# Accessibilité et jeux vidéo : Quid de l'architecture logicielle Entités-Composants-Systèmes ?

Délia Garbarini, Mathieu Muratet

# Accessibilité et jeux vidéo : Quid de l'architecture logicielle Entités-Composants-Systèmes ?

Délia Garbarini
Université Paris Lumière, Paris 8, F-93526 Saint-Denis, France

Mathieu Muratet
Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
Université Paris Lumière, INS HEA, 92100 Suresnes, France
mathieu.muratet@lip6.fr

***Résumé*—Les jeux vidéo font partie intégrante de la culture populaire. Le défi pour l'industrie du jeu vidéo est d'autant plus grand que les nombres de joueurs, de jeux et de domaines applicatifs augmentent. Cette augmentation de joueurs inclut les joueurs « handi » et de ce fait, les jeux vidéo doivent s'adapter à ces publics qui peuvent être touchés par une ou plusieurs déficiences passagères ou continues. Les jeux universellement accessibles (UA-Games) ont pour but de créer des interfaces accessibles et manipulables par le plus grand nombre. Pour ce faire, les moteurs de jeu actuels doivent donc être performants mais également évolutifs et modulables. Cet article étudie l'intérêt de l'architecture logicielle Entité-Composant-Système (ECS) pour intégrer des fonctionnalités d'accessibilité dans un jeu existant. Nous prenons comme cas d'étude un jeu sérieux développé en ECS mais non accessible : E-LearningScape. Nous présentons les fonctionnalités d'accessibilité que nous avons intégrées au jeu et nous discutons les avantages et inconvénient de cette approche. Ce retour d'expérience nous montre que l'ECS apporte une souplesse de conception très utile pour intégrer dans les jeux vidéo des fonctionnalités d'interactions non anticipées.**

***Mots clés—Architecture logicielle, design universel, Entités-Composants-Systèmes, fonctionnalité d'accessibilité, jeu vidéo accessible, jeux sérieux, programmation orientée donnée.***

## I. INTRODUCTION

De nos jours, le jeu vidéo est devenu une activité de loisir pour tous et qui en France, est en passe de devenir le premier objet culturel juste derrière le livre. En 2018, le marché français du jeu vidéo connaît une croissance de 15% et vaut 5 milliards d'euros[1]. Un Français sur deux joue aux jeux vidéo, dont 53% de joueurs masculins et 47% de joueurs féminins c'est-à-dire 30% de plus que dans les années 2000[2]. À l'échelle mondiale, le jeu vidéo représente un objet de partage, de socialisation, d'apprentissage et de travail, qui permet d'interagir avec le monde extérieur, de faire partie d'une communauté et de rencontrer des personnes.

Toutefois, un grand nombre de personnes rencontrent des difficultés pour y jouer à cause d'une ou plusieurs invalidités. Pour jouer à un jeu vidéo, les joueurs doivent solliciter leur vue, leur audition, leur toucher, analyser ces informations multimodales et effectuer des actions motrices en retour (capacité à tenir une manette ou à réaliser des mouvements amples) [1, 2, 3].

L'industrie du jeu est grandissante tout comme le nombre de joueurs. Cette augmentation inclut celle du nombre de joueurs handicapés (handi) dont leur déficience temporaire ou chronique révèle des situations de jeu ou des modalités d'interaction inaccessibles. Pour pallier ce problème, il est nécessaire aujourd'hui que les jeux vidéo s'adaptent à l'ensemble de leur public. Pour ce faire, il existe différentes approches dont l'une d'elles consiste à concevoir des jeux vidéo universellement accessibles (UA-Games) [1, 4, 5, 6]. Cette méthode a pour but de créer des interfaces pour tous c'est-à-dire accessibles à autant de joueurs que possible, avec leurs capacités et incapacités.

Pour cela, l'architecture d'un moteur de jeu doit être suffisamment flexible pour permettre au cours du processus de développement du jeu d'ajouter de nouvelles fonctionnalités, de tester différentes modalités d'interaction et de proposer plusieurs représentations de l'information.

Nous nous intéressons dans cet article à l'architecture logicielle Entité-Composant-Système (ECS). Cette architecture logicielle en plein développement dans le domaine du jeu vidéo semble embarquer des mécanismes favorisant la modularité et l'évolutivité. C'est une approche orientée donnée fondée sur la notion de composition et qui se soustrait des mécanismes fondamentaux de la POO tels que l'héritage ou l'encapsulation. Plusieurs travaux démontrent qu'elle est en mesure de répondre aux problèmes qui incombent à la POO dans le domaine du développement des jeux vidéo [4, 7, 8, 9].

