**La théorie des jeux vidéo**

**Introduction**

Les jeux vidéo sont l'un des principaux passe-temps du monde actuel. Environ 3,32 milliards de personnes jouent à des jeux vidéo dans le monde, et leur nombre ne cesse de croître : des joueurs mobiles de tous âges aux personnes qui passent des dizaines d'heures sur leur personnage Elden Ring, en passant par ceux qui les diffusent en ligne pour gagner leur vie et même ceux qui gagnent des milliers en participant à des tournois d'eSports. Les jeux vidéo ne s'adressent plus seulement à un certain public, mais à un grand nombre de personnes aux goûts différents. Mais qu'est-ce qu'un jeu vidéo ? Bien qu'ils aient connu des débuts modestes, les jeux vidéo ont évolué pour englober des professionnels tels que des graphistes, des animateurs, des musiciens, des compositeurs, des écrivains, etc. Ils nécessitent davantage de temps de fabrication et suivent des cycles de commercialisation destinés à créer un engouement. La sortie d'un jeu vidéo est aujourd'hui un événement en soi, avec des critiques et des streamers qui jouent un rôle important dans sa diffusion. Mais à la base, les jeux vidéo ne seraient rien sans les programmeurs qui leurs donne la vie. La conception des jeux a évolué au fil des nombreuses décennies de commercialisation des jeux vidéo, s'adaptant constamment aux nouvelles découvertes de l'industrie du jeu et de la technologie en général, à tel point que la création d'un jeu n'est plus l'apanage des grandes entreprises. À l'issue de ce cours, nous espérons que vous aurez acquis les connaissances nécessaires pour créer votre premier jeu.

**Cycle de vie d’un jeu vidéo**

Étant des produits à plusieurs niveaux, le cycle de développement des jeux vidéo peut durer de nombreuses années et se composer de nombreuses étapes pour garantir que le résultat final est conforme aux normes :

* **Déploiement**

Une fois vos idées principales définies, il est temps de choisir la plateforme sur laquelle vous allez déployer votre projet. Bien que cette décision puisse sembler secondaire, elle est extrêmement importante et aura une incidence sur tous les autres aspects de votre cycle de développement. Par exemple, un jeu développé pour PC offre une plus grande profondeur graphique qu'un jeu conçu pour appareil mobile.

* **Dessin et modélisation**

Chaque jeu a besoin de visuels, c'est une réalité. Comme le développement de jeux vidéo n'est plus l'apanage des grandes entreprises, les graphismes sont disponibles dans de nombreux dépôts en ligne. C'est un avantage tant pour les petites équipes que pour les concepteurs, qui peuvent profiter de ces opportunités pour mettre sous licence leurs sprites, leurs modèles 2D et 3D.

Finalement, vous voulez des graphismes originaux pour que votre jeu se démarque de la concurrence. Pour ce faire, vous pouvez soit travailler seul sur ces graphismes, soit vous associer ou recruter une personne expérimentée dans ce domaine.

* **Moteur de jeu**

Le moteur utilisé pour développer un jeu est certes important, mais il ne devrait pas constituer une préoccupation majeure à ce stade. Tout ce qui concerne le moteur aurait dû être abordé lors de l'étude de faisabilité qui a permis de déterminer la manière dont le projet serait déployé.

* **Conception sonore**

La conception des sons pendant le développement d'un jeu est en fait assez similaire à la conception graphique. Il est possible d'obtenir et d'utiliser des bouche-trous pendant le développement d'un prototype, mais tout comme pour la conception graphique, il est important de distinguer votre jeu en obtenant des licences pour des sons qui seront utilisés plus tard, ou mieux encore, en développant vos propres sons. Cette dernière solution peut s'avérer à la fois plus facile et beaucoup moins coûteuse. Un plugin VST ou un clavier MIDI peut vous fournir des effets pré-mélangés et la possibilité de créer vos propres sons. L'enregistrement personnel peut également constituer une option viable. Vous pouvez ainsi ajouter des composants audios à votre produit sans stress ni coûts liés à l'obtention d'une licence.

