Structure des fichiers Scenarios pour Orbiter

Niveau II
(expert)

Par Papyref

Mars 2005

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

1 – BUT DE CE DOCUMENT

Le présent document a pour objet de donner quelques "recettes" pour vous permettre de réaliser vos scénarios et en particulier de lancer des satellites divers avec des lanceurs divers.
Il complète le document de niveau 1 pour débutant écrit par Mustard qu'il est bon de lire au préalable

2 – OBJETS AU SOL OU EN ORBITE

Si dans un scénario, on a besoin de définir des appareils au sol et des vaisseaux ou satellites en orbite, voilà comment procéder

2.1 – Définir un appareil au sol

Un appareil au sol est facile à définir dans le scénario, à condition bien sûr qu'il soit désigné dans le dossier CONFIG par un fichier <NOM>.cfg comme par exemple Deltaglider.cfg

Il suffit de situer sa position dans le fichier scénario en le nommant dans la rubrique BEGIN_SHIP …… END_SHIP par l'APPELLATION qu'on veut lui attribuer dans ce scénario.
L'appellation peut être differente du nom ou identique suivant son goût

Exemple ci-dessous ou un DeltaGlider appelé GL-01est placé à Cape Canaveral avec un cap de départ à 150°
Rappelons que le cap (Heading) est défini de 0° à 360° en tournant dans le sens des aiguilles d'une montre le nord étant à 0°

```
BEGIN_SHIPS
GL-01:DeltaGlider              ;== appellation suivie de deux points et du nom du fichier cfg
  STATUS Landed Earth
  BASE Cape Canaveral:1
  HEADING 150.00
  PRPLEVEL 0:1.000 1:1.000
  NAVFREQ 402 94 0 0
  XPDR 0
  NOSECONE 0 0.0000
  GEAR 1 1.0000
  AIRLOCK 0 0.0000
END
END_SHIPS
```

GL-01 est l'appellation que l'on souhaite donner à l'appareil et DeltaGlider le nom de son fichier cfg de configuration.
Le nom peut être quelconque (vous pouvez l'appeler Toto si ça vous chante) et c'est celui qu'on retrouvera pour nommer l'appareil dans les rubriques définissant le type de visualisation BEGIN_FOCUS …END_FOCUS et BEGIN_CAMERA….END_CAMERA (notons qu'il n'est pas obligatoire que ces rubriques soient remplies à la conception d'un scénario)

La position est définie en général par **BASE <Nom de la base>** éventuellement suivi de deux points et d'un numéro donnant l'emplacement d'un pad particulier

La position peut aussi être définie par **BASE <Nom de la base>** suivi d'une ligne
POS <Coordonnées> qui situe plus précisément l'emplacement sur la base

On peut utiliser uniquement POS mais il n'y a aucun détails particulier à l'emplacement du tir à part des détails éventuels du terrain qui auraient été prévus à ces coordonnées.

Par exemple pour un tir Apollo ou le Pad est spécifique sur la base de Cape Canaveral on écrit:

```
AS-506:saturn5nasp
  STATUS Landed Earth
  BASE Cape Canaveral:11
  POS -80.6232502 28.6197342
  HEADING 90.0
```

Voilà le scénario minimum pour un DeltaGlider au départ de Cape Canaveral. (nous verrons plus loin comment choisir la date)

```
BEGIN_DESC
Delta-glider, au départ à Kennedy Space Center
END_DESC

BEGIN_ENVIRONMENT
  System Sol
  Date MJD 51981.6008401528
END_ENVIRONMENT

BEGIN_FOCUS
END_FOCUS

BEGIN_CAMERA
END_CAMERA

BEGIN_MFD Left
END_MFD

BEGIN_MFD Right
END_MFD

BEGIN_PANEL
END_PANEL

BEGIN_SHIPS
GL-01:DeltaGlider
  STATUS Landed Earth
  BASE Cape Canaveral:1
  HEADING 150.00
  PRPLEVEL 0:1.000 1:1.000
  NAVFREQ 402 94 0 0
  XPDR 0
  NOSECONE 0 0.0000
  GEAR 1 1.0000
  AIRLOCK 0 0.0000
END
END_SHIPS
```

Une fois lancé un tel scénario, il suffit de régler les vues extérieures et intérieures comme on le souhaite (ouvrir des MFD, zoomer…) et de le sauvegarder pour avoir le scénario définitif ce qui donne par exemple le résultat suivant.

```
BEGIN_DESC
Delta-glider, au départ à Kennedy Space Center
END_DESC

BEGIN_ENVIRONMENT
  System Sol
  Date MJD 51981.6016465764
END_ENVIRONMENT

BEGIN_FOCUS
  Ship GL-01
END_FOCUS

BEGIN_CAMERA
  TARGET GL-01
  MODE Cockpit
  FOV 50.00
END_CAMERA

BEGIN_HUD
  TYPE Surface
END_HUD

BEGIN_MFD Left
  TYPE Surface
  SPDMODE 1
END_MFD

BEGIN_MFD Right
  TYPE Map
```

```
  REF Earth
  BTARGET Cape Canaveral
  TRACK ON
END_MFD


BEGIN_SHIPS
GL-01:DeltaGlider
  STATUS Landed Earth
  BASE Cape Canaveral:1
  POS -80.6758964 28.5227640
  HEADING 150.00          ;=== azimuth (cap) de lancement
  PRPLEVEL 0:1.000 1:1.000
  NAVFREQ 402 94 0 0
  XPDR 0
  GEAR 1 1.0000
END
END_SHIPS
```

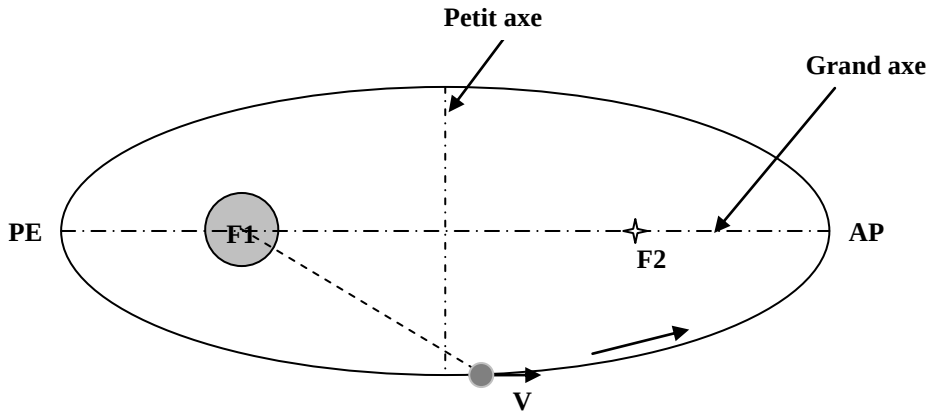
L'azimuth de lancement est important pour placer le vaisseau dans un plan orbital d'inclinaison donnée et réduire au minimum la dépense en carburant ultérieure pour aligner les plans.
Pour les adeptes du calcul voici une formule permettant de calculer cet azimuth en fonction de la latitude de la base de lancement

Azimuth de lancement = arcsin(cos(Inc désirée)/cos(latitude de la base))

2.2 – Définir un appareil en orbite autour d'un corps de référence

C'est un peu plus difficile que pour un appareil posé
Nous supposerons que l'orbite est une trajectoire fermée répétitive de type elliptique ce qui est le cas qui nous intéresse en général

Une orbite de type elliptique possède deux axes de symétrie et deux foyers dont un foyer F1 est occupé par le corps de référence autour duquel gravite le vaisseau ou le satellite à la vitesse V (vitesse tangentielle sur son orbite) L'Apoapsis (AP) et le Periapsis (PE) sont les extrémités du grand axe.



En prenant les notations classiques dans Orbiter, on a :

PeR (Periapsis) = longueur du segment PE-F1
ApR (Apoapsis) = longueur du segment F1-AP

La longueur du grand axe PE-AP = PeR + ApR et par conséquent la valeur du demi grand axe que nous appellerons **SMa** sera:

(1) SMa = (PeR + ApR) / 2

on appelle excentricité de l'orbite la valeur Ecc telle que:

(2) Ecc = (ApR-PeR)/(ApR+PeR)

Si Ecc est nul, l'orbite est circulaire. Plus Ecc s'approche de 1 et plus l'orbite est allongée

Dans Orbiter, on peut définir les paramètres de l'orbite en inscrivant dans le scénario une ligne commençant par ELEMENTS suivi de 7 nombres séparés par un espace (nombres à l'anglo saxonne ou le point représente une virgule) Par exemple :

ELEMENTS 6734919 0.0009 74.513 169.034 328.305 122.0 51544.5

1er nombre Demi grand axe SMa de l'orbite en mètres
2ème nombre Excentricité Ecc
3ème nombre Inclinaison Inc en degrés du plan orbital **par rapport au plan ecliptique**
4ème nombre Longitude **LAN du nœud ascendant en degrés** par rapport au point vernal
5ème nombre Longitude LPe du Periapsis en degrés par rapport à AN
6ème nombre Longitude moyenne à la date de référence en degrés (joue au 1/1000)
7ème nombre Date de référence en MJD

Les trois premiers nombres définissent la forme de l'orbite
Les quatrième et cinquième nombres définissent sa position par rapport à un système d'axe de référence en fixant la longitude du point nodal ascendant et la longitude du Periapsis (voir ma note "Naviguer dans l'espace" page 8)
Les sixième et septième nombres définissent la position du vaisseau sur l'orbite à une date donnée.

Pour construire un scénario nous nous intéressons de façon pratique à trois choses:

- La taille de l'orbite (SMa et Ecc la donneront)
- L'inclinaison de l'orbite par rapport au plan équatorial (par rapport auquel sont situées les bases sur la carte) et non par rapport au plan écliptique (plan de l'orbite de la planète par rapport au soleil)
- Le sens de parcours de l'orbite prograde (Ouest vers Est est ce que l'on a normalement puisqu'on lance vers l'est) ou rétrograde (Est vers Ouest que l'on utilise pour l'insertion lunaire par exemple)

Aucune planète n'a son axe de rotation perpendiculaire à son plan orbital (voir ma note "Naviguer dans l'espace" page 5)
Par exemple la Terre à un axe dont l'inclinaison est de l'ordre de 23°27 actuellement. Il varie un peu dans une très longue période en raison de la précession de l'axe.
Si l'on veut définir une orbite **par rapport au plan équatorial de la planète**, il faut donc compenser la valeur de l'inclinaison Inc donnée dans ELEMENTS par une valeur correspondant à l'inclinaison de l'axe de rotation que l'on appellera Inax

Pour des raisons pratiques de calcul nous allons fixer toujours le quatrième nombre à 180 ° ce qui nous permettra de compenser Inc en ajoutant la valeur de Iax (en fait, pour les connaisseurs, on place le point nodal ascendant dans une direction opposée au point vernal)

Si on désire une orbite prograde (Ouest vers Est)

- On prend Inc = Inax si on veut décrire une orbite équatoriale
- On prend Inc = Inax+90 si on veut décrire une orbite circumpolaire
- On prend Inc = Inax + latitude si on veut pouvoir passer au dessus d'un point ayant cette latitude

Si on désire une orbite rétrograde (Est vers Ouest)

- On ajoute 180 aux valeurs calculées précédemment

Faisons une petite application !
Nous voulons placer un vaisseau autour de la Lune à 50 km d'altitude minimum et 100 km maximum et avec une inclinaison du plan orbital pour passer au dessus de Fra Mauro qui est à 3°65 Sud
Comme la Lune a un rayon moyen égal à 1738000 m. On devra donc avoir (en mètres)

PeR =1738000+50000 = 1788000
ApR=1738000+100000=1838000

La formule **(1)** ci avant va nous donner comme valeur du demi grand axe (en mètres)

SMa = (1788000 + 1838000) / 2 = 1813000

La formule **(2)** va nous donner comme valeur de l'excentricité :

Ecc = (1838000 – 1788000) / (1838000 + 1788000) = 50000/3626000 = 0.013789

L'inclinaison de l'axe lunaire est de 1°54 donc comme on veut passer dans le sens rétrograde il faut prendre Inc = 1.54 + 3.65 + 180 = 185.19

