Étude des caractéristiques des joueurs de Star Shaman :analyses statistiques des données d'un jeu en réalité virtuelle

Aude Bertrand

le cnam
intd

Étude des caractéristiques des joueurs de Star Shaman : analyses statistiques des données d'un jeu en réalité virtuelle.

Mémoire pour l'obtention du

Master Sciences humaines et sociales

mention humanités numériques

Parcours Mégadonnées et analyse sociale
(MEDAS)

Aude BERTRAND

Date et lieu de la soutenance

- 8 juillet 2021
- Cfa du CNAM, 61 rue du Landy LA PLAINE SAINT-DENIS

Membres du jury

- Mme Béatrice ARRUABARRENA, directrice
- Mme Majda SEGHIR, tutrice de mémoire
- M. Olivier Piasentin, tuteur d'entreprise

Promotion (2019-2021)

BERTRAND Aude. Étude des caractéristiques des joueurs de Star Shaman : analyses statistiques des données d'un jeu en réalité virtuelle.
Mémoire professionnel INTD, Titre I, Conservatoire national des arts et métiers
– Institut national des Sciences et Techniques de la Documentation, 2021, 114p. Promotion 4.

Ce mémoire présente les techniques de data mining et de statistique mises en œuvre pour comprendre les caractéristiques des joueurs de Star Shaman, un jeu en réalité virtuelle du studio Ikimasho. Qui sont les joueurs ? Comment évoluent-ils dans le jeu ?
L'analyse de données dans le jeu vidéo a pour but, depuis une dizaine d'années, de garder les joueurs actifs dans le jeu. Ces pratiques concernent aussi le jeu sérieux, notamment dans des domaines tels que la santé qui utilise beaucoup le jeu en réalité virtuelle comme technique de recherche.
Dans la tradition de l'analyse de données dans le jeu vidéo, l'auteur présente la technique de clustering des joueurs, visant à distinguer des groupes de joueurs et leurs caractéristiques propres. Une analyse des facteurs d'abandon du jeu est ensuite développée au moyen d'une régression logistique et d'une forêt aléatoire.

Descripteurs

Réalité virtuelle
Clustering
Partitionnement
Régression logistique
Forêt aléatoire
Jeu vidéo
Data mining
Profilage
Analyse statistique
Apprentissage non supervisé

This master thesis presents data mining technics and statistical technics used to find the characteristics of Star Shaman's players (a videogame in virtual reality by Ikimasho). Who are they ? How do they interact with the game ?
The data analysis in the videogame field is used to hold players in the game. This is also known in serious gaming and other fields like health (using VR a lot, for research purpose). Many researches have used gaming as a technique and do data analysis.
Following videogame practices, the author presents the clustering method used to find groups amongst players and their characteristics. She also analyses the factors in abandoning the game, using logistic regression and random forest.

Keywords

Game analytics
Virtual reality
Clustering
Logistic regression
Random forest
Videogame
Data mining
Profiling
Statistical analysis
Unsupervised learning

TABLE DES MATIERES

REMERCIEMENTS

J’aimerais ici remercier profondément Madame Majda Seghir pour son suivi attentif durant tout le processus d’écriture et d’analyse de ce mémoire.

Mes pensées vont également à Chérif Younis et Gauthier Dine, respectivement directeur technique à Ikimasho et data scientist chez GamePoint, pour leurs conseils et aide dans la mise en place du pipeline de données. Je leur dois une grande partie de mon savoir. Sans leur patience et leur collaboration, je n’aurais pu progresser dans ce travail.

Mes pensées vont également à Olivier et Yann pour avoir accepté de m’inclure dans la vie du studio et de me faire confiance dans mes missions.

Enfin, je remercie Andrée Bertrand, fidèle relectrice de tous mes travaux depuis plus de dix années.

INTRODUCTION

Ce travail porte sur l'analyse de données de jeux vidéo et se base sur une expérience en alternance au sein du studio Ikimasho de janvier 2020 à janvier 2021. Ikimasho est une start-up de production de jeux vidéo en réalité virtuelle, installée à Paris, et qui a sorti son premier jeu, Star Shaman, en octobre 2021. Mon travail, en tant qu'apprentie data analyst, comportait un volet d'analyse des données des joueurs. Son objectif final était de comprendre les types de joueurs de Star Shaman, la manière dont ils évoluaient dans le jeu et ses raisons. Cela devait aussi, à terme, permettre de déterminer les éléments que les joueurs appréciaient et ainsi d'orienter le développement futur du jeu. Ceci a posé d'autres questions : comment suivre les comportements des joueurs ? Quels sont les paramètres et les méthodes à employer ? Très vite, l'objectif s'est recentré sur l'identification de groupes de joueurs et leurs caractéristiques, afin de pouvoir ensuite analyser leur évolution dans Star Shaman séparément.

Ce mémoire a pour objectif de rendre compte de l'analyse des joueurs, en présentant la mise en place du pipeline de gestion des données de jeu et les méthodes employées pour l'analyse. Le clustering a été privilégié dans un premier temps pour constituer des groupes de joueurs ad hoc, en fonction des données de sessions de jeu. J'ai également analysé les facteurs de départ des joueurs, dans le but de comprendre ce qui influençait les joueurs dans un sens ou un autre. Ces analyses ont posé des questions depuis le choix des données à employer, la méthode pour les extraire, jusqu'à la vérification de la validité des modèles utilisés. Ce travail présente donc l'ensemble du processus d'analyse des données de jeu de Star Shaman.

La première partie présente la société Ikimasho, son organisation et mes activités en son sein. Nous reviendrons aussi sur l'histoire de la réalité virtuelle qui est la particularité de ce studio, en l'incluant dans l'histoire du jeu vidéo.

En deuxième partie, je commencerai par un point sur le vocabulaire spécifique au jeu vidéo à Star Shaman et un point sur les données de ce milieu. Je présenterai, en outre, une revue de la littérature professionnelle. En effet, ce mémoire s'inscrit dans la lignée d'autres travaux. Ce sera l'occasion de montrer l'emploi des données dans le jeu en général, que ce soit le jeu vidéo ou le jeu sérieux. En effet, de plus en plus d'activités utilisent les codes du jeu, dans un but non-ludique. Plusieurs travaux portent sur l'analyse de données dans ce cadre et parfois même en réalité virtuelle. Enfin, cette partie présentera aussi l'importance du clustering dans le jeu vidéo et ses principales méthodes.

En troisième et dernière partie, je présenterai l'ensemble des analyses effectuées. Nous commencerons par l'identification des données et la construction du pipeline. Ensuite, je présenterai le clustering effectué pour tenter de trouver les caractéristiques de nos groupes de joueurs. Enfin, je m'attarderai sur la régression logistique et la forêt aléatoire, deux méthodes très différentes pour identifier les facteurs poussant les joueurs à continuer de jouer ou non.

I. PREMIERE PARTIE : LA SOCIETE IKIMASHO

DANS SON CONTEXTE

Dans cette première partie, je propose une présentation de la société Ikimasho au sein de laquelle j'ai réalisé ce travail. Je détaillerai ses activités et sa place dans le milieu de la réalité virtuelle. Dans un second temps, j'expliquerai mes activités en tant que data analyst. Sachant tout cela, nous pourrons alors commencer à appréhender les raisons de cette analyse.

1. UN JEUNE STUDIO DE JEUX EN REALITE VIRTUELLE

a) UNE START-UP DANS UN DOMAINE DE NICHE

Présentation du studio

Ikimasho a été fondé en 2018 par Messieurs Olivier Piasentin et Yann Suquet. L'activité principale du studio est de créer et produire des jeux vidéo en réalité virtuelle, pour son compte ou pour le compte de tiers (sous-traitance).

On y retrouve l'organisation d'une start-up. L'équipe est petite : une quinzaine de personnes, tous métiers confondus. Les profils, lors de mon apprentissage, étaient ceux d'experts dans leurs domaines, jouissant d'une autonomie de travail (voir section B, ci-dessous). L'objectif était de créer une équipe autonome développant le jeu dans les divers domaines de compétence de chacun. Une autre équipe, formée de compétences transversales au studio, était en support. L'analyse de données, le marketing et les activités de gestion du studio en faisaient partie. Ceci est confirmé par la description du studio sur le site l'AFJV (Autorité Française des Jeux Vidéos) :

> *« […] Nous prenons d'abord soin de nos équipes de développement car nous pensons que ce sont elles qui créent le plus de valeur pour nos joueurs. Le studio est donc structuré pour leurs fournir tout le soutien possible de manière à ce qu'à leur tour, elles puissent servir nos joueurs.*
>
> *Dans tout ce que nous faisons, nous déléguons la responsabilité de la prise de décision à la personne la plus à même de faire le bon choix. […]»*[1]

On retrouve cette vision d'experts autonomes apportant leurs propres cultures et compétences sur le site même d'Ikimasho :

> *« We believe in the unifying power of a clear vision, and in unleashing and harnessing our team's creativity at every level. Everything we do, we break down into clearly-defined spheres of ownership that are handled at the lowest practical level and do not overlap. In her sphere of ownership, the leader sets a clear, unifying vision and then unleashes her team's creativity and harnesses their ideas to contribute to the overall effort of the studio. »*[2]

Cette liberté d'action permet aussi des activités de recherche, notamment en termes de systèmes de réalité virtuelle liés à la performance. De ce fait, Ikimasho a reçu le label Jeune Entreprise Innovante (JEI) qui permet à l'Etat de soutenir des PME indépendantes, nouvelles, de moins de 8 ans et dont 15% du budget est lié à la R&D.

[1] https://www.afjv.com/societe/1075-ikimasho.htm
[2] http://www.ikimasho.games/

Situer la VR dans l'histoire du jeu vidéo

L'activité du studio se place dans un contexte d'expansion de la réalité virtuelle et d'épanouissement du jeu vidéo. L'industrie de ce dernier est en effet florissante. Il est souvent fait mention de l'effondrement du jeu vidéo en 1983 mais il s'agit d'un phénomène américain et l'industrie française s'est, elle, développée au même moment d'après Colin Sidre (2020). Au départ très liée aux consoles et bornes d'arcade, l'émergence de nouveaux types de jeux a entretenu cette industrie.

Dans les années 1990 émergent les jeux en réseau qui deviendront des jeux en ligne grâce à l'arrivée d'Internet (Lafrance et Heaton, 1994). D'abord reliés en réseau dans une salle d'arcade ou par le réseau téléphonique, les joueurs commencent à créer des communautés virtuelles. Lafrance et Heaton parlent des premiers jeux en réalité virtuelle dans les salles d'arcade. Chaque joueur se trouvait dans un espace fermé avec un casque diffusant l'image. La cabine était reliée à d'autres et les joueurs échangeaient par micro. Les jeux en réseau se sont ensuite développés grâce à ce que l'on appelait « l'autoroute de l'information », c'est-à-dire le réseau Internet. Les Jeux de Rôle en Ligne Massivement Multijoueur (JDRMM ou MMPORGs en anglais, voir glossaire et II.1) ont fleuri.

Depuis 2005 environ, le jeu vidéo a évolué à la fois dans ses plates-formes et dans son contenu (THIBAULT, 2013). Le modèle économique *Free-to-play*, consistant à permettre de jouer gratuitement en offrant des services ou objets payants, commençait à émerger[3]. Avec l'arrivée des téléphones portables et plus précisément des smartphones, le jeu est devenu mobile et a bénéficié de cette nouvelle économie. Des contenus adaptés aux circonstances de jeu (temps de session plus courts) ont vu le jour. Enfin, avec l'essor des réseaux sociaux puis celui de la tablette, le jeu vidéo a encore augmenté le nombre de ses plates-formes (THIBAULT, 2013).

Aujourd'hui, le marché du jeu vidéo génère des revenus très élevés et dont la croissance est forte. L'AFJV[4] ,citant le rapport Nielson de 2020 sur le jeu vidéo, montre que 126.6 milliards de dollars dans le monde ont été générés par le jeu vidéo en 2020. Plus précisément :

> *« Les jeux gratuits ont continué de générer la grande majorité des revenus des jeux vidéo (78%) [...]*
>
> *Le marché mobile a connu une croissance de 10% en 2020 et représentait 58% du marché total des jeux vidéo. Malgré les confinements et la réduction des déplacements, les revenus mobiles sont restés stables car la majorité des joueurs mobiles (62% aux Etats-Unis) utilisent déjà les appareils mobiles comme plate-forme de jeu principale. »*[5]

La réalité virtuelle (abrégée VR) pourrait être perçue comme une nouvelle plate-forme du jeu vidéo à travers ses casques. Elle rassemble cependant bien d'autres contenus que le jeu : des créations audiovisuelles, des réseaux sociaux comme VRChat et plus anecdotiquement BigScreen (possibilité de regarder des vidéos à plusieurs), ou encore des vidéos de format 4K ou 360° disponibles sur Youtube.

La réalité virtuelle a une histoire plus ancienne qu'il n'y paraît : Sherman & Craig (2018) remontent d'ailleurs aux 19e siècle en recherchant les ancêtres de la réalité virtuelle. On l'a vu, le terme est utilisé pour les jeux d'arcade en réseau de la fin des années 1980. Un brevet pour un premier casque a été déposé par la société VPL Research fin de la décennie. Les recherches se sont poursuivies dans les années 1990, accompagnées de la parution du livre *Virtual Reality* d'Howard Rheingold en 1991. Toutefois, c'est en

[3] Player modeling. Yannakis

[4] https://www.afjv.com/news/10452_les-jeux-video-ont-rapporte-126-6-milliards-de-dollars-en-2020.htm

[5] Idem.

1957 que le cinéaste Morton Heilig invente l'ancêtre de la VR (selon le site français de référence Réalités virtuelles[6]): le sensorama. Resté théorique, celui-ci permet de s'immerger dans un film avec un casque et un système de son. Les premières apparitions de systèmes en réalité virtuelle ont lieu la décennie suivante avec les simulateurs de vol. Un casque a été développé en 1970 à l'université de l'Utah, permettant de voir une scène sous tous les angles en tournant la tête. Cette invention fut suivie d'un gant haptique, c'est-à-dire, un gant permettant de suivre les mouvements de la main et de renvoyer des stimuli à celle-ci. La société Atari fait également plusieurs recherches avant de s'arrêter avec la crise du jeu vidéo en 1983 aux USA. L'étape marquante pour le jeu vidéo est la sortie du casque Sega VR entièrement destiné au gaming.

Les casques modernes des années 2010 sont plutôt le fruit des recherches sur les systèmes virtuels représentant la réalité filmée. On rapproche souvent la réalité virtuelle des travaux de Google sur Google Map et Google Earth. En effet, après une période assez creuse pour la réalité virtuelle au début des années 2000, plusieurs projets sont démarrés et on assiste à une profusion de modèles de casques produits. Google travaille sur le Cardboard, un casque permettant de glisser un smartphone dans la visière. Celui-ci sort en 2015 mais ressemble plus à un système de stéréoscopie. En 2013, la société Oculus crée un premier casque en mode développeur, reliable à un ordinateur. Trois ans plus tard, cette même société sort l'Oculus Rift (en collaboration avec Facebook qui rachète l'entreprise en 2014). En 2015, Samsung sort un casque aussi. L'année d'après, c'est au tour de Valve, éditeur de jeu vidéo, et HTC, entreprise d'informatique, de mettre sur le marché le HTC Vive (Sherman & Craig, 2018). La même année, le système PlaystationVR permet au joueur de console d'avoir un système de VR.

Pour résumer, tous ces casques sont conçus pour le jeu et doivent être reliés à un ordinateur. Le prix du casque et des contrôleurs (manettes à tenir dans les mains) dépasse les 1000 euros. Ces casques de qualité supérieure sont destinés aux gros joueurs.

Oculus crée l'événement en 2018 en sortant un casque autonome et beaucoup moins cher, l'Oculus Go. L'année suivante, la firme sort en même temps que son Rift S (casque traditionnel), l'Oculus Quest, nouveau casque autonome dont la qualité visuelle est fortement supérieure. Il peut s'utiliser seul aussi bien que relié à un ordinateur. Son succès a contribué à l'expansion de la réalité virtuelle en augmentant fortement le nombre de personnes possédant un casque. Une deuxième version avec une qualité visuelle supérieure est sortie en octobre 2020 avec un prix encore plus bas. Cette montée en puissance de la réalité virtuelle a engendré des revenus importants. L'année 2020 a été également un tournant grâce à la crise du COVID qui a favorisé la consommation de contenus en réalité virtuelle.

Toujours selon le rapport Nielsen 2020, cité par l'AFJV[7], les jeux en réalité virtuelle ont eu une croissance de 25% en 2020 par rapport à 2019. Ils ont rapporté 589 millions de dollars dans le monde. Cela semble peu par rapport au jeu vidéo dans son ensemble, mais le marché est encore une niche. Le nombre de joueurs est difficile à estimer car le nombre des ventes de casques Oculus, produits par Facebook, ne sont pas publiques. Il existe des estimations d'après le nombre d'envois par bateau. Seul le site *Finance online*[8] donne un nombre clair et ce, pour les Etats-Unis seulement. Or, les USA sont un des premiers pays consommateur de réalité virtuelle, en concurrence actuellement avec la Chine quant aux montants des achats[9]. Les usagers y seraient 57,4 millions en 2020. C'est sans compter les joueurs occasionnels qui se rendent dans une salle de jeux en VR.

[6] https://www.realite-virtuelle.com/histoire-vr-7-etapes-1511/#:~:text=7%20%C3%A9tapes%20marquantes.-,L'histoire%20de%20la%20VR%20%3A%201957%2C%20le%20Sensorama%20est,utilisateurs%20dans%20un%20monde%20imaginaire.&text=Quoi%20qu'il%20en%20soit,bases%20de%20la%20r%C3%A9alit%C3%A9%20virtuelle.

[7] https://www.afjv.com/news/10452_les-jeux-video-ont-rapporte-126-6-milliards-de-dollars-en-2020.htm

[8] https://financesonline.com/virtual-reality-statistics/

[9] https://www.statista.com/statistics/1076686/ar-vr-spending-worldwide-by-region/

Il existe finalement peu d'études publiques sur le marché de la réalité virtuelle car c'est un marché trop récent. Beaucoup ont été faites par les parties prenantes (salles de VR, fabricants) ou les financeurs (CSA, Union Européenne) à la sortie des premiers casques, et n'ont pas été mises à jour. On trouve néanmoins des articles de plusieurs sites web de référence, tels que l'Association Française pour le Jeu Vidéo[10], *RoadToVR*[11] ou encore *UploadVR*[12]. L'entreprise d'analyse de marchés *Grand View Research* a sorti en mars dernier son rapport payant sur les tendances du marché de la réalité virtuelle entre 2021 et 2028[13]. La commission européenne a commandé un rapport sur la réalité virtuelle, publié en 2017, qui est également une bonne source d'information (Bezegova et al., 2017).

En premier lieu, le marché de la VR concerne de très nombreux secteurs. Le rapport européen mentionne aussi bien la recherche que l'industrie dans des secteurs tels que la santé, la formation, la simulation industrielle. Il semble qu'en 2017 la VR soit assez bien implantée en Europe grâce à son réseau d'infrastructures de recherche : le rapport cite l'Angleterre, la Suède et la France comme des hubs. La VR en France est notamment liée au secteur de l'optique et de l'audio médicaux. Le rapport mentionne en outre que le tissu industriel français renforce la capacité à adopter de nouvelles technologies. *Laval Virtual* est également cité comme le premier salon d'importance dédié à la VR. Grand View Research[14] ajoute aussi les secteurs de l'aérospatial et du commerce dans son article de mars. Entre temps, la VR industrielle s'est en effet fortement développée.

Globalement, le sources citent peu les jeux en réalité virtuelle. Le rapport européen (Bezegova et al., 2017) dresse un panorama des entreprises du secteur en 2017 dans lequel ne figurent pas de studio de jeux vidéo. On retrouve les producteurs de matériels, les institutions de recherche, les financeurs et les influenceurs tels que les festivals. Il y a bien des start-ups citées mais pas de studio. Seraient-ce qu'il y en a, à l'époque, trop peu ? Il faut rappeler que le casque HTC Vive n'est sorti qu'en 2016 et fut l'un des premiers. Les dernières statistiques montrent que le marché de la VR s'est maintenant déplacé en Asie, et notamment sur le marché chinois où les consommateurs adoptent la réalité virtuelle massivement. Quant aux studios de jeu spécifiquement (contrairement aux organisateurs d'événement ou créateurs de produits promotionnels en VR), ils sont encore peu nombreux et relativement petits à l'échelle des studios de jeu vidéo traditionnels. Il en existe une dizaine en France et presque soixante en Europe. Leur taille moyenne est de sept personnes mais les plus gros peuvent comptabiliser jusqu'à trente personnes.

Les dernières tendances en matière de jeu, et que l'on trouve encore très peu, sont les applications sociales et les jeux multi-joueurs. En effet, les contenus en réalité virtuelle sont massivement conçus pour des parties en solo. Or, les tendances vont aux échanges entre les usagers que ce soit sur le réseau social *VR Chat* ou dans le jeu *No man's sky* sorti en 2019.

b) ORGANISATION DE L'EQUIPE ET METIERS

Au studio Ikimasho, l'organisation théorique se faisait par équipes de jeu. Ces dernières étaient soutenues par les profils transversaux : le mien dans la donnée, celui de la responsable du marketing et ceux de nos deux dirigeants. Chaque équipe de jeu devait comprendre des spécialistes des divers métiers, nécessaires au développement d'un jeu vidéo. Toutefois, Star Shaman étant le premier jeu du studio, tout le personnel a travaillé sur le projet. Star shaman est un jeu dans lequel le joueur interprète un shaman des étoiles dont la mission est de régénérer la vie dans les systèmes solaires qu'il traverse. Une faction, dite des « architectes de l'entropie », a en effet entrepris de tuer toute vie et de modeler planètes et

[10] https://www.afjv.com/recherche.php?q=r%C3%A9alit%C3%A9+virtuelle
[11] https://www.roadtovr.com
[12] https://uploadvr.com/
[13] https://www.grandviewresearch.com/industry-analysis/virtual-reality-vr-market
[14] Ibidem.

étoiles selon un concept de perfection géométrique. Les shamans ont donc la charge de régénérer les lunes qui enferment des éléments, les briques élémentaires de la vie, à la suite du passage des architectes. Lorsque toutes les lunes d'une planète sont régénérées, la planète revit. Une fois rassemblé X nombres de chaque élément, le système solaire tout entier peut revivre (voir présentation complète du jeu en I.3).

Pour développer ce jeu, deux grands pôles de compétence sont mis en œuvre : les artistes et les développeurs. On peut noter que de plus grands studios ont une infinité de sous-groupes et métiers supplémentaires.

Parmi les artistes, on trouvait quatre personnes dont les tâches allaient de la conception à la réalisation du contenu graphique du jeu. En effet, il existe une variété de métiers artistiques utiles à la création de jeux vidéo, et chaque phase d'un projet possède ses propres activités également. Le studio étant petit, les personnes étaient multi-tâches mais on retrouvait des spécificités chez chacune. Au début du projet, cette équipe réalisait les « concept arts », c'est-à-dire les premières planches montrant les idées de contenus et leurs styles graphiques. On peut noter que dans les plus grands studios, il existe des « concept artists » uniquement dédiés à cela. Je suis arrivée dans l'équipe alors que le jeu était en développement et j'ai donc vu les dernières versions des concept art qui m'ont aidée à comprendre rapidement l'univers du jeu. Ce fut d'autant plus important que, comme je le découvrais par la suite, Star Shaman était une évolution initiale du projet de la société, Battle Caster. Le style de ce projet initial, totalement différent et très réaliste, ne permettait pas de lancer le produit final sur le casque Quest, nouvellement apparu sur le marché.

Parmi les artistes du studio, on trouve les spécialistes de la 2D. Ils sont responsables des contenus en 2D du jeu, tels que les panneaux du menu ou des paramètres. Ces contenus sont créés sous forme d'objets ou « features » (ex : un panneau de menu, une sphère) qui sont les unités de base d'un jeu. Nous avions une personne responsable de la 2D, et une autre qui intervenait également de temps en temps sur ces contenus. La première produisait également tout le contenu en 2D nécessaire à la responsable marketing pour alimenter les réseaux sociaux et les articles. Outre la création graphique de l'objet, l'artiste lui applique aussi une texture et des couleurs, une ombre et une épaisseur, ou encore des effets spéciaux. Tout le métier de la 2D consiste à créer un objet réaliste et cohérent avec l'univers défini par le concept art. L'aspect immersif de la réalité virtuelle (VR) exige en effet un haut niveau de qualité. L'artiste veillera également à ce que l'objet apparaisse dans le bon plan. Par exemple, certains panneaux de Star Shaman présentaient des défauts d'affichage au début du projet. Certains composants s'affichaient en arrière-plan de ces panneaux et n'étaient donc pas dans le plan visible.

D'autres artistes du studio s'occupaient de la 3D. Il s'agit des objets tels que la sphère du shaman dans laquelle il plonge une main pour générer des sorts ; du tunnel interstellaire par lequel le shaman navigue de planète en planète, ou encore du paysage des planètes. Les mêmes contraintes que pour la 2D s'appliquent, mais avec une complexité plus grande car certains objets sont observés sous plusieurs angles. Par exemple, le joueur régénère des lunes qui se mettent en mouvement. Il faut donc travailler ce mouvement ainsi que les effets spéciaux, le cas échéant, pour les avoir sous tous les angles, avec une qualité optimale. De plus, ces caractéristiques ne doivent pas briser le jeu ou handicaper une autre "feature". Ainsi, nous avons eu des bugs liés au comportement d'un objet qui empêchait d'autres « features » de réagir correctement. En outre, plus un objet est travaillé avec une texture particulièrement soignée, plus il peut être lourd. Cela peut ralentir la performance du jeu et en VR susciter des écrans gelés ou des flous qui sont extrêmement gênants pour le joueur. Une mauvaise performance peut faire quitter un jeu définitivement et/ou donner la nausée. L'artiste doit aussi parfois optimiser ses objets pour enlever le maximum de détails sans que cela impacte la qualité de ce qui est vu par le joueur. Le terme « optimisation » est donc un mot que j'ai entendu souvent ! Un jeu de VR est aussi soumis à un seuil minimum de « rafraîchissement » de l'image : la boutique d'Oculus en fait une condition pour que le jeu figure dans son catalogue.

Enfin, parmi les artistes, nous avions la personne responsable des « skybox ». Le joueur se trouvant sur une planète à chaque niveau, il fallait une personne dédiée à la création des environnements au-dessus de l'horizon.

La seconde partie de l'équipe du jeu, hors profils transversaux, était dédiée au développement. Elle comprenait trois développeurs mais aussi un intégrateur. Le développeur d'un jeu est chargé de faire dialoguer les éléments du jeu au moyen de scripts. Il construit donc la « scène » en s'assurant par exemple que la régénération d'une lune ajoute bien ses éléments au décompte du total, ou encore qu'un tir d'ennemi blessant le joueur lui enlève un point de vie. Le développeur met donc en place la mécanique du jeu. Chacun a sa spécialité (son, paramètres de physique de l'environnement, etc.…) car les jeux sont très complexes, surtout en VR.

L'intégrateur, comme son nom l'indique, intègre les objets réalisés par le pôle artistique dans le moteur du jeu. Il en définit les paramètres, règle l'animation et "débugue" quand cela est nécessaire. Durant le projet, les métiers se sont chevauchés et il n'y a pas eu de séparation nette entre les compétences, chacun développant plutôt une partie du jeu.

À la suite de la sortie de Star Shaman, l'un de nos développeurs a évolué hors de l'équipe de jeu pour se concentrer sur l'évolution et la mise à jour des outils de travail du studio. Il agissait plutôt en tant que coordinateur technique pour s'assurer que chaque membre avait les outils nécessaires à son travail.

Pour terminer cette présentation des métiers, j'aimerais développer rapidement les autres métiers du studio. Nous avions bien sûr un chef de projet et game designer chargé de créer le jeu et sa mécanique ; il partit quelques mois avant le lancement de Star Shaman. Nous avons également eu un consultant responsable de la documentation du jeu afin que nous puissions suivre les évolutions suite aux décisions prises ; il participa à l'équilibrage. Cette dernière tâche est fort importante car elle consiste à s'assurer que le jeu n'est ni trop simple ni trop compliqué, et que le niveau de difficulté est celui souhaité tout au long du jeu (stable ou évoluant dans de bonnes proportions). Nous avons aussi travaillé avec des entreprises consultantes pour des aspects précis du jeu : le son et certains effets spéciaux.

Par ailleurs, nous avions trois activités transversales : la production, le marketing et l'analyse de données. La première était assurée par nos deux dirigeants et consistait à gérer le personnel, le budget, les dossiers de subventions, les relations avec Oculus pour obtenir une page sur leur boutique et l'avancement global du jeu. Le marketing était assuré par ma collègue Sammie. Elle a développé les comptes d'Ikimasho sur les réseaux sociaux, suivi l'élaboration du site Web de Star Shaman et rencontré nos partenaires tels Oculus ou encore le réseau social VRChat pour l'organisation d'un événement en ligne. Enfin, Sammie et moi avons beaucoup collaboré dans le suivi des audiences des canaux de marketing, qu'il s'agisse des réseaux sociaux, des sites Web ou des boutiques en ligne.

2. Les missions d'une data analyst

a) Comprendre son environnement

Je suis arrivée à Ikimasho le 27 janvier 2020, au moment où le studio pensait sortir Star Shaman en avril. J'ai commencé par faire des recherches sur les données dans le jeu vidéo afin de comprendre ce domaine, nouveau pour moi. Je me suis renseignée sur les métiers du jeu vidéo en passant parmi mes collègues pour comprendre ce qu'ils faisaient et ce qu'ils attendaient d'une analyste des données. Ce

métier était nouveau, les attentes ont donc été limitées à deux questions : Le jeu fonctionnerait-il ? Et quel serait le comportement de nos joueurs ?

Durant cette prise de marques, j'ai fait différentes choses telles que me former à Google Analytics pour ensuite installer les balises nécessaires sur nos sites Web et notre page de la boutique Steam. J'ai également commencé à rédiger une feuille de route pour le poste Data dans le studio en lisant des publications sur la réalité virtuelle et les actualités des données, notamment à propos du scandale du studio Nantic. Développeur du jeu mobile à succès Pokemon Go, celui-ci collectait de manière abusive les données des joueurs : l'application, même éteinte, collectait les données et celles-ci géolocalisaient la personne, allant jusqu'à permettre l'identification du domicile et du lieu de travail du joueur.

Parallèlement, je faisais des petits travaux de recherche d'inspiration pour de futures jeux ou des tâches plus directement liées au jeu. J'ai notamment recherché nos obligations envers nos financeurs pour le générique, ou encore rédigé les conditions générales de ventes. Ces dernières ont été particulièrement intéressantes puisque j'ai ainsi pu voir comment se finance un jeu vidéo et quels sont les principaux acteurs en matière de prêts et subventions.

Après mon arrivée, la sortie de Star Shaman a très vite été repoussée à juin. Et malgré ce délai supplémentaire, je manquais de temps pour développer des données du jeu. J'ai fait certes une première analyse des données à suivre, mais je me suis concentrée sur ce qui pouvait être développé dans le temps imparti. J'ai donc commencé par développer l'axe marketing de mes activités.

b) ANALYSE DES CANAUX MARKETING

Les besoins de ma collègue responsable du marketing étaient relativement ouverts : je pouvais suivre toutes les données possibles et lui faire des rapports. Ces rapports ont suscité des réunions hebdomadaires avec les chefs à partir de juin. J'ai donc tout d'abord identifié toutes nos sources de données, les données disponibles pour chacune d'entre elles ainsi que leurs périodes de rétention. Il a fallu identifier les données usuellement suivies en marketing telles que les impressions - le nombre d'occurrences d'un post sur un écran, cela ne signifiant pas qu'il a été lu – et le taux de clics, ou encore le ratio des deux.

Nous avons changé les canaux au fur et à mesure de l'élaboration de la campagne marketing. Par exemple, en ce qui concerne les boutiques en ligne, nous avions prévu de développer Oculus, Steam et Epic. La dernière a été abandonnée, au profit de Viveport particulièrement actif sur le marché chinois. De même, pour les réseaux sociaux, Reddit a été inclus dans le panel et l'intérêt des uns ou des autres réévalué selon leur importance dans notre audience. Nos sources de données furent donc :

- Facebook
- Twitter
- Instagram
- Youtube
- Site web d'Ikimasho : www.ikimasho.games
- Site web de Star Shaman : http://starshaman.games
- Page du jeu Steam
- Page du jeu Oculus
- Boutique Viveport
- Reddit (à partir de novembre 2020)

Collecte des données

La collecte de données a été une étape cruciale car tous les canaux ne donnaient pas aisément des accès. Par exemple, il a été impossible d'avoir accès sur Viveport à d'autres choses que les revenus des ventes. La boutique ne propose pas de rapports sur le nombre de visiteurs de la page du jeu et le nombre d'acheteurs. Ce dernier nombre pourrait se déduire du montant des ventes mais la boutique propose deux manières de jouer : l'achat traditionnel et le paiement d'un abonnement Infinity par mois pour avoir accès à tous les jeux de la plate-forme. Ainsi, pour les abonnés, seule la part de Star Shaman dans le montant mensuel (en fonction du nombre de sessions par jeu accédé) était comptabilisée, rendant ainsi les calculs du nombre de joueurs périlleux. Un autre exemple de canal sans données est Reddit : on peut observer le nombre de votes en faveur de son post ou compter les réponses, mais il n'est pas possible d'extraire ces données de manière systématique. J'ai découvert qu'il existait une API qui permettait d'accéder au nombre de Karma d'un compte ou aux sub-reddit associés, mais cela ne donnait pas d'informations sur le succès d'un post non plus. Il s'agit plus d'une API permettant de faire des actions comme suivre un compte, liker.

Par ailleurs, nous avons aussi eu des difficultés à accéder aux données d'Instagram. Elles ne sont pas accessibles entièrement lorsque le compte a moins de 100 abonnés, ce qui fut notre cas jusqu'à mon départ. Les données restantes ne s'affichaient que sur mobile, médium privilégié par ce réseau social, pour les 7 ou 30 jours précédents. On pouvait ainsi obtenir le nombre de comptes touchés (équivalent des impressions avec détails par accès au profil et actions d'aller sur le site web du studio), le nombre d'interactions avec le contenu, le total des abonnés et leurs pseudonymes, les contenus publiés par le compte. Je notais donc ces informations à la main dans mes rapports. Il a aussi été possible de voir les statistiques des publicités faites avec les impressions et clics, les pays ou régions concernés, ainsi que l'âge des personnes. Enfin, pour chaque post sont disponibles les nombres de likes, de commentaires et de partages, ainsi que les impressions, interactions et les visites de profils à la suite de la lecture.

Pour les autres canaux, j'ai soit installé une balise Google Analytics (GA), soit téléchargé les données en csv, seul format commun à l'ensemble des canaux. Les sites web furent suivis en posant une balise GA dans l'en-tête des pages et en créant une propriété GA pour chacun. J'ai dialogué avec les prestataires chargés de la construction de ces sites. Le site du studio existait déjà mais je souhaitais changer la bannière des cookies. Elle ne permettait pas au visiteur de refuser les cookies et de ne pas envoyer d'informations à GA. Les nouvelles prescriptions de la CNIL exigeaient en effet de laisser ce choix au visiteur. Nous avons donc instauré cette option pour les deux sites Web. Celui de Star Shaman était en cours de création. Nous avons ajouté la balise JTAG de Google Analytics, permettant notamment d'installer plusieurs pipelines de monitoring à l'avenir, pour suivre des objectifs par exemple. Ces derniers permettent de voir combien de visiteurs font une certaine action ou un certain parcours. Nos objectifs se sont cependant limités à l'utilisation de liens UTM pour suivre le nombre d'utilisateurs allant du site du jeu à la boutique Steam. Il s'agit d'un hyperlien ajoutable sur une icône, une image ou dans un texte. Il envoie des informations précises sur le contexte du clic. Nous pouvions donc voir dans GA combien de visiteurs avaient cliqué sur l'icône Steam du site Star Shaman ou combien venaient sur le site depuis une de nos campagnes publicitaires. La boutique en ligne Steam permettait aussi d'ajouter une balise simple de Google Analytics dans les paramètres de la page du jeu afin d'envoyer les données de trafic.

La boutique Oculus est un cas à part car elle mélange les données de trafic et d'utilisation du jeu, Oculus étant aussi un fabricant de casques. Le site web de la boutique propose une section dédiée aux données. On peut y trouver les nombres de visiteurs de la page, d'acheteurs ou encore de « wishlisteurs » (ceux qui mettent le jeu sur leur liste de souhaits). Malheureusement, il n'est pas possible de déterminer les points d'entrée des visiteurs et donc de savoir si le site Star Shaman renvoie des visiteurs ou si telle publicité a fonctionné !

Au cours de ma mission, je me suis rendu compte de deux autres problèmes dont l'un peut sans doute être résolu. Le premier est que Steam envoie les informations de trafic à Google Analytics et propose des rapports de ventes et wishlisting. Mais il n'y a pas de possibilité de faire le lien entre les deux. On ne peut donc pas savoir combien d'acheteurs sont des visiteurs venus du site Star Shaman ou de telle publicité. En interrogeant l'équipe de support, je n'ai pas non plus pu obtenir l'URI de la page d'achat : il s'agit de la partie fixe d'une URL de la page d'achat d'un jeu Steam. Par exemple, l'URI de l'adresse https://store.steampowered.com/login/?purchasetype=self&checkout=1&redir=checkout%2F%3Fpurchasetype%3Dself%26snr%3D1_8_4__503&redir_ssl=1&snr=1_8_4__503, URL d'achat sur la boutique Steam, est : https://store.steampowered.com/?purchasetype=self. Même en achetant un jeu moi-même, je n'ai pu la récupérer. Je n'ai donc pas réussi à créer un objectif d'accès à l'URL qui nous aurait donné les utilisateurs atteignant cette page. Cela ne peut être résolu que du côté de Steam.

Le second point concerne les bannières de cookies qui permettent à l'utilisateur de refuser l'envoi de ses données. Il faudra développer un système pour compter les utilisateurs qui refusent. Actuellement, les données sont biaisées lorsque l'on souhaite connaître le nombre de visiteurs et l'impact de notre publicité. En effet, les visiteurs refusant n'envoient aucune données à Google Analytics, et donc ne sont pas comptabilisés dans les visiteurs.

Pour terminer sur la collecte des données, il faut également mentionner Facebook, Twitter et Youtube. Youtube a une option pour mettre une balise Google Analytics dans les paramètres d'un compte mais l'envoi de données n'a jamais fonctionné malgré mes vérifications. J'ai donc récupéré les données sur https://studio.youtube.com/ qui permet d'extraire dans le format .csv et de croiser certains paramètres.

Facebook a également la possibilité de télécharger les données en .csv, que ce soit par post ou par jour. L'interface spécifique https://analytics.facebook.com aurait été utile, notamment pour extraire les pays, mais elle ne montrait pas de données, et cela jusqu'à mon départ, soit par manque de visiteurs en début de vie de notre compte, soit en raison du RGPD. J'ai donc continué à utiliser les données accessibles gratuitement.

Quant à Twitter, les données sont accessibles sur une page spéciale avec la possibilité de télécharger jusqu'aux derniers 28 jours. Nous avons commencé notre campagne marketing le 16 juin 2020, date à laquelle la première bande-annonce de Star Shaman a été diffusée à l'occasion du festival UploadVR. La deuxième partie du plan marketing était programmée à partir de la sortie, le 22 octobre. Il a donc fallu accumuler les données dans un fichier entre le 16 juin et le 22 octobre afin d'avoir un aperçu sur toute la période, et non sur 28 jours seulement. Ce fut l'une des parties du script R, chargé de traiter les données.

Types de données et traitements

Le script en langage R servait à nettoyer, calculer et fusionner les données des divers canaux. Une version plus optimisée, notamment avec un script spécifique aux fonctions appelées en Python, est restée inachevée car la priorité est allée aux données de jeu à partir du mois de juin.

Les données étaient importées des fichiers csv pour y être nettoyées : passage des dates en format date, remplacement des valeurs nulles[15] par 0 ou encore suppression des données cumulées dans le même fichier sur 7 ou 28 jours. Il fallait aussi extraire les données de la période souhaitée, à savoir à partir du 16 juin 2020. Le script permettait aussi de créer des fichiers csv avec les données des canaux agrégées et thématisées, utilisables ensuite dans le logiciel Tableau pour faire des visualisations. Je calculais les

[15] Il existe de nombreuses techniques pour remplacer une valeur manquante mais dans notre cas, cela signifie qu'il n'y avait pas d'actions ou de post tel jour. Il n'a donc pas été nécessaire de chercher une autre valeur de remplacement.

impressions et l'engagement, la part des impressions par canal, les pays de provenance, les tranches d'âge, les likes et les partages, dans le temps.

Le principal problème était que certaines données ne sont pas disponibles dans certains canaux ou peuvent se calculer à partir d'autres données, uniquement. Par exemple, l'âge des visiteurs n'était connu que par les données Facebook ou Youtube. On ne pouvait donc pas faire de comparaison mais simplement donner les âges comme un fait. Autre exemple, l'engagement est une notion différente d'un canal à l'autre. Twitter le définit comme le « nombre total de fois qu'un utilisateur a interagi avec un Tweet. Ce chiffre tient compte des clics effectués n'importe où sur le Tweet, notamment des Retweets, des réponses, des abonnements, des "J'aime" et des clics sur les liens, la carte, les hashtags, le contenu multimédia intégré, le nom d'utilisateur, la photo de profil et "pour ouvrir le Tweet" »[16] dans sa documentation. A l'inverse, Google Analytics en a une approche plus précise : l'engagement comprend aussi le lent défilement d'une page lorsqu'elle est lue et il n'est pas calculé automatiquement. Quant à Facebook, il donne le rapport entre impressions et interactions (clics, likes, etc.…).

Il a donc fallu adapter les indicateurs pour pouvoir faire une comparaison sans biais de définition. J'ai défini l'engagement comme toute action de la part de l'internaute, quelle qu'elle soit. Je l'ai calculé en nombre de personnes lorsque cet indicateur n'était pas disponible par défaut. Google Analytics propose un taux de rebond, c'est-à-dire le taux de personnes qui quittent la page sans avoir interagi avec elle, et ce, comparé au nombre de sessions. J'ai calculé l'engagement comme l'inverse du rebond et converti le taux en nombre de personnes. Nous avons ainsi pu comparer les taux totaux d'engagement sur une période et pour chaque canal, afin de voir lequel était le plus efficace sur le moment. Les statistiques de likes, de partages et d'abonnés venaient détailler un peu l'engagement et compléter l'analyse. Cela nous a permis de voir des tendances étonnantes comme « l'effet Twitter » : un post ayant un fort engagement aura peu de likes ou de partages, et inversement.

Une fois, les fichiers csv générés, ils furent importés via le logiciel Tableau pour faire une suite de graphiques à copier dans le rapport hebdomadaire. Je générais ainsi une carte mondiale des impressions et de l'engagement, un graphique en courbe de l'évolution du nombre d'impressions par canal, ou encore des tableaux de synthèse Facebook/Twitter contenant le nombre de posts par jour, l'engagement, les likes et les partages.

Dans le rapport figuraient aussi toutes les statistiques manuellement extraites d'Oculus et Instagram. J'y ajoutais le bilan sur les visites générées, des sources de trafic et des campagnes de publicité avec les données de Google Analytics. Enfin, une analyse des mots-clés de recherche et des intérêts de nos audiences a aussi eu lieu.

Par ailleurs, j'avais pour projet, jusqu'en juillet, de créer une page à accès réservé sur le site du studio et d'y afficher les graphiques de la semaine. Toutefois, les données auraient été statiques (pas d'API pour chaque canal permettant le rafraîchissement). J'ai créé un premier graphique dans l'architecture WordPress du site web, au moyen de la libraire JavaScript appelée Chart.js. Toutefois, pour des raisons inconnues, je n'ai pas réussi à afficher les données sur le site et la nécessité de développer le pipeline des données de jeu m'a fait mettre ce projet de côté. Etant donné le caractère statique, nous avons convenu que la priorité était ailleurs. La sortie du jeu ayant été repoussée à l'été, nous avions en effet du temps pour développer un pipeline dans le jeu.

[16] https://help.twitter.com/fr/managing-your-account/using-the-tweet-activity-dashboard

c) Analyse des donnees du jeu Star Shaman

Mon autre grande mission fut de développer l'analyse des données de jeu. Il s'agsait de créer un pipeline, à savoir un ensemble d'étapes, qui conduit à la réception de données de sessions des joueurs. Comme nous le verrons plus bas, l'analyse de jeu permet beaucoup de vérifications et compréhensions de phénomènes (voir partie II.2). Les étapes ont été de décrire les questions auxquelles apporter des réponses, trouver les données nécessaires, créer l'architecture nécessaire dans le projet Unity du jeu, et enfin trouver une plate-forme réceptionnant les données.

