

Structure des fichiers **Scenarios pour Orbiter**

Niveau I
(débutant)

Par Mustard

Novembre 2004 (mise à jour Fev 2005)



Retrouvez le forum francophone et de nombreuses documentations en français sur le site de Dansteph : <http://orbiter.dansteph.com/>

NIVEAU 1, pour débutant

Le but de ce document est principalement orienté vers 2 objectifs. Retrouver les causes d'un plantage d'un add-on et modifier un scénario de départ.

Ce document est le niveau 1 dédié aux débutants, on ne rentrera donc pas trop loin dans les détails, juste ce qu'il faut pour faire des petites manipulations et essayer de retrouver des causes basiques (mais souvent nombreuses) de plantage. Un niveau 2, plus expert, viendra compléter ce document.

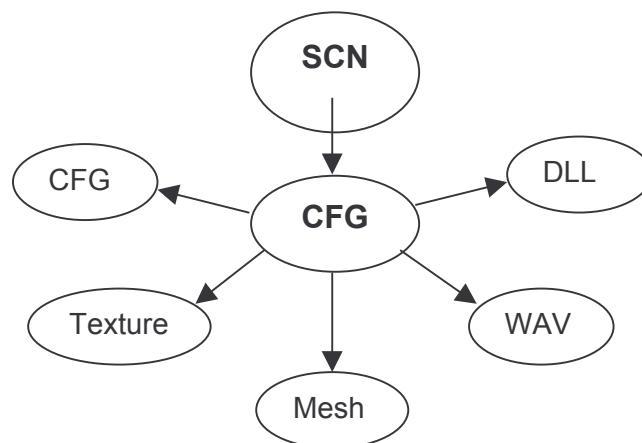
La puissance d'Orbiter réside principalement par sa capacité à recevoir des ajouts de décors, de fusées, d'objets, d'astres, de textures, etc. Ces ajouts sont généralement définies dans leur ensemble par le terme anglophone « add-ons ».

Il n'est pas rare que l'installation de certains de ces add-ons nécessitent des modifications manuelles dans d'autres fichiers ou fassent simplement planter Orbiter. De plus, si un paramètre ou module manque, Orbiter risque certainement de se planter irrémédiablement. Le but va donc être de rechercher le défaut qui nous amène à cet évènement. Cela donnera aussi l'occasion de créer ou modifier des scénarii afin de les adapter à notre volonté.

1) Architecture d'un add-on dans Orbiter

Un add-on est principalement constitué des fichiers suivants :

- .scn : celui-ci est le fichier de démarrage d'un scénario
- .cfg : contient des informations sur l'objet et les liens vers les textures et dll
- .dll : contient les animations ou programmes nécessaires à l'add-on
- .msh : contient les objets en 3D
- .tex : contient les textures
- .wav : contient les sons



Sur ce schéma on voit que le fichier Scenario envoie des liens vers un et même plusieurs fichiers *nom.CFG*. Ce dernier format contient des informations mais aussi des liens vers d'autres fichiers nécessaires. Selon le type d'add-on, la sophistication (avec animation ou avec des parties séparables), les liens se multiplieront.

2) Architecture des fichiers Scenarios de démarrage

Lorsqu'un scénario a été installé, Orbiter ira chercher à son lancement tous les fichiers `.SCN` placés dans le répertoire *scenario* d'Orbiter. A noter qu'une sauvegarde se présente exactement de la même façon.

Ce fichier est primordial car c'est de lui que va partir toute l'arborescence de la structure des fichiers d'un add-on comme vu précédemment.

Ce fichier scénario va contenir toute les informations de date, de positionnement dans le système solaire, de présence des divers objets (stations spatiales ou vaisseaux), de configuration MFD, des caméras, etc.

Pour en expliquer la structure nous allons prendre pour exemple le scenario du tableau suivant. Par la suite nous expliquerons plus en détail les paramètres possibles.

Un fichier scenario comporte plusieurs blocs commençant chacun par un Begin (début) et se terminant par un End (fin) suivi d'un titre. Cela permet de mieux s'y retrouver. C'est aussi une terminologie claire généralement employée en programmation.

L'ordre importe peu mais non retrouvons généralement Desc, Environnement, Focus, Camera, HUD, MFD left, MFD right, station (utilisé dans les ancienne version mais encore présent quelques rare fois), et Ship. Certains n'étant pas obligatoires.