La question que nous nous posons dans ce travail de recherche est la suivante : l'architecture logicielle ECS peut-elle avantager la conception, le développement et l'adaptation de jeux vidéo pour l'intégration de fonctionnalités d'accessibilités ?

Dans un premier temps, nous discuterons des enjeux de l'adaptation dans les jeux vidéo. Nous présenterons ensuite l'architecture logicielle Entités-Composants-Systèmes qui a servi à développer le jeu sérieux étudié dans cet article. La

[1] https://www.afjv.com/news/9546_chiffre-d-affaires-du-marche-francais-du-jeu-video-en-2018.htm, consulté le 16 janvier 2020

[2] https://www.sell.fr/news/bilan-marche-jeu-video-2018 consulté le 16 janvier 2020

section IV présentera le jeu en question et son analyse du point de vue de son accessibilité. Nous discutons l'intérêt de l'architecture ECS pour l'intégration des nouvelles fonctionnalités d'accessibilité dans la section V avant de conclure.

## II. ENJEUX DE L'ADAPTATION DANS LES JEUX VIDÉO

Les jeux vidéo sont des applications multimodales. Ils demandent aux joueurs d'avoir des compétences sensorielles, mentales et motrices nécessaires pour capter les stimuli produits par le jeu et fournir des réponses en retour à travers la maîtrise de périphériques d'entrée (manipuler une manette, un clavier, une souris, une interface tactile...) [1, 2, 3, 10]. Yuan *et al.* [3] ont identifié deux catégories de stimuli. Les stimuli primaires sont essentiels à la bonne compréhension du jeu, un joueur ne percevant pas ces stimuli ne pourra pas jouer au jeu. Les stimuli secondaires complètent les stimuli primaires et ne sont pas essentiel à la compréhension du jeu. Dans une grande majorité de jeux les stimuli primaires sont essentiellement visuels alors que les stimuli secondaires exploitent les canaux auditifs et haptiques (vibrations des manettes).

Ainsi, Bierre *et al.* [5] identifient que l'inaccessibilité d'un jeu vidéo peut bien sûr être dû à la non perception des stimuli primaires et secondaires (en raison d'une déficience visuelle ou auditive par exemple) mais également sur la capacité du joueur à analyser ces stimuli et fournir une réponse en retour. Une surcharge de stimuli visuels et auditifs peut ainsi mettre en difficulté des joueurs avec une déficience cognitive qui auront alors des difficultés à trier l'information et déterminer la réponse à fournir. Une déficience motrice quant à elle limitera le joueur dans la mise en œuvre de la réponse si elle se traduit par la manipulation des commandes de jeu surtout si celle-ci doit être réalisée dans un temps limité.

De nombreux travaux scientifiques contribuent à produire de la connaissance sur l'accessibilité des jeux vidéo. Certains portent sur l'analyse des difficultés rencontrées par les joueurs [3] et d'autres étudient les différentes fonctionnalités à développer pour rendre les jeux vidéo accessibles [1]. Nous notons aussi le travail réalisé par le site GameLover[3] qui vise à analyser des jeux vidéo du commerce sous l'angle de l'accessibilité. Il témoigne ainsi des bonnes et mauvaises pratiques mises en œuvre par les studios de développement des jeux vidéo. L'amélioration de l'accessibilité des jeux vidéo trouve également une inspiration dans les normes initialement développées pour les sites web (WCAG[4] et W3C[5]). Il existe en effet des similitudes notamment pour la gestion des interfaces et menus (espacement des boutons, textes alternatifs, sous titrages, désactivation des animations, modification des contrastes, synthèses vocales). Plus précisément dans le domaine du jeu vidéo l'initiative *Game Accessibility Guideline*[6] propose une liste détaillée de recommandations à suivre en fonction du type de déficience pour améliorer l'accessibilité des jeux vidéo.

L'approche UA-Games ainsi que les travaux cités ci-dessus renvoient aux recherches menées en EIAH sur les dimensions de l'adaptation [11] et les notions d'environnements adaptés, adaptables et adaptatifs. Plongées dans le domaine du jeu vidéo ces dimensions nous permettent de définir les jeux adaptés, adaptables et adaptatifs.

Les jeux adaptés visent un profil de joueur particulier et les modalités d'interaction seront donc fixées lors de la conception. L'environnement créant la situation de handicap, un jeu adapté pour une déficience particulière risque de générer en conséquence une inaccessibilité pour les autres joueurs. Le jeu *A blind Legend*[7] en est un exemple singulier. Ce jeu met le joueur dans la peau d'un personnage aveugle où le seul canal de communication utilisé est le son. Ce jeu est donc par nature inaccessible aux joueurs ayant un trouble de l'audition.

