

# Stardust

MiKaël NAVARRO

26 mars 2004

Les notes qui suivent sont distribuées suivant les termes de la Licence de Documentation Libre GNU (GNU Free Documentation License, version 1.2) de la Free Software Foundation <sup>1</sup>.

Cela signifie que vous pouvez copier, modifier et redistribuer ces notes tout à fait librement, pour autant que vous respectiez un certain nombre de règles qui sont précisées dans cette licence.

Pour l'essentiel, sachez que vous ne pouvez pas vous approprier ces notes pour les redistribuer ensuite (modifiées ou non) en définissant vous-même d'autres droits de copie. Les notes que vous redistribuez, modifiées ou non, doivent obligatoirement inclure intégralement le texte de la licence citée ci-dessus, le présent avis, la *Préface* qui suit, ainsi que la section *Mentions légales*. Vous êtes autorisé à demander une contribution financière à ceux à qui vous redistribuez ces notes, mais la somme demandée ne peut concerner que les frais de reproduction. Vous ne pouvez pas redistribuer ces notes en exigeant pour vous-même des droits d'auteur, ni limiter les droits de reproduction des copies que vous distribuez.

Ces notes sont publiées dans l'espoir qu'elles seront utiles, mais sans aucune garantie.

---

<sup>1</sup><http://www.gnu.org/copyleft/fdl.html>

# Préface

*Stardust* est l'aboutissement de divers travaux en OpenGL sous Linux :

Ça a débuté fin 1997, lors d'un projet de 2ème année de l'ESSI, avec *Visu* un moteur de rendu software (Xlib) sous RedHat 5.2.

En juin 1998, avec le développement de Mesa3d, un projet de fin d'année donna naissance à *glView*, dont le but était la visualisation d'images médicales en trois dimensions.

L'année suivante le mini-projet *glSkel* a mis en place les prémices d'un moteur 3d (TombRaider like, usage des quaternions, glut GameMode, réflexion, ombres portés, environnement mapping, brouillard,..). Parallèlement j'ai commencé le développement du projet *gltrack* dans le but d'implémenter un logiciel de visualisation 3d (obj, asc, 3ds, mdl,..).

Toujours développé en C, j'ai commencé le développement d'un moteur d'animation de personnages, *openSkel*, courant 2002 (cinématique directe et motion capture via des scripts BVH).

C'est dans ce contexte que le projet *Stardust* débuta fin 2002, dont le but est la simulation de vols spatiaux...

# Mentions légales

## Auteurs

*MiKaël NAVARRO* <klnavarro@voila.fr, klnavarro@free.fr>  
Auteur / concepteur de Stardust.

*Nicolas BECAVIN* <becavin.nicolas@wanadoo.fr>  
Conseillé technique.

*Simon DEPIETS* <2df@tuxfamily.org>  
Traduction de la documentation en anglais.

*Gilles MARCEAU*  
Conseillé artistique.

## Licence

*Stardust* est fourni sous licence GPL <sup>2</sup>, sans aucune garantie de fonctionnement. De même, les corrections de bugs et mises à jour ne sont pas garanties. Référez-vous au fichier LICENCE (qui doit être inclus dans cette distribution) pour plus de détails sur les termes de la licence.

## Distribution

<http://freshmeat.net/projects/stardust/>

## Remerciements

A mon épouse, pour son encouragement et son aide précieuse.  
A mes amis Sylvain, Iori, pour leur relecture technique, leur conseils et idées.

---

<sup>2</sup><http://www.gnu.org/copyleft/gpl.html>

# Table des matières

<b>I</b>	<b>Présentation</b>	<b>8</b>
<b>1</b>	<b><i>Stardust</i> l'histoire</b>	<b>10</b>
1.1	Fédérations . . . . .	10
1.1.1	Premier Empire . . . . .	10
1.1.2	Second Empire . . . . .	10
1.1.3	Troisième Empire . . . . .	10
1.2	Les Guerres de Zone . . . . .	11
1.2.1	La Guerre Finale . . . . .	11
1.2.2	Guerre thanguienne . . . . .	11
1.2.3	Les Guerres nimiennes . . . . .	11
1.3	L'Alliance . . . . .	12
<b>2</b>	<b>Guide d'installation</b>	<b>13</b>
2.1	Configuration requise . . . . .	13
2.2	Installation à partir des sources (tarball) . . . . .	14
<b>3</b>	<b>Guide de jeu</b>	<b>15</b>
3.1	Objectifs du jeu . . . . .	15
3.2	Briefing . . . . .	16
3.3	Commandes du jeu . . . . .	16
3.4	Navigation . . . . .	16
3.4.1	Pilotage du vaisseau . . . . .	16
3.4.2	HUD . . . . .	19
3.5	Système d'arme . . . . .	20
3.5.1	Attaque . . . . .	20
3.5.2	Défense . . . . .	21
3.6	Gestion de l'énergie . . . . .	21
3.6.1	Gestion des dégâts . . . . .	21
3.6.2	Répartition des ressources . . . . .	22
3.7	Poste de commandement . . . . .	23
3.7.1	Instructions . . . . .	23
3.7.2	Documentations . . . . .	23
3.7.3	Cartes stellaires . . . . .	23

3.7.4	Envoyer des ordres . . . . .	24
3.7.5	t'Chat . . . . .	24
3.8	Debriefing . . . . .	24
<b>4</b>	<b>Guide technique</b>	<b>25</b>
4.1	Le Soudot . . . . .	25
4.2	Production d'énergie . . . . .	26
4.2.1	Bloc réacteur . . . . .	26
4.2.2	Anti-matière . . . . .	26
4.2.3	Accumulateurs . . . . .	26
4.3	Propulsion . . . . .	26
4.3.1	Moteurs conventionnels . . . . .	26
4.3.2	Vitesse Sub-Luminique . . . . .	26
4.3.3	Voyage dans l'Hyper-Espace . . . . .	27
4.4	Armement et dotations . . . . .	27
4.4.1	Accélérateur de particules . . . . .	27
4.5	Torpilles . . . . .	27
4.5.1	Mines antimatière . . . . .	28
4.5.2	Écran au duralium . . . . .	28
4.5.3	Munitions . . . . .	28
<b>5</b>	<b>Guide de combat</b>	<b>29</b>
5.1	Comprendre ses armes . . . . .	29
5.1.1	Options offensives . . . . .	29
5.1.2	Options défensives . . . . .	30
5.2	Comprendre le canon . . . . .	31
5.3	Comprendre les torpilles . . . . .	31
5.4	Comprendre les boucliers . . . . .	31
5.5	Positionner et manoeuvrer le vaisseau . . . . .	31
5.6	Stratégies générales . . . . .	31
<b>6</b>	<b>Manuel de campagne</b>	<b>33</b>
6.1	Engins spatiaux . . . . .	33
6.1.1	Cartactéristiques . . . . .	33
6.1.2	Classification . . . . .	34
6.1.3	Manoeuvrabilité . . . . .	34
6.2	Le voyage interstellaire . . . . .	35
6.2.1	Propulsion . . . . .	35
6.2.2	Le voyage infralumineux . . . . .	36
6.2.3	Vitesse Sub-Luminique . . . . .	38
6.2.4	Voyage dans l'Hyper-Espace . . . . .	40
6.3	Visions d'ailleurs . . . . .	40
6.3.1	Proxylon III . . . . .	40
6.3.2	Elerac . . . . .	41

6.3.3	Trantor . . . . .	41
6.3.4	Zone de la forêt des Étoiles . . . . .	41
6.3.5	Proxima du Centaure . . . . .	41
6.3.6	Mars . . . . .	41
6.3.7	Biodomes . . . . .	41
<b>II Notes techniques</b>		<b>42</b>
<b>7 Conventions C</b>		<b>44</b>
7.1	Modularité . . . . .	44
7.2	Présentation . . . . .	46
7.3	Documentation . . . . .	47
7.4	Règles de nommage . . . . .	49
7.5	Implémentation C . . . . .	49
7.6	Réutilisation . . . . .	51
7.6.1	Modularité . . . . .	51
7.6.2	Robustesse . . . . .	51
7.6.3	Simplicité . . . . .	52
<b>8 Analyse</b>		<b>53</b>
8.1	Cas d'utilisation . . . . .	53
8.2	Scénario . . . . .	55
8.2.1	Un joueur local en mode campagne . . . . .	55
8.2.2	Un joueur local en mode arcade . . . . .	55
8.2.3	Un joueur distant en mode arcade . . . . .	56
8.3	Diagramme d'état . . . . .	57
8.4	Diagramme de classes . . . . .	60
8.4.1	Engine . . . . .	61
8.4.2	Scene . . . . .	61
8.4.3	SpaceObject . . . . .	61
8.5	Structure . . . . .	62
8.5.1	Librairies . . . . .	62
8.5.2	Quelques références . . . . .	63
8.5.3	Quelques exemples . . . . .	63
<b>9 Dynamique dans <i>Stardust</i></b>		<b>64</b>
9.1	Cinématique du point . . . . .	64
9.1.1	Cadre spatio-temporel . . . . .	64
9.1.2	Coordonnées cartésiennes . . . . .	65
9.2	Dynamique du point . . . . .	65
9.2.1	Masse et quantité de mouvement . . . . .	65
9.2.2	Lois fondamentales de la dynamique . . . . .	65
9.3	Équations Différentielles Ordinaires . . . . .	66

9.3.1	EDO avec conditions initiales . . . . .	66
9.3.2	Méthode d'Euler . . . . .	66
9.3.3	Méthode d'Euler Cauchy corrigé . . . . .	66
9.3.4	Méthode de Taylor d'ordre plus élevé . . . . .	67
9.3.5	Méthodes de Runge-Kutta . . . . .	67
9.3.6	Autres méthodes . . . . .	67
9.4	Programmation du moteur cinématique . . . . .	68
9.4.1	Position du problème . . . . .	68
9.4.2	Méthode d'Euler . . . . .	68
9.4.3	Méthode d'Euler Cauchy corrigé . . . . .	69
9.4.4	Méthodes de Runge-Kutta . . . . .	69
<b>10</b>	<b>La flotte</b>	<b>70</b>
10.1	Charte graphique . . . . .	70
10.1.1	Manoeuvrabilité . . . . .	70
10.1.2	Zone d'activité . . . . .	70
10.1.3	Description . . . . .	71
10.1.4	Volumes . . . . .	71
10.2	Fichiers de spécifications . . . . .	71
10.2.1	Specs UNIVERSE . . . . .	71
10.2.2	Specs MISSION . . . . .	72
10.2.3	Specs SCENE . . . . .	72
10.2.4	Specs SHIP . . . . .	72
10.2.5	Specs OBJ . . . . .	75
10.2.6	Blender . . . . .	77
<b>III</b>	<b>Annexes</b>	<b>79</b>
<b>11</b>	<b>The GNU General Public License</b>	<b>80</b>
<b>12</b>	<b>GNU Free Documentation License</b>	<b>88</b>
	<b>Bibliographie</b>	<b>98</b>



**Première partie**

**Présentation**

*Stardust - A Free Space Flight Simulator for a Free Universe !*



FIG. 1 – Independence-War

**Genre** : Simulation spatiale.

**Type** : Simulateur 3D de combat spatial (type Independence-War).

Le but du projet *Stardust* est de créer un simulateur 3D de combat spatial open-source, portable et extensible reproduisant au mieux la cinématique des vols spatiaux.

Ce projet permettra de gérer des dizaines de vaisseaux dans un environnement 3D, de combattre ou coopérer avec d'autres joueurs via la prise en charge du réseau ou encore de combattre contre des ennemis contrôlés par Intelligence Artificielle.

Le code source du projet est entièrement disponible sous GNU General Public Licence<sup>3</sup>.

*Mikaël NAVARRO*

---

<sup>3</sup><http://www.gnu.org/copyleft/gpl.html>

# Chapitre 1

## *Stardust* l'histoire

Il était une fois...

### 1.1 Fédérations

Entre le troisième et le cinquième millénaire inclus, trois Fédérations furent créées à partir des mondes terriens disparates de la galaxie, chacun caractérisé par une “zone franche” propre (systèmes planétaires exploités par le commerce libre), une zone frontière différente comprenant les divers habitants extra-terrestres de la cité stellaire, et aussi par des centres administratifs différents.

De temps à autres, au cours de chaque période fédérative, un Empire terrien était proclamé de sorte que les périodes séparant ces Empires soient appelées, pour plus de commodité, Interrègnes.

Les Empires, ainsi que leurs fondateurs, sont les suivants :

#### 1.1.1 Premier Empire

Proclamé et fondé sur la Terre par Jaar K'rath Mellows, un Terrien. Il dura 530 ans.

#### 1.1.2 Second Empire

Fondé par Kriston Abel Farenzio, un marchand et un politicien bootien, sur Véga. Il entretenait d'étroites relations avec la Terre et dura 400 ans.

#### 1.1.3 Troisième Empire

Proclamé par Birikoff Brishoff Mac Suriyamana, un nomade et un chef religieux célèbre.

Le Troisième Empire naquit sur Véga mais, lorsque cette dernière se déclara Principauté indépendante (la paix fut signée sur ces entrefaites à la suite de

la Seconde Convention Ursienne), le centre de l'Empire se transporta tout d'abord sur le monde artificiel de Mechanistra puis gagna le système solaire, plus particulièrement Mars.

## **1.2 Les Guerres de Zone**

Nom collectif désignant les quatorze conflits qui se déroulèrent entre la fin du cinquième millénaire et celle du sixième.

La plupart des querelles opposaient des Terriens, toutefois certaines races extra-terrestres furent entraînées dans les conflits, notamment les Fabériens qui, à l'époque de la sixième guerre de Zone, possédaient quelques intérêts dans un certain nombre de monde de la Fédération (ou de l'Empire).

L'ensemble des guerres de Zone s'étala sur soixantes-dix années.

### **1.2.1 La Guerre Finale**

Elle se déroula principalement dans le système de Tucana M. A cette occasion, furent utilisés Mechanistra et d'autres mondes satellites artificiels dotés d'une considérable puissance de feu.

La Guerre Finale, qui dura quelques dix années terrestres, fut la plus acharnée de toutes les quatorze Guerres de Zone; elle débuta par une insignifiante querelle à propos d'une planète industrielle mais se transforma bientôt en un gigantesque conflit opposant les Humains aux Fabériens et à leurs alliés de l'Empire sarosien.

### **1.2.2 Guerre thanguienne**

Livrée dans l'espace environnant le système de la Lyre à l'Empire thanguien, petite enclave étrangère dans l'Espace impérial qui souhaitait transformer ses planètes colonisées afin de modifier leur environnement. Le Troisième Empire déclara la guerre, et vainquit les Thanguiens lors de quatre longues, tragiques (et extrêmement profitables pour les vainqueurs) confrontations.

