

ARCHITECTURE LOGICIEL RPG

SOMMAIRE :

Ø Présentation RPG

Ø Architecture RPG

Ø Explication utilisation architecture

Ø Autre Architecture utilisée

Ø Comparaison avec concurrent

I) Présentation :

Les RPG (Role-Playing Games) sont des jeux de rôle dans lesquels les joueurs incarnent des personnages et vivent des aventures fantastiques ou réalistes, en fonction de l'univers du jeu. Dans un RPG, le joueur doit créer son personnage, en choisissant son apparence physique, ses compétences et ses caractéristiques. Ensuite, il doit explorer un monde virtuel, interagir avec des personnages non-joueurs et accomplir des quêtes qui lui permettront de progresser dans l'histoire et de gagner de l'expérience pour améliorer son personnage. Les RPG peuvent être joués en solo ou en multijoueur en ligne, permettant ainsi aux joueurs de collaborer ou de s'affronter dans des batailles épiques. Les RPG se caractérisent également par leur grande durée de vie, avec des histoires complexes et des quêtes multiples qui peuvent prendre des dizaines ou des centaines d'heures à terminer. Les RPG peuvent être classés en plusieurs sous-genres, tels que les RPG d'action (qui combinent des éléments de jeu d'action avec des éléments de jeu de rôle), les RPG tactiques (qui se concentrent sur la stratégie et la tactique au lieu de l'exploration), les RPG en ligne massivement multijoueur (MMORPG) et les RPG de simulation de vie (qui permettent aux joueurs de vivre une vie virtuelle complète dans un monde fantastique). Certains des RPG les plus populaires incluent la série Final Fantasy, Skyrim, The Witcher, Mass Effect et World of Warcraft. Les RPG offrent une expérience immersive unique qui permet aux joueurs

de vivre des aventures fantastiques et de développer des personnages uniques et puissants.