Les jeux adaptables sont les plus répandus, en effet les jeux récents intègrent de plus en plus de menus permettant aux joueurs de personnaliser les modalités d'interaction à leur convenance (vitesse de déplacement du dispositif de pointage, couleurs/contrastes, touches/boutons de contrôle, choix de la difficulté…). Ces paramètres permettent donc à tous les joueurs (y compris « handi ») d'adapter le jeu à leur convenance pour améliorer leur expérience de jeu.

Les jeux adaptatifs sont quant à eux plus rares car ils doivent intégrer un modèle du joueur ou de la résolution des défis proposés afin d'adapter automatiquement les interfaces ou les contenus aux difficultés rencontrées par les joueurs.

Les travaux en ergonomie nous éclairent également sur la manière dont les joueurs s'approprient un jeu vidéo. Les théories de l'activité visent à comprendre l'activité d'un sujet lorsqu'il tente d'atteindre un objectif [12]. L'activité est donc différente de la tâche qui, elle, est prescrite. Une tâche précise ce qui doit être fait (l'objectif) et la procédure à suivre pour atteindre cet objectif. L'activité est singulière, finalisée et médiatisée par un instrument [13]. L'instrument ici fait référence à un artefact associé à des schèmes d'utilisation [14]. Au regard des théories de l'activité nous considérons qu'un moteur de jeu vidéo est un artefact sur lequel les joueurs vont construire des schèmes pour interagir avec lui. Ces schèmes ne pourront se construire que si le joueur est en capacité d'accéder aux stimuli primaires (voire secondaire) et de produire une réponse en retour. Un jeu vidéo quant à lui exploite un moteur (l'artefact) et propose aux joueurs des objectifs souvent prescrits sous la forme d'une mission ou d'une quête à réaliser, d'un niveau à terminer ou d'un score à dépasser. Les règles du jeu contraignent les actions du joueur et calibrent la complexité d'une situation de jeu. Un jeu vidéo peut donc être abstrait à la notion de tâche selon les théories de l'activité. Ainsi, l'activité du joueur est observée à travers ses actions qui dépendent de ses compétences, de l'accessibilité du moteur et de la complexité des situations de jeu proposées (incluant les objectifs). Ce processus progressif, appelé genèse instrumentale, permet aux joueurs de s'approprier l'artefact (transformer l'artefact en instrument) et fait appel à deux types de transformation : l'instrumentalisation dans laquelle le joueur va adapter l'artefact à ses besoins (facette des jeux vidéo adaptables) et l'instrumentation dans laquelle l'artefact

[3] https://www.game-lover.org/, consulté le 16 janvier 2020
[4] WCAG = Web Content Accessibility Guidelines
[5] W3C = World Wide Web
[6] http://gameaccessibilityguidelines.com/full-list/, consulté le 16 janvier 2020
[7] http://www.ablindlegend.com/, consulté le 16 janvier 2020

influence les actions du joueur (facettes des jeux vidéo adaptés et adaptatifs).

Améliorer l'accessibilité d'un jeu vidéo consiste donc à favoriser la genèse instrumentale et donc à travailler les trois facettes de l'adaptation (adapté, adaptable et adaptatif). De nombreux travaux proposent des normes d'accessibilité minimales, des bonnes pratiques de conception et des stratégies d'adaptation afin de viser l'accessibilité universelle des jeux vidéo. La question du « quoi » est étudiée, mais la question du « comment » est quant à elle peu abordée. Quel impact l'ajout d'une nouvelle fonctionnalité non anticipée a-t-il sur le code d'un jeu en cours de développement ? Comment l'architecture logicielle du jeu peut-elle aider à intégrer de nouvelles fonctionnalités ?

## III. Entités-Composants-Systèmes : une architecture logicielle issue du jeu vidéo

Historiquement le développement de jeux vidéo repose majoritairement sur le paradigme de programmation orientée objet (POO) dont le C#, C++ et Java sont les langages de programmation les plus utilisés. Néanmoins les principes inhérents à la POO tel que l'encapsulation, l'envoi de message et l'héritage peuvent rendre difficile la maintenance et l'évolutivité d'un moteur de jeu vidéo. Avec une approche orientée objet, les développeurs modélisent les éléments du jeu sous la forme de classes pouvant être spécialisés en sous-classes, etc. Le processus de développement des jeux vidéo étant hautement itératif, l'ajout de nouvelles mécaniques de jeu ou modalités d'interaction peuvent entraîner des modifications de la modélisation initialement envisagée. Ces modifications ont alors des impacts forts sur les développements et requièrent des phases de restructuration de code importantes, coûteuses en temps de développement et sources de bugs. La rigidité de l'arbre d'héritage est donc un frein dans ce contexte [15].