### **1.2.3 Les Guerres nimiennes**

Les historiens écrivent qu'il s'agit d'un conflit avec des extra-terrestres en apparence pacifiques ayant atteint un très haut niveau de recherche en matière temporelle. L'Empire, redoutant de ne plus pouvoir lui-même maîtriser le temps, attaqua les planètes nimiennes faisant ainsi disparaître la menace. L'Empire déclara que trois mondes terriens avaient été détruits par une flotte nimienne et que l'attaque du monde nimien n'était qu'une guerre de représaille. On estime maintenant que les planètes terriennes furent détruites par l'Empire lui-même qui utilisa à cet occasion des vaisseaux nimiens capturés.

### 1.3 L'Alliance

L'Empereur fabérien Je'Kiriath, membre de l'énigmatique caste des Epines, fut considéré comme l'un des plus illustres empereurs fabériens, il fut assassiné par un groupe de militaires dissidents qui furent à leur tour abattus par des troupes anti-républicaines et des habitants de la capitale Kurix.

Dès sa mort, à mesure que sa politique anti-colonialiste à l'égard des mondes frontières s'effaçait, les premières tensions entre les Empires terrien et fabérien se firent jour.

C'est dans ce contexte que naquit le mouvement indépendantiste, l'Alliance, composé d'un mélange de pirates, de terroristes et de représentants d'organisations politiques clandestines.

Comme par le passé, de nouvelles colonies de pionniers se sont établies sur de nouveaux territoires afin d'en exploiter ses ressources, mais comme autrefois, les intérêts de la Terre et ceux des colonies se sont affrontés. L'Alliance concrétise les aspirations de ces colons en matière d'autodétermination.

## Chapitre 2

# Guide d'installation

### 2.1 Configuration requise

A l'heure actuelle seule la Version GNU/Linux est maintenue :

- GNU/Linux(i386, kernel 2.4.x, XFree86 4.1.0 ou plus);
- Processeur 400 MHz avec 64 Mo de RAM;
- Carte son;
- Clavier; Joystick/GamePad optionnel.

Vous devez posséder OpenGL<sup>1</sup> (OpenGL 1.2 ou plus | Mesa<sup>2</sup> 3.4 ou plus) ainsi qu'une carte accélératrice 3D supportant OpenGL.

Dépendances :

- libSDL<sup>3</sup> 1.2.1;
- zlib<sup>4</sup> 1.1.4 (support des fichiers gzippés);
- libxml2<sup>5</sup> 2.4.19 (documents xml);
- libtiff<sup>6</sup> 3.5.5 (images au format tiff, recommandé).

Pour compiler à partir des sources :

- autoconf et autoheader 2.13
- automake 1.4
- make 3.79.1
- gcc 2.95 ou 3.0+

---

<sup>1</sup><http://www.opengl.org>

<sup>2</sup><http://www.mesa3d.org>

<sup>3</sup><http://www.libsdl.org>

<sup>4</sup><http://www.zip.org/zlib/>

<sup>5</sup><http://xmlsoft.org/>

<sup>6</sup><http://www.libtiff.org>

## 2.2 Installation à partir des sources (tarball)

Décompresser les sources avec :

```
~> gzip -dc stardust-x.x.x.tar.gz | tar xvf -
```

Puis :

```
~> cd stardust-x.x.x
~> ./configure && make
~> su
~> make install
```

Pour lancer Stardust il suffit alors de taper :

```
~> stardust
```

Et pour obtenir l'aide :

```
~> stardust --help
```

# Chapitre 3

## Guide de jeu

### 3.1 Objectifs du jeu

*Stardust* comporte deux mode de jeu : le mode *Campagne* et le mode *Arcade*.

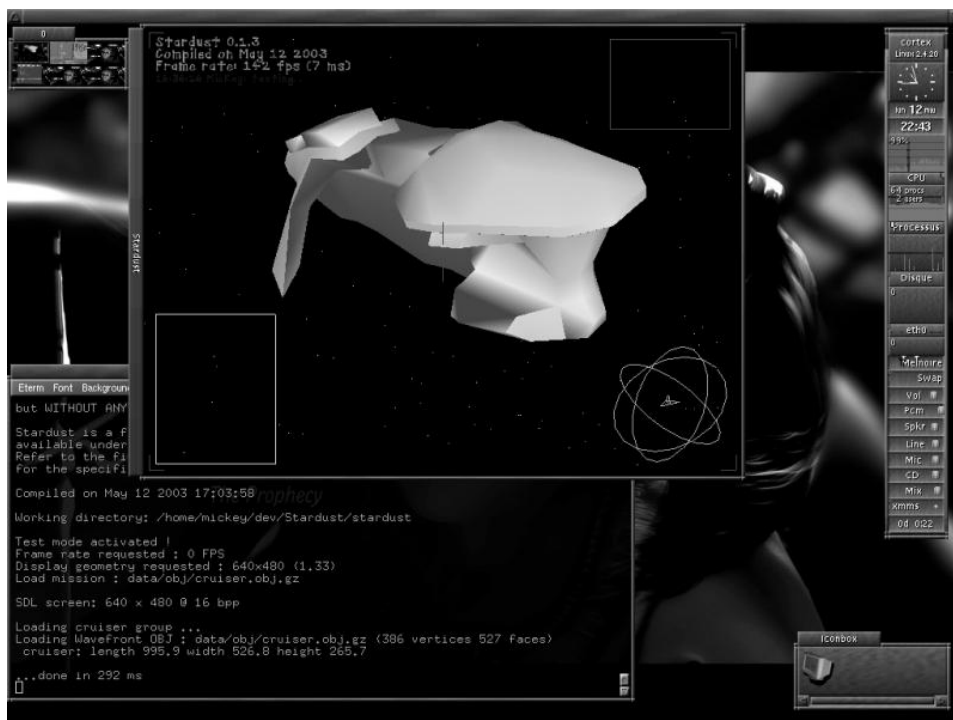


FIG. 3.1 – Stardust



En mode Campagne le jeu se subdivise en missions qui rentre dans le cadre d'un scenario.

Pour réussir, vous devrez accomplir tous les objectifs de cette mission. Toutefois, certaines missions ont des objectifs cachés qui peuvent vous rapporter des points supplémentaires et/ou changer le cours de l'histoire.

En mode Arcade, comme en mode Campagne, vous devrez accomplir des missions mais vous aurez la possibilité de coopérer et/ou combattre avec d'autres joueurs via le net.

## 3.2 Briefing

Avant de commencer une mission, la première chose à faire est de consulter le Briefing. Ce document contient vos ordres de mission sous forme de texte, en outre il vous détaillera le contexte de la mission.

## 3.3 Commandes du jeu

Pause	P
Quitter	Esc
Options	Shift + O

TAB. 3.1 – Commandes du jeu

Pour changer de vue :

Vue avant (vue à partir du vaisseau du joueur)	Home
Vue poursuite (vue depuis l'arrière du vaisseau)	End
Vue circulaire (rotation de la caméra)	Alt + flèches

TAB. 3.2 – Changement de vue

## 3.4 Navigation

### 3.4.1 Pilotage du vaisseau

Le système de commandes de Stardust est très similaire à celui d'un simulateur de vol classique, mais contrairement à un avion, un vaisseau spatial n'a pas besoin d'avancer pour changer son orientation.

Le vaisseau peut se déplacer selon n'importe quelle direction et obéir à plusieurs modes de pilotage. En mode assisté (le plus simple) le vaisseau se déplacera principalement vers l'avant.

Tangage arrière	2* / ↓
Tangage avant	8* / ↑
Roulis à gauche	4* / ←
Roulis à droite	6* / →
Lacet à gauche	1*
Lacet à droite	3*
Interchanger axe roulis et lacet	Y
Augmenter la vitesse	+(*)
Baisser la vitesse	-(*)
Accélérer	Q
Décélérer	W

TAB. 3.3 – Commandes de pilotage

\* Touches du clavier numérique.

(\*) Ajustement de la vitesse préprogrammé.

Les manoeuvres d'un vaisseau spatial se font en utilisant les propulseurs à réaction conventionnels. Le gaz ou le plasma expulsé permet de produire un effet de réaction. Cela permet d'accélérer et de virer, mais pour maintenir une maniabilité élevée une vitesse de moins de  $2000m.s^{-1}$  est conseillé.

### Modes de propulsion

Le mouvement Newtonien d'un vaisseau dans l'espace, pouvant évoluer librement dans toutes les directions pose parfois des problèmes aux pilotes, même les plus expérimentés. Par conséquent, tous les vaisseaux sont équipés d'un mode de pilotage assisté utilisant les propulseurs pour annuler le glissement et maintenir la vitesse avant du vaisseau.

Mode de pilotage assisté/libre	M
--------------------------------	---

TAB. 3.4 – Pilotage assisté

En mode libre, le système d'assistance est désactivé.

### Propulseurs Latéraux

Ces commandes sont utilisés pour des manoeuvres de combat particulières ou pour positionner le vaisseau au millimètre près.

Latéral haut	Shift + 8* / Shift + ↑
Latéral bas	Shift + 2* / Shift + ↓
Latéral gauche	Shift + 4* / Shift + ←
Latéral droit	Shift + 6* / Shift + →

TAB. 3.5 – Propulseurs latéraux

\* Touches du clavier numérique.

### Mode Sub-Luminique

Le mode de déplacement Sub-Luminique permet d'atteindre des vitesses de l'ordre de 33 % de la vitesse de la lumière.

Mode sub-luminique (on/off)	L
-----------------------------	---

TAB. 3.6 – Mode Sub-Luminique

Ce mode est utiliser pour parcourir des distances inter-planétaires.

### Hyper-Espace

L'Hyper-Espace permet à un vaisseau de faire des sauts quasi instantanné entre différents systèmes solaires.

Saut dans L'Hyper-Espace (jump)	J
---------------------------------	---

TAB. 3.7 – Hyper-espace

Pour passer en hyper-espace, vous devez traverser un point de transit, appelé *porte des étoiles*. Pour se faire vous devez satisfaire les conditions :

- Avoir une vitesse entre  $100$  et  $2500m.s^{-1}$  ;
- Activer le mode hyper-espace (J) ;
- et enfin traverser la porte (dans le bon sens).

### Pilotage automatique

L'ordinateur de bord possède, en outre, des fonctions pré-programmées vous assistant lors d'opérations de pilotage de base :

Arrêt auto	F5
Approche auto (de la cible)	F6
Formation auto (avec la cible)	F7
Amarrage auto (avec la cible)	F8

TAB. 3.8 – Pilotage automatique

### 3.4.2 HUD

Les données de pilotage les plus importantes sont affichées sur le HUD, c'est-à-dire un écran holographique dans le champ de vision du pilote; Il offre :

- Le mode de pilotage ainsi que la vitesse ;
- L'état actuel des systèmes du vaisseau ;
- Un indicateur montrant les contacts en cours ;
- Les détails du contact principal ;
- Un radar tri-dimensionnel (orbe) ;
- Une grille de référence en 3D ;
- Un historique des traînées (vaisseaux, torpilles) ;
- Un viseur pour le canon...

Bascule de l'affichage du HUD	H
Consoles	F1-F4

TAB. 3.9 – HUD

Rq : Si le vaisseau est touché, il se peut que le HUD disparaisse brièvement.

#### Grille de référence

Le HUD projette un quadrillage dans le champ de vision du pilote, ce qui permet à ce dernier d'apprécier les distances ainsi que sa vitesse.

#### Contacts

Tous les vaisseaux, stations ou débris situés à portée de capteur sont enregistrés dans l'ordinateur comme contact :

Aliens	Bleu
Ennemis	Rouge
Contacts non-identifiés	Jaune
Vaisseaux neutres	Vert

TAB. 3.10 – Contacts

Le HUD affiche en plus sur le contact en cours :

- Le nom du Contact ;
- La distance du contact en kilomètres ;
- Les dégats recus (en %).

Désigner le contact central	T
-----------------------------	---

TAB. 3.11 – Désigner le contact

## Orbe

L'orbe, quant à lui, est un système radar en 3D capable d'indiquer la position des contacts à la fois devant, derrière, au dessus et au dessous du vaisseau. Le HUD représente, sous forme d'une sphère, l'espace qui entoure le vaisseau (à portée de senseur) ; Celui-ci se trouve au centre de la sphère. Les contacts y sont représentés par des points lumineux, adoptant le code de couleur décrit précédemment.

## 3.5 Système d'arme

L'artilleur est en charge de ce poste.

### 3.5.1 Attaque

Pour tirer sur un vaisseau ennemi, le processus est le suivant :

1. Sélectionner le système d'arme ;
2. Désigner la cible ;
3. Tirer.

Sélectionner le Canon	Entrée
Sélectionner les Torpilles	Retour Arrière
Désigner le contact central comme cible	T
Désigner l'ennemi le plus proche	R
Tir Canon ou Torpille	Espace

TAB. 3.12 – Attaque

### Principales armes

- Le Canon : très efficace à courtes portées (moins de  $5km$ ), ses effets sont presque nuls au delà de  $10km$  ;
- Les Torpilles Autopropulsés : elles se verrouillent sur la signature de la cible.

### 3.5.2 Défense

Leurre CME	Suppr
Bouclier instantané	Insr

TAB. 3.13 – Défense

### Boucliers et leures

- Coque en duralium : cet écran absorbe puis ionise les particules le long de la coque du vaisseau ;
- Les Leures CME (Contre Mesures Electroniques) sont utilisés pour tromper les missiles.

## 3.6 Gestion de l'énergie

Ce poste donne accès aux systèmes du vaisseau et permet de répartir les ressources entre tel ou tel système.

### 3.6.1 Gestion des dégâts

#### Entre 100% et 70%

Entre 100 et 70% de la structure initiale, ca va, rien à signaler ! La peinture est à refaire, mais le vaisseau n'est pas en danger...

### **Entre 70% et 50%**

Aïe ! Les choses se gâtent ! Mais pas de panique encore... L'hyper-espace et le mode sub-luminique sont inutilisables.

### **Entre 50% et 30%**

Les choses deviennent préoccupantes, des avaries mineures (fuites, incendies, fumées...) vous gâchent la vie.

### **Entre 30% et 10%**

Le vaisseau devient aussi manoeuvrable qu'un fer à repasser, les moteurs rendent peu à peu l'âme, la dépressurisation commence.

### **Entre 10% et 1%**

Le vaisseau est mort... Il ne peut ni bouger, ni tirer, ni atterrir. Un vrai cercueil volant.

### **À 0**

Le vaisseau n'est plus qu'une carcasse éventrée et glacée qui dérive au hasard des courants spatiaux.

## **3.6.2 Répartition des ressources**

L'afficheur d'état est un petit moniteur montrant l'état du système :

- TEMP : accumulateur de chaleur dans le dispositif de dissipation ;
- POWER : puissance développée par le réacteur ;
- FUEL : quantité de carburant en réserve ;
- HULL : état de la coque (en %).

