

Le manuel FlightGear

Michael Basler, Martin Spott,
Stuart Buchanan, Jon Berndt,
Bernhard Buckel, Cameron Moore,
Curt Olson, Dave Perry,
Michael Selig, Darrell Walisser,
et d'autres...



Le manuel FlightGear
2 août 2012
Pour *FlightGear* version 2.8.0

Table des matières

1	Préface	5
1.1	Lecture condensée	6
1.2	Instructions pour les grands impatientes	6
1.3	Lecture complémentaire	7
I	Installation	9
2	Vous voulez voler librement ? Choisissez <i>FlightGear</i> !	11
2.1	Encore un autre simulateur de vol ?	11
2.2	Choisir une version	16
2.3	Modèles de dynamique de vol (Flight Dynamics Models, FDM)	16
2.4	A propos de ce guide	17
3	Prévol : installer <i>FlightGear</i>	19
3.1	Installer des scènes	19
3.1.1	MS Windows Vista/7	20
3.1.2	Mac OS X	20
3.1.3	FG_SCENERY	21
3.1.4	Téléchargez automatiquement des scènes en plein vol	21
3.1.5	Créer vos propres scènes	22
3.2	Installer des aéronefs	23
3.3	Installer la documentation	23
II	Flying with <i>FlightGear</i>	25
4	Décollage : comment démarrer le programme	27
4.1	Variables d'environnement	27
4.1.1	FG_ROOT	27
4.1.2	FG_SCENERY	27
4.1.3	Variables d'environnement sous Windows et Mac OS X	27
4.2	Démarrer le simulateur sous Unix/Linux	28
4.3	Démarrer le simulateur sous Windows	29

4.3.1	Lancement à partir de la ligne de commande	30
4.4	Démarrer le simulateur sous Mac OS X	30
4.4.1	Choisir un aéronef et un aéroport	31
4.4.2	Activer le téléchargement des scènes à la volée	31
4.4.3	Activer la carte de navigation (Atlas)	31
4.4.4	Démarrer FlightGear - le simulateur	32
4.4.5	Fonctionnalités avancées	32
4.4.6	Démarrage à partir de la ligne de commande	36
4.5	Command line parameters	37
4.5.1	General Options	37
4.5.2	Features	38
4.5.3	Sound	38
4.5.4	Aircraft	38
4.5.5	Flight model	39
4.5.6	Initial Position and Orientation	39
4.5.7	Environment Options	41
4.5.8	Rendering Options	41
4.5.9	HUD Options	43
4.5.10	Aircraft System Options	43
4.5.11	Time Options	43
4.5.12	Network Options	44
4.5.13	Route/Waypoint Options	44
4.5.14	IO Options	44
4.5.15	Debugging options	45
4.6	Joystick support	46
4.6.1	Built-in joystick support	46
4.6.2	Joystick support via .fgfsrc entries	53
5	En vol : tout sur les instruments, les raccourcis clavier et les menus	55
5.1	Démarrer le moteur	55
5.1.1	Moteur à pistons	55
5.1.2	Aéronefs à turbopropulseurs	56
5.1.3	Aéronefs à réaction	56
5.2	Contrôles clavier	56
5.2.1	Contrôles de l'aéronef	56
5.2.2	Contrôles du simulateur	58
5.2.3	Contrôles du pilote automatique	59
5.3	Actions contrôlées par la souris	60
5.3.1	Mode normal	60
5.3.2	Mode contrôle	61
5.3.3	Mode vue	61
5.4	Entrées du menu	62
5.5	Le tableau de bord	66
5.6	Le collimateur tête haute (<i>Head Up Display, HUD</i>)	67

6	Features	69
6.1	Multiplayer	69
6.1.1	Quick Start	69
6.1.2	Other Methods	69
6.1.3	Troubleshooting	70
6.2	Aircraft Carrier	71
6.2.1	Starting on the Carrier	72
6.2.2	Launching from the Catapult	72
6.2.3	Finding the Carrier - TACAN	72
6.2.4	Landing on the Carrier	73
6.3	Atlas	73
6.4	Multiple Displays	74
6.5	Multiple Computer	74
6.5.1	Setup	74
6.5.2	Basic Configuration	75
6.5.3	Advanced Configuration	75
6.6	Recording and Playback	75
6.7	Text to Speech with Festival	76
6.7.1	Installing the Festival system	76
6.7.2	Running FlightGear with Voice Support	76
6.7.3	Troubleshooting	77
6.7.4	Installing more voices	78
6.8	Air-Air Refuelling	78
6.8.1	Setup	79
6.8.2	Multiplayer Refueling	80
III	Tutorials	81
7	Tutorials	83
7.1	In-flight Tutorials	83
7.1.1	Cessna 172P tutorials	84
7.2	FlightGear Tutorials	84
7.3	Other Tutorials	84
8	A Basic Flight Simulator Tutorial	87
8.1	Foreword	87
8.2	Starting Up	88
8.2.1	MS Windows	88
8.2.2	Linux and other unices	89
8.2.3	In the dark ?	89
8.3	The First Challenge - Flying Straight	90
8.4	Basic Turns	97
8.5	Taxiing on the ground	98

8.5.1	Airspeed	99
8.6	Advanced Turns	101
8.7	A Bit of Wieheisterology	102
8.7.1	Engine control	103
8.7.2	Wings and speed	106
8.7.3	The flaps	108
8.7.4	The stall	109
8.7.5	The trim	110
8.7.6	What direction am I flying ?	111
8.7.7	A look around the panel	112
8.8	Let's Fly	114
8.8.1	A realistic take off	114
8.8.2	Landing	115
8.8.3	Engine Shutdown	117
8.8.4	Aborted Landing	118
8.9	Dealing with the Wind	118
8.9.1	Crosswind Take Off	120
8.9.2	Crosswind Landing	120
8.9.3	Taxiing in the Wind	121
8.10	The autopilot	122
8.11	What Next ?	123
8.12	Thanks	123
8.13	Flying Other Aircraft	124
8.13.1	How to land the Cherokee Warrior II	124
8.13.2	How to take off and land the Piper J3 Cub	125
8.13.3	How to take off and land a jet	126
8.13.4	How to take off and land the P-51D Mustang	130
8.13.5	How to take off and land the B-52 Stratofortress	131
9	A Cross Country Flight Tutorial	133
9.1	Introduction	133
9.1.1	Disclaimer and Thanks	133
9.2	Flight Planning	134
9.3	Getting Up	136
9.3.1	Pre-Flight	136
9.3.2	ATIS	136
9.3.3	Radios	137
9.3.4	Altimeter and Compass	139
9.3.5	Take-Off	140
9.4	Cruising	140
9.4.1	The Autopilot	140
9.4.2	Navigation	141
9.4.3	Mixture	142
9.5	Getting Down	143

9.5.1	Air Traffic Control	143
9.5.2	The Traffic Pattern	144
9.5.3	Approach	145
9.5.4	VASI	147
9.5.5	Go Around	147
9.5.6	Clearing the Runway	148
10	An IFR Cross Country Flight Tutorial	149
10.1	Introduction	149
10.1.1	Disclaimer	150
10.2	Before Takeoff	150
10.2.1	Flight Planning	151
10.2.2	VHF Omnidirectional Range	151
10.2.3	Takeoff	155
10.3	In the Air	155
10.3.1	George I	156
10.3.2	MISON Impossible	156
10.3.3	George II	157
10.3.4	Staying the Course	159
10.3.5	Yet More Cross-Checks	160
10.4	Getting Down	161
10.4.1	Instrument Approach Procedures	161
10.4.2	Nondirectional Beacons	162
10.4.3	Procedure Turn	165
10.4.4	Chasing the Needle	167
10.4.5	FOOTO Time	168
10.4.6	George III	168
10.4.7	ILS Landings	169
10.4.8	Intercepting the Localizer	170
10.4.9	Intercepting the Glide Slope	170
10.4.10	Touchdown, Almost	171
10.4.11	A Confession	171
10.4.12	Touchdown, Not	172
10.4.13	Touchdown	173
10.5	Epilogue	173
11	A Helicopter Tutorial	177
11.1	Preface	177
11.2	Getting started	178
11.3	Lift-Off	179
11.4	In the air	179
11.5	Back to Earth I	180
11.6	Back to Earth II	180

IV	Appendices	183
A	Approche manquée : si rien ne fonctionne	185
A.1	Signaler des problèmes relatifs à FlightGear	185
A.2	Problèmes généraux	186
A.3	Problèmes potentiels sous Linux	187
A.4	Problèmes potentiels sous Windows	188
B	Atterrissage : quelques réflexions complémentaires avant de quitter l'avion	189
B.1	Une ébauche de l'histoire de <i>FlightGear</i>	189
B.1.1	Scènes	190
B.1.2	Aircraft	191
B.1.3	Environment	192
B.1.4	User Interface	192
B.2	Those, who did the work	194
B.3	What remains to be done	202

Chapitre 1

Préface

FlightGear est un simulateur de vol libre et gratuit développé grâce à Internet par une communauté de passionnés de simulation de vol et de programmation. "Le manuel *FlightGear*" a pour but d'offrir aux débutants un guide à l'installation et à l'utilisation de *FlightGear* et aux premières heures de vol. Il n'a pas pour vocation de fournir une documentation complète de toutes les fonctionnalités et ajouts de *FlightGear* mais, plutôt, d'apporter au nouvel utilisateur les meilleures bases pour lui permettre d'explorer ce que *FlightGear* a à offrir.

Cette version du manuel a été écrite pour la version 2.8.0 de *FlightGear*. Les utilisateurs de versions précédentes de *FlightGear* trouveront toujours une utilité à ce document, mais certaines des fonctionnalités qui y sont décrites peuvent être absentes de leur version.

Ce guide est scindé en trois parties et structuré de la façon suivante :

Partie I : Installation

Le chapitre 2, *Vous voulez voler librement ? Choisissez FlightGear !* présente *FlightGear*, la philosophie qui porte le projet et décrit les pré-requis système.

Dans le chapitre 3, *Prévol : installer FlightGear*, vous trouverez les instructions nécessaires à l'installation des binaires et des scènes et avions additionnels.

Partie II : Voler avec FlightGear

Le chapitre 4, *Décollage : comment démarrer le programme*, décrit comment lancer le programme une fois installé. Il comprend un survol des nombreuses options en ligne de commande ainsi que des fichiers de configuration.

Le chapitre 5, *En vol : tout sur les instruments, les raccourcis clavier et les menus*, décrit comment utiliser le programme, c'est-à-dire à proprement parler comment voler avec *FlightGear*. Ceci comprend une liste exhaustive (nous l'espérons) des raccourcis claviers prédéfinis, une vue d'ensemble des entrées des menus, des descriptions détaillées du tableau de bord et du collimateur tête haute (Head Up Display, HUD), ainsi que des astuces sur l'utilisation des fonctions de la souris.

Le chapitre 6, *Fonctionnalités* décrit certaines des fonctionnalités particulières que *FlightGear* offre à l'utilisateur avancé.

Partie III : Tutoriels

Le chapitre 7, *Tutoriels*, fournit des informations sur les nombreux tutoriels disponibles pour les apprentis pilotes.

Le chapitre 8, *Tutoriel de base du simulateur de vol*, propose un tutoriel sur les bases du vol, illustré avec de nombreux exemples sur la façon dont les choses apparaissent dans *FlightGear*.

Le chapitre 9, *Un tutoriel de vol à travers la campagne*, décrit un vol simple à travers la campagne dans la région de San Fransisco, qui peut être réalisé avec l'installation par défaut.

Le chapitre 10, *Un tutoriel de vol IFR à travers la campagne*, propose un vol similaire à travers la campagne comprenant l'utilisation des instruments afin de permettre de voler sans danger dans les nuages sous le régime de vol aux instruments (Instrument Flight Rules, IFR).

Annexes

Dans l'Annexe A, *Approche manquée : si rien ne fonctionne*, nous essayons de vous aider à surmonter certaines difficultés fréquemment rencontrées lors de l'utilisation de *FlightGear*.

Dans la dernière Annexe B, *Atterrissage : quelques réflexions complémentaires avant de quitter l'avion*, nous souhaiterions remercier ceux qui le méritent, dresser une vue d'ensemble du développement de *FlightGear* et souligner ce qu'il reste à accomplir.

1.1 Lecture condensée

Pour ceux qui ne veulent pas lire ce document de bout en bout, nous suggérons de lire les sections suivantes dans l'ordre afin de disposer des minimas pour s'envoler :

Installation :	3
Démarrer le simulateur :	4
Utiliser le simulateur :	5

1.2 Instructions pour les grands impatientes

Nous savons que la plupart des gens détestent lire les manuels. Si vous êtes certain que votre carte graphique prend en charge OpenGL (vérifiez dans votre documentation ; par exemple, la plupart des cartes graphiques NVIDIA le font) et que vous utilisez Windows, Mac OS-X ou Linux, vous pouvez probablement

au moins sauter la Partie I de ce manuel et utiliser des binaires pré-compilés. Ils peuvent être trouvés, accompagnés de leurs instructions d'installation, à l'adresse suivante :

<http://www.flightgear.org/Downloads/>.

Si vous utilisez Linux, vous devriez pouvoir trouver *FlightGear* dans les paquets de votre distribution.

Une fois que vous avez téléchargé et installé les binaires, reportez-vous au Chapitre 4 pour obtenir des détails sur le démarrage du simulateur.

1.3 Lecture complémentaire

Bien que ce guide d'introduction contienne toutes les informations nécessaires, nous vous recommandons vivement de jeter un œil dans d'autres documents, notamment en cas de difficultés :

- un petit **dépliant** à imprimer. Bien pratique en vol, il peut être trouvé dans le paquetage de base dans le répertoire `/FlightGear/Docs` et est également disponible à l'adresse :
<http://www.flightgear.org/Docs/FGShortRef.pdf>.
- des **documentations utilisateurs** complémentaires sur des fonctionnalités particulières sont disponibles dans le paquetage de base dans le répertoire `/FlightGear/Docs`.
- il existe un **wiki** *FlightGear* officiel disponible à l'adresse <http://wiki.flightgear.org>.

Première partie

Installation

Chapitre 2

Vous voulez voler librement ? Choisissez *FlightGear* !

2.1 Encore un autre simulateur de vol ?

N'avez-vous jamais voulu piloter un avion par vous-même, mais manqué d'argent ou de compétences pour le faire ? Etes-vous un véritable pilote désireux de progresser sans devoir décoller ? Voulez-vous tenter quelques manœuvres dangereuses sans risquer votre vie ? Ou voulez-vous simplement vous amuser avec un jeu plus sérieux sans aucune violence ? Si l'une de ces questions s'applique à vous, alors les simulateurs de vol sur PC sont ce qu'il vous faut.

Vous avez peut-être déjà acquis de l'expérience aux commandes de Microsoft © Flight Simulator ou d'un autre simulateur de vol sur PC disponible dans le commerce ? Leurs prix se situant habituellement aux alentours de 70 euros, en acheter un ne devrait pas être une grosse difficulté, en sachant que, malgré la chute des prix, la configuration matérielle exigée pour faire fonctionner un simulateur de vol sérieux sur PC revient à peu près à 1 000 euros.

Avec tant de simulateurs de vol disponibles sur le marché, pourquoi passer des milliers d'heures de conception et de programmation pour construire un simulateur de vol libre et gratuit ? Et bien, il y a de nombreuses raisons, mais en voici les principales :

- tous les simulateurs du commerce ont un inconvénient majeur : ils sont conçus par un petit groupe de développeurs définissant leurs propriétés en fonction de ce qui leur est important et en fournissant des interfaces limitées aux utilisateurs. Toute personne ayant déjà essayé de contacter le développeur d'un logiciel du commerce sait que se faire entendre dans cet environnement est un réel défi. *A contrario*, *FlightGear* est conçu par des utilisateurs et pour des utilisateurs, tout étant fourni *de base*.
- les simulateurs commerciaux sont généralement un compromis entre fonctionnalités et facilité d'utilisation. La majeure partie des éditeurs commerciaux veulent pouvoir toucher un large segment d'utilisateurs, comprenant

de véritables pilotes, des débutants et même des joueurs occasionnels. Au final, il y a donc toujours un compromis entre date de mise sur le marché et budget. Comme *FlightGear* est libre et ouvert, il n'a aucun besoin de ce genre de compromis. Nous n'avons aucun éditeur sur le dos, nous sommes tous des volontaires qui définissons nous-mêmes nos propres dates de lancement. Nous sommes également libres de prendre en compte des marchés qu'aucun autre éditeur commercial ne considérerait rentable, comme la communauté de la recherche scientifique.

- en raison de la fermeture de leur code source, les simulateurs du commerce doivent se passer de la contribution de développeurs imaginatifs et talentueux. Avec *FlightGear*, des développeurs de tous niveaux et débordants d'idées ont la possibilité d'apporter des améliorations énormes sur le projet. La contribution à un projet aussi grand et complexe que *FlightGear* est une grande récompense et offre aux développeurs la fierté de participer au futur d'un grand simulateur.
- Enfin, et-au delà de ces considérations, c'est juste du pur plaisir ! Je suppose que vous pourriez nous comparer aux vrais pilotes qui assemblent des avions en kit ou qui construisent eux-mêmes leurs avions. Bien sûr, nous pourrions aller acheter un avion déjà construit et paré à voler, mais il y a vraiment un sentiment si particulier quand on construit quelque chose de ses propres mains.

Les points mentionnés ci-dessus forment la base des motivations pour lesquelles nous avons créé *FlightGear*. Avec ces motivations à l'esprit, nous nous sommes lancés dans la création d'un simulateur de vol de haute qualité visant à être civil, multi-plateforme, ouvert, auto-maintenu, et pouvant être étendu par les utilisateurs eux-mêmes. Jetons un coup d'œil plus détaillé à chacune de ces caractéristiques :

- **Civil** : le projet vise essentiellement la simulation de vol civile. Il devrait permettre de simuler aussi bien l'aviation générale que civile. Notre but à long terme est de faire approuver *FlightGear* par la FAA comme plateforme d'entraînement au vol. Malheureusement pour les utilisateurs intéressés, ce n'est pas pour le moment un simulateur de combat ; cependant, ces fonctionnalités ne sont pas explicitement exclues. C'est juste que nous n'avons pas eu de développeur sérieusement intéressé par les systèmes nécessaires à la simulation de combat.
- **Multi-plateformes** : les développeurs essaient de maintenir le code aussi indépendant que possible de la plate-forme. Cette motivation est basée sur leur observation que les gens intéressés par la simulation de vol utilisent toute une variété de matériel informatique et de systèmes d'exploitation. Le code actuel prend en charge les Systèmes d'exploitation suivants :
 - Linux (toutes distributions et plate-formes),
 - Windows NT/2000/XP/Seven (plate-formes Intel/AMD),
 - Windows 95/98/ME,
 - BSD UNIX,

- Sun-OS,
 - Mac OS X
- A notre connaissance, il n'existe pas aujourd'hui d'autre simulateur de vol, commercial ou libre, prenant en charge une gamme aussi étendue de plateformes.
- **Ouvert** : le projet n'est pas restreint à un cadre statique ou élitiste de développeurs. Toute personne se sentant capable de contribuer est plus que bienvenue. Concernant les droits d'auteur, le code (y compris la documentation) sont protégés sous les termes de la licence GNU General Public License (GPL).

La licence GPL est souvent mal comprise. En termes simples, elle déclare que vous pouvez copier et distribuer librement le(s) programme(s) auxquels elle s'applique. Vous pouvez les modifier si vous le souhaitez et même demander de l'argent pour la distribution du logiciel concerné, qu'il s'agisse d'une version modifiée ou originale. Cependant, lors de la distribution du logiciel, vous devez en fournir le code source aux destinataires et il doit conserver le copyright original. En résumé :

"Vous pouvez faire ce que vous désirez du logiciel sauf le rendre non-libre".

Le texte intégral de la GPL peut être obtenu à partir du code source de *FlightGear* ou à l'adresse suivante : <http://www.gnu.org/copyleft/gpl.html>.

- **Auto-maintenu et pouvant être étendu par les utilisateurs** : à la différence de la plupart des simulateurs commerciaux, dans *FlightGear*, les formats des scènes et des aéronefs, les variables internes, les APIs et tout le reste sont accessibles par l'utilisateur et documentés depuis le début. Même sans aucune documentation explicite du développement (qui naturellement doit être écrite à un moment ou un autre), on peut toujours consulter le code source pour voir comment quelque chose fonctionne. Le but des développeurs est de construire un moteur de base sur lequel les concepteurs des scènes, des tableaux de bord, et peut-être les auteurs de scénario d'aventures, de procédures de contrôle du trafic aérien (Air Traffic Control, ATC), les artistes audio, et d'autres, peuvent se greffer. Nous espérons que le projet, y compris ses développeurs et ses utilisateurs, tireront avantage de la créativité et des idées des centaines de "pilotes virtuels" talentueux de par le monde.

Sans aucun doute, le succès du projet Linux, initié par Linus Torvalds, a inspiré certains des développeurs. Non seulement Linux a montré que le développement distribué de projets hautement sophistiqués à travers Internet est possible, mais il a également prouvé qu'un tel effort peut surpasser le niveau de qualité de produits commerciaux concurrents.

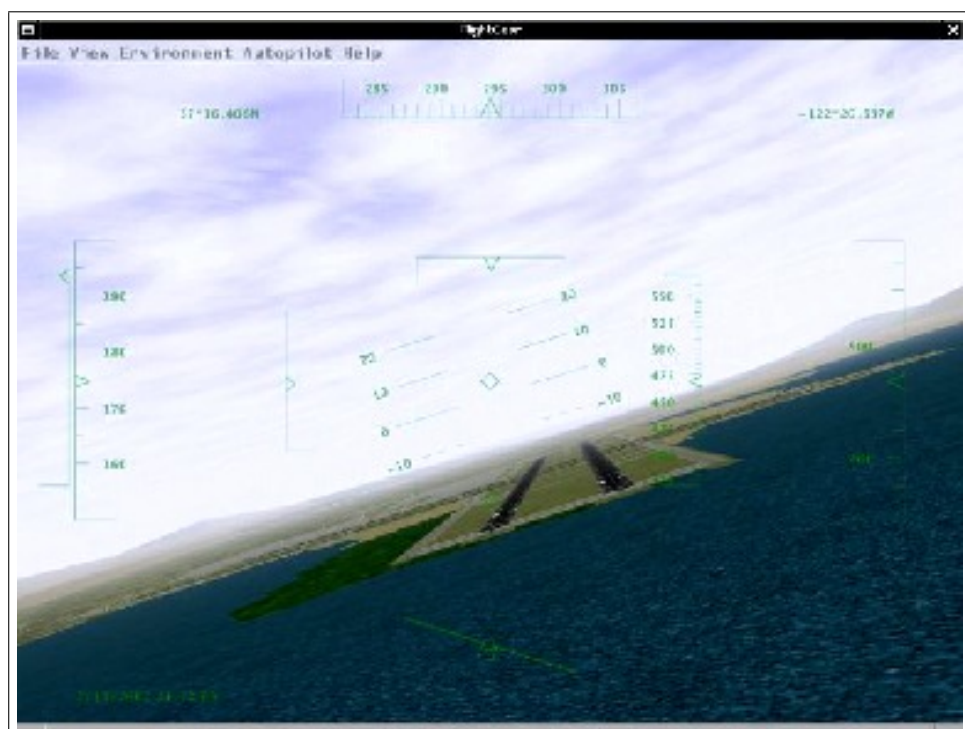


Image 1 : *FlightGear* sous UNIX : mauvaise approche de l'aéroport de San Francisco International, par l'un des auteurs de ce manuel...

En comparaison avec d'autres simulateurs de vol récents, les pré-requis système de *FlightGear* ne sont pas exagérés. Un processeur AMD Athlon64 de vitesse moyenne, ou un Intel P4, ou même un AMD Athlon/K7 décent ou un Intel PIII devraient être suffisants pour faire fonctionner *FlightGear* assez bien, pour peu que vous ayez une carte graphique 3D appropriée.

Un des pré-requis les plus importants pour faire fonctionner *FlightGear* est une carte graphique dont le pilote prenne en charge OpenGL. Si vous ne savez pas ce qu'est OpenGL, la vue d'ensemble proposée sur le site Internet d'OpenGL :

<http://www.opengl.org>

l'explique très bien : "Depuis son introduction en 1992, OpenGL est devenue l'interface de programmation d'application (Application Programming Interface, API) 2D et 3D la plus employée et la mieux prise en charge de l'industrie..."

FlightGear ne fonctionne pas (et ne fonctionnera jamais) sur une carte graphique ne prenant en charge que Direct3D/DirectX. Contrairement à OpenGL, Direct3D est une interface propriétaire, étant limitée au système d'exploitation Windows.

Vous devriez pouvoir faire fonctionner *FlightGear* sur un ordinateur doté d'une carte graphique 3D ne prenant pas en charge l'accélération matérielle OpenGL (et même sur des systèmes sans matériel graphique 3D du tout). Cependant, l'absence de prise en charge matérielle de l'accélération OpenGL peut mettre sur les genoux

même la plus rapide des machines. Les symptômes typiques de l'absence d'accélération matérielle sont des taux d'images (frame rate) inférieurs à 1 image par seconde.

Toute carte graphique 3D moderne prenant en charge OpenGL fera l'affaire. Pour les pilotes de carte graphique Windows qui prennent en charge OpenGL, visitez la page d'accueil de votre fabricant de carte vidéo. Notez que, parfois, les pilotes OpenGL sont fournis par les fabricants des processeurs graphiques et non par les fabricants des cartes. Si êtes sur le point d'acheter une carte graphique pour faire fonctionner *FlightGear*, une carte NVIDIA GeForce est recommandée, car elles ont tendance à bénéficier d'une meilleure prise en charge OpenGL que les cartes AMD/ATI Radeon. 256 Mo de mémoire graphique dédiée sera plus qu'approprié - de nombreuses personnes font fonctionner *FlightGear* sans souci avec moins.

Pour installer les exécutable et les scènes par défaut, vous aurez besoin d'environ 500 Mo d'espace disque libre. Au cas où vous voudriez/deviez compiler le programme vous-même, vous aurez besoin d'à peu près 500 Mo supplémentaires pour le code source et pour les fichiers temporaires créés durant la compilation. Cela ne comprend pas l'environnement de développement, qui variera en taille en fonction du système d'exploitation et de l'environnement utilisés. Les utilisateurs de Windows peuvent s'attendre à avoir besoin d'approximativement 300 Mo d'espace disque supplémentaire pour l'environnement de développement. Les machines fonctionnant sous Linux ou autres UNIX devraient avoir la plupart des outils de développement déjà installés, il est donc probable qu'il ne faille que peu d'espace disque supplémentaire sur ces plate-formes.

En ce qui concerne les effets sonores, n'importe quelle carte graphique devrait suffir. En raison de sa conception flexible, *FlightGear* prend en charge une large un large éventail de joysticks, de manches et de palonniers sous Linux et Windows. *FlightGear* peut également fournir des interfaces aux sièges de vol *full-motion*.

FlightGear est développé principalement sous Linux, un clone libre et gratuit d'UNIX (ainsi qu'un bon nombre d'outils GNU) développé coopérativement via Internet dans plus ou moins le même esprit que *FlightGear* lui-même. *FlightGear* fonctionne également et est en partie développé sous plusieurs versions de Windows. Il est également possible de compiler *FlightGear* sur Mac OS X ainsi que sur diverses stations de travail UNIX/X11. Dès l'instant où vous disposez d'un compilateur approprié installé, *FlightGear* peut-être compilé sous toutes ces plate-formes. Le principal compilateur sur toutes ces plateformes est le compilateur libre GNU C++ (le compilateur Cygnus Cygwin sous Win32).

Si vous voulez lancer *FlightGear* sous Mac OSX, vous aurez besoin de Mac OS X 10.4 ou supérieur. Les pré-requis minimums sont soit un Power PC G4 à 1,0 GHz ou un Mac Intel, mais nous vous suggérons un MacBook Pro, Intel iMac, Mac Pro ou Power Mac (Power PC G5) pour un confort de vol optimal.

2.2 Choisir une version

Nous vous recommandons d'utiliser la dernière version officielle, généralement produite annuellement et qui est utilisée pour créer les binaires pré-compilé. Elle est disponible à l'adresse :

<http://www.flightgear.org/Downloads/> Si vous voulez vraiment obtenir la toute dernière version du code (la plus évoluée et, parfois, la plus instable), vous pouvez cloner les sources à partir de l'adresse :

<http://www.gitorious.org/fg>.

pour obtenir le code le plus récent. Une description détaillée de la préparation de code en vue de l'exécution de *FlightGear* peut être trouvée à l'adresse :

<http://wiki.flightgear.org/GIT>.

2.3 Modèles de dynamique de vol (Flight Dynamics Models, FDM)

Historiquement, *FlightGear* était basé sur un modèle de vol qu'il a hérité de LaRCsim (ainsi que l'avion Navion). Comme cela imposait plusieurs limitations (la plus importante, beaucoup de caractéristiques étaient codées en dur au lieu de passer par des fichiers de configuration), quelques tentatives ont été faites pour développer ou inclure des modèles de vol alternatifs. En conséquence, *FlightGear* prend aujourd'hui en charge différents modèles de vol, qui doivent être choisis au lancement de l'application.

- Le plus important est probablement le modèle de vol JSB développé par Jon Berndt. Le modèle de vol JSB fait partie d'un projet autonome appelé *JSBSim* :
<http://jsbsim.sourceforge.net/>.
- Andrew Ross a créé un autre modèle de vol appelé *YASim* pour *Yet Another Simulator* (encore un autre simulateur). *YASim* prend une approche fondamentalement différente de nombreux autres FDM, car il se base sur l'information géométrique plutôt que sur les coefficients aérodynamiques. *YASim* dispose d'un FDM particulièrement avancé pour les hélicoptères.
- Christian Mayer a développé le modèle de vol d'un ballon à air chaud. Curt Olson intégra ensuite un mode de déplacement spécial, appelé "UFO", qui vous aide à voler rapidement d'un point A à un point B.
- Enfin, il existe le modèle de vol UIUC, développé par une équipe de l'Université d'Illinois à Urbana-Champaign. Ce travail était initialement tourné vers la modélisation d'un aéronef rencontrant des conditions de givrage, mais il inclut maintenant l'aérodynamique "non-linéaire", qui a pour conséquence d'améliorer le réalisme dans des attitudes extrêmes, comme le décrochage et les vols à angle d'attaque élevé. Deux bons exemples qui illustrent

cette capacité sont le deltaplane Airwave Xtreme 150 et le Wright Flyer de 1903. De plus amples détails du modèle de vol UIUC peuvent être obtenus à l'adresse :

<http://www.ae.illinois.edu/m-selig/apasim/Aircraft-uiuc.html>

Il est même possible de piloter l'affichage de la scène de *FlightGear* en utilisant un FDM externe fonctionnant sur un ordinateur différent ou par l'intermédiaire d'un canal nommé sur la machine locale (bien que cela ne soit pas recommandé aux utilisateurs découvrant *FlightGear*).

2.4 A propos de ce guide

Il y a peu, voire pas, de ressources dans ce guide qui y soient exclusivement présentées. Vous pourriez même dire comme Montaigne "Je n'ai fait qu'un bouquet de fleurs et n'ai rien fourni de moi-même que le lien qui les assemble." La majeure partie (mais heureusement pas l'intégralité) des informations présentes ici peut également être obtenue à partir du site Internet de *FlightGear* situé à l'adresse :

<http://www.flightgear.org/>

Le manuel FlightGear a pour vocation d'être un premier pas vers une documentation complète de *FlightGear*. Il s'adresse à l'utilisateur final qui n'est pas intéressé par le fonctionnement interne d'OpenGL ou par la création de ses propres scènes. Nous espérons qu'un jour il y aura, pour accompagner ce document, un *Guide du programmeur de FlightGear*, un *Guide de création des scènes de FlightGear*, décrivant les outils de scènes actuellement livrés avec *TerraGear* ; et un paquetage *Ecole de pilotage FlightGear*.

Merci de nous aider à améliorer ce document en nous soumettant vos corrections, améliorations, suggestions et traductions. Tous les utilisateurs sont invités à contribuer aux descriptions des configurations alternatives (cartes graphiques, systèmes d'exploitation, etc.). Nous serons plus qu'heureux de les inclure dans les futures versions du manuel *FlightGear* (bien entendu sans oublier le crédit attribuable à leurs auteurs).

Chapitre 3

Prévol : installer *FlightGear*

Pour faire fonctionner *FlightGear*, vous devez en installer les binaires. Une fois que vous aurez fait ceci vous pourrez, si vous le souhaitez, installer les paysages et avions additionnels.

Les binaires pré-compilés de la dernière version sont disponibles pour :

- Windows - toutes versions,
- Mac OS X,
- Linux.

Pour les télécharger, rendez-vous sur la page :

<http://www.flightgear.org/Downloads/binary.shtml>

et suivez les instructions qui y sont présentes.

3.1 Installer des scènes

Des scènes détaillées de *FlightGear* sont disponibles pour le monde entier, vous permettant de voler n'importe où, des sommets de l'Himalaya à la campagne du Kansas. Le paquetage de base de *FlightGear* comprend les scènes d'une petite zone autour de San Francisco, donc pour aller voler dans d'autres parties du monde vous aurez besoin de télécharger des scènes additionnelles.

Chaque portion de scène est regroupée dans une archive compressée (appelée *tarball*, en anglais), qui correspond à une tuile de 10 degrés par 10 degrés. Chaque archive est nommée en fonction de la tuile de 10x10 degrés qu'elle représente, par exemple w130n50.tgz.

Vous pouvez télécharger les scènes à partir d'une carte cliquable ici :

<http://www.flightgear.org/Downloads/scenery.html>

Sinon, vous pouvez soutenir le projet *FlightGear* en achetant un lot complet des scènes du monde entier sur DVD à l'adresse :

<http://www.flightgear.org/dvd/>

Une fois le fichier compressé téléchargé sur votre ordinateur, il vous faut localiser le répertoire *Scenery* de votre installation *FlightGear*.

- Pour Windows, ce répertoire devrait être :
`c:\Program Files\FlightGear\data\Scenery.`
- Pour les machines fonctionnant sous la famille Unix, il s’agit généralement du répertoire :
`/usr/local/share/FlightGear/data/Scenery.`
- Pour Mac OS X, il s’agit généralement de :
`/Applications/FlightGear.app/Contents/Resources/data/Scenery.`

Pour installer les scènes, décompressez le fichier compressé dans le répertoire *Scenery*. La plupart des systèmes d’exploitation propose des outils pour décompresser des fichiers type *tarball*. Si vous ne parvenez pas à le décompresser, installez un programme d’extraction comme 7-zip (<http://www.7-zip.org/>).

Notez que vous ne devez pas décompresser les fichiers de scènes numérotés présents au sein du fichier *tarball*, comme par exemple *958402.gz* - cette action sera réalisée à la volée par *FlightGear*.

Une fois que ce fichier est décompressé, les répertoires *Terrain* et *Objects* contiendront de nouveaux sous-répertoires où se situent vos nouvelles scènes.

Pour utiliser ces nouvelles scènes, rendez-vous simplement à l’aéroport de démarrage de votre choix situé au sein de la nouvelle scène. Si vous utilisez l’assistant de démarrage de *FlightGear*, vous devrez appuyer sur le bouton *Rafraîchir* avant de choisir votre aéroport.

3.1.1 MS Windows Vista/7

Si vous utilisez Windows Vista ou Windows 7, vous pourriez être confronté au fait que Windows installe les scènes (et aéronefs) téléchargés dans votre Virtual Store :

```
c:\Users\(Votre Nom)\AppData\Local\VirtualStore\Program Files\FlightGear\Scenery
```

S’il le fait, vous devrez copier manuellement les répertoires *Terrain* et *Objects* vers votre véritable répertoire *Scenery* de *FlightGear*, comme cela a été décrit ci-dessus.

3.1.2 Mac OS X

Vous pouvez installer les données des scènes et des avions téléchargés en utilisant l’interface de lancement (GUI launcher). Appuyez sur *Installez des données additionnelles* dans l’onglet *Fonctionnalités avancées* » *Autres*, ceci ouvrira la fenêtre d’exploration des fichiers. En choisissant un ou plusieurs fichiers de scènes, ceci installera les données de scènes dans `/Applications/FlightGear.app/Contents/Resources/data/Scenery`. Les formats acceptables pour les données de scènes sont zip, tar.gz, tgz, tar, et dossier extrait. Si l’installation via le GUI launcher ne fonctionne pas, pour quelque

raison que ce soit, vous avez toujours une possibilité alternative d'installer les données. Ouvrez le fichier de données en appuyant sur "Ouvrir le fichier de données" dans l'onglet Autres. Vous ouvrirez ainsi une fenêtre Rechercher pour le répertoire des données. Glisser le répertoire d'un avion vers le répertoire data/Aircraft (ou un répertoire de scènes vers un répertoire data/Scenery) sous le répertoire données permettra d'aboutir au même résultat.

3.1.3 FG_SCENERY

Si vous préférez conserver vos scènes téléchargées séparées de l'installation de base, vous pouvez le faire en précisant votre variable d'environnement FG_SCENERY.

Il s'agit de l'emplacement auquel *FlightGear* recherche des fichiers de scènes. Il contient une liste de répertoires qui seront analysés dans l'ordre. Les répertoires sont séparés par des ":" sous Unix (dont Mac OS X) et par des ";" sous Windows.

Par exemple, sous Linux, une variable d'environnement FG_SCENERY paramétrée à :

```
/home/jsmith/WorldScenery:/usr/local/share/Flightgear/data/Scenery  
précisera à FlightGear de rechercher des scènes en priorité dans le répertoire  
/home/jsmith/WorldScenery, suivi de :  
/usr/local/share/Flightgear/data/Scenery.
```

Sous Windows, une variable d'environnement FG_SCENERY paramétrée à :

```
c:\Program Files\FlightGear\data\Scenery;c:\data\WorldScenery  
cherchera d'abord pour des scènes dans le répertoire : c:\Program Files\FlightGear\data\Scenery  
suivi de : c:\data\WorldScenery
```

Le paramétrage de variables d'environnement sur différentes plate-formes dépasse le cadre de ce document.

3.1.4 Téléchargez automatiquement des scènes en plein vol

FlightGear est capable de télécharger automatiquement les scènes pendant que vous volez, si vous avez à votre disposition une connexion Internet permanente. Créez un répertoire de travail 'fonctionnel' pour *TerraSync*, accessible en écriture pour l'utilisateur et faites pointer *FlightGear* vers ce répertoire en utilisant la variable FG_SCENERY (comme expliqué ci-dessus). Ne laissez **pas** *TerraSync* télécharger des scènes dans le répertoire Scenery créé lors de l'installation.

Au sein de *FlightGear* à proprement parler, choisissez l'option Téléchargement de scènes sous le menu Environnement. Ensuite, sélectionnez simplement le répertoire que vous avez créé ci-dessus et activez le téléchargement automatique des scènes.

Le problème de la poule et de l'œuf est présent lorsque vous démarrez pour la première fois dans une nouvelle zone. *FlightGear* s'attend à trouver les scènes pour cette zone, mais il est possible qu'il ne les ait pas encore récupérées. Aussi, dès que *TerraSync* a chargé la nouvelle tuile (ce que vous pouvez vérifier à partir

de la section Statut de la fenêtre Téléchargement de scènes), cliquez sur le bouton Rafraîchissement manuel pour recharger les scènes. Si vous rencontrez toujours des difficultés, redémarrez *FlightGear*.

