Les opérations sur les chaînes de caractères	Multi-structurel et Relationnel	L'apprenant apprend comment appliquer des opérateurs sur des chaînes de caractère, ainsi le savoir acquis lors des deux derniers cours est investi dans la compréhension de celui de ce troisième cours.
Cours 4 : Les entrées et les sorties	Multi-structurel et Relationnel	A ce niveau, l'apprenant opère sur des fonctions de sortie pour afficher des résultats, ces fonctions prennent en paramètres des valeurs numériques, des opérateurs, mais également des chaînes de caractères que l'apprenant maîtrise déjà.

Cours 5 : La conversion de type	Multi-structurel et Relationnel	La conversion de type
Cours 6 : Les variables	Multi-structurel et Relationnel	L'apprenant achève son apprentissage des bases en manipulant des variables, capables de stocker des valeurs. Une fois une valeur stockée (peut être sous forme numérique ou textuelle), l'apprenant peut afficher sa valeur en utilisant les fonctions de sortie déjà apprises et maîtrisées. L'apprenant peut également effectuer des opérations sur les variables

Il est évident que le degré 'Abstrait étendu' ne figure pas sur le tableau. Il s'agit d'un degré de maîtrise qui considère que l'apprenant est capable de mettre en application son savoir. Or, le tableau ne détaille que des phases d'assimilation des connaissances. Par conséquent, Nous avons conçu une phase de conclusion de chaque cours, le test du Professeur Globrain. En effet, il s'agit à la fin de chaque cours de répondre à une série de question de la part du professeur pour valider l'acquisition des savoirs. C'est pourquoi, dans ce qui suit, nous allons décrire cet aspect ludique du cours, en relation avec les deux taxonomies, celle de Benjamin et Bloom et celle de Solo.

5.3.3. *Globrain's quest* et la taxonomie de Bloom, exemple du premier cours sur les opérations en Python

Au fil de la progression du joueur, des obstacles se dressent et augmentent graduellement la difficulté du jeu. Pour réussir, le Pr. Globrain a préparé une série de cours que l'apprenant doit comprendre et assimiler et qui suivent, comme cité auparavant, la taxonomie de Solo. Cette dernière régit la succession de la totalité des cours de l'application. Mais qu'en est-il de la structure de chaque cours ?

Visuellement, la scène du cours illustrée dans la figure qui suit (Fig 45), est celle qui se rapproche le plus du style d'une bande dessinée. En effet, le cours est expliqué à la façon d'un monologue du professeur. Un monologue adressé au personnage principal, incarné par l'apprenant.

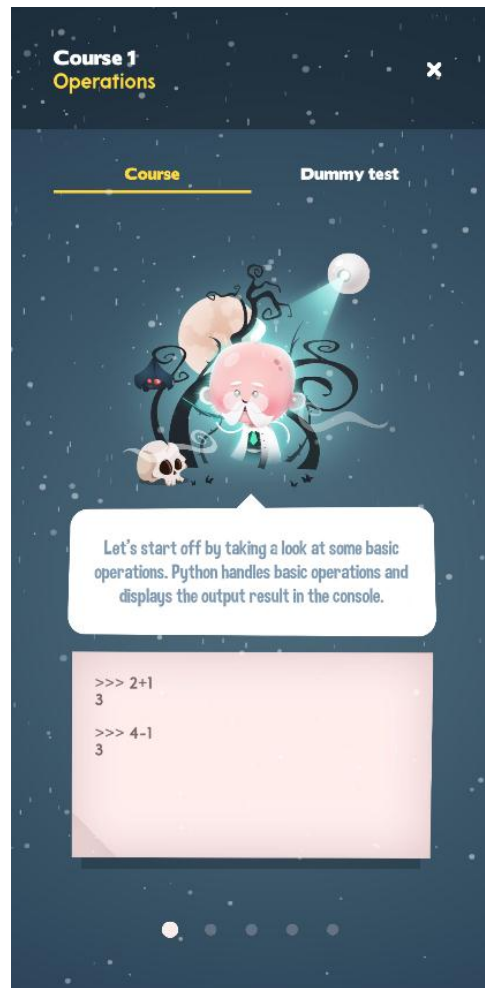


Fig 45. Niveau 1, les opérations dans Python, Globrain's Quest

Il faut se rappeler que d'un point de vue scénaristique, le Pr.Globrain est dans son laboratoire et n'accompagne jamais physiquement le héros dans son aventure. Toutefois, Ordy, le petit globe blanc figurant dans la même illustration projette une image holographique du professeur qui expose son cours. Le monologue du Pr.Globrain est divisé en plusieurs parties, chacune expliquant les morceaux de code affichés en dessous et sont accessibles à travers une série de points en bas de la scène, facilitant ainsi la navigation entre les différentes explications du professeur.

En somme, des syntaxes simples, des explications concises, des personnages animés et des ambiances sonores constituent l'aspect global d'un cours dans *Globrain's Quest*. Tous les cours du jeu suivent le même schéma mais augmentent naturellement en complexité, au fur et à mesure de la progression dans le cours.

Toutefois, tous ces éléments sont agencés de façon à appliquer les principes de la taxonomie de Bloom. Evoquée en introduction de cette section, la réalisation de l'application a été faite en considérant l'état de l'art technique et pédagogique des sections III et IV, c'est notamment le cas pour le cours dispensé dans *Globrain's Quest*. En effet, chaque élément de cours est en adéquation avec un aspect hiérarchique de la taxonomie de Bloom :

- **Les morceaux de code pour la connaissance :**

Les lignes de code servent principalement à présenter la syntaxe d'un langage dont l'apprenant ne connaît pas forcément la structure ni les composants. C'est une première étape de familiarisation avec le sujet à travers ses grandes lignes sans forcément en comprendre le sens ou l'objectif. Suivant l'exemple de la leçon concernant les opérations arithmétiques, l'apprenant est donc à cette phase de connaissance où il commence à définir et identifier les différents opérateurs, lister les types d'opérations et mémoriser les résultats.

- **Le monologue du professeur pour la compréhension :**

Chaque morceau de code est accompagné d'une explication visant à décrire sa syntaxe et ses règles. Ce module vient compléter l'étape de connaissance en dotant l'apprenant d'un outil lui permettant de comprendre la nature et l'utilité de la syntaxe qui lui est présentée.

- **Le compilateur pour l'application :**

Pour avoir un cours complet dans l'application, la pratique doit nécessairement accompagner la théorie. En effet, avoir la possibilité de manipuler du code dans le cours permet au joueur de mieux cerner l'architecture de Python afin de mieux le comprendre. La figure suivante illustre le compilateur où l'apprenant peut modifier le code et générer autant de résultats que voulu. Le principe de bac à sable évoqué par James Paul Gee (Gee & Morgridge, s. d.) est illustré dans cet aspect du jeu qui permet une sorte d'expérimentation de la part de l'apprenant, ce dernier étant libre de manipuler le code à souhait. Dans cet exemple, le joueur peut modifier un code permettant de faire une opération d'addition, il peut modifier l'opérateur et les valeurs.

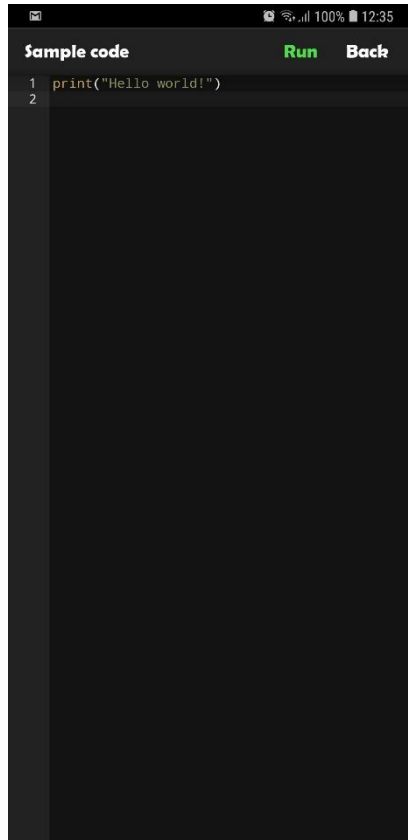


Fig 46. Compilateur Python, Globrain's Quest

A ce stade de l'apprentissage, la compréhension du joueur se reflète à travers sa capacité à opérer directement sur le code. Il peut ainsi en écrivant son propre code en Python, utiliser les différents opérateurs, modifier les paramètres et interpréter les résultats obtenus de son exercice. L'apprenant devient acteur de son apprentissage et participe à sa formation en s'exerçant sur la syntaxe.