Nous prendrons ELEMENTS 1813000.0 0.013789 185.19 180.0 dans notre scénario

```
BEGIN_DESC
Delta-glider, au départ à Kennedy Space Center
END_DESC

BEGIN_ENVIRONMENT
  System Sol
  Date MJD 51981.6008401528
END_ENVIRONMENT

BEGIN_FOCUS
END_FOCUS

BEGIN_CAMERA
END_CAMERA

BEGIN_MFD Left
END_MFD

BEGIN_MFD Right
END_MFD

BEGIN_PANEL
END_PANEL

BEGIN_SHIPS
GL-01:DeltaGlider
  STATUS Orbiting Moon
  ELEMENTS 1813000 0.013789 185.19  180.0  ;=== 180 obligatoire comme on l'a vu
  PRPLEVEL 0:1.000 1:1.000
  NAVFREQ 402 94 0 0
  XPDR 0
  NOSECONE 0 0.0000
  GEAR 1 1.0000
  AIRLOCK 0 0.0000
END
END_SHIPS
```

Comme le voit, il n'y a aucune information sur la vitesse du Deltaglider dans le scénario.
Heureusement Orbiter va travailler pour nous !
Si nous lançons notre scénario puis nous le sauvegardons le miracle se produit. En ouvrant la sauvegarde, voilà ce que l'on obtient pour notre Deltaglider

```
BEGIN_SHIPS
GL-01:DeltaGlider
  GL-01:DeltaGlider
  STATUS Orbiting Moon
  RPOS 1158798.46 -124203.31 1367408.71
  RVEL 1256.831 98.003 -1078.959
  AROT 0.00 0.00 0.00
  PRPLEVEL 0:1.000 1:1.000
```

```
  NAVFREQ 402 94 0 0
  XPDR 0
  GEAR 1 1.0000
END
END_SHIPS
```

Orbiter a transformé les données de ELEMENTS en informations de position RPOS, de vitesse sur l'orbite RVEL et de rotation éventuelle du vaisseau sur lui-même AROT

Si l'on veut aller au plus simple on peut se placer sur une orbite circulaire ou SMa =ApR=PeR et on écrira alors dans notre scénario initial

ELEMENTS <valeur du rayon de l'orbite souhaité> 0.0 <inclinaison de l'orbite> 180.0

Si on lance le scénario, la sauvegarde contient les informations complètes de vitesse et rotation sur lui-même du vaisseau

Exemple:

On veut réaliser une orbite terrestre équatoriale géostationnaire
Le rayon doit être de 42.164M
L'axe terrestre est incliné de 23.27° par rapport au plan de l'écliptique qui correspond à une inclinaison nulle. Si on veut se trouver dans un plan équatorial on doit prendre Inc= 23.27
On écrira dans le scénario ELEMENTS 42164000 0.0 23.27 180

```
BEGIN_SHIPS
GL-01:DeltaGlider
  STATUS Orbiting Earth
  ELEMENTS 42164000 0.0 23.27 180.0
  PRPLEVEL 0:1.000 1:1.000
  NAVFREQ 402 94 0 0
  XPDR 0
  NOSECONE 0 0.0000
  GEAR 1 1.0000
  AIRLOCK 0 0.0000
END
END_SHIPS
```

Et on obtient une orbite géostationnaire comme vous pouvez le vérifier en faisant l'essai

3 – REGLAGE DU TEMPS

Exprimé en MJD (**M**odified **J**ulian **D**ay) qui est un système de datation issue du calendrier Julien.
Le calendrier Julien a pour origine le 1er janvier – 4712 (4713 avant JC) à midi.

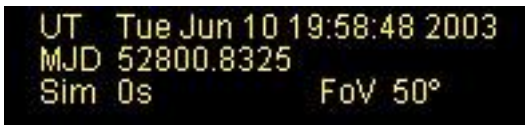
Vers 1950 les scientifiques de l'espace ont décidé d'introduire le calendrier MJD en adoptant la règle suivante MJD = JD(Jour Julien) – 2400000,5 ce qui fait commencer le jour MJD à 0h ce qui est plus pratique.
Le calendrier MJD commence en conséquence le 17/11/1858 à 0h.

L'année solaire ou année tropique compte 365,242217 jours. Elle est donc plus longue que l'année calendaire de 365 jours. Pour cette raison, pour se recaler par rapport au soleil on prend une année bissextile calendaire de 366 jours tous les 4 ans pour les années divisible par 4 sans donner de reste..
Comme cela compense un peu trop puisqu'on ajoute 1 jour au lieu de 0,968868 sur 4 ans on supprime une année bissextile toutes les années divisibles par 100 sauf si elles sont divisibles par 400 (pour cette raison 2000 était bissextile)

Année	2000	2001	2002	2003	2004	2005	2006	2007
MJD	51544	51910	52275	52640	53005	53371	53736	54101

On ajoute une partie décimale au jour MJD pour préciser une heure dans la journée en donnant la fraction du jour écoulé depuis 0h

Dans Orbiter le temps est exprimé de deux manières comme on peut le voir sur le HUD



- Date et temps UT qui s'écrit : Jour, mois, date, heure, minute, seconde, année
- En temps MJD exprimé par un nombre dont la partie entière est le nombre de jour calculé depuis le 17/11/1858 et la partie décimale donne la partie du jour écoulée depuis 0h (minuit)

De façon pratique la partie décimale donne le pourcentage à prendre sur 86400 secondes pour avoir le temps passé depuis minuit

Si on prend l'exemple précédant:

52800 correspond au mardi 10 juin 2003 et 0,8325 représente la fraction du jour écoulé.
Une journée vaut 86400 seconde donc il s'est écoulé 86400 x 0,8325 = 71928 secondes
Une heure fait 3600 secondes d'où 71928s = 19h + 3528s
Une minute fait 60s d'où 71928s = 19h + 58mn + 48s
Nous retrouvons bien l'heure UT affichée

ON RETIENDRA QUE:

La date MJD est celle qui figure dans un scénario en rubrique Environment

```
BEGIN_ENVIRONMENT
  System Sol
  Date MJD 52800.8325
END_ENVIRONMENT
```

La partie entière est le nombre de jours écoulés depuis la mise en œuvre du calendrier Julien le 17/11/1858

La partie décimale est en général exprimée à 4 décimales, ce qui donne une précision de moins de 9 secondes mais on peut utiliser jusqu'à 10 décimales pour une précision de l'ordre de 1/10000 de seconde ce qui pratiquement n'a pas d'intérêt pour nous si nous réalisons un scénario.

De façon simple retenons qu'avec deux décimales la une précision est de 14 minutes environ avec trois décimales elle est de 86 secondes et avec trois décimales elle est de 8,6 secondes.

Retenons également de façon simple que 25% = 6h du matin; 50% = 12h (midi) 75% = 18h et 0% = 0h (minuit)

Par exemple 52800.25 correspond au 10 juin 2003 à 6h du matin et 52800,50 à 12h

On peut trouver sur Internet des petits programmes de calcul réalisant la conversion MJD vers Date et vice versa.

4 – CHARGES A LANCER

4.1 – Types de charge

Il y a dans Orbiter deux sortes de charges à lancer :

- Les charges simples comprenant un seul élément (HST par exemple)
- Les charges composites comprenant un ou plusieurs éléments (Huygens, Galileo par exemple)

Les charges peuvent également avoir des animations comme déploiement de panneaux solaires, largage de parachutes…etc

Elles sont lancées par deux types de lanceurs:

- Les fusées (Ariane, Delta, Proton, Soyouz…) qui embarquent en général des satellites civils ou militaires pour les placer sur orbite
- La navette qui embarque soit des satellites soit des éléments de construction orbitale pour les placer en orbite avec possibilité de manipulation par un bras

4.2 - Définition des charges

Les charges sont appelées **Payload** dans les textes et fichiers. Elles se divisent en plusieurs catégories.

4.2.1 - Charge simple sans animation (exemple satellite Carina)

Il lui correspond un fichier de configuration dans le dossier CONFIG, qui est dans ce cas Carina.cfg. Ce fichier se présente comme suit

```
; === Configuration file for vessel class ESA Carina ===
ClassName = carina
MeshName = carina
Size = 5.0

Mass = 5000; empty mass [kg]
MaxFuel = 0  ; max fuel mass [kg]
Isp = 0      ; fuel specific impulse [m/s]

MaxMainThrust = 0
MaxRetroThrust = 0
MaxHoverThrust = 0
MaxAttitudeThrust = 5e2
COG_OverGround = 2.0
CameraOffset = -.715 .865 -2.5
CW = 10 10 5
LiftFactor = 0.0
CrossSections = 2.45 6.25 2.45

; === Attachment specs ===
BEGIN_ATTACHMENT
P 0 0 0  0 -1 0  0 0 1  XS      ←-------- -point d'attache avec le lanceur
P 0 0.9 0.1  0  1 0  0 0 1  GS  ←-------- point d'ancrage par un bras
END_ATTACHMENT
```

Le fichier comprend:

a) les éléments minimum permettant à Orbiter d'appeler la charge

ClassName = nom attribué à la classe de ce type de module
MeshName = mesh attribué au module pour sa représentation graphique
Size = facteur d'échelle pour la représentation graphique (optionnel)

b) les éléments définissant complètement la charge:

   Série de données pour préciser les masses, les caractéristiques éventuelles des moteurs, les caractéristiques géométriques et dynamiques …etc

c) les éléments "Attachment specs" qui définissent le point d'arrimage XS avec le lanceur et le (ou les) point(s) d'ancrage possible GS par un bras (il peut y en avoir éventuellement plusieurs)
   Nous verrons plus loin comment définir les points XS et GS

Dans certain cas une telle charge peut posséder en plus des ports de docking qui lui permettront de s'arrimer à d'autres modules qui en possèdent pour créer des assemblages.
Le fichier de configuration contient alors une liste des ports de docking

Par exemple voilà le fichier de configuration d'un module à deux ports d'arrimage participant à l'assemblage d'une station.

```
; === Configuration file for the International Space Station ===
Classname = ttmod1
Meshname = ttmod1
Mass = 8e3
Size = 6.0
Inertia = 7.7 7.7 2.3

; === Docking ports ===    points d'arrimage inter modules
BEGIN_DOCKLIST
0 0  5   0 0  1    0 1 0
0 0 -5   0 0 -1    0 1 0
END_DOCKLIST

; === Attachment specs ===
BEGIN_ATTACHMENT
P 0 0 0 1 0 0 0 0 1 XS
P -2.3 0 0 1 0 0 0 0 1 GS
END_ATTACHMENT
```

On voit la rubrique supplémentaire Docking ports et on peut remarquer que ce module n'a pas de moteur. Il sera positionné et arrimé par le bras de manipulation à l'assemblage.

4.2.2 - Charge simple avec animation (exemple HST)

HST comprend des animations comme l'ouverture et le déploiement de panneaux solaires.
En conséquence, le fichier de configuration fait appel à un fichier module HST STS-109.dll qui se trouve dans le dossier MODULE d'Orbiter et qui est le programme de gestion des animations.

```
; === Configuration file for HST Hubble Space Telescope ===
ClassName = HST
Module = HST_STS-109
MeshName = HST_STS-109

MaxFuel = 1000
ISP= 500000000
MaxAttitudeThrust = 1250
Inertia = 15.30 15.47 1.85
CrossSections = 42.74 45.66 17.31

; === Attachment specs ===
BEGIN_ATTACHMENT
P  0.000  0.0 -6.3   0 0 -1  0 -1 0  XS ; BAPS Service Mission latch
P  0.575 -1.9 0.0  0 -1 0  0 0 -1  GS ; RMS fixture 1
P -0.423 -1.9 0.0  0 -1 0  0 0 -1  GS ; RMS fixture 2
END_ATTACHMENT
```

On voit dans ce cas que deux points GS sont prévus pour l'ancrage du grappin du bras de la navette.
Pas de points de docking puisque HST est toujours seul

4.2.3 - Charge composite

Une telle charge est par principe toujours animée puisqu'elle se compose de plusieurs parties qui se détacheront successivement les unes des autres.
Sa représentation et son animation vont faire appel à un module logiciel spécifique appelé par le fichier **<nom de la charge>.cfg** situé dans le dossier de configuration.
Deux cas peuvent se présenter :

- Le programmeur a intégré dans un seul fichier **<nom>.dll** la gestion complète de la charge (représentation,animation, algoritmes…)
  Le fichier de configuration de la charge fait appel directement à ce module qui se situe dans le dossier MODULE

Par exemple voilà le contenu du dossier Saturn5nasp.cfg pour Apollo

```
; === Configuration file for vessel class Saturn 1B ===
ClassName = saturn5Nasp      ;=== nom de classification
Module = saturn5Nasp         ;=== appel au fichier Saturn5nasp.dll dans le dossier MODULE
EnableFocus = FALSE
Help = NCPP,NCPP
```

Les deux premières lignes sont obligatoires.
Enable Focus = TRUE permet si on le désire de pouvoir choisir et contrôler l'élément correspondant en ouvrant le tableau de choix par F3

- Le programmeur utilise pour la gestion et l'animation un ensemble de fonctions préprogrammées contenues dans des fichiers types spacecraft.dll et multistage.dll développés par VINKA, qui sont contenus dans le dossier MODULE
  Le fichier de configuration de la charge fait appel indirectement à un de ces modules par l'intermédiaire en général d'un fichier de type <nom de la charge>.ini situe dans un sous-dossier SPACECRAFT du dossier CONFIGURATION

Prenons par exemple le satellite Cassini :
Dans le dossier de configuration il va y avoir un fichier de configuration très simple

```
;===Satellite Cassini
classname = Cassini      ;=== nom de classification
Module = spacecraft      ;=== appel au dossier Spacecraft
```

Classname= Cassini va permettre d'appeler le fichier Cassini.ni situé dans le dossier Spacecraft qui définira complétement Cassini et ses données d'animations et l'execution se fera par spacecraft.dll.