L'interface suivante permet de répartir les ressources d'énergie entre les principaux systèmes :

Augmenter la puissance des boucliers	Ctrl + ↑
Augmenter la puissance des armes	Ctrl + →
Augmenter la puissance des propulseurs	Ctrl + ←

TAB. 3.14 – Répartition des ressources

## **3.7 Poste de commandement**

C'est à partir de là que le maître à bord supervise les opérations et a accès à des fonctions de commandements spécifiques.

### **3.7.1 Instructions**

A tout moment le commandant du vaisseau peut avoir accès à ses ordres de mission, ainsi qu'à un récapitulatif des objectifs à atteindre. Il est utile de consulter la liste des objectifs de mission pour les passer en revue et voir ce qui a été réalisé et ce qu'il reste à accomplir.

### **3.7.2 Documentations**

L'ordinateur du vaisseau possède une base de données contenant un certain nombre de documents : les fichiers du personnel à bord, ainsi que d'autres informations sur les types d'armes, les missiles, le fonctionnement des boucliers,...

### **3.7.3 Cartes stellaires**

La base de données stellaire contient les coordonnées précises de plus de 5000 étoiles répertoriées. Ces dernières sont nécessaires pour paramétrer les sauts dans l'hyper-espace.



### 3.7.4 Envoyer des ordres

Vous pouvez diriger les aliers à l'aide d'un simple code numérique :

Rapport des aliers	1
En formation avec moi	2
Attaquez ma cible	3
Défendez ma cible	4
En formation avec ma cible	5
Stop	6
Retrait	7
Ammarez-vous avec ma cible	8

TAB. 3.15 – Envoyer des ordres

### 3.7.5 t'Chat

L'équipage d'autres vaisseaux peut avoir besoin de vous parler. Pour converser avec eux utilisez la commande :

Envoyer un message	Tab + texte
--------------------	-------------

TAB. 3.16 – t'Chat

## 3.8 Debriefing

Cet écran vous présente un récapitulatif de vos performances et la liste des objectifs atteints et ceux restant à accomplir.

# Chapitre 4

## Guide technique

### 4.1 Le Soudot

Corvette de classe Soudot, également appelé ■ l'Inviolable ■.

#### Caractéristiques principales

Classe :	Soudot
Catégorie :	Corvette
Type :	CS-525x
Manoeuvrabilité :	élevée
Puissance :	$1,5 \cdot 10^9 W$
Rayon d'action :	Inter-stellaire
Autonomie :	3 <i>mois</i>
Équipage :	40 + 4
Longueur :	151 <i>m</i>
Tonnage à vide :	9500 <i>t</i>
Volume frêt :	100 <i>m</i> <sup>3</sup>

#### Caractéristiques détaillés

Accélération maximale :	$7m.s^{-2}$
Vitesse maximale :	$50km.s^{-1}$
Vitesse sub-luminique :	0.377 <i>c</i>
Systèmes offensifs :	1 Cannon Wingrove 2 tourelles Harvey Lances torpilles
Systèmes défensifs :	Coque au duralium Leurres CME
Systèmes de détection :	Portée 100000 <i>km</i>
Systèmes de propulsion :	Anneau à plasma

## 4.2 Production d'énergie

### 4.2.1 Bloc réacteur

Construit à partir de cellules accélératrices, qui poussent les particules sub-atomiques, le bloc réacteur produit du plasma et de l'énergie nucléaire.

Le plasma est canalisé pour produire la puissance de propulsion, l'énergie nucléaire sert, quant à elle, à produire l'énergie électrique nécessaire pour l'entretien de l'accélérateur de particules et la maintenance des autres modules.

### 4.2.2 Anti-matière

Les réserves d'anti-matière sont habituellement faibles, mais une grande quantité d'énergie peut être produite en très un temps très court.

### 4.2.3 Accumulateurs

Les accumulateurs consistent en des réserves d'énergie électrique, pas suffisante pour maintenir le stockage de l'anti-matière ou du plasma mais suffisant pour maintenir les modules de survie ou établir des communications pendant plusieurs jours.

## 4.3 Propulsion

### 4.3.1 Moteurs conventionnels

Selon les principes de la physique Newtonienne, les propulseurs fonctionnent sur le principe d'action/réaction, le plasma chargé d'énergie est émis à très grande vitesse et canalisé par des déflecteurs.

Ce système permet une grande maniabilité et de fortes accélérations ( $> 10g$ ).

### 4.3.2 Vitesse Sub-Luminique

Ce système a pour effet de modifier l'espace-temps sur une zone localisée, une bulle, autour du vaisseau. L'espace-temps est comprimé à l'avant du vaisseau et étiré à l'arrière, ce qui provoque de minuscules sauts qui répétés, des millions de fois par seconde, permet les voyages interplanétaires.

Ce procédé requiert une énorme quantité d'énergie, qui est la plupart du temps fournie en brûlant une certaine quantité d'anti-matière afin de modifier l'espace-temps dans une zone localisée autour du vaisseau.

### 4.3.3 Voyage dans l'Hyper-Espace

Ni les propulseurs classiques, ni la vitesse sub-luminique ne permettent de se déplacer sur des distances interstellaires, le mode hyper-espace, quant à lui, permet de franchir des distances considérables en un temps infime.

#### Saut dans l'Hyper-Espace

La théorie repose sur le fait que l'espace est replié, en des endroits appelés *trous de vers*, et communique avec d'autres points de l'espace.

Cette distorsion de l'espace, qui permet à un vaisseau de franchir des distances considérables en un temps infime, produit un effet négatif sur la masse de l'astronef en convertissant les micro-particules en particules d'anti-matière et cela à taux constant ; lorsque plus d'un pour cent des micro-particules ont subi cette transformation, le navire devient instable.

#### Les portes des étoiles

Les *portes des étoiles* sont des brèches dans le tissu de l'espace, généralement associées à un affaiblissement naturel de la pesanteur et parfois à la distorsion de champs spaciaux, provoquée par une distorsion de l'hyperespace et des effets spatio-temporels.

La plupart des portes des étoiles sont maintenant localisées, connues et gardées par les forces terriennes et fabériennes associées. Certaines portes des étoiles s'enfoncent dans l'Univers et font encore l'objet d'explorations.

## 4.4 Armement et dotations

### 4.4.1 Accélérateur de particules

Utilisé par la plupart des vaisseaux de la Fédération, ce type d'arme offre une grande puissance de feu pendant un court laps de temps.

Utilisé lors de brefs engagements, il lui est substitué, pour les combats plus long, par les faisceaux Harvey, montés sur tourelles et dont le tir est assisté par ordinateur.

## 4.5 Torpilles

Elles accélèrent très vite jusqu'à atteindre sa vitesse maxi et ont une portée de 100000km. Elles se calent sur la signature de la cible.

### 4.5.1 Mines antimatière

Particules explosives d'antimatière, renfermées dans une coque métallique se désintégrant à la moindre vibration et caractérisée par un champ de distorsion.

### 4.5.2 Écran au duralium

Écran protecteur fixé à la coque des vaisseaux de guerre et créé par la désintégration subatomique phasée du duralium.

### 4.5.3 Munitions

Le Soudot comporte trois soutes, deux soutes principales et une soute secondaire, sous la section principale de la coque, ne pouvant accueillir que des missiles disrupteurs (2) et des leurres (8).

Les deux soutes principales ont une capacité limitée de 4 packs pouvant accueillir :

Leurres CME (Balises CME)	8 / pack
Torpilles	4 / pack
Disrupteurs	2 / pack

TAB. 4.1 – Dotations

En outre d'autres dotations (missiles, sondes) peuvent être accroché sur des points de montage externe, au nombre de 4.

# Chapitre 5

## Guide de combat

*“Avant d’étudier les faiblesses de l’ennemi apprenez à connaître votre propre vaisseau.”*

### 5.1 Comprendre ses armes

#### 5.1.1 Options offensives

##### **Canon Wingrove**

Le tir n’est pas assisté mais la puissance de la salve est très importante (1 tir/min, portée efficace 5km).

##### **Canon à plasma (Harvey)**

Arme tirant avec précision une salve de particules directement prélevé sur l’accélérateur de particules.

Monté sur tourelle, la conduite de tir est assistée par ordinateur. On distingue trois types de tourelles :

- Tourelles légères : 1 tir/3s, portée efficace 2km ;
- Tourelles standards : 1 tir/6s, portée efficace 5km ;
- Tourelles lourdes : 1 tir/min, portée efficace 12km (vaisseaux de +500m).

##### **Torpilles**

Missiles à tête chercheuse munis d’une tête nucléaire à micro fusion lourde. Elles se calent sur la signature de la cible et permettent d’atteindre un objectif hors d’atteinte des lasers.

### **Missiles Disrupteur**

Missile désactivant le mode de propulsion Sub-Luminique et perturbant les systèmes électroniques grâce à une décharge d'énergie sous forme de vibrations électromagnétiques.

### **Conteneur d'Anti-matière**

Larger ses conteneurs d'anti-matière peut être utilisé comme une arme, mais son emploi peut s'avérer incontrôlable.

### **Eperonnage**

Le pilote peut éperonner le vaisseau adverse en surchargeant ses boucliers. La manoeuvre est délicate et risque d'endommager aussi son vaisseau.

### **Ordres aux Aliens**

Communiquer avec ses alliés pour répartir et coordonner l'attaque.

## **5.1.2 Options défensives**

### **Boucliers**

Champs d'énergie (duralium) protégeant le vaisseau.

### **Leurre CME**

Contre mesures électroniques permettant de leurrer les torpilles.

### **Canon**

Le canon peut atteindre les torpilles adverses, mais cela reste difficile à mettre en pratique.

### **Obstruction**

S'abriter derrière un objet.

### **Manoeuvres Evasives**

Eloignement du danger.

### **Mode Sub-Luminique**

Fuir le danger. Risque d'être désactivé par des missiles disrupteurs.

## **Aliers**

Utiliser un alier pour vous soutenir.

## **5.2 Comprendre le canon**

- Approchez-vous de la cible, les dégâts sont d'autant plus importants que vous êtes près de la cible ;
- Utilisez un canon Wingrove seulement si vous êtes sûre d'atteindre la cible ;
- Gardez votre ennemi dans votre champ de tir ;
- Essayez de vous trouver sur l'arrière des vaisseaux ennemis ;
- Aachevez les vaisseaux endommagés.

## **5.3 Comprendre les torpilles**

- Utilisez les torpilles pour affaiblir les cibles ;
- Utilisez la fonction tire et oublie pour mettre plusieurs ennemis hors de combat ;
- Utilisez les torpilles si la cible est hors d'atteinte des canons.

## **5.4 Comprendre les boucliers**

- Surveillez vos arrières, vous êtes vulnérables sur l'arrière ;
- Utilisez les deux boucliers (en divisant les ennemis) ;
- Utilisez le bouclier instantané pour profiter de l'effet pare-chocs ;
- Alternez les boucliers pour éviter leur épuisement.

## **5.5 Positionner et manoeuvrer le vaisseau**

- La distance est une défense ;
- Surveillez vos arrières ;
- Gardez une vitesse moyenne raisonnable pour garder une bonne maniabilité ;
- Utilisez le mode libre pour effectuer des manoeuvres de renversement ou pour mitrailler les gros vaisseaux ;
- Esquivez avec les propulseurs latéraux.

## **5.6 Stratégies générales**

- Abbatez les vaisseaux les plus faibles en premier ;
- Aachevez le vaisseau que vous avez commencé à attaquer ;



- Utilisez vos alliés ;
- Si vous êtes endommagé, suivez bien la flotte (pour profiter de son soutien).

## Chapitre 6

# Manuel de campagne

### 6.1 Engins spatiaux

#### 6.1.1 Caractéristiques

**Classe :**

**Catégorie :**

**Type :**

**Manoeuvrabilité :**

**Puissance :**

**Rayon d'action :**

**Autonomie :**

**Équipage :**

**Longueure :**

**Tonnage à vide :**

**Volume fret :**

#### Caractéristiques détaillés

**Accélération maximale :**

**Vitesse maximale :**

**Vitesse sub-luminique :**

**Systèmes offensifs :**

**Systèmes défensifs :**

**Systèmes de détection :**

**Systèmes de propulsion :**

### **6.1.2 Classification**

Ce facteur, qui se rattache à la maniabilité, sépare les vaisseaux en catégories de tailles et de limites d'accélération.

#### **Type 1**

Engins de 10 à 50m de long ; chasseurs, navettes scouts, intercepteurs, petits yachts. On y trouve aussi des vaisseaux plus grands où la masse est compensée par la puissance des propulseurs : croiseurs d'interception.

#### **Type 2 à 4**

Entre 50 et 200m, les constructeurs proposent des modèles conservant une certaine rapidité de réaction. On y trouve la majorité des véhicules qui sillonnent l'espace : transports (3), cargos (3 ou 4), navettes et barges sol/orbitales (2 ou 3), destroyers (3), yachts de luxe (3), croiseurs de combat (4),...

#### **Type 5 à 10 (et plus)**

Poids lourds de l'espace, les paquebots, destroyers lourds et autres long courriers, longs de 300m à 1 ou 2km. Les vaisseaux de ce type ne se posent jamais sur une planète.

#### **Type X**

Cailloux lancés dans l'espace, les engins de type X n'ont aucune capacité de manoeuvre. Fusée porteuse, station orbitale ou vaisseau-monde.

### **6.1.3 Manoeuvrabilité**

#### **Extrême**

Chasseurs légers où la manoeuvre est anticipée d'une fraction de secondes par les ordinateurs en fonction des commandes manuelles et oculaires du pilote.

#### **Élevée**

Vaisseaux équipés pour le combat ; Vaisseaux militaires d'interception, appareils d'exploration ou pirates et contrebandiers.

#### **Moyenne**

Engins spatiaux lourds (cargos), ou ceux pour qui la propulsion sert à voyager sans chercher à faire d'accrobaties (yachts...).

## **Faible**

Barges ou quelques navettes de bas prix.

## **6.2 Le voyage interstellaire**

### **6.2.1 Propulsion**

#### **Propulsion chimique**

La combustion de l'hydrogène et de l'oxygène produit des gazs dont la détente pousse la fusée. Elle engendre une poussée très importante mais avec un rendement médiocre (des quantités prodigieuses de carburant sont nécessaires).

#### **Propulsion thermonucléaire**

Un flot d'hydrogène liquide traverse un réacteur à coeur nucléaire solide, où le gaz est chauffé à plus de  $2500K$  avant d'être éjecté à grande vitesse par la tuyère de la fusée.

La propulsion nucléaire fournit deux fois plus de quantité de mouvement que les fusées chimiques les plus performantes, et les réacteurs produisent de l'électricité pour le véhicule spatial.

#### **Propulsion ionique**

Un gaz, comme le césium ou le xénon, passe dans une chambre où il est ionisé par un canon à électrons. Une tension électrique appliquée entre deux grilles de métal extrait les ions positifs, qui traversent la grille et sont éjecté dans l'espace. Simultanément une cathode, à l'arrière du moteur, évacue les électrons dans un faisceau d'ions, de manière que le véhicule spatial n'accumule pas de charges négatives.

