

**Faculté des sciences économiques,  
sociales, politiques et de communication**

# **Évaluation du sentiment d'apprentissage de la programmation à travers le jeu vidéo**

Autrice : Audrey Loos  
Promoteur : Thibault Philippette  
Année académique 2022-2023  
Master 120 en sciences et technologies de  
l'information et de la communication à finalité  
spécialisée : conception et évaluation de médias éducatifs



## Remerciements

Il est indispensable pour moi de remercier toutes les personnes qui ont été à mes côtés durant le marathon qu'a constitué ce mémoire.

En premier lieu, je tenais à remercier mon promoteur, Monsieur Thibault Philippette, pour son accompagnement, son dynamisme, ses conseils avisés et sa disponibilité lors des multiples rendez-vous (un peu trop structurés de ma part) que nous avons eu ces deux dernières années. Je souhaite également remercier Arnaud Claes pour l'aide apportée dans la mise en place de mon protocole expérimental, mais également son soutien et sa présence lors de la réalisation de mes expérimentations.

Ensuite, je remercie grandement tous mes participants pour leur dévouement lors de leur session de jeu et de leur entretien, me permettant de réaliser une récolte de résultats fluide et même amusante.

Je tiens à remercier ma famille de m'avoir soutenue et encouragée à donner le meilleur de moi-même malgré les moments difficiles. De plus, parce que je les considère aussi comme ma famille, j'aimerais remercier tous les membres du Kot du Grenier avec qui j'ai eu la chance de vivre ces deux dernières années, à qui j'ai fait traverser mes multiples émotions lors de cette aventure, mais qui m'ont toujours encouragée sans relâche dans la réalisation de mes objectifs.

Je remercie particulièrement Laura et Laureline pour les (trop) nombreuses relectures de ce mémoire et leurs précieuses suggestions d'amélioration, ainsi que Laureline, Manon et Martin pour l'aide apportée lors des retranscriptions d'entretiens. Enfin, je tenais à remercier mes amis Charlotte, Elsa, Laureline, Manu, Martin et Yseult pour leur soutien infaillible et pour avoir cru en moi plus que moi-même.



# Table des matières

<b>INTRODUCTION</b>	<b>1</b>
<b>DESCRIPTION DE LA PROBLÉMATIQUE</b>	<b>3</b>
1 État de l'art . . . . .	3
1.1 Le jeu et le jeu vidéo . . . . .	3
1.1.1 Le jeu . . . . .	3
1.1.2 Le jeu vidéo . . . . .	4
1.2 L'apprentissage par le jeu et le jeu vidéo . . . . .	5
1.2.1 Définition et fondements théoriques de l'ap- prentissage par le jeu . . . . .	5
1.2.2 Apprendre via le jeu vidéo . . . . .	7
1.2.3 L'engagement dans le jeu . . . . .	10
1.3 L'apprentissage de la programmation . . . . .	12
1.4 L'apprentissage de la programmation par le jeu vidéo .	14
2 Question de recherche . . . . .	15
2.1 Le sentiment d'apprentissage . . . . .	15
2.1.1 Le sentiment de compétence . . . . .	15
2.1.2 Étudier le sentiment de compétence . . . . .	16
2.2 Définition de la problématique . . . . .	18
2.2.1 Le sentiment d'apprentissage à travers le jeu vidéo . . . . .	18
2.2.2 Élaboration de la question de recherche . . .	20
3 Hypothèses . . . . .	21
<b>DÉMARCHE MÉTHODOLOGIQUE</b>	<b>24</b>
1 Expérimentation . . . . .	24
1.1 Eye tracker . . . . .	25
1.2 Mise en place . . . . .	27
1.3 Entretien . . . . .	28

2	Choix des jeux . . . . .	29
2.1	Algo-bot . . . . .	29
2.2	Minecraft : Education Edition . . . . .	31
3	Choix des participants . . . . .	36
<b>RÉSULTATS</b>		<b>39</b>
1	Expérience de jeu . . . . .	39
1.1	Algo-bot . . . . .	39
1.2	Hours of Code 2020 (Inclusion) . . . . .	42
1.3	Hours of Code 2021 (Timecraft) . . . . .	44
1.4	Conclusion des auto-confrontations . . . . .	46
2	Entretien . . . . .	47
2.1	Questions thématiques . . . . .	48
2.2	Conclusion de l'entretien . . . . .	56
<b>CONCLUSION</b>		<b>57</b>
1	Discussion des résultats . . . . .	57
1.1	Motivation à apprendre . . . . .	57
1.2	Sentiment de compétence . . . . .	60
1.3	Sentiment de performance . . . . .	61
1.4	Hypothèse principale et question de recherche . . . . .	63
2	Perspectives et limites . . . . .	65
3	Conclusion générale . . . . .	67
<b>BIBLIOGRAPHIE</b>		<b>69</b>

# Introduction

Les jeux vidéo sont devenus des formes incontournables de jeu dans nos sociétés. Ils ont désormais pénétré de nombreux domaines de l'activité humaine : le divertissement et les loisirs, bien sûr, mais également les mondes professionnel et de l'éducation. De plus, au cours des trois dernières décennies, les outils utilisés pour inciter les élèves à programmer sur ordinateur ont considérablement évolué, notamment les jeux vidéo éducatifs proposant des leçons d'informatique. Partant de cette idée fondatrice, nous nous sommes intéressés, dans le cadre de ce mémoire, à l'évaluation du sentiment d'apprentissage de la programmation à travers le jeu vidéo.

Nous avons construit notre question de recherche sur base des lectures présentant les fondements théoriques des différents concepts clés de notre question de recherche. Pour pouvoir mesurer les éventuels effets que les jeux vidéo pourraient avoir, trois notions récurrentes ont été relevées de la littérature : la motivation à apprendre, le sentiment de compétence et le sentiment de performance. Celles-ci ont composé la base de la construction de notre méthodologie.

C'est à travers une quasi-expérimentation, rassemblant neuf participants <sup>1</sup>, que nous avons récolté nos résultats, via une méthode qualitative. Notre méthodologie était divisée en deux phases.

Dans un premier temps, nous avons mis en place une expérience confrontant les participants recrutés à une leçon de programmation sur un jeu vidéo déterminé. Nous avons, pour cette recherche, sélectionné trois types de jeux différents, à savoir Algo-bot et deux leçons de programmation du jeu Minecraft : Education Edition. De plus, le niveau de programmation préalable a constitué le critère principal pour recruter les joueurs. La variation du sentiment de maîtrise de la programmation a permis de comparer les effets d'ap-

---

1. Pour des raisons d'ergonomie de lecture, ce mémoire n'est pas rédigé en écriture inclusive mais il s'adresse néanmoins tant aux hommes qu'aux femmes, ainsi qu'aux personnes s'identifiant par un autre genre.

apprentissage selon le type du jeu.

Dans un second temps, un entretien a été conduit directement après la session de jeu avec chaque participant, retraçant sa performance via une auto-confrontation suivie d'un entretien semi-dirigé. Les questions posées avaient pour objectif de répondre aux effets que les trois notions fondant notre question de recherche pouvaient avoir.

Enfin, nous avons analysé les résultats obtenus afin d'identifier ces éventuels effets sur le sentiment d'apprentissage. Si ceux-ci sont positifs, nous pourrions qualifier le jeu vidéo comme étant un bon média éducatif stimulant le ressenti des apprenants sur leur apprentissage de la programmation.

# Description de la problématique

## 1 État de l’art

### 1.1 Le jeu et le jeu vidéo

#### 1.1.1 Le jeu

La notion de jeu est assez vaste, et une multitude de scientifiques se sont attelés à la tâche de la définir. Le jeu peut être caractérisé comme un « *système dans lequel chaque joueur est engagé dans un conflit artificiel, défini par des règles, qui aboutit en un résultat quantifiable. Les jeux pour apprendre peuvent être définis comme des jeux avec des objectifs d’apprentissage spécifiques* » (Salen & Zimmerman, 2004 ; cités par Plass, Homer, Mayer et Kinzer, 2020). En effet, le jeu pour apprendre s’est intégré petit à petit dans notre monde, apportant à lui seul de nombreux avantages au niveau de l’apprentissage. La psychologie du développement a depuis longtemps reconnu le jeu comme étant une des formes les plus naturelles d’apprentissage : le jeu est essentiel pour le développement cognitif, physique, social et émotionnel de l’enfant (Ginsburg, 2007 ; Hirsh-Pasek, Berk, & Singer, 2009—cf. Lillard et al., 2013 ; cités par Homer, Raffaele et Hendersor, 2020).

Le jeu possède plusieurs caractéristiques. Tout d’abord, il est basé sur des règles qui sont clairement définies. Un jeu est interactif, permettant au joueur de faire des actions et donnant un système de réponses et de feedbacks. De plus, il est challengeant, incluant le plus souvent un élément de chance. La progression dans le jeu est souvent cumulative, en fonction des actions précédentes. Enfin, le jeu invite et motive à l’engagement du joueur (Mayer, 2014 ; cité par Plass et al., 2020). Piaget (1962) caractérise le jeu comme étant motivé en premier lieu par le plaisir et étant une opportunité de mettre en avant et de renforcer ce que l’on connaît déjà (en activant les schémas existants dans de nouvelles situations). D’autre part, Vygotsky (1978) met en avant une plus

grande variété de motivations pour l'engagement dans le jeu, qui est selon lui un outil essentiel à l'apprentissage de nouvelles choses (Homer, Raffaele, & Henderson, 2020). Ces deux auteurs ont apporté des réflexions utiles sur le rôle du jeu pour l'éducation. Dans cette recherche, nous nous positionnerons plus vers le socio-constructivisme de Vygotsky, ce dernier accordant plus d'attention aux éléments sociaux et contextuels de l'apprentissage.

### **1.1.2 Le jeu vidéo**

Les jeux vidéo sont devenus des formes incontournables de jeu dans nos sociétés. Ils ont désormais pénétré de nombreux domaines de l'activité humaine : le divertissement et les loisirs, bien sûr, mais également les mondes professionnel et de l'éducation. Ce phénomène sociétal est omniprésent dans le quotidien des générations actuelles : 95% des adolescents aux États-Unis sont en ligne, et 45% disent être en ligne quasiment tout le temps, essentiellement via leur smartphone (Smith & Anderson, 2018). 99% des garçons et 94% des filles jouent aux jeux vidéo (Pew Research Center, 2008), les garçons jouant plus de 42h et les filles plus de 30h par semaine (Homer, Hayward, Frye, & Plass, 2012).

Les individus réagissent différemment au terme « jeu », chargé d'émotions, selon qu'ils ont joué ou non à des jeux vidéo pendant leur enfance : il s'agit essentiellement d'un problème de fossé entre les générations, car les enfants qui ont grandi depuis les années 1980 ont été exposés aux jeux vidéo toute leur vie (Zyda, 2005).

Les dictionnaires ont tendance à définir un jeu comme une épreuve physique ou mentale, jouée selon des règles spécifiques, dans le but d'amuser ou de récompenser les participants (Larousse, s. d.). Lorsque l'on cherche une définition du terme plus spécifique de jeu vidéo, nous sommes susceptibles de rencontrer une description telle que « un jeu joué contre un ordinateur », qui serait plus précisément à formuler comme « un jeu joué avec un ordinateur » (Zyda, 2005). Cependant, cette formulation est à revoir aujourd'hui tant le jeu vidéo peut être soutenu par plusieurs plateformes, chacune appor-

tant des caractéristiques et fonctionnalités précises au type de jeu appliqué. En effet, un jeu sur console ou sur smartphone ne se joue pas de la même façon de par la singularité du support. Toutefois, nous allons nous concentrer pour cette recherche sur des jeux vidéo se jouant sur un ordinateur.

Après cette brève définition du jeu (vidéo) et de ses caractéristiques, nous allons aborder dans la partie suivante la question de l'apprentissage par le jeu, et plus particulièrement par le jeu vidéo.

## **1.2 L'apprentissage par le jeu et le jeu vidéo**

### **1.2.1 Définition et fondements théoriques de l'apprentissage par le jeu**

L'apprentissage basé sur le jeu (*game-based learning*) implique qu'une tâche d'apprentissage soit redesignée pour la rendre plus intéressante, remplie de sens et, au final, plus efficace à apprendre, qu'une tâche qui ne serait pas tournée vers le jeu (Plass, Homer, Mayer, & Kinzer, 2020). Cela implique une nouvelle pédagogie, et une nouvelle façon d'apprendre qui, si elle est réussie, ne serait pas juste considérée comme une bonne tâche d'apprentissage mais aussi comme un bon jeu (Plass et al., 2020).

Les mécaniques de jeu constituent le *gameplay* principal : c'est un « *ensemble de règles qui définissent la manière de jouer à un jeu vidéo, d'interagir avec l'environnement, lors d'une partie, et qui sont destinées à produire une expérience de jeu agréable* » (Office québécois de la langue française, 2010). Les mécaniques d'apprentissage ont comme but premier l'apprentissage et sont designés directement sur des approches de théories de l'apprentissage : le choix des mécaniques a un effet sur les résultats d'apprentissage (Plass et al., 2012; Plass, O'Keefe, Homer, Case, & Hayward, 2013; cités par Plass et al., 2020). L'apprentissage est facilité quand les mécaniques de jeu et les objectifs d'apprentissage sont en phase (Plass, Homer, & Kinzer, 2015). L'intégration de multiples perspectives d'apprentissage est nécessaire pour que les jeux d'apprentissage atteignent leur plein potentiel (Plass et al., 2020).

Selon Plass et al. (2020), il existe quatre fondements théoriques se soutenant mutuellement dans les jeux pour l'apprentissage :

1. Fondements motivationnels : l'habileté qu'ont les jeux à motiver les joueurs est l'argument le plus cité pour utiliser les jeux en apprentissage (Plass et al., 2015). Les apprenants apprécient et trouvent les activités intéressantes. Le fait de pouvoir recommencer le jeu est l'un des aspects qui motivent. L'échec n'est pas un résultat à éviter : il permet justement de résoudre les différentes tâches. L'échec peut être une étape nécessaire dans l'apprentissage (Kapur, 2008 ; Kapur & Bielaczyc, 2012 ; Kapur & Kinzer, 2009 ; Plass et al., 2010). Les jeux minimisent les conséquences de l'échec, et encouragent la prise de risques, l'exploration, et le fait d'essayer de nouvelles choses (Hoffman & Nadelson, 2010). L'échec permet au joueur d'évaluer sa propre stratégie et de se rendre compte si ses objectifs sont atteints ou non (Barab, Waren, & Ingram-Globe, 2009 ; Kim, Park, & Baek, 2009).
2. Fondements cognitifs : les jeux permettent de sélectionner l'information pertinente du jeu, l'arranger mentalement en une structure cohérente et l'intégrer à ses propres connaissances (Mayer, 2009, 2014).
3. Fondements affectifs : les jeux peuvent avoir des effets sur les émotions des apprenants de nombreuses façons. Il a été montré que les émotions facilitent l'apprentissage (Mayer & Estrella, 2014 ; Plass, Heidig, Hayward, Homer, & Um, 2014 ; Craig et al., 2014 ; D'Mello & Graesser, 2014 ; Graesser, D'Mello, & Strain, 2014).
4. Fondements socioculturels : la recherche sur les aspects sociaux des jeux a révélé des différences entre le jeu individuel, le jeu collaboratif et le jeu compétitif, montrant qu'un jeu de maths visant à automatiser l'arithmétique était plus intéressant lorsque d'autres étaient impliqués soit comme compétiteurs ou en tant que collaborateurs (avec de meilleurs résultats dans la version compétitive du jeu) (Plass, O'Keefe, et al., 2013).

Ces quatre dimensions nous ont permis de poser la base de notre travail. Elles soulèvent des concepts intéressants pour la définition de notre question

de recherche, à savoir la motivation, l'engagement, la gestion de l'échec ou encore les émotions ressenties chez les apprenants.

### 1.2.2 Apprendre via le jeu vidéo

En 2005, avec l'arrivée d'internet à haut débit dans le large public, les MMO (abréviation de MMOG, *massively multiplayer online game*) comme World of Warcraft (2006) ont apporté de nombreuses opportunités éducatives, à tel point que les éducateurs les intégrèrent dans leur classe (Delwiche, 2006 ; cité par Homer et al., 2020). « *Le jeu présente beaucoup de qualités dans le contexte scolaire, grâce aux nombreuses compétences qu'il stimule, mais aucune recherche ne présente de programme scolaire complet, et encore moins pour des profils spécifiques. Cela nous amène à interroger les éventuels effets que ces jeux éducatifs peuvent avoir* » (Karsenti & Bugmann, 2018). Un jeu éducatif a la particularité de présenter des objectifs d'apprentissage clairs et précis, liés explicitement aux concepts ou aux compétences que les apprenants doivent acquérir. Celui-ci se veut engageant et motivant pour inciter plus spécifiquement les étudiants à participer activement à l'apprentissage. Il est important que le jeu vidéo soit adapté au niveau de chaque apprenant, en proposant des activités adaptées à leurs compétences et progression respectives. Enfin, le jeu éducatif doit pouvoir fournir un feedback pour permettre aux étudiants de comprendre leurs progrès et identifier les différents éléments où ils doivent encore s'améliorer.

Gee (2007, cité par Homer et al., 2020) argumente que le jeu vidéo incarne les meilleures pratiques d'apprentissage, en listant 36 principes d'apprentissage qu'il a pu trouver dans le jeu vidéo. Homer et al. (2020) expliquent que le travail de Gee a contribué à l'engouement actuel pour l'utilisation du jeu vidéo dans l'apprentissage. « *Depuis, de nombreux acteurs ont développé leur propre version éducative de jeu populaire, notamment Minecraft Education. Les avantages associés au jeu dans l'éducation sont évidents dans le contexte de l'apprentissage émotionnel et de l'engagement. Cela dit, tous les jeux ne sont pas engageants et toutes les formes de jeu ne sont pas considérées comme positives ou bénéfiques pour l'apprentissage* » (Boyd, 2018).