Ces étapes seront décrites en 3e partie. J'aimerais ici mentionner l'aide précieuse apportée par Gauthier Dine dans ce travail et le choix de la plate-forme. Il est lui-même data analyst et data architect chez GamePoint (un studio de jeux accessibles sur mobile, web et réseau social). Nous avions pour projet de créer notre infrastructure de stockage et traitement des données. Toutefois, le temps étant compté et les compétences pour ce faire n'existant pas dans l'équipe du studio, nous avons cherché une plate-forme commerciale pour réceptionner, nettoyer et stocker les données. Nous avons pris Delta DNA, qui permet tout cela, pré-agrège certaines données et donne accès à de nombreuses fonctionnalités d'analyse.

Après la sortie du jeu, je fus chargée de monitorer nos joueurs et faisais un rapport hebdomadaire. Les premières analyses furent simples : je vérifiais l'évolution du nombre d'installations et de joueurs, le nombre de jours joués avant d'arrêter le jeu, les pays concernés, etc... Je pouvais aussi faire des analyses plus poussées en croisant les joueurs n'ayant que peu progressé dans le jeu, avec d'autres critères pour comprendre leurs difficultés. Ces requêtes SQL me permettaient de comprendre quelles étapes avaient été faites par les joueurs ayant joué 1,2 ou 3 jours seulement. Dans le même ordre d'idée, j'ai développé des *funnels* (voir partie II sur l'utilité et III.1 sur le choix des données) qui sont des entonnoirs permettant de comprendre à quelle(s) étape(s) les joueurs abandonnent le jeu. Certaines des données suivies servaient uniquement à ce besoin. Nous avons ainsi pu identifier un problème avec le tutoriel en voyant des joueurs s'arrêter avant la fin de celui-ci.

Par la suite, j'ai étoffé un peu les analyses, notamment en calculant la fréquence à sept jours : le nombre de jours joués dans les sept précédents une date en particulier, et ce, sur une période. D'autres graphiques ont suivi en croisant les joueurs ayant joué 1, 2, 3 (etc.…) jours et les étapes par lesquelles ils étaient passés. J'ai également calculé des résumés statistiques des vies, scores, montants de monnaie entre autres, pour chaque groupe de joueurs. Ces graphiques furent la base de tableaux de bord dynamiques créés dans l'interface de la plate-forme Delta DNA. Il suffisait de les mettre à jour chaque semaine pour voir les tendances et aller ensuite fouiller dans les données pour chercher des raisons à ces découvertes.

On le voit, les missions autour de la data au studio étaient très variées. L'analyse des données de jeu est présentée de manière approfondie en partie III, notamment dans les étapes de la gestion de ce projet et les choix réalisés tout au long du développement. J'aimerais à présenter décrire le jeu Star Shaman, dont une partie des joueurs sont la base de nos analyses.

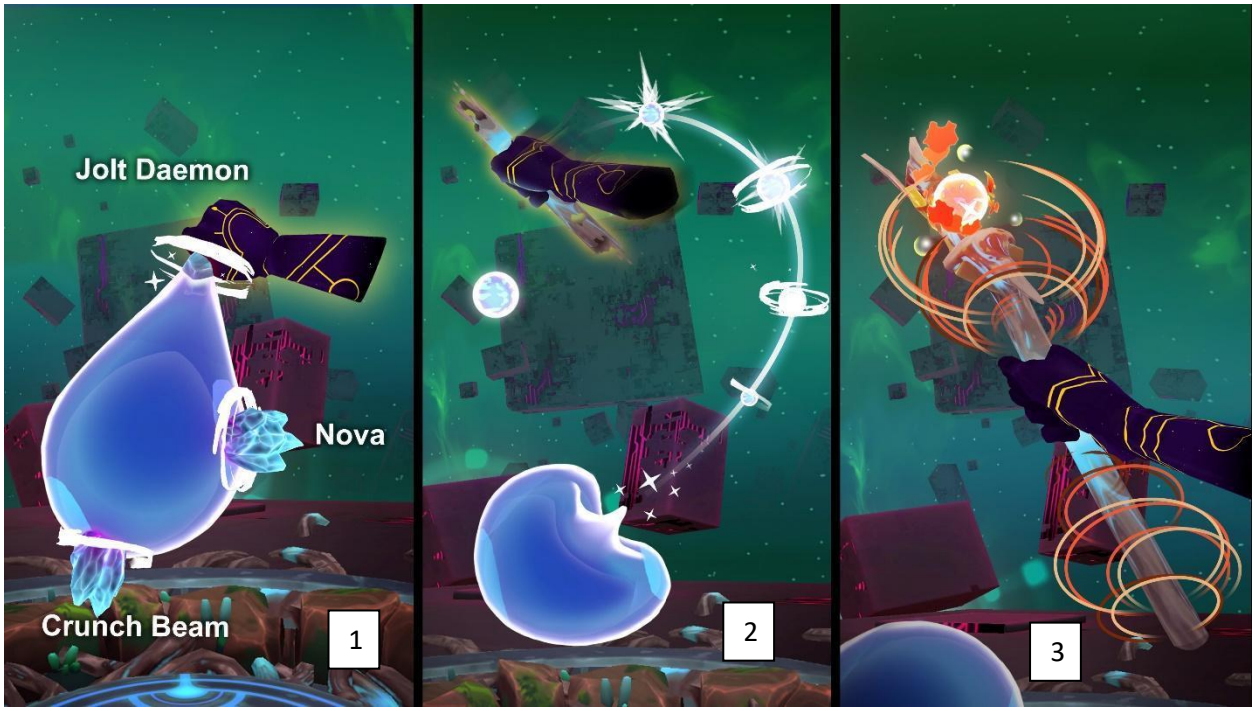
3. PRESENTATION DE STAR SHAMAN ET PREMIERE APPROCHE DE LA PROBLEMATIQUE DES COMPORTEMENTS DE JOUEURS

Connaître les actions des joueurs est un moyen de savoir qui ils sont et quels éléments sont impactant dans le jeu. Nous verrons en partie II que c'est un moyen de vérifier si les joueurs sont ceux que l'on a ciblés, et de savoir quels éléments sont appréciés ou non des joueurs afin d'éventuellement en développer de nouveaux. L'analyse de données sert à garder les joueurs dans le jeu en connaissant leurs motivations et leurs manières d'évoluer, ainsi que les points d'achoppement. Dans cette section, je présenterai d'abord le jeu Star Shaman et ses fonctionnalités ; leur compréhension me paraît essentielle pour comprendre le développement du pipeline de données et les analyses en partie III. Une discussion sur le point de départ de l'analyse du comportement des joueurs suivra.

Le joueur incarne un shaman des étoiles, faction résistante au pouvoir des Architectes de l'Entropie. Ces derniers vont de planète en planète pour éradiquer la vie et transformer les planètes et leurs lunes en perfections géométriques. La vie est un type d'infection selon eux, et doit être détruite au profit de leur canon esthétique. Les shamans s'opposent à cette vision de la perfection et lui préfèrent la diversité de la vie. Le joueur incarne donc un résistant qui se bat pour la biodiversité, dans la droite ligne des préoccupations environnementales actuelles.

Le shaman passe par trois systèmes solaires qu'il doit régénérer en permettant le retour à la vie, grâce à des sortilèges. La vie est constituée d'éléments que les architectes ont enfermés dans les lunes de chaque planète. Il existe cinq catégories d'éléments : azote, phosphore, soufre, eau et carbone. Le shaman doit rassembler un nombre X de chaque élément pour régénérer tout le système solaire. Une fois les X éléments rassemblés, il se rend sur la planète-berceau, le cœur du système, pour y verser les éléments dans cinq piliers diffusant la vie dans tout le système. A ce moment, un ennemi tente de l'en empêcher. Il s'agit du « Boss » final qui est un ressort traditionnel des fins de niveaux dans un jeu vidéo : c'est un ennemi particulièrement puissant que l'on affronte en fin de niveau. S'il est battu, le joueur est victorieux et passe au système solaire suivant, jusqu'à gagner le jeu lors du dernier système. Si le joueur meurt face au Boss, il peut recommencer le jeu. Star Shaman est conçu comme un roguelite, c'est-à-dire un jeu très dur à maîtriser et dans lequel le joueur meurt très souvent jusqu'à en apprendre suffisamment les rouages pour survivre. Ainsi, jusqu'à la mise à jour du 29 janvier 2021, le joueur après sa mort recommençait le jeu depuis le tout début, en perdant la plupart des avantages acquis. Depuis, il existe un mode plus simple dans lequel le joueur reprend au début du système solaire dans lequel il a trouvé la mort.

En ce qui concerne la manière de jouer, le joueur va de planète en planète. La plupart sont des planètes à obeloïds, sortes de machines créées par les Architectes et qui éliminent la vie. Le joueur possède une baguette dans sa main dominante ainsi qu'une sphère de laquelle il tire des sorts et des protections. Le joueur y plonge sa main, attrape un cristal qu'il tire en-dehors jusqu'à obtenir un effet visuel et sonore d'éclatement. Des bulles apparaissent alors qu'il doit faire éclater avec la main dominante pour charger le sort dans la baguette. Il peut ensuite le lancer contre les obeloïds se trouvant sur la planète. En plongeant sa main non-dominante, il peut de la même manière attraper un bouclier et se défendre contre les tirs/lasers des ennemis.



Le joueur charge un sort dans sa baguette :

1.Attraper un cristal et le tirer hors de la sphère

2.La sphère éclate et libère des bulles qu' il faut frapper

3.Le sort est chargé dans la baguette

Le joueur régénère successivement chaque lune de la planète, en affrontant à chaque fois une vague d'obeloïds. Il prend les éléments de vie contenus dans chaque lune. La planète elle-même se régénère automatiquement lorsque toutes ses lunes sont soignées. Chaque planète constitue un niveau. Et la difficulté, qui se traduit par le nombre de lunes ainsi que le nombre et la vitesse des ennemis par lune, s'accroît avec le nombre de niveaux. Elle augmente d'une unité lorsque le joueur a régénéré quatre planètes dans le système solaire 1, puis s'ensuivent d'autres paliers propres aux systèmes solaires. À chaque fin de niveau, un autre paramètre est calculé : il s'agit du score qui dépend du nombre d'ennemis abattus et du nombre de coups reçus. Enfin, la « currency », ou monnaie de jeu, est calculée en prenant un pourcentage du score.

Le joueur presse la détente de sa manette pour lancer le sort sur l'ennemi.

Une fois l'ennemi détruit, le joueur frappe sa sphère et la lune se régénère en prenant une nouvelle forme.

Ici une forme sans animation.

Planète du deuxième système solaire : avant et après régénération.

La monnaie sert à acheter de nouveaux sorts et de nouvelles protections. Ceci se fait sur un autre type de planète, appelé « Shop » dans le jargon de l'équipe et « Reliquaire de Kibele » dans le jeu. Nous l'appellerons magasin. Il existe un magasin par système solaire, où le joueur peut dépenser sa monnaie pour augmenter ses capacités. Il faut d'ailleurs noter que lorsque le joueur meurt, une fraction de son score détermine la monnaie qui lui reste d'une partie à l'autre. Les sorts et protections achetés disparaissent cependant.

Enfin, le système solaire comprend aussi des planètes-événements. Tout comme les magasins, elles ne contiennent pas d'obeloïds. Elles peuvent donner des points de vie et des éléments au joueur. Le système solaire 1 en contient deux et les suivants entre deux et quatre. Au fur et à mesure, le joueur rencontre des personnages de l'univers du jeu.

Parmi les avantages de Star Shaman, il faut noter que la structure de l'univers est générée aléatoirement à chaque nouvelle partie. C'est ce que l'on appelle la génération random, anglicisme venu de « random generation ». Cela signifie que des éléments du jeu sont générés aléatoirement. Dans notre cas, il s'agit de contenu mais aussi de nombre d'éléments.

Dans le cas du contenu, les lunes des planètes sont tirées aléatoirement parmi les lunes disponibles et répondant aux critères du niveau en particulier. Ainsi, le joueur ne se retrouvera jamais avec la ou les mêmes lunes lors de sa première partie, pas plus qu'on ne peut déterminer quels types de lunes seront tirés à chaque niveau. Ceci permet d'introduire tout d'abord un hasard stimulant et de l'émerveillement aussi ; à chaque niveau, la surprise est de savoir quelles lunes apparaîtront. Chaque lune a un graphisme « mort » et un autre (doublé parfois d'une animation) lorsque le joueur la régénère. Le joueur peut ainsi espérer voir l'ensemble des lunes existantes (une bonne trentaine) et cela crée un effet de collection. Certaines étant très rares, cela peut constituer une motivation à jouer. En outre, la génération aléatoire du type de lune réduit l'ennui du joueur qui meurt souvent et doit alors refaire le parcours. Cette variété stimule son envie de continuer à jouer.

En ce qui concerne le nombre d'éléments, il existe par exemple un nombre de lunes minimum et maximum par planète en fonction du système solaire joué. Le nombre de lunes est déterminé au hasard entre ces deux bornes. Il en va de même pour le nombre de planètes-événements, de 2 à 4 dans les systèmes 2 et 3. Enfin, le type d'ennemis est également choisi de la sorte. Dans les paramètres du jeu, les planètes à obeloïds sont divisées en plusieurs groupes et chacun se voit associer un groupe d'ennemis. À l'intérieur du groupe d'ennemis se trouvent des listes d'ennemis avec un coefficient de probabilité d'apparition. À chaque niveau, les ennemis sont donc générés en fonction de la planète et à partir de cette liste d'ennemis, en fonction des coefficients. Il en existe de plusieurs sortes avec une animation particulière et une manière d'attaquer différente : tirs, lasers, feu, miroirs, etc...

C'est un document maître qui régit tous les paramètres de la génération aléatoire et contient toutes les clés (sous forme de caractères et chiffres) utilisées par le jeu pour accéder aux paramètres et éléments. Voici le sommaire de ce document qui liste l'ensemble des onglets et des niveaux dans le jeu, régi par la génération aléatoire :

	A
1	Name
2	Data_Master_Infos
3	Galaxy_Manager
4	Planet_Type_List
5	Solar_System_1
6	Solar_System_2
7	Solar_System_3
8	Level_Recipes
9	Moon_Rarity
10	DifficultyManager
11	Level_Settings
12	Spawner_Group
13	Spawner_List_Gobelin_Life
14	Spawner_List_Gobelin_Currency
15	Spawner_List_Gobelin_Mana
16	Spawn_List_Asia_Batch1_Diff1
17	Spawn_List_Asia_Batch1_Diff2
18	Spawn_List_Asia_BatchElite_Diff1
19	Spawn_List_Iceland_Batch1_Diff3
20	Spawn_List_Iceland_Batch1_Diff4
21	Spawn_List_Iceland_BatchElite_Diff3
22	Spawn_List_America_Batch1_Diff5
23	Spawn_List_America_Batch1_Diff6
24	Spawn_List_America_BatchElite_Diff5
25	Spawn_List_Batch2_Diff1
26	Spawn_List_Boss_1
27	Spawn_List_Boss_2
28	Spawn_List_Boss_3
29	Spawn_List_Boss_4
30	Spawn_List_Boss_5
31	Spawn_List_Level1_Wave1_Tuto
32	Spawn_List_Level1_Wave2_Tuto

- L'onglet *Galaxy_Manager* assure le nombre de systèmes solaires présents dans jeu
- *Planet_Type_List* contient les types de planètes et les clés de localisation utilisées par le jeu pour les reconnaître et les générer.
- Trois onglets avec les règles concernant le nombre par type de planètes pour chacun.

Etc...

Pour terminer cette présentation de Star Shaman, il me faut encore parler de quelques subtilités. Le magasin dans lequel les joueurs achètent des sorts se compose de différents niveaux que l'on débloque en fonction du score total : les paliers sont 5000, 10.000, 15.000, 20.000. À chaque niveau, de nouveaux sorts sont débloqués et les existants bénéficient d'une version améliorée (plus de dégâts ou de protection). Ces sorts plus puissants permettent au joueur d'avancer dans les systèmes plus difficiles et d'affronter des boss plus forts.

Par ailleurs, comme on l'a vu, Star Shaman est un roguelite dans lequel le joueur meurt très souvent. Toutefois, les joueurs avancés peuvent choisir de réinitialiser le jeu, c'est-à-dire de supprimer leur partie et de recommencer à zéro. Cela sert notamment lorsque l'on a perdu beaucoup de points de vie sur une planète et que le score du niveau est mauvais. Le joueur peut s'estimer loin de débloquer un palier de magasin et donc loin d'avoir les sorts nécessaires pour la suite. Il peut donc décider de recommencer à zéro afin de repartir sur de nouvelles bases. Dans ce cas, contrairement à la situation de mort, il ne conserve rien, pas même un montant de monnaie minimal.

Pour résumer, le travail d'analyse des données doit à la fois vérifier la cohérence de l'expérience du joueur, l'équilibrage de la difficulté ou encore l'intérêt des éléments spéciaux (la génération random

ou les planètes événements sont-ils appréciés ?). Nous n'avons pas pu monitorer toutes les données nécessaires à ces objectifs, car comment mesurer la satisfaction du joueur ? Nous y reviendrons en partie III.1 notamment.

Nous avons cependant développé un solide pipeline pour vérifier la cohérence et les points d'achoppement. Suite aux premières analyses simples des joueurs de Star Shaman, nous avons rapidement repéré des comportements qui nécessitaient une explication. Beaucoup de joueurs de la première version partaient au bout d'une journée, sans que l'on sache s'ils allaient revenir dans le jeu après quelques semaines ou non. En effet, je ne disposais d'aucun point de comparaison avec d'autres jeux car les données ne sont pas publiques. Je ne savais donc pas si un joueur parti depuis une semaine était perdu ou non. De fait, beaucoup de joueurs sont revenus au bout de 3 semaines, voire 4. Par ailleurs, certaines données comme le score renvoyaient des valeurs aberrantes pour quelques joueurs : il fallait donc savoir si cela était une erreur du pipeline de données, ou un bug. Je reviendrai en partie III sur les problèmes liés au pipeline et qui ont demandé des rectifications du pipeline de données.

Avec la toute première version de Star Shaman, nous avons eu également des retours sur les réseaux sociaux nous poussant à changer quelques paramètres du jeu et cela fut encore une autre raison de monitorer les joueurs plus finement. Nous voulions voir si les changements apportés modifiaient la situation. Enfin, la difficulté rencontrée par les joueurs (visible à travers le temps des sessions ou le nombre de celles-ci), nous faisait penser que nous n'avions que peu de joueurs experts, notre cible initiale pourtant.

Tout cela a demandé que j'utilise des méthodes avancées pour analyser le comportement, à savoir l'évolution du joueur dans Star Shaman. Je les développerai dans la partie suivante, en lien avec les travaux déjà effectués dans le domaine du jeu ou de la réalité virtuelle. Ce sera aussi l'occasion de mettre en perspective nos besoins avec ceux rencontrés par d'autres et développés dans la littérature professionnelle.

II. DEUXIEME PARTIE : CADRE THEORIQUE

Cette deuxième partie propose un état de l'art des données dans le jeu vidéo, et plus précisément du profilage des joueurs. Nous y aborderons successivement les concepts nécessaires à la compréhension de ce travail, la mise en pratique de l'analyse de données dans le jeu vidéo et enfin le profilage en lui-même.

L'analyse de données dans le domaine du jeu vidéo existe depuis les années 2000. J'ai notamment trouvé un article de 2004 concernant l'équilibrage du jeu grâce aux données des joueurs. On observe que les articles se multiplient à partir de 2010, décennie durant laquelle les méthodes de machine learning commencent à être appliquées. C'est véritablement à partir de 2017 que les articles sont très nombreux. Ces articles abordent les diverses utilités de l'analyse de données que nous détaillerons en deuxième partie de ce chapitre.

1. De quoi parle-t-on ? Glossaire sommaire

J'aimerais d'abord présenter plusieurs notions qui aideront à la compréhension du travail autour des profils des joueurs et du domaine de la réalité virtuelle. En effet, des mots sont apparus, spécifiques à l'un ou à l'autre domaine. Bien plus, les appellations ont changé au fur et à mesure des années. Certains termes ont évolué dans leur sens et peuvent porter à confusion. Un glossaire récapitulatif se trouve en partie V.

a) Breves notions autour du jeu

La réalité virtuelle tout d'abord a évolué dans son sens depuis Heaton et Lafrance qui en 1994 (Heaton & Lafrance, 1994) décrivait le jeu BattleTech comme de la réalité virtuelle. Il s'agissait d'un jeu d'arcade pour lequel le joueur était assis dans un environnement qui se referme. Un casque lui permettait de se croire dans un cockpit d'avion et des écouteurs lui permettaient de communiquer avec d'autres personnes installées à d'autres bornes. Toutefois, je rejoins les auteurs dans leur affirmation que la VR est dans la tête et dans l'interaction que la personne ressent. Aujourd'hui, je définirais la réalité virtuelle, d'après les lectures de la littérature professionnelle, comme un ensemble de technologies permettant à une personne d'être immergée dans un environnement virtuel et d'y évoluer grâce à l'interaction avec son corps. Sherman & Craig (2018) vont plus loin, en ajoutant la notion d'impact sur les sens :

> *« […] a medium composed of interactive computer simulations that sense the participant's position and actions and replace or augment the feedback to one or more senses, giving the feeling of being mentally immersed or present in the simulation (a virtual world). »* (Sherman & Craig, 2018)

Nous l'avons dit, beaucoup de contenus de réalité virtuelle sont aujourd'hui des jeux. Ceux-ci ont différents gameplay. Cette notion fait référence aux mécanismes différents des jeux. Pour résumer, *Beat Saber*, sorti en 2019, est défini comme un jeu de rythme (traduction personnelle de « rythm game »). Il s'agit de couper des cubes avec des sabres lasers selon un ensemble de règles. Le gameplay n'est donc à l'évidence pas du tout le même que pour *In Death : Unchained*, sorti en 2020, qui demande au joueur d'évoluer dans un château en abattant des ennemis au tir à l'arc. Ce dernier jeu demande de la stratégie, permet des évolutions alternatives du joueur dans l'univers, et propose des niveaux. Nous ré-utiliserons cette notion de gameplay, plus bas dans cette partie, car elle est au centre de la littérature professionnelle.

C'est également le cas des concepts de serious game et de gamification. Le premier, jeu sérieux en français, renvoie à un jeu dont l'objectif n'est pas récréatif. On peut y inclure tous les jeux éducatifs ou dits de team building (utilisés en entreprise pour analyser et faire évoluer l'interaction dans une équipe).

Ainsi tous les articles concernant la réalité virtuelle dans la santé dont nous parlerons plus loin font référence à des jeux sérieux. Il existe une littérature très abondante pour tout lecteur désireux d'en savoir plus. On trouve entre autres : Alvarez, 2019, Perez-Colado et al., 2018, Petsani et al., 2018, JAYACHANDRAN ET AL., 2017. Et concernant l'histoire du jeu : Oppermann & Slessareff, 2016.

Notre second concept, la gamification, est intimement lié au jeu sérieux. En effet, il s'est développé dans son sillage car il s'agit de l'usage des codes du jeu dans des situation non-ludiques. Alors que le jeu sérieux a apporté les mécanismes du jeu dans l'apprentissage ou la gestion d'équipe, la gamification va plus loin en utilisant les mécanismes sans le jeu. D'après Andrzej Marczewski, la gamification est « [the] use of games and game-like solutions in non-game contexts ». Il peut s'agir d'utiliser une méthodologie dans la résolution d'un problème en s'appuyant sur le mécanisme d'un jeu, ou simplement d'utiliser un élément de jeu comme des points ou récompenses pour gérer une équipe.

La gamification est intimement liée au développement de profils de joueurs. Beaucoup d'auteurs ayant développé des profils théoriques (voir II.3) évoluent dans le secteur de la gamification. J'aimerais ici terminer ces définitions générales par un point sur la différence entre les profils, le profiling, le clustering, le partitionnement et le modeling. Ces concepts sont très souvent mélangés mais ils ne recouvrent pas tout à fait la même réalité. Charles & Black (2004) parlent de modeling lorsque des types de joueurs ou de comportement sont créés à partir de données. Le profiling, ou profilage en français, reviendrait alors à classer les joueurs dans ces types. C'est Yannakakis (Yannakakis et al., 2013) qui fait explicitement la distinction :

> *« We also make a distinction between player modeling [10, 26] and player profiling.*
>
> *The former refers to modeling complex dynamic phenomena during gameplay interaction, whereas the latter refers to the categorization of players based on static information that does not alter during gameplay — that includes personality, cultural background, gender and age. » (Yannakakis et al., 2013)*

Selon lui et ses co-auteurs, le modeling consiste à observer des phénomènes durant le jeu. Et le profilage, à catégoriser les joueurs selon des données socio-démographiques et de personnalité qui n'ont rien à voir avec le jeu. Quant au clustering et à son équivalent français de partitionnement, il s'agit d'un terme technique pour désigner le fait de partager un échantillon en groupes ou clusters au moyen d'un algorithme d'apprentissage non supervisé. Nous détaillerons celui des k-means plus bas, lors de notre analyse. Les clusters peuvent être des profils ou d'autres types de groupes. Notre analyse portera en partie sur du partitionnement des joueurs mais nous aborderons aussi le profilage dans la revue de la littérature professionnelle car les deux sont liés. Partitionner dit trouver des groupes dont on pourra ensuite extraire les caractéristiques afin de créer des profils (profilage).

b) VOCABULAIRE DU PROJET STAR SHAMAN

Le projet Star Shaman a son vocabulaire propre, qu'il s'agisse des termes techniques de jeu vidéo ou des noms spécifiques au projet. Plusieurs termes entrent d'ailleurs en conflit avec mon propre vocabulaire des données. Je vais donc les aborder point par point. Tout d'abord, j'utiliserai lors de la présentation des analyses (voir partie III) les concepts de paramètres et variables. En effet, ceux-ci sont des mots communs en statistiques. Une variable désigne une caractéristique commune aux individus analysés : dans notre cas, le système solaire parcouru au moment de l'envoi des données de jeu est une caractéristique commune à nos joueurs. Un paramètre désigne un élément constitutif d'une loi statistique. Or, dans l'architecture du jeu, ces deux mots ont un tout autre sens. Variable désigne un élément du jeu qui contient une information telle que le numéro du système solaire. Cette variable peut être attachée à un

autre objet du jeu si celui-ci utilise le numéro de système solaire. De même, un paramètre est une caractéristique d'un objet comme la vitesse d'un projectile ou la taille d'un ennemi.

Nous utiliserons les termes de la manière suivante. En partie III.1.c, je parlerai du dictionnaire de données brutes à disposition pour l'analyse. Le terme de "paramètres" désignera les informations communes à chaque événement de jeu et qui sont envoyées avec lui pour créer nos données. Utiliser "variables" porterait en effet trop à confusion avec celle de l'architecture du projet. A partir de la section III.3, nous utiliserons uniquement le vocabulaire statistique. Ainsi les variables seront de nouveau à comprendre comme les caractéristiques communes aux joueurs.

Ceci me permet d'aborder également les notions de *game event* et événement data qui sont intrinsèquement liées lorsque l'on s'intéresse au pipeline d'envoi des données de jeu (voir partie III.2 pour sa description). En effet, nous verrons que les données sont envoyées sous forme de tableau avec en ligne des événements et en colonnes les paramètres. Par exemple, à chaque début de niveau, l'événement correspondant est envoyé avec plusieurs paramètres comme son nom, le système solaire où il se déroule, la langue utilisée par le joueur, etc... Or ma notion d'événement entre en conflit avec celle du moteur de jeu Unity dans lequel Star Shaman fut développé. Un *game event* désigne toute situation se produisant dans le jeu, tel le fait de réussir le geste pour charger un sort dans la baguette, ou de changer de système solaire. Il est possible d'en créer autant que nécessaire. Ils servent à générer un comportement en réponse à l'événement (ex : baisser les points de vies lorsque le joueur est touché). L'envoi des données est donc basé sur ces *game events* car ils déclenchent des fonctions de création, traitement et envoi des données. Par souci de clarté, j'emploierai donc le terme d' « événement data » pour désigne les événements envoyés sur notre base de données et qui constituent mes données brutes. En effet, un *game event* peut déclencher un ou plusieurs des événements data. A l'inverse, un événement data peut se trouver relié à plusieurs *game events*.

Pour terminer les définitions techniques, il me faut enfin parler de la concurrence entre *analytics*, *metrics* et data. Ce dernier mot renvoie dans le projet à tout autre chose que le pipeline de données. Il est pris sous son sens d'information. Lors de la construction du pipeline, les éléments de celui-ci ont plutôt été nommés avec le mot *Analytics* pour faire la distinction. Nous parlerons aussi beaucoup de *metrics* qui représentent nos données pour l'analyse, car il s'agit là de données calculées et agrégées à partir d'autres. Ce terme est toutefois employé diversement dans la littérature et Anders Drachen (et al., 2013) l'emploie, à l'inverse, pour les données brutes. Je garde le sens de la majorité des auteurs : les *metrics* sont des données non brutes.

Par ailleurs, Star Shaman a des éléments dont le vocabulaire est utilisé pour la suite de ce travail. En premier lieu, un boss est un ennemi que l'on retrouve en fin de niveau dans un jeu vidéo. Dans Star Shaman, le joueur affronte un boss (même terme en français) à la fin de chaque système solaire. C'est en effet l'ultime étape pour régénérer ce système solaire et gagner en fin de 3e système. Je parlerai donc souvent de boss 1, boss 2, boss 3 pour désigner les boss de chaque fin de système solaire.

Parmi les planètes que le joueur peut visiter, le shop est particulier. Il s'agit d'une planète-magasin sur laquelle on peut acheter ou mettre à jour des sorts et protections, avec de la monnaie virtuelle. J'emploierai aussi le terme de planète-magasin afin de faciliter la compréhension. A noter que le terme officiel dans le jeu est « Reliquaire de Kybele ».

Un autre type de planète est le *random event* que je traduis par planète-événement. Le joueur y rencontre des personnages et y fait des découvertes. Le nom original tient au fait que parmi l'ensemble des planètes événements, dans les systèmes 2 et 3, un certain nombre d'entre eux sont piochés aléatoirement.

c) QUELQUES ABREVIATIONS DES DONNEES ET DES TYPES DE JEU

Je termine ce glossaire général avec quelques types de jeux et de données couramment utilisés et leurs abréviations. En effet, celles-ci sont très courantes mais indéchiffrables pour le néophyte. Le lecteur pourra trouver une liste complète des types de données dans l'ouvrage d'Anders Drachen (et al., 2013) et dans celui de Coupart Thibault (Thibault, 2013). L'une des abbréviations les plus utilisées est DAU pour Daily Active Users. C'est le nombre de joueurs actifs, c'est-à-dire effectivement en train de jouer, par jour. On le décline aussi par semaine (WAU) ou au mois (MAU). Attention, certains auteurs utilisent le terme de Daily unique active users pour bien distinguer le fait de compter le nombre de connections et le nombre de personnes, ces dernières pouvant se connecter plusieurs fois dans une journée.

Le taux de conversion est le taux de joueurs ou de personnes accédant à une démonstration ou une publicité et qui deviennent ensuite des joueurs payants. On peut ensuite calculer l'ARPU (Average Revenue Per User) : l'argent gagné sur une période divisé par le nombre de joueurs au total. On peut décliner cette donnée sur les nouveaux joueurs également. La comparaison des deux sert beaucoup en marketing. Il en va de même pour l'UAC (User Acquisition Cost) : coût du jeu pour le studio divisé par le nombre de joueurs.

Par ailleurs, la notion de viralité (virality, k-factor) est très employée. On la connaît dans le domaine des réseaux sociaux. Il s'agit ici du taux de conversion de nouveaux joueurs qui vont acheter et jouer au jeu. Si l'on compare le ratio du nombre de joueur s'arrêtant de jouer sur le nombre de nouveaux, on obtient le taux de churn (churn rate, attrition rate).

On mesurera aussi le taux de rétention, à savoir le taux de joueurs revenant après leur première expérience. Cela s'accompagne du nombre moyen de sessions par joueur que l'on décline par jour, semaine et mois.

En ce qui concerne les types de jeu, il en existe de très nombreux. J'aimerais aborder les MMOG ou MMOs (Massive Multiplayer Online Games) qui se sont développés dans les années 1990. Ces jeux en ligne massivement multi-joueurs sont hébergés par des serveurs auxquels se connectent un très grand nombre de joueurs qui sont alors capables d'évoluer ensemble et d'interagir. Les univers de ces jeux sont persistants : le jeu continue de tourner et l'histoire de se dérouler, même lorsque le joueur n'est pas connecté. Il existe une variante appelée MMPORG (massively multiplayer online role-playing game) quand c'est un jeu de rôle.

En ce qui concerne la monétisation du jeu, nous l'avons dit en première partie, les free-to-play ont imposé un nouveau paradigme en proposant le début ou tout le jeu gratuitement. Ces jeux, notamment sur téléphone mobile, proposent des éléments ou des options payantes. On trouve tous types de jeux fonctionnant sur ce modèle mais pas encore en réalité virtuelle. Pokemon Go de la firme Niantic est un jeu mobile appelant le joueur à attraper des pokemons et proposant des options payantes. Le joueur se voit proposer la possibilité d'agrandir son sac à dos dans lequel il range les boules permettant d'attraper les pokemons.

Je n'entrerai pas d'avantage dans les détails car les données utilisées dans les analyses suivantes sont propres à Star Shaman et ne sont pas des données de marketing. Mais je tenais à définir quelques concepts que nous retrouverons dans la description de la littérature professionnelle. A présent, nous pouvons nous pencher sur le cœur du sujet : l'analyse de données dans le jeu vidéo et ses objectifs.

2. A QUOI SERT L'ANALYSE DE DONNEES DANS LE JEU ?

Les objectifs de l'utilisation des données dans le jeu sont multiples. Nous élargissons ici la recherche à l'ensemble de la sphère du jeu mais le monitoring des données s'est fortement développé dans le jeu vidéo en particulier. Cela est lié à des acteurs très divers que sont les studios, les chercheurs, le public. Les objectifs de l'analyse ont évolué depuis le début des années 2000, avec le développement de nouveaux genres de jeu et de nouvelles plates-formes. On peut les résumer ainsi :

- Équilibrer le jeu : « balancing » du jeu
- Vérifier qu'il n'y a pas de bugs
- Voir si les joueurs suivent le « flow » prévu
- Conserver ses joueurs
- Comprendre les interactions
- Surveiller un phénomène

Ces deux derniers objectifs sont notamment recherchés dans des domaines utilisant le jeu comme technique pour un objectif non-ludique. Comprendre les interactions, notamment entre personnes ou entre le joueur et le jeu, est une manière pour des chercheurs de comprendre l'humain sous l'angle psychologique ou encore anthropologique. Nous aborderons d'ailleurs, plus loin et rapidement, l'impact de la psychologie dans le domaine du jeu vidéo. Quant à la surveillance ou la compréhension d'un phénomène, beaucoup de travaux dans le domaine de la santé (et de très nombreux en réalité virtuelle notamment) utilisent les données pour ce faire. Enfin, les domaines de l'éducation et de l'apprentissage, qui utilisent beaucoup le serious game, sont également de grands fournisseurs d'articles sur la compréhension d'un phénomène, ici l'apprentissage par un sujet.

Plusieurs de ces objectifs de l'analyse de données se recoupent. Ainsi comprendre des interactions et surveiller un phénomène sont des processus quelquefois liés dans le domaine de la santé, notamment dans l'étude de la mémoire des personnes âgées. Par ailleurs, je distingue l'équilibrage, la vérification des bugs, le fait de vérifier que les joueurs suivent le flow, de l'objectif de conserver les joueurs. On pourrait argumenter qu'un jeu déséquilibré, des bugs et/ou des narrations transverses sont autant de facteurs de la perte des joueurs. Toutefois, il me semble que les jeux ont évolué et que depuis une dizaine d'années, il existe des méthodes pour conserver les joueurs, et qui ne sont pas relatives à la bonne santé du jeu mais plutôt aux joueurs eux-mêmes. En effet, ces méthodes permettent d'attirer, d'inciter à l'action et de fidéliser les joueurs. Beaucoup sont issues du marketing. Les trois premiers buts de l'analyse de données ont pour objectif de créer un bon jeu, alors que le quatrième sert à ancrer les joueurs.

Ils sont tous les quatre développés par Anders Drachen, Magy Seif El-Nasr et Alessandro Canossa dans leur ouvrage « Game Analytics: Maximizing the Value of Player Data » (Drachen et al., 2013) qui fait autorité en matière d'analyse des données dans le jeu vidéo. Le titre est tout à fait parlant. A leur suite, je pense que le développement des MMOGs (Jeux en ligne massivement multi-joueurs), puis des jeux sur réseaux sociaux et sur mobile, a induit une nouvelle forme de production des jeux qui se concentre sur les joueurs. Le jeu est devenu un outil modifiable pour fidéliser les joueurs. Les jeux ne sont plus figés dans leurs versions finales lors de leur sortie, mais sont constamment en évolution pour s'adapter à leurs joueurs :

> *« Regardless of the platform, users no longer expect " fire-and-forget" games; modern games on PC, console, and mobile devices must analyze user feedback and experience and use this information to adjust their offerings, thus making themselves appealing to their customers. »*
>
> (Drachen et al., 2013, p.54)

Les auteurs ne manquent pas de souligner les jeux free-to-play (voir définition en 2.1), dans lesquels il faut étudier les usagers avec soin pour les pousser à payer, car il s'agit là du modèle économique même de ce type de jeu. Nous ne sommes donc plus dans une économie où le joueur achète le jeu, une bonne fois pour toutes. Et il semble que cela date de 2009 environ :

« We can definitely attribute the starting point of this method with the rise of Zynga and its Farmville game, published on Facebook in 2009. » (Thibault, 2013)

Autre exemple de manière de conserver ses joueurs, en réalité virtuelle cette fois, Beat Saber est un jeu de rythme dans lequel le joueur fend des cubes au sabre laser sur des musiques électroniques. Dans ce cas, le jeu est payant mais propose de nouvelles musiques, notamment de groupes célèbres comme Linkin Park. Il s'agit ici de produire de nouveaux contenus pour conserver les joueurs.

Pour résumer, conserver ses joueurs est devenu l'objectif principal de l'analyse de données dans le jeu vidéo. Il subordonne tous les autres. De ce fait l'analyse de données a pris une place primordiale dans le développement d'un jeu vidéo et servira aussi bien le responsable marketing que le game designer :

« […] analyzing gameplay data can help us prove or disprove a hypothesis and rule out common misconceptions of a game designer, who may have a skewed view of the nature of his or her game. Oftentimes a visualization is so clear, that it dismisses any arguments. » (Drachen et al., 2013, p. 86-87)

Une littérature abondante s'offre aux chercheurs et professionnels de la donnée. Je détaillerai chaque objectif identifié en donnant des exemples d'articles ou ouvrages.

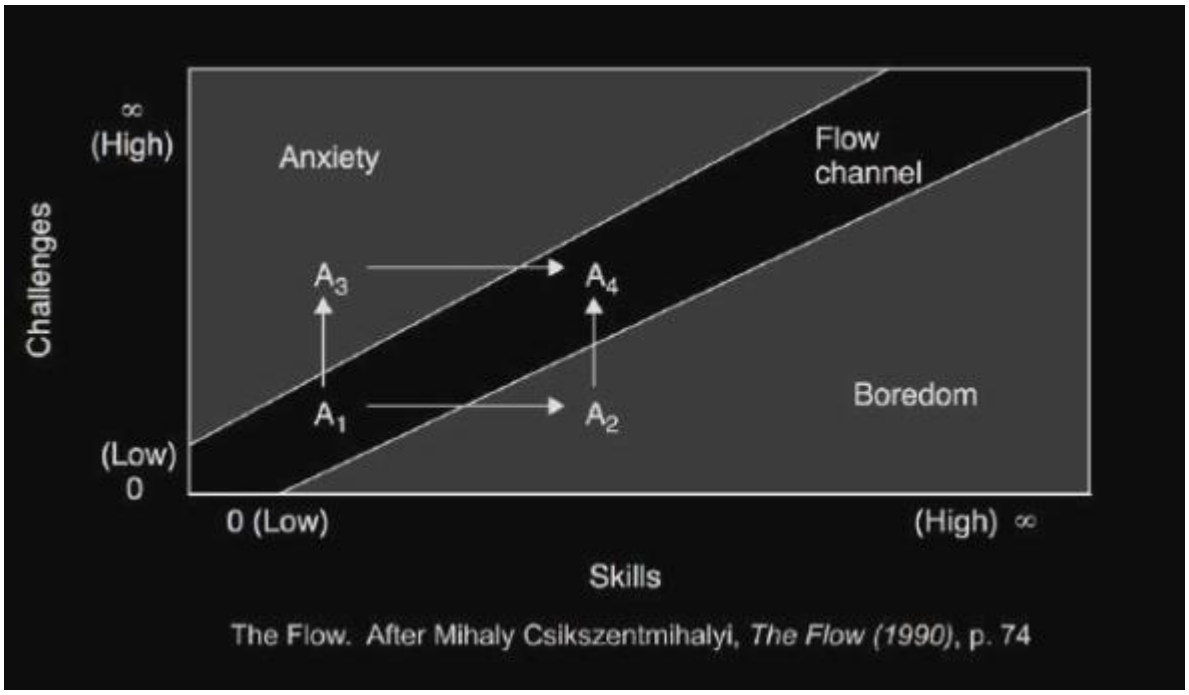
a) SIX OBJECTIFS EN UN

Le balancing

L'une des toutes premières utilisations des données, rencontrées dans la littérature professionnelle, cherchait à moduler la difficulté rencontrée par les joueurs et leur offrir la meilleure expérience possible. Il s'agissait d'équilibrer le jeu pour offrir l'expérience la plus personnalisée. Daryl Charles et Michaela Black (2004) mettaient en avant l'avantage de créer une courbe d'apprentissage personnalisée afin que chaque joueur puisse intégrer le jeu à son rythme et sans frustration. En outre, ils notaient que deux joueurs aux compétences identiques ne trouveraient pas les mêmes éléments du jeu intéressants ; il fallait donc prendre plusieurs facettes en compte dans l'équilibrage. Cette volonté d'équilibrer le jeu a évolué avec les années. Charles et Black plaidaient pour un système dynamique, à l'inverse de ce qui se faisait à l'époque. En effet, le joueur pouvait sélectionner un niveau de compétences en début de jeu mais restait considéré comme débutant, intermédiaire ou expert tout au long du jeu : or, l'apprentissage fait gagner en compétences et le jeu risquait de devenir trop simple. Par ailleurs, il était également possible de paramétrer les capacités de son avatar (force, agilité entre autres) afin de moduler son niveau. Les auteurs de l'article proposaient un système dynamique qui adapte la difficulté en direct dans le jeu, en réévaluant le profil du joueur tout au long de la partie. Cela servirait à adapter la force des ennemis créés par l'ordinateur ou à donner des bonus pour encourager les joueurs.

D'autres articles ont par la suite montré de nombreuses variantes d'équilibrage dynamique durant le jeu (Yannakakis, 2013 par exemple). Je retiens notamment l'article de William Rao Fernandes et Guillaume Levieux (Fernandes & Levieux, 2019) qui propose une méthode d'ajustement de la difficulté des niveaux d'un jeu en fonction des résultats d'une régression logistique. Celle-ci est personnalisée et se veut

adaptable à tout jeu. Selon les auteurs, tous les types de jeu partagent un point commun : le jeu est un ensemble de défis (« challenges ») que l'on réussit ou non. C'est donc la difficulté de ces défis qui doit être équilibrée dynamiquement afin que le joueur se situe dans le « flow ». Le flow est un état idéal qui se produit lorsque la difficulté égale les compétences du joueur telles qu'il les perçoit. Il s'agit d'une théorie inventée par Mihaly Csikszentmihalyi, psychologue enseignant aux Etats-Unis, et qui se schématise ainsi :



Cité dans *Actionable gamification. Beyond Points, Badges, and Leaderboards.* (Chou, 2014)

Si la difficulté est grande et que le joueur a peu de compétences, il devient anxieux et peut quitter le jeu. Si au contraire, ses compétences perçues dépassent la difficulté, il s'ennuiera. Le flow est donc l'aire se situant juste entre les deux et dans laquelle le joueur interagira de manière optimale avec le jeu.

L'objectif de Fernandes et Levieux (2019) est double. D'une part, il s'agit d'offrir une solution d'équilibrage utilisant le peu de données en début de partie et uniquement celles du joueurs (le système n'est pas préalablement entraîné sur un panel de joueurs) : le système est donc hautement personnalisé. Il doit pouvoir classer le joueur dans un type pour s'adapter à lui très vite, en prédisant la possibilité d'un échec :

> *« […] instantly target a specific failure probability while using as few data points as possible, i.e. while having only observed a few attemps of the player to win the challenge. »* (Fernandes & Levieux, 2019)

Leur recherche a abouti à un système basé sur une régression logistique qui fait moins de 20% d'erreurs en moins de 2 minutes.

