

Manual para generar las plantillas para la Campaña Dinámica BADC

Traducido por E69_espiral

1. Vistazo a las Plantillas

Las Plantillas en B.A.D.C. no son un único archivo, son muchos, y están hechas de diferentes maneras.

geo_obj.data: Contiene la mayoría de datos importantes. Tiene 4 grupos de datos:

- 1) La localización de los aeródromos, su nivel de daño, los lugares de parking, a qué bando pertenece, nombre en código, etc.
- 2) Coordenadas del Sector, tipo (atacable o no), TTL, bando a que pertenece, etc.
- 3) Nombre de la ciudad, coordenadas, nivel de daño, radio de Suministros, zonas de almacenaje, bando, etc.
- 4) Una lista con la relación de ciudades y sectores, para saber qué sectores tiene una ciudad y que no será atacable hasta que la ciudad tenga un 50% de daño.

city.mis: Es un archivo de misión creado usando el FMB (Full Mission Builder). Es el archivo que lleva más tiempo hacer. Aquí se determina qué ciudades son parte de la Campaña y las zonas de almacenaje. Cada una de esas zonas es un lugar atacable. El archivo indica qué tipo de objetos se pondrán allí y dónde se pondrán. **NO** tiene los objetos. Los indicadores son una combinación de “way-points” de aviones a diferentes alturas para representar diferentes objetos (se explicará más tarde). Luego el generador pondrá esos objetos “debajo” de las líneas de los WP.

tanks_wp.mis: Es la plantilla de emplazamiento de los tanques. Cada sector de 10x10 Km. necesita tener 2 ó más “puntos”. Luego el generador utilizará 2 de esos “puntos” para colocar el Campamento y el grupo inicial de 9 tanques. Como vemos son necesarios, al menos, 2 “puntos”. Cuantos más “puntos” mejor, así el generador tendrá más combinaciones de “puntos” para usar, haciendo más impredecible dónde estará el emplazamiento del Campamento y de los tanques. Estos “puntos” son tanques estáticos, realmente no importa qué tipo de objeto sea, el generador buscará las coordenadas del objeto (los “puntos”). Los sectores en los que sea imposible colocar 2 emplazamientos de tanques, serán inatacables.

red_obj.mis: Una misión que contiene 1 Li-2 por aeródromo. Nada más.

blue_obj.mis: Lo mismo, pero con 1 Ju-52, y los números de los objetos comienzan con 10000.

frontline.mis: Este archivo contiene el estado de la línea del frente. Tiene una bandera por cuadrícula, excepto para los sectores inatacables.

aircrafts.data: Este archivo tiene la información de los aviones volables, tipo, tareas (escolta, interceptor,...) etc.

imap.data: Este archivo es una imagen HTML para ayudar a seleccionar el objetivo al crear una misión. Contiene los nombres de los sectores (y adicionalmente nombres de las ciudades y aeródromos) para ser mostrados como "pop-up" al colocar el puntero del ratón sobre él.

Para crear plantillas para un mapa han de ser hechos todos los archivos descritos arriba; cada archivo tiene unas condiciones particulares que cumplir. La primera impresión sobre las Plantillas es que son muy complicadas, pero no es así.

2. Descripción de las Plantillas de emplazamientos

En esta guía daré una explicación archivo por archivo y algunas herramientas para semi-automatizar la creación de Plantillas. Pero antes de comenzar a mirar cada archivo mejor explicaré la diferencia entre "static objects templates" (plantillas de objetos estáticos) y "places template" (plantilla de emplazamientos).

Static objects templates son archivos creados con el FMB y tienen todos los objetos del mapa. Usualmente contienen miles de camiones, aviones, AAA, trenes, etc. distribuidos por todo el mapa. Más tarde, cuando se

ataca un emplazamiento, el generador de la misión busca un grupo de objetos cerca del lugar atacado, y “copia” y “pega” esos objetos en la misión creada.

Este sistema tiene ventajas e inconvenientes. Son fáciles de generar, puede llevar varias horas hacerlas, pero más tarde son fáciles de usar por el generador con “copiar y pegar”. El inconveniente es que no se pueden cambiar tipos de objetos, se necesita una “plantilla roja” y una “plantilla azul”, los objetos están siempre en los mismos emplazamientos y tras varios ataques la gente puede conocer dónde está la AAA y dónde los almacenes (el mejor lugar para bombardear).

BADC utiliza 2 archivos “static objects templates”: red_obj.mis y blue_obj.mis. Sólo contienen aviones estáticos para ubicar las Bases, 1 por base, y sólo se usan para que las Bases muestren el color correcto sobre el mapa.

Places templates: Estas plantillas **no** tienen objetos que luego estén la misión. Tienen los “emplazamientos” donde nos gusta tener los objetos colocados. Luego el Generador insertará los objetos en el emplazamiento correcto dentro de la misión que estamos generando. Eso es todo.

Las plantillas de emplazamientos no son difíciles de hacer, sólo se necesitan algunas reglas sobre cómo son marcados estos emplazamientos para que luego el Generador entienda dónde y cómo los objetos tienen que ser insertados. BADC utiliza las plantillas de emplazamiento para aeródromos, way-points de tanques y objetivos de las ciudades. Veamos un ejemplo general, más tarde veremos las reglas detalladas de cada plantilla.

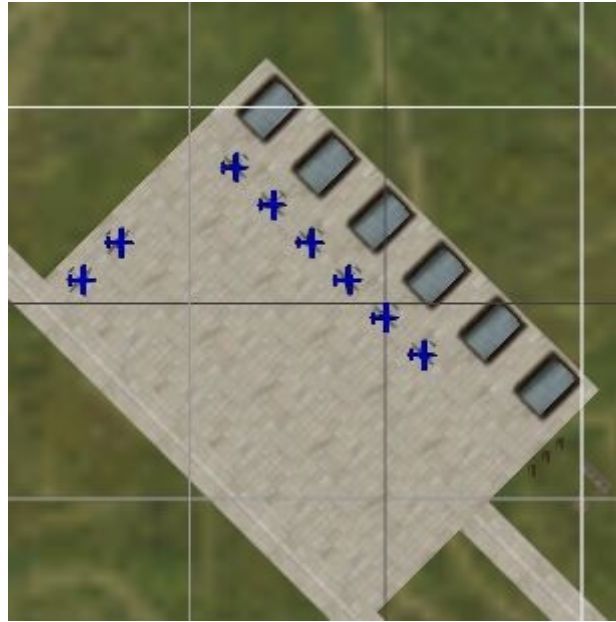
Digamos que estamos creando una “**static template**” (plantilla estática) para un aeródromo; emplazamos varios aviones y podemos ver algo así:



Después de que estos aviones sean seleccionados (tipo) y colocados tenemos que rotarlos uno por uno y alinearlos. Ahora, usemos el mismo aeródromo y la misma idea de la “plantilla de emplazamiento”. Queremos tener algunos aviones alineados en dos líneas, como se muestra en la siguiente imagen:

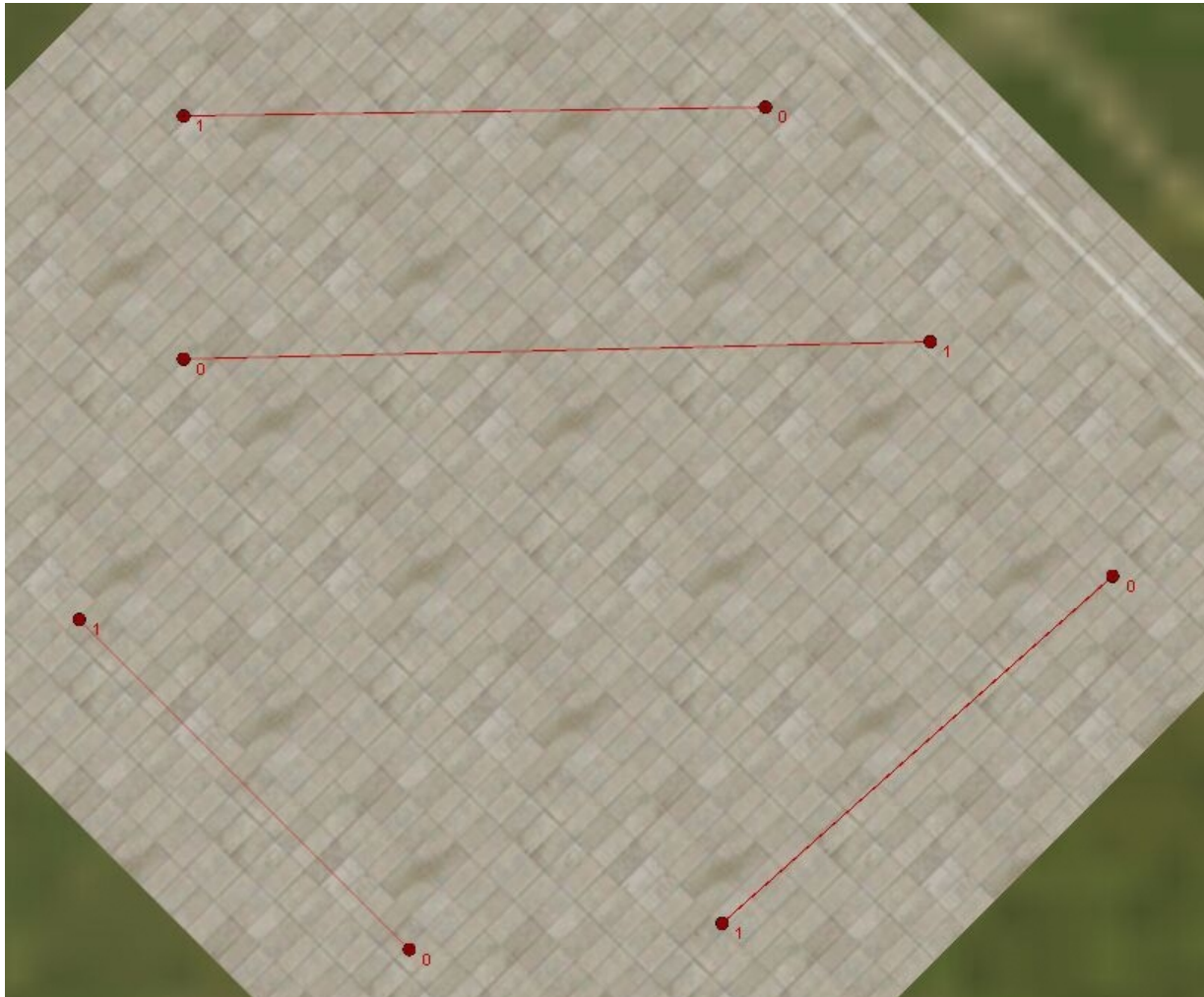


Luego el Generador insertará los aviones como en la siguiente imagen (están correctamente alineados y rotados, aunque pueden no verse bien debido al tamaño de la imagen):

