Chapitre 6

Mesure de la difficulté

Sommaire

6.1 Objectif 2
6.2 Les différentes variables 2
6.2.1 L'effort nécessaire 3
6.2.2 Perception 7
6.2.3 Décision 9
6.2.4 Action 12
6.3 Modèle simplifié 13
6.4 Les challenges 15
6.5 Mesure de la performance 16
6.6 Mesure du niveau du joueur 17
6.7 Equation générale de la difficulté 19
6.8 Trace d'évènements 21
6.9 Calcul du niveau du joueur 22
6.10 Calcul de la difficulté 24
6.11 Apprentissage du joueur 26
6.11.1 Correspondance entre capacités locales 26
6.12 Temps et difficulté 27
6.13 Durée d'un challenge et théorie de la fiabilité 28
6.14 Tracer la courbe de difficulté 29
6.15 Synthèse 30

6.1 Objectif

Dans ce chapitre, nous décrivons une méthode de mesure de la courbe de difficulté d'un jeu vidéo. Nous avons mis au point cette mesure en nous imposant la contrainte de n'exploiter qu'un enregistrement du comportement du joueur dans l'univers du jeu. Des méthodes plus invasives, comme l'utilisation de questionnaires directs ou d'un système de mesure de variables physiologiques permettraient d'éclairer plus précisément de nombreuses facettes de la difficulté d'un jeu vidéo. Mais en nous limitant à une mesure indirecte et transparente, c'est à dire sans questionner directement le joueur ni l'équiper d'un appareil de captation particulier, nous proposons un système de mesure plus léger. Un tel système sera exploitable dans le contexte de tests à grande échelle, et en particulier dans le cadre de jeux disponibles sur internet et permettant l'enregistrement du comportement des joueurs sur un serveur de jeu. Les ordinateurs et consoles de salons permettent en effet aujourd'hui d'accéder à de nombreux systèmes de jeu en ligne, ce qui facilite particulièrement la réalisation de ce type de **playtest** [1] . En contrepartie, ces tests se déroulent dans un environnement non contrôlé et seul le comportement du joueur dans l'univers virtuel nous est accessible.

Le fait de limiter ainsi notre mesure implique qu'une partie des variables utiles à l'évaluation de la courbe de difficulté ne seront pas observables. Nous pouvons néanmoins définir un certain nombre d'hypothèses qui, si elles sont respectées, permettent de contraindre ces variables non observables, et d'obtenir une mesure correcte de la courbe de difficulté d'un gameplay.

6.2 Les différentes variables

Dans la section **??**, nous avons défini la difficulté d'un jeu comme la quantité d'effort que le joueur doit investir pour atteindre son objectif. Cette définition donne une bonne intuition de la notion de difficulté. Cependant, afin d'établir une mesure correcte de la difficulté, il est important de détailler précisément les différentes variables qui conditionnent cet effort. Il est ensuite nécessaire d'identifier précisément les différentes relations qu'entretiennent ces variables, ainsi que les variables que nous pouvons effectivement mesurer. Si certaines variables ne nous sont pas accessibles, nous devrons fournir un ensemble de règles permettant de les contraindre et d'être ainsi tout de même capable d'effectuer notre mesure. La figure 6.1 présente un modèle complet des différentes variables impliquées dans une mesure de la difficulté, ainsi que leurs relations.

1. Un playtest et un processus par lequel un game designer teste un nouveau jeu, pour en éliminer les erreurs logicielles ou pour l'améliorer, avant sa mise sur le marché.

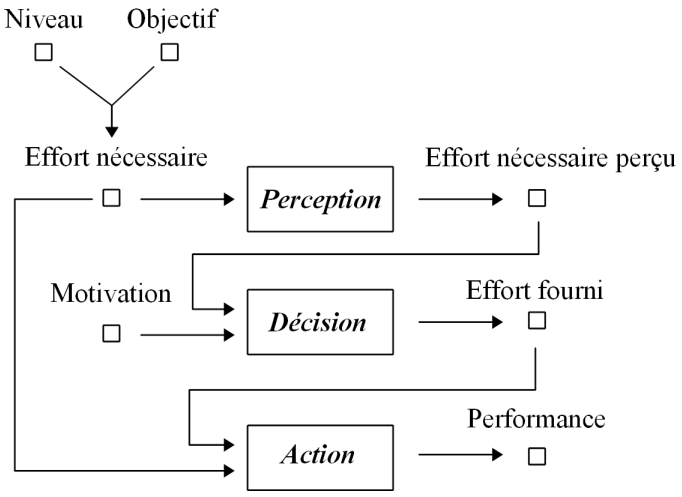

FIGURE 6.1 – Relations entre les variables du modèle de difficulté.

Les rectangles au texte en italique sont des processus, les carrés sont des variables. Cette figure décrit les liens entre les variables de notre modèle de la difficulté d'un jeu vidéo. L'objectif et le niveau du joueur déterminent l'effort nécessaire, c'est à dire la difficulté de la tâche. Dans un premier temps, le joueur *perçoit* l'effort nécessaire. Le joueur va ensuite *décider*, à partir de l'effort nécessaire perçu et de sa motivation, de l'effort qu'il va effectivement investir. Finalement, lorsque le joueur *agit*, le rapport entre l'effort fourni et l'effort nécessaire détermine la performance qu'il va atteindre.

Dans les sections suivantes, nous allons décrire l'ensemble des processus et des variables présentées dans la figure 6.1. Notre objectif consiste à mesurer la variable *effort nécessaire*, c'est à dire la difficulté de la tâche. Cette variable n'est malheureusement pas directement mesurable à partir du comportement du joueur, mais peut être déduite d'autres variables. Nous allons donc identifier les variables que nous sommes capables de mesurer, et les hypothèses sous lesquelles leur mesure nous permet de déduire la difficulté du jeu.

6.2.1 L'effort nécessaire

La première partie du modèle présenté dans la figure 6.1 montre que l'*effort nécessaire* est la résultante d'un *objectif* et du *niveau du joueur*. Dans cette section, nous détaillons cette partie du modèle, isolée dans la figure 6.2.

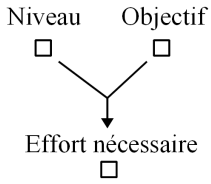

FIGURE 6.2 – Relation entre niveau du joueur et effort nécessaire pour réussir le challenge.

6.2.1.1 Variables impliquées

La difficulté est un effort déployé pour atteindre un objectif. Nous définissons d'abord ce que la variable *objectif* du modèle représente exactement :

> • **Objectif :** un objectif est une fonction binaire de l'état du jeu. Pour qu'un objectif soit *atteint*, le joueur doit parvenir à modifier l'état pour satisfaire cette fonction.

L'objectif dont nous souhaitons calculer la difficulté est l'objectif du *challenge* en cours. Un challenge décrit le rapport qu'entretient un joueur avec un objectif particulier du jeu, à un instant donné [2]. L'état d'un challenge permet par exemple de savoir si l'objectif a été atteint ou si le joueur est toujours en train d'essayer de l'atteindre, ou a échoué. Comme le définissent Rollings et Adams, le **gameplay** [3] d'un jeu est une suite de challenges [Rollings 03]. Chaque challenge est considéré comme *atomique*, c'est à dire que le joueur n'est pas en mesure d'interrompre un challenge, ni d'effectuer plusieurs challenges en parallèle. De cette manière, nous savons que lorsqu'un challenge est actif, le joueur se consacre à l'objectif correspondant, et que son comportement peut être interpréter en fonction de ce contexte précis.

> • **Challenge :** un challenge correspond à un objectif que le joueur doit atteindre. L'état d'un challenge décrit l'avancée du joueur vis à vis de cet objectif. Un challenge *actif* est en cours d'exécution par le joueur. Un challenge ne peut ni être interrompu, ni exécuté en parallèle avec un autre challenge.

Nous définissons deux types d'objectifs : l'*objectif du challenge* et les *sous-objectifs*. L'objectif du challenge est l'objectif général que poursuit le joueur, et correspond à la variable objectif de notre modèle. C'est l'effort nécessaire pour atteindre cet objectif que nous voulons évaluer. Mais pour atteindre l'objectif principal, le joueur va poursuivre des sous objectifs, qui lui permettent de s'en rapprocher. Dans un jeu, chaque challenge propose un objectif

2. Voir sections 6.4 et **??** pour une description plus complète des challenges

3. Rollings et Adams définissent le gameplay de la façon suivante : *Les challenges, ainsi que les actions que le joueur peut entreprendre pour les réussir, constituent le gameplay. D'un manière plus large, on peut considérer le gameplay comme ce que fait le joueur lorsqu'il joue, c'est à dire le type de problèmes qui lui sont posés et la manière dont il les résout.*

général, mais les sous objectifs que le joueur va poursuivre sont quant à eux souvent redondants entre challenges, se sont les briques de gameplay que le game designer utilise pour construire les challenges.

- **Objectif du challenge :** objectif que le joueur doit atteindre pour réussir le challenge.

- **Sous-objectif :** objectif intermédiaire et poursuivi de manière répétitive, qui permet au joueur de progresser vers l'objectif du challenge.

Nous pouvons illustrer la différence entre les sous objectifs et l'objectif du challenge par l'exemple d'un jeu de plateforme. Chaque parcours que doit traverser le joueur, c'est à dire chaque challenge, est unique. Mais pour traverser le parcours, le joueur saute de plateformes en plateformes. Il existe un nombre limité de types de sauts que le joueur va effectuer : différentes combinaisons de plateformes stables ou mouvantes, plus ou moins éloignées ou plus ou moins larges. Chaque saut correspond à un sous objectif. Il va poursuivre ces sous objectifs de manière répétitive, et progresser petit à petit vers la fin du niveau, c'est à dire vers l'objectif du challenge.