D'autre part, leur système permet de paramétrer tout ratio de difficulté. Ils comparent notamment leur système à ceux qui augmentent la difficulté en cas de réussite et la baisse en cas d'échec : ces systèmes ont un ratio de 0.5. Les auteurs ont donc imaginé une solution permettant d'augmenter la difficulté en fonction d'un méta-paramètre : theta. Theta est fixé à 0 en début de partie et augmente de 0.05 à chaque victoire : la difficulté passe donc à 0.5, soit une difficulté moyenne, si le joueur gagne 10 fois (10*0.05) ; la difficulté baisse à nouveau de 0.05 en cas d'échec. Outre theta, le modèle contient également une

valeur maximum et minimum de difficulté atteignable par le modèle. Quand la probabilité de succès est atteinte pour un défi, la régression est utilisée et theta devient la probabilité d'échec du challenge. Cette méthode permet d'obtenir une courbe d'apprentissage totalement personnalisée et une probabilité d'échec rapidement prévisible pour que le jeu puisse y réagir.

Il existe de très nombreux autres articles sur le sujet du balancing qui tous tentent de répondre au problème de la personnalisation et de la finesse d'évolution.

Identifier les bugs

Identifier des bugs dans le jeu est un autre objectif de l'utilisation des données dans le jeu vidéo. Il existe pour cela plusieurs méthodes qui utilisent les données des joueurs. Une des plus simples consiste à créer un entonnoir pour voir à quelle étape le nombre de joueurs diminue. Un entonnoir ou plus communément *funnel* signifie que l'on suit une série d'actions qui se déroulent dans un ordre très précis, comme les actions dans le tutoriel du jeu et les grandes étapes. Si 30% des joueurs ne finissent pas le tutoriel, il est possible qu'un bug les en empêche. Pour peu que l'on ait alors suivi leur parcours étape par étape durant le tutoriel, on pourra déterminer laquelle est responsable de la perte de joueurs. Le tutoriel est une source de perte des joueurs citée à de nombreuses reprises dans la littérature (Voir aussi Thibault, 2013, chapitre 2).

L'utilisation d'un entonnoir est un emprunt au marketing qui analyse la conversion des utilisateurs à un objectif, en plusieurs étapes. L'ouvrage d'Anders Drachen, Magy Seif El-Nasr et Alessandro Canossa, par exemple, cite à de nombreuses reprises les entonnoirs comme outils d'analyse des cohortes de joueurs et de leur conversion sur le web aux jeux en lignes (Drachen et al., 2013).

> *« These are techniques adopted – and sometimes subsequently adapted – from other areas where Business intelligence is applied, notably web analytics. Examples include acquisition analysis, funnel analysis, A/B testing and cohort analysis » (Drachen et al., 2013)*

La recherche de bugs peut bien entendu servir dès la phase Beta du jeu, lors des tests fonctionnels (Drachen & Canossa, 2009). Ainsi l'entonnoir permet de voir rapidement les écueils majeurs.

Quelques auteurs mentionnent le suivi des types de morts (Drachen et al., 2014, Aung et al., 2019). Les bugs ne sont pas toujours comptabilisés comme une cause de la mort des joueurs car ils sont difficiles à percevoir et un bug peut tout à fait impacter l'envoi des données lui-même. Mais suivre les autres causes de morts des joueurs est une manière de vérifier si un jeu contient des bugs : certaines morts ne sont pas justifiées et on peut alors poser l'hypothèse qu'il s'agit d'un problème. Sifa et ses co-auteurs (2013) mentionnent l'exemple du joueur mourant durant les cinématiques, c'est-à-dire les passages de narrations en vidéo. On trouve ces dernières au début du jeu, pour introduire le scénario ou lors d'étapes (fin d'un niveau par exemple). Ces passages ne requièrent aucune action de la part du joueur. Il est passif et regarde l'histoire lui être contée. Mourir à ce moment est donc impossible et relève du bug. Je n'ai pas trouvé plus de détails sur la mise en œuvre du monitoring. Il est souvent fait par défaut et non mentionné dans les recherches.

De fait, l'identification de bugs n'est pas beaucoup abordée par les auteurs parlant de l'analyse de données. On trouvera des références aux test de pré-sortie du jeu, notamment chez Drachen et Canossa (2009). Ceux-ci conseillent en effet de monitorer les données de jeu des tests en phase alpha (premier

prototype) et Betâ (version complète avant sortie). Hullet et ses co-auteurs (2011) sont une exception : ils détaillent l'ensemble des tests possibles avant la sortie du jeu et l'analyse de leurs données.

Enfin, je précise d'expérience qu'il est aussi possible de vérifier l'existence de combinaisons de données qui ne devraient pas être réalisables. Nous verrons en partie III que quelques individus ont été retirés de certaines analyses car ils présentaient des données absurdes. Par exemple, trois joueurs présentaient un numéro de système solaire égal à 2 pour leurs premiers événements. Or, tous les joueurs commencent au premier système dans Star Shaman et ne peuvent donc avoir un numéro 2 comme identifiant du système de leurs premiers événements. Ce sont des erreurs dans les données que l'on peut vérifier aisément : des joueurs au nombre de vies supérieur au maximum possible, un système solaire avec moins de planètes générées que le minimum nécessaire, etc.

Vérifier que les joueurs suivent le flow prévu

J'ai déjà introduit la notion de flow telle qu'elle fut conçue. Par extension, ce terme désigne à présent dans la littérature le chemin ou la narration principale servant de fil conducteur au joueur dans les jeux narratifs. En effet, la plupart des MMOGs (voir définition en II.1) sont dits d'exploration libre : l'évolution du monde est sujette aux actions des joueurs qui ne sont guidées que par le but général du jeu. La narration de ces jeux est donc très légère. Mais, il en existe tout de même une à suivre. A titre d'exemple, Myat Aung et ses co-auteurs (2019) ont analysé *Just cause 2* dans lequel le joueur doit mettre à bat un régime tyrannique en semant le chaos par des actes de rébellion. Les actes du joueur sont libres, il n'est pas guidé dans ses choix. La seule limitation provient de ses équipements. Dans ce contexte, la trame narrative contraint peu le joueur et l'analyse de ses mouvements n'en est que plus complexe.

> *« Since OWGs [Open World Games, jeux à monde ouvert] allow players to move freely instead of limiting their trajectory to set paths, this increases the complexity of making data-driven calculations as each trajectory can be theoretically unique. »* (Aung et al., 2019)

A l'inverse, certains jeux peuvent contenir volontairement plusieurs trames narratives et des voies de traverse que le joueur peut emprunter s'il le souhaite. Le risque est que le joueur se perde dans de fausses pistes et soit frustré. Nous détaillerons donc ici les manières de vérifier que les joueurs suivent la trame principale.

On peut utiliser un funnel pour voir si les étapes sont faites, et dans l'ordre, comme nous l'avons déjà expliqué. Drachen et Canossa vont beaucoup plus loin (2009) en parlant de récréer l'expérience des joueurs dans le jeu *Kane & Lynch*. Le joueur y incarne deux criminels allant de missions en missions. Les auteurs de l'article détaillent les raisons de monitorer les mouvements du joueur à travers les cartes de chaque niveau et notamment :

> *« The focus of this case study is to show how gameplay metrics can be applied to examine player navigation through level maps, and whether the players deviate from the intended path of the level design. »* (Drachen & Canossa, 2009)

Le joueur a une liberté de mouvement assez grande dans l'espace et peut donc s'écarter assez facilement du chemin prévu. Les auteurs ont créé, avec l'aide du designer, un chemin théorique large de deux mètres pour mesurer l'écart entre le cheminement des joueurs et ce chemin théorique. 85% des joueurs du niveau considéré empruntait le chemin prévu. L'analyse des points géographiques du niveau, où la proportion baissait, montrait les points de difficulté des joueurs et peut-être des situations trop difficiles. Bien d'autres auteurs font part d'analyses spatio-temporelles des joueurs pour détecter des comportements inattendus de joueurs. Le lecteur peut se référer aussi à Bauckhage et al., 2014, article dédié au clustering spatial à l'aide de deux techniques expliquées.

Myat Aung (et al., 2019) utilise aussi des données spatio-temporelles pour enregistrer le chemin pris par le joueur et combine avec des données d'équipement telles que le type de véhicule. Cela permet de vérifier qu'un joueur expérimenté disposant de tel véhicule a bien exploré l'entièreté de la carte par rapport à un joueur néophyte. C'est le sens des clusters déterminés par les auteurs qui combinent à la fois la répartition sur la carte, l'évolution dans le temps et les véhicules à disposition. Si des joueurs vont au-delà de ce qui est la norme, on peut ainsi les repérer. Cela inclut ceux qui mettent plus de temps pour un niveau ou qui dépensent plus de matériel.

Comprendre son audience

Il s'agit ici non d'observer le comportement tel qu'au point précédent, mais plutôt d'approcher les raisons de celui-ci.

> *« Basically, the purpose of game analytics is to solve a very old question that has existed since the beginning of the media: why do people play a game and what do they enjoy the most while playing it?*
>
> *Beside being a subjective intuition, game analytics definitely appeared to be the best way to answer this question. » (Thibault, 2013)*

Ainsi que Coupart Thibault le note, la raison d'un comportement est une interprétation subjective. Les données ne donneront jamais une réponse nette et définitive.

> *« When starting, try to avoid complex questions that involve external factors in the game, even if they seem crucial for you.*
>
> *For example, trying to understand why people stopped playing your game over a long period of time is usually impossible.*
>
> *Old players might stop playing because another game came out or they just got bored. Data cannot make miracles at this point of the engagement. » (Thibault, 2013, p. 68)*

En revanche, par recoupement, il est possible de déduire des choses. Anders Drachen et ses co-auteurs (2013) ont notamment établi un processus d'analyse au chapitre 4 de leur ouvrage qui part du nombre de téléchargements du jeu pour explorer successivement les raisons du fait que le jeu ne rapporte pas assez d'argent. Il s'agit là d'un exemple de jeu dans lequel le joueur est poussé à payer. Cependant, le processus, très général et propre à l'industrie du jeu vidéo, s'applique à de nombreux autres types de jeux.

Les méthodes pour comprendre son audience sont liées au profilage et à la segmentation des joueurs que nous développerons dans la section suivante. Classer les joueurs permet d'identifier des comportements récurrents et des types de joueurs, et donc de s'y adapter en matière de contenus (Rodrigues & Brancher, 2018) ou de modifier l'équilibre des profils (Bartle, 1996) pour favoriser certains types de joueurs. La littérature est très abondante à ce sujet. Il existe beaucoup d'approches particulières, spatio-temporelles notamment, qui tiennent compte de la géographie et de la temporalité du jeu.

La temporalité est soulignée par la plupart des auteurs comme une manière de prendre en compte la nécessaire évolution du comportement du joueur au fur et à mesure du jeu. Une facette de l'analyse des joueurs comprend le temps et compare les données à plusieurs instants. Drachen et ses co-auteurs (2014) insistent sur le fait que dans leur partitionnement, un joueur peut changer de groupe au fur et à mesure du jeu. Selon eux, trop peu d'études font du clustering à plusieurs points dans le temps.

Chris Lewis et Noah Wardrip-Fruin prennent le temps en compte dans leur analyse des joueurs de World of Warcraft entre deux patchs. C'est l'un des plus fameux MMORPGs (voir glossaire en II.1) et il se décline en différentes franchises (Lewis & Wardrip-Fruin, 2010). L'objectif est de comparer des résultats d'une mise à jour du jeu à l'autre, et de voir les potentiels impacts des changements réalisés.

Comme mentionné dans le point précédent, Myat Aung et ses co-auteurs ont combiné les aspects temporel et spatial dans leur analyse de *Just Cause 2* (Aung et al., 2019) pour comprendre le cheminement de leurs joueurs dans un univers ouvert et permissif. Ils ont collecté des données en trois dimensions pour tenir compte du temps et ont cartographié les déplacements des joueurs à l'aide de cartes asynchrones (figurant l'ensemble des informations dans le temps). Une première analyse comptait les déplacements d'un joueur entre deux points pour identifier les itinéraires d'importance. Dans un second temps, Myat Aung et ses co-auteurs ont publié une carte asynchrone des joueurs en fonction du moment où ceux-ci quittaient le jeu, rapidement après une première mission ou plus tard. Chaque couleur correspondait à un groupe et l'on pouvait voir les tendances de déplacement : seuls les joueurs quittant rapidement le jeu n'exploraient pas la carte. Enfin, nous citerons encore la carte des modes de transports utilisés (points de couleur selon le mode) et du nombre d'ennemis tués (traits) qui permettait d'observer des groupes plus ou moins habiles et leurs transports favoris. Cette carte permet de ne pas tenir compte du temps, c'est-à-dire, de mettre à plat tous les joueurs, quel que soit leur groupe à tous les instants t du jeu. Enfin, une analyse par niveau et mission leur a permis un même travail par tranche temporelle cette fois. Ainsi, Myat Aung (et al., 2019) peut comprendre les raisons de la dispersion ou de l'utilisation d'un artefact par ses joueurs, grâce à l'ensemble de ces analyses.

On citera également Drachen & Canossa (2009) au sujet de *Fragile Alliance*, un jeu de tir multi-joueurs. Afin de comprendre les actions des joueurs en fonction de leurs rôles (mercenaire, police et même traitre), les auteurs ont étudié les mouvements des joueurs sur la carte des niveaux du jeu. Ils ont ainsi pu voir que les joueurs devenant des traîtres, après avoir tué un mercenaire comme eux, le faisaient essentiellement à la fin de la carte. Il s'agit de l'endroit où le niveau prend fin et où le joueur garde pour lui seul l'argent volé s'il est un traître. Savoir cela permet de comprendre ce qui motive des mercenaires à en tuer d'autres. Ce n'est pas la volonté d'handicaper un joueur (Bartle, 1996 et les killers) mais tout simplement l'appât du gain en fin de parcours.

Nous avons ici essentiellement développé les approches spatio-temporelles. Il existe toutefois beaucoup d'autres manières de comprendre pourquoi les joueurs évoluent d'une manière ou d'une autre dans un jeu.

Comprendre les interactions

Comprendre les interactions de joueurs, ou de personnes si l'environnement n'est pas un jeu, sert beaucoup lorsque des données de réseaux sociaux existent. Je m'attarde sur un article en particulier qui combine l'analyse des données du jeu *Destiny* et celles des comptes twitter des joueurs. Les auteurs décrivent les objectifs de cette démarche : voir les relations entre le comportement sur Twitter et le temps de jeu, pour en tirer des profils, découvrir des sous-communautés twitter et ainsi voir les principaux sujets d'intérêts dans le temps :

> *« •Assessing the relationship between, and establishing player profiles with respect to, tweeting behaviour and playtime;*
>
> *•Uncovering subgroups within the Destiny community based on how people identify themselves in their public Twitter profile descriptions;*
>
> *•Identifying broad topics of interest to the community, their variation over time, and if and how these interests differ between subgroups. » (Wallner et al., 2019)*

En 14 mois et avec 3000 joueurs, les auteurs ont élaboré à l'aide de la méthode des k-means, sept clusters de joueurs aux comportements différents. Le cluster 1 regroupe 43% des joueurs : ils jouent peu et ont une activité faible sur twitter. Une grande partie des joueurs n'interagit donc pas, que ce soit dans le jeu ou sur Twitter. D'autre groupes jouent plus mais n'interagissent pas sur Twitter. A l'inverse, le cluster 3 contient les influenceurs : ils sont beaucoup retweetés. On voit donc ici que les interactions sociales sont fort diverses parmi ces joueurs : on a beaucoup de groupes aux comportements différents.

Un autre article me paraît intéressant à mentionner ici, d'autant plus qu'il traite aussi d'une application en réalité virtuelle à l'usage des personnes âgées (Ladly et al., 2017). Mais c'est de l'autre application présentée qu'il s'agit ici : le projet *Postcards memories*. Il a notamment pour but d'inviter les personnes à "socialiser".

> *« Our research questions explore : 1) How might a multimodal, personalized digital media application with embodied, social interaction create a virtual reality to enhance social connection and improve memory recall ? […]»*

Selon les auteurs, la réalité virtuelle n'est pas inclusive et elle ne prend entre autres pas en compte les limitations physiques ou de connaissances technologiques. Les personnes âgées sont donc exclues de cette technologie et l'article présente un exemple d'application qui leur serait utile.

Le projet se constituait d'une tablette permettant aux personnes âgées de créer des cartes postales virtuelles avec du texte et de la vidéo, et de les envoyer notamment à la famille. L'objectif était de voir si cela incitait les personnes à interagir plus avec leur environnement immédiat (soignants) ou à distance (famille, amis). Un questionnaire semi-oral complétait les commentaires en temps réel des personnes utilisant l'application. Le résultat fut que l'interaction avait augmenté de 1 point sur 5 grâce à cette tablette et un autre projet en VR. On voit ici que l'interaction est un sujet important dans des domaines très divers qui dépassent le jeu vidéo

Surveiller un phénomène

Le dernier objectif de l'analyse de données dans le jeu, surveiller un phénomène, se rencontre aussi dans d'autres domaines que le jeu à but ludique. Le serious game (voir définition en II.1) en fait un usage important. Cela est souvent couplé à l'observation d'interactions comme dans le cas de notre article sur les personnes âgées et leurs interactions. Le second projet de cet article (Ladly et al., 2017) consistait à donner accès dans un environnement virtuel à des vidéos d'actualité des 24 années précédentes. Deux systèmes monitoraient les mouvements du corps et des mains. Le but était de comprendre si naviguer sur l'interface présente dans cet environnement virtuel pouvait améliorer la mémoire tout en préservant une activité ludique :

> *« Can such new media and VR/AR applications assist elders by enhancing memory, affect, and other cognitive abilities ? Can we produce feelings of pleasure and delight in our elder audiences ? »* (Ladly et al., 2017)

Les résultats montraient une amélioration de 0,88 points sur l'échelle des auteurs.

Le domaine de la santé est une source inépuisable d'article sur la surveillance de ce genre de phénomène (Li, 2020, Petsani et al., 2018) et la VR y est beaucoup employée. On trouve notamment une application VR ludique combinée à un vélo (KASSIM, 2018) pour monitorer les données physiologiques du cycliste ; ou un jeu sérieux sur l'achat de produits sains pour les personnes obèses (Jayachandran et al., 2017), etc. Les citer tous, relève de l'impossible. L'article de McGregor (et al., 2017) paraît être intéressant car il aborde le monitoring du stress. Les auteurs ont développé une application de gestion du stress en VR –

on s'éloigne légèrement du jeu – qui capte les données physiologiques de la combinaison utilisée par le testeur. Elle permet notamment de capter le rythme de la respiration, en plus de l'oxygénation du sang et d'autres facteurs.

Pour revenir au jeu, Perrot (et al., 2016) propose aussi un article intéressant car il traite de l'utilisation des données des joueurs de jeux d'argent. Le but de l'auteur est de dépister les comportements excessifs en effectuant un clustering des joueurs. Créer des groupes de joueurs dont certains sont sujets à une addiction au jeu permet de prédire ensuite le comportement de nouveaux joueurs et éventuellement d'entamer des actions de préventions. Cette étude s'est en effet appuyée sur les données de l'ARJEL, l'autorité de régulation des jeux en ligne. On pourrait imaginer son intérêt à pouvoir prédire des addictions.

Dans un autre registre Richard Bartle (Bartle, 1996) est particulièrement célèbre pour avoir créé dans les années 1990, des profils théoriques de joueurs (que nous développerons dans la section suivante) en se basant sur ses connaissances des premiers MUD (*Multi-users dongeons*). Les MUDs sont des espaces virtuels soit tournés vers la sociabilisation (forums), soit vers le jeu : on peut en rapprocher certains des MMOGs. Selon Bartle, l'évolution du nombre de joueurs de tel ou tel type est un phénomène à surveiller. En effet, un type de joueurs peut en chasser un autre, et ainsi déséquilibrer le public du jeu, voire éliminer les joueurs initialement visés. Bartle incite donc les développeurs du jeu, quoi que sans donner de méthode technique, à surveiller ces phénomènes et à faire des actions correctives si nécessaire, en introduisant des éléments favorisant le type de comportement voulu. Bartle développe aussi le concept d'interaction, en décrivant les conséquences des rencontres entre chaque type de joueur. A titre d'exemple, un des types de joueurs est appelé « Achievers ». Ces joueurs sont mus par l'appât des récompenses, les trophées, le fait d'être admirés. Si un *achiever* interagit avec un autre, il y a deux possibilités : soit ils seront des rivaux farouches, soit ils coopéreront pour obtenir une grande distinction. Dans tous les cas, l'interaction sera forte et une amitié durable peut en ressortir.

b) FOCUS SUR LE JEU VIDEO

Après avoir détaillé chacune des facettes de l'analyse de données, il me paraît intéressant de se focaliser sur le jeu vidéo. Il ressort que les données de jeux vidéo ont des caractéristiques qui ressortent de l'ensemble de la littérature.

Les données de jeu sont rarement seules

Les données de jeu sont presque toujours accompagnées d'autres données obtenues par des canaux différents. La plupart des études utilisent des questionnaires pour compléter les données des sessions de jeu. Ce fut même longtemps le seul canal de données dans le jeu vidéo (Drachen et al., 2014). Ils permettent notamment de rassembler les données psychologiques et émotionnelles difficiles à capter par ailleurs. J'aimerais ici citer Nicolas Ducheneaut et Nicolas Yee (Ducheneaut & Yee, 2012) dont l'article décrit le travail sur les joueurs de World of Warcraft, le premier jeu en ligne massivement multi-joueur, pour déterminer la personnalité de chacun. Parmi les méthodes employées figure un questionnaire de 20 questions basées sur l'inventaire des 5 grands facteurs. Il s'agit d'une méthode traditionnelle en psychologie pour mesurer cinq traits de la personnalité. Le fait que le jeu soit agréable par exemple se mesure avec les données des émoticônes envoyées par les joueurs et certaines parties du questionnaire :

> *« La définition de ce trait informe que les individus qui ont un score élevé en agréabilité tendent à être amicaux, soucieux des autres et coopératifs, alors*

que ceux avec un score plus faible tendent à être suspicieux, antagonistes et compétitifs. Dans wow, les individus agréables utilisent plus d'émotions positives (13, 15, 16), comme les câlins et les applaudissements, et ils préfèrent également les activités non violentes comme l'exploration (24), les professions (27), les événements mondiaux (29), la cuisine (30) et la pêche (31). À l'inverse, les joueurs peu agréables préfèrent les aspects plus compétitifs et antagonistes du jeu. Ils aiment tuer d'autres joueurs (8, 9). » (Ducheneaut & Yee, 2012)

Le questionnaire sert par ailleurs à établir le profil socio-démographique des joueurs. Ainsi, les données créées par cet outil, combinées à celles du jeu, affinent la connaissance des chercheurs sur leur population cible et permettent d'interpréter le profilage effectué en termes d'émotions.

Il en va de même pour Felipe Paiva et ses co-auteurs (Paiva et al., 2018) qui utilisent un questionnaire pour comprendre les préférences de types et mécaniques de jeux auprès de joueurs de jeux de cartes à collectionner. Les réponses ont notamment servi à comprendre les caractéristiques des groupes de joueurs :

« This suggests that the opinions players have about themselves are closely related to the way decks are built and the corresponding buying habits. Moreover, there is also a relationship between the level of interest in the narrative and how players' choose their decks for performance and/or fun purposes.» (Paiva et al., 2018)

Anders Drachen explique également en quoi un questionnaire peut se combiner avec les données de jeu. Il permet notamment de collecter les types de jeu ou les mécaniques aimées par les joueurs, les noms des jeux déjà achetés, en plus des informations dont nous parlions plus haut. Ainsi, le questionnaire permet d'établir des profils de joueurs à long terme pour une compréhension plus profonde de ceux-ci :

« It is important to decide if you are interested in either short-term game mechanic fi xes or long-view profiling and modeling of data for a deeper understanding. If a long-term view is desired, such as studying various factors that may contribute to player churn, it means committing to collecting basic profile data prior to the start of the alpha or beta phase as well as soliciting input from players by presenting popups with survey questions for more in-depth understanding. » (Drachen et al., 2013)

Certains auteurs mentionnent aussi l'intérêt de travailler avec une personne spécialisée en psychologie pour créer et interpréter les résultats. Outre les questionnaires, il existe bien d'autres types de données qui renforcent celles du jeu. Nous en avions mentionné quelques-uns plus haut dans les usages de l'analyse de données. En voici un aperçu en détail.

Les types de données sont divers

Il existe plusieurs classifications des données employées dans le jeu vidéo. Drachen, Seif et Canossa décrivent les plus abouties (Drachen et al., 2013). Ils les résument ainsi : *generic metrics* pour les données applicables à tous les jeux vidéo, *genre specific metrics* selon le type de jeu et *game specific metrics* propre à chaque jeu. Dans la pratique, ils identifient trois sources de données :

> *« • Customer metrics : Covers all aspects of the user as a customer, e.g. cost of customer acquisition and retention. These types of metrics are notably interesting to professionals working with marketing and management of games and game development.*
>
> *• Community metrics : Covers the movements of the user community at all levels*
>
> *of resolution, e.g. forum activity. These types of metrics are useful to e.g. community managers.*
>
> *• Gameplay metrics : Any variable related to the actual behavior of the user as a player – inside the game, e.g. object interaction, object trade, and navigation in the environment. Gameplay metrics are the most important to evaluate game*
>
> *design and user experience, but are furthest from the traditional perspective of*
>
> *the revenue chain in game development, and hence are generally under prioritized. » (Drachen et al., 2013)*

Ils vont plus loin en mentionnant des features qui sont les valeurs en entrée des algorithmes de machine learning. Elles ne correspondent pas toujours aux variables elles-mêmes et furent mentionnées dans l'étude de Tomb Raider – Underworld (Drachen et al., 2014) puis reprises dans leur ouvrage général :

> *« Once variables are measured and turned into metrics data, it might be necessary to*
>
> *extract features. Features are the concrete values entered in complex data mining*
>
> *operations and used as input for different algorithms. » (Drachen et al., 2013)*

En ce qui me concerne, je remarque que l'on distingue six types de données dans le jeu vidéo : les données marketing, les données sociales, les données de gameplay (que j'appelle données de jeu), les physiologiques, les paramètres du jeu et enfin les données psychologiques et/ou émotionnelles. Nous avons déjà abordé les données physiologiques dans le domaine de la santé (voir II.2.a).

Nous avons mentionné que les données psychologique ou émotionnelles sont souvent récoltées par un questionnaire, tant il est difficile de juger les émotions et réactions psychologiques d'un joueur en observant sa progression dans le jeu. Certains articles recommandent des systèmes de captation des mouvements du corps et des expressions faciales des joueurs car elles montrent nos émotions. Mais ces auteurs sont marginaux.

J'aimerais revenir sur l'article de Rafael de Albuquerque et Francisco Fialho (de Albuquerque & Fialho, 2015) qui recherchent les facteurs du fun, notion proche de la motivation des joueurs et de leur immersion dans le jeu. A partir de 27 éléments, ils créent six dimensions du fun qui les amènent à trouver huit profils de joueurs. Ce qui me paraît intéressant est leur prise en compte des émotions du joueur. Il faut dire que

le fun est une émotion. Les auteurs prennent d'abord en compte le sentiment d'immersion qui devient une dimension de leur analyse associée aux affirmations suivantes :

« [1] Play games where I can do impossible things which I could not do in real life

[2] Admire fantasy worlds and impossible things

[3] Feel that I am in control of the game, I know what can happen and how to react

[4] Feel that I am in command, that I am powerful and I can choose what I want

[5] Feel immersed in the game and everything looks like real » (de Albuquerque & Fialho, 2015)

On voit que l'immersion est associée au feeling, à la sensation. Elle se rapproche donc des émotions car celles-ci sont intimement liées aux sensations. Les émotions se retrouvent dans de nombreux items de leur étude, notamment liés aux composantes de l'empathie et du grotesque. Le grotesque contient notamment les *wild emotions*, les sentiments de peur, de pouvoir, etc...

« Grotesque: this term is associated with the appreciation of evilness, cruelty,

ugliness, bizarre, and randomness. […]

We can imagine that all these factors are associated with wild emotions, as

described by Poole (2000), and the voluntary submission to primitive feelings

such as danger and chaos. » (de Albuquerque & Fialho, 2015)

Les auteurs analysent aussi la joie, le sentiment d'accomplissement et bien d'autres émotions des joueurs. Les résultats de leurs études leur permettent notamment de remettre en cause la violence dans les jeux car les sentiments violents sont finalement peu appréciés par les joueurs.

« The appeal of violence and of the forbidden (such as stealing, killing or

destroying) was given some of the lowest scores, which suggests the hypothesis that

maybe the violence frequently found in commercial games are not indispensable to

their success. » (de Albuquerque & Fialho, 2015)

D'autres auteurs abordent la compréhension des émotions du joueur, mais nous ne les citerons pas ici car cela demanderait une étude à part entière. Le lecteur pourra se référer à l'étude des traits de personnalité des joueurs de *Fallout 3* (Spronck et al., 2012).

Parmi les autres types de données, on trouve les données marketing. J'ai découvert deux facettes de ces dernières : le monitoring des joueurs et celui des actions marketing. Le premier vise à contrôler le nombre d'achats du jeu ou dans le jeu, le nombre d'installations du jeu, les DAU (voir glossaire en II.1), l'engagement ou encore le taux de rétention. Pour ce qui est des actions marketing, il existe les sources de trafic vers le jeu ou les boutiques en ligne le vendant, le taux de conversion à la suite d'une campagne de publicité et enfin la viralité ou l'ARPU (voir II.1). Toutes ces données font partie des indicateurs de

performance, communs à tous les jeux, d'après Thibault (2013). Elles sont aussi bien développées dans l'ouvrage d'Anders Drachen et ses co-auteurs (Drachen et al., 2013). Ces données marketing ne sont pas à confondre avec les données sociales qui sont utilisées en marketing.

Dans ce cas, il s'agit des informations socio-démographiques, telles que l'âge et le sexe, mais également des informations utiles aux relations sociales à travers le jeu. J'entends par relations sociales les données de réseaux sociaux (identifiant, nombre d'abonnés, etc.…) et celles des préférences (types, mécaniques de jeu, autres titres achetés) qui montrent le type de joueur et les communautés auxquelles le joueur est lié. Encore une fois, Anders Drachen est à citer en référence (Drachen et al., 2013) pour ses mentions des « player social interactions ». Mais je retiens également Wallner (et al., 2019) pour les données Twitter. Ces derniers auteurs mentionnent aussi comme données intéressantes le nombre de comptes suivis, de tweets envoyés et de retweets. Comme nous l'avons vu, l'aspect social est également assez développé dans le domaine de la santé, notamment chez Ladly (et al., 2017). L'un des types de joueurs de Richard Bartle (1996) est nommé *socializer* : ce sont les joueurs qui animent les forums et discutent avec les autres. Quoique Bartle ne parle pas d'analyse des données, on remarque que dès les premiers MUDs, la composante sociale est partie prenante du jeu vidéo.

Par ailleurs, les paramètres de jeu sont également un type particulier. Ils font partie des données de jeu car récoltés durant les sessions, mais ils ne sont pas liés au gameplay. Il s'agit par exemple du niveau de difficulté choisi par le joueur, sa main dominante, la langue du jeu, le nombre de sessions, etc… (Drachen et al., 2013, Fernandes & Levieux, 2019). Drachen (et al., 2014) développe quatre paramètres ajustables par le joueur, dont le temps après un saut pour sauvegarder une action d'attraper un objet, ou encore les paramètres de munitions :

> *« […] players can change various parameters of the game, and four*
>
> *of these impact directly on gameplay and were therefore included: Ammo adjustment, enemy hit points, player hit points, and saving grab adjustment (which adjusts the time a player has to secure a handhold after a jump). » (Drachen et al., 2014)*

Enfin, les données de gameplay (voir glossaire en II.1) sont directement liées à celui-ci, à la narration, c'est-à-dire, aux actions faites dans le jeu qui ne sont pas du paramétrage. Charles & Black (2004) font bien la distinction entre les données de paramétrage et celles de gameplay :

> *« Two sources of information can be used to identify the player-type for a game: firstly the information that a player provides when they begin the game by setting basic preferences and inputting information about themselves. The second source of information should be taken from the player's gameplay habits and performance in-game.»*

Il s'agit des passages de niveaux, des affrontements avec des ennemis ou des récompenses reçues au fil de la narration, etc… (Petsani et al., 2018, Drachen et al., 2014). Ces récompenses ne sont pas à confondre avec les bonus et offres spéciales issues des campagnes marketing. Drachen (et al., 2013) résume en quelques phrases le vaste éventail des données de gameplay :

> *« […] Covers all in-game actions and behaviors of players, including navigation,*
>
> *economic behavior as well as interaction with game assets such as objects and entities. »*

Ces données servent notamment au game designer pour éventuellement corriger, améliorer, créer des éléments en fonction des difficultés ou goûts des joueurs. La plupart des articles de recherche se basent sur des données de jeu (seuls quelques-uns n'utilisent qu'un questionnaire), et je retiens encore une fois, comme premier exemple, Aung (et al., 2019) pour la variété des données collectées : cela va du ratio d'ennemis tués sur le nombre de morts, au nombre de véhicules requis par le joueur, en passant par le nombre de ressources collectées, et bien d'autres. J'aimerais ici expliquer le ratio : il sert à mesurer l'expertise du joueur. Si le ratio est petit, alors le joueur est mort plus souvent qu'il n'a tué d'ennemis. Le joueur est donc considéré de niveau faible : il peut soit être débutant, soit n'avoir pas un bon niveau. Beaucoup d'auteurs créent ainsi des données agrégées pour mesurer l'efficacité, l'expertise ou encore l'intérêt du joueur. Dans Drachen (et al., 2013), c'est le cas du *skill level* qui est une agrégation de données permettant de mesurer le niveau de compétence du joueur.

Toutes ces données de gameplay sont spécifiques au jeu et d'autres auteurs collectent des données bien différentes de celles vues plus haut. Cai (et al., 2019) utilise peu de données de gameplay. En effet, les auteurs ont tenté de partitionner 120 000 joueurs, en faisant face nombre massif de données disponibles. Les données de gameplay utilisées sont agrégées : le nombre de quêtes réussies, le temps de jeu total, le temps par jour moyen. Les autres données sont des paramètres ou sont d'ordre social. Avec aussi peu de données qu'une dizaine, les auteurs parviennent à déterminer les variables importantes pour le clustering et démontrent ainsi comment faire face au nombre très important de données dans un jeu vidéo.

Pour clore cette présentation des objectifs de l'analyse de données dans le jeu, au sens large, j'aimerais à nouveau souligner le fait que ces objectifs se combinent. En effet, dans l'industrie du jeu vidéo, ils sont liés dans un unique but : retenir les joueurs dans le jeu et les faire acheter s'il s'agit d'un jeu free-to-play. Nous avons vu que l'analyse utilise des données de tous types et de sources différentes. Toutes ne proviennent pas des sessions de jeu : elles peuvent aussi être collectées par questionnaire, sur les réseaux sociaux ou encore les sites intéressants pour le marketing (boutique en ligne, site web). Mais tous les auteurs font face à une réalité : un jeu vidéo génère beaucoup de données. La plupart d'entre eux utilisent le clustering pour faire ressortir des groupes de joueurs de cette masse. Nous allons donc à présent nous intéresser plus directement au clustering pour comprendre ce dont il retourne et présenter les méthodes, dont celle que nous emploierons en troisième partie.

3. INTRODUCTION AU CLUSTERING

Le clustering est une méthode de partitionnement des données. Elle permet de créer des groupes basés sur les caractéristiques des individus considérés. Il s'agit de méthodes d'apprentissage non-supervisé, c'est-à-dire utilisant des données non étiquetées ou non classées par groupes au préalable. L'objectif est d'utiliser un algorithme ou un modèle statistique pour trouver les caractéristiques communes de certaines données qui vont ensuite former des groupes. Une fois le résultat obtenu, on peut alors voir quelles sont ces caractéristiques clivantes.

En ce qui concerne le jeu vidéo, le clustering est une méthode (en réalité, il recouvre plusieurs méthodes de machine learning) très utilisée. En effet, un jeu peut générer des millions de données (Drachen et al., 2013), notamment les plus grands MMOGs (voir glossaire en II.1). Il ressort de la littérature que la difficulté réside dans le nombre de données, aussi bien que dans le fait de ne pas avoir d'hypothèse de départ quant aux comportements des joueurs.

« However, due to the huge volume and extreme complexity in online game data collections, grouping players is a challenging task. » (Cai et al., 2019)

En effet, comment discerner des comportements lorsque le panel contient plusieurs milliers de joueurs (120.000 dans Cai et al., 2019 !) générant des données aussi variées qu'un nombre de niveaux réalisés, des lieux visitées, des équipement utilisés ou encore des types d'ennemis affrontés ? Le clustering permet de trouver des schémas dans ces données massives, et fait partie de ce qu'Anders Drachen et ses co-auteur (Drachen et al., 2014) appellent la catégorisation.

« The term categorization is here used as a catch-all for any analytical technique which collapses a high number of users into a few profiles, irrespective of the specific method applied (e.g.segmentation, clustering and classification). » (Drachen et al., 2014)

Ils mentionnent aussi tout l'intérêt du clustering :

« […] Behavioral datasets from major commercial game titles of the "AAA" grade generally feature high dimensionality and large sample sizes, from tens of thousands to millions, covering time scales stretching into several years of real-time, and evolving user populations. This makes dimensionality-reduction methods such as clustering and classification useful for discovering and defining patterns in player behavior. » (Drachen et al., 2014)

Le profilage est quant à lui une méthode différente qui demande de créer les profils avant (profils pré-existants) ou après (partitionnement) l'analyse des données, en incluant un facteur psychologique. Les auteurs utilisent volontiers le terme de profiling pour parler à la fois de profilage et de partitionnement. Cependant, l'objectif final reste de comprendre qui sont les joueurs d'un jeu :

« The purposes of profiling can be varied, from design evaluation, progression analysis, user experience evaluation, and even purely explorative. Jointly, profiling helps build an understanding of the users. » (Aung et al., 2019)

A noter dans Drachen et al., 2014 que les auteurs mentionnaient à l'époque combien le clustering, à partir des données des joueurs, était peu répandu par rapport aux tentatives de créer des profils théoriques :

« Research focusing on clustering and classification remains infrequent. Categorizing players into behavioral types has been an important topic in game research for decades, and since the seminal essay by Bartle [20], which divided players into four types based on the authors personal experience, has generated a number of attempts to develop player behavior categories, initially from survey data but increasingly from in-game behavioral telemetry, e.g. » (Drachen et al., 2014)

Nous allons au préalable aborder le profilage de manière historique. En effet, dès les années 1990 et donc bien avant l'analyse de données en tant que telle, des profils de joueurs théoriques sont développés par Richard Bartle. Je me propose donc de décrire son approche, ainsi que certaines des plus importantes qui ont suivi et qui ont donné lieu au profilage. Nous verrons ensuite le clustering spécifiquement, en expliquant les méthodes applicables. Nous terminerons par une synthèse des lectures en réfléchissant à l'application possible de tous ces concepts à Star Shaman.

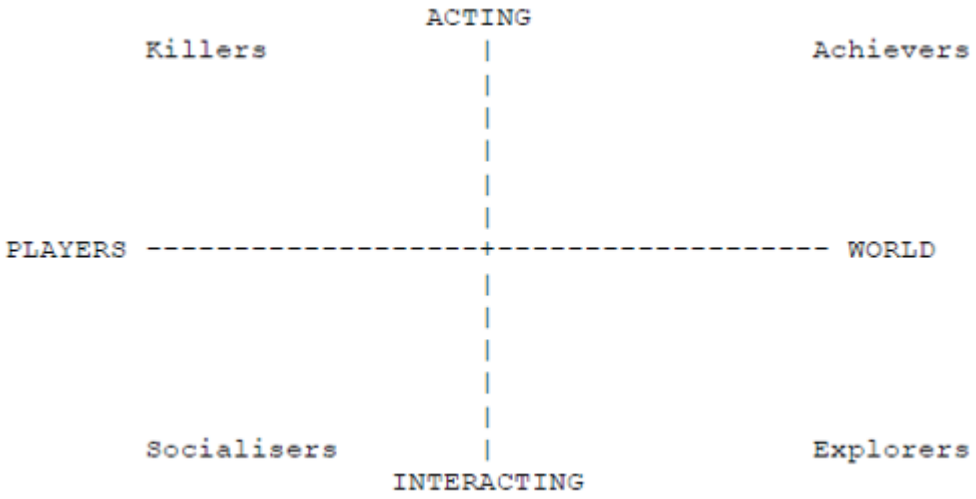
a) LE DEVELOPPEMENT DE PROFILS THEORIQUES DEPUIS LES ANNEES 1990

Les profils théoriques se sont multipliés depuis Richard Bartle en 1996 et, avec le développement de la gamification (voir glossaire en II.1), ils ont donné lieu à des grilles d'analyses très poussées. La plupart ne concernent plus les jeux uniquement, mais l'ensemble des processus auxquels la gamification s'applique : résolution de problème, synthèse de protéine du Sida, recrutement (Marczewski, 2015) et formation (Chou, 2014). Je présente ici trois approches pour fixer les idées.

Richard Bartle et les premiers jeux en réseaux

Richard Bartle a mené une enquête en 1996 sur les joueurs d'un des premiers MUD ou Multi-users Dungeon (Bartle, 1996). Ce sont les ancêtres des MMOGs (voir glossaire en II.1) en ce qu'ils sont des jeux pour la plupart. Toutefois, certains ressemblent plus à des réseaux sociaux car ce sont leurs forums qui rassemblent les joueurs.

Bartle décrit quatre profils en fonction de deux axes sur le schéma ci-dessous. En abscisses, il observe si le joueur est plus enclin à interagir avec les autres joueurs ou avec le monde du MUD. En ordonnées, on note s'il aime interagir ou agir.



Quatre profils en ressortent : *killers*, *achievers*, *socializers* et *explorers*. Les premiers sont une minorité et représentent le type de joueurs honnis par les autres. Leur motivation réside dans tuer les autres joueurs et devenir ainsi des prédateurs. Ils en retirent la satisfaction d'avoir affronté un personnage qui se défend et a de nombreux équipements. Le fait que le joueur tué a perdu ce qu'il avait mis du temps à construire, rend la victoire plus satisfaisante pour un *killer*.

Les *achievers* sont motivés par l'accumulation de richesse et de prestige. Ces joueurs aiment faire des quêtes, recevoir des récompenses et être admirés des autres joueurs. Ils sont très réceptifs aux défis et particulièrement sensibles au fait de tout perdre s'ils sont tués par un *killer*. Ils sont très attachés aux fonctionnalités de leaderboard (tableau des scores des joueurs avec leurs rangs). Ils ne tuent que des rivaux potentiels.

Les *socializers* sont des joueurs portés sur la discussion avec d'autres. Ils sont très actifs dans les forums et évoluent peu dans le monde en lui-même. Ils aiment aider les autres et deviennent des sources d'informations très utiles à force de discussions. Ils peuvent notamment donner des conseils aux joueurs et informer les *achievers* de nouveaux défis. Toujours d'après Bartle, ce sont des victimes très faciles des *killers* car ils n'accumulent pas de quoi se défendre en cas d'attaque. Ils peuvent tuer, mais c'est rare, et uniquement pour venger un ami.

Enfin, les *explorers* sont des explorateurs. Ils aiment parcourir le monde du jeu, son univers. Ils aiment comprendre comment le jeu fonctionne et éventuellement trouver des bugs. S'ils accumulent des équipements, c'est essentiellement pour se défendre face aux *killers*. Ils ne sont pas forcément intéressés par la narration du jeu mais plutôt par les lieux développés. Ils discutent dans les forums pour obtenir des informations, mais les autres joueurs ne les intéressent pas trop. Gagner des points leur sert à se rendre dans des lieux accessibles aux joueurs plus expérimentés, mais en aucun cas à obtenir des récompenses. Ces joueurs peuvent tuer pour se défendre.

Ces profils sont bien évidemment des archétypes et Bartle développe son analyse avec les interactions entre ces types. Elles servent à donner des conseils aux développeurs afin de favoriser l'un ou l'autre type de joueurs, et ainsi garder le public visé du jeu. Voici quelques-unes des interactions selon Bartle. Lorsqu'un *achiever* rencontre un autre, il peut y avoir rivalité ou coopération pour faire un défi complexe. Cela génère une solide amitié entre les joueurs. Entre un *achiever* et un *explorer*, l'équilibre est précaire. Si les derniers sont trop nombreux, les *achievers* auront moins à faire et quitteront le MUD. Dans cette interaction, on a soit de mauvais joueurs qui s'intéressent à la structure du jeu parce qu'ils ne sont pas compétents, soit des excentriques très bons. Ces derniers donnent de bonnes informations aux *achievers* qui progressent ainsi rapidement.

Les relations entre *explorers* et *socialisers* sont quasiment inexistantes, les premiers ne trouvant aucun intérêt aux seconds, sauf à discuter d'un de leurs thèmes d'expertise. Les *socialisers* sont particuliers car ils s'auto-entretiennent. Si leur nombre augmente à un moment, ils vont en attirer d'autres constamment. Si certains quittent le jeu, ils en entraîneront d'autres. Un nombre élevé de *socialisers* fera venir des *killers* et il faut donc garder un équilibre pour que les premiers restent.

