Faites vos jeux !

Réflexions sur les approches de facilitation et d'incitation à la création vidéoludique

Damien Djaouti[1&2], Julian Alvarez[1&2], Jean-Pierre Jessel[1], Gilles Methel[2]

[1]IRIT, Université Toulouse III, France, [2]LARA, Université Toulouse II, France.
djaouti@irit.fr, alvarez@irit.fr, jessel@irit.fr, methel@univ-tlse2.fr

MOTS-CLES :

Jeu vidéo, création, outils, facilitation, structure, game design, level design.

RESUME :

Cet article propose une réflexion d'ensemble sur différentes approches utilisées par l'industrie vidéoludique au cours de son histoire pour inciter et faciliter la création de jeux vidéo par les joueurs.
Nous référant à un modèle structurel du jeu vidéo en tant qu'artefact, nous étudierons quatre approches visant à faciliter et inciter les amateurs à créer ou modifier leur propres jeux : les menus de configuration, les éditeurs de niveaux, les mods et les usines à jeux.
L'objectif de cet article est de mettre en évidence la complémentarité de ces différentes approches par l'analyse des différents processus de facilitation qu'elles emploient.

INTRODUCTION

Poussé par le courant du « Web 2.0 », qui prône notamment un Internet centré sur le contenu généré par l'ensemble de ses utilisateurs, le secteur du multimédia dans son ensemble est invité à questionner son mode de fonctionnement général afin de trouver des moyens pour inciter et faciliter la création des amateurs.

Si le secteur de l'Internet semble aujourd'hui découvrir de tels moyens, il existe un domaine pour lequel ces pratiques semblent déjà fortement ancrées au sein de son histoire et culture : **le monde des jeux vidéo**, en tout cas quand il est pratiqué sur ordinateur (PC, Mac, Amstrad CPC, Amiga, Commodore 64…)

En effet, des approches logicielles visant à inciter des joueurs non professionnels à la création vidéoludique sont apparues dès le début des années 1980, pour véritablement se démocratiser au sein de la communauté des joueurs sur ordinateur dans les années 1990, avec la montée en puissance des mods (« Counter-Strike »…) ou encore des usines à jeux (« Klik n'Play », « Game Maker », « RPG Maker »…). Cette démocratisation semble d'ailleurs pouvoir être imputée en grande partie à Internet, qui a permit aux créateurs amateurs de diffuser leurs œuvres, voire leurs outils de création.

Quelles sont alors les différentes approches explorées par l'industrie vidéoludique pour inciter les joueurs à s'adonner à la création ? Comment peuvent s'opérer de tels processus de facilitation, et de quelle manière influencent-ils la création en elle-même ?

Pour tenter d'apporter réponse à ces questionnements, cet article se consacrera à l'analyse de plusieurs approches distinctes d'incitation et de facilitation de la création vidéoludique.

Afin de pouvoir les analyser, nous proposerons tout d'abord un modèle structurel du jeu vidéo, en nous appuyant sur les conclusions de différents travaux théoriques sur ce dernier vu en tant « qu'artefact résultant d'un processus de design ».

En utilisant ce modèle structurel théorique comme grille de lecture, nous passerons ensuite en revue quatre approches et outils de facilitation de la création vidéoludique : les « menus de configuration », les « éditeurs de niveaux », les « mods » et enfin les « usines à jeux ».

1 Un modèle structurel du jeu

D'après Salen & Zimmerman[1], un jeu est un artefact résultant d'un processus de design de la part d'un ou plusieurs auteurs: *« Game design is the process by which a game designer creates a game, to be encountered by a player, from which meaningful play emerges »*[2] [p.80]

Nous renseignant alors plus en détail sur la nature de cet artefact, les deux auteurs nous indiquent que *« a game is a system in which players engage in a artificial conflict, defined by rules, that results in a quantifiable outcome »*[3]. [p.80]

Approfondissant la nature systémique des jeux, Juul[4], également rejoint par Järvinen[5] sur ce point, voit le jeu comme un système à état variable: *« In a litteral sense, a game is a state machine : A game is a machine that can be in different states, it responds differently to the same input at different times, it contains input and output functions and definitions of what state and what input will lead to what following state.[…] When you play a game, you are interacting with the state machine that is the game. »*[6] [p.60]

Par une approche similaire, Paolo Tajè[7] propose de voir le jeu comme une structure possédant six couches, dont quatre sont explicitement internes à l'artefact créé par le game designer : *« Token »* (les éléments du jeu), *« Prop »* (les propriétés de ces éléments), *« Dyn »* (Les actions du joueur sur ces éléments) et *« Goal »* (les objectif du jeu que le joueur doit accomplir).

---

[1] Salen Katie & Zimmerman Eric, « The Rules of Play », MIT Press, 2004.
[2] Traduction personnelle : *« Le"Game design" est le processus par lequel un concepteur de jeux crée un jeu, visant à être utilisé par un joueur, utilisation de laquelle émergera le sens du jeu [ndT : "meaningful play", notion difficile à traduire littéralement] »*
[3] Traduction personnelle : *«Un jeu est un système dans lequel les joueurs participent à un conflit artificiel, définit par des règles, et qui aboutit à un résultat quantifiable ».*
[4] Juul Jesper, « Half-Real: Videogames between real rules and fictionnal worlds », MIT Press, 2005.
[5] Järvinen Aki, « Games without Frontiers: Theories and Methods for Game Studies and Design », Mémoire de doctorat, Université de Tampere, Finlande, 2008.
[6] Traduction personnelle : *« Au sens littéral, un jeu est une machine à état variable : un jeu est une machine qui peut être dans différents états, qui peut répondre de façon différente à la même entrée, il possède des fonctions d'entrée et de sortie et des définitions spécifiant les transitions entre les différents états. […] Quand vous jouez à un jeu, vous interagissez avec la machine à état variable qu'est le jeu. »*
[7] Tajè Paolo, « Gameplay Desconstruction : Elements and Layers », GameCareerGuide, 2007. (http://www.gamecareerguide.com/features/355/gameplay_deconstruction_elements_.php)

Sans oublier que le modèle MDA[8] divise globalement l'expérience vidéoludique en trois parties : *« Mechanics »* (règles), *« Dynamics »* (interaction joueur-jeu dans le cadre de ces règles) et *« Aesthétics »* (émotions internes au joueur suite à ces interactions).
D'après ce modèle, la partie créée directement par le créateur du jeu est celle des « Mechanics », qui constituent donc le fameux « artefact » résultant du processus de design.

Nous pouvons également nous appuyer sur les approches similaires proposées par Crawford[9], Adams et Morris[10] ainsi que Björk et Holopainen[11].

En synthèse de ces travaux sur la structure du jeu vidéo en tant qu'artefact logiciel, nous proposons alors un modèle structurel présentant le jeu vidéo comme un système mettant en relation différentes composantes.
Chacune de ces composantes, bien qu'en étroite relation avec les autres, est alors liée à des méthodes de conception différant de celles utilisées pour les autres parties du système.

Pour résumer, nous considérerons un jeu vidéo comme **un système à état variable**.

Ce système est composé par un ensemble « **d'éléments de jeu »**, l'état de l'ensemble de ces éléments à un instant donné représentant « l'état actuel » du jeu. Chacun de ses éléments est définit par un ensemble de « propriétés »[12], dont l'état courant détermine l'état général de l'élément en question.

