



# Scripting missions templates

Jeu. 16/07/2022



# Progression

- 1) Les liens
- 2) Les grands principes
- 3) Les pre-requis
- 4) Les features
- 5) Cas d'utilisation





# Les liens.

Les sources des missions et des scripts :

<https://github.com/JTFF-mission-scripts/jtff-templates>

Les missions compilées :

<https://drive.google.com/drive/u/0/folders/1oJxXmdGU4sKZ2ZFznLCIFxhHph4YB6Q7>

Les librairies publiques fortement couplées avec les templates :

- MOOSE : [https://github.com/FlightControl-Master/MOOSE\\_INCLUDE/blob/master/Moose Include Static/Moose.lua](https://github.com/FlightControl-Master/MOOSE_INCLUDE/blob/master/Moose Include Static/Moose.lua)
- MIST : <https://github.com/mrSkortch/MissionScriptingTools/blob/master/mist.lua>
- SKYNET : <https://github.com/walder/Skynet-IADS/blob/master/demo-missions/skynet-iads-compiled.lua>



# Les grands principes

Les scripts présents dans les templates ont pour objectif de fournir un **environnement standardisé, riche et facile à mettre en œuvre** dans vos missions.

Il s'agit donc de vous faciliter la vie pour ce qui est des « à côtés » de vos missions (peut être aussi un peu sur le cœur de vos missions mais ce n'est pas le but principal de ces templates).

Les scripts se présentent donc sous la forme de plusieurs scripts implémentant des fonctions spécifiques (la gestion du trafic civil, la gestion des ATIS de vos aérodromes, la gestion des tankers de vos missions, la gestion de l'airboss de votre CSG, etc...). A vous d'intégrer dans vos missions les fonctionnalités qui vous semblent intéressantes à implémenter.





# Les grands principes

Chaque fonctionnalité est découpée en 2 parties :

- Le script en lui-même : c'est un fichier lua qui implémente le comportement de la feature. C'est codé une bonne fois pour toutes, vous n'avez pas besoin de le modifier. Mais si vous êtes un cadord en lua, vous pouvez contribuer au code et me filer votre identifiant github, je vous ajoute en tant que développeur avec joie comme Dan l'a déjà fait (et c'est le moment où j'en profite pour un big-up à Dan qui a abattu un super boulot sur ces scripts déjà depuis plusieurs mois)
- Le fichier de settings qui va avec : celui-ci est aussi un fichier lua mais qui ne contient que le paramétrage nécessaire au fichier script. C'est celui que vous allez modifier pour l'adapter à votre mission.



# Pre-requis : Les softs utiles.

## Scripts templates:

- Nodejs : <https://nodejs.org/dist/v16.14.0/node-v16.14.0-x64.msi>

## Gestion de version git:

- Git bash : <https://github.com/git-for-windows/git/releases/download/v2.35.1.windows.2/Git-2.35.1.2-64-bit.exe>

## Editeur de code (IDE):

- Notepad++ : <https://notepad-plus-plus.org/downloads/v8.3.1/>
- SublimeText : <https://download.sublimetext.com/Sublime%20Text%20Build%203211%20x64%20Setup.exe>
- Eclipse : [http://eclipse.org/downloads/download.php?file=/ldt/products/stable/1.4.2/org.eclipse.ldt.product-win32.win32.x86\\_64.zip](http://eclipse.org/downloads/download.php?file=/ldt/products/stable/1.4.2/org.eclipse.ldt.product-win32.win32.x86_64.zip)
- IntelliJ IDEA : <https://www.jetbrains.com/fr-fr/idea/download/download-thanks.html?platform=windows&code=IIC>

## Visualisation des logs :

- Glogg : [http://glogg.bonnefon.org/files/glogg-latest-x86\\_64-setup.exe](http://glogg.bonnefon.org/files/glogg-latest-x86_64-setup.exe)