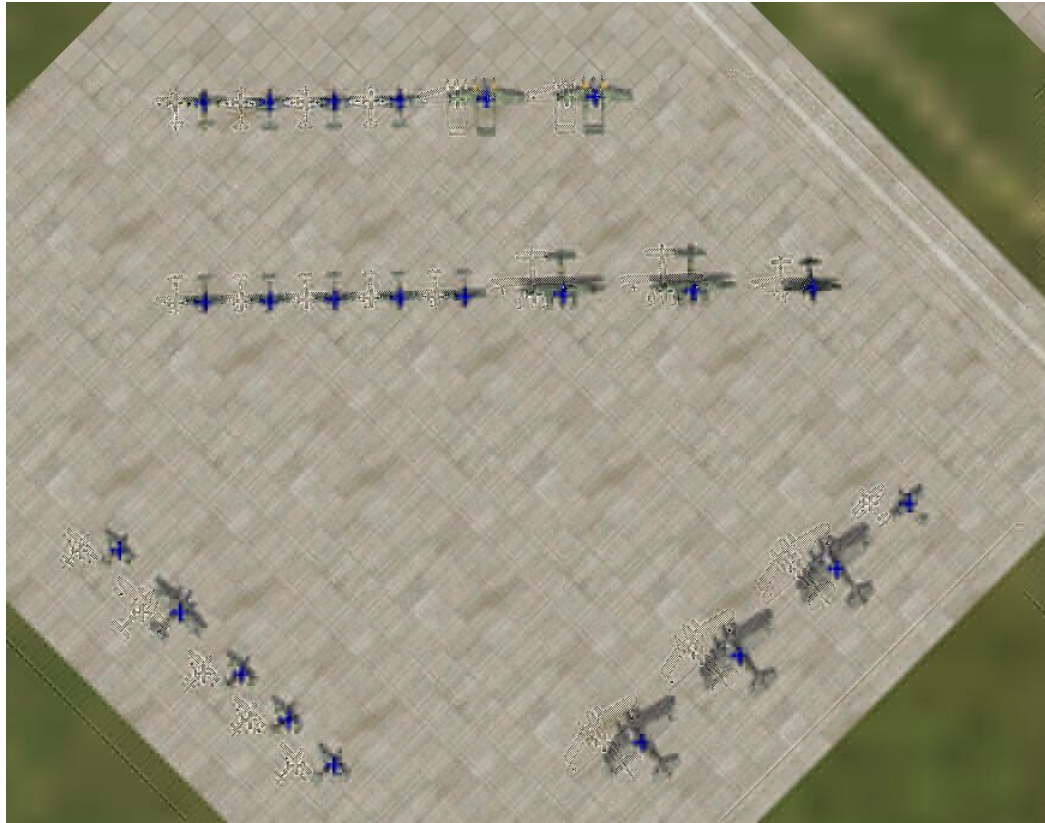


Así que, **¿cómo hacemos estas líneas para que más tarde el Generador coloque los objetos?** Usamos un avión (no pilotable) y establecemos WP; este es el modo en que “dibujamos” las líneas. Puede que no entiendas esto la primera vez que lo leas, pero es muy fácil. Cuando colocas un vuelo, el FMB escribe las coordenadas de los WP. Colocando WP y “dibujando” una línea con un avión haremos 2 cosas: decirnos dónde están las coordenadas y mostrarnos dónde serán colocados los objetos.

Veamos un ejemplo de trabajo; la siguiente imagen muestra varios WP dónde me gustaría colocar algunos aviones. Yo uso un B-17 para “dibujar” estos WP:



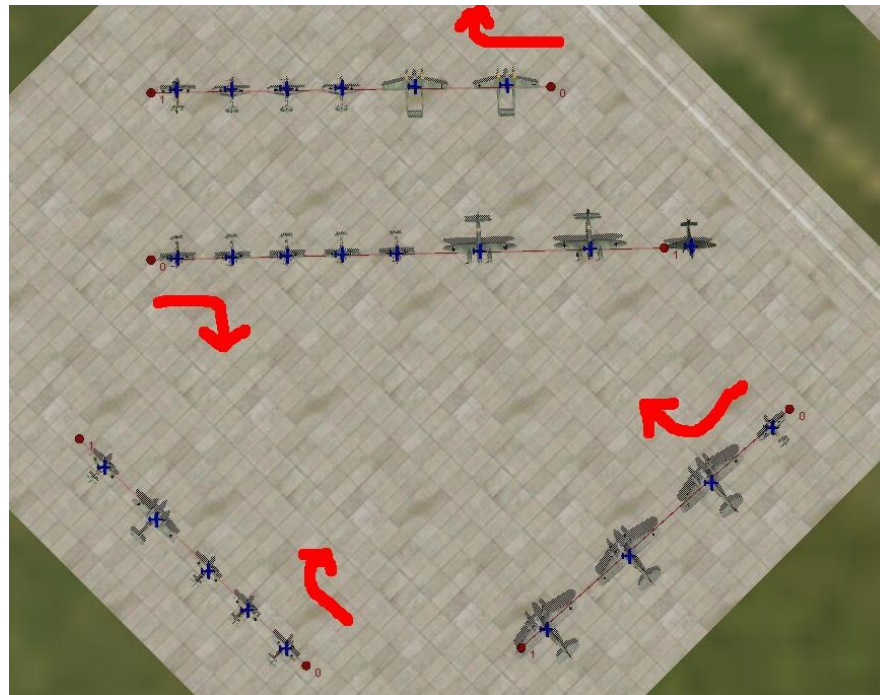
La siguiente imagen muestra cómo son colocados los aviones:



Compara estas dos últimas imágenes, mira la **dirección** de los WP y cómo son colocados los aviones. Notarás que los aviones siempre están colocados con el morro apuntando a la derecha de las líneas. Cuando digo

“derecha” significa: imagina que estás en el WP0 y miras hacia el WP1, los aviones estarán mirando a la “derecha” de esa dirección.

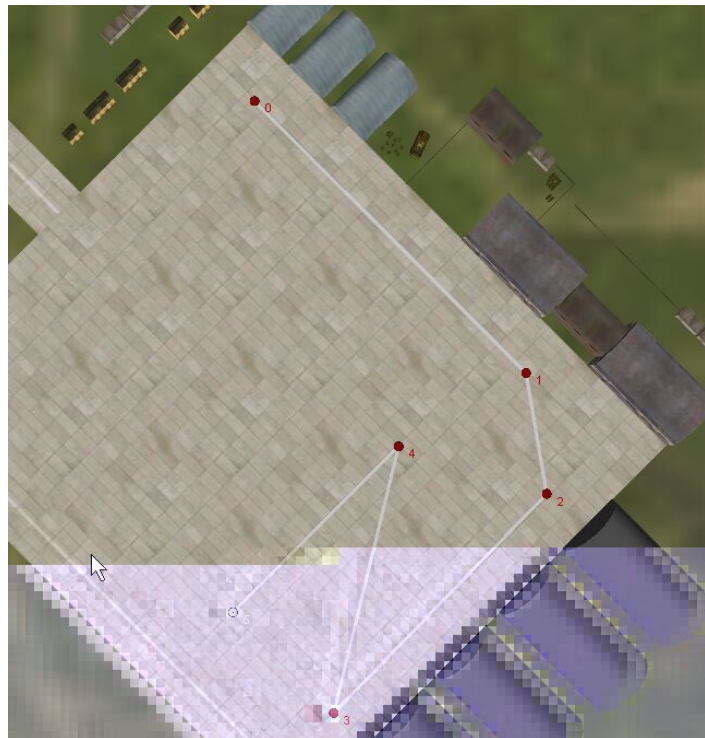
Para una más fácil comprensión combiné esas dos imágenes en una sola y dibujé una gran flecha roja en la “derecha” de esa dirección. Cuando hagas plantillas no verás los aviones, y tendrás que crear esta imagen en tu mente:



Ésta es la forma en que trabajan las **“place templates”**. Usa una misión para colocar vuelos, nunca coloques los objetos. Confía en que luego el Generador usará ese archivo con los WP del vuelo para leer las coordenadas

y colocar los objetos como deseas. Puedes utilizar cualquier avión para “dibujar” los emplazamientos, no importa el bando o el tipo de avión. El Generador colocará aviones estáticos rojos si la Base es soviética, o aviones estáticos azules si la Base es del bando alemán. Asegúrate de entender todo este punto antes de leer los siguientes.