Par exemple, pour le jeu *« Pong »* les éléments sont : les raquettes, la balle, les murs et les compteurs de score. L'état de ces éléments correspond à la valeur actuelle de chacune de leurs propriétés, par exemple leur position spatiale, ou encore la valeur numérique qu'ils affichent dans le cas des compteurs.

Mais pour qu'il ait jeu, ces éléments doivent être soumis à un ensemble de **« règles de jeu »** qui permettent la modification de l'état des éléments du jeu, et donc de l'état du jeu lui-même. Structurellement, une règle se compose de deux parties : la **« condition »** *(SI le joueur appuie sur le bouton haut)* et **« l'action »** *(ALORS modifier la propriété « position spatiale » de l'élément « raquette du joueur » de 3 pixels vers le haut).*

Nous pouvons alors diviser le processus de création de tout jeu vidéo en deux « composantes » :

- La création des éléments et de leur état initial, communément appelé **« Level Design »**.
- La création des règles, usuellement dénommée **« Game Design »**.

[8] Hunicke Robin & LeBlanc Marc & Zubek Robert, « MDA: A formal approach to Game Design and Game Research », actes du colloque « Nineteenth National Conference on Artificial Intelligence », 2004.
[9] Crawford Chris, « Chris Crawford on Game Design », New Riders, 2003.
[10] Adams Ernest & Morris Dave, « Game Architecture and Design: A New Edition », New Riders, 2003
[11] Björk Staffan & Holopainen Jussi, « Patterns in Game Design », Charles River Media, 2005.
[12] Référence à la philosophie de programmation informatique dite « orienté objet ».

Typologie des règles de jeu

Pour le cas spécifique du Game Design, nous pouvons aller plus loin et proposer une typologie des règles de jeux, en synthèse des travaux de Frasca[13] et Järvinen[14].
Ainsi, Frasca nous dit qu'un créateur de jeu vidéo dispose de quatre moyens pour diffuser un message : le *thème*, les *règles de manipulation*, les *règles d'objectif* et les *règles meta*.

De son coté, Järvinen met en évidence 5 type de règles : les règles définissant le rôle des éléments de jeu, les règles définissant les actions autorisées, les règles d'environnement, les règles d'implémentation du thème ainsi que les règles définissant les moyens d'action et la communication d'information aux joueur.

En étendant la typologie de règles que nous avons proposé dans nos précédents travaux[15], nous proposons alors une typologie reposant sur cinq types de règles :

- Règles **« Play »** : Règles définissant les moyens d'actions du joueur.
  Ces règles sont caractérisées par une condition liée à l'interface entrante du système, et une action agissant sur les éléments du jeu.
  Exemple : SI j'appuie sur la flèche droite du clavier ALORS mon avatar bouge de 10 pixels vers la droite.

- Règles **« Game »** : Règles définissant les objectif à accomplir.
  Ces règles sont caractérisées par une condition liée aux éléments du jeu et une action produisant un jugement explicite de la performance du joueur. Ce jugement peut ensuite communiqué vers l'interface sortante du système (ajout d'une règle Display) ou directement sur les éléments du jeu (ajout d'une règle World).
  Exemple : SI tout les éléments « ennemis » sont détruits ALORS le joueur a gagné.

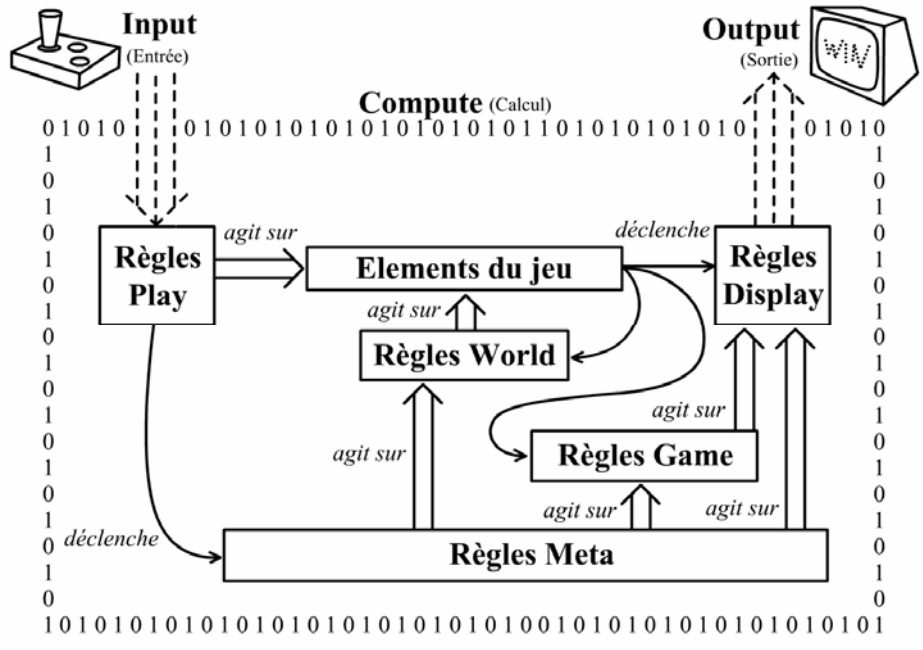
- Règles **« World »** : Règles faisant évoluer le système de manière autonome.
  Ces règles sont caractérisées par une condition liée aux éléments du jeu et par une action qui agit également sur les éléments du jeu.
  Exemple : règles de la Physique (gravité), intelligence artificielle…

- Règles **« Display »** : Règles déterminant la représentation de l'univers virtuel.
  Ces règles se caractérisent par une condition liée aux éléments de jeu et par une action destinée à l'interface sortante du système. Ces règles permettent de « matérialiser » une représentation du monde virtuel sur les périphériques de sortie.
  Exemple : Aspects graphiques, sonores et tactiles du jeu, et toutes les règles qui permettent de modifier ces aspects.

- Règles **« Meta »** : Règles de modification des règles.
  Ces règles sont caractérisées par une condition liée à l'interface entrante et par une action ciblant les règles du jeu. Ces règles permettent de modifier les règles des autres catégories, et permettent ainsi au joueur de « modifier » le jeu.
  Exemple : Menu de configuration présentés dans le paragraphe suivant

---

[13] Frasca Gonzalo, « Simulation versus Narrative: Introduction to Ludology », dans l'ouvrage *"The Videogame Theory Reader"*, Routledge, 2003
[14] Järvinen Aki, « Making and Breaking Games: A typology of Rules », DIGRA Level Up Conference, 2003.
[15] Djaouti Damien & Alvarez Julian & Jessel Jean-Pierre & Methel Gilles, « Play, Game, World: Anatomy of a videogame », actes du colloque *« International Conference on Computer Games: AI, Animation, Mobile, Educational & Serious Games »* (Cgames 2007), 2007.



Typologie des règles de jeux : Play, Game, World, Display et Meta.

2 Du « jeu avec outils » vers « l'outil à jeux »

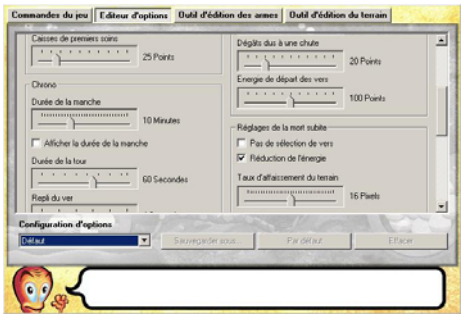
En nous appuyant sur ce modèle, nous allons à présent tenter d'analyser différentes approches destinées à inciter les joueurs à la création ou la modification de jeux vidéo, notamment en leur facilitant l'accès à la création par le biais d'outils étudiés en ce sens.

