



Document réalisé avec OpenOffice Janvier. 2007 PAV.

# Wolfenstein Enemy Territory



## I) Introduction au script initial de la map.

Dans Enemy Territory , le principe général est le suivant. Il y a les aires de jeu appelées «maps» dans lesquelles les joueurs évoluent et il y a le script propre à la map que l'on trouve dans le dossier map du fichier **map.pk3** ou **pack0.pk3** pour les maps originales E.T (oasis.script, fueldump.script...). Ce fichier script est lu au lancement de la map, il contient tous les événements (triggers) qui seront exécutés lorsqu'un joueur agit sur une «entité». Un poste de commandement est une entité et lorsqu'un ingénieur le construit alors la partie du script liée à sa construction est exécutée, (afficher l'image du command post construit, informer les joueurs que le command post est construit, etc...) Les objectifs primaires et secondaires sont des triggers particuliers nommés dans le script, **wm\_objective\_status** et identifiés par un numéro unique.

Les autres triggers ont un nom, un état et la valeur de cet état ou parfois seulement un nom et un état. Par exemple dans fueldump en ce qui concerne le footbridge nous avons ceci :

<b>footbridge build final</b>	lorsque le pont de bois est construit. (nom, état, valeur)
<b>footbridge death</b>	lorsqu'il est détruit. (nom, état)

Pour connaître le n° exacte d'un objectif primaire et ou secondaire, il faut ouvrir le fichier script de la map. Nous pouvons aussi connaître le n° de l'objectif en le visualisant dans le **limbo** mais dans ce cas là les objectifs sont incrémentés de un et en fait l'objectif n° 2 dans le limbo correspond à l'objectif n° 1 du script, le 3 du limbo est le 2 du script, etc... Ceci s'explique par le fait que dans le limbo le n° 1 est une présentation général de la mission et non pas un objectif. Voici le début du script de **oasis.script** (figure 1). Ici, les objectifs sont numérotés de 1 à 8, (primaires 1,2,3), (secondaires 4 ,5,6,7,8). Par convention, la team est aussi identifiée par un numéro, **0 pour Axis, 1 pour Alliés** (2 pour spectateur à titre d'information car non géré dans les scripts). Puis vient l'état de l'objectif **0 = neutre, 1 = atteint, 2 = non atteint**. Ce qui donne une syntaxe de ce type 1 1 1, 1 0 1, 4 0 2, etc... Vous noterez qu'au début du jeu tous les objectifs sont neutres pour les deux équipes (tous les **wm\_objective\_status** se terminent par 0). Quand les Alliés s'empareront du **forward spawn point** alors ce trigger sera déclenché et prendra comme valeur 4 1 1. 4 étant le n° de l'objectif, 1 team alliés , 1 objectif atteint. Quand les Axis construiront leur **command post** alors le trigger correspondant prendra comme valeur 8 0 1. Ce sont ces informations (entre autre), qu'utilise bobot, c'est pourquoi une map et ou un script mal réalisés ne donneront pas de bons résultats quant aux comportements des bots.

**Figure 1 : oasis.script**

```
game_manager
{
    // Objectives
    // 1: Primary1 : Destroy the North gun
    // 2: Primary2 : Destroy the South gun
    // 3: Primary3 : Breach Old City wall
    // 4: Secondary1 : Capture forward spawn point
    // 5: Secondary2 : Drain/flood cave system by repairing/damaging the Oasis pump
    // 6: Secondary3 : Drain/flood cave system by repairing/damaging the Old City pump
    // 7: Allied command post
    // 8: Axis command post
```

```
// Objective overview status indicators
// wm_objective_status <objective> <team (0=Axis, 1=Allies)>
// <status (0=neutral 1=complete 2=failed)>

wm_objective_status 1 1 0
wm_objective_status 1 0 0
wm_objective_status 2 1 0
wm_objective_status 2 0 0
wm_objective_status 3 1 0
wm_objective_status 3 0 0
wm_objective_status 4 1 0
wm_objective_status 4 0 0
wm_objective_status 5 1 0
wm_objective_status 5 0 0
wm_objective_status 6 1 0
wm_objective_status 6 0 0
wm_objective_status 7 1 0
wm_objective_status 7 0 0
wm_objective_status 8 1 0
wm_objective_status 8 0 0
```

Quand un `wm_objective_status` ou un autre trigger est déclenché, E.T passe l'information à bobot qui regarde alors dans son `map.nodscript` si cet événement a été défini et si c'est le cas bobot exécutera ses propres instructions liées au trigger.

## II) Bobot Nodscript

Dans le `nodscript` bobot il n'est pas besoin de dire «les bots doivent faire ça ou aller là ...» tout ceci est pris en charge par le code bobot. Il suffit simplement en général de changer le type d'un node et les bots sauront quoi faire. Le script se révèle extrêmement simple. Il y a peu d'instructions et cela suffit largement pour obtenir un comportement des bot satisfaisant

Ces instructions qui sont au nombre de quatre sont les suivantes:

**Set, wait, destroy\_connec, create\_connec** et ses dérivées

**set** <n° node paramètres> : Change le type d'un node

**wait** <n millisecondes> : Exécute les instructions qui suivent n millisecondes après, exemple `wait 5000`, attend cinq secondes avant d'exécuter ce qui suivra le `wait`.