# Les features

- 0xx ) Les bases
  - 010-root menus.lua
  - 020-mission functions.lua
- 110-set clients.lua
- 120-tankers.lua
- 13x ) Nava Ops:
  - 130-airboss.lua
  - 135-pedro.lua
- 140-beacons.lua
- 150-awacs.lua
- 160-atis.lua
- 17x ) A2A
  - 170-cap zone training.lua
  - 172-cap zone war.lua
  - 173-fox zone-training.lua
  - 176-random air traffic.lua
- 19x ) A2G
  - 190-ranges.lua
  - 193-training ranges.lua
  - 196-fac ranges.lua
  - 199-skynet.lua



# Les features

## 0xx : Les bases

- **010-root\_menus.lua :**

C'est le script qui va ajouter le menu de base dans le menu de com radio F10

Cela va créer dans chaque coalition un menu radio intitulé : « My coalition resources ».

Les autres scripts vont populer ce menu en fonction des scripts que vous embarquerez dans votre mission,

Pas de fichier settings pour celui-là

- **020-mission\_functions.lua :**

C'est la librairie JTFF ce script ne fait rien mais contient l'ensemble des fonctions générales qui pourront être appelées par les scripts que vous embarquerez dans votre mission.

C'est un peu l'équivalent de MOOSE ou de MIST...

Pas de fichier settings pour celui-là





# Les features

## 110-set\_clients.lua

Ce script va gérer les évènements de chaque joueur humain :

Quand un joueur entre dans un avion, il va populer le menu F10 avec:

- les informations des tankers présents sur la map
- Une option pour rendre les avions du groupe du joueur immortel ou non

Pas de fichier settings pour celui-là



# Les features

## 120-tankers.lua



Ce script va gérer les tankers sur la mission :

En fonction des paramètres renseignés dans le fichier settings associé, ce script va programmer le décollage, le pattern et le retour des tankers que vous voulez placer dans votre mission.

Ce script permet d'implémenter 2 logiques distinctes pour les tankers :

- Des tankers « mission » qui auront le comportement défini dans votre fichier settings et s'y conformeront quoi qu'il arrive
- Des tankers « à la demande » qui auront un comportement défini par défaut dans votre fichier settings mais qui pourront à tout moment être re-routés à un autre endroit et avec d'autres paramètres en utilisant la vue F10 et les marqueur pendant la mission.

Ce script prend en entrée 2 variables qui seront définies dans votre fichier settings :

- TankersConfig : c'est les paramètres de vos tankers « mission »
- OnDemandTankersConfig : c'est les paramètres par défaut de vos tankers « à la demande »



# Les features

## 120-tankers.lua



```
TankersConfig = {  
  {  
    enable = true,  
    autorespawn = true,  
    patternUnit = 'Escorte CSG-1 CVN-71-1',  
    benefit_coalition = coalition.side.BLUE,  
    baseUnit = 'CSG-1 CVN-2-1',  
    terminalType = AIRBASE.TerminalType.OpenMedOrBig,  
    groupName = 'nanny-1 #IFF:4775FR',  
    airboss_recovery = true,  
    --escortgroupname = 'jolly_hornet #IFF:7323FR',  
    missionmaxduration = 105,  
    altitude = 8000,  
    speed = 320,  
    tacan = {  
      channel = 104,  
      morse = 'SHL',  
    },  
    freq = 264.250,  
    fuelwarninglevel = 35,  
    racetrack = {  
      front = 40,  
      back = -10,  
    },  
    modex = 102,  
    callsign = {  
      alias = 'Shell',  
      name = CALLSIGN.Tanker.Shell,  
      number = 1,  
    },  
  },  
  { ... },  
}
```

Dans cet exemple :

On a un tanker mission :

- Qui sera actif
- Qui redécollera en boucle
- Qui va tourner autour de l'unité 'Escorte CSG-1 CVN-71-1'
- Qui va décoller depuis le porte-avion : 'CSG-1 CVN-2-1'
- Qui va être pris sur le modèle du groupe : 'nanny-1 #IFF:4775FR'
- Qui va être géré par l'Airboss
- Qui volera au maximum 105 minutes
- À une altitude de 8000 pieds
- A une vitesse de 320 nœuds
- Sur un TACAN 104Y 'SHL'
- Écouter sur une fréquence de 264,250 MHz
- Sera RTB quand il ne lui restera plus que 35% de fuel
- Aura un racetrack de 40nm devant l'unité de pattern 'Escorte CSG-1 CVN-71-1'
- Et 10nm devant son unité de pattern : 'Escorte CSG-1 CVN-71-1'
- Portera un modex : 102
- Et un callsign : Shell-1-1