Un des bénéfices majeurs de *TerraSync* est qu'il récupère toujours la dernière et meilleure version des scènes à partir du *FlightGear World (Custom) Scenery Project* et vous permet ainsi d'obtenir les mises à jour incrémentales indépendamment des versions complètes des scènes mondiales, qui sont généralement synchronisées avec les versions de *FlightGear*.

Utiliser *TerraSync* comme un outil séparé

Il est également possible d'utiliser *TerraSync* comme un outil externe.

Sur Mac OS X ou Windows, le simple fait de cocher "Télécharger les scènes à la volée" dans l'interface graphique de lancement permet d'exécuter *TerraSync* de manière automatique dans un processus séparé pour télécharger les scènes autour de votre avion, de telle sorte que vous n'avez pas besoin de spécifier du tout ni l'option atlas, ni FG_SCENERY.

Sinon, vous pouvez lancer le programme *terrasync* directement. Il communique avec *FlightGear* en utilisant le protocole 'Atlas'. Il vous suffit donc d'appeler *FlightGear* avec les paramètres de ligne de commande suivants :

```
--atlas=socket, out, 1, localhost, 5505, udp
```

et précisez à *TerraSync* le numéro de port que vous utilisez ainsi que le répertoire de destination :

```
terrasync -p 5505 -S -d /usr/local/share/TerraSync
```

Notez que *TerraSync*, lorsqu'il est appelé avec le paramètre de ligne de commande "-S", comme cela est recommandé, va télécharger les scènes à l'aide du protocole Subversion sur HTTP. Aussi, si votre accès Internet est configuré pour utiliser un proxy HTTP, informez-vous de la manière de configurer le client Subversion "libsvn" pour l'utilisation d'un proxy. Si vous utilisez Mac OS X 10.5, l'interface GUI launcher spécifie automatiquement -S si svn est disponible.

3.1.5 Créer vos propres scènes

Si vous êtes intéressé par la création de vos propres scènes, jetez un œil à TerraGear - les outils qui sont utilisés pour générer les scènes pour *FlightGear* :

<http://wiki.flightgear.org/TerraGear>

L'arbre des sources les plus activement maintenues de la chaîne d'outils TerraGear est colocalisée sur le serveur des données géospatiales Mapserver de *FlightGear* :

<http://mapserver.flightgear.org/git/gitweb.pl>.

3.2 Installer des aéronefs

Le paquetage de base de *FlightGear* contient uniquement un petit nombre des aéronefs effectivement disponibles pour *FlightGear*. Les développeurs ont créé une large gamme d'aéronefs, des chasseurs de la seconde guerre mondiale comme le Spitfire aux avions de transport de passagers comme le Boeing 747.

Vous pouvez télécharger les aéronefs à partir de la page :

<http://www.flightgear.org/download/aircraft/>

Téléchargez simplement le fichier et décompressez-le dans le sous-répertoire `data/Aircraft` de votre installation. L'aéronef est téléchargé sous la forme d'un fichier `.zip`. Une fois que vous l'aurez décompressé, un nouveau sous-répertoire sera créé dans votre répertoire `data/Aircraft`, contenant le nouvel aéronef. La prochaine fois que vous lancerez *FlightGear*, le nouvel aéronef sera disponible.

Sous Mac OS X, vous pouvez utiliser le GUI launcher pour installer les fichiers de l'aéronef téléchargé comme décrit dans la section 3.1.2.

3.3 Installer la documentation

La plupart des paquetages cités ci-dessus incluent la documentation complète de *FlightGear*, qui comprend une version au format PDF du *Manuel FlightGear*, qui a été conçu pour un affichage et une impression de qualité en utilisant le logiciel Acrobat Reader d'Adobe, disponible à l'adresse :

<http://www.adobe.com/products/acrobat/>

De plus, si elle est correctement installée, la version HTML de ce document est accessible à partir du menu Aide de *FlightGear*.

Enfin, le code source comporte un répertoire `docs-mini` qui contient de nombreuses idées et solutions pour des problèmes spécifiques. Il s'agit donc également d'un bon endroit pour trouver d'avantage d'informations.

Deuxième partie

Flying with *FlightGear*

Chapitre 4

Décollage : comment démarrer le programme

4.1 Variables d'environnement

Il existe deux variables d'environnement qui doivent être définies pour faire fonctionner *FlightGear*. Elles indiquent à *FlightGear* où trouver ses données et ses scènes.

Vous pouvez les paramétrer de plusieurs façons en fonction de votre plateforme et de vos besoins.

4.1.1 FG_ROOT

Il s'agit de l'emplacement où *FlightGear* recherchera ses fichiers de données comme les aéronefs, les emplacements des balises de navigation, les fréquences des aéroports. Il s'agit du sous-répertoire `data` de l'emplacement où vous avez installé *FlightGear*, par exemple : `/usr/local/share/FlightGear/data` ou `c:\Program Files\FlightGear\data`.

4.1.2 FG_SCENERY

Il s'agit de l'emplacement où *FlightGear* recherchera ses fichiers de scènes. Il s'agit d'une liste de répertoires qui seront analysés de manière séquentielle. Les répertoires sont séparés par “:” sous Unix et “;” sous Windows, par exemple :

```
/home/joebloggs/WorldScenery:/usr/local/share/FlightGear/data/Scenery
```

ou

```
c:\Program Files\FlightGear\data\Scenery;c:\Program Files\FlightGear\data\WorldScenery.
```

4.1.3 Variables d'environnement sous Windows et Mac OS X

Les assistants graphiques sous Windows et le GUI launcher sous Mac OS X définissent de manière interne ces variables d'environnement de telle sorte que

vous n'avez pas besoin de les définir vous-même. Cependant, si jamais vous lancez *FlightGear* depuis la ligne de commande, vous devez définir ces variables de manière explicite.

4.2 Démarrer le simulateur sous Unix/Linux



Fig. 3 : Paré à décoller : en attente à la position de démarrage par défaut de l'aéroport de San Francisco Intl., KSFO.

Avant de pouvoir démarrer *FlightGear*, vous devez définir deux variables d'environnement :

- Vous devez ajouter `/usr/local/share/FlightGear/lib` à votre `LD_LIBRARY_PATH`
- `FG_ROOT` doit être paramétré pour pointer vers le répertoire contenant les données de votre installation de *FlightGear*, par exemple : `/usr/local/share/FlightGear`
- `FG_SCENERY` doit être une liste de répertoires de scènes, séparés par " : ". Ce fonctionnement est semblable à celui de `PATH` lorsqu'on recherche des scènes. par exemple : `$FG_ROOT/Scenery:$FG_ROOT/WorldScenery`.

Pour les ajouter dans le Bourne shell (et compatibles) :

```
LD_LIBRARY_PATH=/usr/local/share/FlightGear/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH
FG_HOME=/usr/local/share/FlightGear
```



```
export FG_HOME
FG_ROOT=/usr/local/share/FlightGear/data
export FG_ROOT
FG_SCENERY=$FG_ROOT/Scenery:$FG_ROOT/WorldScenery
export FG_SCENERY
```

ou en C shell (et compatibles) :

```
setenv LD_LIBRARY_PATH=\
  /usr/local/share/FlightGear/lib:$LD_LIBRARY_PATH
setenv FG_HOME=/usr/local/share/FlightGear
setenv FG_ROOT=/usr/local/share/FlightGear/data
setenv FG_SCENERY=\
  $FG_HOME/Scenery:$FG_ROOT/Scenery:$FG_ROOT/WorldScenery
```

Une fois que ces variables d'environnement ont été paramétrées, démarrez tout simplement *FlightGear* en utilisant la commande `fgfs --option1 --option2...`. Les options de ligne de commande sont décrites dans le chapitre 4.5.

4.3 Démarrer le simulateur sous Windows

Les binaires pré-compilés pour Windows sont livrés avec un assistant graphique pour démarrer *FlightGear*. Cliquez tout simplement sur l'élément *FlightGear* Lancer du menu Démarrer, ou double-cliquez sur l'icône présente sur le Bureau. L'assistant vous permet de choisir votre aéronef, l'aéroport de démarrage et la piste, l'heure du jour, la météo en cours, et de nombreux autres paramètres.

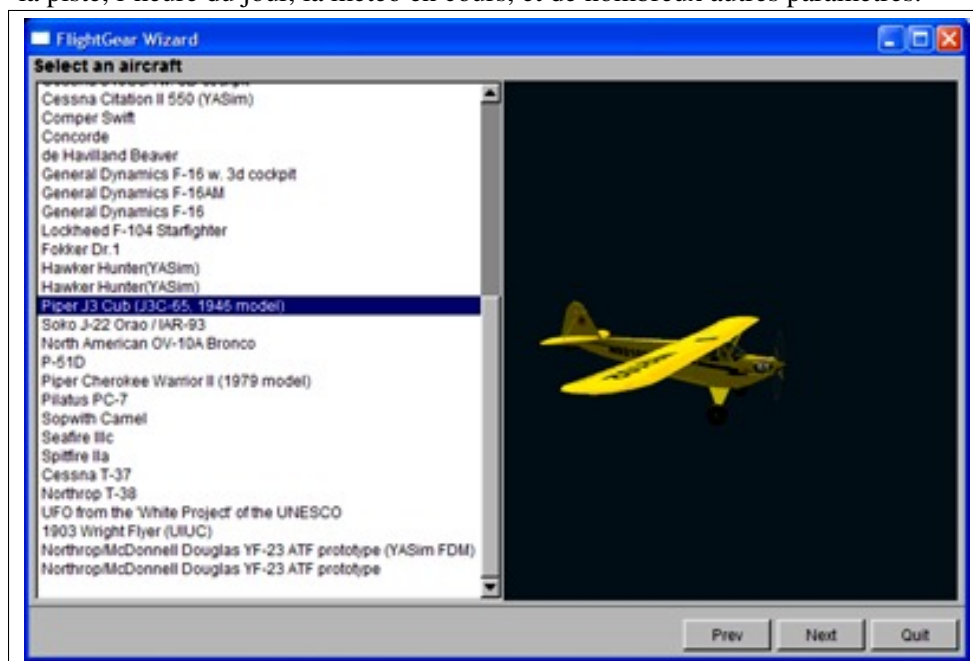


Fig. 4 : L'assistant de démarrage de *FlightGear*

La première fois que vous le lancerez, il vous sera demandé de paramétrer vos variables `FG_ROOT` (généralement `c:\Program Files\FlightGear\data`) et `FG_SCENERY`. Cette dernière variable doit contenir une liste de répertoire(s) où vous avez installé des scènes, typiquement :

```
c:\Program Files\FlightGear\data\Scenery.
```

Si vous indiquez des valeurs incorrectes ou si vous modifiez les répertoires de vos scènes après coup, vous pourrez corriger ces valeurs et appuyant sur le bouton "Précédent" à partir de la première page de l'assistant.

4.3.1 Lancement à partir de la ligne de commande

Alternativement, vous pouvez démarrer *FlightGear* à partir de la ligne de commande. Pour cela, vous devez paramétrer les variables d'environnement `FG_ROOT` et `FG_SCENERY` manuellement.

Ouvrez une invite de commandes, placez-vous dans le répertoire où sont positionnés vos binaires (généralement quelque chose comme `c:\Program Files\FlightGear\bin\`) puis paramétrez les variables d'environnement en tapant :

```
SET FG_HOME="c:\Program Files\FlightGear"
SET FG_ROOT="c:\Program Files\FlightGear\data"
SET FG_SCENERY="c:\Program Files\FlightGear\data\Scenery"
```

et démarrez *FlightGear* (dans la même fenêtre d'invite de commandes, car les variables d'environnement sont valides uniquement localement au sein de la même invite de commandes) par l'intermédiaire de la commande :

```
fgfs --option1 --option2... Les options de ligne de commande
```

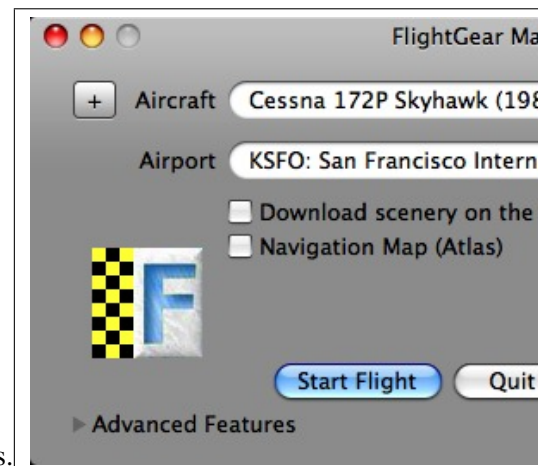
sont décrites dans le chapitre 4.5. Naturellement, vous pouvez créer un fichier *batch* avec un éditeur de texte Windows (comme le bloc-notes) contenant les lignes ci-dessus. Pour obtenir les meilleures performances à l'exécution, il est recommandé de réduire (icônifier) la fenêtre de sortie pendant que *FlightGear* est en fonctionnement.

4.4 Démarrer le simulateur sous Mac OS X

Le paquetage de binaires pré-compilés pour Mac OS X est livré avec un assistant en mode graphique. Double-cliquez simplement sur l'icône *FlightGear* dans le dossier /Applications pour lancer la fenêtre GUI launcher comme démontré à la Fig. 5. L'assistant vous permet :

- de choisir un aéronef et un aéroport
- d'activer/désactiver le téléchargement automatique des scènes (en utilisant *TerraSync*)

- d’activer/désactiver la carte de navigation (Atlas)
- de démarrer *FlightGear*



Les sections suivantes décrivent l’utilisation de ces fonctionnalités.

Fig. 5 : L’assistant graphique pour Mac OS X

4.4.1 Choisir un aéronef et un aéroport

Dans la fenêtre de l’assistant, vous pouvez choisir un aéronef et un aéroport en cliquant sur les boutons situés à droite du nom de l’aéroport/aéronef. L’image miniature de l’aéronef sélectionné s’affichera dans le panneau image de la fenêtre. L’aéroport que vous choisissez ici sera le point de départ de votre vol. *FlightGear* utilise les codes aéroport à quatre lettres de l’OACI, qui se compose d’une ou deux lettres de préfixe, suivies de une ou deux lettres correspondant au code aéroport. Par exemple, le code OACI de l’aéroport international de San Francisco est KSFO. Dans cet exemple, “K” est le préfixe pour les Etats-Unis, et SFO est le code aéroport pour l’aéroport international de San Francisco. Vous pouvez utiliser l’onglet *Fonctionnalités avancées* » *Position* ou *Aéronef* pour trouver un aéroport ou un aéronef.

4.4.2 Activer le téléchargement des scènes à la volée

En activant *Télécharger les scènes à la volée* dans la fenêtre de l’assistant, les données des scènes seront téléchargées pendant que vous volez grâce à *TerraSync*. L’outil de téléchargement des scènes installera ces données dans le répertoire

```
/Applications/FlightGear.app/Contents/Resources/data/Scenery-Terrasync.
```

4.4.3 Activer la carte de navigation (Atlas)

La version Mac de *FlightGear* comprend Atlas, la carte de navigation qui vous aide à suivre votre plan de vol, pour votre confort. L’activation de cette option lancera automatiquement Atlas lorsque vous démarrerez votre vol. Vous n’avez

pas besoin de spécifier d'options Atlas. L'assistant graphique les précisera pour vous.

4.4.4 Démarrer FlightGear - le simulateur

Cliquer sur le bouton *Démarrer le vol* dans la fenêtre de l'assistant ouvre une nouvelle fenêtre - la fenêtre principale de FlightGear. *FlightGear* démarre immédiatement le chargement des tonnes de données nécessaires pour la simulation. Le temps de chargement peut être un peu long sur des machines lentes.

4.4.5 Fonctionnalités avancées

FlightGear possède de nombreuses fonctionnalités et options que vous pouvez spécifier lors du démarrage. Quelques-unes d'entre elles ne peuvent pas être modifiées à partir du menu de la fenêtre principale de *FlightGear*. Pour activer/désactiver ces fonctionnalités, cliquez sur le bouton en forme de triangle dans l'angle inférieur gauche de la fenêtre de l'assistant. Ceci affiche l'onglet fonctionnalités avancées. Tous les paramètres sont sauvegardés de manière à ce que vous n'ayez pas à les respcifier à chaque fois.

Onglets Généralités, Fonctionnalités et Rendu

Ces onglets vous permettent de définir certaines options de *FlightGear* :

- **Sauvegarder les préférences à la fermeture** : permet à *FlightGear* de sauvegarder les préférences que vous avez modifiées à partir du menu dans la fenêtre principale de *FlightGear* (fgfs) vers le fichier `$HOME/.fgfs/autosave.xml`.
- **Contrôles** : précise le moyen de contrôle que vous utiliserez dans *FlightGear*. *auto*, *joystick*, *souris et clavier* sont disponibles. Vous pouvez conserver ce paramètre en mode *auto* sauf si vous tenez à le paramétrer manuellement.
- **Unités** : définit l'unité utilisée dans *FlightGear*. Les *yards* et les *mètres* sont disponibles.
- **Heure du jour** : fixe l'heure du jour. Les options disponibles sont *réelle*, *aube*, *matin*, *midi*, *après-midi*, *crépuscule*, *soir* et *minuit*.
- **Saison** : définit la saison du monde *FlightGear*. Vous pouvez choisir été ou hiver.
- **Son** : la désactivation de cette option coupera le son dans FlightGear.
- **Panneaux d'instruments** : précise si un panneau 2D est activé par défaut. Vous pouvez également activer/désactiver le panneau 2D à partir du menu *FlightGear*.
- **Objets aléatoires** : précise si *FlightGear* affichera des objets aléatoires ou non.
- **Modèles IA** : précise si *FlightGear* gèrera les aéronefs et navires IA.

- **L’horloge n’avance jamais** : définit si l’horloge avance dans *FlightGear* ou non.
- **Pas de consommation de carburant** : laisse l’avion ne consommer aucun carburant lorsqu’il vole.
- **Démarrer dans un état figé** : démarre *FlightGear* dans un état figé. Cette option a l’air de ne pas fonctionner actuellement.
- **Activer l’affichage tête haute (HUD)** : affiche le HUD au démarrage, je pense. Vous pouvez activer/désactiver le HUD en appuyant sur ‘h’ lorsque vous volez.
- **HUD 3D** : active le HUD 3D si un avion sélectionné le prend en charge.
- **Météo réelle** : active la récupération des conditions météo réelles en utilisant les METAR.
- **Effet de l’horizon** : active l’effet de l’horizon.
- **Nuages** : active l’affichage des nuages.
- **Nuages 3D** : active l’affichage des nuages 3D. Vous devez activer les nuages 3D en activant la case *Nuages 3D* dans le menu *Affichage » Options de rendu* dans la fenêtre du simulateur *FlightGear*.
- **Turbulence** : précise la force des turbulences en vol. 0.0 est calme et 1.0 est forte. Vous devez spécifier une valeur allant de 0.0 à 1.0.
- **Visibilité** : spécifie la visibilité en mètres. Vous pouvez donner une valeur de 5 000 ou moins si *FlightGear* tourne lentement sur votre machine.
- **Mode plein écran** : l’activation de ce paramètre lancera *FlightGear* en mode plein écran.
- **Taille de la fenêtre** : définit la taille de la fenêtre *FlightGear*. Sera ignoré si le mode plein écran est activé.
- **Fondu du ciel** : active ou désactive le fondu du ciel dans *FlightGear* en fonction du temps et de l’environnement proche. NE DESACTIVEZ PAS cette option, sous peine de voir *FlightGear* dysfonctionner.
- **Textures** : active ou désactive les textures sur les pistes, les immeubles et les objets des scènes. La désactivation vous donnera quelques images par seconde en plus, spécialement sur les Macs G4.
- **Fil de fer** : demande à *FlightGear* de dessiner le monde en fils de fer, de telle sorte que vous puissiez visualiser la manière dont le monde de *FlightGear* est rendu. Ceci ne devrait être activé que pour des besoins de débogage ou de développement.
- **Brouillard** : précise la manière dont le brouillard est rendu. *désactivé, plus rapide, plus beau* sont les options disponibles.

Onglet Favoris

La liste des favoris offre un moyen de conserver un état nommé des options actuelles, à la façon du marque-pages d’un navigateur Internet. Pour sauvegarder l’état actuel des options, appuyez sur le bouton “+” dans le haut de la fenêtre ou en-bas de l’onglet Favoris. Une fois que les favoris ont été ajoutés, vous pouvez

basculer d'une configuration à l'autre en double-cliquant sur une ligne du tableau dans l'onglet Favoris. L'appui sur le bouton “-” (ou sur la touche suppression) sur un favori dans le tableau efface le favori sélectionné.

Position

Vous pouvez trouver des aéroports ou des porte-avions en cherchant en utilisant une recherche par mot-clé dans la zone de texte filtre. Les mots-clés disponibles sont :

- Pour les aéroports :
 - une partie du nom de l'aéroport (par exemple : international, ranch, ou civil),
 - le nom du pays (comme Japon, USA, ou France) si connu,
 - le nom de l'emplacement (comme la ville ou le nom abrégé de l'état aux Etats-Unis),
 - code IATA (comme SFO, LAX, HND) si disponible,
 - code OACI (comme KSFO, KLAX, RJTT) - fonctionne pour tous les aéroports.
- Pour les porte-avions :
 - nom du porte-avions (Nimitz ou Eisenhower),
 - “carrier”.

Les aéroports et les porte-avions qui correspondent au mot-clé sont affichés dans la vue du tableau sous la zone de texte filtre. Vous pouvez également ouvrir cet onglet en cliquant sur le bouton situé à l'extrémité droite du nom de l'aéroport sur le panneau supérieur. Lorsque vous choisissez un aéroport, les pistes disponibles s'affichent dans le bouton pop-up “piste”. Vous pouvez choisir une piste ou le laisser sur “défaut.”

Onglet Aéronefs

Vous pouvez trouver les aéronefs en cherchant avec un mot clé dans la zone de texte filtre. Les mot-clés disponibles sont :

- une partie du nom de l'aéronef (par exemple : c172, zero, shinden)
- le nom d'un fichier de configuration sans le -set.xml (par exemple : a6m2, p51d)
- le nom du FDM (Modèle de dynamique de vol) (par exemple : yasim, jsb)
- le statut (par exemple : alpha, beta, production, développement)

Une liste des aéronefs correspondant au mot clé s'affichent dans la vue du tableau sous la zone de texte filtre. Le nom d'aéronef sur le panneau supérieur est synchronisé avec l'aéronef actuellement sélectionné. Vous pouvez également ouvrir cet onglet en cliquant sur le bouton situé à l'extrémité droite du nom de l'avion sur le panneau supérieur.

Si vous voulez trouver plus d'aéronefs à partir d'Internet, cliquez sur le bouton “Obtenir des aéronefs” en bas de l'onglet Aéronefs. Ceci ouvrira un navigateur

pour vous mener à la page Liens Aéronefs sur le site Internet *FlightGear Mac OS X*. Vous pouvez consulter certains liens qui s’y trouvent et télécharger l’aéronef que vous souhaitez. Une fois que l’aéronef est téléchargé, ouvrez l’onglet “Autres” pour l’installer.

Si vous constatez un comportement anormal lorsque vous recherchez un aéronef, supprimer un fichier de cache de la base de données des aéronefs peut résoudre votre problème. Ouvrez

`/Applications/Utilities/Terminal.app` et tapez les commandes suivantes pour supprimer le fichier de cache :

```
cd /Applications/FlightGear.app/Contents/Resources  
rm AircraftCache.rb
```

Notez que vous ne devez pas lancer *FlightGear* à partir de l’image disque d’un paquetage binaire précompilé car le répertoire de l’image disque montée est en lecture seule, ce qui vous empêche d’installer toute donnée ajoutée. Vous devez installer *FlightGear* en copiant l’icône *FlightGear* depuis l’image disque montée vers le répertoire `/Applications`.

Onglet réseau

Cet onglet contient deux fonctionnalités réseau de *FlightGear*. L’une est le mode multijoueurs et l’autre est FGCOM (un système de contrôle aérien vocal). Pour activer le mode multijoueurs, précisez les items suivants :

- **Activer le mode multijoueurs** : active ou désactive le mode multijoueurs. L’activation ouvrira une nouvelle fenêtre du navigateur pour afficher la carte sur un serveur multijoueurs.
- **Indicatif** : précise le nom affiché sur la carte. Un nom d’utilisateur peut contenir des lettres de l’alphabet, des nombres, des tirets-bas (*underscores*) et des points. *FlightGear* se terminera si vous précisez un indicatif avec des caractères invalides. Un nom d’utilisateur contenant 8 caractères ou plus sera tronqué avec les seuls 7 premiers caractères.
- **Serveur** : précise le serveur auquel se connecter. `mpserver02.flightgear.org` est disponible à l’heure de l’écriture de ce document.
- **Votre Mac** : définit l’adresse IP de votre Mac. Généralement, l’assistant détecte l’adresse de manière automatique. Si quelque chose n’est pas exact avec votre connexion réseau, effacez le contenu de ce champ texte.
- **Port** : doit être égal à 5000.

FGCOM vous permet d’effectuer une véritable communication audio en utilisant des paramètres radio. Vous pouvez parler sur une fréquence radio précise (COM1) en appuyant sur la barre d’espace. Vous pouvez écouter d’autres joueurs parlant sur la fréquence si d’autres joueurs sont sur cette fréquence et que vous êtes à leur portée. Les options pour FGCOM sont listées ci-dessous :

- **Active FGCOM en mode multijoueurs** : indique si vous utilisez FGCOM lorsque le mode multijoueurs est activé.
- **Serveur FGCOM** : précise un serveur VOIP.

Des informations complémentaires concernant FGCOM et le mode multijoueurs sont disponibles au chapitre 6.

Onglet Autres

Vous pouvez préciser toutes les options autres que aéronef, aéroport et les options affichées dans l'assistant. En entrant des options séparées par des espaces comme présenté à la Figure 11, vous passerez des options additionnelles à *FlightGear*. Vous pouvez voir toutes les options disponibles en appuyant sur "Voir Options." Certaines options peuvent faire dysfonctionner *FlightGear*. Si vous rencontrez de tels dysfonctionnements avec une option particulière, merci de nous le signaler.

Pour installer des aéronefs et des scènes additionnelles, appuyez sur "Installer des données additionnelles." Vous pouvez préciser les fichiers ou répertoires multiples à installer dans le répertoire des données *FlightGear*. Les types de fichiers acceptés sont :

- zip
- tar
- tar.gz
- tar.bz2
- dossier
- 7zip (seulement si 7zip est installé)

Vous verrez la fenêtre d'information lorsque toutes les données ont été correctement installées, dans le cas contraire les messages d'erreurs y seront affichés. Vous pouvez choisir des données d'aéronefs et de scènes en même temps. Si vous choisissez un fichier archivé qui ne contient pas de fichiers d'aéronefs, il sera extrait dans le répertoire des données, mais sera ignoré. Lorsque vous aurez terminé l'installation d'un nouvel aéronef, vous pouvez choisir l'aéronef concerné dans l'onglet "Aéronefs".

4.4.6 Démarrage à partir de la ligne de commande

Vous pouvez également démarrer le simulateur à partir de la ligne de commande sur Mac OS X. Pour le faire, ouvrez Terminal.app (situé dans /Applications/Utilitaires) et tapez les commandes suivantes :

```
cd /Applications/FlightGear.app/Contents/Resources
./fgfs --option1 --option2 ....
```

Reportez-vous au chapitre 4.5 pour obtenir plus d'informations détaillées sur les options en ligne de commande. Contrairement aux autres plate-formes, vous n'avez pas besoin de préciser manuellement les variables d'environnement comme

FG_ROOT et FG_SCENERY tant que vous utilisez un paquetage binaire précompilé.

4.5 Command line parameters

Following is a complete list and short description of the numerous command line options available for *FlightGear*. Most of these options are exposed through the *FlightGear* launcher delivered with the Windows binaries.

If you have options you re-use continually, you can create a preferences file containing a set of command-line options that will be set automatically. You can create the file with any text editor (notepad, emacs, vi, if you like).

- On Unix systems (including Mac OS X), put the command line options in a file called `.fgfsrc` in your home directory.
- On Windows, put the command line options in a file called `system.fgfsrc` in the FG_ROOT directory (e.g. `c:\Program Files\FlightGear\data`).

4.5.1 General Options

- `--help`
Display the most relevant command line options.
- `--help --verbose`
Display all command line options.
- `--version`
Display the current *FlightGear* version.
- `--fg-root=path`
Tells *FlightGear* where to look for its root data files if you didn't compile it with the default settings.
- `--fg-scenery=path`
Allows specification of a path to the base scenery path, in case scenery is not at the default position under `$FG_ROOT/Scenery`; this might be especially useful in case you have scenery on a CD-ROM.
- `--fg-aircraft=path`
Allows specification of a paths to aircraft. Defaults to `$FG_ROOT/Aircraft`.
- `--language=code`
Select the language for this session. e.g. pl, nl, it, fr, en, de.
- `--restore-defaults`
Reset all user settings to their defaults
- `--enable-save-on-exit, --disable-save-on-exit`
Enable or disable saving of user-preferences on exit from the simulator.
- `--enable-freeze, --disable-freeze`
Control whether *FlightGear* starts paused or not. Defaults to not paused.
- `--control=mode`

Specify your primary control mode (joystick, keyboard, mouse) Defaults to joystick.

- `--enable-auto-coordination, --disable-auto-coordination`
Switches auto-co-ordination between aileron and rudder on/off. Auto-coordination is recommended for users without rudder pedals or a 'twist' joystick. Defaults to off.
- `--browser-app=path`
Specify location of your web browser. E.g : `--browser-app="C:\Program Files\Internet Explorer\iexplore.exe"` (Note the "" because of the spaces!).
- `--config=path`
Load additional properties from the given path. E.g :
`--config=./Aircraft/X15-set.xml`
- `--units-feet`
Use feet as the unit of measurement.
- `--units-meters`
Use meters as the unit of measurement.

4.5.2 Features

- `--enable-ai-models, --disable-ai-models`
Enable or disable other aircraft/AI-models in the simulator.
- `--ai-scenario=name`
Enable a specific AI scenario (e.g. `--ai-scenario=vinson-demo`). May be used multiple times.

4.5.3 Sound

- `--enable-sound, --disable-sound`
Enable or disable sound.
- `--show-sound-devices`
Show the available sound devices.
- `--sound-device=device`
Specify the sound device to use for audio.
- `--enable-intro-music, --disable-intro-music`
Enable or disable playing an audio sample when *FlightGear* starts up.

4.5.4 Aircraft

- `--aircraft=aircraft`
Load the specified aircraft, for example : `--aircraft=c172p`. For available choices check the directory `$FG_ROOT/Aircraft`, and look for files ending in `"-set.xml"`. When specifying the aircraft, drop the `"-set.xml"`

from the filename. Alternatively, use the `--show-aircraft` option described below to list the available aircraft. For information on downloading additional aircraft, see Section 3.2.

- `--show-aircraft`
Print a sorted list of the currently available aircraft types.
- `--min-status=status`
Display only those aircraft with a specified minimum declared status, one of `alpha`, `beta`, `early-production`, `production`. For use with `--show-aircraft`.
- `--aircraft-dir=PATH`
Aircraft directory relative to the executable location. Defaults to `$FG_ROOT/Aircraft`.
- `--vehicle=name of aircraft definition file`
Synonym for `--aircraft`.
- `--livery=Name`
Set the aircraft livery.

4.5.5 Flight model

- `--fdm=abcd`
Select the core flight model. Options are `jsb`, `larcsim`, `yasim`, `magic`, `balloon`, `external`, `pipe`, `ada`, `null`. This option can normally be ignored, as the `--aircraft` option will set the FDM correctly.
- `--aero=aircraft`
Specifies the aircraft aeronautical model to load. This option can normally be ignored, as the `--aircraft` option will set the aircraft model correctly.
- `--model-hz=n`
Run the Flight Dynamics Model with this rate (iterations per second).
- `--speed=n`
Run the Flight Dynamics Model this much faster than real time.
- `--trim`, `--notrim`
Trim (or not) when initializing JSBSim. Defaults to `trim`.
- `--on-ground`, `--in-air`
Start up at ground level (default), or in the air. If specifying `--in-air` you must also set an initial altitude using `--altitude`, and may also want to set an initial velocity with `--vc`. Note that some aircraft (notably the X15) must be started in mid-air.
- `--enable-fuel-freeze`, `--disable-fuel-freeze`
Control whether fuel state is constant (e.g. frozen) or consumed normally (default).

4.5.6 Initial Position and Orientation

- `--airport=ABCD`
Start at a specific airport. The airport is specified by its ICAO code, e.g.

- `--airport=KJFK` for JFK airport in New York. For US airport without an ICAO code, try prefixing the 3 character code with 'K'.
- `--parking-id=ABCD`
Start at a specific parking spot on the airport.
- `--runway=NNN`
Start at the threshold of a specific runway (e.g. 28L). If no runway or parking ID is specified, a runway facing into the wind will be chosen for takeoff.
- `--vor=ABCD, --ndb=ABCD, --fix=ABCD`
Set the starting position relative to a VOR, NDB or FIX. Useful for practising approaches.
- `--carrier=NAME`
Start on an aircraft carrier. See 6.2 for details of carrier operations.
- `--parkpos=NAME`
Start at a particular parking position on the carrier. Must be used with `--carrier`. Defaults to a catapult launch position.
- `--offset-distance=nm, --offset-azimuth=deg`
Start at a particular distance and heading from a position set using `--airport, --vor, --ndb, --fix, --carrier`.
- `--lon=degrees, --lat=degrees`
Start at a particular longitude and latitude, in decimal degrees (south, west negative).
- `--altitude=feet`
Start at specific altitude. Implies `--in-air`. Altitude is specified in feet unless you also select `--units-meters`, in which case altitude is in meters. You may also want to set an initial velocity with `--vc` to avoid stalling immediately.
- `--heading=degrees, --roll=degrees, --pitch=degrees`
Set the initial orientation of the aircraft. All values default to 0 - heading North, in straight and level flight.
- `--uBody=X, --vBody=Y, --wBody=Z`
Set the initial speed along the X, Y and Z axes. Speed is in feet per second unless you also select `--units-meters`, in which case altitude is in meters per second.
- `--vNorth=N, --vEast=E, --vDown=D`
Set the initial speed along the South-North, West-East and vertical axes. Speed is in feet per second unless you also select `--units-meters`, in which case altitude is in meters per second.
- `--vc=knots, --mach=num`
Set the initial airspeed in knots or as a Mach number. Useful if setting `--altitude`, unless you want to stall immediately !
- `--glideslope=degrees, --roc=fpm`
Set the initial glide slope angle in degrees or as a climb rate in feet per minute. May be positive or negative.

4.5.7 Environment Options

- `--ceiling=FT_ASL[:THICKNESS_FT]`
Sets an overcast ceiling at a particular height, and with an optional thickness (defaults to 2000ft).
- `--enable-real-weather-fetch, --disable-real-weather-fetch`
Control whether real-time weather information is downloaded and used.
- `--metar=METAR STRING`
Use a specific METAR string, e.g. `--metar="XXXX 012345Z 0000KT 99SM CLR 19/M01 A2992"`. The METAR may be specified in most common formats (US, European). Incompatible with `--enable-real-weather-fetch`.
- `--random-wind`
Sets random wind direction and strength.
- `--turbulence=n`
Sets turbulence from completely calm (0.0) to severe (1.0).
- `--wind=DIR@SPEED`
Specify the surface wind. Direction is in degrees, and speed in knots. Values may be specified as a range by using a colon separator; e.g. `--wind=180:220@10:15`.
- `--season=param`
Sets the simulated season. Valid parameters are `summer` (default), `winter`.
- `--visibility=meters, --visibility-miles=miles`

4.5.8 Rendering Options

- `--aspect-ratio-multiplier=N`
Set a multiplier for the display aspect ratio.
- `--bpp=depth`
Specify the bits per pixel.
- `--enable-clouds, --disable-clouds`
Enable (default) or disable cloud layers.
- `--enable-clouds3d, --disable-clouds3d`
Enable (default), disable 3D clouds. Very pretty, but depend on your graphics card supporting GLSL Shaders, which some older, or less powerful graphics cards do not.
- `--enable-distance-attenuation, --disable-distance-attenuation`
Enable or disable more realistic runway and approach light attenuation.
- `--enable-enhanced-lighting, --disable-enhanced-lighting`
Enable or disable more realistic runway and approach lights.
- `--enable-fullscreen, --disable-fullscreen`
Enable, disable (default) full screen mode.
- `--enable-game-mode, --disable-game-mode`
Enable or disable full screen display for 3DFX graphics cards.
- `--enable-horizon-effect, --disable-horizon-effect`
Enable (default), disable celestial body growth illusion near the horizon.

- `--enable-mouse-pointer, --disable-mouse-pointer`
Enable, disable (default) extra mouse pointer. Useful in full screen mode for old Voodoo based cards.
- `--enable-panel, --disable-panel`
Enable (default) the instrument panel.
- `--enable-random-buildings, --disable-random-building`
Enable, disable (default) random buildings. Note that random buildings take up a lot of memory.
- `--enable-random-objects, --disable-random-objects`
Enable (default), disable random scenery objects.
- `--enable-random-vegetation, --disable-random-vegetation`
Enable (default), disable random vegetation such as trees. Requires a graphics card that supports GLSL Shaders, which some older, or less powerful graphics cards do not.
- `--enable-rembrandt, --disable-rembrandt`
Enable, disable (default) an experimental feature that includes enhanced lighting and realtime shadows.
- `--enable-skyblend, --disable-skyblend`
Enable (default), disable fogging/haze.
- `--enable-specular-highlight, --disable-specular-highlight`
Enable (default), disable specular highlights.
- `--enable-splash-screen, --disable-splash-screen`
Enable or disable (default) the rotating 3DFX logo when the accelerator board gets initialized (3DFX only).
- `--enable-textures, --disable-textures`
Enable (default), disable use of textures.
- `--enable-wireframe, --disable-wireframe`
Enable, disable (default) display of wireframes. If you want to know what the world of *FlightGear* looks like internally, try this !
- `--fog-disable, --fog-fastest, --fog-nicest`
Set the fog level. To cut down the rendering efforts, distant regions vanish in fog by default. If you disable fog you'll see farther, but your frame rates will drop. Using `--fog-fastest` will display a less realistic fog, by increase frame rate. Default is `--fog-nicest`.
- `--fov=degrees`
Sets the field of view in degrees. Default is 55.0.
- `--materials-file=file`
Specify the materials file used to render the scenery. Default : `Materials/regions/materials`
- `--geometry=WWWxHHH`
Defines the window/screen resolution. E.g. `--geometry=1024x768..`
- `--shading-smooth, --shading-flat`
Use smooth shading (default), or flat shading which is faster but less pretty.
- `--texture-filtering=N`
Configure anisotropic texture filtering. Values are 1 (default), 2, 4, 8 or 16.

- `--view-offset=xxx`
Allows setting the default forward view direction as an offset from straight ahead. Possible values are `LEFT`, `RIGHT`, `CENTER`, or a specific number of degrees. Useful for multi-window display.

4.5.9 HUD Options

- `--enable-anti-alias-hud, --disable-anti-alias-hud`
Control whether the HUD (**Head Up Display**) is shown anti-aliased.
- `--enable-hud, --disable-hud`
Control whether the HUD is displayed. Defaults to disabled.
- `--enable-hud-3d, --disable-hud-3d`
Control whether the 3D HUD is displayed.
- `--hud-culled, --hud-tris`
Display the percentage of triangles culled, or the number of triangles rendered in the HUD. Mainly of interest to graphics developers.