#### **Propulsion à effet Hall**

Les moteurs à effet Hall utilisent un champ électrique pour accélérer des particules chargées positivement (en général, des noyaux de xénon). Le moteur crée différemment le champ électrique : un anneau d'aimants engendre d'abord un champ magnétique radial, qui fait circuler les ions autour de lui. Ce déplacement crée alors un champ électrique axial, qui assure l'éjection.

#### **Propulsion magnétohydrodynamique**

Ces moteurs accélèrent des particules chargées en utilisant des champs magnétiques plutôt que des champs électriques. Le système est composé d'une anode creuse, avec tige centrale formant cathode. Une différence de potentiel

ionise le gaz propulseur et engendre un puissant courant électrique radial, qui s'écoule à travers le gaz et le long de la cathode. Ce courant engendre un champ magnétique circulaire, qui interagit avec le courant passant dans le gaz et accélère les particules dans une direction axiale.

Le gaz propulseur peut être de l'argon, du lithium ou de l'hydrogène, par ordre croissant de rendement.

## **VASIMIR**

Le Variable Specific Impulse Magnetoplasma Rocket, ou fusée à magnétoplasma à impulsions spécifiques variables, comble la lacune entre systèmes à fortes et faible poussée. De l'hydrogène est d'abord ionisé par des ondes radio, puis guidé vers une chambre centrale où l'on fait régner un champ magnétique. Là, les particules longent les lignes de champ magnétique en tournant autour d'elles à une fréquence qui ne dépend que de leur charge électrique et de leur masse. En bombardant les particules avec des ondes radio à la même fréquence, on les chauffe jusqu'à des températures de dix millions de Kelvins. Une tuyère magnétique transforme enfin le déplacement en spirale en déplacement axial, ce qui engendre la poussée. En réglant le chauffage et le convertisseur magnétique de sortie, le pilote commande la vitesse d'éjection.

## **Les voiles solaires**

Les voiles solaires sont poussées par la faible pression des vents solaires.

## **Moteur thermochimique à double flux**

Système de propulsion conventionnel des vaisseaux de guerre et de transport des Premières et Seondes Fédérations. Ce mécanisme devait son extraordinaire rendement à l'alternance de deux flux : désintégration et régénération du borinium / phase hostile d'intervention de l'antimatière.

Il lui est maintenant préféré les anneaux accélérateur à plasma, procurant une plus grande autonomie.

### **6.2.2 Le voyage infralumineux**

Les deux types de navire, décrits ci-dessous, ne permettent qu'une colonisation des étoiles les plus proches de celle du monde d'origine.

## **Les vaisseaux à génération**

Dans lesquels s'entassent des familles entières, qui partent pour un voyage sans retour.

### **Les mange-poussières**

Beaucoup plus efficaces, ils montent à des vitesses proches de celle de la lumière. Les lois de la physique font que le temps propre du navire est comprimé. Une étoile située à 10 années-lumière sera atteinte au bout de 10 ans, mais l'équipage aura vieilli de quelques mois au pire.

Leur nom vient du fait qu'ils se ravitaillent en avalant les poussières du milieu interstellaire.

### 6.2.3 Vitesse Sub-Luminique

#### L'énergie négative

Avant d'aller plus loin, précisons que l'*énergie négative* n'est pas l'antimatière, dont l'énergie est positive. Quand un électron heurte son anti-particule (le positon), ils se détruisent mutuellement, produisant des rayons gamma, dont l'énergie est positive.

#### Bulle d'espace-temps

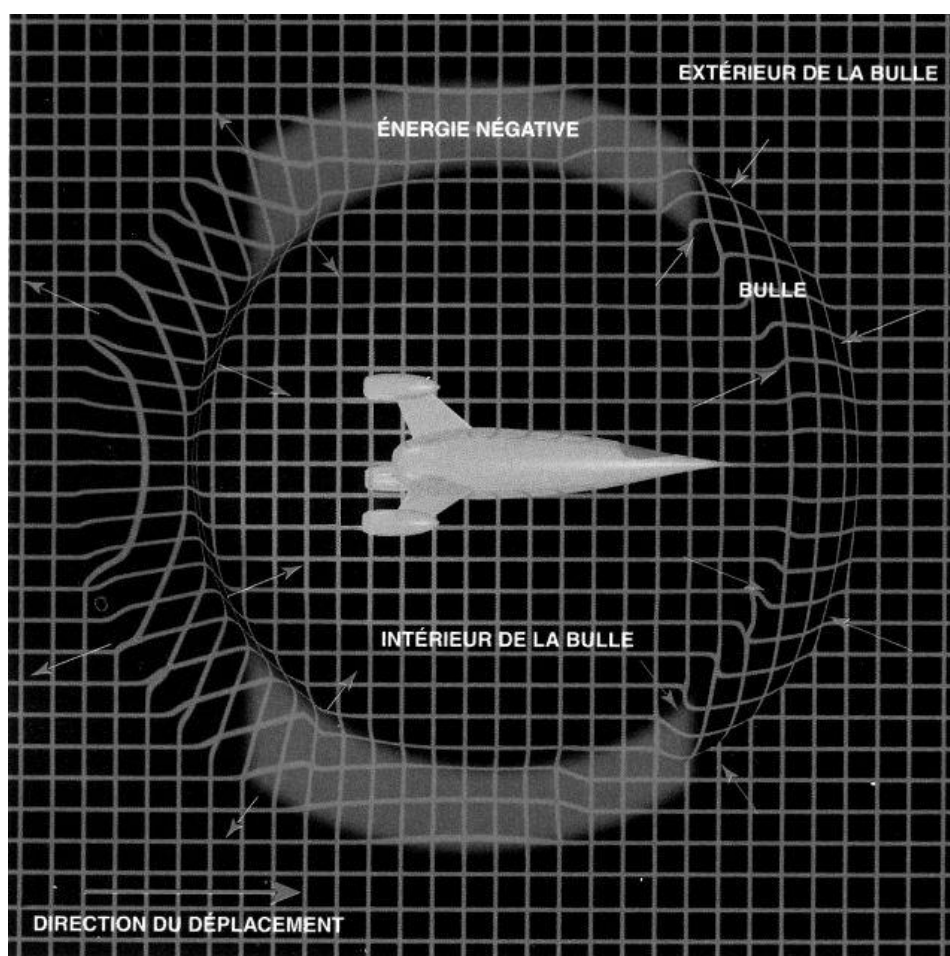


FIG. 6.1 – Bulle d'espace-temps

L'espace-temps est contracté à l'avant de la bulle, réduisant ainsi la distance qui la sépare de sa destination, et dilaté à l'arrière, accroissant la distance qui la sépare de son point de départ. Le vaisseau lui-même ne bouge pas par

rapport à l'espace immédiatement autour de lui, les membres d'équipage ne ressentent aucune accélération.

Une telle bulle nécessite de l'énergie négative, sur les flancs de la bulle.

### **Einvélocité**

Mesure agréée de la vitesse relative espace-temps. Dérivée de la relativité einsteinienne.

### **Ziemen**

Unité de mesure de l'interaction de la matière (énergie-espace à vitesse élevée) proportionnelle au déplacement temporel qui peut ainsi atteindre un point critique. Certains champs d'énergie toutefois créent des déplacements beaucoup plus importants sans que pour autant le temps ne s'accroisse.

### **Facteur C**

Un vaisseau croisant à une vitesse sub-luminique a les plus grandes difficultés à maintenir une vitesse s.-t. constante sous les effets de l'onde gravitationnelle, du flux spatial et de la distorsion spontanés. Le facteur C est obtenu par l'équilibre entre l'einvélocité et la capacité de fluctuation calculée en Ziemens, auxquelles viennent s'ajouter la masse, le temps et l'énergie intrinsèque émise.



## 6.2.4 Voyage dans l'Hyper-Espace

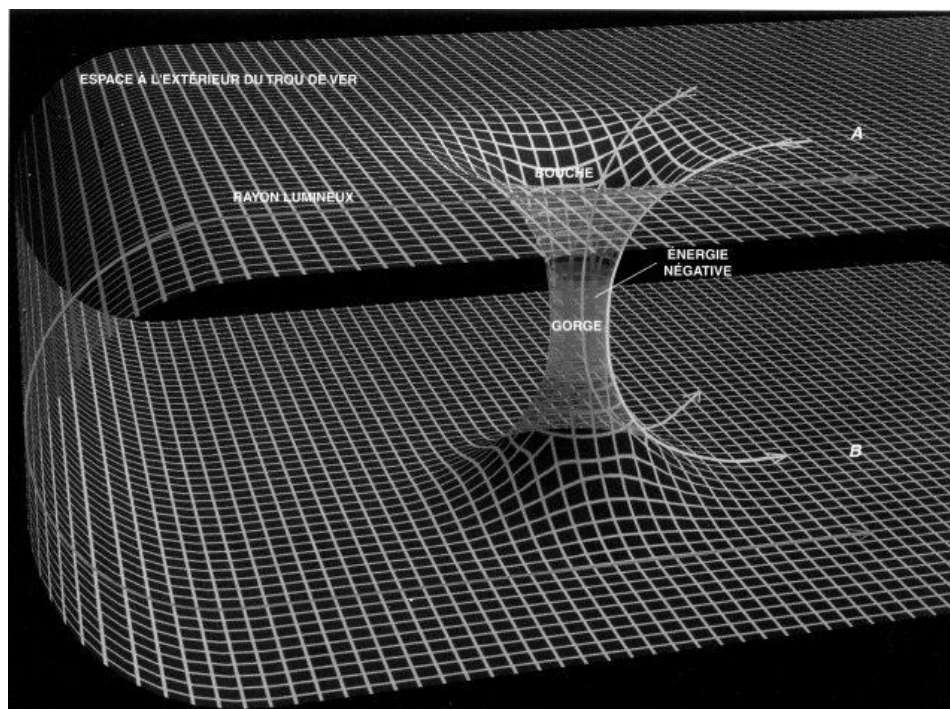


FIG. 6.2 – Trou de ver

Un *trou de ver* est un tunnel entre deux régions éloignées de l'espace. Les rayons lumineux peuvent entrer par une des ouvertures et ressortir de l'autre. La propagation serait beaucoup plus longue si le rayonnement n'empruntait pas ce raccourci.

Dans la gorge, de l'énergie négative doit être maintenu pour empêcher le trou de ver de s'effondrer.

## 6.3 Visions d'ailleurs

### 6.3.1 Proxylon III

Planète colonisée par la Terre, bien que dépourvue de toute vie sauvage, possédait une atmosphère et des caractères morphologiques de type terrien, parmi lesquels des océans profonds et des fosses marines. Transformée en une base militaire, elle comprenait 400 installations mi-civiles, mi-militaires autonomes.

### **6.3.2 Elerac**

Principal complexe industriel d'armement des Mondes Indépendants ; Usines pour la plupart édifiées sur la planète inhabitée de Weston Fall.

### **6.3.3 Trantor**

Monde colonisé entièrement recouvert de métal et d'édifices. La planète devint le centre administratif de la galaxie, mais sa vulnérabilité lui valut d'être détruite durant le Second Empire.

### **6.3.4 Zone de la forêt des Étoiles**

[A faire...]

### **6.3.5 Proxima du Centaure**

[A faire...]

### **6.3.6 Mars**

[A faire...]

### **6.3.7 Biodomes**

[A faire...]

**Deuxième partie**

**Notes techniques**

Mes espoirs sont de créer un simulateur de combats spatiaux en 3D orienté :

**Action** : Le joueur devra avoir la possibilité de se déplacer librement dans un environnement 3D (grâce au clavier et/ou souris/joystick), de faire feu avec des lasers ou de tirer des torpilles. Il aura à sa disposition un *head-up-display* lui indiquant la position de ses ennemis (orbe 3D) ainsi que l'état de son vaisseau...

**Et stratégie** ; En permettant au joueur de donner des ordres à ses alliés ou en jouant en coopération avec d'autres joueurs.

Cette analyse est une nouveauté pour moi, aussi veuillez m'excuser auprès des puristes d'UML et de la conception Objet.

Vous remarquerez d'ailleurs que cette analyse est très orientée fonctionnel, on n'efface pas plusieurs années d'analyse fonctionnelle comme cela...

Ce projet est décomposé en 3 phases de développement, chacune de ces phases portant un nom comme spécifié ci-dessous :

**StarHunter** : Mise en place de l'environnement 3D, déplacements, collisions, son...

**StarFighter** : Interactions, simulation des combats spatiaux, intelligence artificielle.

**StarDust** : Mise en réseau ; Mise en place du mode coopératif via le net.

Les diagrammes ont été fait avec Dia<sup>1</sup> d'Alexander Larsson. Je remercie également Alexander Rawass pour son projet TuxFleet<sup>2</sup> qui m'a grandement aidé.

---

<sup>1</sup><http://www.lysator.liu.se/alla/dia/dia.html>

<sup>2</sup><http://tuxfleet.sourceforge.net>

## Chapitre 7

# Conventions C

Ce chapitre décrit des standards de programmation en C destiné à écrire des applications de manière plus efficace, plus facile à maintenir et des applications portable qui prennent en compte la contrainte de performance.

Le C ANSI (ou ISO) est obligatoire. Il existe aujourd'hui des compilateurs pour toutes les machines. Les autres implémentations de C sont tolérées uniquement sur des machines spécifiques ou plus ou moins obsolètes.

La version ANSI du C est plus sûre que les versions précédentes. Toutefois, de par sa nature même, le langage C est un langage dangereux (c'est d'abord le langage d'écriture d'Unix). Il est donc conseillé d'utiliser un outil comme "purify" pour traquer les erreurs sur les pointeurs (fuites de mémoire, etc.).

### 7.1 Modularité

Nous aborderons une approche de type objet avec une programmation C que nous qualifierons de modulaire (sans héritage).

Une unité de compilation C ANSI doit comporter une interface implémentée sous forme d'un fichier d'inclusion (module.h) déclarant les données et fonctions (types, constantes, macros, prototypes de fonctions) et d'un corps (module.c) définissant les données et l'implémentation des fonctions.

La programmation modulaire est indispensable pour garantir la cohérence, dans l'application, aussi bien en ce qui concerne la déclaration des données que l'appel des fonctions.

Un module est vu comme un composant qui offre un ensemble de services. Il doit notamment comporter :

- Une fonction de création :  
`Module *module_new(void)`  
`{...}`
- Une fonction de terminaison :  
`int module_destroy(Module`  
`*module) {...}`
- Et une fonction d'initialisation :  
`int`  
`module_init(Module *module) {...}`

Les fichiers d'inclusion qui déclarent des fonctions ou des variables externes doivent être inclus dans le fichier qui définit ces fonctions et variables. De cette manière, le compilateur peut alors vérifier que l'utilisation de chaque fonction est conforme à sa déclaration.