# Les features

## 120-tankers.lua



```
OnDemandTankersConfig = {
  --Texaco 3 KC135MPRS
  {
    enable = true,
    type = "tx3",
    benefit_coalition = coalition.side.BLUE,
    baseUnit = AIRBASE.Syria.Incirlik,
    terminalType = AIRBASE.TerminalType.OpenBig,
    groupName = 'TEXACO 3-1 #IFF:4275FR',
    missionmaxduration = 135,
    altitude = 26000,
    speed = 420,
    tacan = {
      channel = 106,
      morse = 'TX3',
    },
    freq = 317.5,
    modex = 23,
    callsign = {
      alias = 'Texaco-3',
      name = CALLSIGN.Tanker.Texaco,
      number = 3
    }
  },
  --Arco 7 KC135
  { ... },
  --Shell 4 KC135
  { ... },
  --Texaco 1 MPRS
  { ... },
  --ARCO 1 KC135
  { ... },
  --ARCO 2 KC135
  { ... }
}
```

Dans cet exemple :

On a un tanker à la demande :

- Qui sera actif
- Qui sera identifié sur les marqueurs en vue F10 comme tx3
- Qui va décoller depuis l'aerodrome d'Incirlik
- Qui va etre pris sur le modele du groupe : **TEXACO 3-1 #IFF:4275FR**
- Qui restera en position 135 minutes
- À une altitude de 26000 pieds
- A une vitesse de 420 nœuds
- Sur un TACAN 106Y 'TX3'
- Ecouterà sur une frequence de 317,5 MHz
- Portera un modex : 23
- Et un callsign : Texaco-3-1

C'est un tanker à la demande. Il ne décollera donc pas au début de la mission mais seulement quand un joueur le déclenchera avec un marqueur en vue F10



# Les features

## 130-airboss.lua

Ce script va gérer les Porte aéronefs sur la mission :

En fonction des paramètres renseignés dans le fichier settings associé, ce script va programmer les opérations aéronavales, les fréquences radio, les TACANs et ICLS des porte-aéronefs que vous voulez placer dans votre mission.

Ce script permet d'implémenter 2 logiques distinctes d'opération aéronavales pour les PA :

- Les cyclicOps.
- L'alpha-strike.

Ce script prend en entrée 1 variable qui sera définie dans votre fichier settings :

- AirBossConfig : c'est les paramètres de vos Porte Aeronefs



# Les features 130-airboss.lua



```
AirBossConfig = {
  {
    enable = true,
    carriername = 'CSG-1 CVN-71-1',
    alias = 'roosevelt',
    coalition = 'blue',
    enable_menumarkzones = true,
    enable_menusmokezones = true,
    enable_niceguy = true,
    handleAI = true,
    recoverytanker = 'nanny-1 #IFF:4775FR',
    recoveryops = {
      mode = 'cyclic',
      cyclic = {
        event_duration_minutes = 60,
        event_ia_reserved_minutes = 15,
      }
    },
    tacan = {
      channel = 71,
      mode = 'X',
      morse = 'RSV',
    },
    icls = {
      channel = 11,
      morse = 'RSVLSO',
    },
    freq = {
      base = 127.3,
      marshall = 127.5,
      lso = 127.4
    },
    infintepatrol = true,
    controlarea = 65,
    recoverycase = 1,
    maxpatterns = 5,
    maxstacks = 8,
    difficulty = AIRBOSS.Difficulty.NORMAL,
    menurecovery = {
      enable = true,
      duration = 30,
      windondeck = 30,
      offset = 0,
      uturn = true
    },
    releayunit = {
      marshall = 'MarshallRelay',
      lso = 'LSORelay',
    },
    singlecarrier = false,
    operationsstatspath = "C:/airboss-stats",
    operationstrapsheetpath = "C:/airboss-trapsheets"
  },
}
```