4.5.10 Aircraft System Options

- `--adf=[radial:]frequency`
Set the ADF frequency and radial.
- `--com1=frequency, --com2=frequency`
Set the COM1/COM2 radio frequency.
- `--dme=nav1|nav2|frequency`
Set the DME to NAV1, NAV2, or a specific frequency and radial.
- `--failure=system`
Fail a specific aircraft system. Valid systems are `pitot`, `static`, `vacuum`, `electrical`. Specify multiple times to fail multiple systems.
- `--nav1=[radial:]frequency, --nav2=[radial:]frequency`
Set the NAV1/NAV2 radio frequency and radial.

4.5.11 Time Options

- `--enable-clock-freeze, --disable-clock-freeze`
Control whether time advances normally or is frozen.
- `--start-date-gmt=yyyy:mm:dd:hh:mm:ss, --start-date-lat=yyyy:mm:dd:hh:mm:ss, --start-date-sys=yyyy:mm:dd:hh:mm:ss`
Specify the exact startup time/date. The three functions differ in that they take either Greenwich Mean Time, the local time of your virtual flight, or your computer system's local time as the reference point.
Incompatible with `--time-match-local, --time-match-real`.
- `--time-match-local, --time-match-real`
`--time-match-real`, is default : Simulator time is read from the system clock, and is used as is. When your virtual flight is in the same timezone as where

your computer is located, this may be desirable, because the clocks are synchronized. However, when you fly in a different part of the world, it may not be the case, because there is a number of hours difference, between the position of your computer and the position of your virtual flight.

The option `--time-match-local` takes care of this by computing the timezone difference between your real world time zone and the position of your virtual flight, and local clocks are synchronized.

Incompatible with `--start-date-gmt`, `--start-date-lat`, `--start-date-sys`.

- `--time-offset=[+/-]hh:mm:ss`

Specify a time offset relative to one of the time options above.

- `--timeofday=param`

Set the time of day. Valid parameters are `real`, `dawn`, `morning`, `noon`, `afternoon`, `dusk`, `evening`, `midnight`.

4.5.12 Network Options

- `--multiplay=dir,Hz,host,port`, `--callsign=ABCD`

Set multiplay options and aircraft call-sign. See Section 6.1.

- `--httpd=port`, `--telnet=port`

Enable http server or telnet server on the specified port to provide access to the property tree.

- `--jpg-httpd=port`

Enable screen shot http server on the specified port.

- `--proxy=[user:password@]host:port`

Specify a proxy server to use.

4.5.13 Route/Waypoint Options

- `--wp=ID[@alt]`

Allows specifying a waypoint for the GC autopilot ; it is possible to specify multiple waypoints (i.e. a route) via multiple instances of this command.

- `--flight-plan=[file]`

This is more comfortable if you have several waypoints. You can specify a file to read them from.

NB : These options are rather geared to the advanced user who knows what he is doing.

4.5.14 IO Options

More detailed descriptions of the various IO parameters can be found in the README.IO file within the Docs directory of your *FlightGear* installation.

- `--atlas=params`

Open connection using the Atlas protocol (used by Atlas and TerraSync).

- `--atcsim=params`

- Open connection using the ATC Sim protocol (atc610x).
- --AV400=*params*
- Open connection to drive a Garmin 196/296 series GPS
- --AV400Sim=*params*
- Open connection to drive a Garmin 400 series GPS
- --generic=*params*
- Open connection using the Generic (XML-defined) protocol.
- --garmin=*params*
- Open connection using the Garmin GPS protocol.
- --joyclient=*params*
- Open connection to an Agwagon joystick.
- --jsclient=*params*
- Open connection to a remote joystick.
- --native-ctrls=*params*
- Open connection using the FG native Controls protocol.
- --native-fdm=*params*
- Open connection using the FG Native FDM protocol.
- --native-gui=*params*
- Open connection using the FG Native GUI protocol.
- --native=*params*
- Open connection using the FG Native protocol.
- --nmea=*params*
- Open connection using the NMEA protocol.
- --opengc=*params*
- Open connection using the OpenGC protocol.
- --props=*params*
- Open connection using the interactive property manager.
- --pve=*params*
- Open connection using the PVE protocol.
- --ray=*params*
- Open connection using the RayWoodworth motion chair protocol.
- --rul=*params*
- Open connection using the RUL protocol.

4.5.15 Debugging options

- --enable-fpe
- Enable abort on a Floating Point Exception.
- --fgviewer
- Instead of loading the entire simulator, load a lightweight OSG viewer. Useful for checking aircraft models.
- --log-level=*LEVEL*
- Set the logging level. Valid values are bulk, debug, info, warn, alert.
- --prop: [type:]name=value

Set property name to value

Example : `--prop:/engines/engine[0]/running=true` starts the simulator with running engines.

Another example :

`--aircraft=c172p`

`--prop:/consumables/fuels/tank[0]/level-gal=10`

`--prop:/consumables/fuels/tank[1]/level-gal=10`

fills the Cessna for a short flight. You may optionally specific the property type (double, string, boolean).

- `--trace-read=params`

Trace the reads for a property ; multiple instances are allowed.

- `--trace-write=params`

Trace the writes for a property ; multiple instances are allowed.

4.6 Joystick support

Could you imagine a pilot in his or her Cessna controlling the machine with a keyboard alone ? For getting the proper feeling of flight you will need a joystick/yoke plus rudder pedals, right ? However, the combination of numerous types of joysticks, flightsticks, yokes, pedals etc. on the market with the several target operating systems, makes joystick support a non-trivial task in *FlightGear*.

FlightGear has integrated joystick support, which automatically detects any joystick, yoke, or pedals attached. Just try it ! If this does work for you, lean back and be happy ! You can see what *FlightGear* has detected your joystick as by selecting Help -> Joystick Information from the menu.

Unfortunately, given the several combinations of operating systems supported by *FlightGear* (possibly in foreign languages) and joysticks available, chances are your joystick does not work out of the box. Basically, there are two alternative approaches to get it going, with the first one being preferred.

4.6.1 Built-in joystick support

General remarks

In order for joystick auto-detection to work, a joystick bindings xml file must exist for each joystick. This file describes what axes and buttons are to be used to control which functions in *FlightGear*. The associations between functions and axes or buttons are called "bindings". This bindings file can have any name as long as a corresponding entry exists in the joysticks description file

`/FlightGear/joysticks.xml`

which tells *FlightGear* where to look for all the bindings files. We will look at examples later.

FlightGear includes several such bindings files for several joystick manufacturers in folders named for each manufacturer. For example, if you have a CH Products joystick, look in the folder

```
/FlightGear/Input/Joysticks/CH
```

for a file that might work for your joystick. If such a file exists and your joystick is working with other applications, then it should work with *FlightGear* the first time you run it. If such a file does not exist, then we will discuss in a later section how to create such a file by cutting and pasting bindings from the examples that are included with *FlightGear*.

Verifying your joystick is working

Does your computer see your joystick? One way to answer this question under Linux is to reboot your system and immediately enter on the command line

```
dmesg | grep Joystick
```

which pipes the boot message to `grep` which then prints every line in the boot message that contains the string “Joystick”. When you do this with a Saitek joystick attached, you will see a line similar to this one :

```
input0: USB HID v1.00 Joystick [SAITEK CYBORG 3D USB] on  
usb2:3.0
```

This line tells us that a joystick has identified itself as SAITEK CYBORG 3D USB to the operating system. It does not tell us that the joystick driver sees your joystick. If you are working under Windows, the method above does not work, but you can still go on with the next paragraph.

Confirming that the driver recognizes your joystick

FlightGear ships with a utility called `js_demo`. It will report the number of joysticks attached to a system, their respective “names”, and their capabilities. Under Linux, you can run `js_demo` from the folder `/FlightGear/bin` as follows :

```
$ cd /usr/local/FlightGear/bin  
$ ./js_demo
```

Under Windows, open a command shell (Start|All Programs|Accessories), go to the *FlightGear* binary folder and start the program as follows (given *FlightGear* is installed under `c:\Flightgear`)

```
cd \FlightGear\bin  
js_demo.exe
```

Under Mac OS X, open Terminal.app (`/Applications/Utilities/`) and run `js_demo` as follows :

```
$ cd /Applications/FlightGear.app/Contents/Resources
$ ./js_demo
```

On our system, the first few lines of output are (stop the program with Ctrl-C if it is quickly scrolling past your window !) as follows :

```
Joystick test program.
Joystick 0: "CH PRODUCTS CH FLIGHT SIM YOKE USB "
Joystick 1: "CH PRODUCTS CH PRO PEDALS USB"
Joystick 2 not detected
Joystick 3 not detected
Joystick 4 not detected
Joystick 5 not detected
Joystick 6 not detected
Joystick 7 not detected
+-----JS.0-----+-----JS.1-----+
| Btns Ax:0 Ax:1 Ax:2 Ax:3 Ax:4 Ax:5 Ax:6 | Btns Ax:0 Ax:1 Ax:2 |
+-----+-----+
| 0000 +0.0 +0.0 +1.0 -1.0 -1.0 +0.0 +0.0 . | 0000 -1.0 -1.0 -1.0 . . . . |
```

First note that `js_demo` reports which number is assigned to each joystick recognized by the driver. Also, note that the “name” each joystick reports is also included between quotes. We will need the names for each bindings file when we begin writing the binding xml files for each joystick.

Identifying the numbering of axes and buttons

Axis and button numbers can be identified using `js_demo` as follows. By observing the output of `js_demo` while working your joystick axes and buttons you can determine what axis and button numbers are assigned to each joystick axis and button. It should be noted that numbering generally starts with zero.

The buttons are handled internally as a binary number in which bit 0 (the least significant bit) represents button 0, bit 1 represents button 1, etc., but this number is displayed on the screen in hexadecimal notation, so :

```
0001 ⇒ button 0 pressed
0002 ⇒ button 1 pressed
0004 ⇒ button 2 pressed
0008 ⇒ button 3 pressed
0010 ⇒ button 4 pressed
0020 ⇒ button 5 pressed
0040 ⇒ button 6 pressed
... etc. up to ...
8000 ⇒ button 15 pressed
... and ...
```

0014 ⇒ buttons 2 and 4 pressed simultaneously
... etc.

For Linux users, there is another option for identifying the “name” and the numbers assigned to each axis and button. Most Linux distributions include a very handy program, “jstest”. With a CH Product Yoke plugged into the system, the following output lines are displayed by jstest :

```
jstest /dev/js3
Joystick (CH PRODUCTS CH FLIGHT SIM YOKE USB ) has 7 axes and 12 buttons. Driver version is 2.1.0
Testing...(interrupt to exit)
Axes: 0: 0 1: 0 2: 0 3: 0 4: 0 5: 0 6: 0 Buttons: 0:off 1:off 2:off 3:on 4:off 5:off 6:off 7:off
8:off 9:off 10:off 11:off
```

Note the “name” between parentheses. This is the name the system associates with your joystick.

When you move any control, the numbers change after the axis number corresponding to that moving control and when you depress any button, the “off” after the button number corresponding to the button pressed changes to “on”. In this way, you can quickly write down the axes numbers and button numbers for each function without messing with binary.

Writing or editing joystick binding xml files

At this point, you have confirmed that the operating system and the joystick driver both recognize your joystick(s). You also know of several ways to identify the joystick “name” your joystick reports to the driver and operating system. You will need a written list of what control functions you wish to have assigned to which axis and button and the corresponding numbers.

Make the following table from what you learned from `js_demo` or `jstest` above (pencil and paper is fine). Here we assume there are 5 axes including 2 axes associated with the hat.

Axis	Button
elevator = 0	view cycle = 0
rudder = 1	all brakes = 1
aileron = 2	up trim = 2
throttle = 3	down trim = 3
leftright hat = 4	extend flaps = 4
foreaft hat = 5	retract flaps = 5
	decrease RPM = 6
	increase RPM = 7

We will assume that our hypothetical joystick supplies the “name” QUICK STICK 3D USB to the system and driver. With all the examples included with *FlightGear*, the easiest way to get a so far unsupported joystick to be auto detected, is to edit an existing binding xml file. Look at the xml files in the sub-folders

of `/FlightGear/Input/Joysticks/`. After evaluating several of the xml binding files supplied with *FlightGear*, we decide to edit the file `/FlightGear/Input/Joysticks/Saitek/Cyborg-Gold-3d-USB.xml`. This file has all the axes functions above assigned to axes and all the button functions above assigned to buttons. This makes our editing almost trivial.

Before we begin to edit, we need to choose a name for our bindings xml file, create the folder for the QS joysticks, and copy the original xml file into this directory with this name.

```
$ cd /usr/local/FlightGear/Input/Joysticks
$ mkdir QS
$ cd QS
$ cp /usr/local/FlightGear/Input/Joysticks/Saitek/
Cyborg-Gold-3d-USB.xml QuickStick.xml
```

Here, we obviously have supposed a Linux/UNIX system with *FlightGear* being installed under `/usr/local/FlightGear`. For a similar procedure under Windows with *FlightGear* being installed under `c:FlightGear`, open a command shell and type

```
c:
cd /FlightGear/Input/Joysticks
mkdir QS
cd QS
copy /FlightGear/Input/Joysticks/Saitek/
Cyborg-Gold-3d-USB.xml QuickStick.xml
```

Next, open `QuickStick.xml` with your favorite editor. Before we forget to change the joystick name, search for the line containing `<name>`. You should find the line

```
<name>SAITEK CYBORG 3D USB</name>
```

and change it to

```
<name>QUICK STICK 3D USB</name>.
```

This line illustrates a key feature of xml statements. They begin with a `<tag>` and end with a `</tag>`.

You can now compare your table to the comment table at the top of your file copy. Note that the comments tell us that the Saitek elevator was assigned to axis 1. Search for the string

```
<axis n="1">
```

and change this to

```
<axis n="0">.
```

Next, note that the Saitek rudder was assigned to axis 2. Search for the string

```
<axis n="2">
```

and change this to

```
<axis n="1">.
```

Continue comparing your table with the comment table for the Saitek and changing the axis numbers and button numbers accordingly. Since QUICKSTICK USB and the Saitek have the same number of axes but different number of buttons, you must delete the buttons left over. Just remember to double-check that you have a closing tag for each opening tag or you will get an error using the file.

Finally, be good to yourself (and others when you submit your new binding file to a *FlightGear* developers or users archive!) by taking the time to change the comment table in the edited file to match your changed axis and button assignments. The new comments should match the table you made from the `js_demo` output. Save your edits.

Several users have reported that the numbers of axes and buttons assigned to functions may be different with the same joystick under Windows and Linux. The above procedure should allow one to easily change a binding xml file created for a different operating system for use by their operating system.

Telling *FlightGear* about your new bindings xml file

Before *FlightGear* can use your new xml file, you need to edit the file `/FlightGear/joysticks.xml`, adding a line that will include your new file if the “name” you entered between the name tags matches the name supplied to the driver by your joystick. Add the following line to `joysticks.xml`.

```
<js-named include="Input/Joysticks/QS/QuickStick.xml"/>
```

You can tell how *FlightGear* has interpreted your joystick setup by selecting Help -> Joystick Information from the Menu.

Some hints for Windows users

Basically, the procedures described above should work for Windows as well. If your joystick/yoke/pedals work out of the box or if you get it to work using the methods above, fine. Unfortunately, however, it’s likely you may encounter a few problems.

The first one concerns users of non-US Windows versions. As stated above, you can get the name of the joystick from the program `js_demo`. If you have a non-US version of Windows and the joystick .xml files named above do not contain that special name, just add it on top of the appropriate file in the style of

```
<name>Microsoft-PC-Joysticktreiber </name>
```

No new entry in the base `joysticks.xml` file is required.

Unfortunately, there is one more loophole with Windows joystick support. In the case that you have two USB devices attached (for instance, a yoke plus pedals), it's possible that the same driver name will be reported twice. In this case, you can at least get the yoke working by assigning it number 0 (out of 0 and 1). For this purpose, rotate the yoke (aileron control) and observe the output of `js_demo`. If figures in the first group of colons (for device 0) change, assignment is correct. If figures in the second group of colons (for device 1) change, you have to make the yoke the preferred device first. For doing so, enter the Windows "Control panel", open "Game controllers" and select the "Advanced" button. Here you can select the yoke as the "Preferred" device. Afterward you can check that assignment by running `js_demo` again. The yoke should now control the first group of figures.

Unfortunately, we did not find a way to get the pedals to work, too, that way. Thus, in cases like this one (and others) you may want to try an alternative method of assigning joystick controls.

Some hints for Mac OS X users

Thanks to Mac OS X, most of HID compatible joysticks are recognized and usable on Mac OS X. Some of joysticks are already defined properly in FlightGear. However, sometimes your joystick does not work as you expect due to some reasons such as missing joystick ID, and misconfigured buttons or axis. In such cases, you need to modify a joystick configuration file.

The basic procedures in configuring joysticks under Mac OS X are the same as those of Linux as described above. The main differences are the path to the joystick configuration files and the means of finding a joystick name. The joystick configuration files are at `Contents/Resources/data/Input/Joysticks` under the FlightGear.app folder. You can open the data folder by choosing "Advanced Features » Others » Open data folder" on the GUI launcher. You can also open it by right-clicking the FlightGear icon at Applications folder and choose "Show Package Contents" to access the folders inside the application. To find a joystick name, follow the steps below :

1. Open *System Profiler* by choosing *Apple Menu | About this Mac | More Info...*
2. Select USB from the *Hardware* list in the left pane.
3. Find the name of your joystick (e.g. Logitech Extreme 3D) from the USB Device Tree.
4. Add a name tag with the recognized joystick name to a proper configuration XML file (e.g. `<name>Logitech Extreme 3D</name>`)
5. Check if your device name is shown properly by choosing *Help | Joystick information* from the menu on the *FlightGear* simulator window.

In case your joystick doesn't work as expected even it is recognized properly, you need to modify the button and/or axis settings. Some Mac users reported that

twisting a joystick controls the throttle instead of the rudder. This is a typical joystick misconfiguration on Mac OS X. To fix this issue, you need to edit two axis tags in a joystick configuration file.

In many cases, the rudder is assigned to axis 2 and the throttle to axis 3 in driver level. Editing or adding a mac tag inside an axis tag for the rudder (and throttle) will fix this problem. The example below is a part of a joystick configuration file that assigns axis 3 for rudder control.

```
<axis>
  <desc>Rudder</desc>
  <number>
    <linux>2</linux>
    <mac>3</mac> <!-- This must be 2 -->
    <windows>3</windows>
  </number>
  . . . . .
```

4.6.2 Joystick support via .fgsrc entries

Fortunately, there is now a utility available which makes things easier for the average, non-XML savvy user.

To configure your joystick using this approach do the following :

1. Open a command shell. Under Windows this is known as a Command prompt and can be found under Start -> All programs -> Accessories
2. Change to the directory /FlightGear/bin (e.g. cd c:\FlightGear\bin).
3. Run the fgjs tool.

The program will tell you which joysticks, if any, were detected. Now follow the commands given on screen, i.e. move the axis and press the buttons as required. Be careful, a minor touch already “counts” as a movement. Check the reports on screen. If you feel something went wrong, just re-start the program.

After you are done with all the axes and switches, the directory above will hold a file called fgsrc.js. The contents on this file can be copied into your personal preferences file (.fgsrc/system.fgsrc). See 4.5 for details of how to create this file.

Note that output of fgjs is UNIX formatted. As a result, Windows Editor may not display it properly. It may be helpful to use an editor capable of handling UNIX files as well. My favorite freeware file editor for that purpose, although somewhat dated, is PFE, which can be obtained from <http://www.lancs.ac.uk/staff/steveb/cpaap/pfe/>.

The axis/button assignment of fgjs should at least get the axis assignments right, but the output may need some tweaking. For example, some axes may be moving the opposite way they should, or the dead zones may be too small. For instance, the elevator axes on my USB CH Yoke was reversed so I had to change

```
--prop:/input/joysticks/js[1]/axis[1]/binding/factor=-1.0
```

into

```
--prop:/input/joysticks/js[1]/axis[1]/binding/factor=1.0
```

To help with tuning, here is a short introduction into the assignments of joystick properties. Basically, all axes settings are specified via lines having the following structure :

```
--prop:/input/joysticks/js[n]/axis[m]/binding/command=property-scale
--prop:/input/joysticks/js[n]/axis[m]/binding/property=/controls/steering option
--prop:/input/joysticks/js[n]/axis[m]/binding/dead-band=db
--prop:/input/joysticks/js[n]/axis[m]/binding/offset=os
--prop:/input/joysticks/js[n]/axis[m]/binding/factor=fa
```

where

- n = number of device (usually starting with 0)
- m = number of axis (usually starting with 0)
- steering option* = elevator, aileron, rudder, throttle, mixture, pitch
- dead-band* = range within which signals are discarded.
Useful to avoid jittering for minor yoke movements
- offset* = Applies an offset after any *factor*.
Useful to handle devices where the neutral position does not map to 0.
- factor* = controls sensitivity of that axis ; defaults to +1,
with a value of -1 reversing the behavior

You should be able to at least get your joystick working along these lines. Concerning all the finer points, such as getting the joystick buttons working, John Check has written a very useful README included in the base package under `$FG_ROOT/Docs/README.Joystick.html`. In case of any trouble with your input device, it is highly recommended to read this document.

Chapitre 5

En vol : tout sur les instruments, les raccourcis clavier et les menus

Voici une description des principaux systèmes de contrôle du programme et de pilotage de l'avion. On suppose que le lecteur est déjà familiarisé avec le vol, peut-être grâce à de l'expérience acquise sur d'autres simulateurs. Si vous êtes un vrai débutant, les tutoriels de la section 7 sont plus appropriés pour vous apprendre à voler avec FlightGear.

Un petit livret, destiné à être imprimé, et contenant une liste des touches, est disponible à l'adresse :

<http://www.flightgear.org/Docs/FGShortRef.pdf>.

Une référence à la plupart des contrôles du clavier se trouve dans le menu Aide du simulateur.

5.1 Démarrer le moteur

Suivant le type d'aéronef, vous pouvez avoir à démarrer le(s) moteur(s) avant de pouvoir voler. Les instructions ci-dessous sont génériques. Consultez l'aide de l'aéronef ou les tutoriels de l'avion pour obtenir des instructions plus spécifiques.

Une fois le moteur démarré, vous devez vérifier si les freins de parking sont serrés. Si c'est le cas, lâchez-les en appuyant sur la touche 'B'.

5.1.1 Moteur à pistons

Pour les aéronefs disposant d'un moteur à pistons, les magnétos sont contrôlés par les touches '{' et '}'. Sur la plupart des aéronefs, le démarreur s'actionne en appuyant sur la touche 's'. Sur les aéronefs multi-moteurs, vous pouvez choisir quel moteur contrôler. Utilisez '^' pour sélectionner tous les moteurs à la fois. La plupart des magnétos ont 4 positions - OFF, LEFT, RIGHT et BOTH. Ainsi, pour

démarrer le moteur sélectionné, appuyez 3 fois sur la touche ‘}’, puis maintenez la touche ‘s’ enfoncée jusqu’au démarrage.

Notez que la procédure de démarrage est souvent plus complexe pour les puissants avions de chasse de la seconde guerre mondiale. Consultez l’aide de l’avion pour avoir plus de détails.

5.1.2 Aéronefs à turbopropulseurs

Le démarrage d’un moteur turbopropulsé ne nécessite généralement que de déplacer le levier de commande de *Off* à *Idle*, en appuyant sur la touche ‘m’.

5.1.3 Aéronefs à réaction

Le démarrage d’un aéronef à réaction est bien plus complexe, et les contrôles varient selon l’appareil.

1. Placer l’interrupteur général (*cutoff*) sur la position ON.
2. Actionner le démarreur (*starter*).
3. Lorsque les moteurs approchent des 5% N1, remettez le *cutoff* sur OFF.
4. Coupez le démarreur une fois que le moteur a atteint sa vitesse opérationnelle.

5.2 Contrôles clavier

Bien que les joysticks, les volants et les pédales de palonnier soient pris en charge, vous pouvez choisir de voler avec *FlightGear* avec l’usage du clavier, seul ou utilisé conjointement avec la souris, comme décrit ci-dessous. Mais, quelle que soit la façon dont vous contrôlez votre aéronef, vous serez obligé d’utiliser le clavier, ne serait-ce que pour quelques actions.

L’attribution de fonctions à des touches (ne sont pas codés en durs, mais ajustables par l’utilisateur. Vous pouvez contrôler et modifier ces paramètres via le fichier `keyboard.xml`, qui peut être trouvé dans le répertoire principal de *FlightGear*. C’est un fichier texte au format ASCII lisible par un humain. Bien que les modifications à ce fichier ne soient pas recommandées aux débutants, des utilisateurs plus aguerris peuvent trouver utile de modifier l’attribution de fonctions aux touches, au gré de leurs souhaits, par exemple pour les faire correspondre à ce qui peut être trouvé sur d’autres simulateurs.

5.2.1 Contrôles de l’aéronef

Pour parvenir à contrôler pleinement votre avion en utilisant votre clavier pendant la phase de vol, vous devriez vous assurer que NumLock est activé, et que la fenêtre *FlightGear* est active. Les touches suivantes assurent le contrôle les surfaces de base de l’aéronef.

Touche	Action
9 / 3	Commande des gaz
4 / 6	Aileron
8 / 2	Gouverne de profondeur
0 / Entrée	gouverne de direction
5	Centrage des ailerons/gouverne de profondeur/direction
7 / 1	Trim de la gouverne de profondeur trim

Tableau 1 : *Contrôles de base de l'aéronef*

Les touches suivantes contrôlent les moteurs.

Touche	Action
!	Sélectionne le premier moteur
@	Sélectionne le deuxième moteur
#	Sélectionne le troisième moteur
\$	Sélectionne le quatrième moteur
~	Sélectionne tous les moteurs
{	Décrémente le magnéto sur le moteur sélectionné
}	Incrémente le magnéto sur le moteur sélectionné
s	Démarrage du/des moteur(s) sélectionné(s)
M / m	Appauvrir/Enrichir le mélange du moteur sélectionné
N / n	Diminue/Augmente le nombre de tours par minute du moteur sélectionné

Tableau 2 : *Touches de contrôle du moteur*

Les touches suivantes contrôlent les systèmes secondaires de l'aéronef.

Touche	Action
b	Appliquer tous les freins
, / .	Appliquer le frein gauche/droit (utile pour le freinage différentiel)
l	Active le verrouillage de la roue de queue
B	Active le frein de parking
g/G	Monter/descendre le train d'atterrissage
Espace	Appuyez pour parler (Push To Talk, PTT)
- / _	Entrée/menu du clavardage clavier du mode multijoueurs
[/]	Rentre/déploie les volets
j / k	Rentre/déploie les aérofreins
Ctrl-B	Active les freins de vitesse

Tableau 3 : *Contrôles secondaires de l'aéronef*

Les contrôles du pilote automatique sont les suivants.

Touche	Action
Retour arrière	Active le pilote automatique
Ctrl-A	Active le verrouillage en altitude
Ctrl-G	Active le verrouillage sur la pente ILS (NAV 1)
Ctrl-H	Active le verrouillage du cap
Ctrl-N	Active le verrouillage du NAV 1
Ctrl-S	Active la manette des gaz automatique
Ctrl-T	Active le verrouillage en mode suivi de terrain (au-dessus du sol)
Ctrl-U	Ajoute 1000 piers à votre altitude (urgence)
F6	Active le mode cap du pilote automatique
F11	Dialogue altitude du pilote automatique
8 / 2	Ajustement de l'altitude
4 / 6	Ajustement du cap
9 / 3	Ajustement de la manette des gaz automatique

Tableau 4 : *Contrôles du pilote automatique*

5.2.2 Contrôles du simulateur

Pour changer la direction de la vue, vous devez désactiver NumLock. Les contrôles disponibles sont les suivants :

Touche pavé numérique	Angle de vue
Shift-8	Vers l'avant
Shift-7	Avant/Gauche
Shift-4	Gauche
Shift-1	Arrière/Gauche
Shift-2	Arrière
Shift-3	Arrière/Droit
Shift-6	Droit
Shift-9	Avant/Droit

Tableau 5 : *Directions de la vue*

De plus, les touches suivantes vous permettent de contrôler l'affichage :

Touche	Action
P	Active/désactive le tableau de bord
c	Alterne cockpit 3D/2D (si les deux sont disponibles)
S	Change le style de tableau de bord (complet/simple)
Ctrl-C	Active la visibilité des zones cliquables du panneau/cockpit
h	Active le HUD (collimateur tête haute)
H	Change la luminosité du HUD
i / I	Minimise/augmente le HUD
x / X	Zoom +/-
v / V	Alterne les modes de vue dans un sens ou l'autre
Ctrl-V	Réinitialise les modes de vue vers le mode pilot
z / Z	Augmente/diminue la visibilité (brouillard)
F10	Active/désactive le menu

Tableau 6 : Options d'affichage

Enfin, les contrôles généraux suivants du simulateur sont disponibles.

Touche	Action
b	Appliquer tous les freins
, / .	Appliquer le frein gauche/droit (utile pour le freinage différentiel)
l	Active le verrouillage de la roue de queue
B	Active le frein de parking
g/G	Monter/descendre le train d'atterrissage
Espace	Appuyez pour parler (Push To Talk, PTT)
- / _	Entrée/menu du clavardage clavier du mode multijoueurs
[/]	Rentre/déploie les volets
j / k	Rentre/déploie les aérofreins
Ctrl-B	Active les freins de vitesse

Tableau 7 : Contrôles généraux du simulateur.

5.2.3 Contrôles du pilote automatique

FlightGear prend en charge deux types de pilote automatique - un pilote automatique générique, qui fonctionne avec tous les aéronefs (même ceux qui, en temps normal, n'auraient pas de pilote automatique), et des pilotes automatiques spécifiques à un aéronef en particulier et qui sont contrôlés à partir du cockpit.

Le pilote automatique générique est contrôlé via les touches suivantes :

Touche	Action
Retour arrière	Active le pilote automatique
Ctrl-A	Active le verrouillage en altitude
Ctrl-G	Active le verrouillage sur la pente ILS (NAV 1)
Ctrl-H	Active le verrouillage du cap
Ctrl-N	Active le verrouillage du NAV 1
Ctrl-S	Active la manette des gaz automatique
Ctrl-T	Active le verrouillage en mode suivi de terrain (au-dessus du sol)
Ctrl-U	Ajoute 1000 piers à votre altitude (urgence)
F6	Active le mode cap du pilote automatique
F11	Dialogue altitude du pilote automatique
8 / 2	Ajustement de l'altitude
4 / 6	Ajustement du cap
9 / 3	Ajustement de la manette des gaz automatique

Tableau 6 : Contrôles du pilote automatique.

Ctrl + T est particulièrement intéressant car il va faire se comporter votre aéronef comme un missile de croisière, en effectuant un suivi de terrain. Ctrl + U peut être utile si vous sentez que vous n'allez pas tader à vous écraser.

Quand le pilote automatique est activé, certaines des touches du pavé numérique fonctionnent différemment et ajustent le pilote automatique au lieu des commandes de vol à proprement parler :

Touche	Action
p	Mettre le simulateur en pause
a / A	Accélérer/ralentir la vitesse de simulation
t / T	Accélérer/ralentir la vitesse de l'horloge
Ctrl-R	Ralenti instantané
F3	Sauvegarder la capture d'écran
ESC	Quitter le programme

Tab. 7 : Contrôles additionnels du pilote automatique.

Notez que certains claviers utilisent “.” au lieu de “,”.

5.3 Actions contrôlées par la souris

De même qu'elle permet de sélectionner des éléments du menu ou de cliquer sur des commandes au sein du cockpit, votre souris peut également être utilisée dans *FlightGear* à de nombreuses fins utiles.

Il y a trois modes souris : Normal (par défaut), Contrôle et Vue. Vous pouvez basculer de l'un à l'autre en utilisant le bouton droit de la souris.

5.3.1 Mode normal

En mode Normal, vous pouvez contrôler les menus et les commandes du tableau de bord. L'activation de ce mode est indiquée par un curseur disposant d'une

flèche normale.

Pour actionner un commutateur ou un interrupteur, cliquez simplement sur celui-ci avec le bouton gauche ou milieu de la souris.

Pour faire tourner un bouton sur une radio ou une commande linéaire comme la manette des gaz, cliquez sur le côté gauche de celui-ci pour faire diminuer sa valeur, et sur le côté droit pour l'augmenter. Cliquez avec le bouton gauche de la souris pour réaliser de petits ajustements, avec le bouton droit pour des ajustements plus importants. Certaines commandes, comme les radios, prennent également en charge la molette de la souris.

Un appui sur Ctrl-C illumine les zones et les objets cliquables.

5.3.2 Mode contrôle

En mode Contrôle, vous pouvez contrôler les commandes de vol de l'aéronef en bougeant la souris. L'activation de ce mode est indiquée par la présence d'un curseur de souris en forme de croix.

Dans ce mode, le déplacement de la souris vers la droite ou vers la gauche contrôle les ailerons et le roulis de l'avion. La déplacer vers l'avant ou vers l'arrière contrôle la gouverne de profondeur et change l'inclinaison de l'avion.

Maintenir le bouton gauche appuyé tout en bougeant la souris vers la droite ou la gauche contrôle le palonnier. Tenir appuyé le bouton du milieu tout en bougeant vers l'arrière ou l'avant la souris permet de contrôler la manette des gaz.

Enfin, la molette peut être utilisée pour paramétrer le trim.

Ce mode est particulièrement pratique si vous n'avez pas de joystick. On a ainsi un bien meilleur contrôle de l'aéronef qu'avec le clavier. Si vous avez l'intention de piloter régulièrement votre avion avec la souris, il vous est recommandé d'activer l'auto-coordination, qui solidarise ailerons et palonnier. Ceci peut être réalisé en activant l'option `--enable-auto-coordination`, ou en choisissant auto-coordination à partir de l'assistant.

5.3.3 Mode vue

En mode Vue, vous pouvez regarder autour de vous en utilisant la souris. L'activation de ce mode est indiquée par la présence d'un curseur de souris à double flèche.

Bouger simplement la souris vous permet de visualiser l'espace autour votre position actuelle. Ceci est particulièrement utile pour regarder autour du cockpit ou par une fenêtre latérale. On peut utiliser la molette pour zoomer ou dé-zoomer. Cliquer sur le bouton gauche de la souris repositionne la vue à sa position initiale, en général droit devant.

En maintenant appuyé le bouton du milieu de la souris tout en bougeant cette dernière vous permet de faire varier le point de vue vers la gauche/la droite et vers le haut/le bas. En bougeant la souris tout en maintenant à la fois le bouton du milieu

et la touche Ctrl enfoncés vous permet de faire bouger le point de vue vers l'avant et vers l'arrière.

5.4 Entrées du menu

La barre de menu permet d'accéder à différentes options du simulateur et de l'aéronef. La plupart des avions ont leurs propres entrées de menus, pour changer leur immatriculation ou démarrer automatiquement leurs moteurs. Ces éléments peuvent être trouvés tout à la fin de la barre de menu. Pour afficher ou cacher la barre de menu, appuyez sur F10. Vous pouvez aussi afficher le menu automatiquement en actionnant votre souris en haut de l'écran.

La barre de menu propose les menus et options suivantes.

– Fichier

- **Réinitialiser (Shift-Esc)** réinitialise le vol à la position de démarrage sélectionnée. Très pratique si jamais vous êtes perdu ou que quelque chose tourne mal.
- **Capture d'écran (F3)** sauvegarde une capture d'écran comme `fgfs-screen-XXX.jpg`.
- **Répertoire des captures** vous permet de préciser le répertoire où les captures d'écran seront sauvegardées.
- **Configuration du son** permet de configurer le volume pour différents canaux sonores, et s'ils doivent être entendus en-dehors de l'aéronef.
- **Quitter (Esc)** quitte le programme.

– Affichage

- **Options d'affichage** permet de configurer diverses options d'affichage, notamment si le tableau de bord 2D, le taux de rafraîchissement d'images et les messages de clavardage sont affichés.
- **Options de rendu** configure diverses fonctionnalités graphiques. Ceci vous permet de gérer le compromis entre les effets visuels sympathiques, comme les ombres, les nuages 3D et les réflexions spéculaires et le taux de rafraîchissement des images. Pour vous permettre d'arriver à un compromis acceptable, l'activation de l'option "Afficher le taux de rafraîchissement" option dans le menu Options d'affichage, vous permettra de visualiser le taux d'images par seconde (*frames per second, fps*) actuel dans l'angle inférieur droit de l'écran. La plupart des personnes trouvent qu'un taux d'environ 20 fps est suffisant pour voler. Ce taux est dépendant des fonctionnalités graphiques que vous avez activé, la visibilité du moment (paramétré par *Z/z*), le nombre d'objets visibles et leur niveau de détails (*Level Of Detail, LOD*).
- **Options de la vue** configure les vues activées.
- **Options d'affichage du cockpit** configure la vue au sein du cockpit, le mouvement de la tête du pilote, le voile noir dû à la prise importante de *G* positifs et le voile rouge dû à la prise important de *G* négatifs.
- **Ajuster le niveau de détail** fixe la distance jusqu'à laquelle différents

- niveaux de détails sont affichés. Ceci affecte les textures et objets affichés par le simulateur.
- **Ajuster la position de la vue** Affiche le décalage de vue actuel. Vous pouvez ajuster cette valeur en déplaçant les contrôles. Sinon, vous pouvez réaliser de petits ajustements de votre point de vue en utilisant la souris (voir ci-dessus).
 - **Ajuster les propriétés du HUD** configure les paramètres du collimateur tête haute (Head Up Display (HUD)), comme la transparence ou l'anti-crénelage.
 - **(Dés)Activer le tunnel ILS** (dés)active un tunnel virtuel ou vous guider jusqu'à la piste sur un plan d'approche normal. Utile si vous avez des difficultés à trouver le bon plan d'approche pour atterrir.
 - **Rejeu instantané (Ctrl-R)** contrôle la fonctionnalité de rejeu instantané - un bon outil pour vérifier la qualité de vos atterrissages !
 - **Options de vue stéréoscopique** configure l'affichage stéréoscopique (en relief 3D), en utilisant des lunettes Rouge/Vert ou d'autres méthodes d'affichage.
 - **Position**
 - **Positionner l'aéronef au sol** positionne l'aéronef sur la piste de n'importe quel aéroport installé. Vous avez besoin de connaître le code OACI de l'aéroport à partir duquel vous souhaitez démarrer (par exemple KSFO pour San Francisco International).
 - **Positionner l'aéronef en vol** positionne l'aéronef à un point arbitraire en l'air. Vous devez sélectionner un point au sol connu, par exemple un aéroport, un VOR, des coordonnées longitude/latitude et une position relative par rapport à ce point, par exemple une distance, direction, altitude. Vous pouvez également configurer votre vitesse et cap initiaux. Ceci s'avère utile pour s'entraîner aux approches.
 - **Choisir un aéroport depuis la liste** positionne l'aéronef sur un aéroport. Vous pouvez choisir parmi tous les aéroports que vous avez installé. Cliquer sur Apply vous positionnera sur cet aéroport sur la piste en service en fonction de la direction du vent.
 - **Attitude aléatoire** Place l'aéronef avec une attitude, un cap et une vitesse aléatoires. Utile pour s'entraîner à la récupération depuis des attitudes inhabituelles.
 - **Position de la tour** configure la tour de l'aéroport utilisée pour les vues de la tour et 'regarder à partir de la vue de la tour'.
 - **Pilote automatique** Ce menu est uniquement disponible pour les aéronefs qui disposent du pilote automatique par défaut configuré. Certains aéronefs peuvent avoir leur propre pilote automatique, qui peut être configuré à partir du tableau de bord : dans ce cas, ce menu est désactivé.
 - **Paramètres** configure le pilote automatique de l'aéronef. Vous pouvez paramétrer le pilote automatique de différentes manières - du simple maintien du niveau des ailes au suivi d'un ILS.