L'existence de sous objectifs communs à différents challenges est nécessaire à la construction d'un gameplay : si chaque challenge reposait sur des sous objectif entièrement nouveaux, la manière de jouer chaque challenge serait également entièrement nouvelle, et en quelque sorte, le joueur jouerait à chaque fois à un nouveau jeu. Un jeu vidéo repose sur un gameplay principal, exploité au cours de différents challenges. Grâce à la redondance des sous objectifs, le joueur peut répéter des séquences d'actions de nombreuses fois et progresser dans sa maitrise d'un gameplay.

Pour atteindre l'objectif du challenge, le joueur doit fournir un effort. Il est nécessaire de définir la notion d'effort, et avec elle la variable *effort nécessaire*.

- **Effort nécessaire :** effort que le joueur doit fournir pour atteindre l'objectif du challenge. Cette variable correspond à la difficulté que nous souhaitons mesurer.

Le joueur dispose d'une certaine quantité de ressources. Il dispose par exemple de ressources attentionnelles, qu'il consacre aux tâches de son choix. Il peut également choisir d'être plus ou moins actif cérébralement pour résoudre un problème, et consommer une quantité d'énergie plus ou moins importante. Sans rentrer dans le détail des différentes ressources exploitées pour jouer à un jeu vidéo, on peut simplement poser comme principe que le joueur dispose d'une certaine quantité de ressources, et que cette quantité est bornée.

L'effort peut être ensuite défini comme l'exploitation de ces ressources. Le dictionnaire Larousse définit l'effort de la façon suivante « *Mobilisation volontaire de forces physiques, intellectuelles, morales en vue de résister ou pour vaincre une résistance* ». Plus précisément, nous définirons l'effort du joueur comme le rapport entre la quantité de ressources investie et la quantité de ressources disponibles. Si le joueur investit la totalité des ressources disponibles

dans une tâche, son effort est considéré comme maximal. On définit donc ici l'effort comme une valeur relative à chaque sujet : si deux joueurs possèdent par exemple une mémoire de travail de taille différente, mais l'exploitent au maximum, on considère que leur effort est similaire.

De la même manière, l'*effort nécessaire* est le rapport entre les ressources que le joueur doit investir et les ressources dont il dispose. L'effort nécessaire est donc dans l'intervalle $[0, +\infty]$ car un challenge peut demander au joueur d'investir une quantité de ressources bien supérieure à celle dont il dispose.

Nous distinguons l'objectif du challenge des sous objectifs car le premier décrit un objectif général, valide le temps d'un challenge, et les derniers permettent de décrire plus précisément le comportement du joueur. Les mesures liées aux sous objectifs nous permettent ainsi d'établir un modèle du joueur, de calculer son niveau :

- **Niveau du joueur :** effort que doit fournir le joueur pour valider un sous-objectif du challenge. Plus le niveau du joueur est élevé, moins l'effort investi est important.

D'un point de vue général, le niveau du joueur regroupe les savoirs et compétences qui lui permettent de convertir un effort en une certaine performance. Si deux joueurs investissent un effort similaire pour réussir un challenge, ils n'obtiendront pas forcément le même résultat car leur savoir sont différents, leur stratégies de résolutions sont différentes, plus ou moins efficaces, et plus ou moins bien exécutées. Ces savoirs et compétences sont acquis challenge après challenge, lorsque le joueur tente d'atteindre les sous-objectifs, et représentent la maîtrise qu'à le joueur du gameplay.

L'effort nécessaire, c'est à dire la difficulté du challenge, est conditionné à la fois par le niveau du joueur et par l'objectif à atteindre. En effet, la capacité du joueur à atteindre les sous objectifs du challenge détermine l'effort qu'il devra fournir pour atteindre l'objectif général du challenge.

6.2.1.2 Mesurabilité des variables

Le *niveau du joueur* correspond à l'ensemble des savoirs et compétences du joueur, au sujet du jeu que nous analysons. Comme nous l'avons décrit précédemment, certains jeux demandent aux joueurs de découvrir des informations enfouies dans l'univers du jeu : par exemple la position d'un objet caché, ou un chemin dans un labyrinthe. Ces informations peuvent être communiquées au joueur de manière extrinsèque, sous la forme d'un « *walk-through* » par exemple, c'est-à-dire une description des différentes informations nécessaires à la réussite des challenges. Les compétences, quant à elles, sont acquises par la pratique.

Dans un jeu de combat par exemple, le joueur doit à la fois découvrir les coups spéciaux de son personnage et être également capable de les exécuter. La séquence de touches est une information, d'ailleurs bien souvent disponible dans le manuel du jeu ou communiqué au

travers des différents tutoriaux. Le fait d'être capable, en jouant, d'exécuter la séquence de touches au bon moment est une compétence, et ne peut être acquise que par l'entrainement.

Cependant, dans notre modèle, nous proposons de ne pas faire de distinction entre savoirs et compétences. Tout comportement du joueur dans le jeu vidéo est soumis à ces deux aspects de l'apprentissage : le joueur doit d'abord découvrir que l'univers du jeu est manipulable d'une certaine manière, que certaines séquences d'actions sont autorisées et permettent d'obtenir un résultat utile. Ensuite, ce savoir n'est pas forcément suffisant pour garantir le succès du joueur : les contraintes psycho-motrices de manipulation de l'univers du jeu demandent au joueur de pratiquer suffisamment ces comportements pour acquérir le temps de réflexe et la précision nécessaires. Mais dans tous les cas, la somme des savoir et compétences du joueur sert une seule et même utilité : sa capacité à atteindre les sous objectif du challenge, et sera pour nous résumée à çà.

Que ce soit pour mesurer le niveau du joueur ou la difficulté du jeu, il s'agit d'évaluer l'effort nécessaire pour atteindre un objectif. En quelque sorte, le modèle de la difficulté d'un challenge est récursif, c'est à dire que le niveau du joueur pourrait également être décrit par le même schéma, à ceci près qu'il s'agirait d'atteindre un sous objectif au lieu de l'objectif d'un challenge. La figure 6.2.1 montre simplement le lien de causalité entre sous objectif et objectif général : la difficulté à atteindre un objectif général dépend de la difficulté à atteindre les sous objectif dont il dépend. Malgré leur apparente similitude, nous souhaitons cependant distinguer la difficulté à réussir un challenge de celle à atteindre un sous objectif : la généralité et la redondance des sous-objectifs font qu'ils correspondent à des savoir et compétences accumulés par le joueur, la ou la difficulté de l'objectif du challenge est bien plus spécifique et correspond à l'exploitation de ces savoirs et compétences dans un contexte particulier.

Le niveau du joueur ou la difficulté du challenge ne sont pas directement mesurable : seul l'effort effectivement fourni par le joueur pourrait être mesuré, ou déduit de la performance du joueur, c'est à dire du fait qu'il atteigne ou non l'objectif. Cependant ces déductions ne sont valides que sous un certain nombre d'hypothèses. Les sections suivantes présentent les hypothèses nécessaires à la mesure d'un effort nécessaire à partir de la performance obtenue, valables autant pour la mesure du niveau du joueur que pour celle de la difficulté du challenge.

6.2.2 Perception

Le premier processus intervenant dans notre modèle de la difficulté d'un challenge (Figure 6.1) est un processus perceptif. La figure 6.3 isole ce processus et les variables impliquées.

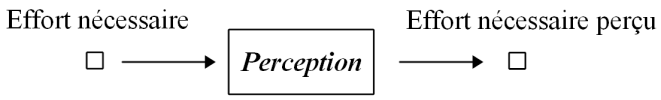

FIGURE 6.3 – Perception de l'effort.

6.2.2.1 Variables impliquées

Nous avons défini l'effort nécessaire dans la section précédente. La variable que fournit le processus perceptif est la suivante :

- **Effort nécessaire perçu :** l'effort que le joueur suppose devoir investir pour atteindre son objectif.

Au travers de ce processus perceptif, le joueur s'appuie sur son expérience du jeu pour déterminer l'effort qu'il doit investir pour atteindre son objectif. S'il a déjà tenté d'atteindre un objectif similaire, il peut se souvenir de l'effort qu'il avait investi et du résultat qu'il avait obtenu, et de cette manière, utiliser son expérience pour évaluer l'effort que la tâche demande.

Cette étape est importante car elle montre que le comportement du joueur n'est pas forcément directement lié à l'effort que demande la tâche, c'est à dire à sa difficulté. Le joueur doit d'abord percevoir cette difficulté pour ajuster son comportement, ce dont nous devons tenir compte dans notre mesure de la difficulté.

6.2.2.2 Mesurabilité des variables

Nous ne sommes pas capables de mesurer directement l'effort perçu par le joueur. Nous pourrions utiliser des questionnaires mais nous sortirions du cadre de notre mesure, qui doit n'être réalisée qu'à partir du comportement du joueur dans le monde du jeu, sans autre source d'information.

Néanmoins, nous pouvons simplifier notre modèle en formulant certaines hypothèses sur la perception de l'effort par le joueur.

6.2.2.3 Hypothèses

Nous supposons que dans un jeu vidéo, et sous certaines hypothèses, le joueur percevra toujours l'effort nécessaire comme maximal. Pour que le joueur puisse supposer qu'il peut atteindre son objectif sans être totalement investi, les deux conditions suivantes doivent être réunies :

– Le joueur doit avoir déjà atteint un objectif équivalent

– Le joueur doit avoir réussit à atteindre cet objectif similaire sans avoir investi un effort maximal

Nous supposons en effet que pour que le joueur prenne le risque de considérer qu'il n'est pas nécessaire d'investir un effort maximal pour réussir, il faut qu'il ait l'expérience, sûrement répétée, d'avoir atteint un objectif similaire en investissant effectivement un effort limité. Nous supposons donc que l'effort perçu par le joueur sera maximal si une des deux conditions suivantes est vérifiée :

– Les objectifs que le joueur doit atteindre sont en apparence suffisamment variés

– Le joueur ne parvient pas à atteindre ses objectifs de manière répétée

Nous formons l'hypothèse que plus le joueur sera soumis à l'échec et plus les objectifs seront variés, moins le joueur sera capable d'inférer l'effort nécessaire. Il n'aura alors d'autre choix que de considérer l'effort nécessaire comme maximal, afin d'assurer sa victoire.