Plusieurs concepts théoriques clés, issus de la théorie de l'apprentissage et pertinents pour l'apprentissage via les jeux vidéo, ont été présentés par Homer et al. (2020) :

- L'engagement : c'est souvent le haut niveau d'engagement qui a fait dire aux éducateurs que les jeux peuvent être de bons outils d'apprentissage (par exemple, Gee, 2007 ; Plass et al., 2015 ; Prensky, 2006). Quand ils jouent à un jeu vidéo, les joueurs décrivent souvent le fait d'être dans un état d'immersion profonde, ce qui fait partie du potentiel éducatif des jeux vidéo (Hamari et al., 2016).
- La motivation : dans le contexte des jeux vidéo, Siang et Rao (2003) ont réécrit la pyramide des besoins de Maslow pour expliquer les motivations des joueurs. Les motivations peuvent être intrinsèques ou extrinsèques. Dans les environnements de jeux vidéo, les joueurs se voient pousser à atteindre les objectifs eux-mêmes (autonomie), sont soutenus et autorisés à réessayer jusqu'à ce qu'ils atteignent leurs objectifs (compétence) et travailleront souvent en collaboration ou partageront leurs réalisations avec d'autres joueurs (relation). Les étudiants ayant un intérêt situationnel plus élevé ont obtenu de meilleurs résultats d'apprentissage. Ce résultat est un exemple de la façon dont les caractéristiques du jeu peuvent avoir des effets différents selon les apprenants et plaide en faveur de la nécessité de comprendre comment personnaliser au mieux l'expérience d'apprentissage.
- Les différences individuelles et l'adaptation : Kickmeier-Rust et Albert (2010) identifient trois grandes catégories, à savoir la présentation de matériel (aspect et ressenti), le séquençage des personnalisations (pour correspondre aux préférences, aux objectifs, aux connaissances antérieures et d'autres attributs) et l'aide à la résolution des problèmes (donner des conseils, des astuces, des stratégies, et d'autres aides si un apprenant éprouve des difficultés).
- Le design émotionnel et affectif : l'échec dans un environnement d'apprentissage ludique consiste généralement à rejouer une séquence en

tenant compte des réflexions apportées par les échecs précédents. En ce sens, l'échec n'est pas un résultat indésirable, mais plutôt un résultat attendu et souvent considéré comme nécessaire au processus d'apprentissage (Kapur, 2008 ; Plass et al., 2015). Les effets négatifs de l'échec sont donc réduits et la persistance est encouragée. La possibilité de réessayer donne également aux joueurs l'occasion de réguler leur propre apprentissage, puisqu'ils sont en mesure de se fixer des objectifs, de contrôler la réalisation de ces objectifs et d'évaluer l'efficacité des stratégies utilisées pour tenter d'atteindre leurs objectifs (Kim, Park, & Baek, 2009). Les émotions positives, comme l'amusement, donnent au joueur un sens de l'autonomie et font développer chez lui une valeur intrinsèque d'apprentissage. Plass et Kaplan (2016), dans leur approche intégrée cognitive-affectif de l'apprentissage avec les médias, affirment que les émotions jouent un rôle essentiel dans la sélection, l'évaluation et la mise en œuvre de l'apprentissage, l'organisation et l'intégration d'informations visuelles, verbales et auditives pour créer des modèles mentaux intégrés.

Ces quatre concepts théoriques clés croisent les quatre fondements du jeu présentés précédemment. En effet, nous retrouvons encore une fois ici les concepts d'engagement, de motivation ou encore de gestion de l'échec et des émotions dans le jeu. Ils sont, cette fois-ci, qualifiés comme pertinents pour l'apprentissage via les jeux vidéo.

### Le jeu vidéo en contexte scolaire

Karsenti & Bugmann (2018) expliquent qu'il est important de fournir un cadre pédagogique et un encadrement pour l'utilisation du jeu en contexte scolaire. « *Le jeu éducatif peut accompagner un enseignement traditionnel et stimuler l'apprentissage des TIC. [...] Il stimule les compétences du 21e siècle, à savoir l'engagement, la créativité, la collaboration et la résolution de problèmes* » (Karsenti & Bugmann, 2018). Les avantages associés au jeu dans l'éducation sont évidents dans le contexte de l'apprentissage et de l'engagement émotionnels (Boyd, 2018). Certains élèves se voient augmenter leur

propre sentiment de compétences ou d'estime de soi (Karsenti & Bugmann, 2018). Cependant, les avantages ne s'appliquent pas à tous les types de jeux.

*« Dans les jeux vidéo, les joueurs ont de nombreuses occasions de renforcer leurs actions. Les conséquences qu'impliquent les actions d'un joueur sont souvent immédiates. Les jeux vidéo suscitent une participation et un engagement constants de la part du joueur sur de longues périodes - un comportement qui est nécessaire pour un apprentissage réussi. [...] C'est dans la mémoire à court terme que les joueurs coordonnent les informations reçues du registre sensoriel et mobilisent leurs compétences pour atteindre les objectifs du jeu. La répétition de ce processus permet aux joueurs de subir un changement cognitif, de nouvelles informations étant encodées dans la mémoire à long terme. En conséquence, l'apprentissage se produit à mesure que les joueurs acquièrent des compétences pour atteindre les objectifs du jeu »* (Homer et al., 2020). Compléter un niveau implique d'apprendre des compétences dans le jeu (Maxmen, 2010). Il y a une motivation à se dépasser pour valider les différents niveaux (Karsenti & Bugmann, 2018). Bonenfant & Philippette (2018) expliquent qu'en études du jeu, Ian Bogost (2008, 2011a) a été le premier à faire valoir le pouvoir de la rhétorique procédurale des jeux vidéo. Cette rhétorique implique de poser un nombre donné d'actions dans un certain ordre afin d'accomplir certains objectifs. Plus précisément, des opérations logiques organisent le sens des actions performées par les joueurs en visant un but à atteindre (Bonenfant & Philippette, 2018).

### **1.2.3 L'engagement dans le jeu**

*« La notion d'engagement réfère à une disposition subjective qui motive la réalisation d'une action civique et, plus largement, de communication »* (Bonenfant & Philippette, 2018). Dans cet ordre d'idées, s'engager dans un jeu consiste à poser le choix de participer à une expérience souhaitée (Bonenfant & Philippette, 2018). Pour qu'il y ait engagement, il faut un investissement de temps, d'effort et d'attention de la part du joueur ainsi qu'une manipulation appropriée, c'est-à-dire une combinaison entre l'inté-

rêt du joueur de poursuivre l'activité ainsi que des contrôles et réponses du système adaptés à ce qui est exigé — autrement dit, une balance entre compétence et défi nécessaire à l'exercice du jeu (Bonenfant & Philippette, 2018). « *Il y aurait engagement lorsque les schémas à interpréter par les lecteurs ou les joueurs ne sont pas préalablement connus et exigent, pour ainsi dire, des efforts interprétatifs. A contrario, l'immersion et, ultimement, le flow font état d'une expérience où les schémas sont si bien maîtrisés que l'interprète peut être immergé au point d'en oublier le dispositif* » (Bonenfant & Philippette, 2018).

Le fun ou plaisir est alors défini comme une réaction émotionnelle agréable produite par la capacité d'être engagé dans un comportement ludique qui émerge d'une interaction avec le jeu ou les stratégies de ludification (Fizek, 2014). En considérant le glissement possible d'une motivation extrinsèque vers une motivation intrinsèque de plaisir et de satisfaction personnelle, on peut qualifier le type d'engagement de l'individu : celui-ci passerait en effet d'un engagement comportemental à un engagement « optimal », où l'investissement dans un comportement se combine avec un investissement plus affectif (Bonenfant & Philippette, 2018).

Un des arguments principaux de l'apprentissage par le jeu, c'est son habilité à engager les apprenants (Gee, 2007 ; Prensky, 2003 ; cités par Plass et al., 2020). Selon Schwartz & Plass (cités par Plass et al., 2020), il y a quatre types d'engagement :

1. L'engagement comportemental : les actions réalisées.
2. L'engagement cognitif : traitement cognitif de l'information dans le but de rendre le jeu significatif et construire des schémas mentaux (Mayer, 2014).
3. L'engagement affectif : réponses ou connexions émotionnelles de la part du joueur sur certains éléments du jeu (par exemple les personnages (Plass et al., 2019)).
4. L'engagement socioculturel : interactions sociales à travers le jeu mais aussi la culture émergente qui en résulte.

Ces quatre types d'engagement ne sont pas sans rappeler les fondements

du jeu énoncés précédemment. En effet, ces engagements traduisent les concepts précités. Ceux-ci seront utilisés lors de la définition de la question de recherche et des différents questionnements associés.

Ce bref tour d’horizon des rapports entretenus entre jeux (vidéo) et apprentissage nous permet, dans la partie suivante, d’approcher l’objet de notre recherche, à savoir l’apprentissage de la programmation.

### **1.3 L’apprentissage de la programmation**

Le codage informatique - une activité qui implique la création, la modification et la mise en œuvre d’un code informatique et qui expose les élèves à la pensée informatique<sup>2</sup> (*computational thinking*) - fait partie intégrante de l’enseignement actuel des sciences, de la technologie, de l’ingénierie et des mathématiques (STEM) (Grover & Pea, 2013). La programmation créative vise à engager les participants dans une démarche critique, empathique et créative de résolution de problèmes d’une certaine complexité et authenticité, tout en faisant appel à l’usage de stratégies et de processus des sciences informatiques pour la création d’une ou plusieurs solutions (Lepage & Romero, 2017). Le codage informatique comprend des activités de création, de modification et d’évaluation du code informatique ainsi que des connaissances sur les concepts et les procédures de codage (Tondeur et al., 2019 ; cités par Scherer et al., 2021). En fin de compte, les éducateurs en informatique le considèrent comme un véhicule pour enseigner la pensée informatique à travers, par exemple, (a) l’abstraction et la généralisation des modèles, (b) le traitement systématique de l’information, (c) les systèmes de symboles et les représentations, (d) la pensée algorithmique, (e) la décomposition des problèmes, (f) le débogage et la détection systématique des erreurs (Grover & Pea, 2013). Ces compétences partagent des similitudes considérables avec la résolution de problèmes généraux et la résolution de problèmes dans des domaines spécifiques (Shute et al., 2017 ; cités par Scherer et al., 2021).

---

2. La pensée informatique implique la résolution de problèmes, la conception de systèmes et la compréhension du comportement humain, en s’appuyant sur les concepts fondamentaux de l’informatique (Wing, 2006).

Plusieurs arguments peuvent venir encourager l'apprentissage de la programmation. *« Apprendre à coder, c'est aussi développer d'autres compétences comme la résolution de problèmes mathématiques, la pensée critique, les compétences sociales, la gestion de soi et les compétences académiques. [...] Dans un contexte scolaire, apprendre l'informatique soulève trois atouts : préparer les futurs travailleurs au monde de l'IT dans un but économique ; préparer les étudiants à être de futurs entrepreneurs de l'innovation ; gagner des compétences et connaissances propres au 21e siècle. [...] De plus, la programmation peut améliorer le niveau de compréhension de matières diverses »* (Popat & Starkey, 2019).

Au cours des trois dernières décennies, les outils utilisés pour inciter les élèves à coder sur ordinateur ont considérablement évolué, et les langages de programmation visuelle tels que Scratch<sup>3</sup> simplifient la création et la compréhension du code informatique (Scherer, Siddiq, & Sánchez-Scherer, 2021). Cet exemple de version ludifiée d'un environnement de programmation permet aux élèves d'être plus confiants dans leur manière d'apprendre car ils ont appréhendé la difficulté à travers un jeu (Maxmen, 2010). L'environnement d'apprentissage qu'offrent les jeux vidéo est malléable, dans le sens où il permet aux concepteurs de programmes éducatifs d'introduire les principes de mathématique dans leur programme, et ainsi appréhender l'apprentissage contextualisé (Homer et al., 2020). Un bon exemple à présenter est le projet RAPUNZEL mené par Plass, Goldman, Flanagan et Perlin (2009 ; cités par Homer et al., 2020) qui avait pour objectif de développer et évaluer un jeu en ligne pour apprendre les compétences de programmation de base aux jeunes filles. Plass et al. (2009 ; cités par Homer et al., 2020) ont montré qu'après avoir joué au jeu, les étudiantes de sixième avaient des améliorations significatives de leurs compétences, notamment l'estime de soi, l'auto-efficacité en informatique et l'auto-efficacité dans la programmation.

La résolution de problèmes par la modélisation mathématique exige des

---

3. Scratch (<https://scratch.mit.edu/>) est un langage de programmation graphique qui a pour objectif l'apprentissage du codage.

élèves qu'ils décomposent un problème en ses parties (p. ex., variables), qu'ils comprennent leurs relations (p. ex., fonctions), qu'ils utilisent des symboles mathématiques pour représenter ces relations (p. ex., équations) et qu'ils appliquent des algorithmes pour obtenir une solution - des activités qui imitent le processus de codage (Scherer et al., 2021). Les modèles de transfert existants suggèrent que plus les tâches pendant l'enseignement d'un domaine (par exemple, le codage) sont similaires à celles d'un autre domaine (par exemple, la résolution de problèmes mathématiques), plus les élèves sont susceptibles de transférer leurs connaissances et leurs compétences entre les domaines (Taatgen, 2013 ; cité par Scherer et al., 2021). Cette nécessité du transfert de connaissance est l'un des objectifs éducatifs les plus importants, car quand ce transfert à lieu, cela indique qu'il y a un apprentissage significatif (*meaningful learning*) (Mayer, 2002).

#### **1.4 L'apprentissage de la programmation par le jeu vidéo**

L'informatique est un domaine en constante croissance et omniprésent dans nos sociétés. Des connaissances de base deviennent presque indispensables dans le monde professionnel. De plus, apprendre la programmation, c'est pouvoir développer des compétences pratiques comme la résolution de problèmes, la création de solutions logicielles ou encore la prise de décision. La créativité tient également sa place dans les compétences développées à travers la programmation. Enfin, les différentes aptitudes de pensée informatique exercées permettent à l'apprenant d'améliorer sa capacité d'analyse et sa résolution de problèmes de manière logique et méthodique. Toutes ces compétences semblent proches de celles engagées par le jeu vidéo lorsqu'il est utilisé dans un contexte d'apprentissage. Nous pouvons retenir que le jeu engage, motive et améliore la gestion de l'échec chez le joueur. Si nous choisissons des jeux dont les mécaniques coïncident avec des objectifs d'apprentissage de la programmation, nous pourrions identifier des effets sur cet apprentissage.

**C'est pour l'ensemble de ses raisons que nous avons trouvé intéressant de choisir la programmation comme domaine de compétences à ana-**

lyser via l'apprentissage par le jeu vidéo. Nous allons pouvoir, dans la partie suivante de ce rapport, établir notre question de recherche et les hypothèses associées.

## 2 Question de recherche

### 2.1 Le sentiment d'apprentissage

#### 2.1.1 Le sentiment de compétence

*« Dans une perspective cognitive, la majeure partie des théories sur le contrôle dans la vie humaine affirment d'une façon ou l'autre que le niveau de motivation, les états affectifs et les actions des personnes sont fondées plus sur ce qu'elles croient être capables de faire que sur leurs capacités réelles objectives. D'où la nécessité d'orienter la recherche vers les croyances que les gens ont sur leurs capacités, leurs sources et leurs conséquences » (Bandura, 1995).*

Le sentiment de compétence peut se définir comme faisant référence aux croyances qu'une personne entretient à propos de ses capacités à organiser et exécuter les actions requises pour gérer les situations « prospectives ». C'est donc une des dimensions du concept de soi, c'est-à-dire de la représentation qu'un individu se fait de lui-même (Bandura, 1995). *« Le terme « sentiment de compétence » reflète donc d'une certaine façon la certitude préalable qu'une personne a d'avoir la capacité à effectuer une activité, à progresser dans un domaine donné ou à contrôler certaines situations, et d'atteindre ainsi un résultat donné. Le jugement sur l'efficacité d'une action entreprise est au contraire une réaction interne aux résultats de cette action, par voie de feedback. Le sentiment de son efficacité sera mieux apprécié par l'intermédiaire des jugements que la personne portera sur sa progression en cours d'activité, ou rétroactivement après que l'activité sera finie » (Ruph, 1997).* Le sentiment de compétence est donc une émotion ressentie par la personne lorsqu'elle est capable de réaliser une tâche ou une activité de manière efficace et avec succès. Cette émotion est positive car elle encourage et amène

l'individu à s'efforcer de réussir et se dépasser.

Pour la théorie sociale cognitive, le sentiment de compétence académique (*academic self-efficacy*) réfère aux jugements que l'on porte sur ses propres capacités à organiser et à exécuter les actions en vue d'atteindre un certain type d'apprentissage (Ruph, 1997). Nous définirons dans ce travail ce sentiment de compétence académique comme étant le sentiment d'apprentissage.

Au plan des compétences académiques, le fait de posséder les connaissances et les habiletés suffisantes à un apprentissage ne veut pas dire nécessairement qu'on les appliquera avec efficacité face à des conditions difficiles (Bandura, 1993). Le feedback social et évaluatif accompagnant l'instruction formelle influence nettement les croyances en sa compétence, qui à son tour augmente le développement des compétences académiques (Zimmerman, 1995; cité par Ruph, 1997). Quand le feedback pour les succès précédents est attribué à l'effort, les apprenants perçoivent plus de progrès, entretiennent une plus haute motivation et une perception plus forte de leur compétence pour des apprentissages suivants (Schunk, 1987; cité par Ruph, 1997). Ils deviennent motivés et déterminés à réussir les tâches qui leur sont données. La fréquence et l'immédiateté du feedback affecte aussi la perception de sa compétence (Schunk, 1983b; cité par Ruph, 1997).

### **2.1.2 Étudier le sentiment de compétence**

Bandura (1995) explique que les recherches sur les divers effets du sentiment de compétence personnelle montrent que les personnes qui ont un sentiment de compétence faible dans un domaine donné fuient les tâches difficiles qu'elles perçoivent comme des menaces personnelles. De ce fait, elles ont des aspirations moindres et un faible engagement envers les buts qu'elles se sont choisis. Au contraire, un solide sens de sa compétence personnelle augmente les réalisations personnelles et le bien-être de plusieurs façons. Les tâches difficiles sont abordées comme des défis à relever plutôt que des menaces à contourner. Une telle perception favorise une motivation intrinsèque et un engagement en profondeur dans les activités (Bandura, 1995). « *La croyance dans sa compétence joue un rôle clé dans l'autorégulation de la motivation*

[...]. *Les gens se motivent et guident leurs actions en anticipant par la pensée les résultats probables de leurs actions en fonction des croyances qu'ils entretiennent sur leurs capacités* » (Ruph, 1997).

Une discordance entre le sentiment de compétence et la compétence réelle peut résulter, soit dans un excès de confiance en soi, soit dans un manque de confiance en soi, quand un individu aborde une tâche d'apprentissage nouvelle pour lui (Bandura, 1995). En conséquence, les premiers essais sont les plus sensibles : ils déterminent presque complètement le sentiment de compétence et l'engagement futur dans les tâches semblables (Kanfer, Ackerman et Heggstad, 1996 ; cités par Ruph, 1997). Bandura (1995) indique qu'un sens durable de ses compétences ne s'acquiert qu'en passant au travers des difficultés grâce à l'effort et à la persévérance. Après qu'une personne ait été convaincue qu'elle possède ce qu'il faut pour réussir, elle va donc persévérer face à l'adversité et se reprend facilement après des échecs (Bandura, 1995). « *Le sentiment de sa compétence influence les attributions causales : les personnes qui se considèrent comme hautement compétentes attribuent leurs échecs au manque d'effort ou à des conditions adverses alors que ceux dont le sentiment de compétence est bas attribuent leurs échecs à une aptitude faible (facteur interne incontrôlable)* » (Ruph, 1997). Les apprenants qui ont un sentiment de compétence élevé se révèlent meilleurs dans leur gestion du temps, sont plus persévérants, sont moins susceptibles de rejeter prématurément de bonnes hypothèses et sont meilleurs dans la résolution de problèmes conceptuels que des apprenants d'habileté égale mais avec un sentiment de compétence plus faible (Bouffard-Bouchard, Parent et Larivée, 1991 ; cités par Ruph, 1997).

Ainsi, le sentiment d'apprentissage peut être défini comme étant l'évaluation que nous faisons de notre propre capacité à apprendre de nouvelles choses. Celui-ci peut être influencé par notre niveau de confiance en nos propres compétences. Nous pouvons dès à présent mettre en lien cette définition avec le domaine choisi pour notre recherche, à savoir l'apprentissage de la programmation à travers le jeu vidéo, et établir notre question de recherche.

