Principales Classes et leurs différents rôles

Même si nous sommes capable d'expliquer toute la démarche et les raisons qui nous on amenées à choisir cette structure. Elles ne sont pas rédigées dans ce document qui n'est donc qu'un simple résultat

Attention ce document ne constitue pas une liste exhaustive et exacte des classes, ni des attributs et des méthodes qu'elles contiennent. Il permet seulement de comprendre grossièrement la logique et la structure du Programme ainsi que l'utilité et le rôle de chaque classe.

En outre, le nom des attributs et des méthodes est représentatif de leur(s) rôle(s) mais ne sera pas forcément celui dans le programme. C'est pourquoi le type de retour et les paramètres des méthodes ne sont pas indiqués.

Sauf indications contraires les attributs sont privés et les méthodes publiques.

Table des matières

Les 16 principales classes du programme

Classes de bases du jeu
Construction
AbstractUnit Classe Abstraite
AbstractPlayer Classe abstraite
HumanPlayer Héritée de AbstractPlayer
ArtificialPlayer Héritée de AbstractPlayer

Gestion de l’affichage des textures
Texture
Fenetre

Gestion de la carte et des positions
Terrain
ConstructionDisposition
UnitDisposition
Map

L’interface et les menus
AbstractBouton Classe abstraite
ActionBouton Héritée de AbstractBouton
SousMenuBouton Héritée de AbstractBouton
Menu

Mécanique et fonctionnement du Jeu
MatriceGestionJeu

Classes de bases du jeu

Construction

Représente la forme d'un bâtiment comme par exemple un château

Attributs principaux

- Tableau dynamique à double dimension d'entiers représentant les différentes ‘’pièces’’ d'une construction. Les dimension seront fixées mais propre à chaque construction
- Taille et position de la construction
- Nom / Résistance de la construction etc...
- Pointeur vers le joueur propriétaire
- Pointeur vers un Objet de type Texture

Méthode principale

- Affichage aux bonnes coordonnées en slide-mapping

AbstractUnit Classe Abstraite

Représente une unité

Attributs principaux

- Nom / vie / force etc...
- Taille et position de l'unité
- Pointeur vers le joueur propriétaire
- Pointeur vers un Objet de type Texture

Méthode principale

- Affichage aux bonnes coordonnées

AbstractPlayer Classe Abstraite

Représente un Joueur de la partie. Cette classe a un **comportement Polymorphique.**

Rq1 : *Cette classe ne possède pas d'attributs, c'est normal car un joueur n'a pas l'accès direct aux données du jeu.*
Ainsi, peu importe comment la classe est programmée. Il est certain qu'elle ***ne modifie aucune donnée*** *et se* ***contente des paramètres fournis*** *par la Matrice de Gestion du Jeu pour prendre des décisions.*

Rq2 : *Cette classe ne possède qu'une seule méthode, en effet le rôle d'un joueur et de* ***prendre des décisions***

Méthode principale

- Prendre une décision (**Méthode Virtuelle Pure**)

  En effet tout joueur (Humain ou ''IA") doit savoir prendre une décision quand la Matrice de Gestion du Jeu en à besoin

HumanPlayer Héritée de AbstractPlayer

Représente un Joueur Humain et gère la prise de décisions y compris l'interface (grâce à une ou plusieurs instances de Menu)

Attribut principal

- Menu ou liste de Menu utile(s) à la prise de décision

Méthode principale

- Prendre une décision (**Méthode Virtuelle**)

  *C'est dans cette partie qu'on retrouve l'utilisation des menus mais surtout la **programmation événementielle**. En effet la prise de décisions humaines nécessite des éventements, c'est à dire l'**attente de certaines interactions** du Joueur avec l'interface*

ArtificialPlayer Héritée de AbstractPlayer

Représente un Joueur muni d'une ‘'intelligence artificielle''. Cette classe se contente d'appliquer l'**algorithme de prise de décisions** en fonction des paramètres fournis par la Matrice de Gestion du Jeu

Rq : *Autrement dit les Joueurs gérés par l'ordinateur ont accès au **mêmes données** qu'un joueur Humain*

Attributs principaux

Ces attributs peuvent différer selon l'algorithme choisis dans la méthode de prise de décisions. Il sont la **mémoire de cette ‘'intelligence artificielle''**

Méthode principale

- Prendre une décision (**Méthode Virtuelle**)

Gestion de l'affichage des textures (en partie programmée)

Cette ensemble de classe permet de **simplifier** et de rendre plus cohérente (avec la logique objet du programme) l'utilisation de la SDL.
Rq : *Ces classes sont les seules utilisant directement la SDL*

Texture

Représente une image (une texture) qui peut être **affichée sur l'instance de Fenetre**