BEGIN_DESC Launch Space Shuttle Atlantis. END_DESC	Descriptif de la mission. Dans le panneau de démarrage d'Orbiter vous verrez ce descriptif en cliquant sur le scenario. Vous pouvez y inscrire ce que vous voulez.
BEGIN_ENVIRONNEMENT System Sol Date MJD 51982.9069682177 END_ENVIRONNEMENT	L'environnement désigne le système solaire. Ici il renvoie au fichier <code>Sol.cfg</code> , qui décrira le contenu du système utilisé. En dessous vous avez la date selon le calendrier Julien. L'add-on « <code>MJDcalc</code> » vous permettra de la convertir en une date classique.
BEGIN_FOCUS Ship STS-101 END_FOCUS	Vous trouverez ici l'objet dont on a le contrôle au départ.
BEGIN_CAMERA TARGET STS-101 MODE Extern POS 2.89 -0.51 -70.39 TRACKMODE TargetRelative FOV 50.00 BEGIN_PRESET Cockpit:STS-101:40.00 Ground:STS-101:40.00:Earth -80.62339 28.61960 110.00 Track:STS-101:50.00:RELATIVE 2.890 -0.009 -1.229 Ground:STS-101:10.00:Earth -80.65298 28.58028 20.00 Track:ISS:40.00:RELATIVE 1.983 -0.474 -0.522 Track:Earth:40.00:GLOBAL 2.896 2.132 -0.331 END_PRESET END_CAMERA END_CAMERA	Vous avez ici les paramètres de la caméra (cible, mode de vue, position de la caméra en externe, type de vue en externe, et champ de vision) La zone PRESET correspond aux positionnements présélectionnés de la caméra virtuelle ou des vues extérieures observateurs.
BEGIN_HUD TYPE Surface END_HUD	Réglage du mode du viseur tête haute HUD
BEGIN_MFD Left TYPE Surface END_MFD	Réglage du MFD gauche avec les paramètres selon le mode utilisé.
BEGIN_MFD Right TYPE Orbit PROJ Ship REF Earth END_MFD	Réglage du MFD droit avec les paramètres selon le mode utilisé.

BEGIN_SHIPS	Début de la partie réservée aux vaisseaux
ISS:ISS STATUS Orbiting Earth RPOS 6190017.90 1969787.51 -1755348.12 RVEL 2669.244 -7069.642 1478.082 AROT 30.00 0.00 50.00 DOCKINFO 1:0,SH-15 NAVFREQ 0 0 END	Informations spécifiques au premier vaisseau. ISS de gauche étant le nom donné au vaisseau dans la partie, et ISS de droite étant le lien vers le nom du fichier ISS.cfg comportant toutes les infos et les liens de l'objet. S'en suit toutes les informations utiles de l'engin (référence, position, vecteur vitesse et rotation, ports, fréquences radios, etc.). Notez que le vaisseau Shuttle-15 est arrimé à ISS. 1:0 signifiant que le port 1 d'ISS est arrimé avec le port 0 du SH-15. La déclaration inverse doit être sur le bloc SH15
SH-15:ShuttleA STATUS Orbiting Earth PRPLEVEL 0:0.999 1:0.983 DOCKINFO 0:1,ISS NAVFREQ 578 0 DOCKSTATE 1 1.0000 PODANGLE 0.0000 0.0000 END	Informations sur le second vaisseau SH-15 étant son nom dans la partie et ShuttleA étant le nom du fichier ShuttleA.cfg où se trouve toutes les infos de cet engin. S'en suit le status (en orbite, au sol ou arrimé) la quantité de fuel, sa position en cas d'arrimage (ici le port 0 du ShuttleA est sur le port 1 d'ISS).
STS-101:Atlantis STATUS Landed Earth BASE Cape Canaveral:11 POS -80.6232502 28.6197342 HEADING 2.00 PRPLEVEL 0:1.000 1:1.000 2:1.000 NAVFREQ 0 0 CONFIGURATION 0 CARGODOOR 0 0.0000 GEAR 0 0.0000 KUBAND 0 0.0000 ARM_STATUS 0.5000 0.0000 0.0000 0.5000 0.5000 0.5000 CARGO_STATIC_MESH Carina_cradle CARGO_STATIC_OFS 0.000 -1.650 0.050 END	Ce vaisseau est nommé STS-101 dans la partie et va chercher le fichier Atlantis.cfg (dans Orbiter/config) (A noter qu'on peut trouver des liens du type STS-101 :shuttle/Atlantis . Dans ce cas le fichier sera dans le répertoire Orbiter/config/shuttle) Vous trouvez ensuite le statut (ici posée sur Terre), la base qui renvoie ici vers le fichier Cape Canaveral.cfg , sur le pas de tir 11 (voir fichier canaveral.cfg). Ensuite vous avez les infos orientation, fuel, radio, état des parties mobiles, etc. Certains paramètres peuvent être spécifiques à un engin, notamment pour les parties mobiles, ici par exemple on voit l'antenne Ku band, le bras, le train d'atterrissage, les portes de la soute, un petit satellite nommé carina accroché dans la soute cargo.
END_SHIPS	fin de la partie réservée aux vaisseaux