Attention, le nom Classname doit être le même que celui du fichier ini du dossier Spacecraft

Un fichier de type peut s'ouvrir avec un éditeur de texte et les données peuvent éventuellement être modifiées
Voilà le fichier Cassini.ini contenu dans le dossier Spacecraft

```
[CONFIG]
MESHNAME="Cassini"
SIZE=10
EMPTY_MASS=2199
FUEL_MASS=3132
MAIN_THRUST=445
HOVER_THRUST=445
RETRO_THRUST=0
ATTITUDE_THRUST=100
ISP=3091.9
TRIM=0.05
PMI=(18.91,12.16,14.40)
CW_Z_POS=.5
CW_Z_NEG=.5
CW_X=.5
CW_Y=.5
CROSS_SECTION=(15.67,13.3,16.81)
COG=.5
```

```
PITCH_MOMENT_SCALE=0.00005
BANK_MOMENT_SCALE=0.00005
ROT_DRAG=(.5,.5,.5)
LAND_PT1=(0,-1,3)
LAND_PT2=(-3,-1,-3)
LAND_PT3=(3,-1,-3)

[EX_MAIN_0]
OFF=(.1,.25,-2)
DIR=(0,0,-1)
LENGTH=.75
WIDTH=.1

[PAYLOAD_0]
MESHNAME="huygens"
OFF=(-1.5,0,.25)
MODULE=huygens
NAME=huygens
MASS=319
SPEED=(-1,0,0)
ROT_SPEED=(0,0,0)
```

Une rubrique [CONFIG] contient les paramètres définissant le module de base:

- Meshname donne **entre guillemets** le nom et le chemin du fichier Mesh utilisé (en général il est dans le dossier MESHES d'ou pas de précision supplémentaire sur le chemin)

Attention : Un mauvais chemin ou un fichier Mesh inexistant ou nul sont des causes assez fréquentes de plantage pour les Add-ons

- Les informations de masse (à vide, carburant emporté…)
- les informations sur les caractéristiques des moteurs éventuels (type, poussée…)
- les informations géométriques et aérodynamiques (coefficients de traînées, moment d'inertie, points d'appui…)

Une ou plusieurs rubriques peuvent donner ensuite des informations complémentaires précisant les positionnements de certains points et en particulier des points de docking si il y en a et les séquences d'animation

Enfin et c'est le plus important, une rubrique PAYLOAD donne tous les éléments de définition pour la deuxième partie de l'élément qui est considérée comme charge embarquée dans l'élément de base:

- **MESHNAME** donne le nom et le chemin du fichier Mesh utilisé (en général il est dans le dossier racine MESHES d'ou pas de précision supplémentaire sur le chemin)
- **MODULE** donne le nom et le chemin du fichier module utilisé (en général il est dans le dossier racine MESHES d'ou pas de précision supplémentaire sur le chemin)
- **NAME** donne le nom de la charge désignée correspondant à module
- Ensuite on trouve des éléments définissant entre autre la masse et l'offset pour positionner le nouvel élément sur l'élément de base

L'assemblage peut être plus complexe et continuer de même, si la Payload comprend à sont tour une Payload et ainsi de suite.

ON RETIENDRA QUE:

Un objet est défini par un fichier <Nom de l'objet>.cfg situé dans le dossier CONFIG

Ce fichier peut faire appel pour les cas complexes:

- à un fichier <nom de l'objet>.dll situé dans le dossier module qui est un programme traitant l'ensemble de l'affichage et de l'animation de l'objet

- à un fichier <nom de l'objet>.ini situé dans le dossier SPACECRAFT du dossier CONFIG (ou éventuellement dans un sous dossier de SPACECRAFT)

Les points éventuels d'arrimage sur un porteur (fusée ou navette) et de saisie par bras d'un objet sont définis dans son fichier cfg

Les points de docking possibles sont définis soit dans le fichier cfg, soit dans le fichier ini soit dans le fichier dll pour un objet complexe.

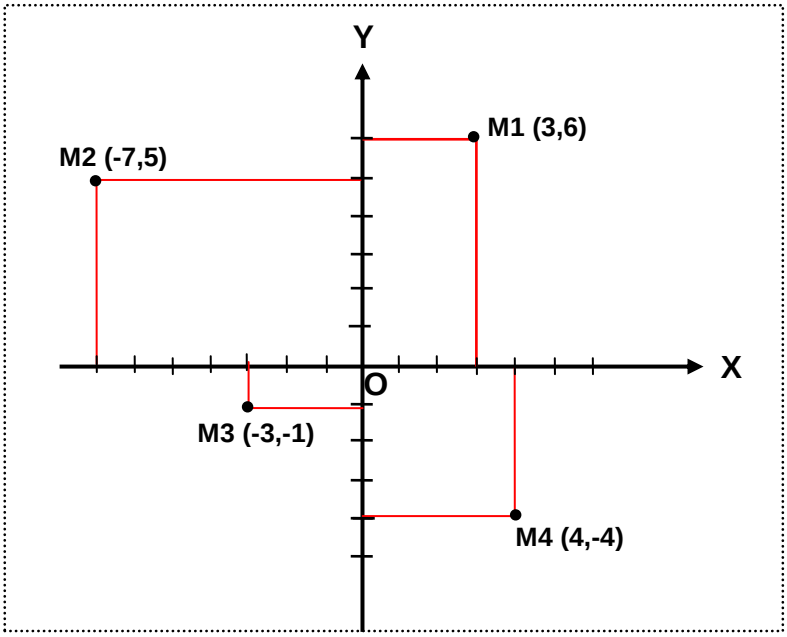
5 - POINTS D'ATTACHE, D'ANCRAGE ET DE DOCKING

Si l'on veut embarquer certaines charges dans la navette, il est possible qu'ils n'aient aucun points d'attache et d'arrimage par le bras et il faut les définir.
Il se peut aussi que l'on veuille modifier ou ajouter des points de docking qui permettent à un module de s'arrimer à un autre module ou permettent l'arrimage d'un vaisseau.
Le but de ce chapitre est de préciser comment faire ?

5.1 – Un peu de géométrie

Pour bien comprendre la façon de définir le positionnement des points nous allons rappeler quelques notions géométriques.

Plaçons nous pour simplifier dans un espace à deux dimensions qui est le plan.
Traçons deux axes se coupant à angle droit en un point 0 et que nous appellerons X et Y.
Graduons ces axes à intervalles réguliers et donnons une orientation en plaçant une flèche en bout d'axe



Nous avons défini un système d'axes de référence. L'axe X est appelé axe des abscisses et l'axe Y est appelé axe des ordonnées.

Si nous voulons définir la position d'un point dans ce système nous donnerons les valeurs mesurées des distances horizontales et verticales du point par rapport aux axes.
Les valeurs seront positives ou négatives suivant que l'on mesure dans le sens de la flèche placée sur l'axe ou en sens inverse.

Sur la figure ci-dessus nous avons écrit entre parenthèse les valeurs qui correspondent aux positions des différents points le premier nombre étant l'abscisse et le second l'ordonnée.

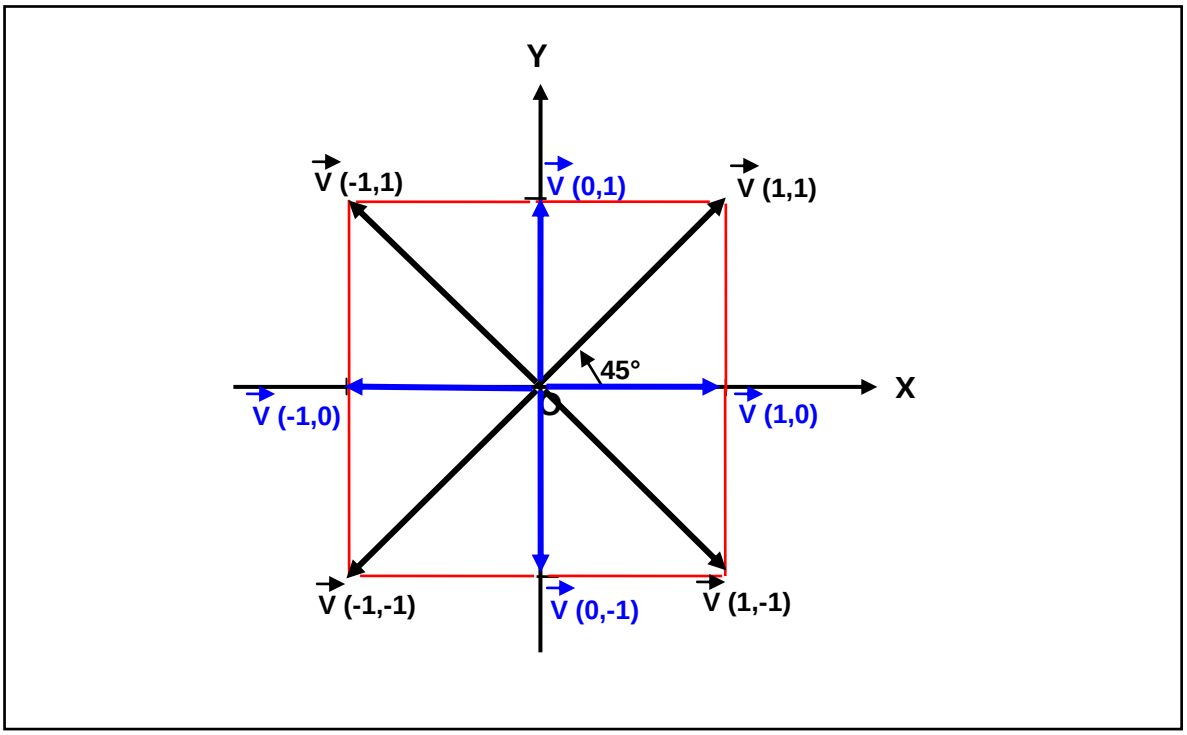
Sur la figure j'ai pris pour simplifier des valeurs entières pour les mesures mais bien entendu on peut prendre des valeurs fractionnaires et définir par exemple un point M (1.3 , -4.8) Je note la virgule de la mesure par un point comme on le trouvera dans Orbiter à la manière anglo-saxonne

OK vous me suivez ? Alors on continue !

Nous savons maintenant repérer un point et nous allons essayer de repérer une direction dans le même système d'axes. Elle servira à préciser le sens d'un déplacement ou la direction à avoir après rotation.

Une direction est représentée par une flèche donnant le sens du déplacement. Cette flèche que nous nommerons ici V est appelée vecteur et on la représente symboliquement par la lettre du nom surmontée d'une flèche.

Comme pour un point on peut mesurer la distance qui sépare le bout de notre flèche de chaque axe. La différence est que l'on mesure en prenant 1 comme unité au maximum.
Le bout du vecteur se déplace donc sur un carré de coté égal à 2 unités comme on le voit sur la figure ci-après.
On exprime la direction du vecteur par les deux nombres compris entre 0 et 1 qui lui sont associés.
Sur la figure ci-après j'ai représenté les vecteurs correspondant à des directions espacées de 45°.
Dans Orbiter seules ces directions sont prises en compte.



Maintenant il faut faire preuve d'imagination pour se représenter ce qui va se passer dans l'espace qui est a trois dimensions
.