## 2.2 Définition de la problématique

### 2.2.1 Le sentiment d'apprentissage à travers le jeu vidéo

Comment mettre en lien la notion de sentiment d'apprentissage définie précédemment et la théorie sur l'apprentissage via le jeu vidéo ? Comment évaluer le sentiment d'apprentissage ressenti par un individu jouant à un jeu vidéo éducatif ? Quelles sont les compétences qu'il va développer à travers le jeu ? C'est un champ d'application et de recherche qui ouvre sur des pistes d'analyses intéressantes à suivre.

Dans un premier temps, le jeu vidéo offre un feedback permettant au joueur de comprendre ses progrès et d'identifier les différents éléments à améliorer. Que cela soit à la fin d'un niveau ou d'une mission proposée par le scénario du jeu, le feedback lui permet de se rendre compte de sa performance, afin d'améliorer les éléments auxquels il a échoué ou qu'il a moins bien réussi, mais aussi les tâches réalisées correctement qui lui ont permis d'atteindre ou de se rapprocher de ses objectifs d'apprentissage. Ainsi, il peut juger l'efficacité d'une action en temps réel : cette façon de faire lui a-t-elle permis de progresser dans les objectifs d'apprentissage fixés préalablement ? Nous pouvons ainsi définir le jugement et l'évaluation objectives de ses propres actions résultant comme étant un sentiment de performance.

De plus, le feedback social et éducatif, dans lequel peut s'inscrire l'utilisation d'un jeu vidéo éducatif, comme une classe par exemple, influence les croyances sur ses propres compétences. Si le joueur reçoit un feedback de ce qu'il a réalisé dans le jeu, la perception de son sentiment de compétence sera affectée en fonction des efforts mis en place pour atteindre les objectifs académiques proposés par le jeu.

Le jeu vidéo engage le joueur dans la réalisation des tâches proposées. S'il est engagé, il va chercher à effectuer ses tâches comme un défi à relever, tout en restant concentré. Il va pouvoir voir ainsi son évolution et sa progression dans le jeu. Le feedback ici reflète l'engagement du joueur : celui-ci se sentira performant de par son évolution et son implication dans le jeu. Plus il sera

engagé, plus il mettra d'efforts dans la réalisation de ses tâches, et plus il se sentira performant et ressentira un effet positif sur son sentiment d'apprentissage.

Recevoir un feedback positif entretient une plus haute motivation chez le joueur. Celui-ci s'est vu atteindre ses objectifs, en ayant la possibilité de réaliser ses tâches en plusieurs essais. Si le sentiment de compétence est fort, le joueur participera de manière active à l'apprentissage proposé par le jeu vidéo en résolvant des énigmes, en faisant des choix et en prenant des décisions. L'augmentation de la motivation intrinsèque par le jeu vidéo est importante pour stimuler positivement le sentiment d'apprentissage. Le fait d'atteindre ses propres objectifs soi-même et de pouvoir répéter les actions pour réussir sont des éléments importants afin que le joueur trouve les activités intéressantes, ce qui contribuera à un meilleur ressenti d'apprentissage. La croyance en sa compétence joue un rôle clé dans l'autorégulation de la motivation. À quel niveau la motivation apportée dans le jeu influence le ressenti d'apprentissage? Si le joueur est motivé en jouant au jeu, il réalisera les activités, difficiles ou non, avec plus d'entrain et verra mieux sa progression dans les objectifs qu'il a à atteindre. C'est une des questions à laquelle nous tenterons de répondre.

Le glissement d'une motivation extrinsèque vers une motivation intrinsèque offert par le jeu vidéo et un sentiment d'apprentissage positif développeront chez le joueur un investissement plus affectif et un engagement optimal. Une meilleure perception de ses compétences augmente l'engagement dans les activités proposées. Le jeu vidéo permet d'éviter de fuir des tâches jugées plus difficiles en engageant le joueur dans la réalisation de ses objectifs. Il faut ici un bon équilibre entre les compétences stimulées et le niveau du défi afin de garder le joueur engagé. L'immersion offerte par le jeu vidéo peut pallier cet équilibre. L'engagement dans ce cas-ci serait de type comportemental et cognitif. En effet, le joueur chercherait à réaliser les bonnes actions et entraînerait son traitement de l'information et sa construction de bons sché-

mas mentaux. Comme l'explique la littérature, un haut niveau d'engagement fait du jeu vidéo un bon outil d'apprentissage (Gee, 2007 ; Plass et al., 2015 ; Prensky, 2006 ; cités par Homer et al., 2020).

L'échec est nécessaire à l'apprentissage, et c'est une composante principale du jeu vidéo. Il permet au joueur d'évaluer sa propre stratégie et de se rendre compte des objectifs atteints. Si le joueur pense qu'il en est capable à travers le jeu vidéo, il se sentira motivé et effectuera les actions en conséquence. Il est important de réguler la différence entre compétence réelle et le sentiment de compétence, afin de ne pas tomber dans de l'excès ou dans un manque de confiance en soi. Si le joueur est convaincu de ses compétences, il aura une meilleure gestion de l'échec, qui est offerte par le jeu, et une persévérance dans les différents niveaux pour accomplir les objectifs fixés. De plus, le jeu vidéo peut avoir des effets sur les émotions de l'apprenant : celles-ci peuvent faciliter l'apprentissage. Si le sentiment d'apprentissage est stimulé grâce au jeu vidéo, le joueur obtiendra une meilleure gestion du temps, une meilleure résolution de problème, ainsi qu'une persévérance dans la réalisation des tâches. Ces compétences vont lui permettre de mieux comprendre et retenir les concepts en les mettant directement en pratique.

### **2.2.2 Élaboration de la question de recherche**

La combinaison du jeu vidéo et de la programmation pour une recherche semble être naturelle, de par la nature numérique des deux domaines. Cette association peut amener à stimuler le sentiment d'apprentissage au travers de la motivation, l'engagement et le ressenti auto-déclaré des joueurs. Ces trois éléments ont fondé la base de la question de recherche menée lors de ce travail.

C'est sur base de ces éléments d'analyse que nous pouvons construire notre sujet et question de recherche. L'objectif de cette recherche a été de chercher à **évaluer le sentiment d'apprentissage de la programmation via le jeu vidéo**. Ce sujet nous amène à la question de recherche suivante : **quels**

## **effets le jeu vidéo peut-il avoir sur le sentiment d'apprentissage de la programmation ?**

Le jeu vidéo, comme nous l'avons vu précédemment, est un formidable outil d'apprentissage. Si ses effets sont positifs, et que le jeu vidéo se révèle efficace et pertinent à ce type d'apprentissage, il pourrait intégrer les classes d'école ou les centres de formation en plus du cadre de formation informelle. Un élève motivé par le jeu, engagé au vu de sa progression et exprimant un ressenti d'apprentissage positif pourrait qualifier le jeu vidéo comme un bon média d'apprentissage, car le sentiment d'apprentissage ressenti au final serait stimulé et positif.

Les éléments précédemment retenus et fondant notre définition du sentiment d'apprentissage via le jeu vidéo, à savoir la motivation, l'engagement et le ressenti d'apprentissage, ont fixé la base de l'élaboration de nos hypothèses de recherche.

### **3 Hypothèses**

Suite à cette revue de la littérature scientifique, nous avons établi l'hypothèse de recherche suivante : **l'utilisation du jeu vidéo éducatif stimulera le sentiment d'apprentissage de la programmation**. Cette hypothèse se connecte directement à l'objet même de notre recherche, à savoir l'évaluation du sentiment d'apprentissage de la programmation à travers le jeu vidéo.

Notre hypothèse principale peut se découper en trois sous-hypothèses basées sur notre définition du sentiment d'apprentissage. Chaque sous-hypothèse sera testée via des indicateurs. Un indicateur est une variable observable et mesurable dont l'état permet de falsifier l'hypothèse (Culot & Ligurgo, 2021). Chaque sous-hypothèse amène des effets attendus différents que nous avons jugé intéressants à analyser pour répondre à l'hypothèse principale. Si la majorité des sous-hypothèses est réfutée, alors l'hypothèse principale sera invalidée et nous pourrions qualifier le jeu vidéo comme n'étant pas un média

éducatif stimulant le sentiment d'apprentissage.

#### Sous-hypothèse n°1

**Les joueurs seront plus motivés à apprendre la programmation après avoir joué au jeu vidéo.** En effet, la motivation joue un rôle clé dans le sentiment d'apprentissage. Est-ce que celui-ci sera stimulé grâce à la motivation que le jeu vidéo peut amener ? Si un joueur est motivé, il verra son sentiment d'apprentissage évoluer de manière positive. Le jeu vidéo aurait alors pour effet de motiver l'apprenant dans la résolution de sa tâche et lui permettre un meilleur ressenti d'apprentissage. Pour cette première sous-hypothèse, nous avons quatre indicateurs :

1. Une échelle d'auto-perception sur la motivation à apprendre à programmer suite à la session de jeu.
2. La réticence à apprendre la programmation.
3. La volonté de rejouer à d'autres jeux pour apprendre à programmer.
4. L'intérêt du joueur face aux exercices demandés et sa concentration tout le long de la session de jeu.

#### Sous-hypothèse n°2

**Les jeux vidéo dédiés améliorent le sentiment de compétence à programmer.** Nous rejoignons ici la notion de ressenti d'apprentissage. Le joueur exprime-t-il un quelconque ressenti d'apprentissage lorsqu'il a réalisé les tâches demandées ? La gestion de l'échec dans les niveaux proposés a-t-elle permis au joueur de comprendre ses erreurs et d'évaluer son propre progrès dans le jeu ? Comment le joueur a-t-il perçu la notion d'échec dans le jeu, et quel effet cela a-t-il pu avoir sur son sentiment de compétence ? Pour cette deuxième sous-hypothèse, il y a trois indicateurs :

1. Une échelle d'auto-perception sur l'impression du joueur à avoir appris un peu plus à programmer suite à la session de jeu.
2. L'avis du joueur sur le format du jeu vidéo en tant qu'outil d'apprentissage de la programmation.

3. L'évolution de plus en plus rapide dans les exercices.

Sous-hypothèse n°3

**Pouvoir suivre l'évolution des exercices réussis dans le jeu vidéo permet aux joueurs de se sentir performants.** L'engagement du joueur tout du long de son évolution dans le jeu joue-t-il un rôle dans sa performance ? Un joueur engagé va-t-il mieux évaluer sa propre performance et son apprentissage suite à sa session de jeu ? Le jeu vidéo permet-il d'orienter les efforts d'apprentissage en fonction du feedback reçu, définissant l'évolution du joueur dans la tâche confiée ? Ce feedback donne-t-il du sens à la progression ? Enfin, cette troisième sous-hypothèse comporte un seul indicateur :

1. Une échelle d'auto-perception sur le sentiment de performance suite à l'évolution dans le jeu.

Ces indicateurs sont directement intégrés au protocole d'expérimentation que nous avons mené lors de cette recherche. Ils sont donc facilement identifiables, et les résultats répondront directement aux sous-hypothèses.

# Démarche méthodologique

## 1 Expérimentation

Pour pouvoir évaluer ce sentiment d'apprentissage de la programmation à travers le jeu vidéo, une quasi-expérimentation a été mise en place, via une méthode qualitative. Une quasi-expérimentation est une étude qui se déroule sur le terrain et qui implique un changement dans une variable-clé indépendante et d'intérêt, mais qui assouplit l'un ou les deux critères définissant les expériences de terrain en laboratoire : l'affectation aléatoire aux conditions de traitement et la manipulation contrôlée de la variable indépendante (Grant & Wall, 2009).

En effet, notre recherche s'est basée principalement sur une mise en place expérimentale, qui va s'articuler entre deux jeux vidéo, répartis en trois types de gameplay. Ces types, à savoir la programmation via une barre d'instructions, via des blocs ou via un langage de programmation, traduisent la « difficulté » de l'apprentissage de la programmation (simple, moyen, soutenu).

Ensuite, cette quasi-expérimentation a rassemblé trois types de joueur. La connaissance préalable en programmation a constitué le critère principal pour définir le type de joueur. Ceux-ci ont été divisés en trois catégories, pour trois participants dans chacune d'elle : « débutant » (n'ayant jamais programmé), « intermédiaire » (ayant des connaissances de base en programmation et/ou programmeur occasionnel - dans le cadre d'un cours par exemple) et « confirmé » (ayant des connaissances approfondies en programmation et/ou programmeur régulier).

La variation du sentiment de confort et de maîtrise de la programmation permet de comparer les effets d'apprentissage selon le type de jeu. Un joueur débutant la programmation aura-t-il un meilleur ressenti d'apprentissage de la programmation via une barre d'instructions ? Nous tenterons d'identifier ces effets lors de l'analyse des résultats.

	Algo-bot ( <i>simple</i> )	MEE Bloc ( <i>moyen</i> )	MEE Python ( <i>soutenu</i> )
Débutant	1	1	1
Intermédiaire	1	1	1
Confirmé	1	1	1

FIGURE 1 – Tableau croisé des participants et des jeux sélectionnés

## 1.1 Eye tracker

Lors de l'expérimentation, un eye tracker a été mis en place comme une méthode supplémentaire d'évaluation et d'analyse des jeux vidéo utilisés, afin de fournir des informations relatives à la manière dont les joueurs visualisent l'interface du jeu et explorent les niveaux auxquels ils jouent (Almeida, Veloso, Roque, & Mealha, 2011). Il avait également comme objectif d'identifier des indicateurs qui permettraient d'évaluer l'immersion du participant dans un environnement de jeu (Cierro, Philippette, Francois, Nahon, & Watrin, 2020). En effet, le logiciel associé à l'outil permet d'obtenir le retour vidéo de la performance de jeu du participant, accompagné de son parcours oculaire (Figure 2). Les données statistiques mesurées par l'outil n'ont pas été prises en compte dans ce cas-ci, car l'eye tracker a servi principalement à l'utilisateur pour réaliser une rétrospective face à son propre parcours oculaire en vidéo. La confrontation du participant à sa performance permet d'obtenir un meilleur retour critique de sa part, de rediscuter des éléments que lui et l'expérimentatrice ont jugé intéressants, et de chercher les stratégies mises en place par le joueur (Bojko, 2005). Ces éléments intéressants ont notamment été identifiés via le parcours oculaire superposé à la vidéo du jeu. Lorsque le participant fixait un item du décor ou du gameplay plus longtemps, le « point de regard » (*gaze point*) s'intensifiait en devenant plus gros, ce qui rend la lecture du parcours oculaire et des éléments clés limpide.

Lors de l'analyse, ces retours vidéo ont permis de repérer et de comparer des éléments intéressants et semblables dans les performances du joueur qui n'auraient pas été identifiés directement lors de son auto-confrontation. De

plus, c'est via le logiciel de cet eye tracker que l'expérimentatrice observait les participants à distance, en tagguant aux moments opportuns dans le logiciel les éléments clés de la réalisation des tâches.



FIGURE 2 – Exemple de suivi du parcours oculaire lors d'une session de jeu

L'utilisation de ce genre d'outil peut amener la question de sa précision et des problèmes techniques. Ici, rien à signaler du côté technique. Au niveau de la précision, les participants sont restés relativement tous bien statiques et dans le cadre de l'appareil. Deux participants seulement portaient des lunettes lors de l'expérimentation : l'information captée par l'appareil est moins fiable lorsque c'est le cas, à cause des reflets de lumière sur les verres. Quoiqu'il en soit, le fait d'avoir des pertes de fréquences ou de linéarité dans le parcours oculaire n'a pas posé de problèmes majeurs pour l'auto-confrontation ou l'analyse des résultats. En effet, l'auto-confrontation se déroulait directement après la session de jeu, ce qui réduisait le risque de perte de mémoire entraînant des hésitations dans la verbalisation d'aspects plus précis (Philipette, 2016). Le message qui ressort des vidéos est lisible dans son ensemble, même avec quelques pertes dans le cas où le participant perdait la connexion avec l'outil.

L'eye tracker utilisé, le Gazepoint GP3 HD 150 Hz, est un modèle non-invasif qui se fixait directement à la base de l'écran. Le participant devait

se tenir dans une position confortable mais fixe après avoir réglé l'angle de captation de l'appareil. Cette contrainte physique demandait au participant un minimum de son attention pour pouvoir rester droit et face à l'outil. Dès lors, il pouvait ressentir une gêne dans une certaine mesure de ne pas pouvoir se positionner naturellement et bouger selon ses envies ou besoins.

Ce sont les logiciels Gazepoint qui ont été utilisés avec cet outil, permettant une multitude de manipulations :

- Gazepoint Control pour allumer l'eye tracker ;
- Gazepoint Analysis pour créer le projet, le type de captation (ici capture d'écran), contrôler le démarrage de l'expérimentation et récupérer les fichiers en fin de session ;
- Gazepoint Remote pour lancer et observer l'enregistrement à distance sur un autre ordinateur.

Un dernier élément important à noter ici est que le fait d'avoir deux jeux (et trois missions différentes) fait évidemment varier les zones d'intérêt observées par les participants. La nature du jeu influence ces zones d'intérêt (Cierro et al., 2020). Algo-bot est un jeu 2D avec une barre d'action et une zone d'observation fixes. Minecraft : Education Edition est un jeu immersif en 3D où le joueur doit se déplacer et explorer des zones étendues, interagir avec le décor et les personnages non-joueurs. En ce sens, nous comparerons les zones d'intérêt uniquement entre les trois joueurs qui ont réalisé la même mission.

## **1.2 Mise en place**

Après avoir été accueilli, le participant était briefé sur la mission qu'il allait devoir réaliser. Chaque participant s'est vu attribuer un jeu, en fonction de son niveau, comme expliqué précédemment. Le jeu était déjà lancé sur la machine, ainsi que le logiciel relié à l'eye tracker. Une fois le brief terminé et les formulaires de consentement signés (cf. Annexe 3), le participant était installé devant l'ordinateur. Il recevait pour consignes de s'asseoir confortablement, à distance raisonnable de l'écran. Ses mains et coudes devaient se trouver dans une position naturelle, pour que l'expérimentation se déroule le

plus sereinement possible. Une fois le participant confortablement installé, le calibrage de l'eye tracker était lancé. Si le calibrage était jugé satisfaisant, Gazepoint Analysis était caché sur le second écran, et l'écran d'accueil du jeu était présenté au participant. Après une brève explication finale sur la mission qu'il devait effectuer, le participant pouvait démarrer la session de jeu une fois l'expérimentatrice sortie de la pièce. À côté de lui était disposée une feuille avec les consignes de lancement, qui le guidait de l'écran d'accueil jusqu'à la mission choisie pour l'expérimentation. Une fois la pièce quittée et l'enregistrement écran lancé sur Gazepoint Remote, l'expérimentatrice pouvait regarder à distance le participant commencer la mission. Elle notait chaque élément intéressant de la session de jeu, qui pouvait être rediscuté lors de l'auto-confrontation qui allait suivre.

Une fois la mission terminée, l'expérimentatrice notait le temps de réalisation de la session de jeu, avant de rejoindre le participant dans la pièce.