L'Entités-Composants-Systèmes (ECS) est une architecture logicielle principalement utilisée pour le développement de jeux vidéo [16, 17]. Cette architecture utilise une approche orientée donnée et s'articule autour de trois concepts. Les entités (premier concept) représentent les objets du jeu mais ne contiennent ni donnée, ni méthodes. Une entité est une simple référence vers une collection de composants (deuxième concept) qui contiennent eux les données. Les composants décrivent les aspects d'une entité comme sa couleur, sa taille, sa vitesse, etc. Un composant peut être ajouté ou supprimé dynamiquement à une entité. Enfin les systèmes (troisième concept) définissent la logique de jeu. Ils accèdent aux composants des entités afin de les traiter et les mettre à jour. Ils modifient ainsi les données du jeu et mettent en œuvre la simulation.

ECS est une architecture logicielle où la simulation est dirigée par les données et se base sur la notion de composition au contraire de l'approche orientée objet focalisée sur l'encapsulation et l'héritage.

L'ECS a été développée pour répondre à deux problématiques : améliorer la modularité des codes informatiques et améliorer les performances des moteurs de jeu. Concernant la modularité, l'approche dirigée par les données permet d'ajouter de nouvelles mécaniques de jeu ou modalités d'interaction avec un impact limité sur le code existant. Pour intégrer une nouvelle caractéristique, le développeur doit (1) définir les composants nécessaires au stockage des données, (2) ajouter ces composants aux entités concernées et (3) implémenter les systèmes qui traiteront ces composants. Ainsi ajouter une nouvelle mécanique de jeu a un impact limité sur les codes existants. Du point de vue de la performance, l'ECS permet de contrôler l'organisation des données en mémoire et permet ainsi d'optimiser l'accès aux composants. Dans cet article nous étudions l'ECS pour sa promesse de modularité et nous souhaitons l'évaluer au regard de l'ajout de fonctionnalité d'accessibilité. Nous ne détaillons donc pas les questions d'optimisation.

Pour répondre aux besoins d'interaction spécifiques des joueurs, les travaux de Garcia *et al.* [4] abordent également cette approche d'architecture fondée sur les données et les composants en faveur du développement de jeux vidéo accessibles. Ils étudient l'intérêt de l'ECS pour créer différentes représentations au sein d'un jeu, modifiables en temps réel et sans impact sur la logique de jeu. Pour restituer une information sous différentes modalités, il suffit de créer des composants qui contiennent les données des différents stimuli (graphique, audio, haptique) et des systèmes spécifiques à chaque stimuli traités pour manipuler ces composants. Lorsque le jeu est en cours d'exécution, la présentation modale transmis au joueur ne dépend que du choix des composants attachés à une entité du jeu. Ainsi, il est possible de passer d'une représentation à l'autre en temps réel, en fonction des besoins et capacités du joueur offrant une expérience de jeu plus accessible. De plus, leurs travaux mettent en avant l'avantage de cette architecture à pouvoir utiliser une source de donnée externe (fichier XML) pour définir uniquement les paramètres spécifiques au jeu. Ainsi, les joueurs pourraient accéder à ce fichier pour créer et générer de nouveaux composants, personnaliser les entités, modifier les préréglages et créer des configurations par défaut adaptées à chaque profil de joueur.

Les avantages de l'ECS au *runtime* mis en avant par Garcia *et al.* [4] nous confortent dans notre hypothèse que cette architecture serait adaptée et utile pour modifier plus simplement un jeu vidéo existant dont l'accessibilité n'aurait pas été prise en compte lors de la conception.

## IV. Présentation et analyse du jeu E-LearningScape

Dans cette section nous présentons l'artefact sur lequel nous avons travaillé, il s'agit du jeu sérieux open source E-LearningScape[8] (voir Fig. 1). Ce jeu a été développé initialement sans prendre en compte la question de l'accessibilité mais a été entièrement développé en exploitant l'architecture logicielle ECS (ce jeu a été développé sous Unity avec le module FYFY[9]). Il s'agit donc d'un cas d'étude intéressant pour notre problématique que nous rappelons ici : l'architecture logicielle ECS est-elle avantageuse pour intégrer des fonctionnalités d'accessibilité non anticipées lors de la conception dans un jeu vidéo non accessible ?

