



# GÉNIE LOGICIEL

VERSION

# CLASSIQUE

NICOLAS MARIE  
LOIC LALANNE  
PIERRE-ANTOINE LARGUET  
ROMAIN GRANIE  
ROMAN BADANIN

# SOMMAIRE

- \* INTRODUCTION
- \* ANALYSE DU SYSTÈME
- \* MODELE VDM++ CLASSIQUE
- \* IMPORTANCE DE CE MODÈLE POUR LE DÉVELOPPEUR
- \* TRANSLATION DU MODÈLE VDM++ (C++ / JAVA)

# I - INTRODUCTION

UN INTERCEPTEUR DE MISSILE (BOUCLIER ANTIMISSILE) EST UN SYSTÈME D'AUTO-DÉFENSE PAR CONTRE MESURE. SON ACTIVITÉ EST L'INTERCEPTION DIRECTE PAR COLLISION DE TYPE HTK (HIT-TO-KILL)

ELLE INTÈGRE DEUX GROUPES DE FONCTIONS:

- LA DÉTECTION, LA LOCALISATION ET L'IDENTIFICATION D'UNE MENACE:

LA PRÉSENCE D'UN MISSILE QUI PÉNÈTRE DANS UNE DEMI-SPHÈRE DE 25KM CENTRÉE SUR LE BOUCLIER, EST DÉTECTÉE,

SA POSITION CALCULÉE ET SON TYPE IDENTIFIÉ

- DE CONTRÔLE DE TIRS D'ANTIMISSILES:

LE SYSTÈME ORDONNE LE TIR D'UN ANTIMISSILE POUR CONTRER LE TYPE DE MISSILE ET DONC NEUTRALISER LA MENACE.

DANS CE PROJET, NOUS NOUS INTÉRESSONS À LA GESTION DE L'INTERCEPTEUR DE MISSILE, LA GESTION EST UNE TACHE TRÈS COMPLIQUÉ A FAIRE PAR UN HUMAIN SANS LOGICIEL, C'EST POUR CELA QU'IL LUI FAUT UN LOGICIEL SÛR QUI FONCTIONNE CORRECTEMENT, CAR LA MOINDRE DÉFAILLANCE PEUT PROVOQUÉ DE GRAVE CONSÉQUENCE.

ON CONSIDÈRE QUE L'INTERCEPTEUR ÉTUDIÉE EST CONFIGURÉE EN 3 PHASES, DÉCRITE PLUS LOIN.

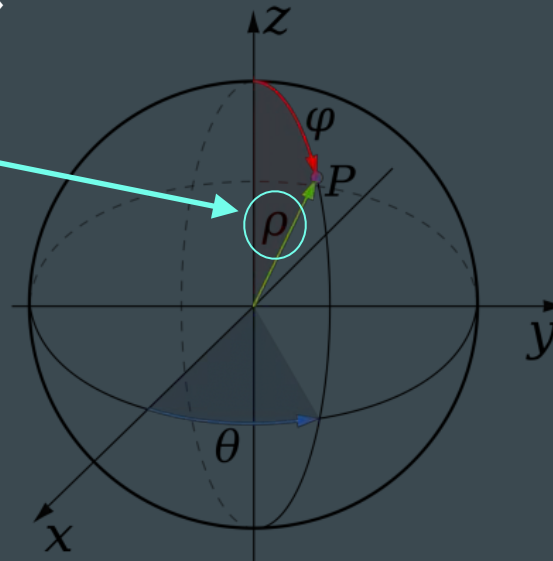
**UN DÉTECTEUR DE MISSILE:** ACTIVE LES CAPTEURS

IL ANALYSE LES INFORMATIONS DÉLIVRÉES PAR SATELLITE ET RADAR ET DÉTECTE LA PRÉSENCE D'UNE MENACE

ON CONSIDÈRE QUE LA POSITION DE LA MENACE EST DETERMINER PAR 3 CAPTEURS:

## CAPTEUR: DISTANCE

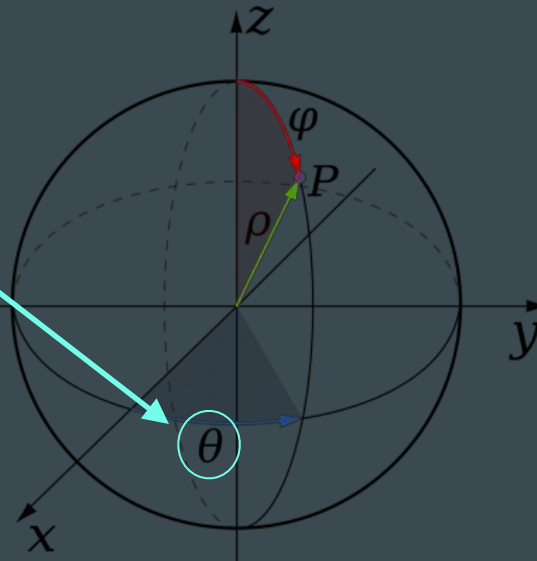
LA DISTANCE ON LA DÉFINIT PAR LE PETIT «P»



IL DONNE UNE ESTIMATION DE LA DISTANCE «P» À LAQUELLE SE TROUVE À UN INSTANT DONNÉ LE MISSILE À INTERCEPTER

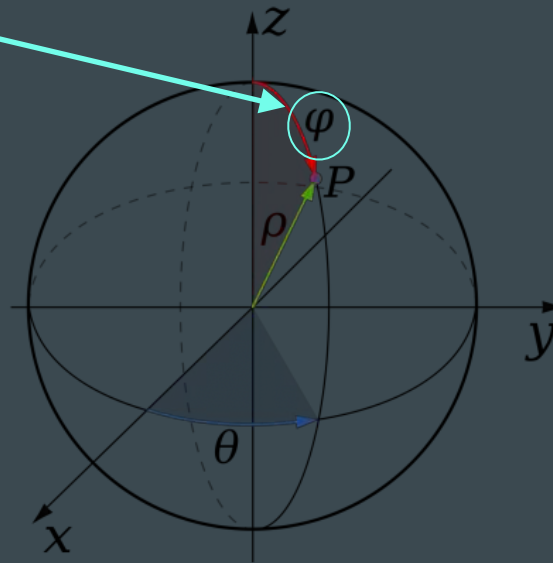
## CAPTEUR: LONGITUDE

LA LONGITUDE, NOTÉE  $\theta$  QUI DÉSIGNE L'ANGLE MESURÉ DEPUIS L'AXE DES ABSCISSES  $x$  ET COMPRIS ENTRE  $-180^\circ$  ET  $180^\circ$



## CAPTEUR: LATITUDE

LA LATITUDE, NOTÉE  $\phi$  , QUI DÉSIGNE, L'ANGLE DEPUIS LE PLAN ÉQUATORIAL (ENTRE  $-90^\circ$  ET  $90^\circ$ )



# CONTRÔLEUR

CES 3 CAPTEURS TRANSMETTENT LEURS DONNÉES AU SYSTÈME D'IDENTIFICATION, QUI EXPLOITE UNE BASE DE DONNÉES DE MENACES POTENTIELLES ET DÉTERMINE QUEL EST LE TYPE DE MISSILE À CONTRER.