Dans cet exemple on a un PA avec ces paramètres :

- Qui est activé
- Qui est basé sur l'unité du mission editor : **CSG-1 CVN-71-1**
- Qui fait partie de la coalition blue
- Qui va activer les menu radios pour placer des marqueurs sur la vue F10
- Qui va activer les menus radios pour placer des smokes
- Qui va activer son mode « niceguy »
- Qui va gérer les IA dans son marshall comme les joueurs
- Qui va proposer un recovery tanker basé sur l'unité du mission editor : **nanny-1 #IFF:4775FR**
- Qui va opérer en mode cyclic-ops :  
Avec des events qui durent 60 minutes (2/3 recovery puis 1/3 departure) **mode = 'cyclic'**  
**cyclic = {**  
  **event\_duration\_minutes = 60,**  
  **event\_ia\_reserved\_minutes = 15,**  
**}**
- Qui va activer un TACAN 71X 'RSV'
- Un ICLS 11 'RSVLSO'
- Qui va opérer les fréquences :
  - 127,3 MHz pour l'approche
  - 127,5 MHz pour Marshall
  - 127,4 MHz pour la tour/LSO
- Qui va contrôler les aéronefs dans le 65 nm autour de lui
- Qui va opérer en case I
- Aurotiser 5 patterns avant de renvoyer au marshall
- Gérer jusqu'à 8 stacks dans le marshall
- Qui va noter les patterns en niveau Difficile
- Qui va populer un menu Airboss dans les menu radios F10
- Qui va simuler les émissions radio sur SRS avec les unités du mission editor:
  - MarshallRelay
  - LSORelay
- Qui va stocker les stats de recovery et trapsheets dans :
  - C:\airboss-stats
  - C:\airboss-trapsheets

Pensez à créer ces 2 statiques sur le PA pour permettre les communications sur SRS

```
releayunit = {
  marshall = 'MarshallRelay',
  lso = 'LSORelay',
}
```

Pensez à créer ces 2 répertoires sur le serveur qui hébergera la mission



# Les features

## 135-pedro.lua

Ce script va gérer les Helicos de sauvetage autour des porte aeronefs sur la mission :

En fonction des paramètres renseignés dans le fichier settings associé, ce script va programmer le comportement général des hélico de sauvetage que vous voulez placer dans votre mission.

Ce script prend en entrée 1 variable qui sera définie dans votre fichier settings :

- PedrosConfig : c'est les paramètres de vos Pedros



# Les features

## 135-pedro.lua



```
PedrosConfig = {  
  {  
    enable = true,  
    autorespawn = true,  
    patternUnit = 'CSG-1 CVN-71-1',  
    baseUnit = 'Escorte CSG-1 CVN-71-2',  
    groupName = 'Pedro-1 #IFF:7325FR',  
    freq = 127.6,  
    modex = 704  
  },  
  { ... }  
}
```

Dans cet exemple un Pedro avec ces paramètres :

Qui sera activé

Qui respanwera à chaque fois qu'il se pose ou qu'il meure

Qui suivra l'unité du mission editor : CSG-1 CVN-71-1

Qui décollera de l'unité du mission editor : Escorte CSG-1 CVN-71-2

Qui sera reproduit sur le groupe du mission editor : Pedro-1 #IFF:7325FR

Qui opérera sur la fréquence : 127,6MHz

Qui portera le modex : 704



# Les features

## 140-beacons.lua

Ce script va gérer les TACANs que vous pourriez vouloir placer sur votre mission :

En fonction des paramètres renseignés dans le fichier settings associé, ce script va activer un TACAN sur l'unité de votre choix.