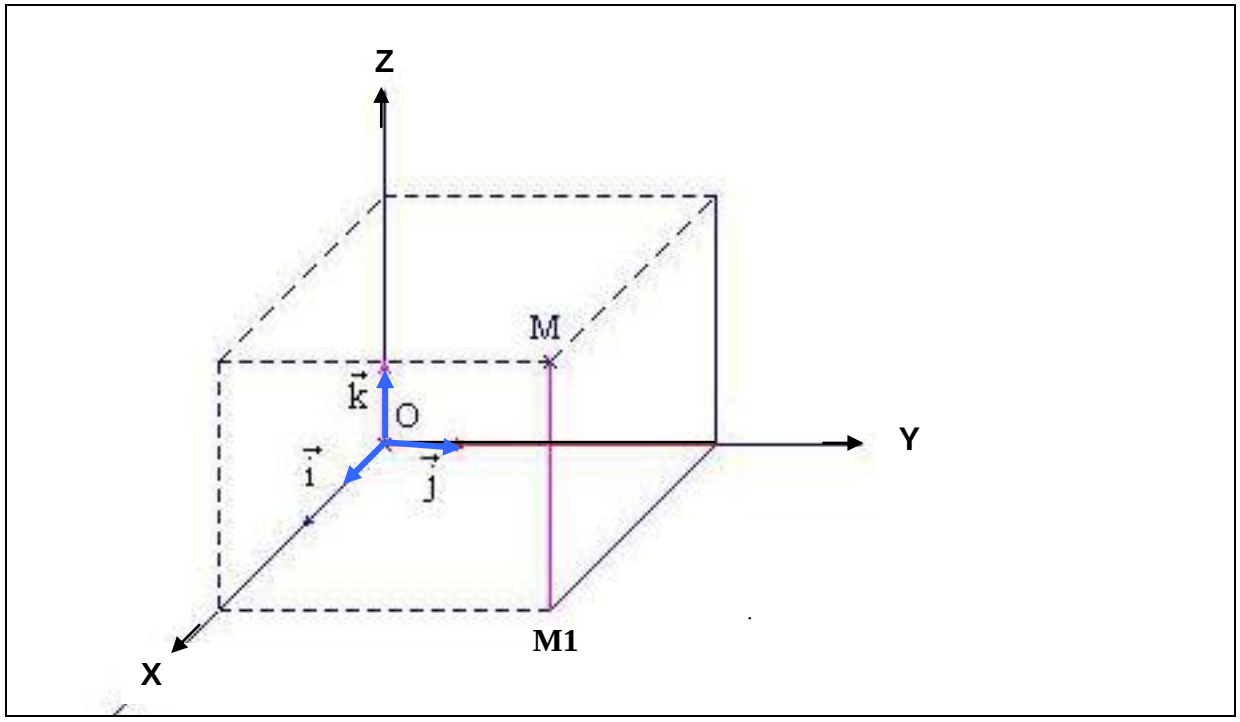
Nous ne sommes plus dans un plan mais dans un volume et il va falloir prendre un système de référence à 3 axes X,Y et Z pour pouvoir représenter une position. Le troisième axe Z est perpendiculaire à l'origine O au plan des axes X et Y

Comme on le voit sur la figure ci-dessous un point M dans l'espace correspond à un point dans le plan (comme M1 dans notre figure dans le plan) mais il est déplacé vers le haut ou le bas sur une perpendiculaire en M1.
La valeur de ce déplacement dans la direction de l'axe Z s'appelle la côte.

La position d'un point dans ce système à trois dimensions se notera **M(3,5,3)** par exemple si il se trouve à 3 unités au dessus du plan XY.

On peut voir également qu'une direction peut être représentée par un vecteur de direction (une flèche donnant le sens) qui peut se décomposer en 3 vecteurs i,j et k projeté sur les axes X,Y,Z..
Par convention, la longueur de ces vecteurs est égale à 1.
On notera un vecteur de déplacement par exemple V(0,0,1)

De façon pratique on retiendra comment est noté un vecteur placé sur un des axes :
V(1,0,0) il est sur l'axe X
V(0,1,0) il est sur l'axe Y
V(0,0,1) il est sur l'axe Z



Pour définir la rotation d'un corps par rapport à un point on peut définir d'abord l'axe de rotation en donnant les valeurs de i,j,k qui le définissent puis donner la direction du vecteur de rotation en définissant la direction de ce vecteur dans le plan perpendiculaire à l'axe de rotation en donnant ses valeurs i,j,k.

Par exemple, si je veux définir le pivotement du cube représenté ci-dessus autour du point O de manière à garder M1M vertical mais à avoir M1 sur l'axe Y je vais définir la rotation par 6 nombres

0 0 1 pour dire que l'axe de rotation est sur l'axe Z (puisque X et Y sont nuls)
0 1 0 pour dire que la rotation amène sur l'axe Y (puisque X et Z sont nuls)

C'est cette notation qui est utilisée dans ce que nous allons examiner par la suite

ON RETIENDRA QUE :

Dans un système d'axes de référence XYZ :

- Un point M est repéré par ses trois coordonnées (abscisse, ordonnée et côte) et s'exprime sous la forme

M(abscisse, ordonnée, côte)

Les valeurs de l'abscisse, l'ordonnée et la côte pouvant être positive ou négatives

- Un déplacement (translation ou rotation) est repéré par un vecteur V et s'exprime sous la forme

V(vecteur i, vecteur j , vecteur k)

Les valeurs de i,j et k étant comprises entre 0 et 1 dans le cas général et égales à 1 pour des déplacement parallèles aux axes ou des rotations à 45°

5.2 – Application aux points d'attache et d'ancrage

Les points d'attache et d'ancrage d'un élément embarqué sont définis dans le fichier de configuration **<Nom>.cfg** de l'élément

- Un point de type XS sert pour attacher le module au lanceur. Il n'y en a qu'un
- Un point de type GS sert pour ancrage éventuel du module par un bras de manipulation. Il peut y en avoir plusieurs

On définit un point dans la liste d'attachement par une notation de type P suivi de 9 nombres et un repère XS ou GS

P < a b c d e f g h i > XS
P < a b c d e f g h i > GS

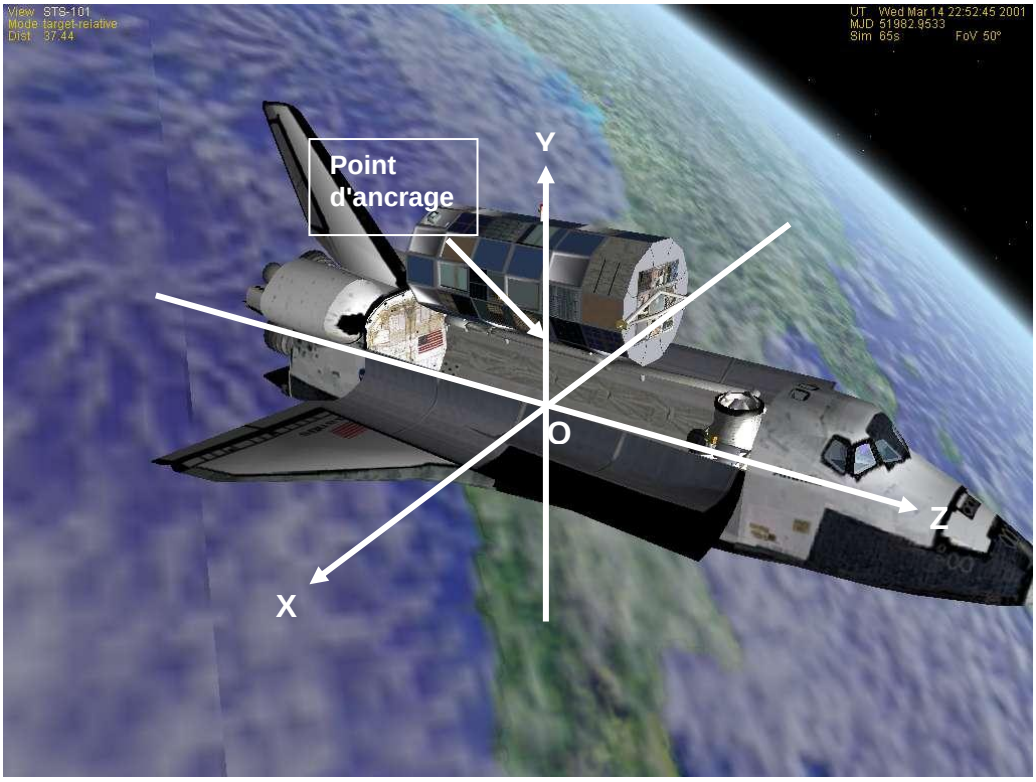
Voilà par exemple ce que l'on trouve dans le fichier LDEF.cfg

```
; === Attachment specs ===
BEGIN_ATTACHMENT
P 0  2.6  0   0  1 0  0 0  1  XS
P 0.05 -2.6  0   0 -1 0  0 0 -1  GS
END_ATTACHMENT
```

Les nombres a, b et c qui suivent le P donnent en valeur positive ou négative les distances en mètres suivant les 3 axes du point XS ou GS mesurées dans le système de référence de l'objet auquel il se rattache et dont l'origine O est :

- Le point d'attache en soute de la navette pour XS
- Le centre de gravité de l'élément embarqué pour GS

Les valeurs du déplacement suivant les axes sont données dans l'ordre des axes X,Y et Z



Sur cette photo avec le satellite LDEF largué sont figurés les 3 axes de référence passant par le point d'attache de la navette.

La navette qui porte le satellite a un point d'attache théorique situé au milieu de la soute et il n'est pas modifiable.
Il faut donc positionner le point d'attache du satellite pour qu'il se trouve au bon endroit par rapport au point de soute

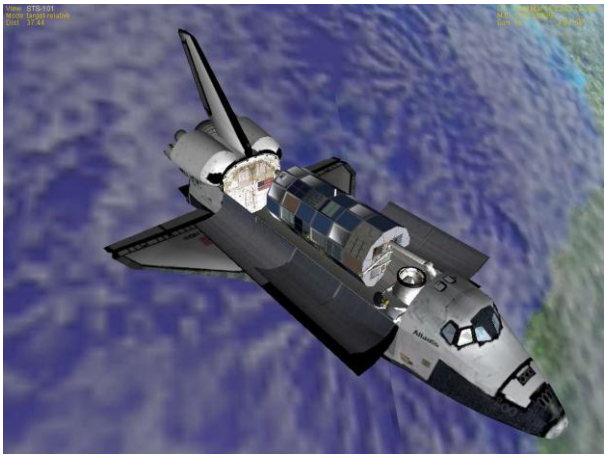
Les trois premiers nombres du point d'attache de LDEF sont 0, 2.6 et 0 ce qui veut dire que par rapport au point d'attache de la navette le point G est remonté de 2;6 m au dessus du point d'attache défini pour la navette dans le sens de l'axe de lacet et que le satellite est centré sur les deux autres axes (2,6 m est à peu près le rayon du satellite)

Voilà ce que l'on obtient en faisant varier des valeurs par rapport aux valeurs d'origine 0 2.6 et 0 pour changer le positionnement du point d'attache du satellite.

P 0 0 0 0 1 0 0 0 1 XS
on descend de 2,6 m

P 0 4.6 0 0 1 0 0 0 1 XS
on monte de 2 m

P 2 2.6 0 0 1 0 0 0 1 XS
On décale de 2 m à droite vu de derrière



P -2 2.6 0 0 1 0 0 0 1 XS
On décale de 2 m à gauche vu de derrière

P 0 2.6 2 0 1 0 0 0 1 XS
On décale de 2 m vers l'arrière

P 0 2.6 -2 0 1 0 0 0 1 XS
On décale de 2 m vers l'avant

On peut donc positionner un satellite dans la soute en faisant varier les trois valeurs de XS

NOTA :
Nous verrons plus loin que l'on peut réaliser le positionnement d'une autre façon en agissant directement sur le scénario pour fixer un décalage (offset du point théorique de la navette)

Les 6 derniers nombres d,e,f,g,h,et i qui suivent le P servent à définir :

- pour le point XS l'orientation du module dans la soute par rapport aux axes de références de la navette
- pour le point GS la rotation du point d'ancrage du satellite par rapport à ses axes de référence

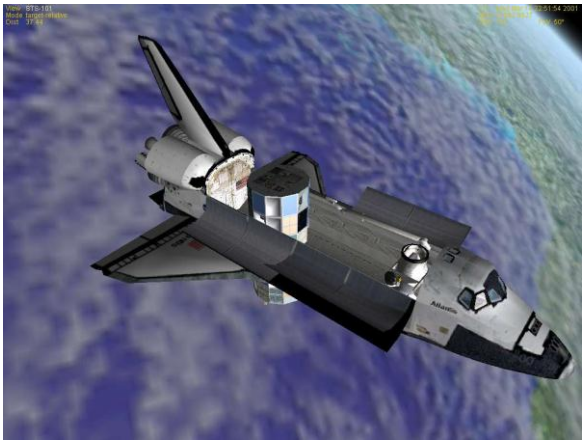
Puisqu'il défini un axe de rotation X,Y ou Z, le premier groupe d e f ne peut avoir comme valeur qu'une des 3 combinaisons 0 0 1 ou 0 1 0 ou 1 0 0

Puisqu'il définit un vecteur de rotation par rapport à l'axe précédent, un nombre du groupe g h i ne peut avoir la valeur 1 si le nombre de rang identique du groupe d e f est égal à 1

Il est difficile de manipuler ces données qui sont les valeurs unitaires d,e et f du vecteur donnant l'axe de rotation et les valeurs unitaires g,h et i du vecteur donnant l'angle de rotation dans le système d'axe de référence auquel il se rapporte.
En principe il n'est pas nécessaire de les modifier mais si le besoin s'en fait sentir il faut procéder à des expériences en modifiant les valeurs par groupe de 3 et en voyant le résultat graphique.