**creat\_connect** <n° node départ n° node arrivée> : créer une connexion de type neuronale entre deux nodes.

**create\_connec\_basic** <n° node départ n° node arrivée> : créer une connexion basic entre deux nodes

**create\_connec\_echelle** <n° node départ n° node arrivée> : créer une connexion échelle entre deux nodes

**create\_connec\_axis** <n° node départ n° node arrivée> : créer une connexion pour les axis entre deux nodes

**create\_connec\_allies** <n° node départ n° node arrivée> : créer une connexion pour les alliés entre deux nodes

**destroy\_connec** <n° node départ n° node arrivée> : détruire une connexion entre deux nodes

**Nota :**

*Les bots ne bougeront pas si dans le script il n'y pas au moins les paramètres `objectif_pri` ou `protection`. Seuls les paramètres font bouger les bots, sans eux les bots resteront à leur spawn point, préférant fumer une clope ou boire une bière plutôt qu'aller se faire flinguer pour rien comme des omni-bot.*

## Paramètres du set

*Nota : Les paramètres peuvent être joints les uns aux autres dans n'importe quel sens.*

*Exemple :*

*set 5 move objectif\_pri tank mobil linked heavy\_machinegun*

*set 8 move protection axis duck*

**all** : tous les nodes sont concernés. (Concerne les nodes). Pas de priorité sur ce node.

Exemple : *set all move* (tous les nodes sont des voies de passage).

**x to y** : du node x au node y, (Concerne les nodes).

Exemple: *set 5 to 20 move tankway*.

Les nodes du n° 5 au n° 20 sont tous flagués comme move + tankway. (avec  $y-x > 1$ ) Exemple : set 3 to 4 ne fonctionne pas, set 3 to 5 fonctionne.

**move** : les bots peuvent empreinter ce node comme voie de passage. Quand on crée un node, il est move par défaut. Un node qui n'est pas **move** sera ignoré par les bots, donc n'oubliez jamais de mettre **move** après le n° du node dans la commande **set**. (Concerne toutes les classes de bots). Pas de priorité sur ce node.

**objectif\_pri** : ce node est un objectif prioritaire. (Concerne toutes les classes de bots sauf l'engineer en général). Ce node de haute priorité et sera choisi parmi les premiers par les bots.

Exemple : Les Alliés doivent aller sur un objectif prioritaire (node de n° 5).

*Set 5 move objectif\_pri axis*

*Pourquoi axis, alors que ce sont les Alliés qui doivent y aller ?*

Tout simplement parce que c'est forcément un objet qui appartient à l'équipe adverse, sinon les Alliés n'auraient pas de raison d'y aller. Donc pour faire aller les Alliés sur un objectif prioritaire, il faut préciser la **team adverse**.

Exemple :

*set 5 move objectif\_pri axis* (ce sont les **Alliés** qui vont sur le node 5)

*set 5 move objectif\_pri allies* (ce sont les **Axis** qui vont sur le node 5)

*set 5 move objectif\_pri* (pas de team précisée, **Axis** et **Alliés** vont sur le node 5)

A titre d'exemple, nous avons un node n°5 (document que les Alliés doivent prendre), à gauche du node n° 5, le node n° 8 et à droite du node n° 5 le node n° 24. Si on fait *set 5 move objectif\_pri*, tous les bots iront sur le node n° 5 et forcément les Axis rencontrant les Alliés, il y a attaque – défense. Mais le mieux est de faire :

*set 5 move objectif\_pri axis* (les Alliés vont sur le node n° 5 pour prendre le document Axis)

*set 8 move protection axis* (Les Axis protègent le node n° 8, à gauche du node n° 5)

*set 24 move protection axis* (Les Axis protègent le node n° 24, à droite du node n° 5)

De cette façon chaque team connaît sa mission plus précisément et plus logiquement. Il faudra faire pareil pour les objectifs secondaires.

**objectif\_sec**: ce node est un objectif secondaire. (Concerne toutes les classes de bots).

Exemple : Les Alliés doivent aller sur un objectif secondaire (node de n° 5). Ce node de priorité moindre et sera choisi après les nodes **objectif\_pri** et **protection**.

*Set 5 move objectif\_sec axis*

*Pourquoi axis, alors que ce sont les Alliés qui doivent y aller ?*

Tout simplement parce que c'est forcément un objet qui appartient à l'équipe adverse, sinon les Alliés n'auraient pas de raison d'y aller. Donc pour faire aller les Alliés sur un objectif secondaire, il faut préciser la **team adverse**.

Exemple :

*set 5 move objectif\_sec axis* (ce sont les **Alliés** qui vont sur le node 5)

*set 5 move objectif\_sec allies* (ce sont les **Axis** qui vont sur le node 5)

*set 5 move objectif\_pri* (pas de team précisée, **Axis** et **Alliés** vont sur le node 5)

**protection** : c'est un node à protéger. (Concerne toutes les classes de bots). Ce node de haute priorité et sera choisi parmi les premiers par les bots.

Exemple : Les Alliés doivent protéger un endroit (node de n° x).

### ***Set x move objectif\_sec allies***

Cette fois ci il faut préciser la **même team** car si c'est un endroit à protéger c'est qu'il appartient à la **même team**. En effet les Alliés n'ont pas de raison de protéger un emplacement Axis.

