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

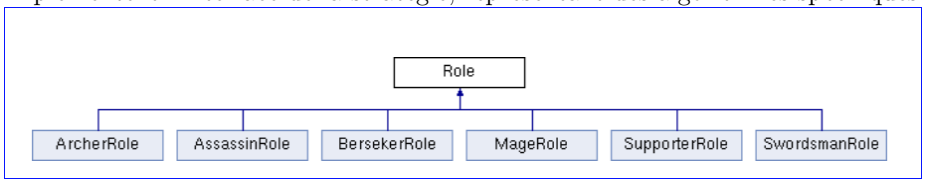
```
sudo apt-get update
sudo apt-get install doxygen
```

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.

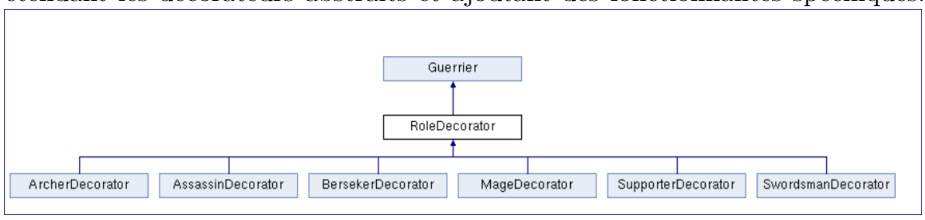
Composants clés : - **Contexte :** Objet utilisant une stratégie, capable de changer de stratégie à tout moment. - **Stratégie :** Interface ou classe abstraite décrivant l'algorithme. - **Implémentations concrètes :** Différentes classes qui implémentent l'interface de la stratégie, représentant des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



Liste des tâches réalisées "Commits" au cours du projet :

Number	Commit ID	Date	Comment
1	cc39a04	2023-12-01 19:34:21	first commit
2	ef5158d	2023-12-01 19:35:47	add a blank makefile

Number	Commit ID	Date	Comment
3	2b60d55	2023-12-01 21:48:32	Testing Basic Makefile
4	5f780c4	2023-12-11 18:21:07	Classe de Base Guerrier
5	2d50cb8	2023-12-11 19:54:18	Classe num 1 : Humain
6	d998826	2023-12-18 20:21:34	CLasses Elfe Orque Titan
7	0534af6	2023-12-18 20:30:17	Add Mersenne Twister
8	03bf226	2023-12-21 17:41:31	Add Fonctions but without Implementation
9	942d9f1	2023-12-30 23:22:01	Implementation & Fix Bugs after 9 Days of Debugging
10	bfc0a48	2024-01-01 18:13:21	Update Guerrier .cpp .hpp Working fine
11	fe42969	2024-01-01 19:32:21	Update Humain .cpp .hpp Working fine
12	08233b5	2024-01-01 20:02:42	Update Elfe .cpp .hpp Working fine
13	722d349	2024-01-01 21:22:13	Update Orque .cpp .hpp Working fine
14	9701341	2024-01-01 21:52:48	Update Titan .cpp .hpp Working fine
15	e96bacf	2024-01-01 22:13:28	Fix some Bugs
16	9d6f499	2024-01-01 22:43:13	Fix some bugs in the makefile
17	2677fbd	2024-01-02 02:12:31	Add Files : Simulation Statistique
18	6c0ed04	2024-01-02 04:31:21	Update Makefile Want To Use SDL2 SDL2 Image
19	ac3de67	2024-01-05 20:01:11	Add Terrain cpp hpp Working with SDL2 Everything is working fine
20	72ed59d	2024-01-05 21:12:56	Ad Gitignore file to ignore .O files
21	d9e15e5	2024-01-05 21:24:05	Add Valgrind and edit makefile to work with it
22	5def4ec	2024-01-06 01:11:24	Add Doxygen and it's working
23	9e79c07	2024-01-06 01:53:36	Add Doxygenn doc for all the files

Number	Commit ID	Date	Comment
24	bd6777b	2024-01-06 01:55:21	Add docygen command to the makefile
25	0236854	2024-01-06 03:21:34	Add Doxygen doc for all functions variables and all but with the help of chatgpt
26	a16b500	2024-01-07 19:54:03	optimiser le code
27	b45ce56	2024-01-07 19:57:44	update the .gitignore
28	5b29b6f	2024-01-08 20:11:58	add mana to Guerrier
29	39841a3	2024-01-10 21:01:14	Add Roles Using Decorator & Strategy Design Patterns
30	05cb411	2024-01-12 23:41:25	Add Doxygen Docs for the Roles Files with the help of chatgpt
31	6c4a209	2024-01-13 20:12:53	Add CDC
32	890ff7a	2024-01-13 20:41:28	Modify CDC et ajouter les diagrames de gantt et pert previsionnel
33	975ddf5	2024-01-13 21:13:43	Add diagrammes de classe generee par doxygen
34	a80a837	2024-01-14 13:42:36	Update Readme.md and add a pdf version of it