2.1 Les « menu d'options »

Ajustement de règles existantes

Accompagnant historiquement la migration des jeux vidéo des salles d'arcade vers les consoles de salon et autres ordinateurs personnels, les « menus d'options » permettent de paramétrer les règles du jeu.

Street Fighter 2



Worms 2

Parmi les exemples les plus courants de ces « options » nous trouvons la possibilité de configurer l'interface du jeu (associer un bouton de l'interface entrante à une action dans le jeu). On peut également rencontrer le paramétrage des règles du jeu lui-même par le choix d'un « mode de jeu » général ou plus précisément en ajustant la « difficulté du jeu ».

Si l'on se réfère au modèle structurel présenté précédemment, quelles sont les composantes d'un jeu vidéo que permet de modifier un « menu d'option » donné ?

Nous observons qu'il s'agit ici de la possibilité de modifier les règles du jeu, par le biais de « règles Méta » préalablement définies par le game designer : le joueur ne peut exercer son pouvoir de modification que dans un cadre défini et immuable.

Ces règles Méta permettent ensuite de modifier, respectivement :

- Les règles « Play », dans le cadre des configurations des touches : on modifie la partie « condition » des règles Play. Si le menu propose par exemple de modifier le nombre de projectiles disponibles dans un jeu de tir, il s'agit également d'une modification des règles Play.
- Les règles « Game », pour le cas du réglage du nombre de « vies » ou « continues », de la durée de la partie… et parfois même dans le cadre du paramétrage de la « difficulté » : par exemple, en modifiant la résistance des ennemis, on modifie la condition d'une règle Game.
- Les règles « World » lorsque le menu propose de modifier des paramètres plus spécifiques, par exemple la « puissance de la gravité » ou la « vitesse du jeu ».
- De choisir entre plusieurs ensembles de règles à utiliser, ces dernières pouvant alors être de chacun des types définit par le modèle théorique (cas du « mode de jeu »)
- On notera que dans le cas spécifique du réglage de la difficulté, cela peut aussi permettre de modifier l'état initial du jeu : il est possible de choisir entre plusieurs états initiaux comportant par exemple un nombre différent d'ennemis, de bonus, etc…

D'une manière synthétique, on retiendra que les menus d'options ou de configuration permettent au joueur de personnaliser les règles et éventuellement l'état l'initial du jeu dans un cadre prédéfini par des règles de type « Méta ».
Conçu selon une logique claire de facilitation, aucune connaissance technique ou théorique particulière n'est nécessaire à l'utilisation de ces menus. Nous pouvons finalement observer qu'il s'agit là d'un moyen d'accès extrêmement simple d'utilisation de modification d'un jeu vidéo, la contrepartie étant qu'il demeure forcément très limité : en effet, toutes les modifications permises par ces menus doivent être explicitement prévues par les créateurs originels du jeu.

2.2 Les « éditeurs de niveaux »

Création des états initiaux

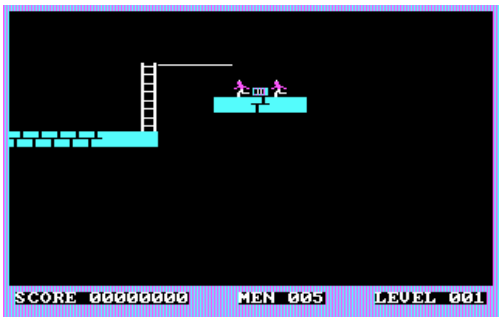
En parallèle aux menus de configuration, il existe de nombreux logiciels permettant de créer des « états initiaux », autrement dit de générer un « univers virtuel » composés d'un nombre fini d'éléments possédant chacun un état de départ.

Originellement destinés aux professionnels réalisants le « Level Design » d'un jeu, ces outils, qui sont généralement des programmes distincts de celui du jeu, ont progressivement été adoptés par les joueurs. Notons qu'à de rares exceptions près, ces outils d'édition de niveau sont toujours spécifiques à un jeu donné.

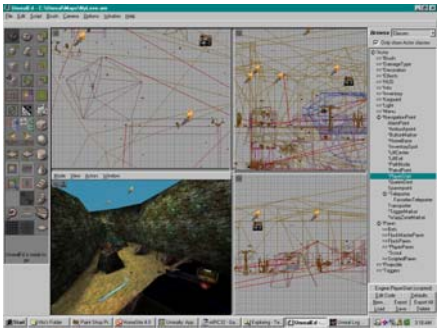
Si une des premières initiatives de ce genre, remonte à 1983 avec le jeu « Lode Runner »[16], la généralisation des éditeurs de niveaux ne fut pour autant pas immédiate. En effet, si une vague

[16] L'histoire veut que l'éditeur de niveau pour « Lode Runner » ait originellement été utilisé par Doug Smith, géniteur du jeu, pour atteindre l'objectif des 150 niveaux qu'il devait réaliser pour le jeu complet. A court

de jeux avec éditeurs de niveaux arriva dans les années 80, et même sur console à l'image du jeu « Excite Bike » en 1984, cette vague fut stoppée par l'arrivée des jeux en « 3D temps réel », au début des années 1990.



Lode Runner


UnrealED (Unreal II)

Est-ce la complexité supplémentaire de la création d'univers tridimensionnels qui aura rebuté les créateurs de l'industrie ?
Toujours en est-il que les premiers éditeurs de niveaux « 3D » disponibles pour le grand public furent créés par des amateurs, aux compétences de « hackers », et qui n'avaient à priori aucun lien avec les créateurs originels des jeux.
Les éditeurs de niveaux « officieux » les plus connus furent sans doute ceux des jeux « Wolfenstein 3D » et « Doom », ces titres faisant partie des pionniers de la 3D temps réel videoludique, en tout cas pour le monde des ordinateurs personnels.
Bien qu'au départ réticente à laisser au joueur la possibilité de modifier ses jeux en dehors des cadres qu'elle avait définie, l'industrie du jeu, sous la pression de ses clients fortement attiré par ces créations « officieuses », en est arrivé en moins d'une décennie à livrer avec certains de ses jeux les outils qu'elle utilise pour les créer, à l'image des « éditeurs d'univers » des derniers opus de la série « Elders Scrolls », les jeux « Morrowind » et « Oblivion ».

Si, pour ces cas précis, l'utilisation de ces logiciels[17] implique un niveau de connaissance technique équivalent à celui des « level designers » professionnels, il est souvent d'usage de créer des éditeurs de niveaux destinées au joueurs, plus facile d'accès car nécessitant un savoir technique moindre.
Cette facilitation de l'utilisation passe, en plus du recours à des règles d'ergonomie et d'utilisabilité sur l'interface, à la limitation des possibilités de l'éditeur de niveaux.
Il n'est d'ailleurs pas rare que ces éditeurs de niveaux « accessibles » soient réalisés d'après les logiciels professionnels, dont les possibilités sont alors bridées.

En nous référant au modèle présenté en première partie, que pouvons-nous observer par rapport aux « éditeurs de niveaux » ?