* **Prototype**

Maintenant que vous disposez de vos composants audiovisuels et des principales caractéristiques de votre jeu, vous pouvez passer à la création d'un prototype qui les présente. Un bon prototype doit permettre une jouabilité totale, tout en mettant l'accent sur les idées fondamentales du jeu plutôt que de servir comme copie conforme. Il s'agit plutôt d'un moyen de prouver que les concepts fondamentaux de votre jeu sont suffisamment solides pour devenir un produit à part entière.

* **Licences**

L'octroi de licences, comme le démontrent les sections de cette liste relatives aux composants audiovisuels, est un aspect extrêmement important du développement d'un jeu. Même si vous développez vos propres composants, vous serez toujours confronté à d'autres exigences en matière de licence, comme le moteur de votre jeu, par exemple, qui nécessite généralement un abonnement ou un plan de paiement proportionnel aux revenus générés par votre jeu. Selon l'ampleur de votre projet, vous aurez peut-être besoin de services d'acteurs vocaux pour donner vie à vos personnages, et ces coûts doivent également être pris en compte.

Il est important de suivre l'évolution des licences pour chaque élément de votre projet. Toute négligence peut entraîner des problèmes financiers considérables pour vous et votre équipe.

* **Tests Alpha**

Les tests alpha sont la première version d'un projet complet. Il est amusant de constater que la principale préoccupation des développeurs au cours de cette phase est de casser le jeu, de tester les limites de son code, ainsi que la qualité et la cohérence du résultat final. C'est très important, car un jeu bogué sera toujours mal perçu, quelles que soient les idées qui l'ont inspiré. Par conséquent, cette phase consiste en de multiples cycles de collecte de données, de correction des problèmes, de tests et de re-tests jusqu'à ce que le produit final soit suffisamment bon pour être soumis à des bêta-testeurs extérieurs.

* **Tests Beta**

Contrairement au test alpha, qui est principalement réalisé en interne, le test bêta fait appel à des testeurs tiers pour obtenir des opinions en retour. Ce type de contribution est extrêmement important lors du lancement d'un produit, car il permet de tâter le terrain tout en conservant la possibilité de modifier et d'améliorer le jeu en réponse aux critiques.

* **Pré-lancement**

La période de pré-lancement marque le début d'une campagne de marketing intense. Il s'agit notamment de le promouvoir sur les réseaux sociaux, de contacter les créateurs de contenu, les critiques et les influenceurs pour qu'ils contribuent à faire connaître le jeu en en parlant ou en le diffusant auprès de leur public. Une bonne campagne marketing peut véritablement dynamiser la sortie d'un jeu. Il est également possible de lancer une démo limitée en ligne ou lors d'événements. Capcom, le développeur de Street Fighter, met généralement à disposition des versions du jeu avec un nombre limité de personnages pendant les tournois de la version précédente.

* **Financement**

Même si l'on croit souvent à tort que le financement intervient au début du cycle de vie, c'est en fait tout le contraire. Les joueurs ont besoin d'un point d'ancrage, de quelque chose de plus qu'une simple idée de jeu pour savoir que leur contribution a réellement de la valeur.

L'efficacité du crowdfunding dépend entièrement de la qualité de la campagne marketing, car seuls 2 % des joueurs intéressés financent généralement un jeu.

* **Propriété intellectuelle**

Il vous revient de protéger votre projet contre toute fuite ou piratage. Les testeurs alpha internes et les acteurs vocaux doivent signer des accords de non-concurrence et de non-divulgation afin de limiter la quantité d'informations pouvant être dévoilées sans votre consentement ou celui de votre équipe.

Il est également extrêmement important d'enregistrer votre produit pour en protéger les droits d'auteur, faute de quoi vous risquez de ne pas pouvoir lutter contre les tentatives de vol ou de piratage, et votre projet ne vous appartiendra plus. Idéalement, cela devrait être fait avant le test bêta, avant que quiconque en dehors de votre équipe n'y ait accès.