Exemple :

***set 5 move protection axis*** (ce sont les **Axis** qui vont sur le node 5)

***set 5 move protection allies*** (ce sont les **Alliés** qui vont sur le node 5)

***set 5 move protection axis allies*** (**Axis** et **Alliés** vont sur le node 5)

**axis** : Flag supplémentaire les les objectifs, les mines, ou les protections.

**allies** : Flag supplémentaire les les objectifs, les mines ou les protections.

**tank** : Le node est lié à un véhicule. (Fonctionne avec **mobil**, doit être suivit d'un node **tankway**). (Concerne toutes les classes de bots)

**mobil** : Node mobile qui suit un véhicule en mouvement. (Fonctionne avec **tank**). (Concerne les nodes).

Exemples :

***set 5 move tank mobil***

***set 5 move tank mobil heavy\_machinegun*** (et le ms42 du tank est en plus utilisable).

***set 5 move tank mobil linked*** (et le tank est réparable).

***set 5 move tank mobil heavy\_machinegun linked*** (toutes les fonctionnalités du tank).

***set 5 move tank mobil heavy\_machinegun linked objectif\_pri*** (et toutes les teams vont sur le tank ).

**tankway** : C'est un node dont la connexion est reliée à un node de type **tank** (véhicule en mouvement) afin que les bots retrouvent ce véhicule facilement. (doit être précédé d'un node de types **tank mobil**). (Concerne les véhicules).

***Utilisation correcte d'un véhicule en mouvement (un tank par exemple).***

***Le node du tank est le 0 (exemple)***

***Le long du chemin du tank on a disposé 10 nodes tankway (exemple 1 à 10). Les connexions entre le tank et les tankway doivent être à double sens.***

***Au début quand on fait le waypointing , le node 0 du tank doit être relié au 1er node tankway (ici le n°1) et au second node tankway (ici le n° 2) et à aucun autre node car le node du tank étant mobile vous auriez des connexions pouvant mettre les bots dans des situations délicates. Bien sûr tous les nodes tankway doivent être reliés entre eux 1 à 2, 2 à 3 etc... Les nodes tankway (1 à 10) peuvent et doivent être reliés à d'autres nodes. Regardez pour exemple le fichier venice.nod ou goldrush.nod ou tout autre map.nod ou il y a un véhicule en mouvement.***

***Dans le nodscript on écrira :***

***set 0 move tank mobil heavy\_machinegun linked objectif\_pri***

***set 1 to 10 move tankway***

**linked** : Le node est lié à une entité et prend ses attributs. (Concerne l'engineer qui saura quoi faire). Spécifique aux objets dynamitables ou constructibles.

Exemple : ***set 5 move linked***

***Nota : Pourquoi se servir de linked plutôt que de construction et dynamite ?***

***Parce que le type linked est dynamique, il met à jour le type du node dès qu'un évènement s 'est produit dessus.***

***Exemple : set 5 move linked***

***Au début il faut construire un command post.***

***linked transformera ce node en :***

***set 5 move construction axis*** si les **Axis** doivent le construire.

***set 5 move construction allies*** si les **Alliés** doivent le construire.

***Quand le command post est détruit, linked transformera ce node en :***

***set 5 move dynamite axis*** si les **Alliés** doivent le détruire.

***set 5 move dynamite allies*** si les **Axis** doivent le détruire.



*Il est plus facile et rapide d'écrire dans l'initiation du script (ou ailleurs).*

**trigger init**

```
{  
set 5 move linked           // les bots sauront quoi faire automatiquement  
set x ...  
set x ...  
}
```

*Que si le **wm\_objective\_status** du command post est 3*

*// Construit par les Alliés*

**trigger 3 1 1**

```
{  
set 5 move dynamite allies // Les Axis doivent le détruire  
}
```

*// Failed par Alliés (Détruit par les Axis)*

**trigger 3 1 2**

```
{  
set 5 move construction allies // Les Alliés doivent le construire  
}
```

*// Construit par les Axis*

**trigger 3 0 1**

```
{  
set 5 move dynamite axis // Les Alliés doivent le détruire  
}
```

*// failed par Axis (Détruit par les Alliés)*

**trigger 3 0 2**

```
{  
set 5 move construction axis // Les Axis doivent le construire  
}
```

**Construction** : Un objet à construire (préciser à quelle team appartient cet objet). (Concerne l'engineer). Node de haute priorité.

Exemple :

**set 5 move construction axis** si les Axis doivent le construire.

**set 5 move construction allies** si les Alliés doivent le construire.

**dynamite** : Un objet à détruire avec une dynamite. (préciser à quelle team appartient l'objet à détruire). (Concerne l'engineer). Node de haute priorité.

Exemple :

**set 5 move dynamite axis** c'est un objet axis que les Alliés doivent le détruire.

**set 5 move dynamite allies** c'est un objet allié que les Axis doivent le détruire.

Vous pouvez ajouter **charge** à la suite si l'objet peut aussi être détruit avec un **satchel**.

**set 5 move dynamite charge axis**

**set 5 move dynamite charge allies**

**mine** : Une mine à placer par un engineer. Il faut préciser quelle team plante la mine. (Concerne l'engineer). Ce node est de priorité moindre à **construction** et **dynamite**.