- **Gestionnaire de routes** prépare la feuille de route (points de cheminement) pour le pilote automatique. Les points de cheminement peuvent être des aéroports ou des points fixes. Les cap, distance et temps restant vers le point de cheminement actuel peuvent être affichés dans le HUD.
- **Point de cheminement précédent** sélectionne le point de cheminement précédent à partir de la feuille de route.
- **Point de cheminement suivant** sélectionne le point de cheminement suivant à partir de la feuille de route.
- **Environnement**
 - **Météo** vous permet de paramétrer la météo actuelle, choisir un scénario météo ou utiliser la météo telle qu'elle est diffusée par la station météo la plus proche (généralement un aéroport) via les METAR. Vous pouvez choisir la météo simple, qui paramètre explicitement les conditions, ou la météo avancée, qui simule un éventail plus large de systèmes météorologiques, au détriment de la précision des conditions locales.
 - **Paramètres du temps** vous permet de définir l'heure actuelle dans le simulateur, d'accélérer la simulation, et de changer la vitesse à laquelle l'heure défile dans le simulateur. Affiche l'heure locale et l'heure UTC.
 - **Paramètres de feu de forêt** définit si le crash d'un avion au sol provoque un feu de forêt réaliste, qui peut s'étendre et peut être éteint à l'aide d'un aéronef bombardier d'eau approprié.
 - **Téléchargement de scènes** configure le téléchargement automatique des scènes via Internet en utilisant la fonctionnalité *TerraSync*.
- **Equipement**
 - **Carte** affiche une carte mobile qui affiche les aéroports, les balises de radionavigation, etc.
 - **Chronomètre** affiche un chronomètre basique, utile pour les approches aux instruments.
 - **Carburant et chargement** vous permet de paramétrer les niveaux de carburant et de chargement au sein de l'aéronef. Uniquement disponible sur certains aéronefs.
 - **Paramètres radio (F12)** fixe les fréquences et radiales utilisées par les radios et équipements de navigation.
 - **Paramètres GPS** configure les points de cheminement et visualise les informations de route pour le GPS.
 - **Paramètres des instruments** vous permet de caler la pression de l'altimètre et l'indicateur de cap.
 - **Pannes aléatoires** configure des pannes aléatoires de différents systèmes et instruments de l'aéronef, en précisant le temps moyen entre pannes (*Mean Time Between Failure, MTBF*) ou les cycles moyens entre pannes (*Mean Cycles Between Failure, MCBF*).
 - **Pannes système** configure des pannes aléatoires des systèmes de l'aéronef, comme la pompe à vide.
 - **Pannes des instruments** configure la panne d'un instrument spécifique

de l'aéronef.

- **IA**
 - **Paramètres de trafic et de scénario** configure les scénarios IA actifs. Notez que cette configuration ne prendra effet que lors du prochain redémarrage du simulateur.
 - **Contrôle des ailiers** vous permet, suivant les aéronefs, de contrôler vos ailiers IA.
 - **Contrôle du ravitailleur** vous permet de créer dynamiquement un avion de ravitaillement en vol, si votre aéronef le prend en charge. Voyez la section 6.8 pour de plus amples détails.
 - **Contrôle du porte-avions** vous permet de contrôler un porte-avions IA.
 - **Paramètres des passerelles d'embarquement** vous permet de contrôler les passerelles d'embarquement sur certains aéroports.
- **Multijoueurs**
 - **Paramètres** vous permet de configurer le mode multijoueurs en paramétrant un indicatif et un serveur.
 - **Fenêtre de clavardage** vous permet de discuter avec un autre aéronef dans l'environnement multijoueurs.
 - **Menu de clavardage (-)** vous permet d'envoyer des messages de discussion standards avec d'autres avions ou un organisme de contrôle aérien dans l'environnement multijoueurs. Certains menus contiennent des sous-menus d'options.
 - **Liste des pilotes** affiche une liste des autres pilotes multijoueurs à portée, avec leur distance, relèvement et altitude.
 - **Choix du porte-avions multijoueurs** affiche une liste des porte-avions multijoueurs disponibles.
- **Débogage** : ce menu contient différentes options qui ne sont pas l'objet de ce guide.
- **Aide**
 - **Aide (s'ouvre dans le navigateur)** ouvre le système d'aide dans une fenêtre de navigateur.
 - **Aide de l'aéronef** affiche des informations spécifiques à l'aéronef.
 - **Touches communes aux aéronefs** liste les touches de base pour contrôler l'aéronef.
 - **Touches de base du simulateur** liste les touches de base pour contrôler le simulateur.
 - **Informations sur le joystick** affiche des informations sur tout joystick en fonction, y compris des informations sur l'attribution des axes et des boutons.
 - **Tutoriels** vous permet de démarrer un tutoriel au sein du simulateur pour l'avion actuel. Cette option n'est disponible que pour certains aéronefs. Voir Tutoriels ci-dessous pour plus de détails.
 - **A props de FlightGear** affiche des informations sur cette version de FlightGear.

5.5 Le tableau de bord



Fig. 6 : Le cockpit 3D d'un Cessna 172.

Les aéronefs au sein de *FlightGear* peuvent avoir à la fois un tableau de bord à deux dimensions et un cockpit à trois dimensions. Le cockpit à trois dimensions offre une vue beaucoup plus réaliste, comme si on était à la place du pilote, mais la lecture du tableau de bord peut être difficile avec des moniteurs de dimensions réduites.

Le Cessna 172P par défaut (c172p) dispose à la fois d'un cockpit à deux et trois dimensions. Le cockpit à 3 dimensions est activé par défaut lorsque vous démarrez *FlightGear*, mais vous pouvez y substituer le tableau de bord à deux dimensions en choisissant Affichage->Options d'affichage->Show 2D Panel à partir du menu, ou en appuyant sur la touche "P".

Tous les boutons et leviers peuvent être actionnés avec la souris. Pour changer une commande, cliquez simplement avec le bouton gauche/milieu sur le levier/bouton en question. Pour les commandes qui ont une large gamme de positions, utilisez le bouton du milieu de la souris pour des ajustements plus grands. En général, un clic sur le côté droit de la commande augmentera sa valeur, alors qu'un clic sur le côté gauche la diminuera.

Certains instruments (en particulier les radios) supportent également l'utilisation de la molette de la souris pour l'ajustement de leurs valeurs.

5.6 Le collimateur tête haute (*Head Up Display, HUD*)

FlightGear fournit également un collimateur tête haute ou HUD (*Head Up Display*) HUD . On trouve généralement les HUD sur les aéronefs militaires et sur quelques avions à réaction modernes. Cependant, *FlightGear* vous permet également d'utiliser le HUD sur de nombreux aéronefs de l'aviation générale. Pour activer le HUD, appuyez sur 'h'.

Le HUD, présenté à la Fig. 7 affiche tous les principaux paramètres de vol de l'aéronef. Au centre, vous trouvez l'indicateur d'assiette (en degrés) avec l'indicateur des ailerons au-dessus et l'indicateur de la gouverne de direction dessous. Une échelle correspondante pour la peut être trouvé à gauche de l'indicateur d'échelle avec un indicateur de *trim*. En bas se trouve un simple indicateur de taux de virage.

Il y a deux échelles à l'extrême gauche : celle qui est à l'intérieur indique la vitesse (en noeuds) alors que celle à l'extérieur indique la position de la manette des gaz. Les deux échelles à l'extrême droite affichent votre altitude - celle de gauche indique l'altitude au-dessus du sol, alors que celle de droite indique l'altitude au-dessus du niveau de la mer, les deux étant indiquées en pieds.

Enfin, le HUD indique deux informations complémentaires. Dans le coin supérieur gauche vous aurez la date et l'heure, ainsi que votre position actuelle en latitude et longitude.

Vous pouvez modifier la couleur du HUD en utilisant les touches "H" ou "h". Un appui sur la touche "i/I" minimise/agrandit le HUD.

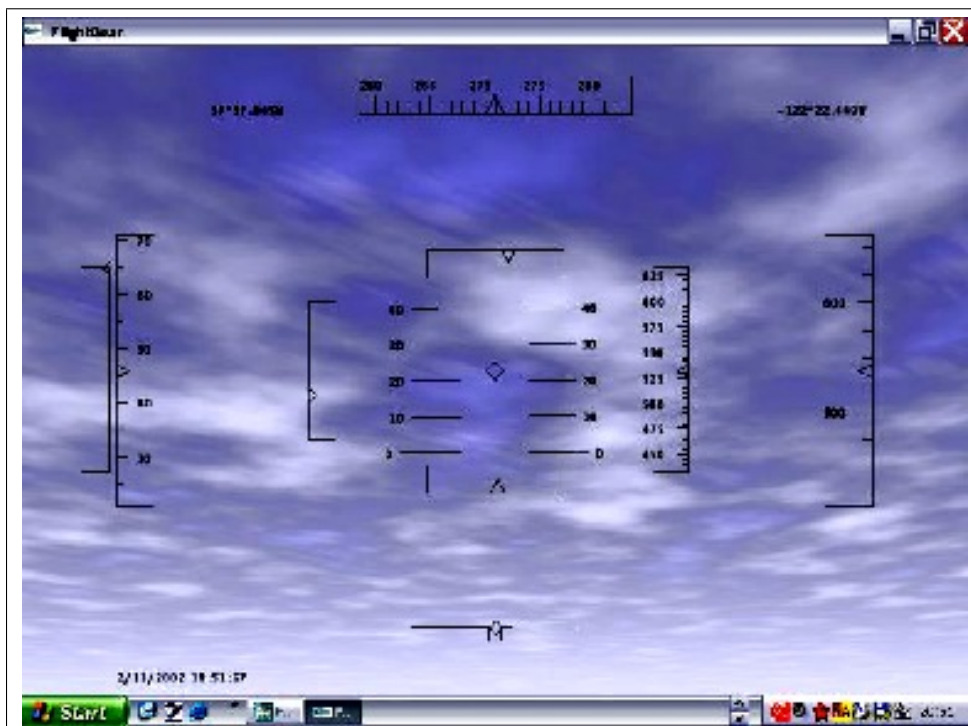


Fig. 7 : Le collimateur tête haute, ou HUD (*Head Up Display*).

Chapitre 6

Features

FlightGear contains many special features, some of which are not obvious to the new user. This section describes how to enable and make use of some of the more advanced features.

Many of the features are under constant development, so the information here may not be completely up-to-date. For the very latest information (and new features), see the *FlightGear* Wiki, available from <http://wiki.flightgear.org/>

6.1 Multiplayer

FlightGear supports a multiplayer environment, allowing you to share the air with other flight-simmers. For server details and to see who is online (and where they are flying), have a look at the excellent multiplayer map, available from <http://mpmap02.flightgear.org>

Click on the 'server' tab to see a list of multiplayer servers.

6.1.1 Quick Start

You can connect to the MP environment from the MP Settings dialog under the Multiplayer menu. Simply select the server closest to you from the list, enter a callsign (which will be seen by other players), and click "Connect".

To see a list of other pilots are in the area, select Pilot List from the Multiplayer menu.

All the standard MP servers are interconnected, so there is no need to be on the same server as people you are flying with.

6.1.2 Other Methods

If you are connecting to a non-standard server, or the above method does not work, you can also connect using the methods below.

Using the FlightGear Launcher

The final screen of the FlightGear Launcher has a section for Multiplayer. Simply select the checkbox, enter the hostname and port number you noted above and choose a callsign to identify yourself. Your callsign can be up to 7 characters in length. You must also check The AI models checkbox under Features to make other aircraft visible.

Using the Command Line

The basic arguments to pass to fgfs for multiplayer are these :

```
--multiplay=out,10,<server>,<portnumber>  
--multiplay=in,10,<client>,<portnumber>  
--callsign=<anything>  
--enable-ai-models
```

Where

1. <portnumber> is the port number of the server e.g. 5000.
2. <server> is the name of the multiplayer server e.g. mpserver01.flightgear.org.
3. <client> is the name of your computer, or the IP address ip address of the network interface being used by FG to connect to the server - even if that's a local 192.168 type address. e.g. 192.168.0.1
4. <callsign> is the call sign to identify yourself, up to 7 characters, e.g. N-FGFS.

Once the simulator has loaded, you should see yourself displayed on the map. If you don't, check the console for error messages and see the Troubleshooting section below.

6.1.3 Troubleshooting

To get multiplayer to work, we need information about the IP address of our computer and the ability to communicate with the server. How to get this information depends on your configuration and is described below.

Those using a USB modem to connect to the Internet

First of all, you need to know the IP address of the network interface you're going to be running FG multiplayer over. If your Internet connection is via an ADSL modem that plugs directly into your computer with a USB connection, you should be able to find your IP address by visiting <http://www.whatismyip.com> . Please note that this address may very well change every now and again - if MP stops working, check this first.

Those using some kind of Ethernet router to connect to the Internet

Otherwise, your connection is likely via some kind of router that connects to your computer via an RJ-45, or "Ethernet" connector (similar shape to most Western telephone plugs), or by a wireless link. You need to find the IP address of that network interface.

Under Linux, this can be found by logging in as root and typing "ifconfig". You may find more than one interface listed, beginning with "lo" - ignore that one. You should have something like "eth0" or "wlan0" also listed - look through this block of text for "inet addr". This will be followed directly by the number you're looking for, e.g. "inet addr :192.168.0.150"

Under Windows XP, click start, run, and type "cmd". In the terminal window which appears, type "ipconfig" This should show you your IP address - note it down.

With Windows 98, click start, run, and type "winipcfg" to get information about your IP address.

If It Still Doesn't Work

You MUST give your local, behind-the-router IP address for MultiPlayer to work. Trust me on this one !

You should check that your firewall is not causing problems - either turn it off temporarily or add an exception to allow incoming connections on port 5000.

If it's still just not working for you, ask nicely on the FlightGear IRC channel and someone should be able to assist.

6.2 Aircraft Carrier

FlightGear supports carrier operations on the Nimitz, (located near San Francisco), Vinson, San Antonio, Foch, and Eisenhower. The carriers are equipped with working catapult, arrestor wires, elevators, TACAN and FLOLS.

To enable the carrier, you must edit your preferences.xml file in \$FG_ROOT using a text editor (e.g. Notepad under Windows). Search for the word "nimitz". You ought to find something that looks like this ;

```
<!--<scenario>nimitz_demo</scenario>-->
```

You should remove the "comment" marks so that it looks like this ;

```
<scenario>nimitz_demo</scenario>
```

Also ensure that the line above that referring to ai being enabled is set to "true"
Save the file and quit the text editor.

6.2.1 Starting on the Carrier

You are now ready to start FlightGear. To position your aircraft on the carrier at startup, use the following command line options :

```
--carrier=Nimitz --aircraft=seahawk
```

Please note the uppercase “N” in “Nimitz”.

If you are using the Windows or OS X launcher to run FG, you should find a text entry box in the gui that allows you to specify command line options, add the above options there.

Note that several FG aircraft are carrier capable, but the seahawk is possibly the easiest to fly to begin with.

6.2.2 Launching from the Catapult

Once FlightGear has started, you should ensure that the parking brakes are off and press and hold “L” to engage the launchbar. You must hold down “L” until the launch bar has engaged. You should notice the aircraft being pulled into alignment with the catapult and see the strops appear and hold down the aircraft. This will only happen if your aircraft is close enough to the correct spot on the catapult ; as a rough guide, for the default parking position the seahawk’s nose should be roughly level with the deck observation bubble.

To get the carrier into as good a position as possible for launch, select the “ATC/AI” menu, then check the “Turn into wind” box under the “AI Carrier” section. You should now notice the carrier begin to pick up speed and turn into the wind, and naturally the deck may tilt somewhat as it turns. You should wait for this maneuver to finish and the deck to return to level before moving on to the next stage.

Being attached to the catapult, you should spool up the engines to full power, ensure the brakes are off and that all flight controls are in a suitable position for launch (stick held right back with the seahawk.) When ready, press “C” to release the catapult. Your aircraft will be hurled forward off the deck, and you should be able to raise the undercarriage and climb slowly away, being careful to avoid stalling.

6.2.3 Finding the Carrier - TACAN

Actually finding the carrier in a vast expanse of open water can be very difficult, especially if visibility is poor. To assist with this task, Nimitz is equipped with TACAN, which allows a suitably-equipped aircraft (including the Seahawk) to obtain a range and bearing to the carrier. First, you must set the appropriate TACAN channel, 029Y in this case, in the radios dialogue (ctrl-r or choose Equipment/Radio Settings from the FG menubar). You should, if within range, notice the DME instrument show your distance from the carrier, and the ADF instrument

(next to the DME in the seahawk) should indicate a bearing to the carrier. Turn to the indicated heading and you should see the DME dial indicate your closing in on the carrier.

6.2.4 Landing on the Carrier

This is the most difficult part of the operation, as in real life. You might well find Andy Ross' tutorial on operating the A4 Skyhawk useful. It is available from here :

http://wiki.flightgear.org/A-4F_Skyhawk_Operations_Manual

Once you have used the TACAN to locate the carrier, you must line up with the rear of the flight deck. As this part of the deck is at an angle to the course of the vessel, you may need to correct your alignment often. Ensure that the aircraft is in the correct configuration for approach (the Help/Aircraft Help menu should contain useful data for your aircraft) and that the gear and the arrestor hook are down.

As you approach you should see, on the left hand side of the deck, a set of brightly coloured lights - called the Fresnel Lens Optical landing System (FLOLS). This indicates your position on the landing glideslope. You will see a horizontal row of green lights, and when approximately on the glideslope, an orange light (known in some circles as the "meatball") approximately in line with the green lights. When approaching correctly, the meatball appears in line with the green lights. If you are high it is above, and when low it is below. If you are very low the meatball turns red. If you fly to keep the meatball aligned you should catch number 3 wire.

Carrier landings are often described as "controlled crashes" and you shouldn't waste your time attempting to flare and place the aircraft gently on the deck like you would with a conventional landing - ensuring that you catch the wires is the main thing.

Immediately your wheels touch the deck, you should open the throttles to full power, in case you have missed the wires and need to "go around" again ; the wires will hold the aircraft if you have caught them, even at full power.

If you wish, you can then raise the elevators from the ATC/AI menu, taxi onto one of the elevators, lower it (uncheck the box on the menu) and taxi off into the hangar.

Don't be discouraged if you don't succeed at first - it's not an easy maneuver to master. If after a little practice you find the Seahawk too easy, you could move on to the Seafire for more of a challenge !

6.3 Atlas

Atlas is a "moving map" application for FlightGear. It displays the aircraft in relation to the terrain below, along with airports, navigation aids and radio frequen-

cies.

Further details can be found on the Atlas website :

<http://atlas.sourceforge.net>

6.4 Multiple Displays

FlightGear supports multiple displays. Using some straightforward XML, you can configure multiple "slave cameras" that are offset from the main view, so you can use multiple monitors to display a single view of the simulator. For example, you can have one display showing the view straight ahead, while two additional displays show the view to either side.

Information on configuring multiple displays can be found in the README.multiscreen file in the docs directory of your *FlightGear* installation.

6.5 Multiple Computer

FlightGear allows you to connect multiple instances of the program using the very flexible I/O subsystem, and display completely different views and controls on different computers. This can be used in combination with the Multiple Display support to create a more sophisticated environment with separate cockpit panel displays and even a separate control station allowing an instructor to fail instruments, change the weather etc.

An example of this is the 747 cockpit project.

<http://www.flightgear.org/Projects/747-JW/>

6.5.1 Setup

Each instance of *FlightGear* can support a single display. Due to the complexity of the FDM and graphics, *FlightGear* is very processor-intensive, so running multiple instances of *FlightGear* on a single machine is not recommended.

You will therefore need a computer for each view of the simulation you wish to display, including the panel. The computers obviously must be networked and for simplicity should be on the same subnet.

One computer is designated as the master. This computer will run the FDM and be connected to controls such as yokes, joysticks and pedals. As the machine is running the FDM, it usually only displays a simple view, typically the main panel, to maximize performance.

All other computers are designated as slaves. They are purely used for display purposes and receive FDM information from the master.

6.5.2 Basic Configuration

Creating a basic configuration for multiple displays is straightforward. The master computer needs to broadcast the FDM and control information to the slaves. This is done using the following command line options :

```
--native-fdm=socket,out,60,,5505,udp
--native-ctrls=socket,out,60,,5506,udp
```

The slave computers need to listen for the information, and also need to have their own FDMs switched off :

```
--native-fdm=socket,in,60,,5505,udp
--native-ctrls=socket,in,60,,5506,udp
--fdm=null
```

6.5.3 Advanced Configuration

The options listed above will simply show the same view on both machines. You will probably also want to set the following command-line options on both master and slave computers.

```
--enable-game-mode (full screen for glut systems)
--enable-full-screen (full screen for sdl or windows)
--prop:/sim/menubar/visibility=false (hide menu bar)
--prop:/sim/ai/enabled=false (disable AI ATC)
--prop:/sim/ai-traffic/enabled=false (disable AI planes)
--prop:/sim/rendering/bump-mapping=false
```

If using the master computer to display a panel only, you may wish to create a full-screen panel for the aircraft you wish to fly (one is already available for the Cessna 172), and use the following options.

```
--prop:/sim/rendering/draw-otw=false (only render the panel)
--enable-panel
```

6.6 Recording and Playback

As well as the Instant Replay feature within the simulator, you can record your flight for later analysis or replay using the I/O system. Technical details of how to record specific FDM information can be found in the `$FG_ROOT/protocol/README.protocol` file.

To record a flight, use the following command line options :

```
--generic=file,out,20,flight.out,playback
```

This will record the FDM state at 20Hz (20 times per second), using the playback protocol and write it to a file `flight.out`.

To play it back later, use the following command line options :

```
--generic=file,in,20,flight.out,playback
--fdm=external
```

The `playback.xml` protocol file does not include information such as plane type, time of day, so you should use the same set of command line options as you did when recording.

6.7 Text to Speech with Festival

FlightGear supports Text To Speech (TTS) for ATC and tutorial messages through the festival TTS engine (<http://www.cstr.ed.ac.uk/projects/festival/>). This is available on many Linux distros, and can also be installed easily on a Cygwin Windows system. At time of writing, support on other platforms is unknown.

6.7.1 Installing the Festival system

1. Install festival from <http://www.cstr.ed.ac.uk/projects/festival/>
2. Check if Festival works. Festival provides a direct console interface. Only the relevant lines are shown here. Note the parentheses !

```
$ festival
festival> (SayText "FlightGear")
festival> (quit)
```

3. Check if MBROLA is installed, or download it from here :
<http://tcts.fpms.ac.be/synthesis/mbrola/>

See under "Downloads" in "MBROLA binary and voices" (link at the bottom ; hard to find). Choose the binary for your platform. Unfortunately, there's no source code available. If you don't like that, then you can skip the whole MBROLA setup. But then you can't use the more realistic voices. See below for details of more voices. Run MBROLA and marvel at the help screen. That's just to check if it's in the path and executable.

```
$ mbrola -h
```

6.7.2 Running FlightGear with Voice Support

First start the festival server :

```
$ festival --server
```


Now, start *FlightGear* with voice support enabled. This is set through the `/sim/sound/voices/enabled` property. You can do this through the command line as follows.

```
$ fgfs --aircraft=j3cub \  
      --airport=KSQL \  
      --prop:/sim/sound/voices/enabled=true
```

Of course, you can put this option into your personal configuration file. This doesn't mean that you then always have to use *FlightGear* together with *Festival*. You'll just get a few error messages in the terminal window, but that's it. You cannot enable the voice subsystem when *FlightGear* is running.

To test it is all working, contact the KSFO ATC using the `'` key. You should hear "your" voice first (and see the text in yellow color on top of the screen), then you should hear ATC answer with a different voice (and see it in light-green color).

You can edit the voice parameters in the `preferences.xml` file, and select different screen colors and voice assignments in `$FG_ROOT/Nasal/voice.nas`. The messages aren't written to the respective `/sim/sound/voices/voice[*]/text` properties directly, but rather to aliases `/sim/sound/voices/atc,approach,ground,pilot,ai-plane`.

6.7.3 Troubleshooting

On some Linux distros, *festival* access is restricted, and you will get message like the following.

```
client(1) Tue Feb 21 13:29:46 2006 : \  
  rejected from localhost.localdomain  
not in access list
```

Details on this can be found from :

http://www.cstr.ed.ac.uk/projects/festival/manual/festival_28.html#SEC130.

You can disable access restrictions from `localhost` and `localhost.localdomain` by adding the following to a `.festivalrc` file in `$HOME` :

```
(set! server_access_list ' ("localhost"))  
(set! server_access_list ' ("localhost.localdomain"))
```

Or, you can just disable the access list altogether :

```
(set! server_access_list nil)
```

This will allow connections from anywhere, but should be OK if your machine is behind a firewall.

6.7.4 Installing more voices

I'm afraid this is a bit tedious. You can skip it if you are happy with the default voice. First find the Festival data directory. All Festival data goes to a common file tree, like in FlightGear. This can be `/usr/local/share/festival/` on Unices. We'll call that directory `$FESTIVAL` for now.

1. Check which voices are available. You can test them by prepending "voice_" :

```
$ festival
festival> (print (mapcar (lambda (pair) (car pair)) \
                        voice-locations))
(kal_diphone rab_diphone don_diphone us1_mbrola \
 us2_mbrola us3_mbrola en1_mbrola)
nil
festival> (voice_us3_mbrola)
festival> (SayText "I've got a nice voice.")
festival> (quit)
```

2. Festival voices and MBROLA wrappers can be downloaded here :

<http://festvox.org/packed/festival/1.95/>

The "don_diphone" voice isn't the best, but it's comparatively small and well suited for "ai-planes". If you install it, it should end up as directory `$FESTIVAL/voices/english/don_diphone/`. You also need to install "festlex_OALD.tar.gz" for it as `$FESTIVAL/dicts/oald/` and run the Makefile in this directory. (You may have to add "`-heap 10000000`" to the festival command arguments in the Makefile.)

3. Quite good voices are "us2_mbrola", "us3_mbrola", and "en1_mbrola". For these you need to install MBROLA (see above) as well as these wrappers : `festvox_us2.tar.gz`, `festvox_us3.tar.gz`, and `festvox_en1.tar.gz`. They create directories `$FESTIVAL/voices/english/us2_mbrola/` etc. The voice data, however, has to be downloaded separately from another site :
4. MBROLA voices can be downloaded from the MBROLA download page (see above). You want the voices labeled "us2" and "us3". Unpack them in the directories that the wrappers have created : `$FESTIVAL/voices/english/us2_mbrola/` and likewise for "us3" and "en1".

6.8 Air-Air Refuelling

As the name suggests, Air-Air Refueling (AAR) involves refueling an aircraft (typically a short-range jet fighter) from a tanker aircraft by flying in close formation. There are two types of refueling system supported. The KC135-E tanker has a boom that connects to a receiver on the refueling aircraft. The smaller KA6-D deploys a hose, into which the refueling aircraft inserts a probe.

A number of aircraft support AAR, including the T-38, Lightning, A-4F, Vulcan, Victor (which can also act as a tanker) and A-6E. You can tell if a particular aircraft support AAR by looking at the AI/ATC menu. If the "Tanker" menu item is enabled, the aircraft support AAR.

6.8.1 Setup

To set up AAR, simply start FlightGear with an AAR-enabled aircraft, take-off and climb to 15,000ft. Once cruising at this altitude, select AI/ATC->Tanker, and select "Request", which will spawn a new tanker in clear air at approximately your altitude.

FlightGear will report the altitude, speed, and TACAN identifier of the tanker. Program your TACAN with the TACAN identifier reported by the tanker (from the Equipment->Radio Settings dialog, or your cockpit controls). Depending on your aircraft, the tanker may also appear on your radar. If you require more help to find the tanker, you can select "Get Position" to be told the tanker location relative to yourself

Turn to an appropriate heading, guided by the TACAN bearing (you should try a "leading" approach to close in on the tanker) and look for the tanker on the radar or nav. screen. Around 5nm away, you should reduce your speed to around 20kts faster than the tanker - a "slow overtake". The KC135 will be visible from about 10nm, the KA6-D, being smaller, just over 1 nm. If you find yourself overshooting, deploy your airbrakes.

Close to within 50ft of the tanker (don't get too close, or you may collide with the tanker itself). You should see indication in the cockpit that you are receiving fuel - there is a green light in the A4 fuel gauge, and you should see the indicated tank load increase.

Once your tanks are full, or you have taken as much fuel as you wish, close the throttle a little, back away from the tanker and continue your intended flight.

Successfully refueling is not easy and can take a lot of practise, as in real life. Here are some helpful hints for making contact.

1. Approach the tanker slowly. It is very easy to overshoot and be unable to spot where the tanker has gone.
2. If you are having difficulty matching the speed of the tanker due to the throttle being too sensitive, try deploying your airbrakes. This will require more power to achieve the same speed and will reduce the throttle sensitivity.
3. To reduce your workload, you may be able to use the autopilot to stay at the correct altitude and/or speed. This is technically cheating, though NASA recently demonstrated that an advanced autopilot can perform AAR without pilot intervention.
4. Bear in mind that as you receive fuel your aircraft will become heavier and the center of gravity will move, affecting the trim of the aircraft.

5. The tanker aircraft fly a clock-wise "race-track" pattern in the sky. While it is possible to stay connected during these turns, you may find it easier to wait until the tanker has settled on its new course before refueling. The tanker aircraft provide warnings when they are intending to turn.

6.8.2 Multiplayer Refueling

Refuelling is possible within a Multiplayer session using the KC135 or Victor. The pilot of this aircraft should use the callsign "MOBIL1", "MOBIL2" or "MOBIL3". Other numbers are acceptable, but only these three have A-A TACAN channels assigned. These are 060X, 061X and 062X respectively.

If the receiving aircraft uses a YASim FDM, there are no further complications. Should the receiving aircraft be JSBSim based, the user must make sure that there are no AI tankers in their configuration. This means disabling (commenting out) all refuelling "scenarios" in the relevant aircraft-set.xml and in preferences.xml.

MP refuelling works in exactly the same way as AI refuelling and is a fun challenge. It is best to ensure that your network connection is as free from interruptions as possible; the MP code does a degree of prediction if there is a "blip" in the stream of packets and this can make close formation flight very difficult or even impossible.

Troisième partie

Tutorials

Chapitre 7

Tutorials

If you are new to flying, an advanced simulator such as *FlightGear* can seem daunting : You are presented with a cockpit of an aircraft with little information on how to fly it.

In real life, when learning to fly you would have an instructor sitting next to you to teach you how to fly and keep you safe.

While we cannot provide a personal instructor for every virtual pilot, there are a number of tutorials available that you can follow to become a proficient virtual pilot.

7.1 In-flight Tutorials

FlightGear contains an in-flight tutorial system, where a simulated instructor provides a virtual ‘lesson’. These vary between aircraft from simple tutorials teaching you how to start the engines on the aircraft, to full lessons teaching you how to fly for the first time. To access tutorials, Select Start Tutorial from the Help menu.

The tutorial system works particularly well with the Festival TTS system (described above).

For simplicity, run tutorials with AI aircraft turned off from the Options item on the AI/ATC menu. Otherwise, ATC messages may make it difficult to hear your instructor.

Each tutorial consists of a number of discrete steps which you must complete. Your instructor will provide directions on how to complete each step, and observe how you perform them, providing additional guidance if required.

Within a tutorial, to ask your instructor to repeat any instructions, press ‘+’. You can pause the tutorial at any time using the ‘p’ key. To stop the tutorial select Stop Tutorial from the Help menu.

7.1.1 Cessna 172P tutorials

If this is your first time flying, a number of tutorials exist for the Cessna 172P designed to teach you the basics of flight, in a similar way to a real flight school. The tutorials are based around Half-Moon Bay (KHAF) and Livermore Municipal (KLVK) airports near San Francisco. Both these airports are provided in the base package. To start the tutorials, select the Cessna 172P aircraft, and a starting airport of KHAF or KLVK, using the wizard, or the command line :

```
$ fgfs --aircraft=c172p --airport=KHAF
```

When the simulator has loaded, select Start Tutorial from the Help menu. You will then be presented with a list of the tutorials available. Select a tutorial and press Next. A description of the tutorial is displayed. Press Start to start the tutorial.

7.2 FlightGear Tutorials

The following chapters provide *FlightGear* specific tutorials to take the budding aviator from their first time in an aircraft to flying in the clouds, relying on their instruments for navigation. If you have never flown a small aircraft before, following the tutorials provides an excellent introduction to flight.

Outside of this manual, there is an excellent tutorial written by David Megginson – being one of the main developers of *FlightGear* – on flying a basic airport circuit specifically using *FlightGear*. This document includes a lot of screen shots, numerical material etc., and is available from

<http://www.flightgear.org/Docs/Tutorials/circuit>.

7.3 Other Tutorials

There are many non-*FlightGear* specific tutorials, many of which are applicable. First, a quite comprehensive manual of this type is the Aeronautical Information Manual, published by the FAA, and available at

<http://www.faa.gov/ATPubs/AIM/>.

This is the Official Guide to Basic Flight Information and ATC Procedures by the FAA. It contains a lot of information on flight rules, flight safety, navigation, and more. If you find this a bit too hard work, you may prefer the FAA Training Book,

<http://avstop.com/AC/FlightTraingHandbook/>,

which covers all aspects of flight, beginning with the theory of flight and the working of airplanes, via procedures like takeoff and landing up to emergency

situations. This is an ideal reading for those who want to learn some basics on flight but don't (yet) want to spend bucks on getting a costly paper pilot's handbook.

While the handbook mentioned above is an excellent introduction on VFR (Visual Flight Rules), it does not include flying according to IFR (Instrument Flight Rules). However, an excellent introduction into navigation and flight according to Instrument Flight Rules written by Charles Wood can be found at

<http://www.navfltsm.addr.com/>.

Another comprehensive but yet readable text is John Denker's "See how it flies", available at

<http://www.av8n.com/how/>.

This is a real online text book, beginning with Bernoulli's principle, drag and power, and the like, with the later chapters covering even advanced aspects of VFR as well as IFR flying

Chapitre 8

A Basic Flight Simulator Tutorial

8.1 Foreword

Aviation is about extremes :

- An airplane is quite fragile and flies at high speeds. Yet it is one of the safest forms of transport.
- Pilots must constantly follow rules and procedures. Yet an airplane is a symbol of freedom.
- With a little training, flight a small aircraft is easy. Yet if a problem occurs, you must be able to resolve it in a few seconds.
- Many flight tutorials are written with a lot of humor. Yet not taking flying seriously will bring you down to earth prematurely.

The aircraft used in this tutorial is the [Cessna 172p](#). This is the aircraft used in many real life flight schools and a great airplane to fly.



The following articles complement this tutorial and will answer most questions that may arise as you read through. This first one in particular is a good introduction to the airplane's main components and controls :

- <https://www.gleim.com/aviation/learn-to-fly/>
- http://www.pilotfriend.com/training/flight_training/aero/principa.htm

- <http://en.wikipedia.org/wiki/Aircraft>
- http://en.wikipedia.org/wiki/Flight_controls
- http://en.wikipedia.org/wiki/Airplane_flight_mechanics
- http://en.wikipedia.org/wiki/Aircraft_engine_controls
- <http://www.firstflight.com/flt1.html>
- <http://flightsims.ig-wilson.com/index.php?f16land>
- <http://www.navfltsm.addr.com/>

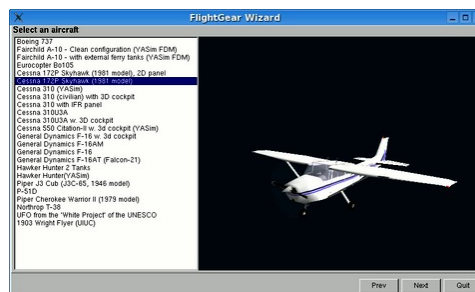
This tutorial is accurate to the best of my knowledge, but will inevitably contain some mistakes. I apologize in advance for these.

8.2 Starting Up

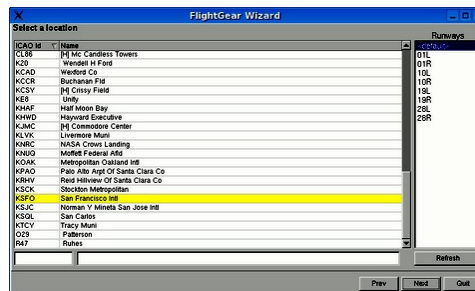
There are a number of different ways to start *FlightGear* based on your platform and the distribution you are using

8.2.1 MS Windows

On MS Windows, *FlightGear* has a GUI Wizard in which you can choose your aircraft and starting position. First choose the Cessna 172p airplane as shown below. To match this tutorial do not choose the 2D panel version. (You may however find in the future that the 2D version is more appropriate for training) Press the **Next** button to choose your airport.



You can start from any airport for this tutorial, but I will assume that you will start from FlightGear's default airport of San Francisco (KSFO) :



Once you have selected KSFO and pressed the **Next** button, you can set any number of options for the simulator. For your first flight, I suggest starting at noon. I would also recommend that you start with a small resolution of 800×600 . Later on you can play around with the options and use a higher resolution, but this obviously adversely affects performance. Press the **Run** button and the *FlightGear* will start with the options you selected.



If you have problems running the latest version FlightGear on your Windows system, you may want to try an earlier version with lower graphics requirements (for example 0.9.8) You can find previous releases on the FTP mirrors mentioned at the top of the *FlightGear* download page : .

If you are running under Windows Me and the flight simulator suddenly starts stuttering, with the frame rate dropping, try killing all tasks except Explorer and Systray before you launch FlightGear. If one of the tasks you kill is an antivirus or such protection software, this is an obvious security risk. Also, on one Windows Me machine, a *FlightGear* of 800×600 yielded good results, while a lower resolution of 640×480 triggered much lower FPS levels (Frames Per Second).

8.2.2 Linux and other unices

On Linux and other Unix-like systems, you may have to run *FlightGear* from the command line. If you have installed *FlightGear* but cannot find it in your menu system, try the following :

- From a terminal window (also named “console” window) try running the `fgrun` command. If installed, this will run the same GUI Wizard as under Windows as described above.
- Alternatively, open a terminal window and type the following command :
`fgfs --timeofday=noon --geometry=800x600.`

8.2.3 In the dark ?

Without the `--timeofday=noon` option, *FlightGear* will start at the current time in San Francisco - often night-time if you are in Europe. To change the time of day in the simulator to daytime, select **Weather->Time of Day** from the menu and select **Noon**.



If running *FlightGear* from a menu (e.g. under KDE or Gnome), you can edit the FlightGear launch icon properties and change the simple `fgfs fgfs` command to something like `fgfs --timeofday=noon --geometry=1024x768`, or include whatever command options you wish. Further details of the command line options can be found in Chapter 4, *Takeoff : How to start the program*.

8.3 The First Challenge - Flying Straight

Once *FlightGear* is started you will see the following window and hear the sound of an engine :



On startup, the aircraft is at the end of the runway with the engine running at low power. The airplane will occasionally tremble a little, but it won't move.

About the keyboard.

- In this tutorial, a lowercase key letter indicates you should simply press that key. An uppercase means you must press shift and that key. (The **↑ Shift** keys are those two keys with a hollow fat arrow pointing upwards.) In other words : if you are told to type “v”, simply hit the **v** key briefly. If you are told to type “V”, press the **Shift** key down and while you have it pushed down, hit the **v** key, then release the **Shift** key. (In short : V is the same as **Shift-v**.)

- The tutorial will assume you have the **NumLock** switched on. When switched on, you should find a small green light on at the right of your keyboard. Press the **NumLock** key repeatedly until the lamp is on.