THE DUMMY TEST pour l'application:

L'onglet « Dummy test » est la seconde activité contenue dans ce premier cours. Il s'agit en réalité d'un questionnaire visant à évaluer la compréhension de l'apprenant du cours tout juste appris. Cette forme d'évaluation formative doit toutefois trouver sa place dans le schéma scénaristique du jeu pour une nouvelle fois éviter de briser l'immersion et l'aspect ludique. Pour réussir l'intégration de cet exercice, le scénario admet que le Pr.Globrain vérifie la compréhension du héros en préparant une marionnette à l'image des ennemis du jeu. Cette marionnette est l'ennemi principal de chaque questionnaire dans le jeu. En effet, comme le montre la figure (Fig 47), le premier

questionnaire possède des règles mises en évidence sous le titre et qui affichent au joueur le nombre de bonnes réponses à faire, le nombre de mauvaises réponses autorisé et le temps imparti pour chaque question. Dans ce cas, il faut réussir quatre bonnes réponses successives, sans aucune restriction de temps.

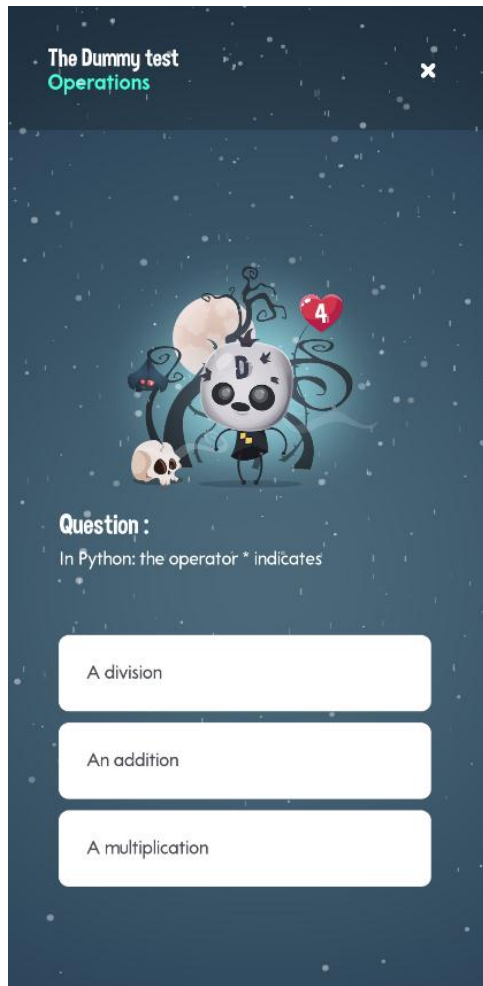


Fig 47. Dummy test 1, Globrain's Quest

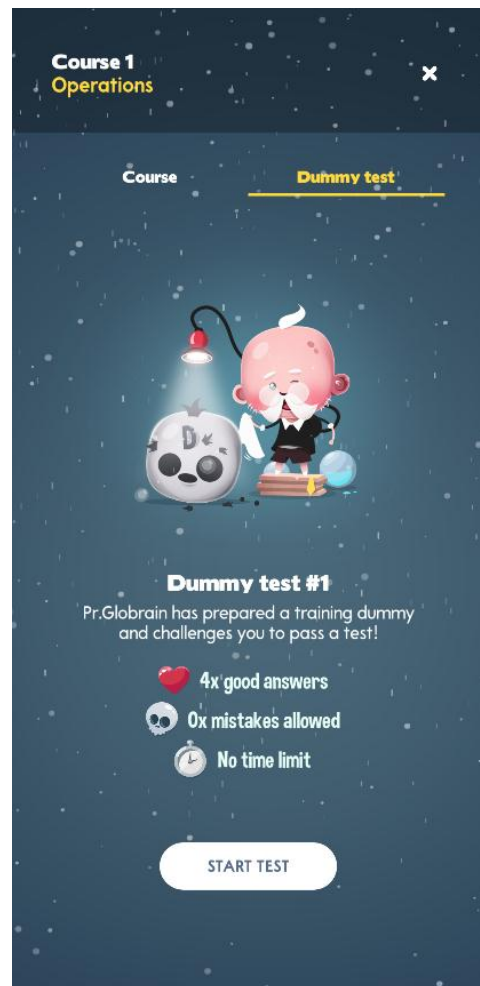


Fig 48. Dumy test 1, Globrain's Quest

Visuellement (Fig 47), cette scène est semblable aux précédentes dans la mesure où elle présente les mêmes éléments du visuel du premier niveau. Techniquement par contre, les choix du joueur dans ce questionnaire ont un effet immédiat sur la marionnette. En effet, une bonne réponse déduit un point de vie au nombre total de cœurs, qui s'élève à quatre dans cet exemple. Ainsi, quatre bonnes réponses sont requises pour réussir cette première épreuve. Les questions s'affichent sous forme de choix multiples où une seule réponse est correcte parmi les trois choix affichés. Les réponses justes doivent être successives car le héros est tenu de ne commettre aucune erreur sous peine d'échouer au test. Ce premier test, relativement simple, fait figure d'entrée en matière aux différents formats d'évaluations intégrés au sein de l'application.

Ce mécanisme classique prend un aspect ludique avec l'introduction de la marionnette et le fait qu'elle soit un élément d'entraînement pour le joueur. La véritable première épreuve ne se présentant que lors du deuxième niveau, et comprend des mécanismes de jeu plus évolués. D'un point de vue pédagogique, ce questionnaire représente une situation problématique où l'application du savoir acquis permet de résoudre le problème. Il s'agit du niveau de maîtrise correspondant à « l'application » dans la taxonomie de Bloom.

Dans le cas où l'apprenant réussit le questionnaire, une fenêtre s'affiche contenant un certain nombre d'informations (figure 41) :

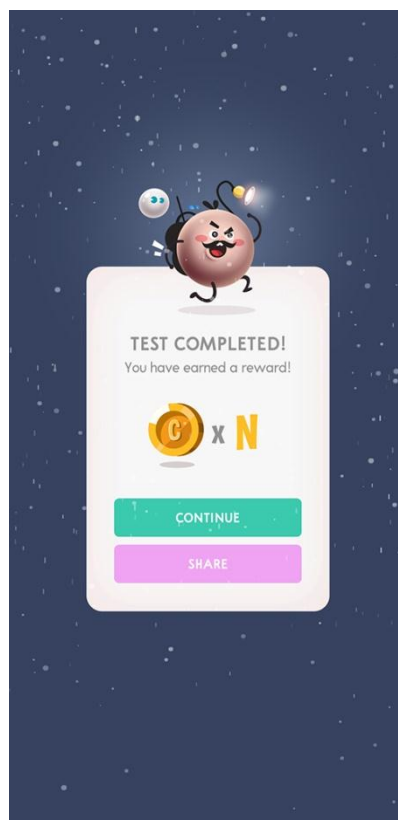


Fig 49. Dummy test 1 – popup victoire, Globrain's Quest

Une illustration du personnage principal célébrant sa victoire, accompagné d'Ordy (le personnage blanc au dessus du héros). De plus, cette fenêtre récompense le joueur avec des pièces d'or que ce dernier peut dépenser dans l'application pour acquérir certains objets. Ce système de récompenses n'est encore pas intégré à l'application, mais figure parmi les modules à développer dans une phase ultérieure car représentant un aspect de motivation qui encourage le joueur à mieux apprendre pour réussir les épreuves qui l'attendent tout au long de son aventure.