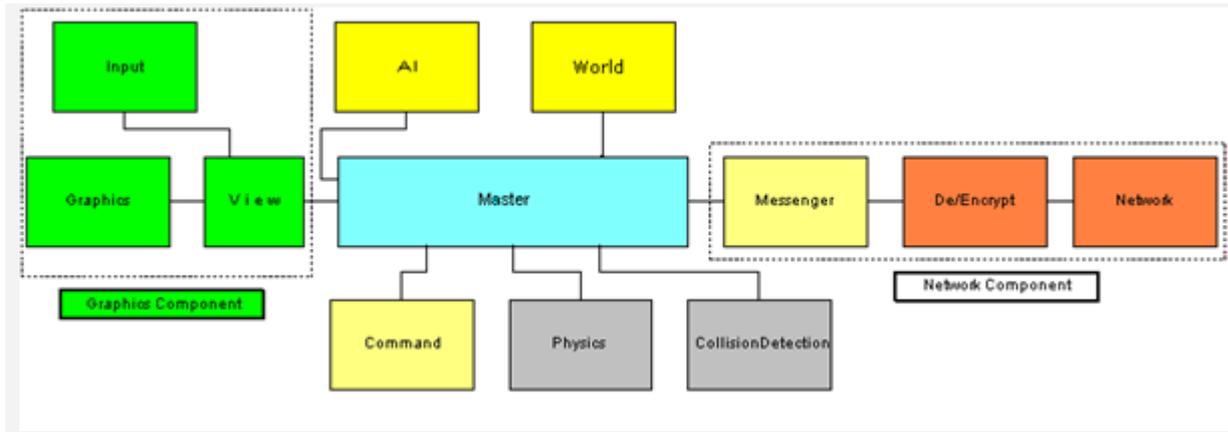
II) Architecture RPG

Il existe plusieurs types de RPG, nous avons donc décidé de parler du MMORPG. Les MMORPG utilisent l'architecture client-serveur, l'architecture client-serveur est un modèle informatique qui est utilisé pour fournir des services informatiques. Elle est composée de deux parties principales : un client et un serveur. Le client est un programme ou une application qui est installé sur l'ordinateur ou l'appareil de l'utilisateur. Il envoie des demandes au serveur et reçoit les réponses correspondantes. Le client est responsable de l'interface utilisateur et de l'interaction avec l'utilisateur. Le serveur, quant à lui, est un programme ou une application qui est installé sur un ordinateur ou un appareil qui est connecté à un réseau. Il reçoit les demandes du client, effectue les traitements nécessaires, et renvoie les résultats au client. Le serveur est responsable du stockage des données, de la gestion des connexions réseau et des opérations de traitement des données. L'architecture client-serveur permet aux clients d'accéder à des ressources et des services qui ne sont pas disponibles localement sur leur ordinateur ou leur appareil, tels que des fichiers, des bases de données ou des applications. Les clients et les serveurs peuvent être situés sur des ordinateurs différents, mais connectés par un réseau, ce qui permet aux clients d'accéder aux services et aux ressources du serveur à distance. Cette architecture est utilisée dans de nombreuses entreprises telles que les banques et les institutions financières, Amazon Web Services, Microsoft Azure. Elle est très populaire car elle permet une communication efficace entre les clients et les serveurs, et permet également une gestion centralisée des données et des ressources.

III) Explication utilisation architecture

Client

L'architecture de l'application côté client est conçue pour déléguer le plus de traitements possibles aux clients afin d'améliorer l'efficacité des serveurs.



Le composant "Master" est responsable de la transmission des messages et des informations entre les différents modules du client. Il pilote sept composants : AI, Graphics, Command, Physics, Collision et Network.

Le "World" stocke localement une copie des données du serveur pour limiter les requêtes du client au serveur. Le "World" tient le rôle de modèle dans le MVC.

Le composant "AI" gère les objets liés au joueur, mais non contrôlés par ce dernier. Le module Graphics est responsable de l'affichage graphique, de la vue et des entrées système. Le module Inputs permet l'abstraction du type d'entrée, tel que joystick, clavier, souris, manette, etc.

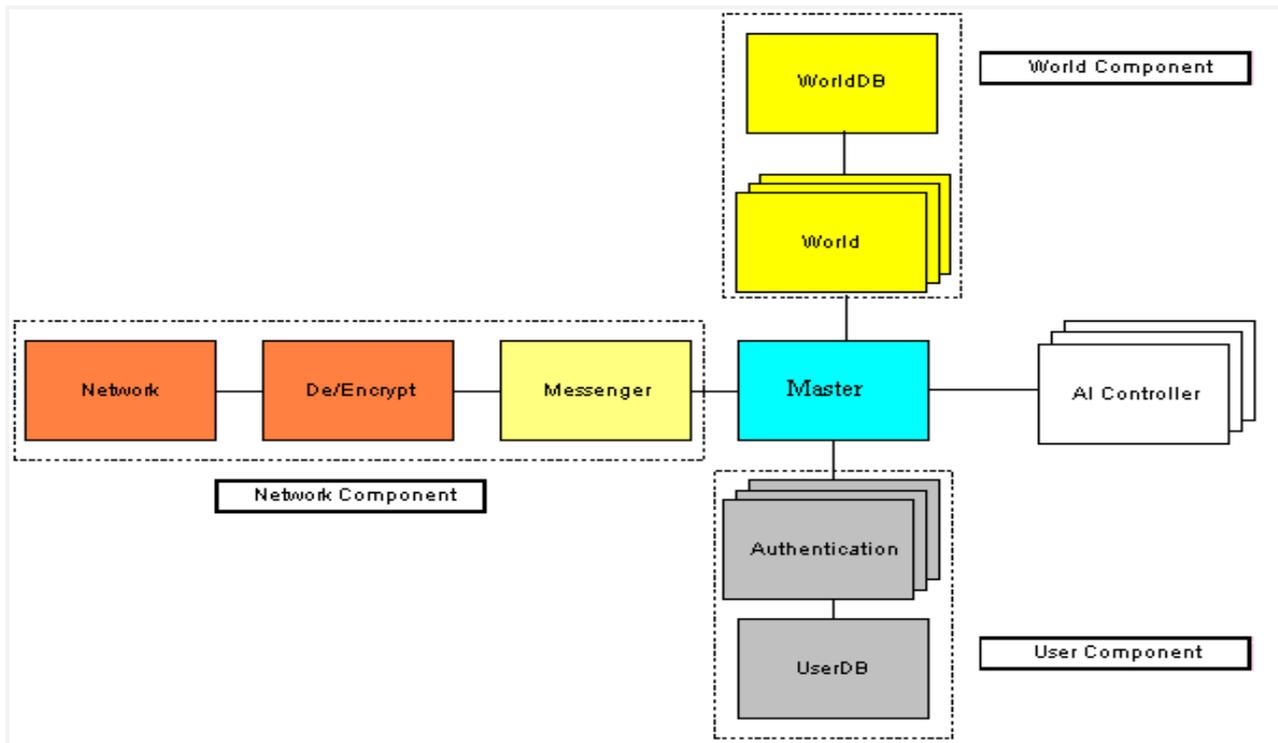
Le composant "Command" prend en charge des fonctions qui n'ont pas directement d'impact sur le gameplay, telles que la synchronisation de l'horloge du client avec celle du serveur et les statistiques relatives aux actions du joueur.

