

**Manuale per generare i  
Templates per la Campagna Dinamica  
BADC**

**Versione 1.0**

# INDICE

1. I FILES TEMPLATE: PANORAMICA GENERALE .....	3
2. DESCRIZIONE DEI PLACES TEMPLATES .....	4
3. CREARE I TEMPLATES .....	10
3.1 <i>Prima fase</i> .....	10
3.2 <i>Seconda Fase</i> .....	14
3.3 <i>Terza fase</i> .....	22
4. AEREI .....	23
4. IMMAGINE DELLA MAPPA .....	27
6. CONFIGURAZIONE FINALE .....	27
7. COMMENTI.....	29

# 1. I files Template: panoramica generale

---

In badc non c'è un singolo file template, ma molti, e tutti creati in modo differente.

**geo\_obj.data** : contiene dati molto importanti. Ci sono 4 gruppi principali di dati:

- 1) la localizzazione degli aeroporti, il loro grado di danneggiamento, i posti per il parcheggio (degli aeromobili), la fazione a cui appartengono, codici dei nomi ecc.
- 2) le coordinate dei settori, il tipo (se attaccabile o meno), TTL, la fazione a cui appartengono, ecc.
- 3) i nomi delle città, le loro coordinate, danni, raggio dei rifornimenti, zone spot, la fazione a cui appartengono, ecc
- 4) una lista di relazioni fra i settori e le città, per conoscere quale settore comprende una città e non sarà quindi attaccabile finché quella particolare città non avrà un danneggiamento pari ad almeno il 50%

**city.mis** : questo è un file di missione creato con il FMB (Full Mission Builder). E' il tipo di file che richiede più tempo. Qui voi decidete quali città faranno parte della campagna, e create le zone Spot. Ciascuna di queste zone spot sono posti che possono essere attaccati. Il file indica che tipo di oggetti ci saranno in tali zone, e dove saranno piazzati. Ma in questo file NON CI SONO gli oggetti. Gli indicatori sono una combinazione di waypoints (WP) aerei a differenti altitudini, per rappresentare oggetti differenti (questo punto verrà spiegato in seguito). In seguito il generatore posizionerà gli oggetti "sotto" queste linee di WP.

**tanks\_wp.mis** : questo è il template relativo alla disposizione dei carri armati. Ogni settore di 10x10 km deve avere 2 o più "punti". In seguito il generatore userà 2 di questi "punti" per posizionarvi accampamenti e i gruppi iniziali di 9 carri. Come potete vedere sono necessari almeno due punti, ma di più sarebbe meglio, in modo che il generatore possa creare maggiori combinazioni che non con 2 soli punti, rendendo così maggiormente imprevedibile la disposizione degli accampamenti e dei carri in movimento. Questi "punti" sono degli oggetti "carri" statici, il tipo di carro è indifferente, il generatore considererà le coordinate (i punti). In quei settori dove è impossibile piazzare 2 punti carri, non sarà possibile sferrare attacchi.

**red\_obj.mis** : una missione che contiene un Li-2 in ogni aeroporto. Niente di più.

**blue\_obj.mis** : lo stesso, ma con Ju 52, e numeri di oggetti che iniziano con 10000.

**frontline.mis**: questo file contiene informazioni sullo stato del fronte: ha una bandiera per settore, con eccezione di quelli inattaccabili.

**aircraft.data**: questo file contiene informazioni sugli aerei disponibili, tipi, missioni, classi ecc..

**imap.data** : questo file è una immagine HTML per aiutare la selezione degli obiettivi durante la fase di creazione delle missioni. Contiene i nomi dei settori (e in più degli accampamenti e delle città), che vengono mostrati come pop-up al passaggio del puntatore del mouse.

Per creare i templates per una mappa, è necessario creare tutti i files sopra descritti, e ciascun file deve rispettare determinate condizioni. Come prima impressione i files templates possono sembrare molto complessi, ma in realtà non lo sono.

## 2. Descrizione dei Places templates

---

Questa guida fornisce una spiegazione per ogni tipo di file, e qualche strumento per la creazione semi-automatica di template. Ma prima di iniziare ad esaminare ciascun file, è meglio spiegare la differenza fra "static objects templates" (per gli oggetti) e "places templates" (per i posizionamenti).

Static objects templates sono files create con il FMB, e contengono tutti gli oggetti della mappa. Usualmente contengono migliaia di auto, aerei, AAA, treni ecc., tutti posizionati in una particolare mappa. In seguito, quando viene attaccato un determinato luogo, il generatore di missioni cercherà un set di oggetti nelle vicinanze del posto attaccato, li “copierà” e “incollerà” nella missione creata.

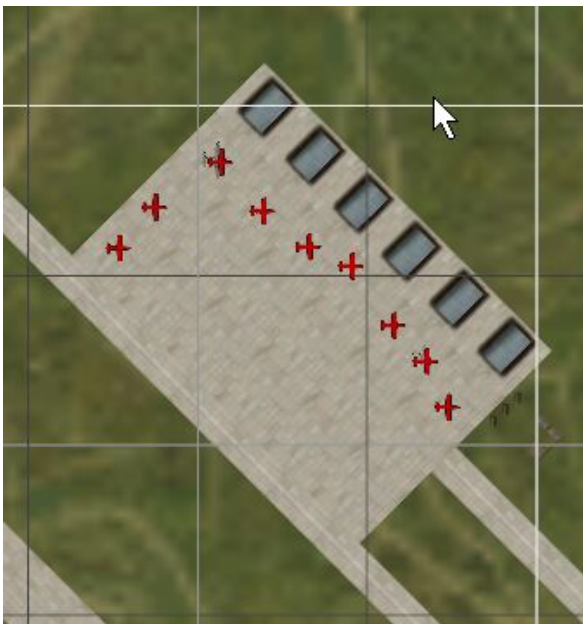
Questo sistema presenta vantaggi e svantaggi. Sono facili da creare, anche se possono richiedere diverse ore, ma in seguito è semplice per il programma generatore usarli con un semplice “copia e incolla”. Lo svantaggio è che non è possibile cambiare il tipo di oggetti, c'è bisogno di un template blu e uno rosso, gli oggetti sono tutti sempre negli stessi posti, e dopo vari attacchi nello stesso luogo i giocatori possono ricordare dove si trovano le difese contraeree e dov'è il “jackpots” (il miglior posto da bombardare).

Badc usa 2 files static objects templates: red\_obj.mis e blue\_obj.mis. Questi contengono solo aerei statici da piazzare sulle basi aeree, 1 per base, usati solo affinché le basi mostrino il corretto colore sulle mappe (non ci sono basi aeree bianche).

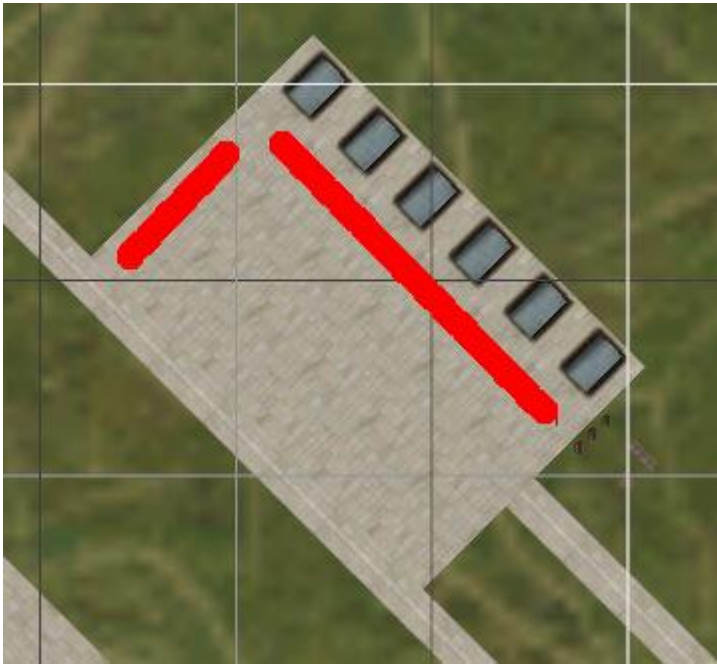
Places templates: questi template non contengono gli oggetti che in seguito saranno nella missione. Ciò che contengono invece sono i “punti” in cui ci piacerebbe che siano posizionati gli oggetti. In seguito il generatore inserirà gli oggetti nel posto giusto nella missione che staremo generando. Tutto qui.

I places templates non sono difficili da creare, è necessario solo seguire qualche regola su come sono segnalati questi posti, affinché in seguito il generatore capisca dove e come gli oggetti debbano essere inseriti. Badc usa "place" templates per gli aeroporti, per i WP dei carri e gli obiettivi nelle città. Vediamo un esempio in generale, in seguito vedremo in dettaglio le regole proprie per ogni singolo template.

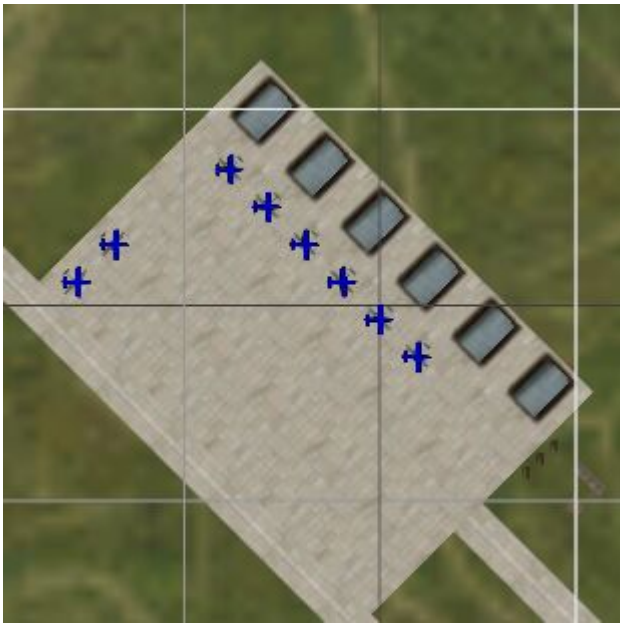
Poniamo che vogliamo creare uno static template per un aeroporto, piazieremo diversi aerie, ed avrà un aspetto simile a questo:



Dopo aver selezionato il tipo di aereo ed averli piazzati, dovremo ruotarli uno per uno, ed in seguito allinearli. OK, ora usiamo lo stesso aeroporto, ed usiamo il concetto del "place template" Noi vogliamo avere qualche aereo allineato in 2 file, come quelle mostrate nell'immagine seguente:

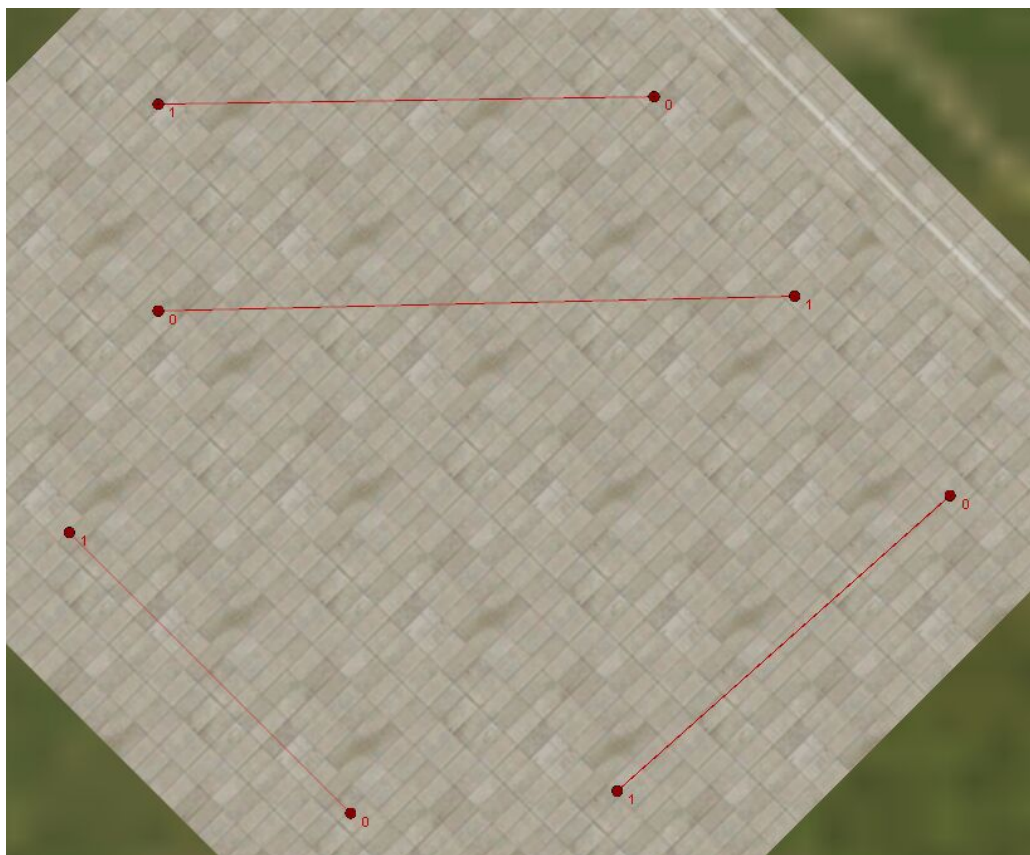


Così in seguito il generatore inserirà gli aerei come mostrato nell'immagine sotto (correttamente allineati e ruotati, forse non si vede bene a causa del formato dell'immagine):



Quindi, **come facciamo creare queste linee** così che il generatore piazzì gli oggetti correttamente? Usiamo un aereo (uno qualsiasi purché flyable) e predisponiamo un **volo (flight)** con dei waypoints, in questo moto “disegniamo” le linee. Forse non è facile da comprendere immediatamente, ma è realmente semplice. Quando piazzate un “volo”, il FMB scrive le coordinate dei waypoints. Piazzando i waypoints e "disegnando" una linea con un aereo farà due cose: ci dice dove sono le coordinate e ci mostra dove saranno piazzati gli oggetti.

Vediamo un'altro esempio “in lavorazione”... la prossima immagine mostra alcuni waypoints dove vogliamo piazzare qualche aereo. Ho usato un B-17 per “tracciare” questi waypoints:

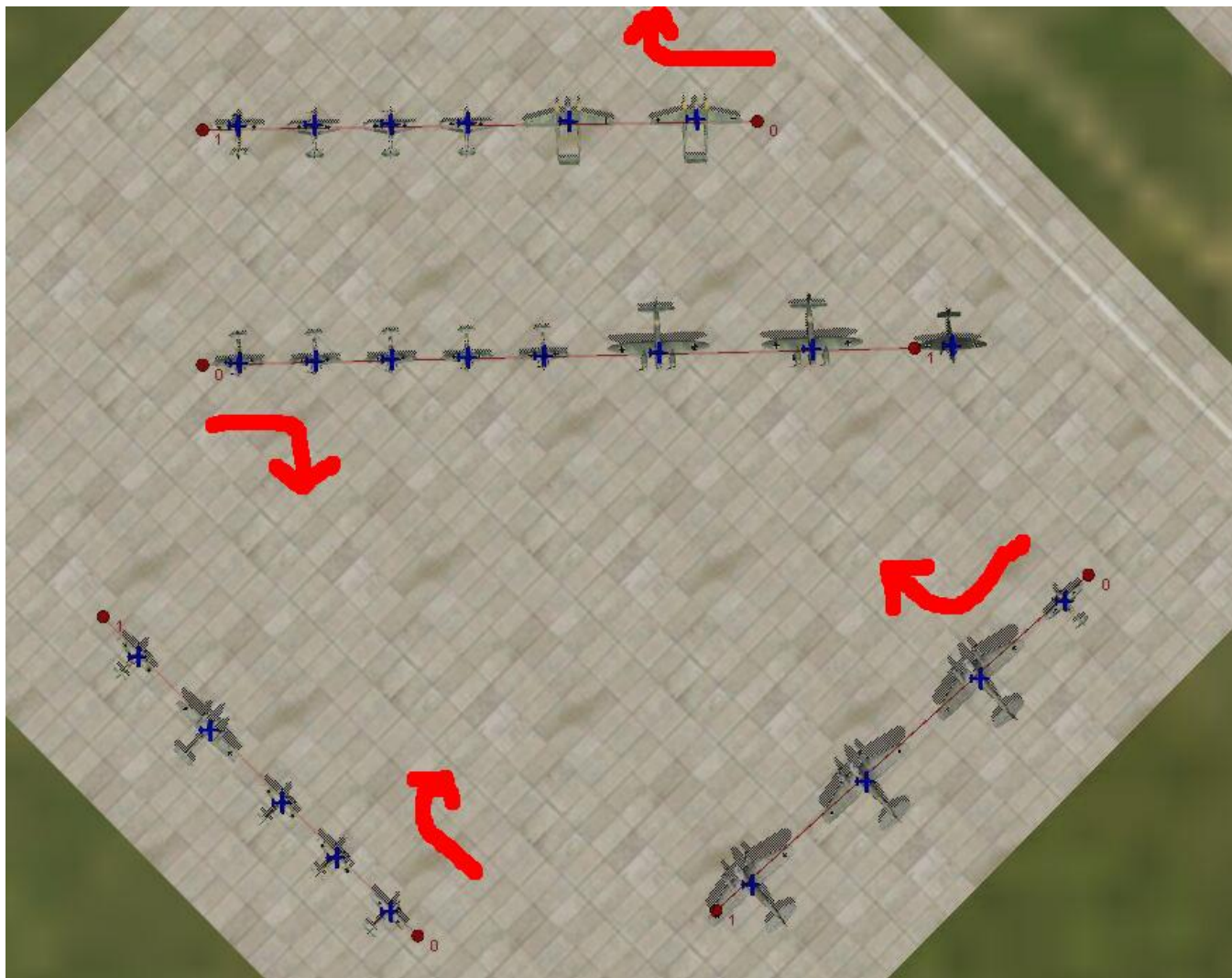


La prossima immagine mostra come sono piazzati gli aerei:



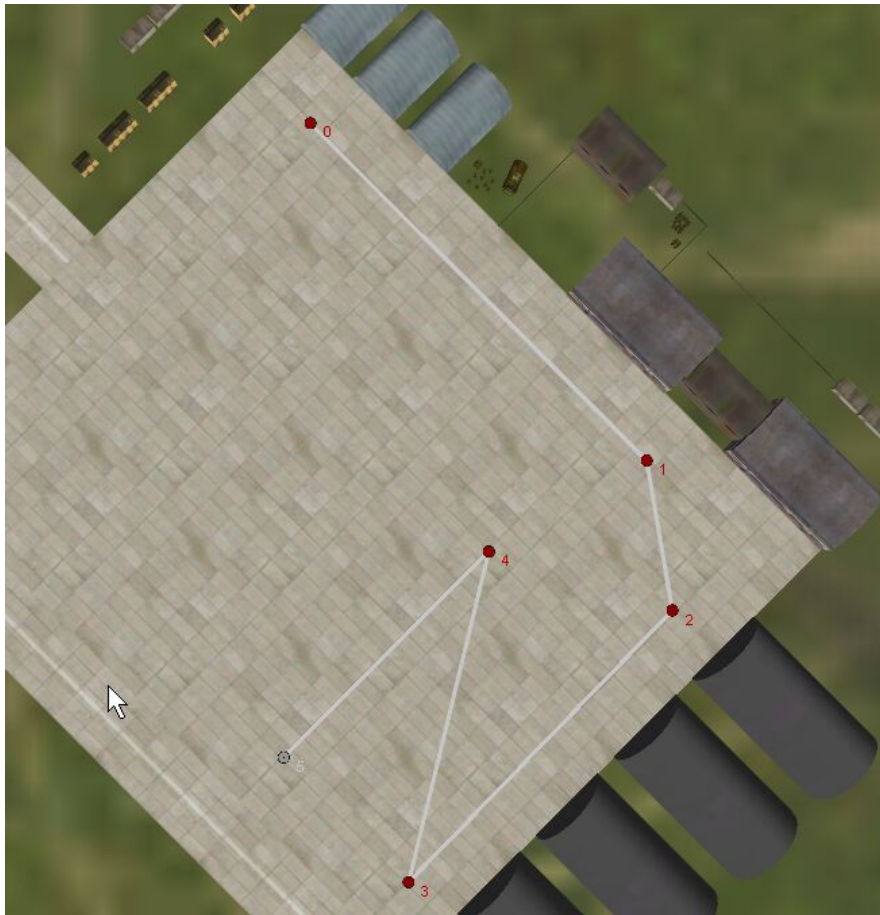


Confrontate le due immagini, e guardate la **direzione** dei waypoints e come sono piazzati gli aerei. Noterete che gli aerei sono sempre piazzati con il muso rivolto a destra della linea, e quando dico “destra” voglio dire: immaginate di stare in piedi sul WP0 e di guardare nella direzione del WP1, gli aerei guarderanno a destra di questa direzione. Per miglior comprensione ho combinato le 2 immagini e disegnato una grossa freccia rossa con la direzione a destra. Quando creerete un template non vedrete gli aerei, e dovrete avere questa immagine in mente:



Questo è il modo in cui funziona un place template. Dovrete solo usare una missione per piazzare i “voli”, non dovrete piazzare oggetti. Abbiate fiducia nel fatto che in seguito il generatore userà il vostro file con i waypoints per leggere le coordinate e disporre gli oggetti a vostro piacimento. Potrete usare qualsiasi aereo per “disegnare” le linee, di qualsiasi tipo o forza. Il generatore piazzerà aerei statici russi se quella particolare base sarà sovietica, o aerei statici della LW se sarà una base tedesca. E’ importante che abbiate compreso bene questo primo punto, prima di leggere i prossimi consigli.