3) Le scenario plante

Orbiter est très sensible et ne pardonne pas beaucoup le moindre fichier ou paramètre manquant. S'il n'a pas ce qu'il veut ce sera probablement un retour sur le bureau de Windows (Crash To Desktop ou CTD en anglais).

Même si votre fichier scenario est cohérent, cela peut encore entraîner un plantage à son lancement. Dans ce cas cela est sûrement dû à un mauvais lien. Il faut donc remonter les filières des fichiers cfg appelés.

Dans notre exemple ci-dessus nous avons 4 appels, sol.cfg, ISS.cfg, Shuttle.cfg et Atlantis.cfg Les fichiers cfg se trouvent tous dans le répertoire Orbiter/config ou dans des sous-dossiers de ce répertoire.

A) Sol.cfg

Ce fichier informe Orbiter sur le système solaire utilisé pour la partie. Vous y trouverez toutes les planètes, les lunes, et les astéroïdes présents. Mais ne figurera ici que le classement hiérarchique de ces astres, et non pas les caractéristiques de leur orbite.

<pre> ; === Configuration file for solar system === Name = Sol Star1 = Sun Planet1 = Mercury Planet2 = Venus Planet3 = Earth Earth:Moon1 = Moon </pre>	<p>Descriptif Nom du système</p> <p>Star1 défini le soleil. Il renvoie vers le fichier sun.cfg pour en avoir les paramètres physique.</p> <p>Ensuite, nous avons la liste des planètes avec le lien vers le</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Structure des fichiers scénario pour Orbiter

Planet4 = Mars Mars: Moon1 = Phobos Mars: Moon2 = Deimos Planet5 = Jupiter Jupiter: Moon1 = Io Jupiter: Moon2 = Europa Jupiter: Moon3 = Ganymede Jupiter: Moon4 = Callisto Planet6 = Saturn Saturn: Moon1 = Mimas Saturn: Moon2 = Enceladus Saturn: Moon3 = Tethys Saturn: Moon4 = Dione Saturn: Moon5 = Rhea Saturn: Moon6 = Titan Planet7 = Uranus Uranus: Moon1 = Miranda Uranus: Moon2 = Ariel Uranus: Moon3 = Umbriel Uranus: Moon4 = Titania Uranus: Moon5 = Oberon Planet8 = Neptune Neptune: Moon1 = Triton Neptune: Moon2 = Proteus Neptune: Moon3 = Nereid Planet9 = 1 Ceres Planet10 = 2 Pallas Planet11 = 4 Vesta Planet12 = 10 Hygiea	fichier cfg correspondant. Par exemple Planet1 renvoie vers le fichier Mercury.cfg . Ce dernier fichier contient les informations physiques de l'astre ainsi que les bases et VOR) Pour définir une lune appartenant à une planète vous noterez l'architecture employée : <i>Nom de la planète/MoonX=nom du cfg de la lune</i> (Moon se traduisant par lune). On comprend malgré tout facilement cela en regardant ce tableau. A savoir qu'un astre tel qu'un astéroïde devra être défini comme planète. Vous pouvez définir autant de planètes et lunes que vous le souhaitez.
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Si vous installez un add-on ajoutant un astre dans votre système, il vous suffira de l'ajouter dans cette liste en incrémentant le dernier numéro des planète ou lunes.