Para poder usar muchos aviones pilotables y acelerar la creación de plantillas, usamos el mismo **vuelo** para hacer varios emplazamientos. ¿Cómo? Bien, el Generador buscará en **parejas** de WP porque sabemos que una línea está definida por 2 WP. Así que, cuando crees emplazamientos puedes hacerlo añadiendo puntos usando el mismo vuelo, como se muestra en la siguiente imagen:



Estos WP indican 3 emplazamientos: coloca aviones desde WP0 al WP1, coloca aviones desde el WP2 al WP3, coloca aviones desde el WP4 al WP5. Cuando “dibujes” estos WP tienes que pensar en parejas de WP. Me digo a mi mismo cuando hago estas plantillas:

Colocando WP0 pienso: aviones **desde aquí**

Colocando WP1 pienso: **hasta aquí**

Colocando WP2 pienso: aviones **desde aquí**

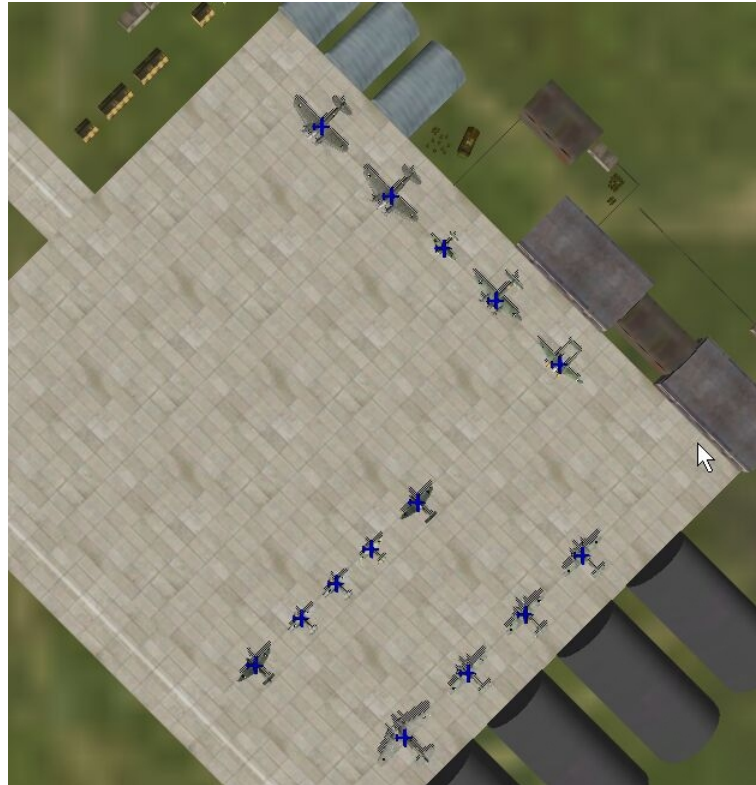
Colocando WP3 pienso: **hasta aquí**

Colocando WP4 pienso: aviones **desde aquí**

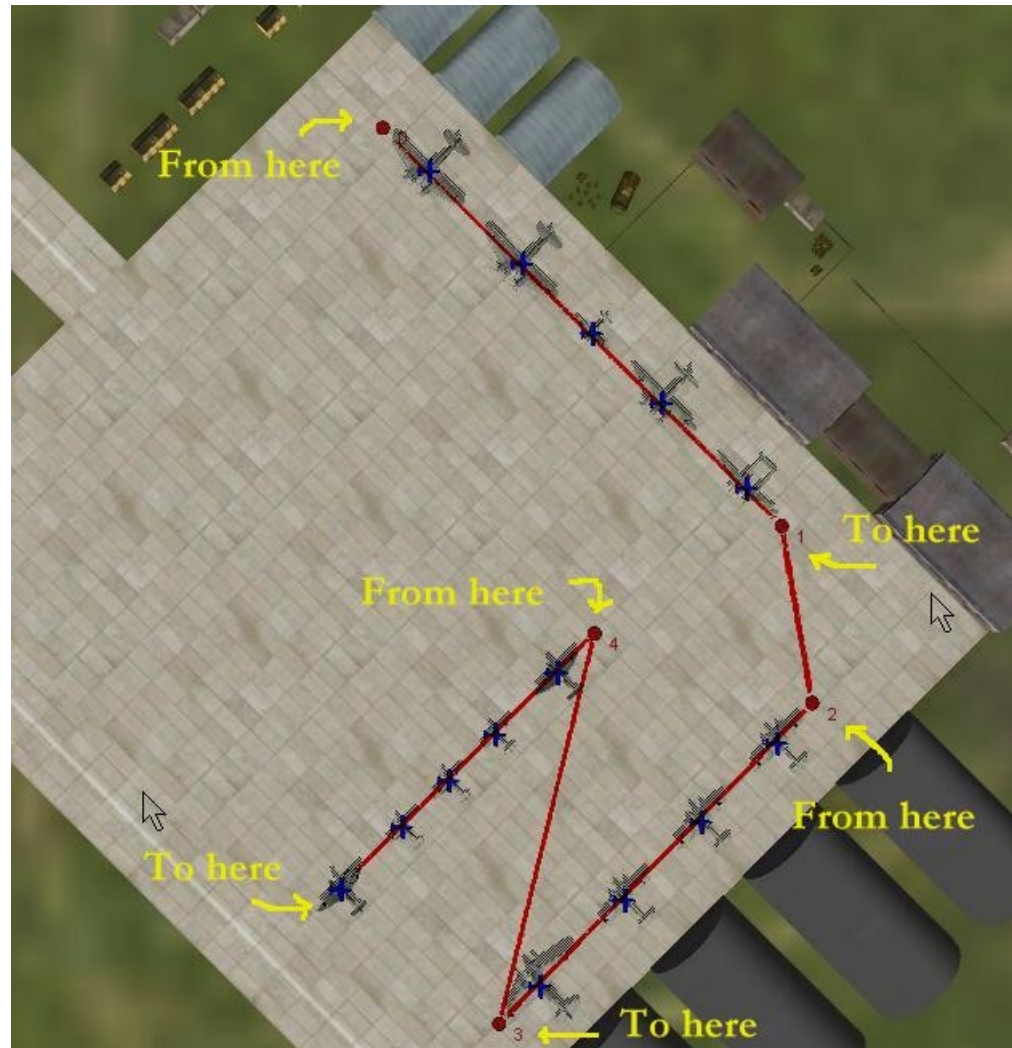
Colocando WP5 pienso: **hasta aquí**

Como puedes ver, eres libre de seleccionar los puntos de inicio entre emplazamientos. En este ejemplo, las líneas entre el WP1 y el WP2 no definen un emplazamiento, es sólo una línea de reposicionamiento y tienes que ignorarla. Lo mismo se aplica a la línea entre el WP3 y el WP4.

Realmente es muy fácil después de hacerlo varias veces. La siguiente imagen muestra como son colocados los aviones en los emplazamientos vistos en la imagen anterior:



Y la siguiente imagen es la combinación de ambas, mostrando lo que tienes que “imaginar” en el momento de “dibujar” tus plantillas. Recuerda, es **MUY** importante que utilices parejas de WP, porque esas parejas definen las líneas. Sólo recuerda, nemotécnicamente, “desde aquí-hasta aquí”.



3. Creando plantillas

Crear una plantilla para un mapa significa que tienes que hacer varios archivos. Puedes hacer manualmente cada uno de ellos, pero esto suele producir errores, porque montones de coordenadas tienen que ser escritas y un error en una coordenada producirá una misión incorrecta. Así que suministro varias herramientas para hacer plantillas de una manera semi-automática.

Creamos plantillas en varios **estados** (o pasos), porque algunos pasos necesitan datos de los pasos anteriores. Por ejemplo, el "geo_obj.data" contiene cuatro (4) grupos principales de datos. Uno de estos grupos son las definiciones de Sector, y para escribir las definiciones de Sectores necesitas tener los WP de tanques. Y los WP de tanques se harán usando una plantilla creada después de finalizar el "stage 1" (paso 1 ó estado 1), donde definimos aeródromos y lugares de emplazamiento de aviones. No te preocupes ahora por esto, lo único importante ahora es saber que hay varios "stages".

Esta creación semi-automática de plantillas usa "scripts" en "Perl". Así que necesitas tener instalado "Perl". Si no lo tienes no será posible hacer plantillas tal y como describo en este manual.

El proceso de creación de plantillas necesita que trabajes con el FMB, y de cuando en cuando ejecutes algunos "scripts", así que para hacer las cosas más fáciles, crea una carpeta en **IL-2\missions**. Yo suelo nombrar la carpeta con el nombre de las plantillas del mapa en construcción, por ejemplo, para hacer las plantillas para "normandie1", tengo una carpeta: **C:\UbiSoft\IL-2 1946\Missions\normandie1**, en la que meto los "scripts" de "Perl" y donde colocaré los archivos de las plantillas. Una vez que han finalizado los renombro añadiendo 2 ó 3 letras antes de cada archivo para reconocerlos más fácilmente.

Copia todos los "scripts" suministrados con este manual a la carpeta que quieras.

3.1 Primer paso

Comencemos a crear plantillas para el mapa Normandie1. Tal y como describimos antes, yo coloco todos los “scripts” en una carpeta llamada “normandie”. Luego lanzo el IL-2 y cargo el mapa “normandie1”. Lo primero que hay que hacer es colocar un objeto “airfield campfire” (hoguera) en la esquina superior derecha, mejor un poquito fuera de la última coordenada (si puedes; si no, lo más lejos que puedas en la esquina superior derecha). Esto será utilizado por los “scripts” para determinar los límites del mapa. Guarda este archivo como **“initial.mis”**. Ahora este archivo nos proporcionará datos importantes: los límites y el nombre del mapa.