Al fine di evitare di usare molti aerei flyable planes e per velocizzare la creazione di un template, possiamo usare lo stesso volo per determinare diversi punti. Come? Bene, il generatore cercherà coppie di waypoints, dal momento che sappiamo che una linea è definita da 2 punti. Quindi, quando create i punti, potrete usare lo stesso “volo” aggiungendo WP, come nell’immagine seguente:



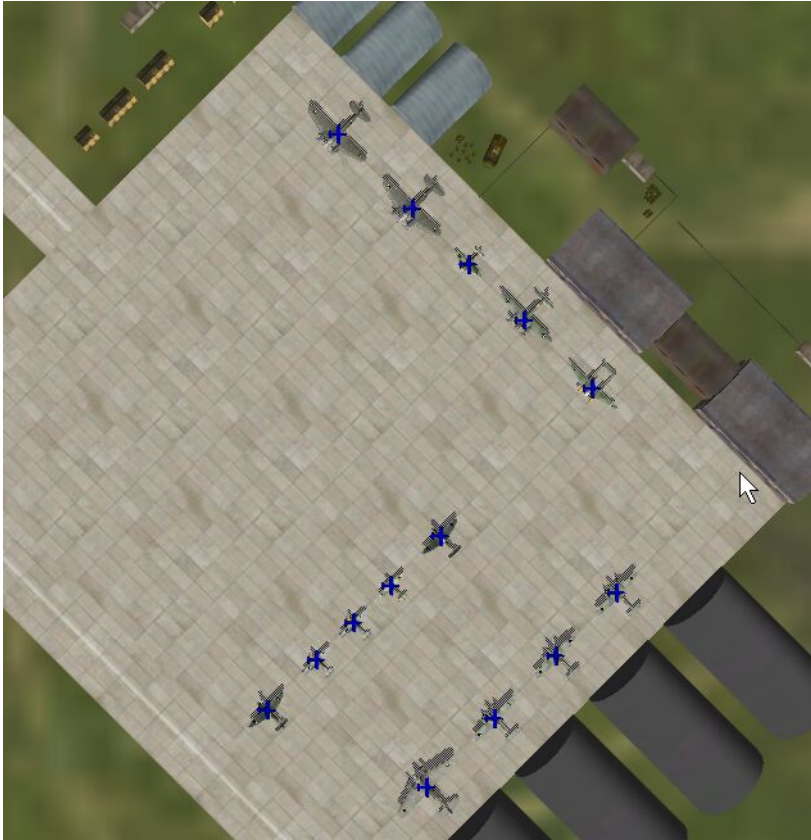
Questi waypoints indicano 3 luoghi: il posto di aerei dal WP0 al WP1, il posto per aerei dal WP2 al WP3, il posto per aerei dal WP4 al WP5. Quando “disegnate” questi waypoints dovrete considerare coppie di waypoints. Quando faccio questi template io dico a me stesso:

piazzando il WP0 penso: aerei da qui  
piazzando il WP1 penso: a qui  
piazzando il WP2 penso: aerei da qui  
piazzando il WP3 penso: a qui  
piazzando il WP4 penso: aerei da qui  
piazzando il WP5 penso: a qui

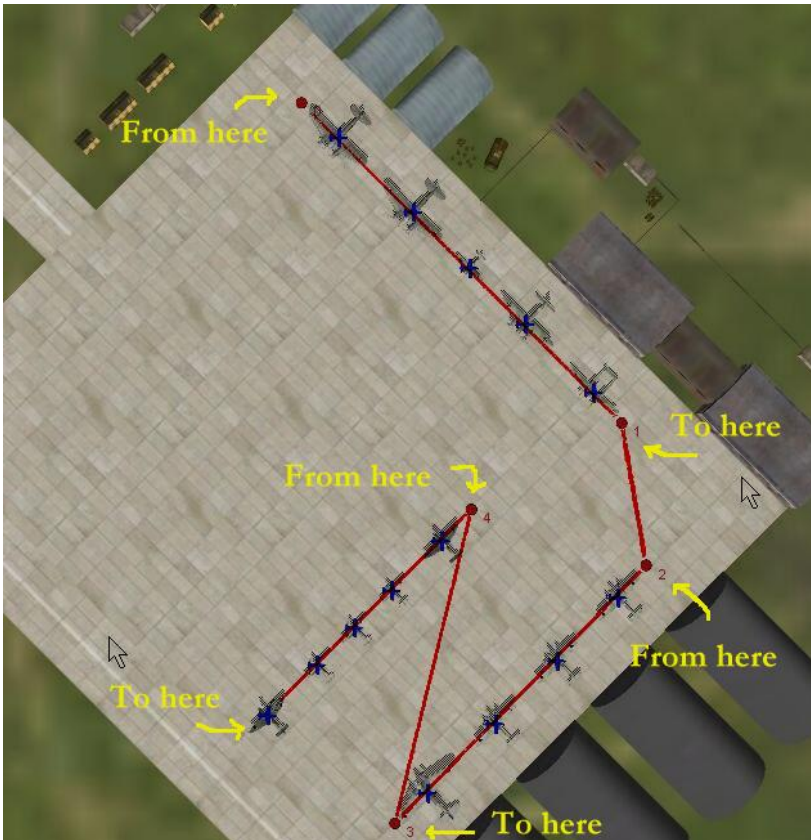
Come potete vedere, siete liberi di selezionare i punti di inizio fra i luoghi. In questo esempio, la linea fra il WP1 e il WP2 non definisce un luogo, è solo una linea di riposizionamento e potrete ignorarla. Lo stesso dicasi per la linea fra WP3 e WP4.

E' veramente facile, dopo che avrete provato alcune volte. La prossima immagine mostra come sono disposti gli aerei usando i luoghi definiti nell'immagine precedente:





E la prossima immagine è una combinazione di entrambe, mostrando quello che dovrete “immaginare” nel momento in cui “disegnate” il vostro template. Ricordate è MOLTO importante usare coppie di waypoints, perché definiscono le linee. Ricordatevi di usare la tecnica mnemonica “da qui...a qui”.



## 3. Creare i templates

---

Per creare un template per una mappa, è necessario creare svariati files. Potreste fare manualmente ciascuno di questi, ma ciò generalmente comporta la possibilità di errori, perché devono essere scritte una quantità di coordinate, ed un solo errore su una coordinata determinerà una missione sbagliata. Quindi ho pensato a fornire alcuni strumenti per creare templates in maniera semi-automatica.

Creeremo templates in diverse **fasi** (o passi), perché qualche fase necessita di dati dalla fase precedente. Ad esempio, i Geo\_obj.data contengono 4 gruppi principali di dati. Uno di questi gruppi sono le definizioni dei settori, e per scrivere le definizioni dei settori avrete bisogno di avere i waypoints dei carri armati. Ed i waypoints dei carri verranno decisi usando un template creato dopo aver terminato la “fase 1”, dove avremo definito gli aeroporti e i luoghi di parcheggio. Non preoccupiamoci ora di questo, la sola cosa importante per ora è sapere che ci sono diverse “fasi”

Questa creazione semi-automatica di template usa degli **scripts perl**. Così è necessario che abbiate installato **perl**. Se ne siete sprovvisti, non sarà possibile creare templates come descritto in questo manuale.

Il processo di creazione di template necessita che voi lavoriate con il FMB, e di quando in quando adoperiate qualche script, così per rendere le cose semplici, create una cartella all'interno di FB/missions. Generalmente rinomino la cartella con il nome del template della mappa che sto creando, così, ad esempio, per fare templates per la Normandia1 avrò la seguente cartella: **"D:\Ubi Soft\IL-2 Sturmovik Forgotten Battles\Missions\normandie1"** dove piazza tutti gli scripts perl e dove tengo i files template. Una volta finite li rinomino, aggiungendo un codice di 2 o 3 lettere prima di ogni file per un riconoscimento più facile.

Copia tutti gli scrips forniti con questo manuale nella cartella che preferisci.

### 3.1 Prima fase

---

Cominciamo a costruire templates per la mappa Normandie1. Come detto prima io metto tutti gli scripts in una nuova cartella che chiamo normandie. Quindi lancio IL2FB, e carico la mappa "normandie1". La prima cosa da fare è piazzare un oggetto "airfield champfire" nell'angolo il più possibile in alto a destra, meglio se leggermente fuori dell'ultima numero/lettera del settore (se ci riuscite, altrimenti piazzatelo il più possibile in alto a destra della mappa). Questo sarà usato dagli scripts per determinare i limiti della mappa. Salvate questo file come "initial.mis". Questo file ora fornisce dati importanti: i limiti della mappa ed il nome della mappa.



L'immagine Above mostra la localizzazione dell'oggetto "airfield champfire" per la mappa normandie1. Questa mappa ha in alto a destra le coordinate O-10, così l'oggetto "airfield champfire" è piazzato nel settore P-11. naturalmente, le coordinate P-11 non sono mostrate nel FMB. Questo quadrato in alto a destra avrà differenti coordinate su differenti mappe.

Questo nuovo file **initial.mis** conterrà i dati degli aeroporti, oltre ad altre informazioni di cui vi parlo. Per ciascun aeroporto della mappa che ci piacerebbe usare per la campagna, abbiamo bisogno di raccogliere alcune informazioni: le coordinate della località, le due direzioni ("headings") per decollo ed atterraggio, l'orientamento della pista, i "punti" dove ci piacerebbe avere aerei parcheggiati ed il punto per l'aereo statico che servirà per determinare la fazione dell'aeroporto (rosso o blu).

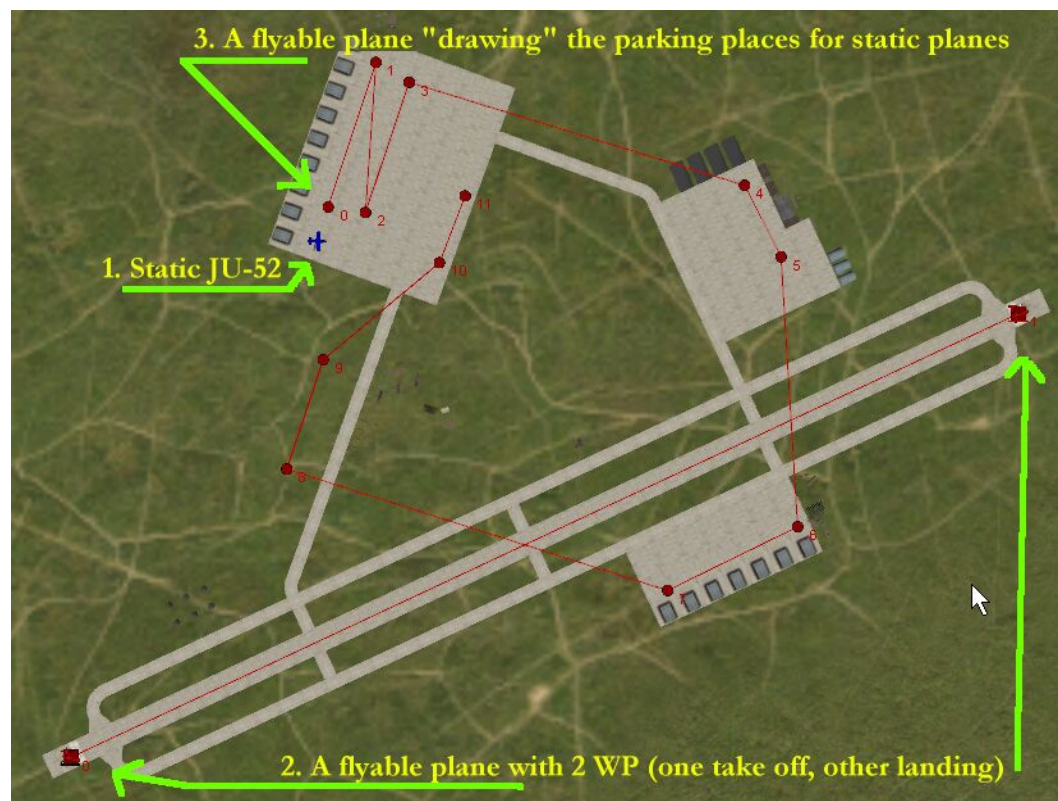
Avremo tutti questi dati aggiungendo 3 oggetti in ogni aeroporto:

il primo oggetto sarà uno Ju-52, da piazzare da qualche parte a nostro piacimento.

il secondo oggetto è un "volo" (qualsiasi aereo, io uso un B-17) con 2 way points: il waypoint 0 è un decollo (**takeoff**) e il waypoint 1 è un atterraggio (**landing**). Questo "volo" fornirà informazioni sulle coordinate dei punti di decollo e atterraggio, e anche sul punto centrale (center location) dell'aeroporto, come pure dell'orientamento (headings) della pista.

l'ultimo oggetto è un "volo", un **altro differente dal precedente** (sempre qualsiasi aereo, io uso sempre il B-17) che definisce i posti di parcheggio per gli aerei, come descritto nella sezione relative ai "Places templates" Potrete aggiungere fino a 9 posti, ovvero fino a 18 waypoints. Ricordatevi l'orientamento "a destra" e la tecnica mnemonica "da qui...a qui".

Una volta piazzati tutti e 3 questi oggetti, selezionate un altro aeroporto e ripetete il processo aggiungendo un altro Ju 52, un altro volo di B-17 con 2 WP (un take-off e un landing), e un'altro volo di B-17 per disegnare i parcheggi. La prossima immagine è uno screenshot con tutti questi 3 oggetti per un aeroporto.