Press **v**, to view the aircraft from the outside. Type **v** repeatedly to scroll through a number of different views until you return to the cockpit. Typing **V** will cycle backwards through the views.) :



In real life, we would have inspected the airplane all around to check everything is working, nothing is hampering the moving parts, and nothing is obstructing the instrument openings. In the simulator, this is already done for us before we start.

Hold the **Page Up** key down for about eight seconds. You will hear the engine sound rise.

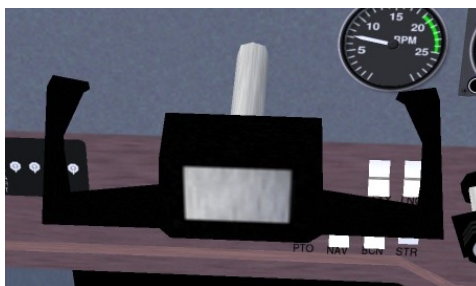
The airplane will start accelerating down the runway. As it does so, it will drift to the left, before finally taking off, banking to the left, falling to the ground and crashing (probably).

You can see a replay of the crash using the **View -> Instant Replay** menu. Click the **Replay** button at the bottom of the dialog window, then use **v** and **V** to see the airplane from the outside. The picture below shows the end part of the flight. You can take a snapshot by typing the **F3** key. You can also use the **F10** key to toggle the menu bar on or off.



Having observed your crash, exit from *FlightGear* (using **File->Quit**) and restart the simulator using the same options as before.

In order to fly straight you need the airplane's control yoke :



You can control the yoke using a joystick, or by moving the mouse. To use the mouse you need to be in mouse yoke mode. Get in that mode by clicking the right mouse button. The mouse cursor becomes a + sign. Move the mouse and see the yoke moving accordingly. Type **v** to see the plane from the outside. If you move the mouse again you will see the tail elevator and the ailerons at both wings ends move. If your viewpoint is too far from the aircraft to see any movement, type **x** a few times to zoom in. Type **X** to zoom back out. **Ctrl-x** returns the view to the default zoom level. Type **v** to change the view back to the cockpit.

Clicking the right mouse button again gets you in mouse view mode. In this mode the mouse cursor will be a \leftrightarrow sign. This allows you to look around easily. Clicking the left mouse button will re-center the view. A further right click will return you to the normal mouse mode.

To summarize, the right mouse button cycles the mouse through three modes :

- *Normal mode*. This mode allows you to click on the menu and on the instrument panel.
- *Yoke mode*. The mouse controls the yoke (+ pointer shape).
- *View mode*. The mouse controls the view direction (\leftrightarrow pointer shape).

Try taking off again using the mouse to control the yoke. Right-click to put the mouse in yoke mode (+pointer shape) and raise the engine throttle to maximum by holding the **Page Up** key down. Do not try to keep the airplane rolling straight

on the runway using the mouse/yoke. Let it drift leftwards. Wait till it rises in the air. Then use the mouse to try and get the airplane to fly straight. (If you want to control the airplane on the ground see section 8.5.)

You will find that you must prevent the airplane from banking to the left :



... or to the right :



... or from plunging to the ground :



Try to fly more or less straight, with the horizon stable slightly above the airplane nose :



Whatever your skills at video games or simpler simulators, you will probably not succeed at first. The airplane will crash, probably quite soon after take-off. This is the moment where most candidates get desperate and abandon trying to fly a simulator or a real aircraft. Just hold tight and keep trying. Eventually you will develop a feel for the subtle control inputs required.

The most common error is moving the mouse forwards to bring the nose up. In fact, you must pull the yoke by moving the mouse backwards to do this.

Equally, when you want to lower the airplane's nose, you must move the mouse forwards. This can seem odd, but all airplane control yokes are designed that way. With time, you will wonder how you every thought it worked any other way. You will also find that small mouse movements have a large effect on the aircraft. You may find that decreasing your mouse sensitivity may help initially.

If you have difficulty visualising this, the following analogy may help. Imagine a soccer ball is on your desk and you have "glued" your hand to the top of it. If you move your hand forwards the ball will roll forwards and your fingers will point to to the desk. If you move your hand backwards the ball will roll back and your fingers will now point up at the ceiling. Your hand is the airplane :



Another common error is the assumption that the control inputs directly match airplane bank. In other words, you believe if the control yoke is level, the airplane will fly level. This is not true. The yoke controls the *rate* at which the airplane banks. If the airplane is banked 20° to the left and the control yoke is level, the

airplane will stay banked at 20° left until some other force affects it. If you want to return the airplane to level flight, you have to turn the control yoke slightly to the right (move the mouse slightly rightwards) and keep it slightly to the right for a while. The airplane will turn slowly rightwards. Once it is level with the horizon, bring the control yoke level too. Then the airplane will stay level (until some other force changes its orientation).

A third error is trying to find “the right position” for the yoke/mouse. Naturally, you will want to find the fine tuning that will leave the airplane fly straight. Actually there is no such ideal yoke position. The airplane is inherently unstable in the air. You must constantly correct the airplane’s attitude and keep it flying straight with tiny movements of the mouse. This may seem to take all your concentration initially, but just like driving a car, keeping the aircraft straight and level will soon become second nature. For longer flights, you will eventually use the autopilot to keep the airplane level, but this is outside the scope of this tutorial.

To help fine-tune your senses to the control inputs required, keep your eyes on the outside scenery and not get fixated on the instruments or the yoke. Check the angle of the horizon and its height above the airplane’s nose. The horizon line and the airplane engine cover are your main flight instruments. Look at the instrument panel only once in a while.

While the mouse is in yoke control mode (+ pointer shape), don’t move it close to the FlightGear window edges. Once the mouse leaves the window, it stops controlling the aircraft, often at the worse possible moment ! If you wish to use the mouse outside of the window, first go back to standard mouse mode by clicking two times on the right mouse button.

You can also control the yoke using the four **keyboard arrow** keys or the keypad **8**, **2**, **4** and **6** keys. While initially this may seem easier than the mouse, you cannot make the very fine adjustments required for accurate flying, so it is much better to persevere with the mouse.

You may hear beeping sounds while flying around the airport. These are landing aid signals. Don’t worry about them for the moment.

You will know that you have mastered this when you can make the aircraft climb steadily in the air. The next step is to learn to keep the aircraft at a constant altitude, or to make it ascend or descend slowly and under your control.

Keeping the aircraft at a constant altitude involves observing the altimeter and making small changes with the mouse forwards or backwards to stop the aircraft ascending or descending respectively.

The altimeter instrument is at the middle top of the instrument panel. The long needle shows hundreds of feet, the short needle shows thousands of feet. The altimeter below shows an altitude of 300 feet, approximately 100 meters.



As you ascend or descend the altimeter will change accordingly, turning anti-clockwise as you descend, and clockwise as you gain height. If you see the altimeter “unwinding” you will be able to tell that you are losing height and move the mouse backwards slightly to raise the nose. After a while you will notice that when flying level the nose of the aircraft is always in the same position relative to the horizon. This is the aircraft attitude for level flight. By putting the nose in that same position, you will achieve almost level flight without having to reference the instruments. From there you can fine-tune your altitude.

Beware : an altimeter does not automatically show the absolute altitude above sea level. You must adjust for the local air pressure. The little black knob on the lower left side of the altimeter allows you to adjust the altimeter. Start *FlightGear* and stay on the ground. Click (in normal mouse mode) inside the black knob. A click on the left half makes the altimeter turn back. On the right half the altimeter turns forward. Use that little knob to tune in the current altitude. The principle is you use the knob when you are sure about the altitude. If you know you are at 1,100 feet altitude, tune in 1,100 feet on the altimeter. Clicking with the middle mouse button makes the knob turn faster. Type **Ctrl-c** to see the two button halves highlighted.

To make settings the altimeter easier, airports advertise their altitude in various ways. They may provide a radio service (called ATIS in the USA) to broadcast the current air pressure at sea level. This is expressed in inches of mercury. The altimeter contains a small scale inside which is calibrated in this way. You can set your altimeter using this scale. Alternatively, if you are on the ground and know the altitude of the airport, you can simply adjust your altimeter until it displays the correct altitude.



Note that there is an important difference between “altitude above sea level” and “altitude above the ground”. If you fly near Mount Everest at an altitude of 24,000 feet above sea level (AMSL), your altitude above the ground (AGL) will be much less. Knowing the altitude of the ground around you is obviously useful.

8.4 Basic Turns

While if you had enough fuel you could return to the same airport by flying straight head for thousands of miles, being able to change direction will make your flying more enjoyable and useful.

Once you are able to fly more or less straight, it is time to learn to turn. The principle is simple :

- When the airplane is banked to the left, it turns to the left.
- When the airplane is banked to the right, it turns to the right.



To turn, you do not need high levels of bank. 20° is more than enough for a safe and steady turn. The turn coordinator indicates your angle of bank by showing a depiction of your aircraft from behind. The picture below shows the turn coordinator when the airplane is banked 20° to the right. You can also tell the bank angle by observing the angle of the horizon.

Try the following : keep the airplane banked around those 20° for a few minutes and keep your eyes outside the aircraft. You will see the same ground features appear again and again, every 120 seconds. This shows you need 120 seconds to make a 360° turn (or 60 seconds for a 180° turn). This is particularly useful when navigating. Whatever speed the airplane is flying, if you bank at 20° you always need 60 seconds to make a 180° turn in the Cessna 172P.

So, by banking the airplane to the left or to the right, you make it turn to the left or to the right. Keeping the airplane level with the horizon keeps it flying straight.

The little purple ball in the bottom of the turn indicator shows the sideways forces. In real life you would feel these as your turn, however it is not possible to simulate these, so you must simply keep an eye on the ball. If you turn neatly (using the rudder a little bit), the ball will remain centered. If the ball is pushed say rightwards, this means you the pilot too are pushed rightwards. Like in a car turning to the left. During a neat turn in an airplane, even a strong turn, the passengers never endure a sideways force. They are only pushed a little harder on their seats by the centrifugal force.

By experimenting you will notice you can make much steeper turns by banking the airplane to high angles and pulling back on the yoke. Turns at over 60° bank angle are the realm of aerobatics and military flying, and dangerous is aircraft such as the Cessna.

8.5 Taxiing on the ground

While *FlightGear* starts you by default conveniently lined up on the runway and ready to go, you may be wondering how to get your aircraft from its hangar, along the taxi-ways to the runway. This is taxiing.

The picture below shows the instrument. It displays how fast the engine is turning in hundreds of revolutions per minute (RPM).



Type the **Page Up** key a few times, until the tachometer is showing 1,000 RPM (as shown above). If required type the **Page Down** key to decrease the engine speed.

At roughly 1,000 RPM, the airplane will move forward on the runway, but it will not accelerate and take off.

Type the “.”key (**Shift-;** on Azerty keyboards). The airplane will make a sharp turn to the right. If you keep the “.”key down the airplane will halt. When you type the “.” key, you are activating the brake on the right wheel of the airplane.

To activate the brake on the left wheel, use the “,” key.

The “,” and “.” keys simulate two brake pedals located at your feet on a real airplane. Using the throttle and the brake pedals you can control the speed of the aircraft and cause it to turn on the ground.

The brakes can be very useful when taxiing slowly on the runway. You can also steer the nose-wheel of the aircraft. In a real airplane this is done by pushing the rudder pedals with your feet. You push with your feet on the side you want to turn towards. If you don't have real rudder pedals, there are two ways to control the virtual rudder pedals :

- Using the keypad **0** and **Enter** keys . If you type the keypad **Enter** key say seven times, you will see the airplane firmly turns to the right and stays turning that way. Type the keypad **0** key seven times to get the airplane back rolling (almost) straight.
- Using the mouse. While the mouse is in yoke control mode (+ pointer shape), if you hold the left mouse button down, the mouse controls the rudder pedals instead of the yoke. The rudder pedals are connected to both the rudder and nose-wheel. This method is much more precise.

Start the simulator, Type **v** or **V** to view the airplane from the outside and keep **x** down a couple of seconds to zoom in on the airplane. Look at the front wheel and keep keypad **0** down. Then keep keypad **Enter** down. See the front wheel turn. Click on the right mouse button to get in yoke control mode (+ pointer

shape). Keep the left mouse button down to get in rudder control mode and move the mouse to the left and to the right. Note that the rudder, that big vertical control surface at the rear of the plane, moves together with the front wheel.

I tend to control the rudder pedals using the mouse while the front wheel is on the ground and use the keypad **0** and **Enter** keys once it has lifted off. In other words : I keep the left mouse button down while the front wheel is on the ground. This allows for a precise and easy rudder control on the ground. Then I simply release the left mouse button once the front wheel lifts off.



8.5.1 Airspeed

Just like driving a car, it is good to know how fast you are traveling. The aviation equivalent of a speedometer is the airspeed indicator (ASI), calibrated in nautical miles per hour (knots).



A knot is 1.85325 kilometer/hour. So, if you want to have a rough idea of your speed in flight expressed in km/h, multiply the knots displayed by 2. A knot is 1.15115 miles per hour, so very roughly, 1 knot is 1 mph. Note that some aircraft ASIs (in particular the Piper J3 Cub) display mph instead of knots.

The airspeed indicator displays the speed of the aircraft compared to the surrounding air, not the speed compared to the ground like a car speedometer does. If the plane is halted on the ground and there is a 10 knot wind blowing from straight ahead, the airspeed indicator will display 10 knots airspeed, although the plane will not be moving relative to the ground.

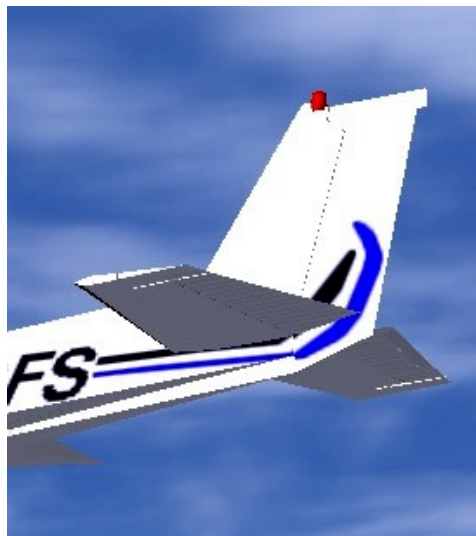
When the airplane rolls over the runway at more than 40 knots, you must prevent the front wheel from touching the ground. The nosewheel is not designed for high speeds and in real life would shimmy and wear out.

During take off, once over 40 knots you can make the front wheel leave the ground by pulling back gently on the control yoke. Don't turn sharply at high speed on the ground. Doing so may cause the aircraft to tip over.

The picture below shows the front wheel slightly lifted. Don't overdo this. Keep the airplane's white nose cover well below the horizon. You just need to lift the plane's nose very slightly.



Question : if the front wheel no longer touches the runway, how do you steer the airplane ? Answer : still using the rudder pedals. As mentioned above, the rudder pedals are linked to both the nose-wheel and the tail rudder, that big vertical moving part at the tail of the plane :



At airspeeds above 40 knots, the rudder has enough air-flow over it to steer the airplane.

Note the front wheel and the tail rudder don't make the airplane turn at exactly the same rate. So when the rudder takes over the front wheel, you must adapt the rudder pedals angle. That means fast typing keypad **0** and keypad **Enter** (or hold the left mouse button down and tightly control the rudder with the mouse).

Once you've become familiar with the nose-wheel and rudder, you can use these new controls to keep the airplane straight on the runway during take-off.

Say the airplane is heading too much to the right. You type keypad **0** a few times to make it turn back to the left. Don't wait till the aircraft has straightened up

completely. Type keypad **Enter** before the aircraft reaches the direction you wish to travel. Otherwise you will find that you will over-correct and have to turn back again. If you use the mouse, such corrections are much easier and more precise.

To summarise : two methods exist to steer the airplane on the ground : the differential brakes on the side wheels and the rudder pedals. This control redundancy is very common in aviation. If one method fails, you still have another method available to perform the task.

You may be wondering why the aircraft drifts to the left when it rolls on the ground, forcing you to compensate with a little push on the right rudder pedal ? The main reason is the flow of air produced by the propeller. It blows along the airplane body, but also corkscrews around the airplane fuselage. The upper part of that slight vortex pushes the vertical tail to the right. This causes makes the front of the aircraft to yaw to the left.

You can center all yoke and rudder controls by typing **5** on the keypad. This is a good preflight precaution. Sometimes it can “save your life” in flight if you find yourself with controls all over the place !

8.6 Advanced Turns

As with turning on the ground, there are two methods of turning in the air. You can use the wing ailerons (steered by the yoke/mouse) as described above or you can use the tail rudder (steered by the rudder pedals / the keypad keys **/0** and **Enter**).

Why two ways ? Partially for redundancy, but mainly because they are complementary. The main effect of the rudder is yaw (rotation around the vertical axis), while the main effect of the ailerons is roll (rotation around the longitudinal axis).

- When flying close to the ground, it is better not to bank the airplane in order to turn. The rudder is used more instead. Acting on the rudder pedals allows you to turn the airplane without excessive banking.
- When the plane is just above the runway, the two side wheels need to be at the same height above the runway for landing. That means the wings must be level with the horizon. The plane is not allowed to bank. You keep the plane wings level with the horizon by using the yoke/mouse/aileron. Note this does not need to be perfect. A bank of a few degrees is harmless.
- In flight, especially at high speed, the rudder is an in-efficient way to turn the aircraft :
 - It causes the airplane to present its flank to the airstream, increasing drag.
 - The airplane turns very slowly.
 - You will lack control while turning.
 - At high flight speed the centrifugal force will be disturbing or even dangerous.

Using the yoke/mouse/aileron allows for efficient, fast, reliable and comfortable turns.

- The rudder can be vital when the wings are stalled. Indeed, during a stall the wing ailerons become less effective or even useless. (Note that some airplanes can go in a very dangerous stall if you overdo the rudder control at low speed.)

When you turn in flight, using the ailerons, you still need the rudder a little bit. You add a little bit of rudder. This allows you to compensate for the adverse yaw created when you roll using the ailerons. In a real aircraft, you can feel this sideways motion. In the simulator, you can check this visually on the turn coordinator. In the picture below the little ball is pushed rightwards during a strong turn to the right using the ailerons. That means you the pilot endure a rightwards force too. You can compensate this by pushing the right rudder pedal (type the keypad **Enter** key a few times). In normal flight you should use the rudder to keep the little ball centered.



So, in normal flight use the ailerons to turn, while close to the ground at low speed use the rudder. However, one method never completely cancels out the other. You still need the rudder at high altitudes and speeds. Reciprocally you have to use the ailerons a little bit when close to the ground, to keep the wings level with the horizon.

Even when taxiing, you should use the ailerons. Otherwise, strong winds can blow the aircraft onto its side. To counteract this, you should turn the ailerons into the wind. This raises the aileron in the wind, helping to keep the wing down.

You should avoid making quick and aggressive movements of the rudder. On the ground at high speed this can make the airplane turn too sharply. In flight at low speed it can cause a very dangerous type of stall. In flight at high speed it can cause all kinds of aerodynamic and physical discomfort. Instead, make gentle movements of the rudder.

I recommend you practise turning with the rudder in flight. Fly at a low speed of about 70 knots. Try to keep the altitude stable by increasing and decreasing the engine power. Use the rudder to turn towards a ground feature and maintain a heading, then turn the aircraft towards a new heading. See how the plane yaws. Learn to anticipate rudder control. Don't try to make steep turns. Use the yoke/aileron to keep the wings level constantly.

8.7 A Bit of Wieheisterology

Wieheisterology comes from the German phrase “Wie heißt Er” – “What’s that name”. This section is about gauges, switches and controls of the aircraft. While in the simulator you can take off and land a basic airplane with just the engine throttle and the yoke, but you will need all the controls to perform securely and efficiently.

8.7.1 Engine control

An airplane engine is designed for simplicity, reliability and efficiency. Rather than use advanced electronic ignition and fuel injection systems found in modern cars, they instead use older technology that doesn't rely on electrical power. That way, the plane can still fly even if it suffers complete electrical failure.

Magneto

On the bottom left, below the instrument panel you will find the magneto switch and engine starter :



To see the switch, either type **P** to get the schematic instrument panel or type **Shift-x** to zoom out (**x** or **Ctrl-x** to zoom back in).

You can move the switch with the **{** and **}** keys (use the **Alt Gr** key on Azerty keyboards).

You are probably aware that the fuel inside a car engine is ignited by electric sparks. Modern car engines use electronic ignition. An airplane engine uses a more old-fashioned (but more reliable) magneto ignition instead. For redundancy, it contains two such magnetos : the “left” one and the “right” one. When you change the magneto switch on OFF, both magnetos are switched off and the engine will not run. With the magneto switch on L you are using the left magneto. On R you are using the right magneto. On BOTH you use both. In flight you will use BOTH.

Given that you use both magnetos in flight, why have the switch ? The reason is that during your pre-flight checks you will verify that each of the magnetos is working correctly. To do this, increase the RPM to about 1500 then switch the magneto switch to L and observe the tachometer. You should observe a slight drop in RPM. If the engine cuts out, the left magneto is broken. If you do not see an RPM drop, then the switch may be faulty, as both magnetos are still switched on. You can then perform the same test on the right magneto. Of course, in the simulator, the magnetos are unlikely to fail !

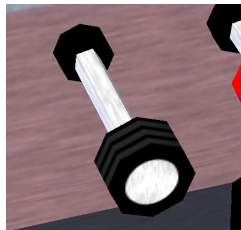
Should one of the two magnetos fail in flight, the other one will keep the engine running. The failure of one magneto is rare, the failure of both simultaneously is almost unheard of.

You may have typed **{** to shut the engine down. To start the engine again after doing so, type **}** three times in order to put the magneto switch on BOTH. Then use the starter motor by pressing the **s** for a few seconds, till the engine is started.

You can also turn the magneto switch and start the engine by clicking left and right of the switch in normal mouse mod). Type **Ctrl-c** to see the two click sides highlighted by yellow rectangles.

If you turn the switch to OFF, the engine noise stops. If you quickly turn the switch back to L, the engine starts again as the propeller is still turning. If you wait for the propeller to stop, placing the switch on L, R or BOTH won't start the engine. (Once the engine is halted, always place the magneto switch to OFF.)

Throttle

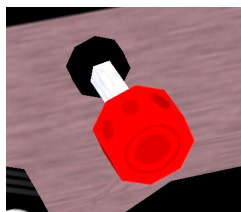


You already know that you increase the engine power by pushing that throttle rod in (**Page Up** key). You decrease the power by pulling the control out (**Page Down** key). You can also click left and right of the lever (middle mouse button for quicker moves, **Ctrl-c** to highlight the left and right halves).

What does “increase the power” actually mean? Does it mean you increase the amount of fuel delivered to the engine? Yes, but this is not enough to fully understand what you are doing. You need to be aware that the engine is also fed with a huge amount of air. The engine's cylinders burn a mixture of fuel and air. Fuel alone wouldn't burn. Only a mixture of fuel and air can detonate and move the engine pistons. So when you push the throttle in, you increase both the fuel and the air fed to the engine.

Mixture

The amount of air compared to the amount of fuel is critical. The proportion of the two has to be tuned closely. This is the purpose of the mixture lever. The picture below displays the mixture lever, pulled out far too much.



When the mixture lever is fully pushed in, you feed the engine with an lots of fuel and little air. This is known as a “rich” mixture. When the lever is pulled

out completely, there is an excess of air, known as a “lean” mixture. The correct position to produce maximum power is in between these two extremes, usually quite close to fully pushed in.

When you start the engine and when you take off, you need a fuel-rich mixture. That means the mixture lever should be pushed in. A fuel-rich mixture allows the engine to start easily. It also makes the engine a little more reliable. The drawback is that a part of the fuel is not burned inside the engine. It is simply wasted and pushed out the exhaust. This makes the engine more polluting, it decreases the energy the engine can deliver and it slowly degrades the engine by causing deposits of residues inside the cylinders.

Once in normal flight, you have to pull the mixture lever a little, to get a more optimal mixture. Check this out by doing the following. Start the simulator. Put the parking brakes on with key B (that is **Shift-b**). Push the throttle in to its maximum. The engine RPM should now be close to the maximum. Slowly pull on the mixture lever (using the mouse in normal pointer mode). You will see the RPM increases a little. You get more power, without increasing the fuel intake. You waste no fuel and it pollutes less. If you continue to pull the mixture lever, the RPM will decrease back away, because now there is too much air. The excess of air slows the explosions down inside the cylinders and decreases the explosion temperature, hence the thermodynamic yield decreases. You have to tune in the optimal mixture. For thermodynamic reasons, the best mixture isn't exactly at maximum power - it is better for the engine to be running very slight richer or leaner than maximum power. This also avoids the possibility of the fuel detonating explosively damaging the engine. You can find the maximum power point by the fact you get the highest RPM. (Another method is to check the engine exhaust temperature. Roughly, this is the point at which you get the highest temperature.)

The mixture control allows you to burn less fuel for the same speed and distance, and therefore fly farther and pollute less. However, if you mis-manage it, it can cause serious problems. Suppose you go flying at high altitude and pull out the mixture lever accordingly. At high altitude there is less oxygen available so the correct mixture will be quite lean - i.e. with little fuel being used. Then you descend back in order to land. If you forget to push the mixture lever in as you descend, The fuel/air mixture will become far too lean and the engine will simply halt.

When landing, you have to tune back in a mixture that is a little too rich in fuel. This means pushing the mixture lever in. That way the engine becomes a little more reliable and will be better adapted to a decrease in altitude.

I wrote above that placing the magneto on OFF is not the right way to stop the engine. The right method is to pull the mixture lever. First pull the throttle out completely, to get the engine to minimum power and fuel consumption. Then pull the mixture lever, till the engine stops because the mixture contains too much air. This ensures the engine doesn't get choked by waste fuel residues. Finally, turn the magneto switch to OFF to ensure the engine won't start again accidentally.

An important warning : you may think the RPM indicator reflects the engine power. Wrong. Two things make the RPM increase : the engine power and the

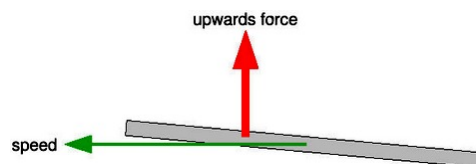
airplane speed. To check this, fly to a given altitude then pull the engine power to minimum. Try out diving to the ground then rising back to altitude. You will see the RPM varies significantly as does your airspeed. It rises while diving and decreases while climbing.

One pitfall of this is when you intend to tune the engine power in for landing. Suppose you're descending towards the airport, flying fast. You know the ideal RPM for landing is around 1,900 RPM. So you pull the throttle till you get 1,900 RPM. You think you tuned in the appropriate RPM. You think you shouldn't bother any more about it. But when you level off, the plane's speed starts to decrease, along with the RPM. A few minutes later, you get the low flight speed you wanted. You don't see the RPM is now far too slow. You will either lose altitude or stall. Or both. Be cautious with the throttle and with the RPM indicator. Either pull on the throttle more steadily or be mentally prepared to push it back in quickly.

8.7.2 Wings and speed

Say you are flying with full engine power. Dropping the nose a little makes you lose altitude and raising the nose a little makes you gain altitude. You may think this is quite straightforward. The plane travels in the direction it is heading ; the direction the propeller is heading. This is not the best way to think about it. This model would be fine for a rocket, but not for an airplane. A rocket is lifted by its engine, while a plane is lifted by its wings. That's a huge difference.

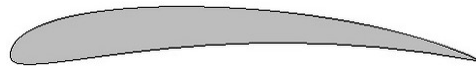
Get a big rigid square of cardboard, hold it horizontally in your hand with your arm stretched out and make it do fast horizontal movements while rotating your torso. When the cardboard moves flat through the air, it experiences no lift force. If you twist your arm slightly to give the cardboard a slight upward angle, you will feel it tends to lift in the air. There is an upward force acting on the cardboard. That's the way a wing holds a plane in the air. The wings have a slight upward angle and lift the airplane. The more angle you give the cardboard, the more lift force. (Till you give it too steep an angle. Then you will rather feel a brake force. The cardboard is "stalling" (see below).)



- When you pull the yoke, the airplane's nose rises up. Hence the wings travel through the air at a steeper angle. Hence the lift force on the wings is stronger. Hence the plane rises in the air.
- When you push the yoke, the airplane's nose dives. Hence the wings travel through the air with less angle. Hence the lift force on the wings decreases. Hence the plane descends.

What matters is the angle the wings travel through the air. This is the angle of attack.

I wrote above that when the wings travel through the air with no angle of attack, they don't produce lift. This is false. It would be true if the wings were a flat plate like the cardboard. But they aren't. The wings are a slightly curved airfoil. This makes them create lift even when traveling through the air at no angle of attack. Actually, even with a little negative angle of attack they still create a lift force. At high speed the airplane flies with the wings slightly angled towards the ground !



The angle at which the wings travel through the air matters. Something else matters too : the speed. Take the cardboard again in your hand. Hold it with a given slight angle and don't change that angle. Move it at different speeds through the air. The faster you move the cardboard, the more upward force it experiences.

- When you increase the engine power, the plane increases speed, the lift force on the wings increases and the plane gains altitude.
- When you decrease the engine power, the plane decreases speed, the lift force on the wings decreases and the plane loses altitude.

To make things a little more complicated : when rising in the air, the airplane tends to lose speed. When descending, it tends to gain speed.

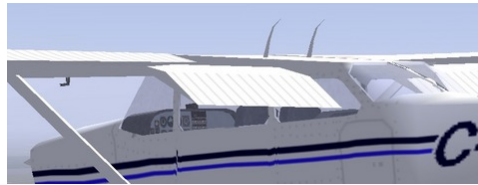
That's all a matter of compromises. If you want to fly at a constant altitude and at a given speed, you will have to tune both the engine power and the yoke/elevator (or better : the trim (see below)), till you get what you want. If you want to descend yet keep the same speed, you have to push the yoke a little and decrease the engine power. And so on. You constantly have to tune both the engine power and the yoke. However, during a normal flight you can simplify this by simply choosing a comfortable engine power level then relying on the yoke and trim for altitude.

A very interesting exercise you can perform with the simulator is to fly straight with full engine power. Get maximum speed while keeping in horizontal flight. Then decrease the engine power to minimum. Pull steadily on the yoke to keep the plane at constant altitude. The plane slows down steadily, meanwhile you have pull more and more on the yoke to stay level. Since the speed decreases the lift from the wing will decrease, but you compensate the loss of speed by increasing the wing angle of attack. This proves the plane does not necessarily travel in the direction its nose is heading. In this experiment we make the nose rise in order to stay at constant altitude. Once the plane is flying very slowly, and the nose is very high, you may hear a siren yell. That's the stall warning (see below). This indicates that the angle of attack is too high for the airfoil to produce lift. The wings are no longer producing lift and the plane quickly loses altitude. The only way to correct this is push the yoke forwards to reduce the angle of attach, making the nose drop, then apply full power to gain speed and finally bring the yoke carefully back to level flight.

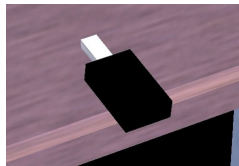
Question : is it better to control the airplane's speed and altitude with the yoke or with the throttle ? Answer : it depends on what exactly you intent to do and on the situation you are in. In normal flight, as said above, you tend to set a comfortable engine power level, forget about it and rely on the yoke and trim. During take off and landing the procedures are quite strict about the use of yoke and throttle. You do the opposite : control the speed with the yoke and trim, control the altitude and descent speed with the engine throttle. This will be discussed further below.

8.7.3 The flaps

The flaps are situated at the rear of the wings, either side of the aircraft fuselage.



You deploy the flaps and retract them back in by using the flaps control lever :



You can either click on it with the mouse or use the [and] keys. Key [to retract the flaps one step,] to deploy them one step at a time. Type **v** to view the plane from the outside and try out [and]. (On the schematic instrument panel the flaps lever is located at the lower right.)

In the Cessna 172P. there are four flaps settings :

- 0° - for normal flight and normal take off.
- 10° - for short field take off, when you want to gain altitude while flying slowly. Or during the first stange of an approach to land.
- 20° - to slow the aircraft and lose altitude quickly, for example when descending towards the runway to land.
- 30° - To lose altitude even more quickly.

The flaps are somewhat delicate. Do not deploy the first step of flaps above 110 knots. Do not deploy the second or third stage of flaps above 85 knots.

The flaps create large amounts of drag on the aircraft and brake the plane at high speed. This is one more reason not to forget to pull the flaps back in once you fly above 85 or 110 knots.

To check the flaps position visually, either use the mouse view mode to look at the back of the wing, or type **Shift-right arrow** to shift the view to the right and then quickly **Shift-up arrow** to get back to front view.

Flaps increase wing lift by altering the shape of the airfoil. The wing lifts more at a given speed with the first stage of flaps set. Hence you will get in the air a little sooner during take off. It also has the effect to make the plane fly with a lower nose attitude. This is useful as it provides a better view of the runway when taking off or landing.

The flaps also increase drag on the aircraft. The second and third stage of flaps produce much more drag than lift, so they are used to brake the plane. This is particularly useful when landing, because the airplane glides very well. If you cut down the engine power completely, the plane will descend, yet but too slowly. You need to deploy two or three flaps steps in order to brake and really descend towards the ground.

The fact that the flaps brake during landing makes you need more engine power during the landing. This can seem odd. Why not simply throttle the engine down to minimum and use less flaps steps? The answer is that it is better to have a strongly braking plane and lots of engine power, as the plane reacts faster to your commands. Should the engine fail, then just retract flaps as needed and glide to the runway.

What can you do if you have full flaps extended and need to increase your rate of descent further? Slowly push the rudder pedals on one side. This will make the plane present its flank to the air stream and brake. Compensate the turning by using the ailerons (yoke). This is known as side-slipping, and is a very effective way to lose height progressively as it is easy to stop at any point.

8.7.4 The stall

An aircraft relies on the smooth flow of air over the surface of the wing to produce lift. However, if the wing is at too high an angle of attack, this flow is broken, and the wing no-longer produces lift. With no lift, the aircraft cannot fly, and quickly drops back to earth. This is known as a stall.

A stall is an emergency situation, whatever the While it can happen at any speed, it commonly occurs in slow flight. A given aircraft has a specific stall speed, at which no angle of attack can produce enough lift. You should always keep your aircraft well above the stall speed. To help, aircraft are equipped with stall sirens that sound when the angle of attack is approached.

If you encounter a stall, the remedial action is to immediately drop the nose, and apply full power, bringing the nose level when flying speed has been attained again. However, doing so will cause the aircraft to lose altitude, which you may not have to spare when landing or taking off!

A spin occurs when one wing stalls before the other, which can occur in a steep turn at low speed. As one wing is still flying, the aircraft turns around the stalled wing, spinning tighter and tighter. To get out of a spin, you need to apply rudder to straighten out the spin into a normal stall, then recover as above.

Aircraft like the Cessna 172 and Piper Cub, have benign stalls, and are unlikely to enter a spin. High performance jets, such as the F16 have much more aggressive

stalls, and can easily enter a spin.

To practise this in the simulator, do the following :

- Fly at constant altitude and attitude.
- Reduce engine power, raising the nose to avoid entering a descent.
- Continue to reduce power until the stall begins.
- Try to control the plane while it stalls and descends to the ground.
- Keep the yoke pulled to the maximum and the plane in a steady attitude, the wings parallel with the horizon. Try to change direction.
- Recover by lowering the nose, applying full power, and correcting the attitude once flying speed has been regained.

You can also experiment with stalls with different flap settings, and high speed stalls by making abrupt attitude changes.

Experiment with different aircraft. Compared with the Cessna 172 the Cessna Citation jet, stalls much more aggressively and with little warning..

8.7.5 The trim

The trim is the dark big vertical wheel with gray dots located at the middle below the instrument panel :



On *FlightGear*, the keys **Home** and **End** adjust the trim. **Home** rolls the wheel upwards while the **End** rolls the wheel downwards. You can also click on the upper or lower half of the trim wheel.

In first approximation, the trim does the same as the yoke : it acts on the elevator. Turning the trim wheel downwards is the same as pulling on the yoke. Yet there is a key difference between the trim and the yoke. The trim remains in position after you make a change, while the yoke only continues to affect the elevator while you apply pressure and returns the elevator to neutral when you release it.

During cruise flight, the required elevator position to keep the aircraft at constant altitude will not be completely neutral - it will vary depending on the air outside the aircraft, the current fuel level, and the payload. Obviously, holding the yoke continually to retain a constant attitude would quickly become tiring. By using the trim to “trim out” the elevator force required for cruise flight, the yoke can be kept neutral.

During take off the trim should be neutral. Otherwise you may find that it either refuses to take-off with the normal level of yoke control, or takes off too quickly.

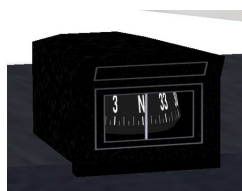
During landing, try to get the yoke/mouse/elevator towards neutral position by tuning the trim. This makes making small adjustments to your attitude and position easier. On the Cessna 172p this means trim on neutral. On the Cherokee Warrior II this means the trim a little “pulled”.

The trim wheel movement is much slower than the yoke, allowing for delicate changes in trim. Be patient.

8.7.6 What direction am I flying ?

Knowing the direction you are going is obviously a good idea. There are three basic ways to determine the direction you are flying :

- Look through the windows. If you are flying regularly from the same airport, you will learn to recognize the ground features such as roads, hills, bridges, cities, forests. In a simulator, you only have a narrow view of the virtual outside world. Several ways exist to allow you to pan your virtual head inside the airplane :
 - Use **Shift** and the four arrow keys to look to the front, rear, left and right.
 - Use **Shift** and the keypad keys to look in the four directions mentioned above and in four diagonal directions in-between.
 - Use the mouse in view mode (right button, \leftrightarrow). This allows you to look in every direction, including up and down. Click the left mouse button to bring the view back to straight ahead.
- The magnetic compass. This is located above the instrument panel. The compass is simple, but is affected by the acceleration of the aircraft, and magnetic abnormalities on the ground. Also, the compass points towards magnetic north rather than true north. This deviation varies depending on your location.



- The directional gyro (picture below) or “heading indicator”. The gyro is powered by a vacuum system. The gyro is set to match the magnetic compass, and is not affected by magnetic issues, or aircraft movement. However, due to gyroscopic precession and friction in the instrument, over time it drifts and must be reset by reference to the magnetic compass on occasion. To reset the HI, during cruise flight, use the black knob on the bottom left of the instrument (normal mouse pointer mode, click left or right half of the knob, middle mouse button to move faster, **Ctrl-c** to highlight halves). (The red knob, bottom right, is used to tell the autopilot what direction you wish to fly (**HDG** = “heading”).



8.7.7 A look around the panel

Finally, let's have a look at the instrument panel, combining the instruments described above with some new ones.

The six-pack

Let us start with the most important instruments any simulator pilot must know, these are known as the “holy six” or the “six-pack” . In the center of the instrument panel (Fig. 5), in the upper row, you will find the artificial horizon (attitude indicator) displaying pitch and bank of your plane. It has pitch marks as well as bank marks at 10, 20, 30, 60, and 90 degrees.

Left of the artificial horizon, you'll see the airspeed indicator. Not only does it provide a speed indication in knots but also several arcs showing characteristic velocity ranges you have to consider. At first, there is a green arc indicating the normal operating range of speed with the flaps fully retracted. The white arc indicates the range of speed with flaps in action. The yellow arc shows a range which should only be used in smooth air. The upper end of it has a red radial indicating the speed you must never exceeded, unless you want to break up the plane in mid-flights. . .

Below the airspeed indicator you can find the turn indicator. The airplane in the middle indicates the roll of your plane. If the left or right wing of the plane is aligned with one of the marks, this would indicate a standard turn, i.e. a turn of 360 degrees in exactly two minutes.