Le module Physics permet de prédire l'état d'un objet suivant la physique du monde du MMORPG et permet d'abstraire toutes les règles physiques du reste de l'application. Le module "CollisionDetection" gère la gestion des collisions.

Enfin, le module "Network" permet la communication entre le client et le serveur et contient le "Messenger", responsable de la formation des messages pour le serveur, le module d'encryption et le module de gestion de la communication avec le serveur.

Serveur

Nous allons ensuite regarder l'architecture côté serveur. Le schéma ci-dessous présente une vue logique des différents composants nécessaires au fonctionnement du jeu. Le Master est le composant qui va régir l'interaction de tous les autres modules. D'un point de vue matériel, c'est lui qui va s'assurer que tous les serveurs fonctionnent bien et relayer les messages entre serveurs.



Le module World contient la base de données contenant toutes les informations relatives au monde du jeu et aux objets qu'il contient. Du fait de la lenteur des accès aux bases de données, c'est une copie en mémoire de cette base de données permettra au Master d'accéder aux données du jeu sans ces surcoûts. Compte tenu de la taille du monde et du nombre d'objets à gérer, cette copie mémoire de la base de données est répartie sur plusieurs serveurs. Souvent plusieurs dizaines voire plus (ou bien plus!).

Le module AI Controler est lui aussi généralement déployé sur plusieurs serveurs. Il s'agit des serveurs qui vont contrôler l'IA, l'Intelligence Artificielle. Ces derniers interagissent avec le Master comme des pseudos-clients.

Le module User contient lui la base de données des utilisateurs. Cette base de données contient toutes les informations telles que le profil du joueur, ses statistiques, ses login et mots de passe de connexion au jeu... Dans l'architecture d'un MMORPG, il y a toujours un pool de serveurs dédiés à l'authentification des joueurs voulant jouer. Le Master redirige les nouveaux clients vers les serveurs d'authentification qui gèrent tous les aspects liés à la sécurité. Une fois authentifié, un client se voit attribuer l'adresse de son serveur de jeu (module World) par le Master.

On retrouve aussi le Messenger qui va former et interpréter les messages à destination ou en provenance des clients, et les modules Network et Encrypt/Decrypt pour le cryptage et la communication réseau. Alors que l'administration du monde du

Le jeu ou l'authentification des clients sont des fonctions distribuées sur plusieurs serveurs, dont c'est la fonction principale, les modules Messenger, Encrypt/Decrypt et Network sont présents sur chaque machine. Dans la section suivante, nous nous intéresserons surtout aux interactions entre les clients et les serveurs de jeu.