Exemple :

**set 5 move mine allies** (les engineer Alliés plante une mine sur le node 5).

**set 5 move mine axis** (les engineer Axis plante une mine sur le node 5).

**set 5 move mine allteam** (les engineer Alliés ou Axis plante une mine sur le node 5).

**Charge** : Un objet à détruire avec un satchel. (Concerne le covertops). Ce node est de priorité moindre à **objectif\_pri objectif\_sec** et **protection**.

Exemple : Les Alliés covertops doivent placer une charge sur un objet (node de n° x).

**set x move charge axis**

**Pourquoi axis, alors que ce sont les Alliés qui doivent placer la charge ?**

Tout simplement parce que c'est forcément un objet qui appartient à l'équipe adverse, sinon les Alliés n'auraient pas de raison d'y placer une charge. Donc pour faire placer une charge par les Alliés sur un objet, il faut préciser la **team adverse**.

Exemple :

**set 5 move charge axis** (ce sont les *Alliés* qui vont placer la charge sur le node 5).

**set 5 move charge allies** (ce sont les *Axis* qui vont placer la charge sur le node 5).

**blockable** : bloquer un bot sur un node. (Pour qu'il reste sur un bateau, exemple dans venice). (Concerne toutes les classes de bots). Pas de priorité sur ce node, la commande est exécutée quand le bot est sur ce node.

Exemple : **set 5 move blockable**

**allteam** : Team Axis et Alliés. (pour les nodes *mine* et *blockable* actuellement)

**healer** : Un node sur un cabinet de soin. (les deux team s'en serviront). (Concerne toutes les classes de bots)

Ce node est géré par la logique floue, Les bots iront dessus en fonction de la distance du cabinet, de leur état de santé ...

Exemple : **set 5 move healer**

**supplier** : Un node sur un cabinet de munitions. (les deux team s'en serviront). (Concerne toutes les classes).

Ce node est géré par la logique floue, Les bots iront dessus en fonction de la distance du cabinet, du nombre de munitions qui leur reste ...

Exemple : **set 5 move supplier**

**Nota :**

Si un même node est sur un cabinet de soin et sur un cabinet de munitions ou entre les deux parce que les cabinets sont très proches, on peut écrire **set 5 move healer supplier**.

**duck** : Marcher courbé doucement ou se mettre à genoux (dépend de l'action en cours du bot). (Concerne toutes les classes de bots). Pas de priorité sur ce node, la commande est exécutée quand le bot est sur ce node.

Exemple : **set 5 move duck**.

**heavy\_machinegun** : MG nest , mitrailleuse du tank. (les deux team s'en serviront). (Concerne toutes les classes sauf l'engineer). Node de basse priorité.

Exemple : **set 5 move heavy\_machinegun**

**prone** : Se coucher sur ce node. (Concerne toutes les classes de bots). Pas de priorité sur ce node, la commande est exécutée quand le bot est sur ce node.

Exemple : **set 5 move prone**

**take** : Une porte à ouvrir, une manette à enfoncer (toutes équipes confondues). (Concerne toutes les classes de bots). Pas de priorité sur ce node, la commande est exécutée quand le bot est sur ce node.

**Exemple : set 5 move take**

**Nota :** *L'engineer ne va que sur les nodes qui lui sont spécifiques (construction, dynamite, mine). Il se consacre essentiellement à la construction, destruction et pose de mines. S'il n'y a pas ce genre de node dans une map, en dernier recours il ira quand même sur les objectifs primaires, secondaires et endroits à protéger pour ne pas rester inactif. Il ne faut pas confondre un objectif primaire/secondaire avec un objectif à détruire ou à construire. Dans godlrush l'or est un objectif primaire, il ne concerne pas directement un engineer. Dans oasis le big gun nord et sud sont des objectifs primaires mais ne concernent que les engineers.*

*Si pour oasis on écrit (avec le node 5 sur le gun nord)*

**set 5 move linked objectif\_pri axis**, tous les bots alliés iront sur ce node, l'engineer avec le « linked » et les autres avec le « objectif\_pri ». Linked transforme ce node en set 5 move dynamite axis donc pas la peine de rajouter axis. + objectif\_pri que l'on a mis on a en réalité **set 5 move dynamite objectif\_pri axis**.

**set 5 move linked objectif\_pri allies**, tous les bots axis et alliés iront sur ce node (sauf l'engineer axis s'il a autre chose à construire ou détruire), l'engineer allié avec le « linked » et les autres avec le « objectif\_pri ». Linked transforme ce node en set 5 move dynamite axis + objectif\_pri allies que l'on a mis. On a donc en réalité **set 5 move dynamite axis objectif\_pri allies**.

**set 5 move linked** (tous les bots engineer alliés iront sur ce node). Linked transforme ce node en **set 5 move dynamite axis**.

Dans godlrush si on écrit (avec le node 5 sur l'or dans la banque)

**set 5 move objectif\_pri allies** (tous les bots axis iront sur ce node, \* sauf les engineers).

**set 5 move objectif\_pri axis** (tous les bots alliés iront sur ce node, \* sauf les engineers).

**set 5 move objectif\_pri** (tous les bots axis et alliés iront sur ce node, \* sauf les engineers).

**set 5 move linked objectif\_pri** (tous les bots axis et alliés iront sur ce node, \* sauf les engineers)

linked ne transforme pas ce node en construction ou dynamite car ce n'est pas un objet dynamitable ou constructible.