Sous cette hypothèse, nous pouvons considérer que l'effort perçu par le joueur sera considéré comme maximal, et n'est donc plus une variable du modèle, mais une constante.

6.2.3 Décision

Le processus suivant de notre modèle de la difficulté d'un challenge (Figure 6.1) est une prise de décision. A partir de l'effort perçu et de sa motivation, le joueur va déterminer quel effort fournir. Nous avons isolé ce processus et ses variable dans la figure 6.4.

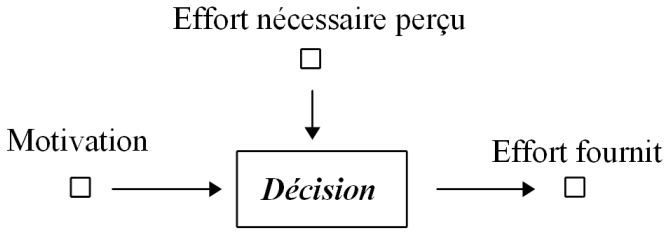

FIGURE 6.4 – Détermination de l'effort fourni.

6.2.3.1 Variables impliquées

Le processus de décision traite tout d'abord l'effort nécessaire perçu. Mais comme nous l'avons défini dans la section suivante, sous certaines hypothèses, cette variable est une constante. L'autre variable traitée par ce processus est la suivante :

- **Motivation :** La motivation du joueur correspond à sa volonté d'atteindre l'objectif qui lui est proposé.

La décision du joueur concerne l'effort qu'il va effectivement fournir :

- **Effort fourni :** L'effort fourni correspond à l'effort que le joueur investit effectivement pour atteindre l'objectif.

Le joueur n'est en effet pas nécessairement attiré par l'objectif qu'on lui propose. En effet, le comportement des joueurs varie en fonction de leurs préférences. Comme l'a par exemple étudié Richard Bartle [Bartle 96], on peut distinguer plusieurs profils motivationnels chez les joueurs de **MMORPG** [4] . Certains joueurs s'attachent à réussir l'ensemble des challenges que propose le gameplay, d'autres recherchent la compagnie des joueurs, ou veulent avant tout découvrir l'ensemble du monde virtuel. Les motivations des joueurs peuvent influencer l'effort fourni pour deux raisons différentes.

Tout d'abord, le profil motivationnel du joueur peut l'amener à limiter l'effort investi dans la réalisation de la tâche proposée par le gameplay. Si la tâche proposée ne correspond pas aux motivations du joueur, son investissement pourra être moindre. D'une manière encore plus subtile, par exemple, le fait que le profil général du joueur soit basé sur une recherche de la réussite ou l'évitement de l'échec influencera l'effort investi pour atteindre l'objectif [Capa 08].

Ensuite, les préférences du joueur peuvent l'amener à investir une partie de son effort dans d'autres objectifs, à priori non proposés par le gameplay. Dans un **First Person Shooter** [5] comme Crysis (Crytek - EA) par exemple, le joueur peut emprunter un chemin plus dangereux, plus long, simplement pour pouvoir, par exemple, utiliser un véhicule qu'il affectionne particulièrement. Le monde virtuel de Crysis est très ouvert, et le parcours du joueur y est faiblement contraint, ce qui laisse beaucoup de place aux objectifs personnalisés, non prévisibles par le game designer. Le joueur peut ainsi agrémenter le gameplay de meta-objectifs personnels, qui reflètent plus précisément ses préférences. En conséquence, une partie de l'effort fourni par le joueur sera attribué à ces meta-objectifs ce qui, en contrepartie, limitera l'effort investi dans l'objectif principal du gameplay.

6.2.3.2 Mesurabilité des variables

Nous ne sommes pas en mesure de mesurer l'effort fourni par le joueur. Nous pourrions par exemple tenter de le déduire au moyen d'un dispositif de captation des battements cardiaques et de l'activité électrique des muscles de la face [Capa 08], mais ces techniques sont particulièrement complexes à mettre en place et sortent du cadre de notre modèle, nous souhaitons effectuer nos mesures sans avoir recours à un dispositif de captation particulier.

4. Massively Multiplayer Online Role Playing Game : jeu de role massivement multijoueurs en ligne.
5. First Person Shooter, jeu de tir en vue subjective.

La motivation du joueur peut quant à elle être évaluée à partir de questionnaires, mais nous souhaitons également nous passer de ce type de mesures, particulièrement difficiles à mettre en place lorsque les tests sont réalisés hors laboratoire, comme par exemple lors d'une partie jouée sur internet. Les joueurs viennent alors principalement pour jouer et leur motivation à remplir des questionnaires hors environnement contrôlé n'est pas du tout assurée.

Nous devons donc définir sous quelles hypothèses les différentes variables du processus de décision peuvent être négligées.

6.2.3.3 Hypothèses

Nous supposons que si les deux règles suivantes sont valides simultanément, l'impact de la motivation du joueur sur son comportement sera minimisé :

- En recrutant pour le playtest des joueurs non rémunérés, issus d'une communauté de joueurs attirés par le type de gameplay proposé, la motivation des joueurs sera maximisée.
- Plus le gameplay du jeu sera fermé, c'est à dire limitera la capacité du joueur à créer des meta objectifs, plus on pourra considérer que le joueurs investissent leur effort pour atteindre l'objectif principal.

En effet, les joueurs peuvent être recrutés en fonction de leur appartenance à une communauté de joueurs attirés par le type de gameplay proposé, et rémunérés au minimum, de manière à ce que leur motivation intrinsèque soit la plus importante possible. De nombreux jeux proposent par exemple des versions betas dites *fermées*, c'est à dire uniquement accessibles après l'obtention d'une clé de déverrouillage, distribuées en nombre limité sur des sites spécialisés. La seule rétribution consiste à pouvoir jouer gratuitement et avant la sortie officielle du jeu, et garantit donc en partie le niveau de motivation général des joueurs.

Dans le présent modèle, nous ne chercherons également pas à prévoir les préférences du joueur et partons du principe que son objectif est l'objectif principal fourni par le gameplay, et qu'il n'investit pas une partie de son effort pour atteindre d'autres objectifs. Cette hypothèse est une des plus importantes à souligner. Pour reprendre la classification de Richard Bartle, notre modèle s'attache principalement à modéliser le comportement des *achievers*, qui cherchent à atteindre les objectifs principaux du gameplay, le plus efficacement possible. Cette hypothèse implique qu'en l'état, notre modèle aura plus de mal à fournir des résultats précis et pertinents pour des jeux à gameplay très ouvert, dotés d'univers très riches et contraignant faiblement le joueur. Lors de l'interprétation des résultat fournis par notre mesure, le game designer devra tenir compte du fait qu'un gameplay très ouvert peut amener les joueurs à développer des comportements éloignés de ceux d'un *achiever*, compromettant alors la validité des résultats.

Sous cet ensemble d'hypothèses, on peut considérer que l'effort investi par le joueur est similaire à l'effort qu'il suppose devoir investir, et nous pouvons donc négliger la variable de motivation du joueur. En respectant également les hypothèses de la section précédente, nous pouvons considérer que l'effort nécessaire perçu est maximal et donc que l'effort investi par le joueur est également maximal. L'effort fourni devient donc une constante.

6.2.4 Action

Le dernier processus intervenant dans notre modèle de la difficulté d'un challenge (Figure 6.1) est le passage à l'action du joueur. La figure 6.5 isole ce processus et les variables impliquées.

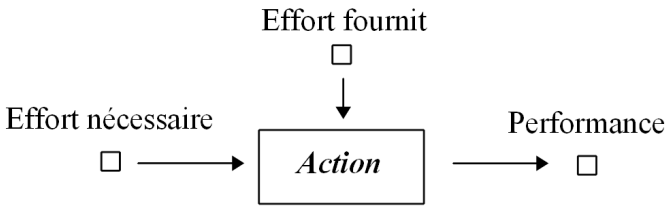

FIGURE 6.5 – Action du joueur.

6.2.4.1 Variables impliquées

L'effort fourni et l'effort nécessaire ont été décrits respectivement section 6.2.3 et section 6.2.1. Lorsque le joueur passe à l'action, ces deux variables conditionnent la performance qu'il va obtenir.

- **Performance :** résultat obtenu par le joueur à la fin du challenge.

Le dictionnaire Larousse donne plusieurs définitions de la performance : « *Résultat obtenu dans l'exécution d'une tâche.* », « *Résultat chiffré (en temps ou en distance) d'un athlète ou d'un cheval à l'issue d'une épreuve.* ». Ces définitions montrent les deux aspects primordiaux d'une mesure de performance : le fait qu'il s'agisse d'un résultat, et qu'il se mesure par rapport à une épreuve, ou une tâche, c'est à dire que la performance se mesure à l'issue d'un intervalle temporel identifié.

Dans le contexte du jeu vidéo, l'intervalle temporel, équivalent d'une tâche ou d'une épreuve, sera le *challenge*. Un challenge se définit par un début et une fin identifiables, et un objectif que le joueur doit atteindre : détruire un ennemi ou trouver un objet par exemple. Il est toujours possible de déterminer, une fois le challenge terminé, si le joueur à réussit ou non à atteindre l'objectif qui lui est associé.