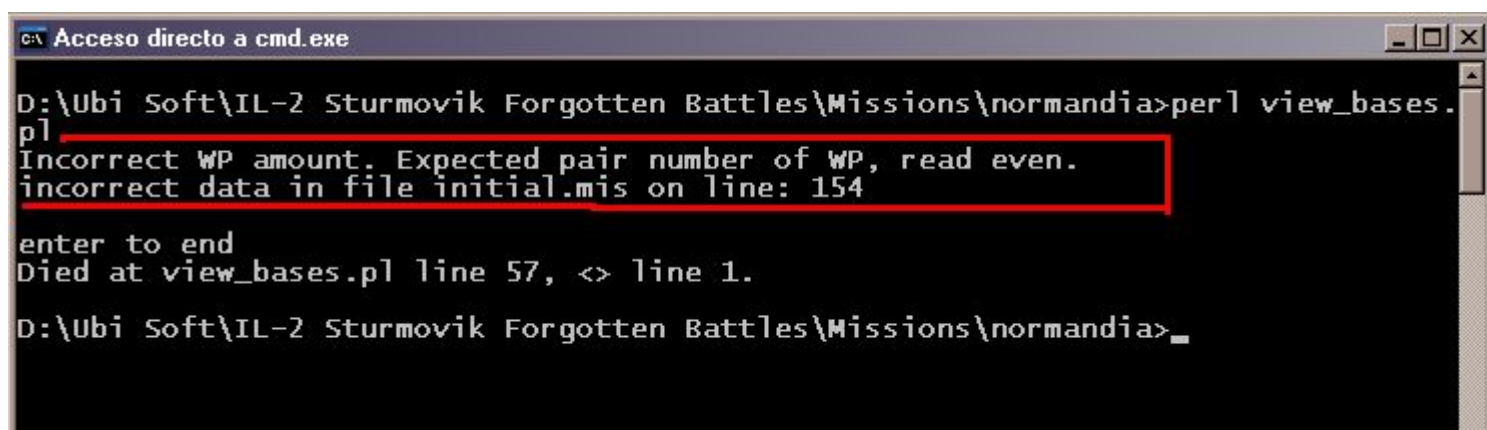
Volta per volta dovreste controllare come sono piazzati gli aerei, se l'orientamento è corretto, se avete predisposto correttamente le coppie di WP per il parcheggio degli aerei e così via. Per vedere una preview del vostro lavoro, lanciate lo script "view\_bases.pl". Questo script leggerà il vostro file "initial.mis" e creerà un nuovo file chiamato "preview\_bases.mis". Caricate questo file nel FMB e controllate come sono piazzati gli aerei. Questo script segnalerà un errore se avrete un numero dispari (non pari) di waypoints. La preview mostrerà solo aerei tedeschi, come se la base fosse sotto controllo Tedesco, in seguito il generatore piazzerà aerei LW o VVS a seconda della missione.

Ricordate di continuare a lavorare sul file "initial.mis". E' un errore frequente caricare una preview, e quindi continuare ad aggiungere Ju 52, posti di parcheggio e decollo/atterraggio sul file di preview. La missione preview è solo questo, una missione per dare uno sguardo se i vostri waypoints saranno pronti per il generatore, il file importante è "initial.mis". Quindi, dopo aver controllato con la missione preview, ricordatevi di caricare nuovamente il file "initial.mis" per continuare a lavorare su questo.

La prossima immagine mostra come potrete vedere una base in un file preview\_base.mis. Notate che lo Ju 52 statico non è visibile. Questo perché la preview mission mostra esclusivamente i posti di parcheggio degli aerei.



Questo è come appare un messaggio di errore quando provate a fare una preview e il numero totale dei waypoints è dispari (non pari). Quando ciò succede è necessario caricare "initial.mis" nel FMB e cercare dove risiede il problema. Un consiglio è quello di lanciare una preview ogni volta che finite di lavorare su un aeroporto. Così che se si verifica un problema di WP dispari, il problema sarà sempre nell'ultima base su cui avete lavorato. Per di più, il messaggio di errore mostrerà qual è il numero della linea dove risiede l'errore di WP dispari, così se lo credete potrete cancellare manualmente quella linea, aprendo il file "initial.mis" con un editor di testo. (non consigliato, è preferibile controllare i WP nel FMB).



```
C:\ Accesso diretto a cmd.exe

D:\Ubi Soft\IL-2 Sturmovik Forgotten Battles\Missions\normandia>perl view_bases.pl
Incorrect WP amount. Expected pair number of WP, read even.
incorrect data in file initial.mis on line: 154

enter to end
Died at view_bases.pl line 57, <> line 1.

D:\Ubi Soft\IL-2 Sturmovik Forgotten Battles\Missions\normandia>
```

Questo è tutto ciò di cui abbiamo bisogno per finire la prima fase, un file "initial.mis" con un oggetto champfire, e ciascuna base aerea con 3 oggetti. Una volta che avrete fatto pratica nel costruire questi aeroporti, ciascuno di essi non vi richiederà che un paio di minuti. Io lavoro in questo modo: come prima cosa posiziono il file champfire, quindi aggiungo uno Ju 52 so ogni aeroporto che intendo usare, in modo che forniscano un facile aiuto visuale per trovare le basi. In seguito aggiungo tutti i punti di decollo ed atterraggio usando un B-17 per ogni aeroporto. In questo modo, quando do' un'occhiata alla mappa nel suo insieme, devo vedere qualche cosa di blu (gli Ju 52) e qualche cosa di rosso (I WP del B-17). Infine aggiungo le linee di parcheggio per ogni base, lanciando le preview volta per volta.

Se guardate nel file initial.mis che fornisco come esempio potrete vedere le delimitazioni della linea del fronte. Non ne avete bisogno per ora, è qualcosa che aggiungeremo nelle prossime fasi, e spiegheremo più tardi.

Ora siamo pronti per lanciare lo script 1\_stage.pl (lanciarlo in una finestra cmd: perl -w 1\_stage.pl). Questo creerà 5 files:

1. **blue\_obj.mis**: uno Ju 52 per base, nei punti indicati dallo Ju 52. Questo file è pronto per l'uso.
2. **red\_obj.mis**: uno Li-2 per base, nei punti indicati dallo Ju 52. Questo file è pronto per l'uso.
3. **tank\_wp.mis**: un template iniziale per i carri armati, lavoreremo su questo nella prossima fase.
4. **city.mis**: un template vuoto dove aggiungeremo le città da attaccare e gli obiettivi nella prossima fase.
5. **geo\_obj.data**: è un contenuto parziale di questo file, dove ora sono contenute tutte le descrizioni degli aeroporti, linee di parcheggio, orientamenti, Danni ecc.

Se siete curiosi, potrete guardare nel file geo\_obj.data appena creato, potrete vedere svariate linee con le informazioni su ciascun aeroporto. Ad esempio queste sono le linee che rappresentano l'aeroporto mostrato nelle immagini precedenti. Le informazioni sono: informazioni generali sull'aeroporto (prima linea), coordinate dell'orientamento della pista (AF06:Hx), seguite da alcune linee con le coordinate dei punti di parcheggio e le angolazioni per il piazzamento degli aerei (AF06:Px)

```
AF06,aerodromo--D03,38788.61,29109.76,2,-D,03,2,0:2
AF06:H1,38048.61,28764.76,
AF06:H2,39530.43,29455.75,
AF06:P1,38451.23,29622.01,38526.13,29846.98,378,
AF06:P2,38509.59,29613.09,38578.11,29816.26,378,
AF06:P3,39102.59,29655.61,39159.29,29544.57,152,
AF06:P4,39185.28,29124.04,38982.10,29024.82,243,
AF06:P5,38386.74,29213.82,38443.44,29383.92,378,
```



AF06:P6,38625.36,29535.12,38665.52,29639.07,381,  
#--- End AF06 (aerodromo--D03)

Non preoccupatevi se non capite queste informazioni, non avete bisogno di comprenderle per continuare a lavorare sui templates.

## 3.2 Seconda Fase

Questa fase è quella che richiede più tempo, perchè bisogna lavorare sul template dei waypoints dei carri armati (tank) e su quello delle città. Dividerò questa fase in due parti, “A” e “B” una per ciascuno dei due template.

### **A- Template tanks WP**

Il template dei carri armati contiene informazioni sui “punti” dove il generatore in seguito piazzerà un campo di difesa - qualche oggetto statico, artiglieria contraerei AAA (Anti Aircraft Artillery), e carri armati stazionari – o un gruppo di carri in movimento (il gruppo di attacco). Ciascun settore necessita di ameno 2 punti, in modo che sia possibile piazzare entrambi i gruppi, uno in ciascun punto. Più punti sarebbe meglio, in tal caso il generatore in seguito potrà considerare differenti combinazioni di questi punti, rendendo le missioni meno repetitive. Quindi, per apportare qualche variazione sulla selezione dei punti, si consiglia di inserire un totale di questi per ogni settore da 3 fino ad un massimo di 5.

Ci sono settori dove i carri non possono muoversi, ad esempio mare, laghi, foreste ecc. In questi settori non piazzeremo nessun punto, tutto qui, non ci saranno “punti di partenza”, e tali settori saranno inattaccabili..

Andate nel FMB e caricate la missione "tank\_wp.mis" create nella prima fase. Dovreste vedere qualcosa simile a questo ( è mostrata solo una piccolo parte):

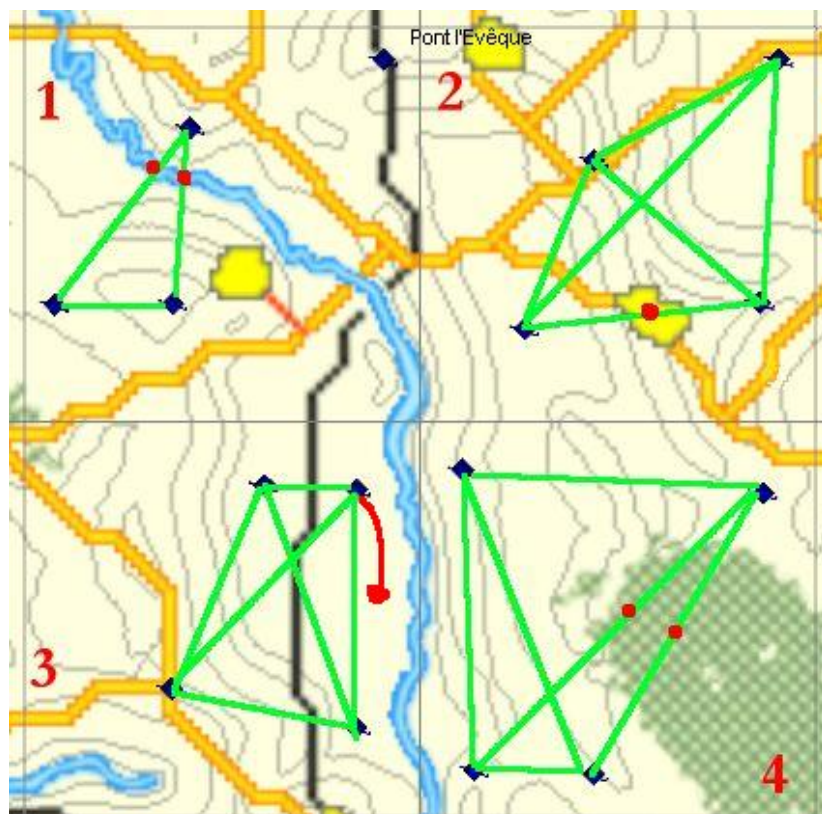


Potete vedere in tutti i settori 5 carri, uno a ciascun angolo ed uno al centro. Questo file è creato in questo modo per velocizzare il processo. E' infatti più semplice eliminare e muovere oggetti già piazzati in una missione. Ciascuno di questi carri statici che vedete rappresenta un "punto". Come per i punti di parcheggio usiamo i "voli" per poi in seguito piazzare degli oggetti (gli aerei), in questa fase usiamo carri stazionari per indicare le coordinate dove in seguito verranno piazzati un accampamento, o un gruppo di carri in movimento. Nell'immagine ho numerato i settori da 1 a 4, solo come facile riferimento per questo manuale.