IV) Autre architecture utilisée

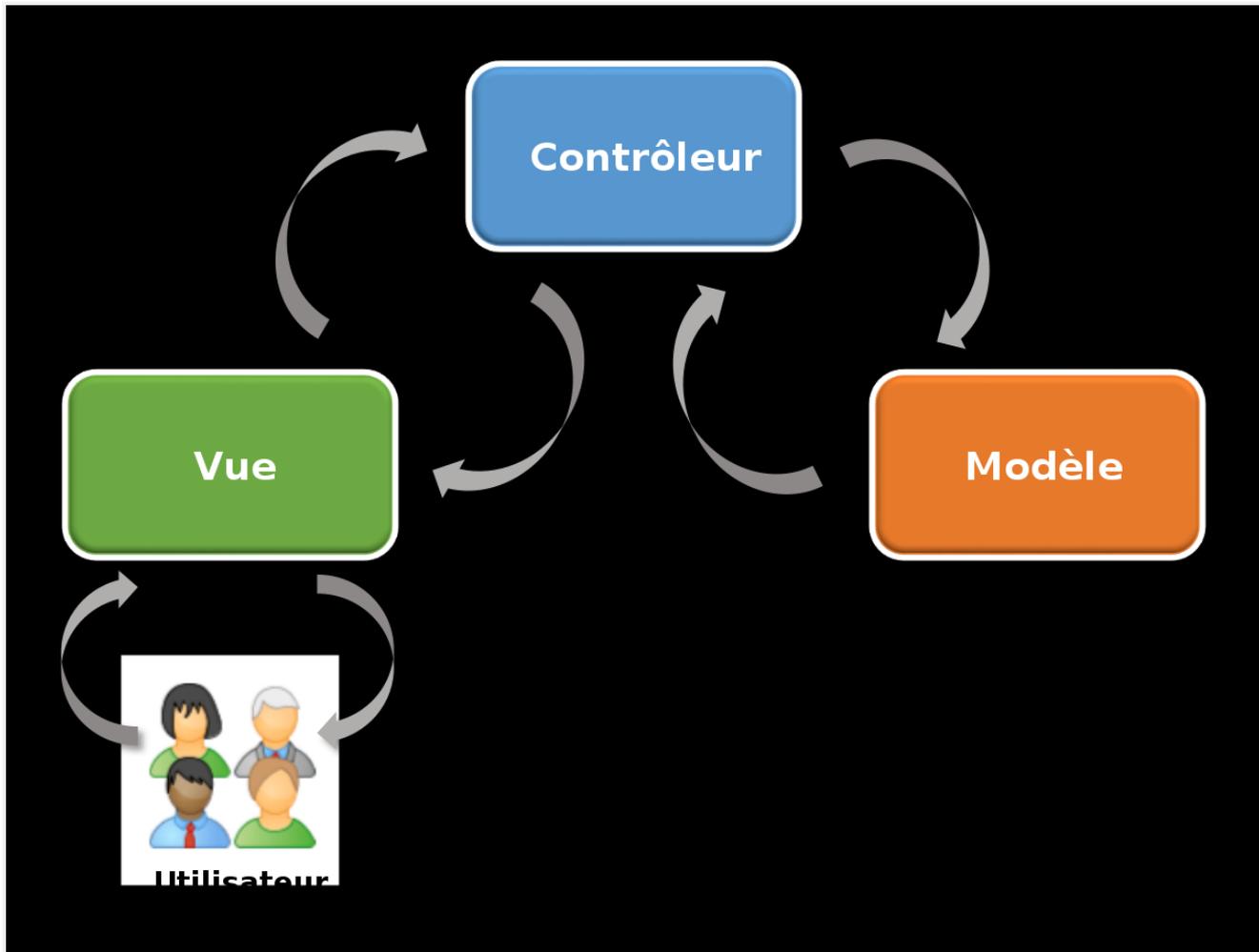
Comme dit précédemment il existe plusieurs types de RPG qui n'ont pas tous la même architecture nous allons alors parler des J-RPG. Le J-RPG (Japanese Role-Playing Game) est un genre de jeux vidéo de rôle originaire du Japon. Les J-RPG sont souvent caractérisés par une forte narration, des personnages mémorables et des univers de fiction riches et détaillés.

Les J-RPG se concentrent souvent sur le développement de personnages et les interactions entre eux, et les choix du joueur peuvent influencer le déroulement de l'histoire. Les combats sont généralement basés sur des mécanismes de tour par tour et impliquent souvent l'utilisation de magie et d'autres compétences spéciales.

Les J-RPG sont souvent associés à des franchises emblématiques telles que Final Fantasy, Dragon Quest, Persona, Tales of, ou encore Chrono Trigger, et ont une base de fans dédiée à travers le monde.

Les J-RPG sont appréciés pour leur histoire et leurs personnages attachants, ainsi que pour leur système de combat stratégique et leur style graphique distinctif. Ils sont souvent considérés comme des jeux "à l'ancienne" en raison de leur focus sur l'histoire et la progression des personnages, ce qui peut différer des styles de jeu plus récents axés sur l'action.