\* Sauf les engineers : s'il y a un objet à construire ou à détruire ou des mines à poser en priorité.

---

A titre d'exemple nous prendront le nodescript de fueldump, vous verrez qu'avec peu d'instructions les bots peuvent réaliser les objectifs. Nous avons disposé des nodes dans la map et relevé le n° de ceux qui nous intéressent.

## Fichier fueldump.nodescript

### Les Nodes

0	: Tank
1 à 20	: Le long de la route qu'empreinte le Tank
36	: Construire Grand pont Alliés
3<==>4	: Connexion de chaque coté du Grand pont
7<==>8	: Connexion lorsque la porte tunnel est détruite
110<==>133	: Connexion lorsque le mur latéral du bunker est détruit
111<==>95	: Connexion lorsque la porte du bunker est détruite
130	: Construire/Détruire barricade de protection fuel coté gauche
137	: Construire/Détruire barricade de protection fuel coté droit
156 et 157	: Deux nodes ou détruire le fuel dump
80	: Command post Axis
74	: Mirador MG42 Axis (partie 1)
164	: Cabinet de soin et munitions Alliés (partie 2)
159	: Command post Alliés
99	: MG42 Alliés/Axis (coté bunker)
43	: Construire foot-bridge
30	: Devant la porte du bunker axis
134	: Devant le command post alliés
59	: Cour intérieure axis coté mur latéral
58	: Cour intérieure axis coté porte
96	: Coté cour Mg42 axis
81 et 151	: Devant le pont coté alliés
62	MG42 sur le bunker axis
32	: MG42 Alliés (partie 1)

162 et 163 : Porte hangar gauche et droite

Au début du jeu tous les objectifs sont neutres comme nous l'avons vu dans le script original de la map. Nous ferons pareil. La syntaxe est la suivante.

```
trigger init           // début du jeu
{
    // parenthèse de début de trigger.
set all move           // Initialisation des nodes, ils sont tous des voies de passages
set 164 move supplier <Je_vais_recharger_mon_arme // cabinet de munitions
}
// parenthèse de fin de trigger.
```

### Que doivent faire les Axis ?

- 1) Monter le poste de commandement.
- 2) Monter le mirador Mgnest.
- 3) Défendre.

```
trigger init
{
set all move
set 164 move supplier
```

```
// Construire / Détruire le mirador axis
set 74 move linked <Je_me_dirige_vers_le_mirador
// Construire / détruire le poste de commandement Axis
set 80 move linked <Je_vais_au_poste_de_commandement_axis
// nodes à protéger par les axis. Les axis resteront dans ces parages en défense.
set 77 move protection axis <Je_vais_en_haut_de_la_colline
set 69 move protection axis <Je_vais_en_haut_du_bunker
}
```

### Que doivent faire les Alliés ?

- 1) Amener le tank devant la porte du tunnel et la détruire.
- 2) Construire le pont principal.
- 3) Construire le footBridge.
- 4) Construire leur Mgnest.

```
trigger init
{
set all move
set 164 move supplier
set 74 move linked <Je_me_dirige_vers_le_mirador
set 80 move linked <Je_vais_au_poste_de_commandement_axis
set 77 move protection axis <Je_vais_en_haut_de_la_colline
set 69 move protection axis <Je_vais_en_haut_du_bunker
```

```
set 0 move objectif_pri mobil tank linked heavy_machinegun <Je_vais_vers_le_tank //Tank
set 1 to 20 move tankway //Tank way
// Construire / détruire le MG42 alliés.
set 32 move linked <I'm_going_to_the_allied_mgnest
// Construire / détruire le grand pont
set 36 move linked prone <I'm_going_to_the_bridge
// Construire / détruire le footbridge, prone
set 43 move linked prone <I'm_going_to_the_foot bridge
}
```



Vous noterez les phases commençant par < , ce sont des messages que vous pouvez mettre à la suite de la commande. Le bot le dira à sa team en chat quand il aura pris la décision d'aller sur ce node. Les mots doivent être séparés par \_.

Il aura fallut 11 commandes pour initialiser le jeu.

```
trigger init
{
set all move
set 164 move supplier
set 74 move linked <Je_me_dirige_vers_le_mirador
set 80 move linked <Je_vais_au_poste_de_commandement_axis
set 77 move protection axis <Je_vais_en_haut_de_la_colline
set 69 move protection axis <Je_vais_en_haut_du_bunker
set 0 move objectif_pri mobil tank linked heavy_machinegun <Je_vais_vers_le_tank
set 1 to 20 move tankway
set 32 move linked <I'm_going_to_the_allied_mgnest
set 36 move linked prone <I'm_going_to_the_bridge
set 43 move linked prone <I'm_going_to_the_foot bridge
}
```

Maintenant les triggers objectifs principaux et secondaires qu'E.T envoie à bobot.

// Alliés construisent le pont, on crée juste une connexion entre les deux nodes se trouvant de chaque coté du pont afin que les bots puissent passer.

**trigger 1 1 1**

```
{
create_connec 4 5
create_connec 5 4
}
```

//Axes détruit le pont ou alliés n'arrivent pas à le construire, on détruit ces connexions, les bots ne passeront plus.

