



Analyse et Programmation Orientées Objets / C++

Simulateur TRAFALGAR
Gestion des radars

Fonctions du simulateur

F1 : installer le champ de bataille

F2 : **calculer les déplacements des navires**

F3 : appliquer les consignes de mouvement

F4 : appliquer les consignes de tir

F5 : appliquer les consignes de débarquement

F6 : **diffuser l'état courant du champ de bataille**

F7 : déterminer et appliquer les conditions météo

Etat courant du champ de bataille

1. **Positions et vitesses réelles des navires**
2. **Etat courant des dégâts matériels**
3. **Effectifs des troupes à bord**
4. **Situation des troupes à terre**
5. **Description de tous les échos radar**
6. **Etats courants de la météo**

Classe Trafalgar (feuille n°23)

Interface unique entre le simulateur et le reste de la suite logicielle

Modélise un ensemble de navires qui se déplacent dans un ensemble d'îles.

Attributs de la classe Trafalgar

➡ Version 1.0.0

```
static map <string, Ile*> m_pIles;
```

```
static map <string, Navire*> m_pNavires;
```

```
static bool m_status;
```

Services de la classe Trafalgar

➔Version 1.0.0

static void **observer** ();

static map <string, Ile>& **diffuser** ();

static map<string, Navire>& **diffuser**
(const
string&);

Méthodes de la classe Trafalgar

➡ **Version 1.1.0**

static void observer (const string&);

Attributs de la classe Navire

➡ **Version 1.5.0**

list <Radar*> m_pRadars;

Services de la classe Navire

➔ **Version 1.5.0**

void observer ();

Attributs de la classe Radar

➔ Version 1.1.0

```
int          m_portee;  
int          m_precision;  
etatRadar    m_etatRadar;  
list <echo* > m_pEchos;
```

Services de la classe Radar

➡ Version 1.1.0

void **observer** (const string&, int);

Méthodes de la classe Radar (1)

➡ **Version 1.1.0**

```
list <echo* >* observer (const Point&,
                        list <Triangle>&,
                        list <Point>&);
```

Méthode de classe privée

Méthodes de la classe Radar (2)

➔ **Version 1.1.0**

**echo* calculerEchoRadar(const Point&,
const Point&);**

Méthode d'instance privée

Méthodes de la classe Radar (3)

➡ **Version 1.1.0**

bool cibleCachee (const Point&
 const Point&
 list <Triangle>&);

Méthode de classe privée

Méthodes de la classe Radar (4)

➡ Version 1.1.0

Point* cibleMasquee (const Point&
const Point&
list <Point>&);

Méthode de classe privée