Exemples de rotation (c'est un peu plus difficile !)

P 0 2.6 0 **0 1 0 1 0 0** XS



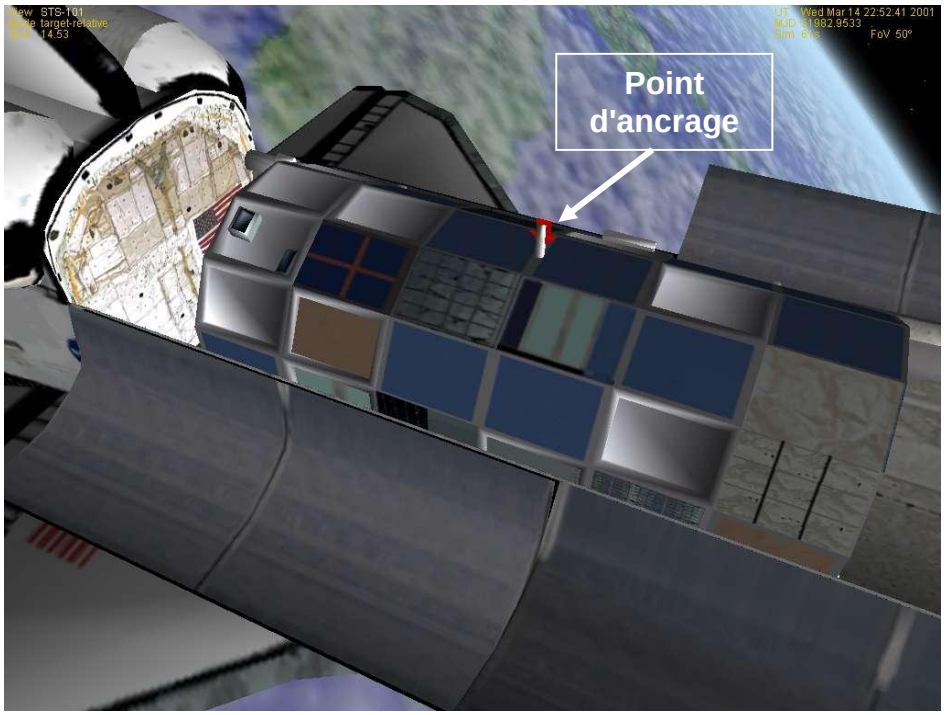
P 0 2.6 0 **0 0 1 0 1 0** XS

Le raisonnement est le même pour le ou les point d'ancrage GS que l'on positionne sur le satellite par rapport au centre de gravité

Sur la figure ci-dessous, on voit un point d'ancrage situé sur le dessus de LDEF.
Les valeurs sont: P 0.05 -2.6 0 0 -1 0 0 0 -1 GS

Le point d'ancrage est donc situé en léger décalage à droite de 0,05 m et remonté à 2;60 m (G est 2,6 m en dessous du point)

Les 6 derniers nombres 1 ou 0 permettent de définir la position de la flèche rouge de repérage. Dans cet exemple 0 -1 0 montre qu'elle est dirigée vers le bas suivant l'axe vertical Y et 0 0 -1 la place à plat dans le sens longitudinal. (on pourrait écrire 0 0 1 et on aurait la même visualisation en symétrie)



La soute d'une navette peut recevoir au maximum des charges de 5,5 m de diamètre et de longueur 16 m environ d'où des déplacements de points qui ne doivent pas dépasser en gros 2,7 m.en largeur et 8m en longueur par rapport au centre de gravité.
Ca donne une enveloppe pour le réglage si l'on veut ajouter des points sur un satellite n'en ayant pas et que l'on destine à la navette.

5.3 – Points de Docking

Ces points peuvent être définis dans le fichier cfg de configuration de l'objet ou dans le fichier ini du dossier Spacecraft. Attention ! leur repère dans le scénario va de 0 à N-1 si il y a N points

Dans le fichier de configuration ils sont définis comme ci après :

```
; === Docking ports ===    points d'arrimage inter modules
BEGIN_DOCKLIST
0 0  5   0 0  1    0 1 0   ;== attention c'est le point 0 dans le scenario ne le prenez pas pour le point 1
0 0 -5   0 0 -1    0 1 0   ;== attention c'est le point1 dans le scenario ne le prenez pas pour le point 2
END_DOCKLIST
```

Comme pour les points XS et GS, les trois premiers nombres donnent la position du port de docking par rapport au centre de gravité et les 6 nombres suivants donnent la direction du vecteur d'approche et la position en rotation par rapport à ce vecteur pour que le docking soit correct (c'est la position donnée par le triangle rouge MFD de docking lorsqu'il devient blanc)

Dans un fichier ini chaque point de docking est défini comme suit

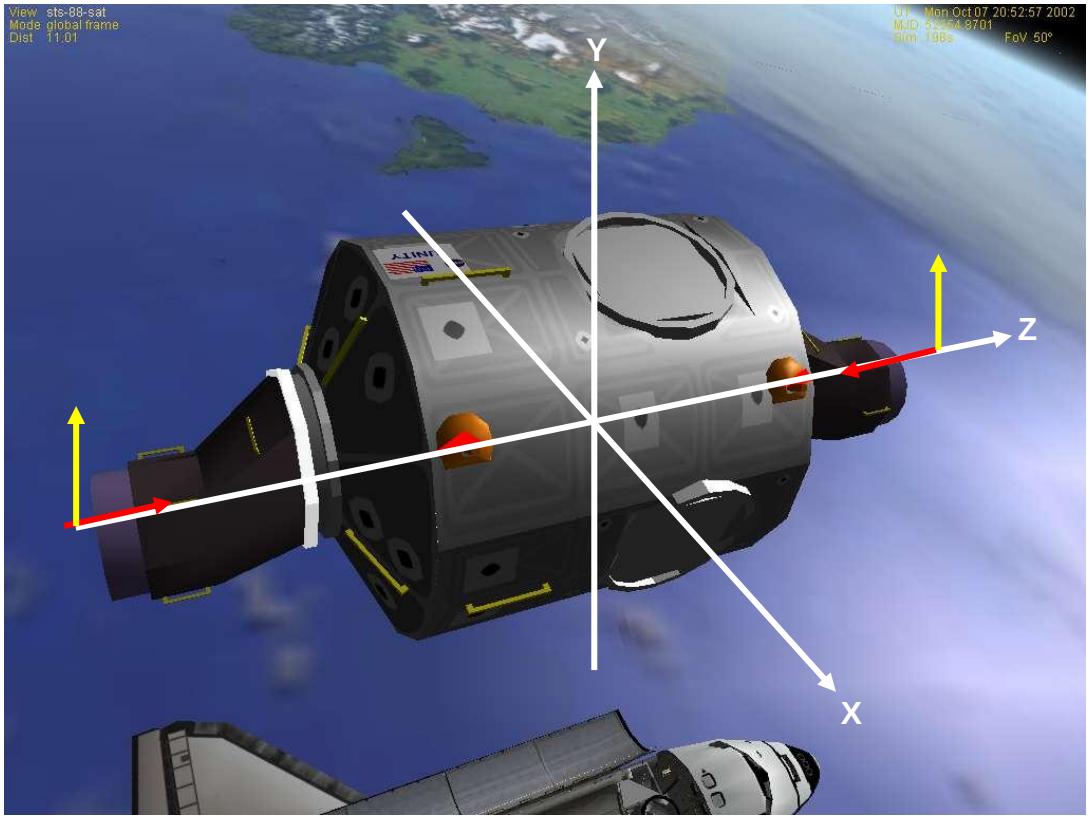
```
[DOCK_0]
POS=(0,0.07,5.64)
DIR=(0,0,1)
ROT=(0,1,0)
```

POS contient les coordonnées du point par rapport au point de référence de l'objet (c'est l'équivalent des 3 premiers nombres pour un point GS)

DIR contient les cordonnées du vecteur unitaire donnant la direction pour le docking (c'est l'équivalent des nombres n° 4,5 et 6 pour un point GS

ROT contient les coordonnées du vecteur unitaire définissant la position en rotation par rapport au vecteur défini par DIR pour que le docking soit correct (c'est l'équivalent des nombres n° 7,8 et 9 pour un point GS)

Prenons par exemple ce module Unity base de construction pour ISS.



Et la liste de ses points de docking

```
; === Docking ports ===
BEGIN_DOCKLIST
0 0 5   0 0 1  0 1 0
0 0 -5  0 0 -1  0 1 0
END_DOCKLIST
```

Dans la docklist on trouve deux points de coordonnées 0 0 5 et 0 0 -5. C'est logique puisque les deux sas qui se trouvent en bout sont décalés de 5 m vers l'avant et 5 m vers l'arrière.
On trouve ensuite 0 0 1 pour un point et 0 0 -1 pour l'autre. C'est toujours logique puisque pour les deux points le vecteur d'approche est dans la direction de l'axe longitudinal mais en sens opposé pour un point par rapport à l'autre

Enfin on trouve 0 1 0 comme 3 derniers nombres qui donne la même position en rotation par rapport à l'axe d'approche

ON RETIENDRA QUE :

Un objet embarqué a trois types de points particuliers qui se définissent par rapport à 3 axes de référence

- Un point d'attachement obligatoire permettant la fixation dans le véhicule de transport

- Un ou plusieurs points d'ancrage optionnels

- Un ou plusieurs points de docking (si le module doit être assemblé)

La position de ces points est définie dans le fichier de configuration cfg pour le point d'attachement et les points d'ancrage et dans le fichier de configuration cfg ou le fichier ini du dossier Spacecraft, ou le fichier dll pour les points de docking

A titre d'exemple de modification, nous allons traiter le cas d'un module appelé ttmod1 que l'on trouve dans le pack Stationpart1.zip téléchargeable sur Orbithangar.
Son fichier de configuration est le suivant :

```
; === Configuration file for the International Space Station ===

Meshname = ttmod1
Mass = 8e3
Size = 6.0
Inertia = 7.7 7.7 2.3

; === Docking ports ===
BEGIN_DOCKLIST
0 0  5   0 0  1    0 1 0
0 0 -5   0 0 -1    0 1 0
END_DOCKLIST

; === Attachment specs ===
BEGIN_ATTACHMENT
P 0 0 0 0 0 1 0 1 0 parent
C 0 2 0 1 0 0 0 1 0 child
END_ATTACHMENT
```

Les notations pour les points d'attachement ne conviennent pas dans la version Orbiter actuelle.
On repérait avant dans Orbiter le point pour le "parent" ici la navette et le point pour l'enfant (child) qui est le module.
Nous supprimons la ligne parent et remplaçons C <9 nombres> child par P <9 nombres>XS

```
; === Attachment specs ===
BEGIN_ATTACHMENT
P 0 2 0 1 0 0 0 1 0 XS
END_ATTACHMENT
```

Si nous embarquons le module dans la navette comme nous allons le voir plus loin voilà le résultat !

Il est placé à l'horizontale et déborde largement de la soute

Si vous avez suivi mes explications préalables il faut modifier les valeurs des 3 derniers nombres de XS pour le faire pivoter dans une bonne position en écrivant 001 au lieu de 010 pour s'aligner sur l'axe longitudinal.