* **Lancement**

Le voici. Tout ce sur quoi vous et votre équipe avez travaillé depuis le début culmine ici pendant cette période. Même si la plupart des jeux ont une « date de lancement », le processus de lancement peut s'étaler sur de nombreux mois. La façon dont vous gérerez ce processus déterminera le succès de votre jeu sur le marché. Chaque diffusion de vidéo, chaque interview en ligne, chaque interaction avec la communauté ou les réseaux sociaux et chaque couverture médiatique est importante pour que le fruit de votre travail soit vraiment mis en lumière.

* **Après-lancement**

Idéalement, la fin du cycle de développement d'un jeu devrait coïncider avec le début d'un autre. Il est important de toujours respecter les calendriers afin d'obtenir des résultats cohérents. Il peut falloir plus de 10 projets pour devenir un développeur de jeux reconnu, mais il ne faut pas se décourager et il est important d'utiliser chaque processus du cycle de vie comme une expérience d'apprentissage pour la prochaine fois.

Maintenant que nous avons détaillé le processus global de développement d'un jeu, il est temps de se concentrer sur la structure technique d’un jeu vidéo habituel.

**GDD**

Le GDD (game design document) est un document essentiel à la conception de votre jeu vidéo. Tenter de travailler sur un jeu sans ce document est un moyen infaillible de perdre le fil de ce que fait chaque composant et d'abîmer complètement le code au cours du développement. Le GDD peut être utilisé par plusieurs développeurs, ce qui permet à l'équipe de cibler les problèmes, de résoudre les conflits de conception et d'organiser efficacement les tâches à venir.

Le GDD doit commencer par une section intitulée « Présentation du jeu ». Comme son nom l'indique, il s'agit d'une vue d'ensemble de ce que votre jeu implique et de quelques points présentant les principales caractéristiques. Une image contenant texte, capture d’écran, Police, document

Description générée automatiquement

Le GDD de Silent Hill, un titre reconnu comme l'un des meilleurs jeux d'horreur et un pilier du genre.

Ensuite, vous devez ajouter une section intitulée « Description du jeu ». Contrairement à la section « Présentation du jeu », celle-ci doit être plus approfondie en décrivant et en expliquant le déroulement général de votre jeu. Décrivez chaque fonctionnalité et montrez comment le joueur interagira avec le jeu.

Enfin, la section la plus fluide est celle des « éléments du jeu ». Peu de concepts de jeu restent inchangés du début à la fin, et il est important de tenir cette section à jour pour assurer la cohérence des idées proposées.

Bien que la documentation puisse sembler fastidieuse, elle est extrêmement importante pour garder vos idées organisées et obtenir un produit plus propre. Elle vous rappellera plus tard que vous avez travaillé dur et que vous méritez votre succès.

**Conception des niveaux**

La conception des niveaux est une phase extrêmement importante du processus de développement d'un jeu, qui englobe la création des étapes, des cartes et des missions. Son objectif est de rendre chaque élément interactif du jeu intéressant, captivant et amusant pour le joueur. La conception des jeux a beaucoup évolué au cours de ces nombreuses décennies d'existence. Au début, les jeux n'étaient développés qu'en 2D et la conception des niveaux était donc plus linéaire. Il s'agissait principalement de placer des obstacles, des bonus et des ennemis sur une carte en 2D à défilement latéral. Mais les choses ont pris un tournant décisif dans les années 90, lorsque les jeux en 3D sont arrivés sur le marché. Une toute nouvelle dimension de la conception des niveaux est alors ajoutée à la formule, et depuis lors, les jeux n'ont cessé de devenir de plus en plus complexes. Certains jeux, comme Dark Souls (2011) et le récent Elden Ring (2022), sont de véritables merveilles en termes d'architecture vidéoludique, repoussant sans cesse les limites de l'interconnectivité entre les niveaux. Au fur et à mesure que la technologie utilisée pour créer des jeux vidéo s'améliore, le niveau de complexité des jeux et les responsabilités des concepteurs augmentent également. Plus un niveau est détaillé, plus il reçoit d'éloges de la part des joueurs. Il ne s'agit pas nécessairement de prouesses graphiques, mais plutôt de rendre chaque niveau mémorable, qu'il s'agisse d'un environnement unique ou d'interactions et d'événements amusants avec lesquels le joueur peut interagir. Les niveaux ne sont cependant pas conçus selon des règles strictes. Même au sein d'un même genre, les approches peuvent varier considérablement. Par exemple, deux des séries de jeux de plateforme les plus populaires sont Super Mario Bros et Sonic the Hedgehog. L'objectif est généralement d'aller d'un point A à un point B, mais les méthodes pour augmenter le score sont très différentes d'un jeu à l'autre. Les jeux de Mario comportent généralement des objets à collectionner cachés et sont donc plus lents, et plus longs. Les jeux de Sonic sont axés sur la vitesse, et il s'agit d'aller du début à la fin le plus rapidement possible. Votre score augmente proportionnellement à la vitesse à laquelle vous avez franchi le niveau. Bien qu'elles puissent sembler minimes, ces différences expliquent en partie pourquoi les deux séries attirent des publics différents. De nombreuses compagnies AAA disposent même d'équipes exclusivement dédiées à la conception des niveaux.