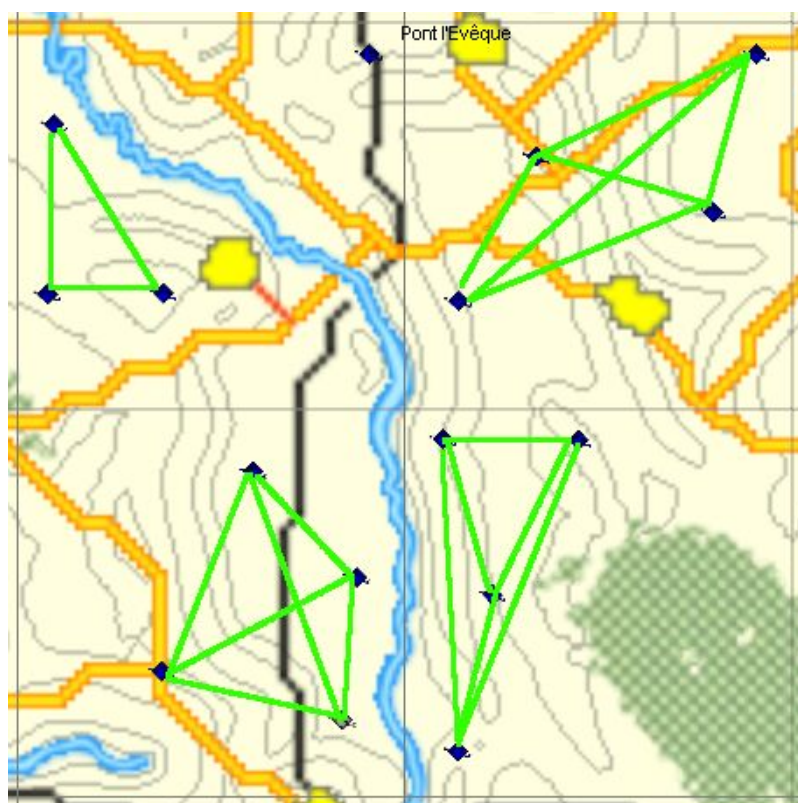
L'immagine precedente mostra 4 settori. La prima cosa da fare è eliminare tutti e 5 i punti dai settori non attaccabili (mare, lagune e i settori con ricoperti da forsete). Quindi bisogna ricollocare i punti seguendo semplici regole. I punti (in effetti i carri, in questo template) non devono trovarsi troppo vicini. Tutti devono trovarsi all'interno del settore, e fiumi, laghi, città o foreste non devono attraversare il loro cammino. Se non è possibile piazzarli tutti e 5, provate ad eliminarne qualcuno per cercare di trovare più facilmente dei posti corretti. La prossima immagine mostra come editare questi punti. Il settore 1 non è attaccabile (mare), ed ho eliminato tutti e 5 i punti. Il settore 2 ha 4 punti, il settore 3 ne ha 5 ed il settore 4 solo 3:



Come potete vedere, tutti i punti rispettano i requisiti. Ora mostrerò invece un'immagine con un piazzamento incorretto dei punti. Aggiungerò qualche linea per congiungerli, e piazzerò un punto rosso sopra le linee verdi per evidenziare le inesattezze. Dovrete "visualizzare" nella vostra mente queste linee quando muovete i punti.



Nel settore 1 c'è un punto piazzato dall'altra parte di un fiume. Nel 2 il percorso fra due punti passa attraverso una città. Il piazzamento nel settore 3 è corretto, ma due punti sono troppo vicini, e sarebbe meglio posizionarne uno leggermente più distante, come mostrato dalla linea rossa. Il settore 4 ha due punti piazzati in una foresta. Nell'immagine successiva mostro la disposizione corretta di questi punti:





Dovrete controllare la disposizione dei punti settore per settore. Assicuratevi che ogni settore abbia almeno 2 punti, ma è consigliabile averne un minimo di 3. E assicuratevi anche di non piazzare punti al di fuori del settore. Per rendere il lavoro più comodo, usate lo zoom-in e -out in modo che le linee della griglia mostrino quadrati di 10x10. E' anche consigliabile, per zone con colline e montagne, controllare esattamente il posto in cui piatterete un punto, per evitare di sgegliere magari il precipizio di una montagna....

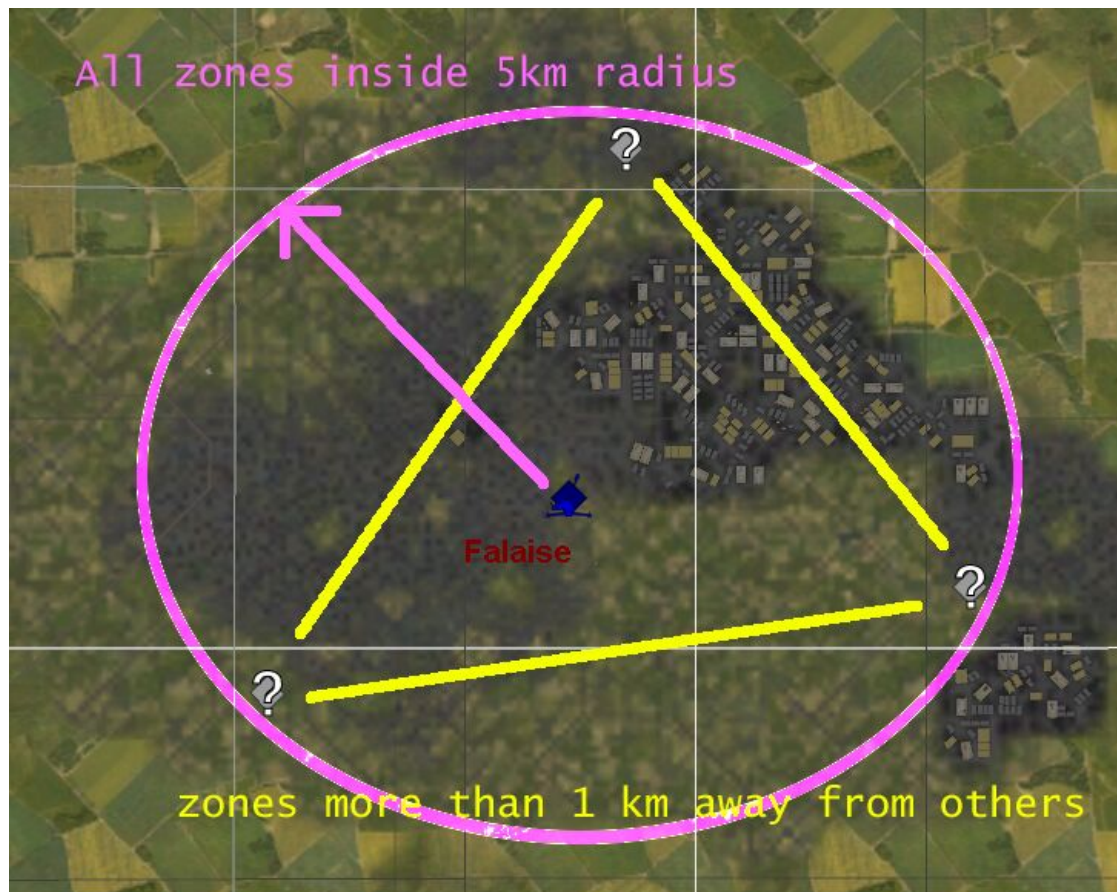
Questo template prende un po' di tempo per la realizzazione, ma quando avrete preso la mano, potrete costruire diversi settori in un minuto. La cosa più importante è rendere non attaccabili quei settori che non possono contenere i "punti" per i carri armati. Tutti gli altri settori possono essere editati o corretti in seguito (problemi con una collina, o punti su diverse sponde del fiume ecc.)

## **B- Template per le città**

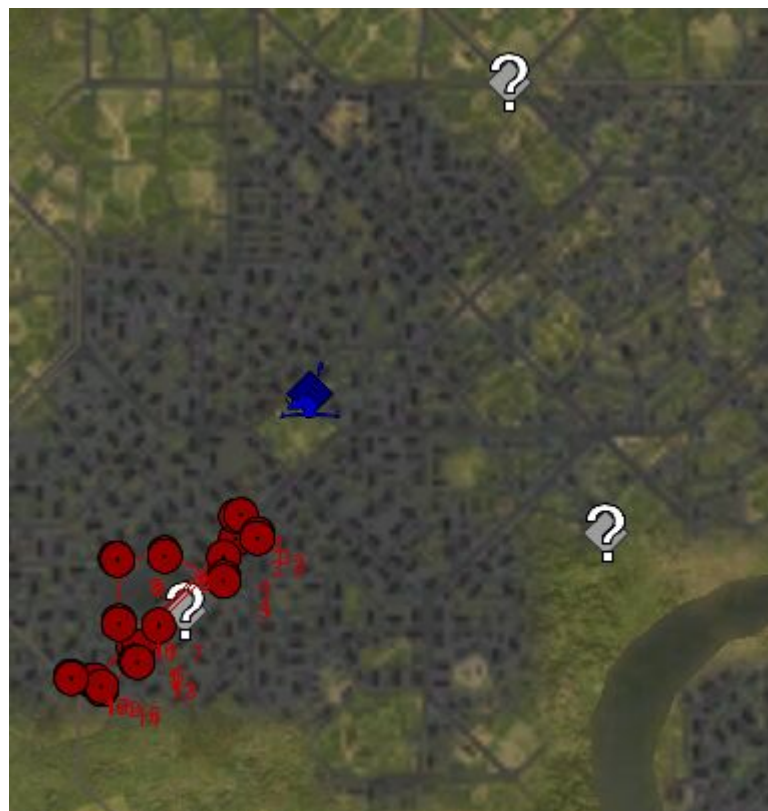
Ora iniziamo con il template per le città. Questo tipo di template contiene le città che vogliamo includere come obiettivi. Ogni settore conterrà solo una città. Le città non possono essere più vicine di 10km l'una dall'altra. Caricate il city.mis nel FMB. La prima cosa da fare è piazzare un oggetto "observation ballon 90m" (Pallone di osservazione da 90m) sopra ogni città che vogliamo includere. In questo modo sarà facile controllare che non piatteremo 2 città troppo vicine. La prossima città mostra come sarà il template dopo aver aggiunto alcune città. Le linee sono state disegnate solo per evidenziare le distanze da controllare



Dopo aver finito, zoomate vicino a ciascuna città ed aggiungete alcuni oggetti "champfires". Ogni oggetto "champfire" rappresenterà una zona attaccabile, predisporne almeno 1, fino ad un massimo di 9 per ciascuna città. Tutti i posti segnati con un "fire" (zone) devono essere vicini alla città, non piazzateli mai ad una distanza superiore a 5 km. Inoltre, ogni zona "bersaglio" deve essere separata di almeno 1 km o più da ogni altra zona bersaglio. In questo modo, lascerete spazio per il posizionamento degli oggetti senza il rischio di sovrapporli. L'immagine successiva mostra una città con 3 zone (3 fires), tutte all'interno del limite di 5 km, e tutte separate fra loro di almeno 1 km.



Piazzate un numero maggiore di zone nelle città più importanti (5 o 6 può andar bene) ed un numero minore nelle città più piccolo (2 o 3). Una volta finito di piazzare i segnalatori (le “fire zones”), dovrete piazzare gli oggetti su ciascuno di essi. L’immagine sotto mostra una zona completata con oggetti e le rimanenti zone ancora vuote.