Below the plane, still in the turn indicator, is the inclinometer. It indicates whether the rudder and ailerons are co-ordinated. During turns, you always have to operate aileron and rudder in such a way that the ball in the tube remains centered ; otherwise the plane is skidding. A simple rule says : “Step on the ball”, i.e. step onto the left rudder pedal when the ball is on the left-hand side.

If you don't have pedals or lack the experience to handle the proper ratio between aileron/rudder automatically, you can start *FlightGear* with the option `--enable-auto-coordination`.

To the right-hand side of the artificial horizon you will find the altimeter showing the height above sea level (not ground !) in hundreds of feet. Below the altimeter is the vertical speed indicator indicating the rate of climbing or sinking of your plane in hundreds of feet per minute. While you may find it more convenient

to use than the altimeter in certain cases, keep in mind that its display usually has a certain time-lag.

Further below the vertical speed indicator is the propellor tachometer, or RPM (rotations per minute) indicator, which displays the rotations per minute in hundreds. The green arc marks the optimum region for cruise flight.

The group of the main instruments further includes the gyro compass being situated below the artificial horizon. Besides this one, there is a magnetic compass sitting on top of the panel.

Four of these gauges being arranged in the from of a “T” are of special importance : The air speed indicator, the artificial horizon, the altimeter, and the compass should be scanned regularly during flight.

Supplementary Instruments

Beside the six-pack, there are several supplementary instruments. To the very left you will find the clock, obviously being an important tool for instance for determining turn rates. Below the clock there are several smaller gauges displaying the technical state of your engine. Certainly the most important of them is the fuel indicator - as any pilot should know.

The ignition switch is situated in the lower left corner of the panel (cf. Fig. 4). It has five positions : “OFF”, “L”, “R”, “BOTH”, and “START”. The first one is obvious. “L” and “R” do not refer to two engines (as the Cessna 172 only has one) but the two magnetos, providing redundancy in the case of a failure.. The two switch positions can be used for test puposes during preflight. During normal flight the switch should point on “BOTH”. The extreme right position is for using a battery-powered starter (operated with the “s” key).

The handle below the yoke is the parking brake. In the vertical position, the parking brake is ON. The parking brake is operated with the “B” key.

Radios

The right hand side of the panel is occupied by the radio stack. Here you find two VOR receivers (NAV), an NDB receiver (ADF) and two communication radios (COMM1/2) as well as the autopilot.

The communication radio is used for communication with air traffic facilities ; it is just a usual radio transceiver working in a special frequency range. The frequency is displayed in the LEDs. Usually there are two COM transceivers ; this way you can dial in the frequency of the next controller to contact while still being in contact with the previous one.

The COM radio can be used to listen to the current weather conditions at an airport, known as ATIS. To do this, simply dial in the ATIS frequency of the relevant airport. You can find this by selecting `ATC/AI->Frequencies` from the menu, and selecting the 4-letter ICAO code of a nearby airport.

Each COM radio has two frequencies configured - an 'active' frequency which the pilot is transmitting and receiving on, and a 'standby' frequency which may be changed. In this way, you can continue to listen on one frequency while tuning another one.

You can change the radio frequency using the mouse. For this purpose, click left/right to the circular knob below the corresponding number. The corresponding switch left to this knob can be used for toggling between the active/standby frequency.

Use of the autopilot and radio navigation equipment is covered in later tutorials. For the moment you can ignore these radio instruments as long as you are strictly flying according to VFR (visual flight rules).

8.8 Let's Fly

By now you will be able to keep on runway while taking off by using the rudder and you're able to fly straight, descend, climb and make gentle turns. This section will describe a slightly more realistic approach to taking off and landing, and introduce some of the more subtle concepts you should be aware of.

8.8.1 A realistic take off

The following general rules apply during a normal take-off :

- The nose-wheel should be lifted from the runway at approximately 40 knots.
- Immediately after take-off, you should accelerate to 70 knots, to stay well above stall speed, in case of a gust of wind, or engine failure.
- Don't fly much above 75 knots to ensure you gain height as quickly as possible.
- Follow the runway heading until at 500 feet. This way, if you suffer an engine failure, you can easily land back on the runway you left.
- Don't over-fly buildings until at least 1,000 ft.
- Close to the ground, turns should be gentle and well coordinated using the rudder.

So, you need to take off and rise in the air at a steady speed of around 75 knots. However, when you raise the nose slightly at 40 knots, the aircraft will probably take-off at around 55 knots. To accelerate quickly to 75 knots, lower the nose slightly immediately on take-off, then raise it once 75 knots has been achieved. You are using the yoke to control the speed of the aircraft.

Putting this all together with what you have learned previously, a normal take-off using the mouse will consist of the following :

1. Adjust the altimeter to the correct altitude, based on the airport altitude. For reference, KSFO is at sea level - 0ft.
2. Check aileron and elevator are neutral by observing the yoke position.
3. Change the mouse to control mode by pressing the right mouse button.

4. Hold the left mouse button down to control the rudder.
5. Apply full power (hold **PgUp** until the throttle is fully in).
6. As the aircraft accelerates down the runway, keep it in the center by making small adjustments using the mouse.
7. As 40kts is reached, release the left mouse button, and pull back slightly to raise the nose-wheel. You are now controlling the yoke with the mouse.
8. The aircraft will fly off the runway at approximately 55 knots.
9. Lower the nose slightly to accelerate to 70 knots
10. Keep alignment with the runway.
11. Use the yoke to keep the ASI at 70 knots as you climb. If the airspeed is dropping, lower the nose. If the airspeed is increasing, raise the nose slightly.
12. Once you reach 500 feet, turn to your required heading, staying away from buildings until you are over 1,000ft.

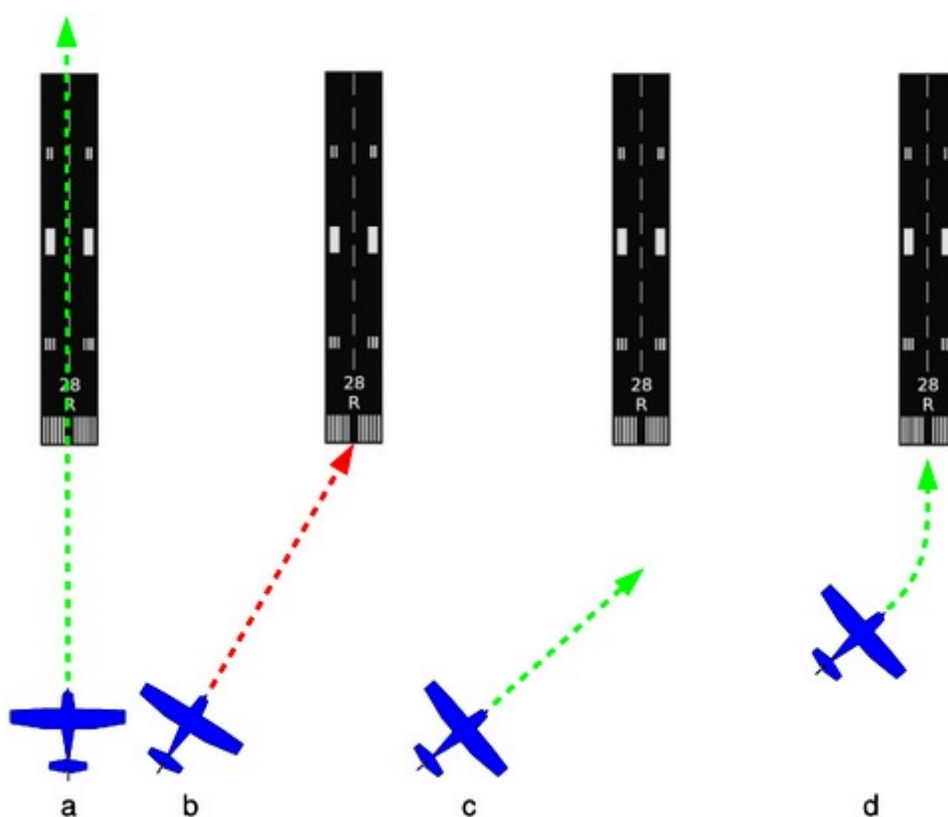
8.8.2 Landing

The rules for landing are almost identical to that of take-off, but in reverse order :

- Close to the ground, turns should be gentle and well coordinated using the rudder.
- Stay above 500ft until on final approach to the runway
- Approach the runway at approximately 70 knots.
- Touch down on the two main wheels at 55kts.
- Let the nosewheel touch down at 40kts.

Landings are much easier if you have an aiming point picked out on the runway. By observing the aiming point, you can easily tell if you are descending too fast or too slowly. If the aiming point appears to move upwards, you are descending too fast,

Obviously, you need to be lined up with the runway. That means your flight direction has to match the middle line of the runway (drawing (a) below). In order to arrive at this, don't aim at the start of the runway (b). Rather aim at a fictitious point well in front of the runway (c). And begin to turn gently towards the runway well before you reach that fictitious point (d). Note the turns and bankings you make for these flight corrections are often very soft. You wouldn't even notice them on the turn coordinator. This is one example where you better rely on the outside horizon line than on the inside flight instruments.



A straight in landing using the mouse would consist of the following :

1. 1,500ft above the airport, and a couple of miles out, an in-line with the runway, reduce power to approximately 1500rpm. This will slow you down somewhat and start a gradual descent.
2. Once below 115 knots, apply one step of flaps (J). This will increase lift, but also slow you down.
3. Re-trim the aircraft so you continue to descend.
4. At around 1,000 feet, apply another step of flaps (J). This increase drag significantly, but also improve the view over the nose.
5. Tune the speed using the elevator and trim : push the yoke if you are flying below 70 knots, pull the yoke if you are flying above 70 knots. If using a joystick, use the trim to relieve any pressure on the elevator.
6. Tune the altitude using the engine throttle. Add power if you are descending too fast, reduce power if you are too high. It is much easier to work out if you are too high or too low by observing the numbers on the runway. If they are moving up the screen, you are descending too fast - increase power. If they are moving down, you are too high and need to reduce power.
7. Make minor adjustments to heading to keep aligned with the runway.

8. at about 500ft, apply the final step of flaps. (**J**). This increase drag significantly, so be prepared to increase power to keep your descent constant.
9. When you are just above the runway, reduce power to idle, and use the yoke to gently pull back the aircraft to horizontal. This is the “round-out” and should result in the aircraft flying level with the runway a couple of feet above the surface. Performing the round-out at the correct height is a difficult task to master. To make it easier, observe the horizon rather than getting fixated on the aiming point.
10. Keep the wings level using small inputs with the yoke. We want both wheels to touch down at the same time.
11. Continue pulling back on the yoke. The main wheels should touch down at about 55 knots. This is the “flare”.
12. As you touch down, be ready to use the rudder to keep the aircraft straight (keypad **0** and keypad **Enter**)
13. Once you are below 40 knots, lower the nose-wheel to the ground.
14. Hold down the left mouse button to control the nosewheel/rudder using the mouse.
15. Once below 30 knots, use the brakes **b** to slow the aircraft further.



Once the plane is halted or at very low speed, you can release the **b** key and add a little engine power to taxi to the parking or hangar.

8.8.3 Engine Shutdown

To shut the engine down :

- Set the parking brake, type **B**.
- Engine throttle to minimum (hold **Page Down** down for a while).
- Pull the mixture lever to halt the engine (mouse in normal pointer mode, click on the left of the red mixture lever to pull it out).
- Rotate the magneto switch to OFF (a few hits on **{**).

8.8.4 Aborted Landing

You must be mentally prepared to abort landing any time the landing doesn't look good, or due to external factors such as :

- an order from the control tower
- incorrect speed or landing angle when there is insufficient time to correct it.
- a strong gust blow of wind
- birds flying over the runway.

To abort the landing, apply full power (hold **PgUp**), raise the nose to climb, and once you are climbing, retract the flaps (key[]).

Landing is much harder than taking off. Landing on a large runway, such as KSFO (San Francisco, the default) is much easier than smaller runways such as KHAF (Half Moon Bay, about 10 miles to the south west of KSFO).

To practise landings, use the command line below in a terminal window to start the simulator in flight and heading for the runway. The airplane is placed 5 miles ahead of the runway, at an altitude of **1500** feet and a speed of about 120 knots.

```
fgfs --offset-distance=5 --altitude=1500 --vc=120 --timeofday=noon
```

Approaching to land at 65 knots instead of 70 knots allows to use a much shorter runway length. However, this requires better control, particularly as it is much closer to the stall speed. It is quite different from landing at 70 knots.

8.9 Dealing with the Wind

Consider a hot air balloon. Think of it as being in the middle of a gigantic cube of air. The cube of air may move at high speed compared to the ground, but the balloon itself is completely static in the middle of the cube. Whatever the wind speed, persons aboard a hot air balloon experience not a breath of wind.

In the same way, an aircraft flies in the middle of a gigantic cube of air and flies relative to that air mass. The motion of the cube of air relative to the ground has no effect on the aircraft.

You, the pilot, on the contrary, are interested in the speed of the surrounding air compared to the ground. It can make you drift to the left or to the right. It can make you arrive at your destination much later or much sooner than planned.

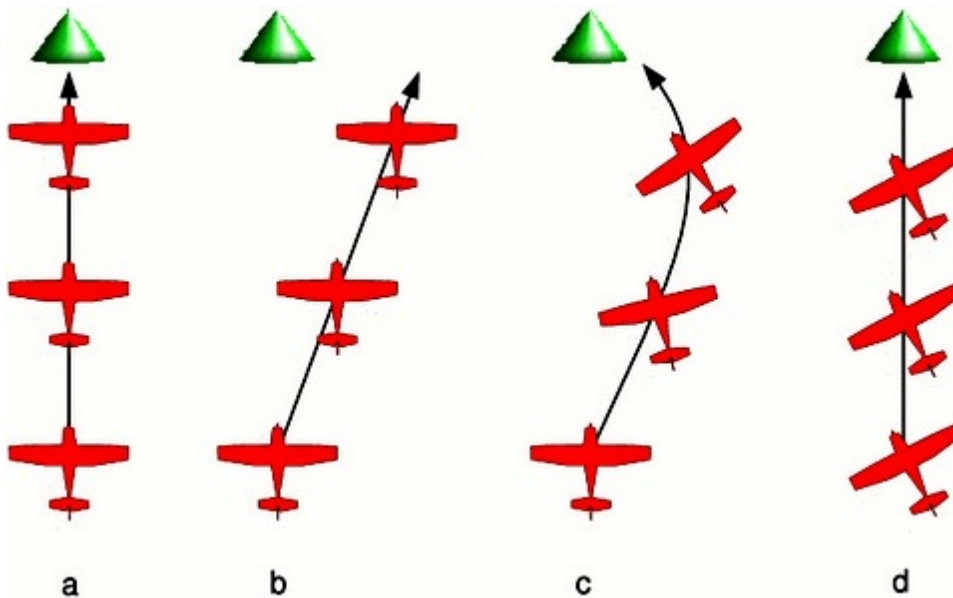
When the wind blows in the same direction as you fly, the speed of the wind adds itself to the airspeed of the plane. Hence you move faster compared to the ground. You will arrive earlier at your destination.

When the wind blows in the opposite direction (towards the nose of the plane), the speed of the wind subtracts itself from the airspeed of the plane. Hence you move slower compared to the ground. You will arrive later at your destination and have more time to enjoy the landscape.

The two cases above are quite simple. More complex is when the wind blows towards the side of the airplane. Consider the diagram below.

- On picture (a) there is no wind. The pilot wants to reach the green hill situated to the North. He heads for the hill, towards the North, and reaches the hill

- after a while. When there is no wind, you just head towards your destination and everything's fine.
- On picture (b), the pilot keeps heading to the North. Yet there is wind blowing from the left ; from the West. The airplane drifts to the right and misses the hill.
 - On picture (c), the pilot keeps heading towards the hill. This time he will arrive at the hill. Yet the plane flies a curved path. This makes the pilot lose time to get to the hill. Such a curved path is awful when you need to make precise navigation.
 - Picture (d) shows the optimal way to get to the hill. The plane is directed to the left of the hill, slightly towards the West and the wind. That way it compensates for the wind and remains on a straight path towards the hill.



How much to the left or to the right of the object must you head? At what angle? Serious pilots use geometry and trigonometric computations to calculate the correct angle. You need no computations at all to fly roughly straight. The trick is to choose an aiming point in the direction you wish to fly, then observe how it moves. You will become aware if you are drifting leftwards or rightwards. Then let your instinct slowly head the plane to the right or to the left to compensate the obvious drift. To begin with, you may need to think about what you are doing. Very soon this will become automatic, just like when you learned to fly straight. You will no more keep the plane headed towards the object. You will rather keep it flying towards the object.

The faster the flight airspeed compared to the wind speed, the less correction you will need.

8.9.1 Crosswind Take Off

Taking off when the wind is coming from the side is tricky. Airport designers avoid this by placing runways so that they face into the prevailing wind. Often airports have multiple runways, placed such that there will be a runway facing straight into wind as much of the time as possible.

Taking off with a wind blowing straight towards the nose of the aircraft makes life easier as it is the speed of the wing relative to the air that causes lift. When there is no wind, the aircraft must accelerate to 55 knots to take off. However, if there is a 10 knot head-wind, the aircraft has an airspeed of 10 knots standing still and only has to accelerate to 45 knots relative to the ground to take off. This shortens take-off distances.

Just as a headwind shortens take-off, a tail-wind increases take-off length. Anything more than a knot or two makes a huge difference to take-off distance. As (most) runways can be flown from either end, you can easily take off from the other end of the runway and benefit from the headwind.

The main way to know the wind direction and speed is to go to the control tower or ask the control tower by radio. A necessary and complementary tool are the windsocks at both ends of the runway. They show the wind direction and speed. The longer and the stiffer the windsock, the more wind there is. The windsock on the picture below shows an airspeed of 5 knots :



Unfortunately, sometimes there isn't a runway facing the wind, and you have to take off when the wind is blowing from the side.

The technique is as for a normal take-off with two changes :

- During the take-off roll, the aircraft will try to “weather-cock” into wind. You must react by using the rudder to keep the aircraft running straight. You will have to apply the rudder at quite a strong angle to stay aligned with the runway. You will need to keep applying rudder throughout the take-off.
- As you take off, the aircraft will react to the rudder and try to turn. You will need to correct for this using the ailerons. Once the aircraft is in the air, you can reduce the rudder pressure and aileron, then correct for the wind, to keep aligned with the runway as described above.

8.9.2 Crosswind Landing

Landing in a crosswind is very similar to the take-off :

- Stay aligned with the runway by compensating for the crosswind.
- As you begin to round-out, use the yoke to straighten the aircraft so it is pointed down the runway. Apply rudder to stop the aircraft turning.
- The aircraft will land on one wheel first. Use the rudder to keep the aircraft pointed straight down the runway as the other wheel touches down.

The technique described here is the [slip landing](#). Another crosswind landing technique is the [crab landing](#).

8.9.3 Taxiing in the Wind

Under 10 knots wind the Cessna 172p seems not to need particular precautions when taxiing. Yet any sudden increase in wind speed can tilt it and tumble it over. So best apply the recommendations whenever there is wind.

To train taxiing on the ground when there is wind, configure the simulator for a strong wind, like 20 knots. Such a wind can tilt the plane and blow it away tumbling any moment. One single error during taxiing and the plane is lost.

Main rule is you must *push the yoke towards the wind*. This deserves some physical explanation :

- When the wind is blowing from 12 o'clock, this is quite logical. The yoke is pushed (towards 12 o'clock) and the elevator makes the tail rise a little. That's the most stable position to avoid the plane be tilted by the wind.
- When the wind comes from 10 o'clock, pushing the yoke towards 10 o'clock means that the elevator is almost neutral, while the left aileron is upward and the right aileron is downward. This pushes the left wing down and lifts the right wing. Again, that's the most stable position to avoid the plane be tilted by the wind.
- When the wind blows from 8 o'clock, you would think you should invert the position of the ailerons, to keep the left wing being pushed down. Hence you should push the yoke to 4 o'clock. Wrong ! Keep pushing the yoke to 8 o'clock. The reason is the downward position of the aileron on the right wing makes it act like a slat. This increases the lift on the right wing and this is all we want. Symmetrically, the upward position of the left aileron decreases the lift of the left wing.
- When the wind comes from the rear, from 6 o'clock, the yoke is pulled (towards 6 o'clock). The upward position of the elevator tends to make the tail be pushed down. Once again this is the best. Strong wind can push the tail against the ground. This is impressive but the tail is conceived to withstand this.

If you want to move towards the wind, you will need more engine power. When the wind blows from the rear you may need no engine power at all. Always keep the engine power to the minimum needed.

Especially when turning, move very slowly. Make little changes at a time. Take your time and closely survey the yoke angle. Constantly keep it pushed towards the wind. Constantly try to reduce the engine power. Keep in mind using the brakes too

firmly may shortly tilt the plane at an angle that allows the wind to tilt it and blow it away.

8.10 The autopilot

An autopilot is not an “intelligent” pilot. It just takes over simple tasks for the pilot. You still are the pilot aboard and have to keep aware of everything. Be prepared to shut the autopilot down as they often go wrong, both in real life, and in the simulator.

The autopilot is mounted to the right of the yoke :



Switch it on by pressing the **AP** button. The autopilot then controls the roll of the aircraft. It keeps the wings level with the horizon. This is displayed in the picture below by the “**ROL**” marking. To switch the autopilot off press **AP** again.



If you press the **HDG** button the autopilot will try to keep the plane flying towards the direction tuned on the directional gyro by the red marking (see section 8.7.6). “**HDG**” stands for “heading”. Press again on the **HDG** button to get back to roll control mode (or **AP** to switch the autopilot off).



The buttons **ALT**, **UP** and **DN** are used to tell the autopilot either to control the vertical speed **VS** or the altitude **ALT**.

For more advanced use of the autopilot, see the reference document for the autopilot modeled in the Cessna 172 :

https://www3.bendixking.com/servlet/com.honeywell.aes.utility.PDFDownloadServlet?FileName=/TechPubs/repository/006-18034-0000_2.pdf

8.11 What Next ?

This tutorial has introduced you to the basics of flight in the Cessna 172. From here you can explore the many features that *FlightGear* has to offer.

Once you have mastered the content of this tutorial, you may want to look at the other tutorials in this Manual, covering flying to other airports, flying using instruments when clouds obscure the ground, and flying helicopters.

This tutorial has skipped over a number of topics that a real-life pilot would have to consider :

- How to follow real checklists.
- How to make emergency landing on very short fields, after an engine failure.
- How to navigate with regard to the laws of the air, charts, laws, radio beacons and weather conditions
- How to create a flight plan and fly it accurately.
- How to place people, fuel and baggage in an airplane to get a correct center of gravity.
- How to deal with the control tower and with other airplanes.
- How to deal with several fuel tanks and their systems.
- How to deal with the failure of every possible part of the plane.

This tutorial has also not covered features of more advanced aircraft, including :

- retractable landing gear
- variable pitch propellers
- multiple engines
- jet engines.

8.12 Thanks

I wish to thank :

- Benno Schulenberg who corrected lots of mistakes in my English in this tutorial.
- Albert Frank who gave me key data on piloting and corrected technical errors.
- Vassilii Khachaturov who learned me new things about *FlightGear*.
- Roy Vegard Ovesen for pointing me to the official Autopilot Pilots Guide.
- Dene Maxwell for his solution for problems under Windows Me.
- Mark Akermann and Paul Surgeon for their remarks.
- Michael “Sam van der Mac” Maciejewski who made the translation in Polish and converted the tutorial for use in the manual.

- The *FlightGear* mailing list users for their hearty welcome.
- 4p8 webmaster my friend Frédéric Cloth for the web space used by this tutorial.

8.13 Flying Other Aircraft

I cross-checked all the data about the Cessna 172p, a pilot friend verified I did not write too much rubbish and I made numerous virtual test flights. This section contains less reliable data about other airplanes based on my experience in the simulator. You may find it useful as an introduction to those airplanes but bear in mind my only goal was to make flights that seem OK and acquire basic knowledge.

8.13.1 How to land the Cherokee Warrior II

The Cherokee Warrior II has some advantages upon the Cessna 172p. Thanks to its low wings it is far less sensitive to crosswind. Fully extended flaps are provide more braking and allow it to land on a much shorter distance.

Take off is the same as for the Cessna 172p in *FlightGear*. In real life their take off checklists are not exactly the same.

You have to get used to some minor differences of the Cherokee Warrior II for the landing :

- During the steady horizontal flight before landing, the trim must be pulled a little below neutral in order to get the yoke around neutral.
- The optimal tachometer RPM during landing is at a lower RPM than the tachometer green zone. Roughly, keep the needle vertical.
- Only put use two steps of flaps during landing. Don't decrease the engine throttle too much.
- If you remain at two flaps deployed during landing, the round-out and flar will be similar to the Cessna 172p. However, using the third set of flaps will slow the aircraft down dramatically. It will very quickly touch the runway then come to a near halt. Be prepared to lower the front wheel very soon. (It is possible to use the third flaps step during the descent towards the runway, instead of tuning the engine power down. Oscillating between two steps and three steps allows to aim the runway start. Yet keep two flaps steps and tune the engine seems easier. An interesting stunt is to fly stable till nearly above the runway start, then tune the engine to minimum and deploy three flaps steps. The plane almost falls to the runway. It's impressive but it works.)

In real life, an advantage of the Cessna 172p upon the Cherokee Warrior II is the fuel reservoirs of the Cessna are located in the wings close above the center of the plane and higher than the engine. What's more an automatic system switches between the reservoirs. That means you almost don't have to bother for the way the fuel gets to the engine in flight. On the contrary, on the Cherokee Warrior II the reservoirs are located separately, on both wings and lower than the engine.

That means you have to constantly switch between the two reservoirs in flight. Should one reservoir become much lighter than the other, this would destabilize the airplane. The fact the reservoirs are lower than the engine means you have to control the fuel pumps and the backup fuel pumps.

Some links :

- http://en.wikipedia.org/wiki/Piper_Cherokee
- <http://www.alioth.net/flying/pa28-161/index.html>
- <http://freechecklists.net/Resources/Piper/PA-28-151+Warrior/>

8.13.2 How to take off and land the Piper J3 Cub

The Piper J3 Cub is a very different airplane from the Cessna 172p and the Cherokee Warrior II. The Cessna 172p and the Cherokee Warrior II are nose-wheel airplanes, while the Piper J3 Cub is a tail wheel airplane. Take off and landing with tail wheel airplanes is more difficult. You have to tightly use the rudder pedals when rolling over the runway. The yoke often needs to be pulled backwards to the maximum. The Piper J3 Cub is a good introduction to tail-wheel aircraft and it is quite easy to take off and land provided you follow an appropriate procedure. Stall speed seems to be a little below 40 mph (the airspeed indicator is in mph) (about 27 knots). Take-off is below 50 mph.

My take off procedure for the Piper Cub is to fully pull the yoke backwards then throttle the engine to maximum. Once the front wheels clearly rises from the ground, gently push the yoke back to neutral, towards a normal flight close above the runway. Let the plane accelerate to 50 mph. Then pull the yoke to keep a little more than 50 mph while rising in the air.

The landing procedure is quite different to that of 172, as the aircraft is very light, and has no flaps.

1. Fly at say 500 feet constant altitude and "exactly" 52 mph speed towards the runway. Let the engine cover eat up the runway start. The engine cover will hide the runway completely. To see where the runway is, push the yoke/mouse very shortly then stabilize again in normal flight.
2. Once the runway start matches with the set of instruments (if you could see through the instrument panel), reduce the throttle to a near minimum and begin the dive towards the runway start. Keep 52 mph using the yoke. Add some throttle if you are going to miss the runway edge. (Keep in mind just a little wind is enough to change things a lot for the Piper J3 Cub).
3. Make the rounding and pull the throttle to minimum. Do not pull steadily on the yoke. Instead let the wheels roll on the runway immediately.
4. Once the wheels roll on the runway, *push* firmly on the yoke, to its maximum. This rises the tail in the air. You would think the propeller will hit the runway or the airplane will tilt over and be damaged. But everything's fine. The wings are at a strong negative angle and this brakes the plane. (Don't push

the yoke this way on other airplanes, even if their shape seems close to that of the Piper J3 Cub. Most of them will tumble forwards.)

5. The yoke being pushed in to its maximum, push the left mouse button and keep it pushed to go in rudder control mode. Keep the plane more or less centered on the runway. This is quite uneasy. One tip is to stop aiming the rudder to say the left already when the plane just starts to turn to the left.
6. Once the speed is really low (and the rudder control stabilized), you will see the tail begins to sink to the ground. Release the left mouse button to go back to yoke control. Pull the yoke backwards completely, to the other extreme. The tail now touches the ground and the nose is high up. Now you can use the wheel brakes (**b**). (If you use the brakes too early, the plane nose will hit the ground.)

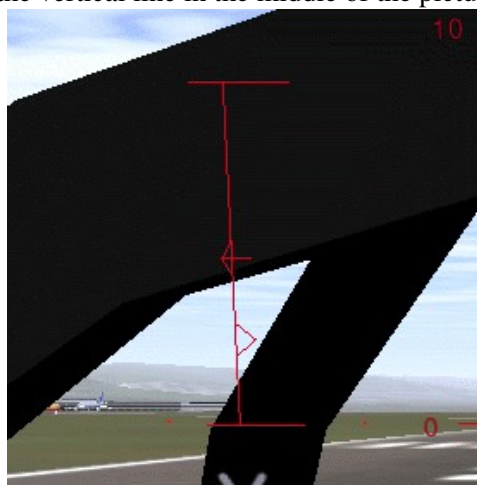
The take off procedure mentioned above is symmetrical to the first landing procedure. There exists a second take off procedure, symmetrical to the second landing procedure. Yet I don't succeed it properly so I won't write about it.

8.13.3 How to take off and land a jet

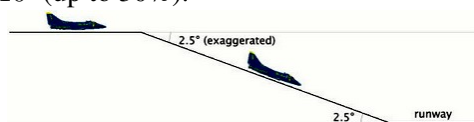
Take off on a jet is easy but you must have fast reflexes. My favorite jet on FlightGear is the A-4 Skyhawk. You get it with the `-aircraft=a4-uiuc` parameter on Linux, provided it is installed.

This is the "calm" procedure to take off :

- Ask for a red and full HUD by typing **h** two times. The engine throttle indicator is the leftmost on the HUD.
- The airspeed indicator is the one labeled "KIAS" on the upper left side of the instrument panel. You can also use the airspeed indicator on the HUD, of course.
- Tune in $\frac{1}{2}$ engine power.
- Keep the yoke pulled in $\frac{1}{2}$ of its total way (picture below : the red arrow on the right side of the vertical line in the middle of the picture).



at the dotted red lines labeled “-10”. A feature on the ground situated there is situated 10° below the ideal horizon. In other words : when you look to objects “hidden” by the lines labeled “0”, you have to lower your eyes of 10° to look at objects "hidden" by the dotted lines labeled “-10”. This implies, and it is very important, that a person in a rowboat, “hidden” by the dotted lines labeled “-10”, has to rise his eyes up 10° to look at your plane. He sees you 10° above the horizon. In the picture above, the start of the runway is situated at 64% of the way towards the red “-10” dotted lines. That means you have to lower your eyes of $6,4^\circ$ to look at the runway start. This also means that if you start now to descent towards the runway start, the descent path will be of $6,4^\circ$ (too steep). So, the HUD allows to measure precisely the angle of the descent path. On a jet plane you need an angle of $2,5^\circ$ (up to 3°), that is 25% of -10° (up to 30%).



- Once descending towards the runway start, aim at it using the yoke/mouse. And keep 150 knots speed using the engine throttle lever.
- Keep measuring the angle between the ideal horizon and the runway start. It must keep $2,5^\circ$ (that is 25% of 10°) :
 - o If the angle increases above $2,5^\circ$, you are above the desired path and you must loose altitude faster. Both decrease the engine power and dive the nose a little.
 - o If the angle decreases below $2,5^\circ$, you are under the desired path. I wouldn't say you should gain altitude, rather you should loose altitude less fast. Both add a little engine power and rise the nose a little.
- Once very close to the runway start, do no rounding. Don't pull steadily on the yoke like you would for the Cessna 172p. Simply let the plane touch the ground immediately, at high speed. Let it smash on the runway, so to say. All three wheels almost together. Just throttle the engine down to minimum. (If you try to pull steadily on the yoke and hover over the runway while the plane nose rises steadily, on a F-16 you would scrape the plane rear and probably destroy it.)
- Keep the key **b** down to brake and use the rudder to stay aligned with the runway. Make only very little tunings with the rudder, otherwise the plane will tumble on one of its sides.

The HUD in a real jet contains a symbol to show towards what the airplane is moving. It is shown in the picture below. When you are flying at constant altitude, that symbol is on the ideal horizon line. Once you dive towards the runway start, you simply have to place that symbol on the runway start. This is quite an easy and precise way to aim at the runway start. (The diamond in the center of the FlightGear HUD sometimes can help but it does not have the same purpose. It shows towards what the airplane nose is pointing. For example is you descent towards the ground

at low speed, the symbol would be somewhere on the ground while the FlightGear diamond will be up in the sky.) (By the way, the HUD on the virtual B-52 on FlightGear has that symbol. It is great to use while landing.)



Also, a real HUD shows a dotted line at $-2,5^\circ$, to help find the correct descent path. Simply keep that dotted line on the runway threshold.

In addition to airspeed, military fast jet pilots rely on using the correct angle of attack during approach. The Angle Of Attack (AoA) is the angle at which the wings are pitched against the relative airflow. The advantage of keeping to an optimal AoA is that the optimal AoA for landing does not depend on the plane load, while the optimal airspeed speed does. By ensuring that the AoA is correct for every landing, you will land at the correct speed, whatever the plane load.

The Angle of Attack is displayed within the HUD, and/or as a set of three lights shown below. When the upper ∇ is lit, your angle of attack (AoA) is too high and you need to pitch down. When the lower \wedge is lit, your AoA is too low and you need to pitch up. The center \circ indicates your the AoA is OK. Obviously, as you pitch up or down your airspeed and descent rate will change, so you will need to change your throttle setting appropriately.



The Cessna 172 and the A-4 Skyhawk are two extremes. Most other airplanes are in-between these two extremes. If you trained them both (and one or two tail wheel airplanes), you should be able to find out how to take off and land most other airplanes.

160 knots seems an appropriate landing speed for the F-16 Falcon. Also you need to throttle down the engine to minimum just before the plane should touch the runway. Otherwise it will hover over the runway. Don't bother for the flaps. It seems they are deployed automatically with the landing gear. (Read the section [8.7.4](#) about the stall).

140 up to 150 knots and all 8 flaps steps deployed seem appropriate to land the virtual Boeing 737. But don't trust me especially on that one. I just made a few experiments and didn't search for serious data. The landing speed varies a lot depending on the plane load, I suppose 140 knots is for a plane with no load. The

Boeing 737 seems to like a gentle rounding before the wheels touch the runway. Start the rounding early.

In the take off procedure for the Cessna 172 and the A-4 Skyhawk I recommend you pull the yoke/mouse/elevator to $\frac{1}{2}$ the total way, from the start on. This seems to be a bad practice on the Pilatus PC-7. Keep the elevator neutral. Let the plane accelerate and wait till the speed gets over 100 knots. Then pull calmly on the yoke. During landing, deploy full flaps once you start plunging to the runway but don't decrease the engine throttle. Decrease it only when the hovering above the runway starts. 100 knots seems a good landing speed.

For the Cessna 310 too you better leave the elevator neutral during the acceleration on the runway. The plane will raise its nose by its own provided you deployed one flaps step. (If you keep the yoke pulled from the start on, the nose will rise sooner and you will get yawful yaw problems.)

(Some virtual airplanes, like some big airliners or fast aircraft, need faster physical computations. Then add the `-model-hz=480` parameter to the command line. If the plane is difficult to control during landings, try this.)

The angle at which you land a Cessna 172p is far steeper than the narrow $2,5^\circ$ for a jet. Nevertheless you are allowed to land the Cessna at a narrow angle too. (Provided the terrain around the runway allows for this, of course.) If you have passengers who have ears problems with the variation of air pressure. . .

8.13.4 How to take off and land the P-51D Mustang

Should you ever get a chance to pilot a [P-51 Mustang](#), just say no. It is quite dangerous to take off and land. That's the kind of airplane you fly only when your country is in danger. You need a lot of training. Yet once in the air the P-51 Mustang seems no more dangerous to its pilot than other common military airplanes. It is quite easy to pilot.

At low and medium altitude the P-51 wasn't better than the Spitfire and the Messerschmitts. The big difference was at high altitude. The P-51 kept efficient and maneuverable while enemy fighters were just capable to hang in the air. This was an advantage at medium altitude too because the P-51 was able to plunge towards enemy airplanes from high altitude. Another key difference was the P-51 is very streamlined. Hence it was capable to fly much further than the Spitfire. These two differences let the P-51 Mustang fulfill its purpose : escort Allied bombers all the way to their targets in Germany. This allowed the bombings to be much more efficient and contributed to the defeat of the Nazis.

To get the P-51D Mustang on Linux use the `-aircraft=p51d` command line parameter.

To take off the P-51D Mustang in *FlightGear*, deploy one flaps step, pull and keep the yoke completely backwards, push the engine throttle to maximum and keep the left mouse button pressed to control the rudder and keep on the runway. Once you reach exactly 100 mph, suddenly push the rudder 1/3 of its total way to the right. Immediately release the left mouse button and push the yoke to rise

the tail (don't push it too much, as the sooner the wheels leave the ground the better). From now on, keep the left mouse button released. Only make very short adjustments to the rudder. Let the plane rise from the runway and get to altitude at a speed of say 150 mph. Don't forget to retract the landing gear and the flaps.

Don't make too steep turns. You would lose control on the plane and crash.

To land, deploy full flaps and lower the landing gear from the start on. 130 mph speed seems fine, up to 140 mph. Make an approach from 1,000 feet altitude and a dive at a low angle, like for a jet. Once over the runway, shut the engine down completely (key `{`). Don't hover over the runway. Get the wheels rolling soon (like for a jet). Hold the left mouse button down to steer the plane using the rudder. Once the tail sinks in, briskly pull the yoke (left mouse button shortly released) to force the tail on the runway. Go on steering the plane using the rudder. Now the tail is firmly on the ground, use the brakes if you want.

8.13.5 How to take off and land the B-52 Stratofortress

The B-52F bomber implemented in *FlightGear* is a success. It is one of my favorite airplanes. I'm sorry it was conceived to terrify me. One single B-52 bomber can wipe out every main town of my country and rise a nightmare of sicknesses and children malformation for centuries. All B-52 bombers united can wipe out mankind and almost every kind of plants and animals on Earth.

The differences between the virtual B-52F bomber and the Cessna 172p are these :

- The B-52F starts with the flaps deployed and the parking brakes set.
- There are only two flaps steps : retracted and deployed. When deployed they are only meant to make the wings lift more, not to brake. If you want to brake, you need the spoilers. They deploy on the upper side of the wings. Use the key `k` to deploy the spoilers and the key `j` to retract them. There are seven steps of spoilers.
- The main landing gear of the Cessna 172p is composed of two wheels, one on each side of the airplane. In order for these wheels to leave and touch the ground altogether, you need to keep the wings parallel with the ground. The main landing gear of the B-52F is composed of a set of wheels at the front and a set of wheels at the rear. This implies that in order for these wheels to leave and touch the ground altogether, you need to keep the airplane body parallel with the ground.

This is my procedure to take off the virtual B-52F :

- *Push* the yoke $\frac{1}{3}$ of the total way.
- Push the engine throttle to maximum.
- Release the parking brakes (key `B`).
- Push down the left mouse button to control the rudder pedals and keep the airplane on the runway
- The whole runway length is needed till the B-52F rises from the ground (KSFO).