**trigger 1 1 2**

```
{
destroy_connec 5 4
destroy_connec 4 5
}
```

***Nota : On aurait pu écrire ce trigger 1 0 1 car 1 1 2 est équivalent à 1 0 1, puisque objectif 1 raté par les alliés est égal à objectif 1 réussi par les axis, dans la plupart des cas.***

// Alliés détruisent la porte du tunnel

**trigger 2 1 1**

```
{
create_connec 7 8 // création des connexions entre les deux nodes se trouvant de
create_connec 8 7 // chaque coté de la porte du tunnel.
```

// On remet à zéro tous les nodes du début de partie car ces objectifs n'ont plus d'intérêt et il ne faut pas que les bots y retournent.

**set all move** // tous les nodes ne sont que des voies de passages.

// Tous les nodes étant à zéro, on réinitialise le tank, comme au début, car lui aussi à été remis à zéro par la commande set all move.

**set 0 move objectif\_pri mobil tank linked heavy\_machinegun <Je\_vais\_vers\_le\_tank**

**set 1 to 20 move tankway**

// on déclare les nouveau objectifs.

**set 95 move protection allies axis <Je\_me\_dirige\_vers\_la\_porte\_du\_bunker**

```
set 130 move linked <Je_vais_a_la_barriere_de_protection_gauche
set 137 move linked <Je_vais_a_la_barriere_de_protection_droite
set 159 move linked <Je_vais_au_poste_de_commandement_alliés
set 99 move linked <Je_vais_au_MG42_du_poste_de_commandement
set 163 move take <Je_vais_fermer_la_portes_gauche_du_hangard
set 162 move take <Je_vais_fermer_la_portes_droite_du_hangard
set 148 move heavy_machinegun <Je_vais_au_mg42_du_bunker
set 102 move mine axis // Axis posent des mines
set 92 move mine axis // Axis posent des mines
set 110 move mine axis // Axis posent des mines
}
```

//Alliés détruisent la porte du dépôt

```
trigger 3 1 1
```

```
{
```

//création de la connexion entre les 2 nodes se trouvant de chaque coté de la porte.

```
create_connec 95 111
```

```
create_connec 111 95
```

```
set 156 move linked <Je_vais_au_depot_de_carburant_est
```

```
set 157 move linked <Je_vais_au_depot_de_carburant_ouest
```

```
set 131 move objectif_pri <Je_fonce_du_cote_du_depot_de_carburant_ouest
```

```
set 134 move objectif_pri <Je_fonce_du_cote_du_depot_de_carburant_est
```

```
}
```

// Alliés détruisent le mur latéral.

```
trigger 5 1 1
```

```
{
```

//création de la connexion entre les 2 nodes se trouvant de chaque coté du mur.

```
create_connec 110 133
```

```
create_connec 133 110
```

// on a plus besoin du tank ni de son tankway, on remet ces nodes à zéro (pas obligatoire), les autres nodes sont toujours valables.

```
set 0 to 20 move
```

```
}
```

Et c'est fini pour les objectifs principaux et secondaires.

On passe aux autres triggers qui ne sont pas des wm\_objective\_status.

//footbridge construit, les bots passent

```
trigger footbridge etat final
```

```
{
```

```
create_connec_basic 22 41
```

```
create_connec_basic 41 22
```

```
set 43 move dynamite charge allies // node que les axis peuvent dynamiter-charge
```

```
set 41 move dynamite charge allies // node que les axis peuvent dynamiter-charge
```

```
}
```

//footbridge détruit, les bots ne passent plus

```
trigger footbridge etat death
```

```
{
```

```
destroy_connec 22 41
```

```
destroy_connec 41 22
```

```
set 43 move linked prone
```

```
set 41 move
```

```
}
```

// les Alliés reconstruisent sur ce node

// il n'est plus dynamite-charge

```
//Axis Command Post constructed
trigger axis_radio_built built final
{
set 80 move dynamite charge axis           // node que les Alliés peuvent dynamiter-charge
}
```

```
//Allied team has destroyed the Axis Command Post
trigger axis_radio_built etat death
{
set 80 move linked                         // Axis reconstruisent sur ce node
}
```

```
//Allied Command Post constructed
trigger allied_radio_built built final
{
set 159 move dynamite charge allies       // node que les Axis peuvent dynamiter-charge
}
```

```
//Axis team has destroyed the Allied Command Post
trigger allied_radio_built etat death
{
set 159 move linked                       // Alliés reconstruisent sur ce node
}
```

```
//Barricade gauche du fuel dump détruite, les bots peuvent aller dans le dépôt de fuel par ce coté là
trigger reardump etat death
{
create_connec 136 157
create_connec 157 136
set 137 move linked                       // Axis reconstruisent sur ce node
set 136 move                             // il n'est plus dynamite-charge
}
```

```
//Barricade gauche du fuel dump construite, les bots ne peuvent plus aller dans le dépôt de fuel par ce coté là.
trigger reardump etat final
{
destroy_connec 136 157
destroy_connec 157 136
set 137 move dynamite charge axis       // node que les Alliés peuvent dynamiter-charge
set 136 move dynamite charge axis       // node que les Alliés peuvent dynamiter-charge
}
```

```
//Barricade droite du fuel dump détruites, les bots peuvent aller dans le dépôt de fuel par ce coté là
trigger frontdump etat death
{
create_connec 128 156
create_connec 156 128
set 128 move                             // Axis reconstruisent sur ce node
set 130 move linked                     // il n'est plus dynamite-charge
}
```

```
//Barricade droite du fuel dump construite, les bots ne peuvent plus aller dans le dépôt de fuel par ce coté là.
trigger frontdump etat final
{
destroy_connec 128 156
destroy_connec 156 128
set 128 move dynamite charge axis       // node que les Alliés peuvent dynamiter-charge
}
```