En tenant compte des hypothèses précédentes, l'effort fourni par le joueur sera toujours maximal, c'est à dire qu'il investira toutes les ressources dont il dispose. Mais l'effort nécessaire peut être supérieur à l'effort fourni, auquel cas le joueur n'atteindra pas l'objectif et sa performance diminuera. Si par contre, il fournit un effort supérieur à l'effort nécessaire, il parviendra à atteindre son objectif.

6.2.4.2 Mesurabilité des variables

La mesure de performance la plus simple, évidente et non discutable est une mesure binaire du résultat du joueur : si le joueur a atteint l'objectif, le résultat est maximal, sinon il est nul. De nombreux jeux proposent des mesures plus précises de la performance du joueur, un score, qui évalue le comportement du joueur durant le challenge. Cette mesure est heuristique, définie à priori par le game designer, comme une formule sur différentes variables du jeu, comme par exemple le nombre de points de vie perdus, le nombre d'ennemis abattus ou de blocs empilés. L'inconvénient d'une telle mesure est la difficulté à prouver sa validité : les choix des variables et de la formule qui les combine sont réalisés à priori et ne reflètent pas forcément la performance du joueur de manière précise, mais fournissent une certaine image de cette performance. Face au même challenge, deux game designer peuvent proposer des heuristiques de score différentes, sans qu'il soit possible de déterminer laquelle est la plus précise. La seule mesure sans équivoque est la mesure binaire, car la réussite ou l'échec du joueur fait partie des règles du jeu, est n'est donc pas discutable.

Le résultat du joueur fait partie intégrante de la notion de jeu, et un jeu vidéo détermine forcément si le joueur à réussit à atteindre l'objectif du challenge. Il est donc tout à fait possible d'introduire une portion de code spécialisée chargée d'enregistrer le résultat du joueur, et donc une mesure de la performance du joueur est tout à fait réalisable.

A partir de l'ensemble des hypothèses que nous avons formulé, nous sommes en mesure de simplifier notre modèle de la difficulté d'un jeu vidéo. Nous présentons ce modèle simplifié dans la section suivante.

6.3 Modèle simplifié

Le modèle simplifié de la difficulté, valide si l'on respecte l'ensemble des hypothèses définies précédemment, est présenté figure 6.6.

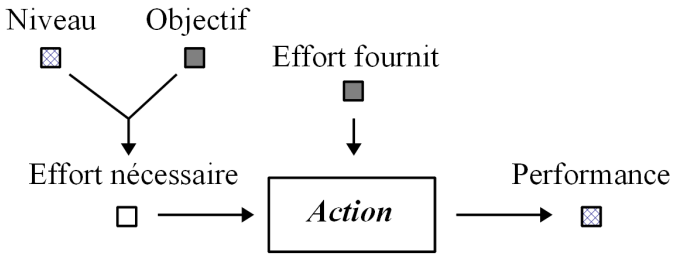

FIGURE 6.6 – Modèle simplifié de difficulté.

Les carrés noirs sont des constante, les carrés hachurés sont des variables mesurables, le carré blanc représente la variables inférée.

En posant un certain nombre d'hypothèses, nous avons pu ramener le modèle complet de la difficulté à un modèle plus simple. Ce modèle nous permet d'établir les propositions suivantes :

Tout d'abord, on remarque que la performance est une image directe de l'effort nécessaire, c'est à dire de la difficulté du challenge. Les hypothèses de la section précédente nous permettent de considérer l'effort fourni par le joueur comme une constante, et donc la performance du joueur dépend uniquement de l'effort nécessaire. En conséquence, la mesure de la performance du joueur va nous permettre d'avoir une image de la difficulté du challenge, qui n'est pas directement mesurable[6]. De la même manière, il donc également possible de mesurer le niveau du joueur à partir de la performance du joueur vis à vis des sous objectifs.

La mesure de la performance et du niveau du joueur seront effectuées de manière statistique, à partir de la fréquence de réussite. Ce type de mesure soulève un problème particulier : ni le niveau du joueur ni la difficulté du challenge ne sont des valeurs constantes. Elles évoluent à chaque fois que le joueur joue, c'est à dire pratique et apprend. Cependant, le niveau du joueur sera plus facile à évaluer car les sous objectifs peuvent être mesurés de manière répétitive sur un court intervalle de temps, pendant lequel on peut faire l'hypothèse que le joueur à peu progressé. Le fait de pouvoir évaluer le niveau du joueur nous permet ensuite de construire la statistique de la difficulté du challenge.

En effet, la difficulté du challenge est directement conditionnée par le niveau du joueur. Nous sommes donc capables de sélectionner, parmi toutes les mesures de performance d'un challenge, celles pour lesquelles la difficulté était similaire : il suffit d'avoir au préalable

6. On peut supposer que la relation entre la performance mesurée et l'effort nécessaire est décroissante et monotone : plus l'effort nécessaire va augmenter, plus la performance du joueur va diminuer. Cette relation n'est valable que si le joueur maintient un effort constant, c'est à dire si l'ensemble des hypothèses de la section précédente sont respectées.

mesuré le niveau du joueur et de sélectionner les mesures pour lesquelles ce niveau est équivalent. De cette manière, nous pourrons mesurer la difficulté de manière statistique, en observant la performance de joueurs pour lesquels la difficulté était similaire.

Nous reprenons et détaillons ce principe de mesure dans la section suivante. Nous présentons notre méthode de mesure du niveau du joueur et de sa performance, ainsi que le calcul de la courbe de difficulté d'un jeu vidéo.

6.4 Les challenges

Comme nous l'avons défini précédemment, la performance du joueur se mesure dans un contexte donné : le challenge. Un challenge correspond, dans un jeu vidéo, à un objectif que le joueur peut chercher à atteindre. Un challenge peut ainsi avoir plusieurs états, selon la relation que le joueur entretient avec cet objectif. L'automate qui définit les états et transitions d'un challenge est représenté figure 6.7.

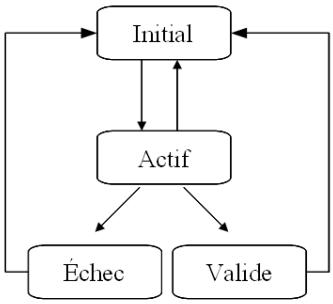

FIGURE 6.7 – Automate d'un challenge.

L'état par défaut d'un challenge est l'état *initial*. Dans cet état, le joueur ne cherche pas encore à atteindre l'objectif du challenge, il est par exemple occupé à atteindre les objectifs des challenges précédents. Si l'objectif du challenge devient un des objectifs du joueur, le challenge passe à l'état *actif*. Lorsque le challenge est à l'état actif, le comportement du joueur est considéré comme lié à ce challenge et ses actions sont donc interprétées en relation avec l'objectif courant. Les deux états *valide* et *échec* correspondent aux deux issues possibles du challenge, c'est à dire lorsque le challenge est *terminé*. Si le joueur arrive à atteindre son objectif, alors le challenge est dans l'état *valide*. Par contre, si le joueur ne parvient pas à atteindre l'objectif alors que le challenge est considéré comme terminé, ce dernier est à l'état *échec*.

Dans un jeu d'aventure par exemple, lorsque le joueur rencontre un groupe d'ennemis, son objectif consiste à les éliminer, et le challenge correspondant à ce groupe d'ennemis devient actif. Toutes les actions du joueur sont alors interprétées en fonction de l'objectif

courant. Chaque coup d'épée en l'air est une action manquée qui nous renseigne par exemple sur ses compétences au combat. Si le joueur perd tous ses points de vie, alors le challenge passe à l'état échec. Si par contre, il parvient à tuer tous ses ennemis sans avoir perdu tous ses points de vie, le challenge passe à l'état valide, le joueur a réussit. On définit donc E l'ensemble des états d'un challenge, ainsi que l'ensemble $E_T \subset E$ des états terminaux d'un challenge :

$$\begin{aligned} E &= \{initial, actif, valide, echec\} \\ E_T &= \{valide, echec\} \end{aligned} \tag{6.1}$$

Etant donné un ensemble de challenges C, on suppose donc qu'il existe une fonction qui donne l'état actuel d'un challenge du jeu :

$$Etat(c) : C \to E \tag{6.2}$$

Depuis le moment où le joueur va découvrir un jeu vidéo jusqu'à sa dernière partie, celui-ci va se confronter de manière répétitive aux différents challenges. Dans le cadre de la mesure difficulté, nous pouvons conserver l'état terminal de chaque tentative de validation du challenge, numérotée $i \in \mathbb{N}$ pour un joueur noté $j \in J$. Nous définissons ensuite la fonction $EtatHisto(c, i, j) : C \times \mathbb{N} \times J \to E_T$ qui fournit l'état terminal d'un challenge lors de la i^{eme} tentative de validation du joueur j :

$$EtatHisto(c, i, j) : C \times \mathbb{N} \times J \to E_T \tag{6.3}$$

Pour simplifier les notations suivantes, on considèrera l'ensemble H de tous les tuples (c, i, j) pour lesquels nous avons enregistré le résultat d'une tentative de validation.

– H est l'ensemble des tuples (c, i, j) pour lesquels $EtatHisto(c, i, j)$ est définie.

Notre objectif consiste à calculer la difficulté d'un challenge. Cette mesure ne doit s'appuyer que sur les seules variables mesurables à partir d'un enregistrement du comportement du joueur dans l'univers virtuel du jeu vidéo : sa performance et son niveau. Les sections suivantes présentent de quelle manière calculer la performance du joueur et son niveau, pour aboutir finalement à l'expression de la difficulté d'un challenge.