Au cas où le joueur dépasse la limite d'erreurs autorisées, le test est considéré comme étant un échec et la fenêtre suivante s'affiche (fig num) :

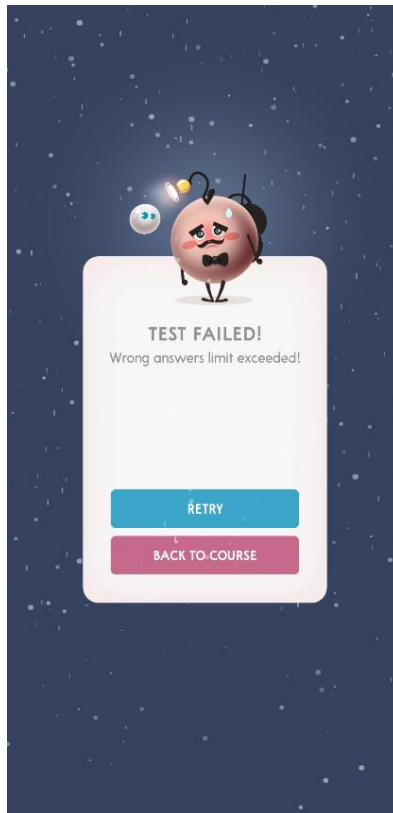


Fig 50. – Dummy test 1 – popup défaite, The Stronglobes

Cette figure représente une illustration du *Capitaine Globrave*, le représentant insatisfait de son échec. Aucune récompense n'est octroyé dans ce cas et le joueur se voit proposer deux options : repasser le test ou revenir au cours pour réviser d'avantage. Selon James Paul Gee (Gee & Paul, 2003), commettre des erreurs est un aspect positif dans l'apprentissage. A travers le test de la marionnette, échouer à une question permet au joueur d'orienter son attention vers le point où l'erreur a potentiellement été commise. En somme, ce test est un exercice reposant sur des mécanismes techniques très simples mais rendu intéressant en l'incluant dans le scénario de façon subtile dans un but purement pédagogique : rendre l'apprentissage plus dynamique et plus engageant.

Toutefois, ce questionnaire n'est qu'une entrée en matière ludique vers la variété d'épreuves prévues dans le jeu. Le joueur sera confronté à la première épreuve dans le niveau 2 du chapitre 1.

CHAPITRE 1 NIVEAU 2 :

Le deuxième niveau du premier chapitre constitue la première épreuve du jeu. Elle oppose le héros à un LOREM contrôlé par une puce protégée par un code brouillé. L'objectif scénaristique est de décoder la puce pour libérer le LOREM. Comme illustré sur la figure 43, la présentation du deuxième niveau suit les mêmes codes visuels que le premier, à savoir les titres, le bouton d'accès au niveau et l'illustration représentant des personnages et quelques éléments de décor.

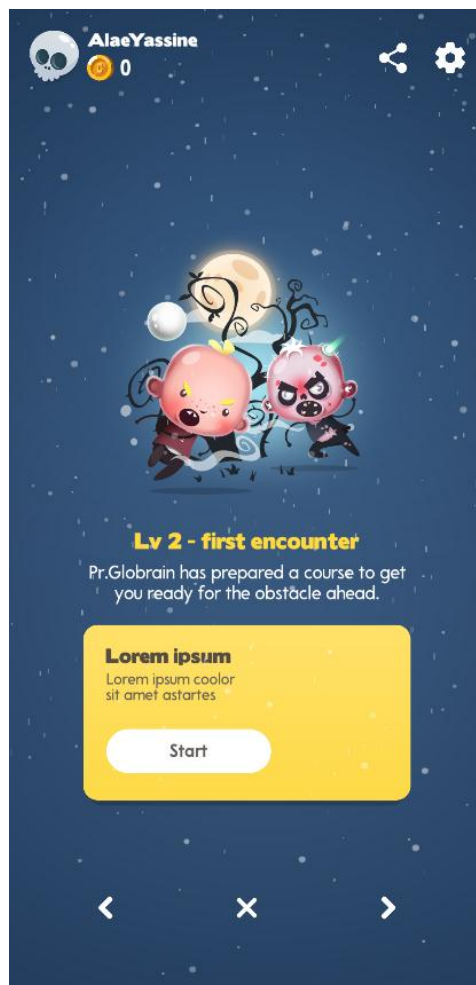


Fig 51. – niveau 2 chapitre 1, Globrain's quest

Une fois dans la scène de l'épreuve, une fenêtre explique l'objectif du niveau. Dans ce cas, l'apprenant doit compléter une série de lignes de code pour décoder la puce. Cette série contient du code en Python de certaines opérations incomplètes. Les éléments manquants varient entre opérateurs, valeurs ou résultats. De plus, la dernière ligne de code à compléter est construite à la façon d'une énigme où du calcul mental est requis.

À l'instar de l'épreuve du questionnaire où l'apprenant choisit la bonne réponse à une question directe, l'épreuve du deuxième niveau propose au joueur une palette de cases blanches contenant chacune une proposition pour compléter une ligne de code. Sur la figure (Fig 52), le code incomplet au centre contient des creux que le joueur est supposé compléter à partir des choix qui lui sont fait en dessous.

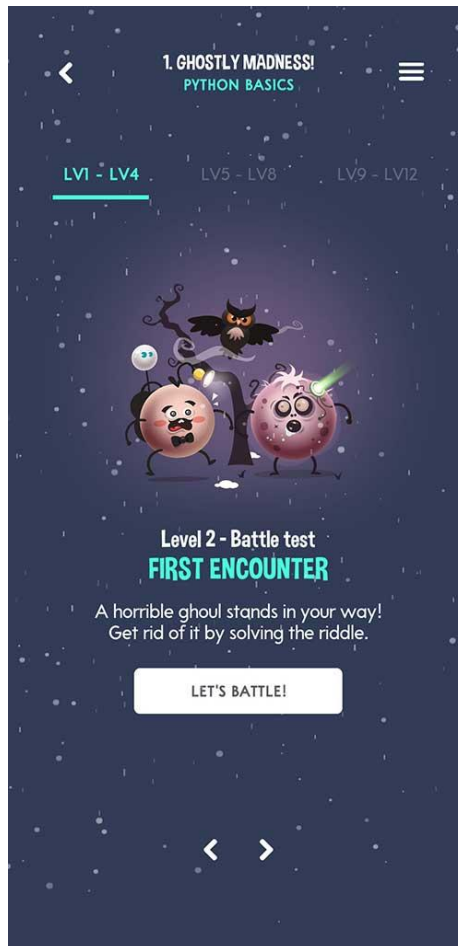


Fig 52. – niveau 2 chapitre 1, Globrain's quest

La série de points au dessus du code représente les trois étapes de l'épreuve, l'objectif étant de réussir les trois pour décoder la puce. L'interaction sous forme de drag & drop représente une évolution par rapport aux réponses directes face aux questions de compréhension de l'épreuve précédente. Cette évolution reflète la progression de l'apprenant dans son apprentissage car ce dernier entreprend dans ce nouvel exercice une activité d'observation d'un code incomplet, d'analyse et de compréhension de la structure du code et une déduction des éléments manquants. Son choix de réponse émane ainsi d'un travail de réflexion sur la syntaxe de Python.

5.4. Synthèse

En conclusion de cette section décrivant le prototype *Globrain's Quest*, il est important de noter que les différents éléments qui le composent à savoir le cours, le compilateur, le questionnaire et l'épreuve en Drag&drop, sont des modules basiques mais nécessaires dans la première phase d'apprentissage du langage Python. Cette entrée en matière comporte donc une quantité suffisante d'éléments pour diversifier l'expérience de jeu et introduire l'apprenant de manière ludique à la programmation. Toutefois, un ensemble de modules est prévu pour faire évoluer l'application :

- Le système multi-agent évoqué précédemment constitue l'étape suivante de l'évolution du prototype vers une première version complète. A travers ce système, l'application joue le rôle d'un tuteur prenant en considération un ensemble d'éléments mesurables et liés directement à l'apprenant. De cette façon, l'expérience de jeu peut être personnalisée en fonction des capacités de chaque joueur.
- L'intégration de nouvelles approches d'interaction car la progression du joueur dans son apprentissage lui permet d'aborder des épreuves aux possibilités d'interaction plus évoluées et aux requis plus complexes. Procéder à travers des techniques de jeu plus interactives est à la fois une manière de créer une richesse ludique dans l'application et une confirmation de l'évolution des aptitudes de l'apprenant. Ce dernier pouvant écrire du code, analyser des syntaxes complexes et produire des résultats de plus en plus concrets.
- Un avantage encore plus motivant est celui de l'aspect multijoueurs qui permet aux apprenants d'échanger des informations, de collaborer pour résoudre différents problèmes ludiques et comparer leurs impressions.

Chapitre 6: Vers un jeu multi-joueurs grâce au SMA

Un système multi-agent (SMA) est important dans notre contexte de recherche. Présenter un jeu statique à l'apprenant ne serait pas le bon choix puisque les profils et les rythmes d'apprentissage des joueurs diffèrent. Tandis que certains joueurs progressent, d'autres pourraient avoir du mal à terminer un niveau et auraient besoin de plus d'attention. L'intégration d'un système multi-agent, dans ce contexte, fournit aux joueurs des instructions automatisées basées sur leurs performances dans le jeu. Avant de présenter notre vision d'un système multi-agents. Il convient de définir l'approche multi-agent et son utilisation dans le contexte de jeux sérieux.