Clairement, il s'agit ici d'outils qui n'ont pas vocation à la modification des règles de jeu, mais uniquement de l'état initial du jeu, d'un « niveau ».
Ainsi, ces outils permettent de définir, par des interfaces logicielles spécialement étudiées, les éléments de jeu présents au démarrage du jeu, ainsi que leur état de départ (position spatiale, vie, vitesse, etc…), à l'image de ceux utilisés sur le site http://www.whosegame.com.

---

d'idées après la création d'une trentaine de niveaux, il invita les enfants de son quartier à utiliser l'éditeur de niveau, les rémunérant pour chacune de leur création qui serait utilisé pour la version commercialisée du jeu.
[17] Ou pour tout autre éditeur de niveau professionnel livré tel quel, à l'image de « UnrealEd » ou « Build ».

De ce point de vue, les « composantes » de notre modèle théorique modifiables par les éditeurs de niveaux « professionnels » et les éditeurs de niveaux « amateurs » sont identiques. Les différences entre le niveau d'usage des outils proviendront généralement du type de jeu (un éditeur « 3D » étant plus complexe à manier qu'un éditeur « 2D »), et dans l'approche de l'interface : si un logiciel « amateur » visera par exemple à être convivial[18], un outil professionnel devra quant à lui répondre en premier lieu à des critères d'efficacité, quitte à nécessiter un apprentissage spécifique pour son utilisation.

Par exemple, l'éditeur de niveau du jeu 2D « Splodder » ne permet que de créer et déplacer des éléments, alors que l'éditeur de niveau 3D « Worldcraft », utilisé pour le jeu « Half-Life », permet en plus d'associer des scripts d'intelligence artificielle, des propriétés physiques..., à chaque élément du niveau.

2.3 La mode des « Mods »

Modification libre des règles de jeux existants

Loin de se limiter à la création de l'état initial du jeu, les joueurs ont rapidement cherché à modifier les règles des jeux qu'ils pratiquent, au delà des quelques « réglages » parfois proposés par des menus de configuration.

A l'image de la généralisation des éditeurs de niveaux, un usage spontané de la part de nombreux joueurs a conduit ces derniers à modifier des jeux existant, créant ainsi des formes « d'œuvres dérivées » qu'ils proposent ensuite gratuitement aux autres joueurs.
Afin de rester dans les barrières d'une certaine tolérance légale, ces versions modifiées de jeux existants, baptisées « mods »[19], ne sont pas autonomes et nécessitent de posséder le jeu originel pour être utilisées[20].

Counter Strike (Half-life)

Quake Rally (Quake III)

Originellement créé par des amateurs aux compétences de « hackers »[21], les premiers « mods », à l'image des premiers éditeurs de niveaux pour jeux 3D, étaient souvent officieux et obtenu à partir de techniques « d'ingénierie inverse »[22], permettant de retrouver le code source du jeu afin d'en proposer une version modifiée[23].

---

[18] Pour un ensemble de textes approfondissant la notion de convivialité au sein des interfaces logicielles, vous pouvez référer aux actes du colloque « Ludovia 2007 », http://www.ludovia.org
[19] Raccourci évident du mot « modification », en anglais.
[20] Principe bien connu dans le monde vidéoludique de « l'addon », ou « additiel » en bon français.
[21] Virtuose de l'informatique.
[22] Cheminement intellectuel qui consiste à partir du produit finit pour en retrouver le mode de conception. Très utilisé en informatique pour arriver à reconstituer le code source de tout type de logiciel.
[23] Historiquement et culturellement, la pratique du « mod » est sûrement rattachable au caractère « open source » des premiers jeux vidéos sur ordinateur. En effet, un jeu comme « Space War » (1962) a connu un nombre

Au départ réticente à ce genre de pratique, l'industrie du jeu vidéo finit par accepter le principe des « mods », jusqu'à l'encourager en distribuant des outils permettant de créer facilement des modifications pour certains jeux. Baptisés « Software Developement Toolkit », ou SDK, ces ensembles d'outils comprennent des « éditeurs » destinés à modifier les différentes parties d'un jeu : éditeur de niveau, éditeur de graphisme, éditeur de sons, éditeur de règles, voire à l'extrême tout ou partie du « code source informatique » du jeu…

Au-delà de la simplification de la création de mods, l'industrie à également connu des initiatives visant à encourager leur distribution : certains jeux, à l'image de « Max Payne », proposent au joueur, lorsqu'il lance le jeu, de sélectionner un « mod » à employer, par le biais d'un menu de configuration. De même, certains éditeurs ont mis en place sur le site officiel de leurs jeux des sections d'échanges et de distribution de « mods ».

On notera au passage que, contrairement aux menus de configuration et aux éditeurs de niveaux, le terme « mod » ne désigne pas ici des outils de modification ou de création vidéoludique, mais les modifications en elles-mêmes (règles, images, niveaux…)

On peut alors faire un parallèle entre ces « mods » et les différents « mode de jeu » parfois proposés par les menus de configuration. La différence étant que les « modes de jeu » émanent des créateurs originels du jeu et sont livrés directement avec ce dernier.

D'un point de vue structurel, quelles sont les composantes d'un jeu vidéo qui sont modifiées par les « mods » ?

Pour trouver réponse à cette question il suffit de regarder les outils de création de mods distribués par les studios de développement de jeux, et force est de constater qu'il est impossible d'en dégager une tendance unique.
En effet, selon le jeu, on sera en mesure de trouver différent type d'outils, chacun permettant de modifier une composante : état initial pour l'éditeur de niveau, règles « Display » pour les éditeurs de graphismes ou de sons, et règles « Play », « Game » et/ou « World » pour les éditeurs de scripts et le code source.
Cependant, pour chaque jeu les types d'outils ne sont pas les mêmes : là où certains ne proposent qu'un éditeur de niveaux, d'autres proposent le code source intégral du jeu.

La seule tendance générale que l'on peut apparemment dégager en observant ces outils concerne leur niveau de complexité élevé. En effet, qu'il s'agisse des outils utilisés pour le développement du jeu ou d'outils étudiés pour faciliter et inciter à la création, tous nécessitent un certain apprentissage.
Si pour les éditeurs de graphismes et de sons ce savoir repose essentiellement sur des compétences en création graphique ou sonore « généralistes », la modification des règles du jeu par langage de programmation nécessitera, en plus de solides compétence en développement dans le langage utilisé, s'initier à la façon spécifique dont le jeu à été écrit.

---

exponentiel de variantes au fur et à mesure qu'il arrivait dans de nouvelles universités, nombre des nouveaux étudiants à découvrir ce jeu (voire le jeu vidéo tout court) s'empressant d'en réaliser une version modifiée selon leur goût personnel, à partir de la manipulation directe du code source.
Cette pratique se retrouve aussi dès les premiers jours des bornes d'arcade, sous le couvert d'une pratique alors complètement illégale en la forme de la création de « bootlegs ». Concrètement, il s'agit de bornes d'arcade qui proposent des versions modifiées, sans autorisation aucune, de jeux déjà existants. Ces « bootlegs », certes rares et réalisés par des professionnels peu scrupuleux, constituent sans conteste un modèle préfigurant de l'apparition des « mods », nombre de ces derniers étant également nés du mauvais coté de la barrière de la légalité.