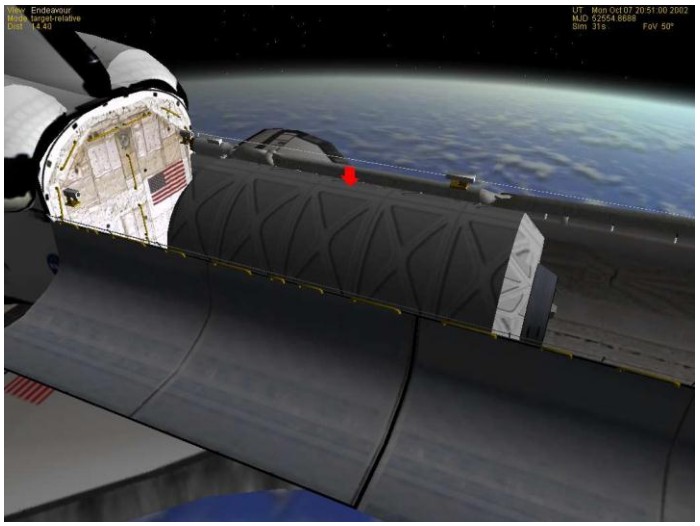
```
; === Attachment specs ===
BEGIN_ATTACHMENT
P 0 2 0 1 0 0 0 0 1 XS
END_ATTACHMENT
```

Cette fois ci il est aligné mais il est décalé par rapport à l'axe.
Il faut modifier les 3 premiers nombres 0 2 0 pour le recentrer.
La valeur 2 est mauvaise maintenant du fait du pivotement car elle nous décale latéralement aussi va-t-on la supprimer pour avoir:

```
; === Attachment specs ===
BEGIN_ATTACHMENT
P 0 0 0 1 0 0 0 0 1 XS
END_ATTACHMENT
```

et l'on obtient le résultat de la photo de droite en bonne position

Il ne nous restera plus qu'à définir un point d'ancrage pour pouvoir utiliser le module pour construire notre station spatiale.



```
; === Attachment specs ===
BEGIN_ATTACHMENT
P 0 0 0 1 0 0 0 0 1 XS
P -2.2 0 0 -1 0 0 0 0 1 GS
END_ATTACHMENT
```

Et voilà de quoi saisir et manipuler au bras ! (flèche rouge)

NOTA –
Pour modifier ou améliorer un élément embarqué il faut faire attention aux changements éventuels d'axes de référence entraînés soit par la rotation soit du fait de la conception.
Pour le point d'ancrage on pouvait s'attendre à écrire 0 -2.2 0 pour les trois premiers et en fait il faut écrire -2.2 0 0 ce qui montre que les axes X et Y sont inversés.
Il sera bon avant toute modification d'embarquer le module et de vérifier sur l'affichage comment varie sa position en faisant varier successivement les trois premiers nombres après P (une valeur de +2 par exemple par rapport à la valeur d'origine) On sera ainsi sûr d'avoir correctement identifié les axes X,Y,Z

6 – EMBARQUEMENT DANS LA NAVETTE

Nous avons un élément que nous voulons lancer et nous avons éventuellement ajouté ou corrigé les points d'attache, d'ancrage et de docking nécessaires dans le fichier cfg de l'élément et/ou le fichier ini

6.1 – Base du scénario

Il faut prévoir deux vaisseaux au minimum dans le scénario qui sont la navette choisie et l'élément embarqué.

Supposons que l'on lance le 14 juillet 2005 le module **CARINA** que nous appelons symboliquement Papy avec la navette Endeavour appelée symboliquement ST104
Il faut déclarer ce module comme attaché à la navette en **Status Orbiting Earth que la navette soit au sol ou pas** en spécifiant que son point 0 est attaché au point 0 de la navette désignée par son appellation en utilisant la terminologie **ATTACHED 0:0,ST104**

```
BEGIN_DESC
ST104 Endeavour avec Carina
END_DESC

BEGIN_ENVIRONMENT
  System Sol
  Date MJD 53565.5        ;=== le 14 juillet à midi
END_ENVIRONMENT

BEGIN_FOCUS
  Ship ST104
END_FOCUS

BEGIN_CAMERA
  TARGET ST104
  MODE Extern
  POS 3.25 -146.59 -25.21
```

```
  TRACKMODE TargetRelative
  FOV 50.00
END_CAMERA

BEGIN_HUD
END_HUD

BEGIN_MFD Left
END_MFD

BEGIN_MFD Right
END_MFD

BEGIN_SHIPS
ST104:Endeavour          ;== navette appelée ST-104 : nom de son fichier cfg
  STATUS Landed Earth
  BASE Cape Canaveral:11
  HEADING 0.00
  PRPLEVEL 0:1.000 1:1.000 2:1.000
  NAVFREQ 0 0
  CONFIGURATION 0
  CARGODOOR 0 0.0000
  GEAR 0 0.0000
  SAT_OFS_X 0.000
  SAT_OFS_Y 0.000
  SAT_OFS_Z 0.000
  ARM_SH_P 0.000
  ARM_SH_Y 0.000
  ARM_EL_P 0.000
  ARM_WR_P 0.000
  ARM_WR_Y 0.000
  ARM_WR_R 0.000
  TGT_HEADING  42.000
END
Papy:Carina                ;== Sattelite appelé Papy : nom de son fichier cfg
  STATUS Orbiting Earth
  ATTACHED 0:0,ST104       ;== attaché par les ports 0 à la navette appelée ST-104
END
END_SHIPS
```

Si le point d'attache du module est correctement défini, il se place bien dans la soute et il n'y a rien d'autre à faire.
Si le module est décalé, on peut modifier les valeurs de SAT_OFS_X, SAT_OFS_Y et SAT_OFS_Z pour rectifier la position sur un ou plusieurs axes (ce sont les valeurs d'offset (décalage) du centre de gravité du module par rapport au point d'attache de la navette)

6.2 - Cas du module avec un berceau

Le berceau support est une partie fixe qui reste dans la navette après lancement du module.
Ce berceau appelé CRADLE en anglais est défini en ajoutant les lignes suivantes dans la définition de la navette

```
CARGO_STATIC_MESH <nom du mesh du berceau>
CARGO_STATIC_OFS 0.000 0.000 0.000
```

Les trois nombres qui suivent CARGO_STATIC_OFS sont les valeurs de déplacement du berceau sur les 3 axes et peuvent être modifiés pour ajuster sa position.
Par exemple pour le satellite Carina on a:

```
CARGO_STATIC_MESH Carina_cradle
CARGO_STATIC_OFS 0.000 -1.650 0.050
```

La représentation du berceau peut être utilisée avec un autre satellite si elle convient

A titre d'exemple, nous allons créer un scénario idiot ou l'on embarquerai la sonde Cassini qu'on peut trouver dans l'add-on Cassini.zip en la plaçant sur le berceau de Carina (ça n'est pas terrible mais c'est pour l'exemple !)

Le fichier Cassini.cfg d'origine est le suivant :

classname = cassini
Module = spacecraft/Cassini

C'est un module composé qui possède un fichier Cassini.ini dans le dossier Spacecraft

Pour pouvoir lier Cassini à la navette il faut rajouter un point d'attachement (par exemple celui de Carina pris dans le dossier Carina.cfg)
On peut aussi rajouter un point d'arrimage si on le désire

Pour pouvoir voir le satellite attaché à la navette, il faut rajouter le nom du Mesh lu dans Cassini.ini dans le fichier cfg

Enfin on peut se contenter pour simplifier d'écrire dans la ligne module spacecraft seul si le classname est rigoureusement celui du fichier ini

Et voilà le fichier Cassini.cfg modifié ou j'ai ajouté un joli titre (après un point virgule la ligne n'est pas prise en compte par le logiciel)

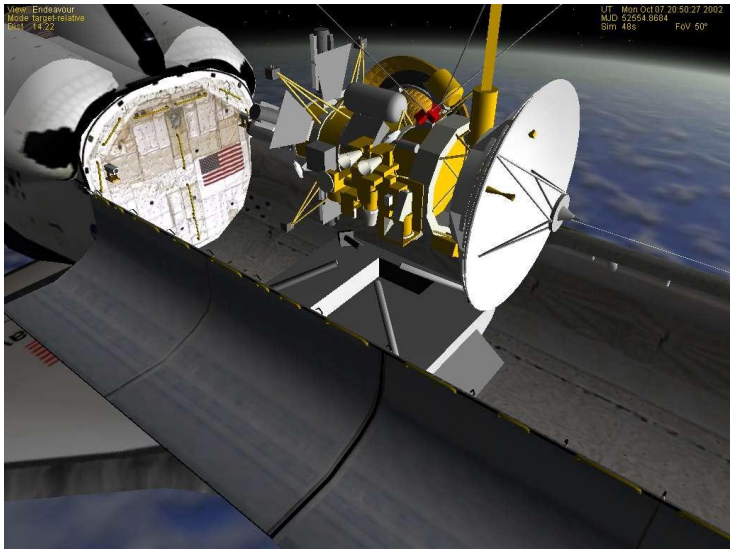
; === Configuration file for vessel class Cassini ===
classname = cassini
Module = spacecraft
Meshname = Cassini

; === Attachment specs ===
BEGIN_ATTACHMENT
P 0 0 0 0 -1 0 0 0 1 XS
END_ATTACHMENT

Il n'y a plus qu'à embarquer dans la navette en écrivant le scénario ci-dessous:

```
BEGIN_DESC
ST104 en orbite avec Cassini embarqué
END_DESC

BEGIN_ENVIRONMENT
  System Sol
  Date MJD 52554.8678210764
END_ENVIRONMENT

BEGIN_FOCUS
  Ship ST104r
END_FOCUS

BEGIN_CAMERA
  TARGET ST104
  MODE Extern
  POS 3.25 -146.59 -25.21
  TRACKMODE TargetRelative
  FOV 50.00
END_CAMERA

BEGIN_HUD
  TYPE Surface
END_HUD

BEGIN_MFD Left
END_MFD

BEGIN_MFD Right
END_MFD

BEGIN_SHIPS
ST104:Endeavour
  STATUS Orbiting Earth
  RPOS -1733857.76 -6231476.36 1319382.70
  RVEL -6894.555 1199.604 -3395.129
  AROT 160.82 62.33 -34.32
  VROT -0.06 -0.00 -0.00
  PRPLEVEL 0:0.878
  NAVFREQ 0 0
  CONFIGURATION 3
  CARGODOOR 1 1.0000
```

```
  GEAR 0 0.0000
  SAT_OFS_X 0.000
  SAT_OFS_Y 0.000
  SAT_OFS_Z 0.000
  ARM_SH_P 0.000
  ARM_SH_Y 0.000
  ARM_EL_P 0.000
  ARM_WR_P 0.000
  ARM_WR_Y 0.000
  ARM_WR_R 0.000
  CARGO_STATIC_MESH Carina_cradle
  CARGO_STATIC_OFS 0.000 -1.650 0.050
END
Sat_Cassini:Cassini
STATUS Orbiting Earth
ATTACHED 0:0,ST104
END
END_SHIPS
```



Je viens de le lâcher et on peut voir le berceau Carina en dessous

On voit avec cet exemple qu'il est possible d'embarquer avec des supports beaucoup d'éléments personnalisés ou non dans la navette en suivant la technique exposée

ON RETIENDRA :

Pour embarquer un module dans la navette il faut :

- Qu'il possède un fichier <NOM>.cfg contenant au moins un point d'attachement XS
- Qu'il soit défini dans la liste des vaisseaux BEGIN_SHIP……END_SHIP en tant que STATUS ORBITING EARTH
- Qu'il soit défini comme ayant son port 0 attaché au port 0 de la navette

Si le module est composé de plusieurs modules, le fichier de configuration doit faire appel au fichier ini maître qui se trouve dans le dossier Spacecraft et il doit contenir les lignes suivantes

Classname = <le nom du fichier ini maître>
Module = Spacecraft
Mesh =<le nom de la mesh du fichier ini maître>

BEGIN_ATTACHMENT
P < 9 nombres > XS ;=== pour un point d'attache obligatoire
P < 9 nombres > GS ,=== pour un point d'arrimage optionnel (plusieurs possibles)
END_ATTACHMENT

6.3 - Comment embarquer plus d'un module ?