Ce script prend en entrée 1 variable qui sera définie dans votre fichier settings :

- BeaconsConfig : c'est les paramètres de vos balises TACAN



# Les features

## 140-beacons.lua



Dans cet exemple on a une balise avec ces paramètres :

- Qui sera activée
- Qui sera nommée « My beacon »
- Qui émettra sur l'unité du mission editor :
- Qui émettra sur 101X 'MBC'

```
BeaconsConfig = {  
  enable = true,  
  name = 'My beacon',  
  unitName = 'MyBeaconUnit-1',  
  tacan = {  
    channel = 101,  
    band = 'X',  
    morse = 'MBC'  
  }  
}
```



# Les features

## 150-awacs.lua

Ce script va gérer les awacs sur la mission :

En fonction des paramètres renseignés dans le fichier settings associé, ce script va programmer le décollage, le pattern et le retour des awacs que vous voulez placer dans votre mission.

Ce script permet d'implémenter 2 logiques distinctes pour les tankers :

- Des tankers « mission » qui auront le comportement défini dans votre fichier settings et s'y conformeront quoi qu'il arrive
- Des tankers « à la demande » qui auront un comportement défini par défaut dans votre fichier settings mais qui pourront à tout moment être re-routés à un autre endroit et avec d'autres paramètres en utilisant la vue F10 et les marqueur pendant la mission.

Ce script prend en entrée 2 variables qui seront définies dans votre fichier settings :

- TankersConfig : c'est les paramètres de vos tankers « mission »
- OnDemandTankersConfig : c'est les paramètres par défaut de vos tankers « à la demande »



# Les features

## 150-awacs.lua



```
AwacsConfig = {  
  {  
    enable = true,  
    autorespawn = true,  
    patternUnit = 'Escorte CSG-1 CVN-71-1',  
    benefit_coalition = coalition.side.BLUE,  
    baseUnit = 'CSG-1 CVN-2-1',  
    terminalType = AIRBASE.TerminalType.OpenMedOrBig,  
    groupName = 'WIZARD #IFF:4773FR',  
    airboss_recovery = true,  
    escortgroupname = 'jolly_hornet #IFF:7323FR',  
    missionmaxduration = 120,  
    altitude = 28000,  
    speed = 380,  
    freq = 377.800,  
    fuelwarninglevel=45,  
    racetrack = {  
      front = 80,  
      back = -20  
    },  
    tacan = {  
      channel = 100,  
      morse = 'WZD',  
    },  
    modex = 705,  
    callsign = {  
      alias = 'Wizard',  
      name = CALLSIGN.AWACS.Wizard,  
      number = 1  
    }  
  },  
  { ... }  
}
```

Dans cet exemple :

On a un awacs :

- Qui sera actif
- Qui redécollera en boucle
- Qui va tourner autour de l'unité Escorte CSG-1 CVN-71-1
- Qui va décoller depuis le porte-avion : CSG-1 CVN-2-1
- Qui va être pris sur le modele du groupe : WIZARD #IFF:4773FR
- Qui va être géré par l'Airboss
- Qui sera escorté par le groupe du mission editor : jolly\_hornet #IFF:7323FR
- Qui volera au maximum 120 minutes
- À une altitude de 28000 pieds
- A une vitesse de 380 nœuds
- Sur un TACAN 100Y 'WZD'
- Ecouterà sur une frequence de 377,8 MHz
- Sera RTB quand il ne lui restera plus que 45% de fuel
- Aura un racetrack de 80nm devant l'unité de pattern
- Et 20nm devant son unité de pattern :
- Portera un modex : 705
- Et un callsign : Wizard-1-1

```
racetrack = {  
  front = 80,  
  back = -20  
},
```



# Les features

## 160-atis.lua

Ce script va gérer les ATIS sur les terrains importants de la mission :

En fonction des paramètres renseignés dans le fichier settings associé, ce script va programmer les informations données dans l'ATIS sur chaque terrain ou un ATIS sera présent.