### **1.3 Entretien**

Lorsque la session de jeu était terminée, un briefing était donné au participant concernant la suite, sans trop donner de détails. Des entretiens enregistrés étaient ensuite lancés, composés de deux parties.

Afin d'épaissir la valeur des données récoltées, l'aspect compréhensif a été pris en compte à travers une auto-confrontation (Latzko-Toth, Bonneau, & Millette, 2017). Grâce à l'enregistrement vidéo de la session de jeu capturée sur Gazepoint Analysis, les éléments importants, qui avaient été repérés au préalable lors du visionnage à distance, étaient partagés avec le participant. Il a été demandé au participant d'expliquer ces éléments avec ses propres mots, sans retenir son ressenti personnel de la performance qu'il venait de réaliser, et qu'il confrontait en direct avec son propre parcours oculaire. Cette méthode, utilisant l'eye-tracking pour l'enregistrement de ce parcours oculaire, laisse au participant la possibilité de moins se surcharger cognitivement que s'il était amené à commenter sa performance en direct (Renshaw, Stevens, & Denton, 2009). La verbalisation rétrospective permet d'obtenir des informa-

tions pertinentes auprès des utilisateurs si elle est effectuée directement après la réalisation des tâches (Eger, Ball, Stevens, & Dodd, 2007). Les zones d'intérêt visionnées à l'écran encourageaient la réflexion du participant sur pourquoi il avait réalisé telle ou telle action à cet endroit là, ou pour notifier des éléments de gameplay qui ont particulièrement attiré son attention. L'auto-confrontation était donc nécessaire pour éviter que l'analyse ne se focalise uniquement sur les zones observées par le participant, mais intègre également les intentions de celui-ci autour de la tâche performée (Eger et al., 2007).

La deuxième partie de l'entretien, qui suivait directement la confrontation, était semi-dirigée (Van Campenhout, Marquet, & Quivy, 2017). Elle se composait en une série de questions posées par l'intervieweuse, regroupées en trois thèmes, à travers une discussion. Le premier thème reprend des questions relatives à l'expérience de jeu, et à la définition du sentiment d'apprentissage du participant (cf. Annexe 1.1). Elles ont été construites pour répondre directement aux indicateurs des sous-hypothèses. Le deuxième thème rassemble des questions plus générales sur le niveau de programmation et l'approche que le participant a du jeu vidéo, dans le milieu éducatif ou non (cf. Annexe 1.2). Enfin, la troisième partie est composée de questions établies pour définir le profil du participant (cf. Annexe 1.3).

## **2 Choix des jeux**

Les deux jeux utilisés pour cette recherche ont été sélectionnés selon différents critères. L'objectif était de trouver des jeux à but éducatif, avec des notions de programmation facilement assimilables pour le temps donné d'une expérimentation. Celle-ci ne devait pas être trop longue, au risque de décourager le participant.

### **2.1 Algo-bot**

Le premier jeu qui a été retenu pour cette quasi-expérimentation est Algo-bot. Ce jeu éducatif fonctionne sous les concepts de la programmation et de

l'algorithmique. Grâce à des flèches et des déplacements, le joueur a pour mission de faire bouger le petit robot Algo-bot jusqu'à son objectif, à la fin de plusieurs niveaux, à savoir sauver le vaisseau Europa d'un tragique accident. Le robot aura parfois des tâches à réaliser, qui apportent des instructions supplémentaires que le joueur va devoir prendre en compte pour réussir les énigmes proposées (Figure 3). La difficulté est progressive : les premiers niveaux apportent les bases de la logique de programmation à travers des tutoriels.



FIGURE 3 – Écran de jeu du niveau 5 d'Algo-bot

Algo-bot (<https://www.algobot.be>) est un jeu indépendant, qui a été développé et édité par Fishing Cactus, sous la tutelle du Centre de compétence numérique Technobel. Il est disponible sur machine Microsoft Windows, Linux et Mac OS, mais également depuis l'année dernière sur tablette Android et iOS.

Ce jeu est considéré comme le jeu avec le niveau de difficulté « simple ». Les participants à notre expérimentation ont dû réaliser les neuf premiers niveaux du jeu, jusqu'à ce qu'une cinématique spéciale se déclenche, dans laquelle le robot fuit en catastrophe la salle qui est en train d'exploser. Dans ces

niveaux, le joueur progresse à travers plusieurs éléments importants du principe de programmation. Aucun niveau de programmation n'est requis pour jouer à ce jeu, car la mécanique utilisée demande uniquement de faire jouer sa logique informatique. En effet, le joueur ne code pas réellement les actions que le robot doit effectuer, mais il lui indique les opérations via des flèches et des commandes qu'il place dans une barre d'instructions, située en bas de l'écran (Figure 3). Le robot peut avancer, tourner dans les quatre directions, agir avec un interrupteur, porter un objet et le poser. Le principe de programmation sur lequel repose le jeu est la compréhension du séquençage d'instructions qui se suivent. À partir du quatrième niveau, le joueur voit la possibilité d'utiliser des fonctions, éléments essentiels en algorithmique. C'est un nouveau principe amené par le jeu, qui pousse la réflexion un cran plus loin en termes de logique informatique. Les instructions tournent parfois en boucle, ce qui permet au participant de comprendre le principe de récursivité. Les joueurs ont à disposition toutes ces actions, mais également un bouton stop, un bouton play, un bouton retour et un bouton indice en bas à droite de l'interface de jeu (Figure 3). Celui-ci a été repéré à l'avance comme zone d'intérêt : si le joueur venait à l'utiliser, c'est une étape qui serait discutée lors de l'entretien.

## **2.2 Minecraft : Education Edition**

Le second jeu sélectionné pour cette quasi-expérimentation est Minecraft : Education Edition (<https://education.minecraft.net>). Tout le monde connaît Minecraft : il s'agit d'un jeu de construction développé, édité et distribué par la société Mojang Studio. Depuis mai 2020, c'est le jeu vidéo le plus vendu de tous les temps : 200 millions d'exemplaires vendus sur toutes les plateformes, avec plus de 126 millions de joueurs actifs par mois (Froment, 2020). Le but premier du jeu est de laisser libre cours à son imagination en plaçant des blocs dans un univers uniquement composé de cubes. Le joueur peut construire des outils grâce aux ressources récoltées, et partir à l'aventure explorer le vaste monde dans lequel il se trouve. À côté de cet aspect solitaire, le côté multijoueur du jeu est largement développé et exploité par la commu-

nauté mondiale. Celle-ci est toujours très active, notamment sur Internet via les serveurs existants.

Minecraft propose également une version un peu différente de celle que l'on connaît, à savoir Minecraft : Education Edition. Cette version se veut un outil d'apprentissage : elle est composée de plus de 600 cours variés, sur différents domaines habituellement proposés dans les programmes scolaires. Ces cours sont prêts à l'emploi pour les enseignants désireux de soutenir des activités stimulant l'intérêt des élèves.

Plusieurs études ont déjà été réalisées, démontrant l'efficacité de l'apprentissage via Minecraft : Education Edition. En effet, dans une étude réalisée en 2019 par l'Université de Queensland, les étudiants ont montré un engagement plus prononcé au niveau de l'apprentissage après avoir réalisé un cours sur le jeu, mais également une augmentation de l'estime de soi et de leurs capacités en mathématiques (Dezuani & Macri, 2020). Le sentiment d'apprentissage qui en ressort est positif, avec une valeur ajoutée sur les matières reconnues plus difficiles habituellement par les élèves, à savoir les STEM. Dans une étude menée à Montréal auprès de 118 élèves du primaire, les compétences du 21<sup>e</sup> siècle sont mises en avant dans les résultats : l'engagement, la créativité, la collaboration et la résolution de problèmes (Karsenti & Bugmann, 2018). Ils en ont conclu que l'expérience sur le jeu avait boosté la motivation intrinsèque des élèves, mais aussi que la créativité induite à partir du jeu augmentait le sentiment d'apprentissage des matières comme les mathématiques, l'informatique ou encore la géographie. Les élèves ont également développé encore une fois un meilleur engagement et une meilleure estime de soi.

En Belgique, la volonté des acteurs économiques du numérique de promouvoir l'innovation digitale dans les écoles se fait sentir. Il est réellement urgent de réinventer les pratiques d'apprentissage (Jenart, 2022). La création à Mons de la toute nouvelle Minecraft Education Academy a été réalisée en ce sens. Technocité et Microsoft sont à l'origine du projet, ayant pour objectif d'initier les enseignants et leurs élèves aux outils proposés par le jeu. « *Ainsi, les enseignants qui seront formés à la Minecraft Education Academy pour-*

*ront déployer cette solution dans leurs écoles et tirer un avantage éducatif d'un des jeux vidéo les plus populaires au monde auprès des jeunes. Cette expérience ludique et immersive dans un environnement virtuel va favoriser l'acquisition de réflexes numériques et qui sait, susciter des vocations chez les élèves, conclut Willy Borsus, ministre du numérique. » (Ducobu, 2021).*

En ce sens, Minecraft : Education Edition a constitué dans cette expérimentation les deux niveaux de difficulté restants, à savoir « moyen », avec la leçon *Hours of Code 2020 (Inclusion)*, et « soutenu », avec la leçon *Hours of Code 2021 (TimeCraft)*. Ces leçons, toutes les deux implémentées dans le catalogue des exercices d'informatique du jeu, proposent au début de l'exercice de choisir entre la programmation en blocs ou en Python. Quand le participant s'est vu attribuer le niveau moyen, il jouait à la version 2020 avec la programmation en blocs. Pour le niveau soutenu, le joueur était sur la version 2021, avec la programmation en Python. Cette différence de « langage », définissant notre type de jeu, permettait d'offrir aux participants une appréhension et une réflexion différente du jeu, le langage en blocs était plus intuitif que le codage systématique et syntaxique d'un langage de programmation traditionnel comme Python.

#### Minecraft : Education Edition - Hours of Code 2020 (Inclusion)

Hours of Code 2020 (Inclusion) est une leçon d'une durée de plus ou moins une heure, mettant en scène l'histoire des villageois et des illageois. Ces deux villages habitent dans le même espace, mais n'ont presque aucun contact. Le joueur est amené à utiliser ses compétences de codage pour tenter de rapprocher les deux villages, au travers de 6 exercices répartis dans ce monde immersif. Au niveau de l'histoire, la compassion et l'empathie sont les maîtres mots mis en avant dans cette leçon.

Après s'être retrouvé plongé dans le monde, le joueur a droit à un petit tutoriel d'utilisation de la machine pour coder, le *Code Builder* (Figure 4). C'est l'outil utilisé dans Minecraft : Education Edition pour programmer les instructions à faire faire au petit robot présent dans le jeu, appelé l'agent. Une

fois le tutoriel réussi, le joueur se lance librement dans les deux villages à la recherche des missions à réaliser. Celles-ci sont toutes différentes, mais sont à chaque fois accompagnées d'un contexte, et d'instructions claires lorsque le Code Builder est activé. Dans deux d'entre elles, le jeu permet d'utiliser les boucles pour réaliser son code plus efficacement. C'est une mécanique qui a été mise en évidence lors des quasi-expérimentations : elle permet de voir si le joueur essaie de comprendre les outils mis à sa disposition, et la logique qui se cache derrière. À chaque réalisation d'une mission, le jeu indique distinctement l'avancement de celles-ci, et renvoie (facultativement) le joueur dans la pièce centrale où il a démarré la leçon, pour voir la croissance progressive d'un arbre. Celui-ci symbolise distinctement la progression de la leçon.



FIGURE 4 – Code Builder version blocs, Hours of Code 2020 (Inclusion)

#### Minecraft : Education Edition - Hours of Code 2021 (Timecraft)

Hours of Code 2021 (TimeCraft) ressemble, sur le principe, à sa prédécesseuse, mais avec un contexte différent. La leçon transporte le joueur dans un voyage dans le temps et l'espace. Il aura pour mission, grâce à ses talents en codage et son fidèle agent, de réparer les erreurs présentes dans des grands moments de l'Histoire et sauver l'avenir de l'Humanité.

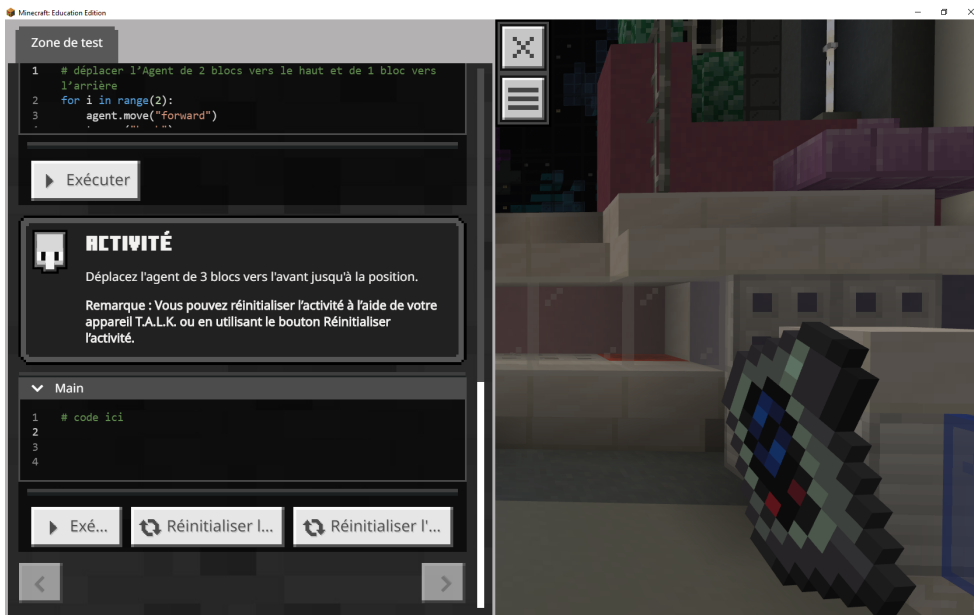


FIGURE 5 – Code Builder version Python, Hours of Code 2021 (Timecraft)

Après une brève cinématique, le joueur est amené à réaliser un très court tutoriel sur les commandes de base de Minecraft (déplacement, saut, orientation, interaction) avant d'arriver devant le PNJ<sup>4</sup> principal. Celui-ci lui explique le contexte, et le lance sur le tutoriel d'utilisation du Code Builder. Une fois terminé, le joueur est amené à choisir, sur l'ordinateur de la pièce centrale, la première mission sur les trois qu'il va être amené à réaliser. Cette première mission est imposée, contrairement aux deux suivantes qui sont libres. Chaque mission représente un événement historique majeur (comme par exemple la construction de la Muraille de Chine). Une fois sélectionnée, le joueur se lance dans la machine à remonter dans le temps et découvre le tableau de jeu à élucider. Tout comme la version 2020, les consignes données sont claires, et le parcours est linéaire. Ici, comme le joueur code en Python et non en blocs, il est libre d'utiliser les commandes qu'il veut (Figure 5). Des exemples sont donnés à chaque exercice, ce qui permet au joueur de les utiliser et d'apprendre par lui-même comment les commandes présentées fonctionnent. À chaque mission, le joueur se voit chercher une salle secrète où il doit choisir l'agent qui pose tous ces problèmes dans le passé. À la fin des trois missions, le joueur doit sélectionner le « méchant », et la leçon se termine par sa fuite.

---

4. Personnage Non Joueur.

### 3 Choix des participants

Les neuf participants ont été sélectionnés parmi les étudiants de l’UCLouvain, afin de permettre un critère stable sur plusieurs points. En premier lieu, l’âge des participants se situait dans une fourchette rapprochée, entre 22 et 26 ans. Ensuite, les compétences développées lors d’un parcours universitaire permettaient aux participants de faire preuve d’un sens critique nécessaire sur leur propre performance, essentiel lors des entretiens d’auto-confrontation, et d’un niveau d’anglais suffisant pour se sentir à l’aise sur les deux des trois jeux présentant quelques consignes en anglais. Enfin, le fait de choisir des étudiants de l’UCLouvain permettait une facilité de recrutement parmi mes connaissances.

Comme expliqué précédemment, la connaissance préalable en programmation a constitué le critère principal pour définir le type du joueur, et donc la recherche du participant. Il fallait trouver trois étudiants n’ayant jamais programmé (catégorie débutant), trois étudiants ayant des connaissances de base en programmation et/ou programmeur occasionnel ; dans le cadre d’un cours par exemple (catégorie intermédiaire), et enfin, trois étudiants possédant des connaissances approfondies et une maîtrise en programmation (catégorie confirmé). Afin d’anonymiser les participants, une nomenclature spécifique a été appliquée (Figure 6). Celle-ci combine à la fois le niveau en programmation du participant, mais également le jeu auquel il a été attribué. Cette méthode permettra dans la suite de ce rapport d’identifier directement les individus personnellement.

	Algo-bot	MEE Blocs	MEE Python
Débutant	N1-AB	N1-M2020	N1-M2021
Intermédiaire	N2-AB	N2-M2020	N2-M2021
Confirmé	N3-AB	N3-M2020	N3-M2021

FIGURE 6 – Tableau de la nomenclature utilisée pour identifier les participants

## Profil des participants

Les participants de niveau débutant étudient tous les trois dans le domaine des sciences humaines et n'ont jamais touché à la programmation. N1-M2020 et N1-M2021 avaient déjà joué à des jeux vidéo pour apprendre, comme la logique ou l'histoire. N1-AB est le participant qui joue le plus fréquemment parmi les trois et indiquait avoir développé certaines compétences à travers cette pratique (comme l'apprentissage de l'anglais, améliorer ses réflexes ou encore sa précision).

Les participants de niveau intermédiaire sont trois étudiants en master en sciences et technologies de l'information et de la communication. Ce master propose une petite base en programmation, notamment via l'apprentissage des principes de base : les trois participants ont d'ailleurs défini leur niveau de programmation comme étant « débutant » ou « limité ». N2-AB, N2-M2020 et N2-M2021 qualifient leur niveau d'aisance sur un ordinateur de « bon », comparé aux participants de niveau débutant qui se situaient à un niveau « moyen ». Le niveau d'aisance sur un ordinateur peut être une variable à prendre en compte dans la performance de jeu, et la réflexion que le participant peut en tirer. Enfin, les trois participants de niveau intermédiaire expliquent avoir déjà joué à des jeux vidéo pour apprendre, notamment lors d'un des cours du master.

Les participants de niveau confirmé sont des étudiants de master en sciences informatiques. Ils considèrent leur niveau d'aisance sur un ordinateur et leur niveau de programmation comme étant « très bon ». En ce qui concerne la fréquence de jeu, N3-AB ne joue jamais aux jeux vidéo, contrairement aux deux autres participants qui sont plutôt très réguliers dans la pratique. Parallèlement aux réponses des participants de niveau débutant, deux informaticiens ont déjà joué à des jeux pour apprendre la logique ou la stratégie, et le dernier explique avoir quand même développé ses réflexes et sa précision grâce aux jeux vidéo (non éducatifs) auxquels il a joué.