**set 130 move dynamite charge axis**

// node que les Alliés peuvent dynamiter-charge

}

//Porte 2 du garage, ouvertes les bots passent

**trigger tbdoor2\_lever1 etat up**

{

**create\_connec 16 160**

**create\_connec 160 16**

}

//Porte 2 du garage, fermée les bots ne passent pas

**trigger tbdoor2\_lever1 etat down**

{

**destroy\_connec 16 160**

**destroy\_connec 160 16**

}

//Porte 1 du garage, ouvertes les bots passent

**trigger tbdoor1\_lever1 etat up**

{

**create\_connec 161 165**

**create\_connec 165 161**

}

//Porte du garage, fermée les bots ne passent pas

**trigger tbdoor1\_lever1 etat down**

{

**destroy\_connec 161 165**

**destroy\_connec 165 161**

}

// Fin du script fueldump.nodscript

Les trigger de type footbridge, tbdoor1\_lever1, ne sont pas des wm\_objective\_status mais souvent les déclenchent. Ils se trouvent dans le script original de la map. Vous devez respecter la syntaxe suivante dans le script bobot.

Trigger NomDuTrigger EtatDuTrigger ValeurDuTrigger

{

}

Voici ce que l'on trouve dans fueldump.script en ce qui concerne le footbridge

// =====

// Footbridge on Allied side of the map

// =====

**footbridge**

{

spawn

{

wait 200

constructible\_class 2

setstate footbridge invisible

setstate footbridgematerials default

setstate lms\_footbridge invisible

}

buildstart final

```

{
    setstate footbridge underconstruction
}

built final
{
    setstate footbridge default
    setstate footbridgematerials invisible
    // Some kind of UI pop-up to alert players
    wm_announce "Allied team has constructed the Foot Bridge!"
}

decayed final
{
    setstate footbridge invisible
    setstate footbridgematerials default
}

death
{
    setstate footbridge invisible
    setstate footbridgematerials default
    // Some kind of UI pop-up to alert players
    wm_announce "Axis team has destroyed the Foot Bridge!"
}
}

```

Il ne faudra retenir que le nom du trigger footbridge (1ère ligne, ici en gras) puis ses états (en gras) et leur valeur, en général ils sont suivis d'une annonce E.T (wm\_announce) et d'un wm\_objective\_status.

**Nota : On trouvera dans le script de la map trois syntaxes relatives aux triggers.**

Exemples :

- 1) footbridge built final
- 2) footbridge death
- 3) documents trigger stolen

Dans ces exemple nous prendrons :

- 1) footbridge : Le nom du trigger
- 2) death : Quand il est détruit
- 1) built final : Quand il est construit.

3) documents : Le nom du trigger

3) stolen : Quand il a été volé

- 1) **built** est l'état , final est sa valeur, dans ce cas là il faut prendre l'état et sa valeur.
- 2) **death** est l'état sans valeur, dans ce cas là il faudra intercaler entre footbridge et death la constante **etat** afin de respecter la syntaxe trigger Nom Etat Valeur.
- 3) **trigger** est un mot clef bobots, il doit également être remplacé par la constante **etat**.

Ce qui donnera dans le nodscript botots

Trigger footbridge etat death // On a ajouté la constante etat

```

{
    à faire...

```



```
}
```

Trigger footbrige built final // On a pris le nom l'état et sa valeur tels qu'ils sont dans le script.

```
{
```

à faire...

```
}
```

Trigger documents etat stolen // On a pris le nom, remplacé trigger par etat, et pris la valeur.

```
{
```