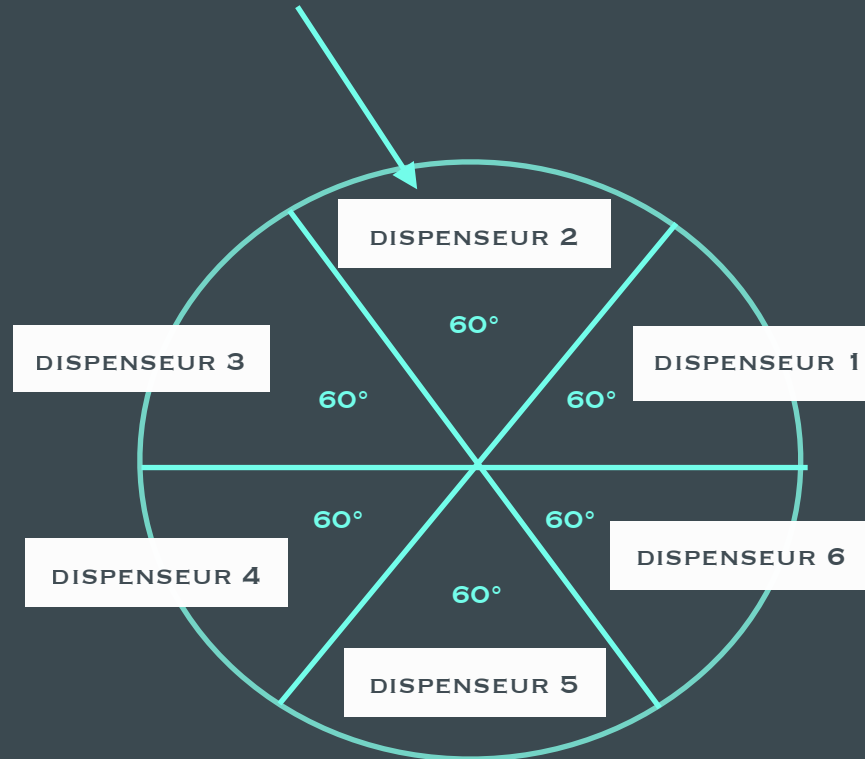
LE **CONTRÔLEUR** EST LANCÉ POUR DÉTERMINE EN DÉROULANT UN ALGORITHME LES ORDRES AUX DISPENSEURS QUI DÉCLENCHENT DES TIRS



UN **CONTRÔLEUR** DE TIR DOIT ÊTRE CAPABLE D'ACTIONNER LES DISPENSEURS DANS UN DÉLAI DE **250ms** APRÈS RÉCEPTION DE L'INFORMATION COMPLÈTE DES 3 CAPTEURS.

## LES DISPENSEURS

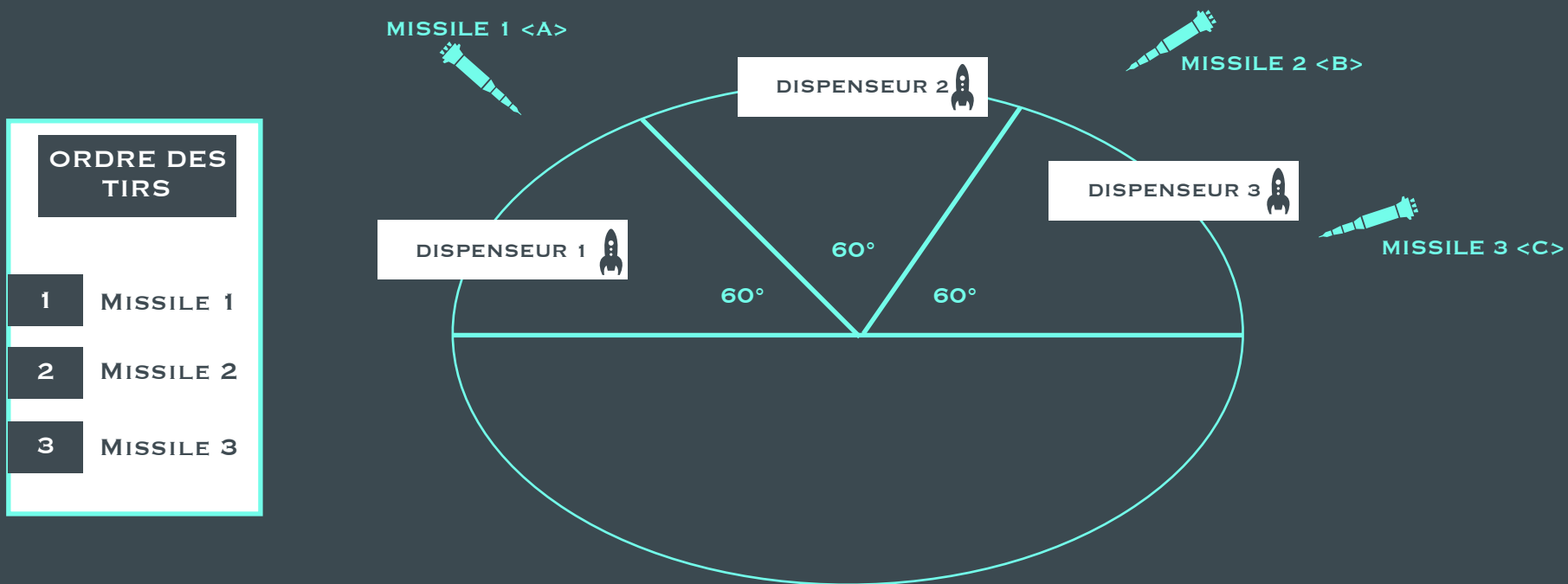
DISPOSÉ DE FAÇON CIRCULAIRE DE SORTE QUE CHAQUE **DISPENSEUR** COUVRE UN CHAMPS DE TIR DE DE  $60^\circ$