- Once the B-52F leaves the ground, around 190 knots seems appropriate to get to altitude.
- Retract the flaps and the landing gear.

To land, the B-52F's HUD offers that great airplane-shaped symbol I talked about in the section about jets. So you just have to put that symbol on the airplane threshold (a few pixels further seems optimal) and keep the runway start 2,5° below the ideal horizon line. 130 up to 140 knots seems a good landing speed. (Instead of the speed you can make use of the AOA indicator displayed on the schematic instrument panel (**P**).). Simply keep the AOA at 3°. I must confess I prefer to tune the speed rather than the AOA.) If the plane gets to the runway at 130 up to 140 knots, simply "let it smash" on the runway. Otherwise, if the speed is higher, make a rounding and a short hover. The brakes seem to be very effective **b**). They allow to stop the B-52F on roughly the same short runway length as the Cessna 172p.

Replays of the flights are a delight. They allow to check the plane body left the runway and landed back parallel with it. One of the points of view is situated inside the B-52F rear turret, which allows you to be your own passenger and to compare what you see with what you experienced as a passenger in airliners. The key **K** allows to visualize the airplane trajectory.

To cause an accident with the B-52 do this :

- Make a steep turn with a very strong bank ; the wings nearly perpendicular to the ground.
- Try to get the plane back level. It will obey but very slowly. You will get aware that the turn will go on for a while and that you will turn further than your intended flight direction.
- Do something that accelerates the stabilization on some airplanes : push the rudder to an extreme, opposite to the current turn. This will suddenly make the airplane drop from the sky.

Chapitre 9

A Cross Country Flight Tutorial

9.1 Introduction



FIGURE 9.1 – Flying over the San Antonio Dam to Livermore

This tutorial simulates a cross-country flight from Reid-Hillview (KRHV) to Livermore (KLVK) under Visual Flight Rules (VFR). Both airports are included in the standard FlightGear package, so no additional scenery is required.

I'll assume that you are happy taking off, climbing, turning, descending and landing in FlightGear. If not, have a look at the tutorials listed above. This tutorial is designed to follow on from them and provide information on some of the slightly more complicated flight systems and procedures.

9.1.1 Disclaimer and Thanks

A quick disclaimer. I fly microlights rather than Cessnas in real life. Most of this information has been gleaned from various non-authoritative sources. If you find an error or misunderstanding, please let me know. Mail me at stuart_d_buchanan-at-yahoo.co.uk.

I'd like to thank the following people for helping make this tutorial accurate and readable : Benno Schulenberg, Sid Boyce, Vassilii Khachaturov, James Briggs.

9.2 Flight Planning

Before we begin, we need to plan our flight. Otherwise we'll be taking off not knowing whether to turn left or right.

First, have a look at the Sectional for the area. This is a map for flying showing airports, navigational aids, and obstructions. There are two scales of sectionals for VFR flight - the 1 :500,000 sectionals themselves, and a number of 1 :250,000 VFR Terminal Area Charts which cover particularly busy areas.

They are available from pilot shops, or on the web from various sources. You can access a Google-map style interface here :

<http://www.runwayfinder.com/>

Simple search for Reid-Hillview. An extract from the chart is shown in Figure 9.2.

If you want a map of the entire area showing exactly where the plane is, you can use Atlas. This is a moving-map program that connects to FlightGear. See Section 6.3 for details.

So, how are we going to fly from Reid-Hillview to Livermore ?

We'll be taking off from runway 31R at KRHV. KRHV is the ICAO code for Reid-Hillview airport, and is shown in the FlightGear wizard. (It is marked on the sectional as RHV for historic reasons. To get the ICAO code, simply prefix a 'K'.)

The 31 indicates that the magnetic heading of the runway is around 310 degrees, and the R indicates that it's the runway on the right. As can be seen from the sectional, there are two parallel runways at KRHV. This is to handle the large amount of traffic that uses the airport. Each of the runways can be used in either direction. Runway 31 can be used from the other end as runway 13. So, the runways available are 13R, 13L, 31R, 31L. Taking off and landing is easier done into the wind, so when the wind is coming from the North West, runways 31L and 31R will be in use. The name of the runway is written in large letters at the beginning and is easily seen from the air.

Once we take off we'll head at 350 degrees magnetic towards Livermore (KLVK). We'll fly at about 3,500ft above sea-level. This puts us at least 500ft above any terrain or obstructions like radio masts on the way.

We'll fly over the Calaveras Reservoir then the San Antonio Reservoir. These are both large bodies of water and we can use them as navigation aids to ensure we stay on the right track.

Once we get about 10 miles out of Livermore (above the San Antonio Reservoir), we'll contact the Livermore Air Traffic Control (ATC) to find out where we should land. We'll then join the circuit and land.

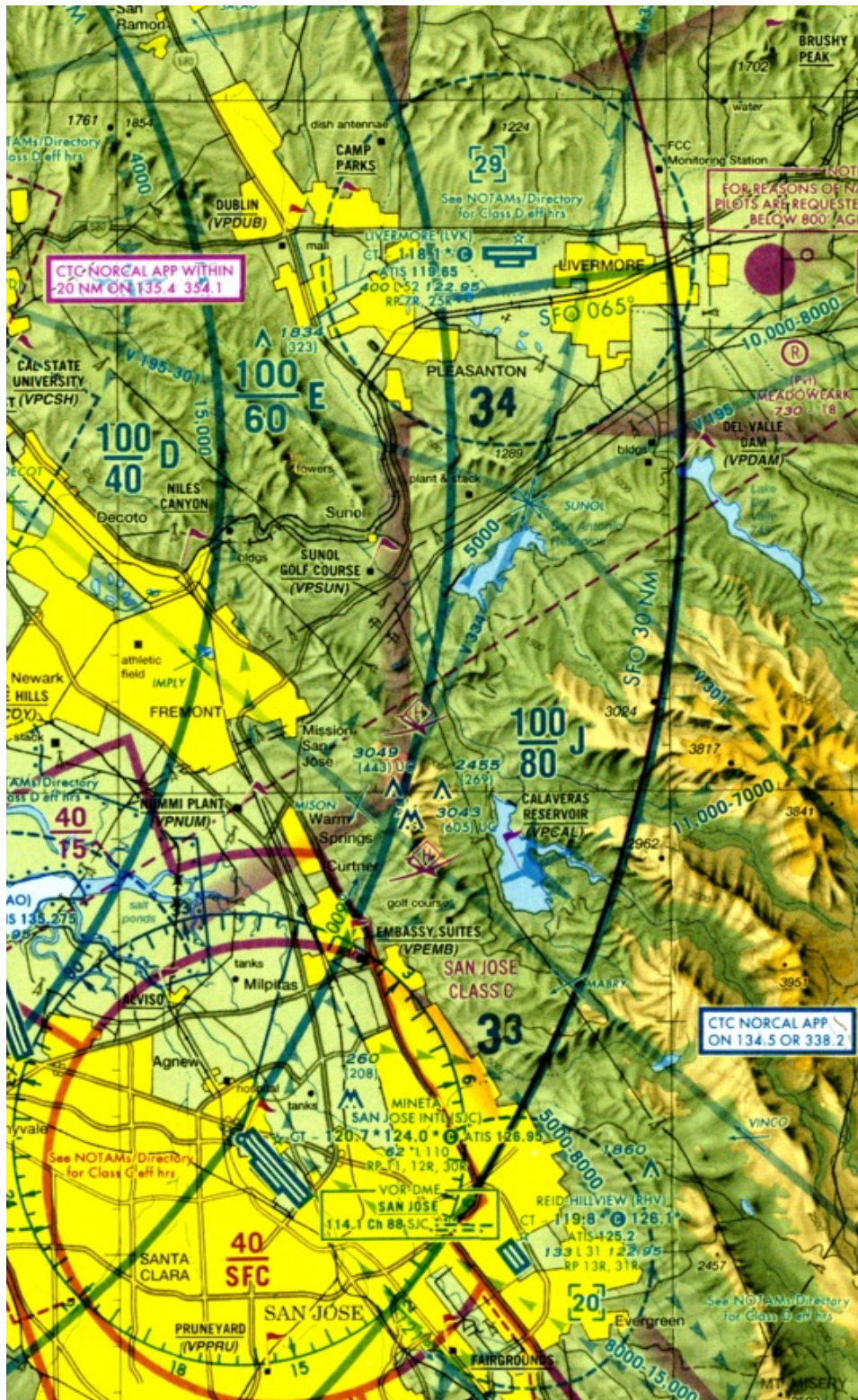


FIGURE 9.2 – Sectional extract showing Reid-Hillview and Livermore airports

9.3 Getting Up

OK, we know where we're going and how we'll get there. Time to get started.

Start FlightGear using the Wizard (or command-line if you prefer). We want to use a C172P and take off from runway 31R at Reid-Hillview of Santa Clara County (KRHV). Dawn is a nice time to fly in California.

If you want, you can fly in the current weather at KRHV by clicking the Advanced button on the final screen of the Wizard, selecting Weather from the left-hand pane, selecting 'Fetch real weather' and clicking OK.



FIGURE 9.3 – On the runway at KRHV

9.3.1 Pre-Flight

Before we take off, we need to pre-flight the aircraft. In the real world, this consists of walking around the aircraft to check nothing has fallen off, and checking we have enough fuel.

In our case, we'll take the opportunity to check the weather, set our altimeter and pre-set things that are easier to do when you're not flying.

The weather is obviously important when flying. We need to know if there is any sort of cross-wind that might affect take-off, at what altitude any clouds are (this is a VFR flight - so we need to stay well away from clouds at all times), and any wind that might blow us off course.

We also need to calibrate our altimeter. Altimeters calculate the current altitude indirectly by measuring air pressure, which decreases as you ascend. However, weather systems can affect the air pressure and lead to incorrect altimeter readings, which can be deadly if flying in mountains.

9.3.2 ATIS

Conveniently, airports broadcast the current sea-level pressure along with useful weather and airport information over the ATIS. This is a recorded message that

is broadcast over the radio. However, to listen to it, we need to tune the radio to the correct frequency.

The ATIS frequency is displayed on the sectional (look for 'ATIS' near the airport), but is also available from within FlightGear. To find out the frequencies for an airport (including the tower, ground and approach if appropriate), use the ATC/AI menu and select Frequencies. Then enter the ICAO code (KRHV) into the dialog box. The various frequencies associated with the airport are then displayed. Duplicates indicate that the airport uses multiple frequencies for that task, and you may use either.

Either way, the ATIS frequency for Reid-Hillview is 125.2MHz.

9.3.3 Radios

We now need to tune the radio. The radio is located in the Radio Stack to the right of the main instruments. There are actually two independent radio systems, 1 and 2. Each radio is split in two, with a communications (COMM) radio on the left, and a navigation (NAV) radio on the right. We want to tune COMM1 to the ATIS frequency.



FIGURE 9.4 – The C172 communications stack with COMM1 highlighted

The radio has two frequencies, the active frequency, which is currently in use, and the standby frequency, which we tune to the frequency we wish to use next. The active frequency is shown on the left 5 digits, while the standby frequency is shown on the right. We change the standby frequency, then swap the two over, so the standby becomes active and the active standby. This way, we don't lose radio contact while tuning the radio.

To change the frequency, click on the grey knob below the standby frequency (highlighted in Figure 9.5), just to the right of the 'STBY'. Using the left mouse button changes the number after the decimal place, using the middle button changes the numbers before the decimal place. Click on the right side of the button to change the frequency up, and the left of the button to change the frequency down. Most of the FlightGear cockpit controls work this way. If you are having difficulty clicking on the correct place, press Ctrl-C to highlight the hot-spots for clicking.



FIGURE 9.5 – COMM1 adjustment knob



FIGURE 9.6 – COMM1 switch

Once you have changed the frequency to 125.2, press the white button between the words ‘COMM’ and ‘STBY’ to swap the active and standby frequencies (highlighted in Figure 9.6). After a second or so, you’ll hear the ATIS information.

9.3.4 Altimeter and Compass



FIGURE 9.7 – Altimeter calibration knob

Listen for the ‘Altimeter’ setting. If you are not using ‘real weather’, the value will be 2992, which is standard and already set on the plane. If you are using ‘real weather’, then the altimeter value is likely to be different. We therefore need to set the altimeter to the correct value. To do this, use the knob at the bottom left of the altimeter (circled in red in Figure 9.7), in the same way as you changed the radio frequency. This changes the value in the little window on the right of the altimeter, which is what you are trying to set, as well as the altitude displayed by the altimeter.

The other way to set the altimeter is to match it to the elevation above sea-level of the airport. The elevation is listed on the sectional. For KRHV it is 133ft. This means you can double-check the pressure value reported over ATIS.



FIGURE 9.8 – Heading adjust knob

We will also take the opportunity to set the heading bug on the compass to 350 - our bearing from KRHV to KLVK. To do this, use the red button on the compass housing (highlighted in Figure 9.8), just as you've done before. Use the left mouse button for small adjustments, and middle mouse button to make big adjustments. The value of 350 is just anti-clockwise of the labeled value of N (North - 0 degrees).



FIGURE 9.9 – Take-off from KRHV

9.3.5 Take-Off

OK, now we've done that we can actually take off!. In my case this usually involves weaving all over the runway, and swerving to the left once I've actually left the ground, but you'll probably have better control than me. Once above 1000ft, make a gentle turn to the right to a heading of 350 degrees. As we've set the heading bug, it will be easy to follow. We're aiming for a fairly prominent valley.

Continue climbing to 3,500 ft at around 500-700 fpm. Once you reach that altitude, reduce power, level off to level flight and trim appropriately. Check the power again and adjust so it's in the green arc of the RPM gauge. We shouldn't run the engine at maximum RPM except during take-off.

9.4 Cruising

OK, we've taken off and are on our way to Livermore. Now we can make our life a bit easier by using the autopilot and our plane more fuel efficient by tuning the engine. We'll also want to check we're on-course

9.4.1 The Autopilot

We can make our life a bit easier by handing some control of the aircraft over to 'George' - the autopilot.

The autopilot panel is located towards the bottom of the radio stack (highlighted in Figure 9.10). It is easily distinguishable as it has many more buttons than



FIGURE 9.10 – The C172 Autopilot

the other components on the stack. It can work in a number of different modes, but we are only interested in one of them for this flight - HDG. As the names suggest, HDG will cause the autopilot to follow the heading bug on the compass, which we set earlier.

To set the autopilot, press the AP button to switch the autopilot on, then press the HDG button to activate heading mode. While the autopilot is switched on, it will use the trim controls to keep the plane on the heading. You can change the heading bug, and the autopilot will maneuver appropriately. However, the autopilot doesn't make any allowances for wind speed or direction, it only sets the heading of the airplane. If flying in a cross-wind, the plane may be pointed in one direction, but be travelling in quite another.

You should use the trim controls to keep a level flight. You can use the autopilot for this, but it is a bit more complicated.

Once the aircraft has settled down under the autopilot's control, we can pay more attention to the outside world and higher level tasks.

9.4.2 Navigation



FIGURE 9.11 – The Calaveras Reservoir

As we noted above, we're going to be travelling over a couple of reservoirs. When you leveled off, the first (Calaveras) was probably right in front of you. You can use them to check your position on the map. If it looks like you're heading off course, twist the heading bug to compensate.

9.4.3 Mixture



FIGURE 9.12 – The Calaveras Reservoir

As altitude increases, the air gets thinner and contains less oxygen. This means that less fuel can be burnt each engine cycle. The engine in the C172 is simple and doesn't automatically adjust the amount of fuel to compensate for this lack of oxygen. This results in an inefficient fuel burn and a reduction in power because the fuel-air mixture is too 'rich'. We can control the amount of fuel entering the engine every cycle using the mixture control. This is the red lever next to the throttle. By pulling it out, we 'lean' the mixture. We don't want the mixture too rich, nor too lean. Both these conditions don't produce as much power as we'd like. Nor do we want it perfect, because this causes the fuel-air to explode, rather than burn in a controlled manner, which is a quick way to trash an engine.

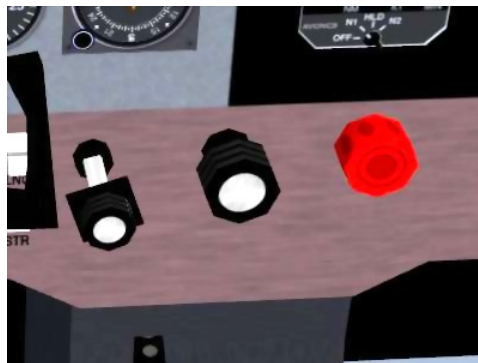


FIGURE 9.13 – Mixture Control

The mixture is controlled by the red lever to the right of the yoke. You may need to pan your cockpit view to see it.

To pan the cockpit view, right-click with the mouse-button within the Flight-Gear window until the cursor becomes a double-headed arrow. Moving the mouse now pans the view. Once you can see the mixture lever clearly, right-click again to change the mouse back to the normal arrow.



FIGURE 9.14 – Fuel Flow and EGT gauges

Pull the mixture lever out slowly (use Ctrl-C to see the hot spots), leaning the mixture. As you do so, you'll see various engine instruments (on the left of the panel) change. Fuel flow will go down (we're burning less fuel), EGT (Exhaust Gas Temperature) will go up (we're getting closer to a 'perfect mixture') and RPM will increase (we're producing more power). Pull the mixture lever out until you see the EGT go off the scale, then push it in a bit. We're now running slightly rich of peak. While at 3,500ft we don't need to lean much, at higher altitudes leaning the engine is critical for performance.

9.5 Getting Down

Once you reach the second reservoir (the San Antonio Reservoir), we need to start planning our descent and landing at Livermore. Landing is a lot more complicated than taking off, assuming you want to get down in one piece, so you may want to pause the simulator (press 'p') while reading this.

9.5.1 Air Traffic Control

In the Real World, we'd have been in contact with Air Traffic Control (ATC) continually, as the bay area is quite congested in the air as well as on the ground. ATC would probably provide us with a 'flight following' service, and would continually warn us about planes around us, helping to avoid any possible collisions. The FlightGear skies are generally clear of traffic, so we don't need a flight following service. If you want to change the amount of traffic in the sky, you can do so from the AI menu.

Livermore Airport is Towered (towered airports are drawn in blue on the sectional), so we will need to communicate with the tower to receive instructions on how and where to land.

Before that, we should listen to the ATIS, and re-adjust our altimeter, just in case anything has changed. This is quite unlikely on such a short flight, but if flying hundreds of miles it might make a difference. To save time when tuning radios, you can access the Radio Settings dialog from the Equipment menu. The Livermore ATIS frequency is 119.65MHz.

An ATIS message also has a phonetic letter (Alpha, Bravo, . . . Zulu) to identify the message. This phonetic is changed each time the recorded message is updated. When first contacting a tower, the pilot mentions the identifier, so the tower can double-check the pilot has up to date information.

Besides the altitude and weather information, the ATIS will also say which runway is in use. This is useful for planning our landing. Normally, due to the prevalent Westerly wind, Livermore has runways 25R and 25L in use.

Once you've got the ATIS, tune the radio to Livermore Tower. The frequency is 118.1MHz. Depending on the level of AI traffic you have configured on your system, you may hear Livermore Tower talking to other aircraft that are landing or departing. This information is not played over the speakers, it is only displayed on the screen.

Once the frequency goes quiet, press the ' key. This will bring up the ATC menu. Click on the radio button on the left to select what you wish to say (you only have one option), then OK.

Your transmission will be displayed at the top of the screen. It will indicate who you are (type and tail number), where you are (e.g. 6 miles south), that you are landing, and the ATIS you have.

After a couple of seconds, Livermore Tower will respond, addressing you by name and telling you what runway to use, which pattern is in use and when to contact them, for example

“Golf Foxtrot Sierra, Livermore Tower, Report left downwind runway two five left.”

To understand what this means, we'll have to describe the Traffic Pattern.

9.5.2 The Traffic Pattern

With the number of aircraft flying around, there have to be standard procedures for take-off and landing, otherwise someone might try to land on-top of an aircraft taking off.

The Traffic Pattern is a standard route all aircraft must follow when near an airport, either taking off or landing. The traffic pattern has four stages (or 'legs'), shown in Figure 9.15. The 'downwind' mentioned above refers to one of these, the one with the number 3.

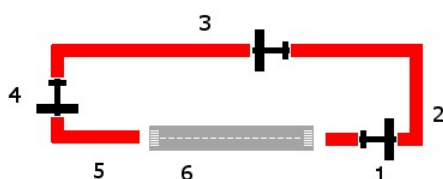


FIGURE 9.15 – The Traffic Pattern

1. Aircraft take off from the runway and climb. If they are leaving the airport, they just continue climbing straight ahead until clear of the pattern and then do whatever they like. If they are returning to the runway (for example to practise landing), they continue climbing until they reach a couple of hundred feet below 'pattern altitude'. This varies from country to country, but is usually between 500ft and 1000ft Above Ground Level (AGL). This is called the *upwind* leg.
2. The pilot makes a 90 degree left-hand turn onto the *crosswind* leg. They continue their climb to 'pattern altitude' and level out.
3. After about 45 seconds to a minute on the crosswind leg, the pilot again makes a 90 degree left turn onto the *downwind* leg. Aircraft arriving from other airports join the pattern at this point, approaching from a 45 degree angle away from the runway.
4. When a mile or so past the end of the runway (a good guide is when the runway is 45 degrees behind you), the pilot turns 90 degrees again onto the *base* leg and begins the descent to the runway, dropping flaps as appropriate. A descent rate of about 500fpm is good.
5. After about 45 seconds the pilot turns again onto the *final* leg. It can be hard to estimate exactly when to perform this turn. Final adjustments for landing are made. I usually have to make small turns to align with the runway properly.
6. The aircraft lands. If the pilot is practising take-offs and landings, full power can be applied and flaps retracted for takeoff, and the aircraft can take off once more. This is known as 'touch-and-go'.

Most patterns are left-handed, i.e. all turns are to the left, as described above. Right-hand patterns also exist, and are marked as 'RP' on the sectional. ATC will also advise you what pattern is in use.

9.5.3 Approach

We're approaching Livermore airport from the South, while the runways run East/West. Due to the prevailing Westerly wind, we'll usually be directed to either runway 25R or 25L. 25R uses a right-hand pattern, while 25L uses a left-hand



FIGURE 9.16 – Sectional extract showing approaches to Livermore

pattern. Both the patterns are illustrated in Figure 9.16. Depending on the runway we've been assigned, we'll approach the airport in one of two ways. If we've been asked to land on runway 25R, we'll follow the blue line in the diagram. If we've been asked to land on runway 25L, we'll follow the green line.

We also need to reduce our altitude. We want to end up joining the pattern at pattern altitude, about 1,000ft above ground level (AGL). Livermore airport is at 400 ft above sea-level (ASL), so we need to descend to an altitude of 1400 ASL.

We want to begin our maneuvers well before we reach the airport. Otherwise we're likely to arrive too high, too fast, and probably coming from the wrong direction. Not the best start for a perfect landing :).

So, let's start descending immediately.

1. First switch off the autopilot by pressing the AP switch.
2. Return mixture to fully rich (pushed right in). If we were landing at a high airport, we'd just enrich the mixture slightly and re-adjust when we reached the pattern.
3. Apply carb-heat. This stops ice forming when the fuel and air mix before entering the cylinder, something that can often happen during descent in humid air. The carb-heat lever is located between the throttle and mixture. Pull it out to apply heat.
4. Reduce power quite a bit. Otherwise we might stress the airframe due to over-speeding.

5. Drop the nose slightly to start the descent.
6. Trim the aircraft.

Use your location relative to the airport and the two towns of Pleasanton and Livermore to navigate yourself to the pattern following the general guide above.

Once you're established on the downwind leg, you'll need to report to ATC again. Do this in the same way as before. They will then tell you where you are in the queue to land. 'Number 1' means there are no planes ahead of you, while 'Number 9' means you might want to go to a less busy airport! They'll also tell you who is ahead of you and where. For example 'Number 2 for landing, follow the Cessna on short final' means that there is a single aircraft in front of you that is currently on the final leg of the pattern. When they land and are clear of the runway, they'll tell ATC, who can then tell you 'Number 1 for landing'.

9.5.4 VASI



FIGURE 9.17 – On Final at Livermore with VASI on the left

Once on final, you'll notice two sets of lights on the left of the runway (enhanced in Figure 9.17). This is the VASI and provides a nice visual clue as to whether you're too low or too high on approach. Each set of lights can either be white or red. White means too high, red means too low. White and red together means just perfect. On a Cessna approaching at 60kts, a descent rate of about 500fpm should be fine. If you are too high, just decrease power to increase your descent rate to 700fpm. If you are too low, increase power to decrease your descent rate to 200fpm.

9.5.5 Go Around

If for some reason it looks like you're going to mess up the landing you can abort the landing and try again. This is called a 'Go Around'. To do this

1. Apply full power
2. Wait until you have a positive rate of climb - i.e. your altitude is increasing according to the altimeter.

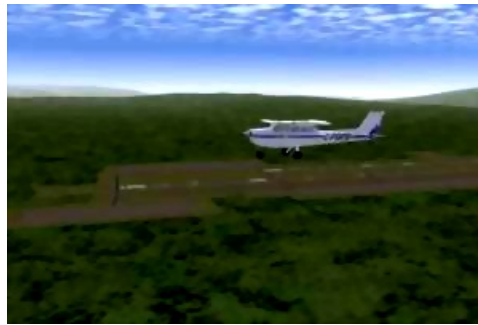


FIGURE 9.18 – Missed approach at Livermore

3. Raise your flaps to 10 degrees (first-stage).
4. Tell ATC you are ‘going around’
5. Climb to pattern height
6. If you aborted on final approach, continue over the runway to re-join the pattern on the crosswind leg. If on base, fly past the turn for final, then turn and fly parallel to the runway on the opposite side from downwind to rejoin on the crosswind leg.
7. Fly the complete pattern, telling ATC when you are on downwind, and try again.

9.5.6 Clearing the Runway

Once you’re on the ground, you should taxi off the runway, then tell ATC you are clear. At high-altitude airports, you would lean the engine to avoid fouling the spark-plugs with an over-rich mixture. Find somewhere nice to park, shut down the engine by pulling mixture to full lean, then throttle off and magnetos to off (knob on the bottom left of the panel). Switch off the avionics master switch, tie down the aircraft, then go get that hamburger !

I hope this tutorial is of some use. If you have any comments, please let me know at [stuart_d_buchanan {at} yahoo.co.uk](mailto:stuart_d_buchanan@yahoo.co.uk).

Chapitre 10

An IFR Cross Country Flight Tutorial

10.1 Introduction

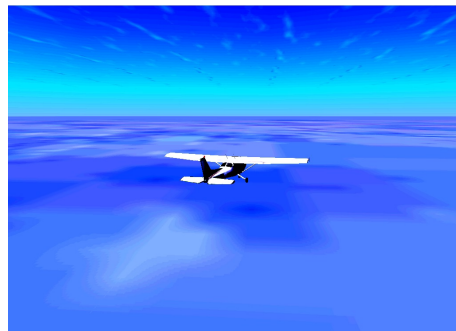


FIGURE 10.1 – Flying over the San Antonio Dam to Livermore. I think.

In the cross country flight tutorial, you learned about VFR flight, and in the course of the flight you were introduced to most of the flight instruments in the C172p. Now we're going to do an Instrument Flight Rules (IFR) flight. In this flight you'll be introduced to the remaining instruments, learn a bit about IFR flight, and learn many, many TLAs (Three-Letter Acronyms).

We'll fly the same flight, from Reid-Hillview (KRHV), runway 31R, to Livermore (KLVK), runway 25R, only this time we'll do it in IFR conditions : a ceiling 200 feet above ground level, and 800 meter visibility. This tutorial assumes you've completed the cross country flight tutorial.



FIGURE 10.2 – On runway 31R at KRHV

10.1.1 Disclaimer

This is not intended to teach you how to fly IFR. Rather, it is meant to give a flavour of what IFR flying is like, and remove the mystery of the panel instruments not covered by the cross country flight tutorial.

I'm not a pilot. Like the previous tutorial, this information has been gleaned from various non-authoritative sources. If you find an error or misunderstanding, please let me know. Mail me at bschack-flightgear -at- usa -dot- net.

10.2 Before Takeoff

We need to tell FlightGear about our flight conditions. There are different ways to set our “desired” weather, but we'll use the global weather menu. After launching FlightGear, click **Environment** ⇒ **Global Weather** to bring up the weather dialog. Then click on **METAR Source** and select **CAT I minimum**.

This gives us a low ceiling and reduced visibility. Unfortunately, it also gives us rather stiff winds. If you don't want to deal with them, then you can easily turn off the winds :

- Click on **METAR Source** and select **Manual input**.
- In the METAR string at the bottom, change “15015KT” (15 knot winds coming from 150°) to “15000KT” (0 knot winds coming from 150°), and hit **Apply**.

10.2.1 Flight Planning

After your weather settings settle down, you'll see something like Figure 10.2. Those clouds don't look very friendly, and it's hard to even see past the end of the runway. Maybe we should just *drive* there in the Cessna. We had been planning to practice ground steering anyway . . .

So how do you get from A to B when you can't see? There are a variety of ways that have evolved over the years, with various advantages and disadvantages. Our flight will use all of the navigation instruments the standard Cessna C172p has, just to give a taste of what's possible.

Our entire route, and the aids we'll be using, are shown in Figure 10.3. Our route is in green, the navigational aids blue and red. The route looks a bit crazy — in fact, you might wonder if we're *more* lost using our fancy equipment than just flying by the seat of our pants — but there is a method to the madness. Rather than overwhelming you with details by explaining it all now, I'll explain it bit by bit as we go along.

10.2.2 VHF Omnidirectional Range

The first bit will involve VOR¹ (VHF (Very High Frequency) Omnidirectional Range) navigation, and will get us to a point about 5 nm (nautical miles) south of Livermore.

VOR stations are indicated on the sectional by a big bluish-green circle with compass markings around the outside. I've helped you by marking their centers with a big blue dot as well. Reid-Hillview is very close to one, San Jose, which you can see in Figure 10.3. Near the centre of the circle, in a bluish-green rectangle, is the station information. According to the station information, it's a VOR-DME station (I'll explain DME later), its name is San Jose, its frequency is 114.1 MHz (or Channel 88, which is an alternative way to say the same thing), and its identifier, or "ident", is SJC (which in Morse code is $\cdots \cdot\text{---} \text{---}\cdot$).

To tune into a VOR station, we use one of the NAV receivers, which are paired with the COMM receivers (see Figure 10.4). And we navigate using the corresponding VOR gauge. We'll choose the NAV1 receiver (and VOR1 gauge) in this case (NAV2 would have worked just as well). Before setting the frequency, check out the VOR1 gauge. It should look like VOR1 on the left in Figure 10.5. The important thing is the red "NAV" flag. That means there's no VOR signal, so we can't trust the gauge.

The NAV receiver has an active frequency, a standby frequency, and a tuning knob, just like the COMM receiver.² Tune it to 114.1, and press the swap button. If you look at VOR1, you should notice that the red "NAV" flag has disappeared,

NAV1 \Rightarrow 114.1³

1. See http://en.wikipedia.org/wiki/VHF_omnidirectional_range for more information.

2. Operation of the COMM receivers was covered in the cross country flight tutorial.

3. All important actions and events will be given in the margin. This should provide a nice summary of the flight, uncluttered by the verbiage of the text.

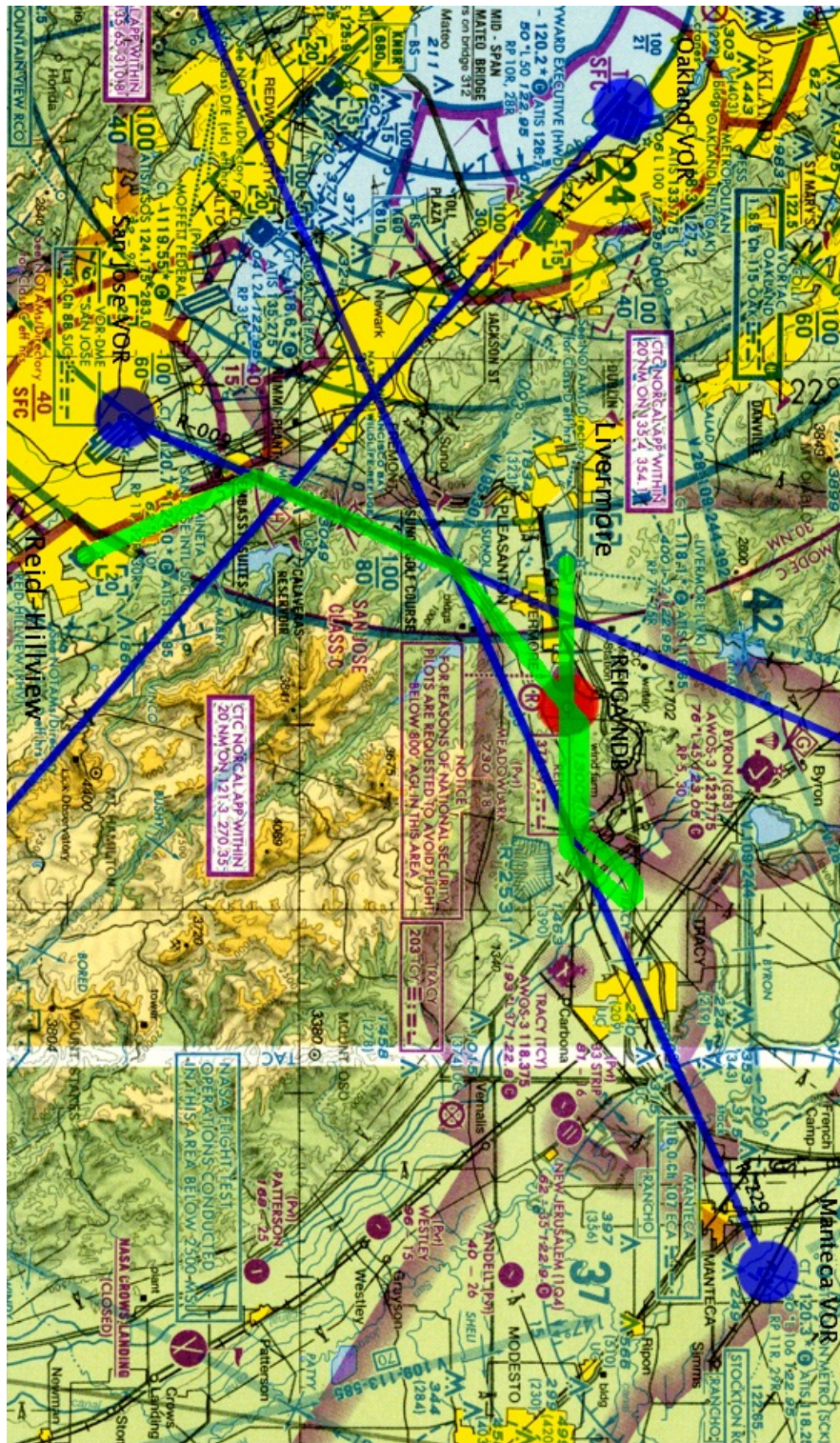


FIGURE 10.3 – Green : our route, Blue : VORs and radials, Red : NDBs



FIGURE 10.4 – IFR navigation Instruments

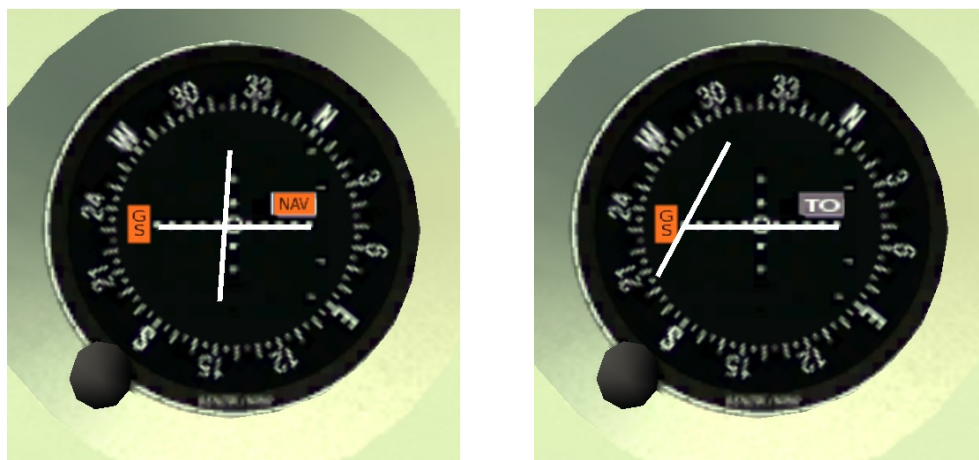


FIGURE 10.5 – VOR1, before and after tuning



FIGURE 10.6 – Ident volume knob and hot-spots

to be replaced with a “TO” flag, as shown on the right of Figure 10.5. That means we’re receiving a signal. But is it the correct one? What if we accidentally set the wrong frequency?

To confirm that we’re tuned into the correct VOR, we listen for its ident. If you can’t hear the ident, or if it doesn’t match the chart, don’t trust the needle. So far, you probably haven’t heard a thing. That seems strange. After all, the red “NAV” flag turned off. We must be in range of *some* station, right? Why can’t we hear anything?

When in doubt, look for the simplest solution. In this case, check the volume. There’s a little knob on the NAV receivers, with “PULL IDENT” written beside it (see Figure 10.6). That’s our ident volume control. It’s pointing down and to the left in the picture, which means the volume is off. Click on the right side of the ident knob to increase the volume. *Now* you should hear the ident : ··· ·---- ·-···. Phew.

Back to VOR1. There's a knob on the lower left, called the OBS (Omni Bearing Selector). As the name vaguely suggests, it is used to select a bearing. If you turn it, you should see the vertical needle, called the CDI (Course Deviation Indicator) move.⁴ Try to center the needle. It should center when the little arrow at the top points to somewhere around 277. That number, and the TO flag means: "Flying at a heading of 277° will lead you directly *to* the station".

That's great, except, according to our route, we don't want to go *to* the station. We actually want to intercept the light blue line labelled "009°" (the "9 degree radial") coming *from* the station. How do we do that? Simple. Set the OBS to 9. When we fly across the radial, the needle will center, and the flag will say FROM. This tells us: "flying at a heading of 9° will lead you directly away *from* the station", which is what we want. At that point we'll turn right to a heading of 9°.

VOR1 OBS ⇒ 009

One final thing — set the heading bug on the directional gyro to our current heading (about 310°).

Heading bug ⇒ 310

10.2.3 Takeoff

Now we're ready to take off. It took a long time to get ready, but in fact there's really more that we should have done. For example, there are all the things mentioned in the previous cross country flight tutorial. And there are other IFR preparations that we should have made. But again, in the interests of not overwhelming your brains, I'm only feeding you a bare minimum of information, and feeding it in trickles. This brings us to the most important control you have — the 'p' key. Use this often, especially when a new concept is introduced.

Okay. Now take off, keeping a heading of 310° for now. Establish a steady rate of climb. We plan to climb to 4000 feet. There's just one problem though — those ugly-looking clouds are standing in our way.

Take off; climb on runway heading

10.3 In the Air

If this is your first attempt at IFR flight, you will find it impossible to fly once you enter the clouds. When you enter the clouds, you will be momentarily disconcerted by the lack of visual cues. "No matter," you then think. "I'll just keep things steady." In a few moments, though, you'll probably notice dials and needles spinning crazily, and without knowing it, you'll be flying upside down, or diving towards the ground, or stalling, or all three.

It takes practice to get used to flying without external visual clues, although it's a skill that you definitely *must* master if you want to fly IFR. For now though, we'll use "George", the autopilot, to make this part of flying easier.