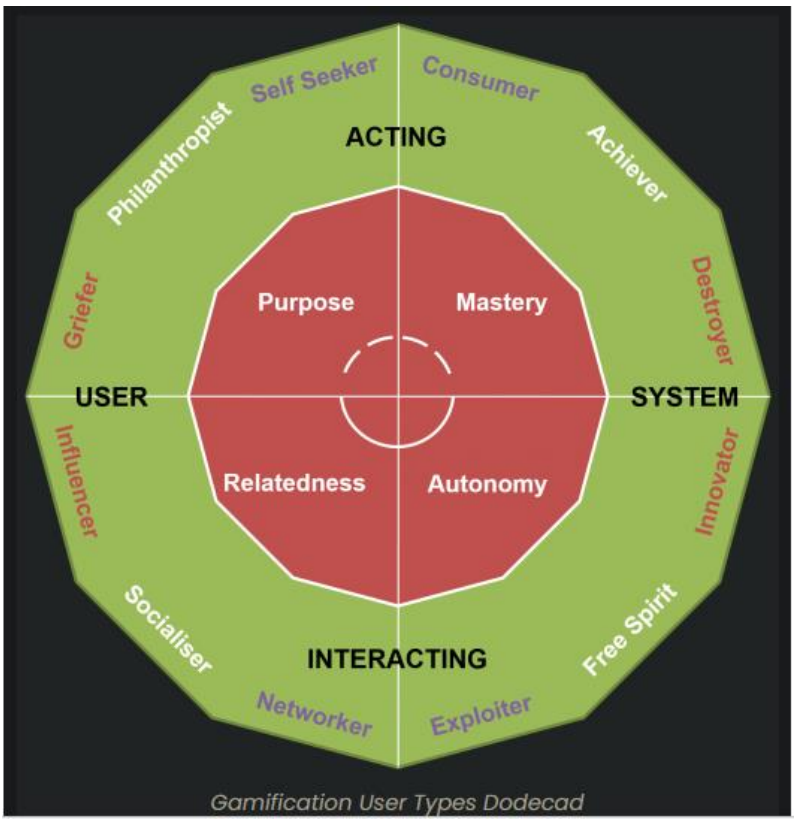
Une dernière interaction particulière réside entre les *explorers* et les *killers*. Les seconds attaquent peu les premiers car ceux-ci les ignorent assez souvent. Cela empêche les *killers* de se satisfaire de l'attaque. Leur réputation est encore plus atteinte si l'*explorer* les bat. Les *explorers* quant à eux représentent une population inerte car seule une très forte augmentation des *killers* peut les faire quitter le jeu.

Beaucoup d'autres interactions et conseils sont développés par Richard Bartle ; je ne les développerai pas ici. Il est important de noter que ces profils sont la base des analyses de joueurs depuis lors ; ils sont toujours discutés aujourd'hui dans la littérature professionnelle. Richard Bartle les a développés pour les MUDs des années 1990. Et certains n'étant pas considérés comme des jeux mais plutôt comme des réseaux sociaux, ces profils s'appliquent à un certain type d'univers.

Le dodécaèdre de Andrzej Marczewski

Andrzej Marczewski (2015) a adapté les profils de Richard Bartle au jeu moderne et à tous les secteurs touchés par la gamification. Au-delà des profils seuls, il fait le lien entre des profils et leurs motivations. Richard Bartle avançait déjà les motivations de chacun des types de joueurs. Elles sont développées et actualisées dans le dodécaèdre (ci-dessous). On y retrouve les deux axes de Bartle, le monde ayant été remplacé par le système car un jeu moderne est avant tout une certaine mécanique de jeu à laquelle on peut appliquer toutes sortes d'univers.

Marczewski se base sur quatre motivations qui viennent compléter les axes : avoir une mission, maîtriser, être autonome et interagir.



Le dodécaèdre ci-dessus est l'aboutissement d'une réflexion à partir de six types. Marczewski reprend les quatre types de joueurs de Bartle et en ajoute deux. Dans les six types on trouve les socialisers, disruptors, free spirits, achievers, philanthropists et les players. Les socialisers sont motivés par le lien social, l'interaction avec les autres comme chez Bartle. Les free spirits aiment créer et explorer et sont plus ou moins l'équivalent des explorers de Bartle. Les achievers sont aussi les mêmes : ils aiment maîtriser les choses et remporter des défis. Enfin les philanthropists sont des altruistes qui n'espèrent pas de récompenses en échange de leur aide. Ils sont motivés par un but noble, une cause.

La complexité du dodécaèdre vient du fait que selon Marczewski, les players et disruptors sont des profils extrinsèques, à savoir, qui ne recherchent rien de particulier dans le jeu. Les premiers ont envie de jouer, quel que soit le jeu et ils ne requièrent pas forcément d'éléments particuliers pour le faire : ils accumulent les récompenses pour évoluer dans le système du jeu, mais ne recherchent rien pour leur profit personnel ou pour ce qui est d'enrichir leur personnalité. Les seconds n'ont pas envie de jouer et sont motivés par le changement : ils cherchent à bousculer positivement ou négativement le jeu. Ces deux profils extrinsèques se déclinent chacun en quatre sous-profils, d'où les douze types de joueurs du dodécaèdre.

Les disruptors ne sont pas les killers de Bartle car ils peuvent avoir une influence positive. Ils sèment le chaos dans le système pour des raisons positives ou non. Trouver des bugs est plutôt positif et c'est l'objectif des improvers. Les influencers vont aussi faire changer le système en influençant les autres joueurs et en devenant éventuellement des supporters des mises à jour. Enfin, les griefers sont les killers de Bartle, et les destroyers vont hacker le système pour trouver des bugs mais veulent aussi détruire le système qui ne leur plaît pas. On voit ici que l'approche de Marczewski est beaucoup plus fine que celle de Bartle car adaptée aux jeux modernes.

Les sous-types de players sont pensés dans ce sens également. Les Self-seekers sont comme les philanthropist mais sans cause : ils aident contre rémunération. Les consumers ressemblent aux achievers car ils aiment les récompenses mais sans le *challenge*. S'ils peuvent les acquérir sans défi, cela leur convient. Ils désirent avant tout évoluer dans le jeu. Les networkers vont socialiser pour obtenir des

informations ou des récompenses, et non par envie d'échanger. Enfin, les exploiters vont explorer le système mais non par motivation de l'exploration. Ils recherchent, comme tous les players, un moyen d'avancer dans le jeu et donc d'obtenir des récompenses.

Dans un article plus récent auquel Marczewski a participé (Tondello et al., 2019), les auteurs s'interrogent sur les profils théoriques imaginés par lui et d'autres chercheurs. En effet, la plupart notent que les jeux en ligne sont la cible de ces recherches, et que les profils ne s'appliquent pas forcément aux autres jeux, comme, par exemple, les jeux de tirs qui représentent pourtant une importante part de marché dans le jeu vidéo. En réalité virtuelle, les jeux de tirs sont d'ailleurs assez nombreux.

« Hamari and Tuunanen (2014) conducted a systematic review of these and other player type models to investigate their commonalities. The authors note that MMOs and online games are more frequently covered than other genres in several of these studies, and thus that this compromises the generalizability of these models. » (Tondello et al., 2019)

La motivation dans le serious game par Yu-Kai Chou

Une autre approche de la motivation a été développée pour le serious game (voir glossaire en II.1) par Yu-Kai chou. Se basant sur Bartle, Marczewsi et d'autres, il a mis en forme l'Octalysis (Chou, 2015), polygone à huit côtés rassemblant les motivations des personnes. Son approche est légèrement différente des autres car elle ne prend en compte que la motivation à prendre part au jeu. Chou n'établit pas de profils mais on peut utiliser son cadre d'analyse pour en créer.

Chou liste huit motivations dont voici notre traduction de l'anglais : le sens, l'autonomisation, l'influence sociale, l'imprévisibilité, l'évitement, le manque, la propriété et l'accomplissement. La première correspond au but ou à la cause, déjà cité par Marczewski : le joueur aura l'impression de participer à quelque chose de plus grand que lui ou d'avoir été choisi dans un but particulier. Ceci peut être une impression développée lorsqu'un événement rare se produit par hasard pour un joueur. L'autonomisation consiste à créer, modifier et essayer des variations dans le processus créatif pour aboutir à une création commentée publiquement. Une fois enclenché, ce processus s'auto-alimente chez les joueurs et ne nécessite pas d'éléments additionnels développés.

Quant à l'influence sociale, c'est une motivation pour certains joueurs qui recherchent la relation, le fait d'être accepté. A ce titre, elle peut aussi engendrer de l'envie et de la jalousie : certaines personnes pourront vouloir ce que d'autres possèdent pour obtenir du prestige ou envier ceux qui sont reconnus par le groupe pour leurs compétences/réalisations.

L'imprévisibilité s'accompagne de la curiosité : c'est le fait d'aimer ne pas savoir ce qui va se passer. Chou précise d'ailleurs que c'est la base de l'addiction au jeu, à un livre ou une série. Par ailleurs, l'évitement concerne ceux qui essayent d'éviter une situation ou un sentiment, et auxquels l'imprévisibilité fait plutôt peur. Ils cherchent à éviter une perte de points ou l'obligation de reconnaître l'inutilité de leurs actions, etc... Cela concerne également ceux qui ne peuvent manquer une promotion ou un événement ayant lieu dans le jeu. Dans un autre registre, la pénurie ou le manque est lié à l'évitement : c'est éviter de manquer de quelque chose dans le jeu. Ces personnes reviendront donc dans le jeu si une récompense leur est offerte. Certains sont aussi motivés par la propriété, le fait d'accumuler et de posséder.

Enfin, le sentiment d'accomplissement réunit ceux qui aiment développer ou créer, et ceux qui aime remporter des défis. C'est pour eux la raison d'être des points, des panneaux de victoires dans un jeu. Ils montrent les compétences de la personne. Le sentiment d'accomplissement est aussi, d'après Chou, le plus facile à satisfaire dans le développement d'un jeu.

Les motivations de Chou ont leurs natures propres : certaines rendent la personne puissante sans sentiment d'urgence. D'autres induisent l'urgence, l'obsession et parfois l'addiction et font se sentir mal l'utilisateur. L'Octalysis place d'ailleurs les motivations selon leur côté positif ou négatif : ce sont les concepts de *white hat gamification* et *black hat gamification*. Les motivations se répartissent aussi entre impact à long-terme ou court-terme sur la personne.

Lorsque la personne fait quelque chose parce qu'elle a peur de manquer ou d'agir, ou parce qu'elle a du mal à atteindre un but, alors les motivations sont négatives et le joueur (s'il s'agit d'un jeu) ne se sent pas bien. Chou précise que cela peut aider à atteindre un objectif personnel si cela est équilibré par une éthique personnelle.

J'aimerais ici insister sur deux points qui m'ont fait retenir le travail de Yu-Kai Chou. Tout d'abord, il aborde une potentielle neuvième motivation qui serait la sensation. Faire une action pour le plaisir de nos sens serait une motivation. Néanmoins Chou ne la garde pas car c'est un aspect plus physique que psychologique selon lui. Ceci me fait réfléchir à la réalité virtuelle : la sensation, du fait de la technique et de son impact sur nos sens (impact parfois négatif avec la nausée), a-t-elle une place plus importante en VR ? Ceci est peut-être un élément à prendre en compte dans notre analyse et il rejoint l'immersion que nous avons brièvement abordée (voir II.2.b). Nous en reparlerons lors du choix de nos données (voir III.3).

Le second point intéressant, en ce qui nous concerne, dans l'Octalysis de Chou est la capacité à utiliser ce modèle dans le temps. Un jeu comporte des phases :

> *« These phases are: Discovery (why people would even want to try out the experience), Onboarding (where users learn the rules and tools to play the game), Scaffolding (the regular journey of repeated actions towards a goal) and Endgame (how do you retain your veterans). »* (Chou, 2014)

D'après Chou, il est important de refaire l'analyse à chacune car la motivation change à chaque étape.

En conclusion de cette rapide présentation des profilages théoriques existants, j'aimerais insister sur leur aspect psychologique. Ces approches vont bien au-delà du partitionnement qui se base sur les données disponibles depuis le jeu. Anders Drachen présente d'ailleurs la méthode des cinq facteurs venus du domaine de la psychologie comme étant une méthode à utiliser dans l'analyse d'un jeu vidéo : elle se base sur cinq traits de la personnalité (Drachen et al., 2013, Yannakakis et al., 2013). De telles méthodes requièrent le travail d'une personne versée en psychologie. Dans le cas de Star Shaman, il s'agira de comprendre les caractéristiques des joueurs à partir de leurs sessions de jeu uniquement. Nous utiliserons donc le clustering pour établir des groupes de joueurs, le terme de profils portant finalement à confusion.

b) METHODES DE PARTITIONNEMENT DES JOUEURS

Dans la littérature de l'analyse de données de jeu vidéo, on trouve plusieurs méthodes selon les besoins des auteurs. Charles & Black (2004) et Fernandes & Levieux (2019) utilisent la régression logistique pour créer un système d'ajustement de la difficulté. Benmakrelouf et al., 2015 emploie une régression linéaire multiple afin de déterminer l'impact de variables sur l'engagement dans un serious game (voir glossaire en II.1). Ces deux types de régressions sont décrits par Thibault (201)3 qui en explique le fonctionnement. Bicalho (et al., 2019) et Thibault (2013) mentionne aussi les arbres de décisions pour répartir les joueurs dans les profils, notamment ceux de Richard Bartle. Mais, j'ai plutôt focalisé mes recherches sur le clustering de joueurs. Celui-ci peut se faire avec différentes méthodes.

Comme nous l'avons dit, il s'agit de classer les joueurs en groupes en fonctions des caractéristiques intrinsèques et inconnues au départ de leurs données. Deux familles de méthodes d'apprentissage non-

supervisé sont particulièrement employées dans la littérature pour l'analyse des groupes de joueurs. Il s'agit des analyses factorielles et du clustering en tant que tel. Les premières permettent de déterminer des facteurs d'un phénomène et contiennent notamment l'Analyse en Composantes Principales (ACP) utilisée dans le jeu vidéo. Mais les principales méthodes croisées dans les articles sont celles de clustering proprement dit, surtout la méthode des k-means, et l'Archetypal analysis. Nous allons décrire ces trois méthodes afin de présenter leurs fonctionnements.

L'ACP est une méthode de regroupement des variables en composantes. En effet, les données de jeu vidéo sont massives et contiennent de très nombreuses variables. Il faut donc une méthode qui regroupe celles-ci en quelques composantes à partir desquelles l'analyse est faite. Les composantes regroupent les variables similaires (Rodrigues & Brancher, 2018), c'est-à-dire dont la distance euclidienne est faible. Celle-ci est calculée à partir d'un regroupement hiérarchique : il s'agit d'une méthode de clustering agrégeant les individus proches. Chaque individu forme une classe indépendante au départ du processus et, à chaque itération, les plus proches sont agrégés formant ainsi des classes de plus en plus grandes et réduisant le modèle à quelques branches. Pour l'ACP, ce ne sont pas les individus mais les variables qui sont ainsi regroupées. D'après Rodrigues & Brancher (2018), le partitionnement des joueurs en a été fortement amélioré.

Toutefois, l'Analyse en composantes principales est parfois critiquée car ses résultats manquent de clarté. En effet, les composantes sont des entités abstraites qui ne sont pas interprétables. Anders Drachen et ses co-auteurs mentionnent une étude sur 70.000 joueurs de World of Warcraft. L'étude compare le clustering par l'ACP, la méthode des k-means, l'Archetypal Analysis et la factorisation par matrice non négative (Drachen et al., 2013). Les résultats de l'ACP semblent peu clairs aux auteurs et donc non accessibles à des non-spécialistes. Charles & Black (2004) vantent les mérites de l'ACP, une première façon d'explorer les données et de voir quelles variables sont corrélées entre elles. On peut ensuite chercher pourquoi certaines sont corrélées, et en tirer des conclusions sur le jeu :

> *« Then using statistical neural network approaches such as Principal Component Analysis or Factor Analysis we may explore the data so as to identify the correlational (or high order statistical) relationship between the variables. »* (Charles & Black, 2004)

La méthode des k-means est souvent privilégiée en matière de partitionnement, pour la clarté de ses résultats. Charles & Black (2004) la comparent aux analyses factorielles dont l'ACP. Contrairement à celle-ci, ce n'est pas la corrélation entre variables qui est intéressante mais la manière dont on peut créer des groupes de joueurs :

> *« Whereas a clustering method would group players together so that we can label these groups as novice, normal or advanced, on the basis of the complete data point. »* (Charles & Black, 2004)

La méthode permet de regrouper les individus autour de valeurs moyennes de chaque variable pour chaque groupe. Ces valeurs sont appelées centroïdes : elles ne représentent pas un individu en particulier mais une valeur moyenne sur l'ensemble des variables auxquelles on va rattacher successivement les individus les plus proches. Il faut donc déterminer d'avance le nombre de clusters que l'on souhaite. L'algorithme utilisé détermine alors un même nombre de centroïdes qui seront modifiés en fonction des résultats des itérations. Pour chaque individu, l'algorithme calcule le centroïde dont la distance est la plus faible, et maximise la distance au centroïdes des autres clusters. Ainsi, les k-means déterminent deux paramètres : l'homogénéité interne des clusters en minimisant la distance entre les individus, et la distance inter-cluster qui est maximisée (Thibault, 2013). Pour des explications détaillées sur les concepts

de distance, le lecteur peut se référer à Drakopoulos (et al., 2020). Le résultat se présente sous la forme d'un graphique en deux dimensions avec des groupes de points de couleurs différentes.

Wallner (et al., 2019) utilise la méthode des k-means pour définir les profils de 3000 joueurs de Destiny.

« [..] we extracted the total time spend playing (PvPand PvE combined) and measures of Twitter activity (numberof tweets, retweets, and replies as well as the total number of people a user is following) and indicators of popularity (number of followers, average number of retweets a tweet receives, average number of likes a tweet receives) in order to correlate the measures and to develop player profiles through k-means clustering » (Wallner et al., 2019)

Il mentionne d'ailleurs un inconvénient important des k-means : la méthode est sensible aux valeurs extrêmes, c'est-à-dire aux individus dont tout ou partie des caractéristiques sont trop éloignées de celles des autres joueurs. Nous en rencontrerons d'ailleurs dans notre analyse (voir Partie III). Les auteurs de l'article ont donc supprimé certains individus de leur jeu de données avant clusterisation. Sept groupes furent trouvés, qui départagent les joueurs tant sur les données de jeu que sur celles de leurs comptes Twitter.

Benmakrelouf (et al., 2015) compare k-means et régression dans leurs usages respectifs :

« The regression models showed that the number of access to the game, completed quests and advantages used contribute significantly to the scores and the gaming duration, while the clustering revealed three forms of players' participation: beginner, intermediate and advanced; » (Benmakrelouf et al., 2015)

Cela montre deux types de résultats que l'on souhaite obtenir et l'importance de choisir la bonne méthode. En effet, la régression a permis aux auteurs de trouver des facteurs du score et de la durée de jeu. Ils ont pu identifier quelles variables les impactaient. A l'inverse, la méthode des k-means sert à la classification, à savoir, regrouper les individus et analyser un phénomène. Lors de l'analyse (partie III), je ferai appel aussi bien à la régression qu'à la classification, et nous verrons la différence entre les questions qui se posaient au départ.

Pour terminer, les k-means peuvent être utilisés en combinaison avec d'autres méthodes. Les auteurs traitant de données spatio-temporelles mentionnent cette méthode comme une de leur panel. Dans Bauckhage (et al., 2014), les auteurs obtiennent des waypoints ou étapes des joueurs en appliquant cette méthode, et peuvent ensuite construire une carte des manières de naviguer dans tel niveau de jeu. Ils sont partis du fait que les étapes des joueurs dans un niveau représentent des densités et donc en quelque sorte des clusters. Leur but était de trouver un moyen d'afficher les étapes par lesquels passaient les joueurs en tenant compte de la temporalité, et donc de faire une carte asynchrone. C'est aussi ce genre de cartes qui sont réalisées par Aung (et al., 201 et les k-means ont servi aussi à identifier les points de passage des joueurs. Les auteurs ont d'abord identifié n lieux par joueurs puis la même chose pour chacun des joueurs, aboutissant à une représentation asynchrone des déplacements :

« That is, using k-means clustering to cluster locational information of players we extracted central waypoints and later created a waypoint graph encoding movement information between the automatically extracted waypoints. Note that the extracted waypoint graphs (especially for OWGs) are usually time or location-asymmetric. » (Aung et al., 2019)

On l'a vu, les k-means se basent sur des caractéristiques moyennes puisque le centroïde prend les valeurs moyennes des variables en compte. C'est tout l'inverse pour l'Archetypal Analysis qui cherche des individus extrêmes pour avoir des groupes aux caractéristiques bien distinctes. C'est Anders Drachen

et ses divers co-auteurs qui parlent de cette méthode employée dans plusieurs recherches. Elle utilise une matrice convexe et il en résulte des archétypes de joueurs. Les centroïds des clusters sont pris dans les points des parties extrêmes de la matrice (Drachen et al., 2013). Les auteurs précisent d'ailleurs que les souvenirs mémorables se font grâce aux expériences uniques et extraordinaires. En ce sens, il leur semble donc logique de chercher des valeurs extrêmes qui reflètent ces expériences.

« Searching for certain extremal elements in a set of data as it is done for AA accommodates human cognition, since memorable insights and experiences typically occur in form of extremes rather than as averages » (Drachen et al., 2013)

Il faut noter que les individus ne sont pas assignés à un cluster unique. L'Archetypal analysis calcule plutôt la relation de chaque point avec les vecteurs de base de la matrice. Les joueurs sont regroupés selon ces appartenances. A l'affichage des résultats sous forme de graphique, chaque joueur est attribué au cluster le plus proche.

La référence du cluster n'est pas un point abstrait comme pour les k-means, mais un individu extrême. Ceci rend les résultats très aisément interprétables (Drachen et al., 2014). Les profils se basant sur des individus et non sur des composantes abstraites ou des points abstraits, ils sont facilement lisibles.

Nous ne développerons pas plus cette méthode car nous ne la retenons pas. A la suite de Bicalho (et al., 2019), je souhaite rattacher chaque joueur à un cluster directement. J'utiliserai donc plutôt la méthode des k-means. En effet, je souhaite pouvoir interpréter les caractéristiques des clusters. Je cherche donc à créer des groupes en me basant sur les variables directement, sans passer par des composantes.

Pour finir cette présentation de la littérature professionnelle, j'aimerais évoquer quelques réflexions à propos de Star Shaman. En effet, à la lecture des recherches et ouvrages, j'ai réalisé quelques limites de l'analyse des joueurs. Tout d'abord, je n'ai pas trouvé de références sur le clustering ou le profilage de jeux en réalité virtuelle. Or, le type de jeu et son univers engendrent des données spécifiques, comme nous l'avons vu dans toute cette deuxième partie. Je me suis donc demandé quelles étaient les particularités de Star Shaman. Est-ce l'immersion, le mouvement ? A la suite de Chou (2015), faut-il considérer la sensation ? Je me suis rendu compte que nos données ne monitoraient pas cela. En effet, nous ne suivons pas combien de fois le joueur fait le geste nécessaire à charger un sort dans sa baguette. L'opération demande-t-elle plusieurs essais ? Est-ce l'apanage des débuts de jeu ou cela perdure-t-il dans le temps ? A défaut de suivre la gestuelle des joueurs, j'ai utilisé plusieurs variables pour tenter d'approcher la mesure de l'immersion. Je reviendrai là-dessus en partie III.3.

Ensuite, j'ai remarqué que beaucoup d'auteurs se basaient sur les blessures du joueur et ses tirs combinés, que ce soit en nombres ou en termes de qualité. Nous ne monitorons pas le nombre de sorts lancés, ni leurs types. Nos données seront donc fort différentes et uniquement basées sur les données de jeu. Contrairement à Wallner (et al., 2019) ou à Albuquerque & Fialho (2015) par exemple, nous n'avons pas de données extérieures au jeu, ni de questionnaire dont les réponses seraient combinables.

III. TROISIEME PARTIE : DEVELOPPEMENT D'UN CADRE D'ANALYSE DES JOUEURS

Dans cette dernière partie, nous allons détailler les étapes de l'analyse des joueurs de Star Shaman. Comme nous l'avions esquissé en partie I.3, l'objectif de la première analyse est d'établir une classification des joueurs en fonction de leurs caractéristiques de jeu. Il s'agit de voir si des groupes se distinguent, puis d'observer quelles sont leurs caractéristiques. Le studio Ikimasho pourrait ainsi savoir comment améliorer le jeu, par l'introduction de nouveaux éléments/systèmes solaires ou le raffinement de paramètres par exemple. Ces objectifs en tête, j'ai utilisé la méthode de clustering des k-means pour créer des groupes. Nous allons dans cette partie détailler la méthode et ses résultats.

Toutefois, nous verrons que les résultats ne permettent pas de classifier les joueurs. J'ai donc poursuivi un second objectif : déterminer les facteurs impactant le fait de quitter le jeu ou de continuer à jouer. A l'aide d'une variable booléenne déterminant si le joueur avait continué Star Shaman après l'extraction des données au 26 mars, j'ai utilisé deux méthodes : la régression logistique et la forêt aléatoire. Cela permet d'identifier les variables importantes pour prédire un départ ou la continuation du jeu. Cela est donc aussi un bon moyen de comprendre les éléments importants du jeu dans le comportement des joueurs.

Cette troisième et dernière partie se compose donc de la description de tout le processus d'analyse, depuis la récupération des données jusqu'aux résultats des modèles. La première section détaille la réflexion autour des données, notamment le choix des données suivies. La deuxième explique le pipeline mis en place pour les générer et les récupérer. La troisième section présente le clustering et les résultats. Enfin, la quatrième section présente la recherche de facteurs impactant le départ des joueurs et ses résultats.

1. REFLEXION SUR L'ORGANISATION DES DONNEES

Dans un premier temps, j'aimerais revenir sur le travail de réflexion à propos des données. En effet, la première étape est de se demander quelles données doivent être suivies dans le jeu et pourquoi. En effet, nous l'avons vu en partie 2, il y a des données générales telles que le nombre total de joueurs, le nombre par étape du funnel ou encore le nombre de joueurs actifs quotidiens. Nous verrons en section 2 que cela était facilement calculable sur la plate-forme DeltaDNA qui réceptionne les données brutes. Toutefois, les données spécifiques à Star Shaman étaient à imaginer et à créer.

a) COMMENT CHOISIR LES DONNEES A SUIVRE

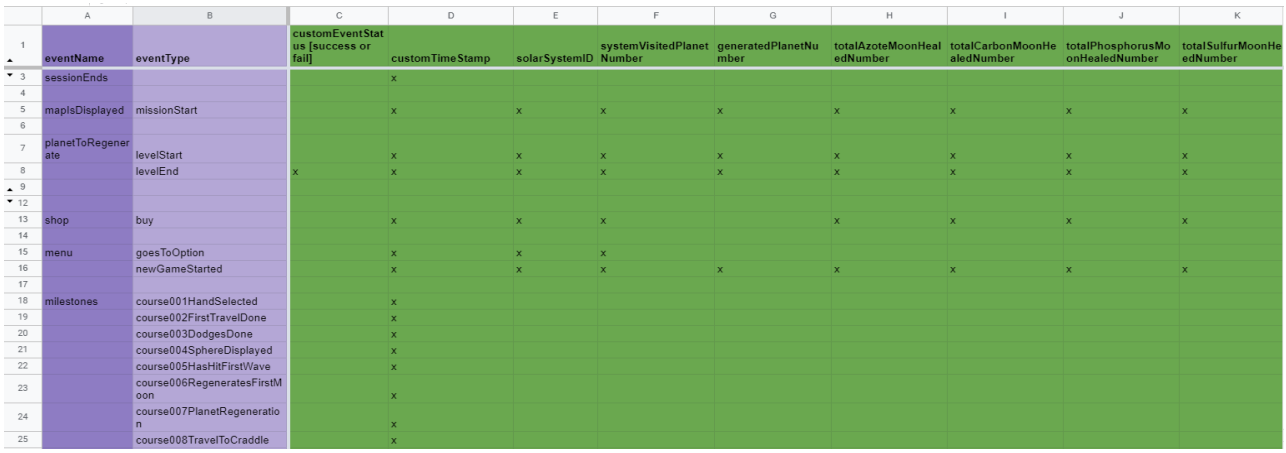
Un jeu vidéo génère de très nombreuses données : les suivre toutes requiert une architecture et une base de données très puissantes. D'après Anders Drachen (Drachen et al., 2013, chapitre 6), le type de revenu du studio (jeux en freemium, achat unique) détermine la temporalité de l'envoi des données et donc de l'analyse. Les jeux sociaux demandent une analyse du réseau (voir partie II) et donc des données streamées. Dans le cas de Star Shaman, afin de préserver la performance du jeu, j'ai dû choisir quelles données devaient être enregistrées, ce qui m'a amenée à rejeter l'idée du stream. Sur les conseils de Monsieur Gauthier Dine, data analyst et data architect du studio GamePoint (voir section I.3.c), j'ai privilégié un suivi simple des joueurs. A ce moment de la réflexion, un clustering n'était pas encore envisagé. J'ai adopté son système en boucle, basé sur le mouvement des joueurs dans le jeu. Le principe repose sur la question suivante : quels sont les types d'événements communs à l'ensemble du jeu ? Les suivre permet de réduire le nombre d'événements enregistrés tout en gardant une vue de l'ensemble du jeu.

Nous avons remarqué que l'événement central du jeu a lieu lorsque le joueur affiche la carte du système solaire. A ce moment, il a le choix entre aller régénérer une planète, se rendre sur une planète-événement ou dans un magasin. À tout moment, le joueur peut aussi aller dans le menu des options. Il y a donc deux boucles : aller dans le menu ou sur une planète (avec trois sous-boucles selon la planète). Dans chaque boucle, nous avons monitoré les sous-boucles à l'aide de types d'événements (voir l'ensemble des événements et types dans les sous-parties suivantes). Par exemple, l'événement « aller dans le menu » avait deux types différents : « aller dans les paramètres » ou « re-démarrer le jeu à zéro ». Enfin, une troisième boucle consiste à suivre les visites au magasin : cette planète permet d'acheter des protections, des sorts ou des mises à jour de l'un ou l'autre. Nous l'avons définie comme une boucle à part car l'action faite n'est pas la régénération mais l'achat ; nous souhaitions monitorer ce que le joueur acquiert. Comme nous l'avons déjà dit en I.4.d, nous n'avons finalement pas pu avoir de données concernant ce que le joueur achète. Aussi cette boucle est-elle restée simplement un événement retraçant si le joueur achète ou non.

Outre ces boucles, j'ai créé un *funnel* (voir définition en partie II) pour suivre les joueurs dans plusieurs étapes de Star Shaman : les étapes du tutoriel, la première visite au magasin, le premier système solaire, le deuxième, le troisième. Ces événements servent au suivi du nombre de joueurs. On peut ainsi voir rapidement si une étape du jeu perd beaucoup de joueurs, ce qui montre une trop grande difficulté ou un bug. Pour terminer, d'autres données ont été ajoutées aux boucles et au *funnel* : l'événement de fin de session de jeu et celui lorsque le joueur quitte en sauvegardant. Le premier montre lorsque le joueur quitte le jeu sans passer par la sauvegarde : il utilise le bouton spécifique de sa manette pour sortir du jeu. De fait, il est très peu enregistré car l'événement n'a pas le temps d'être envoyé au moment où le joueur quitte le jeu. Et si la personne se reconnecte plus de 32 jours après, la base de données Delta DNA n'accepte pas l'événement envoyé lors du redémarrage. C'est pourquoi nous avons enregistré lorsque les joueurs utilisent le bouton « sauvegarder et quitter » : le temps que le jeu s'enregistre, l'événement est envoyé.

Une fois les données à suivre choisies et leurs événements correspondant déterminés, j'ai créé une *Tracking Bible*. Ce document sert à résumer les événements suivis et les données que l'on reçoit pour chacun. En effet, comme nous le verrons plus bas, chaque événement génère l'envoi de paramètres. Le travail sur les boucles et la bible de données permet de déterminer les paramètres, c'est-à-dire les données communes aux événements. Cela permet aussi de réduire le nombre de paramètres. En effet, la facturation des services de bases de données repose généralement, quel que soit le fournisseur, sur le nombre d'événements reçus et souvent aussi le nombre total de paramètres.

A titre d'exemple de paramètre commun, j'aimerais citer le *timestamp*. Il s'agit de l'horodatage : le jour et l'heure à laquelle est reçu l'événement. Ceci est commun à tous les événements et permet de les ordonner dans le temps. C'est donc un paramètre à inclure par défaut. La bible de données permet de lister en ligne les événements, et en colonne les paramètres associés.



	A	B	C	D	E	F	G	H	I	J	K
1	eventName	eventType	customEventStatus [success or fail]	customTimeStamp	solarSystemID	systemVisitedPlanetNumber	generatedPlanetNumber	totalAzoteMoonHealedNumber	totalCarbonMoonHealedNumber	totalPhosphorusMoonHealedNumber	totalSulfurMoonHealedNumber
3	sessionEnds			x							
4											
5	mapIsDisplayed	missionStart		x	x	x	x	x	x	x	x
6											
7	planetToRegenerate	levelStart		x	x	x	x	x	x	x	x
8		levelEnd	x	x	x	x	x	x	x	x	x
9											
12											
13	shop	buy		x	x	x		x	x	x	x
14											
15	menu	goesToOption		x	x	x					
16		newGameStarted		x	x	x	x	x	x	x	x
17											
18	milestones	course001HandSelected		x							
19		course002FirstTravelDone		x							
20		course003DodgesDone		x							
21		course004SphereDisplayed		x							
22		course005HasHitFirstWave		x							
23		course006RegeneratesFirstMoon		x							
24		course007PlanetRegeneration		x							
25		course008TravelToCraddle		x							

Sur l'extrait ci-dessous, on observe que le nombre de cœurs (*hearts number*) du joueur, c'est-à-dire de vies, est commun à l'ensemble des boucles et de l'événement central (*map is displayed*).

eventName	eventType	totalWaterMoonHealedNumber	systemNumberOfHealedMoons	generatedMoonNumberInSolarSystem	heartsNumber	~~manaMaxNumber~~	currencyAmou
sessionEnds							
mapIsDisplayed	missionStart	x	x		x	~~x~~	x
planetToRegenerate	levelStart	x	x	x	x	~~x~~	x
	levelEnd	x	x	x	x	~~x~~	x
shop	buy	x	x	x	x	~~x~~	x
menu	goesToOption		x	x	x	~~x~~	x
	newGameStarted	x	x	x	x	~~x~~	x
milestones	course001HandSelected						
	course002FirstTravelDone						
	course003DodgesDone						
	course004SphereDisplayed						
	course005HasHitFirstWave						
	course006RegeneratesFirstMoon						
	course007PlanetRegeneration						
	course008TravelToCraddle						
	course009TotemsRegenerated						

Extrait des premiers événements et derniers paramètres de la bible de données

Pour conclure sur le choix des données, j'aimerais m'attarder sur l'importance de connaître très précisément les questions auxquelles l'on souhaite répondre à l'avance et les méthodes potentiellement employables à l'avenir. En effet, nous avons dû rajouter des données par la suite car les dirigeants du studio souhaitaient savoir si les joueurs quittaient le jeu d'une manière ou d'une autre, ou parce que le clustering demandait des données supplémentaires. Dans le premier cas, il s'agit d'une évolution normale des questionnements. Dans le second cas, cela aurait pu être prévu en amont. J'ai notamment rajouté le numéro de version du jeu dans les paramètres, pour éventuellement comparer des clustering entre les joueurs du début et ceux de la mise à jour de janvier. Enfin, je me suis aperçu du manque d'un événement lors du clustering. Afin de capter la spécificité de la réalité virtuelle dans le clustering, j'aurais aimé vérifier que les gestes pour charger un sort de la sphère à la baguette étaient réussis et j'aurais éventuellement voulu compter combien de fois le joueur avait dû le répéter avant d'y arriver. Nous avions un booléen *IsGestureComplete* qui aurait permis de suivre cela. Cependant, dans mon attention à ne pas monitorer trop de détails, je n'ai pas vu l'intérêt d'aller à ce niveau de détail lors de la mise en place, d'autant plus que nous ne pouvions pas distinguer les sorts entre eux. Cela m'a manqué par la suite.

Dans tous les cas, introduire le suivi de nouveaux événements implique d'attendre la mise à jour suivante, et surtout de développer et tester entièrement le pipeline avant. Afin de faire comprendre l'évolution de la mise en place du pipeline de données, nous allons rapidement présenter l'évolution du projet de traitement des données de jeu et les décisions successives que nous avons prises.

b) LES ETAPES DE DEVELOPPEMENT DU TRAITEMENT DES DONNEES

Lorsque le traitement des données de jeu a été décidé, en mai 2020, Star Shaman en était à sa version alpha. La mécanique de jeu était formalisée, les illustrations et objets créés pour la plupart. Il manquait le paramétrage, l'équilibrage, le codage de certaines parties de la mécanique de jeu. Ainsi, j'ai commencé à travailler sur les données alors que le jeu était assez avancé, mais la construction de son architecture n'était pas encore terminée, d'où de nombreux changements dans les variables utilisables et certaines décisions prises quant à l'implémentation du système data dans l'architecture.

Par ailleurs, Star Shaman fut lancé le 21 octobre 2020 et plusieurs mises à jour ont suivi, ajoutant ou retranchant des phases de tutoriel et des éléments de jeu. J'ai donc dû m'adapter en ajoutant ou en enlevant des événements du pipeline. De plus, j'ai dû corriger une erreur de pipeline.

Afin de donner un aperçu de l'évolution du projet, voici le résumé du planning de la mise en place du suivi de données jusqu'à mon départ le 26 janvier 2021 :

Mai 2020	Premières réflexions sur les données à suivre
Mai - Juin 2020	Première version de la bible de données
Juillet - Août 2020	Première version du pipeline technique
Fin Août - 20 Octobre 2020	Deuxième version du pipeline et tests
21 Octobre 2020	LANCEMENT de Star Shaman
22 Octobre - 6 Novembre 2020	Premiers retours sur l'incomplétude des données et
7 Novembre 2020	1ère mise à jour
8 Novembre -14 Décembre 2020	Implémentation de la version 2 de la bible de données
15 Décembre 2020	2ème mise à jour
16 Décembre 2020 – 26 Janvier 2021	Implémentation de la version 3 de la bible de données
29 Janvier 2021	3ème mise à jour

Suite au lancement, je me suis aperçu qu'une erreur se produisait quasiment sur l'ensemble des joueurs (ceux ayant acheté auprès de la boutique Oculus) : leurs données étaient envoyées vers

l'environnement de développement et non vers celui de production de la base de données Delta DNA ! Je n'avais en effet pas vérifié la dernière version du jeu mise dans les boutiques et l'une d'entre elles contenait la mauvaise option d'envoi des données (voir pipeline en section 2 ci-dessous).

Les données de joueurs se confondaient donc avec nos tests intensifs suite aux premiers retours des joueurs et en prévision de la mise à jour de novembre. Étant donné que tester le jeu signifiait réinstaller presque à chaque fois la version sur le casque, chaque membre de l'équipe générait plusieurs identifiants. Il n'est pas possible de trouver cette information en allumant le jeu depuis un casque autonome. Aussi, ne fut-il pas possible de différencier les joueurs de nos tests. Nous avons corrigé cela dans la mise à jour de novembre ; et encore a-t-il fallu attendre quelques joueurs afin que tous les joueurs mettent Star Shaman à jour et donc voir leurs données arriver dans le bon environnement de la base de données. En raison de cet incident, j'ai décidé de n'utiliser que les événements à partir du 7 novembre 2020 dans mes analyses.

Les données supplémentaires que je voulais rajouter/ modifier lors de la mise à jour du 7 novembre furent repoussées, par manque de temps de test, à la mise à jour de décembre. Le premier événement du *funnel* – ou *milestone* dans ma bible de données – se déclenchait de manière intempestive dès que le joueur mettait la main dans la sphère au début du tutoriel. Or, cette étape avait pour objectif de choisir sa main dominante et fut difficile à comprendre pour les joueurs : l'idée était de savoir si le joueur était droitier ou gaucher pour adapter la main recevant la baguette de shaman ainsi que les côtés de la sphère (celui de la main dominante recevant toujours les sorts). Les joueurs passaient donc beaucoup de temps à mettre et enlever la main de la sphère avant de comprendre quoi faire. J'avais par conséquent prévu de changer l'événement-jalon correspondant à la section de la main dominante pour le déclencher uniquement lorsque celle-ci serait définitivement validée et enregistrée. Or, ce fut repoussé à la mise à jour de décembre lorsque le système fut changé pour un panneau de texte avec un bouton sélectionnable par main. J'ai donc décidé s'il fallait garder l'événement ou non : ce fut le cas en l'adaptant au nouveau système. J'ai ainsi obtenu des données répétitives pour cet événement, m'obligeant à documenter sa signification afin que l'on se rappelle qu'il ne recouvre pas exactement le même système avant et après le 15 décembre 2020.

De la même manière, le numéro de système solaire nous a posé un problème et la correction fut repoussée à décembre. En effet, je savais que lors de l'envoi de l'événement marquant le passage d'un système solaire à un autre, le numéro de système solaire serait augmenté de 1 très rapidement. En pratique, le joueur passant du système 1 à 2 envoie un événement au moment où le numéro a déjà été changé pour 2 par le système de jeu. Il faut donc documenter que l'événement de type « system finished » renvoie, non le numéro du système quitté, mais celui du nouveau. Or, cette règle ne s'appliquait pas à tous les casques. Certains, plus lents à modifier le numéro de système solaire, renvoyaient l'événement avec le numéro du système quitté. J'ai dû me greffer à un autre moment du jeu pour recevoir l'événement plus tôt et donc le numéro de système quitté par le joueur.

Une autre donnée repoussée à la mise à jour de décembre fut l'événement lorsque le joueur sauvegarde et quitte le jeu. Il s'agit d'un bouton dans les options que nous souhaitions monitorer. Comme il s'agissait d'une nouveauté, cela demandait beaucoup de tests pour être sûre de la réception de l'événement dans la base de données. En effet, inclure une nouveauté dans le pipeline (voir détail du pipeline en 2^{e} section ci-dessous) inclut de:

- l'intégrer dans l'architecture du projet de jeu localement sur ma machine,
- tester l'envoi vers la base de données depuis un casque autonome et un casque relié à l'ordinateur,
- vérifier le format de la réception et la complétude des données
- publier le changement dans l'architecture commune du projet
- et enfin de re-tester.

Cela peut prendre facilement trois jours. J'ai dû également vérifier que les événements/éléments du jeu, auxquels mes événements data étaient reliés, étaient bien conservés et actifs. En effet, nous avons retravaillé le tutoriel pour les deux premières mises à jour, et certains événements data ne se déclenchaient plus. À chaque mise à jour, je devais donc faire une série de tests globaux pour vérifier qu'un événement n'avait pas été désactivé.

Pour terminer, nous allons aussi aborder brièvement ce qui n'a pas pu être suivi, et ce, pour diverses raisons. La première est que certains événements ont disparu. C'est notamment le cas du fait que le joueur attrape pour la première fois son bouclier; cela faisait partie du tutoriel initial. Je n'ai pas pu garder cet événement car aucun élément dans le jeu ne permettait de l'y rattacher.

Par ailleurs, d'autres événements/éléments du jeu ne sont pas monitorés parce que rien ne permettait de les identifier. Je voulais notamment faire une boucle sur les planètes-événements en identifiant le nom du personnage associé à la planète. Cela m'aurait permis de voir qui le joueur avait rencontré et combien de fois. Or, le nom de la planète (chaque élément du jeu a un identifiant) n'était pas une variable dans l'architecture du jeu. Comme Star Shaman incluait une génération aléatoire du nombre de planètes selon leurs types, ces informations étaient prises dans un fichier séparé, hors du projet, ce qui ne permettait pas de récupérer les identifiants. Aussi ai-je adapté le paramètre « type » de l'événement de fin de niveau. Ce dernier se déclenchait dès que le joueur quittait une planète, quelle qu'elle soit. J'ai donc adapté le type de l'événement (voir ci-dessous la liste des événements et paramètres) pour qu'il mentionne le type de planète quitté. Je pouvais ainsi savoir combien de fois les joueurs avaient été sur la planète-magasin ou sur une planète-événements.

Enfin, il me paraissait nécessaire de monitorer quels sorts étaient achetés/mis à jour dans le magasin et lesquels étaient ensuite utilisés. Nous aurions ainsi pu comparer, voir quels étaient les sorts préférés (achetés et utilisés), les sorts délaissés (achetés mais non utilisés) etc. Or, les sorts n'avaient pas d'identifiant à utiliser comme variable dans un événement. On ne pouvait donc les distinguer et je ne pus suivre cela.