Toute fonction qui figure dans le corps du module et dont la déclaration ne figure pas dans l'interface ne doit être utilisée qu'à l'intérieur du module. Pour éviter tout risque d'utilisation externe et pour augmenter les performances, il faut les déclarer en `static`.

Il est préférable d'éviter l'utilisation de variables globales externes car elles nuisent à la conception objet et à la réutilisabilité.

Pour éviter les variables externes, il faut concevoir les applications dans une approche objet. Les données sont alors internes aux modules et accédées par des méthodes de lecture et d'écriture.

Pour des questions de performances, il peut être nécessaire d'assouplir cette règle. Dans ce cas, on utilisera des macros pour l'accès aux variables externes pour simuler l'approche objet.

## Interface

L'interface doit contenir suffisamment d'informations pour quiconque appelle le module. La structure d'un fichier d'interface (`module.h`) est la suivante :

<cartouche d'en tete>

```
#ifndef _MODULE_H
#define _MODULE_H
```

```

<inclusions (standard C)>
<autres inclusions (applicatif)>

<constantes>
<macros>
<types exportes>

<declarations des prototypes et des macros d'accès>

#endif /* _MODULE_H */

```

## Implémentation

La structure type d'un fichier d'implémentation (module.c) est la suivante :

```

<cartouche d'en tete>

<inclusions (standard C)>
#include "module.h"
<autres inclusions (applicatif)>

<definitions locales (constantes, macros, types, variables static)>
<definitions des fonctions statics>

<definitions des fonctions>

```

## 7.2 Présentation

- Dans un cadre général, il faut éviter les fichiers trop gros, une taille maximale de 1000 lignes semble raisonnable, ainsi que la longueur des lignes des fichiers source ne doit pas dépasser 80 colonnes ;
- Les commentaires doivent être placés judicieusement et décrire un algorithme, un hardcode, a bug, ...
 

```

/* Commentaire simple autorise */
/* Commentaire developpe sur
   plusieurs lignes autorise. */
/*
 * Commentaire developpe sur plusieurs lignes autorise.
 * (generalement utilise pour les prototypes).
 */

```
- Il ne devrait y avoir qu'une seule instruction par ligne, cela facilite le débogage ;
- Éviter les affectations emboîtées ;
- L'opérateur "?" ne doit être utilisé que dans des cas simples ;

- Les instructions composés doivent être de préférence formatés comme ci-dessous :
 

```

if(condition) {
    ...
}
else { /* commentaire */
    ...
} /* end if */
while(condition) {
    ...
} /* end while */
do {
    ...
while(condition);
for(...) {
    ...
} /* end for */

```
- Et pour l'éguillage :
 

```

switch (expression) {
    case A:
        ...
        break; /* si pas de break, mettre une explication */
    case B:
        ...
        break;
    ...
    default: /* obligatoire, meme s'il est vie */
        ...
        break;
} /* end switch */

```
- Les typedefs doivent être formatés comme les instructions :
 

```

typedef struct _str {
    ...
} str_t;

```
- L'indentation souhaité est de 2 espaces à partir de la colonne 1 ;
- Les commentaires doivent être indentés comme les instructions.

### 7.3 Documentation

Il faut penser à mettre un copyright et la licence GPL en en-tête de chaque fichier source, par exemple dans le cadre du projet Stardust :

```

/*
 * Stardust :-)

```



```

* A Free Space Flight Simulator for a Free Universe!
* Copyright (C) 2003 MiKael NAVARRO
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License as published by
* the Free Software Foundation; either version 2 of the License, or
* (at your option) any later version.
*
* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.
*
* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
*/

```

L'ensemble des commentaires représente la documentation du système. Une partie de ces commentaires (en-tête de fonctions) doit pouvoir être extrait automatiquement.

On peut automatiser la génération d'en-tête et l'extraction de commentaires avec "sed" ou "awk" sous GNU/Linux.

Il peut aussi être intéressant de mettre en valeur certains points spécifiques :

- TODO : description d'une section à compléter
- BUG : [BugID] description d'un bug connu
- TRICKY : description d'une astuce
- WARNING : description d'un point délicat
- COMPILER : codage permettant de contourner un problème de compilateur

Les en-tête de modules doivent être de la forme :

```
/*
 * NOM: Module.
 * ROLE: Description du role du module (objet).
 */
```

Et les en-tête de fonctions :

```
/*
 * Description du role de la fonction.
 * Resultats.
 * Pre/post conditions.
 */
int any_function(void);
```

## 7.4 Règles de nommage

- Seul les caractères alphanumériques (a-z, A-Z, 0-9) sont autorisés dans les identifiants;
- Les identifiants ne doivent pas dépasser 31 caractères (portabilité);
- L'usage du tiret (-) est interdit, ainsi que le souligné (\_) en début d'identificateur (réservé système);
- Chaque identificateur doit décrire ce qu'il est ou ce qu'il fait;
- Tous les identifiants (fonctions, variables, exceptés les noms de typedef) doivent être en minuscule, séparés par des soulignés (\_);
- Tout identifiant utilisé en dehors de son module (fonctions) doit être préfixé par le nom de ce module : `module_any_function`;
- Les noms de struct ou de typedef doivent être écrits en minuscule, sans espaces ni souligné, avec des majuscules en début de chaque mot (ex : `SpaceObject`);
- L'identificateur d'une fonction doit désigner une action. Les fonctions de debug ou de trace doivent être préfixé de "debug\_" ou "trace\_";
- Le nom d'un booléen doit correspondre à son sens quand il vaut TRUE.
- Les pointeurs sont préfixés par le caractère 'p';
- Les types déclarés par un typedef devraient être suffixé par '\_t';
- Les données constantes devraient être préfixées par le caractère 'k';
- Les énumérations et les macros doivent être en majuscule séparé par des soulignés.

## 7.5 Implémentation C

Utiliser les constructions C ANSI (plus sûres et portables) :

- Utiliser la signature des fonctions C ANSI;
- Déclarer 'void' une fonction qui ne retourne pas de résultats;
- Utiliser systématiquement sizeof pour le calcul des tailles de données (surtout dans malloc);
- Toute instruction switch doit comporter les cas default;
- Attention aux 'unions' qui sont peu portables;
- Éviter l'utilisation des pointeurs génériques (void \*);
- Éviter l'utilisation du qualificateur 'volatile'.

## Généralités

- Éviter d'utiliser le goto, à part, pour le traitement d'erreur;
- Ne pas utiliser de constantes littérales dans le code;
- Les données constantes doivent être déclarés 'const', cela permet au compilateur d'effectuer des contrôles supplémentaires et des optimisations;
- Ne pas dupliquer le code, créer plutôt une fonction ou éventuellement une macro;
- Attention les types 'unsigned' (sauf 'unsigned int') sont très dépendant du compilateur;
- Préférer le type 'unsigned int' pour les compteurs de boucle;
- Testez les nombres réels avec les opérateurs <= ou >= mais jamais avec == ou !=, en raison des arrondis au cours des calculs;
- Initialisez vos données, dès la déclaration (en particulier les pointeurs);
- Préférer les variables locales aux variables globales;
- Définir plusieurs variables intermédiaires n'est pas toujours mauvais pour les performances, croyez l'optimisateur local;
- L'usage du cast doit être limité;
- Les compilateurs utilisent des conventions différentes pour retourner les structures. Les pointeurs sur les structures ne posent pas de problèmes;
- Vérifier les codes de retour du système;
- Utiliser avec prudence les appels systèmes comme 'exit' et 'system';
- Vérifier les données externes à l'aide des préconditions ('assert').

## Fonctions et macros

Dans la mesure du possible on se restreindra à l'utilisation des fonctions C ANSI. Dans certains cas on tolérera les fonctions POSIX. Les fonctions qui ne sont ni ANSI C, ni POSIX, ni BSD socket et ni IPV SV sont à proscrire.

Une macro de type fonction doit être encadrée de parenthèses, ceci permet de l'utiliser dans les expressions, les appels de fonctions, ...

Une macro de type procédure doit être déclaré dans un bloc 'do { ... } while(0)'.

## 7.6 Réutilisation

Il est nécessaire d'avoir fait une bonne conception avant de coder.

Dans tous les cas il faut documenter les particularités et dépendances dues au contexte.

En complément des règles des chapitres précédents, les principales qualités favorisant la réutilisation sont la modularité, la robustesse et la simplicité.

### 7.6.1 Modularité

La modularité favorise l'autonomie des composants vis à vis du contexte. Il est donc plus facile ensuite d'extraire et utiliser ces composants comme des "boîtes noires" sans se soucier de leur implémentation.

- Restreindre l'utilisation des variables globales. Minimiser les références externes. La référence à un objet global impose un contexte requis sur l'utilisation d'une fonction. Souvent l'information peut être passée en paramètre.
- Encapsuler les données. Les données sont encapsulées si elles ne sont pas accessibles de l'extérieur du module où elles sont déclarées (non extern). Les structures de données internes sont spécifiques à un algorithme de fonction. Si vous les exportez, vous limitez la flexibilité de l'algorithme.
- Minimiser les fonctions exportées. Les fonctions exportées sont visibles de tous les modules. Lorsqu'une fonction exportée est utilisée par d'autres modules, il est coûteux de changer son interface. Les fonctions déclarées static sont internes à un module. Elles peuvent être détruites ou modifiées avec un impact limité sur le reste de l'application.

### 7.6.2 Robustesse

Lorsqu'on veut réutiliser des composants on cherche avant tout des composants qui ne plantent pas, même en recevant des paramètres inappropriés.

- Se protéger contre les erreurs. Un logiciel doit se protéger contre les données utilisateurs incorrectes ; Des données utilisateurs incorrectes ne doivent jamais provoquer de plantage. Il faut aussi se protéger contre les codes d'erreurs des services système.
- Valider les arguments. Les fonctions externes, celles accessibles par les autres modules, doivent vérifier rigoureusement leurs arguments pour éviter le plantage (utilisation de pré-conditions). N'incluez pas d'arguments qui ne puissent pas être validés.
- Utiliser un mécanisme de Traces. Il est souhaitable de mettre en place un mécanisme de Traces dès le début du codage. L'ajouter à posteriori lorsqu'on a des problèmes de mise au point est fastidieux.

### 7.6.3 Simplicité

Pour faciliter la réutilisation et la maintenance d'un composant, il faut en faciliter la compréhension. Il est donc préférable d'éviter la complexité découlant d'une exploitation trop poussée des spécificités d'un langage de programmation. En général la simplicité a aussi une bonne influence sur la robustesse.

- Limiter le nombre de paramètres.
- Éviter les imbrications trop profondes d'instructions; Cela nuit à la lisibilité du code.
- Créer des fonctions cohérentes et compréhensibles. Une fonction est compréhensible si quelqu'un d'autre que son créateur peut comprendre son code. Garder des fonctions petites ( $\leq 100$  lignes) et cohérentes.
- Optimiser le code après que le programme fonctionne et seulement si c'est nécessaire.

# Chapitre 8

## Analyse

Première ébauche.

### 8.1 Cas d'utilisation

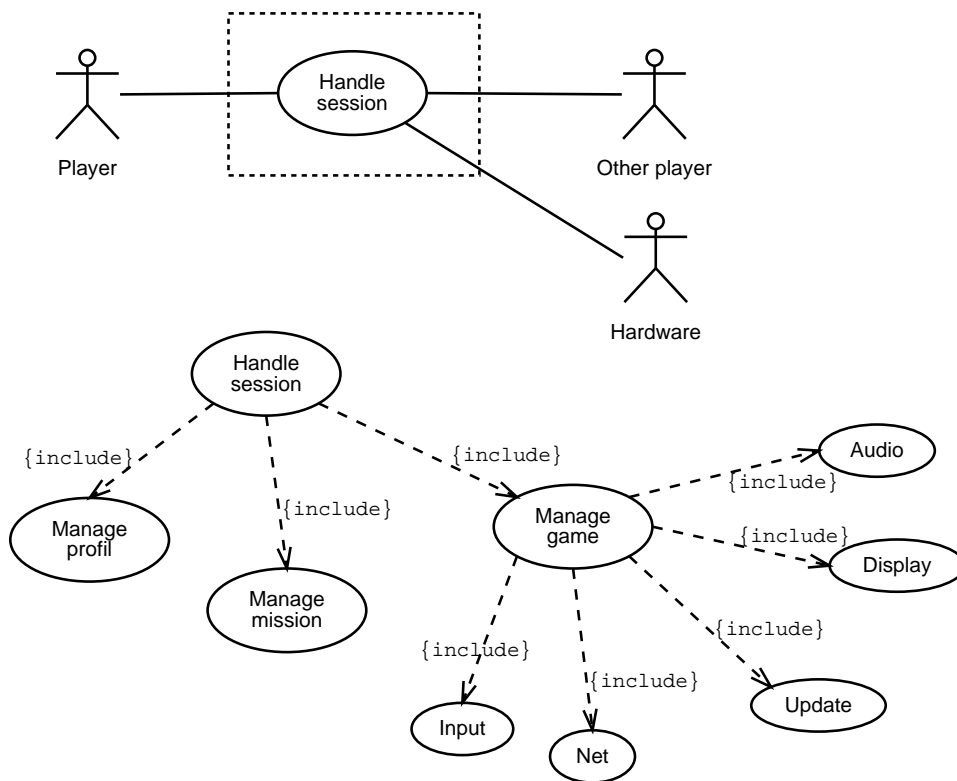


FIG. 8.1 – Cas d'utilisation

## **Input**

Reçoit les commandes (keyboard (attention aux claviers non-US), joystick, gamepad, mouse) du joueur et les enregistrent pour un usage ultérieur (libSDL).

## **Network**

Informe les autres joueurs (ordinateurs) sur un réseau de l'état de l'engine.

## **Update**

C'est le cerveau du jeu.

## **Display**

Prend avantage de l'accélération 3D pour l'affichage (libSDL + OpenGL).

## **Audio**

Musique d'ambiance (ogg vorbis), sons événementiels (OpenAL).

## 8.2 Scénario

### 8.2.1 Un joueur local en mode campagne

Précondition : Le système est fonctionnel.

1. Le scénario commence au lancement de l'application ;
2. Le joueur choisi de jouer en mode campagne ;
3. Le système affiche les profils disponibles ;
4. Le joueur crée son profil ;
5. Le système présente la liste des missions déjà effectués et propose de nouvelles missions ;
6. Le joueur choisi une mission ;
7. Le système charge les données de la mission, crée l'univers, affiche le briefing et le jeu commence ;
8. Le joueur interagit avec le monde ;
9. Le système réagit en conséquence ;
10. Le joueur accomplit tout les objectifs de mission ;
11. Le système sauvegarde le profil, affiche le debriefing et propose de nouvelles missions ;
12. Le joueur décide d'arrêter ;
13. Le jeu s'arrête.

Postcondition : Le système est fonctionnel.

### 8.2.2 Un joueur local en mode arcade

Précondition : Le système est fonctionnel.