6.1. L'architecture multi-agent dans notre contexte

Un système multi-agent est important dans notre contexte de recherche. Présenter un jeu statique à l'apprenant ne serait pas le bon choix puisque les profils et les rythmes d'apprentissage des joueurs diffèrent. Tandis que certains joueurs progressent, d'autres pourraient avoir du mal à terminer un niveau et auraient besoin de plus d'attention. L'intégration d'un système multi-agent, dans ce contexte, fournit aux joueurs des instructions automatisées basées sur leurs performances dans le jeu. Avant de présenter notre vision d'un système multi-agents, il convient de définir l'approche multi-agent et son utilisation dans le contexte des jeux sérieux.

Un système multi-agents est un système composé d'un ensemble d'agents dits autonomes, ces derniers sont en interaction en vue d'accomplir une tâche ou résoudre un problème. L'aspect autonomie est très important et exprime la capacité des agents à contrôler leurs états et leurs actions sans interventions extérieures. Les agents évoluent et interagissent dans un environnement partagé. Selon (Ferber 1997), un agent est une entité capable d'agir dans un environnement et communiquer avec les autres agents. Chaque agent possède ses propres ressources et a une vision de l'environnement dans lequel il évolue. Cela, n'exclut pas le cas où il peut éventuellement n'avoir aucune vision de son environnement. Chaque agent a des objectifs à atteindre, en fonction des ressources et des compétences dont il dispose, sa perception de l'environnement et des interactions avec les autres agents (Ferber 1999). Les interactions des agents leur permettent de participer à la réalisation d'un objectif global. Une autre définition a été proposée par (Demazeau et Rocha Costa 1996) qui stipulent qu'un agent est une entité ré-

elle ou virtuelle au comportement autonome, pouvant évoluer dans un environnement dans lequel elle est capable d'agir et d'interagir avec les autres entités.

Une des spécificités des agents est leur capacité à communiquer dans un environnement donné. Il s'agit de l'envoi d'un message à un ou plusieurs récepteurs qui vont à leur tour interpréter le contenu selon les communications menées auparavant.

Les SMA sont utilisés dans plusieurs contextes: la résolution des problèmes en utilisant l'intelligence artificielle dans des environnements distribués mais également utilisés en Génie Logiciel pour créer des composants logiciels de plus en plus autonomes. Selon (Morvan 2009), un SMA est dit multi-agents si :

- « Un groupe (E) d'entités placées dans un environnement (V).
- Un groupe (A) d'agents avec $(A) \subseteq (E)$.
- Un système d'action sous forme de percepteurs et d'effecteurs qui permettent aux agents d'intervenir et d'agir dans leur environnement.
- Un système d'interaction et de communication entre agents.
- Et enfin, une organisation qui permet de structurer l'ensemble des agents. »

Dans la littérature, les agents se déclinent en deux grandes catégories: les agents dits cognitifs et les agents dits réactifs. Cette distinction est liée au degré de perception de l'agent de son environnement. Un agent est réactif s'il ne dispose que d'une présentation limitée de son environnement en fonction de ses perceptions. Dans ce cas, l'agent réagit au stimulus (appelé perception) en réalisant des actions appropriées. Par contre, un agent est dit cognitif s'il possède une représentation de son environnement, et a la capacité de raisonner, planifier et réaliser des actions en interactions avec les autres agents et l'environnement.

Selon (Mostafa et al. 2017), un agent dit :

- Autonome : un agent qui fonctionne sans l'intervention directe de l'homme ou d'autres agents et dispose d'une certaine forme de contrôle sur ses actions et ses états.
- Situé: un agent qui fait partie d'un environnement, observe son entourage et agit en fonction des entrées qu'il reçoit.
- Intéreactif: un agent qui interagit avec d'autres agents via un langage de communication commun.
- Réactif: un agent qui perçoit une partie de son environnement et réagit en temps utile aux changements qui se produisent.

- Coordination: un agent qui configure ses actions en fonction des exigences de son environnement afin d'atteindre son objectif.
- Reproduction: un agent qui est capable de se reproduire et de s'adapter aux changements.

Nous avons eu recours à l'approche SMA dans la perspective de décomposer notre application en plusieurs composantes autonomes. Chaque composante appelée agent doit réaliser des objectifs de façon autonome tout en interagissant avec les autres éléments de l'environnement. Notre jeu sérieux repose sur un système multi-agents. Le système comprend divers agents intelligents interagissant pour fournir aux élèves le meilleur scénario d'apprentissage. Ces agents imitent les décisions du tuteur humain en analysant le progrès des joueurs et en leur fournissant les meilleures instructions pour progresser. Le but est de rendre le jeu flexible et dynamique, permettant au joueur d'être guidé tout au long du processus d'apprentissage en fonction des décisions prises par le système. Le système multi-agent comprend 4 agents: "Agent de comportement", " Agent de Diagnostic de la Performance "; " Agent de conception du système pédagogique" et "agent d'assistance". Chaque agent effectue une tâche spécifique, partage des informations et communique avec d'autres agents. Le fonctionnement de chaque agent sera détaillé dans la section qui suit :

- **Agent de comportement:** cet agent recueille des données sur le comportement des joueurs dans le jeu. Par exemple, l'agent recueille des informations sur le temps écoulé, l'interaction du joueur avec différents objets des scènes et les étapes poursuivies par l'apprenant (actions) pour résoudre un exercice.
- **Agent de diagnostic de la performance:** cet agent est chargé de comparer l'état actuel de la maîtrise de l'étudiant avec celui qui est désiré en utilisant les données fournies par l'agent de comportement.
- **Agent de conception du système pédagogique:** cet agent joue le rôle d'un tuteur humain en prenant des décisions sur le contenu qui devrait être présenté à l'apprenant selon les résultats de l'agent de diagnostic de la performance. Le jeu est divisé en plusieurs scènes; chaque scène contient de nombreux objets avec de nombreux objectifs d'apprentissage. L'agent de conception pédagogique décide quel environnement sera présenté avec des objectifs d'apprentissage spécifiques et des tâches précises.

- **Agent d'assistance:** Cet agent est activé par le joueur lui-même ou par l'agent de diagnostic de performance pour fournir des commentaires aux étudiants. D'une part, l'aide est activée si le joueur demande explicitement de l'aide. L'agent d'assistance envoie à la fois une notification au tuteur et à d'autres apprenants qui travaillent sur une même tâche pour encourager l'apprentissage collaboratif. Si un des apprenants accepte d'assister le demandeur d'aide, l'assistant ouvre une fenêtre de conversation entre les deux apprenants. L'assistant doit travailler intelligemment pour fournir de l'aide à l'apprenant uniquement lorsque cela est justifié. Par exemple, l'aide n'est pas activée à moins que l'apprenant n'ait travaillé pendant un certain temps sur la tâche. Cette information est donnée par l'agent de comportement. D'autre part, l'agent est implicitement activé lorsqu'il est informé par l'agent de comportement que l'apprenant met trop de temps à terminer une tâche. L'agent d'assistance envoie une notification à l'agent de conception de système pédagogique qui décide de ce qui sera présenté à l'apprenant en fonction de la situation signalée par l'agent de comportement.