On le voit, la mise en place du monitoring des données de jeu a donné lieu à de nombreux changements dans les événements et paramètres suivis. La section suivante présente l'ensemble des données à disposition, ainsi que les listes des événements et des paramètres, pour une meilleure compréhension de ce qui était à disposition pour les analyses.

c) DICTIONNAIRE DES DONNEES

Ce dictionnaire a pour but de présenter les variables à disposition pour l'analyse des joueurs. Certaines sont brutes et directement issues des informations envoyées lors du jeu. D'autres sont conçues par l'agrégation de données brutes ou par mes calculs. Dans les deux cas, certaines furent automatiquement mises à disposition par la plate-forme DeltaDNA et sont signalées comme telles. Il en existe bien plus mais nous ne mentionnerons ici que celles qui sont nécessaires à l'analyse ou que nous avons considérées lors de nos choix. D'autres sont plus techniques et servent en cas d'administration d'une base de données vers laquelle les données seraient renvoyées.

Nous prenons le parti de les appeler « paramètres » dans les tableaux car il s'agit du terme employé pour la création du pipeline du jeu vers la plate-forme, et c'est également le terme utilisé sur la plate-forme. En effet, le terme variable dans ce contexte renvoie plutôt aux variables du code du jeu qui alimentent ces paramètres (voir section 2 ci-dessous). Nous reparlerons de variables mais uniquement au sens statistique dans les analyses, pour désigner les colonnes des jeux de données extraites pour ces analyses.

En résumé, DeltaDNA offre une base de données avec une table *Events* qui reçoit les données brutes et sept autres tables qui proposent des *metrics* ou données agrégées. Chaque ligne de la table *Events* correspond à un événement et les colonnes correspondent aux paramètres. Les événements sont stockés dans l'ordre d'arrivée sur le serveur et non par identifiant de joueurs. Les autres tables sont ordonnées par joueurs ou par sessions. Nous présentons d'abord les paramètres car ils sont communs à plusieurs événements et ils permettent de comprendre comment un événement est construit. En effet, ce dernier est une combinaison de paramètres. Il ne se déclenche pas au même moment selon les valeurs des paramètres. Par exemple, l'action d'aller dans le menu se divise en deux sous-événements que sont « aller dans les options » et « réinitialiser le jeu ». Ainsi comprendre les paramètres permet de saisir les événements suivis.

Tableau des paramètres issus des actions du jeu

Il s'agit des données brutes reçues par Delta DNA et stockées dans la table *Events* de la base. Les paramètres ci-dessous sont les colonnes de la table. Certaines données sont automatiquement envoyées par le script de la plate-forme présent dans le jeu, et d'autres sont déduites automatiquement. Ces dernières sont suivies d'un * dans le tableau.

PARAMETRES	NOM FRANÇAIS	DESCRIPTION	EXEMPLE	TYPE
IDENTIFIER UN EVENEMENT PRECIS				
eventName	NOM	Nom de l'événement réalisé par le joueur	*sessionEnds*	Qualitatif
eventType	TYPE	Sous-événement, déclinaison de l'événement	*level starts* par rapport à *Niveau*	Qualitatif
Status	STATUS	Fin de niveau uniquement : réussite ou échec	*success/fail*	Qualitatif
PARAMETRES DE TEMPS				
customTimeStamp	HORODATAGE LOCAL	Jour et heure dans le fuseau local du joueur	"2020-11-30 13:11:32.202"	Qualitatif ordinal
eventTimeStamp *	HORODATAGE DE L'ÉVÉNEMENT	En horaire UTC	"2021-01-05 23:40:11.288"	Qualitatif ordinal
collectInsertedTimestamp *	HORODATAGE DE RÉCEPTION	Jour et heure lors de la réception de l'événement sur le server de Delta DNA	2021-02-16 11:34:00.642	Qualitatif ordinal
eventDate *	DATE DE L'ÉVÉNEMENT	Date à laquelle l'événement se produit	2021-01-25	Qualitatif ordinal
VOIR LA PROGRESSION DANS LE JEU				
solarSystemID	NUMÉRO DE SYSTÈME SOLAIRE	Système dans lequel se trouve le joueur à l'instant t	1 à 3	Qualitatif

systemVisitedPlanetNumber	NOMBRE DE PLANÈTES VISITÉES DANS LE SYSTÈME	Calculé depuis le début du système solaire courant	4	Quantitatif discret
generatedPlanetNumber	NOMBRE DE PLANÈTES EXISTANTES DANS LE SYSTÈME SOLAIRE	Système solaire courant du joueur	15	Quantitatif discret
systemNumberOfHealedMoons	NOMBRE DE LUNES REGENEREES DANS LE SYSTÈME	Calculé depuis le début du système courant du joueur	4	Quantitatif discret
generatedMoonNumberInSolarSystem	NOMBRE DE LUNES GÉNÉRÉES DANS LE SYSTÈME	Nombre de lunes présentes dans le système courant	30	Quantitatif discret
DETAILS DE CE QUE LE JOUEUR A RÉGÉNÉRÉ				
totalAzoteMoonHealedNumber	NOMBRE DE LUNES AZOTE RÉGÉNÉRÉES	Nombre calculé depuis le début du jeu.	7	Quantitatif discret
totalCarbonMoonHealedNumber	NOMBRE DE LUNES CARBONE RÉGÉNÉRÉES	Nombre calculé depuis le début du jeu.	8	Quantitatif discret
totalPhosphorusMoonHealedNumber	NOMBRE DE LUNES PHOSPHORE RÉGÉNÉRÉES	Nombre calculé depuis le début du jeu.	7	Quantitatif discret
totalSulfurMoonHealedNumber	NOMBRE DE LUNES SOUFRE RÉGÉNÉRÉES	Nombre calculé depuis le début du jeu.	8	Quantitatif discret
totalWaterMoonHealedNumber	NOMBRE DE LUNES EAU RÉGÉNÉRÉES	Nombre calculé depuis le début du jeu.	7	Quantitatif discret
PARAMÈTRES DU JOUEUR				
User ID *	IDENTIFIANT DU JOUEUR	Identifiant unique, sauf si le joueur réinstalle le jeu et réinitialise l'envoi des données.	4e15b4ee-17b3-4969-9aee-1a4e1e6861ad	Qualitatif
heartsNumber	NOMBRE DE COEURS	Nombre de points de vie du joueur à l'instant t (en forme de cœurs)	1 à 5	Quantitatif discret
currencyAmount	MONTANT DE MONNAIE	Montant virtuel possédée par le joueur	522	Quantitatif discret
userScore	SCORE DU JOUEUR	Gagné à l'instant t : n'est pas cumulatif.	14570	Quantitatif discret
difficultyLevel	NIVEAU DE DIFFICULTÉ	Il augmente selon le nombre de planètes régénérées : on passe au niveau 2 après 4 planètes.	1 à 6	Qualitatif ordinal
scoreMultiplier	MULTIPLICATEUR DE SCORE	% donnant la part du score donné en	1 à 5	Qualitatif ordinal

		monnaie au joueur à la fin d'un niveau		
handUsed	MAIN DOMINANTE	Gaucher ou droitier ?	*Right hand*	Qualitatif
gameLanguage	LANGUE DU JEU	Codé en chiffre : Français, anglais, allemand, japonais, italien, espagnol /Espagnol d'Amérique du Sud ou chinois simplifié	3	Qualitatif
userCountry *	PAYS	Pays du joueur	FR	Qualitatif
gaUserStartDate*	DATE DU PREMIER JEU	Date à laquelle le joueur allume le jeu pour la première fois.	2021-01-25	Qualitatif ordinal
userLanguage *	LANGUE DU JOUEUR	Langue du pays	En	Qualitatif
PARAMÈTRES DE SESSION				
Session ID *	IDENTIFIANT DE SESSION	Numéro unique par session de jeu	2905a7fd-6193-467d-a74b-d850e11e06c4	Qualitatif
msSinceLastEvent *	MILLISECONDES DEPUIS LE DERNIER ÉVÉNEMENT	Nombre de millisecondes depuis le dernier événement du joueur	11470	Quantitatif
PARAMÈTRES TECHNIQUES				
DeviceName *	NOM DE L'APPAREIL	Nom de l'ordinateur du joueur ou « Oculus Quest » pour les casques autonomes	« Oculus Quest »	Qualitatif
DeviceType *	TYPE D'APPAREIL	PC ou TABLET pour les casques autonomes	TABLET	Qualitatif
Platform *	PLATE-FORME	PC ou ANDROID pour les casques autonomes	PC_CLIENTS	Qualitatif
sdkVersion *	VERSION DU MOTEUR DE JEU	Jeu développé dans le moteur Unity : version d'Unity	`"sdkVersion": "Unity SDK v4.13.1"`	Qualitatif
appVersion	VERSION DU JEU	Version du jeu joué : permet d'identifier ceux qui jouent aux mises à jour. Introduit dans celle du 29/01/2021	`11.2.0`	Qualitatif ordinal

eventID *	IDENTIFIANT DE L'ÉVÉNEMENT	Identifiant unique de l'événement attribué à l'arrivée sur le serveur de DeltaDNA	2828067667598934018	Qualitatif
eventUUID *	IDENTIFIANT DE L'ÉVÉNEMENT	Identifiant plus long de l'événement	1aca2877-3ffa-418b-876e-fe5b6ece9a70	Qualitatif
clientVersion *	VERSION CLIENT	Numéro de build ? 6.23 au 4/09/2020	7.4.0	Qualitatif ordinal
hardwareVersion *	VERSION DU HARDWARE	Nom de l'appareil : doublonne deviceName	Oculus Quest	Qualitatif
operatingSystem *	SYSTÈME d'EXPLOITATION	Doublonne DeviceType	ANDROID	Qualitatif
operatingSystemVersion *	VERSION DU SYSTÈME D'EXPLOITATION		7.1.1	Qualitatif
timezoneOffset *	FUSEAU HORAIRE	Code du fuseau horaire du joueur	+0800	Qualitatif
manufacturer *	FABRIQUANT	Uniquement Oculus : fabricant de l'appareil	Oculus	Qualitatif

Tableau des données agrégées et des données de niveau calculées.

Les données agrégées sont issues de la table user_metrics générée automatiquement par DeltaDNA et qui contient des données par joueur. Les autres ont été calculées en fonction des données brutes.

La release est calculée en fonction du premier horodatage et du dernier :

- Si le joueur n'a joué qu'entre le 7 novembre 2020 13h UTC et le 15 décembre 2020 18h59 UTC, la personne est considérée comme un joueur de la mise à jour de novembre : « November Only »
- Si le joueur a commencé après le 7 novembre 2020 13h UTC et fini après le 15 décembre 2020 18h59 UTC, il est considéré comme un joueur des deux mises à jour : « Both »
- Si le joueur a commencé après le 15 décembre 2020 18h59 UTC, il est considéré comme un joueur de décembre : « December Only »

Ce paramètre est imprécis. Le paramètre *appVersion* a été introduit pour la mise à jour du 29 janvier afin de récupérer le numéro de version du jeu et donc de savoir précisément qui joue, dans quelle version.

*Les noms suivis d'une étoile sont générés automatiquement par la plate-forme Delta DNA

PARAMÈTRES	NOM FRANÇAIS	DESCRIPTION	EXEMPLE	TYPE
PARAMÈTRES DE GAMEPLAY DU JOUEUR				
FIELDUSERSCOREMIN *	SCORE MINIMUM	Score minimum atteint par le joueur toutes sessions confondues	0	Quantitatif discret

FIELDUSERSCOREMAX *	SCORE MAXIMUM	Score maximum atteint par le joueur toutes sessions confondues	75000	Quantitatif discret
AVERAGESCORE	SCORE MOYEN	Score moyen atteint par le joueur toutes sessions confondues	15000	Quantitatif discret
PARAMETRES DE TEMPS DU JOUEUR				
USER_LAST_SEEN_DATE *	DERNIÈRE FOIS QUE JOUEUR A ÉTÉ VU	Date à laquelle le joueur a joué pour la dernière fois	2021-01-22	Qualitatif ordinal
USER_FIRST_SEEN_DATE *	PREMIÈRE FOIS QUE JOUEUR A ÉTÉ VU	Date à laquelle le joueur a joué pour la première fois fois	2020-10-22	Qualitatif ordinal
TOTALDAYSPLAYED *	NOMBRE TOTAL DE JOURS JOUÉS	Nombre de jours où le joueur s'est connecté toutes sessions confondues	7	Quantitatif discret
TOTALSESSIONSPLAYED *	NOMBRE TOTAL DE SESSIONS JOUÉES	Nombre de sessions total depuis le premier jour	8	Quantitatif discret
TOTALTIMEPLAYEDMS *	TEMPS TOTAL JOUES EN MILLISECONDES	Depuis le premier jour	14000	Quantitatif discret
FIRSTGAMESESSIONEVENTTIMESTAMP *	HORODATAGE DU PREMIER ÉVÉNEMENT	Jour et heure de la première action enregistrée : plus précis que la première date	2020-11-07-14 :00 :00.000	Qualitatif ordinal
LASTGAMESESSIONEVENTTIMESTAMP *	HORODATAGE DU DERNIER ÉVÉNEMENT	Jour et heure de la dernière action enregistrée : plus précis que la dernière date	2020-11-23-14 :00 :00.000	Qualitatif ordinal
RELEASE	MISE À JOUR	Nom de la mise à jour à laquelle le joueur joue : définie en fonction des horodatages des actions et non sur la base d'un numéro de version donc imprécis	«Both releases »	Qualitatif
PARAMÈTRES DE NIVEAUX DU JOUEUR				
MAX(NBLEVELSYSTEM1)	NOMBRE MAXIMUM DE NIVEAU POUR LE SYSTÈME 1	Nombre maximum de niveaux réussis dans le système 1. Le joueur peut faire plusieurs systèmes	15	Quantitatif discret

		1 car en mourant, il retourne au début du jeu. Donc je prends ici le nombre maximum.		
MAX(NBLEVELSYSTEM2)	NOMBRE MAXIMUM DE NIVEAU POUR LE SYSTÈME 2	Nombre maximum de niveaux réussis dans le système. Le joueur peut faire plusieurs systèmes 2 car en mourant, il retourne au début du jeu. Donc je prends ici le nombre maximum.	15	Quantitatif discret
MAX(NBLEVELSYSTEM3)	NOMBRE MAXIMUM DE NIVEAU POUR LE SYSTÈME 3	Nombre maximum de niveaux réussis dans le système 3. Le joueur peut faire plusieurs systèmes 3 car en mourant, il retourne au début du jeu. Donc je prends ici le nombre maximum.	15	Quantitatif discret
levelNumberSystem1	NOMBRE TOTAL DE NIVEAU RATÉS POUR LE SYSTÈME 1	Nombre total de niveaux ratés dans le système 1 = nombre de morts. Le joueur peut faire plusieurs systèmes 1 car en mourant, il retourne au début du jeu.	15	Quantitatif discret
levelNumberSystem2	NOMBRE TOTAL DE NIVEAUX RATÉS POUR LE SYSTÈME 2	Nombre total de niveaux ratés dans le système 2. Le joueur peut faire plusieurs systèmes 2 car en mourant, il retourne au début du jeu.	15	Quantitatif discret
levelNumberSystem3	NOMBRE TOTAL DE NIVEAUX RATÉS POUR LE SYSTÈME 3	Nombre total de niveaux réussis dans le système 3. Le joueur peut faire plusieurs systèmes 3 car en mourant, il retourne au début du jeu.	15	Quantitatif discret
COMPTER LE NOMBRE D'ÉVÉNEMENTS PAR JOUEUR				
EVENTMENUCOUNT*	NOMBRE D'ÉVÉNEMENTS MENU	Nombre de fois total où le joueur a	3	Quantitatif discret

		envoyé l'événement MENU = nombre de fois où le joueur a ouvert un menu de paramètres		
EVENTSHOPCOUNT *	NOMBRE D'ÉVÉNEMENTS SHOP	Nombre de fois total où le joueur a envoyé l'événement SHOP = nombre de fois où le joueur a acheté quelque chose	12	Quantitatif discret
EVENTGAMESTARTEDCOUNT *	NOMBRE D'ÉVÉNEMENTS GAME STARTED	Nombre de fois total où le joueur a envoyé l'événement GAME STARTED = nombre de fois où le joueur a démarré le jeu	12	Quantitatif discret
EVENTGAMEENDEDCOUNT *	NOMBRE D'ÉVÉNEMENTS GAME ENDED	Nombre de fois total où le joueur a envoyé l'événement GAME ENDED = nombre de fois où le joueur a quitté le jeu ATTENTION : cet événement est envoyé à la session suivante donc beaucoup sont perdus si les joueurs ne reviennent pas. PAS fiable	12	Quantitatif discret
TOTALEVENTSGENERATED *	NOMBRE TOTAL D'ÉVÉNEMENTS GÉNÉRÉS	Nombre total d'événements reçus pour un joueur	1400	Quantitatif discret
PARAMETRES TECHNIQUES DU JOUEUR				
FIELDDEVICENAMELAST *	DERNIER NOM D'APPAREIL	Au cas où le joueur utilise plusieurs appareils, ici on enregistre le dernier. Le multi-plateforme n'était pas encore en place donc pas de possibilité d'accéder à ses jeux	Oculus Quest	Qualitatif

		depuis différents appareils. Doublonne deviceName		
FIELDDEVICETYPE LAST *	DERNIER TYPE D'APPAREIL	Idem pour le type. Doublonne deviceType	PC	Qualitatif

Liste des événements

Les événements sont déclenchés lors d'une ou plusieurs actions précises du joueur. Certains définissent le début et la fin d'une session de jeu. Il est important de préciser que les derniers événements précédant la sortie du jeu par le joueur ne sont pas envoyés avant que le jeu ne s'éteigne : ils ne sont pas générés assez rapidement avant la fin de la session pour être envoyés sur-le-champs. Ils sont donc enregistrés et envoyés lors de la session suivante. Or, DeltaDNA n'enregistre pas les événements qui présentent un écart de 32 jours et plus entre leur horodatage et celui de l'arrivée sur le serveur. Nous avons donc perdu certains événements, mais disposons d'un fichier les listant avec l'identifiant du joueur concerné, envoyé par mail par DeltaDNA.

Par ailleurs, les événements sont définis par rapport au cheminement du joueur dans le jeu et aux boucles définies plus haut dans le choix des données (voir sous-partie a). En effet, pour chaque système solaire, le joueur se rend d'une planète à une autre. Entre deux, la carte du système est affichée. Il s'agit donc d'une boucle car le joueur revient toujours à la carte entre deux niveaux. Or, il y a trois types de planètes. Il y a donc trois boucles à suivre : faire un niveau classique sur une planète à régénérer (ou planète à obeloïds, *obeloïds planet*), aller dans le magasin et éventuellement acheter puis aller sur une planète - évènement. Cette dernière ne génère aucun événement propre à elle car le joueur n'a pas de choix ou d'action à y faire. Seule la fin d'un niveau permet de savoir que le joueur était sur une telle planète. La planète-magasin, appelée Reliquaire de Kibele dans le jeu, génère en plus de la fin de niveau, un événement propre car le joueur a la possibilité d'acheter un sort ou un objet.

Enfin, il y a aussi des événements pour mesurer la progression dans le jeu et créer un entonnoir pour comprendre à quelle étape les joueurs abandonnent le jeu. Ce sont les *milestones* du jeu qui permettent aussi de vérifier que le jeu fonctionne à toutes les étapes. Perdre 25% des joueurs dans une étape indiquerait que quelque chose dysfonctionne entre les deux : l'étape suivante ne se charge pas, un élément du jeu bloque la progression, etc... A ces *milestones* s'ajoutent depuis le 29 janvier 2021 des *Achievements* qui marquent le passage d'un joueur par une situation précise et sont désignés comme des trophées (pratique très courante dans le jeu vidéo). Par exemple, un joueur ayant réussi à vaincre le premier *Boss* du système solaire 1 débloquera le premier trophée ou « badge » qui marque sa réussite.

* Générés automatiquement par DeltaDNA

NOM	TYPE	NOM FRANCAIS	DECRIPTION
SESSION			
gameStarted *	/	JEU ALLUMÉ	Premier événement d'une session.
sessionEnds	/	FIN DE SESSION	Souvent non enregistré car n'est pas envoyé assez vite avant fermeture du jeu.

saveAndQuit	/	SAUVEGARDE ET FIN DE JEU	Lorsque le joueur enregistre sa progression. Ferme le jeu automatiquement et donc souvent non envoyé assez vite.
ÉVÉNEMENTS FORMANT LES BOUCLES			
mapIsDisplayed	missionStart	AFFICHAGE DE LA CARTE	Lorsque la carte du système solaire est affichée.
planetToRegenerate	• levelStart • levelEnd (2 statuts : succès ou échec : voir paramètres) • levelEndShop • levelEndRandomEvent	PLANÈTE A REGENERER • Début de niveau • Fin de niveau planète classique • Fin de niveau magasin • Fin de niveau planète événement	Se déclenche lorsque le joueur va sur une planète, quel que soit sont type d'où un seul *levelStart*. Mais aussi à la fin du niveau : un type par catégorie de planète >> Il n'était pas prévu que *levelStart* se déclenche pour toutes les planètes mais il n'a pas été possible de rectifier avant la sortie du jeu, d'où la création de types différents pour la fin de niveau
shop	buy	MAGASIN	Se déclenche uniquement à l'achat. L'événement général MAGASIN permet de prévoir d'autres actions possibles que l'achat dans le futur. >> était prévu une option initialement : rafraîchir les produits proposés pour avoir un autre jeu de sorts et objets
Menu	• goesToOption • newGameStarted	MENU • va dans les options • réinitialise le jeu	Le joueur se rend dans un menu de paramètres : 1. bouton *Settings* dans le menu de départ, le seul qui permette de réinitialiser le jeu en plus des

			paramètres d'ergonomie 2. bouton de la manette gauche pour tous les paramètres d'ergonomie
ÉVÉNEMENTS DE L'ENTONNOIR			
milestones	• course001HandSelected • course002FirstTravelDone • course003DodgesDone • course004SphereDisplayed • course005HasHitFirstWave • course006RegeneratesFirstMoon • course007PlanetRegeneration • course008TravelToCraddle • course009TotemsRegenerated • course010TravelMapGrabbed • shopTuto001tutoLaunched • *shopTuto002Shieldpurchased* • systemFinished • winsGame	JALON • tutoriel étape 1 : main dominante sélectionnée • tutoriel 2 : premier voyage dans le tunnel interstellaire fait • tutoriel 3 : a réussi à éviter les tirs ennemis • tutoriel 4 : la sphère du shaman est apparue • tutoriel 5 : a tué le premier ennemi • tutoriel 6 : a régénéré la première lune • tutoriel 7 : a régénéré la première planète • tutoriel 8 : est arrivé sur la planète berceau	Jalons pour l'entonnoir et voir à quelle étape les joueurs se perdent ou le jeu ne fonctionne pas correctement. Se déclenchent lorsque le joueur fait chaque étape, avec le type approprié. *shopTuto002Shieldpurchased* : événement non utilisé à partir de la mise à jour du 15 décembre 2020. >> Le bouclier ne s'achète plus à la fin du tutoriel. Le joueur le prend dans sa sphère durant le tutoriel. Il n'est pas possible d'identifier le nom de ce que le joueur récupère dans la sphère, donc cette étape n'a plus été monitorée.

		• tutoriel 9 : a régénéré le premier pilier • tutoriel 10 : la carte est affichée pour la première fois • tutoriel magasin 1: le tutoriel démarre • tutoriel magasin 2 : le joueur achète le bouclier • système solaire terminé • a gagné le jeu	
EVENEMENTS DÉBLOQUANT LES TROPHEES			
achievements	• 1stDeath • 1stRegeneration • boss1Dead • boss2Dead • boss3Dead • gameWon • levelWithoutWounds • shopLevel5 • shopLevel10 • shopLevel15 • shopLevel20	TROPHÉES • 1ere mort • 1ere régénération (planète du tutoriel) • 1er boss mort • 2e boss mort • 3e boss mort • Jeu gagné • A réussi un niveau sans blessure (le nombre de points de vie	Se déclenchent la première fois que le joueur remplit les conditions.

		reste le même) ● A atteint le niveau 5 du magasin ● Idem pour le niveau 10 ● Idem pour le niveau 15 ● Idem pour le niveau 20	

2. CREATION DU PIPELINE DE GENERATION ET RECEPTION DES DONNEES

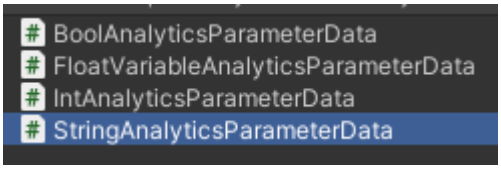
Le pipeline de génération des données depuis le jeu se compose de plusieurs étapes, depuis la génération des données jusqu'à leur mise à disposition sur la plate-forme Delta DNA (DDNA). Nous avons choisi cette dernière afin de déléguer la gestion de la base de données et d'avoir des fonctionnalités offrant des premiers *metrics* automatiquement calculés. De plus, les fonctionnalités de rapport automatiques et *dashboard* nous semblaient intéressantes.

Le pipeline en lui-même comporte quatre éléments : l'*Analytics manager* qui crée les événements et paramètres lorsqu'une combinaison de facteurs se produit dans le jeu, le wrapper qui appelle les fonctions nécessaires à la création et l'envoi des événements, les routines de réception et traitement de données par Delta DNA, et enfin, la mise à disposition de ces dernières. Je ne détaillerai pas les routines DDNA car cela n'a pas de rapport direct avec le sujet. Mais les principales règles de réception ou refus des données à l'entrée de la base seront abordées.

a) L'ARCHITECTURE DES EVENEMENTS

La première étape fut d'intégrer les événements et paramètres cités plus haut dans le projet du jeu. Le jeu en lui-même fut développé dans le moteur de jeu Unity, et la gestion des données devait y être intégrée en C#. Pour une meilleure compréhension de la suite, je rappelle que les données répondent à des événements qui se produisent dans le jeu et que l'on nommera *event*, terminologie en usage dans un projet Unity. Ces events sont des objets qui permettent de déclencher une réponse : les conséquences d'un tir ennemi ou l'envoi de données. Par exemple, l'event *planetToRegenerate* se déclenche lorsque le joueur se rend sur une planète et démarre un niveau. Les *events* sont à distinguer des simples événements qui sont les événements data envoyés en réponse et réceptionnés par DeltaDNA.

Les événements et leurs paramètres sont créés au moment où une combinaison *d'events* et d'options sont réunis. Plusieurs scripts génériques ont été développés par notre directeur technique, afin de créer l'architecture des paramètres nécessaires en fonction de leur type (nombre entier, chaîne de caractères etc.…).

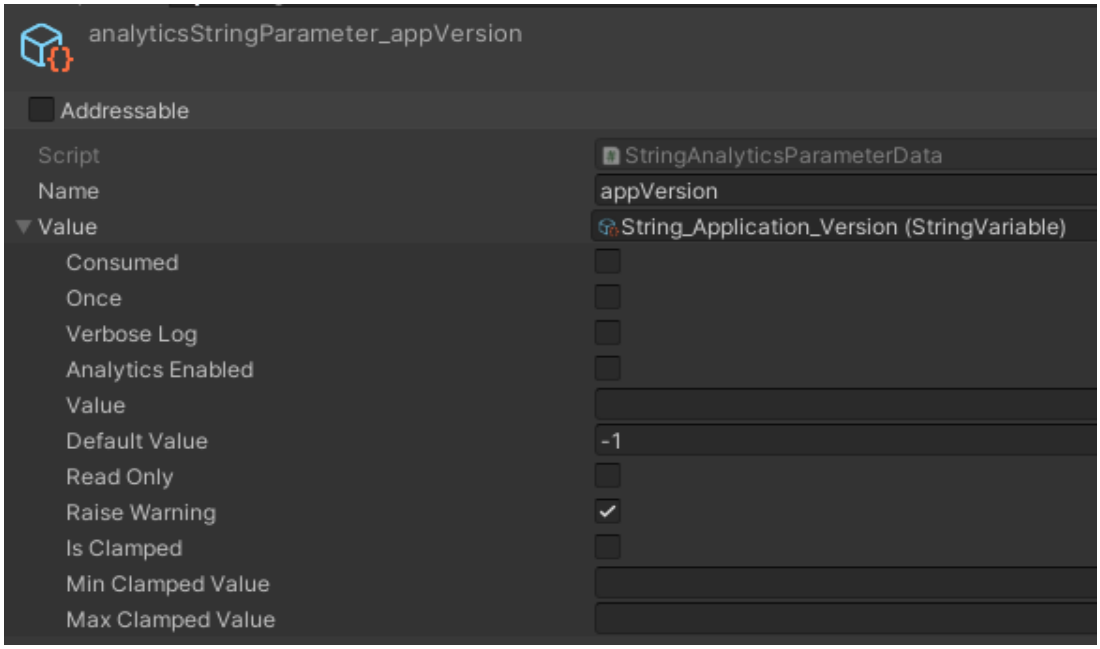


Liste des scripts créés pour chaque type de variable.

A titre d'exemple, le script construisant un paramètre contenant du caractère se construit ainsi:

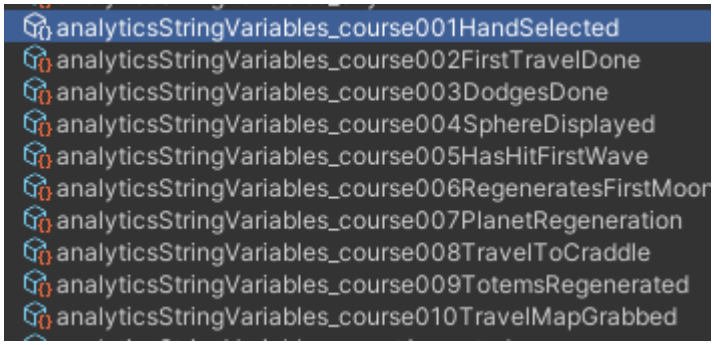
```
using ScriptableObjectArchitecture;
using UnityEngine;

namespace Analytics
{
  [CreateAssetMenu( fileName = "New String Analytics Parameter", menuName = "Analytics/ String Parameter" )]
  public class StringAnalyticsParameterData : AnalyticsParameterData
  {
    #region Public

    public StringVariable m_value;

    #endregion


    #region Main

    public override object GetValue()
    {
      if(m_value == null)
      {
        Debug.LogError( m_errorMessage + this.name, this );
        return "";
      }
      return m_value.Value;
    }

    #endregion
  }
}
```

Il permet d'avoir un champ de nom de paramètre, un autre pour la valeur du paramètre et un dernier pour la valeur par défaut lorsqu'aucune autre n'est déclarée.

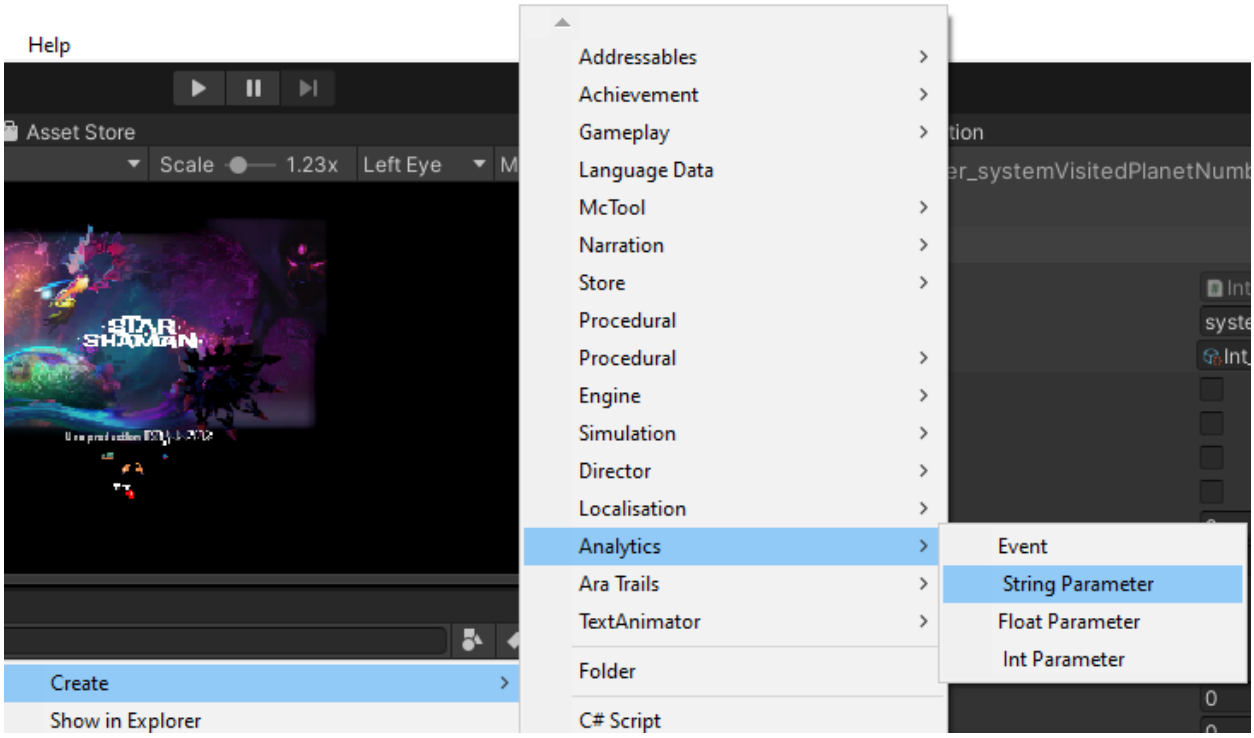


En effet, il est alors possible d'utiliser une variable du jeu (ci-dessus le numéro de version codé en caractères), et de la déposer dans le champ V*alue* afin qu'au moment de la création de tout événement utilisant ce paramètre, la valeur du numéro soit automatiquement récupérée. Si aucune variable n'est déclarée, le paramètre contient la valeur par défaut -1. Cela se voit lors des tests et permet de rectifier.

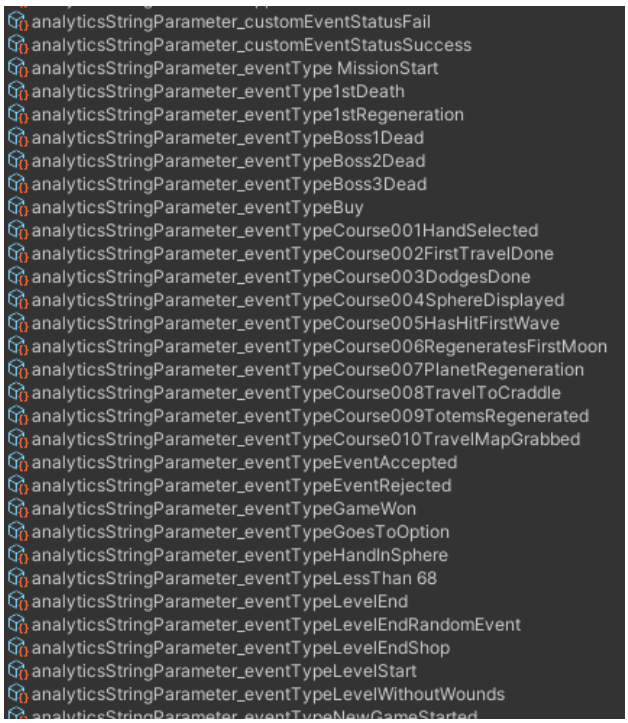
Certaines variables n'existaient pas dans le projet et ont été créées spécialement. Ce fut notamment le cas des valeurs en caractères des types d'événements tels que le nom des étapes de l'événement *milestone* (voir section précédente) :



Une fois les scripts créés, les paramètres pouvaient être créés un par un en suivant les règles de nommage internes.

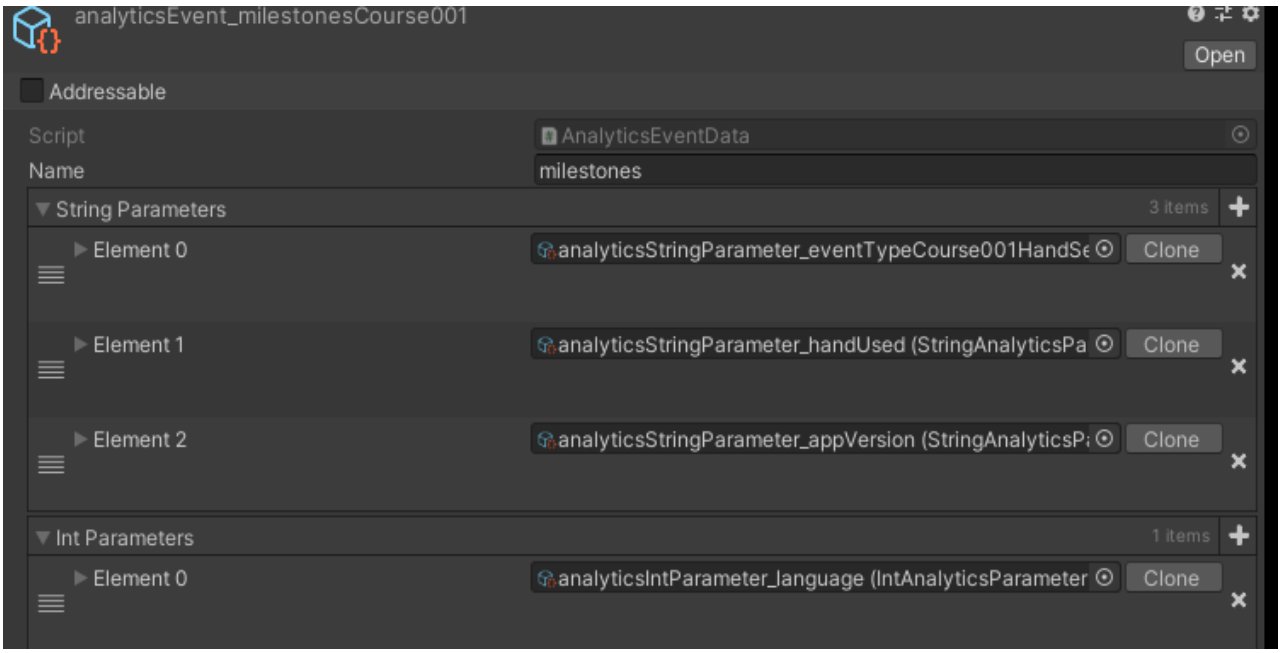


Certains *events* du jeu déclenchaient des événements différents. Par exemple, la fin de niveau déclenchait soit une réussite en cas de régénération de la planète, soit un échec et la mort du joueur. Le paramètre *customEventStatus* ne pouvait donc contenir la même valeur dans l'un et l'autre cas. Deux paramètres différents portant le même nom furent donc créés avec un paramètre « *status* » dont la valeur était différente. Il en fut de même pour les *eventType,* paramètres utilisés (voir section précédente) pour préciser de nombreux événements. Une fin de niveaux renvoyait ainsi une valeur différente selon le type de planète terminée. Les *eventtype* sont les paramètres les plus nombreux de ce fait.



Une fois les paramètres créés, je pus les utiliser pour former des événements. Encore une fois, certains portaient un même nom mais se déclinaient en plusieurs variantes. Par exemple, les étapes du tutoriel étaient suivies en déclenchant des événements à chaque étape passée par le joueur. Ce furent autant d'événements créés qui se nommaient *milestones* mais ils se déclinaient en plusieurs *eventTypes* :

Dans chaque champ de l'événement, un paramètre est déposé, lequel contient une variable avec une valeur. Cette architecture contient donc tous les éléments permettant ensuite d'appeler un événement et de l'envoyer vers la base de données. Chaque mise à jour ou changement dans le jeu nécessite de revoir ces événements, d'en supprimer/ajouter et de tester l'ensemble du pipeline à nouveau.

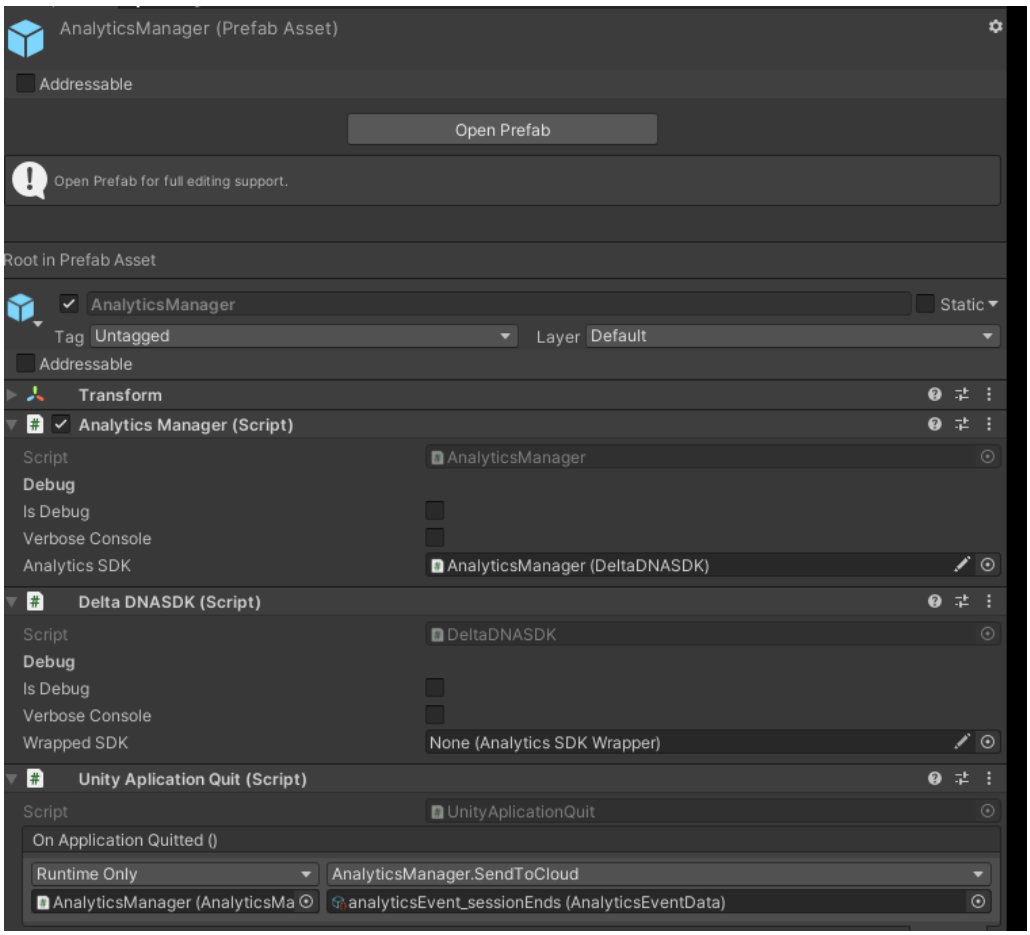


Exemple d'un des événements appelés *Milestones* et de ses paramètres

b) DECLENCHEMENT DES EVENEMENTS

Dans le projet Unity, chaque événement data est rattaché à un *game event* du jeu en tant que réponse. Les données sont envoyées en réponse au déclenchement de l'événement. Cela permet notamment de réduire la quantité d'information par rapport à un stream continu de données, et donc de sauvegarder la performance du jeu, très importante en réalité virtuelle. Cette dernière demande beaucoup d'effort de la part du cerveau et du système visuel chez le joueur. Un manque de performance entraîne le brouillage de l'image, le "gel" de l'écran et donc une forte gêne physique chez le joueur. L'équipe technique du projet veillait donc à réduire tout ce qui aurait pu lui porter atteinte. Par ailleurs, pour un data analyst, l'afflux continu de données peut rapidement noyer l'information importante. Hormis quelques jeux très puissants -en temps réel sur serveur et avec un nombre de joueurs gigantesque-, envoyer les informations uniquement lorsque des événements se déclenchent est une méthode plus efficace.

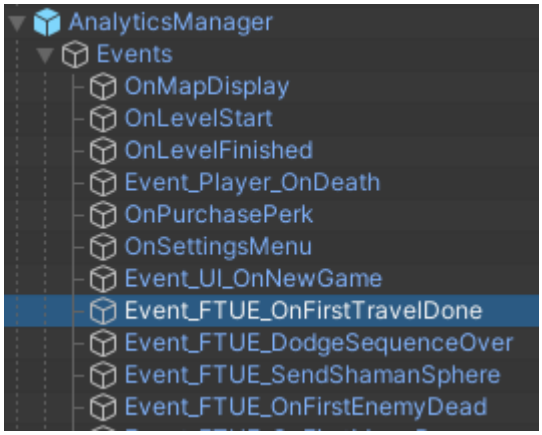
Un projet Unity contient des managers qui rassemblent la partie de l'architecture d'un jeu sur un élément en particulier : la planète-magasin, la génération de niveaux ou encore l'interface (UI). Un *Analytics manager* fut donc créé pour rassembler les événements liés aux données, ainsi que les fonctions à appeler pour la création d'un évènement et son envoi. Il s'agit d'un objet Unity auquel sont attachés les scripts contenant les fonctions nécessaires.