Ce script prend en entrée 1 variable qui sera définie dans votre fichier settings :

- **AtisConfig** : c'est les paramètres de vos ATIS

**!!** : il est possible aussi d'utiliser le module DATIS, dans ce cas pas besoin de scripts, vous éditez juste un objet dans le mission editor comme décrit dans la doc de DATIS : <https://github.com/rkusa/DATIS>



# Les features

## 160-atis.lua



```
atisConfig = {  
  {  
    enable = true,  
    airfield = AIRBASE.Syria.Incirlik,  
    radio = {  
      freq = 270.1,  
      power = 100,  
      modulation = radio.modulation.AM,  
      relayunit = 'ATISRelay-1-1',  
      tower = {  
        360.1,  
        360.2,  
        360.3,  
        122.1}  
      },  
    },  
    active = {  
      number = '05',  
      side = 'L'  
    },  
    },  
    tacan = {  
      channel = 21  
    },  
    },  
    ils = {  
      freq = 109.3,  
      runway = '05'  
    },  
    },  
    srs = {  
      path = "C:\\SRS"  
    },  
    },  
  }  
}
```

Dans cet exemple :

On a un atis :

- Qui sera actif
- Qui sera situé sur l'aerodrome de Incirlik

TO BE COMPLETED



# Les features

## 170-cap\_zone\_training.lua

Ce script va gérer

TO BE COMPLETED



# Les features

## 170-cap\_zone\_training.lua



```
TrainingCAPConfig = {  
  {  
    enable = true,  
    coalitionCAP = coalition.side.RED,  
    name = 'North_Turkey',  
    --patrolZoneGroupName = 'CAP zone Red',  
    --engageZoneGroupName = 'engage zone Red',  
    patrolZoneName = 'red_north_patrol_zone',  
    engageZoneName = 'red_north_engage_zone',  
    CAPGroups = {  
      'REDNavyCAP-1',  
      'REDNavyCAP-2',  
      'REDNavyCAP-3',  
      'REDNavyCAP-4',  
      'REDNavyCAP-5',  
      'REDNavyCAP-6',  
      'REDNavyCAP-7',  
      'REDNavyCAP-8'  
    },  
    skill = 'Excellent',  
    capParameters = {  
      patrolFloor = 20000,  
      patrolCeiling = 40000,  
      minPatrolSpeed = 250,  
      maxPatrolSpeed = 400,  
      minEngageSpeed = 250,  
      maxEngageSpeed = 900,  
      engageFloor = 5000,  
      engageCeiling = 55000  
    }  
  }  
}
```

Dans cet exemple :

On a une zone d'entrainement A2A :

- Qui sera actif
- Qui sera dans la coalition RED

TO BE COMPLETED



# Les features

## 172-cap\_zone\_war.lua

Ce script va gérer un système complète de défense A2A à l'intérieur d'une zone prédéfinie de la mission :

Le système sera constitué :

- D'une zone à défendre.
- D'un senssemble EWR et AWACS assurant la détection des menaces et coordonnant la défense.
- D'un ensemble d'aérodromes ou porte-aéronefs constituant les bases des différents escadrons.
- D'un ensemble d'un ou plusieurs escadrons sur chacune de ces bases avec chacun une mission assignée de type :
  - CAP dans une sous-zone ou dans la zone complète.
  - Interception lorsqu'une menace est détectée à l'intérieur de la zone à défendre.



# Les features

## 172-cap\_zone\_war.lua

```
WarCAPConfig = {
  {
    enable = true,
    coalitionCAP = coalition.side.RED,
    name = 'Russian Defense',
    debug = true,
    borderZoneGroupName = 'border-russia',
    --borderZoneName = 'red_north_patrol_zone',
    detectionGroupingRadius = 15,
    engageRadius = 100,
    gciRadius = 300,
    ewrPrefixes = {
      "EW_RUS",
      "108"
    },
  },
  CAPBases = {
    {
      baseName = "kusnetsov",
      patrolSquadrons = {
        {
          enable = true,
          groupName = "CAP-RU-3",
          groupForce = 4,
          groupNumber = 8,
          patrolInAirNumber = 2,
          patrolZoneName = "Patrol_Crimea"
        }
      },
      interceptSquadrons = {
        {
          enable = true,
          groupName = "INT-RU-3",
          groupForce = 2,
          groupNumber = 2
        }
      }
    }
  },
  skill = 'Excellent',
  capParameters = {
```