Si ces outils peuvent permettre, au prix d'un apprentissage plus ou moins long et complexe, de modifier toutes les composantes d'un jeu donné, il reste une composante qui ne sera à priori pas modifiable : les règles « Méta ».
En effet, les règles « Méta », ou règles de modifications des règles, sont le type de règle utilisé pour écrire les outils de modification proposés. On comprendra donc que la vaste majorité des créateurs de jeux industriels, afin de conserver la liberté de définir ce qui est modifiable et ce qui ne l'est pas, ne donnent pas la possibilité aux joueurs de « créer leurs propres outils » ou de modifier les outils eux-mêmes.
Seul des joueurs possédant alors de solides compétences de « hackers » seront à même d'écrire leur propres outils, à l'image des outils officieux qui furent utilisés lors de la création des premiers « mods ».

Du point de vue des usages, le niveau de complexité général de ces outils n'est pas sans incidence : contrairement à l'usage des menus de configuration ou des éditeurs de niveaux, souvent utilisés par un créateur solitaire, les « mods » sont très souvent créés par des équipes de créateurs, en particuliers lorsque leurs modifications portent sur plusieurs composantes simultanées.

Une sorte de catégorisation empirique existe d'ailleurs au sein des communautés de moddeurs[24] : d'un coté les *« Partial Convertion »*, qui ne modifient qu'une composante à la fois[25], et de l'autre les *« Total Convertion »* qui en modifient plusieurs simultanément, au point parfois de ne plus permettre de reconnaître le jeu originel.

Si les mods riches en modifications, à l'image des « Total Convertion », sont souvent le fruit de plusieurs créateurs associés, la raison en est très simple : la modification en profondeur de chacune des composantes nécessite un savoir distinct de celui utilisés pour modifier les autres composantes.
Dans le monde professionnel, chaque « corps de métier » prend en charge la création d'une composante spécifique, il en est donc de même pour les moddeurs travaillant en équipe : les « levels designers » modifieront les états initiaux, les « infographistes » et « designers sonores » s'occuperont des règles « Display », respectivement pour la partie visuelle et sonore, et enfin les « programmeurs » auront à leur charge de transformer les règles « Play », « Game », et « World ».

En résumé, on observera que les outils de création de « mods », bien qu'à priori destinées à des amateurs, sont plutôt des outils nécessitants un savoir technique pouvant égaler celui des professionnels, en raison d'un processus de facilitation qui s'appuie le moins possible sur une limitation des possibilités offertes par les outils de modifications.
En revanche ces outils, distribués gratuitement, restent toujours spécifiques à un jeu (ou à un ensemble de jeux) donné.

Dans la pratique, l'acquisition de compétences et de méthodes de travail professionnelles par des amateurs afin de pouvoir modifier leurs jeux favoris n'est pas sans soulever la question de la frontière entre monde professionnel et monde amateur. Cette frontière semble d'autant plus floue lorsque l'on remarque que certains « mods », après diffusion gratuite par des amateurs, sont parfois édités commercialement par des éditeurs, à l'image de « Counter-Strike », de « Day of Defeat » ou encore de « Tactical Ops ».

[24] Terme anglophone désignant un auteur de « mod ».
[25] A l'image des « Mutators » pour « Unreal Tournament », dont le nom est très explicite sur le fait qu'ils ne modifient que les règles du jeu, ne permettant de fait que la création de variantes du jeu originel.

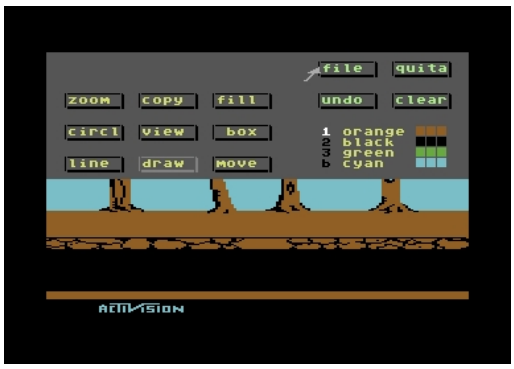
2.4 Les « usines à jeux » et autres « construction kit »

Création de jeux autonomes

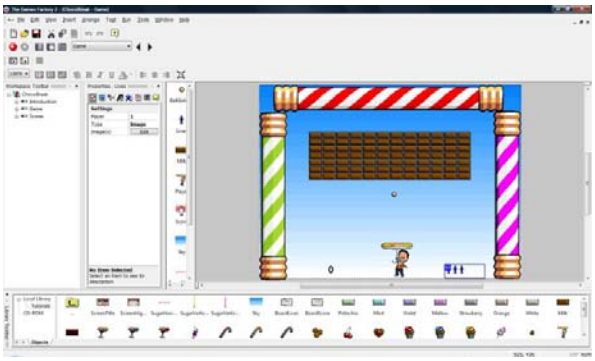
Etape à ce jour la plus complète de l'évolution d'outils visant à faciliter et inciter à la création vidéoludique, il existe nombre de logiciels permettent de créer des jeux vidéos complets et autonomes[26].
Permettant généralement de créer toutes les composantes d'un jeu vidéo, ces logiciels se présentent donc comme des compilations d'outils de « Game Design » et de « Level Design ».
A la différence des outils de « modding[27] », ces outils ne sont pas liés à un jeu donné, et constituent donc des outils de création vidéoludique génériques.

Parmi les noms les plus célèbres de ces outils, dont certains sont d'ailleurs commercialisés, on retrouve notamment la grande famille « click » initié en 1994 par « Klik n'Play », rapidement suivi par les différents opus de « The Game Factory » et « Multimédia Fusion ».
Parmi les pionniers du genre à être reconnus commercialement, nous identifions « Pinball Construction Set » en 1983, ou encore « GameMaker » sur Commodore 64 en 1985.

Mais la grande famille des « usines à jeux » compte aussi des titres tels que « Game Maker », « 3D Game Studio », « The 3D GameMaker », « RPG Maker », « M.U.G.E.N », « Adventure Studio », « Adventure Construction Set », « F.P.S. Creator », « MegaZeux », « Racing Destruction Set », « Shoot'em Up Construction Kit »… et la liste est encore longue.



GameMaker



The Games Factory 2

A l'image de cette grande variété de logiciels existants, force est constater qu'on se retrouve face à un cas similaire à celui outils de moddings, c'est-à-dire que ces outils existent en des complexités et possibilités de création très variables.

Cependant, nous constatons au premier abord qu'il existe des logiciels « professionnels », à l'image de « Virtools », « Multimedia Fusion » ou « 3D Game Studio », qui nécessitent un

[26] Ces différents outils apparaissent à juste titre comme un évolution des « librairies orientées jeu vidéo » et autres « éditeurs de contenus » utilisées par les programmeurs. Concrètement, il s'agit de « librairies de code informatique » prêtes à l'emploi qui permettent d'accélérer le développement d'un jeu vidéo, ainsi que d'outils dédiés à la création des composants graphiques et sonores. De nos jours on trouve des ensembles d'outils complets destinés aux professionnels, que l'on appelle « middleware » ou « moteur de jeu » (Unreal Engine, Renderware, id Tech…)
La différence entre un « middleware » et une « usine à jeux » est parfois ténue, et se situe souvent dans la cible choisie par l'éditeur du logiciels (professionnels ou amateurs), que l'on peut observer par le prix du logiciel. En conséquence, un logiciel « professionnel » sera pensé pour améliorer la productivité et réduire la durée du développement, alors qu'un logiciel « amateur » visera à faciliter son accès au plus grand nombre.
[27] Terme anglophone désignant l'action de créer un « mod »

savoir technique très élevé, et incitent d'ailleurs les créateurs de jeu utilisant ces outils à se regrouper en équipes, comme pour les moddeurs.
A l'image des outils de modding, il s'agit donc ici d'outils destinés en premier lieu à des professionnels, qui sont ensuite détournés dans les usages par des amateurs.