Scripts contenus dans l'*Analytics Manager*

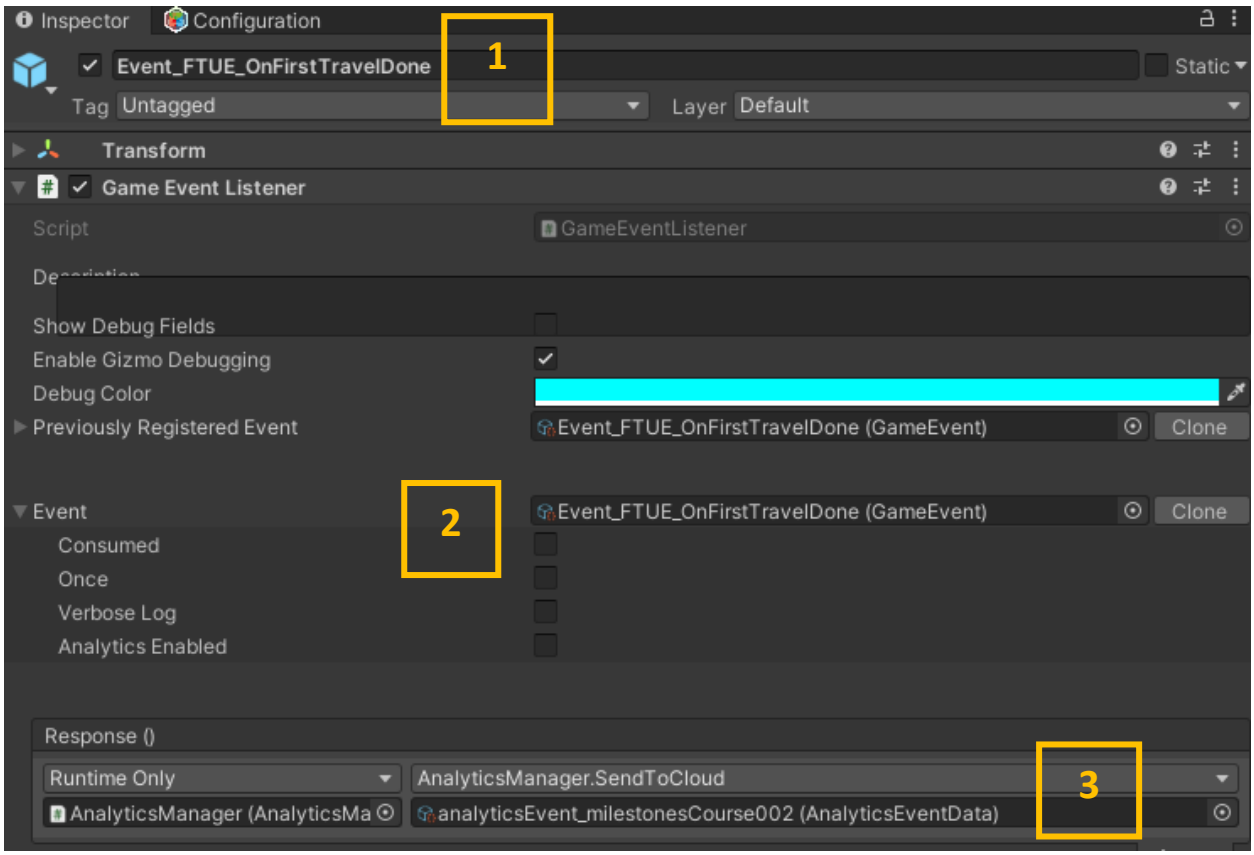
Pour commencer la description de cet objet fondamental du pipeline, j'aborderai les scripts que j'ai développés sous la supervision de notre directeur technique. Ils sont de différentes natures. L'un contient les fonctions qui transforment l'événement appelé en événements acceptables par DeltaDNA (avec les bons champs, le bon type de valeurs, etc.…). Un autre encore appelle les fonctions de Delta DNA pour déclencher l'envoi. Ainsi donc, dès qu'un *game event* est déclenché (*raised*) par le joueur, un événement data est envoyé en réponse, grâce aux scripts qui le mettent en forme et l'envoient.

L'objet *Analytics manager* est inclus dans les managers du projet et des objets-enfants lui sont ajoutés. Ces objets portent les noms des *game events* auxquels ils réagissent.



De même, un événement data se décline en plusieurs objets. Par exemple, les étapes du tutoriel (ci-contre appelé FTUE, appellation officielle dans le projet) sont déclinées en plusieurs *game events* dont chacun a un évènement data en réponse.

Par exemple, le joueur arrivant sur la planète du tutoriel, par le tunnel interstellaire, déclenche le *game event OnFirstTravelDone* qui déclenche en réponse l'évènement data *milestones* avec comme eventType *course002FirstTravelDone :*



L'objet-enfant contient un champ nom (1), un script *Listener* (2) et une réponse (3). Le *Listener* est un objet d'Unity qui permet d'écouter un game *event* et d'envoyer une réponse. C'est la base de l'architecture d'un jeu. Dans le champ *Event* du listener, le *game event* est glissé afin de déclarer lequel est suivi. Dans le champ réponse, l'objet Analytics Manager est glissé, ainsi que l'événement data qu'il traite et envoie.

Certains objets enfants avaient également un script attaché qui vérifiait qu'une condition était remplie avant d'envoyer la réponse. Par exemple, l'étape de régénération de planète du tutoriel était déclenchée par un événement qui s'appliquait à toutes les fins de niveaux réussis. Le script vérifiait donc que la variable booléenne *!m-boolTutorialCombat* contenait la valeur indiquant que l'on se trouvait dans le tutoriel, et cela avant d'envoyer l'événement.

En dernier point, il faut aborder quelques événements data hors du manager. En effet, certains se déclenchent lorsque des objets qui ne sont pas des événements entrent en action. N'ayant pas trouvé comment les suivre au moyen d'une *Listener*, j'ai placé mes réponses directement sur les objets en question dans leurs managers respectifs. Lors des premières versions du jeu, il existait un tutoriel de la planète-magasin dont je suivais deux étapes. Mes événements data furent placés directement en réponse à l'affichage des cartons de texte correspondant à l'achèvement de l'étape. Un objet enfant vide fut créé dans l'*Analytics Manager* afin de garder la trace de l'emplacement de cette réponse.

Il en fut de même pour les *achievements*. Ceux qui ne se déclenchaient que la première fois qu'une situation se produisait virent ma réponse placée dans le champ *Is true* ou *Is False,* pour respecter cette condition.

c) La question du Wrapper

Pour terminer la description du pipeline du côté du projet, j'aimerais aborder brièvement la question du Wrapper. Notre directeur technique m'a appris à en créer un afin de garantir la viabilité du pipeline dans le temps. En effet, nous pourrions changer de fournisseur de base de données dans le futur. Au lieu d'appeler les fonctions de Delta DNA directement pour mettre en forme et envoyer l'événement, il fallait donc passer par un intermédiaire avec des fonctions génériques. Ces dernières se chargent d'appeler les fonctions de DDNA, rendant plus simple tout changement éventuel de fournisseur.

Ci-dessous, deux fonctions du Wrapper appellent chacune une fonction de Delta DNA. En cas de changement de fournisseur, il suffira de changer les fonctions appelées et non l'entièreté des scripts.

```
1 référence
private void ConnectToProvider()
{
    m_analyticsSDK.InitializeSDK();
}

0 références
public void SendToCloud( AnalyticsEventData analyticsEvent )
{
    m_analyticsSDK.SendingData(analyticsEvent);
}

#endregion
```

Ainsi donc, le pipeline se résume ainsi : un *game event* déclenche un événement data (si les conditions d'un éventuel script sont remplies) et cela appelle les fonctions du Wrapper qui appellent en retour celles du fournisseur.

d) Du cote de Delta DNA

Les événements sont réceptionnés par DDNA, sous forme de fichier json, qui vérifie leur validité en fonction d'une liste blanche que j'ai créée sur la plate-forme. Tout événement non enregistré, avec un nom différent, un paramètre en plus ou moins, ou encore dont le type d'un paramètre n'est pas bon, est rejeté et non enregistré dans la base. Selon le motif de refus, on peut les voir dans une liste d'erreurs.

La plate-forme DDNA comporte deux environnements, un de développement permettant de tester le pipeline et l'autre de production appelé LIVE. La liste blanche se construit dans l'environnement DEV.

Il faut d'abord déclarer les paramètres (n°1, avec nom et types de valeur) puis créer les événements avec leurs paramètres associés (2). Les paramètres étant peu nombreux et partagés par la majorité des événements, cette partie fut relativement simple. Chaque mise à jour apportait son lot de changements cependant.

L'*event browser* permet de tester l'arrivée des événements et leurs bonnes réceptions avec quelques minutes de latence. Bien que l'ordre d'affichage soit lié à l'ordre d'arrivée et non de création, les événements sont aisément repérables. Il est d'ailleurs possible de filtrer par joueur afin de ne voir que ses propres événements ou encore par nom d'événement. Par défaut, les événements invalides ou erreurs sont affichés. Il est possible de voir le détail pour repérer les erreurs de valeurs ou de types.

Le pipeline technique est donc le reflet du travail d'identification des événements à suivre, de leurs paramètres et des *events* /variables les représentants dans l'architecture du jeu. Nous allons à présent aborder l'analyse des joueurs en elle-même et l'utilisation des données récoltées pour ce faire.

3. DETERMINER LES PROFILS DE JOUEURS A L'AIDE DE LA METHODE DES K-MEANS

Notre premier objectif est de créer des profils de joueurs à l'aide de la méthode des k-means. J'ai commencé par déterminer les composantes de l'analyse, c'est-à-dire les variables à utiliser dans le clustering. Nous détaillerons ici leurs créations et la méthodologie suivie pour le clustering. Pour finir, nous analyserons les résultats des clusterings.

J'ai pris les événements des joueurs depuis le 7 novembre 2020 (1ère mise à jour de Star Shaman, voir ci-dessus 1.b.) afin d'avoir les données propres sans problème de mélange avec les tests de l'équipe. Les données ont été téléchargées le 27 mars 2021 à 17h52 pour les événements, le 10 avril 2021 pour les données de pays et premières connexions (inchangées, quelle que soit la date de téléchargement), ainsi que le 19 mai 2021 pour la variable de dernier horodatage nécessaire aux nouvelles analyses. Nous avons ainsi un échantillon initial de 3083 joueurs et joueuses de Star Shaman.

a) DETERMINER LES COMPOSANTES DE L'ANALYSE

Je me suis demandé comment résumer les actions des joueurs de Star Shaman afin de distinguer des groupes : quelles variables seraient intéressantes pour décrire les joueurs de ce jeu et comment appréhender la réalité virtuelle ? D'emblée se posait la question du déséquilibre de nos joueurs en ce qui concerne le pays, le type de casque. Une modalité de chaque variable concentre la très grande majorité des joueurs : 97% des joueurs utilisent un Oculus Quest et 57,21% des joueurs sont aux Etats-Unis. Il ne semblait donc pas possible de faire une analyse de l'ensemble des joueurs sans qu'un biais soit introduit.

Par ailleurs, la majorité de nos joueurs quitte le jeu très rapidement. On observe sur les résumés statistiques que trois quarts des joueurs sont restés 45 minutes et moins dans le jeu, soit l'équivalent de deux jours et moins. Faut-il donc prendre exemple sur le travail de Myat Aung et ses co-auteurs (Aung et al., 2019) qui divisent leur échantillon en deux, en fonction du niveau auquel partent les joueurs de *Just Cause* ?

> *« This dataset is then split into two parts, one for Early Dropouts, those stopping shortly after starting to play, and Committed Players who keep playing for a longer period. This split allows the development of behavioral profiles for specific segments of players, which in turn helps inform about differences in player experience. »*
>
> (Aung et al., 2019)

Mais le critère de séparation est compliqué à trouver si l'on ne se contente pas de la médiane de temps joué. Faut-il considérer ceux qui partent au bout d'un jour, 2 jours, avant la fin ou à la fin du tutoriel ? J'ai eu rapidement une confirmation: le départ des joueurs n'est pas lié à la version du jeu et celle-ci n'est donc pas un critère de séparation non plus. 62.5% d'entre eux partent après un jour de jeu, quelle que soit la date d'achat ou d'installation.

Je prends finalement en compte le pays et le nombre de niveaux pour séparer l'échantillon en quatre. Nous avons ainsi quatre sous-échantillons :

- Les joueurs des Etats-Unis partis avant la 3^{e} planète régénérée : ils n'ont réussi au maximum qu'un niveau après le tutoriel.
- Les joueurs de tous les pays, y compris les Etats-Unis, partis avant la 3^{e} planète régénérée.
- Les joueurs des Etats-Unis partis à partir de la 3^{e} planète régénérée : ces joueurs ont continué à jouer après le premier niveau.

- Les joueurs de tous les pays, y compris les Etats-Unis, partis à partir de la 3^{e} planète régénérée.

Le nombre de planètes régénérées permet de situer tous les joueurs sur un même plan : quel que soit le temps mis pour arriver au même niveau, certains sont partis plus rapidement. 35% des joueurs ne régénèrent que 2 planètes maximum, celles-ci comprenant la planète du tutoriel ! Quant à la division Etats-Unis / Tous les pays, il s'agit de comparer les résultats entre les Etats-Unis seuls et l'ensemble des pays : s'ils sont les mêmes, alors on peut y voir l'influence des joueurs des Etats-Unis.

Les variables de l'analyse furent cherchées parallèlement à cette réflexion et ont abouti au critère de séparation des joueurs sur le nombre de niveaux de jeu. J'ai tout d'abord déterminé des composantes qui décrivent les joueurs : la performance, les compétences, l'immersion et la progression. La performance regroupe les actions du joueur dans le jeu et donne une estimation de son comportement. Les compétences regroupent les variables permettant de déterminer le niveau d'un joueur. S'il est débutant, le nombre de parties sera élevé car il sera mort fréquemment et sans doute rapidement, d'où le calcul du nombre de morts par heure de jeu. Le ratio du nombre de planètes régénérées sur le nombre de planètes visitées permettra de voir l'efficacité du joueur également : est-il avancé et régénère-t-il la quasi-totalité des planètes visitées ?

Cela est équilibré par la progression qui doit montrer la différence entre le jeu dans la première partie et celui de la dernière. En effet, le joueur peut devenir très performant et se classer parmi les meilleurs. J'observe donc par exemple le nombre de morts par système solaire, l'évolution des scores, notamment par niveau. Le fait de vérifier que le joueur se trouve dans le même système solaire permet aussi de distinguer ceux qui progressent ou non.

Enfin, l'immersion est un facteur très important en réalité virtuelle. Elle est difficile à appréhender sans données émotionnelles et/ou de mouvements du corps, ou encore sans questionnaire. Je propose d'essayer de l'appréhender par le fait qu'un joueur pris par l'univers jouera souvent et/ou longtemps malgré les revers qu'il pourrait connaître. D'un autre côté, on pourrait également voir l'immersion au fait qu'il visite plus de choses que les joueurs faisant le minimum pour progresser dans le jeu. Dans les systèmes solaires 2 et 3, le nombre de planètes est supérieur à celui qui est nécessaire à l'obtention des éléments de vie requis. Avoir plus d'éléments de vie ne donne aucun avantage. Faire plus de planètes expose au risque d'être tué par les obeloïds mais représente une chance de gagner plus de monnaie pour acheter des sorts. Il me semble que suivre si le joueur régénère toutes ces planètes donne un aperçu de la motivation et donc de l'immersion.

En raisonnant de la sorte, j'aboutis aux variables suivantes :

A. Performance
   - Ratio nombre de morts/planètes régénérées
   - Nombre de morts au système 1, 2 et 3 séparément
   - Score moyen par niveau : cette variable dépend du nombre d'ennemis touchés et des tirs reçus ; c'est donc un bon aperçu de la performance
   - Montant moyen de monnaie gagné par niveau
   - Nombre de planètes événements (Random events)
   - Nombre de clics sur les options

B. Compétences du joueur
   - Ratio nombre de planètes régénérées/nombre de planètes visitées
   - Nombre de morts moyen par heure
   - Nombre de parties au total

- Temps moyen jusqu'à complétion du dernier système fini : il s'agit d'un temps moyen en heures car le joueur peut faire plusieurs fois son dernier système solaire s'il meurt.
- Nombre de planètes régénérées : il s'agit de l'équivalent du nombre de niveaux.
- Nombre de boss tués
- Système solaire maximum atteint
- Le joueur a-t-il gagné le jeu ? ***** : remplacé par un temps pour gagner (avec 0 minutes pour ceux qui ne gagnent pas).

C. Immersion
- Nombre de sessions
- Temps moyen par session en minutes
- Temps total de jeu en minutes
- Tous les random events sont-ils fait au système 1 ? * : au système solaire 1, les planètes à obeloïds sont toutes obligatoires pour obtenir assez d'éléments de vie. Remplacé par le temps jusqu'à complétion de la dernière planète-événement. Avec 0 minute si le joueur n'a pas réussi.
- Toutes les planètes sont-elles visitées au système solaire 2 ? *
- Toutes les planètes sont-elles visitées au système solaire 3 ? *

D. Progression
- Différence entre le score minimum de la 1ère et la dernière partie
- Différence entre le score maximum de la 1ère et la dernière partie
- Différence entre le montant de monnaie moyen par niveau de la 1e et la dernière partie
- Différence entre les scores moyens par niveaux de la 1e et la dernière partie
- Le joueur est-il dans le même système solaire en début et en fin de jeu? * Cette variable n'est pas remplaçable par une variable numérique pour la clusterisation. Je l'ai mise de côté.
- Le joueur est-il arrivé dans le système 3 ? * : Remplacé par le temps pour arriver au système 3 et 0 minutes pour ceux qui n'y sont pas.
- Nombre de parties par système solaire 1
- Nombre de parties par système solaire 2
- Nombre de parties par système solaire 3

Les variables suivies d'un astérisque ont posé problèmes car le clustering prenant en compte des distances, il n'était pas possible d'introduire des booléens. Après leurs créations, j'ai donc créé des variables numériques pour les remplacer. Il s'agit souvent du temps mis pour arriver à l'étape, ce qui permet de mettre 0 minutes à ceux qui ne sont pas arrivés là. Deux de ces booléens n'ont pas été remplacés ni même utilisés, quelle que soit la méthode, car ils contenaient une seule modalité. En calculant si les joueurs avaient fini l'ensemble des planètes des systèmes solaires 2 et 3, je me suis aperçu que ce n'était le cas d'aucun !

Une fois les quatre jeux de données créés selon les critères précités, je trouve des incohérences. Dans le jeu de tous les pays avec les joueurs à départs rapides, quatre d'entre eux ont un nombre de morts dans le 2^{e} système solaire alors qu'ils ne peuvent pas y être arrivés. Je supprime ces joueurs problématiques (deux Français, un Italien et un Canadien) du jeu, après avoir identifié la source du problème dans les données brutes, car les données sont visiblement erronées et/ou tronquées.

De même, deux autres joueurs français ont des événements dans le système solaire 3, alors qu'ils n'ont pas évolué dans le premier. Ils sont supprimés du jeu également. L'un deux est un nouveau joueur car l'événement *newPlayer* est envoyé. Il devrait donc commencer au système 1. Or, l'événement envoyé lors de son passage dans les options est marqué au système solaire 3. Enfin, un dernier joueur français a renvoyé un ID de système solaire 2 lors du redémarrage d'une session. Je l'enlève également.

Après calculs et corrections, ainsi que suppressions de doublons, on obtient quatre jeux de données :

- 491 personnes pour les Etats-Unis, avec un départ avant la 3e planète régénérée
- 1211 personnes pour les Etats-Unis, avec un départ à partir de la 3e planète régénérée
- 1076 personnes pour l'ensemble des pays, avec un départ avant la 3e planète régénérée
- 1892 personnes pour l'ensemble des pays, avec un départ à partir de la 3e planète régénérée

Il faut noter que la plupart des variables, dans les jeux de données des joueurs partis avant la 3ème régénération, contiennent des valeurs nulles car ces joueurs quittent le jeu avant de faire certaines actions. Aussi, j'enlève les variables concernant les systèmes solaires 2, 3 et la victoire. Je dois également enlever deux variables qui ne contiennent qu'une modalité chacune et qui ne sont donc pas clivantes pour cette population. Il s'agit du temps pour terminer la dernière planète événement au système solaire 1 : ce temps est de zéro car aucun joueur parti rapidement du jeu ne les a terminées. Enfin, tous les joueurs sont restés dans le même système solaire entre le début du jeu et les derniers événements du joueur : la variable d'évolution n'est pas utilisable non plus.

Nous avons donc deux jeux de données des joueurs partis avant la 3ème régénération réduite à 18 variables, au lieu de 29.

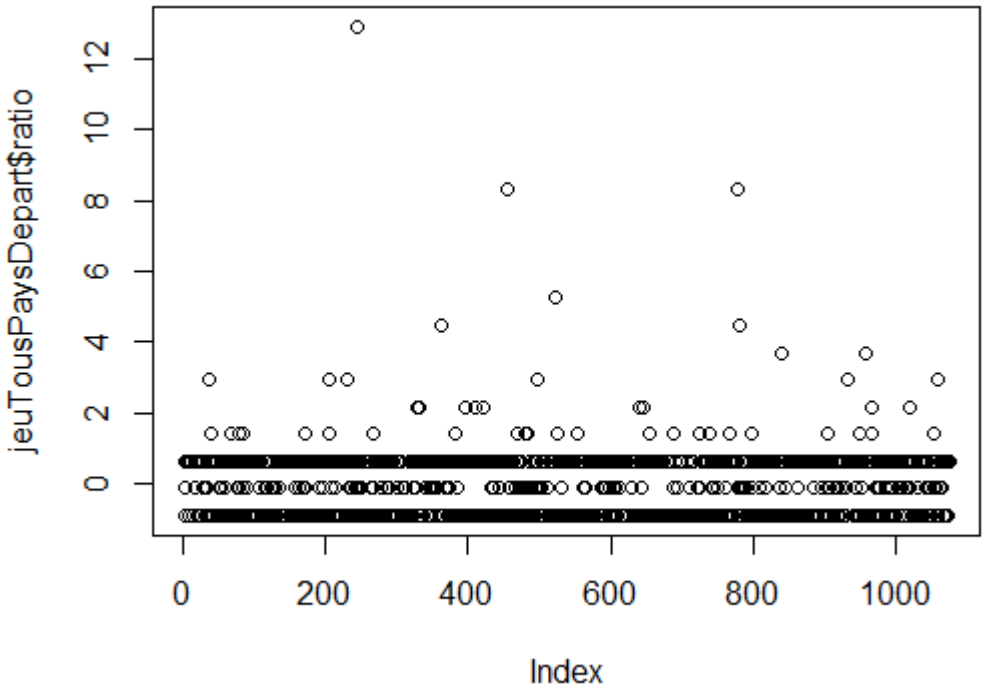
b) DESCRIPTION DES METHODES K-MEANS

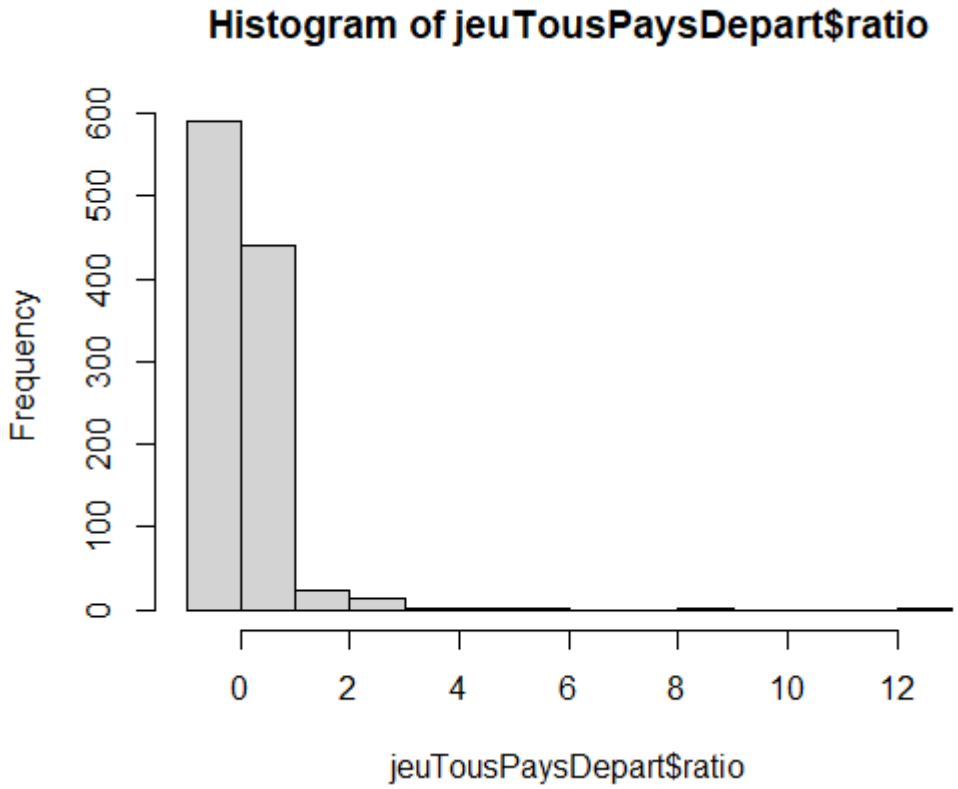
Nous allons ici décrire les clusterisations effectuées sur les quatre jeux de données, une par une. Une comparaison des résultats est présentée en sous-section suivante. Jusqu'à quatre hypothèses de travail ont été utilisées pour chaque jeu de données afin d'obtenir un clustering satisfaisant quant aux individus à valeurs extrêmes (appelés par la suite *outliers*). Ces individus sont extraordinaires par rapport aux autres joueurs et présentent des valeurs extrêmes pour certaines variables. Or, la méthode des k-means est sensible à leur influence sur le jeu de données. En effet, les valeurs extrêmes augmentent les moyennes des valeurs d'une variable et placent des individus en marge du partitionnement, étirant par-là les clusters. Nous aborderons donc rapidement une hypothèse intermédiaire de suppression de leur influence avant d'aborder le partitionnement final.

Supprimer l'impact des *outliers*

Un premier partitionnement sur les données telles que calculées n'a pas abouti tout d'abord en raison de la présence d'*outliers* : les clusters se chevauchaient et n'avaient pas d'homogénéité interne forte. Or, ce sont les deux critères d'un clustering réussi : maximiser la distance entre les clusters pour avoir des groupes aux caractéristiques bien distinctes et minimiser la distance au sein du groupe (homogénéité forte) pour que ses individus aient les mêmes caractéristiques. J'ai donc cherché une méthode pour supprimer l'influence des individus extrêmes.

J'ai généré des nuages de points et des histogrammes de chaque variable afin de déterminer à partir de quelle valeur, de chaque côté de la distribution, un individu était considéré comme extrême. Par exemple, dans le cas du ratio mort/planètes régénérées, nous avons les graphiques suivants :





99% des valeurs se répartissent jusqu'à 2,947 : 99% des joueurs ont donc un ratio de 2.9 et moins ; ils sont morts jusqu'à 2,9 fois plus qu'ils n'ont régénéré de planètes. Le nuage de point suggère en revanche une relative proximité par rapport à la masse des joueurs des valeurs jusqu'à 4, voire 5. Je choisis donc de fixer le seuil à 4, afin de ne pas introduire de biais en retirant systématiquement 1% des valeurs.

Après avoir fait de même avec toutes les variables, j'ai remplacé les valeurs extrêmes des joueurs par ces seuils. Cette méthode n'a été appliquée qu'au jeu de données de l'ensemble des pays, dont les joueurs

étaient partis avant la 3ème régénération. En effet, les résultats furent moyens : les clusters se chevauchaient et la variance expliquée diminuait. Cette dernière est le pourcentage de variance des valeurs des joueurs qui est expliquée par le partitionnement. Mon pourcentage de 54% signifiait donc que seuls 54% des valeurs disparates des joueurs trouvaient un sens dans le clustering réalisé.

Méthode finale utilisée

La méthode des seuils n'ayant pas fonctionné, j'ai finalement décidé d'identifier visuellement les outliers sur les clusters obtenus. Il est très difficile de déterminer si un individu est un outlier ou simplement un individu singulier dont la singularité doit être conservée car elle est représentative d'un comportement intéressant pour l'analyse. J'ai donc vérifié le nombre de valeurs extrêmes de chaque individu repéré et les ai supprimées quand je confirmais leur nature d'outliers. Par ailleurs, pour certains jeux de données, des clusters avec cinq joueurs avaient été créés. Cela me semblait peu par rapport au nombre total de joueurs : j'ai décidé de fixer un minimum de 5% du nombre total des joueurs pour former un cluster, soit 54 individus.

Je compare ensuite les valeurs des joueurs aux résumés statistiques des variables de mon jeu de données initial. Je construis pour chaque jeu de données un tableau indiquant si le candidat outlier a une valeur extrême (dans le 1er quartile ou le 4ème). Je précise aussi s'il s'agit du maximum ou du minimum. En comptant le nombre total de valeurs extrêmes et de minimums/maximums, je peux juger si le joueur est réellement un outlier ou non. Par exemple, beaucoup de joueurs du jeu de l'ensemble des pays, avec départ avant la 3ème régénération, ont des valeurs égales à celles de 25% et 75% des joueurs. Il s'agit des bornes supérieures du 1er et 3ème quartile. Ces valeurs sont considérées comme dans la norme acceptable. Dernier exemple, dans le jeu des Etats-Unis dont les joueurs continuent après la 2ème régénération, j'avais identifié l'individu 426 qui ne présente aucune valeur extrême dans mon tableau. C'est donc le seul exemple de joueur que je ne supprime pas car je ne le considère pas comme un outlier.

J'ai finalement retiré des jeux de données respectivement onze, treize, quatre et sept joueurs ; 35 joueurs sont ainsi retirés de l'analyse.

Une fois les quatre jeux de données nettoyés de leurs outliers, je refais mon clustering en tentant quatre groupes à chaque fois. Ce nombre me paraît intéressant pour l'analyse. Pour chaque jeu, j'obtiens des effectifs déséquilibrés, des chevauchements de clusters et/ou une homogénéité intra-cluster trop faible. A titre d'exemple, les joueurs de l'ensemble des pays partis avant la 3ème régénération sont répartis dans des clusters de 362, 220, 55 et 429 individus respectivement. Le 3e cluster est nettement disproportionné. En observant le graphique ci-dessous, on voit très nettement que la distance entre les clusters est nulle : le chevauchement est élevé. Quant au cluster 2, son homogénéité est très faible, notamment en raison de l'individu 683 qui est très isolé. Ce dernier semble être un outlier qui n'apparaissait pas avant et que la nouvelle échelle de graphique indique. C'est le cas pour tous les jeux de données : après suppression des outliers, d'autres semblent apparaître. Leur analyse révèle qu'ils n'en sont pas et je les garde dans leurs jeux de données respectifs.

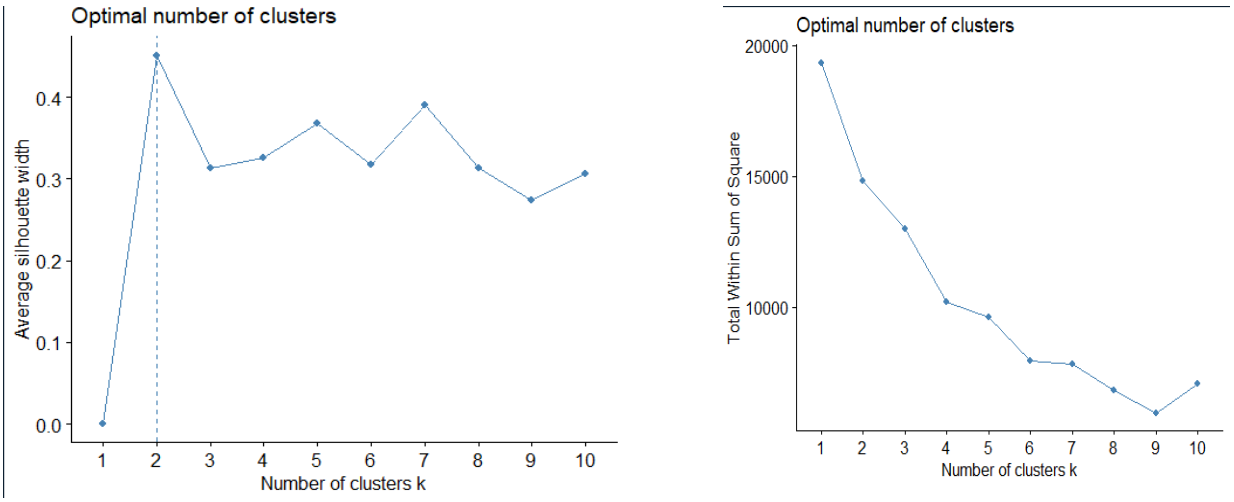
Partitionnement à 4 clusters des joueurs de tous les pays, parti du jeu avant la 3e régénération.

Puisque le partitionnement à quatre clusters n'aboutit pas, je décide d'employer les méthodes de sélection du nombre de clusters optimal. Il en existe trois : la méthode Elbow, le coefficient de Silhouette et la méthode Gap. Le premier calcule la somme totale des carrés, à savoir l'homogénéité interne des clusters pour chaque nombre de clusters considéré. Le coefficient de Silhouette, mesure la distance moyenne des points d'un groupe et celle entre les points de plusieurs groupes (à savoir l'homogénéité interne des clusters et la distance maximale entre les clusters). La statistique de Gap calcule la variation dans un cluster et la compare à la valeur statistique dans un modèle sans cluster, où nous n'aurions qu'un seul groupe de joueurs. Le principe de ces méthodes consiste à afficher les résultats de chacune sous forme de graphique pour voir à quel nombre de clusters le clustering est optimal.

La méthode Gap ne convergera jamais dans nos analyses. Malgré le fait d'avoir renseigné un nombre d'itérations, l'algorithme n'ira pas au-delà des 10 itérations car aucun partitionnement stable ne sera trouvé, quel que soit le jeu de données. Une autre particularité de l'emploi de ces méthodes est qu'elles renvoient toutes des nombres de clusters très différents, et ce pour chaque jeu de données. Le coefficient de Silhouette renvoie pour chacun un nombre optimal de 2, alors que l'indicateur d'Elbow renvoie entre 7 et 9 clusters selon le jeu. Une telle différence dans le nombre optimal de clusters indique en général une incapacité à partitionner correctement due aux données. A nouveau, voici ci-dessous l'exemple du jeu de données de l'ensemble des pays avec les joueurs partis avant la 3ème régénération : les graphiques du coefficient de Silhouette et de la méthode Elbow sont présentés afin de montrer la grande différence dans le nombre optimal de clusters. Les deux graphiques ne se lisent pas de la même manière. Celui de Silhouette, à droite, représente le coefficient de Silhouette en ordonnées : celui-ci doit être maximisé et il faut donc chercher le nombre k de clusters (en abscisse) dont le coefficient correspondant est le plus élevé (ici, k=2). A l'inverse, l'indicateur d'Elbow se base sur la somme totale des carrés, soit l'homogénéité de tous les clusters, et doit donc être minimisé. On recherche alors le nombre k pour la somme des carrés la plus basse (ici k=9).

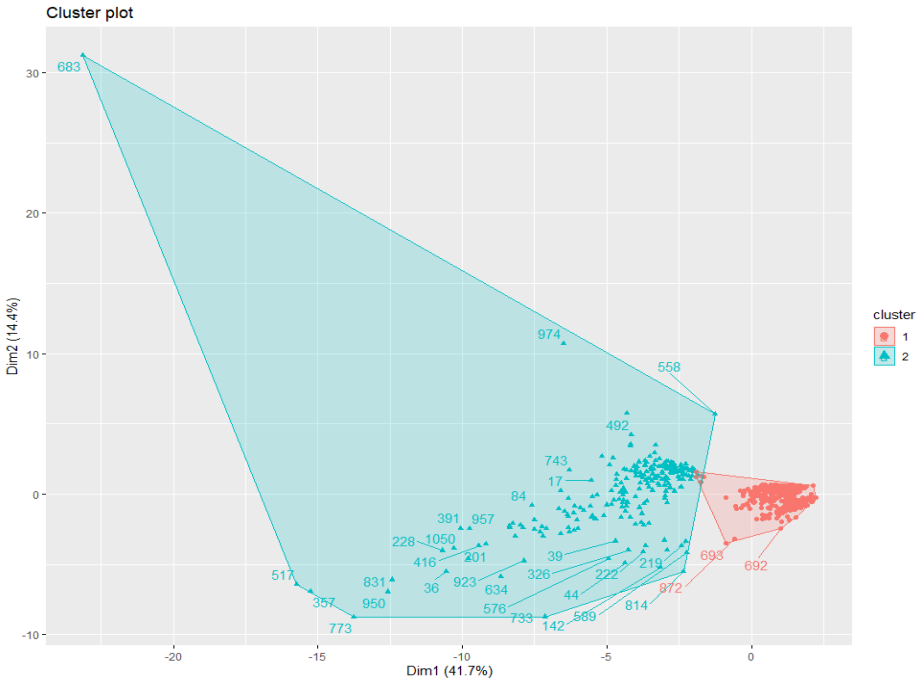
Graphique de la méthode Silhouette et graphique de la méthode Elbow.

Jeu de données : Tous pays, joueurs quittant avant 3ème régénération

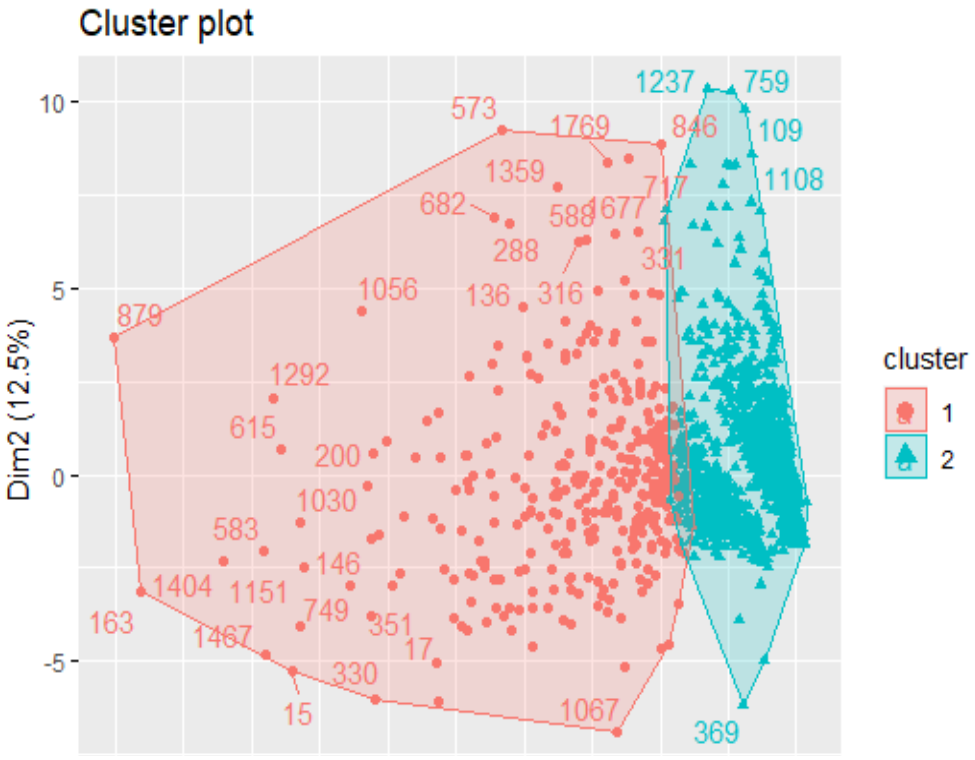


Intéressée par les critères d'homogénéité et de distance inter-cluster (vu le niveau de chevauchement de mon clustering), je retiens la méthode Silhouette plutôt. D'ailleurs, avoir 7 à 9 clusters par jeu de données me paraît peu souhaitable. C'est un trop grand nombre de groupes pour une analyse fiable. Il y a risque de se perdre dans des caractéristiques marginales de joueurs.

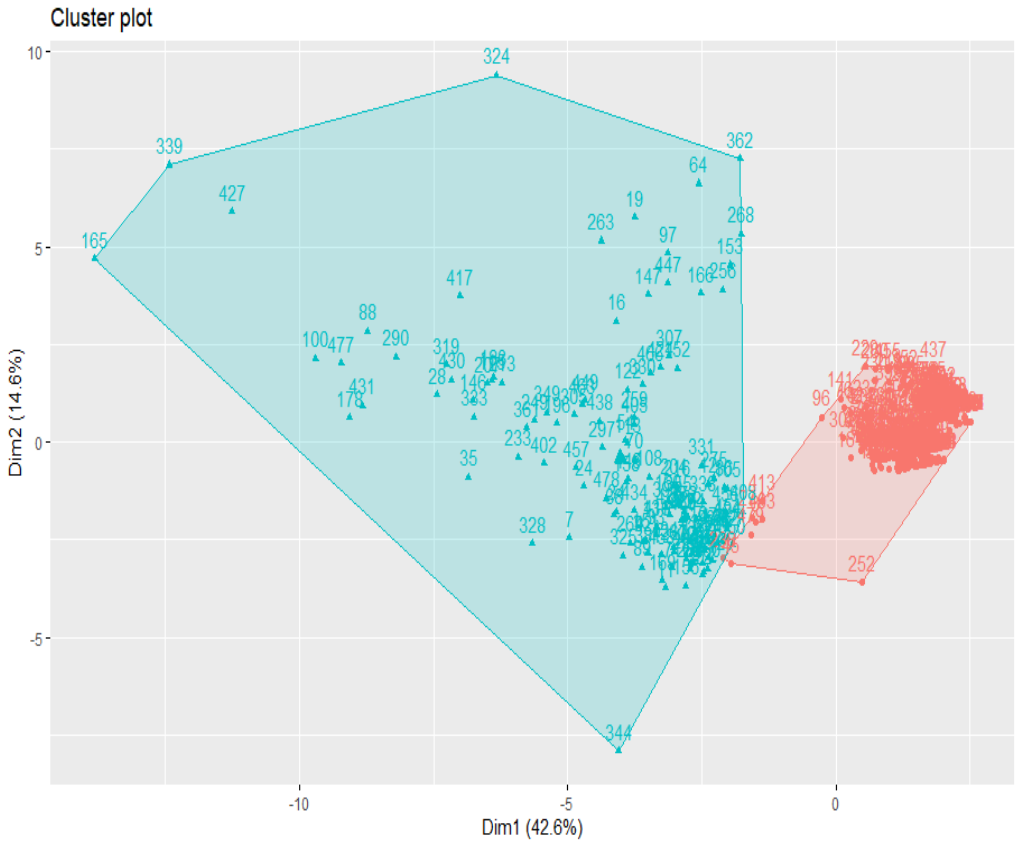
Je réalise donc un nouveau partitionnement en jouant sur 2 ou 3 clusters quand le résultat paraît être amélioré ainsi. Toutefois, le résultat sera toujours en faveur de deux clusters, quel que soit le jeu de données. Les clustering suivants se dessinent :



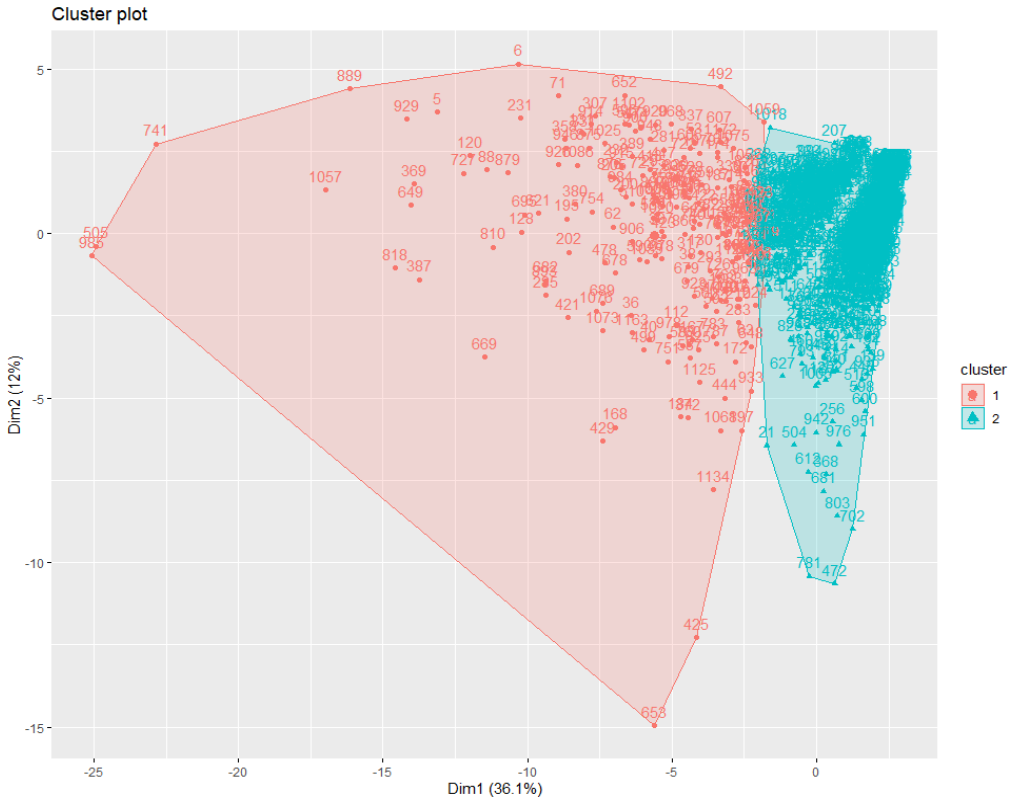
Jeu Tous pays – Joueurs quittant avant la 3ème régénération



Jeu Tous pays – Joueurs restant après la 2ème régénération



Jeu USA – Joueurs partant avant la 3ème régénération



Jeu USA – Joueurs restant après la 2ème régénération

Le chevauchement des clusters est toujours présent mais réduit au minimum. Cela serait-il le reflet d'un problème dans le clustering ? La pertinence du clustering serait remise en question par ces résultats. En effet, il semble que les données ne permettent pas de créer des groupes de joueurs aux caractéristiques distinctes. Nous allons toutefois tenter une interprétation.