La chose la plus importante à retenir est de toujours concevoir en ayant un objectif clair en tête. Un bon concepteur de niveaux donnera toujours la priorité à l'immersion. Chaque obstacle, chaque élément de l'environnement et chaque ennemi doit avoir une raison d'être, une utilité, plutôt que de simplement exister.

La difficulté est un aspect parfois controversé de la conception des niveaux. Dans l'idéal, la difficulté doit être gratifiante et ne doit pas punir le joueur sans raison. Elle doit corriger la négligence, mais aussi amener les joueurs à être plus attentifs aux indices. Elle ne doit pas être aléatoire, mais soigneusement conçue pour développer les compétences du joueur à mesure qu'il progresse dans le jeu.

En règle générale, la difficulté augmente à mesure que le joueur progresse. Plus le joueur progresse dans le contenu d'un jeu, plus celui-ci devient difficile. Toutefois, cette augmentation doit être progressive. Les pics de difficulté suivis d'une stagnation sont généralement mal perçus, car ils sont artificiels et source de frustration.

En ce qui concerne les aspects plus techniques, tout dans un niveau est généralement créé à l'aide de nœuds. Les nœuds constituent le squelette de tout jeu : ils peuvent servir à simuler un mouvement, à jouer un son, ou à peu près n'importe quelle « action » que vous souhaitez voir prendre à votre composant.

Comme les nœuds englobent des données utilisées pour créer de nombreuses choses différentes, il est important de les organiser, sous peine de se retrouver avec un code désordonné. C'est pourquoi ils sont regroupés en scènes. Une scène est généralement associée à un personnage, un élément, un menu ou, plus important encore, à un niveau entier. Une scène peut elle-même contenir d'autres scènes. Une scène peut également être réutilisée, par exemple lorsqu'un objet ou un ennemi peut être trouvé plusieurs fois dans un même niveau ou jeu.

Une image contenant texte, capture d’écran, multimédia, logiciel

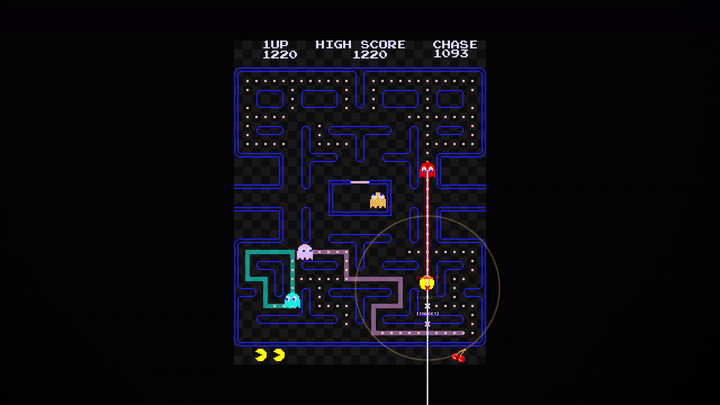
Description générée automatiquementPar conséquent, la construction d'un niveau de jeu consiste plutôt à créer d'abord des composants plus petits, puis à les étendre de plus en plus jusqu'à ce que l'on obtienne le résultat souhaité.