Un détail doit être notifié avant de poursuivre dans la description des résultats. Les six participants qui se sont vu attribuer leur expérience de jeu

sur une des deux leçons de Minecraft : Education Edition avaient déjà joué à Minecraft auparavant. La maîtrise relative du gameplay a sans aucun doute influencé la performance de jeu, comparé aux participants qui ont joué sur Algo-bot, qui eux ne connaissaient pas du tout les mécaniques et principes du jeu. Les deux participants de niveau intermédiaire avaient déjà utilisé le Code Builder, outil propre dans la version éducative de Minecraft, contrairement à N3-M2020 et N3-M2021 qui l'ont découvert lors de leur expérimentation. Cependant, aucun des participants n'avait joué à la leçon qui lui a été attribuée, pour le bien de cette recherche. C'est sur cet aspect que nous allons nous concentrer, mais il était important d'expliquer le contexte au préalable. Plusieurs études montrent en effet que les joueurs ayant une expérience avec le jeu vidéo étudié ont une approche plus objective du jeu, savent quoi faire et regarder, réalisent les tâches plus vite et ont un meilleur temps de réaction (Almeida et al., 2011).

# Résultats

Les neuf expérimentations se sont déroulées sans accroche majeure. Elles ont été réalisées durant deux semaines. Un seul élément est cependant à signaler : les participants N2-M2021 et N3-M2021, qui ont été les deux premiers à réaliser l'expérimentation, n'ont pas pu être confrontés à leur performance sur le logiciel Gazepoint Analysis dû à un problème de fichier. Nous avons alors visionné les vidéos brutes du jeu pour cette partie, sans le tracé du regard. Les notes prises lors de l'observation externe ont permis de pallier ce désagrément. Le problème de fichier a été résolu pour les sept autres expérimentations.

## 1 Expérience de jeu

Chaque joueur a réalisé un parcours différent dans le jeu. Certains ont rencontré des difficultés sur certaines missions, d'autres ont utilisé des outils, comme par exemple les boutons donnant des indices, pour pouvoir avancer dans leurs tâches. Les réflexions qui sont ressorties lors de l'auto-confrontation ont été enrichissantes, permettant de mieux comprendre le ressenti du joueur face à sa performance. Nous allons dans cette section décrire les sessions de jeu et comparer les éléments intéressants à relever. Afin de rester pertinent, les performances des joueurs seront comparées uniquement entre les trois joueurs ayant réalisé la même leçon.

### 1.1 Algo-bot

Pour rappel, Algo-bot propose aux joueurs neuf niveaux à résoudre à l'aide d'une barre d'instructions. Ce premier type de gameplay est considéré comme étant le niveau « simple ».

Les trois participants ont réalisé leur session de jeu entre 26 et 37 minutes. Nous avons observé que N1-AB et N3-AB ont évolué de plus en plus vite dans

les premiers niveaux proposés. Concernant l'aisance sur le jeu, N3-AB a semblé être plus serein que les deux premiers participants en ayant comme objectif d'aller le plus vite possible. Il n'a eu aucun mal à démarrer ses tâches, en comprenant presque instantanément ce qu'il devait faire. Similairement, N2-AB nous a indiqué vouloir réaliser l'expérimentation assez rapidement, dans un souci de performance. Il a expliqué manquer de logique et être stressé par le temps qu'il mettait pour réaliser ses niveaux. Nous avons observé à ce moment-là qu'effectivement, à la place de lire les dialogues introduisant le niveau suivant, le participant analysait déjà à l'avance l'exercice pour gagner du temps.

Les trois joueurs ont porté de l'intérêt aux exercices qu'ils devaient réaliser, mais avec une nuance différente pour chacun. N1-AB et N2-AB ont expliqué être contents lorsqu'ils réussissaient un niveau du premier coup. De plus, N1-AB et N3-AB ont exprimé une volonté de refaire les niveaux pour lesquels le score n'était pas optimal.

Nous tenions à ajouter une remarque concernant la performance de N2-AB : l'ordinateur sur lequel nous observions le participant s'est éteint lors de la session de jeu, entraînant, durant un moment, l'interruption de la réalisation de l'exercice. Le temps que le participant a mis pour réaliser l'ensemble des niveaux a donc été rallongé. Enfin, comme nous le verrons plus loin dans l'auto-confrontation, le participant a expliqué avoir été frustré que nous l'arrêtions en pleine concentration. Il a également mentionné que cette coupure lui a rappelé qu'il participait à une expérimentation, ce qui a eu un effet sur son état d'esprit pour la fin de l'exercice.

Chaque participant a eu des difficultés sur des niveaux du jeu différents. N1-AB et N2-AB ont fait plusieurs erreurs en inversant leur gauche et leur droite. De plus, les trois joueurs ont émis une même remarque concernant les pictogrammes des actions « poser » et « prendre » qui apparaissent dès le cinquième niveau. Ils ont fait tous les trois la même erreur en inversant les actions, jugeant les pictogrammes comme étant difficilement différenciables.

N1-AB a eu des difficultés concernant le niveau 7 du jeu : « *J'ai dû vraiment me concentrer et du coup ce niveau-là était vraiment fatiguant pour moi, visuellement et pour construire le truc. [...] J'étais un peu blasé du niveau parce que vraiment, au niveau de ma logique, ça a été vraiment compliqué* ». Le participant a perdu sa concentration au fur et à mesure des erreurs. Il prenait moins le temps de vérifier son code avant de le lancer car il perdait patience. En parlant du niveau 8 : « *Celui-là a été plus simple et ça m'a instantanément calmé et remis dans le jeu. Je le trouvais super agréable comme petit jeu quand même. Je suis content de pas avoir fini sur ce niveau-là [le 7] parce que je pense que j'aurais été trop gavé* ».

Pour N3-AB, le dernier niveau s'est révélé être le plus compliqué. Le participant a commencé à perdre patience. Il a enclenché une série d'essai-erreurs<sup>5</sup> en ne prenant plus le temps de vérifier le parcours du robot à chaque fois. Malgré cette frustration, le participant a expliqué qu'il voyait l'exercice comme un puzzle qu'il fallait résoudre. Il a avoué qu'il voulait aller trop vite, et qu'il n'a donc pas pris le temps d'analyser la situation.

Pour contrer ces difficultés et avancer dans le jeu, plusieurs stratégies ont été mises en place. Nous avons noté que N1-AB et N3-AB observaient en alternance les éléments du jeu et la barre d'instructions pour comprendre où se situaient leurs erreurs potentielles. Intuitivement, N1-AB a eu recours à l'essai-erreur pour comprendre les nouvelles instructions qui se présentaient à lui. Il découpait son problème en plusieurs parties en prenant le temps de vérifier à chaque étape si ce qu'il avait fait fonctionnait correctement. N2-AB a expliqué faire de l'essai-erreur uniquement lorsqu'il était moins à l'aise sur un niveau.

À partir du niveau 4, les fonctions sont intégrées au gameplay. N1-AB a expliqué : « *J'ai eu plus facile dans les niveaux où la fonction devait être mise dès le départ, en première, parce que je la voyais directement. Quand elle*

---

5. L'essai-erreur (*trial and error*) dans le jeu vidéo est défini comme étant l'absence de stratégie systématique en jouant à un jeu. Cela implique des actions et des réactions aux circonstances, aux conséquences et aux feedbacks obtenus dans le cadre du jeu. C'est en observant et en participant activement au jeu que le joueur établit au fur et à mesure les connaissances nécessaires pour savoir comment jouer (Dempsey & Others, 1996).

*arrivait au milieu du schéma, c'est vrai que j'avais un peu plus de mal à la placer. Je voyais pas où elle commençait et où elle s'arrêtait* ». Pour pallier cette difficulté, N1-AB a opté pour la solution de construire son code sans la fonction, et ensuite remplacer les patterns d'instructions qui en ressortaient par une fonction. N2-AB a adopté la même stratégie sur quelques niveaux. Contrairement aux deux précédents, N3-AB a quant à lui directement implémenté sa fonction dans son code : *« J'essaie de voir une espèce de récursion dans le schéma à faire. [...] Je cherche à mettre le plus d'éléments possibles dans la fonction »*.

Au niveau des outils proposés par le jeu, seul N1-AB a utilisé le bouton d'aide pour comprendre comment le *gameplay* fonctionnait lors de la première mission. N2-AB a eu le même problème, mais après avoir cherché un court instant, il a trouvé le bouton d'arrêt et a pu commencer à construire son code.

Enfin, N2-AB a partagé avec nous une réflexion concernant l'interruption de son jeu : *« Je crois qu'à un moment donné [...] j'ai dû compter avec mes doigts. Genre j'ai fait comme ça sur l'écran parce que j'avais du mal. Il fallait que je simule le bonhomme qui avance »*. Cette remarque confirme le fait qu'il a dû se reconcentrer après avoir été interrompu.

## **1.2 Hours of Code 2020 (Inclusion)**

Le deuxième jeu étudié est une leçon du jeu Minecraft : Education Edition. Hours of Code 2020 (Inclusion) est notre type de jeu proposant aux joueurs d'utiliser la programmation en blocs pour réussir les six missions présentées par le scénario.

Les participants ont réalisé cette leçon de niveau « moyen » entre 23 et 40 minutes. N1-M2020 a mis un certain temps à comprendre le lien entre le Code Builder et l'agent. Ensuite, il a réalisé les missions de plus en plus vite. N2-M2020 a été très à l'aise avec les outils proposés et a démarré les missions rapidement. Enfin, N3-M2020 lisait bien les consignes et prenait son temps au début du jeu, puis a accéléré pour aller de plus en plus vite. Les trois

joueurs ont émis la même remarque sur l'arbre qui pousse au fur et à mesure des missions, symbolisant la progression dans le jeu : venir le voir à chaque fin de mission n'était pas nécessaire.

Concernant l'intérêt porté aux exercices, N1-M2020 a lu les instructions à la lettre tout au long de la leçon. Après avoir réussi la première mission, il a exprimé vouloir faire mieux pour la suite. Il est resté concentré en lisant l'ensemble des consignes, et pour la deuxième mission, a pris le temps de compter le nombre de blocs à poser avant de réaliser son code. N2-M2020 a indiqué être satisfait d'avoir réussi un exercice du premier coup : « *Surtout que je suis super nul en programmation. Et ce genre d'exercice avec des blocs, je trouve ça hyper sympa parce que tu vois directement l'agent avancer* ». Enfin, N3-M2020 a avoué vouloir aller trop vite car les niveaux lui paraissaient trop simples.

Au niveau des difficultés rencontrées, les trois participants ont soulevé des éléments différents. N1-M2020 se demandait ce qu'il devait faire au début. Il a dû apprendre par lui-même que les blocs de code pouvaient s'emboîter pour combiner les actions en une fois. Le joueur a avoué, lorsque nous observions qu'il éprouvait des difficultés à recentrer son agent, qu'il aurait souhaité ré-initialiser la mission. Il ne savait pas comment le faire et a alors continué en essayant de réparer ses erreurs manuellement.

Ensuite, N2-M2020 a exprimé vouloir aller trop vite sur deux missions, entraînant de nombreuses erreurs. Il en était conscient, et une fois la mission réussie, il a exprimé avoir été soulagé d'avoir enfin réussi l'exercice.

Enfin, arrivé sur le cinquième exercice, N3-M2020 a expliqué avoir eu un peu plus de mal à résoudre ce problème. En effet, il a utilisé ici des boucles imbriquées dans un souci de vouloir aller plus vite et être plus efficace. Celles-ci lui ont finalement porté préjudice car elles comportaient une erreur de logique qu'il a mis du temps à identifier.

Les joueurs ont utilisé plusieurs stratégies durant leur expérimentation.

N1-M2020 a expliqué n'avoir pas tout de suite compris comment la boucle fonctionnait, mais il a pris le réflexe de devoir l'utiliser. Il a fonctionné par essai-erreur pour tenter de comprendre dans quel sens il fallait aller lors d'un niveau plus compliqué, avec pour objectif d'être le plus efficace possible : *« C'est pour ça que je faisais par segment tu vois. Je me dis que si le segment foire, je peux le voir facilement, et je peux le reprendre pour être sûr de le peaufiner correctement »*. Il a également ajouté que les messages d'erreurs présents dans le chat l'ont aidé à comprendre qu'il y avait un problème quelque part.

La stratégie établie par N2-M2020 consistait à découper l'exercice général en sous-problèmes, pour ensuite les exécuter tous un par un afin d'avancer correctement en corrigeant uniquement ce qui était nécessaire : *« C'est parce que je savais pas si on pouvait revenir à 0. Et du coup, par sûreté, je me suis dit que je vais faire étape par étape, comme ça on est sûr que le truc soit bien fait et qu'il n'y a pas de problème »*. Pour tenter de comprendre les consignes et comment réussir un niveau difficile, N2-M2020 n'a pas hésité à parler au PNJ et lire l'indice proposé dans le Code Builder.

Enfin, N3-M2020 a adopté immédiatement des réflexes qui lui ont permis d'agir assez rapidement. Il se positionnait systématiquement correctement afin de voir clairement le schéma de jeu sur lequel il allait devoir réfléchir. Il a essayé plusieurs fois des commandes avant de les corriger pour comprendre comment l'agent devait avancer.

### **1.3 Hours of Code 2021 (Timecraft)**

Enfin, pour rappel, notre dernière leçon utilisée dans le cadre de cette recherche, intitulée Hours of Code 2021 (Timecraft), utilise un langage de programmation, Python, pour résoudre les trois missions proposées par l'histoire. Ce jeu est notre troisième type de gameplay employé lors de cette recherche.

Cette leçon a été réalisée par les joueurs entre 21 et 45 minutes. N1-M2021 a perdu beaucoup de temps entre les niveaux car il prenait le temps d'explorer l'environnement de jeu et de tester dans les exercices les possibilités offertes

par le gameplay et le Code Buidler. Un point commun entre N2-M2021 et N3-M2021 a été le fait de vouloir aller trop vite, entraînant des erreurs. N3-M2021 a même expliqué avoir réalisé le tutoriel sans même comprendre que le code servait à faire avancer l'agent, confirmant le fait que le joueur cherchait à vouloir aller le plus vite possible. De plus, les deux joueurs ont expliqué avoir trouvé les dialogues du PNJ principal trop longs, ralentissant leur progression dans le jeu.

N1-M2021 est sans doute le joueur qui a porté le plus d'intérêt à la leçon. Il a expérimenté beaucoup de choses en étant très curieux de l'environnement dans lequel il se trouvait. Nous avons relevé qu'il avait assimilé la syntaxe et le fonctionnement de Python dès la première mission, malgré le fait qu'il n'avait jamais codé. Le joueur possédait de bons réflexes de joueur de jeu vidéo, comme par exemple l'utilisation du *chat* pour noter les éléments qu'il jugeait importants de se souvenir. Concernant les instructions du jeu, N1-M2021 a toujours bien pris le temps de les lire, N2-M2021 y a fait attention uniquement au début et à la fin de son expérience, et N3-M2021 a avoué les avoir lues uniquement au début.

En observant sa performance lors du deuxième exercice, qui lui a semblé plus difficile, N2-2021 a expliqué : « *C'est là que je me suis pris la tête parce que j'avais juste lu en diagonale et j'avais pas vu qu'il fallait positionner son agent au-dessus du bloc alors que je l'ai pas fait, ça m'a saoulé. Donc voilà, je prends vraiment le temps de lire toutes les commandes* ». Cependant, il a refait la même erreur plus tard en voulant utiliser une boucle dans son code, sans prendre le temps de lire complètement les consignes. Quant à N3-M2021, il a expliqué avoir eu du mal avec son dernier exercice car il ne comprenait pas comment il devait placer ses blocs. Il a pris le temps d'écrire une fonction car il pensait que les actions allaient devoir se répéter.

Au niveau des stratégies mises en place par les joueurs, nous avons retrouvé de l'essai-erreur chez N1-M2021 et chez N3-M2021. De son côté, N2-

M2021 a pris le temps d'analyser les schémas de jeu, en comptant les blocs que l'agent devait parcourir, avant de construire son code dans le Code Builder. Lorsqu'il se trompait, il corrigeait son erreur en découpant le problème en plusieurs étapes.

N3-M2021 a fait appel au PNJ pour relire les instructions à suivre dans l'exercice qui lui a posé problème. Il n'a pas voulu utiliser l'indice car il le considérait comme « *étant de la triche* ». Un élément intéressant à noter est le fait que N3-M2021 ait retrouvé instantanément ses réflexes de programmeur, en utilisant des fonctions et les commentaires. Il a été dès le début très confiant avec l'interface : « *Ce sont les mêmes raccourcis que dans VSCode*<sup>6</sup> ».

Enfin, nous avons relevé une similitude chez N1-M2021 et N2-M2021. Les deux joueurs ont préféré taper leur code directement au lieu de copier-coller les exemples donnés. N2-M2021 a expliqué : « *Je préfère parce qu'en fait j'ai peur qu'en copiant j'oublie de modifier un détail important qui casse le code. [...] Taper, ça permet de se ré-approprier le truc* ».

## 1.4 Conclusion des auto-confrontations

Plusieurs similitudes émanent de ces auto-confrontations, que cela soit entre les types de jeux ou entre les niveaux des joueurs. Tout d'abord, les débutants ont réalisé leur expérimentation dans un temps croissant en fonction du type de jeu. Cependant, nous avons pu voir que la prise en main des différents outils de programmation (barre d'instructions, blocs et Python) s'est faite de manière progressive chez les trois joueurs. Par ailleurs, pour les joueurs qui ont réalisé leur session sur l'une des deux leçons Minecraft : Education Edition, nous pouvons noter que les joueurs confirmés ont été plus rapides que les joueurs débutants. Cependant, pour l'ensemble des participants, l'évolution dans le jeu a été de plus en plus rapide. Ils devaient tous prendre le temps de comprendre comment le jeu fonctionnait (même si les joueurs de Minecraft : Education Edition y avaient déjà tous joué) pour ensuite appliquer les mêmes stratégies jugées comme étant efficaces aux exercices suivants.

---

6. Visual Studio Code est un éditeur de code redéfini et optimisé pour la création et le débogage d'applications Web et Cloud (<https://code.visualstudio.com/>).

Ensuite, nous avons relevé qu'il n'y avait pas que les débutants qui éprouvaient des difficultés. Les joueurs confirmés ont également eu des difficultés pour réaliser certains niveaux. Ces trois joueurs (et certains joueurs intermédiaires) ont voulu aller trop vite. Ils réalisaient alors des erreurs évitables, en ne prenant pas le temps de lire l'ensemble des consignes. Cette prise de confiance trop rapide peut s'expliquer par le fait que les joueurs ayant l'habitude de programmer ont jugé les jeux comme étant simples, et ont alors compté sur leurs réflexes pour résoudre les différentes énigmes. En parallèle, nous avons pu observer que les joueurs débutants (et certains joueurs intermédiaires) avaient moins de réserves à utiliser les boutons d'aide proposés dans les jeux, mais également à prendre le temps d'explorer l'environnement de jeu.

Un élément intéressant à soulever concerne la stratégie mise en place par l'ensemble des joueurs, à savoir l'essai-erreur. Chacun, d'une façon ou d'une autre, a pris le temps de déconstruire son problème en plusieurs étapes pour comprendre comment la logique fonctionnait, mais surtout pour identifier l'erreur et la corriger. La majorité des joueurs analysait les niveaux à l'écran en alternant leur regard entre le code qu'ils étaient en train de construire et le schéma de jeu proposant l'exercice à résoudre. De plus, tous les joueurs étaient globalement satisfaits lorsqu'ils réussissaient un exercice du premier coup.