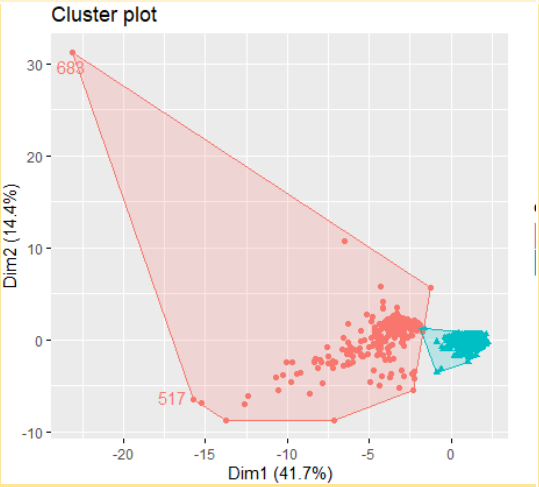
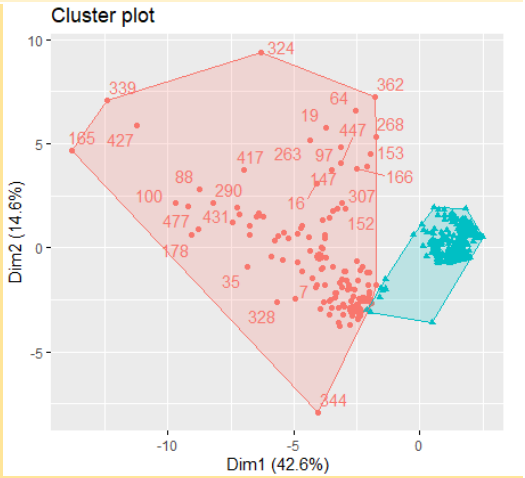
c) Comparaison des resultats du clustering

L'utilisation de l'algorithme des K-means montre que, quel que soit le jeu de données, deux clusters sont à retenir pour obtenir le meilleur partitionnement. Je vais ici comparer les résultats de l'ensemble des pays et ceux des Etats-Unis, pour comprendre si les joueurs américains influent sur les caractéristiques de tous les joueurs, ou si l'on peut discerner une influence des autres pays. Deux tableaux mettent en vis-à-vis les résultats : le premier pour les joueurs quittant rapidement le jeu et le second pour les autres.

Départs avant la 3ème planète régénérée

Le jeu des Etats-Unis semble mieux expliqué par le clustering : +1,8% de variance expliquée par rapport à l'ensemble des pays. L'homogénéité intra-cluster est meilleure. Les effectifs des clusters y sont répartis de manière plus tranchée : ¼ versus ¾, contre 1/3 versus 2/3 pour tous les pays. Les clusters se chevauchent sur quelques points avec les deux jeux.

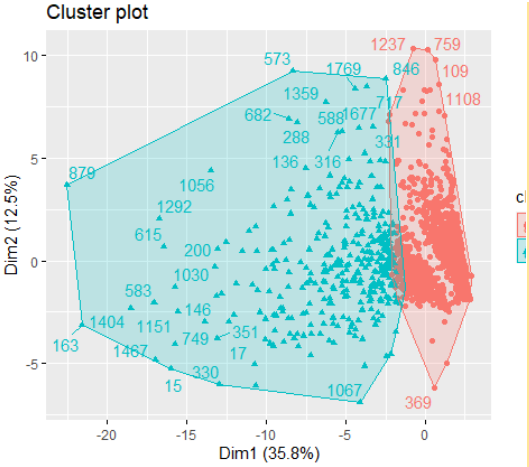
Les différences de répartition des effectifs et de variance me font penser que le comportement des joueurs des Etats-Unis n'explique pas entièrement la variation des valeurs dans le jeu de données de tous les pays. Le clustering semble montrer un effet des autres pays dans les résultats, notamment dans la différence d'effectifs entre les deux clusters.

	Tous Pays	USA
Nombre de clusters retenus	2	2
Effectifs des clusters	383 (35,93%) 683 (64,35%)	132 (27,1%) 355 (72,9%)
Homogénéité intra-cluster	5310 4222	1655 2656
% de variance expliquée par le clustering	56,16%	57,24%
Graphique des clusters		

Départs à partir de la 3e planète régénérée

En ce qui concerne les joueurs restant plus longtemps dans le jeu, la répartition des effectifs est identique (80/20%) mais s'inverse pour les Etats-Unis avec un premier cluster plus petit que le second. L'homogénéité intra-cluster est plus grande, essentiellement pour le second cluster. Toutefois, la variance expliquée par le clustering est légèrement inférieure (-0.82) lorsque l'on considère les Etats-Unis uniquement.

Au vu des effectifs et de la variance, il me semble que le clustering met en évidence l'influence des Etats-Unis sur l'ensemble des joueurs restant dans le jeu. Je ne constate pas de nette différence entre l'ensemble des pays et les Etats-Unis.

	Tous Pays	USA
Nombre de clusters retenus	2	2
Effectifs des clusters	344 (18,31%) 1535 (81,69%)	211 (17,51%) 994 (82,49%)
Homogénéité intra-cluster	14189 17861	13881 8829
% de variance expliquée par le clustering	48,28%	48,1%
Graphique des clusters		

Cette comparaison de l'ensemble des pays et des Etats-Unis me semble montrer une limite du clustering. Il ne paraît pas permettre de trouver des comportements parmi les joueurs. La différence entre l'ensemble des pays et les Etats-Unis n'est pas toujours flagrante non plus. Je poursuis donc par l'analyse des variables clivantes pour corroborer ces résultats.

d) VERIFIER LA PERTINENCE DES RESULTATS AVEC LES VARIABLES DISCRIMINANTES

Une fois les clusters définis, j'aimerais déterminer quelles variables sont vraiment discriminantes et lesquelles séparent les joueurs en clusters. Je commence par observer les moyennes des variables dans chaque cluster : si elles sont significativement différentes, cela signifie que la variable est discriminante. Cela permet aussi de vérifier qu'un nombre important de variables sont clivantes et que le clustering est fiable. Or, les indicateurs cités plus hauts (différence USA/tous pays, distance inter-clusters) semblent plutôt indiquer que le clustering n'apporte pas d'informations sur les joueurs.

Départs avant la 3ème planète régénérée

Je commence par comparer les joueurs ayant quitté le jeu rapidement, entre l'ensemble des pays et les Etats-Unis. Le tableau en annexe 1 récapitule les moyennes, ou centres pour chaque variable : il compare les moyennes des deux clusters du jeu de données de l'ensemble des pays. On observe que, sur l'ensemble des joueurs ayant quitté Star Shaman avant la 3ème planète régénérée, les variables pour lesquelles les valeurs sont les plus éloignées sont :

- Le ratio morts / planètes régénérées
- Le nombre de planètes régénérées
- Le temps moyen d'une session en minutes
- Le temps total de jeu

En effet, la distance entre les centres des clusters pour ces variables est égale à 1 ou est supérieure. Bien d'autres variables ont des moyennes relativement éloignées, avec une distance entre 0,4 et 0,6 : le nombre de morts au système 1, le montant moyen de monnaie, le nombre de planètes-événements jouées, l'évolution du montant moyen de monnaie, etc... Enfin, les variables suivantes ont des moyennes similaires (différence <0,2) et ne sont pas discriminantes à mon sens :

- Le score moyen
- Le nombre de clics sur les options
- Le nombre de sessions jouées
- L'évolution du score minimum entre la 1ère et la dernière partie
- Idem pour le score maximum
- Idem pour le score moyen

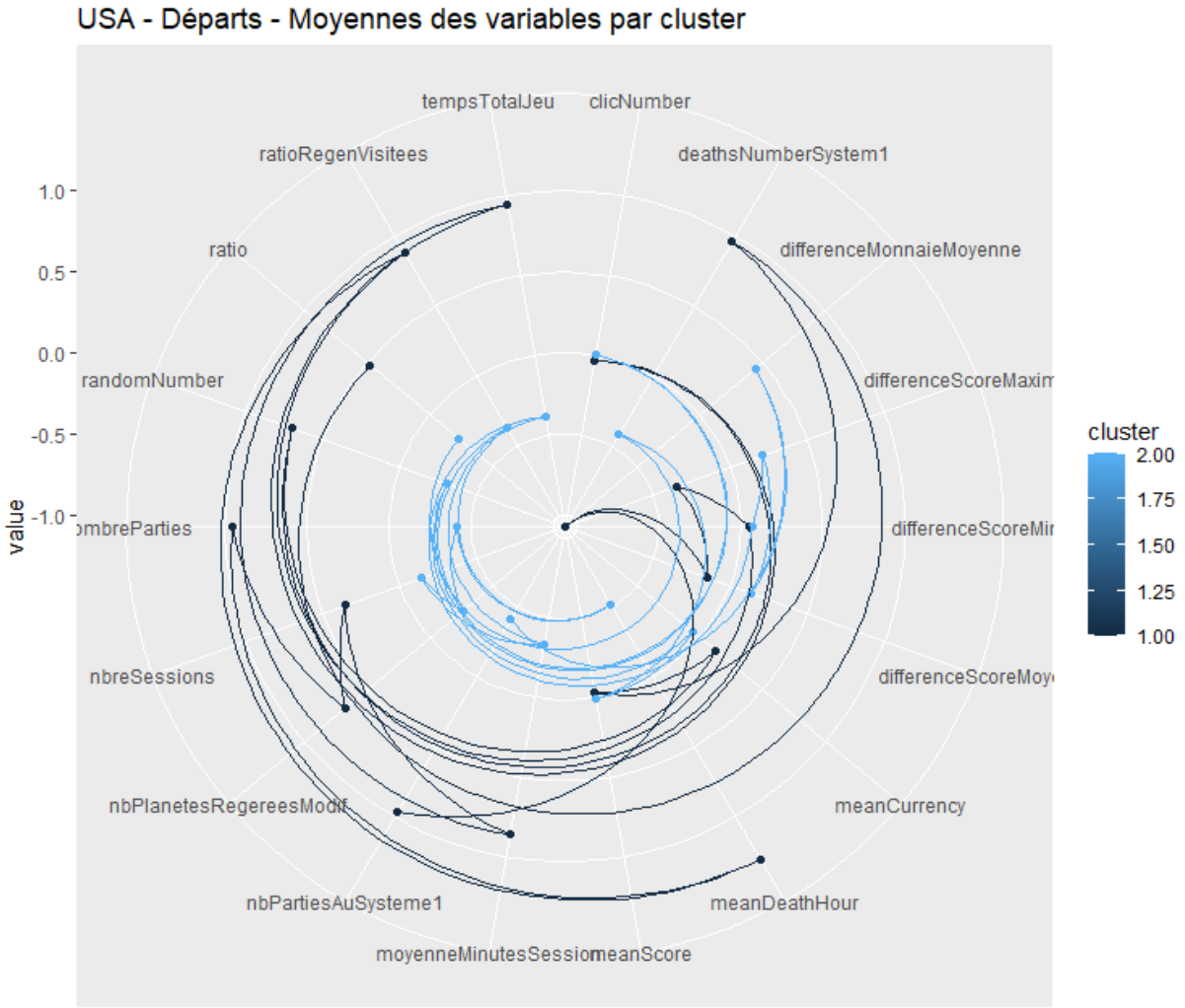
Si l'on compare aux joueurs américains (voir tableau des moyennes en annexe 1), on observe que les écarts des moyennes ne sont pas de même ordre, sauf pour le temps total de jeu et l'évolution du score minimum. Les variables clivantes sont plus nombreuses. Elles incluent d'ailleurs toutes celles remarquées sur l'ensemble des pays :

- Le ratio mort / nombre de planètes régénérées
- Le nombre de planètes-événements faites
- Le ratio de planètes visitées/régénérées
- Le nombre de morts moyen par heure
- Le nombre de parties
- Le nom de planètes régénérées
- La durée moyenne d'une session
- Le temps de jeu total
- L'évolution du montant de monnaie moyen
- Le nombre de parties au système 1

J'émets l'hypothèse que les joueurs des autres pays nivellent les différences, d'où certaines variables qui ne sont plus significatives dans la classification de l'ensemble des joueurs. En outre, pour les joueurs américains, les écarts entre moyennes des clusters des variables non significatives sont plus disparates : allant d'autour de 0 à 0.5 environ. Ce ne sont pas les mêmes que pour l'ensemble des pays, à l'exception du nombre de clics sur les options.

En ce qui concerne les joueurs américains, c'est l'évolution du montant de monnaie moyen qui offre le plus d'écart en moyenne entre les deux clusters, d'où le point du cluster 1 au centre du graphique radar ci-dessous. Ce graphique permet de comparer les deux clusters en plaçant chaque variable sur un axe. Attention, la légende est continue car l'extension permettant de la rendre discrète n'est pas encore accessible sur la dernière version de R.

On voit par ailleurs que les moyennes du cluster 2 sont beaucoup plus proches du centre de la grille de coordonnées polaires (entre 0 et 0.5). A l'inverse, les moyennes du cluster 1 sont dans les marges du graphique. La distinction visuelle entre les deux clusters est nette, ce qui traduit l'écart globalement plus grand entre les moyennes des clusters.



Départs à partir de la 3ème planète régénérée

D'après les tableaux ci-dessous, la majorité des variables semblent être clivantes dans le jeu de données sur l'ensemble des joueurs. On trouve facilement des écarts de plus de 1 entre les moyennes des groupes par rapport aux chiffres des joueurs aux départs rapides. Les variables les plus clivantes sont : le nombre de parties aux systèmes 2 et 3, le nombre de morts au système 2 et le temps total de jeu. Comparé

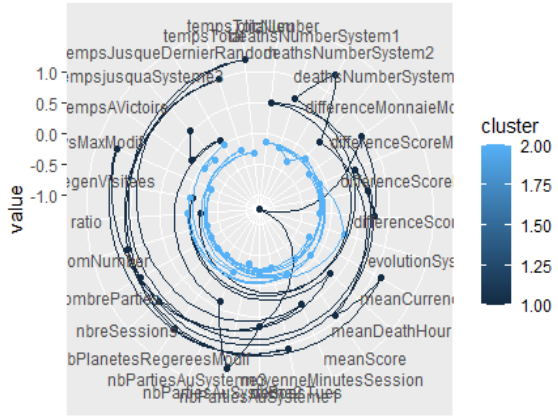
aux joueurs à départ rapide, le nombre de planètes régénérées est toujours un facteur de classification important, mais le temps moyen d'une session et le ratio morts/nombres de planètes régénérées ne le sont plus du tout.

Pour ce qui est des Etats-Unis, il est également très facile d'avoir des écarts supérieurs à 1 entre les moyennes des clusters, et ce sur les mêmes variables que l'ensemble des joueurs. Curieusement, les moyennes du clusters 2 sont très similaires entre les deux jeux de données. En revanche, celles du cluster 1 sont plus élevées dans le jeu des USA, à l'exception de la durée moyenne de session et du temps de jeu jusqu'à la dernière planète-événement qui sont légèrement en-dessous.

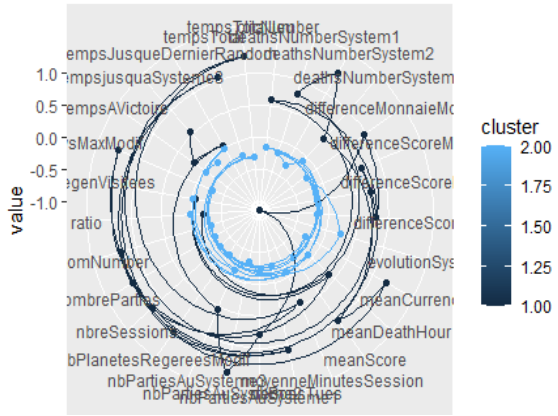
Par rapport aux joueurs à départs rapides des Etats-Unis, le ratio mort / nombre de planètes régénérées, le ratio de planètes visitées/régénérées, le nombre de morts moyen par heure et l'évolution du montant de monnaie moyen ne sont plus des facteurs de classification.

La similarité entre les deux échantillons est confirmée par les graphiques en radar ci-dessous, comparant les moyennes des groupes pour chacun des deux jeux de données. Ils semblent être les mêmes. Cela confirme notre première déduction d'après la variance et les effectifs des groupes : les joueurs américains influencent grandement le comportement des joueurs partant après la 2ème régénération. Toutefois, les moyennes plus élevées des joueurs américains semblent montrer un comportement plus affirmé, que les joueurs d'autres pays atténueraient.

Tous pays - joueurs restant - Moyennes des variables par cluster



USA - joueurs restant - Moyennes des variables par cluster



Le fait que l'écart entre moyennes est souvent supérieur à 1, montre que cet écart est similaire d'une variable à une autre. Se peut-il que cela révèle une importance égale des variables dans le clustering et confirme que les données ne permettent pas de tirer des informations sur les joueurs à partir du clustering ?

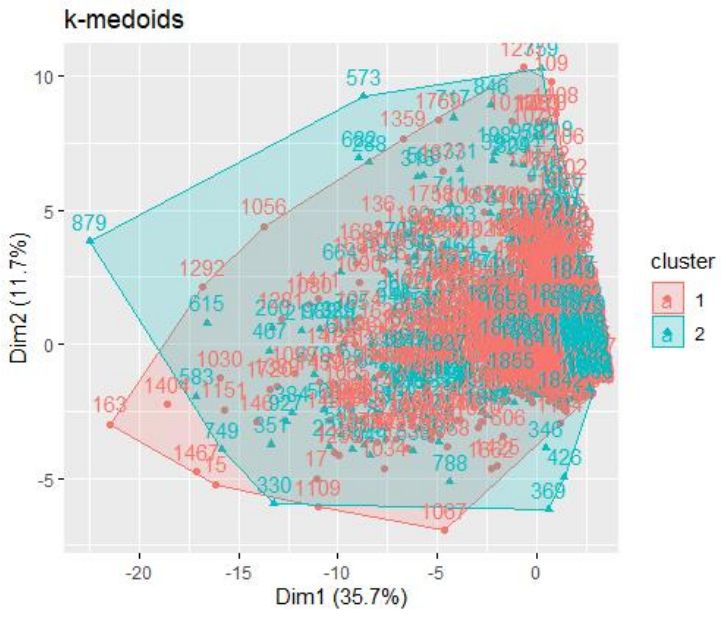
e) CONCLUSION SUR LE CLUSTERING

En rassemblant l'ensemble de nos résultats et de nos conclusions sur l'importance des variables, il semble que le clustering ne permette pas une analyse des comportements. Les données ne seraient pas clivantes et ne permettraient pas de distinguer des groupes de joueurs.

En effet, en voyant la différence moindre entre les variances expliquées des joueurs partant avant trois régénérations (56-57%) et ceux d'après (48%), il me semble que le clustering n'a pas de variables assez discriminantes pour établir un profilage solide. De plus, les clusters se chevauchent toujours malgré la suppression des outliers et le fait de relancer l'algorithme. La distance inter-cluster sur laquelle se base l'algorithme des k-means est très faible. Le chevauchement en est la manifestation graphique. Je pense qu'il n'est pas possible de séparer en groupes, les joueurs avec les données disponibles : ces variables ne sont pas discriminantes. On le voit notamment pour les joueurs partant à partir de la 3ème régénération ; les écarts entre les moyennes des clusters sont similaires d'une variable à une autre.

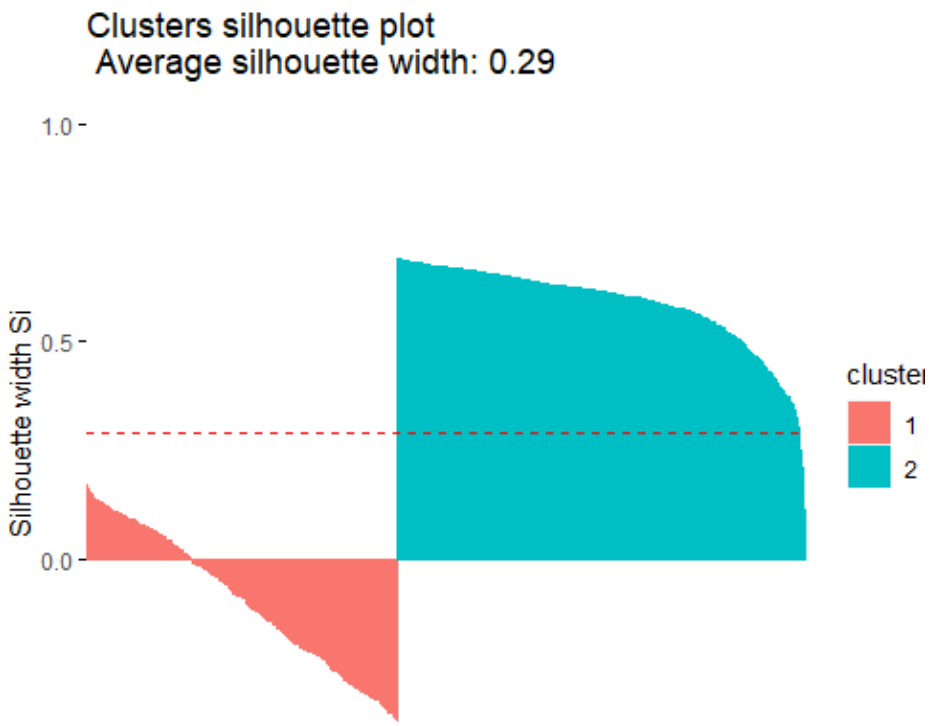
Afin de compléter notre propos, nous avons également essayé de profiler avec la méthode des k-médoïds qui est plus robuste aux outliers. La fonction PAM de R permet de la mettre en œuvre. Contrairement à l'algorithme des k-means, le centre d'un cluster n'est pas défini par la moyenne de chaque variable mais par un objet central. Le médoïde est un individu existant dans le jeu de données et dont la distance aux autres membres du cluster est minimale. À chaque itération de l'algorithme, et donc à chaque nouvel individu considéré, ce médoïde change pour aboutir au meilleur centre possible pour le cluster.

Je ne présente pas cette méthode ni ses résultats car le chevauchement des clusters est maximal. J'ai utilisé les jeux sans les outliers, en testant trois puis deux clusters. L'exemple le plus marquants des résultats fut obtenu avec le jeu des joueurs de tous les pays qui ont quitté après la 2ème planète régénérée. Il donne les deux clusters suivants :



On voit parfaitement bien l'impossibilité de distinguer des groupes de joueurs dans cet ensemble compact.

Bien plus, si l'on prend les joueurs américains uniquement, on obtient un coefficient de Silhouette négatif pour le cluster 1 dénotant un mauvais classement des individus de ce cluster. Le coefficient négatif révèle que les individus ont été mal classés en étant attribués au cluster 1.



Coefficient de Silhouette pour les clusters des joueurs américains, quittant le jeu à partir de la 3ème planète régénérée – méthodes des K-médoïdes.

Le clustering, avec notre partition des données entre les joueurs quittant avant la 3ème régénération et les autres, donne des résultats peu probants. Je ne pousse donc pas l'analyse des caractéristiques des groupes plus avant. Toutefois, en observant les graphiques des clusters, j'observe à chaque fois une distinction nette dans le nuage de points, qui n'est pas pris en compte comme séparation des groupes par l'algorithme des k-means. Peut-être faut-il utiliser une méthode plus poussée comme une de celles de la famille des SVM.

Je décide de me concentrer plutôt sur la question suivante : pourquoi les joueurs quittent-ils le jeu ? En effet, c'est ce qui nous avait conduits, au départ, à partitionner les joueurs par nombre de planètes régénérées. Nous allons donc nous concentrer sur les facteurs de départ du jeu. Pour cela, nous utilisons deux autres méthodes : la régression logistique et la forêt aléatoire.

4. ESTIMER L'IMPACT DES FACTEURS SUR LE FAIT DE QUITTER LE JEU

Suite aux résultats du clustering, j'ai abordé les joueurs de Star Shaman sous un angle nouveau. Puisque l'on ne peut pas retenir de groupes de joueurs dont nous aurions analysé le comportement, je me concentre sur la question de leur départ du jeu. En effet, nous avions établi en préambule de nos analyses que 35% des joueurs quittaient Star Shaman avant d'avoir atteint le 2ème niveau. La plupart d'entre eux ne dépassaient pas le tutoriel. Je souhaite donc comprendre ce fait et identifier les facteurs de départ des joueurs, quel que soit le moment de ce départ.

Cette section présente les méthodes retenues pour mener cette analyse ainsi que leurs résultats. J'utilise deux méthodes dont les objectifs et donc les techniques ne sont pas les mêmes. J'ai tout d'abord utilisé une régression logistique. Elle permet d'estimer l'impact des variables sur le départ des joueurs ou de comparer l'impact entre plusieurs modalités d'une variable : on obtient une probabilité, pour chaque variable, de rester joueur à Star Shaman par rapport à partir. J'ai ensuite utilisé une forêt aléatoire pour classer les joueurs entre ceux partant et les autres. Cette méthode permet d'identifier les variables clivantes et donc aussi celles qui ont un impact sur le fait de rester jouer.

a) PREPARATION DES DONNEES : DE NOUVELLES VARIABLES

Pour commencer, j'ai dû créer de nouvelles variables. J'ai tout d'abord créé la variable binaire « restant » avec comme modalité 0 si le joueur a quitté le jeu et 1 s'il a continué de jouer. Il s'agit de la variable-réponse par laquelle les joueurs seront classés en fonction de l'impact des autres variables. Pour ce faire, j'ai téléchargé le 19 mai 2021 le dernier horodatage de chaque joueur (« FIELDEVENTTIMESTAMPLAST »). Il s'agit du jour et de l'heure du dernier événement enregistré. Ainsi, savais-je avec certitude si le joueur avait bel et bien quitté le jeu après le 26 mars, date à laquelle s'arrêtaient les données de jeu de notre échantillon de 3083 joueurs. J'ai choisi de ne pas prendre l'horodatage du dernier événement de jeu enregistré, parce que j'aurais dû laisser passer une période de deux semaines pour être sûre que le joueur a réellement quitté le jeu. En effet, certains joueurs se reconnectent au jeu après 2 voire 3 semaines seulement. Seuls les joueurs ayant fait leur dernière action jusqu'au 13 mars inclu, auraient été classés comme ayant quitté le jeu. Avec les données de mai, nous savons réellement si la personne est partie, au jour près. Ainsi une personne réellement partie après avoir joué le 20 mars n'est pas classée restante. De même avec une personne ayant son dernier horodatage le 26 mars mais ayant quitté le jeu ensuite. Savoir avec certitude presque deux mois après si les joueurs étaient encore en jeu ou non est donc précieux. Toutefois, il faut préciser d'emblée que très peu de joueurs avaient continué à jouer : seuls 131 sur 2975 (taille de notre échantillon après nettoyage, voir plus bas) ont joué après le 26 mars.

J'ai également inclus la variable du pays laissée de côté dans le clustering. L'analyse porte en effet sur l'ensemble des joueurs cette fois, sans distinction de temps de jeu ou de pays. Aussi la variable pays est-elle devenue importante : elle permet de voir si des joueurs d'une certaine nationalité ont tendance à jouer plus longtemps. A partir de la variable pays, j'ai créé cinq autres variables de nationalité. Les joueurs américains sont en effet la très grande majorité (57% environ) et je souhaite continuer de monitorer l'impact de ces joueurs. Une variable booléenne permet donc de savoir si le joueur est américain (1) ou non (0). D'autres pays ont un nombre élevé de joueurs par rapport à la masse : la France, l'Angleterre, le Canada. Ils ont tous plus de 5% des joueurs et me semblaient à ce titre impactant. Chacun a donné lieu à un booléen. La Chine est légèrement sous le seuil des 5% mais rassemble plus de joueurs que la masse des autres pays. J'ai donc créé un booléen pour elle également.

Par ailleurs, j'ai créé plusieurs variables dichotomiques dite d'étape (voir liste en annexe 2), pour savoir précisément si partir à une étape du jeu est significatif par rapport à une autre. Elles servent à comparer les joueurs partis avant et après l'étape en question. De plus, il est possible de comparer la significativité de deux de ces variables.

Enfin, j'ai retiré du jeu de données les variables booléennes marquant si le joueur a fait toutes les planètes ou non aux systèmes 2 et 3. Comme nous l'avions indiqué pour le clustering, je me suis aperçu après calcul qu'aucun joueur n'avait rempli la condition. Ces variables à une modalité ne peuvent donc pas servir dans les analyses. Quant aux variables de temps qui avaient été créées pour remplacer les booléens inutilisables avec la méthode des k-means, elles ont été retirées du jeu également pour ne pas faire double emploi.

Dans un second temps, j'ai également vérifié les corrélations entre les variables. Une des hypothèses fondamentales de la régression logistique est que les prédicteurs ne sont pas colinéaires. Ceci peut se vérifier en calculant les coefficients de corrélation de Pearson. Cela permettra de ne garder que les variables non colinéaires et d'utiliser un modèle valide de régression. La forêt aléatoire n'est, quant à elle, pas sujette à erreur si les variables sont colinéaires.

Je vérifie les corrélations à l'aide d'un corrélogramme (voir annexe 2). Ce graphique permet d'afficher à la fois les coefficients entre les variables deux à deux, et une croix si la relation entre deux variables n'est pas significative. Cela signifierait que l'on ne peut statuer sur le fait qu'il y a ou non corrélation entre les deux. Je garde un seuil de significativité assez bas et classique, c'est-à-dire de 5% ; il n'est pas nécessaire de l'augmenter ni de le diminuer dans le domaine du jeu vidéo. De plus, j'ai choisi de voir une corrélation entre variables lorsque le coefficient est de 0.6 et plus. En effet, je ne souhaite éliminer qu'un faible nombre de variables et ne retiens que les corrélations fortes.

Le corrélogramme montre des corrélations fortes entre le nombre de parties, le nombre de sessions, de planètes régénérées, le temps de jeu total, le nombre de boss tués. Les corrélations les plus importantes sont entre le nombre de boss et celui de planètes régénérées, et entre ce dernier et le temps total de jeu (plus de 0.9 !). Le système maximum atteint est aussi corrélé au nombre de planètes régénérées, de boss tués et au temps de jeu total. Par conséquent, si je souhaite garder le nombre de planètes régénérées (par rapport au nombre de boss notamment), alors je dois retirer l'ensemble des autres variables mentionnées.

Je garde, outre la variable-réponse, vingt-et-une variables finalement. La première régression n'a pas donné de résultats satisfaisants. Or, une autre des hypothèses fondamentales de la régression logistique suppose que les données ne contiennent pas de valeurs extrêmes. Et comme nous avions identifié les outliers lors des recherches de clustering, je les considère comme des individus à enlever et les supprime donc de mon jeu de données. N'ayant à nouveau pas obtenu de résultats satisfaisants, j'en ai déduit que mon choix de variables n'étais pas optimal. J'avais peu de prédicteurs significatifs et même un prédicteur ne permettant pas de générer des *odds*.

J'ai donc finalement utilisé une méthode de choix des variables dite *Stepwise* pour déterminer les prédicteurs à conserver. J'ai ensuite refait une régression logistique.

b) MODELE DE REGRESSION LOGISTIQUE

Dans cette section, nous détaillerons la régression logistique employée pour trouver les facteurs poussant à rester dans le jeu. La méthode de la régression logistique permet d'estimer l'impact de facteurs sur une variable à deux modalités. Elle renvoie des *odds* c'est-à-dire des ratios de probabilité entre le nombre de joueurs restés dans le jeu et celui des joueurs partis. Par transformation, on obtient les coefficients de chaque variable et donc l'impact sur le fait de continuer à jouer ou non.

La méthode stepwise permet d'utiliser un algorithme sélectionnant les variables à retenir dans un modèle. Il existe plusieurs formes et je choisis la bidirectionnelle. C'est une combinaison des formats *forward* et *backward*. Le premier lance les calculs en ajoutant une variable à chaque itération : le résultat renvoie le modèle le plus optimal et donc les variables à employer. Le second fait l'inverse : à chaque itération une variable est enlevée. La méthode bidirectionnelle fait les deux à chaque itération en enlevant les variables qui ne sont plus significatives et en ajoutant d'autres. Elle prend en entrée un modèle de régression avec l'ensemble des variables.

Cependant, lancer la méthode sur l'ensemble des variables génère plusieurs problèmes. Il m'a fallu déclarer un nombre d'itérations de minimum 75 pour que le modèle puisse converger. De plus, certaines variables ont des modalités avec peu d'individus. Par exemple, le fait qu'un joueur arrive au système 3 a des effectifs déséquilibrés : parmi les joueurs restant dans le jeu (variable « restant » =1), seuls deux sont arrivés au système solaire 3. Ainsi donc, rien que pour lancer la méthode stepwise, je dois retirer les variables problématiques. Je remets mes variables choisies après analyse de corrélation dans le modèle et introduit les autres une à une. Ce sont finalement les variables des pays et du dernier horodatage qui posent problèmes.

Une fois la méthode stepwise lancée, j'obtiens en sortie les variables à retenir (voir annexe 3): le nombre de clics, le nombre de boss (un à la fin de chaque système solaire) tués, le nombre de sessions, la différence du montant moyen de monnaie entre les 1ère et dernière partie, les nombres de parties aux systèmes 2 et 3, le fait de partir avant de régénérer la lune du tutoriel ou non, le fait de partir avant de régénérer la planète du tutoriel ou non, les variables de nationalités américaine, anglaise et canadienne.

Les paramètres de la régression sont les suivants : je prends 0.1 comme seuil de significativité (le seuil usuel étant de 0.05) car j'accepte d'avoir une marge d'erreur plus grande. Ainsi, toute p-value en-deçà de 0.1 confirme la significativité de la variable dans le modèle et le fait que celle-ci soit un facteur de continuer à jouer ou non. J'avais aussi normalisé les données numériques afin d'éviter un biais lorsque les unités ne sont pas les mêmes. Sept variables sont significatives au seuil de 0.1 (voir annexe 3) : le nombre de boss tués, le nombre de sessions, la différence de monnaie, les nombres de parties aux système 2 et 3, le fait d'être américain ou anglais.

En calculant l'exponentiel des coefficients de la régression, j'obtiens les *odds ratio* de ces variables. Un *odds ratio* est la probabilité de rester par rapport à partir, et ce, induite par une variable, toutes les autres étant égales par ailleurs. Lorsque la variable est discrète, tel le nombre de boss tués, l'*odds ratio* permet de calculer la probabilité de rester par rapport à partir du jeu pour l'ensemble des joueurs. Si la variable est plutôt booléenne, le coefficient donnera la probabilité des joueurs à modalité 1 par rapport à ceux dont la modalité est 0. Par exemple, le fait d'être américain (modalité 1) réduit fortement la probabilité de rester par rapport à partir. Il faut enfin noter que je ne calcule pas les intervalles de confiance des ratios, car R les calcule au seuil de 5% par défaut, alors que notre seuil est à 10%.

Les ratios des variables significatives de ce modèle sont les suivants :

names	x
(Intercept)	0.0537231
nbBossTues	0.6763479
nbreSessions	1.4568970
differenceMonnaieMoyenne	1.1681317
nbPartiesAuSysteme2	1.6692284
nbPartiesAuSysteme3	0.8386435
americain1	0.5928409
anglais1	0.3583083

Trois variables augmentent la probabilité de rester par rapport à quitter le jeu : le nombre de sessions, la différence de monnaie entre la 1ère et la dernière partie, le nombre de parties au système solaire 2. Leurs ratios sont au-dessus de 1. Celui de la monnaie en revanche est très proche de 1, ce qui montre une probabilité presque égale entre rester et quitter Star Shaman. Les deux autres montrent clairement augmentation de la probabilité de rester dans le jeu.

A l'inverse, les autres variables diminuent la probabilité de rester dans le jeu par rapport à partir. Leurs ratios sont inférieurs à un. Le fait d'être anglais est particulièrement impactant, au vu du ratio proche de zéro : les Anglais (notés « anglais 1 ») ont une probabilité de rester en jeu beaucoup plus faible que de partir. L'interprétation est la même pour la nationalité américaine. Il est curieux de constater que parmi les variables de nationalité, ce soient deux pays anglophones qui ressortent : cela peut-il être lié à la langue du jeu ? Il faudrait comparer le pays à la langue utilisée pour jouer.

Pour terminer cette analyse, je vérifie la robustesse de ce modèle. Tout d'abord, je réalise une analyse de la déviance de celui-ci. En effet, la déviance est l'équivalent de la variance d'un modèle linéaire. Elle permet de mesurer l'écart entre les valeurs réelles et celles du modèle. Une déviance importante rend le modèle caduc. La régression logistique sous R compare notre modèle à celui, théorique, réduit à la constante. Les résultats montrent que notre modèle a une déviance moindre par rapport à celui réduit à la constante. En outre, la p-value est très petite. Notre modèle semble donc expliquer beaucoup plus le départ des joueurs que celui réduit à la constante.

Une autre manière de vérifier que la dispersion des données est bien dans la tranche acceptable par la théorie est de calculer s'il y a sur-dispersion. Si cela est le cas, alors les erreurs standards des paramètres seront sous-estimées et les conclusions erronées. Cela se calcule sous la forme d'un ratio de la déviance résiduelle sur le nombre de degrés de liberté du modèle. Il n'y a pas de seuil au-delà duquel il est certain qu'il y a sur-dispersion, mais l'on suspecte la chose si le ratio dépasse 1. Or, ici le ratio est de 0.32.

Par ailleurs, il faut aussi vérifier les hypothèses fondamentales de la régression logistique. Comme nous l'avions souligné en début d'analyse, les prédicteurs ne doivent pas être colinéaires. Bien que les variables identifiées par le modèle Stepwise respectent cette condition, je calcule les coefficients de chaque variable pour vérifier. Je ne trouve pas de colinéarité. Quant à l'hypothèse de lien linéaire entre les prédicteurs et les *log odds*, il s'agit de la dernière hypothèse fondamentale à vérifier. Les *log odds* sont les logarithmes des *odds*, c'est-à-dire les logarithmes des fractions nombre de joueurs restant dans le jeu / nombre de joueurs quittant le jeu. On peut vérifier rapidement cette condition en utilisant le modèle pour faire une prédiction sur un échantillon de test. Les probabilités ainsi générées servent à calculer les *log odds* qui sont ensuite mis en graphique (voir annexe 3). On observe qu'aucune variable indépendante n'a de relation linéaire aux *log odds* : il faudrait donc modifier les variables pour améliorer le modèle de régression.

Enfin, on observe que la prédiction avec notre modèle ne permet pas d'identifier les joueurs restant dans le jeu : ils sont systématiquement classés comme étant parti. Ceci m'amène à tenter une autre méthode : la forêt aléatoire.

c) UTILISATION D'UNE FORET ALEATOIRE

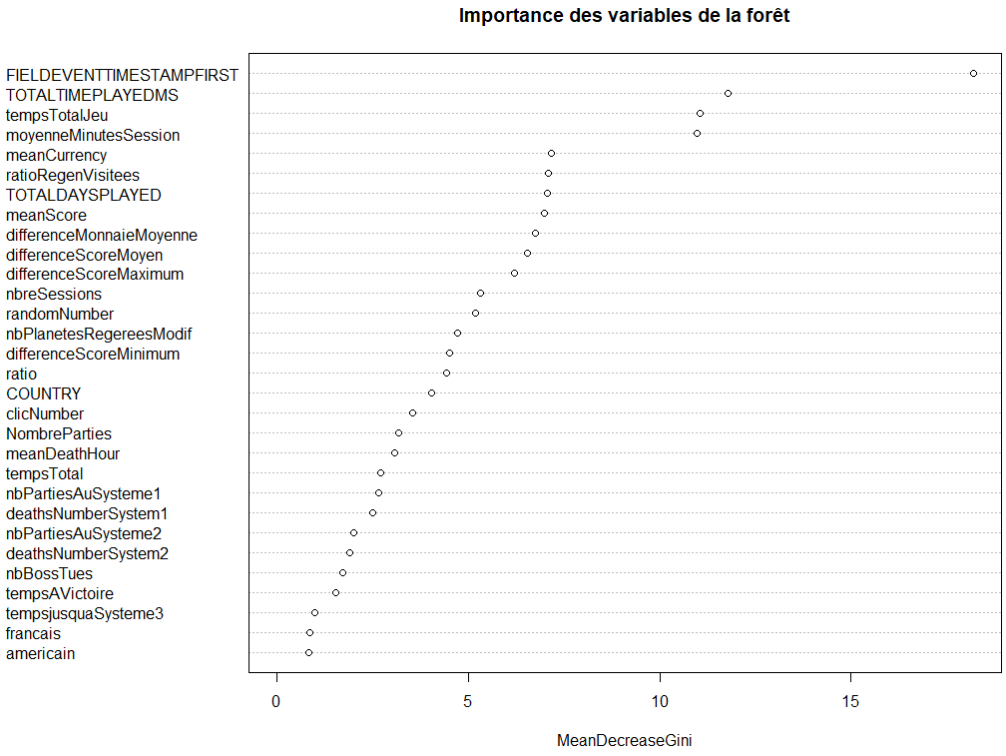
La forêt aléatoire permet de classer les individus en fonction des variables prédictrices. Il s'agit d'un ensemble d'arbres de décision. Chacun fait un nombre d'itérations durant lesquelles il scinde l'échantillon des joueurs en fonction de critères successifs. Au fur et à mesure des itérations, des branches se construisent dans lesquels les sous-groupes sont scindés en groupes toujours plus petits. Un arbre de décision possède un paramètre pour déterminer le nombre d'itérations maximum. Cependant un arbre seul est très sensible aux outliers et au sur-apprentissage. Ce dernier arrive lorsque le modèle s'adapte très bien aux données qu'on lui donne lors de sa construction et n'est plus capable de tourner avec des données différentes.

Afin d'éviter cela, j'ai opté pour une forêt qui, en multipliant les arbres, classe les joueurs en prenant la moyenne de l'ensemble des résultats des arbres. Par exemple, si 310 arbres, sur les 500 par défaut du modèle, classent un joueur comme étant resté dans le jeu, la forêt retient le choix de la majorité. C'est une première manière d'éviter les biais. En outre, chaque arbre a un échantillon d'individus et une partie des prédicteurs en entrée. Il est ainsi possible de multiplier les combinaisons joueurs/variables.

La forêt standard se compose de 500 arbres et chacun a sept variables observées dans notre cas. Nous avons séparé notre jeu de données en deux sous-échantillons : l'échantillon d'entraînement du modèle avec 70% des individus et celui de test avec les 30% restant. Toutes les variables du jeu initial sont reprises car la forêt n'est pas sensible aux corrélations ou aux variables dont une modalité est quasiment absente.

D'après les résultats, on observe plusieurs choses. Tout d'abord, l'erreur Out-Of-Bag (OOB) indique que 4,37% des individus OOB sont mal classés. Cette erreur représente le taux de mauvais classements des individus qui n'ont pas été pris dans les échantillons de chacun des arbres. Ils servent en quelque sorte d'échantillon de test à l'intérieur de l'échantillon d'entraînement. Ensuite, la matrice de confusion (voir annexe 4) sur les données d'entraînement montre que le classement des individus n'est pas optimal. Tous les joueurs ayant quitté le jeu sont classés comme tels, mis à part un individu. Mais aucun des joueurs restant dans le jeu n'est classé comme tel.

Je cherche ensuite à déterminer les variables impactant la classification, c'est-à-dire les facteurs de départ des joueurs. On peut les calculer en utilisant la moyenne des indices de Gini pour chaque variable sur l'ensemble des arbres. L'indice de Gini est une mesure du fait qu'une variable réduit le chaos dans le jeu de données et permet le classement. Une variable dépasse les autres de loin sur le graphique ci-dessous : c'est le premier horodatage. Le jour et l'heure du premier événement enregistré du joueur déterminerait son départ ou non. Viennent ensuite le temps total de jeu en millisecondes et minutes, et le temps moyen d'une session. On voit ici que les deux premières sont colinéaires (seule l'unité change) mais ce n'est pas un problème pour la forêt. On voit surtout dans ce graphique que l'ensemble des variables a un indice en-dessous de 20, ce qui est relativement peu.