Les J-RPG utilisent généralement une architecture logicielle basée sur le modèle MVC (Modèle-Vue-Contrôleur). Cette architecture divise l'application en trois composants principaux :



1. Le Modèle : Gère les données de l'application, telles que les personnages, les ennemis, les objets, les quêtes, etc. Dans un JRPG, le modèle est souvent représenté par un système de base de données ou de fichiers de sauvegarde.
2. La Vue : Gère la présentation de l'interface utilisateur, telle que les menus, les dialogues, les cartes, les graphismes, les effets spéciaux, etc.
3. Le Contrôleur : Gère la logique de l'application et coordonne les interactions entre le modèle et la vue. Dans un JRPG, le contrôleur est souvent représenté par un système de combat, des quêtes, des événements, etc.

Cette architecture permet une séparation claire des préoccupations et une évolutivité facile du code. Elle permet également à différentes équipes de développeurs de travailler sur différents composants de l'application en même temps, ce qui accélère le développement et améliore la qualité du code.

V) Comparaison avec concurrent

Il est difficile de dire quels sont les concurrents directs des RPG car ils couvrent un énorme éventail de genres et de sous-genres de jeux. Cependant, certains genres de jeux peuvent être considérés comme des concurrents indirects en raison de leur popularité auprès des mêmes types de joueurs.

Nous avons alors décidé de parler d'un type de jeux proche mais assez différent à la fois les MOBA, les MOBA ou Multiplayer Online Battle Arena sont des jeux vidéo en ligne dans lesquels deux équipes de joueurs s'affrontent dans une arène virtuelle. Les joueurs contrôlent des personnages avec des compétences uniques, appelés héros, et travaillent en équipe pour détruire la base ennemie. Les MOBA sont des jeux de stratégie en temps réel qui exigent une communication entre les joueurs pour atteindre la victoire. Les MOBA sont devenus très populaires ces dernières années et sont considérés comme des sports compétitifs. Le jeu MOBA le plus connu est Dota 2

La plupart des MOBA utilise une architecture à 3 couches, cette architecture permet de séparer les différentes fonctionnalités du jeu en couches distinctes, ce qui facilite la maintenance et le développement du jeu:

- 1) La couche de présentation : c'est la couche supérieure de l'architecture à 3 couches et elle est responsable de l'interface utilisateur du jeu. Elle inclut les graphismes, les animations, les menus, les boutons et les contrôles. Cette couche est conçue pour être facilement modifiable, elle permet de personnaliser le jeu en fonction des préférences du joueurs comme modifier les contrôles ou les touches de sort pour les personnages ou encore la qualité des graphismes du jeu.
- 2) La couche logique : c'est la couche intermédiaire de l'architecture à 3 couches et elle est responsable de la logique de jeu du jeu. Elle inclut des éléments tels que la gestion des niveaux, des quêtes, des ennemis, des personnages, des armes, des règles du jeu et des algorithmes. Cette couche est souvent conçue pour être facilement extensible, ce qui permet de développer de nouvelles fonctionnalités et de les intégrer facilement dans le jeu. C'est grâce à cette couches que l'on peut ajouter des nouveaux personnage jouable ou de nouveau niveaux
- 3) La couche de persistance : c'est la couche inférieure de l'architecture à 3 couches et elle est responsable de la gestion des données du jeu. Elle inclut des éléments tels que les bases de données, les fichiers de sauvegarde, les fichiers de configuration et les fichiers de log. Cette

couche est souvent conçue pour être facilement accessible, ce qui permet aux développeurs de récupérer et de stocker des données importantes du jeu.

La principale différence entre les MOBA et les RPG est que les MOBA sont des jeux de stratégie compétitifs en temps réel avec des parties courtes, tandis que les RPG sont des jeux de rôle avec des parties plus longues axées sur la narration, la progression du personnage et l'exploration de l'environnement de jeu.

Ce qui explique la différence d'architecture entre les deux la ou l'architecture client-serveur est utilisée dans de nombreuses applications informatiques pour permettre à un grand nombre d'utilisateurs d'accéder simultanément à une application ou à une base de données centralisée l'architecture à trois couches permet une plus grande flexibilité dans le développement et la maintenance de l'application. Chaque couche peut être modifiée sans affecter les autres, ce qui facilite les mises à jour et les modifications et est facilement évolutive, ce qui permet d'ajouter ou de retirer des couches sans affecter les autres.