## **2 Entretien**

Une fois l'auto-confrontation terminée, un entretien semi-dirigé a été mené. L'objectif était de partager une discussion avec chaque participant, afin qu'il développe le plus fidèlement possible son expérience et ses réponses aux questions. Cet entretien était divisé en trois thèmes. Nous allons, dans cette section, analyser les résultats des réponses aux neuf questions du premier thème. Celles-ci avaient pour objectif de répondre aux indicateurs des sous-

hypothèses que nous avons établies pour cette recherche (cf. Annexe 1.1). Les deux thèmes suivants, répondant à des questions plus générales sur l'approche du participant sur le jeu vidéo (cf. Annexe 1.2) et son profil (cf. Annexe 1.3), ont été analysés dans la section « Choix des participants ».

## 2.1 Questions thématiques

### 1. « Sur une échelle de 1 à 10, comment noterais-tu ta motivation à apprendre à programmer suite à cette session de jeu ? »

Cette première question, qui demandait aux participants de noter leur motivation à apprendre à programmer sur une échelle de 1 à 10, peut être résumée en un graphique (Figure 7).

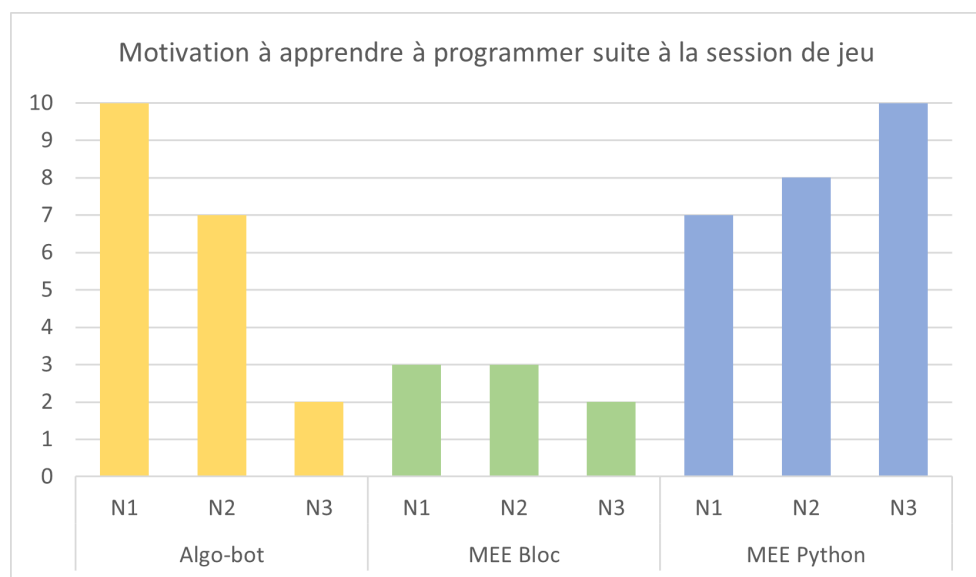


FIGURE 7 – Question n°1

Les trois participants ayant joué à Algo-bot ont attribué une moyenne de 6.33 sur 10. Pour N1-AB, le jeu était agréable, intuitif, et il était faisable malgré les doutes qu'il pouvait avoir avant de commencer. N2-AB a expliqué qu'il avait trouvé satisfaisant de voir que ses exercices fonctionnaient. Enfin, N3-AB a trouvé le jeu amusant, mais ne s'est pas senti particulièrement motivé par la session de jeu.

Hours of Code 2020 (Inclusion) a reçu une moyenne de 2.66 sur 10. Les

joueurs ne se sont pas senti motivés par cette leçon. N1-M2020 ne voit pas l'utilité d'apprendre davantage la programmation, et N2-M2020 a expliqué que la motivation qu'il éprouvait à apprendre à programmer n'avait rien à voir avec le jeu. Cependant, N3-M2020 a indiqué pouvoir donner la note de 9 au lieu de 2 s'il s'imaginait n'avoir jamais programmé. Cette réflexion remonterait la moyenne à 5.

Les joueurs qui ont réalisé la leçon Hours of Code 2021 (Timecraft) lui ont donné une moyenne de 8.33 sur 10. N1-M2021 a expliqué que le jeu lui avait donné envie d'apprendre à programmer car il avait trouvé satisfaisant de voir que son code fonctionnait : « *En tout cas, avec ce que montre le niveau, ça a l'air plus accessible que ce que je ne l'aurais cru* ». N2-M2021 a trouvé le jeu très ludique. Enfin, N3-M2021 a trouvé le jeu était très motivant pour apprendre à programmer.

Nous pouvons observer sur ce graphique que N3-AB et N3-M2020 ont mis une note relativement basse : ils expliquent qu'ils étaient déjà intéressés par la programmation avant de jouer. Le critère de sélection des participants peut justifier cette réflexion. Cependant, N3-M2021 a eu une réponse très différente. Il a interprété la question avec un avis extérieur à son expérience personnelle, en répondant comme s'il n'avait jamais programmé.

Nous pouvons établir que Hours of Code 2021 (Timecraft) semble être le jeu le plus motivant en général pour apprendre à programmer. Cependant, Algo-bot s'avère être le jeu le plus motivant pour les joueurs débutant la programmation.

## 2. « Sur une échelle de 1 à 10, comment noterais-tu ton impression d'avoir appris un peu plus à programmer suite à cette session de jeu ? »

Cette deuxième question comprenait également une échelle de 1 à 10, nous permettant de résumer les notes attribuées en un graphique (Figure 8).

Algo-bot s'est vu attribuer une moyenne de 2.33 sur 10. Ce résultat se traduit par les commentaires des trois joueurs. N1-AB a expliqué avoir eu

l'impression de jouer plus que de n'avoir réellement appris à programmer. De plus, N2-AB indique n'avoir rien appris de plus par rapport à ses cours. Néanmoins, N3-AB a indiqué que le jeu pourrait être intéressant pour quelqu'un qui n'a jamais programmé une fois qu'il comprend la logique qu'il y a derrière les exercices.

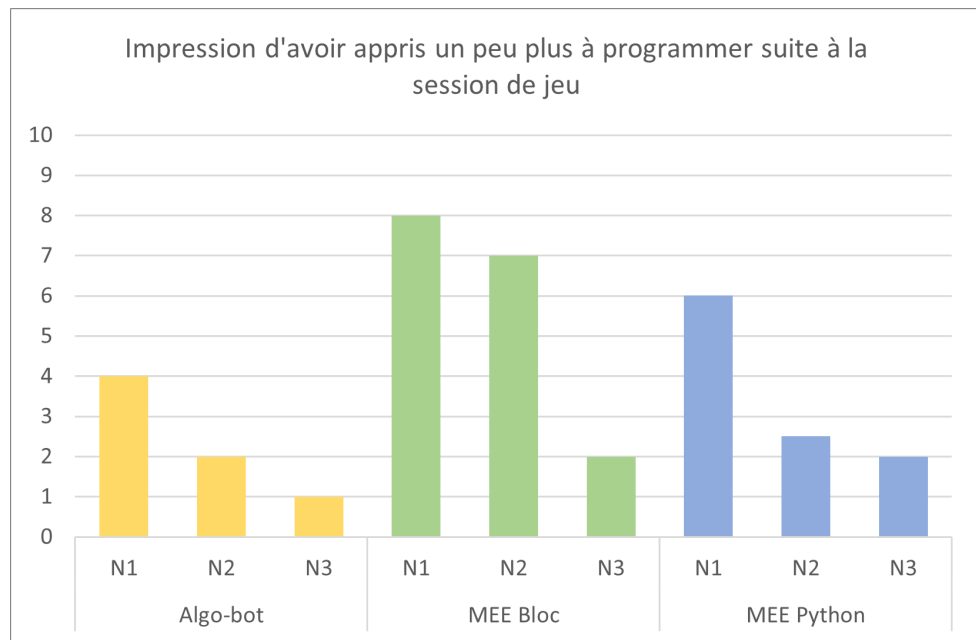


FIGURE 8 – Question n°2

Pour Hours of Code 2020 (Inclusion), la moyenne obtenue est de 5.66 sur 10. N1-M2020 a expliqué qu'une fois qu'il avait compris la logique derrière les missions, il était lancé dans son aventure. N2-M2020 a pu retravailler sa logique avec le jeu. Enfin, N3-M2020 a trouvé que la leçon était intéressante pour quelqu'un qui n'a jamais programmé afin d'apprendre la logique. S'il avait été dans ce cas-là, il aurait attribué la note de 9.

Les joueurs ayant réalisé la leçon Hours of Code 2021 (Timecraft) lui ont donné une moyenne de 3.5 sur 10. Mis à part N1-M2021 qui a expliqué avoir acquis les bases nécessaires grâce au jeu, N2-M2021 et N3-M2021 ont indiqué avoir déjà les connaissances suffisantes avant de réaliser leurs tâches.

Dès lors, Hours of Code 2020 (Inclusion) se révèle être le jeu qui donne l'impression d'avoir acquis de nouvelles connaissances en programmation.

Cependant, il faut nuancer les résultats. Plusieurs joueurs expliquent qu'ils ont pu, grâce au jeu, retravailler leur logique. Il y a une tendance décroissante des notes en fonction du niveau du joueur. En effet, plus le joueur est débutant dans la programmation, plus il aura l'impression d'avoir appris à programmer.

3. « Sur une échelle de 1 à 10, à quel point t'es-tu senti performant en voyant que tu évoluais dans le jeu ? »

Cette troisième question demandait aux participants de noter sur une échelle de 1 à 10 leur sentiment de performance en voyant qu'ils évoluaient dans le jeu. Nous avons également construit un graphique, résumant ainsi les notes attribuées lors de l'entretien (Figure 9).

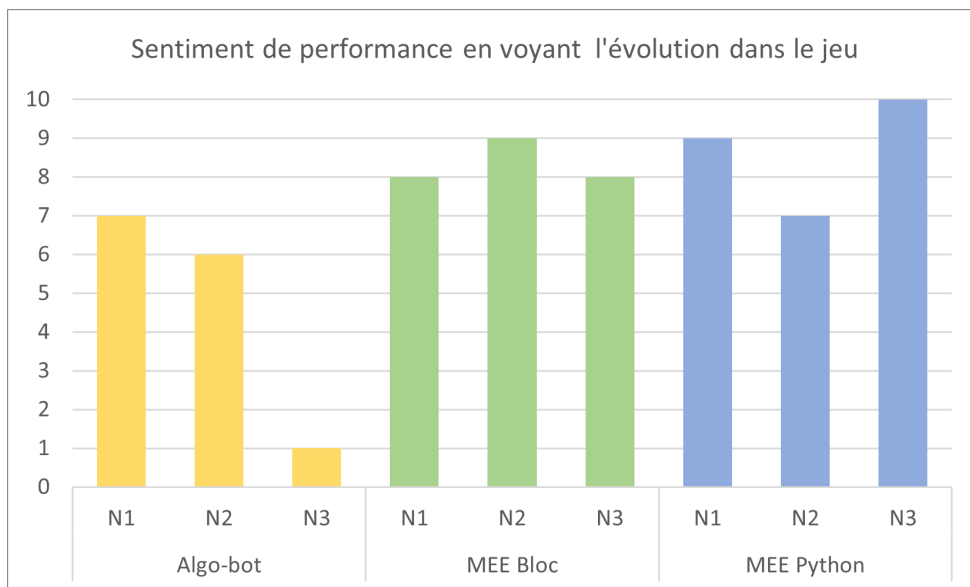


FIGURE 9 – Question n°3

Algo-bot a reçu une moyenne de 4.66 sur 10 pour cette question. N1-AB a senti que sa session de jeu se déroulait bien lorsqu'il avait le bon score. N2-AB a expliqué avoir été content de finir et n'a pas vraiment senti une évolution dans le jeu. Quant à N3-AB, il a mis une note de 1 en expliquant qu'il avait fini sa session de jeu sur une mauvaise impression dû à un niveau plus compliqué. Il a eu alors le sentiment d'être moins performant.

Concernant Hours of Code 2020 (Inclusion), les participants lui ont at-

tribué une moyenne de 8.33 sur 10. Mis à part N2-M2020 qui a trouvé son avancement et ses réussites très satisfaisants, les notes de N1-M2020 et N3-M2020 ne reflètent pas vraiment l'explication donnée par la suite. En effet, N1-M2020 a défini sa performance comme étant « en dents de scie ». Quant à N3-M2020, il a attribué un 8 car il a quand même réalisé plusieurs erreurs qui, selon lui, ont entaché sa performance.

La moyenne obtenue par Hours of Code 2021 (Timecraft) est de 8.66 sur 10. N1-M2021 a trouvé très satisfaisant de voir qu'il réussissait les missions et N3-M2021 a expliqué avoir vu la performance qu'il réalisait au fur et à mesure grâce au scénario qui avançait. N2-M2021, quand à lui, a attribué seulement un 7 car la deuxième épreuve a été plus compliquée pour lui.

Les deux leçons de Minecraft : Education Edition ont reçu des moyennes assez hautes. Les erreurs ont apparemment ralenti les joueurs dans leur évolution du jeu, ce qui a eu un effet sur leur sentiment de performance. Par ailleurs, nous pouvons observer que plus un joueur a un niveau élevé en programmation et plus le jeu est « compliqué », plus son sentiment de performance est élevé.

#### 4. « Étais-tu réticent à apprendre à programmer ? L'es-tu toujours ? »

Sur neuf participants, N2-AB est le seul à avoir avoué être réticent à programmer. Cependant, il a expliqué qu'il l'était un peu moins après la session, et ce grâce au jeu. Pour tous les autres, la réponse était négative pour diverses raisons. Les trois joueurs débutants ont indiqué la même chose : ils n'étaient tout simplement pas intéressés par la programmation ou n'en voyaient pas l'utilité. Toutefois, ils ne sont pas contre en refaire plus tard.

#### 5. « Comptes-tu rejouer à d'autres jeux pour apprendre à programmer ? »

Six participants sur neuf ont répondu qu'ils comptaient rejouer à d'autres jeux vidéo pour apprendre la programmation, si ceux-ci sont relativement similaires à ceux qu'ils ont découvert lors de leur expérimentation. N2-M2021 a expliqué : « *De base j'aime bien ça, mais j'ai pas un gros niveau donc via*

*un jeu ce sera plus intéressant. J'aurais moins l'impression de perdre mon temps* ». Seuls N2-M2020 et N3-M2020 ont répondu à la négative à cette question, ce dernier expliquant que pour lui le code était destiné uniquement au boulot. N2-AB n'a pas répondu à la question.

6. « Qu'as-tu pensé du format « jeu vidéo » pour apprendre la programmation ? Trouves-tu ça efficace ? »

Tous les retours obtenus pour cette question ont été positifs. Pour la plupart des participants, le jeu vidéo est un format très efficace pour apprendre la programmation. N1-AB nous a partagé une réflexion intéressante : *« Déjà, il y a le fait que j'ai pas l'impression d'avoir échoué sur quelque chose, donc ça aide pour garder une concentration et ne pas être frustré trop vite. De plus, je me suis bien amusé : j'ai retenu des trucs que si j'avais fait sans m'amuser, j'aurais pas retenu »*. N1-M2021, N2-AB et N2-M2021 trouvent le jeu vidéo efficace pour apprendre les bases car c'est le média éducatif, selon eux, qui se prête le mieux à l'apprentissage de la programmation. N1-M2021 a également mentionné le fait que le jeu vidéo permettait d'apporter du concret dans cet apprentissage, tout comme N2-M2020 et N2-M2021. N3-M2021 a expliqué : *« Je trouve ça bien, je pense que ça pourrait ouvrir plus de possibilités pour tout le monde et que les jeunes passent déjà leur temps sur les jeux vidéo alors autant l'utiliser à bon escient »*.

7. « Cite-moi un avantage du jeu vidéo pour apprendre à programmer. »

Les réponses à cette question ont été diverses et variées. Pour N1-AB, le jeu vidéo permet d'apporter une concentration plus active lors de l'apprentissage. Pour N1-M2020, le visuel est joli et les récompenses obtenues à chaque étape permettent d'améliorer le sentiment d'apprentissage. N1-M2021 a expliqué que le jeu vidéo apporte du concret et le scénario permet de se voir progresser. N2-AB trouve que l'essai-erreur qu'offre le jeu vidéo permet un meilleur apprentissage, mais également les récompenses lors des différents niveaux. Pour N2-M2020, le jeu vidéo est immersif : le joueur est plongé dans l'histoire et il va alors tout faire pour réussir les différentes tâches. Pour

N2-M2021, l'avantage du jeu vidéo, c'est le fait que ça soit ludique. Ensuite, N3-AB définit le jeu vidéo comme étant plus sympa visuellement qu'un cours pratique. Pour N3-M2020, apprendre en s'amusant grâce au jeu est un avantage. Enfin, pour N3-M2021, le jeu vidéo permet d'augmenter la créativité du joueur.

8. « Cite-moi un inconvénient du jeu vidéo pour apprendre à programmer. »

Un inconvénient en particulier est revenu plusieurs fois : apprendre la programmation à travers le jeu vidéo n'apporte pas suffisamment de compétences précises. Les joueurs n'ont pas l'impression d'avoir les capacités pour programmer car le jeu ne leur a appris que les bases. Les principes de programmation sont différents car ajustés au jeu : il n'y a pas de réel apprentissage de la syntaxe. De plus, l'interface, pour N1-M2020 et N3-M2021, peut poser problème avec des visuels qui peuvent être parfois trop confus. Par ailleurs, N2-M2021 a expliqué : « *Ça prend peut-être plus de temps pour se lancer dans le jeu alors que si je veux juste une réponse claire pour débiter je peux aller sur OpenClassrooms<sup>7</sup> et trouver cette information* ». Enfin, N2-M2020 a relevé le fait que le jeu vidéo, et plus particulièrement les jeux en 3D comme Minecraft, peut donner la nausée à certains joueurs.

9. « Sur une échelle de 1 à 10, comment évaluerais-tu ton expérience de jeu ? Tu t'es plus amusé ou ennuyé ? »

Cette dernière question a permis de conclure la session de chaque joueur avec une note d'appréciation globale sur son expérience de jeu (Figure 10).

Tout d'abord, Algo-bot s'est vu attribuer une moyenne de 7.33 sur 10. N1-AB a expliqué avoir trouvé le jeu très chouette et avoir vécu une expérience intéressante. N2-AB a trouvé le jeu joli, mais son expérience a parfois été déroutante car il n'aime pas le code.

Ensuite, Hours of Code 2020 (Inclusion) a reçu une moyenne de 8.66 sur

---

7. OpenClassrooms (<https://openclassrooms.com/fr/>) est un site de formation en ligne, proposant des cours gratuits et payants et des forums dans divers domaines, et notamment la programmation.

10. N1-M2020 a beaucoup apprécié son expérience : « *J'aime bien moi tout ce qui est un peu éducatif, jouer et en même temps apprendre. C'était chouette, et je me suis dit que s'il y avait d'autres trucs ou quoi ça pourrait être marquant à refaire une fois* ». N2-M2020 a trouvé que l'histoire donnait envie de jouer, et N3-M2020 a trouvé l'interactivité amusante malgré la leçon qui était parfois trop simple pour lui.