[8] https://github.com/Mocahteam/E-LearningScape, consulté le 16 janvier 2020

[9] https://github.com/Mocahteam/FYFY, consulté le 16 janvier 2020

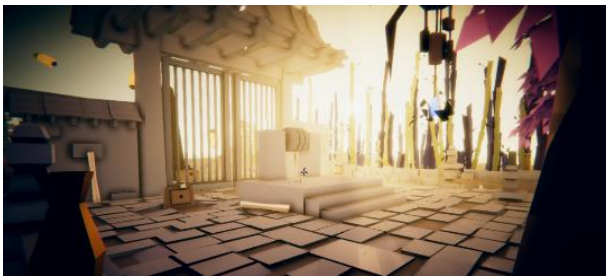

Fig. 1. Capture d'écran du jeu E-LearningScape

### A. *Présentation du jeu*

E-LearningScape est une adaptation numérique de l'*escape game* LearningScape[10] développée par SAPIENS[11] et le CRI[12]. Dans cette version originale du jeu, les participants jouent le rôle de marchands de sable immergés dans le rêve de Camille, une jeune enseignante-chercheuse à la veille de son premier enseignement. Leur défi : aider Camille à répondre à toutes ses questions avant son réveil.

L'adaptation numérique, que nous avons pris comme cas d'étude, fait collaborer deux à cinq participants autour d'un même écran d'ordinateur. Les joueurs évoluent dans un univers virtuel, découvrent des fragments de rêve leur donnant accès à du matériel dans le monde réel. Ils résolvent alors des énigmes à l'intérieur et à l'extérieur du jeu vidéo, ces deux facettes s'alimentant mutuellement. Le jeu est décomposé en quinze énigmes réparties dans quatre salles virtuelles. Les énigmes questionnent les joueurs dans le domaine de la pédagogie (par exemple : le changement de posture, la notion de feedbacks constructifs, le triangle pédagogique, la construction de grilles d'évaluation…). L'objectif de ce jeu est de tester et accroître leurs connaissances dans le domaine de l'éducation mais aussi de permettre aux joueurs de travailler l'esprit d'équipe et la cohésion.

### B. *Analyse du jeu E-LearningScape*

Nous avons analysé le jeu E-LearningScape à l'aide de la grille d'analyse fournie par le *Game Accessibility Guideline*. Ses recommandations sont réparties sur six catégories (motricité, cognition, vision, audition, parole et général). Chacune est décomposée en trois niveaux (basique, intermédiaire et avancé), c'est-à-dire allant des considérations les plus simples qui s'appliquent à la plupart des mécaniques de jeu, aux adaptations les plus complexes ou spécifiques à certaines mécaniques de jeu. Chacune d'entre elle obéit à un équilibre qui cible le nombre de bénéficiaire, l'impact entre les joueurs et le coût de mise en œuvre. Ainsi sur les 121 recommandations proposées nous en avons retenu 79 appropriées au jeu E-LearningScape. Par exemple nous avons éliminé toutes les recommandations relatives au multijoueur, au chat, aux dispositifs de réalité virtuelle ou mobile et aux sous-titrages de voix qui n'ont pas lieu dans ce jeu. Sur ces 79 recommandations, 10 d'entre elles étaient déjà intégrées au sein du jeu. Nous en avons intégré 33 supplémentaires afin de satisfaire le niveau basique des catégories motricité, cognition et vision (les catégories audition et parole n'étant pas sollicitées par le jeu E-LearningScape). La version actuelle du jeu intègre donc 43 recommandations sur les 79 applicables au jeu. Les diagrammes présentés dans la figure 2 donnent une vue du niveau d'accessibilité du jeu E-LearningScape avant et après intégration des nouvelles fonctionnalités. La série « référence » permet d'apprécier la ventilation des 79 recommandations retenues dans les différentes catégories et niveaux.