Cependant, si pour les outils de « moddings », qui ne sont que des « suppléments gratuits » livrés avec certains jeux, les créateurs d'outils pouvaient s'affranchir de proposer des outils faciles d'accès en justifiant de leur puissance de modification, la donne est ici différente pour des logiciels qui sont vendus ou distribués pour eux-mêmes.

En effet, même si la portée et la puissance des outils rejoint celles des outils de mods, le but n'est pas ici de modifier un jeu, donc de créer à partir d'une base, mais d'inciter le joueur à créer une œuvre originale.
La problématique relative aux moyens d'opérer une facilitation sans pour autant réduire la liberté de création prend alors ici une importance capitale, plus que dans les autres approches.

Et effectivement, au-delà des outils « professionnels » sus-cités, on remarque qu'il existe des outils plus simples d'accès, et dont le savoir technique requis, s'il n'est certes pas inexistant, reste très modeste comparée à celui demandé par les logiciels provenant directement du monde professionnel.
Pour autant, ces logiciels ne peuvent pas faire trop de concessions sur la liberté de création qu'ils offrent, sous réserve de connaître un échec commercial[28].

Quelles sont donc les solutions alors mises en œuvres par ces créateurs de logiciels pour essayer de concilier liberté créative et facilité d'accès ?

Plusieurs approches différentes semblent coexister :

- *La limitation sur les composantes créables, voire de la liberté créative :*
  Des logiciels comme « The 3D Game Maker » ne permettent en fait de modifier que l'état initial (« Level Design ») et la représentation du jeu (règles « Display »), le reste des règles n'étant pas modifiable.
  Les éléments modifiables ne le sont d'ailleurs qu'à partir de bibliothèques fournies avec le logiciel, facilitant grandement la création tout en la limitant sévèrement.
  Cela permet même à ce logiciel de proposer un mode « création aléatoire », dans lequel un jeu est créé automatiquement par la machine qui assemble alors de manière aléatoire les divers blocs de sa bibliothèque, et configure de même les quelques options proposées au créateur.
  Rare logiciel à opter pour une voie de la facilitation par la limitation extrême, au point de pouvoir se demander s'il permet de « créer » ou simplement de « choisir un choix préexistant », il n'a apparemment pas remporté un grand succès.

- *La limitation des options des outils :*
  Solution visant à réduire grandement le niveau de connaissance technique requis pour la création vidéoludique tout en conservant la possibilité de créer tout les types de composantes.
  Elle consiste, au lieu de supprimer des outils comme précédemment, de tous les

[28] La donne est un peu différente pour les logiciels gratuits, qui soient optent pour la voie de la complexité pour proposer une grande liberté créative, soient la limite énormément pour devenir très simple d'accès. Les logiciels proposant une approche entre ces deux extrêmes sont rares et donc payants, étant apparemment plus complexes et long à inventer en raison d'un existant assez limité ou peu convaincant en la matière.

conserver mais d'en réduire leur puissance, réduisant de fait ce qu'ils permettent de créer sans pour autant condamner toute liberté créatrice.
Le joueur conserve néanmoins une grande palette de possibilité créatives, et chaque logiciel bridant ses propres outils de manière différente, la diversité entre les nombreux logiciels existants confère au joueur un certain choix dans les limitations dont il pourra s'accommoder.
Ainsi, un logiciel de création « 2D » ne pourra pas permettre la création de jeux en « 3D ». Autre exemple, pour l'écriture des règles chaque logiciel propose un « langage de programmation » qui lui est propre, qui se trouve généralement être une version plus ou moins simplifié d'un langage de programmation commun (Basic, Javascript, etc…). A noter que des logiciels comme « The Game Factory » permettrent d'écrire des règles sans passer par l'apprentissage d'un langage de programmation, un « éditeur d'événement » basé sur le modèle « condition / action » permettant d'écrire de l'interactivité en quelques clics.

- *L'automatisation optionnelle :*
  Il s'agit d'une méthode très originale que l'on peut observer dans la famille des produits de la « clickteam », qui consiste à proposer simultanément une approche limitée facile d'accès et une approche ouverte plus complexe.
  Ainsi le créateur peut par exemple, pour un élément donné, choisir différents modes de déplacement pré-configurés (mode voiture de course, mode plateforme…) qu'il peut éventuellement configurer, lui permettant ainsi de créer les règles « Play » dédiées au mouvement des éléments d'une manière certes limitée mais aussi très simple d'accès. Pour ceux souhaitant aller plus loin que ces modes « pré-configurés », ce même logiciel permet de créer de toutes pièces les règles « Play » de mouvement des éléments.
  Dans le même esprit, ce logiciel propose également un mode « pas à pas », qui est une méthode de création des règles de jeu intégré dans une session de jeu. Concrètement, après avoir créé les différents éléments de son jeu et leur avoir attribué un état initial, le joueur lance le mode « pas à pas », qui est un mode de jeu qui détectera automatiquement ce qui lui semblera être des « conditions » de règles pertinentes. Il proposera alors au créateur de choisir le ou les actions à associer à cette condition, lui facilitant ainsi la création des règles Play, Game, World et Display.

Au-delà d'une recherche plus poussée que les autres approches sur la problématique de la facilitation et l'incitation tout en conservant une certaine liberté créative, il est assez difficile de ne pas faire le rapprochement entre certains de ces logiciels et les outils de modding de certains jeux.

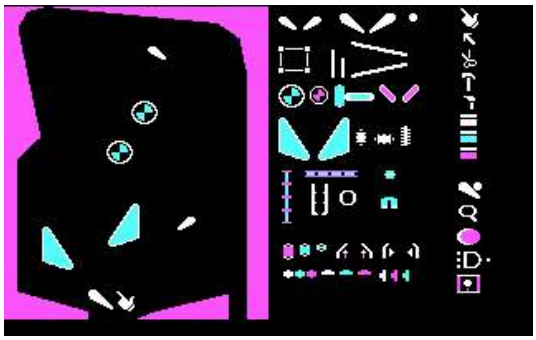
En l'occurrence, certains jeux, par exemple « Quake », livré avec des outils de modifications professionnels[29], ont vu naître des mods qui donnent naissance à des jeux très différents du jeu d'origine, à l'image du célèbre « Quake Rally ».

Pour autant, utiliser « Quake » comme base pour créer un jeu de course n'est pas la voie la plus aisée, ce jeu étant à priori voué à donner naissance à des « mods » du même type de jeu que lui, en l'occurrence des « F.P.S. »[30].

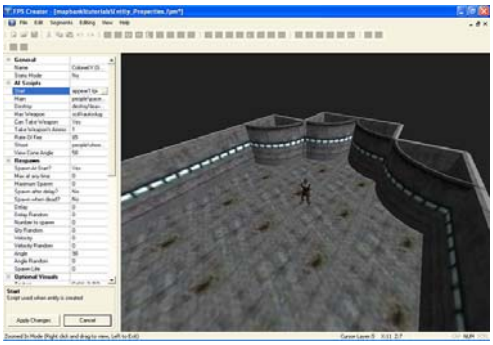
[29] Le jeu possède même son propre langage de programmation, le « Quake C », dérivé du langage C.
[30] Acronyme anglophone de « First Person Shooter », en français « Jeu de tir à la première personne ».
Ce genre de jeu, dont les canons ont été principalement posés par « Wolfenstein 3D » et « Doom », propose au

Ainsi la grande majorité des « mods » d'un jeu donné, resteront globalement dans les canons du genre du jeu d'origine.

