**ESIL**

08

**Rapport PSR**

**WarThread**

**Florent Mas,Sylvain Paillé,Tony Quesnel,Aurélien Fiol,Michaël Lanöe, Jean-Marc Muzi**

Sommaire

[1 Introduction 3](#_Toc198998135)

[2 Présentation du jeu 3](#_Toc198998136)

[2.1 But 3](#_Toc198998137)

[2.2 Règles 3](#_Toc198998138)

[3 Entités du jeu 3](#_Toc198998139)

[3.1 Dieu 3](#_Toc198998140)

[3.2 Fidele 3](#_Toc198998141)

[3.3 Warrior 3](#_Toc198998142)

[3.4 Diagramme UML 3](#_Toc198998143)

[4 Création du réseau et déploiement 5](#_Toc198998144)

[4.1 Création du graphe 5](#_Toc198998145)

[4.2 Déploiement du graphe 5](#_Toc198998146)

[4.3 Visualisation du jeu 5](#_Toc198998147)

[5 Déroulement du Jeu 5](#_Toc198998148)

[5.1 Migration des warriors 5](#_Toc198998149)

[5.2 IA des warriors 5](#_Toc198998150)

[5.2.1 Récolte des informations 5](#_Toc198998151)

[5.2.2 Détermination d’une stratégie 5](#_Toc198998152)

[5.3 Gestion des batailles 6](#_Toc198998153)

[6 Gestion des pannes 6](#_Toc198998154)

[6.1 Simulation d’une panne 6](#_Toc198998155)

[6.2 Détection des pannes 6](#_Toc198998156)

[6.3 Elections d’un nœud 6](#_Toc198998157)

[6.4 Reconstruction d’un nœud 6](#_Toc198998158)