Vous remarquerez souvent que sol.cfg n'est pas celui utilisé dans certain scénario. C'est généralement le cas pour des add-ons passés ou fictifs n'ayant pas les mêmes bases au sol.

Par exemple, sur les missions Mercury, Gemini ou Apollo vous trouverez souvent des sol_1962.cfg ou sol_1975.cfg. Cela parce que la base Cap Canaveral de l'époque n'était pas la même qu'aujourd'hui. En effet les pad39 de la navette par exemple n'existaient pas en 1962. Donc il faut un autre fichier pour la base, avec des bâtiments en moins ou en plus. Par conséquent le scénario renvoie vers un autre fichier sol qui renverra à son tour vers d'autres fichiers bases (avec d'autres bâtiments et pas de tir).

B) Iss.cfg

Ce fichier donne certaines informations sur le vaisseau nommé « ISS ».

<pre> ; === Configuration file for the International Space Station === Meshname = ISS Mass = 450e3 Size = 55.0 Inertia = 395 559 290 CrossSections = 897.2 3641.3 610.8 EnableFocus = FALSE EnableXPDR = TRUE XPDR = 466 ; === Docking ports === BEGIN DOCKLIST 13.67 5.56 0 1 0 0 1 0 588 -4.32 17.6 0 0 1 0 1 0 578 -36 4.21 0 -1 0 0 0 -1 0 568 END DOCKLIST </pre>	Descriptif Meshname renvoie vers le fichier ISS.msh (modélisation 3d de la station). Nous trouvons ensuite quelques paramètres physiques de l'engin. Le XPDR est sa fréquence radio utilisé lors des poursuites pour un arrimage. EnableXPDR doit être sur True (Vrai) pour être actif Le XPDR commence à 108.00 KHz et s'incrémente de 0.05 KHz. C'est-à-dire qu'on a XPDR=0 pour 108 KHz et XPDR=1 pour 108.05 KHz, XPDR=2 pour 108.10 KHz, etc. XPDR=0 à 639 Ensuite nous avons les fréquences des ports d'arrimage selon le même principe que précédemment. 108KHz=0. Ici le port 1 à 588 donne 588x0.05=29.4KHz soit 108+29.4=137.40KHz. Les autres paramètres étant les positions des ports par rapports à
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	l'engin
--	---------

C) ShuttleA.cfg

Ce fichier donne les informations sur le vaisseau. Dans ce cas là, il n'y a pas grand-chose car il s'agit d'un engin de base d'Orbiter

<pre> ; === Configuration file for vessel class ShuttleA === ClassName = ShuttleA Module = ShuttleA </pre>	Descriptif Type d'engin (définie la classe selon la taille dans Orbiter). Ce paramètre n'est pas important. La dernière ligne renvoie vers le fichier ShuttleA.dll du répertoire Orbiter/module. Ce fichier comporte la programmation de celui-ci particulièrement pour les parties mobiles, le comportement, la fumée, etc. Cette dll est inaccessible en lecture car elle a été compilée.
------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Cependant, dans de nombreux add-ons vous trouverez ici de nombreuses informations utiles s'il n'y a pas de lien vers un fichier dll. Mais à ce stade il est difficile de les décrire car c'est très varié et dépendant de l'auteur. Chacun faisant un peu à sa manière. Ainsi vous pourrez trouver des infos sur la position du port d'arrimage, du fuel, du comportement dans l'atmosphère.

Le fichier Atlantis.cfg est de même structure que ce dernier, je ne l'expliquerai donc pas.

D) Les bases et les décors au sol

L'une des causes d'un plantage réside souvent dans les bases ou les décors. Dans notre scénario de départ, on voit que l'un de nos vaisseaux, la navette Atlantis nommée STS101, est positionnée sur une base, au sol. Cette ligne donnait " BASE Cape Canaveral:11".

Cela signifie que la navette est positionnée sur cette base. De plus cette base possède des pas de tir et notre navette est positionnée sur le n°11.

Cette ligne se réfère au fichier Canaveral.cfg. Vous verrez sur la seconde ligne de ce fichier que le nom de cette « surface base » est « Cape Canaveral ». Après le chiffre 11 correspond au 11^{ème} bloc LPAD qui est un bloc de pas de tir.