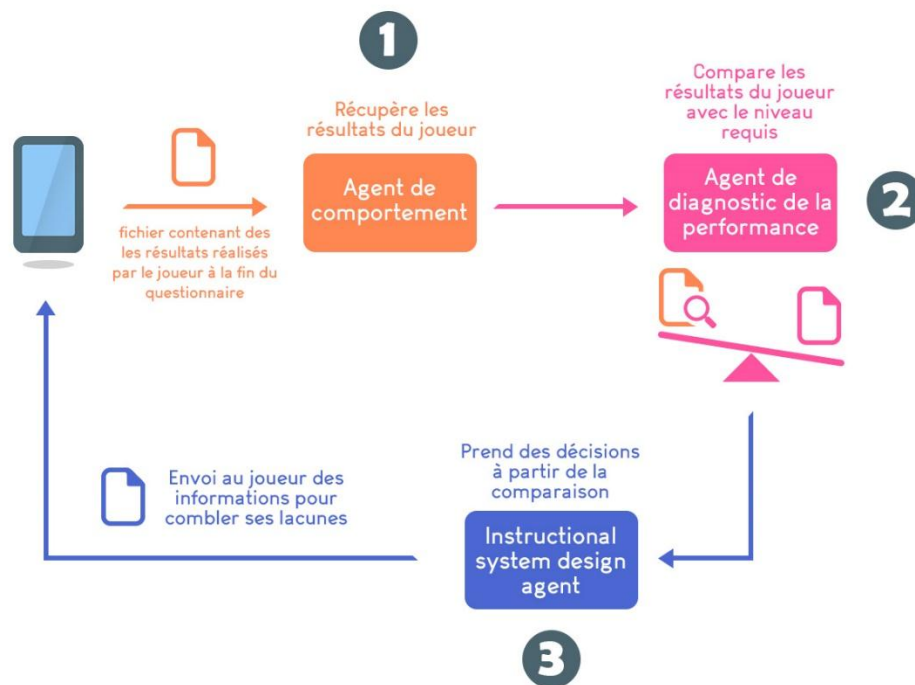


Fig 53. Architecture Multiagent

Pour des raisons de sécurité, la communication entre l'interface du jeu et le système multi-agent s'effectue par le biais de requêtes HTTP. Le mécanisme de sauvegarde uti-

lise un système de points de contrôle profitant ainsi de l'évolution du joueur pour sauvegarder sa progression. Par exemple, lorsque l'apprenant atteint un certain niveau, la fonction de sauvegarde est automatiquement enclenchée, les attributs du jeu sont ensuite envoyés en tant que variables, qui sont collectées et utilisées par le système multi-agent.

L'architecture de notre jeu sérieux, basée sur le système de communication entre agents, est représentée dans la figure (Fig 53).

En utilisant une architecture multi-agent, les jeux sérieux fonctionnent intelligemment et fournissent aux apprenants un contenu pédagogique à la difficulté analysée et ajustée à leurs niveaux. Ces agents ont besoin de données sur le niveau de l'apprenant et ne peuvent pas être performants dès le début de la partie car la collecte s'effectue dans le premier chapitre du jeu. Par exemple, en atteignant un point de contrôle ou en rejouant un niveau que l'apprenant n'a pas réussi à terminer, il aura l'opportunité de faire l'expérience de nouveaux défis avec un nouvel environnement en fonction du fonctionnement des agents intelligents.

Le jeu proposé est conçu pour apprendre le langage de programmation C de manière individualisée et collaborative, les apprenants sont tenus de jouer selon leur propre rythme et peuvent demander de l'aide à tout moment.

6.2. Base de connaissances

Pour organiser le contenu pédagogique à enseigner aussi bien que pour faciliter la communication entre les différents agents, nous avons opté pour la construction d'une base de connaissance via les ontologies. Les ontologies étant un formalisme qui a la particularité d'être compréhensible par les agents humains et les agents logiciels.

L'ontologie est un concept pluridisciplinaire, Une ontologie est une description formelle d'un domaine de connaissances. Concrètement, il s'agit d'un ensemble de concepts et relations permettant de décrire un champ d'informations. Pour mieux définir la notion d'ontologie, il est nécessaire d'appréhender son évolution au cours du temps. Une première définition fut proposée par Neches et ses collègues : « une ontologie définit les termes et les relations de base du vocabulaire d'un domaine ainsi que les règles qui indiquent comment combiner les termes de façon à pouvoir étendre le vocabulaire ». Cette définition fut empruntée de la philosophie où l'ontologie exprime l'ensemble des objets qui existent dans un domaine (Gruber, 1995) et peut être projetée sur l'ingénierie de con-

naissances pour affirmer que l'ontologie est la représentation de concepts, objets et entités, et leurs relations pour représenter un domaine d'intérêt.

En 1993, Gruber propose la définition, qui de nos jours, devient la plus citée dans la littérature: «Une ontologie est une spécification explicite d'une conceptualisation ». Le terme 'spécification' connote une modélisation formelle des connaissances d'un domaine dans le sens où l'ontologie doit être codée dans un langage interprétable aussi bien par les humains que par les machines. Le terme 'conceptualisation' se rapporte à un modèle qui rend compte du sens des termes pour restreindre leur interprétation ou décrit fidèlement une réalité ou un phénomène par le biais de concepts et de relations.

Borst (Borst, Akkermans, & Top, 1997) apporte une légère modification à la définition de Gruber et définit l'ontologie comme « une spécification formelle d'une conceptualisation partagée ». Cette définition s'appuie sur l'idée qu'une représentation à base d'ontologies doit être reconnue et acceptée au sein d'une communauté et n'est pas l'objet d'un individu. Les deux définitions de Gruber et Borst étant populaires, furent combinées dans la définition de Studer 1993 qui définit l'ontologie comme étant : « une spécification formelle, explicite d'une conceptualisation partagée ».

En 1998, Nicola Guarin définit l'ontologie de la façon suivante: la conceptualisation étant une structure $\langle D, R \rangle$: D est le domaine et R est l'ensemble des relations qui existent dans ce domaine. Une ontologie est une théorie de logique qui rend compte du sens d'un vocabulaire par le biais d'axiomes. Comme il a été souligné par Garino une ontologie peut être assimilée à un artefacte informatique qui se compose d'un ensemble d'assomptions explicitant les relations entre les différents éléments. Une ontologie peut être utilisée comme une base de connaissances qui recense et hiérarchise les données nécessaires au fonctionnement d'un système. Ce dernier, exploite l'ontologie par le biais de requêtes et construit de nouvelles connaissances en se basant sur les différentes relations entre les concepts. L'un des avantages des ontologies est de raisonner automatiquement en inférant de nouvelles connaissances.

En ce qui concerne le contenu pédagogique et pour une meilleure organisation, gestion et stockage des connaissances, nous avons développé une ontologie de domaine. Le modèle pédagogique est construit à l'aide l'éditeur Protégé (« protégé », s. d.) .La visualisation des connaissances est assurée par le plugin VOWL. Les principaux avantages de l'utilisation d'une ontologie résident dans sa capacité à assurer la communication entre les agents, la réutilisation et l'organisation des connaissances. L'ontologie du domaine

est une structure indépendante du jeu sérieux qui a pour objectif le regroupement de toutes les connaissances dont le système a besoin pour fonctionner. Dans notre cas, les agents intelligents utilisent l'ontologie pour extraire les contenus pédagogiques à enseigner.

L'ontologie du domaine que nous avons développée est publiée en ligne dans le but de fournir aux applications mobiles un accès fédéré aux données. Un des avantages de l'utilisation de l'ontologie est que de nombreuses applications peuvent partager la même source de données. Par exemple, le système de gestion de l'apprentissage (LMS) peut utiliser la banque de questions dans un module d'évaluation de même que dans notre jeu sérieux. Il facilite également la communication et le partage des connaissances entre agents.

En ce qui concerne le modèle pédagogique, le cours est divisé en plusieurs unités. Chaque unité comprend de nombreuses pages et aborde de nombreuses compétences. Chaque page utilise une ou plusieurs ressources d'apprentissage adaptées aux appareils mobiles. L'ontologie du domaine contient une banque de questions. Chaque question appartient à une catégorie et est liée à une unité d'apprentissage spécifique. Cela implique également un nombre limité d'essais après lequel un feedback est présenté à l'apprenant. Il est également possible de donner des conseils aux élèves lorsqu'ils répondent à une question.

Afin de fournir aux étudiants un feedback automatique et sur mesure, l'ontologie du domaine enregistre les questions ainsi que la ou les bonnes réponses. Cela signifie que les réponses textuelles ne sont pas autorisées car il est difficile pour les agents d'effectuer un traitement de texte automatique. Pour ce faire, il faut choisir la ou les bonnes réponses parmi de nombreuses questions. Pour chaque item du test, nous vous suggérons d'utiliser certains paramètres tels que : difficulté, description, feedback et isCorrect qui est réglé sur vrai s'il s'agit d'une bonne réponse à la question. Cette structure a le potentiel d'aider les agents à choisir les questions au hasard. De plus, l'ontologie est une entité indépendante de tout système et peut être utilisée par les tuteurs pour charger des questions et du nouveau contenu.