La navette a en fait deux points d'attache : un point de soute (point 0) et un point au bout du bras (point 1) plus un point de docking qui est les sas à l'avant.
On ne peut pas rajouter de point d'attache supplémentaire et il faut se débrouiller avec le bras si on veut fixer un deuxième élément dans la soute.
Bien entendu il faut que la taille des éléments permette d'en mettre deux (soute de 5,5 x 16 m)

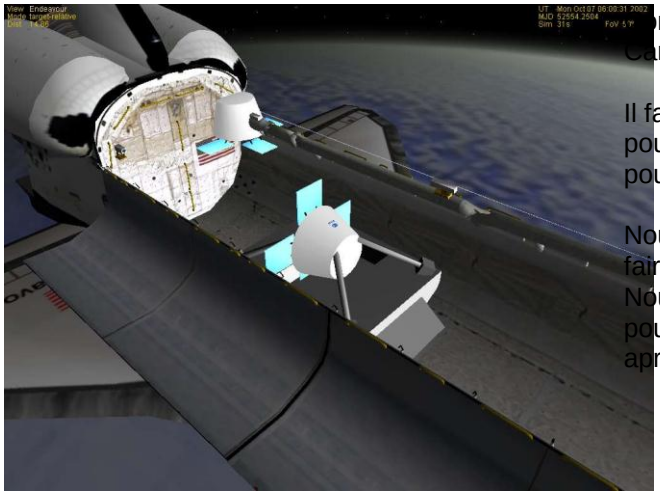
Le premier élément est placé normalement et positionné plus vers l'avant si il le faut en jouant sur l'offset dans la spécification de la navette et le deuxième élément sera déclaré comme attaché au bras et il faudra modifier la spécification de son point d'attachement pour qu'il soit bien positionné.

Supposons que nous voulions embarquer deux satellites Carina dans Endeavour.
Nous écrirons le scénario suivant (sans modification particulière) en ajoutant un deuxième Carina1 attaché au point 1.
Créons un fichier Carina1.cfg en dupliquant le fichier Carina.cfg
Pour donner de la place à Carina 1 nous prenons un offset de 4m pour le berceau et le satellite Carina
Nous présentons Endeavour en orbite soute ouverte (CARGODOOR1 1.0000) pour faciliter le positionnement mais bien entendu on pourrait prendre la navette au départ et faire les essais portes de soute ouverte.

```
BEGIN_DESC
Endeavour avec deux modules Carina
END_DESC

BEGIN_ENVIRONMENT
  System Sol
  Date MJD 52554.25
END_ENVIRONMENT

BEGIN_FOCUS
  Ship Endeavour
END_FOCUS

BEGIN_CAMERA
END_CAMERA

BEGIN_HUD
END_HUD

BEGIN_MFD Left
END_MFD

BEGIN_MFD Right
END_MFD


BEGIN_SHIPS
Endeavour:Endeavour
  STATUS Orbiting Earth
  RPOS -1733857.76 -6231476.36 1319382.70
  RVEL -6894.555 1199.604 -3395.129
  AROT 160.82 62.33 -34.32
  VROT -0.06 -0.00 -0.00
  PRPLEVEL 0:0.878
  NAVFREQ 0 0
  CONFIGURATION 3
  CARGODOOR 1 1.0000
  GEAR 0 0.0000
  SAT_OFS_X 0.000
  SAT_OFS_Y 0.000
  SAT_OFS_Z 4.000     ;== offset pour avancer Carina de 4 m
  ARM_SH_P 0.000
  ARM_SH_Y 0.000
  ARM_EL_P 0.000
  ARM_WR_P 0.000
  ARM_WR_Y 0.000
  ARM_WR_R 0.000
  CARGO_STATIC_MESH Carina_cradle
  CARGO_STATIC_OFS 0.000 -1.650 4.0    ;== offset pour advancer le berceau de 4 m
END
```

```
Carina:Carina
  STATUS Orbiting Earth
  ATTACHED 0:0,Endeavour  ;== Carina attaché normalement
END
Carina1:Carina1
  STATUS Orbiting Earth
  ATTACHED 0:1,Endeavour   ;== Carina1 attaché au bras
END
END_SHIPS
```

Nota :
Il n'est pas possible de figurer un deuxième berceau. La seule solution consiste à modifier le Mesh si on sait le faire pour étendre sa capacité.
Nous devrons nous contenter de cette approche.



Comme on pouvait s'y attendre, le deuxième Carina est au bout du bras mais mal disposé.

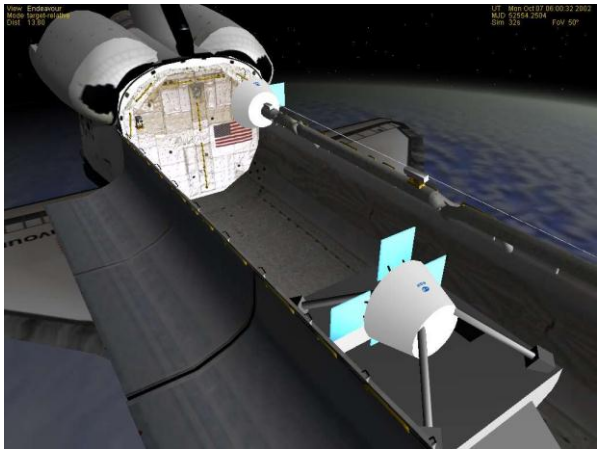
Il faut le le faire pivoter suivant l'axe transversal pour le présenter correctement, puis le décaler pour le centrer sur l'axe longitudinal.

Nous allons modifier le fichier Carina1.cfg pour faire cela en modifiant les valeurs du point XS. Nous garderons les mêmes valeurs pour GS pour pouvoir ensuite le manipuler normalement après largage

La formule du point d'attachement d'origine est **P 0 0 0 0 -1 0 0 0 1 XS**
Il faut d'abord faire pivoter en jouant sur les 6 derniers nombres,.puis centrer ou l'on veut en jouant sur les trois premiers nombres

Il faut faire quelques essais avant d'y arriver car les axes x,y,z par rapport au bras ne sont pas forcément placé dans l'ordre x,y, z de ceux de la navette

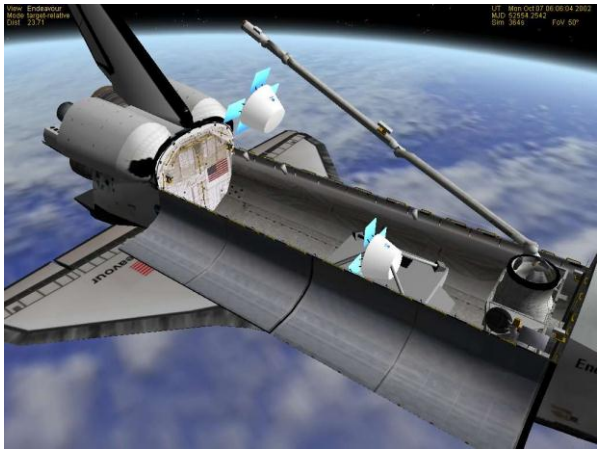
a) on effectue une rotation autour de l'axe transversal
b) on décale de 2.2 en transversal (il y a à peu près 2,20 m entre le bout du bras rangé et l'axe de la navette) on abaisse de 1.5 m et on avance de 4 m



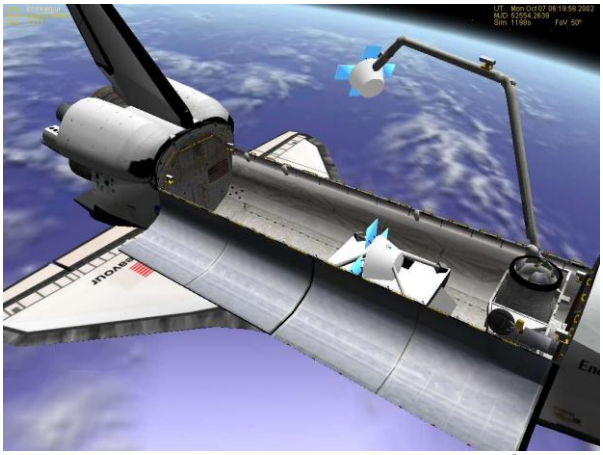
P 0 0 0 0 0 1 0 1 0 XS
Rotation autour de l'axe transversal

P -2.2 1.5 -4 0 0 1 0 1 0 XS
Positionnement au centre

On pourra larguer Carina1 en faisant une release du bras puis le reprendre avec le bras par son point d'ancrage si on veut le manipuler de façon réaliste.
Carina se largue et se manipule normalement



Si on ne fait pas release pour le bras au préalable Carina1 se déplace sans être au bout du bras



Si on fait d'abord release, Carina1 peut être saisi ensuite normalement par le bras

7 – EMBARQUEMENT SUR UNE FUSEE

Il existe de nombreux types de lanceurs, les plus courants étant Ariane, Delta, Proton et Soyouz. Ces lanceurs sont multi-étages et reçoivent une ou plusieurs charges (PAYLOAD) à lancer qui peuvent être également multi étage (Cassini par exemple qui porte Huygens ou Spirit avec son parachute et ses airbags)
Je me contente de vous donner les grandes lignes de conception.

Comme pour les satellites, le lanceur possède un fichier de configuration **<Nom du lanceur>.cfg** qui fait appel pour sa gestion à:
:

- Soit à un module dll qui lui est propre(Apollo NASSP par exemple)
- Soit à des bibliothèques standards de gestion comme M**ultistage.dll** et **Spacecraft.dll (** développés par VINKA que je remercie) ou plus récemment CVE et CVEL

Dans le dossier de configuration, la ligne **Module = <Nom>** permet d'appeler le module adéquat.

Par exemple voilà le fichier **Titan4BCassini.cfg** du lanceur Titan de cette sonde

```
; === Configuration file for vessel class Titan4B ===
classname = titan4Bcassini
Module = multistage      ;== on appelle Multistage.dll
```

Si la fusée et ses charges ne fait pas l'objet d'un module propre dll, on fait appel à un fichier **<Nom>.ini** situé dans le dossier SPACECRAFT ou un de ses sous-dossiers qui définit l'ensemble fusée plus "payloads" ou à des fichiers définissant les différentes charges.

Par exemple l'ensemble fusée Titan avec Cassini au lancement fait l'objet d'un seul fichier ini. Alors que Ariane 5 fait appel à des fichiers particuliers pour les charges.

La rubrique de description du vaisseau dans le scénario peut donc avoir deux types de forme:

- la forme ci-dessous ou l'on nomme la configuration complète dans une ligne CONFIG_FILE

```
BEGIN_SHIPS
titan4Bcassini:titan4Bcassini  ; fichier cfg
  STATUS Landed Earth
  BASE Cape Canaveral:3
  HEADING 90.00
  FUEL 1.000
  CONFIG_FILE Config\boosters\titan4Bcassini.ini  ;fichier ini dans le sous dossier Boosters
  CONFIGURATION 0
  CURRENT_BOOSTER 0
  CURRENT_STAGE 1
```

```
  CURRENT_PAYLOAD 0
  FAIRING 1
END
END_SHIPS
```

- la forme ci-dessous ou l'on nomme les Payloads séparément

```
BEGIN_SHIPS
A5-04:ariane5\A5V       ; ==fichier A5V.cfg dans sous dossier Ariane5
  STATUS Landed Earth
  BASE Kourou:1
  HEADING 135.00
  PRPLEVEL 0:1.000 1:1. 2:1.
  CONFIGURATION 0
  FAIRING SMALL 5.9
  PAYLOAD Carina1 carina carina 0.0 0.0 2.5 5000. 0.
  PAYLOAD _speltra ariane5\speltra5660 ariane5\speltra5660 0.0 0.0 4. 800. 0.
  PAYLOAD PB-01 ShuttlePB ShuttlePB 0. 0. 11. 800. 0.0 0.5
END
END_SHIPS
```

Dans les cas ou le fichier ini de la (ou des) charges est (sont) séparé (s) on pourra substituer une charge à une autre sous réserve de compatibilité de taille et de poids.

8 – AJOUTER UN VAISSEAU PRES D'UNE STATION

Si l'on ouvre le scénario Mir.scn contenu dans le dossier Space Station on est en présence de plusieurs stations et plusieurs vaisseaux.