à faire...

```
}
```

**Voici le fichier fueldump,nodscript fini, tel que s'en sert bobot.**

```
// 04.jan.2007 PAV
```

```
// 1: Primary1 : Empêchez les alliés de construire et de renforcer un pont au-dessus du fleuve gelé.  
// 2: Primary2 : Empêchez le char allié de détruire les portes de tunnel.  
// 3: Primary3 : Neutralisez le char allié avant qu'il puisse détruire les portes de dépôt.  
// 4: Primary4 : Défendez le dépôt de carburant.  
// 5: Secondary1 : Défendez le mur latéral endommagé.  
// 6: Secondary2 : Arrêtez l'Axe dans la construction d'un poste de commandement.  
// 7: Secondary3 : Arrêtez les Alliés dans la construction d'un poste de commandement.
```

```
// Les Nodes -----
```

```
//0 : Tank  
// 1-20 : Route du Tank  
// 36 : Construire Grand pont Alliés  
// 4<==>5 : Traverser Grand pont  
// 7<==>8 : Traverser porte tunnel  
// 110<==>133 : Traverser mur latéral  
// 111<==>95 : Traverser porte du bunker  
// 130 : Construire/Détruire protection fuel gauche  
// 137 : Construire/Détruire protection fuel droite  
// 156 et 157 : Détruire fuel dump  
// 80 : Command post Axis  
// 74 : Mirador MG42 Axis (partie 1)  
// 164 : Health et Ammo Alliés  
// 159 : Command post Alliés  
// 99 : MG42 Alliés/Axis (coté bunker)  
// 43 : Construire foot-bridge  
// 95 : Devant la porte du bunker axis  
// 134 : Cour intérieure axis coté mur latéral  
// 132 : cour intérieure axis coté porte  
// 148 : MG42 sur le bunker axis  
// 32 : MG42 Alliés (partie 1)  
// 162 et 163 : porte hangar gauche et droite
```

```
trigger init
```

```
{
```

```
set all move
```

```
set 26 move healer supplier <I'm_going_to_the_allied_supply_cabinet
```

```
set 0 move objectif_pri mobil tank linked heavy_machinegun <I'm_going_to_escort_the_tank
```

```
set 1 to 20 move tankway
```

```
set 4 move tankway protection allies <I'm_going_to_protect_the_tank
```

```
set 74 move linked <I'm_going_to_the_axis_mirador
```

```
set 32 move linked <I'm_going_to_the_allied_mg_nest
set 36 move linked prone <I'm_going_to_the_bridge
set 43 move linked prone <I'm_going_to_the_foot_bridge
set 80 move linked <I'm_going_to_the_axis_command_post
set 83 move prone
set 72 move prone
set 35 move prone
set 77 move protection axis <I'm_going_to_the_axis_hill
set 69 move protection axis <I'm_going_to_the_top_bunker
set 37 move protection axis
set 130 move linked <I'm_going_to_the_est_base_fortifications
set 137 move linked <I'm_going_to_the_west_base_fortifications
set 99 move linked <I'm_going_to_the_mirador
}
```

// Alliés construisent le pont

```
trigger 1 1 1
{
create_connec 4 5
create_connec 5 4
set 4 move tankway
set 6 move tankway protection allies <I'm_going_to_protect_the_tank
}
```

//Axes détruit le pont

```
trigger 1 1 2
{
destroy_connec 5 4
destroy_connec 4 5
set 6 move tankway
set 4 move tankway protection allies <I'm_going_to_protect_the_tank
}
```

// Alliés détruisent la porte du tunnel

```
trigger 2 1 1
{
set all move
set 0 move objectif_pri mobil tank linked heavy_machinegun
set 1 to 20 move tankway
set 95 move protection allies axis <I'm_going_to_the_bunker_entrance
set 159 move linked <I'm_going_to_the_allied_command_post
set 163 move take <Je_vais_fermer_lesportes_du_hangard
set 162 move take <Je_vais_fermer_lesportes_du_hangard
set 148 move heavy_machinegun <I'm_going_to_the_axis_mg_nest
set 102 move mine axis
set 92 move mine axis
set 110 move mine axis
}
```

//Alliés détruisent la porte du dépôt

```
trigger 3 1 1
{
set 156 move linked <I'm_going_to_the_fuel_dump
set 157 move linked <I'm_going_to_the_fuel_dump
set 132 move protection axis allies <I'm_going_to_protect_the_west_fuel_dump
set 134 move protection axis allies <I'm_going_to_protect_the_east_fuel_dump
}
```

// Alliés détruisent le mur latéral.

trigger 5 1 1

```
{
create_connec 110 133
create_connec 133 110
set 0 to 20 move
}
```

//Axis Command Post constructed

trigger axis\_radio\_built built final

```
{
set 80 move dynamite charge axis
}
```

//Allied team has destroyed the Axis Command Post

trigger axis\_radio\_built etat death

```
{
set 80 move linked
}
```

//Allied Command Post constructed

trigger allied\_radio\_built built final

```
{
set 159 move dynamite charge allies
}
```

//Axis team has destroyed the Allied Command Post

trigger allied\_radio\_built etat death

```
{
set 159 move linked
}
```

//footbridge

trigger footbridge built final

```
{
create_connec_basic 22 41
create_connec_basic 41 22
set 43 move dynamite charge allies
set 41 move dynamite charge allies
}
```

trigger footbridge etat death

```
{
destroy_connec 22 41
destroy_connec 41 22
set 43 move linked prone
set 41 move
}
```

//Barricade west du fuel dump

trigger reardump built final

```
{
destroy_connec 136 157
destroy_connec 157 136
set 137 move dynamite charge axis
set 136 move dynamite charge axis
}
```

```
trigger reardump etat death
{
create_connec 136 157
create_connec 157 136
set 137 move linked
set 136 move
}
```

```
//Barricade east du fuel dump
trigger frontdump built final
{
destroy_connec 128 156
destroy_connec 156 128
set 128 move dynamite charge axis
set 130 move dynamite charge axis
}
```

```
trigger frontdump etat death
{
create_connec 128 156
create_connec 156 128
set 128 move
set 130 move linked
}
```

```
//Portes du garage
trigger tbdoor2_lever1 etat up
{
create_connec 16 160
create_connec 160 16
}
```

```
trigger tbdoor2_lever1 etat down
{
destroy_connec 16 160
destroy_connec 160 16
}
trigger tbdoor1_lever1 etat up
{
create_connec 161 165
create_connec 165 161
}
trigger tbdoor1_lever1 etat down
{
destroy_connec 161 165
destroy_connec 165 161
}
```