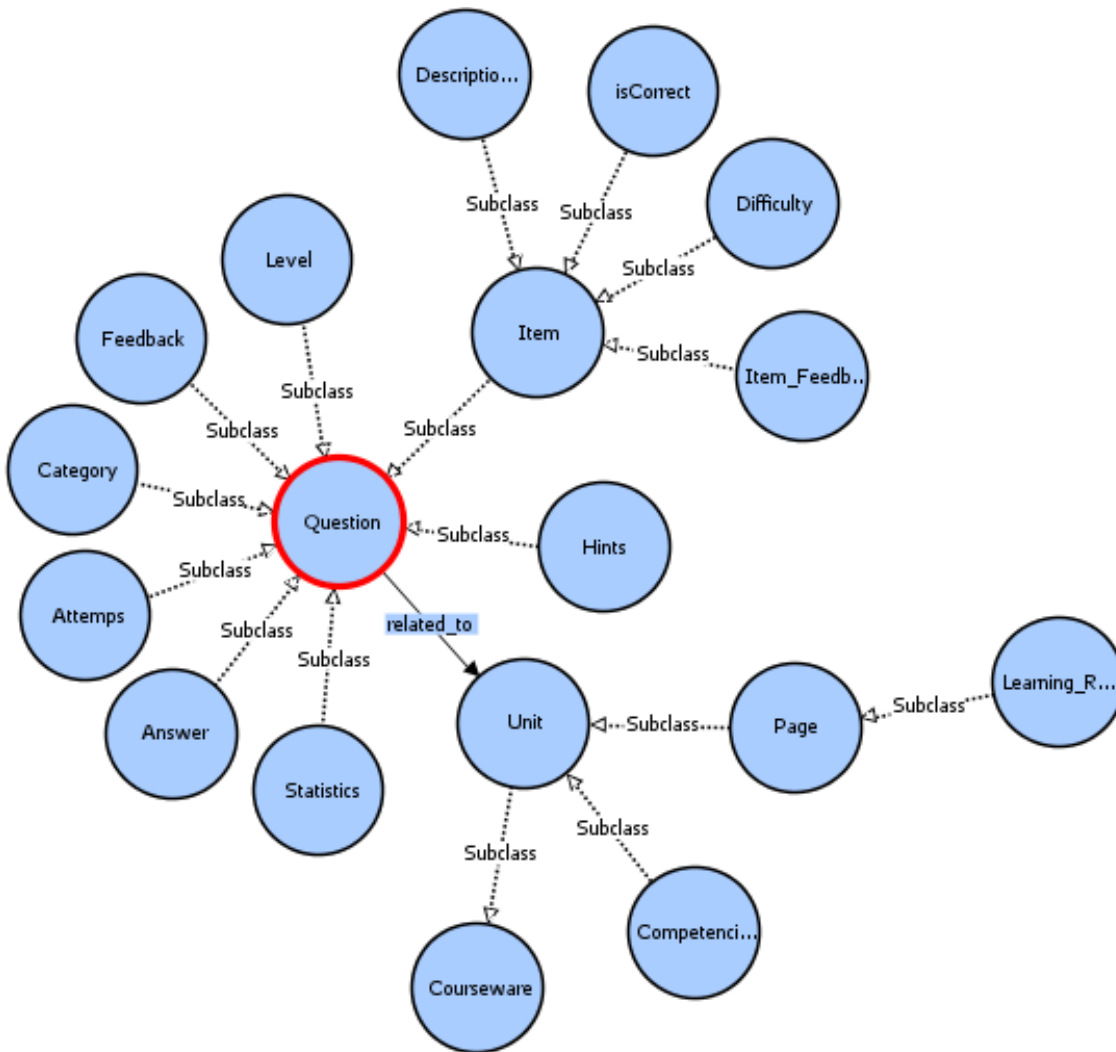


Fig 54. Extrait du modèle pédagogique

6.3. Synthèse

Dans ce chapitre nous avons présenté l'architecture multi-agentS utilisée pour créer un jeu sérieux pouvant inclure l'aspect multi-joueurs. Nous sommes conscients que l'apprentissage collaboratif a le potentiel de motiver l'apprenant et de susciter son intérêt à l'apprentissage. Grâce à l'intelligence des agents, le jeu sérieux pourra fournir aux apprenants un contenu pédagogique à la difficulté analysée et ajustée à leurs niveaux. Le but est de présenter à l'apprenant une meilleure expérience d'apprentissage ludique, permettant au joueur de s'amuser tout en apprenant et d'être guidé tout au long du processus d'apprentissage en fonction des décisions prises par les différents agents. Le système multi-agents comprend 4 agents: "Agent de comportement", "Agent de Diagnostic de la Per-

formance ", " Agent de conception du système pédagogique" et "agent d'assistance". Chaque agent effectue une tâche spécifique, partage des informations et communique avec d'autres agents.

Pour fonctionner, communiquer et échanger des informations les agents font appel à une base de connaissances développée sous forme d'ontologie. L'objectif est d'organiser le contenu pédagogique à enseigner aussi bien que pour faciliter la communication entre les différents agents. L'ontologie est développée avec l'éditeur Protégé avec une sérialisation XML/RDF pour faciliter la structuration des données selon un langage universel qui est le langage XML. L'exploitation de l'ontologie se fait grâce à un langage de requête appelé SPARQL et accessible par tout programme information à travers des APIs.

L'utilisation d'une base de connaissances respectant un langage formel a pour objectif de faciliter la communication des agents dans notre jeu sérieux à travers un standard (XML) et assurer une communication et un échange avec des systèmes existants ou futurs.

Conclusion générale

Les jeux sérieux occupent une place importante dans le domaine éducatif. Ils sont perçus comme des outils innovants et motivants pour transmettre le savoir. Que ce soit dans les disciplines littéraires ou scientifiques, les jeux sérieux exploitent une palette d'éléments interactifs pour rapprocher le sujet de l'apprenant et l'aider à percevoir l'utilité de son apprentissage. Ce projet de recherche a abouti à un prototype et dont l'évolution s'inscrit dans le futur à travers des mises à jour régulières. L'objectif final étant de réaliser une première version de *Globrain's Quest* comprenant l'ensemble des notions de base de Python, enseignées grâce aux différentes techniques et aux pédagogies relevées par ce travail.

En ce qui concerne les techniques utilisées pour développer *Globrain's Quest*, il était important de mener une étude comparative pour déterminer le framework de développement mobile approprié à notre contexte. Parmi les critères retenus, la possibilité de créer une application dite cross-platfrom, une application pouvant être déployée sur la majorité des terminaux mobiles (Android, iOS et Windows phone). Le deuxième critère est l'intuitivité dans la création de l'interface graphique du jeu, ce critère est crucial pour créer des interfaces graphiques au visuel attractif à l'instar des jeux vidéos que les apprenant ont l'habitude de jouer. Au cours de notre processus de recherche, nous avons expérimenté trois frameworks de développement mobile: Stencyl, Xamarin et Unity. La première version du jeu se basait sur le logiciel Stencyl 2.0. Le jeu était appelé « PEROBO » et était développé pour une plateforme web et pouvait être joué en utilisant un navigateur aussi bien qu'un smartphone ou tablette (Yassine et al., 2017a). Cependant, même si le logiciel Stencyl est intuitif et facile à utiliser, il présentait quelques limites: la création d'une scène de jeu se base sur des drag & drop, ce qui limite le développeur à des fonctionnalités restreintes. Le deuxième inconvénient était que l'enchaînement des actions dans le jeu est réalisé en utilisant l'interface graphique. En effet, il était impossible de penser à un système multi-agents pouvant communiquer avec le jeu sérieux parce que le code derrière l'interface graphique est caché par le logiciel. Ces limitations nous ont mené à expérimenter un autre framework de développement mobile: Xamarin.

Xamarin étant un framework de développement mobile populaire et très cité dans la littérature scientifique. Ce dernier permet de créer des applications mobiles natives, c'est-à-dire un même code est utilisé pour créer 3 versions d'applications (Android, iOS et windows).

L'aspect natif représente d'une part la capacité de l'application à fonctionner exactement comme si elle a été spécifiquement conçue pour la plateforme cible. D'autre part, l'application générée peut être déployée sur le *Market Place*. Le *Market Place* désigne l'endroit spécifique dans lequel tout utilisateur cherchera à télécharger et à installer l'application. Le deuxième prototype développé en utilisant le framework Xamarin (Yassine et al., 2018b), présentait des avantages par rapport à celui créé par Stencyl. Tout d'abord, le framework Xamarin nous offre la possibilité de définir l'enchaînement du jeu et les différentes interactions grâce au code. Xamarin se base sur la technologie .Net et sur le langage C# pour définir le code derrière les interfaces graphiques. Il s'agit du type de programmation appelé programmation événementielle. Chaque interaction du joueur avec les composants de l'interface graphique déclenche un événement qui exécute un code et affiche un retour au joueur.