1. Le scénario commence au lancement de l'application ;
2. Le joueur choisi de jouer en mode arcade ;
3. Le système demande au joueur de s'identifier ;
4. Le joueur s'identifie ;
5. Le système affiche les connections disponibles ;
6. Le joueur décide de démarrer une session serveur ;
7. Le système affiche une liste de missions multi-joueurs ;
8. Le joueur choisi une mission ;
9. Le système affiche l'état des connections ;
10. Le joueur valide les connections clientes ;
11. Le système charge les données de la mission, crée l'univers, affiche le briefing et le jeu commence ;



12. Les joueurs interagissent avec le monde ;
13. Le système réagit en conséquence ;
14. Les joueurs accomplissent tout les objectifs de mission ;
15. Le système affiche le debriefing et reaffiche la liste des missions ;
16. Le joueur décide d'arrêter ;
17. Le jeu s'arrête.

Postcondition : Le système est fonctionnel.

### **8.2.3 Un joueur distant en mode arcade**

Précondition : Le système est fonctionnel.

1. Le scénario commence au lancement de l'application ;
2. Le joueur choisi de jouer en mode arcade ;
3. Le système demande au joueur de s'identifier ;
4. Le joueur s'identifie ;
5. Le système affiche les connections disponibles ;
6. Le joueur décide de démarrer une session client ;
7. Le système affiche une liste de "squad" ;
8. Le joueur rejoint une équipe ;
9. Le système affiche l'état des connections ;
10. Le système charge les données de la mission, crée l'univers, affiche le briefing et le jeu commence ;
11. Les joueurs interagissent avec le monde ;
12. Le système réagit en conséquence ;
13. Les joueurs accomplissent tout les objectifs de mission ;
14. Le système affiche le debriefing et reaffiche la liste de "squad" ;
15. Le joueur décide d'arrêter ;
16. Le jeu s'arrête.

Postcondition : Le système est fonctionnel.

### 8.3 Diagramme d'état

#### Handle session

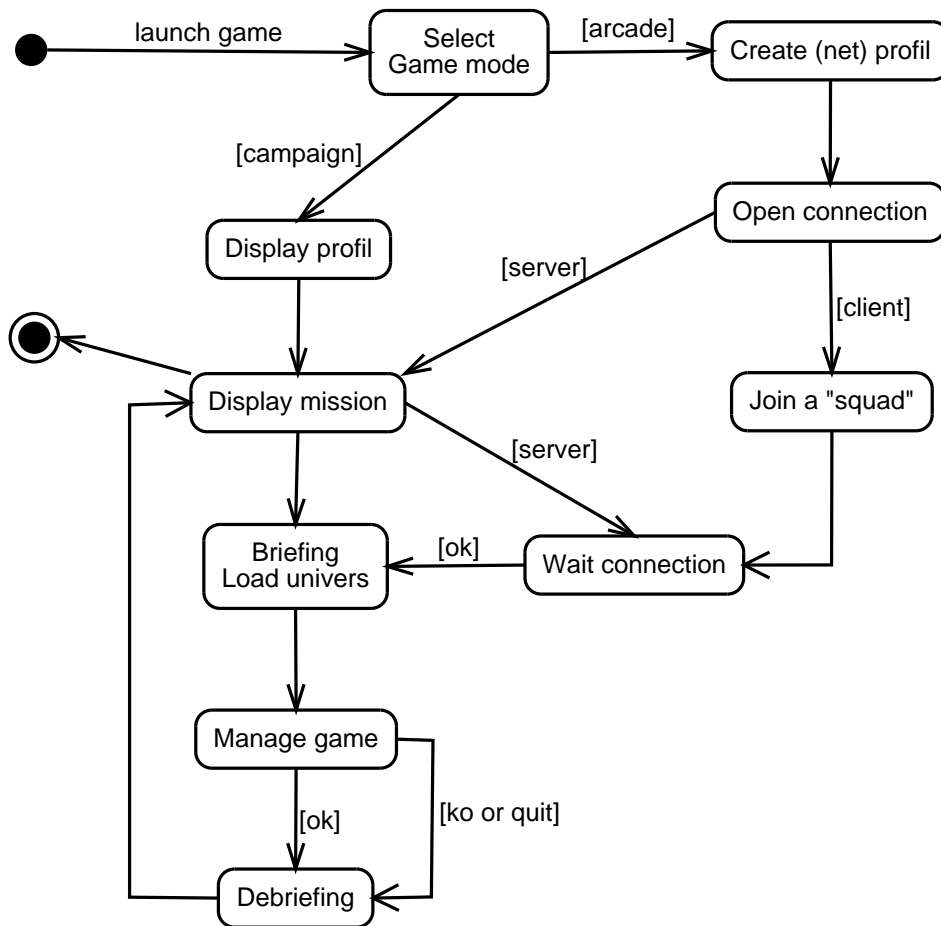


FIG. 8.2 – Diagramme d'état

Game loop (local)

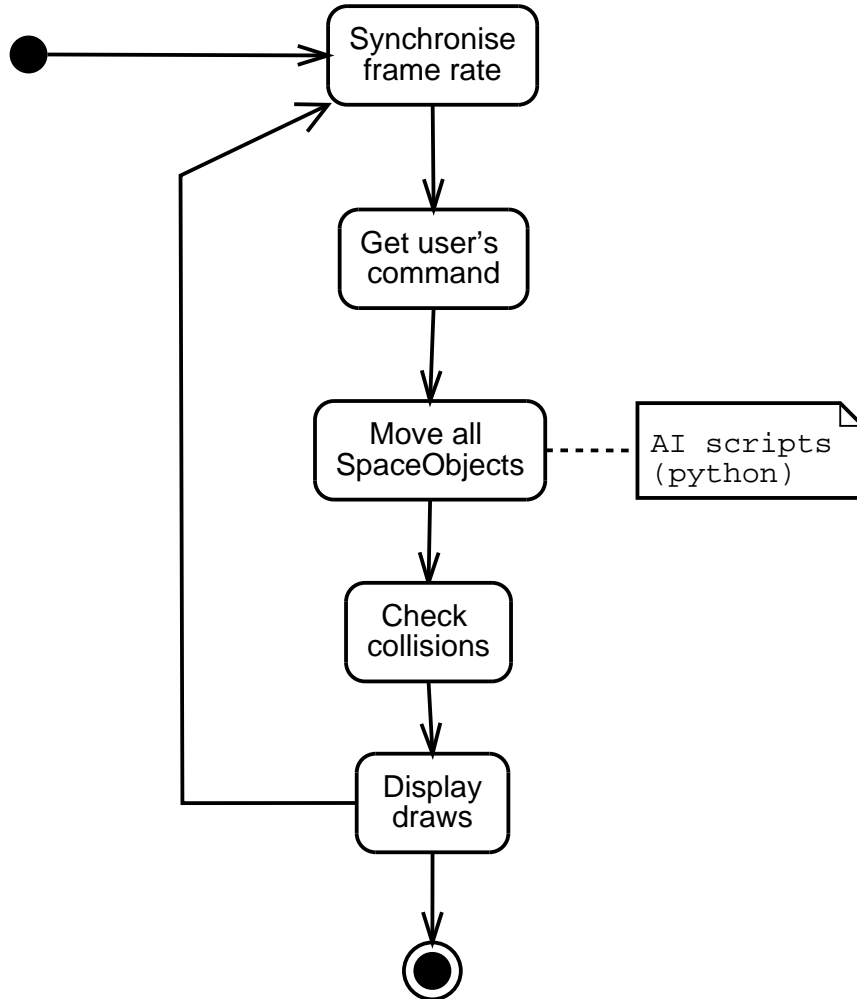


FIG. 8.3 – Game loop (local)

### Game loop (client/server)

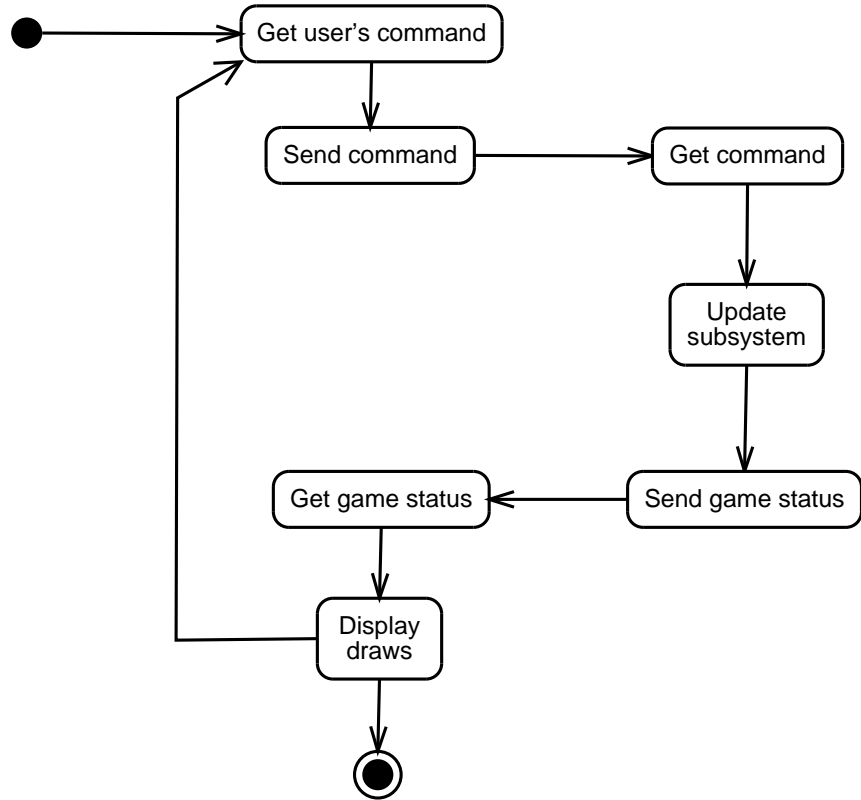


FIG. 8.4 – Game loop (client/server)

[A faire...]

## 8.4 Diagramme de classes

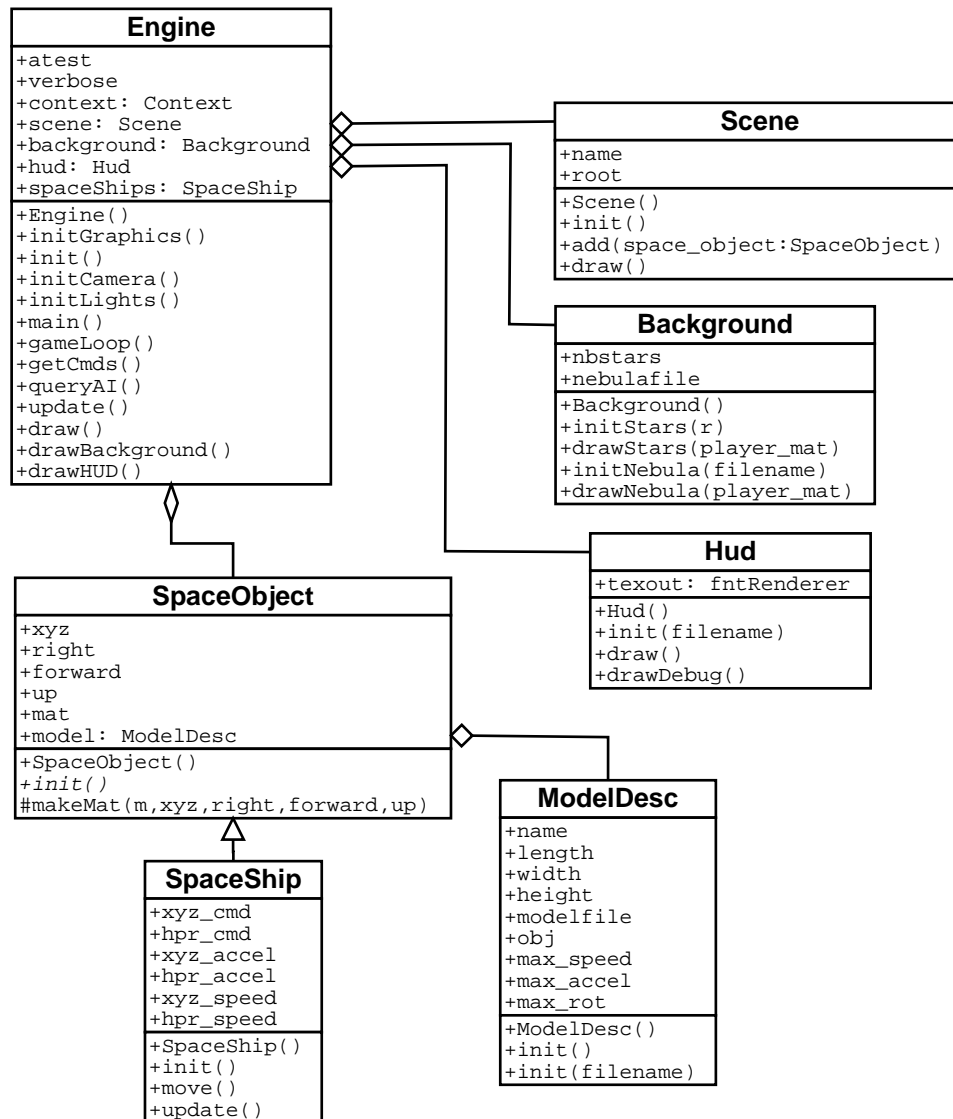


FIG. 8.5 – Diagramme de classes

### 8.4.1 Engine

Le *Core* de Stardust !

Au démarrage, s'initialise ainsi que tout les sous-systèmes (gfx, sound, joystick, net,..), charge les objectifs de mission et entre dans la game loop.

Dans la *game loop* :

- Calcul le temps écoulé depuis la dernière frame ;
- Update (get player commands + AI) ;
- Display (Scene).

### 8.4.2 Scene

L'univers du jeu, dans notre cas la "scene graphe" :

- Background ;
- La classe SpaceObject pour est appelé déplacer tout les objets ;
- Détection des collisions ;
- Affichage.

### 8.4.3 SpaceObject

Un objet dans l'espace. Contient les informations :

- Position ;
- Orientation ;
- SpaceModel.

#### ModelDesc

Description de l'objet :

- Name ;
- Size ;
- Max Speed/Accel/Rot ;
- 3DModel (mesh).

#### SpaceShip

Sous-classe spécialisé de SpaceObject :

- Accel,Speed ;
- Shield ;
- Energy ;
- Throttle...

## 8.5 Structure

Le projet *Stardust* devra satisfaire les conditions suivantes :

- Projet sous licences GPL, LGPL et GFDL <sup>1</sup>;
- Programmation en langage C (gcc sous Linux et utilisation des scripts autoconf/automake pour l'installation);
- Design UML avec Dia (+ Dia2Code).