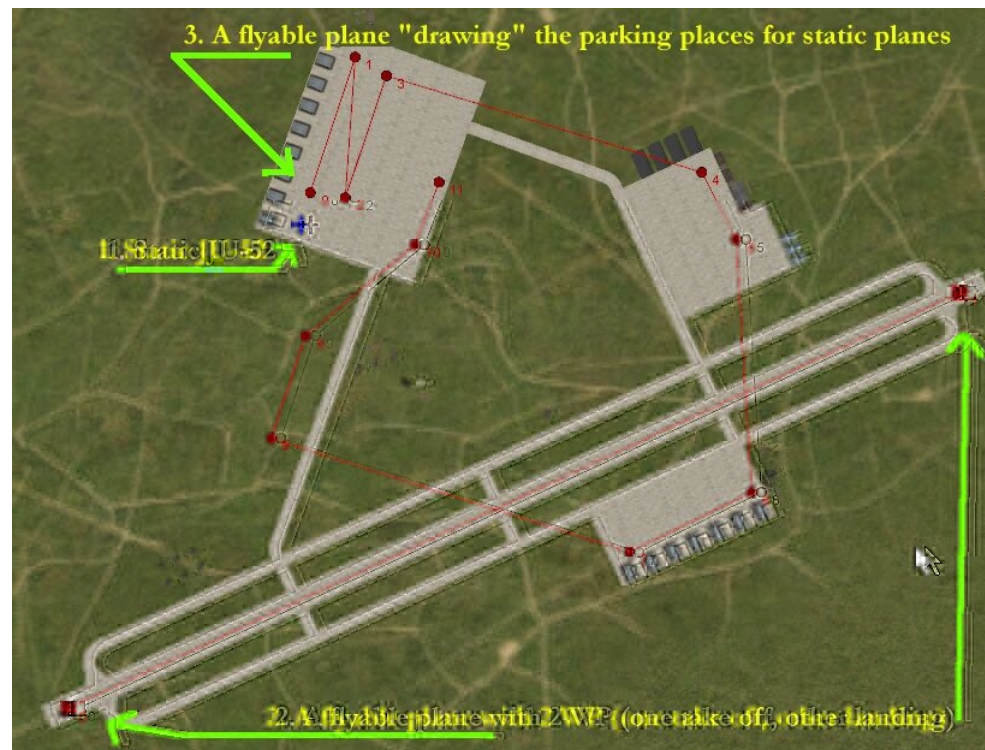
En la imagen de arriba vemos la localización del “airfield campfire” para el mapa “normandie1”. Este mapa tiene en su esquina superior derecha las coordenadas O-10, así que el “airfield campfire” estará situado en las coordenadas P-11. Por supuesto que las coordenadas P-11 no se ven el FMB. Esta cuadrícula superior derecha tendrá diferentes coordenadas en los distintos mapas.

Este nuevo archivo “**initial.mis**” contendrá los datos de los aeródromos, además de otra información que ya dije. Por cada aeródromo del mapa que queramos utilizar en la Campaña, necesitamos hacer acopio de varias informaciones: las coordenadas de localización, las 2 cabeceras para despegue y aterrizaje, la orientación de la pista, los “emplazamientos” en los que queremos colocar los aviones aparcados y el lugar para el avión estático que será utilizado para la identificación del bando del aeródromo.

Todos estos datos son recopilados añadiendo 3 objetos a cada aeródromo:

1. El primer objeto será un Ju-52 estático, colocado donde quieras.
2. El segundo objeto es un vuelo (con cualquier avión; yo uso un B-17) con 2 WP: WP0 es un “**take-off**” y el WP1 es un “**landing**”. Este vuelo proporcionará la información de las coordenadas de los puntos de despegue y aterrizaje, y también proporciona la “localización central” de un aeródromo y las cabeceras de pista.
3. El tercer objeto es un vuelo, pero **otro vuelo** diferente al anterior (con cualquier avión, pero yo también uso un B-17) que contendrá los lugares de parking para los aviones, como se describió en la sección “plantillas de emplazamiento”. Puedes poner hasta 9 emplazamientos, esto es, hasta 18 WP. Recuerda la orientación hacia la “derecha” y la regla nemotécnica “desde aquí-hasta aquí”.

Cuando hayas colocado estos 3 objetos, selecciona otro aeródromo y repite el proceso colocando un nuevo Ju-52, un vuelo de B-17 con 2 WP (uno despegue y otro aterrizaje), y otro vuelo de B-17 para “dibujar” los puntos de parking. La siguiente imagen muestra una captura de pantalla con los 3 objetos colocados en un aeródromo.



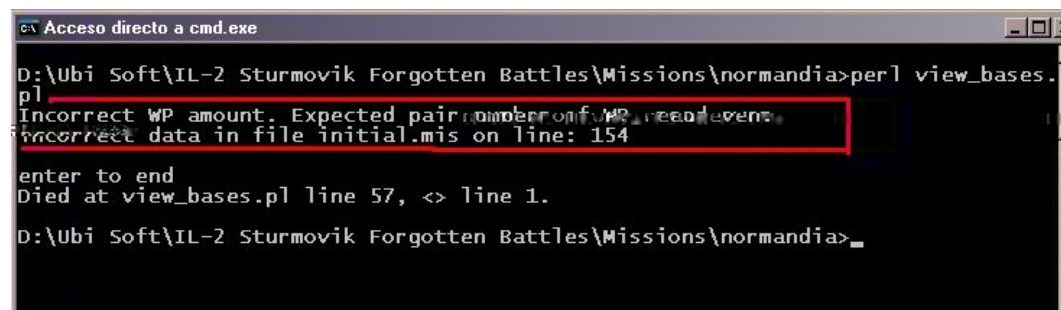
De cuando en cuando deberás comprobar cómo están colocados los aviones, si la orientación es correcta, si tienes correctamente hechas las parejas de WP para los lugares de emplazamiento, y todo lo demás. Para echar un vistazo a tu trabajo, ejecuta el "script" "view_bases.pl". Este "script" leerá tu archivo "initial.mis" y generará un nuevo archivo llamado "preview_bases.mis". Carga este archivo con el FMB y comprueba la colocación de los aviones. Este "script" escribirá un error si no tienes **pares de WP** (incluso si hay montones de WP). La vista previa sólo mostrará aviones alemanes, como si la Base estuviese bajo control alemán; luego el Generador colocará aviones alemanes o soviéticos.

Recuerda guardar el trabajo en “initial.mis”. **Es un error común cargar la vista previa y luego guardarla añadiendo Ju-52, lugares de parking y WP de despegue y aterrizaje.** La misión de vista previa es sólo eso, una misión para comprobar cómo el Generador leerá los WP. El archivo importante es el “initial.mis”. Después de inspeccionar la misión de “vista previa” recuerda volver a cargar “initial.mis” para continuar trabajando en ella.

La siguiente imagen es como verás una Base en la “preview_base.mis”. Observa como el Ju-52 estático no será mostrado. La misión de vista previa sólo muestra los emplazamientos de parking de los aviones.



Así es como se mostrará el mensaje de error cuando trates de hacer una vista previa y los montones de WP no sean pares. Cuando esto ocurra tendrás que cargar el archivo “initial.mis” en el FMB y buscar dónde está el problema. Una recomendación es hacer una “vista previa” cada vez que termines con una Base. Así, si hay un error con las parejas de WP, el problema siempre estará localizado en la última Base en la que has trabajado. Adicionalmente, el mensaje de error mostrará en que número de línea se está produciendo, así que podrás borrar manualmente esa línea abriendo “initial.mis” con un editor de texto (no recomendado; mejor hazlo con el FMB).



```
Acceso directo 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 event.
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>
```

Ésto es todo lo que necesitas para finalizar el primer paso, un “initial.mis” con un *campfire* y cada Base con 3 objetos. Una vez que estés habituado a hacer estas Bases, cada Base no te llevará más que unos pocos minutos. La forma en la que trabajo es haciendo primero el archivo con el *campfire*, luego añado los Ju-52 a todas las Bases que quiero utilizar, de tal forma que constituyen una ayuda visual para localizar las Bases. A continuación añado todos los WP de despegue y aterrizaje utilizando un B-17 para cada Base. De esta manera, cuando echo un vistazo al mapa, tengo que ver algo azul (los Ju-52) y algo rojo (los WP de los B-17). Finalmente añado los parkings en cada Base, haciendo la misión de “vista previa” de cuando en cuando.

Si cargas el “initial.mis” que proporciono como ejemplo verás marcas del frente. No las necesitas todavía. Es algo que añadiremos en los próximos pasos, y se explicará más adelante.

Ahora estamos listos para ejecutar el “script” “1_stage.pl” (haz en una consola cmd de Windows: perl -w 1_stage.pl). Esto creará 5 archivos:

1. **blue_obj.mis:** Un Ju-52 por Base, en los lugares indicados por el Ju-52. Este archivo está listo para usar.
2. **red_obj.mis:** Un Li-2 por Base, en los lugares indicados por el Ju-52. Este archivo está listo para usar.
3. **tank_wp.mis:** Una plantilla inicial para tanques; tendremos que trabajar en ella en el siguiente paso.
4. **city.mis:** Una plantilla vacía a la que añadiremos las ciudades para atacar y los objetivos en el siguiente paso.
5. **geo_obj.data:** Es un contenido parcial de este archivo, que ahora contiene las descripciones de todos los aeródromos, lugares de parking, cabeceras, daños, etc. etc.