On voit aussi par ces résultats que la forêt ne permet pas de prédire le départ ou non des joueurs. Les joueurs restants ne sont pas correctement classés par la forêt et seules les variables de temps, dont deux ont la même signification, permettent de discriminer nos joueurs. Par curiosité, j'essaye une approche sans outliers pour confirmer que cela n'impacte pas la forêt. Les résultats sont similaires. L'erreur Out-Of-Bag est de 4,02% donc très légèrement inférieure. Les joueurs ayant continué à jouer ne sont pas détectés. Quant aux variables d'importance, ce sont les mêmes.

Pour achever la comparaison, je tente également une forêt aléatoire à partir des variables identifiées comme non corrélées et que j'avais utilisées pour la première régression. Lorsque j'utilise le jeu de données nettoyées des variables (même si le lien entre les variables n'a pas d'importance pour un arbre, répétons-le), l'erreur de classification approche les 4,5% et les individus restés à jouer ne sont pas correctement classés non plus. Seule une variable sort du lot : le temps de session moyen et son indice est de 25 environ. Sans les variables de temps, j'observe que d'autres forment le groupe des trois suivantes : le ratio planètes visitées/ planètes régénérées, la différence du montant de monnaie moyen entre la première et la dernière partie, et la différence du score moyen. Leurs indices restent sous 20 comme précédemment.

En conclusion de cette troisième partie, nous avons vu l'ensemble des analyses menées, à la fois pour classer les joueurs et déterminer des comportements de chaque groupe, et pour trouver les facteurs de départ du jeu. Aucune des méthodes n'est réellement satisfaisante : il est difficile de distinguer des groupes ou des facteurs de départ avec les données actuelles. En effet, le clustering ne permet pas de créer des groupes distincts et homogènes. Il semble donc que nos joueurs ne puissent être répartis en groupes aux paramètres distincts. L'homogénéité interne faible des groupes tend à montrer que leurs membres n'ont pas forcément de caractéristiques assez communes pour être regroupés ensemble.

Quant à la régression logistique, elle ne permet en l'état de classer les joueurs restant. L'hypothèse fondamentale de lien linéaire entre les prédicteurs et les *log odds* est rejetée. Toutefois, l'analyse de la variance montre que le modèle avec les variables issues de la méthode *Stepwise* permet d'obtenir un

meilleur résultat qu'un modèle sans variables. Les résultats seraient donc des premières pistes de travail. En retravaillant les variables, il serait peut-être possible d'obtenir un meilleur résultat.

Enfin, la forêt aléatoire ne permet pas plus de classer un joueur restant dans le jeu. Cependant, la méthode n'est invalidée en rien. Il serait peut-être utile d'agrandir l'échantillon pour obtenir plus de joueurs restés dans le jeu et ainsi voir si l'on obtient un meilleur classement. Les résultats sont donc de premières pistes de travail également.

CONCLUSION

Ce mémoire avait pour objectif de présenter un travail d'analyse effectué sur les joueurs de Star Shaman, le jeu en réalité virtuelle du studio Ikimasho. Nous avons présenté tout d'abord le contexte professionnel en décrivant l'entreprise et les missions durant mon apprentissage au studio. Nous avons ensuite abordé un état de l'art de l'analyse de données dans le jeu et du clustering dans le jeu vidéo en particulier. Nous avons notamment vu que l'analyse de données dans le jeu vidéo servait aujourd'hui à retenir les joueurs et, dans le cas très répandu des jeux *free-to-play*, à leur faire dépenser de l'argent dans le jeu. Nous avons aussi présenté l'angle de la recherche, notamment dans les *jeux sérieux* et la réalité virtuelle dans le domaine de la santé. Il ressort de tout cela que les données de jeu sont très souvent accompagnées d'autres types de données, telles que celles de réseaux sociaux ou socio-démographiques ou psychologiques. Ces données peuvent être obtenues par des questionnaires. L'analyse des données de Star Shaman n'utilisait pas ces données supplémentaires et ne pouvait se baser que sur celles récoltées pendant les sessions de jeu. Enfin, la littérature montrait également que le clustering était une méthode de machine learning très employée pour déterminer des groupes de joueurs et comparer leurs cheminements dans un jeu ; un objectif différent poussait aussi certains auteurs à l'utilisation de régressions. La méthode des k-means en particulier était très décrite, de même que les analyses factorielles, l'*archetypal analysis* et les méthodes serveur-machine. Cela m'a permis de décrire trois d'entre elles : les k-means, l'analyse en composante principale et l'*archetypal analysis.*

La dernière partie de ce travail exposait les méthodes employées, et leurs résultats, pour déterminer des caractéristiques des groupes de joueurs de Star Shaman et les facteurs les poussant à continuer à jouer. A la suite des conclusions de revues de la littérature professionnelle, j'ai décrit mes interrogations quant aux données à utiliser dans le clustering et la mise en place du pipeline de production et traitement des données de jeu. Une première conclusion du clustering a montré que cela ne donnait pas de résultats probants. La méthode des k-means ne permettait pas de déterminer des groupes de joueurs, ceux-ci semblant être trop uniformes. Néanmoins, les graphiques de clusters montraient des fractures dans les nuages de points. Il se peut qu'en utilisant une méthode plus poussée, de la famille des SVM, on obtienne des groupes de joueurs définis en fonction de ces fractures. L'identification des facteurs poussant les joueurs à continuer de jouer à Star Shaman n'a pas donné de résultats définitifs non plus. En effet, l'hypothèse fondamentale de linéarité entre les prédicteurs de la régression linéaire et les *log odds* (c'est-à-dire des logarithmes des résultats de la régression) n'est pas vérifiée. Il faut donc retravailler nos variables pour améliorer le modèle. De plus, la prédiction des départs avec notre modèle ne permettait pas de reconnaître les joueurs continuant à évoluer dans le jeu. En revanche, l'analyse de la variance a montré que le modèle était robuste par rapport à un modèle réduit à la constante. Introduire ces variables permet donc une analyse du départ des joueurs mais il faut retravailler celles-ci pour affiner les résultats et pouvoir s'appuyer dessus. Quant à la forêt aléatoire, elle montrait une faible influence de quatre variables et ne parvenait pas à prédire le fait de rester jouer. Il semble qu'augmenter l'effectif de notre échantillon de joueur pourrait augmenter le nombre de ceux restant dans le jeu et améliorer les résultats. Les variables désignées comme clivantes par la forêt sont donc une première piste de réflexion. Il me semble qu'il serait également intéressant de retirer les joueurs ayant gagné le jeu, même s'ils sont peu nombreux. En effet, nous pourrions ainsi comparer les autres joueurs et voir les facteurs qui poussent à rester jouer. Il me semble en effet que comprendre les facteurs de départ est intéressant pour un joueur qui n'a pas encore gagné : va-t-il rester pour essayer de gagner et si oui, pourquoi ?

Les méthodes utilisées ont donc conduit, chacune avec son objectif, au même résultat : les données, telles qu'utilisées actuellement, ne permettent pas de tirer des conclusions probantes quant aux joueurs, à leurs cheminements dans Star Shaman ou encore aux éléments qu'ils apprécient dans le jeu. J'en retire donc deux hypothèses. La première concerne l'impossibilité de distinguer des groupes de joueurs : les données employées ne permettent pas de les discriminer. Soit les joueurs de Star Shaman seraient uniformes et

auraient des caractéristiques trop communes, soit leurs différences se situeraient sur d'autres éléments que ceux suivis. J'avais notamment attiré l'attention sur le fait que le geste – et donc le mouvement si important en VR – n'était pas suivi. La seconde hypothèse de ces enseignements est que les raisons du départ rapide de certains joueurs ne sont pas liées aux éléments suivis dans le jeu. Elles ne résident donc pas dans les types de planètes, un bug à une étape particulière ou encore le système de score et monnaie. En effet, tous ces éléments étaient suivis et auraient donné des résultats lors de la régression notamment.

Ceci m'amène à discuter ces résultats et notamment le futur de la recherche sur ces joueurs. Outre le fait d'inclure le geste dans les données monitorées, j'aimerais mener, à l'avenir, une réflexion plus poussée sur ce qu'il faut suivre en réalité virtuelle. Nous avons déjà identifié que le mouvement était une composante essentielle de la VR et qu'il faudrait l'inclure dans les futurs projets. Je n'ai pu le faire ici car le projet était trop avancé lorsque j'ai identifié cette caractéristique, et cela demande une infrastructure plus complexe. En effet, la multiplication des variables à prendre en compte multiplierait le nombre de données envoyées et exigerait d'autres base de données et fréquence d'envoi, peut-être même un stream de données. De plus, monitorer le mouvement se fait aujourd'hui uniquement à partir de la tête et des mains, à savoir du casque et des manettes. Une recherche a porté sur le développement d'une combinaison pour suivre les mouvements du corps, mais cela est loin d'être commercialisable. Il s'agit donc pour l'analyste de trouver des moyens de suivre ces parties du corps et d'extrapoler le plaisir ou la réussite du joueur dans ses gestes.

Dans un autre registre, la réalité virtuelle est aussi une affaire d'immersion. Comme mentionné, cela n'est pas mesurable aujourd'hui en dehors de questionnaires directement adressés aux joueurs. J'ai tenté une approche par des variables combinant le temps de jeu, le nombre de morts et le nombre de planètes régénérées pour identifier les joueurs persévérants et/ou faisant plus que le minimum requis. Cette approche est imparfaite, c'est pourquoi inclure un questionnaire dans la démarche data me semble une étape nécessaire dans toute analyse future.

Par ailleurs, il serait également intéressant d'inclure des variables de niveau macro dans mes analyses. Dans le cas de Star Shaman, ces variables pourraient être la sortie d'un autre jeu en VR retenant l'attention des joueurs ou encore le fait d'avoir organisé un concours au mois de mars poussant des gens à acheter le jeu pour participer. De telles variables permettent de prendre en compte des événements indépendant du jeu mais qui pourrait avoir une influence sur son audience.

Enfin, je souhaitais comparer les joueurs des différentes versions du jeu lors du lancement initial de ce projet. Je pensais notamment constater que les joueurs d'après la mise à jour de janvier restaient plus longtemps car ils recommençaient le jeu dans le dernier système solaire atteint avant leur mort. Or, identifier les versions successives jouées par une personne ne s'est avéré possible qu'après mon départ, avec la mise en place d'un numéro de version, lors de la préparation de ma dernière mise à jour. Il s'est donc avéré impossible de faire cette analyse, alors que les changements apportés au jeu étaient conséquents d'une version à l'autre. Ceci est aussi un enseignement pour une analyse future. Cependant, l'utilisation de la variable du premier horodatage (jour et heure du premier événement enregistré d'un joueur) dans la forêt aléatoire m'a permis de relativiser l'importance de la version de jeu. En effet, l'importance de cette variable était faible dans le fait de rester jouer, alors que commencer avec telle ou telle version me paraissait être un facteur important quant au fait de rester jouer. Il faudra revoir ce critère lors d'une nouvelle analyse avec un échantillon plus grand de joueur et peut-être en conservant les joueurs n'ayant pas gagné.

IV. ANNEXES

1. CLUSTERING – VARIABLES CLIVANTES

Moyennes des variables dans chaque cluster – jeu contenant tous les pays – joueurs quittant avant la 3ème planète régénérée

```
> profilsK_tousDepart_H3$centers
       ratio deathsNumberSystem1   meanScore meanCurrency randomNumber   clicNumber ratioRegenVisitees meanDeathHour
1  0.6344128          -0.3672396 -0.07985468   -0.3645773   -0.4711833 -0.018753930         -0.3443767   -0.4281471
2 -0.4101195           0.1395181  0.03251396    0.1941984    0.2362834  0.004593405          0.1480513    0.1869205
  NombreParties nbPlanetesRegereesModif nbreSessions moyenneMinutesSession tempsTotalJeu differenceScoreMinimum
1    -0.3635478             -1.3184678 -0.08518813           -0.7144346   -0.6934520            0.08271078
2     0.1370344              0.7315384  0.03577907            0.3851713    0.3546021            0.06107214
  differenceScoreMaximum differenceScoreMoyen differenceMonnaieMoyenne nbPartiesAuSysteme1
1             0.18077342            0.1323389                0.4402410          -0.3672396
2             0.00630039            0.0335173               -0.2248452           0.1395181
```

Tableau des moyennes des variables pour les joueurs partant avant le 3ème niveau, aux USA

```
     ratio deathsNumberSystem1  meanScore meanCurrency randomNumber   clicNumber ratioRegenVisitees meanDeathHour NombreParties
 0.4761894          0.9658923 -0.0352094   0.12946086    0.6957992 -0.031998961          0.8803517     1.3050152     0.9544676
-0.2240318         -0.4135715  0.0126148  -0.05543409   -0.3043806  0.006544397         -0.3731789    -0.5160434    -0.4096921
nbPlanetesRegereesModif nbreSessions moyenneMinutesSession tempsTotalJeu differenceScoreMinimum differenceScoreMaximum
              0.6743553     0.3507564             0.8612429    0.9464624            0.05250030           -0.3452488
             -0.2584365    -0.1446097            -0.3331977   -0.3874484            0.06742954            0.2131757
differenceScoreMoyen differenceMonnaieMoyenne nbPartiesAuSysteme1
          -0.1451723               -1.0713832          0.9658923
           0.1400228                0.4498425         -0.4135715
```

Moyennes des variables – haut : ensemble des pays, bas : USA – joueurs quittant à partir de la 3ème régénération

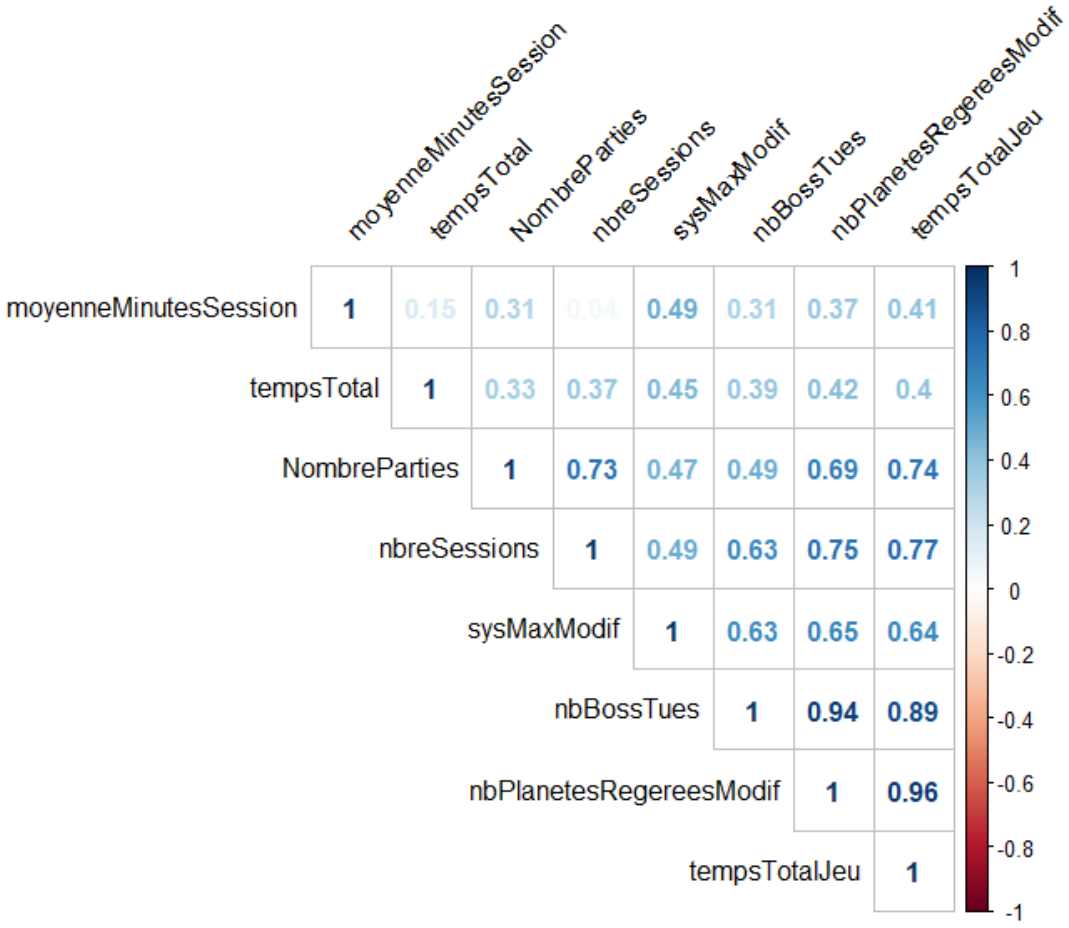
```
> profilsTousRestants_H2$centers
       ratio deathsNumberSystem1 deathsNumberSystem2 deathsNumberSystem3  meanScore meanCurrency randomNumber clicNumber
1 -0.20524053          0.6513194           1.3085338          0.26874170  0.9093547    1.1289766    1.1424717  0.5105743
2  0.02534367         -0.1836261          -0.3179961         -0.09531094 -0.2241111   -0.2715199   -0.2881727 -0.1441250
  ratioRegenVisitees meanDeathHour NombreParties tempsTotal nbPlanetesRegereesModif nbBossTues sysMaxModif nbreSessions
1       -0.052857707     0.3510621     1.1038458  0.9849174              1.1925576  1.0597868    1.433023    1.0587932
2       -0.008924668    -0.0889542    -0.2943262 -0.2330902             -0.3082641 -0.2739402   -0.329536   -0.2784586
  moyenneMinutesSession tempsTotalJeu differenceScoreMinimum differenceScoreMaximum differenceScoreMoyen differenceMonnaieMoyenne
1             0.4592521     1.2050947              0.6575088              0.5312603           0.7438848              0.8893947
2            -0.1059096    -0.3151556             -0.1634156             -0.1334687          -0.1908475             -0.2173692
  evolutionSystem nbPartiesAuSysteme1 nbPartiesAuSysteme2 nbPartiesAuSysteme3 tempsJusqueDernierRandom tempsAVictoire
1      -1.2353418           0.6513194           1.3920090           0.4092555              0.08204859     0.18633718
2       0.2809096          -0.1836261          -0.3345124          -0.1265313             -0.01772345    -0.06835692
  tempsjusquaSysteme3
1           0.5221589
2          -0.1169095
> profilsUSARestant_H2$centers
       ratio deathsNumberSystem1 deathsNumberSystem2 deathsNumberSystem3  meanScore meanCurrency randomNumber clicNumber
1 -0.19244467          0.7860649           1.370657          0.40801718  1.0126444    1.2451806    1.2655364  0.5900603
2  0.01401206         -0.1906921          -0.311937         -0.09229576 -0.2204826   -0.2816938   -0.2886854 -0.1526386
  ratioRegenVisitees meanDeathHour NombreParties tempsTotal nbPlanetesRegereesModif nbBossTues sysMaxModif nbreSessions
1       -0.05778940     0.37777673     1.2669746  1.0578509              1.2404281  1.0772548    1.3955271    1.2123105
2       -0.01606592    -0.08952507    -0.2974368 -0.2363119             -0.2919149 -0.2567989   -0.3010633   -0.2737451
  moyenneMinutesSession tempsTotalJeu differenceScoreMinimum differenceScoreMaximum differenceScoreMoyen differenceMonnaieMoyenne
1            0.43011222     1.2725679              0.7227656              0.6790474           0.8059392              0.9592853
2           -0.09719653    -0.3009629             -0.1663597             -0.1558313          -0.1935381             -0.2139383
  evolutionSystem nbPartiesAuSysteme1 nbPartiesAuSysteme2 nbPartiesAuSysteme3 tempsJusqueDernierRandom tempsAVictoire
1      -1.1438743           0.7860649           1.4312515           0.5414078              0.04612646     0.18220300
2       0.2459245          -0.1906921          -0.3228641          -0.1213768             -0.00937463    -0.06596516
  tempsjusquaSysteme3
1           0.5481735
2          -0.1156881
```

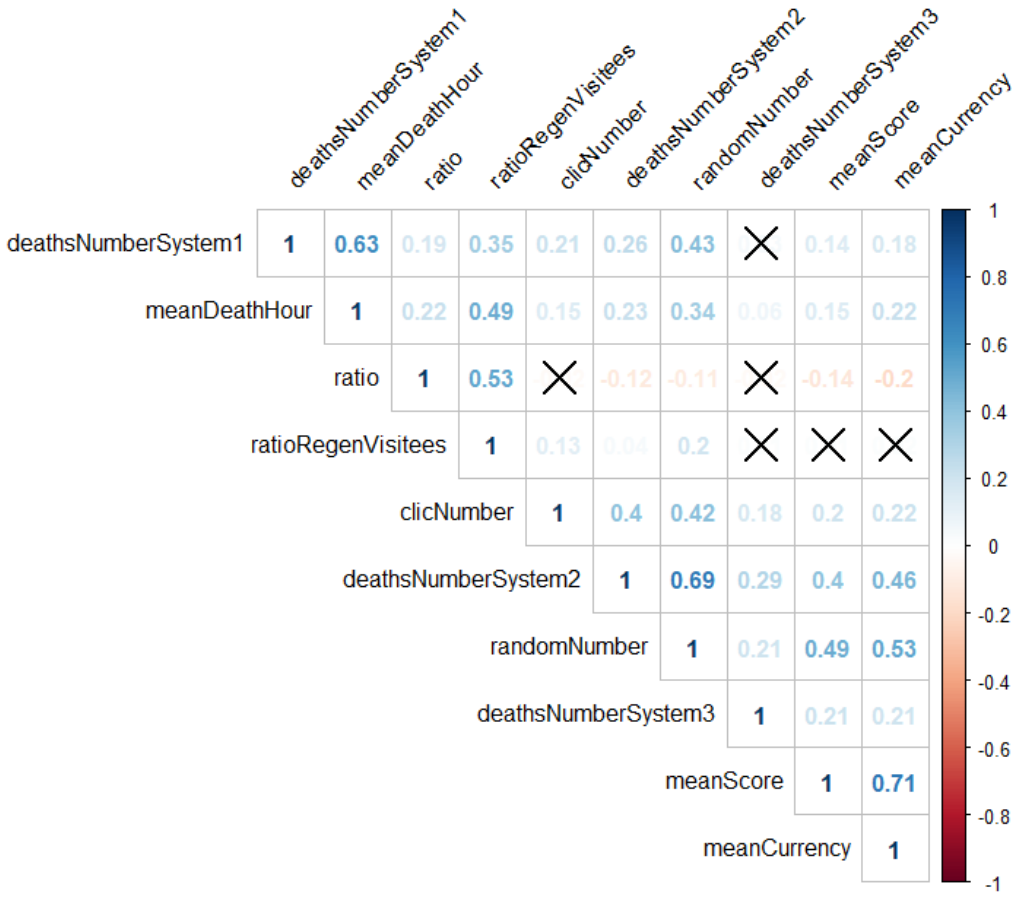
2. REGRESSION LOGISTIQUE – VARIABLES ET CORRELATIONS

Liste des variables dichotomiques créées. Elles servent à comparer les joueurs partis avant et après l'étape en question.

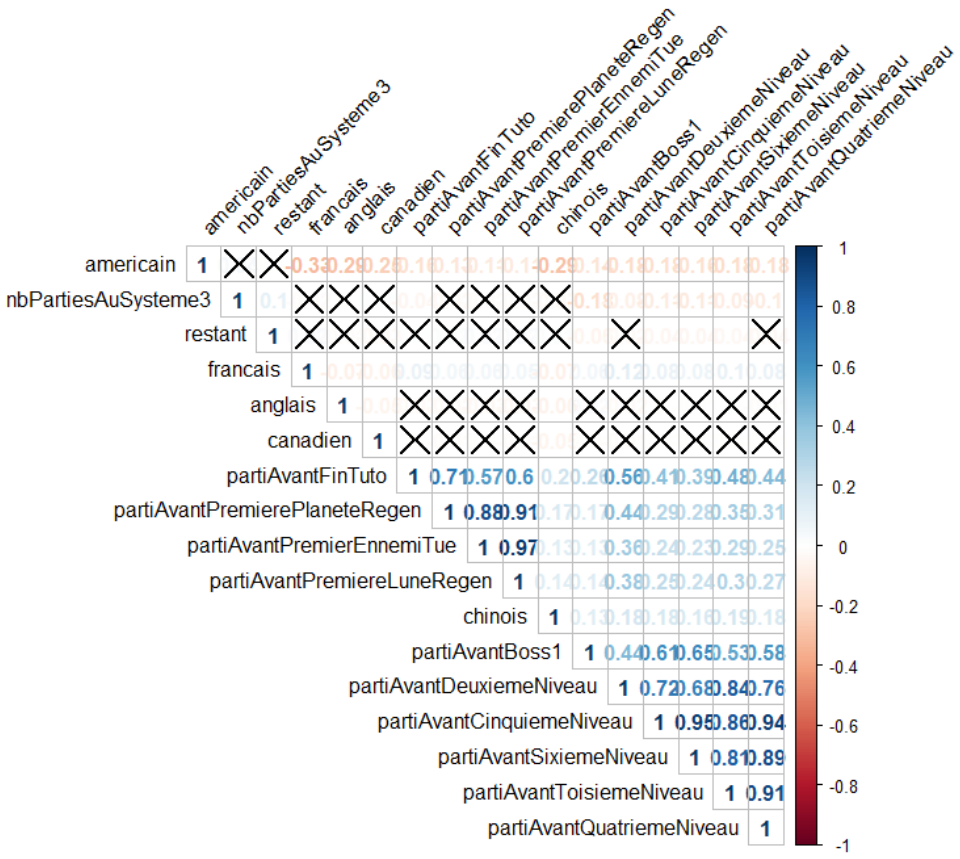
```
 [1] "partiAvantFinTuto"              "partiAvantPremierEnnemiTue"     "partiAvantPremiereLuneRegen"
 [4] "partiAvantPremierePlaneteRegen" "partiAvantDeuxiemeNiveau"       "partiAvantToisiemeNiveau"
 [7] "partiAvantQuatriemeNiveau"      "partiAvantCinquiemeNiveau"      "partiAvantSixiemeNiveau"
[10] "partiAvantBoss1"
```

Extrait du corrélogramme des variables, coupé pour des raisons de lisibilité




deathsNumberSystem1
meanDeathHour
ratio
ratioRegenVisitees
clicNumber
deathsNumberSystem2
randomNumber
deathsNumberSystem3
meanScore
meanCurrency
deathsNumberSystem1 1 0.63 0.19 0.35 0.21 0.26 0.43 0.14 0.18
meanDeathHour 1 0.22 0.49 0.15 0.23 0.34 0.15 0.22
ratio 1 0.53 -0.12 -0.11 -0.14 -0.2
ratioRegenVisitees 1 0.13 0.2
clicNumber 1 0.4 0.42 0.18 0.2 0.22
deathsNumberSystem2 1 0.69 0.29 0.4 0.46
randomNumber 1 0.21 0.49 0.53
deathsNumberSystem3 1 0.21 0.21
meanScore 1 0.71
meanCurrency 1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


evolutionSystem
nbPartiesAuSysteme3
differenceScoreMoyen
differenceScoreMinimum
differenceScoreMaximum
nbPartiesAuSysteme1
differenceMonnaieMoyenne
nbPartiesAuSysteme2
evolutionSystem 1 -0.18 -0.31 -0.25 -0.11 -0.26 -0.38 -0.69
nbPartiesAuSysteme3 1 0.22 0.13 0.07 0.04 0.22 0.33
differenceScoreMoyen 1 0.91 0.84 0.18 0.51 0.36
differenceScoreMinimum 1 0.91 0.17 0.43 0.3
differenceScoreMaximum 1 0.18 0.42 0.21
nbPartiesAuSysteme1 1 0.25 0.27
differenceMonnaieMoyenne 1 0.43
nbPartiesAuSysteme2 1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


americain
nbPartiesAuSysteme3
restant
francais
anglais
canadien
partiAvantFinTuto
partiAvantPremierePlaneteRegen
partiAvantPremierEnnemiTue
partiAvantPremiereLuneRegen
chinois
partiAvantBoss1
partiAvantDeuxiemeNiveau
partiAvantCinquiemeNiveau
partiAvantSixiemeNiveau
partiAvantToisiemeNiveau
partiAvantQuatriemeNiveau
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

3. REGRESSION LOGISTIQUE – RESULTATS ET ODDS

Variables identifiées par la méthode Stepwise

```
Coefficients:
                  (Intercept)                    clicNumber                    nbBossTues
                      -2.9239                       -0.1628                       -0.3910
                nbreSessions      differenceMonnaieMoyenne                nbPartiesAuSysteme2
                      0.3763                        0.1554                        0.5124
          nbPartiesAuSysteme3  partiAvantPremiereLuneRegen1  partiAvantPremierePlaneteRegen1
                     -0.1760                       14.6445                      -14.1990
                   americain1                      anglais1                      canadien1
                     -0.5228                       -1.0264                       -1.6599
```

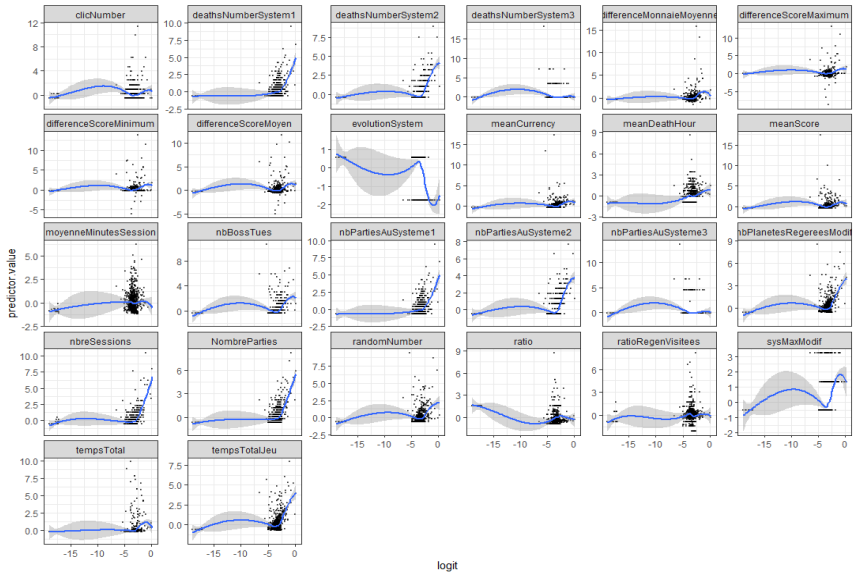
Odds des variables de la méthode Stepwise et significativité.

```
Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -2.92391    0.19133 -15.282  < 2e-16 ***
clicNumber                      -0.16282    0.11584  -1.406 0.159859
nbBossTues                      -0.39105    0.16627  -2.352 0.018682 *
nbreSessions                     0.37631    0.10596   3.551 0.000383 ***
differenceMonnaieMoyenne         0.15541    0.08819   1.762 0.078048 .
nbPartiesAuSysteme2              0.51236    0.13092   3.914 9.09e-05 ***
nbPartiesAuSysteme3             -0.17597    0.09594  -1.834 0.066639 .
partiAvantPremiereLuneRegen1    14.64452  560.10701   0.026 0.979141
partiAvantPremierePlaneteRegen1 -14.19897  560.10697  -0.025 0.979775
americain1                      -0.52283    0.24064  -2.173 0.029807 *
anglais1                        -1.02636    0.61852  -1.659 0.097039 .
canadien1                       -1.65988    1.02286  -1.623 0.104635
```

Odds ratio des variables significatives

names	x
(Intercept)	0.0537231
nbBossTues	0.6763479
nbreSessions	1.4568970
differenceMonnaieMoyenne	1.1681317
nbPartiesAuSysteme2	1.6692284
nbPartiesAuSysteme3	0.8386435
americain1	0.5928409
anglais1	0.3583083

Recherche de lien linéaire entre les valeurs des prédicteurs et les *log odds*.



4. FORET ALEATOIRE – RESULTATS

Matrice de confusion de la forêt aléatoire sur l'ensemble des variables

En ligne, les joueurs partis (0) ou restés (1) ; et en colonne les prédictions de la forêt

```
Call:
 randomForest(formula = restant ~ ., data = dataForetTrain[, c(2:39,      41:56)])
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 7

        OOB estimate of  error rate: 4.37%
Confusion matrix:
     0 1 class.error
0 1991 1 0.000502008
1   90 0 1.000000000
>
```

V. GLOSSAIRE

Analytics : dérivé de *Data Analytics*, la science d'analyse des données. Désigne le pipeline de données dans le projet Star Shaman.

Average Revenue Per User (ARPU) : revenu moyen généré par joueur.

Boss : ennemi très puissant, rencontré en fin de niveau dans un jeu vidéo. Dans Star Shaman, l'ennemi de fin de système solaire.

Churn : ratio des joueurs quittant le jeu sur le nombre de nouveaux joueurs

Clustering (partitionnement) : méthode de répartition des individus en groupe, selon leurs données.

Daily Active Users (DAU) : nombre de joueurs actifs dans le jeu, à une date donnée.

Free-to-play : jeu proposant soit des premiers niveaux gratuits avant paiement, soit un jeu entièrement gratuit avec des options payantes.

Gameplay : mécanisme du jeu influençant le style global.

Gamification : usage des processus et règles du jeu dans un contexte non ludique.

Jeu sérieux : jeu dont l'objectif n'est pas récréatif mais plutôt pédagogique ou clinique par exemple.

Massive Multiplayer Online Game (MMOG ou MMO) : jeu en ligne avec des serveurs puissants pouvant supporter des millions de joueurs. L'univers du jeu est persistant.

Massively Multiplayer Online Role-playing Game (MMPORG) : MMOG sous forme de jeu de rôle.

Metrics : données agrégées, non brutes.

Profiling (profilage) : création de profils d'individus théoriques, en fonctions de notions psychologiques.

Random events : planète sur laquelle le joueur vit un événement. Le joueur ne s'y bat pas mais peut gagner des points ou des éléments.

Réalité virtuelle (VR) : ensemble de technologies permettant à une personne d'être immergée dans un environnement virtuel et d'y évoluer grâce à l'interaction avec son corps.

Shop : planète-magasin sur laquelle le joueur peut acheter des sorts ou protections.

User Acquisition Cost (UAC) : coût moyen du jeu par joueur. Le coût du jeu est divisé par le nombre de joueurs

Viralité (k-factor) : Taux de conversion de joueurs. Taux de joueurs achetant le jeu.

VI. BIBLIOGRAPHIE

Alvarez, J. (2019). *Design des dispositifs et expériences de jeu sérieux* (Doctoral dissertation, Université Polytechnique des Hauts-de-France). https://isidore.science/document/10670/1.cui9h8

Aung, M., Demediuk, S., Sun, Y., Tu, Y., Ang, Y., Nekkanti, S., ... & Drachen, A. (2019, August). The trails of Just Cause 2: spatio-temporal player profiling in open-world games. In *Proceedings of the 14th International Conference on the Foundations of Digital Games* (pp. 1-11).https://doi.org/10.1145/3337722.3337765

Bartle, R. (1996). Hearts, clubs, diamonds, spades: Players who suit MUDs. Journal of MUD research, 1(1), 19. https://www.researchgate.net/publication/247190693_Hearts_clubs_diamonds_spades_Players_who_suit_MUDs

Bauckhage, C., Sifa, R., Drachen, A., Thurau, C., & Hadiji, F. (2014, August). Beyond heatmaps: Spatio-temporal clustering using behavior-based partitioning of game levels. Dans *2014 IEEE conference on computational intelligence and games* (pp. 1-8). IEEE.

https://doi.org/10.1109/CIG.2014.6932865

Benmakrelouf, S., Mezghani, N., & Kara, N. (2015, August). Towards the identification of players' profiles using game's data analysis based on regression model and clustering. Dans *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (pp. 1403-1410). IEEE. https://doi-org.proxybib-pp.cnam.fr/10.1145/2808797.2809429

Bezegova, E., Ledgard, M. A., Molemaker, R. J. (2017). *Virtual Reality and its Potential for Europe*. Ecorys. https://ec.europa.eu/futurium/en/system/files/ged/vr_ecosystem_eu_report_0.pdf

Bicalho, L. F., Baffa, A., & Feijó, B. (2019, October). A game analytics model to identify player profiles in singleplayer games. Dans *2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)* (pp. 11-20). IEEE. https://doi.org/10.1109/SBGames.2019.00013

Cai, Y. D., Ratan, R., Shen, C., & Alameda, J. (2015, July). Grouping game players using parallelized k-means on supercomputers. Dans *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure* (pp. 1-7).

https://dl.acm.org/doi/abs/10.1145/2792745.2792755

Charles, D., Black, M. (2004). Dynamic Player Modelling: A Framework for Player-centred Digital Games. *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education.*

https://www.researchgate.net/publication/251860100_Dynamic_Player_Modelling_A_Framework_for_Player-centred_Digital_Games

Chou, Y. K. (2014). Actionable gamification. Beyond Points, Badges, and Leaderboards. http://ganj-ie.iust.ac.ir:8081/images/c/c5/Actionable_Gamification.pdf

de Albuquerque, R. M., & Fialho, F. A. P. (2015). Fun and games: Player profiles. *The Computer Games Journal*, *4*(1), 31-46. https://link.springer.com/article/10.1007/s40869-015-0003-y

Drachen, A., & Canossa, A. (2009). Towards gameplay analysis via gameplay metrics. Dans *Proceedings of the 13th international MindTrek conference*. Tampere: ACM. https://dl.acm.org/doi/abs/10.1145/1621841.1621878

Drachen, A., El-Nasr, M. S., & Canossa, A. (2013). Game Analytics: Maximizing the Value of Player Data. Springer.

Drachen, Anders & Thurau, Christian & Sifa, Rafet & Bauckhage, Christian. (2014). A Comparison of Methods for Player Clustering via Behavioral Telemetry. Dans *Proceedings of the 8th International Conference on the Foundations of Digital Games*. https://arxiv.org/abs/1407.3950

Drakopoulos, G., Voutos, Y., & Mylonas, P. (2020). Annotation-Assisted Clustering of Player Profiles in Cultural Games: A Case for Tensor Analytics in Julia. *Big Data and Cognitive Computing*, *4*(4), 39. https://doi.org/10.3390/bdcc4040039

Ducheneaut, N., Yee, N. (2012). Les jeux vidéo en ligne, un miroir de la personnalité des internautes ? *Questions de communication.* https://doi.org/10.4000/questionsdecommunication.6571

Fernandes, W. R., & Levieux, G. (2019, November). $$\delta $$-logit: Dynamic Difficulty Adjustment Using Few Data Points. Dans *Joint International Conference on Entertainment Computing and Serious Games* (pp. 158-171). Springer, Cham. https://link.springer.com/chapter/10.1007/978-3-030-34644-7_13

Heaton, L., & Lafrance, J. P. (1994). Les communautés virtuelles ludiques. Réflexions sur les jeux multi-utilisateurs. *Réseaux. Communication-Technologie-Société*, *12*(67), 95-110. https://doi.org/10.3406/reso.1994.2740

Hullett et al., 2011, K., Nagappan, N., Schuh, E., & Hopson, J. (2011, May). Data analytics for game development: NIER track. Dans *2011 33rd International Conference on Software Engineering (ICSE)* (pp. 940-943). IEEE.

https://dl-acm-org.proxybib-pp.cnam.fr/doi/epdf/10.1145/1985793.1985952

Jayachandran, K., Chilakamarri, S., Coelho, D., & Mueller, K. (2017, November). A virtual reality grocery shopping game to improve awareness for healthy foods in young adults. Dans *2017 13th International Conference and Expo on Emerging Technologies for a Smarter World (CEWIT)* (pp. 1-5). IEEE. https://doi.org/10.1109/CEWIT.2017.8263138

Kassim, M. N. H. M. (2018). Data analytics on interactive indoor cycling exercises with virtual reality video games. Dans *4th International Conference on Control, Automation and Robotics (ICCAR)*, Auckland, 2018, pp. 321-326. https://doi.org/10.1109/ICCAR.2018.8384693

Ladly, M. J., Bakker, T., Chadha, K., Farrelly, G., Micak, K., Penn, G., & Rudzicz, F. (2017). Reality recalled: Elders, memory and VR. Dans *2017 23rd International Conference on Virtual System & Multimedia (VSMM)* (pp. 1-9). IEEE. https://doi.org/10.1109/VSMM.2017.8346256

Lewis, C., & Wardrip-Fruin, N. (2010, June). Mining game statistics from web services: a World of Warcraft armory case study. Dans *Proceedings of the Fifth International Conference on the Foundations of Digital Games* (pp. 100-107). https://doi.org/100-107. 10.1145/1822348.1822362

Li, X., & Chen, Y. (2020, May). Auto-Hierarchical Data Algorithm: Focus on Increasing Users' Motivation and Duration In Virtual Reality. Dans *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)* (pp. 150-153). IEEE. https://doi.org/10.1109/ICBDA49040.2020.9101254

Marczewski, A. (2015). User Types. Dans A. Marczewski, *Even Ninja Monkeys Like to Play: Gamification, Game Thinking and Motivational Design* (2nd ed, pp.65-80). CreateSpace Independent Publishing Platform. https://www.gamified.uk/user-types/

McGregor, C., Bonnis, B., Stanfield, B., & Stanfield, M. (2017, April). Integrating Big Data analytics, virtual reality, and ARAIG to support resilience assessment and development in tactical training. Dans *2017 IEEE 5th International Conference on Serious Games and Applications for Health (SeGAH)* (pp. 1-7). IEEE. https://doi.org/10.1109/SeGAH.2017.7939256

Oppermann, L., & Slussareff, M. (2016). Pervasive games. Dans *Entertainment computing and serious games* (pp. 475-520). Springer, Cham. https://doi-org.proxybib-pp.cnam.fr/10.1007/978-3-319-46152-6_18

Paiva, Felipe & Franco, Artur & Mendonça Junior, Glaudiney & Maia, José. (2018). Analyzing Player Profiles in Collectible Card Games. Dans *Proceedings of 17th SBGames*. Foz do Iguaçu, Brazil. http://www.sbgames.org/sbgames2018/files/papers/CulturaFull/188202.pdf

Perez-Colado, I. J., Rotaru, D. C., Freire-Moran, M., Martinez-Ortiz, I., & Fernandez-Manjon, B. (2018, September). Multi-level game learning analytics for serious games. In 2018 10th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games) (pp. 1-4). IEEE. https://doi.org/10.1109/VS-Games.2018.8493435

Perrot, B., Hardouin, J. B., Grall-Bronnec, M., Costes, J. M., & Challet-Bouju, G. (2016). Dépistage des comportements excessifs de jeu sur Internet. Revue d'Épidémiologie et de Santé Publique, 64, S156. https://doi.org/10.1016/j.respe.2016.03.099

Petsani, D., Kostantinidis, E. I., Zilidou, V. I., & Bamidis, P. D. (2018, June). Exploring health profiles from physical and cognitive serious game analytics. In 2018 2nd International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW) (pp. 1-6). IEEE. https://doi.org/10.1109/TISHW.2018.8559562

Rodrigues, L. A., & Brancher, J. D. (2018, October). Improving Players' Profiles Clustering from Game Data Through Feature Extraction. In 2018 17th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames) (pp. 177-17709). IEEE. https://www.researchgate.net/publication/328538791_Improving_Players'_Profiles_Clustering_from_Game_Data_Through_Feature_Extraction

Sherman, W. R., & Craig, A. B. (2018). *Understanding virtual reality—Interface, application, and design* (2e éd.). Elsevier Science.

Sidre, C. (2020). Le contexte d'émergence d'une industrie du jeu vidéo en France (1975-1988): Logiques de production et modèles éditoriaux. Réseaux, 6(6), 31-58. https://doi.org/10.3917/res.224.0031

Sifa, R., Drachen, A., Bauckhage, C., Thurau, C., & Canossa, A. (2013, August). Behavior evolution in tomb raider underworld. Dans *2013 IEEE Conference on Computational Inteligence in Games (CIG)* (pp. 1-8). IEEE. https://doi.org/10.1109/CIG.2013.6633637

Spronck, P., Balemans, I., & Van Lankveld, G. (2012, October). Player profiling with fallout 3. Dans *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (Vol. 8, No. 1). https://ojs.aaai.org/index.php/AIIDE/article/view/12523

Thibault, C. (2013). Game Data Analysis–Tools and Methods. Packt Publishing Ltd.

Tondello, G. F., Mora, A., Marczewski, A., & Nacke, L. E. (2019). Empirical validation of the gamification user types hexad scale in English and Spanish. *International Journal of Human-Computer Studies*, *127*, 95-111. https://doi.org/10.1016/j.ijhcs.2018.10.002

Wallner, G., Kriglstein, S., Drachen A. (2019). Tweeting your Destiny: Profiling Users in the Twitter Landscape around an Online Game. *2019 IEEE Conference on Games (CoG)*, London, United Kingdom, 2019, pp. 1-8. https://doi.org/10.1109/CIG.2019.8848079

Yannakakis, G. N., Spronck, P., Loiacono, D., & Andre, E. (2013). Player modeling. Dans S. M. Lucas, M. Mateas, M. Preuss, P. Spronck, & J. Togelius (Eds.), *Artificial and computational intelligence in games* (6, pp.45-59). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. https://www.um.edu.mt/library/oar/handle/123456789/29725