## TRAITEMENT DES MENACES

LE **MISSILE 1** EST ARRIVÉ EN PREMIER IL EST DE TYPE <A>, IL SERA DONC TRAITÉ EN PREMIERE PAR LE CONTRÔLEUR, PUIS LES 2 AUTRES **MISSILES** SERONT TRAITÉES PAR ORDRE DE PRIORITÉ PAR LE CONTRÔLEUR.



## II - ANALYSE DU SYSTEME

### SYSTEME **RTP**:

- REPRÉSENTE LES CLASSES DE CONFIGURATION DU SYSTÈME
- CREE DES INSTANCES "PUBLIC STATIC" POUR PERMETTRE L'ACCÈS A PARTIR DE TOUTES LES CLASSES DU MODELE.

### SYSTEME **HORLOGE**:

- CLASSE POUR MODÉLISER L'HORLOGE: BASE DE TEMPS
- OPERATIONS:
  - STEPTIME()**; MET A JOUR LA DATE ACTUELLE
  - GETTIME()**; RETOURNE LA DATE ACTUELLE

### SYSTEME **ENVIRONNEMENT**:

- CLASSE DÉCRIT L'ENVIRONNEMENT DANS L'INSTANT MEME (MODÉLISATION DE L'ENVIRONNEMENT)
- OPERATIONS:
  - READDIS()**; RETOURNE LA DISTANCE
  - READLAT()**; RETOURNE LA LATITUDE
  - READLON()**; RETOURNE LA LONGITUDE
  - SETDIS()**; MET A JOUR LA DISTANCE
  - SETLAT()**; MET A JOUR LA LATITUDE
  - SETLON()**; MET A JOUR LA LONGITUDE

#### SYSTEME DETECTEUR:

- CLASSE MODÉLISE UN DÉTECTEUR DE MENACE
- OPERATIONS:
  - DETECTEUR(ID, TYPE); INITIALISE L'ID ET LE TYPE DES CAPTEURS.
  - ACTION(); L'ACTION DU DÉTECTEUR SI UNE MENACE EST DÉTECTÉ.

#### SYSTEME CONTROLEURDISPENSEUR:

- CLASSE MODÉLISE UN CONTROLEURDISPENSEUR
- OPERATIONS:
  - CONTROLEURDISPENSEUR(); INITIALISE L'ID ET LE TYPE.
  - LANCEMENT(); ACTIVE LA DESTRUCTION DE LA MENACE.
  - ACTION(); ACTIONNE L'ALGORITHME, ACTIVE LANCEMENT() ET SUPPRIME LA CIBLE DES MENACES.

#### SYSTEME CONTROLEUR:

- CLASSE MODÉLISE LE CONTRÔLEUR, PILOTE QUI RÉALISE L'ASSERVISSEMENT
- OPERATIONS:
  - ADDMENACE(MENACE); AJOUTE LA MENACE DANS LA LISTE DES MENACES.
  - RMENACE(); SUPPRIME LA MENACE DE LA LISTE DES MENACES.
  - MISEAJOUR(); MET A JOUR LES DONNÉES DE DISTANCE/LONGITUDE/LATITUDE DE LA MENACE.
  - ALGORITHME(); DETERMINANT LA DESTRUCTION D'UN MISSILE
  - ACTION(); RÉCUPÈRE LES DONNÉES DES CAPTEUR, MET A JOUR LES DONNÉES ET DÉROULE L'ALGORITHME.