Si tienes curiosidad puedes mirar dentro del archivo “geo_obj.data” que se acaba de generar y podrás ver varias líneas con la información de cada aeródromo. Por ejemplo, en éste están las líneas que representan el aeródromo mostrado en las imágenes anteriores. Esa información es: información general del aeródromo (primera línea), coordenadas de orientación (AF06:Hx), seguidas por varias líneas con las coordenadas de los lugares de parking y el ángulo para los emplazamientos de los aviones (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)
```

No te preocupes si no entiendes esta información, no necesitas hacerlo para seguir trabajando con las plantillas.

3.2 Segundo paso

Este paso es el que lleva más tiempo, porque tenemos que trabajar en la plantilla de los WP de tanques y en la plantilla de la ciudad. Lo dividiré en dos partes, “A” y “B”, para cada una de ellas.

A- Plantilla de WP de tanques

La plantilla de tanques contiene los “puntos” en los que el Generador emplazará un *Campamento defensivo* (algunos objetos estáticos, AAA, y tanques estáticos) o un *grupo de tanques en movimiento* (el grupo de ataque). Cada Sector necesita al menos 2 “puntos” para que sea posible emplazar los dos grupos, uno en cada “punto”. Cuantos más “puntos” mejor, así el Generador podrá realizar más combinaciones, haciendo las misiones menos repetitivas. El total de “puntos” recomendado para cada Sector es hasta 5, y no menos de 3 para hacer algunas variaciones en la selección.

Hay algunos Sectores en los que los tanques no se pueden mover. Por ejemplo el mar, lagos, sectores cubiertos de bosque, etc. En estos sectores pondremos 0 “puntos”; es decir, no habrá “puntos” iniciales y los sectores serán inatacables.

Vamos al FMB y cargamos la misión "tank_wp.mis", creada en el primer paso. Verás algo como ésto (sólo se muestra una pequeña parte del mapa):



Puedes ver en **todos** los sectores 5 tanques, uno en cada esquina y uno más en el centro del sector. Este archivo está creado de esta manera para acelerar la edición. Es más fácil eliminar y mover objetos ya puestos en la misión. Cada uno de los tanques estáticos que ves es un “punto”. Al igual que en los lugares de parking usábamos un vuelo para que más tarde se colocasen los objetos, aquí usamos un tanque estático para indicar las coordenadas donde se emplazará un *Campamento* o un *grupo de tanques en movimiento*. En esa imagen numeré los Sectores del 1 al 4 sólo para facilitar las referencias en este manual.

La imagen previa muestra 4 Sectores. Lo primero que hay que hacer es borrar los 5 “puntos” de cada Sector inatacable (mar, lagos y sectores que sólo tienen bosque). Luego tienes que recolocar los “puntos” aplicando unas simples reglas. Los “puntos” (actualmente tanques en esta plantilla) necesitan no estar demasiado cerca unos de otros. Todos tienen que estar dentro del Sector; ningún río, lago, ciudad o bosque puede interceptar su ruta. Si no puedes situar los 5, prueba borrando algunos para una más fácil localización de los emplazamientos correctos. La siguiente imagen muestra cómo edité esos “puntos”. El Sector numerado como 1 es inatacable (mar) y eliminé los 5 “puntos”. El Sector 2 tiene 4 “puntos”, el Sector 3 tiene 5 “puntos” y el Sector 4 tiene sólo 3 “puntos”:



Como puedes ver, todos los “puntos” cumplen los requisitos. Ahora mostraré una imagen con emplazamientos incorrectos de los “puntos”. Añadiré algunas líneas para unirlos entre ellos y pondré puntos rojos sobre las líneas verdes para ver dónde está el problema. Estas líneas has de trazarlas mentalmente mientras mueves los “puntos”.



El Sector 1 tiene un emplazamiento al otro lado del río. El Sector 2 tiene dos emplazamientos incorrectamente colocados, porque la ruta entre ellos atraviesa una ciudad. El Sector 3 es correcto, pero dos emplazamientos están demasiado cerca, así que es conveniente mover uno de ellos un poquito más lejos, como muestra la línea roja. El Sector 4 tiene dos rutas atravesando el bosque. En la siguiente imagen se ve cómo se han corregido los emplazamientos:

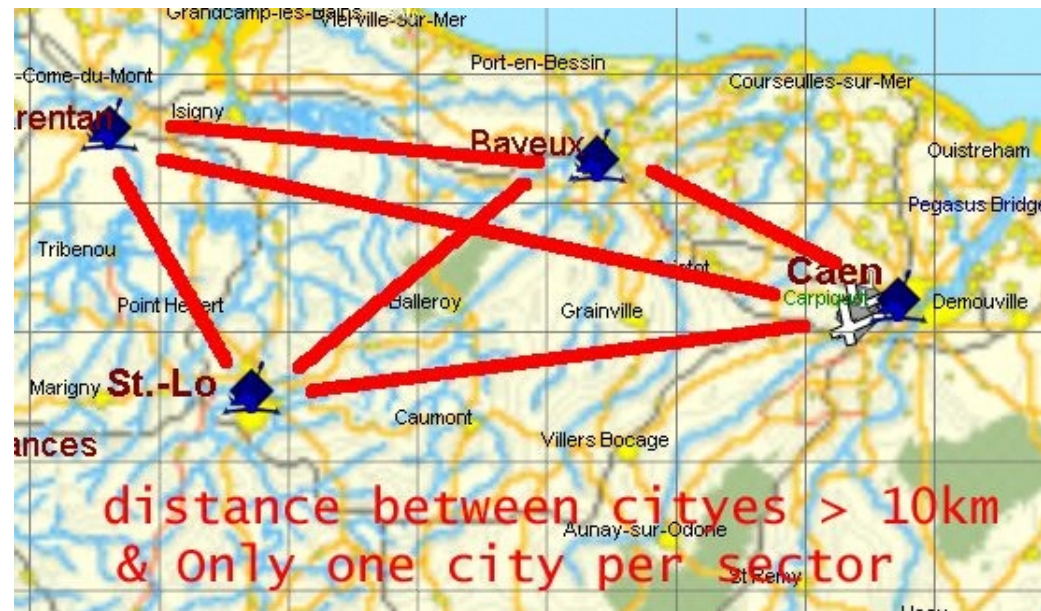


Hay que comprobar, Sector por Sector, los puntos de emplazamiento. Asegúrate de que cada Sector tenga por lo menos 2 WP, aunque es recomendable un mínimo de 3. También has de asegurarte de no colocar emplazamientos fuera del Sector. Para trabajar más cómodamente, pon y quita el “zoom” sobre el mapa para ver sólo 10x10 cuadrículas. También es aconsejable, para zonas con colinas y montañas, ver exactamente el lugar donde colocas un emplazamiento, así evitarás situarlo en precipicios.

Esta plantilla lleva algún tiempo, pero cuando estés habituado a trabajar en ella, podrás hacer varios Sectores en un minuto. La cuestión más importante es hacer inatacables los Sectores en los que no se puedan poner “puntos” de emplazamiento de tanques. Los demás Sectores podrán ser corregidos o editados más tarde (problemas con colinas, o un emplazamiento al otro lado de un río, etc.)

B- Plantilla de ciudad

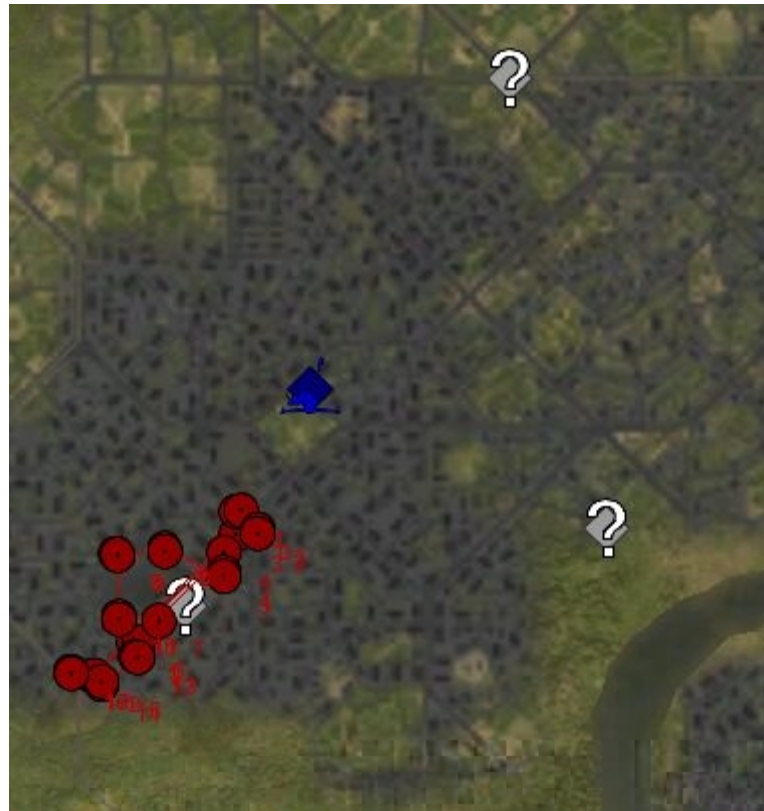
Comenzamos con la Plantilla de ciudad. Esta plantilla contendrá las ciudades que nos gustaría incluir como objetivos. Sólo una ciudad por Sector. La ciudad no puede estar a menos de 10 Kms. de otra. Carga “city.mis” en el FMB. Lo primero que hay que hacer es situar un “*observation balloon 90m*” (globo de observación 90m.) sobre cada ciudad que queramos incluir. De esta forma comprobaremos fácilmente que no tenemos dos ciudades demasiado cerca. La siguiente imagen muestra cómo aparece la Plantilla después de añadir varias ciudades. Las líneas están dibujadas únicamente para ver la distancia a comprobar:



Después de terminar, deberás hacer “zoom” sobre cada ciudad y colocar varios “*campfire*”. Cada “*campfire*” será una “zona atacable”. Añade al menos una y hasta 9 zonas en cada ciudad. Todos los lugares marcados con una luz (zonas) tienen que estar cerca de la ciudad; nunca los sitúes a una distancia mayor de 5 kms. Cada zona necesita estar separada 1 Km. o más de las otras. Así que no la llenes de objetos (“*campfire*”) y no los superpongas. La siguiente imagen muestra una ciudad con 3 zonas (3 “*campfire*”), todas ellas dentro del límite de 5 kms. y todas ellas separadas al menos 1 Km. de las demás.



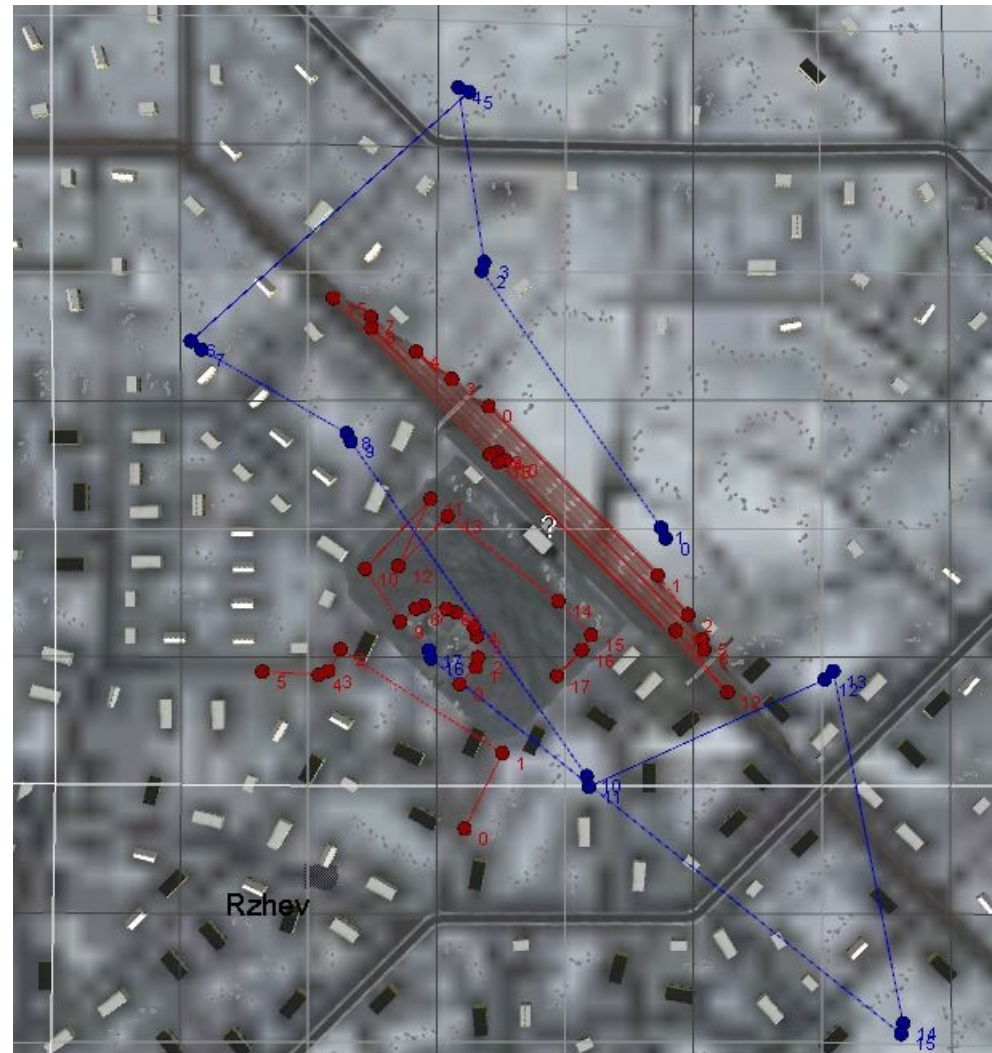
Pon más zonas en las ciudades más importantes (5 ó 6 está bien) y menos zonas en las ciudades pequeñas (2 ó 3). Cuando hayas terminado de situar los “marcadores” de zonas, tendrás que colocar objetos en cada uno de ellos. La siguiente imagen muestra una zona con objetos y otras todavía sin ellos.



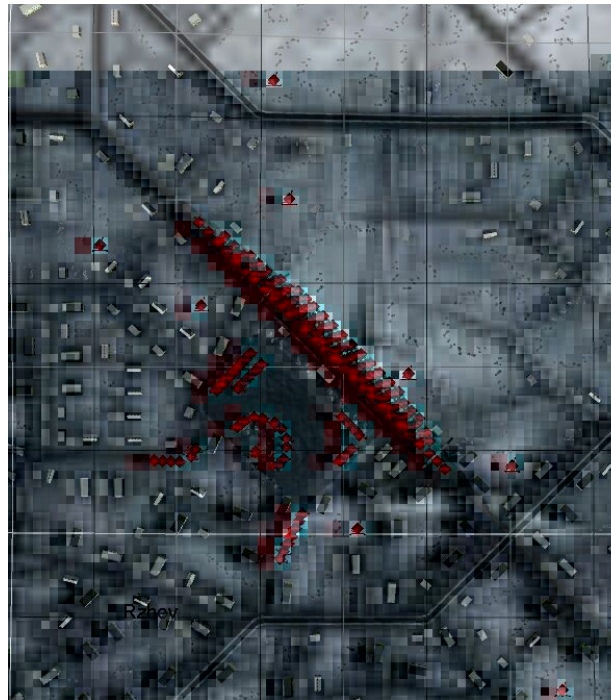
La forma de colocar objetos es muy similar a la descrita en el apartado “place templates” (plantillas de emplazamiento). De hecho es igual, pero con algunas opciones más. Tienes que dibujar parejas de WP. Estas parejas de WP indican dónde serán situados los objetos. Pero ahora tenemos una característica extra. Usando la **altitud** de los WP podemos indicar diferentes tipos de objetos. Piensa en los WP como si fuesen un lápiz y la altitud el tipo de lápiz. Así que, con distintas altitudes podemos “dibujar” diferentes objetos.

- WP a 500 metros son vehículos paralelos a la línea dibujada.
- WP a 1000 metros son vehículos rotados a la “derecha” de la línea.
- WP a 1500 metros son trenes paralelos a la línea; la locomotora estará situada en el primer WP y el último vagón en el siguiente WP.
- WP a 2000 metros son AAA. Sólo se sitúa una AAA por cada pareja de WP.

Nuevamente, tampoco importa qué avión utilices para dibujar los emplazamientos. Pero ahora es importante reflejar la **altitud** correcta para representar los diferentes tipos de objetos que quieras colocar. Suelo utilizar un avión alemán (WP azul) para “dibujar” los emplazamientos de la AAA, y uso un vuelo rojo para situar los emplazamientos de los vehículos. Y para cada emplazamiento uso diferentes vuelos para diferentes objetivos. Por ejemplo, con un B-17 dibujo todas las líneas para los vehículos paralelos a las líneas (altitud de WP 500 m.). Pongo entre 10 y 20 WP, así que “dibujo” entre 5 y 10 emplazamientos. Luego hago un nuevo vuelo de B-17, con la altitud de los WP a 1000 m. y sitúo varios vehículos rotados a la “derecha”. Si hay una vía de tren hago otro vuelo de B-17 situando la altitud de los WP a 1500 m. y añado todos los trenes. Finalmente hago un vuelo alemán (con cualquier avión, yo uso un Ju-87) e inserto varios emplazamientos de AAA. La siguiente imagen muestra como se ve un objetivo.



Como antes, tienes un “script” para previsualizar cómo serán leídos los objetivos por el Generador. Este “script” es denominado “view_city.pl” y leerá el archivo “city.mis” (sobre el que estás trabajando) y crea un nuevo archivo llamado “preview_city.mis”. Después de añadir una o más zonas, ejecuta el “script” y luego carga en el FMB el archivo de previsualización. Comprueba cómo están situados los objetivos, si todo lo que has querido añadir se encuentra en el lugar correcto, etc. Luego vuelve atrás y carga de nuevo “city.mis”. Como antes, no cometas el error de seguir añadiendo zonas en la misión de vista previa. Este “script” también te dirá si hay algún problema con las parejas de WP. La siguiente imagen muestra la previsualización de la zona vista en la imagen puesta antes como ejemplo.



Y la siguiente imagen muestra la combinación de las dos anteriores, en la que se pueden ver los objetos situados sobre las líneas que estabas “dibujando”. Esto es lo que tendrás que imaginar cuando añadas “puntos”. Fíjate que todos los objetos en la previsualización son rojos, pero más tarde el Generador situará los emplazamientos correctos de cada bando basándose en cuál sea el propietario de la ciudad.



Bien, ahora estamos listos para ejecutar el “script” del segundo paso. Este “script” leerá ambas plantillas (“tanks_wp.mis” y “city.mis”) y **añadirá la información al archivo geo_obj.data**. La información añadida es:

Para Información del Sector, líneas como:

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

Para Información de la ciudad, líneas como:

```
# Name F08  
SUC02, SUM-ci ty-F08, 53945. 30, 73549. 92, -, -, -, -, -: 0  
CT02, ci ty-F08, 53945. 30, 73549. 92, 25, ti po, TOTAL_ZONES_HERE, 0, 25: 0  
CT02Z0: 54427. 62, 74109. 38, 1  
CT02Z1: 55014. 97, 73130. 47, 1  
CT02Z2: 54036. 05, 72934. 69, 1
```

Para relaciones entre ciudad y Sector, líneas como:

```
pobl ado, ci ty-H07, sector--H07  
pobl ado, ci ty-F08, sector--F08  
pobl ado, ci ty-I 04, sector--I 04
```

3.3 Tercer paso

Este último paso consiste en la personalización de tus plantillas. El archivo “geo_obj.data” está casi completo, pero necesita unos retoques. Abre el archivo con un editor de texto. Verás que las ciudades no tienen nombres. Y ésto es porque tienen que ser escritos manualmente. Para cada “ciudad-XXX” encuentra el nombre en el FMB

y sustitúyelo. Recomiendo usar “*buscar y reemplazar todo*”, evitando así errores en los nombres. Realmente no tiene importancia si el nombre de la ciudad es incorrecto, pero tiene que ser incorrecto en todas partes.

Por ejemplo, la ciudad en el mapa “normadie1”, en el Sector F-08 es **Bayeux**, así que hacemos estos cambios en el archivo “geo_obj.data”:

Líneas iniciales afectadas:

```
# Name F08
SUC02, SUM-ci ty-F08, 53945. 30, 73549. 92, -, -, -, -, -: 0
CT02, ci ty-F08, 53945. 30, 73549. 92, 25, ti po, 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)
.
pobl ado, ci ty-F08, sector--F08
.
.
```

Ahora sería así:

```
# Bayeux F08
SUC02, SUM-Bayeux, 53945. 30, 73549. 92, -, -, -, -, -: 0
CT02, Bayeux, 53945. 30, 73549. 92, 40, ti po, 3, 0, 40: 0
CT02Z0: 54427. 62, 74109. 38, 1
CT02Z1: 55014. 97, 73130. 47, 1
CT02Z2: 54036. 05, 72934. 69, 1
.
.
.
pobl ado, Bayeux, sector--F08
.
.
```

Fíjate que he cambiado el texto "TOTAL_ZONES_HERE" por el total de zonas, en este caso 3. Además prefiero poner Bayeux con un radio máximo de 40 Kms. en lugar de los 25 por defecto, así que cambié ambos "valores 25" a 40. Estos valores son el *radio máximo de suministro* y el *radio actual de suministro* (ambos valores son iguales porque el nivel de daño es 0).

Una vez que termines de añadir los nombres de las ciudades y fijar el número correcto de zonas y el radio inicial de suministro, hay que definir cuál será la línea inicial del frente para la campaña. Tienes que hacer que todos los objetos en "geo_obj.data" estén "sincronizados" con la línea inicial del frente.

Para lograr ésto, carga el archivo "initial.mis" y pon algunas marcas azules y rojas del Frente. Guarda el archivo y ejecuta el "script" "3_stage.pl". Este "script" leerá la línea de frente que has fijado en "initial.mis" y creará una plantilla "frontline.mis". También sincroniza el archivo "geo_obj.data". Ahora casi todo está hecho.

4. Aviones

El archivo "aircrafts.data" tiene una definición para cada avión que quieras incluir en la campaña. Este archivo también contiene información sobre los nombres de los Escuadrones, los aviones estáticos y los vehículos estáticos. Las líneas que comienzan con la almohadilla (#) se ignoran.

No hay una forma automática de crear este archivo, así que abre un nuevo archivo en un editor de texto, escribe las siguientes indicaciones y luego guarda el archivo como "aircrafts.data".

Las siguientes líneas indican que los Escuadrones de Cazas Rusos pueden ser 176IAP o 3IAK. La selección será aleatoria. El Escuadrón Jabó será 11IAP y Transporte (SUM) y Bombarderos (BA) serán 34BAP. Para los cazas

alemanes tendremos los Escuadrones I_JG3,II_JG3,I_JG27, los Bombarderos usarán Stab_KG27a,I_KG27a,II_KG27a,III_KG53a, y así sucesivamente. Otras líneas como éstas describen diferentes escuadrones para otras combinaciones de país/tarea. Cuando hay más de una definición, una de ellas será seleccionada aleatoriamente. Tendrá que haber al menos un Escuadrón para cada definición país/tarea existente.

```
##--red sqds
rusfi g=176I AP, 3I AK;
rusj ab=11I AP;
rusbom=34BAP;
rustrp=34BAP;

gerfi g=I _JG3, II _JG3, I _JG27;
gerj ab=I _LG2;
gerbom=Stab_KG27a, I _KG27a, II _KG27a, III _KG53a;
gertrp=KGrzbV102a;

romfi g=Eskadri I a54;
romj ab=Eskadri I a54;
rombom=Eskadri I a54;
```

Los nombres de Escuadrones no pueden ser cualquiera que desees. **Tienen que ser exactamente los mismos utilizados por el FMB.** Si quieres añadir nombres de Escuadrones, asegúrate de que están disponibles en el Simulador. Este archivo contendrá la descripción del avión y valores separados por “coma”, “dos puntos” (:) tarea y número total. Los campos están en orden:

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

Army 1 para Aliados, 2 para el Eje. El “country+task” (romfig, gerjab, rusfig, hunfig, itajab, etc..) es una combinación de: 3 letras para el país + 3 letras para el rol. Más tarde será utilizado para poner un nombre al Escuadrón, como III_JG52 ó 84IAP, etc. Fíjate, normalmente sólo 3 países están permitidos (*rus,ger* y *rom*); ésto tiene que ser ampliado para otros países en “badc_gen_1.pl”. Otros archivos se explican por sí mismos.

Las tareas son:

BA : **B**omber **A**ttack. Grupo de Bombarderos que atacan ciudades o aeródromos.

SUM: **S**upply **M**ission. Grupo de aviones de Transporte, haciendo misiones de suministro.

BD: **B**omber **D**efend. Grupo de Bombarderos de Defensa, que atacan a los tanques enemigos.

EBA: **E**scort **BA** bombers. Escolta de Bombarderos.

ESU: **E**scort **S**upply transports. Escolta de Suministros.

EBD: **E**scort **BD** bombers. Escolta de Bombarderos de Defensa.

ET: **E**scort **T**anks. Cuando ataques un Sector, aviones dando cobertura aérea e interceptando **BD enemigos**.

INT: **I**NTercept. En general, interceptar Transportes de SUM y BA enemigos.

El número es la cantidad total de aviones de cada tipo y es igual (no obligatoriamente) para cada entrada. Aquí está una entrada para un P-40 caza/bombardero:

(ADVIERTE QUE LA TAREA DE INTERCEPTOR ES UNA SOLA "I" NO "INT" como se describió arriba)

```
# P40-E
1, rusj ab, P-40E, 1, ai r. P_40E, 500l b, 100, 3000, 400: BD, 6,
1, rusfi g, P-40E, 1, ai r. P_40E, 500l b, 100, 3000, 400: ET, 6,
1, rusfi g, P-40E, 1, ai r. P_40E, defaul t, 100, 3000, 400: EBA, 6,
1, rusfi g, P-40E, 1, ai r. P_40E, defaul t, 100, 3000, 400: EBD, 6,
1, rusfi g, P-40E, 1, ai r. P_40E, defaul t, 100, 3000, 400: ER, 6,
1, rusfi g, P-40E, 1, ai r. P_40E, defaul t, 100, 3000, 400: I , 6,
```

Para obtener los nombres *"class air"*, abre el FMB, carga cualquier mapa pequeño, por ejemplo "online1". Pon un avión del que quieras obtener el *"class air"*, digamos un Fw190-A8, y pon un vuelo con ese avión con armamento en "default". Luego pon otro Fw190-A8 y ponle una bomba que quieras utilizar, por ejemplo una SC500. Guarda esta misión con cualquier nombre, por ejemplo "airnames.mis" (puedes poner tantos vuelos como nombres de *"class air"* necesites). Abre esta misión con un editor de texto y verás algo como esto:

```
[MAIN]
MAP Net1Summer/Load. ini
```



```
TIME 12.0
CloudType 0
CloudHeight 1000.0
army 1
playerNum 0
[Wi ng]
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]
[Bui l di ngs]
[Bri dge]
[House]
```

Puedes ver que el “*class air*” es: **air.FW_190A8**

El armamento por defecto es: **default**

La carga de bombas es: **1sc500**

Bien, hagamos la entrada para este FW190-A8 usando el armamento por defecto y una 1SC500 para bombardero de defensa (BD), y establezcamos 45/300 aviones. Primero escribimos una sola línea para el rol de Interceptor:

```
2, gerfi g, FW-190A8, 1, ai r. FW_190A8, defaul t, 100, 3000, 400: I , 45,
```

Estos valores son:

2 = bando del Eje

gerfig = 3 letras del país (**ger**many)+ 3 letras de rol (**fi**ghter)

FW-190A8 = nombre del avión que se mostrará en los briefings, reportes y estadísticas.

1 = porque es pilotable

air.FW_190A8 = es el “class air” explicado anteriormente

default = es la carga de armamento

100 = carga de combustible

3000 = altitud de crucero

400 = velocidad de crucero

: es un separador

I = tarea de Interceptor (fíjate en que es una sola letra, en lugar de las 3 INT)

45 es el número total de aviones

, coma final

Cuando hayas escrito esta línea, cópiala y pégala varias veces (4 más) para obtener esto:

```
2, gerfi g, FW-190A8, 1, ai r. FW_190A8, defaul t, 100, 3000, 400: I , 45,  
2, gerfi g, FW-190A8, 1, ai r. FW_190A8, defaul t, 100, 3000, 400: I , 45,  
2, gerfi g, FW-190A8, 1, ai r. FW_190A8, defaul t, 100, 3000, 400: I , 45,  
2, gerfi g, FW-190A8, 1, ai r. FW_190A8, defaul t, 100, 3000, 400: I , 45,  
2, gerfi g, FW-190A8, 1, ai r. FW_190A8, defaul t, 100, 3000, 400: I , 45,
```

Finalmente modifícalo para tener también los roles de EBA EBD ET BD, pon 1SC500 para tareas de ET y BD, reduce la altitud de crucero a 1000 y pon el código “gerjab” para el bombardeo defensivo (BD); suelo añadir una línea comentada (empieza con #) con el nombre y más información al principio.

```
# FW-190A8 45/300 planes as Pirx table for map bla-bla at year xxxx ...
2,gerfi g,FW-190A8,1,ai r.FW_190A8,default t,100,3000,400:I,45,
2,gerfi g,FW-190A8,1,ai r.FW_190A8,default t,100,3000,400:EBA,45,
2,gerfi g,FW-190A8,1,ai r.FW_190A8,default t,100,3000,400:EBD,45,
2,gerfi g,FW-190A8,1,ai r.FW_190A8,1sc500,100,3000,400:ET,45,
2,gerj ab,FW-190A8,1,ai r.FW_190A8,1sc500,100,1000,400:BD,45,
```

Si investigas otras cargas de armamento sabrás que los cañones mk108 en el FW190-A8 son denominados "r22mk108", así que si te gustaría añadirlos para las tareas de Intercepción, cambia "default" por "r22mk108" en la línea de rol de Intercepción, que quedaría así:

```
2,gerfi g,FW-190A8,1,ai r.FW_190A8,r22mk108,100,3000,400:I,45,
```

Repite este método para otros aviones. Habitualmente hago una misión "airnames.mis" en la que pongo todos los aviones que necesitaré con diferentes cargas bélicas. Más tarde abro esa misión para leer todos los datos "class air" y de armamentos. En tu caso, puedes necesitar P-51, P-38, P-47, 109-G6, B-25, etc., etc., inclúyelos todos en tu "airnames.mis" usando el FMB.

Luego, en este archivo tendremos distribuciones de aviones estáticos y vehículos estáticos. La línea de los aviones estáticos será algo así:

```
ST100,1,I153,vehi cl es.pl anes.Pl ane$I _153_M62,15:150
ST101,1,I16,vehi cl es.pl anes.Pl ane$I _16TYPE24,15:350
ST102,1,I L2,vehi cl es.pl anes.Pl ane$I L_2_1940Late,20:400
ST103,1,LaGG3,vehic les.pl anes.Pl ane$LAGG_3SERIES4,15:500
```

Y los campos son:

STx+ 2 números: x=1 para aliados y x=2 para el Eje
bando
nombre externo (para los reportes)

nombre del objeto en el IL-2
envergadura aproximada del objeto
: número

El número son valores acumulativos, y el último avión estático tiene que tener el número 1000. Tienen que estar en orden; para el ejemplo de arriba: tenemos 150 I-153 por 1000 aviones, 200 (350-150) 16type24, 50 (400-350) IL2 1940, 100 (500-400) LaGG3, y así sucesivamente. Fíjate que, mezclados con los aviones estáticos, tenemos algunos camiones de combustible, que serán colocados como si fuesen aviones, simulando el repostaje. Por último la definición de vehículos estáticos. Es casi idéntica a la de los aviones estáticos, usando una frecuencia de valores acumulativos hasta 1000. A continuación algunos ejemplos para los vehículos estáticos del bando azul:

```
SV201, 2, VW82, vehi cl es. stati onary. Stati onary$VW82, 15: 80  
SV202, 2, VW82t, vehi cl es. stati onary. Stati onary$VW82t, 15: 160  
SV203, 2, Opel Kadett, vehi cl es. stati onary. Stati onary$Opel Kadett, 15: 260  
SV204, 2, Opel Bl i tz36S, vehi cl es. stati onary. Stati onary$Opel Bl i tz36S, 15: 340  
SV205, 2, Opel Bl i tz6700A, vehi cl es. stati onary. Stati onary$Opel Bl i tz6700A, 15: 440  
SV206, 2, Kettenkrad, vehi cl es. stati onary. Stati onary$Kettenkrad, 15: 500
```

Y los campos son:

SVx+ 2 números: x=1 para los aliados y x=2 para el Eje
bando
nombre externo (para los reportes)
nombre del objeto en el IL-2
envergadura aproximada del objeto
: número

5. Imágenes

La imagen que usaremos será BMP 24bits/píxel, que tendrás que hacer con el programa que prefieras. Haz varias capturas de pantalla del mapa dentro del FMB. Luego éditalas y redimENSIONALAS para obtener 900 píxeles de ancho. Puedes usar otros tamaños, pero no es recomendable más de 900 para no sobrecargar el servidor. Más pequeño puede ser difícil ver la información. Utiliza siempre un tamaño divisible por 4, porque los “scripts” no pueden manejar decimales. Añade a la imagen los nombres de las ciudades. Nombra esta imagen como NM1.BMP. Más tarde usarás una herramienta para crear el archivo de la imagen del mapa (que se llamará “imap.data”). **GIMP** (<http://www.gimp.org.es/> y <http://www.gimp.org/>) tiene un buen editor de BMP.

Cuando hagas la imagen cuenta cuántos píxeles/sector hay. Necesitarás que esa información se escriba en “config.pl” (que se explicará luego). Por ejemplo, si tu imagen tiene 900 píxeles de ancho y el mapa tiene 15 Sectores, el número de píxeles/sector es 60. Pudiera ser algo menos, porque los mapas tienen sectores parciales en los bordes, donde están las coordenadas.

La imagen del mapa es sólo un archivo HTML, apuntando a la imagen (no exactamente, apunta a un “front.jpg”) y luego la definición del mapa, como ésta:

```
<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 al t="" USEMAP="#map">
<MAP NAME="map">
<AREA SHAPE="RECT" COORDS="0, 31, 46, 77" HREF=". /no_click.html " al t="sector--A16">
.
. (other imap lines)
.
</MAP>
```

Advierte que tienes que establecer el ancho y el alto. Normalmente la herramienta de edición de imágenes lo hará.

6. Configuración final

Una vez que tengas todos los archivos, es el momento de renombrarlos, para que sean únicos. Usar una combinación de 2 ó 3 letras será suficiente. Así que renombramos “normandie1” usando NM1.

```
initial.mis -> NM1_initial.mis (no se utilizará, pero guárdalo por si acaso)
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 sólo indica que es la primera imagen).
imap.data -> NM1_imap.data
```

Ahora hay que añadir esta información del mapa al archivo “config.pl”. Primero definimos un nombre de mapa (\$MAP_NAME_LONG), un texto corto con alguna información precisa sobre el mapa. Después hay que incluir varias líneas condicionales:

- 1) La imagen de información del mapa
- 2) El tipo de tanques usados para defensa y ataque por ambos bandos, y la presencia de AAA
- 3) Los nombres de los archivos para este mapa
- 4) El listado de aviones para este mapa (para el "takeslot.pl" form printing; se refiere a la pantalla de selección de avión una vez cargada la misión, pero no sé cómo traducirlo)

El MAP_NAME_LOAD (en nuestro ejemplo es "Normandy/load2.ini") puede obtenerse buscándolo en cualquiera de las plantillas que hemos creado. Abre una con un editor de texto y busca la línea que comienza con "MAP" y copia la información como sigue:

```
$MAP_NAME_LONG="Normandie. June 1944";

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

# 1. información de la imagen del mapa
$ANCHO=900;      # NM1_000.bmp ancho de la imagen en píxeles
$ALTO=600;      # NM1_000.bmp alto de la imagen en píxeles
$H_BLOCK_PIX=60; # tamaño horizontal del sector en píxeles
$V_BLOCK_PIX=60; # tamaño vertical del sector en píxeles
$LETRAS=16;     # número de letras (coordenadas) del mapa, 1 de más (Normandie1 tiene hasta la 0, 15)
$NUMEROS=11;    # números (coordenadas) del mapa, 1 de más (Normandie1 tiene 10)

# 2. tipos de tanques y emplazamientos de AAA
$ALLIED_TANKS_ATTK="Armor. 3-BT7";
$AXIS_TANKS_ATTK="Armor. 3-PzIIIG";
$ALLIED_TANKS_DEF="vehicles.artillery.Artillery\%BT7"; # fíjate en ^\^ antes de ^$"
$AXIS_TANKS_DEF="vehicles.artillery.Artillery\%PzIIIG"; # fíjate en ^\^ antes de ^$"
$AAA_IN_CHAMPS=0;      # poner o no AAA en los Campamentos
$LATE_AAA_IN_CHAMPS=0; # poner o no en los Campamentos AAA avanzada, como nimrod y M16

# 3. nombres de los archivos
$GEOGRAPHIC_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 . datos para el takeslot.pl los nombres de los aviones tienen que ser idénticos a los de 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");
}

```

Los valores en VVS_SUM_PLANES, VVS_BA_PLANES, VVS_AI_PLANES, son los aviones de Suministro, Bombarderos de Ataque, y listado de aviones IA permitidos (disponibles). Lo mismo para los campos con LW_* para los aviones del Eje. Todos los valores son usados en “take_slot.pl” para hacer la selección de los aviones al generar la misión. Los nombres de los aviones tienen que ser **exactamente iguales** que los escritos en el archivo “aircrafts.data”. Los que he puesto en el ejemplo no son para Normadía, son sólo un ejemplo.

Cuando establezcas los tipos de tanques de ATAQUE, asegúrate de incluir en la variable de su nombre (TANK_REGEX) una cadena para identificarlos. Por ejemplo, si añades KV1, tendrás que hacer que esta línea:

```
$TANK_REGEX="Pz|T34|ValentineII|BT7|M4A2|T70"; # la cadena de identificación para el tanque usa | como separador
```

sea como ésta:

```
$TANK_REGEX="Pz|T34|ValentineII|BT7|M4A2|T70|KV1"; # la cadena de identificación para el tanque usa | como separador
```


Fíjate en los nombres de los tanques de defensa; los estáticos, tienen un \$ con una \ delante. Por ejemplo, el nombre en el IL-2 para un BT7 estático es: **vehicles.artillery.Artillery\$BT7**, y tienes que añadir \ antes del \$, así: **vehicles.artillery.Artillery\\$BT7**

7. Comentarios

Seguramente lleva bastante tiempo hacer las plantillas, y tenemos que crear, editar y modificar varios archivos. Después de generar todos los archivos, tienes que comprobar que son correctos; por ejemplo, que se utilizan todos los aviones, que están todas las ciudades y los objetivos de suministro de las ciudades, que la imagen del frente y la imagen de suministros han sido hechas correctamente, etc.

Crearé más plantillas, las mismas que usaré en “**bellum**”: Kursk, Smolensk y Balaton. No haré más plantillas que esas. Todas las preguntas serán bien recibidas y seguro que ayudaré en la creación de cualquier otra plantilla. Al menos a aquellos que demuestren que han empleado tiempo en la comprensión de este manual (a pesar de mi mal inglés).

Usa el *Foro de Bellum* para preguntar (<http://il2fb-bellum.com.ar/foro>)* y solicitar ayuda con las plantillas que estés haciendo (si es posible suministra los archivos con los que estés trabajando).

Nota: Sé que los mapas pequeños de IL-2 funcionan con BADC, pero realmente desconozco si el generador/parser puede manejar mapas grandes, los que tienen 2 letras por Sector (Finlandia, Crimea, etc.).

* No existe ya esa página (*N. del T.*)

NOTAS FINALES (del Traductor):

*He dejado sin traducir algunas palabras, como “**campfire**” (hoguera) porque al trabajar en el **FMB** las encontraremos en inglés, con lo que será más fácil la identificación del objeto.*

*Otras, como “**script**” (“secuencia de comandos”), simplemente por comodidad.*

He reducido el tamaño de algunas de las imágenes para ahorrar espacio y poder colocarlas junto a su texto explicativo, pero se adjunta una carpeta con todas ellas a su tamaño original.

BADC sí funciona con mapas que tienen 2 letras por Sector. Ejemplo de ello es que ahora mismo en <http://war.by-airforce.com/> están volando una Campaña con el mapa de Crimea y tienen previsto que el siguiente sea Finlandia. Además de otras muchas modificaciones interesantes, tales como SUM por tierra y mar, misiones de reconocimiento, ataques aleatorios y sorpresivos a AFs,...

Gracias a E69_metaliving he comenzado a usar un editor de texto fantástico para estas tareas, se trata del “Komodo Edit” (<http://www.activestate.com/komodo-edit>)