Le deuxième prototype présentait des avantages par rapport au premier. Tout d'abord, grâce au code c#, il était envisageable de concevoir un Système Multi-Agents pouvant communiquer avec le jeu sérieux de façon à rendre l'expérience d'apprentissage intelligente et personnalisée selon l'apprenant qui est en face. Cependant, l'inconvénient majeur de Xamarin était la conception des interfaces graphiques du jeu. Xamarin se base sur un langage spécifique appelé XAML permettant de décrire les composants de l'interface graphique en utilisant des balises XML. Quoique le langage XAML est facile à utiliser mais ne permet pas de créer des jeux sérieux au design attractif, et nous estimons que l'expérience utilisateur joue un rôle primordial en suscitant l'intérêt de l'apprenant. Cette limitation nous a encouragé à chercher un framework de développement mobile nous permettant à la fois de créer un jeu au design attractif et aux interactions riches, avec la possibilité d'écrire le code associé à chaque interface en utilisant un langage de programmation. Notre choix s'est porté sur Unity, un framework de développement dédié principalement aux jeux vidéo. Selon (Brett & Simons, 2017), Unity est un moteur de jeu qui dispose de 85% du marché de développement des jeux vidéo. Il est également supporté par une grande communauté. Un avantage majeur qui a influencé notre choix et qui de plus, offre la possibilité d'utiliser le code C# pour coder les différentes interactions dans le jeu.

Ainsi, le dernier prototype du jeu est développé en utilisant Unity. En résulte une expérience utilisateur présentant une nette différence entre le prototype que nous avons développé avec Xamarin. En utilisant Unity, les interfaces graphiques sont fluides et réagissent rapidement

aux évènements déclenchés par l'utilisateur. En tant que concepteur de jeux, nous avons une grande palette de choix de composants utilisés dans l'univers des jeux. Ainsi cet aspect, nous facilite la phase de création des interfaces graphiques qui nécessitait plus d'effort sur Xamarin. Un autre avantage de Unity, également présent dans le framework Xamarin, est la possibilité de développer le fonctionnement du jeu avec le langage de programmation C#. C'est un avantage majeur nous permettant d'ajouter un aspect multi-joueurs au jeu aussi bien la personnalisation de ce dernier grâce à l'approche multi-agents.

La mise en place d'un scénario dans notre jeu représente l'étape fondatrice de notre vision de l'interaction que l'on propose au joueur. En effet, les niveaux du jeu s'enchaînent selon un scénario qui privilégie l'apprentissage pour créer une expérience à la fois ludique et éducative. Avant de concevoir les personnages et les environnements du jeu, il a fallu se tenir à une règle primordiale: l'expérience de jeu ne doit pas empiéter sur l'apprentissage, ce dernier étant l'essence même de notre projet de recherche. Ainsi, nous nous sommes basés sur les taxonomies qui définissent notre choix scénaristique.

En utilisant les taxonomies, nous avons pu inclure dans le scénario un but ultime comme source de motivation du joueur et avons également organisé l'acquisition des connaissances et compétences à travers le jeu de façon graduée allant des concepts les plus simples à leur application dans les situations les plus complexes. Effectivement, un scénario sans réel finalité peut vite ennuyer le joueur qui n'y percevrait pas le sens de ces actions. Le jeu sérieux est donc doté d'un objectif global, qui donne un sens à toutes les actions entreprises par l'apprenant durant son expérience ludique.

Cette première version de l'application peut être considérée comme une itération qui s'adresse à un public indéterminé. En d'autres termes, elle se veut accessible à tous les apprenants, de tous âges et toutes disciplines. C'est pour cela qu'elle fera l'objet d'une expérimentation auprès d'échantillons d'apprenants de différents âges et différents niveaux. L'objectif de cette démarche est de déterminer le public qui sera le plus réceptif à la démarche pédagogique de l'application et à quel degré cette dernière répond aux attentes des apprenants. Ainsi, l'application pourra être orientée vers une audience plus spécifique, avec un contenu plus adapté et des objectifs d'avantage plus précis.

En termes de perspectives, nous nous intéressons à l'application de l'approche multi-agents dans le cadre des jeux sérieux. L'objectif est de rendre l'expérience du jeu personnalisée se-

lon les actions de l'apprenant. En réalité, l'utilisation des agents a pour objectif d'imiter le comportement d'un tuteur humain en matière de prise de décision. C'est-à-dire, les agents ont pour mission de suivre et observer l'avancement du joueur et proposer des rétroactions appropriées et surtout personnalisées selon le besoin. L'approche multi-agents nous permet également de créer un jeu sérieux multi-joueurs pour favoriser l'aspect de compétition qui est parfois source de motivation des apprenants mais aussi encourager l'apprentissage collaboratif. L'objectif du jeu sérieux finalement n'est pas d'isoler l'apprenant mais au contraire de l'encourager à apprendre tout en collaborant avec les autres. Ceci, permet de promouvoir l'apprentissage collaboratif en ligne qui est l'un des piliers du courant pédagogique socio-constructiviste.

A ce stade, nous avons mis en place l'architecture du système multi-agents aussi bien que la base de connaissances qui sera utilisée par le système pour personnaliser l'expérience d'apprentissage. La base de connaissances étant le cœur du système, aggrège toute l'information utile qui sera exploitée par les agents. Elle est créée selon le formalisme des ontologies pour créer une base de connaissances respectant un format standard de représentation des connaissances qui est le langage XML

Ce travail de recherche, faute d'être complet, ouvre la voie à des perspectives: tout d'abord, l'implémentation du système multi-agents exploitant la base de connaissances est l'une des perspectives de ce travail. A ce stade, nous proposons d'utiliser le langage C# ou éventuellement un autre langage pour créer des agents intelligents. La communication entre l'interface du jeu et le système multi-agent s'effectue par le biais de requêtes/réponses HTTP.

Une autre perspective de ce travail est de réaliser une expérimentation auprès des étudiants pour collecter leur avis concernant le jeu et faire une analyse en Machine Learning. Plus spécifiquement, nous nous intéressons à l'étude de la capacité du jeu sérieux à répondre aux problèmes rencontrés par les étudiants lors de l'apprentissage d'un nouveau langage de programmation, et si possible, pour prédire et anticiper la déction des problèmes d'apprentissage.

Bibliographie

- Adesoji, F. A. (2018). Bloom Taxonomy Of Educational Objectives And The Modification Of Cognitive Levels. *Advances in Social Sciences Research Journal*, 5(5). <https://doi.org/10.14738/assrj.55.4233>
- Anderson, L., & Krathwohl, D. (2001). A Taxonomy for Learning and Assessing: a Revision of Bloom Taxonomy.
- Basque, J. (1999). *Inforoute et technologie éducative à l'aube de l'an 2000 : actes du XIII Colloque du Conseil interinstitutionnel pour le progrès de la technologie éducative (CIPTÉ)*. Télé-université.
- Benner, P. (1982). Issues in Competency-Based Testing. *Nursing Outlook*, 30(5), 303- 309.
- BIGGS, J. B., & COLLIS, K. F. (1982). The Evaluation of Learning: Quality and Quantity in Learning. In *Evaluating the Quality of Learning* (p. 3- 15). Elsevier. <https://doi.org/10.1016/B978-0-12-097552-5.50006-5>
- Bocquillon, M., Gauthier, C., Bissonnette, S., & Derobertmeasure, A. (2020). Enseignement explicite et développement de compétences: antinomie ou nécessité?. *Formation et profession*, 28(2), 3-18
- Brett, J., & Simons, A. (2017). Implementation of the Unity Engine for Developing 2D Mobile Games in Consideration of Start-Up/Student Developers. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10345 LNCS, p. 271- 278). Springer Verlag. https://doi.org/10.1007/978-3-319-65849-0_29
- Ciman, M., & Gaggi, O. (2017). An empirical analysis of energy consumption of cross-platform frameworks for mobile development. *Pervasive and Mobile Computing*, 39, 214- 230. <https://doi.org/10.1016/J.PMCJ.2016.10.004>
- Clark, R. C., & Mayer, R. E. (2012). *e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning: Third Edition*. *e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia*