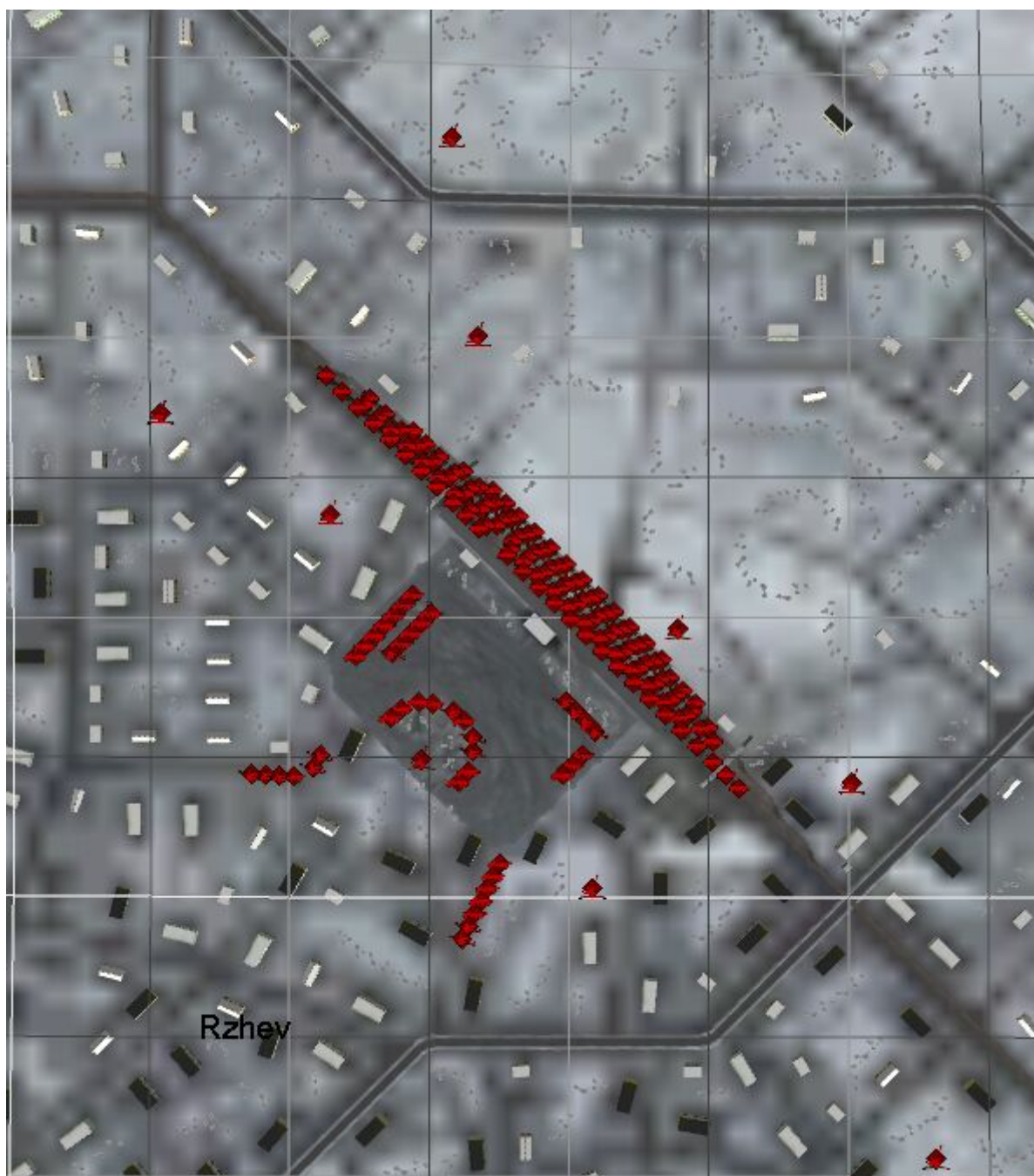
Il metodo per l'inserimento di oggetti è molto simile a quello descritto per i "places templates". In effetti è lo stesso, con qualche opzione in più. Dovrete disegnare coppie di waypoints. Queste coppie indicheranno le linee dove saranno piazzati gli oggetti. Ma in questo caso abbiamo in più un'altra possibilità. Usando l'**altitudine** del WP potremo indicare differenti tipologie di oggetti. Pensate ai WP come ad una matita, e all'altitudine come al tipo di matita. In questo modo, con differenti altitudini, potremo "disegnare" oggetti differenti.

- Wp a 500 metri sono veicoli paralleli alla linea.
- Wp a 1000 metri sono veicoli ruotati a destra della linea.
- Wp a 1500 metri sono treni paralleli alla linea, con la locomotiva piazzata sul primo WP, e l'ultimo vagone piazzato nel WP successivo.
- Wp a 2000 metri sono AAA (artiglieria contraerei), una solo AAA ogni coppia di WP.

Ancora, è indifferente il tipo di aereo usato per tracciare i punti. Ma ora è molto importante usare la corretta altitudine per rappresentare gli oggetti desiderati. Normalmente uso un aereo Tedesco (WP blu) per tracciare i punti per le AAA, e uno sovietico (rosso) per i posti dove saranno piazzati i veicoli. E generalmente per ogni zona uso "voli" differenti per bersagli differenti. Ad esempio, con un B-17 traccio tutti i veicoli paralleli alla linea (altitudine del WP 500 m). Io piazzo circa dieci fino a 20 WP, così da predisporre da 5 a 10 posti. Quindi inserisco un nuovo "volo" di B-17, seleziono l'altitudine a 1000m e piazzo alcuni veicoli ruotati verso destra. Se c'è una linea ferroviaria, aggiungo un ulteriore volo di B-17, seleziono l'altitudine a 1500m e aggiungo tutti i treni. Alla fine inserisco un volo blu (qualsiasi tipo, in genere io uso lo Ju 87) e aggiungo vari punti per le AAA. L'immagine seguente mostra come apparirà un'area con i bersagli.

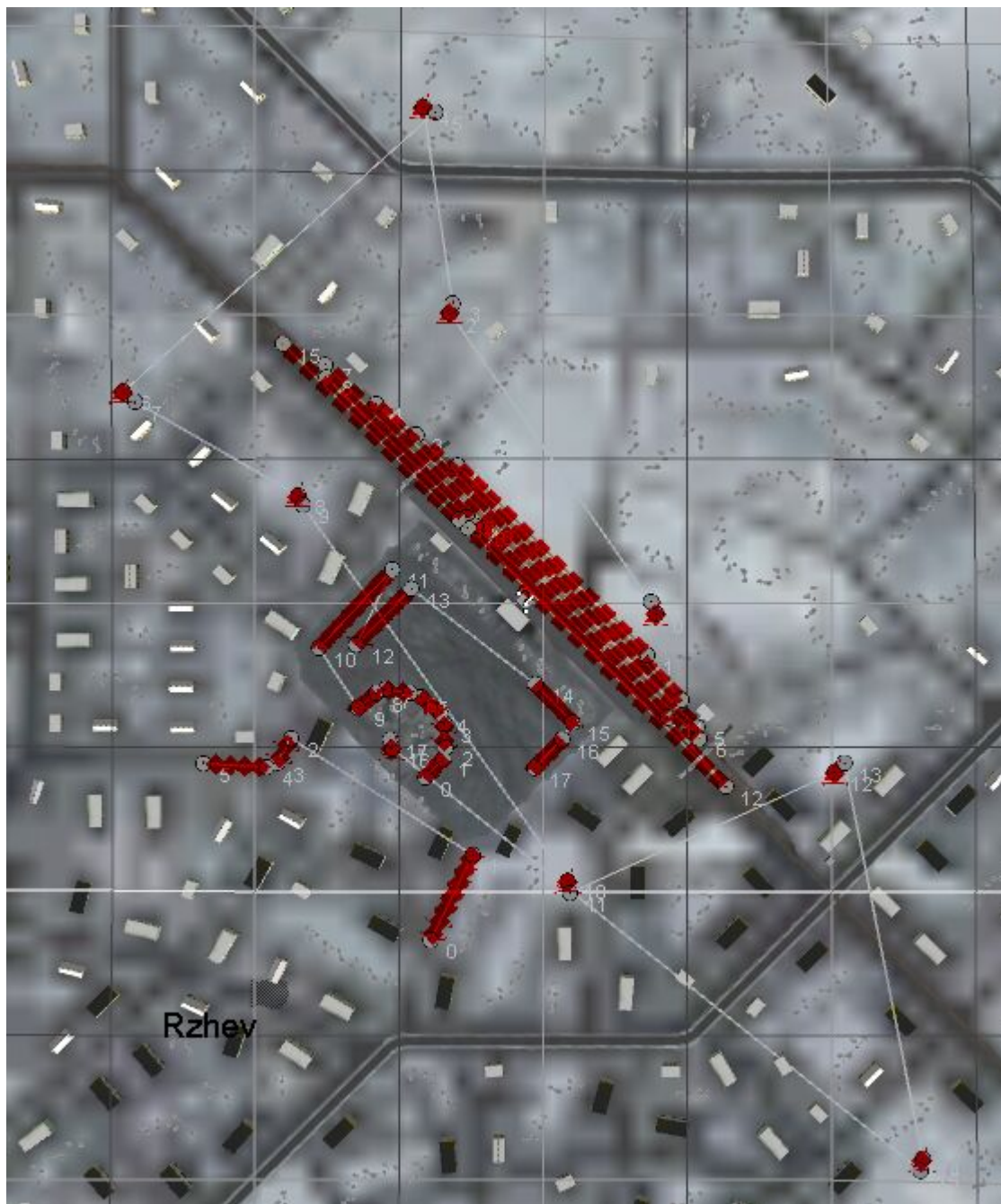


Come in precedenza, avete uno script per controllare in che modo il generatore leggerà i vostri bersagli. Questo script è chiamato "view\_city.pl" e leggerà il file city.mis (su cui state lavorando) creando un nuovo file "preview\_city.mis". Dopo aver aggiunto una o più zone, lanciate lo script e quindi caricate nel FMB la missione preview. Controllate la disposizione dei bersagli, se tutto ciò che avevate pianificato di aggiungere è al posto giusto ecc. In seguito tornate indietro e caricate ancora il file "city.mis". Come prima, non fate l'errore di continuare ad aggiungere zone nella missione di preview. Questo script ugualmente vi segnalerà la presenza di problemi dovuti a waypoints in numero dispari. La prossima immagine mostra la preview della stessa zona dell'esempio precedente.



E l'immagine seguente mostra la combinazione delle due immagini precedenti, dove potrete vedere gli oggetti piazzati sulle linee che avete tracciato. Questa è l'immagine che dovrete avere in mente quando aggiungerete i punti. Notate che tutti i bersagli sulla preview sono oggetti rossi, ma in seguito il generatore piazzerà correttamente gli oggetti dei due eserciti secondo quale sarà la fazione che controllerà la città.





Ok, ora siamo pronti per lanciare lo script della seconda fase 2\_stage.pl. Questo script leggerà entrambi i templates (tanks\_wp e city) e aggiungerà informazioni al file geo\_obj.data. Le informazioni aggiunte sono: informazioni sul settore, come queste linee:

```
SEC-E08,sector--E08,45000,75000,30,1:0
SEC-E09,sector--E09,45000,85000,30,1:3
etc...
```

informazioni sulle città, come queste linee:

```
# Name F08
SUC02,SUM-city-F08,53945.30,73549.92,-,-,-,-:0
CT02,city-F08,53945.30,73549.92,25,tip0,TOTAL_ZONES_HERE,0,25:0
CT02Z0:54427.62,74109.38,1
CT02Z1:55014.97,73130.47,1
CT02Z2:54036.05,72934.69,1
```

relazioni fra settori e città, come queste linee:

```
poblado,city-H07,sector--H07
poblado,city-F08,sector--F08
poblado,city-I04,sector--I04
```

### 3.3 Terza fase

---

Questa ultima fase è la personalizzazione dei vostri templates. Il file `geo_obj.data` è quasi completa ma necessita di essere personalizzato. Aprite il file con un editor di testo. Noterete che non ci sono nomi per le città, dal momento che dovrete essere voi ad aggiungerli manualmente. Per ciascuna città-XXX trovate il nome nel FMB e sostituitelo. Vi consiglio di usare il comando `cerca e sostituisci tutto`, per evitare errori nel riscrivere più volte i nomi. Infatti non è poi importante che il nome di una città sia sbagliato, ma è importante che sia sbagliato in tutti i posti.