6.5 Mesure de la performance

La performance du joueur peut être mesurée sans équivoque si l'on s'appuie sur l'automate de transitions du challenge, présenté à la figure 6.7. En effet, il doit forcément être décidé, pour tout challenge terminé, si le joueur a réussit ou échoué. La notion de résultat valué fait partie de la définition même d'un jeu (voir section **??**), la version la plus simple d'un tel résultat étant une simple valeur binaire, d'échec ou de réussite.

Afin d'obtenir la mesure la plus générale possible, nous avons choisit de nous appuyer uniquement sur la valeur binaire du résultat du challenge. Bien sûr, de nombreux jeux fournissent des résultats valués plus précis au joueur. Mais ces résultats ne traduisent pas forcément avec précision la performance du joueur. Leur valeur est plus précise mais le rapport entre cette valeur et la performance réelle du joueur est toujours discutable. Pourquoi attribuer 100 points à la destruction de l'ennemi A et 200 points pour celle de l'ennemi B ? Y'a t'il forcement une performance deux fois plus importante pour B que pour A ? En nous appuyant uniquement sur le résultat binaire du challenge, nous obtenons une mesure de performance certes en apparence moins précise mais qui repose sur un fondement stable et applicable à n'importe quel jeu.

La performance du joueur $j \in J$ lors de sa i^{eme} tentative de validation du challenge $c \in C$ peut donc être calculée à partir de l'historique des états terminaux de c, par la fonction $Performance(c, i, j)$:

$$Performance(c, i, j) = \begin{cases} 0 & \text{ssi } EtatHisto(c, i, j) = Echec \\ 1 & \text{ssi } EtatHisto(c, i, j) = Valide \end{cases} \tag{6.4}$$

Nous définissons donc la difficulté d'un challenge comme l'opposé de la performance moyenne qu'obtient ce joueur quand il essaie de valider ce challenge. La performance moyenne du joueur correspond à une estimation de la probabilité qu'a ce joueur de gagner, et l'opposé de la probabilité de gagner est la probabilité de perdre. On peut également définir $Performance(c, j)$ comme une variable aléatoire, dont $Performance(c, i, j)$ est une observation pour une tentative i. Notre définition de la difficulté d'un challenge c pour un joueur j est donc la suivante :

$$D(c, j) = Probabilite\{Performance(c, j) = 0\} \tag{6.5}$$

Dans un jeu de plateformes comme Super Mario, par exemple (Figure **??**), la performance moyenne d'un joueur pour un parcours de jeu donné, qu'il doit traverser sans tomber dans les pièges qui lui sont tendus, correspond au rapport du nombre d'échecs sur le nombre total de tentatives.

6.6 Mesure du niveau du joueur

Telle que définie dans la section précédente (eq. 6.5), la difficulté n'est pas estimable de manière statistique, car la probabilité que le joueur échoue varie à chaque fois qu'il essaie. En effet, la probabilité qu'a le joueur de gagner augmente au fur et à mesure qu'il essaie de valider le challenge car ses savoirs et compétences évoluent. Afin de pouvoir calculer une statistique de la difficulté, nous devons calculer cette difficulté non pas pour un joueur donné, mais pour un niveau de joueur donné. Si nous effectuons nos mesures sur un échantillon de

joueurs du même niveau, alors la difficulté sera constante, et nous pourrons l'évaluer de manière statistique. Nous devons donc mesurer le niveau du joueur, et l'intégrer à notre mesure de difficulté.

Dans la section 6.2.1, nous avons défini le niveau du joueur de la façon suivante :

> • **Niveau du joueur :** effort que doit fournir le joueur pour valider un sous-objectif du challenge. Plus le niveau du joueur est élevé, moins l'effort investi est important.

Nous proposons de modéliser le niveau du joueur comme un ensemble de *capacités*. Chaque capacité est une évaluation niveau du joueur, vis à vis d'un sous-objectif particulier.

> • **Capacité :** effort que doit fournir le joueur pour atteindre un sous-objectif particulier.

On remarque la définition d'une capacité est semblable à celle de la difficulté d'un challenge : la difficulté d'un challenge est l'effort nécessaire pour l'objectif du challenge. Evaluer le niveau du joueur revient à calculer la difficulté qu'a le joueur à atteindre certains sous-objectifs. Nous nous heurtons donc au même problème que celui rencontré pour une mesure de la difficulté : le niveau du joueur évolue constamment et empêche a priori d'en calculer une statistique.

Cependant, le niveau du joueur est calculé sur des objectifs à très court terme, sur lesquels la variation du niveau du joueur est minime. Une capacité doit donc être définie sur un but à très court terme, mesurable plusieurs fois pendant la durée du challenge de manière à calculer le niveau du joueur comme une statistique sur ces différentes mesures. Nous supposons donc qu'une mesure du niveau du joueur pendant la durée d'un challenge est stable, et mesurable à partir de la fréquence relative avec laquelle il atteint l'objectif à court terme.

> • **Mesure de capacité :** probabilité que le joueur atteigne un objectif à court terme donné au cours d'un challenge donné.

A chaque capacité correspond donc un but que le joueur cherche à atteindre. Si, à partir d'un enregistrement du comportement du joueur dans le monde virtuel, nous sommes capables d'identifier lorsque le joueur *tente* d'atteindre le but relatif à une capacité, et lorsqu'il *réussit* effectivement à l'atteindre, alors nous pouvons calculer la fréquence à laquelle il parvient à atteindre ce but, et donc évaluer le niveau du joueur, pour cette capacité.

Afin de simplifier encore le modèle, nous proposons de mesurer les capacités du joueur de manière discrète. La précision de discrétisation peut être modifiée sans compromettre la validité du modèle. Dans la suite de ce document, nous fixerons trois niveaux pour chaque capacité : $L = \{Mauvais, Moyen, Bon\}$.

Il est également important de noter qu'une mesure de capacité est relative à un challenge donné. En effet, le challenge détermine le contexte dans lequel le joueur évolue, et conditionne donc l'expression de ses capacités. Dans un jeu de combat, le joueur ne parviendra pas à

effectuer certaines suites d'actions face à un ennemi rapide, alors qu'il y parviendra tout à fait face à un ennemi plus lent. La mesure d'une capacité est donc relative à un challenge particulier, même si elle reflète une caractéristique du joueur.

Finalement, même si le niveau du joueur est considéré stable sur un challenge, nous prenons en compte le fait qu'il varie entre chaque challenge. Le niveau du joueur est donc défini pour une capacité, un challenge, mais également pour une tentative de validation de ce challenge $i \in \mathbb{N}$.

Nous proposons une fonction $Niveau(a, c, i, j)$ qui permet de calculer le niveau $l \in L$ d'un joueur $j \in J$ lors de sa i^{eme} tentative de validation du challenge $c \in C$, pour une capacité $a \in A$:

$$Niveau(a, c, i, j) : A \times C \times \mathbb{N} \times J \rightarrow L \tag{6.6}$$

Toujours dans Super Mario, nous pouvons définir la capacité du joueur à sauter sur certain types de plateformes : saut court, saut long, synchronisation avec une plateforme mouvante, bref, identifier les capacités fondamentales dont le joueur fait preuve tout au long du jeu et qui sont testées de manière répétitive. Ensuite, nous étudions le comportement du joueur lorsqu'il tente de franchir un niveau donné. A chaque fois qu'il est confronté à un saut long par exemple, on comptabilise une tentative pour cette capacité. S'il réussit le saut on comptabilise également une réussite, et s'il se blesse, un échec. La fréquence de réussite nous permet ensuite d'estimer le niveau du joueur, et ce au cours d'une tentative de résolution du challenge général qui consiste à terminer le niveau.

6.7 Equation générale de la difficulté

Le modèle simplifié de la difficulté, présenté figure 6.6, peut être ramené au réseau bayésien de la figure 6.8, si nous ne tenons compte que des variables mesurables, c'est à dire la performance du joueur et son niveau. Les variables du réseau correspondent aux capacités du joueur $\{a_0, a_1, ..., a_n\}$ et ont valeur dans L, et la performance du joueur correspond à la variable $performance$, ayant valeur dans $\{0, 1\}$.

La formule générale de la difficulté peut être déduite de ce réseau bayésien. Nous souhaitons connaitre la probabilité que le joueur perde un challenge, étant donné son niveau pour chaque capacité. Pour réaliser ce réseau bayésien, il faudrait connaitre la probabilité de perdre en fonction de toutes les combinaisons de valeurs possibles des capacités du joueur. Cependant, nous considérons qu'une première mesure simple et utile serait de calculer la difficulté d'un challenge vis à vis de chacune des capacités du joueur, prise séparément. En récoltant suffisamment de données il serait cependant tout à fait possible de spécifier la table de probabilité conditionnelle complète de la variable $performance$, en prenant en compte l'ensemble de ses parents.

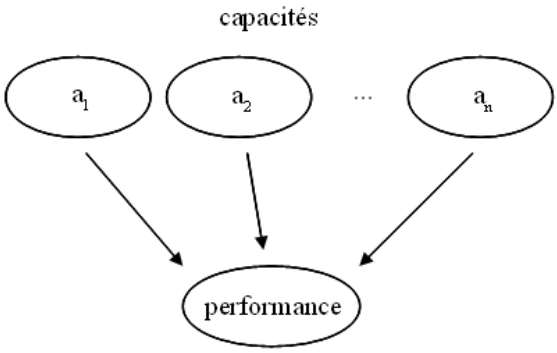

FIGURE 6.8 – Réseau bayésien : causalité entre capacités et performance.