Dans cette image, nous pouvons trouver la scène « Main » qui contient un niveau possède ensuite des scènes « Player », « Enemy » et CanvasLayer. Le joueur et l'ennemi ont également leurs propres nœuds enfants représentant leurs sprites et leur boîte de collision leur permettant d'interagir avec leur environnement. Le joueur possède ses propres sons et une caméra qui le suit lorsqu'il se déplace. CanvasLayer est utilisé pour afficher l'interface utilisateur qui suit le joueur tout au long du niveau (barre de santé, score, durée de jeu, etc.).

**IA ennemie**

L'IA fait fureur aujourd'hui. La technologie GPT a constitué une avancée incroyable et a changé le paysage de nombreux aspects de notre vie. Cependant, l'IA a toujours joué un rôle important dans la conception des jeux.

En 1948, les informaticiens Alan Turing et David Champernowne ont commencé à développer le premier programme informatique algorithmique, Turochamp, qui permettait de jouer aux échecs en ligne. Cependant, les limitations matérielles de l'époque ne permettaient pas de l'exécuter sur des ordinateurs physiques, et une grande partie du code a été perdue. Alors que l'IA ennemie, telle que nous la connaissons, n'était pas vraiment présente dans la plupart des jeux d'arcade, la situation a pris un tournant en 1980, lorsque Namco, l'un des principaux développeurs de jeux d'arcade, a sorti l'un des jeux les plus emblématiques et les plus marquants de l'industrie : Pac-Man.

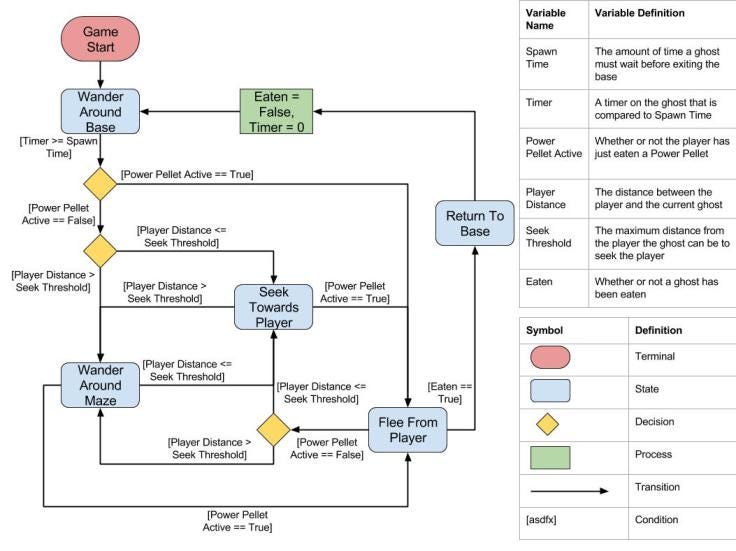


Un visuel de la méthode de chasse de chaque fantôme.

Dans ce jeu, les quatre fantômes (Inky, Blinky, Pinky et Clyde) tentent tous de poursuivre Pac-Man de différentes manières. Blinky est le plus simple, mais peut-être le plus efficace : il se dirige directement vers l'endroit où se trouve Pac-Man en empruntant le chemin le plus court possible. Pinky se dirige vers l'endroit où Pac-Man se trouvera d'ici quelques secondes s'il continue sur sa lancée. Inky se base non seulement sur Pac-Man, mais aussi sur Blinky, et cherche à être à tout moment du côté opposé à Blinky par rapport à Pac-Man. Clyde garde ses distances et espère acculer sa proie à distance. Cependant, se concentrer exclusivement sur un ou deux fantômes revient à se laisser piéger par les autres. C'était la première fois que la difficulté n'était pas liée à la vitesse des balles ou au nombre d'ennemis.