### 8.5.1 Librairies

Mes dernières études sur les libs dispos pour le développement du jeu *Stardust...* m'ont conduites aux conclusions suivantes :

- Utilisation d'OpenGL (ou son équivalent Mesa3d, facilitant ainsi le portage sous d'autres plate-formes);
- Utilisation de la lib SDL\_Video pour créer un context OpenGL;
- Gestion des événements (clavier,souris, joystick...) avec la lib SDL (dispo pour Linux, Unix, BeOS, Win32, ..., PS2 et bientôt Dreamcast), on notera que GLUT n'est plus supporté (et la licence trop restrictive) et que son alternative freeglut comporte encore quelques bugs;
- Affichage du HUD avec OpenGL et la texfont de Mark J. Kilgard;
- Hiérarchisation des objets (scene graph) et gestion du LOD<sup>2</sup>;
- On notera aussi la possibilité d'importer des objets (Wavefront OBJ,..), ainsi qu'une connection avec Blender (plugins Python).

Encore à étudier :

- SDL\_Image, Intro?
- Gestion des données sur disque/cdrom : SDL?
- Son : SDL + SDL\_mixer (ogg, mod,..) et/ou OpenAL?
- AI Python?
- GUI : GLUI, Gtk,...?
- Réseau : SDL\_net.

---

<sup>1</sup><http://www.gnu.org/copyleft/copyleft.fr.html>

<sup>2</sup>Level Of Detail

### 8.5.2 Quelques références

- The Art & Science of Making Games : <http://www.gamasutra.com>
- GameDev.net : <http://www.gamedev.net>
- OpenGL ARB 1.1 : <http://www.opengl.org>
- Mesa3d 3.0 : <http://www.mesa3d.org>
- libSDL 1.2.5 : <http://www.libsdl.org>
- PLIB 1.6.0 : <http://plib.sourceforge.net>
- GLUT 3.7 : <http://www.opengl.org/Documentation/GLUT.html>
- freeglut 1.3 : <http://freeglut.sf.net>

### 8.5.3 Quelques exemples

- TuxFleet : <http://tuxfleet.sourceforge.net>
- FlightGear : <http://www.flightgear.org>
- Tux A Quest for herrings : <http://tuxaqfh.sf.net>
- Tux Kart : <http://tuxkart.sf.net>



## Chapitre 9

# Dynamique dans *Stardust*

Dans *Stardust*, nous avons voulu créer des mouvements qui reproduisent au mieux les sensations de vol spatial. Pour se faire nous allons appliquer les principes de la physique à notre vaisseau spatial.

### 9.1 Cinématique du point

Un solide est un système matériel indéformable.

Un point matériel est un solide suffisamment petit pour que 3 coordonnées seulement suffisent à déterminer sa position.

#### 9.1.1 Cadre spatio-temporel

Un référentiel est l'ensemble d'un repère et d'une horloge :  $(\mathcal{R})(\mathcal{O}xyzt)$ .

Rq1 : En mécanique classique (Newton) toutes les horloges de tous les référentiels indiquent le même temps : le temps  $y$  est une grandeur absolue  $\Rightarrow (\mathcal{R})(\mathcal{O}xyz)$ .

Rq2 : En mécanique relativiste (Einstein) le temps  $y$  est une grandeur relative.

La position de  $M$  dans  $(\mathcal{R})(\mathcal{O}xyzt)$  est donnée par le vecteur  $\overrightarrow{OM}$ .

La vitesse de  $M$  dans  $(\mathcal{R})(\mathcal{O}xyzt)$  est donnée par le vecteur :

$$\overrightarrow{v_{M/\mathcal{R}}} = \left( \frac{d\overrightarrow{OM}}{dt} \right)_R$$

L'accélération de  $M$  dans  $(\mathcal{R})(\mathcal{O}xyzt)$  est donnée par le vecteur :

$$\overrightarrow{a_{M/\mathcal{R}}} = \left( \frac{d\overrightarrow{v_{M/\mathcal{R}}}}{dt} \right)_R$$

### 9.1.2 Coordonnées cartésiennes

$$\begin{cases} \overrightarrow{OM} = x(t)\vec{e}_x + y(t)\vec{e}_y + z(t)\vec{e}_z \\ \overrightarrow{v_{M/R}} = \dot{x}(t)\vec{e}_x + \dot{y}(t)\vec{e}_y + \dot{z}(t)\vec{e}_z \\ \overrightarrow{a_{M/R}} = \ddot{x}(t)\vec{e}_x + \ddot{y}(t)\vec{e}_y + \ddot{z}(t)\vec{e}_z \end{cases}$$

## 9.2 Dynamique du point

### 9.2.1 Masse et quantité de mouvement

Il faut attribuer à un point matériel une grandeur scalaire caractérisant la répugnance qu'il a à se mettre en mouvement ; c'est la masse.

Dans un référentiel  $\mathcal{R}$  le vecteur quantité de mouvement de  $M$  p/r  $\mathcal{R}$  est :

$$\overrightarrow{\rho_{M/R}} = m\overrightarrow{v_{M/R}}$$

### 9.2.2 Lois fondamentales de la dynamique

**Loi 1** *Il existe au moins un référentiel privilégié  $\mathcal{R}$  par rapport auquel tout point matériel  $M$  ne subissant aucune interaction mécanique a un mouvement rectiligne uniforme.*

$$\begin{cases} \overrightarrow{v_{M/R}} = c\vec{t}\vec{e} \\ \overrightarrow{\rho_{M/R}} = c\vec{t}\vec{e} \end{cases}$$

Ces référentiels privilégiés sont appelés référentiels Galiléen :

**Réf. lié à la Terre** pourra être considéré comme Galiléen dans de nombreux cas,

**Réf. Géocentrique** (centre Terre) constituera un bon réf. Galiléen,

**Réf. de Copernic** (centre Soleil, 3 étoiles fixes) est un excellent réf. Galiléen.

Forces : Toute interaction mécanique peut être décrite par un vecteur  $\vec{f}$  (force) qui s'exerce en son point d'application ( $M$ ).

Ex interaction gravitationnelle :

$$\begin{cases} \vec{f}_{1/2} = -\mathcal{G}\frac{m_1m_2}{r^2}\vec{u}_{1/2} \\ \mathcal{G} = 6.67.10^{-11} USI \end{cases}$$

**Loi 2 (PFD)** *Dans un réf. Galiléen un point matériel  $M$  soumis à des interactions dont la résultante est  $\sum \vec{f}$  a un mouvement satisfaisant :*

$$\left(\frac{d\overrightarrow{\rho_{M/R}}}{dt}\right)_R = \sum \vec{f}, \quad \overrightarrow{\rho_{M/R}} = m\overrightarrow{v_{M/R}}$$

pour le point matériel :  $m\overrightarrow{a_{M/K}} = \Sigma \vec{f}$

## 9.3 Équations Différentielles Ordinaires

### 9.3.1 EDO avec conditions initiales

On considère le problème dit de "Cauchy", à savoir une équation différentielle du premier ordre :

$$\frac{dy}{dt} = y'(t) = f(t, y(t)) \text{ en connaissant } t_0 \text{ et } y_0 = f(t_0).$$

Toutes les techniques numériques consistent à discrétiser la variable  $x_{i+1} = x_i + h_i$  et à estimer la valeur de  $y_{i+1}$  en fonction d'une ou plusieurs valeurs  $y_i$  déjà connues ou estimées.

La méthode la plus simple, dite d'Euler, approxime  $y'_{i+1} = \frac{y_{i+1} - y_i}{h_i}$  et permet de tirer  $y_{i+1} = y_i + h_i f(x_i, y_i)$ .

De qualité médiocre, elle conduit dans beaucoup de cas à une erreur importante au bout d'un nombre assez faible de pas  $h_i$ , les méthodes de Runge-Kutta ou de prédiction-correction sont plutôt le standard de résolution efficace d'EDO : l'approximation de  $y_{i+1}$  se fait à partir de plusieurs calculs de  $f(t, y)$  en des points  $x$  convenablement choisis.

### 9.3.2 Méthode d'Euler

On définit la pas  $h$  par la subdivision d'un interval de temps  $[t_0, t_1]$  en  $N$  sous-intervalles :  $h = \frac{t_1 - t_0}{N}$ .

Le développement en série de Taylor à l'ordre 2 donne :

$$\begin{cases} y(t+h) &= y(t) + hy'(t) + o(h^2) \\ &= y(t) + hf(t, y(t)) \end{cases}$$

En itérant, on obtient la solution  $y(t_i)$  aux points ( $t_i = t_0 + hxi$ ) :

$$\begin{cases} y(t_0) &= y_0 \\ y(t_{i+1}) &= y(t_i) + hf(t_i, y(t_i)) \end{cases}$$

### 9.3.3 Méthode d'Euler Cauchy corrigé

De manière équivalente on à :

$$y(t+h) = y(t) + \frac{h}{2} (f(t, y(t)) + f(t+h, y(t)) + hf(t, y(t)))$$

### 9.3.4 Méthode de Taylor d'ordre plus élevé

Ces méthodes sont simplement basées sur un développement en série de Taylor d'un ordre plus élevé que pour la méthode d'Euler :

$$y(t+h) = y(t) + hy'(t) + \frac{h^2}{2!}y''(t) + \dots + \frac{h^n}{n!}y^{(n)}(t) + o(h^{n+1})$$

### 9.3.5 Méthodes de Runge-Kutta

La méthode de Runge-Kutta d'ordre 2 calcul la valeur de la fonction en 2 points intermédiaires :

$$\begin{cases} k_1 = hf(t, y(t)) \\ k_2 = hf(t+h, y(t) + k_1) \\ y(t+h) = y(t) + \frac{k_1+k_2}{2} \end{cases}$$

Méthode de Runge-Kutta d'ordre 4 :

$$\begin{cases} k_1 = hf(t, y(t)) \\ k_2 = hf(t + \frac{h}{2}, y(t) + \frac{k_1}{2}) \\ k_3 = hf(t + \frac{h}{2}, y(t) + \frac{k_2}{2}) \\ k_4 = hf(t+h, y(t) + k_3) \\ y(t+h) = y(t) + \frac{k_1+2k_2+2k_3+k_4}{6} \end{cases}$$

### 9.3.6 Autres méthodes

Ces méthodes sont basés sur l'interpolation polynomiale d'ordre 1,2,3 ou 4. Ces méthodes sont aussi appelées méthodes à pas multiples.

#### Adams-Bashforth à 2 pas

$$y(t+h) = y(t) + h \frac{3f(t, y(t)) - f(t-h, y(t-h))}{2}$$

#### Adams-Bashforth à 3 pas

$$y(t+h) = y(t) + h \frac{23f(t, y(t)) - 16f(t-h, y(t-h)) + 5f(t-2h, y(t-2h))}{12}$$

## 9.4 Programmation du moteur cinématique

### 9.4.1 Position du problème

Nous considérerons un vaisseau spatial, assimilé à un point matériel  $M(\equiv G)$ , dans un référentiel Galiléen  $\mathcal{R}(Oxyz)$ , possédant une accélération  $\vec{a}$ , une vitesse  $\vec{v}$  et une position dans l'espace  $\overrightarrow{OM}$  qui sera noté  $\vec{x}$ .

Dans ces conditions la cinématique et le PFD donne :

$$\sum \vec{f} = m \overrightarrow{a_{M/R}}$$

La résultante des forces appliqués sur le vaisseau (principalement les propulseurs en absence de frottements) sera noté :  $\vec{F} = \sum \vec{f}$  et sera considérée constante. On en déduit alors :

$$\vec{a} = \frac{\vec{F}}{m}$$

Cela permet de dire que sans force, il n'y a pas d'accélération et vis-versa.

On sait de plus qu'une accélération indique un changement de vitesse et que la présence d'une vitesse se traduit par un déplacement dans l'espace.

$$\begin{aligned} \frac{d\vec{v}(t)}{dt} &= \vec{a}(t) = \vec{f}(t, v(t)) = c\vec{t}\vec{e} \\ \frac{d\vec{x}(t)}{dt} &= \vec{v}(t) = \vec{f}(t, x(t)) \\ &\text{avec } h = \frac{dt}{N} = dt \end{aligned}$$

En intégrant successivement ces relations sur un interval de temps  $dt$  on en déduira alors la position  $\vec{x} = \overrightarrow{OM}$ .

L'accélération étant constante l'intégration de la première relation s'effectue de manière évidente et donne :

$$\vec{v}(t + dt) = \vec{v}(t) + a dt$$

Le problème se réduit alors à intégrer la seconde relation :

### 9.4.2 Méthode d'Euler

$$\left\{ \begin{aligned} \vec{x}(t + dt) &= \vec{x}(t) + \vec{v}(t)dt \quad (\text{cf.9.3.2}) \end{aligned} \right.$$

### 9.4.3 Méthode d'Euler Cauchy corrigé

$$\left\{ \begin{array}{l} \vec{x}(t + dt) = \vec{x}(t) + \vec{v}(t)dt + \frac{1}{2}\vec{a} dt^2 + \frac{1}{2}\vec{v}(t) dt^2 \end{array} \right. \text{ (cf.9.3.3)}$$

### 9.4.4 Méthodes de Runge-Kutta

#### Runge-Kutta d'ordre 2

$$\left\{ \begin{array}{l} \vec{x}(t + dt) = \vec{x}(t) + \frac{k_1+k_2}{2} dt \\ \phantom{\vec{x}(t + dt)} = \vec{x}(t) + \vec{v}(t)dt + \frac{1}{2}a dt^2 \end{array} \right. \text{ (cf.9.3.5)}$$

#### Runge-Kutta d'ordre 4

$$\left\{ \begin{array}{l} \vec{x}(t + dt) = \vec{x}(t) + \frac{k_1+2k_2+2k_3+k_4}{6} dt \\ \phantom{\vec{x}(t + dt)} = \vec{x}(t) + \vec{v}(t)dt + \frac{1}{3}a dt^2 \end{array} \right. \text{ (cf.9.3.5)}$$

# Chapitre 10

## La flotte

### 10.1 Charte graphique

But = Assurer une cohérence graphique entre tous les vaisseaux spatiaux.

Après avoir déterminé le nom, la classe (ex : Soudot), la catégorie (ex : Corvette), le type (ou identifiant, ex : CS-525x) plusieurs points sont à considérer avant de commencer le design d'un nouveau vaisseau :

#### 10.1.1 Manoeuvrabilité

La manoeuvrabilité représente les capacités de manoeuvre du vaisseau. Les valeurs permises sont : Extrême, Élevée, Moyenne ou Faible.

#### 10.1.2 Zone d'activité

Aérien*	Atmosphère planétaire
Planétaire*	Proximité d'une planète
Lunaire	Une planète et ses lunes
Stellaire	Différentes planètes d'un système
Inter-stellaire	Différentes étoiles

TAB. 10.1 – Rayon d'action

\* Ces types de vaisseaux (pouvant se poser sur une planète) nécessitent une forme aérodynamique.