Dans ce cadre, l'équation générale de la difficulté, pour une capacité donnée, est la probabilité de d'échouer, sachant le niveau du joueur pour cette capacité. On définit $Performance(c)$ comme une variable aléatoire dont l'observation pour une tentative de résolution i par un joueur j est donnée par la fonction $Performance(c, i, j)$, et $Niveau(a, c)$ une variable dont $Niveau(a, c, i, j)$ est la valeur pour pour une tentative de résolution i par un joueur j. L'équation générale de la difficulté est alors la suivante :

$$D(a, c) = Probability\{Performance(c) = 0 | Niveau(a, c)\} \tag{6.7}$$

On remarque que dans cette équation, la probabilité est calculée pour tout joueur $j \in J$ et tentative $i \in \mathbb{N}$ de l'historique. En effet, cette équation présente une statistique générale, calculée sur l'ensemble des joueurs. Nous ne souhaitons pas calculer la difficulté pour un joueur particulier ou un nombre d'essais donnés, mais uniquement par rapport à la maitrise d'une capacité. La statistique décrite par l'équation précédente est donc calculée pour toutes les tentatives de tous les joueurs de notre échantillon, pour le challenge et la capacité concernés.

Toujours dans Super Mario, par exemple, calculer la difficulté d'un niveau particulier selon l'équation 6.7 revient à prendre toutes les tentatives à terminer un niveau particulier, pour tous les joueurs ayant participé au test. On trie ensuite toutes ces tentatives en fonction du niveau de jeu dont on fait preuve ces joueurs, pour une capacité donnée. Par exemple, tous les joueurs qui ont réussit plus de quatre vingt pourcents des sauts sur une plateforme mouvante. On obtient donc un sous-échantillon de tentatives de résolutions du challenge ou le niveau du joueur est équivalent, au moins pour la capacité de sauter sur une plateforme mouvante. En calculant la fréquence relative d'échec, on obtient la difficulté de ce niveau de Super Mario, pour les joueurs qui maitrisent le saut sur plateformes mouvantes.

Le fait de formuler la difficulté d'un challenge en fonction d'une capacité a plusieurs

avantages :

- Définir la difficulté d'un challenge pour un niveau de joueur donné nous permet de mettre en évidence et de mesurer la causalité entre le niveau de maitrise du gameplay par le joueur et la difficulté d'un challenge particulier
- Si notre mesure était basée uniquement sur un échantillon de joueurs, sans tenir compte de leur niveau, nous devrions être en mesure de construire un échantillon de joueur représentatif de la population de joueurs concernée, aux niveaux variés et correctement répartis. Le fait de calculer le niveau de chaque joueur permet de connaitre le type de joueur pour lesquels la difficulté a été calculée, et donc le domaine de validité de notre mesure.

Cette formulation a également un inconvénient :

- Définir la difficulté d'un challenge pour un niveau de joueur définit par une seule capacité ne permet pas d'obtenir une véritable indépendance des observations. En effet, plus le nombre de capacités profondément impliquées dans la résolution du challenge augmente, plus ce modèle unidimensionnel doit montrer ses limites. Cependant, ce modèle peut en pratique être tout à fait étendu à plusieurs dimensions, il suffit de spécifier la partie conditionnelle de la probabilité d'échouer comme la conjonction de plusieurs niveaux. Cependant, il faudra également augmenter le nombre d'observations pour pouvoir calculer ce modèle plus juste. Notre modèle à une capacité est donc d'autant plus valable que la capacité choisie est une capacité fondamentale, représentative du niveau du joueur.

Les sections suivantes décrivent le calcul du niveau d'un joueur j pour une capacité a, un challenge c et une tentative i donnés, $Niveau(a, c, i, j)$, ainsi que le calcul de la difficulté d'un challenge en fonction de cette capacité $D(a, c)$. Nous souhaitons calculer la difficulté d'un challenge de manière statistique, à partir d'un enregistrement de son comportement dans l'univers du jeu. La première étape du calcul, décrite dans la section suivante, consiste à construire cet enregistrement.

6.8 Trace d'évènements

Notre objectif consiste à mesurer le niveau du joueur et sa performance à partir d'un enregistrement de son comportement dans le monde virtuel. Il nous est impossible d'enregistrer tous les changements d'états de l'univers virtuel : le nombre de variables est bien trop important. Il est cependant possible de définir une trace des évènements significatifs pour notre mesure, par exemple tous ceux liés aux changements d'états des challenges ainsi que ceux décrivant le comportement du joueur vis-à-vis des capacités que nous souhaitons étudier. Nous introduisons donc les notations suivantes :

- un *évènement de jeu*, noté $ge_{a,c} \in Ge_{a,c}$, est un changement important dans l'état du jeu, en relation avec la capacité a que nous souhaitons mesurer ou avec le changement d'état du challenge c pour lequel nous effectuons la mesure.
- une *trace* $t_{a,c} \in T_{a,c}$ est une suite d'évènements de jeu $ge_{a,c}$ enregistrés entre l'instant ou le challenge c devient actif et l'instant où il atteint un l'état échec ou valide, pour une capacité a donnée.
- la fonction $resultat(t_{a,c}) : T_{a,c} \rightarrow \{valide, echec\}$ détermine si, à la fin d'une trace, le joueur a réussit ou échoué à atteindre l'objectif lié au challenge c.
- au cours de tests, un ensemble de traces d'évènements de jeu sera enregistré pour chaque challenge c. Il existe une fonction qui fournit, pour un challenge c, une capacité a, un joueur j, la i^{me} trace $t_{a,c}$ enregistrée : $trace(a, c, i, j) : A \times C \times \mathbb{N} \times J \rightarrow T_{a,c}$

6.9 Calcul du niveau du joueur

La trace d'évènements de jeu, définie dans la section précédente, va nous permettre d'évaluer les capacités du joueur. Comme expliqué dans la section 6.6, une capacité du joueur correspond à la facilité avec laquelle il est capable d'atteindre un but à court terme, avec l'objectif global de valider le challenge. Pour atteindre cet objectif global, le joueur va exploiter sa capacité, c'est à dire chercher à atteindre de manière répétitive le but à court terme qui lui correspond.

Nous avons défini dans la section 6.6 l'ensemble des niveaux du joueur pour une capacité, $L = \{Mauvais, Moyen, Bon\}$, et le niveau du joueur pour une capacité, $Niveau(a, c, i, j) : A \times C \times \mathbb{N} \times J \rightarrow L$. Le reste de cette section explique le calcul de $Niveau(a, c, i, j)$ à partir d'une trace d'évènements du jeu $t_{a,c} \in T_{a,c}$.

Nous définissons les ensembles et fonctions suivantes, nécessaire au calcul de $Niveau(a, c, i, j)$:

- chaque capacité $a \in A$ que le joueur peut utiliser dans un challenge $c \in C$ doit correspondre à un but à court terme $b \in B$.
- un but $b \in B : Ge_{a,c} \rightarrow \{true, false\}$ est une fonction qui détermine, à partir d'un événement de la trace, si le sous objectif est atteint.
- il existe une fonction $but(a, c) : A \times C \rightarrow B$, définie de manière statique avec l'assistance du game designer, qui fournit le but associé à une capacité $a \in A$ et à un challenge $c \in C$.
- nous devons être capables de compter, dans une trace d'événements de jeu $t_{a,c}$, le nombre de fois que le joueur a *tenté* d'atteindre le but $but(a, c)$, grâce à la fonction $nbEssais(t_{a,c}, a, c)$.
- nous devons être capables de compter, dans une trace d'événements de jeu $t_{a,c}$, le

nombre de fois que le joueur a *réussit* à atteindre le but $but(a, c)$, grâce à la fonction $nbReussit(t_{a,c}, a, c)$.

Si nous savons combien de fois le joueur a tenté et réussit à atteindre le but à court terme associé à une capacité, alors nous pouvons estimer la probabilité qu'il a eu d'atteindre ce but :

$$ProbaBut(t_{a,c}, a, c) = \frac{nbReussit(t_{a,c}, a, c)}{nbEssais(t_{a,c}, a, c)} \tag{6.8}$$

Cette probabilité, $ProbaBut(t, a, c)$, permet d'évaluer le niveau du joueur pour cette capacité, dans le contexte d'un challenge particulier. Comme précédemment, nous ne souhaitons pas conserver une valeur continue du niveau du joueur mais attribuer au joueur un niveau parmi $L = \{Mauvais, Moyen, Bon\}$. La correspondance entre valeurs discrètes et continues peut par exemple être calculée de la façon suivante :

$$Niveau(a, c, i, j) = \begin{cases} mauvais & \text{iff } 0 \leq probaBut(trace(a, c, i, j), a) < 0.3 \\ moyen & \text{iff } 0.3 \leq probaBut(trace(a, c, i, j), a) < 0.6 \\ bon & \text{iff } 0.6 \leq probaBut(trace(a, c, i, j), a) \leq 1 \end{cases} \tag{6.9}$$

L'algorithme suivant précise l'ensemble du calcul de $Niveau(a, c, i, j)$.

```
1  function Niveau(Ability a, Challenge c, Index i, Joueur j)
2           return Level
3  {
4    function b = but(a,c); //Le but à court terme
5    var nbreussit = 0; //Compteur
6    var nbtotal = 0; //Compteur
7    var t = trace(a,c,i,j); //La trace correspondante
8
9    pour chaque sous-trace ti de t où le joueur tente d atteindre b
10   {
11     //une tentative de plus
12     nbtotal++;
13
14     pour chaque evenement ej de la trace ti
15     {
16       //Cette condition dépend de chaque couple (a,c)
17       si b(ej) = true
18         nbreussit++;
19     }
20   }
```

```
21
22    //Probabilité d'atteindre le but à court terme b
23    var p = nbreussit / nbtotal;
24
25    //Discrétisation
26    var level = "mauvais";
27    if(p > 0.3)
28          level = "moyen";
29    if(p>0.6)
30      level = "bon";
31
32    return level;
33  }
```

Il existe plusieurs méthodes pour savoir si le joueur à réussit à atteindre un but à court terme donné (ligne 17 de l'algorithme précédent). Le plus souvent, un évènement de jeu vient confirmer la réussite du joueur. Si on mesure la capacité du joueur à viser correctement, la collision entre le projectile et l'ennemi est un évènement qui valide le fait que le joueur à atteint le sous but.