Ad esempio, la città sulla mappa `normadie1`, al settore-F08 è **Bayeux**, così dovremo apportare questi cambiamenti al file `geo_obj.data`:

Inizialmente le linee saranno così:

```
# Name  F08
SUC02,SUM-city-F08,53945.30,73549.92,-,-,-,-,-:0
CT02,city-F08,53945.30,73549.92,25,tip0,TOTAL_ZONES_HERE,0,25:0
CT02Z0:54427.62,74109.38,1
CT02Z1:55014.97,73130.47,1
CT02Z2:54036.05,72934.69,1
.
.(other lines)
.
poblado,city-F08,sector--F08
.
.
```

Ora saranno invece così:

```
# Bayeux  F08
SUC02,SUM-Bayeux,53945.30,73549.92,-,-,-,-,-:0
CT02,Bayeux,53945.30,73549.92,40,tip0,3,0,40:0
CT02Z0:54427.62,74109.38,1
CT02Z1:55014.97,73130.47,1
CT02Z2:54036.05,72934.69,1
.
.
.
poblado,Bayeux,sector--F08
.
.
```

Notate che ho cambiato il testo `"TOTAL_ZONES_HERE"` con il totale delle zone, in questo caso 3. Inoltre ho voluto settare per Bayeux un raggio Massimo di rifornimenti di 40 km invece dei normali 25 km di default. Così

ho cambiato entrambi i valori “25” in “40”. Questi valori rappresentano il raggio massimo di rifornimenti e il raggio attuale (a causa di Danni è 0 entrambi i valori sono uguali).

Una volta terminato di aggiungere tutti i nomi e predisposto il numero corretto di zone e il raggio di rifornimenti iniziali, dovreste definire quale sarà la linea del fronte iniziale della campagna e far sì che tutti gli oggetti nel file geo\_obj.data siano “sincronizzati” con la linea del fronte.

Per fare ciò, caricate "initial.mis" e piazzate qualche marcatore (marker) del fronte blu/rosso. Salvate questo file e lanciate finalmente lo script della 3a fase “3\_stage.pl”. Questo script leggerà la linea del fronte che avrete appena immesso nel file initial.mis e creerà il template frontline.mis. Inoltre sincronizzerà il file geo\_obj.data file. Ora abbiamo quasi finito.

## 4. Aerei

---

Il file aircraft.data ha le definizioni di ciascun aereo che siete interessati ad inserire nella campagna. Inoltre questo file contiene informazioni sui nomi delle unità aeree impiegate e sugli aerei e veicoli stazionari. Le linee che iniziano con (#) vengono ignorate.

Non c'è un sistema automatico per creare questo file, così aprite un nuovo file con un editor di testo e createlo seguendo le indicazioni successive, quindi salvatelo denominandolo “aircrafts.data.”

Le prossime linee indicano che lo Squadrone dei caccia sovietici potrebbe essere o il 176IAP o il 3IAK. La selezione sarà casuale. I compiti di cacciabombardieri saranno appannaggio dello Squadron 11IAP, mentre per il trasporto e il bombardamento ci sarà lo Squadron 34BAP. Per i caccia tedeschi la scelta sarà fra i Gruppi I\_JG3, II\_JG3, I\_JG27, i bombardieri saranno dello Stab\_KG27a, o dei Gruppi I\_KG27a, II\_KG27a, III\_KG53a, e così via. Altre linee simili a queste descriveranno differenti unità per altre combinazioni di paesi e/o compiti.

Quando c'è più di una definizione, ne verrà scelta una casualmente. Per ogni compito/missione (caccia, bombardamento, cacciabombardieri, trasporti ecc) e per ogni fazione (rossi/blu), deve essere indicata almeno una definizione di unità (gruppo, stormo, squadrone ecc.).

```
##--red sqds
rusfig=176IAP,3IAK;
rusjab=11IAP;
rusbom=34BAP;
rustrp=34BAP;

gerfig=I_JG3,II_JG3,I_JG27;
gerjab=I_LG2;
gerbom=Stab_KG27a,I_KG27a,II_KG27a,III_KG53a;
gertrp=KGrzbV102a;

romfig=Eskadrila54;
romjab=Eskadrila54;
rombom=Eskadrila54;
```

I nomi delle unità possono essere quelle che preferite, **ma devono essere esattamente le stesse usate da FB**. Se pensate di aggiungere nomi di squadrons, assicuratevi che siano disponibili fra quelli di FB. Più avanti, questo file contiene la descrizione degli aerei con campi separati da virgole, con la missione e il numero separati dai due punti. I campi sono in ordine:



army, country+task, name of plane, class air, loadout, fuel, crusier alt, crusier speed :  
task, number,

**army:** 1 per gli alleati 2 per l'asse. Il campo **country+task** (romfig, gerjab, rusfig, hunfig, itajab, etc..) è una combinazione 3 lettere per il paese + 3 lettere per il ruolo. Più tardi si darà ad uno squadrone un nome come III\_JG52 or 84IAP ecc.. Notate, attualmente solo 3 nazioni sono supportate (rus,ger and rom), questo sarà ampliato su badc\_gen\_1.pl anche per le altre nazioni. Gli altri campi si spiegano da soli.

I compiti/ruoli (task) sono:

**BA :** Bomber Attack. Un gruppo di bombardieri che attaccano una città o un aeroporto

**SUM:** SUply Mission. Un gruppo di aerei da trasporto che svolgono missione di rifornimento

**BD:** Bomber Defend. Gruppo di bombardieri da difesa, che attaccano carri nemici che avanzano

**EBA:** Escort BA bombers: caccia di scorta a bombardieri in attacco BA

**ESU:** Escort SUply transports: caccia di scorta a trasporti in missione di rifornimento SU

**EBD:** Escort BD bombers: caccia di scorta a bombardieri in missione di difesa BD

**ET:** Escort Tanks: nel caso di attacco ad un settore (con truppe di terra) i caccia forniranno copertura ai carri in attacco, intercettando i bombardieri nemici BD

**INT:** INTERcept: In generale, missioni di intercettazione di trasporti SU e bombardieri BA.

Il numero number è il totale degli aerei di quel tipo ed è uguale per ciascun inserimento (non obbligatorio) Ecco di seguito come appare un inserimento per un cacciabombardiere P-40:

(Notate che il ruolo INTERCETTAZIONE è una singola "I" e non INT come descritto sopra)

```
# P40-E
1,rusjab,P-40E,1,air.P_40E,500lb,100,3000,400:BD,6,
1,rusfig,P-40E,1,air.P_40E,500lb,100,3000,400:ET,6,
1,rusfig,P-40E,1,air.P_40E,default,100,3000,400:EBA,6,
1,rusfig,P-40E,1,air.P_40E,default,100,3000,400:EBD,6,
1,rusfig,P-40E,1,air.P_40E,default,100,3000,400:ER,6,
1,rusfig,P-40E,1,air.P_40E,default,100,3000,400:I,6,
```

Per avere i nomi class air, aprite il FMB e caricate una qualsiasi piccola mappa, ad esempio online1. Quindi inserite un aereo di cui volete conoscere la class air, diciamo un FW 190-A8, così predisponete un "volo" di quell'aereo con l'armamento di default. Quindi inserite un'altro FW 190-A8 e fornitelo di un carico di caduta che vi piacerebbe avere, ad esempio una bomba SC500. Salvate la missione con un nome qualsiasi, ad esempio airnames.mis (potrete aggiungere più voli di altre class air se ne avrete bisogno). Aprite la missione con un editor di testo, avrete qualche cosa simile a questo:

```
[MAIN]
MAP Net1Summer/load.ini
TIME 12.0
CloudType 0
CloudHeight 1000.0
army 1
playerNum 0
[Wing]
g0101
g0100
[g0101]
Planes 1
Skill 1
Class air.FW_190A8
Fuel 100
weapons default
[g0101_Way]
```

```

    NORMFLY 15421.69 25085.94 500.00 300.00
[g0100]
    Planes 1
    Skill 1
    Class air.FW_190A8
    Fuel 100
    weapons 1sc500
[g0100_Way]
    NORMFLY 15421.69 23440.96 500.00 300.00
[NStationary]
[Buildings]
[Bridge]
[House]

```

Qui potrete vedere che la class air è: air.FW\_190A8  
 L'armamento di default (default loadout) è: default  
 L'armamento di caduta (bombload loadout) è: 1sc500

Ok, facciamo l'inserimento di questo FW190-A8 usando default loadouts, e una bomba SC500 per ruoli bomber defense (BD), e predisponiamo 45/300 aerei. Prima scrivete una singola linea, per compiti di intercettazione:

```
2,gerfig,FW-190A8,1,air.FW_190A8,default,100,3000,400:I,45,
```

Questi valori rappresentano:

- **2** = la fazione Asse
- **gerfig** = 3 lettere per la nazione (Germania) + 3 lettere per il ruolo (fighter, caccia)
- **FW-190A8** = è il nome dell'aereo che apparirà sui briefings, reports e stats
- **1** = perchè è human flyable
- **air.FW\_190A8** = è la class air che avete trovato come spiegato sopra
- **default** = è la configurazione di armamento standard
- **100** = è il carico di carburante (100%)
- **3000** = è l'altitudine di crociera
- **400** = è la velocità di crociera
- **:** = è il separatore
- **I** = compito intercettazione (NOTA: è una singola lettera "I", non 3 lettere "INT")
- **45** = è il numero totale degli aerei
- **,** = virgola finale

Una volta fatta questa linea, con il copia e incolla create altre linee (almeno 4 in più) per avere questo:

```

2,gerfig,FW-190A8,1,air.FW_190A8,default,100,3000,400:I,45,
2,gerfig,FW-190A8,1,air.FW_190A8,default,100,3000,400:I,45,
2,gerfig,FW-190A8,1,air.FW_190A8,default,100,3000,400:I,45,
2,gerfig,FW-190A8,1,air.FW_190A8,default,100,3000,400:I,45,
2,gerfig,FW-190A8,1,air.FW_190A8,default,100,3000,400:I,45,

```

Infine fate i cambiamenti in modo da avere compiti/ruoli EBA EBD ET BD tasks, scegliete 1sc500 loadout per compiti ET e BD. Riducete l'altitudine di crociera a 1000 metri e cambiate il codice in gerjab per il

bombardamento difensivo (BD): generalmente all'inizio aggiungo una linea di commento, preceduta dal segno (#), con nomi ed altre informazioni:

```
# FW-190A8 45/300 planes as Pirx table for map bla-bla at year xxxx ...
2,gerfig,FW-190A8,1,air.FW_190A8,default,100,3000,400:I,45,
2,gerfig,FW-190A8,1,air.FW_190A8,default,100,3000,400:EBA,45,
2,gerfig,FW-190A8,1,air.FW_190A8,default,100,3000,400:EBD,45,
2,gerfig,FW-190A8,1,air.FW_190A8,lsc500,100,3000,400:ET,45,
2,gerjab,FW-190A8,1,air.FW_190A8,lsc500,100,1000,400:BD,45,
```

Se controllate altre configurazioni di armamento, saprete che i cannoni Mk108 dei FW190-A8 sono denominati "r22mk108", così forse vi piacerebbe aggiungerli per missioni di intercettazione. Basta solo cambiare default con r22mk108 sulla linea del compito intercettazione, in modo da farla apparire in questo modo:

```
2,gerfig,FW-190A8,1,air.FW_190A8,r22mk108,100,3000,400:I,45,
```

Ripetete questo metodo per gli altri aerei. Generalmente faccio una missione (airnames.mis) dove inserisco tutti gli aerei di cui avrò bisogno con varie configurazioni di armamento. Così in seguito apro la missione per leggere tutti i dati relative ad air.class e loadouts di cui ho bisogno. Nel vostro caso, forse avrete bisogno di P51, P38, P47, 109-G6, B25 etc etc... piazzateli tutti nel vostro airnames.mis usando il FMB.