### 10.1.3 Description

Enfin pour affiner la description du vaisseau il faut aussi renseigner les données suivantes :

- Puissance (en Watt) ;
- Autonomie (heures, jours, mois, années) ;
- Équipage ;
- Longueur et tonnage à vide ;
- Volume de frêt.

Ainsi que toutes autres informations que vous jugerez utiles.

### 10.1.4 Volumes

La classification par type, séparant les vaisseaux en catégories de taille et de maniabilité, va nous permettre de nous donner une indication sur les volumes occupés par les différents systèmes du vaisseau :

Systèmes de propulsion et production d'énergie	30% du volume du vaisseau divisé par le type (cf.6.1.2)
Modules d'habitation et systèmes vitaux	10m <sup>3</sup> par personne +10 à 60% pour plus de confort
Système d'arme :	
Canon wingrove	10m <sup>3</sup>
Tourelle légère	5m <sup>3</sup>
Tourelle standard	15m <sup>3</sup>
Tourelle lourde	50m <sup>3</sup>

TAB. 10.2 – Espaces occupés

## 10.2 Fichiers de spécifications

L'ensemble des fichiers de configuration sont au format XML (utilisation de la libxml2<sup>1</sup> du projet GNOME<sup>2</sup>).

Des exemples de ces fichiers de spécifications sont fournis dans leur répertoires respectifs : data/...

### 10.2.1 Specs UNIVERSE

Ce fichier décrit une partie de l'univers connu :

---

<sup>1</sup><http://xmlsoft.org>

<sup>2</sup><http://www.gnome.org>



**<id> name**

Dans cette ligne réside le nom de l'espace décrit.

**<scenes> file.scn[.gz]**

Cette commande déclare une scène à inclure.

**<cnxs> from to**

Cette commande définit les points de connections (hyper-espace) entre les différentes scènes d'éclairés précédemment.

### 10.2.2 Specs MISSION

Ce fichier décrit la mission à effectuer par le joueur. Il comprend la description de la région de l'espace (universe) où se déroule la mission ainsi que les objectifs de mission :

**<brief> texte**

En addition aux spécifications standards de l'univers, on rajoute ici les objectifs de missions.

### 10.2.3 Specs SCENE

C'est dans ce fichier qu'est décrit un système stellaire, avec ses étoiles, planètes, bases et vaisseaux :

**<id> name nebula.tif**

Dans cette ligne réside l'identification de la section (scène) ainsi que le liens vers la texture représentant le ciel étoilé (nébuleuse).

**<ships> file.shp posx posy posz forward right up**

Cette commande déclare un vaisseau à une position donnée de la scène.

**<player> file.shp posx posy posz forward right up**

Et enfin la position initiale du joueur.

### 10.2.4 Specs SHIP

Au format XML, ce fichier défini l'ensemble des caractéristiques des vaisseaux (ainsi que la référence sur le model 3D au format OBJ) :

**<id> name class category model camp**

Dans cette section réside l'identification du vaisseau.

**<desc> manoeuvrability power range autonomy crew length tonnage freight**

Cette section est uniquement descriptive, les valeurs des champs sont libres. Exception cependant concernant la maniabilité et le rayon d'action.

La manoeuvrabilité (mania) peut prendre les valeurs : low, med, high ou extreme.

Le rayon d'action (range) permet quant à lui (outre l'autonomie) de définir si les voyages inter-planétaires ou dans l'hyper-espace sont possible. Les valeurs acceptés sont : aerial, planetary, lunar, stellar, interstellar.

**<caract> type hit\_points**

On spécifie ici les caractéristiques entrant en considération pour le jeu. C'est-à-dire le type du vaisseaux (X,1-10+) et ses points (bonus).

**<obj> file.obj[.gz]**

On indique ici le fichier dans lequel est décrit la modélisation 3D du vaisseau (nom du fichier OBJ éventuellement gzippé).

**<ai> file.py**

Référence vers le fichier d'intelligence artificielle (script Python).

**<accel> forward back lat up down yaw pitch roll slum**

Cette commande détail les caractéristiques du moteur : accélérations linéaires, angulaires et sub-luminique (m/s/s, deg/s/s).

**<velocity> forward back lat up down yaw pitch roll slum**

Cette commande détail les caractéristiques du vaisseau : vitesses maximales linéaires, angulaires et sub-luminique (m/s, deg/s).

**<off> gun light-turret std-turret heavy-turret torpedo missile cme**

On indique ici le nombre disponible pour chacun de ces types d'armement.

### <def> hull armour

On indique ici la résistance de la coque (hull) et la puissance de protection du champ d'énergie (armour, duralium).

### <detect> range accuracy

Portée en mètres et sensibilité (pourcentage) des détecteurs .

Par exemple dans le cas du Soudot ce fichier ressemblerai à :

```
<?xml version="1.0" ?>
<Stardust>
  <ship>
    <id>
      <name>Default</name>
      <class>Soudot</class>
      <category>Corvette</category>
      <model>CS-525x</model>
      <camp>NVY</camp>
    </id>
    <desc>
      <manoeuvrability>high</manoeuvrability>
      <power>1,5.10^9W</power>
      <range>interstellar</range>
      <autonomy>3months</autonomy>
      <crew>40+4</crew>
      <length>151m</length>
      <tonnage>9500t</tonnage>
      <freight>100m^3</freight>
    </desc>
    <caract>
      <type>3</type>
      <hit_points>1000</hit_points>
    </caract>
    <obj>
      <file>data/obj/corvette.obj.gz</file>
    </obj>
    <ai>
      <file>data/ai/default.py</file>
    </ai>
    <accel>
      <forward>7000.0</forward>
      <back>5000.0</back>
      <lat>3000.0</lat>
```

```

    <up>4000.0</up>
    <down>2000.0</down>
    <yaw>120.0</yaw>
    <pitch>180.0</pitch>
    <roll>240.0</roll>
    <slum>100000</slum>
</accel>
<velocity>
    <forward>70000.0</forward>
    <back>50000.0</back>
    <lat>30000.0</lat>
    <up>40000.0</up>
    <down>20000.0</down>
    <yaw>180.0</yaw>
    <pitch>270.0</pitch>
    <roll>360.0</roll>
    <slum>100000000</slum>
</velocity>
<off>
    <gun>1</gun>
    <light_turret>0</light_turret>
    <std_turret>2</std_turret>
    <heavy_turret>0</heavy_turret>
    <torpedo>16</torpedo>
    <missile>2</missile>
    <cme>8</cme>
</off>
<def>
    <hull>1500</hull>
    <armour>500</armour>
</def>
<detect>
    <range>1000000000</range>
    <accuracy>60</accuracy>
</detect>
</ship>
</Stardust>

```

### 10.2.5 Specs OBJ

Stardust ne supporte qu'une partie du format de fichier OBJ d'Alias Wavefront. Je vous recommande de lire ce tutoriel, puis d'ouvrir des fichiers OBJ afin de vérifier qu'ils soient compatibles avec Stardust.

Le format OBJ est un format de fichier texte, ce qui signifie que vous pouvez l'éditer avec un simple éditeur de texte. Malheureusement, les spécifications originales ne définissent pas comment les lignes se terminent (retour chariot, caractère de fine de ligne, ...).

Le premier caractère de chaque ligne détermine le type de la commande. Si le premier caractère est un #, la ligne est considérée comme un commentaire, de même toute ligne blanche est ignorée.

Le fichier est lu et traité ligne par ligne du début à la fin. Dans la description qui suit, le premier caractère est la commande, suivie par ses arguments. Les arguments entre crochets sont optionnels.

### **# a comment line**

Elle est toujours ignorée. Habituellement la première ligne de tout fichier OBJ est un commentaire indiquant quel outil a généré le fichier, de même il est de coutume de rajouter le nom, le nombre de faces, ...

### **v x y z**

Cette commande spécifie un vertex avec ses trois coordonnées (un point dans l'espace). Les vertexes sont implicitement nommés par leur numéro d'apparition dans le fichier (on commence à 1).

### **vt u v [w]**

Cette commande spécifie les coordonnées de mapping UV (et éventuellement W). Ce sont des réels compris entre 0 et 1.

### **vn x y z**

On définit ici les normales. Souvent non utilisés car recalculés lors de la construction des faces.

### **f v1[/vt1][/vn1] v2[/vt2][/vn2] v3[/vt3][/vn3] ...**

C'est ici que l'on décrit les faces (polygons). Les arguments sont les indices des vertex, des coordonnées de mapping et des normales décrit par les commandes précédentes.

Par convention les faces sont définies dans le sens trigonométrique.

### **g name**

On définit ici un sous-objet; Toutes les faces suivantes sont considérées comme faisant partie de ce groupe.

### **mtllib mat.mtl**

Lien vers le fichier contenant les spécifications des matériaux.

### **usemtl name**

Cette commande permet de déclarer un matériau à utiliser ; Toutes les faces suivant la déclaration utiliseront ce matériau.

Notez que pour les vertexes, qui peuvent être disséminés dans le fichier, seul l'ordre dans lequel ils apparaissent compte. Les faces peuvent aussi être disséminés dans le fichier, à part qu'ils doivent suivre la déclaration des vertexes (je pense ;), et qu'ils font parti de tout groupe/matériau qu'ils suivent.

La forme "normale" d'un fichier OBJ est finalement la suivante :

```
# comment about what application generated this file
# material lib
mtllib mat.mtl
# all of the 'v' commands
v x y z
v ...
# all of the 'vt' commands
vt u v w
vt ...
# all of the 'vn' commands
vn x y z
vn ...
# the object name and its material
g object
usemtl mat
# all of the 'f' commands
f 1/1/1 2/2/2 3/3/3 4/4/4
f ...
```

### **10.2.6 Blender**

Pour faciliter la modélisation des objets 3D sous son modeleur favori, un plugin Python à été développé pour Blender permettant l'export des objets au format OBJ d'Alias Wavefront (Maya) :

Pour l'utiliser, sélectionner l'objet à exporter, charger le script *objexport2.py* dans la *Text window* et enfin exécuter le script (*Alt+P*).

Le résultat est un fichier portant le nom de l'objet avec extension *.obj*.

On notera la possibilité de raffiner successivement le maillage initial (méthodes de Catmull-Clark, de Butterfly, de Loop, ou autres), afin d'obtenir différents niveaux de détails.

Les différents raffinements devront être enregistrés sous la forme *objectnameX.obj*, où X est le niveau de raffinement (0=initial).

Ci-dessous quelques règles à respecter lors de la modélisation des objets :

- Axe z le long du vaisseau, dirigé vers l'arrière;
- Axe x dirigé sur la droite;
- Axe y pointant vers le haut;
- La taille de l'objet peut être indifféremment définie en "Edit Mode" ou "zoomé" par la suite;
- L'unité choisie pour la modélisation est le mètre;
- Enfin, le nombre de faces doit être assez restreint.

Une remarque sur les coordonnées de mapping : Ne pas oublier d'activer *TexFace Make* dans le panneau *Edit (F9)* ainsi que la génération des coordonnées UV (au niveau des faces) dans le panneau *Material (F5)*, bouton *TexFace*. Pour définir le mode de projection utiliser ensuite l'UV-Editor (*FKEY*, *UKEY*).

[A suivre...]

## **Troisième partie**

### **Annexes**



## Chapitre 11

# The GNU General Public License

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.

59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must

make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps : (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law : that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License ; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty ; keep intact all the notices that refer to this License and to the absence of any warranty ; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions :
  - (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
  - (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
  - (c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception : if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you ; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of

Sections 1 and 2 above provided that you also do one of the following :

- (a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange ; or,
- (b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange ; or,
- (c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

- 4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
- 5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or

distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances. It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

### END OF TERMS AND CONDITIONS

## Appendix : How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

one line to give the program’s name and a brief idea of what it does.

Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail. If the program is interactive, make it output a short notice like this when it starts in an interactive mode :

Gnomovision version 69, Copyright (C) yyyy name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w’.

This is free software, and you are welcome to redistribute it under certain conditions; type ‘show c’ for details.

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names :

Yoyodyne, Inc., hereby disclaims all copyright interest in the program  
'Gnomovision' (which makes passes at compilers) written by James Hacker.

signature of Ty Coon, 1 April 1989  
Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.



## Chapitre 12

# GNU Free Documentation License

Version 1.2, November 2002

Copyright ©2000,2001,2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

### Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom : to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation : a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals ; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **"Document"**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as **"you"**. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A **"Modified Version"** of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A **"Secondary Section"** is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The **"Invariant Sections"** are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The **"Cover Texts"** are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **"Transparent"** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called **"Opaque"**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML,

PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The **"Title Page"** means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section **"Entitled XYZ"** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **"Acknowledgements"**, **"Dedications"**, **"Endorsements"**, or **"History"**.) To **"Preserve the Title"** of such a section when you modify the Document means that it remains a section **"Entitled XYZ"** according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties : any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's

license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts : Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version :

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for

example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License

into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## **7. AGGREGATION WITH INDEPENDENT WORKS**

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## **8. TRANSLATION**

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## **9. TERMINATION**

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

### **ADDENDUM : How to use this License for your documents**

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page :

Copyright ©YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this :

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.



# Table des figures

1	Independence-War . . . . .	9
3.1	Stardust . . . . .	15
6.1	Bulle d'espace-temps . . . . .	38
6.2	Trou de ver . . . . .	40
8.1	Cas d'utilisation . . . . .	53
8.2	Diagramme d'état . . . . .	57
8.3	Game loop (local) . . . . .	58
8.4	Game loop (client/server) . . . . .	59
8.5	Diagramme de classes . . . . .	60

# Liste des tableaux

3.1	Commandes du jeu . . . . .	16
3.2	Changement de vue . . . . .	16
3.3	Commandes de pilotage . . . . .	17
3.4	Pilotage assisté . . . . .	17
3.5	Propulseurs latéraux . . . . .	18
3.6	Mode Sub-Luminique . . . . .	18
3.7	Hyper-espace . . . . .	18
3.8	Pilotage automatique . . . . .	19
3.9	HUD . . . . .	19
3.10	Contacts . . . . .	20
3.11	Désigner le contact . . . . .	20
3.12	Attaque . . . . .	21
3.13	Défense . . . . .	21
3.14	Répartition des ressources . . . . .	22
3.15	Envoyer des ordres . . . . .	24
3.16	t'Chat . . . . .	24
4.1	Dotations . . . . .	28
10.1	Rayon d'action . . . . .	70
10.2	Espaces occupés . . . . .	71

# Bibliographie

- [1] Particle Systems Ltd. *Independence War*. Infogrames Multimedia.
- [2] Steven Eisler. *Images de la Science Fiction*. Gründ, Paris, 1980.
- [3] Michel Brassive, Didier Guiserix. *MegaIII*. Casus Belli HS5, 1992.
- [4] Lawrence Ford, Thomas Roman. *L'énergie négative*. Pour la science, Mai 2000.
- [5] AM. Hugues, S. Moisan. *Analysis and Design with the UML*. ESSI, Fev 2003.