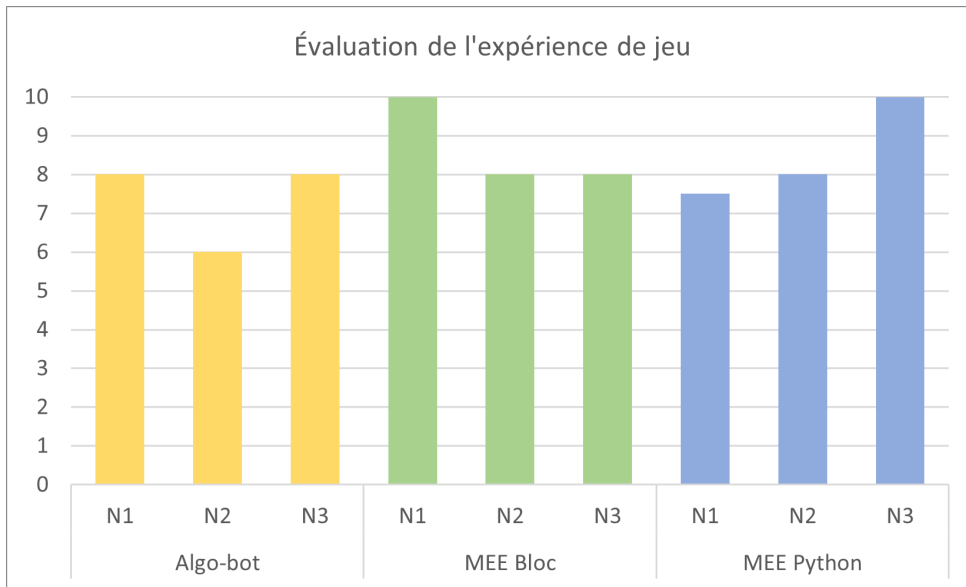


FIGURE 10 – Question n°9

Enfin, les joueurs ont donné à la leçon Hours of Code 2021 (Timecraft) une moyenne de 8.5 sur 10. N1-M2021 a apprécié le fait de devoir choisir parmi plusieurs missions donnant à ses choix une importance à prendre en compte. Quant à N2-M2021 et N3-M2021, ils ont expliqué s'être bien amusés.

Nous pouvons observer que, de manière générale, les participants ont apprécié leur expérience de jeu et se sont amusés. Certains éléments donnés lors des questions précédentes sont revenus dans l'argumentaire de cette dernière question, expliquant les différentes notes attribuées. Les joueurs au niveau confirmé ont trouvé leurs tâches faciles à réaliser, mais ont quand même trouvé l'expérience satisfaisante.

## 2.2 Conclusion de l'entretien

Dans l'ensemble, les résultats montrent que les joueurs ont apprécié leur expérimentation. Ils se sont tous amusés et ont trouvé la réussite des différents exercices satisfaisante.

Certains jeux conviennent mieux que d'autres pour certains aspects. Cependant, il est important de noter qu'ils dépendent toujours du niveau et de l'expérience personnelle des joueurs. Par exemple, Hours of Code 2021 (Timecraft) motive plus les joueurs ayant un niveau de programmation intermédiaire et confirmé, les joueurs débutants préférant Algo-bot. Hours of Code 2020 (Inclusion) a été le jeu qui a donné le plus l'impression aux joueurs d'avoir appris à programmer. Enfin, les joueurs ont évoqué avoir ressenti une meilleure performance grâce à l'évolution du jeu dans Hours of Code 2021 (Timecraft).

Les joueurs confirmés ont fait preuve d'une auto-critique plus poussée concernant leur expérience. Leur ressenti sur le jeu vidéo pour l'apprentissage était biaisé sur certaines questions au vu de leur expérience dans le domaine de la programmation, donnant des résultats légèrement différents.

Le jeu vidéo a été jugé comme étant un média éducatif efficace pour l'apprentissage de la programmation car le format s'y prête bien et rend l'apprentissage concret. Plusieurs avantages ont été évoqués, comme la concentration que le jeu permet d'apporter, le scénario du jeu rendant la progression concrète, la stratégie de l'essai-erreur, l'immersion, le visuel du jeu, l'amusement ressenti et la créativité développée. Toutefois, plusieurs inconvénients ont également été soulevés. Le jeu vidéo, selon les participants de cette recherche, ne permet pas d'acquérir de compétences précises et réutilisables plus tard. Les bases développées restent trop spécifiques au jeu. De plus, l'interface de certains types de jeu peut se révéler troublante pour certains joueurs, rendant l'expérience moins fluide. Enfin, il faut faire attention également à certains environnements de jeu en 3D qui ne conviennent pas à tous les joueurs.

# Conclusion

## 1 Discussion des résultats

L'objectif de cette recherche a été d'évaluer le sentiment d'apprentissage de la programmation à travers le jeu vidéo. Plusieurs éléments peuvent être discutés suite à l'analyse des résultats. Cette discussion a ainsi pour objectif de répondre à nos trois sous-hypothèses au départ des indicateurs mesurés, afin de valider, ou non, notre hypothèse principale.

### 1.1 Motivation à apprendre

La sous-hypothèse n°1 proposait que les joueurs soient plus motivés à apprendre la programmation après avoir joué au jeu vidéo. Durant l'entretien, trois des quatre indicateurs élaborés pour répondre à cette sous-hypothèse ont été questionnés. Le dernier indicateur a été mesuré lors de l'auto-confrontation.

Tout d'abord, concernant la motivation à apprendre à programmer perçue par les joueurs suite à la session de jeu, un score moyen a été attribué à chaque jeu par les joueurs. Algo-bot et Hours of Code 2021 (Timecraft) ont obtenu des notes supérieures à la moyenne (étant de 5 sur 10), respectivement de 6.33 et 8.33 sur 10. Seul Hours of Code 2020 (Inclusion) a reçu une note inférieure à la moyenne. La majorité des joueurs a donc semblé être motivée par leur jeu. Nous retenons cependant que le type de jeu a semblé avoir un effet sur la motivation des joueurs.

Ensuite, concernant la réticence, la réponse est plutôt unanime. L'ensemble des joueurs a partagé ne pas être réticents à apprendre la programmation. Le seul participant ayant émis une réticence avant de jouer a indiqué l'être un peu moins à la fin à la session de jeu. Cela pourrait indiquer que le jeu lui a permis de stimuler sa confiance en lui en le motivant à atteindre ses objectifs.

Six participants sur neuf ont indiqué vouloir rejouer à d'autres jeux pour apprendre à programmer. Le jeu pourrait être vecteur de motivation en induisant une volonté de rejouer, et donc d'en apprendre davantage sur le domaine sélectionné (dans le cadre de cette recherche, la programmation).

Enfin, le quatrième indicateur, qui a été évalué lors de l'auto-confrontation, avait pour objectif d'identifier l'intérêt des joueurs face aux exercices demandés et leur concentration tout au long de la session de jeu. Malgré les difficultés rencontrées, les joueurs ont établi des stratégies leur permettant de continuer les exercices. De plus, certains participants ont exprimé vouloir refaire leurs exercices pour atteindre le score optimal, ce qui peut traduire un intérêt particulier pour le jeu. Le fait de réussir les exercices permettait à la majorité des joueurs d'être satisfaite et de vouloir continuer. Les débutants avaient plus tendance à lire les consignes à la lettre. En ce qui concerne la concentration, elle a été évoquée par les participants en traitant des avantages du jeu vidéo.

Cependant, les participants avec un niveau confirmé de programmation ont tous voulu résoudre rapidement les tâches. Nous pouvons interpréter cette volonté par un intérêt moins prononcé pour les exercices proposés, de par l'expérience personnelle de ces trois joueurs. Ils ont chacun réalisé des erreurs évitables par ce manque de concentration. Nous pouvons établir un lien avec Ruph (1997) : les joueurs ayant un sentiment de compétence plus élevé ont tendance à attribuer leurs échecs à un manque d'efforts.

Nous pouvons retenir que le niveau du joueur, et le type de jeu qui lui est attribué, peuvent avoir un effet sur l'intérêt porté au jeu et à l'apprentissage associé.

La sous-hypothèse n°1 peut être validée suite à la mesure de ces quatre indicateurs. En effet, la majorité des joueurs a exprimé être motivée suite à la session de jeu et l'ensemble des joueurs n'est pas réticent à programmer suite à l'expérience. La plupart a une volonté de rejouer à d'autres jeux pour

apprendre à programmer. Les débutants ont porté plus d'intérêt au jeu que les joueurs confirmés, mais en général, les joueurs sont restés concentrés et ont porté de l'intérêt aux tâches données dans les différents types de jeux.

Concernant les effets que la motivation apportée par le jeu vidéo a pu avoir sur le sentiment d'apprentissage de la programmation, nous pouvons établir ici qu'ils sont principalement positifs. Le jeu vidéo a pour effet de motiver le joueur dans la résolution de sa tâche (Plass et al., 2015). De plus, il rendrait les joueurs moins réticents à l'apprentissage d'une discipline comme la programmation (Grover & Pea, 2013). Le jeu vidéo pourrait être un vecteur de motivation en induisant une volonté de rejouer, et donc en apprendre davantage sur la programmation. Enfin, il permettrait aux joueurs de rester concentrés dans la réalisation des tâches : le jeu vidéo, comme expliqué par Homer et al. (2020), autorise le joueur à réessayer jusqu'à ce qu'il atteigne ses objectifs. Cet intérêt situationnel plus élevé permet d'obtenir de meilleurs résultats d'apprentissage. N1-M2020 a fait une remarque pertinente à ce sujet : *« C'est pour ça que généralement j'essaie de conjuguer cours à l'université et une application un peu plus rigolote, comme Duolingo<sup>8</sup>. C'est très visuel, et pour moi c'est beaucoup plus facile de retenir ce genre d'éléments. Ça me stimule plus. »*.

Malgré tous ces effets positifs, il est important de noter que le type de jeu aurait un effet sur la motivation des joueurs. En effet, Hours of Code 2021 (Timecraft) a été le jeu qui a semblé être le plus motivant pour les joueurs intermédiaires et confirmés. Les joueurs étaient directement confrontés à un vrai langage de programmation. Parmi les trois types de jeu, c'est Algo-bot qui a semblé être le type de jeu le plus motivant pour les joueurs ayant un niveau débutant. Ce jeu étant plus visuel et ne demandant aucune maîtrise de syntaxe, sa prise en main rapide a permis aux joueurs de comprendre la logique des exercices sans s'en rendre compte.

---

8. Duolingo (<https://fr.duolingo.com>) est une application web et téléphone d'apprentissage ludique des langues.

## 1.2 Sentiment de compétence

La sous-hypothèse n°2 supposait que les jeux vidéo dédiés améliorent le sentiment de compétence à programmer. Deux des trois indicateurs élaborés pour répondre à cette sous-hypothèse ont été analysés durant les entretiens. Le dernier indicateur a été mesuré lors de l'auto-confrontation.

Premièrement, en ce qui concerne l'auto-perception des joueurs à avoir appris un peu plus à programmer à la suite de leur session de jeu, seulement trois joueurs ont attribué une note supérieure à la moyenne de 5 sur 10. Pour chaque type de jeu, les résultats indiquent que ce sont les joueurs débutants qui ont attribué les meilleures notes : ceux-ci ont tous les trois expliqué avoir appris des bases ou une certaine logique suite à la session de jeu. Les joueurs de niveaux supérieurs ont émis des réticences ou sont mitigés. En somme, plus le joueur est débutant et plus il aura eu l'impression d'avoir appris à programmer.

Dans un deuxième temps, l'avis sur le format du jeu vidéo a été interrogé. Tous les joueurs ont trouvé le jeu vidéo comme étant un format efficace pour apprendre la programmation. Certains trouvent que c'est le format de média éducatif qui se prête le mieux à l'apprentissage de la programmation. Il apporte du concret, et permet de rester concentré en s'amusant.

Enfin, durant l'auto-confrontation, nous avons pris soin d'observer attentivement l'évolution des joueurs dans leur leçon. En général, la plupart des joueurs ont évolué de plus en plus vite dans le jeu. Les joueurs réutilisaient leurs stratégies gagnantes dans les niveaux suivants une fois la logique entraînée. Nous avons également observé une prise en main progressive des outils de programmation, quel que soit le type de jeu. La majorité des joueurs intermédiaires et confirmés a émis une volonté d'aller le plus vite possible. Cependant, ils ont quand même commis quelques erreurs, ralentissant leur évolution croissante dans le jeu.

Cette deuxième sous-hypothèse peut être validée, mais il y a quelques nuances à apporter. Seuls les joueurs débutants ont ressenti une amélioration de leur sentiment d'apprentissage. Néanmoins, l'ensemble des joueurs a caractérisé le jeu vidéo comme étant un format propice à l'apprentissage de la programmation. Enfin, la majorité des joueurs a évolué de plus en plus vite dans le jeu, malgré les difficultés rencontrées.

Le jeu vidéo peut avoir plusieurs effets sur le ressenti d'apprentissage des joueurs. En effet, malgré le fait que seuls les débutants ont indiqué avoir réellement appris durant leur session de jeu, nous avons pu observer que la gestion de l'échec a été reçue de manière positive. L'échec fait partie des étapes de l'apprentissage pour résoudre les tâches car il permet d'évaluer sa propre stratégie afin d'atteindre les objectifs fixés (Plass et al., 2020). La stratégie de l'essai-erreur a été appliquée ici dans la majorité des cas : celle-ci permet d'identifier son erreur plus facilement et d'assimiler la logique associée. Certains joueurs ont ressenti de la frustration qui s'est transformée en soulagement une fois la tâche réussie. Cela converge avec ce que Maxmen (2010) énonçait : les participants ont été plus confiants sur leur manière d'apprendre car ils ont appréhendé les difficultés à travers un jeu.

Le type de jeu a également l'air d'avoir un effet sur le ressenti d'apprentissage. Hours of Code 2020 (Inclusion), utilisant la programmation en blocs, a été le jeu donnant aux joueurs la meilleure impression d'avoir appris à programmer davantage. Algo-bot a reçu des notes relativement basses : ce type de jeu paraît être moins concret auprès des joueurs au niveau de l'apprentissage. De plus, nous pouvons indiquer ici que le niveau des joueurs a une influence sur le ressenti. Plus un joueur est compétent, moins il aura ressenti une évolution de son apprentissage de la programmation avec le jeu.

### **1.3 Sentiment de performance**

Enfin, nous supposons pour notre troisième sous-hypothèse que l'évolution des exercices dans le jeu vidéo permettait au joueur de se sentir performant. Il se base sur un indicateur du sentiment de performance. Celui-ci a été

mesuré lors de l'entretien.

En effet, nous avons demandé aux participants d'évaluer leur sentiment de performance suite à leur évolution dans le jeu. Les deux leçons sur Minecraft : Education Edition ont reçu des moyennes élevées. Nous avons relevé ici plusieurs commentaires traduisant la même chose : les erreurs rencontrées ont, pour certains joueurs, entaché leur sentiment de performance. Cependant, l'avancement ressenti lors des différents niveaux était positif pour la majorité des joueurs. Notre troisième sous-hypothèse peut être validée.

Le jeu vidéo engage les joueurs, comme expliqué par Homer et al. (2020), ce qui leur permet de mieux évaluer leur propre performance et leur apprentissage. Un joueur engagé, portant de l'intérêt aux exercices et mettant en place des stratégies pour réussir, réussit ses tâches en les décrivant comme satisfaisantes lorsqu'il les termine. L'investissement devient plus affectif lorsque l'engagement du joueur fait passer la motivation extrinsèque à une motivation intrinsèque (Bonenfant & Philippette, 2018). Les participants ont, en effet, exprimé s'être amusés lors de la session de jeu : cette réaction émotionnelle est produite par la capacité d'être engagé dans le jeu (Fizek, 2014). De plus, l'un des avantages cités par les participants est l'immersion apportée par le scénario. Comme l'explique Bandura (1995), un engagement en profondeur, une motivation intrinsèque et le fait de traiter les tâches comme des défis traduisent un sentiment de compétence élevé.

Selon Ruph (1997), le feedback reçu influence les croyances sur ses compétences. Si le feedback reçu pour les succès précédents est attribué à l'effort, les apprenants perçoivent plus de progrès, entretiennent une plus haute motivation et une perception plus forte de leurs compétences. Le jeu vidéo donne un feedback immédiat au joueur. En effet, la résolution des problèmes de programmation demande une compilation du code effectué pour voir le résultat. Cela permet au joueur de repérer ses erreurs facilement et de rester engagé dans l'apprentissage. La stratégie de l'essai-erreur permet d'évoluer et de se sentir performant lorsqu'un exercice plus compliqué a été réussi. Les tâches

difficiles deviennent alors des défis à relever, les joueurs se persuadant qu'ils peuvent réussir (Bandura, 1995). La compétence de résolution de problème propre à la programmation mais également à l'utilisation d'un jeu vidéo est alors stimulée. Nous pouvons ainsi dire que le feedback reçu donne du sens à la progression. Comme citées par les joueurs, les récompenses obtenues à chaque étape permettent d'améliorer le sentiment d'apprentissage, et le scénario permet de se voir progresser.

Ce sentiment de performance dépend également du type de jeu et du niveau du joueur. De ce que nous avons pu observer, Algo-bot permettait moins aux joueurs de s'immerger et de se rendre compte de leur évolution dans le jeu. Après chaque niveau, le joueur est confronté au score qu'il vient de réaliser. Celui-ci pourrait avoir un effet sur le sentiment de performance : un joueur n'ayant pas réalisé le score optimal pourrait dévaluer sa performance.

#### **1.4 Hypothèse principale et question de recherche**

Les trois sous-hypothèses étant validées, nous pouvons valider l'hypothèse principale. L'utilisation du jeu vidéo éducatif stimule le sentiment d'apprentissage de la programmation.

Karsenti et Bugmann (2018) s'interrogeaient sur les effets que les jeux vidéo pouvaient avoir. À travers cette recherche, nous avons pu établir que les effets relevés sont positifs. Le jeu vidéo est efficace et pertinent pour le type d'apprentissage que nous avons envisagé. Il peut, dans ce cas-ci, être considéré comme étant un « bon » média éducatif qui a pour effet de motiver, engager et faire ressentir un sentiment d'apprentissage positif au joueur. Celui-ci voit donc son sentiment d'apprentissage de la programmation stimulé.

Comme l'expliquait Ruph (1997), les premiers essais que les joueurs réalisent sont les plus sensibles car ceux-ci vont déterminer le sentiment de compétence et l'engagement futur dans des tâches semblables. Dans le cadre de cette recherche, nous avons identifié que la plupart de nos participants a émis la volonté de rejouer à des jeux pour apprendre à programmer. L'expérimentation qu'ils ont vécue à travers le jeu vidéo a donc pu augmenter leur senti-

ment de compétence et leur engagement notamment grâce au ressenti positif et à l'amusement qui en a résulté. Enfin, tous les participants ont réussi à atteindre les objectifs d'apprentissage propres à chaque leçon. Ils ont chacun effectué une série d'opérations logiques dans un certain ordre afin de donner du sens aux actions performées pour atteindre les buts (Bonenfant & Philippette, 2018). L'apprentissage s'est donc produit à mesure que les joueurs acquéraient les compétences pour atteindre ces objectifs (Homer et al., 2020).