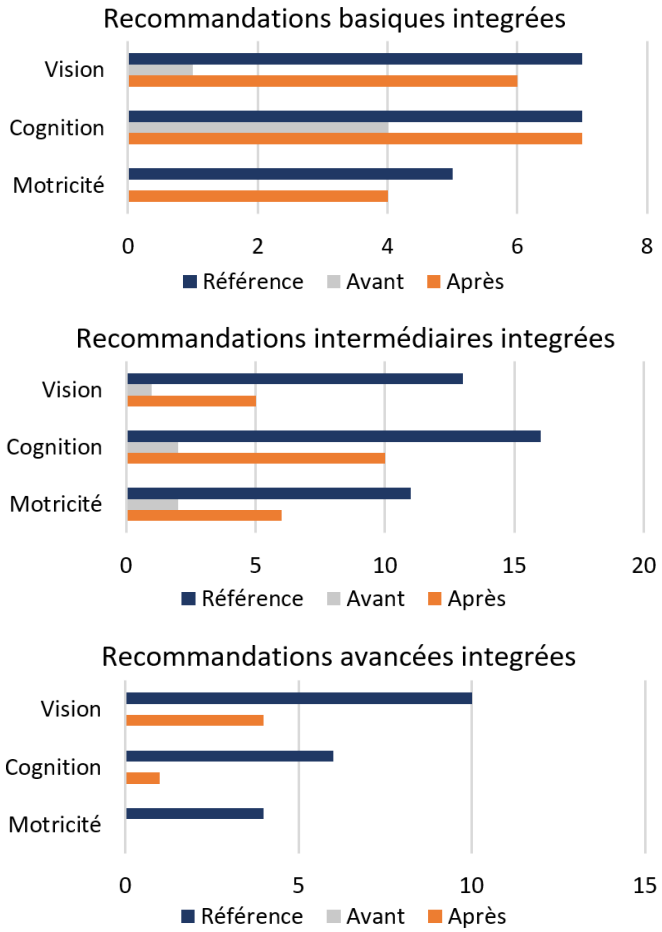


Fig. 2. Recommandations intégrées avant et après l'analyse du jeu

## V. Discussions des adaptations apportées

Suite à l'analyse du jeu E-LearningScape des modifications ont été apportées.

#### 1) *Jeu adapté*

Des intégrations et des changements ont été réalisés modifiant ainsi directement l'interface du jeu et l'interaction que le joueur aura avec la machine.

La manipulation du jeu a été augmentée pour permettre l'utilisation de différents contrôleurs en plus des périphériques

[10] https://sapiens-uspc.com/projets-innovants/learningscape-2/, consulté le 16 janvier 2020

[11] SAPIENS : Service d'Accompagnement aux Pédagogies Innovantes et à l'Enseignement Numérique de Sorbonne paris cité

[12] CRI : Centre de Recherches Interdisciplinaires

clavier/souris initialement prévus : des manettes de console comme la Xbox one, la Xbox adaptative (XAC) de Microsoft, le Nunchuck de Nintendo ou des contacteurs spécifiques. Ainsi, le jeu permet au joueur d'utiliser préférentiellement les périphériques mis à disposition ou d'en connecter d'autres sans avoir à toucher aux paramètres de réglages pour les intégrer. L'environnement Unity propose en natif les outils pour accueillir différents contrôleurs, l'ECS n'a donc pas apporté de plus-value dans ce cas. En lien avec les contrôleurs nous avons repris la navigation des menus pour permettre leur utilisation. Nous avons également ajouté un effet visuel pour permettre aux joueurs de mieux identifier le bouton se trouvant sous le pointeur du dispositif de pointage utilisé. Là encore Unity propose des outils en natif efficaces pour définir l'ordre de navigation dans les interfaces et les différents états de mise en évidence des menus interactifs indépendamment des périphériques d'entrée utilisés.

D'autres fonctionnalités ont pu être intégrées sans avoir recours à la couche ECS car relatives à des paramètres graphiques statiques directement paramétrable dans l'éditeur d'Unity : révisions des couleurs utilisées en limitant les rouges et les verts et ajout d'un contour noir aux éléments textuels.

Un couplage audio a été ajouté aux objets interactifs des différentes scènes. Dans ce dernier cas, l'approche ECS a permis d'intégrer cette fonctionnalité rapidement. En effet un système était déjà en place permettant de mettre en surbrillance les objets interactifs. Nous avons ajouté une seule ligne de code à ce système pour lui demander d'ajouter dynamiquement à l'entité concernée un nouveau composant décrivant le son à jouer. Nous avons ensuite créé un nouveau système permettant de gérer ces composants ajoutés dynamiquement et jouer le son correspondant. L'impact sur le code existant a donc été minime et la notification sonore reste indépendante de la notification visuelle. Nous avons ainsi exploité ce système pour d'autres cas d'usage demandant une notification sonore (click sur un bouton, succès/échec à une énigme). Avec une approche classique d'Unity nous aurions dû créer un nouveau script et l'attacher à la main à tous les objets interactifs du jeu avec un risque d'oubli de certains objets de jeu.

L'ECS a également été utile pour facilement identifier les différentes fonctionnalités déjà en place dans le jeu en vue de les modifier. En effet, chaque fonctionnalité étant implémentée par un système, modifier cette fonctionnalité consistait à identifier le système concerné et concentrer l'effort de compréhension du code sur ce système. Nous avons ainsi modifié le système gérant les objets interactifs pour permettre au joueur de maintenir un bouton enfoncé ou répéter des pressions successives ; et procéder à des glisser/déposer sans avoir à maintenir un bouton enfoncé. Nous avons aussi modifié le système gérant le contrôle de la caméra pour ajouter une fonction de zoom et de changement de point de vue (première/troisième personne).