Più avanti in questo file contiene informazioni sulla distribuzione degli aerei e dei veicoli stazionari. Le linee degli aerei parcheggiati appaiono così:

```
ST100,1,I153,vehicles.planes.Plane$I_153_M62,15:150
ST101,1,I16,vehicles.planes.Plane$I_16TYPE24,15:350
ST102,1,IL2,vehicles.planes.Plane$IL_2_1940Late,20:400
ST103,1,LaGG3,vehicles.planes.Plane$LAGG_3SERIES4,15:500
```

E i campi sono:

- STx+ 2 numeri: x=1 alleati e x=2 Asse (aerei statici)
- fazione (1 alleati, 2 asse)
- il nome dell'aereo (per i reports)
- il nome dell'oggetto in FB
- l'apertura alare approssimata dell'oggetto
- :il numero.

I numeri sono valori cumulativi, e l'ultimo aereo statico deve avere il numero 1000, devono essere in ordine, come nell'esempio sopra: abbiamo 150 I-153 per 1000 aerei, quindi abbiamo 200 (350-150) I-16type24, poi 50 (400-350) IL2 1940, quindi 100 (500-400) LaGG3 e così via. Notate che in mezzo agli aerei statici abbiamo qualche camio cisterna, che saranno piazzati come gli aerei a simulare il rifornimento. Infine la definizione dei veicoli statici. Questo è praticamente lo stesso che per gli aerei statici, usando valori cumulative fino a 1000. Ecco qualche esempio per veicoli dell'asse:

```
SV201,2,VW82,vehicles.stationary.Stationary$VW82,15:80
SV202,2,VW82t,vehicles.stationary.Stationary$VW82t,15:160
SV203,2,OpelKadett,vehicles.stationary.Stationary$OpelKadett,15:260
SV204,2,OpelBlitz36S,vehicles.stationary.Stationary$OpelBlitz36S,15:340
SV205,2,OpelBlitz6700A,vehicles.stationary.Stationary$OpelBlitz6700A,15:440
SV206,2,Kettenkrad,vehicles.stationary.Stationary$Kettenkrad,15:500
```

I campi sono:

- SVx+ 2 numeris: x=1 alleati e x=2 asse (veicoli statici)
- Fazione
- nome del veicolo (per i reports)
- il nome dell'oggetto in FB
- l'apertura alare approssimata dell'oggetto
- :il numero.

## 4. Immagine della mappa

---

L'immagine che deve essere usata è di tipo BMP 24bits/pix , che dovrete fare con il programma che preferite. Prendete diversi screenshots della mappa nel FMB. Quindi apritela e ridimensionatela per averne una di 900 pixels. Potreste usare altre dimensioni, ma non è consigliata una dimensione maggiore di 900 per le performance ed il carico del server. Più piccola potrebbe essere difficile distinguere le informazioni. Usate sempre una larghezza divisibile per 4, perchè gli scripts non gestiscono il riempimento. Aggiungete all'immagine della mappa i nomi delle città. Chiameremo questa immagine NM1.BMP. In seguito usate uno strumento per creare il file dell'immagine della mappa (lo chiameremo imap.data). GIMP ha un buon imagemap editor.

Quando costruite l'immagine, contate quanti pixel per settore avete. In seguito avrete bisogno di questa informazione per scriverla nel config.pl (spiegato dopo). Ad esempio, se la vostra immagine ha una larghezza di 900 pixel e la mappa ha 15 settori, il conteggio di pixel per settore è 60. Forse uno meno perchè le mappe hanno setori parziali alla fine, dove sono le coordinate.

Il file image map è solo un file html, che punta all'immagine (non a questa, punta di default a un "front.jpg" ) e quindi alla definizione della mappa, in questo modo:

```
<br>
<p> Use the map to find the name of the area of attack </p>
<IMG SRC="/images/front.jpg" WIDTH=xxx HEIGHT=xxx BORDER=0 alt="" USEMAP="#map">
<MAP NAME="map">
<AREA SHAPE="RECT" COORDS="0,31,46,77" HREF="/no_click.html" alt="sector--A16">
.
. (other imap lines)
.
</MAP>
```

Notate che dovrete definire larghezza ed altezza, usualmente questo è fatto dallo strumento imagemap.

## 6. Configurazione finale

---

Una volta che avete tutti questi file, è il momento di rinominarli, in modo da renderli unici. Basterà usare una combinazione di 2 o 3 lettere. Così, per la Normandia, li rinomineremo usando NM1:

```

initial.mis      -> NM1_initial.mis  (questo non sarà più usato, ma salvatelo in ogni caso)
geo_obj.data     -> NM1_geo_obj.data
red_obj.mis      -> NM1_red_obj.mis
blue_obj.mis     -> NM1_blue_obj.mis
tank_wp.mis      -> NM1_tank_wp.mis
city.mis         -> NM1_city.mis
frontline.mis    -> NM1_frontline.mis
aircrafts.data   -> NM1_aircrafts.data
NM1.BMP          -> NM1_000.BMP  (000 è solo per dire che è l'originale map image).
imap.data        -> NM1_imap.data

```

Ora dovete aggiungere le informazioni di questa mappa al file config.pl. Per prima cosa definite il nome della mappa (\$MAP\_NAME\_LONG), un breve testo con qualche semplice informazione sulla mappa. **After that several lines has to be included inside a if condition:**

1. l'informazione sull'immagine della mappa
2. i tipi di carri che devono essere usati in difesa ed in attacco da entrambi gli eserciti, e le AAA
3. i nomi dei file per questa mappa. 4) la lista degli aerei da usarsi su questa mappa (per la stampa del form takeslot.pl) Il MAP\_NAME\_LOAD (nel nostro esempio è Normandy/load2.ini) può essere estratto cercando in qualsiasi template appena creato. Apritelo con un editor di testo e cercate la linea che inizia con "MAP" , quindi copiate l'informazione che segue.

```

$MAP_NAME_LONG="Normandie. June 1944";

if ( $MAP_NAME_LONG eq "Normandie. June 1944" ) {

    # 1 . map image information
    $ANCHO=900;      # NM1_000.bmp image width in pixels
    $ALTO=600;       # NM1_000.bmp image height in pixels
    $H_BLOCK_PIX=60; # horizontal sector size in pixels
    $V_BLOCK_PIX=60; # vertical sector size in pixels
    $LETRAS=16;      # map sector letters, 1 in excess (Normandie1 last is 0, that is
15)
    $NUMEROS=11;     # map sector numbers, 1 in excess (Normandie1 last number is 10)

    # 2. tank types and aaa placement
    $ALLIED_TANKS_ATTK="Armor.3-BT7";
    $AXIS_TANKS_ATTK="Armor.3-PzIIIIG";
    $ALLIED_TANKS_DEF="vehicles.artillery.Artillery\$BT7"; # notice "escaped $"
    $AXIS_TANKS_DEF="vehicles.artillery.Artillery\$PzIIIIG"; # notice "escaped $"
    $AAA_IN_CHAMPS=0;      # place or not place AAA on field champs
    $LATE_AAA_IN_CHAMPS=0; # place or not place advanced AAA, like nimrod and M16

    # 3. file names
    $GEOGRAFIC_COORDINATES="NM1_geo_obj.data";
    $FRONT_LINE="NM1_frontline.mis";
    $RED_OBJ_FILE="NM1_red_obj.mis";
    $BLUE_OBJ_FILE="NM1_blue_obj.mis";
    $CITY_PLACES="NM1_city.mis";
    $TANKS_WP="NM1_tank_wp.mis";
    $FRONT_IMAGE="NM1_000.bmp";
    $IMAP_DATA="NM1_imap.data";
    $MAP_NAME_LOAD="Normandy/load2.ini";
    $FLIGHTS_DEF="MM1_aircrafts.data";

    # 4 . data for takeslot.pl planes names should be the exact same used in
    aircrafts.data)
    @VVS_SUM_PLANES=( "Li-2", "TB3-4M-34R" );
    @VVS_BA_PLANES=( "IL2 series1", "IL2 series2", "SB-2M100A", "SB2-M103", "SU-2", "TB3-
4M-34R" );

```



```

    @VVS_AI_PLANES=( "Li-2" , "IL-4-DB3B" , "PE-2 1940" , "R-10" , "SB-2M100A" , "SB2-M103" , "SU-2" );
    @LW_SUM_PLANES=( "HE-111H2" , "JU-52" );
    @LW_BA_PLANES=( "BF-110C4B" , "HE-111H2" , "JU-87B2" , "JU-88A4" );
    @LW_AI_PLANES=( "JU-52" , "BF-110C4B" , "JU-88A4" );
}

```

I valori in VVS\_SUM\_PLANES, VVS\_BA\_PLANES, VVS\_AI\_PLANES sono gli aeree da trasporto disponibili, gli aeree da bombardamento (in attacco) e la lista degli aerei AI. Lo stesso funziona per i campi LW\_\* con aerei dell'asse. Tutti questi valori vengono usati nel file take\_slot.pl per permettere la selezione degli aerei al momento della creazione della missione. I nomi degli aerei devono essere **esattamente** gli stessi come sono scritti nel file aircrafts.data. Quelli che ho stampato per gli esempi non sono per la Normandia, sono solo degli esempi.

Quando predisponete i tipi di carri per l'attacco (ATTACK), assicuratevi di includere in una variabile chiamata (TANK\_REGEX) una stringa per identificarli. Ad esempio se volete aggiungere carri di tipo KV1, dovrete editare questa linea:

```
$TANK_REGEX="Pz|T34|ValentineII|BT7|M4A2|T70"; # tank string identification use | as separator
```

che ora sarà:

```
$TANK_REGEX="Pz|T34|ValentineII|BT7|M4A2|T70|KV1"; # tank string identification use | as separator
```

Notate che i nomi dei carri per la difesa, quelli fermi, devono avere un \$ barrato, aggiungendo un backslash prima del simbolo \$. Ad esempio, il nome dell'oggetto FB per un carro fermo BT7 è:

**vehicles.artillery.Artillery\$BT7**

e voi dovrete aggiungere il backslash prima del simbolo \$,

così: **vehicles.artillery.Artillery\BT7.**

## 7. Commenti

---

Sicuramente i templates richiedono tempo, e molti file devono essere creati, scritti e modificati. Dopo che tutti i files sono stati fatti, dovrete testarli per controllare che siano corretti, ad esempio che siano usati tutti gli aerei, tutti gli obiettivi delle città siano presenti, le immagini della linea del fronte e dei rifornimenti delle città siano creati correttamente e così via.

Io realizzerò altri templates, quelli che userò nella bellum: Kursk, Smolenk e il Balaton. Non farò altri template oltre quelli menzionati prima. Potrete pormi domande e sicuramente fornirò aiuto a chiunque costruirà templates, quantomeno a coloro che proveranno di aver dedicato tempo per comprendere questo manuale (nonostante il mio cattivo inglese). Usate il forum Bellum per le domande ( <http://il2fb-bellum.com.ar/foro> ) e per chiedere aiuto per i problemi con i templates che state creando (possibilmente fornite i files su cui state lavorando).

Nota: so che piccole mappe di FB funzionano con badc, ma non so se il generatore/parser può gestire mappe più estese, quelle che hanno 2 lettere per settore (finlandia, crimea, ecc..)

E' tutto. :-)