Cependant, il est important de nuancer certains éléments de réponse. Nous devons tenir compte du profil des joueurs mais également du type de jeu. Le croisement de ces deux aspects a permis d'identifier différents effets intéressants que peut avoir le jeu sur la stimulation du sentiment d'apprentissage de la programmation.

Comme l'expliquaient Plass et al. (2020), les mécaniques du jeu ont un effet sur l'apprentissage. De plus, celui-ci est facilité par les mécaniques en phase avec les objectifs d'apprentissage (Plass et al., 2015). Lors de cette recherche, nous avons utilisé trois types de jeux différents, à savoir la programmation avec une barre d'instructions, la programmation en blocs et la programmation avec un véritable langage en usage, dans ce cas-ci Python. Les trois jeux étaient basés sur l'apprentissage de la programmation, et comportaient donc des objectifs éducatifs similaires. Nous avons pu relever que les types de jeux utilisés avaient ainsi des effets différents sur les aspects définissant notre sentiment d'apprentissage. En effet, apprendre la programmation en codant avec un langage de programmation est le type d'apprentissage qui semble motiver le plus les joueurs ayant déjà les bases. Concernant les joueurs débutant la programmation, ils ont eu l'air d'être plus motivés par l'apprentissage via la barre d'instructions, malgré le manque d'immersion et de ressenti de progression évalué par les joueurs. En revanche, c'est la programmation en blocs qui a permis aux joueurs de mieux ressentir l'apprentissage de la programmation. Enfin, en plus du type de gameplay propre à chaque jeu et ayant des effets différents, le design du jeu est également à prendre en compte. Celui-ci peut influencer positivement l'expérience du joueur, mais également

devenir une entrave au bon apprentissage si des éléments sont jugés comme étant non pertinents (par exemple, les pictogrammes jugés confus dans Algotbot).

Le profil du joueur a aussi tenu un rôle dans la définition des effets. Le niveau d'aisance du joueur a pu avoir une influence sur la réalisation des tâches demandées. Plus le participant avait un niveau élevé, moins il a ressenti avoir appris quelque chose. Cette affirmation paraît logique de prime abord car les jeux sélectionnés traitaient uniquement de bases de la programmation. Cependant, au travers des entretiens, les joueurs confirmés ont tout de même considéré le jeu comme étant un bon vecteur d'apprentissage, pour toutes les raisons citées lors de cette recherche.

Il est devenu évident, au travers de notre recherche et des résultats récoltés, que stimuler le sentiment de compétence est important pour l'apprentissage. En effet, si ce sentiment est élevé, l'apprenant développera une meilleure gestion du temps, une meilleure persévérance et une meilleure résolution de problème (Ruph, 1997). Si le jeu vidéo est considéré comme étant un média éducatif permettant de stimuler ce sentiment d'apprentissage, en raison des multiples effets identifiés lors de cette recherche, alors il pourrait intégrer davantage les milieux scolaires.

## **2 Perspectives et limites**

Bien que tout ait été mis en oeuvre pour réaliser cette étude exploratoire le plus rigoureusement possible, nous pouvons identifier quelques critiques et perspectives d'amélioration concernant les résultats obtenus et l'analyse qui en a été tirée.

Certains choix méthodologiques peuvent être discutés. Premièrement, l'ensemble des joueurs qui se sont vus attribuer un jeu sur Minecraft : Education Edition avait déjà eu une expérience ultérieure avec le jeu de base. Certains participants avaient également déjà joué à la version éducative, ayant ainsi

une affinité préalable avec le Code Builder, rendant leur expérimentation plus fluide que des joueurs découvrant l'outil de programmation. Afin de rendre les résultats plus cohérents, nous aurions pu sélectionner des joueurs n'ayant aucune expérience avec les jeux utilisés lors de la recherche.

Ensuite, les objectifs d'apprentissage des jeux auxquels les participants ont joué n'étaient pas complètement en phase avec ceux des joueurs. En effet, nous avons sélectionné des étudiants universitaires. Ceux-ci ne correspondent pas au public visé initialement par les jeux, notamment Minecraft : Education Edition qui s'adresse plus spécifiquement aux écoles primaires et secondaires. La quasi-expérimentation mise en place nécessitant des capacités d'auto-réflexion suffisamment développées, ce choix a entravé ce critère de compatibilité des objectifs, rendant les résultats parfois presque incohérents lorsque les critiques réflexives n'étaient pas assez poussées. Cependant, nous avons dans la majorité des cas reçu des commentaires constructifs nous permettant de répondre à nos hypothèses. Dans une future recherche, aligner les objectifs d'apprentissage des participants et des jeux semble être une condition indispensable pour récolter des résultats plus cohérents. Il faudrait alors faire abstraction de cette auto-réflexivité recherchée dans la méthodologie présentée ici, et traiter des données peut-être moins critiques, mais plus naturelles.

De plus, nous avons réalisé les quasi-expérimentations avec uniquement un seul participant par catégorie, croisant son niveau de programmation actuel et le type de jeu qui lui a été attribué (Figure 1). Les résultats récoltés résultent donc d'une unique expérience ressentie par un individu en particulier. Les ressentis exprimés par les joueurs ont été purement subjectifs et personnels. Dès lors, recruter plusieurs participants par catégorie semble être indispensable pour une future étude sur le sujet. En effet, il serait alors possible de croiser les différents profils entre les joueurs d'une même catégorie, mais également leurs comportements et leurs ressentis. Étendre le nombre d'individus permettrait de soutenir davantage les effets analysés dans nos hypothèses.

Enfin, nous sommes restés ici dans une analyse exploratoire. Nous pouvons envisager une analyse plus poussée sur la comparaison entre la percep-

tion des compétences et les compétences réelles des joueurs. Celle-ci pourrait se faire via des questionnaires avant et après la session de jeu. Les questionnaires auraient pour objectif d'évaluer la compréhension des concepts à travers différents exercices, mais également le changement concret de ressenti après avoir joué au jeu. Cette méthodologie étendue viserait à obtenir une meilleure évaluation des effets que le jeu vidéo peut avoir sur l'apprentissage de la programmation.

### **3 Conclusion générale**

En conclusion, nous avons pu observer à travers cette recherche exploratoire et qualitative que les jeux vidéo ont des effets sur le sentiment d'apprentissage de la programmation. Pour y parvenir, nous nous sommes penchés sur trois aspects jugés fondamentaux lors de notre analyse de la littérature existante. Nous avons ensuite mis en place une quasi-expérimentation rassemblant neuf participants autour de trois jeux vidéo de différents types. L'objectif était, suite à la session de jeu réalisée, d'identifier des éléments pouvant définir d'éventuels effets sur le sentiment d'apprentissage à travers une auto-confrontation et un entretien semi-dirigé.

Dans un premier temps, nous avons évalué chez nos participants leur motivation à apprendre à travers le jeu. Les résultats montrent que le jeu vidéo a pour effet de motiver le joueur dans la résolution de ses tâches. Celui-ci est moins réticent à apprendre la programmation après avoir accompli sa leçon, et il souhaite même rejouer à d'autres jeux pour en apprendre davantage. De plus, nous avons constaté que l'ensemble des participants était concentré malgré les différences d'intérêt constatées, celles-ci étant expliquées par les niveaux de programmation différents des joueurs.

Dans un deuxième temps, nous avons tenté d'évaluer si le jeu vidéo pouvait améliorer le sentiment d'apprentissage de la programmation. Les résultats sont plus mitigés ici, démontrant que nos participants n'ont pas particulièrement ressenti avoir appris davantage à programmer après avoir joué. Cependant, la gestion de l'échec positive par la mise en oeuvre de stratégies adaptées

a permis aux joueurs de finir leur leçon de manière confiante et positive.

Enfin, le sentiment de performance des joueurs ressenti face à l'évolution des différents niveaux de jeu a été globalement positif. Le jeu vidéo, de par son immersion, permet aux joueurs de s'engager et de s'investir de manière plus affective. Les joueurs traitent les différentes tâches comme des défis (par la mise en place de stratégies par exemple), et font preuve alors d'un sentiment de performance élevé en obtenant les différentes récompenses apportées par le jeu.

Par conséquent, nous avons établi que les jeux vidéo stimulaient positivement le sentiment d'apprentissage de la programmation. Ils peuvent être considérés comme étant de bons médias éducatifs qui motivent, engagent et améliorent le sentiment de compétence des joueurs. Cependant, nous avons observés que le type de jeu et le niveau préalable en programmation du joueur avaient également plusieurs effets importants à prendre en compte. Plusieurs suggestions d'améliorations ont été évoquées afin de permettre aux futures recherches d'approfondir ce sujet d'étude. Notre méthodologie à petite échelle pourrait servir de base à de futures expérimentations, s'intéressant davantage par exemple au milieu scolaire. L'innovation digitale dans les écoles est devenue un sujet préoccupant dans nos sociétés, et le jeu vidéo pourrait subvenir, à bien des égards, à cette demande.

# Bibliographie

- Almeida, S., Veloso, A., Roque, L., & Mealha, O. (2011). The eyes and games : A survey of visual attention and eye tracking input in video games.. Consulté sur <https://doi.org/10.13140/RG.2.1.2341.3527>
- Bandura, A. (1993). Perceived self-efficacy in cognitive development and functioning. *Educational Psychologist*, 28(2), 117-148. Consulté sur [https://doi.org/10.1207/s15326985ep2802\\_3](https://doi.org/10.1207/s15326985ep2802_3)
- Bandura, A. (1995). Exercise of personal and collective efficacy in changing societies. In A. Bandura (Ed.), *Self-efficacy in changing societies* (p. 1-45). Cambridge University Press. Consulté sur <https://doi.org/10.1017/CB09780511527692.003>
- Bojko, A. (2005). Eye tracking in user experience testing : How to make the most of it. *Proceedings of the 14th Annual Conference of the Usability Professionals' Association (UPA)*. Montréal, Canada.
- Bonenfant, M., & Philippette, T. (2018). Rhétorique de l'engagement ludique dans des dispositifs de ludification. *Sciences du jeu*(10). Consulté sur <https://doi.org/10.4000/sdj.1422>
- Boyd, S. (2018). Playing to investigate the relationship between achievement and the perception of learning. *International Journal of Serious Games*, 5(1). Consulté sur <https://doi.org/10.17083/ijsg.v5i1.214>
- Cierro, A., Philippette, T., Francois, T., Nahon, S., & Watrin, P. (2020). Eye-tracking for sense of immersion and linguistic complexity in the sky-rim game : Issues and perspectives. *ACM Symposium on Eye Tracking Research and Applications*, 1-5. Consulté sur <https://doi.org/10.1145/3379156.3391836>
- Culot, M., & Ligurgo, V. (2021). *LCOMU2663 : Effets éducatifs des médias : 4. Évaluer et mesurer les effets éducatifs* [Unpublished document, Université catholique de Louvain, Louvain-la-Neuve].
- Dempsey, J. V., & Others. (1996, 8-12 Avril). Instructional applications of

- computer games [Rapport de conférence]. In *Annual Meeting of the American Educational Research Association*. New-York, NY. Consulté sur <https://eric.ed.gov/?id=ED394500>
- Deslauriers, L., McCarty, L. S., Miller, K., Callaghan, K., & Kestin, G. (2019). Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom. *Proceedings of the National Academy of Sciences of the United States of America*, 116(39), 19251-19257. Consulté sur <https://doi.org/10.1073/pnas.1821936116>
- Dezuani, M., & Macri, J. (2020). Minecraft : Education Edition for educational impact. Consulté sur <https://research.qut.edu.au/dmrc/projects/mee-minecraft-for-educational-impact/>
- Ducobu, F. (2021). *La première Microsoft Minecraft Education Academy de Belgique va s'implanter à Mons*. Consulté sur <https://www.dhnet.be/regions/mons/2021/12/16/la-premiere-microsoft-minecraft-education-academy-de-belgique-va-simplanter-a-mons-S3Y5ED0D5BERTHF5B34SI5CNAE/>
- Eger, N., Ball, L., Stevens, R., & Dodd, J. (2007). Cueing retrospective verbal reports in usability testing through eye-movement replay. In *Proceedings of HCI 2007 the 21st British HCI Group annual conference University of Lancaster, UK*. Consulté sur <https://doi.org/10.14236/ewic/HCI2007.13>
- Fizek, S. (2014). Why Fun Matters : In Search of Emergent Playful Experiences. In M. Fuchs, S. Fizek, P. Ruffino, & N. Schrape (Eds.), *Rethinking gamification* (p. 273-287). Lüneburg, Meson-Press.
- Froment, E. (2020). *Minecraft s'est vendu à plus de 200 millions d'exemplaires*. Consulté sur <https://geeko.lesoir.be/2020/05/18/minecraft-sest-venu-a-plus-de-200-millions-dexemplaires/>
- Grant, A. M., & Wall, T. D. (2009). The neglected science and art of quasi-experimentation : Why-to, when-to, and how-to advice for organizational researchers. *Organizational Research Methods*, 12(4), 653-686.

- Consulté sur <https://doi.org/10.1177/1094428108320737>
- Grover, S., & Pea, R. (2013). Computational Thinking in K–12 : A Review of the State of the Field. *Educational Researcher*, 42(1), 38-43. Consulté sur <https://doi.org/10.3102/0013189X12463051>
- Homer, B. D., Hayward, E. O., Frye, J., & Plass, J. L. (2012). Gender and player characteristics in video game play of preadolescents. *Computers in Human Behavior*, 28(5), 1782-1789. Consulté sur <https://doi.org/10.1016/j.chb.2012.04.018>
- Homer, B. D., Raffaele, C., & Henderson, H. (2020). Games as playful learning : Implications of developmental theory for game-based learning. In J. L. Plass, R. E. Mayer, & B. D. Homer (Eds.), *Handbook of game-based learning* (p. 22-52). The MIT Press.
- Jenart, D. (2022). *La Belgique se dote d'une MINECRAFT Education academy*. Consulté sur <https://www.digitalwallonia.be/fr/publications/minecraft-education-academy/>
- Karsenti, T., & Bugmann, J. (2018). Quels apports éducatifs du jeu vidéo Minecraft en éducation ? Résultats d'une recherche exploratoire menée auprès de 118 élèves du primaire. *Formation et profession*, 26(1), 89-108. Consulté sur <https://doi.org/10.18162/fp.2018.459>
- Larousse. (s. d.). Jeu. In *Dictionnaire en ligne*. Consulté le 3 janvier 2023, sur <https://www.larousse.fr/dictionnaires/francais/jeu/44887>
- Latzko-Toth, G., Bonneau, C., & Millette, M. (2017). Small data, thick data : Thickening strategies for trace-based social media research. In L. Sloan & A. Quan-Haase (Eds.), *The sage handbook of social media research methods* (p. 199-214). Thousand Oaks : SAGE. Consulté sur <https://dx.doi.org/10.4135/9781473983847.n13>
- Lepage, A., & Romero, M. (2017). Évaluation par compétences d'activités de programmation créative avec l'outil 5c21. In *Cirta 2017* (Vol. 1). Montréal.
- Maxmen, A. (2010). Video games and the second life of science class. *Cell*, 141(2), 201-203. Consulté sur [https://doi.org/10.1016/j.cell.141\(2\), 201-203](https://doi.org/10.1016/j.cell.141(2), 201-203)

.2010.03.045

- Mayer, R. E. (2002). Rote Versus Meaningful Learning. *Theory Into Practice*, 41(4), 226-232. Consulté sur [https://doi.org/10.1207/s15430421tip4104\\_4](https://doi.org/10.1207/s15430421tip4104_4)
- Meunier, S. (2017). Les recherches sur le jeu vidéo en France. *Revue d'anthropologie des connaissances*, 11(3), 379-396. Consulté sur <https://doi.org/10.3917/rac.036.0379>
- Office québécois de la langue française. (2010). Mécanique de jeu. In *Grand dictionnaire terminologique*. Consulté le 8 janvier 2023, sur <https://vitrinelinguistique.oqlf.gouv.qc.ca/fiche-gdt/fiche/26502525/mecanique-de-jeu>
- Pew Research Center. (2008, September 16). *Teens, video games, and civics*. Consulté sur <https://www.pewresearch.org/internet/2008/09/16/teens-video-games-and-civics/>
- Philippette, T. (2016). Étudier les pratiques collectives de jeux vidéo : l'allo-auto-confrontation aux traces de l'activité. *COMMposite*, 19(1), 6-26.
- Plass, J. L., Homer, B. D., & Kinzer, C. K. (2015). Foundations of game-based learning. *Educational Psychologist*, 50(4), 258-283. Consulté sur <https://doi.org/10.1080/00461520.2015.1122533>
- Plass, J. L., Homer, B. D., Mayer, R. E., & Kinzer, C. K. (2020). Theoretical foundations of game-based and playful learning. In J. L. Plass, R. E. Mayer, & B. D. Homer (Eds.), *Handbook of game-based learning* (p. 3-24). The MIT Press.
- Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, 128, 365-376. Consulté sur <https://doi.org/10.1016/j.compedu.2018.10.005>
- Renshaw, T., Stevens, R., & Denton, P. (2009). Towards understanding engagement in games : An eye-tracking study. *On the Horizon*, 17(4), 408-420. Consulté sur <https://dx.doi.org/10.1108/10748120910998425>
- Ruph, F. (1997). *Le sentiment de compétence et l'apprentissage chez l'adulte* (Examen de synthèse, Université de Montréal). Consulté

sur <https://docplayer.fr/1891970-Universite-de-montreal-le-sentiment-de-competence-et-l-apprentissage-chez-l-adulte-question-de-synthese-n-1-par-francois-ruph.html>

- Scherer, R., Siddiq, F., & Sánchez-Scherer, B. (2021). Some evidence on the cognitive benefits of learning to code. *Frontiers in Psychology, 12*. Consulté sur <https://doi.org/10.3389/fpsyg.2021.559424>
- Smith, A., & Anderson, M. (2018). Social media use in 2018. *Pew Research Center, 1*. Consulté sur <https://www.pewresearch.org/internet/2018/03/01/social-media-use-in-2018/>
- Van Campenhoudt, L., Marquet, J., & Quivy, R. (2017). *Manuel de recherche en sciences sociales*. Dunod.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35. Consulté sur <https://doi.org/10.1145/1118178.1118215>
- Zyda, M. (2005). From visual simulation to virtual reality to games. *Computer, 38*(9), 25-32. Consulté sur <https://doi.org/10.1109/MC.2005.297>

Les jeux vidéo sont devenus des formes incontournables de jeu dans nos sociétés. Ils ont désormais pénétré de nombreux domaines de l'activité humaine : le divertissement et les loisirs, bien sûr, mais également les mondes professionnel et de l'éducation. De plus, au cours des trois dernières décennies, les outils utilisés pour inciter les élèves à programmer sur ordinateur ont considérablement évolué, notamment les jeux vidéo éducatifs proposant des leçons d'informatique. Partant de cette idée fondatrice, nous nous sommes intéressés, dans le cadre de ce mémoire, à l'évaluation du sentiment d'apprentissage de la programmation à travers le jeu vidéo. C'est suite à la mise en place d'une quasi-expérimentation basée sur trois types de jeux différents que cette recherche a démontré que les jeux vidéo avaient des effets sur le sentiment d'apprentissage de la programmation. Cependant, certaines nuances sont à prendre en compte, comme discuté à la fin de ce rapport.

Mots-clefs : jeu vidéo, sentiment d'apprentissage, programmation, game-based learning