War Of Races

Ce projet a pour objectif de simuler une guerre entre deux armées composées de guerriers de races différentes (Humain, Elfe, Orque, Titan) occupant des rôles variés tels que Swordsman, Mage, Supporter, Archer, Berserker et Assassin. Le langage de programmation utilisé est le C++, avec la bibliothèque SDL2 pour créer une interface graphique simple. Les armées sont représentées par des guerriers capables de se déplacer et d'attaquer les ennemis. Ce jeu s'inspire du MMORPG japonais "One Piece Treasure Cruise".

Technologies Utilisées

C++

Ce projet est principalement codé en C++. Pour utiliser cette application, vous devez compiler le code source à l'aide d'un compilateur C++ et exécuter l'application générée.

SDL2

SDL2, ou Simple DirectMedia Layer 2, est une bibliothèque open-source en langage C facilitant le développement d'applications multimédias.

Commet installer SDL2 ?

Sous linux :

```
sudo apt-get update
sudo apt-get install libsdl2-dev libsdl2-image-dev
```

Make (makefile)

Make est un outil d'automatisation de construction qui simplifie la compilation et la liaison des projets logiciels. Il utilise un fichier appelé Makefile pour définir un ensemble de règles et de dépendances pour la construction du projet. Le Makefile contient des instructions sur la manière de compiler les fichiers source et de les lier pour créer des programmes exécutables.

Git

Git est un système de contrôle de version distribué qui aide à gérer et suivre les modifications du code source pendant le développement logiciel. Alors que make et le Makefile sont liés à la construction et à la compilation de logiciels, Git se concentre sur le contrôle de version et la collaboration entre les développeurs.

Gitignore

Le fichier .gitignore est utilisé pour spécifier des fichiers et des répertoires que Git doit ignorer lors du suivi des changements dans un projet

Valgrind

Valgrind est un outil de programmation conçu pour le débogage de la mémoire, la détection des fuites de mémoire et le profilage. Il fonctionne en analysant dynamiquement le comportement de votre programme pendant son exécution, identifiant les problèmes liés à la mémoire et fournissant des informations détaillées à leur sujet. Voici quelques fonctionnalités clés et utilisations de Valgrind :

- **Détection des Fuites de Mémoire :** Valgrind peut détecter les fuites de mémoire dans votre programme, vous aidant à identifier les parties de votre code où la mémoire est allouée mais n'est pas correctement désallouée.
- **Détection des Erreurs de Mémoire :** Il peut trouver diverses erreurs liées à la mémoire, telles que la lecture depuis des emplacements mémoire invalides ou l'écriture dans ces emplacements, ce qui peut entraîner un comportement imprévisible ou des plantages.

Comment installer valgrind

Sous linux

```
sudo apt-get update
sudo apt-get install valgrind
```

Doxygen

Doxygen est un outil de génération de documentation pour le code source. Il prend en charge plusieurs langages de programmation, dont C++, C, Java, Python, et d'autres. Doxygen analyse le code source, extrait la documentation intégrée (commentaires spéciaux dans le code) et génère des documents dans divers formats, tels que HTML, LaTeX, et RTF. Il facilite la création de documentation claire et structurée pour les projets logiciels.

comment Installer Doxygen

Sous Linux

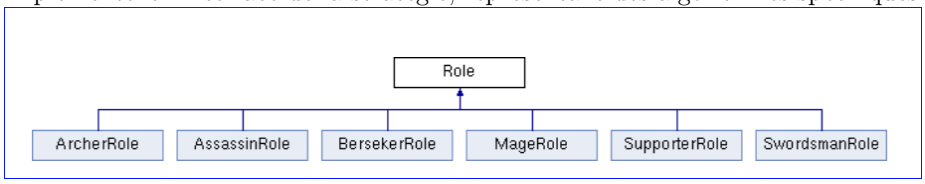
```
sudo apt-get update
sudo apt-get install doxygen
```

Les Patrons de Design

Patron de Stratégie (Strategy Pattern) :

Objectif : Le patron de conception Strategy permet de définir une famille d'algorithmes, de les encapsuler, et de les rendre interchangeables. Ainsi, le client peut dynamiquement choisir l'algorithme à utiliser.

Composants clés : - **Contexte :** Objet utilisant une stratégie, capable de changer de stratégie à tout moment. - **Stratégie :** Interface ou classe abstraite décrivant l'algorithme. - **Implémentations concrètes :** Différentes classes qui implémentent l'interface de la stratégie, représentant des algorithmes spécifiques.



Patron de Décorateur (Decorator Pattern) :

Objectif : Le patron de conception Decorator permet d'attacher dynamiquement des responsabilités supplémentaires à un objet, de manière flexible et sans altérer son code source.

Composants clés : - **Composant :** Interface ou classe abstraite décrivant l'objet de base que le décorateur peut étendre. - **Composant concret :** Implémentation concrète de l'interface. - **Décorateur :** Classe abstraite étendant le composant et contenant une instance du composant concret. Elle peut ajouter des fonctionnalités supplémentaires. - **Décorateurs concrets :** Classes étendant les décorateurs abstraits et ajoutant des fonctionnalités spécifiques.