- Denami, M. (2018). Serious game-based learning: the place of users' verbalization in the acquisition of specific skills. *International Journal of Training and Development*, 22(2), 144- 161. <https://doi.org/10.1111/ijtd.12123>
- Dickson, P. E. (2015). Using Unity to Teach Game Development. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE '15* (p. 75- 80). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2729094.2742591>
- Du, J., Wimmer, H., & Rada, R. (2016). « Hour of code »: Can it change students' attitudes toward programming? *Journal of Information Technology Education: Innovations in Practice*, 15(1), 53- 73. <https://doi.org/10.28945/3421>
- El-Kassas, W. S., Abdullah, B. A., Yousef, A. H., & Wahba, A. M. (2017). Taxonomy of Cross-Platform Mobile Applications Development Approaches. *Ain Shams Engineering Journal*, 8(2), 163- 190. <https://doi.org/10.1016/J.ASEJ.2015.08.004>
- Esper, S., Foster, S. R., Griswold, W. G., Herrera, C., & Snyder, W. (2014). CodeSpells. In *Proceedings of the 14th Koli Calling International Conference on Computing Education Research - Koli Calling '14* (p. 05- 14). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2674683.2674684>
- Fokides, E., & Atsikpasi, P. (2017). Redefining the Framework for Teaching Programming to Primary School Students: Results from Three Pilot Projects. *British Journal of Education, Society & Behavioural Science*, 20(3), 1- 11. <https://doi.org/10.9734/BJESBS/2017/33520>
- Gee, J. P. (2007). *What video games have to teach us about learning and literacy*. Palgrave Macmillan.
- Gee, J. P., & Paul, J. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment*, 1(1), 20. <https://doi.org/10.1145/950566.950595>
- Gomes, A., & Mendes, A. J. (2008). An environment to improve programming education (p. 1). <https://doi.org/10.1145/1330598.1330691>
- Gouws, L. A., Bradshaw, K., & Wentworth, P. (2013). Computational thinking in educational activities. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education - ITiCSE '13* (p. 10). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2462476.2466518>

- Johnston, R. A.-J., & Alexander-James, R. (2015). Constructivism: Theorists and their Theories. *ACE papers, Issue 6: Graduate Student Work - Issues in Contemporary Education, Paper 2*. Consulté à l'adresse <https://researchspace.auckland.ac.nz/handle/2292/25062>
- Kelleher, C., & Pausch, R. (2005, juin). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*. <https://doi.org/10.1145/1089733.1089734>
- Kim, H. J., Karunaratne, S., Regenbrecht, H., Warren, I., & Wunsche, B. C. (2015). Evaluation of Cross-platform Development Tools for Patient Self-Reporting on Mobile Devices. *Proceedings of the 8th Australasian Workshop on Health Informatics and Knowledge Management*, 8, 55- 61.
- Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, 37(3), 14. <https://doi.org/10.1145/1151954.1067453>
- Majgaard, G. (2015). Plants vs zombies as an empowering learning machine. In *Proceedings of the International Conferences on Interfaces and Human Computer Interaction 2015, IHCI 2015, Game and Entertainment Technologies 2015, GET 2015 and Computer Graphics, Visualization, Computer Vision and Image Processing 2015, CGVCVIP 2015 - P* (p. 271- 274).
- Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2013). Towards a new massive multiplayer online role playing game for introductory programming. In *Proceedings of the 6th Balkan Conference in Informatics on - BCI '13* (p. 156). New York, New York, USA: ACM Press. <https://doi.org/10.1145/2490257.2490284>
- Miljanovic, M. A., & Bradbury, J. S. (2016). Robot on! *Proceedings of the 5th International Workshop on Games and Software Engineering - GAS '16*, 33- 36. <https://doi.org/10.1145/2896958.2896962>
- Milne, I., & Rowe, G. (2002). Difficulties in Learning and Teaching Programming—Views of Students and Tutors. *Education and Information Technologies*, 7(1), 55- 66. <https://doi.org/10.1023/A:1015362608943>
- Mouaheb, H., Fahli, A., Moussetad, M., & Eljamali, S. (2012). The serious game: what educational benefits? *Procedia-Social and Behavioral Sciences*, 46, 5502- 5508.

<https://doi.org/10.1016/j.sbspro.2012.06.465>

Muller, M., Buchheister, K., & Boutte, G. (2017). *Multiple perspectives on cognitive development : Radical constructivism, cognitive constructivism, sociocultural theory, and critical theory. Multiple Perspectives on Cognitive Development*. Consulté à l'adresse <https://digitalcommons.unl.edu/famconfacpub/261>

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*, 25(2), 175- 202. <https://doi.org/10.3166/ria.25.175-202>

Paquette, G. (2007). An ontology and a software framework for competency modeling and management. *Journal of Educational Technology & Society*, 10(3), 1–21.

Ribeiro, A., & da Silva, A. R. (2012). Survey on Cross-Platforms and Languages for Mobile Apps. In *2012 Eighth International Conference on the Quality of Information and Communications Technology* (p. 255- 260). IEEE. <https://doi.org/10.1109/QUATIC.2012.56>

Rieger, C., & Majchrzak, T. A. (2019). Towards the definitive evaluation framework for cross-platform app development approaches. *Journal of Systems and Software*, 153, 175- 199. <https://doi.org/10.1016/J.JSS.2019.04.001>

Rubiano, S. M. M., López-Cruz, O., & Soto, E. G. (2015). Teaching computer programming: Practices, difficulties and opportunities. In *Proceedings - Frontiers in Education Conference, FIE* (Vol. 2014). <https://doi.org/10.1109/FIE.2015.7344184>

Sillaots, M., & Maadvere, I. (2013). Students designing educational games. *EAI Endorsed Transactions on Game-Based Learning*, 1(1), e7. <https://doi.org/10.4108/trans.gbl.01-06.2013.e7>

Soobard, R., Rannikmäe, M., & Reiska, P. (2015). Upper Secondary Schools Students' Progression in Operational Scientific Skills – A Comparison between Grades 10 and 12. *Procedia - Social and Behavioral Sciences*, 177, 295- 299. <https://doi.org/10.1016/j.sbspro.2015.02.342>

Wilson, C., & Cameron. (2014). Hour of code. *ACM Inroads*, 5(4), 22- 22. <https://doi.org/10.1145/2684721.2684725>

Yassine, A., Berrada, M., Tahiri, A., & Chenouni, D. (2015). Vers une nouvelle approche

pédagogique d'enseignement de la programmation en se basant sur les techniques des jeux sérieux. In APEL 2015 (Éd.), *1 ère Edition du Workshop International sur les Approches Pédagogiques & E-Learning*.

Yassine, A., Berrada, M., Tahiri, A., & Chenouni, D. (2018a). A cross-platform mobile application for learning programming basics. *International Journal of Interactive Mobile Technologies*, *12*(7), 139- 151. <https://doi.org/10.3991/ijim.v12i7.9442>

Yassine, A., Berrada, M., Tahiri, A., & Chenouni, D. (2018b). A cross-platform mobile application for learning programming basics. *International Journal of Interactive Mobile Technologies*, *12*(7), 139- 151. <https://doi.org/10.3991/ijim.v12i7.9442>

Yassine, A., Berrada, M., Tahiri, A., & Chenouni, D. (2020). A Serious Game for Teaching Python Programming Language. In *Advances in Intelligent Systems and Computing* (Vol. 1076, p. 389- 397). Springer. https://doi.org/10.1007/978-981-15-0947-6_37

Yassine, A., Chenouni, D., Berrada, M., & Tahiri, A. (2017a). A serious game for learning C programming language concepts using solo taxonomy. *International Journal of Emerging Technologies in Learning*, *12*(3), 110- 127. <https://doi.org/10.3991/ijet.v12i03.6476>

Yassine, A., Chenouni, D., Berrada, M., & Tahiri, A. (2017b). ROLE PLAY GAME FOR TEACHING POINTERS USING EDUCATIONAL TECHNOLOGY. In *ICERI2017 Proceedings* (Vol. 1, p. 8758- 8763). IATED. <https://doi.org/10.21125/iceri.2017.2407>

Zapušek, M., & Rugelj, J. (2013). EAI Endorsed Transactions on Game Based Learning Editorial Learning programming with serious games. <https://doi.org/10.4108/trans.gbl.01-06.2013.e6>