Nous avons décrit le calcul du niveau du joueur à partir d'une trace des évènements du jeu. Nous pouvons désormais calculer l'équation générale de difficulté $D(a, c)$ (eq. 6.7). Nous présentons ce calcul dans la section suivante.

6.10 Calcul de la difficulté

Nous allons estimer statistiquement la fonction $D(a, c)$, à partir de l'ensemble de traces $T_{a,c}$ enregistrées. L'ensemble $T_{a,c}$ est partitionné en sous ensembles $T_{a,c}^{mauvais}$, $T_{a,c}^{moyen}$, $T_{a,c}^{bon}$, à l'aide de la fonction $Niveau(a, c, i, j)$:

- $T_{a,c}^{mauvais}$ est l'ensemble des traces $t_{a,c} \in T_{a,c}$ d'index i et de joueur j telles que $Niveau(a, c, i, j) = mauvais$
- $T_{a,c}^{moyen}$ est l'ensemble des traces $t_{a,c} \in T_{a,c}$ d'index i et de joueur j telles que $Niveau(a, c, i, j) = moyen$
- $T_{a,c}^{bon}$ est l'ensemble des traces $t_{a,c} \in T_{a,c}$ d'index i et de joueur j telles que $Niveau(a, c, i, j) = bon$.

Cette partition regroupe les traces en sous ensembles selon le niveau du joueur pour une capacité et un challenge donnés. Chaque niveau correspond à une valeur possible de la partie conditionnelle de la fonction $D(a, c)$. L'étape suivante consiste à calculer chacune des probabilités d'échec, pour chaque niveau du joueur.

A ce stade, nous pouvons déterminer, pour chaque trace d'évènements, si le joueur a réussit à valider le challenge. L'évènement correspondant fait partie de la trace et la fonction

$Resultat(t_{a,c})$ se charge de rechercher cet évènement et de nous indiquer si le joueur à gagné ou perdu. Il nous est donc possible de compter, dans chaque sous ensemble de la partition de $T_{a,c}$ définie précédemment, le nombre de traces où le joueur a perdu. La fonction chargée de compter ces traces dans un sous ensemble $T' \in P(T_{a,c})$ sera nommée $nbPerdu(T') : P(T_{a,c}) \rightarrow \mathbb{N}$.

A partir du nombre de traces ayant aboutit à un échec du joueur, nous sommes en mesure d'estimer la probabilité qu'a le joueur de perdre pour chaque niveau, étant donné une compétence et un challenge, c'est à dire pour chaque sous ensemble $T' \in \{T_{a,c}^{mauvais}, T_{a,c}^{moyen}, T_{a,c}^{bon}\}$.

$$ProbaEchec(T') = \frac{nbPerdu(T')}{card(T')} \tag{6.10}$$

Nous somme donc capables de construire la table de probabilité conditionnelle de l'équation générale de la difficulté (eq. 6.7), pour une capacité a et un challenge c. Nous calculons, pour tout $T' \in \{T_{a,c}^{mauvais}, T_{a,c}^{moyen}, T_{a,c}^{bon}\}$ la probabilité d'échec $ProbaEchec(T')$. Ces valeurs correspondent à la probabilité qu'a un joueur de perdre étant donné son niveau pour une capacité donnée, probabilité conditionnelle exprimée par l'équation générale de difficulté, donnée pour rappel ci-après :

$$D(a,c) = Probability\{Performance(c) = 0 | Niveau(a,c)\} \tag{6.11}$$

L'algorithme suivant montre le calcul de l'équation générale de difficulté.

```
1  function Difficulte(Challenge c, Ability a, Level l)
2           return float
3  {
4    var nbEchec = 0; //Compte les echecs
5    var nbTotal = 0; //Total counter
6
7    //On récupère les traces de ce challenge et cette capacité
8    //pour tous les joueurs et indices temporels
9    pour tout joueur j de J
10   {
11     pour toute trace i enregistrée pour j
12     {
13        var t = traces(a,c,i,j);
14        si Niveau(a,c,i,j) == l
15        {
16          si Resultat(t) == echec
17            nbEchec++;
18
19          nbTotal++;
```

```
20          }
21      }
22    }
23
24    return nbEchec/nbTotal;
25  }
```

6.11 Apprentissage du joueur

Les sections précédentes ont permis de décrire le calcul de la difficulté d'un challenge, étant donné le niveau du joueur. Mais comme nous l'avons suggéré précédemment, le niveau du joueur évolue en fonction du temps : ses savoirs et compétences augmentent, au fur et à mesure qu'il explore l'univers du jeu. Une mesure importante et particulièrement utile pour tracer la courbe de difficulté consiste à être capable d'estimer la progression du joueur, c'est à dire de quelle manière, pour une capacité donnée, un joueur va pouvoir changer de niveau. Les sections suivantes décrivent ce calcul.

6.11.1 Correspondance entre capacités locales

Les capacités définies dans les sections précédentes sont des capacités locales, dans le sens où les scores mesurés ne sont valables que dans le contexte d'un challenge donné. Par ailleurs, si un jeu propose de nombreux challenges au joueur, le nombre de capacités impliquées est lui bien plus limité. La précision avec laquelle un joueur vise est par exemple commune à l'ensemble des challenges de la plupart des First Person Shooter.

On peut ainsi considérer qu'un sous-ensemble des challenges proposés dans un jeu vidéo exploite une même capacité a donnée. Pour ce sous-ensemble particulier de challenges, il serait intéressant d'être capable de déterminer une correspondance entre les différents niveaux mesurés pour la capacité a. En effet, même si le contexte varie, on peut supposer qu'un joueur capable d'une grande précision au cours d'un challenge aura beaucoup plus de chances d'être également parmi les joueur les plus précis dans un autre contexte : le joueur transfère ses savoirs et compétences d'un contexte à l'autre et donc les niveaux observés dans deux challenges différents doivent être liés.

Nous définissons une fonction qui, pour deux challenges successifs c et $c^{'}$ et un niveau l d'une capacité a dans le contexte du challenge c, fournit le niveau correspondant dans le challenge $c^{'}$:

$$CorrespNiveau(l, a, c, c^{'}) : L \times A \times C \times C \rightarrow L. \tag{6.12}$$

Nous pouvons définir un modèle probabiliste qui saisisse la dynamique des transitions entre niveau pour une capacité et un couple de challenges donnés. Un tel modèle peut être établit en observant, parmi toutes les traces enregistrées, la fréquence relative de changement de niveau d'une capacité a pour tous les couples de challenges $(c, c^{'})$ joués de manière successive.

La fonction $CorrespNiveau$ est une fonction partielle, car tous les challenges d'un jeu ne seront pas joués de manière successive. Comme nous le verrons lorsque nous établirons un modèle du scénario d'un jeu vidéo, les challenges sont partiellement ordonnés, et parmi tous les couples de $C \times C$, seuls quelques uns sont des couples valides, que le joueur pourra effectivement jouer de manière successive. Cependant, les couples pour lesquels $CorrespNiveau$ n'est pas défini sont des couples pour lesquels nous n'effectuerons pas de requête, car si aucun joueur ne joue jamais le couple $(c, c^{'})$ alors nous ne chercherons pas non plus à prévoir le niveau d'un joueur pour ce couple.

L'intérêt d'une telle fonction consiste à pouvoir prédire un parcours possible du joueur, à partir d'un challenge de départ et d'un niveau donnés. L'équation générale de difficulté (eq. 6.7) nous permet de connaître la probabilité qu'un joueur a d'échouer à un challenge donné, sachant son niveau. Si nous sommes en mesure de connaitre le challenge suivant auquel le joueur sera soumis, à partir d'un modèle du scénario du jeu par exemple, la fonction que nous venons de présenter permettra de prévoir son niveau pour ce challenge et d'estimer ensuite sa probabilité d'échec. Nous serons donc en mesure d'estimer un parcours probable de ce joueur.

6.12 Temps et difficulté

Afin de tracer la courbe de difficulté d'un jeu, nous devons être en mesure de prévoir la durée d'un challenge. En effet, la courbe de difficulté se décrit dans le temps : le game designer doit être en mesure de connaitre la difficulté d'un challenge, autant que de prévoir la durée pendant laquelle le joueur sera soumis à ce niveau de difficulté. Certains challenges peuvent être particulièrement longs, comme un combat contre l'ennemi le plus puissant du niveau, ou très courts, comme lorsque le joueur doit venir à bout d'un petit groupe d'ennemis. La courbe de difficulté permettra de visualiser le rythme de la difficulté du jeu, et pour être en mesure de la tracer, nous devons pouvoir évaluer la durée des différents challenges.

Dans l'optique d'une mesure générale de la difficulté, nous ne pouvons pas déterminer l'évolution de la difficulté *pendant* la durée du challenge. En effet, nous mesurons uniquement le résultat que le joueur obtient à la fin du challenge, qui représente la concrétisation de l'effort fourni pendant toute sa durée. Cet effort a pu varier au cours du challenge, le joueur ayant pu alterner entre différentes tâches : trouver tout d'abord le bon chemin pour se débarrasser ensuite d'un ennemi par exemple. Une manière de comprendre plus finement la

difficulté d'un tel challenge consisterait à le découper en deux challenges successifs et à en évaluer la difficulté séparément.