*2) Jeu adaptable*

Pour permettre aux joueurs d'adapter davantage l'interface du jeu à son profil, un menu de personnalisation a été créé. Nous y avons intégré des options de contrôle, graphiques et sonores pour permettre la désactivation d'animation, la modification de la vitesse de déplacement et de rotation de la camera, le réglage du volume de la musique du jeu et des effets sonores, le réglage de la taille du curseur de visée pour aider à la sélection d'objet interactif (boutons, indices, champs de saisie) et la modification de l'intensité lumineuse de la scène. Pour toutes ces modifications, l'approche Unity classique était efficace.

L'ECS a apporté un intérêt pour procéder à des modifications massives sur les objets de la scène. Nous avons par exemple ajouté la possibilité de basculer les textes de la police non accessible par défaut à la police Arial. Avec une approche classique d'Unity, nous aurions dû soit ajouter un nouveau script à chaque objet textuel de la scène, soit créer un gestionnaire de textes référençant tous les objets textuels. Dans ces deux approches le risque est d'introduire des erreurs par la suite si un nouvel objet textuel est ajouté (oubli d'ajouter le script d'accessibilité au GameObject ou d'inscrire le GameObject au manager de texte). Avec l'approche ECS, nous avons créé un nouveau système qui, à l'aide des familles, récupère tous les objets de la scène contenant les composants requis à l'affichage de texte. Lorsque l'utilisateur bascule d'une police à l'autre, le système modifie l'ensemble des textes du jeu. Ajouter un nouveau texte accessible au jeu ne demandera alors aux concepteurs qu'à créer ce nouveau texte et l'ajouter à la scène sans se préoccuper des dépendances particulières liées à l'accessibilité (ajout d'un script particulier ou abonnement du GameObject à un gestionnaire particulier). Le même principe a été suivi pour contrôler les effets de transparence des fenêtres du jeu.

L'utilisation de l'approche ECS couplée à une approche classique d'Unity a été pertinente pour réaliser ce premier travail d'adaptation du jeu vers une accessibilité universelle. L'ECS nous a permis d'agir sur un ensemble d'élément, en leur ajoutant ou supprimant des comportements, avec des modifications minimes et peu coûteuses sur la mécanique et l'interface initiale du jeu. Ainsi, nous mettons à disposition des joueurs des options de réglage simples, peu nombreuses et qui touchent un large profil de joueur, dans le but d'augmenter sa perception de l'environnement de jeu, son expérience utilisateur et son confort de jeu.

*3) Jeu adaptatif*

Nous avons ajouté une dimension adaptative au jeu en intégrant un module de suivi des activités du joueur pour calculer en cours de simulation la meilleure aide à apporter au joueur en fonction de ses difficultés. Le module de suivi que nous avons intégré est compatible avec l'approche ECS [18] et permet de suivre l'activité des joueurs sur des objets précis du jeu ou sur des collections d'objets (décrites via les familles de l'ECS).

L'étape ayant impacté le plus fortement le code existant était la génération des traces. Il s'agissait ici de repérer dans les programmes existants les étapes validant les actions du joueur. Produire la trace consistait alors à créer un composant (spécifié par le module de suivi) contenant les données relatives à l'action. La décomposition de la logique du jeu en système a permis de rapidement identifier les systèmes concernés (celui gérant l'inventaire du joueur, celui gérant les déplacements, celui gérant l'activation des objets interactifs…).

Nous avons ensuite créé un système qui traite les composants produits en retour par le module de suivi. Ce système détermine l'aide à apporter au joueur en fonction du label associé à chaque

action de jeu (positif ou négatif) qui se cumulent au cours du temps, de la progression du joueur dans les énigmes et du temps restant avant la fin de la séance. Les aides proposées peuvent être de différentes natures : suggérer de fouiller une zone du jeu, indiquer la position d'un indice, expliquer une mauvaise réponse et acter que le joueur a tous les éléments en main pour résoudre une énigme. Là encore l'architecture ECS apporte de la souplesse pour pouvoir mettre à jour le système d'aide afin de, par exemple, modifier les règles impactant la prise de décision relative au déclanchement de l'aide ou à sa nature.

## VI. CONCLUSION

Nous avons débuté cet article en soulignant l'importance de prendre en compte la question de l'accessibilité dans les jeux vidéo. Nous nous sommes intéressés au cas où la prise en compte de l'accessibilité n'avait pas été anticipée lors des premières phases de développement et devait être intégrée à posteriori. Nous avons donc étudié l'intérêt que pouvait apporter l'architecture ECS dans ce contexte. Notre encrage théorique s'appuie sur les théories de l'activité et les trois dimensions de l'adaptation (adapté, adaptable et adaptatif).