SYSTEME **CAPTEURLONGITUDE:**

- CLASSE MODÉLISE LE CAPTEUR DE LONGITUDE DE LA MENACE
- OPERATIONS:
  - CAPTEURLONGITUDE(ID, TYPE, VAL);** INITIALISATION DE L'ID, TYPE ET VALEUR.
  - ACTION();** MESURER LA LONGITUDE DES MISSILES DE L'ENVIRONNEMENT.

SYSTEME **CAPTEURLATITUDE:**

- CLASSE MODÉLISE LE CAPTEUR DE LATITUDE DE LA MENACE
- OPERATIONS:
  - CAPTEURLATITUDE(ID, TYPE, VAL);** INITIALISATION DE L'ID, TYPE ET VALEUR.
  - ACTION();** MESURER LA LATITUDE DES MISSILES DE L'ENVIRONNEMENT.

SYSTEME **CAPTEURDISTANCE:**

- CLASSE MODÉLISE LE CAPTEUR DE DISTANCE DE LA MENACE
- OPERATIONS:
  - CAPTEURDISTANCE(ID, TYPE, VAL);** INITIALISATION DE L'ID, TYPE ET VALEUR.
  - ACTION();** MESURER LA DISTANCE DES MISSILES DE L'ENVIRONNEMENT.

#### SYSTEME CAPTEUR:

- CLASSE MODÉLISE LES CAPTEURS

- OPERATIONS:

  - GETID(); DONNE L'IDENTITÉ D'UN CAPTEUR.

  - GETTYPE(); DONNE LE TYPE D'UN CAPTEUR.

  - READVALEUR(); RETOURNE LA VALEUR MESURÉE PAR LE CAPTEUR.

  - ACTION(); RENVOYÉE A LA RESPONSABILITÉ DE LA SOUS-CLASSE EN FONCTION DU CAPTEUR UTILISE.

#### SYSTEME ANIMATION:

- CLASSE QUI FAIT INTERAGIR LES DIFFÉRENTS COMPOSANTS DU SYSTÈME

- OPERATIONS:

  - ANIMER(); ACTIVER ENVIRONNEMENT POUR CRÉER UN STIMULI.

### III - MODELE VMD++ CLASSIQUE

LA PARTIE CODE EST SITUÉ DANS LE FICHIER «ONE»

### IV - IMPORTANCE DE CE MODELE POUR LES DÉVELOPPEURS

L'INTÉRÊT DE LA MÉTHODE VDMSL EST SA COMPLIANCE AVEC LA MÉTHODE AGILE. NOTAMMENT CETTE MÉTHODE PERMET AU DÉVELOPPEUR DE METTRE EN PLACE UN MODÈLE QUI PERMET, AU CLIENT COMME AUX DÉVELOPPEURS, DE TESTER LE PROGRAMME ET DE S'ASSURER QUI RÉPOND PARFAITEMENT AU BESOIN DU CLIENT SANS AVOIR À PASSER PAR L'ÉTAPE DÉVELOPPEMENT. LONGUE ET COÛTEUSE, SURTOUT SI LE PROGRAMME NE RÉPOND PAS CORRECTEMENT À LA DEMANDE.

### V - TRANSLATION DU MODELE EN C++/JAVA

NOUS AVONS GÉNÉRER LE CODE SOURCE DU LOGICIEL D'APPONTAGE AUTOMATIQUE, EN C++ ET JAVA. CES DEUX CODES SE SITUENT DANS LE DOSSIER «ONE/GENERATED» POUR LA PARTIE C++ ET «ONE/GENERATED/QUOTES» POUR LA PARTIE JAVA DE L'ARCHIVE DU PROJET.

FIN, **MERCI** POUR VOTRE LECTURE

**PROJET DE:** LOIC LALANNE  
ROMAN BADANIN  
ROMAIN GRANIE  
NICOLAS MARIE  
PIERRE-ANTOINE LARGUET

**M1 TI 2021/2022 GENIE LOciel**



FIN