Comment alors ne pas faire le rapprochement entre les « mods » et les « usines à jeux » qui sont orientés vers un genre spécifique de jeu ?



Pinball Construction Set



FPS Creator

En effet, si nous avons pour l'instant analysé des logiciels de création vidéoludique généralistes[31], il en existe un nombre conséquent qui soit orienté vers un genre précis de jeu vidéo, à l'image de « M.U.G.E.N. », « Fighter Maker » et « KOF'91 » pour les jeux de combats, ou de logiciels aux titres explicites comme « RPG Maker », « Pinball Construction Set », « Shoot'em Up Construction Kit », « Adventure Construction Set »…

L'existence parallèle de logiciels de création vidéoludique « généralistes » et de logiciels « spécifique à un genre » nous permet d'ailleurs de nous interroger sur la façon dont il est possible, au delà d'un cadre technique, d'inciter et de faciliter la création d'un genre précis de jeu vidéo.
En d'autres termes, quelles sont les stratégies qui permettent de rendre un logiciel spécifique à un type de jeu donné (quand ce n'est pas à un jeu donné comme pour le cas des mods) ?

En analysant ces logiciels « spécifiques » et en les comparant à leurs collègues « génériques », nous pouvons observer les approches suivantes :

- Les outils spécifiques possèdent un certain nombre de règles prédéfinies.
  Ces règles caractérisent généralement un genre de jeu donné :
  - *R.P.G. Maker :* combat au tour par tour, système de point de vie, de magie et de compétence, gestion d'un inventaire, d'une équipe des personnages…
  - *F.P.S. Creator :* vue en 3D à la première personne, l'avatar du joueur peut tirer à partir de différentes armes, présence d'un moteur physique…
  - *M.U.G.E.N. :* chaque joueur ne contrôle qu'un avatar, utilisation d'un système de collision et animation, prédéfinition de règles telles que « gel de l'animation à l'écran », jauge de « super coup », etc…

  Ces règles sont généralement immuables ou configurables dans un cadre restreint et prédéfini. Le fait de réaliser un jeu d'un genre différent de celui du logiciel de création[32] nécessite alors de passer par des méthodes de détournements visant à contourner les « facilitations » mises en place par les auteurs du logiciel.

---

jouer d'évoluer librement dans un univers tridimensionnel en vue subjective, tout en éliminant des autochtones hostiles à grand renfort d'armes à feu.

[31] A quelques détails techniques près, comme « 2D ou 3D », « Vectoriel ou Bitmap », etc…

[32] Un exemple de ce type de détournement est le jeu « J-DAR II », shoot'em up réalisé avec « M.U.G.E.N. », qui est un logiciel de création de jeu de combat. L'auteur a détourné les règles de création de « boules de feu », caractérisant les jeux de combats, pour créer un jeu de tir dans l'espace.

- De leur coté, au-delà des orientations et limitations techniques dont nous avons déjà parlé, les outils génériques ne proposent à priori aucune règle qui ne soit pas modifiable.
  Il peuvent néanmoins, afin de faciliter la création de jeu genre donné, proposer des « exemples » de jeu sur lequel pourra se baser le créateur.
  Nous nous retrouvons alors dans un logique proche des « mods », avec une différence fondamentale : là où les « mods » ne proposeront que rarement des outils qui permettent d'outrepasser des règles de bases du jeu originel, le logiciel de création généraliste n'a aucune limite et propose juste un exemple d'utilisation librement modifiable.

A la lumière de ces observations, nous remarquons donc le fort lien qui existe entre les « mods » et les « usines à jeux », notamment par l'existence des « usines à jeux » liées à une genre spécifique. De plus, parmi les outils proposés par ces deux approches, nous retrouvons des « éditeurs de niveaux », ainsi que des « menus de configuration ».

Nous pouvons alors voir, à travers ces approches, trois étapes d'un même processus de facilitation de la création vidéoludique : en premier lieu les logiciels de création généralistes qui n'opèrent que des choix de limitations ergonomiques ou techniques sans pour autant trop orienter la création. Viennent ensuite les logiciels de création spécifique à un genre, qui, par le biais de règles et mécanismes ludiques préconstruits, orientent la création vers un genre donnée. En poussant ce processus à l'extrême, on retrouve des logiciels de création orientés vers un jeu donné.
La limitation et la mise en place de mécanismes ludiques immuables ou à configuration limité sont alors les principaux outils de ce processus de facilitation de la création vidéoludique.

3 Synthèse analytique

En nous basant sur un modèle structurel poussant à voir le jeu comme un système à état variable, nous mettons en évidence deux grandes composantes de l'artefact jeu vidéo qui résulte du processus de création des auteurs du jeu : l'état initial, rattaché au « Level Design » et les règles permettant les changement d'état, crée lors du processus de « Game Design ».

Nous utilisons alors ce modèle comme grille de lecture sur diverses approches et outils destinés à inciter les joueurs, sous-entendu des créateurs non professionnels, à modifier ou créer des jeux par eux-mêmes.

Si ces pratiques sont variées et notre étude loin d'être exhaustive, ce large panorama nous permet déjà de dégager des grandes tendances :

- La création de jeu vidéo est opérée par des auteurs « professionnels », rémunérés pour leur travail, et des auteurs « amateurs », dont la passion remplace le salaire.
- La création jeu vidéo par les professionnels repose sur des outils impliquant un niveau de connaissance technique assez élevé.
- Afin de faciliter l'accès à la création de jeux vidéo, plusieurs approches existent :
  - La modification de jeux existants
    - Les menus de configuration : aucune compétence particulière.
    - Le « modding »
      - Modification des règles

        - Règles Play / Game / World : compétences en programmation informatique.
        - Règles Display : compétences en création infographique ou sonore.
        - Règles Meta : rarement modifiables, car elles servent au créateurs originel pour définir le cadre des modification autorisées.
    - Modification des états initiaux : création de « niveaux », par le biais d'éditeurs dédiés dont certains, créés pour les joueurs, ne nécessitent pas un niveau de compétence technique très élevé.
- La création de jeux « sans modèle préalable »
    - Outil de création « générique » : permettent de faire tout type de jeu.
    - Outil de création « spécifique » : sont orientés pour la construction d'un genre de jeu précis.

Nous pouvons alors observer que la limitation de la portée des outils de création semble être la voie privilégiée pour faciliter l'accès à la création vidéoludique. S'il en résulte indubitablement une limitation de la liberté créative, cette limitation s'accompagne également d'une baisse du niveau de connaissance technique ou théorique requis pour créer ou modifier un jeu vidéo.

Dans la pratique, cette limitation peut porter sur les composantes qui sont modifiées, à l'image des éditeurs de niveaux ne permettant de modifier que le « Level Design », ou sur la profondeur du cadre des modifications autorisées pour une composante donnée, à l'image des « menus de configuration » qui ne permettent qu'un nombre excessivement restreint de modifications sur les règles de jeu.

Si l'on regarde alors ce panorama dans son ensemble, des menus de configuration aussi limités que faciles d'utilisation aux outils professionnels aussi puissants que complexes, on ne peut alors s'empêcher de constater une certaine **complémentarité** entre toutes ces approches. En effet, si elles permettent chacune de modifier des composantes différentes, elles proposent surtout de le faire avec niveau de complexité très varié.