Supposons que nous souhaitions ajouter une navette Endeavour baptisée ST-101 à proximité de ISS
Une première possibilité est de définir la navette comme dockée puis on la largue, on la positionne ou l'on veut avec le RCS en translation et enfin on sauvegarde le scénario
On modifie le scénario d'origine pour y écrire :

```
BEGIN_SHIPS
ISS:ProjectAlpha_ISS
  STATUS Orbiting Earth
  RPOS -2782560.18 -5745474.73 2160729.54
  RVEL -6860.604 3450.801 355.409
  AROT 110.00 -10.00 80.00
  PRPLEVEL 0:1.000
  NAVFREQ 0 0
  XPDR 466
  DOCKINFO 1:0,ST-101   ; on arrime ST101
END
ST-101:Endeavour        ; on ajoute ST101
  STATUS Orbiting Earth
  PRPLEVEL 0:1.000 1:0.993
  CONFIGURATION 3
  CARGODOOR 1 1.0000
  GEAR 0 0.0000
  DOCKINFO 0:1,ISS
END
```

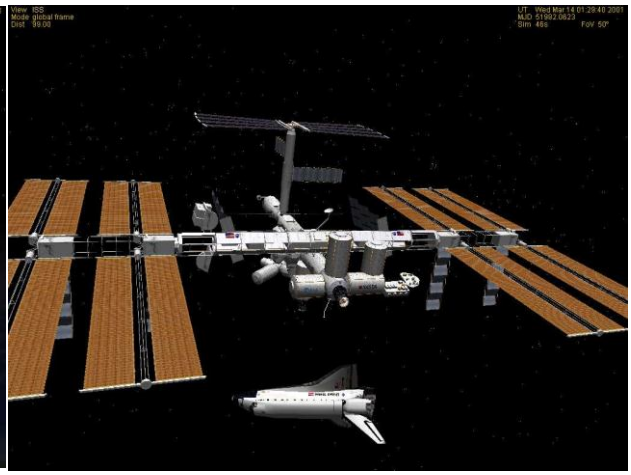
Et il ne reste plus qu'à larguer ST-101 et la positionner en translation

Il y a une deuxième méthode qui consiste donner à la navette le Status identique à celui de la station en copiant RPOS et RVEL puis à modifier légèrement sa position

```
BEGIN_SHIPS
ISS:ProjectAlpha_ISS      ;=== rien à modifier sur ISS
  STATUS Orbiting Earth
  RPOS -2782560.18 -5745474.73 2160729.54
  RVEL -6860.604 3450.801 355.409
  AROT 110.00 -10.00 80.00
  PRPLEVEL 0:1.000
  NAVFREQ 0 0
  XPDR 466
END
ST-101:Endeavour     ;=== on ajoute Endeavour
```

```
  STATUS Orbiting Earth
  RPOS -2782560.18 -5745474.73 2160729.54  ; === même RPOS que ISS
  RVEL -6860.604 3450.801 355.409          ; === même RVEL que ISS
  PRPLEVEL 0:1.000 1:0.993
  CONFIGURATION 3
  CARGODOOR 1 0.0000  ; soute fermée
  GEAR 0 0.0000
END
```

Comme on pouvait s'y attendre (figure 1) la navette est en plein milieu de la station.
En ajoutant ou en retranchant quelques dizaines à la première valeur de RPOS de la navette, on va l'éloigner de la station
Sur la figure 2 vous voyez le résultat en augmentant de 50 la première valeur de RPOS pour avoir
RPOS -2782610.18 -5745474.73 2160729.54



Il ne reste plus qu'à positionner comme on veut la navette avec le RCS et à sauvegarder le scénario.

9 – AJOUTER UN COSMONAUTE

Il est possible d'ajouter un cosmonaute près d'une station ou d'un vaisseau ou dans une navette pour agrémenter le paysage

La première chose est de se fabriquer un cosmonaute de service en créant un fichier cfg.
Il suffit de copier le fichier Nasa_mmu du dossier CONFIG, de le renommer et de le "bricoler" un peu.
Voilà mon fichier Cosmonaute.cfg

```
; === Configuration file for vessel class Cosmonaute ===
ClassName = nasa_mmu
Module = mmu
EnableFocus = TRUE

; === Attachment specs ===
BEGIN_ATTACHMENT
P 0 0 0  0 -1 0  0 0 1 XS
P 0 -1.1  0  0 -1 0  0 0 1 GS
END_ATTACHMENT
```

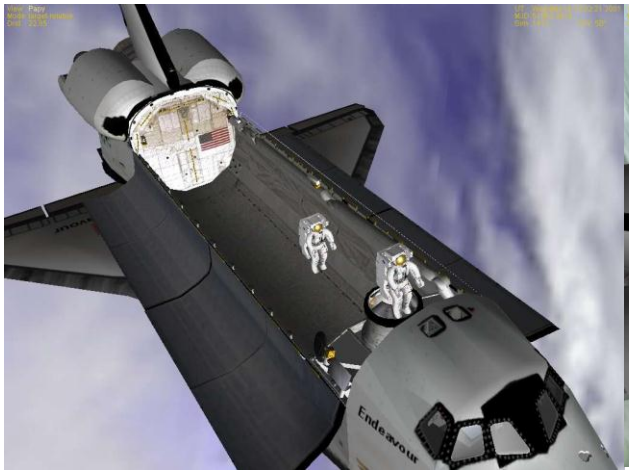
Remarquez que j'ai ajouté EnableFocus = TRUE qui permettra de prendre le contrôle du cosmonaute à l'aide de la fenêtre appelée par F3.
Ceci est valable pour tous les types de vaisseaux !

J'ai ajouté deux points d'attachement:
- un point XS qui permet d'arrimer le cosmonaute dans une soute
- un point GS qui d'agripper le cosmonaute par les pieds avec un bras de navette

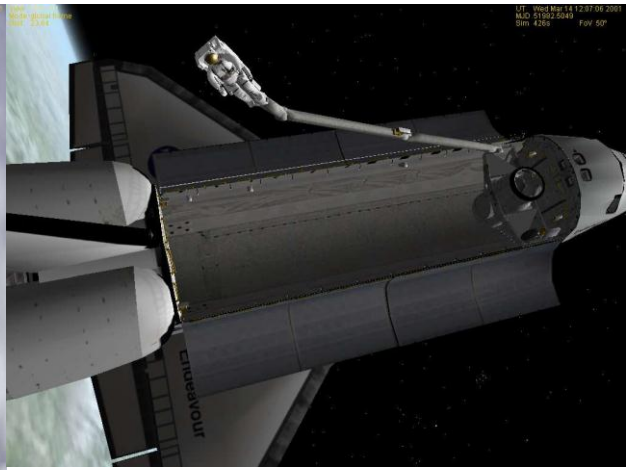
Nous pouvons par exemple placer le cosmonaute dans la navette en écrivant ce qui suit dans le scénario

Il ne faut pas oublier le Fuel pour le cosmonaute si vous voulez pouvoir le manœuvrer après son largage par J puis sa prise en main par F3.

```
BEGIN_SHIPS
STS-101:Endeavour
  STATUS Orbiting Earth
  RPOS 4860858.41 1140979.54 -4384066.58
  RVEL 4883.630 1390.997 5857.353
  AROT 96.69 -74.93 -5.17
  PRPLEVEL 0:1.000
  NAVFREQ 0 0
  CONFIGURATION 3
  CARGODOOR 1 1.0000
  GEAR 0 0.0000
END
Papy:Cosmonaute
  STATUS Orbiting Earth
  ATTACHED 0:0,STS-101
  FUEL 1.0000  ;===ne pas oublier le carburant pour le mmu !
END
END_SHIPS
```



Papy et le mmu de la navette



Papy sur le bras

On peut utiliser en même temps le cosmonaute de la navette mais il est le seul qui peut rentrer. Si on veut amener Papy plus loin, il faut l'agripper avec le bras pour ne pas l'abandonner dans l'espace ce qui serait dommage !

On peut aussi donner au cosmonaute RVEL, RPOS et AROT de la navette et il se trouve au milieu de la soute, mais il n'est pas arrimé. On écrirait dans notre cas

```
Papy:Cosmonaute
  STATUS Orbiting Earth
  RPOS 4860858.41 1140979.54 -4384066.58
  RVEL 4883.630 1390.997 5857.353
  AROT 96.69 -74.93 -5.17
  FUEL 1.0000  ;===ne pas oublier le carburant pour le mmu !
END
```

On peut enfin attacher le cosmonaute au bras (c'est bizarre si le bras est à l'horizontale mais ça marche pour le transport et on peut le largue en release) en écrivant :

```
Papy:Cosmonaute
  STATUS Orbiting Earth
  ATTACHED 0:1,STS-101        ;== le port 1 est le bras
  FUEL 1.0000                 ;===ne pas oublier le carburant pour le mmu !
END
```

On peut ajouter le point GS au cosmonaute de la navette ce qui permet de l'agripper par les pieds si on veut simuler une opération de montage comme ça a était fait dans la réalité.

Il suffit de modifier le fichier Nasa_mmu.cfg pour avoir :

```
; === Configuration file for vessel class Satellite ===
ClassName = nasa_mmu
Module = mmu
; === Attachment specs ===
BEGIN_ATTACHMENT
P 0 -1.1  0  0  -1 0  0 0 1  GS
END_ATTACHMENT
```

9 - APPUIS DES LANCEURS ET AUTRES ELEMENTS

On peut rencontrer des difficultés en utilisant des Pads de lancement non adaptés au lanceur.
Il se peut que la fusée utilisée ne repose pas à la hauteur voulue par rapport au sol ou que le Pad ne soit pas à bonne hauteur par rapport au sol.

La position d'Apollo par exemple peut être corrigée en ajoutant dans sa description une ligne
TCP = hauteur en m de la correction verticale en + ou –

```
BEGIN_SHIPS
AS-506:saturn5nasp
  STATUS Landed Earth
  BASE Cape Canaveral:11
  HEADING 90.00
  FUEL 1.000
  APOLLONO 11
  NASSPVER 50001
  STAGE 5
  STAGESTATUS 0
  DOCKSTATE 0
  MISSNTIME -3600
  TOAPO 185.94
  TOPER 183.16
  TOHDG 72.058
  MOONLAT 23.43
  MOONLONG 0.69
  MOONALT 0.0
  TCP -8.5        ;=== on abaisse de 8,5 m
  LANG English
END
END_SHIPS
```

La position de repos d'éléments décrits par des fichiers ini peut en général être modifiée car ils contiennent une description de 3 points servant à définir la base de sustentation sur la surface ou il repose (c'est l'équivalent des 3 roues d'un train tricycle d'avion)
Par exemple si on prend le Crawler d'Apollo utilisé dans l'excellent Add-on KL-C-39 on trouve dans la section [CONFIG] du fichier CRAW-SAT-V-S5.ini qui définit le Crawler les éléments ci-dessous

```
LAND_PT1=(0,-77,9)
LAND_PT2=(-4,-77,-5)
LAND_PT3=(4,-77,-5)
```

Il s'agit des coordonnées des 3 points de la base théorique d'appui sur le sol (Land Points).
Le nombre du milieu donne la hauteur des points d'appui par rapport à une partie choisie par le programmeur.
En faisant varier cette valeur pour les 3 points on règle la hauteur de la plateforme.
Si on en modifie un seul, on peut voir que le crawler prend une allure de tour penchée. C'est comme si on rallonge un pied à un trépied d'appareil de photo.

Les fichiers ini des satellites possèdent souvent ces points qui permettent de la régler en position sous la coiffe si on veut utiliser un autre lanceur. Voilà celui de Cassini par exemple.

```
LAND_PT1=(0,-1,3)
LAND_PT2=(-3,-1,-3)
LAND_PT3=(3,-1,-3)
```

On peut sûrement trouver d'autres applications je vous laisse le soin de les découvrir…

10 - CONCLUSION

Je pense vous avoir dit l'essentiel de ce qui peut se faire mais vous découvrirez sûrement d'autres possibilités au cours de vos essais.

Dans vos scénarios faîtes très attention à utiliser les deux points, le point virgule et la virgule correctement sinon vous vous exposez à de nombreux plantages.
Ne faites pas de fautes d'orthographe sur les noms des fichiers et des vaisseaux utilisés.

Si un scénario se plante au lancement faites une copie en supprimant tous les éléments Ship sauf 1 et lancez cette copie.
Si ça ne marche pas, ce premier élément est fautif et vous cherchez pourquoi ? (erreur de syntaxe, faute d'orthographe, espace en trop, base de lancement non déclarée, point de docking inexistant …)
Si ça marche rajoutez un élément et refaites l'essai et ainsi de suite jusqu'à trouver l'erreur.

Orbiter crée dans le dossier racine un fichier Orbiter.log qui peut quelquefois aider à trouver les raisons d'un plantage au lancement. Consultez-le !
Vous pouvez l'effacer quand vous voulez, il se régénère.

J'espère que ce document vous sera utile pour réaliser vos scénarios et en particulier ceux qui permettent d'assembler une station.

Bonne chance !

Papyref
Mars 2005