Attribut

- Tableau à double dimension d'entiers représentant chaque case du terrain

Méthodes

- Chargement à partir d'un fichier image
- Définir à partir d'une couleur unie
- Définir à partir d'un dégradé de couleur

SpriteTexture

Représente et gère l'affichage des textures en Sprite entre une et deux dimensions

Fenetre

Représente la fenêtre SDL

Rq : *Étant donné que la SDL ne gère **qu'une seule fenêtre**, l'instance de cette classe doit être unique.*

Attributs

- Pointeurs vers la surface de la fenêtre
- Taille de la fenêtre

Méthodes

- Ajouter une texture à la fenêtre
- Actualiser le contenu de la fenêtre

Gestion de la carte et des positions

Cette ensemble de classe permet de faire la le chargement à partir de fichiers, la génération aléatoire et l'affichage de la carte, des constructions et des unités. Elle sert également à stocker toutes les postions du Jeu

Terrain

Représente la géographie physique du terrain

Attribut principal

- SurfaceAffichage qui représente l'intégralité du terrain

Méthodes principales

- Chargement à partir d'un fichier*
- Génération aléatoire
- Enregistrement dans un fichier
- Affichage

ConstructionDisposition

Indique la position de toute les constructions présentes sur la carte

Attributs principaux

- Tableau dynamique à double dimension de pointeurs vers chaque construction
- Pointeur vers le tableau des constructions contenues dans dans la Matrice

Méthodes principales

- Savoir si une construction est présente sur une case
- Obtenir le Pointeur de la construction sur cette case
- Afficher toutes les constructions

UnitDisposition

Indique la position de toute les unités présentes sur la carte

Attributs principaux

- Tableau dynamique à double dimension de pointeurs vers chaque unité
- Pointeur vers le tableau des unités contenues dans dans la Matrice

Méthodes principales

- Savoir si une unité est présente sur une case
- Obtenir le pointeur de l'unité sur cette case
- Afficher toutes les unités

Map

La map représente le terrain et la disposition de l'ensemble des constructions et des unités.

RQ : *Elle est la seule classe qui est appelée directement pour afficher la carte du Jeu*

Attributs principaux

- Un Terrain
- Un objet de type UnitDisposition
- Un objet de type UnitConstruction

Méthodes principales

- Affichage (actualisation graphique de l'ensemble Terrain, constructions et unités)
- Actualisation graphique des constructions + unités
- Actualisation graphique des unités

L’interface et les menus

Cette partie concerne l’interface permettant aux Joueurs Humains de prendre des décisions et d’interagir avec le programme. Elles permettent de rendre très simple la création, la modification, l’utilisation et l’affichage des menus

AbstractBouton Classe abstraite

Cette classe représente simplement un bouton, *aucune instance ne peut-être crée*

Attributs principaux

- Taille et position du bouton
- Texte affiché sur le menu
- Couleur du fond
- Couleur du texte

ActionBouton Héritée de AbstractBouton

Représente un bouton qui sera associé à une action c’est à dire qui aura un impact direct sur le jeu (via une décision)

Attributs principaux

- Id de l’action qui sera un nom ou un nombre
- + Attributs de AbstractBouton

SousMenuBouton Héritée de AbstractBouton

Représente un bouton qui déclenchera l'**affichage d'un nouveau menu**

Attributs principaux

- Pointeur vers un autre menu différent du Parent
- + Attributs de AbstractBouton

Menu

Permet la création, l'affichage et la **gestion des différents menus** permettant d'interagir avec les joueurs humains

Attributs principaux

- Liste d'AbstractBouton
- Taille et position du menu
- Couleur du fond
- Pointeur vers la liste de tous les menus (**attribut statique**)

Méthodes principales

- Calcul des positions des Boutons que le Menu contient
- Affichage du menu
- Associer la position d'un clic à un menu et surtout à un bouton en retournant l'identifiant de celui-ci (**méthode statique**)

Mécanique et fonctionnement du Jeu

MatriceGestionJeu

Cette classe **gère et coordonne** l'ensemble des mécanismes du jeu **en fonction des décisions des joueurs**. *Instance de l'objet censé être unique*

Rq : *La matrice ne fait qu'interpréter mais ne gère pas le système qui permet à chaque joueur de prendre une décision. Ainsi le traitement effectué par la matrice est le même pour un Joueur et une ''Intelligence artificielle"*

Attributs principaux

- Map
- Tableau dynamique de AbstractPlayer
- Tableau dynamique deAbstractConstruction
- Tableau dynamique de AbstractUnit

Méthodes principales publiques

- Initialisation de la Matrice de Gestion du Jeu
- GameLoop qui est la boucle du jeu

Exemple de méthodes privées

Permettent grandement de simplifier et de découper le code de la fonction GameLoop

- Initialisation du tour d'un joueur
- Déplacement d'une unité
- Défaite d'un joueur

…