Cela a non seulement donné au jeu un niveau de complexité unique, mais a aussi conféré une personnalité aux fantômes. Blinky est un peu bête, Pinky et Inky sont stratégiques et Clyde est patient. Cela a ajouté un incroyable sentiment d'immersion au jeu vidéo, et c'est l’une des plus importantes raisons pour lesquelles ce média a pu prospérer pendant si longtemps.

Cela a préparé le terrain pour 2000, où la complexité de l'IA ennemie a explosé. Deux des jeux les plus connus pour leur IA impressionnante sont Half-Life et Halo 2. Half-Life utilise une approche très simple de l'IA, appelée machine à états finis. Le comportement d'un PNJ est défini par un état, par exemple patrouiller dans une zone de la carte ou attaquer le joueur, puis des événements qui expliquent pourquoi il passe d'un état à l'autre sont créés. Un soldat peut par exemple continuer à patrouiller sur la carte, en se déplaçant entre les points de passage, jusqu'à ce qu'il aperçoive le joueur, après quoi il commencera à tirer. On appelle cela aujourd'hui les ennemis ayant une « portée d'aggro ». Cependant, ils peuvent ensuite battre en retraite si le joueur s'approche trop ou s'ils subissent trop de dégâts, ce qui donne l'impression que l'ennemi est « réel » comme rien d'autre. L'une des évolutions a consisté à créer des machines à états finis hiérarchiques, qui sont, par essence, des machines à états à l'intérieur d'autres machines à états. Ainsi, vous pouvez avoir une machine d'état pour les comportements de patrouille et d'enquête, mais si vous passez au combat, elle bascule sur une machine d'état différente dans le même système. Cela permet à l'IA itinérante et à l'IA de combat d'établir des priorités différentes, ce qui vous permet de structurer le système en fonction de vos besoins. Cette évolution s'est poursuivie avec l'introduction des arbres de comportement dans Halo 2. Dans un arbre de comportement, tous les comportements sont contenus dans une grande structure arborescente. À mesure que vous descendez de la racine vers les comportements situés sur les bords, vous définissez des règles qui dictent quelles branches seront actives en fonction de l'action se déroulant dans le monde pendant que vous jouez. Le jeu peut alors effectuer des actions individuelles ou les enchaîner pour créer des séquences de comportements intelligents. Cette fonctionnalité est très utile dans des jeux comme Halo, où les situations peuvent se présenter rapidement. Par exemple, il est possible d'avoir un sous-arbre entier qui détermine les actions à effectuer si le joueur se trouve dans un véhicule, car les options auxquelles il a accès sont radicalement différentes. Cela ajoute de la profondeur au combat en véhicule et donne l'impression de se trouver dans un véritable scénario de guerre. Il est possible de réinitialiser très rapidement le jeu si quelque chose change. C'est la raison pour laquelle les grognards paniquent lorsque vous tuez un ennemi d'élite devant eux, car le fait de tuer le haut gradé à proximité oblige tous les PNJ à réinitialiser leur arbre de comportement, car il a débloqué un nouveau comportement qu'ils doivent adopter. La fluidité du jeu s'en trouve véritablement améliorée et, encore aujourd'hui, la série Halo est louée pour l'utilisation magistrale de l'IA ennemie. Pour simplifier : dans Halo 1, les ennemis utilisent leur IA pour simuler des « sens » pour percevoir l’information autour de lui, et réagit de manière appropriée : ne rien faire, aller à l’offensive, aller à la défensive, ou s’enfuir. C’est vraiment comme du « if/then » de la programmation qu’on connait tous. Dans Halo 2 : l’IA permet l’entraide et les formations de groupes entre les ennemis. L’IA permet un ennemi d’envoyer des requêtes a d’autre ennemis, et d’effectuer un évènement de groupe.

Voici quelques schémas qui expliquent le fonctionnement de l'IA des ennemis de Pac-Man, Halo et Half-Life :