Nous représentons donc la difficulté d'un challenge comme une courbe plate, d'équation $y = D(a, c)$, sur un segment de droite d'une longueur égale à la durée moyenne observée pour ce challenge. Nous définissons pour cela la fonction $Duree(c, i, j)$, qui fournit la durée d'un challenge pour la i^{eme} tentative d'un joueur i, et $DureeMoyenne(c)$ qui fournit la durée moyenne d'un challenge, calculée sur l'ensemble des traces enregistrées pour ce challenge.

$$Duree(c, i, j) : C \times \mathbb{N} \times J \rightarrow \mathbb{R}. \tag{6.13}$$

$$DureeMoyenne(c) : C \rightarrow \mathbb{R} = \sum_{i,j} \frac{Duree(c, i, j)}{i * j}. \tag{6.14}$$

Néanmoins, il peut être particulièrement intéressant, pour un game designer, d'étudier plus finement la durée d'un challenge qu'à partir d'une simple moyenne. La durée d'un challenge est un élément très important du game design, car la difficulté rythme la progression du joueur. Nous pouvons donc lui fournir une vision complémentaire et plus précise de cette durée, en fournissant sa distribution de probabilité.

6.13 Durée d'un challenge et théorie de la fiabilité

Nous nous inspirons pour cela des théories de la fiabilité, qui décrivent, dans le temps, la probabilité qu'une panne survienne sur un dispositif donné. Nous pouvons tracer le même type de courbe pour un challenge, en remplaçant l'évènement *une panne est survenue* par l'évènement *réussite du joueur* ou *échec du joueur*. Nous somme en effet capables, en considérant la variable aléatoire $Duree(c)$ dont $Duree(c, i, j)$ est une observation pour une trace donnée, d'évaluer la fonction suivante :

$$D(a, c, t) : Probability\{Performance(c) = 0 \wedge Duree(c) < t | Niveau(a, c)\}. \tag{6.15}$$

$D(a, c, t)$ fournit la probabilité, étant donné le niveau du joueur pour une capacité a, qu'il échoue au challenge c avant une durée t. Tracer cette courbe permet au game designer de comprendre si les joueurs échouent plus ou moins rapidement au challenge, et lui fournit une vision plus précise de la durée du challenge. On remarque que notre définition de la difficulté d'un challenge, $D(a, c)$ correspond à la limite, lorsque t tends vers l'infini, de $D(a, c, t)$.

6.14 Tracer la courbe de difficulté

Etant donné les différentes fonction définies dans les sections précédentes, nous sommes en mesure de tracer la courbe de difficulté d'un jeu, à partir d'une requête formulée par le game designer et en partant du principe que nous disposons d'un modèle du scénario du jeu, c'est à dire que nous sommes en mesure de connaitre, à partir d'un challenge et de son état terminal, le challenge suivant que le jeu va proposer au joueur. Ce modèle fera l'objet d'une définition complète au chapitre **??**. Nous pouvons noter cette fonction de la manière suivante :

$$ChallengeSuivant(c, e) : C \times E_T \rightarrow C. \tag{6.16}$$

Pour tracer la courbe de difficulté, nous utilisons les différents modèles probabilistes définis précédemment pour simuler un parcours possible du joueur et mesurer sa difficulté. Etant donné un challenge de départ et le niveau du joueur défini pour chaque capacité, nous pouvons déterminer la probabilité que le joueur a d'échouer, c'est à dire la difficulté du challenge, ainsi que sa durée probable (eq. 6.7 et eq. 6.13). Nous pouvons ainsi tracer une première portion de la courbe de difficulté, pour le niveau de difficulté et la durée que nous venons de calculer.

Il suffit ensuite d'effectuer un tirage en respectant la probabilité d'échec du joueur pour déterminer le résultat du challenge. En fonction de ce résultat, le modèle de scénario doit nous fournir le challenge suivant. Nous sommes également en mesure de déduire le niveau du joueur pour ce nouveau challenge (eq 6.12). A partir de ce nouveau modèle du joueur et du nouveau challenge, nous pouvons répéter la procédure et tracer la portion suivante de la courbe de difficulté.

Le game designer peut répéter sa requête plusieurs fois, et obtenir à chaque fois une courbe de difficulté respectant les probabilités observées lors du playtest, et visualiser ainsi les courbes de difficulté proposées par la portion de gameplay étudiée.

L'algorithme suivant décrit le traçage de la courbe de difficulté :

```
1 function TraceCourbe(Challenge c, Ability a, Level l)
2 {
3   var tailleCourbe = 10; //Taille maximale de la suite de challenge
        mesurée
4
5   if(c != null et tailleCourbe > 0)
6   {
7     d = Difficulte(c,a,l); //Difficulté du challenge pour ce niveau de
          joueur
8     t = Duree(c); //Durée moyenne du challenge
9
```

```
10      //Trace une ligne horizontale de longueur t et de hauteur d.
11      Tracer(d,t);
12
13      //Simule le résultat
14      var tirage = rand();
15      var e = 'valide';
16      if(tirage < d)
17        e = 'echec';
18
19      //Demande le challenge suivant
20      c2 = ChallengeSuivant(c,e);
21
22      //Si il existe un challenge suivant, réévalue le niveau
23      if(c2 != null)
24        l2 = CorrespNiveau(l,a,c,c2);
25
26      c = c2
27      l = l2
28      tailleCourbe--;
29    }
30  }
```

6.15 Synthèse

Dans ce chapitre, nous avons tout d'abord défini le système de variables qui déterminent la difficulté d'un challenge. Nous avons montré que de nombreux facteurs avaient leur importance, comme la perception de l'effort demandé au joueur, ou l'effort réellement fourni par le joueur. Nous avons proposé un ensemble d'hypothèses, sous lesquelles nous pouvons réduire le nombre de variables impliquées et arriver ainsi à mesurer l'effort que demande le challenge, c'est à dire sa difficulté. Ces hypothèses sont, pour rappel, les suivantes :

- Les objectifs que le joueur doit atteindre sont en apparence suffisamment variés
- Le joueur ne parvient pas à atteindre ses objectifs de manière répétée
- En recrutant pour le playtest des joueurs non rémunérés, issus d'une communauté de joueurs attirés par le type de gameplay proposé, la motivation des joueurs sera maximisée.
- Plus le gameplay du jeu sera fermé, c'est à dire limitera la capacité du joueur à créer des meta objectifs, plus on pourra considérer que le joueurs investissent leur effort pour atteindre l'objectif principal.

Nous avons ensuite proposé un modèle qui, étant données les hypothèses précédentes, permet d'évaluer statistiquement la difficulté d'un challenge. Nous estimons la difficulté d'un challenge à partir de la seule mesure sans équivoque de la performance du joueur, son succès ou son échec. Un nombre suffisant d'observations indépendantes nous permet de déterminer une probabilité d'échec, c'est à dire la difficulté du challenge.

Nous avons ensuite proposé d'exprimer la difficulté d'un challenge sous la forme d'une probabilité conditionnée par une mesure des capacités du joueur. Cette formulation a tout d'abord l'avantage d'être plus descriptive, en permettant au game designer de comprendre de quelle manière certaines capacités influent sur la réussite du joueur pour un challenge donné. Ensuite, elle permet d'obtenir une mesure correcte de la difficulté avec un échantillon de joueurs moins importants, mais qui fournissent chacun plusieurs tentatives de résolution du challenge. Ces tentatives ne sont alors plus indépendantes, puisque chaque joueur évolue, apprend à mieux joueur et donc augmente ses chances de réussites. Mais en mesurant le niveau du joueur, la mesure peut être faite pour un ensemble de tentatives réalisées par des joueurs de même niveau, tentatives qui respectent l'indépendance conditionnelle.

Ce modèle présente cependant diverses faiblesses. Tout d'abord, l'indépendance conditionnelle n'est atteinte que si la capacité choisie est la capacité fondamentale. Plus le niveau du joueur est complexe et s'appuie sur une variété de capacités, plus les liens de causalité entre capacité et performance sont nombreux et donc plus l'indépendance conditionnelle ne peut être atteinte qu'en prenant en compte un nombre important de capacités. Le modèle peut être facilement étendu pour tenir compte de plusieurs capacités, mais le nombre de tentatives doit lui aussi augmenter considérablement.

Dans le chapitre suivant, nous présentons une formalisation du parcours du joueur, c'est à dire de la suite de challenges que le joueur va affronter. En effet, notre mesure de difficulté s'appuie sur un découpage du gameplay en une suite de challenges. Le prochain chapitre propose une logique qui permette de spécifier et manipuler graphiquement cette suite de challenges de manière efficace.

Glossary

First Person Shooter First Person Shooter, jeu de tir en vue subjective.. 9, 25

gameplay Rollings et Adams définissent le gameplay de la façon suivante : *Les challenges, ainsi que les actions que le joueur peut entreprendre pour les réussir, constituent le gameplay. D'un manière plus large, on peut considérer le gameplay comme ce que fait le joueur lorsqu'il joue, c'est à dire le type de problèmes qui lui sont posés et la manière dont il les résout.*. 3, 5

MMORPG Massively Multiplayer Online Role Playing Game : jeu de role massivement multijoueurs en ligne.. 9

playtest Un playtest et un processus par lequel un game designer teste un nouveau jeu, pour en éliminer les erreurs logicielles ou pour l'améliorer, avant sa mise sur le marché.. 1, 10, 28, 30

Bibliographie

[Bartle 96] Richard Bartle. *Heart, Clubs, Diamonds, Spades : Players who suit muds.* Volume 1, Issue 1, Journal of MUD Research (http ://www.mud.co.uk/richard/hcds.htm), 1996.

[Capa 08] Rémi L. Capa, Michel Audiffren & Stéphanie Ragot. *The interactive effect of achievement motivation and task difficulty on mental effort.* International Journal of Psychophysiology, vol. 70, no. 2, pages 144 – 150, 2008.

[Rollings 03] Andrew Rollings & Ernest Adams. Andrew rollings and ernest adams on game design. New Riders Publishing, 2003.