Dans cet exemple :

On a une zone de guerre A2A :

- Qui sera active
- Qui sera dans la coalition RED
- Constituée d'une base : le Kusnetsov
  - Qui hébergera 1 escadron de patrouilleurs patrouillant par groupe de 4 et constitué de 8 groupes qui ne seront jamais plus de 2 groupes en l'air en même temps, et qui patrouilleront dans la zone 'Patrol\_Crimea'
  - Qui hébergera 1 escadron d'intercepteurs décollant par groupe de 2 et constitué de 2 groupes
- Guidés par un système EWR et AWACS constitué par tous les groupes du mission editor commençant par **EW\_RUD** et 108
- Avec une consigne d'engagement de 100 nm pour les patrouilleurs
- Avec une consigne d'engagement de 300 nm pour les intercepteurs
- Le système complet sera contenu à l'intérieur de la zone **border-russia**





# Les features

## 173-fox\_zone\_training.lua

Ce script va gérer

TO BE COMPLETED



# Les features

## 173-fox\_zone\_training.lua



```
FoxRangesConfig= {  
  {  
    name = 'North-West',  
    enable = true,  
    safeZoneName = 'red_north_engage_zone',  
    launchZoneName = 'red_north_engage_zone',  
    --safeZoneGroupName = 'engage zone Red',  
    --launchZoneGroupName = 'engage zone Red',  
    debug = false,  
    f10Menu = false,  
    missileLaunchMessages = false,  
    missileDestruction = true  
  },  
  { ... }  
}
```

Dans cet exemple :

On a une zone d'entraînement A2A :

- Qui sera active

TO BE COMPLETED



# Les features

## 176-random\_air\_traffic.lua

Ce script va gérer

TO BE COMPLETED



**Dans cet exemple :**  
**On a un trafic aléatoire :**

- Qui sera actif

TO BE COMPLETED



**Dans cet exemple :**  
**On a un trafic aléatoire :**

- **Qui sera actif**

TO BE COMPLETED

# Les features 190-ranges.lua

Ce script va gérer

TO BE COMPLETED



# Les features

## 190-ranges.lua

```
RangeConfig = {  
  { ... },  
  {  
    name = "Paphos",  
    enable = true,  
    benefit_coalition = coalition.side.BLUE,  
    subRange = {  
      { ... },  
      { ... },  
      { ... },  
      { ... },  
      { ... }  
    }  
  },  
  { ... }  
}
```

Dans cet exemple :  
On a un range A2G :  
• Qui sera actif

TO BE COMPLETED





# Les features

## 190-ranges.lua

Dans cet exemple :  
On a un

```
RangeConfig = {  
  { ... },  
  {  
    name = "Paphos",  
    enable = true,  
    benefit_coalition = coalition.side.BLUE,  
    subRange = {  
      { ... },  
      { ... },  
      {  
        name = "Parking Supply",  
        groupsToSpawn = {"Infantry Paphos", "SCUD Paphos", "Truck Paphos"},  
        AI = false,  
        holdFire = true  
      },  
      { ... },  
      { ... }  
    },  
  },  
  { ... }  
}
```

TO BE COMPLETED



# Les features

## 193-training\_ranges.lua

Ce script va gérer

TO BE COMPLETED



# Les features

## 193-training\_ranges.lua



```
TrainingRangeConfig = {  
  {  
    name = "Akamas",  
    enable = true,  
    targets = {  
      { ... },  
      { ... },  
      { ... },  
      { ... },  
      { ... }  
    }  
  },  
  { ... }  
}
```

Dans cet exemple :  
On a un range A2G :  
• Qui sera actif

TO BE COMPLETED



# Les features

## 193-training\_ranges.lua



```
TrainingRangeConfig = {  
  {  
    name = "Akamas",  
    enable = true,  
    targets = {  
      {  
        type = "Strafepit",  
        unit_name = "Strafe Akamas",  
        foul_line = "Foul Line",  
        boxlength = 2000,  
        boxwidth = 200,  
        heading = nil,  
        inverseheading = true,  
        goodpass = 20  
      },  
      { ... },  
      { ... },  
      {  
        type = "BombCircle",  
        unit_name = "Cercle A Akamas",  
        precision = 25  
      },  
      { ... }  
    }  
  },  
  { ... }  
}
```