L'analyse des corrections apportées au jeu E-LearningScape nous a montré que l'architecture Entités-Composants-Systèmes était un atout pour opérer des transformations impactant un grand nombre d'objets de jeu et éviter des manipulations redondantes sources d'erreurs. Nous avons vu également que le principe de modularité et de décompositions des fonctionnalités du jeu en système, nous a permis de rapidement identifier les parties de programme à modifier. Concernant les limites, l'ECS n'apporte pas de plus-values pour traiter des objets de jeu singulier. Dans ce cas les fonctionnalités classiques d'Unity nous ont semblé plus efficaces.

Nous notons toutefois que les deux approches s'articulent parfaitement et se complètent. L'architecture Entités-Composants-Systèmes semble donc être un atout pouvant contribuer à aider à l'intégration de fonctionnalités d'accessibilité dans les jeux vidéo et notamment lorsque celles-ci n'ont pas été anticipées.

Bien que cette architecture ait émergée de besoins issus du développement de jeux vidéo, ces avantages en terme de modularité pourraient servir des applications plus générales et il nous semblerait intéressant de poursuivre ces travaux pour l'accessibilité d'autres types d'applications (mobile, web, bureautique, simulateur…).

## RÉFÉRENCES


[1] D. Grammenos, A. Savidis et C. Stephanidis, «Designing Universally Accessible Games,» *ACM Computers in Entertainment*, vol. 7, n° %11, p. 29, 2009.

[2] R. J. McCrindle et D. Symons, «Audio space invaders,» chez *Third International Conference on Disability, Virtual Reality and Associated Technologies*, 2000.

[3] B. Yuan, E. Folmer et F. Harris, «Game accessibility: A survey,» *Universal Access in the Information Society*, vol. 10, pp. 81-100, 2011.

[4] F. E. Garcia et V. P. de Almeida Neris, «A Data-Driven Entity-Component Approach to Develop Universally Accessible Games,» *Universal Access in Human-Computer Interaction*, vol. 8514, pp. 537-548, 2014.

[5] K. Bierre, J. Chetwynd, B. Ellis, D. M. Hinn, S. Ludi et T. Westin, «Game Not Over: Accessibility Issues in Video Games,» *Human-computer interaction*, 2005.

[6] D. Grammenos, A. Savidis et C. Stephanidis, «Unified Design of Universally Accessible Games,» chez *Universal Access in Human-Computer Interaction*, Beijing, 2007.

[7] T. Rafaillac et S. Huot, «Polyphony: Programming Interfaces and Interactions with the Entity-Component-System Model,» chez *11th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, Valencia, Spain, 2019.

[8] T. Raffaillac et S. Huot, «Application du modèle Entité-Composant-Système à la programmation d'interactions,» chez *30eme conférence francophone sur l'Interaction Homme-Machine*, Brest, 2018.

[9] P. Gestwicki, «The entity system architecture and its application in an undergraduate game development studio,» chez *International Conference in the Foundations of Digital Games*, 2012.

[10] K. Bierre, M. Hinn, T. Martin, M. McIntosh, T. Snider, K. Stone et T. Westin, Accessibility in Games: Motivations and Approaches, The International Game Developers Association, 2004, p. 37.

[11] K. Sehaba, «Adaptation dynamique des Environnements Informatiques pour l'Apprentissage Humain,» Université Lumière Lyon 2, 2014.

[12] F. Daniellou et P. Rabardel, «Activity-oriented approaches to ergonomics: some traditions and communities,» *Theoretical Issues in Ergonomics Science*, vol. 6, n° %15, pp. 353-357, 2005.

[13] S. Nogry, F. Decortis, C. Sort et S. Heurtier, «Apports de la théorie instrumentale à l'étude des usages et de l'appropriation des artefacts mobiles tactiles à l'école,» *Sticef*, vol. 20, p. 35, 2013.

[14] P. Rabardel, Les hommes et les technologies; approche cognitive des instruments contemporains, Patis: Armand Colin, 1995.

[15] T. Hottou, «Entity Component System et le C++,» 4 Juillet 2017. [En ligne]. Available: https://www.supinfo.com/articles/single/4673-entity-component-system-c. [Accès le 19 Mars 2020].

[16] S. Bilas, «A Data-Driven GameObject System,» chez *Game Developers Conference*, 2002.

[17] B. Capdevila, «Serious game architecture and design: modular component-based data-driven entity system framework to support systemic modeling and design in agile serious game developments,» Université Pierre et Marie Curie, Paris, 2013.

[18] M. Muratet, A. Yessad, T. Carron et A. Ramolet, «Un système d'aide à l'analyse des traces des apprenants dans les jeux sérieux,» *Sticef*, vol. 25, 2018.