Pour être plus précis, nous pouvons même observer une sorte de hiérarchie entre les quatre approches analysées, chacune proposant un champ de possibilités créatives de plus en plus ouvert, au prix d'un niveau de savoir technique requis augmentant en conséquence.

Ainsi, les menus de configurations, limités et facile d'accès, sont intégrés directement dans le jeu, et peuvent même constituer une étape obligatoire au démarrage de la partie.
Les éditeurs de niveaux, qui ne permettent à la base que de définir des « états initiaux », sont un des outils utilisés pour la création de mods et se retrouvent intégrés dans la palette des outils proposés par les « usines à jeux ».
Ces deux dernières approches nécessitent donc un savoir supplémentaire, en plus de la maîtrise des éditeurs de niveaux. En conséquence, il n'est pas rare que les deux dernières approches, surtout celles basées sur des outils à l'origine destinés aux professionnels, soit principalement utilisées par des groupes de créateurs, et non plus des personnes seules.

La diversité et la complexité des savoirs nécessaires à la réalisation de mods tels que « Counter-Strike » poussent les « joueurs-créateurs » à s'organiser en équipe réunissant plusieurs « corps de métier », comme les professionnels, à la différence près que ces derniers sont rémunérés pour leur travail.

Au delà des processus de facilitation et d'incitation mis en place par chacune des approches présentées, il semble alors pertinent de voir une forme de processus global de facilitation et d'incitation à la création ludique de qualité professionnelle opéré par la combinaison de toutes ces approches.

En effet, nombreux sont les joueurs, à l'image des auteurs de cet article, a être devenu des créateurs professionnels de jeux vidéo suite à un parcours de « créateur amateur » impliquant la succession des différentes approches mises en avant dans cet article.
Par exemple, les fondateurs du studio Valve Software, actuellement parmi les développeurs les plus renommés de l'industrie grâce à la création de la série de jeux « Half-life », ont commencés par se faire connaître dans le monde « amateur » par la création du mod « Team Fortress » pour le jeu « Quake ». Ironie du sort, un de leur dernier titre est justement la « suite » de mod, sous forme de jeu complet réalisé dans le monde professionnel cette fois.
A noter également que GooseMan, initiateur du célébrissime mod « Counter-Strike » pour « Half-Life », fut ensuite recruté par ce même studio.

Ces différentes approches, en facilitant et incitant les joueurs la création vidéoludique, contribuent donc également à créer des fortes connections entre le milieu de la création vidéoludique « amateur » et le monde industriel.

Notons également que ces approches ont des incidences sur la pratique même du jeu par les joueurs qui ne sont pas forcément « créateurs », comme l'a par exemple étudiée Maude Bonenfant dans son article d'analyse des usages de World of Warcraft[33] : ces approches, en particulier les mods et menus de configuration, permettent aujourd'hui aux joueurs de s'approprier ou se réapproprier un jeu donné.

4 Conclusion

Au travers de son histoire, nous observons que le jeu vidéo pratiqué sur support informatique est riche de différentes approches destinées à globalement faciliter et inciter les joueurs à la création vidéoludique.

En nous appuyant sur un modèle structurel du jeu en tant qu'artefact résultant d'un processus de design, nous avons alors mis en évidence les différences d'approches entre ces dispositifs logiciels.
Mais au-delà de leurs différences, cette analyse fait ressortir leur complémentarité : chacun de ces dispositifs, de part les limites et outils sur lesquels il repose, implique un niveau de compétences technique plus ou moins élevé de la part des utilisateurs.
D'après les quatre approches présentées dans cet article, nous observons que ces niveaux de compétences se répartissent de très faible (menu de configuration) à très élevé (kit de développement pour les mods ou usines à jeux).

[33] Bonenfant Maude, « WoW, un espace de communication, d'entraide et de partage : l'exemple de l'additiel », colloque « Le jeu vidéo : Une expérience multidimensionnelle », Québec, mai 2008.

Cette complémentarité entre les diverses approches de facilitation de la création vidéoludique semble avoir contribué à plusieurs évolutions notables de l'univers des jeux vidéo. Ces évolutions ne sont d'ailleurs pas sans soulever quelques questions quant à la notion de frontière entre professionnels et amateurs, au vu des forts liens qui unissent ces deux mondes au travers des différentes approches présentées.

Mais au-delà d'un moyen de démocratiser l'accès à la création ou d'attirer nombre de créateurs professionnels en leur permettant « d'apprendre le métier » en tant qu'amateur, ces approches semblent aujourd'hui préfigurer d'un nouveau courant vidéoludique, baptisé « Jeu 2.0 » [34].
En effet, riche d'un passé fourni en démarches de démocratisation de la création vidéoludique, une des voies que semble aujourd'hui vouloir emprunter l'industrie du jeu vidéo est celle du « joueur-créateur »

En d'autres termes, il s'agit d'inclure au maximum les créations des joueurs dans le jeu, voire même de baser les mécanismes du jeu sur la créativité des joueurs. Pour cela nous retrouvons des jeux d'outils inspirées des approches présentées dans cet article, à l'image de « Splodder » ou « Little Big Planet », dont le partage de niveaux créés par les joueurs est au cœur du jeu, grâce à des éditeurs de niveaux simple d'accès.[35]

Autre exemple significatif, très récent lors de la rédaction de cet article, le jeu « Spore », sur un concept de Will Wright, propose au joueur de créer sa propre forme de vie et de la faire évoluer, de l'état de bactérie jusqu'à l'age de la conquête interstellaire, où elle devra alors affronter les formes de vies imaginées par d'autres joueur.
Pour autant, il n'agit pas ici d'un jeu en réseau où chacun dirige une civilisation, mais bel et bien d'un jeu « solo » dont l'état initial du jeu sera basé sur l'agglomération de toute les créations des joueurs ayant lancé le jeu avant vous.
A cet usage, un « éditeur de créatures », permettant de concevoir l'apparence de créatures virtuelles, est au cœur des mécanismes de jeu. Il s'agit clairement d'un « éditeur de graphisme », utilisé en tant que base des règles « Play ».
A titre d'exemple, la campagne promotionnelle du jeu a consisté à diffuser gratuitement l'éditeur de créatures trois mois avant la sortie du jeu, afin que les joueurs, une fois qu'il ont créé leur créatures, aient envie d'acheter le jeu qui leur permettra de jouer avec, l'éditeur ne permettant que de les construire et des les admirer.

Signe de l'avenir apparemment prometteur du concept de « joueur-créateur », plus d'un million de créatures[36] ont été créées de part le monde seulement deux semaines après la publication de cet éditeur.

Quelles seront alors les incidences sociales, économiques ou culturelles de cette forte incitation du joueur à exercer sa créativité dans un cadre vidéoludique ?
C'est ce que nous tenterons d'observer en concentrant nos futurs travaux sur le « Jeu 2.0 »[37]…

[34] Référence évidente au « Web 2.0 », courant de pensée prônant un Internet reposant sur le « contenu créé par les utilisateurs ».
[35] Alvarez Julian, « Du jeu vidéo au Serious Game : Approches culturelle, pragmatique et formelle », mémoire de doctorat, université de Toulouse, 2007. [p 26]
[36] Source : « Sporepédia », base d'échange officielle de créatures : http://www.spore.com/sporepedia (25/06/08)
[37] « Game 2.0 » de son nom original