Dans cet exemple :  
On a un

TO BE COMPLETED

# Les features

## 196-fac\_ranges.lua

Ce script va gérer

TO BE COMPLETED



# Les features

## 196-fac\_ranges.lua



```
FACRangeConfig = {  
  {  
    name = "Range 62",  
    enable = true,  
    benefit_coalition = coalition.side.BLUE,  
    subRange = {  
      { ... },  
      { ... }  
    }  
  },  
  { ... },  
  { ... },  
  { ... },  
  { ... },  
  { ... },  
  { ... }  
}
```

Dans cet exemple :  
On a un range A2G :

- Qui sera actif

TO BE COMPLETED



## TO BE COMPLETED

```
FACRangeConfig = {  
    {  
        name = "Range 62",  
        enable = true,  
        benefit_coalition = coalition.side.BLUE,  
        subRange = {  
            {  
                name = "FAC 62-A",  
                groupsToSpawn = {"FAC C3"},  
            },  
            { ... }  
        }  
    },  
    { ... },  
    { ... },  
    { ... },  
    { ... },  
    { ... },  
    { ... },  
}
```

# Les features 199-skynet.lua

Ce script va gérer

TO BE COMPLETED



# Les features

## 199-skynet.lua

Dans cet exemple :  
On a un range A2G :

- Qui sera actif

```
SkynetConfig = {  
  { ... },  
  { ... },  
  {  
    name = "Bassel El Assad IADS",  
    enable = true,  
    benefit_coalition = coalition.side.BLUE,  
    headQuarter = {"IADS_QG_Bassel"},  
    nodes = {  
      { ... }  
    }  
  }  
}
```

TO BE COMPLETED





# Les features

## 199-skynet.lua

Dans cet exemple :  
On a un range A2G :

- Qui sera actif

```
SkynetConfig = {  
  { ... },  
  { ... },  
  {  
    name = "Bassel El Assad IADS",  
    enable = true,  
    benefit_coalition = coalition.side.BLUE,  
    headQuarter = {"IADS_QG_Bassel"},  
    nodes = {  
      {  
        connection = "IADS_QG_Bassel",  
        ewrs = {"EW_NorthBassel", "EW_SouthBassel"},  
        sites = { ... },  
      }  
    }  
  }  
}
```

TO BE COMPLETED



# Les features

## 199-skynet.lua



Dans cet exemple :  
On a un

TO BE COMPLETED

```
SkynetConfig = {  
  { ... },  
  { ... },  
  {  
    name = "Bassel El Assad IADS",  
    enable = true,  
    benefit_coalition = coalition.side.BLUE,  
    headQuarter = {"IADS_QG_Bassel"},  
    nodes = {  
      {  
        connection = "IADS_QG_Bassel",  
        ewrs = {"EW_NorthBassel", "EW_SouthBassel"},  
        sites = {  
          {  
            sam = "SAM-SA-17M1-Bassel",  
            harmDetectionChance = 65,  
            goLiveRangePercent = 60,  
            pdharmDetectionChance = 90,  
            actAsEW = false,  
            pointDefenses = "SAM-SA-15-PD-Bassel"  
          },  
          {  
            sam = "SAM-SA-19-Bassel",  
            goLiveRangePercent = 70,  
            "SAM-ShortRange-Bassel",  
          },  
        },  
      },  
    },  
  },  
}
```

# Des questions ?

- Grouillez-vous on a les mécanos qui nous attendent...
- Et Tracto, laisse la dame tranquille, regarde tu l'a fait tomber...





# Fin du briefing

- **Allez cassez-vous, et n'oubliez pas de faire vos checklist pré-embarquement : la sécurité avant tout...**