Certains add-ons autres que des bases permettent d'ajouter des décors sous forme de relief du sol et objets tel que des cailloux. C'est le cas des sites historique Apollo ou martiens.

En fait un décor vallonné est comme un bâtiment mais au lieu d'être cubique il est plat et vallonné. Idem pour les cailloux ou rochers.

Donc ces décors sont à considérer comme des bases. Il est important de devoir les configurer manuellement lors de l'installation.

Il faut savoir qu'il est **PRIMORDIAL** de lire les fichiers généralement nommés « readme » (lisez moi) afin d'avoir la procédure d'installation et surtout les paramètres de positionnement géographique du décor.

Prenons un exemple. Vous télécharger l'add-on Tranquilitybase, afin d'avoir le décor de la mission Apollo11.

Vous devez extraire les fichiers comme vous le faites avec tout autre add-on. Mais cela ne suffira pas. Il faut aller ajouter à la main ce nouvel élément dans le fichier de la planète utilisé. Ici moon.cfg (la lune)

Dans ce fichier vous trouvez une partie nommée « ;==Surface bases== », c'est en dessous que vous devrez ajouter votre décor à la liste déjà présente.

Ajoutez donc dans cette liste le nom et les coordonnées du décor que vous trouverez dans le fichier nommé généralement « readme.txt »

Ex :

Avant
; === Surface Bases ===
Brighton: -33.4375 +41.125
Procellarum: -23.41930 -3.01381

Après
; === Surface Bases ===
Brighton: -33.4375 +41.125
Procellarum: -23.41930 -3.01381
Tranquillity : +23.4333 +0.6875

Les indications derrière votre base sont l'emplacement géographique du décor ou de la base (longitude / latitude) que vous aurez dans le readme de l'add-on. Vous avez donc la possibilité de la bouger en modifiant ces valeurs.

E) Les cas particuliers de plantage

Malheureusement il existe encore d'autres raisons de plantage mais ils sont particulièrement complexes à trouver et j'avoue que parfois je ne trouve pas moi même la solution.

Cependant il faut savoir que certains add-ons sont structurés autour des modules Multistage et Spacecraft de Vinka ou de la plus récente bibliothèque CVE et CVElite. Il s'agit d'une librairie C++ d'aide à la création d'add-ons utilisée pour les séparations, les parties mobiles, etc. Malheureusement ces add-ons réalisés avec les modules de Vinka ont un inconvénient majeur, il ne permettent pas de reprendre une sauvegarde lorsque le vaisseau ou la capsule n'a pas été totalement séparé du lanceur. Ainsi si vous sauvez alors que votre capsule est encore sur l'étage 3 de la fusée la partie ne sera pas récupérable.

Attention, lorsque vous quittez une partie (pour modifier un paramètre de base d'Orbiter, comme le fuel illimité par exemple) et que vous revenez dans la partie cela correspond à une reprise de sauvegarde. En effet, dans une sortie d'une partie, le programme sauvegarde la partie dans (*Current state*).scn, puis en revenant avec le bouton Orbiter il recharge en fait ce fichier. Et dans notre exemple vous aurez un beau plantage.

4) Conclusion

Voici donc un petit aperçu rapide et pas trop détaillé de la structure des fichiers scenarios d'Orbiter.

Je suis d'accord avec vous que tout cela fait assez fouillis et que cela aurait pu être un peu plus structuré car on s'aperçoit vite en installant plusieurs add-ons que ça devient vite le bazar. Surtout que les auteurs ne respectent pas toujours les règles, certains vont s'installer des répertoires supplémentaires alors que d'autres utiliseront la racine de Config, certains vont utiliser des fichiers .ini et d'autres des .dll, etc. L'idéal aurait été une structure stricte comme Flight Simulator.

A présent j'espère que cela aura permis d'éclaircir l'architecture utilisée par Orbiter et que les débutants y verront un peu plus clair.

Pour les plus expérimentés ou ceux qui en veulent plus il faudra se référer au document de niveau II ou à la doc officielle qui ne détaille cependant que certains paramètres.

J'espère que vous me pardonneriez si des fautes ou des incompréhensions persistent. Dans ce dernier cas, je vous conseille de vous rendre sur le forum francophone d'Orbiter où vous pourrez poser toutes vos questions qui recevront assurément une réponse.

--
Mustard