# Introduction

# Présentation du jeu

## But

## Règles

# Entités du jeu

## Dieu

## Fidele

## Warrior

## Diagramme UML

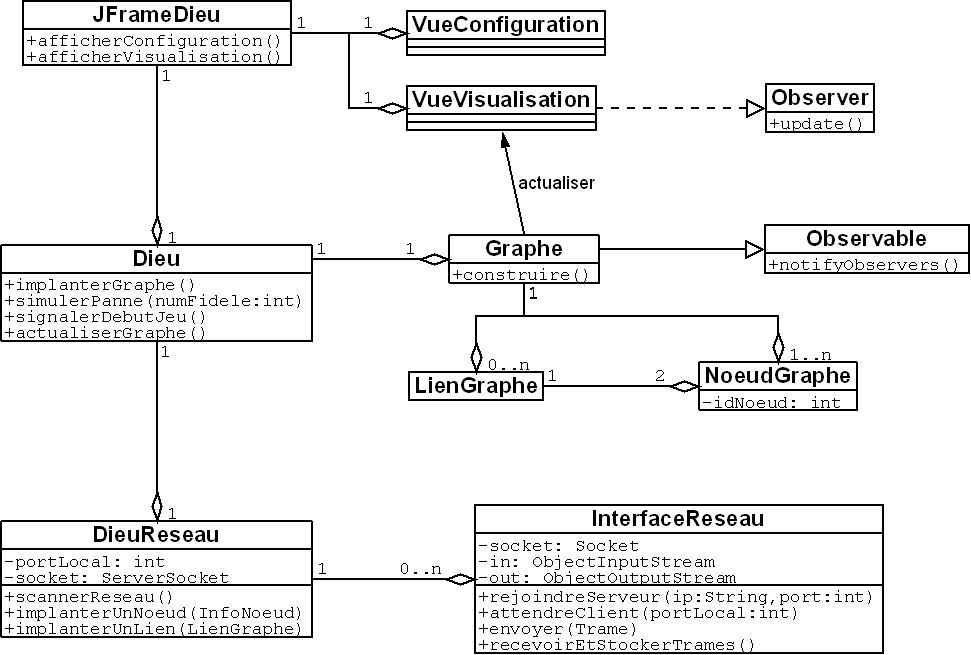


Figure : UML du programme DIEU

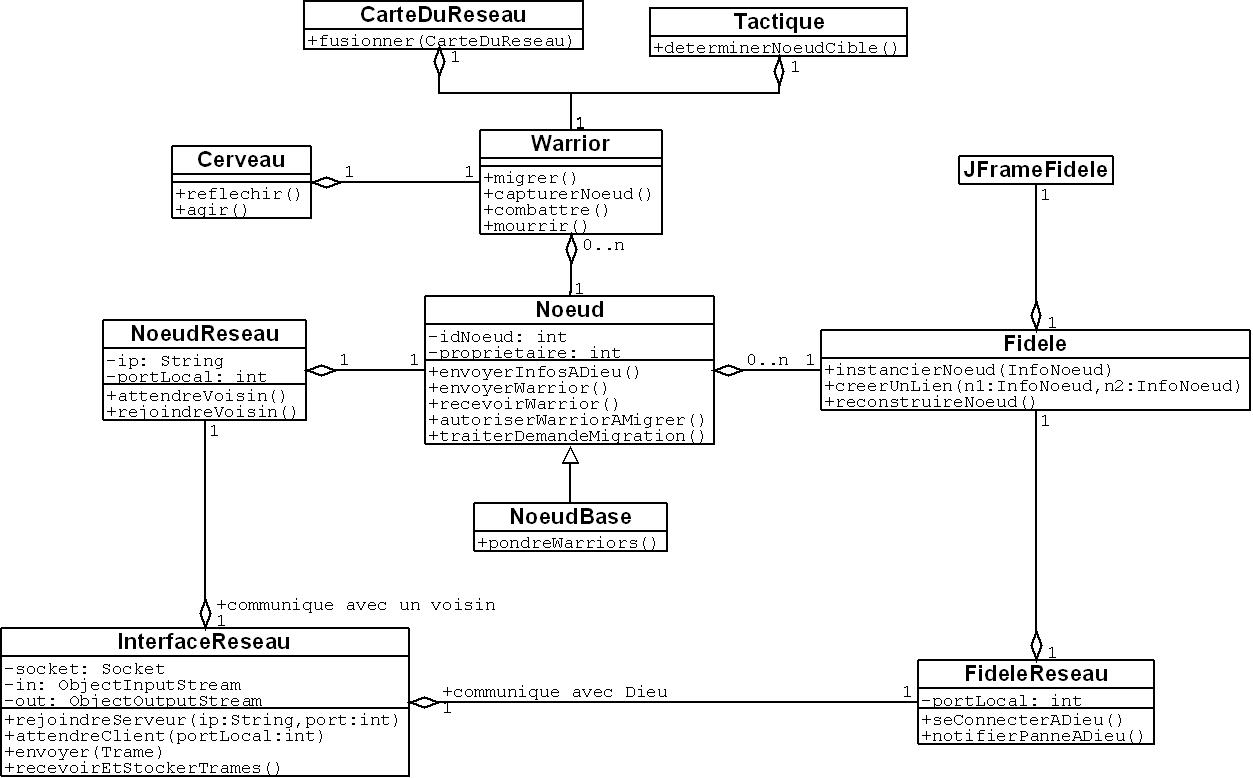


Figure : UML du programme Fidèle

# Création du réseau et déploiement

## Création du graphe

## Déploiement du graphe

## Visualisation du jeu

# Déroulement du Jeu

## Migration des warriors

## IA des warriors

Le déroulement du jeu nécessite un comportement intelligent des warriors (agents mobiles).Nous avons donc implémenté une IA respectant le schéma classique de réflexion d’une entité intelligente :

Cette IA va donc récolter des informations sur la carte présente et agir en conséquence.

### Récolte des informations

### Détermination d’une stratégie

Après la récolte des différentes informations sur la carte. L’IA va choisir un type de stratégie. Il en existe 4 :

* Stratégie indéterminée
* Stratégie Exploratrice
* Stratégie Offensive
* Stratégie Défensive

Le choix va se faire sur différent facteurs : le risque des nœuds aux alentours, le coefficient d’agressivité de l’IA du joueur et le nombre de nœud connus par le warrior.

1. La stratégie Indéterminée : elle permet au warrior d’évoluer sur le réseau de manière aléatoire au début du jeu ou dans le cas ou aucune autre stratégie ne serait applicable.
2. La stratégie Exploratrice : Cette stratégie choisit toujours d’explorer les nœuds inconnus : c’est-à-dire que parmi les voisins du nœud sur lequel se trouve le warrior, elle va choisir un nœud qui n’est pas présent dans la carte du warrior.
3. La stratégie offensive : Elle permet au warrior d’attaquer des nœuds qui possèdent un coefficient « risque » plus ou moins élevé : c'est-à-dire un nœud qui a été capturé par l’ennemi ou accueillant un ou plusieurs warrior ennemis.  
   Ce choix se fait parmi toute la carte en possession du warrior .La stratégie calcule donc **le chemin le plus court** pour accéder au nœud grâce à **l’algorithme de Dijkstra.**Si plusieurs warriors sont nécessaires pour la capture du nœud, le thread attendra un certains laps de temps que d’autres warriors de son camp viennent l’aider.
4. La stratégie offensive : Cette stratégie se met en place lorsque le risque de la majorité des nœuds connus est beaucoup trop élevé. Le warrior va alors défendre sa base puis les nœuds aux alentours.  
   De la même manière que la stratégie offensive les chemins sont calculés avec dijkstra.

Chaque stratégie définit donc une ou plusieurs actions qui sont :

* Migrer directement vers le nœud cible
* Migrer vers le nœud cible en passant par un chemin donné
* Attaquer le nœud
* Attendre l’arrivé de renfort

## Gestion des batailles

# Gestion des pannes

## Simulation d’une panne

## Détection des pannes

## Elections d’un nœud

## Reconstruction d’un nœud