4. The horizontal needle is used in ILS landings, which will be explained later.



FIGURE 10.7 – Autopilot after engaging

10.3.1 George I

Once you’ve established a steady rate of climb and heading, engage the autopilot by pressing the AP button. You should see “ROL” displayed on the left to show that it’s in “roll mode” — it is keeping the wings level. In the middle it will display “VS”, to show it is in “vertical speed” mode — it is maintaining a constant vertical speed. On the right it will *momentarily* display that vertical speed (in feet per minute). Initially, the value is your vertical speed at the moment the autopilot is turned on. In the case of Figure 10.7, the autopilot has set the vertical speed to 300 feet per minute.

When you engage the autopilot, CHECK THIS CAREFULLY. Sometimes the autopilot gets a very funny idea about what your current rate of climb is, like 1800 feet per minute. Our little Cessna cannot sustain this, and if the autopilot tries to maintain this (and it will), you will stall before you can say “Icarus”. This is a bug, to be sure, and a bit annoying, but it is also a useful cautionary lesson — don’t put blind faith in your equipment. Things fail. You have to monitor and cross-check your equipment, and be prepared to deal with problems.

We want a vertical speed of around 500 to 700 feet per minute. Hit the up and down (UP and DN) buttons to adjust the vertical speed to a nice value. Take into account the airspeed as well. We want a sustainable rate of climb.

Engage autopilot; set vertical speed; engage heading mode

Finally, once you’re climbing nicely, hit the heading (HDG) button. On the display, “ROL” will change to “HDG”, and the autopilot will turn the airplane to track the heading bug. Since you set the heading bug to the runway heading, and you took off straight ahead (didn’t you ?), it shouldn’t turn much.

10.3.2 MISON Impossible

It’s around 8 nm to the 009 radial intercept, so we’ve got a bit of time. Since there’s no scenery to admire (eg, see Figure 10.8), we might as well prepare for the next phase of the flight.

If you look along our route, just after we intercept the 009 radial and turn north, we pass by a point labelled MISON (see Figure 10.9 for a closeup of that section of the chart without my fat blue and green lines drawn on top. MISON is in the lower right). Just above and to the left of MISON are two crossed arrows. MISON is an intersection. We’re actually going to pass east of MISON, but the radial passing roughly from northwest to southeast through MISON (and our route) is of interest



FIGURE 10.8 – Typical IFR scenery

to us. We're going to use it to monitor our progress.

Noting our passage of that radial isn't strictly necessary — we can just keep flying along the 009 radial from San Jose until we need to turn. But it's useful for two reasons : First, it's nice to know exactly where we are. Second, it confirms we are where we think we are. If we fly and fly and never cross the radial, alarm bells should start going off.

Looking at the sectional, we see that the radial is the 114 radial from the Oakland VORTAC (VOR TACAN, where TACAN stands for Tactical Air Navigation). Oakland's frequency is 116.8, and its ident is OAK (--- ·- ·-·-). NAV2 should already be tuned to Oakland, but if it isn't, do it now. Turn on NAV2's volume and make sure you're getting the correct ident.

NAV2 ⇒ 116.8

We need to adjust the OBS, to tell VOR2 which radial we're interested in. Set the OBS to 114.⁵ See if you can guess whether the flag should read TO or FROM when we cross the 114 radial. And see if you can guess whether the needle will move from left to right or right to left as we cross the radial.

VOR2 OBS ⇒ 114

A final note : For our purposes, there's nothing magical about the 114 radial — we could have used 113, or 115, or 100, or 090. The reason I chose 114 is because there was a line on the map already drawn along the 114 radial, which saved me the trouble of drawing a line myself.

10.3.3 George II

As we continue towards the 009 radial intercept, let's look a bit more closely at the autopilot. First of all, if you aren't in the habit of trimming the airplane, you'll probably notice a flashing "PT" with an arrow on the autopilot. The autopilot is telling you to adjust the pitch trim. I tend to ignore it because, flying with a mouse, trimming is more trouble than it's worth. Those of you lucky people with yokes

5. If you get tired of clicking on the knobs, much of this can be done more easily using the **Equipment** ⇒ **Radio Settings** dialog.

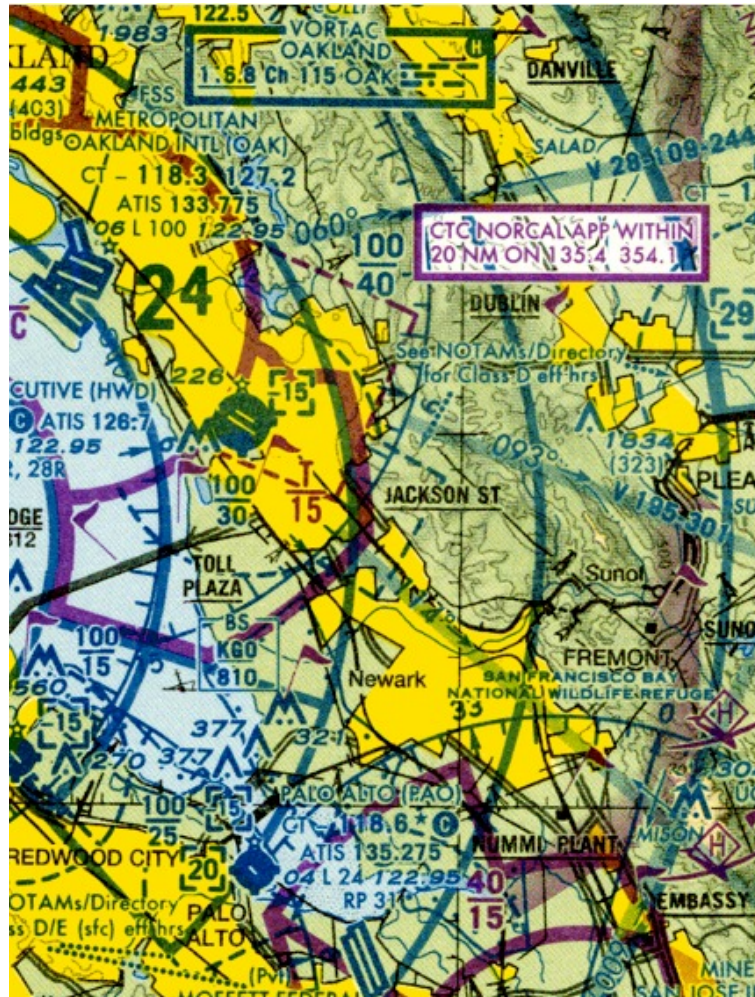


FIGURE 10.9 – Oakland VOR and 114 radial to MISON intersection



FIGURE 10.10 – Autopilot with altitude armed

and joysticks and who find flashing lights annoying might want to trim to get rid of it.

Also, on the right there's a big knob, the altitude select knob, which we can use to dial in a target altitude. We're going to use it. Turn it until you see our desired cruising altitude, 4000 feet, displayed on the right. When you started turning it, "ALT ARM" should have appeared in the autopilot display (as in Figure 10.10). This indicates that you've selected a target altitude. The autopilot will maintain the current rate of climb until reaching that altitude, at which point it will level off and change from vertical speed (VS) mode to altitude hold (ALT) mode. In altitude hold mode it maintains an altitude (in this case our target altitude of 4000 feet).⁶

Set autopilot altitude to 4000

Don't forget that the autopilot won't adjust the throttle, so when it levels out, the airplane (and engine) will speed up. You'll need to adjust the throttle to get a proper cruise.

10.3.4 Staying the Course

At some point you'll intercept the 009 radial (the VOR1 needle will centre). Turn to a heading of 009. You can do this using the heading bug on the directional gyro if you're using the autopilot.

Turn to 009° upon VOR1 intercept

Unless you're good or lucky, the needle probably won't be centered. We need to adjust our course. The CDI needle (the vertical needle on the VOR) tells us where to go. If it's to the left, that means the radial is to the left, so we need to go left. Ditto for right.

It's quite easy in theory, although in practice you may find that it's hard to keep the needle centered, and that you are slaloming down the radial. The key is to realize this: the *position* of the needle tells us where we *are*, the *motion* of the needle tells us what to *do*.

I'll explain. If the needle is to our left, then, yes, the radial is definitely to our left.⁷ But if the needle is *moving* towards us, that means we're going to cross the radial, sooner or later, so our situation is improving, and we probably just need to wait for the needle to center. On the other hand, if the needle is *moving* away, we need to turn towards it to stop, and reverse, its motion.

6. Of course, you don't really have to do this — you could just watch the altimeter, and when it gets to 4000 feet, reduce the vertical speed to 0, or press the ALT button to enter altitude hold mode. But by using the altitude select knob, we've demystified one more mystery button.

7. Unless you're heading in the opposite direction, but that's another story.

Note that the amount we need to turn is difficult to guess correctly at first, so experiment. Try 10° . If the needle moves too fast, cut it down to 5° (ie, turn back 5°). If, on the other hand, the needle moves too slowly, double it to 20° (ie, add another 10°), and see what happens.

10.3.5 Yet More Cross-Checks

Cross-checking your position is always a good thing. The intersection with the Oakland 114 radial is one way. Ahead of that lies the SUNOL intersection. If you look closely, 5 separate radials join at the point, so we have an embarrassment of choices with regards to the intersecting radial. Because it will come in useful later, we're going to use the one coming in from the upper right. Another check of the sectional reveals that this is the 229 radial of the Manteca VORTAC, 116.0 MHz, ident ECA (· ---· ---).

NAV2 \Rightarrow 116.0

VOR2 OBS \Rightarrow 229

You should know the drill by now : Tune NAV2 to 116.0, set the OBS to 229, and check the ident to confirm the station.

Meanwhile, let's introduce another piece of gear on the panel that will cross-check the SUNOL passage. Some VOR stations have a distance capability, called DME⁸ (Distance Measuring Equipment). For example, San Jose does (remember it's a VOR-DME station), as do Oakland and Manteca (VORTACs have DME capabilities).

Using DME, you can find out how far you are, in straight-line distance, from the VOR station. In our scenario, the DME isn't necessary, but we'll use it anyway, just to see how it works, and to reconfirm our position.

DME \Rightarrow N2

The DME is the instrument below the autopilot (refer to Figure 10.4). Make sure it's turned on. The selector to the left of the on/off switch is probably set to N1, where "N1" means "listen to NAV1". Since NAV1 is tuned to San Jose, it's telling us the distance to the San Jose VOR-DME. Switch the DME to N2. It now shows us the distance to the Manteca VOR.

The DME shows you 3 things : the distance in nautical miles to the station, your speed towards or away from the station, and estimated time to the station at the current speed. Note that the distance is the direct distance from your plane to the station (called the "slant distance"), not the ground distance. Note as well that the speed is relative to the station, so unless you're flying directly to or from the station, it will probably be lower than your true groundspeed. For example, the speed from San Jose, which is directly behind us, should be greater than the speed towards Manteca, which is off to the right.

If we look up information about the SUNOL intersection,⁹ it tells us that it is 33.35 nm (as measured by a DME receiver) from ECA on the 229.00 radial (that's what "ECAr229.00/33.35" means).

Now we have two ways to confirm the SUNOL intersection : The VOR2 needle will center, and the DME will read 33.4 or so. Note that the DME doesn't provide

8. See http://en.wikipedia.org/wiki/Distance_Measuring_Equipment for more information.

9. For example, from <http://www.airnav.com/airspace/fix/SUNOL>.

us with a very precise fix here because Manteca is at such an oblique angle. But it does give us a good warning of SUNOL's impending arrival. Moreover, if it has an unexpected value (like 30), it should raise a few alarm bells.

You may be wondering what "HLD" means (the setting between N1 and N2 on the DME). It stands for "hold", and means "retain the current frequency, regardless of whether NAV1 or NAV2 are retuned". For example, if we switch from N2 to HLD, the DME will continue to display (and update) information to Manteca. Even if we retune NAV2, the DME will remain tuned to Manteca. This is handy, because it basically represents a third independent receiver, and in IFR flight two receivers just never seem like enough.

10.4 Getting Down

We're getting close to SUNOL, flying along the 009 radial from San Jose, monitoring our position with the DME. At SUNOL we'll be less than 5 nm from Livermore, somewhere down there in the clouds. Perhaps if we just descended to 700 feet or so (Livermore is at 400, the ceiling is at 750) and headed more or less directly north after SUNOL, we'd get there? A recipe for disaster my friend, and you know it.

10.4.1 Instrument Approach Procedures

As you recall from the previous tutorial, when flying VFR, you don't just point your airplane to the nearest runway to land. You need to fly a pattern. This helps you line up, and helps prevent planes from crashing into one another, which is a Good Thing.

Similarly with IFR landings. There's a procedure to follow. In fact, there are *procedures* to follow. Because of the complexity of landing in IFR conditions, there's no single procedure for all airports. You need to check for your particular airport. In fact, you usually need to check for your particular airport, runway, and navigation equipment.

Our airport is Livermore (KLVK). Let's check the information for that airport. Go to <http://www.airnav.com/airport/KLVK> to see what they've got. Down near the bottom, we have IAPs (Instrument Approach Procedures). There are two listed for runway 25R. One is an ILS (Instrument Landing System) approach, the other a GPS (Global Positioning System) approach. Our plane has no GPS, but it does have ILS capabilities (I'll explain ILS later), so we'll choose that.

Although Livermore only has two different instrument approach procedures, big airports have many many more. If you look at nearby San Francisco, you'll see they have a *slew* of procedures. There are ILS procedures, GPS procedures, LDA procedures, VOR procedures, ... I wouldn't be surprised if they had a procedure for someone with a sextant and an hourglass in there. To learn IFR flight, you'll need to master all of them.

Back to Livermore. If you download the procedure, you'll see something like Figure 10.11 (except for the colour). It's pretty overwhelming at first — it compresses a lot of information in a small space. We'll ignore as much as we can, restricting ourselves to the three parts that have been coloured in. And we'll do those parts on a “need to know” basis — we'll only look at them when we really have to.

Where to start? At the beginning of course. An IAP will have one or more Initial Approach Fixes (IAFs). These are your entry points to the approach procedure and can be found in the “plan view”, which I've coloured purple in Figure 10.11. Our IAP lists two, one in the middle and one on the right (see Figure 10.12 for a close-up).

An IAF is a *fix*, and a *fix* is an identifiable point in space. In fact, we've already encountered another kind of *fix*, namely a VOR intersection. Fixes are also usually named (eg, MISON, SUNOL). The IAF on the right is named TRACY, and consists of a radial, a distance, and an altitude. Specifically, it's 15 DME (15 nm as measured by a DME receiver) along the 229 radial from the ECA (ie, Manteca) VOR.

10.4.2 Nondirectional Beacons

However, we're not going to use TRACY as our IAF. We're going to use the IAF in the middle, which is a marker (LOM stands for “Locator Outer Marker”). We'll worry about what an outer marker is later. For now let's concentrate on the locator part. The locator in an LOM is an NDB ¹⁰ (nondirectional beacon). It's a bit like a VOR, in that it can be used to determine your heading and navigate from place to place. Like a VOR, it has a name (REIGA, in this case), a frequency (374 kHz), and an ident (LV, or ···· ···· in Morse). NDBs also appear on sectionals, as fuzzy red circles with a small circle in the middle, with their identification information placed in a red box nearby. (see Figure 10.13 for a closeup. Don't confuse the NDB, which is fuzzy, with the solid red circle on the left, nor the circle below with the “R” inside).

An NDB station basically broadcasts a signal that says “I'm over here”, and the receiver on the plane can receive that signal and tell you, the pilot, “the station is over there”. You just need to tune the receiver and monitor the correct instruments. The receiver, labelled ADF (Automatic Direction Finder) Receiver, and the corresponding instrument, also labelled ADF, are shown in Figure 10.4.

To tune into REIGA, turn the tuning knob on the receiver until 374 is displayed as the standby (STDBY) frequency. As usual, use the middle mouse button for big changes (100 kHz in this case), and the left mouse button for small changes (1 kHz). Then hit the swap button (labelled “FRQ”). The 374 is now displayed as the selected (SEL) frequency. The needle on the ADF should swing around, eventually pointing ahead to the right, to REIGA. But it won't. Why? Because the receiver is

ADF ⇒ 374

10. See http://en.wikipedia.org/wiki/Non-directional_beacon for more information.

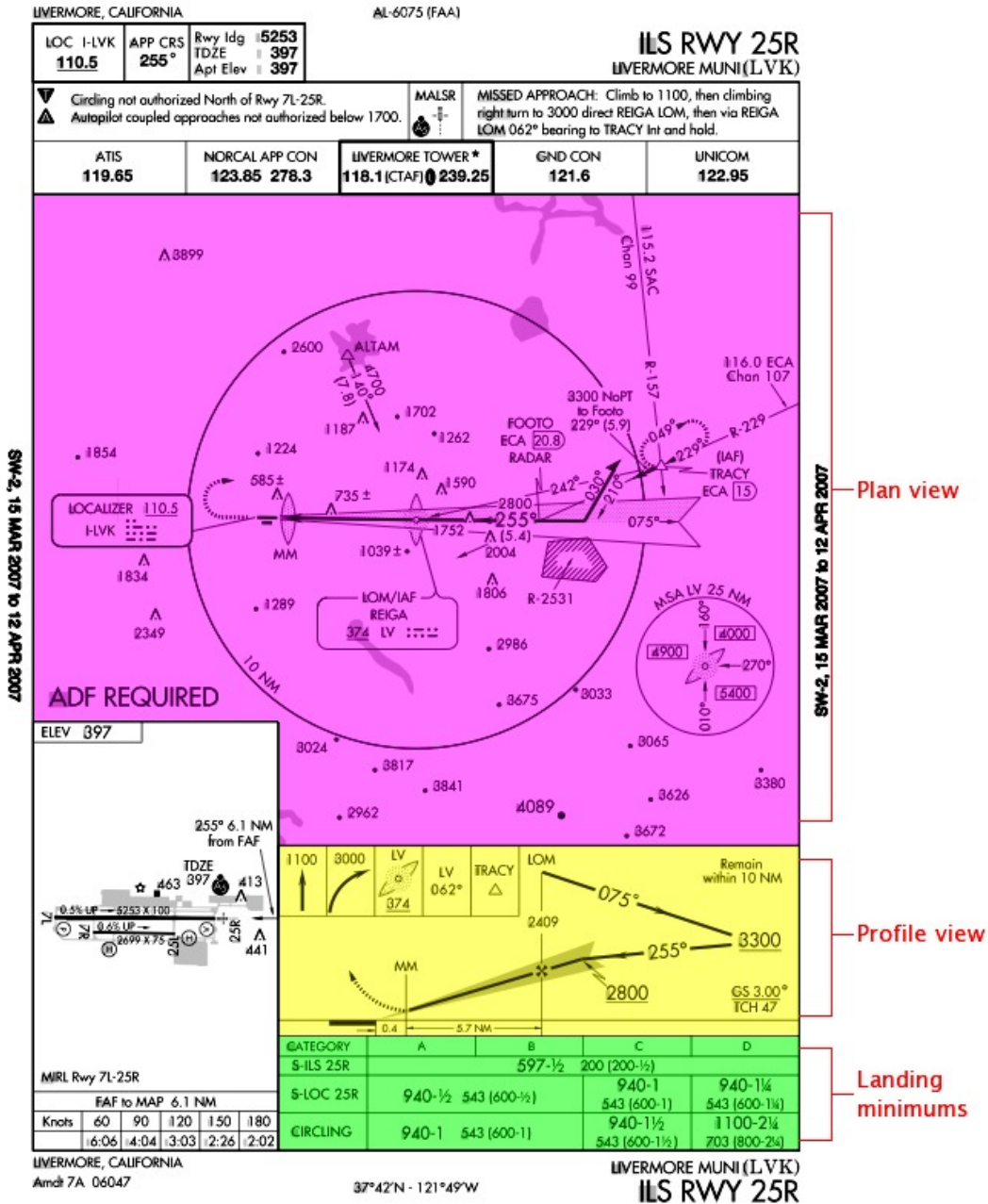


FIGURE 10.11 – ILS approach plate for Livermore runway 25R

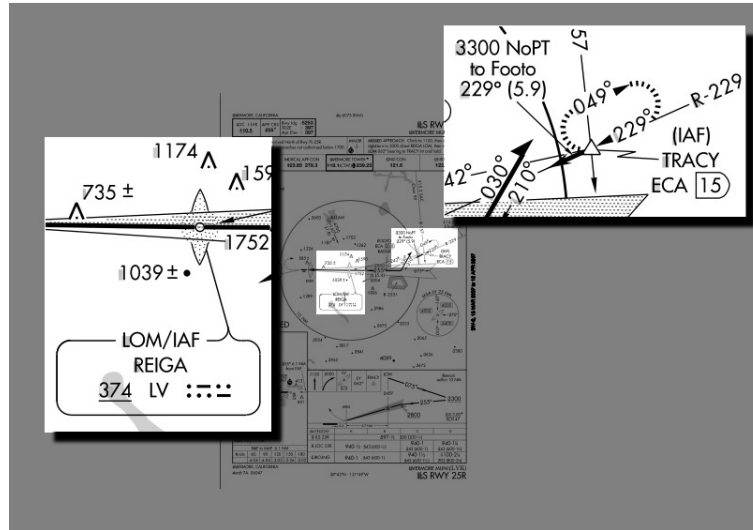


FIGURE 10.12 – Initial approach fixes



FIGURE 10.13 – REIGA nondirectional beacon

in antenna mode (as show by the “ANT” in the upper-left portion of the display).¹¹ Hit the ADF button so that “ADF” shows. Now the needle should swing to point to REIGA.

Like VORs, to be sure we’re really tuned into the right station, we need to hear the ident as well. Like the NAV receivers, the ADF receiver has a dedicated volume control, this one labelled “VOL”. The volume also functions as an on/off switch, so if you turn it all the way counter-clockwise, it will shut the receiver off. Unfortunately, at the time of writing, the ident function doesn’t work.

Notice there’s no OBS to set for an ADF — the needle just points to the station, which is nice. This leads us to our first rule for ADFs :

ADF Rule 1 : The needle points to the station.

Pretty simple. In fact, you may not think it merits a “rule”, but it’s important to emphasize the difference between ADFs and VORs. A VOR, remember, tracks a single radial, which you specify by turning the OBS. An ADF has a knob, and a identical-looking compass card, so it’s tempting to believe it acts the same way. It doesn’t. Turn the ADF heading knob (labelled “HD”) and see what happens. The compass card moves, but the arrow doesn’t. It just *points to the station*.

In our current situation, where we just want to fly to REIGA, that’s all we need to know to use the ADF. If the needle points “over there”, then we’ll fly “over there”, and eventually we’ll pass over REIGA. However, for the sake of practice, and because it will be necessary later, I’m going to give the second rule for ADFs, which explains what the compass card is there for :

ADF Rule 2 : *If* the compass card reflects our current heading, then the needle gives the bearing *to* the station.

In other words, the compass card gives “over there” a number.

Now we’re ready to head to REIGA. Rotate the ADF heading knob until our current heading is at the top (basically, the ADF should match the directional gyro). When we pass the SUNOL intersection, look at the ADF needle, and set the DG bug to that heading (I assume you’re using the autopilot. If not, just turn to that heading). At the end of the turn, the ADF needle should point straight ahead. And if it doesn’t, adjust your heading so that it does.¹²

Cross SUNOL; turn to REIGA

By the way, the closer you get to REIGA, the more sensitive the needle becomes to changes in your heading. Don’t go crazy trying to keep the needle centered as you get close. Maintain a steady heading, and get ready for the . . .

11. Antenna mode, by the way, is usually used to ident an NDB, because it gives better audio reception. While in antenna mode, however, the ADF will *not* point to the station — the needle will be parked pointing directly right.

12. Which is actually bad technique in the presence of a crosswind, but I’m ignoring the wind to simplify the tutorial.

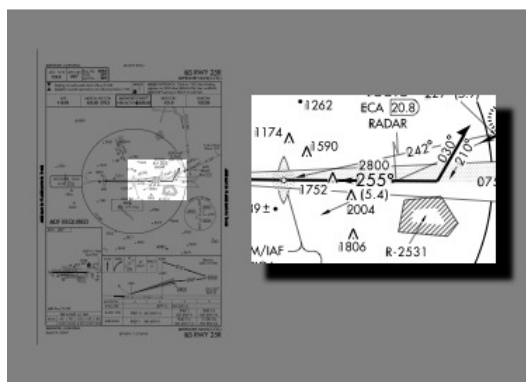


FIGURE 10.14 – Livermore ILS procedure turn

10.4.3 Procedure Turn

So, once we hit REIGA, do we just turn left and head down to the runway? Ah, if only life were so simple. No, we turn right, *away* from the airport, and do a *procedure turn*. We know there's a procedure turn because of the barbed arrow in the plan view (see Figure 10.14). As you can see if you follow the arrow, we need to fly away, on a heading of 075° , then turn left 45° to a heading of 030° . We do a U-turn (to the right, *away* from the airport — that's one of the rules about procedure turns) to come back at 210° , then a 45° right turn to 255° , heading straight towards the runway. All of this turning gives us time to set ourselves correctly on course, at the right altitude, to land on 25R.

Hmmm. I mentioned “right altitude”, but how do we know that is? That's down below, in the profile view (the yellow part of Figure 10.11). You can see that at the top is the LOM, our IAF. Now follow the arrows. After the IAF, we head out at 075° . During the procedure turn we can descend to 3300 feet, but *no lower* (that's what the line *under* the 3300 means). After we finish our procedure turn and are heading back at 255° , we can descend to 2800 feet, but *no lower*, until we intercept the glide slope.

One thing the instrument approach procedure does *not* tell you is the length of the procedure turn. The only constraint is that you must not fly more than 10 nm away from the NDB. You'll notice there's a 10 nm circle drawn around it in the plan view, and a note in the profile view saying “Remain within 10 NM”. They're not kidding. So, since we fly at around 110 knots, two minutes on each leg is reasonable — two minutes at 075° , and two minutes at 030° . On the way back we don't care about times — we just want to intercept 255° .

So, after we pass REIGA, turn right to 075° . Our ADF receiver has a built-in timer, so we'll use that to time our two-minute leg. Hit the “FLT/ET” (flight time/elapsed time) button. The “FRQ” in the middle of the display will disappear, “FLT” will appear on the right, and the standby frequency will be replaced by a



FIGURE 10.15 – ADF with timer running

time. This is the total flight time, and cannot be changed, except by cycling the power. Hit “FLT/ET” again. Now you’ll see “ET” displayed, and a time, probably the same as the flight time. To reset the elapsed time, hit the next switch, labeled “SET/RST”. The timer should reset to 0, then start counting up (see Figure 10.15).¹³ In elapsed time mode, each time you hit “SET/RST”, the time resets to 0. If you want to see the standby frequency again, hit “FRQ” once. The timers will continue to run.

10.4.4 Chasing the Needle

When we approached REIGA, we weren’t particularly concerned about our course — we just aimed for REIGA. Now, however, our course is important. We want to be flying directly away from REIGA *on a course of 075°*.

Now, in an ideal world, after we turned to 075°, the ADF needle would be pointed directly behind you (ie, we’d be on course). Probably it isn’t, so we need to adjust our course. The key to adjusting our course is ADF Rule 2. If we’ve set the compass card correctly, then the needle shows us the current NDB bearing. If we turn and fly until we intercept the 255 bearing, then turn to 075°, we’ll be right on course.

Figure 10.16 shows what I mean. In the figure, the plane, flying along the green line, is initially off course.¹⁴ The heading is correct, 075°, but the station is at 225°, not 255°. To correct this, we turn right (remembering to adjust the ADF compass card to match our new heading). As we fly on this new heading, we get closer to the correct position, crossing the 235 and 245 bearings (shown in red). Finally, when we the ADF needle points to 255°, we turn left to 075°, and readjust the ADF compass card.¹⁵ We are now on course.

Of course, even when you get back on track, that won’t be the end of the story. Your airplane drifts ; your mind drifts ; your compass drifts ; the wind pushes you around. What you find is that you will be constantly making little corrections. That’s okay, as long as we’re close. And anyway, before long (2 minutes actually), we’ll turn left 45° to 030° as part of our procedure turn, at which point we’ll just

Cross REIGA ; fly at 075° away from REIGA for two minutes

13. The timer can also be set to count down from a time you specify — except that feature has not yet been implemented.

14. Way off course, actually. I’ve exaggerated the angles to make the explanation clearer.

15. You might be thinking “Wouldn’t it be nice if there was an ADF where the compass card rotated automatically ?” Well, such an ADF does exist, and it comes with its own acronym — RMI (Radio Magnetic Indicator).

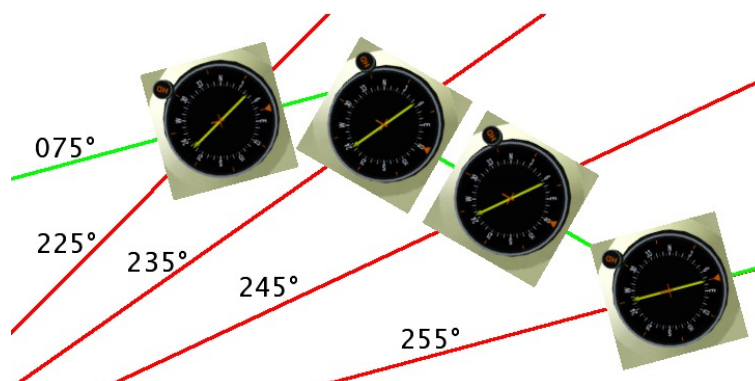


FIGURE 10.16 – Getting back on course

ignore the NDB anyway. Sigh. All that effort for just 2 minutes. Hardly seems worth it.

10.4.5 FOOTO Time

While you're flying outbound, take an occasional look at VOR2, tuned to Manteca, and the DME. Assuming the OBS is still at 229, and the DME still tuned to N2, at some point the needle should center, meaning you've crossed the 229 radial, and, if you're on course, at the same time the DME should read 20.8. How do I know that? If you look at the approach plate (Figure 10.11), you'll notice an intersection, named FOOTO. FOOTO is on the approach, and is defined to be 20.8 DME from ECA. Although this intersection is not strictly necessary for us, it comes for free, and provides good confirmation of our position both outbound and, later, inbound.

Depending on how fast you're flying, you'll probably pass FOOTO close to the time your two minutes at 075° are up. At the end of two minutes, turn left 45° to 030°. Reset the timer, and fly for another two minutes on this heading.

10.4.6 George III

Turn left to 030°; fly for two minutes while descending to 3300

This leg is relatively uneventful, so we'll take advantage of the lull in the action to descend to 3300. Assuming you're using the autopilot, you will need to do a few things :

1. If you're in altitude hold (ALT) mode, you need to get back into vertical speed (VS) mode. Press the ALT button — the "ALT" in the middle of the display should change to "VS", and your current vertical speed (probably 0) should be displayed momentarily on the right.
2. Click the DN button until you get a vertical speed of -500 feet per minute.

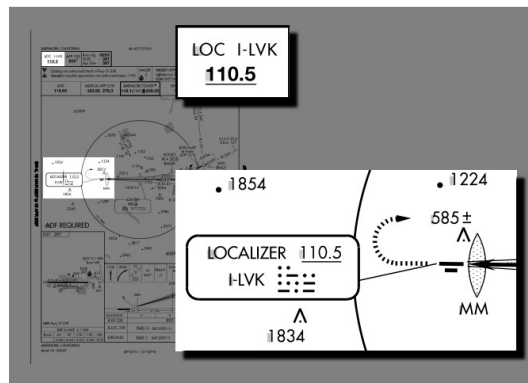


FIGURE 10.17 – Livermore 25R localizer

VOR1 OBS ⇒ 255

is basically a VOR with *one* radial, the approach heading. We don't care about any others, so we don't need an OBS to declare interest in any others. However, it does serve as a useful reminder, so move the OBS to 255, our desired heading.

10.4.8 Intercepting the Localizer

Turn right 180° to 210°
Intercept localizer

We're now ready to intercept the ILS localizer. When the two minutes on the 030° leg have passed, make your U-turn to the right to 210°. Soon after you complete your turn, the vertical (localizer) needle on the ILS will begin to move. And it will move *fast*, much faster than the ADF and VOR needles did. A localizer is 4 times as sensitive as a VOR, relatively small movements of the aircraft make big changes in the needles. You'll probably overshoot, but don't worry, because we have around 5 or 10 minutes to get things straightened out.

Just remember : don't chase the needles. That mantra is now more important than ever. Those needles are sensitive — if you just turn left when the localizer needle is to the left and right when it's to the right, you'll be flying like a drunken sailor. If you're lucky, the runway will be passing underneath you as you swing across the track for the umpteenth time. Luck, though, is something we should not be relying on. Determine on how the needles are moving before making your move.

Slow to 75 knots;
drop a notch of flaps;
descend to 2800

Now that you're heading back inbound at 255°, slow to 75 knots, drop a notch of flaps, and descend to 2800 feet (but no lower). And check for the inbound passage of FOOTO to confirm your position. And pat your head and rub your stomach.

10.4.9 Intercepting the Glide Slope

As we fly towards the runway, don't forget to look at the horizontal needle, the glide-slope needle. When we intercepted the localizer, it should have been high above us, because we were actually under the glide slope. As we levelled out at 2800, the glide slope started coming "down" to us. Eventually, you should see the

needle start to move down. When the needle is horizontal, that means you're on the glide slope.¹⁷ And, soon after we intercept the glide slope, we should pass over the outer marker. Several things will happen more or less simultaneously, all of which confirm your position :

1. You'll hear a continuous series of dashes.
2. The blue light labelled "O" above COMM1 will flash.
3. The ADF needle will swing around.

Once on the glideslope, we need to start descending. What's a good rate ? It depends on our groundspeed. In our case, we're going at 75 knots (there's almost no wind, so our airspeed and groundspeed are the same), and it turns out that we need to descend at around 400 feet per minute. With the autopilot, that's pretty easy — just dial in -400, and you're set (but remember to reduce power to keep our speed at 75 knots, or you'll hit the runway going pretty fast, and be prepared to adjust things if you drift above or below the glide slope).

Without the autopilot, it's also pretty easy — just reduce power. How much ? In this case, with our plane, to around 1700 RPM. Again, it depends on many things — plane, elevation, winds, weight, . . . , so you'll have to adjust things if you see the glide-slope needle start to move up or down. Like the localizer needle though, . . . (are you ready ?) DON'T CHASE IT. Watch how it's moving, then make your adjustment.

Since we're on final approach, you might want to drop a second notch of flaps. This will affect your trim, and you'll have to adjust power a bit as well.

Intercept glideslope ;
cross outer marker ;
drop second notch of
flaps

10.4.10 Touchdown, Almost

After all the excitement of the procedure turn, it will seem like a long way down to the runway from the outer marker. There's not much to do but stare at those needles. In fact, you'll probably stare at them like you've never stared at them before. Take a look around at the other gauges too, though — they have useful things to tell you. Is our airspeed okay ? We don't want to stall. RPMs about right ? If flying manually, you'll want to constantly check the attitude indicator and directional gyro. This being a simulator, we don't have to worry about oil pressure and engine temperature, but you might want to glance over there anyway, just to get into the habit. And I hope you've done things like set the mixture to full rich (you did lean it out while cruising, didn't you ?).

10.4.11 A Confession

I've actually made you do more work than you have to. We've been using the autopilot as a fancy steering wheel, but it's capable of more than that. You may

17. Maybe. There can be false glideslopes, and FlightGear models these, so we have to make sure we're on the real one. One purpose of the procedure turn is to get you in the correct position, at the correct altitude, to intercept the true glideslope.

have noticed that the autopilot has some buttons I haven't explained — NAV, APR, and REV. Well, using those buttons, the autopilot can :

NAV : Track a VOR radial.

APR : Do a direct ILS approach, tracking both the localizer *and* the glideslope.

REV : Intercept the ILS before the procedure turn (ie, head *away* from the localizer).

So, in fact, even more of the work you've done could have been done by the autopilot. After takeoff, you could have asked it to track the 009 radial from SJC all the way to SUNOL in NAV mode ; at SUNOL, you could have asked it to fly the “back-course approach” from I-LVK in REV mode ; done the procedure turn in HDG mode ; finally, tracked the localizer and glideslope in APR mode.

However, I didn't give you this information for two reasons. First, flying by hand (even with the autopilot gently holding your hand, as we've been doing) gives you a better idea of what's happening. Second, the autopilot doesn't behave quite as the official manual says it should for some of these functions — best stick to the features that are known to work well.

10.4.12 Touchdown, Not

Although ILS approaches can get us close to the runway, closer than VFR, NDB, or VOR approaches can, we still need *some* visibility to land,¹⁸ so we need a way to decide if landing is possible or not. That's what the landing minimums section of the procedure plate is for (coloured green in Figure 10.11). In the category labelled “S-ILS 25R” (that's us), you'll see “597-1/2 200(200-1/2)”. This tells us that we can track the glide slope down to an altitude of 597 feet (200 feet above the runway). At 597 feet we make our decision — if we can't see the runway, then we have to execute a missed approach. 597 feet is our *decision height* (DH).

In addition to the altimeter, this particular approach also has another indication that we're close — a middle marker (MM). This marker will sound — in this case, a dot dash series — and the yellow light labelled “M” above COMM1 will flash. Passage over the middle marker should coincide with reaching decision height.¹⁹

So, what if you can't see the runway at decision height ? As you might have expected, just as you can't land willy-nilly, you can't just go around willy-nilly. There's a Procedure. A Missed Approach Procedure. This is shown in several places on the approach plate (see Figure 10.18) : At the top, where it says “MISSED APPROACH”, in the plan view, where you can see a dashed arrow coming off the end of the runway and a dashed oval on the right, and in the profile view, where a series of boxes shows graphically what to do. In our case, these all tell us to :

18. Well, unless it's a Category IIIC ILS approach.

19. As you may have guessed, the remaining light — white, and labelled “A” — indicates passage of the inner marker. Our approach doesn't have one, but San Francisco's runway 28R does. While passing over it, you should hear a rapid, high-pitched series of dots. Why is it labelled “A” and not “T” ? Because in ancient times, it was also used to identify passage over “airway” markers along radio range tracks.

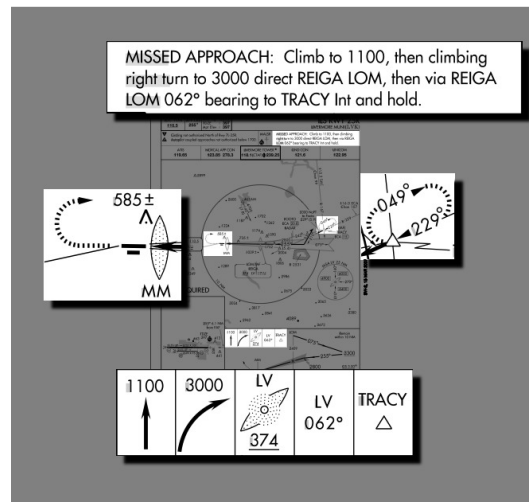


FIGURE 10.18 – Missed approach procedure

1. Climb straight ahead to 1100 feet
2. Make a climbing right turn to 3000 feet
3. Fly to REIGA
4. Fly outbound from REIGA at 062°
5. Fly a holding pattern at the TRACY intersection

The holding pattern, as you might have guessed, is a place where you can “park” while sorting things out, and has its own set of procedures and techniques which we won’t go into here, because . . .

10.4.13 Touchdown

In our ideal simulator world, you probably won’t have to execute a missed approach. Assuming you stayed on the glide slope, you should have popped out of the murk at the decision height, and with 800 metre visibility, the runway should have been in view soon after. With the runway in sight, you could then turn wildly to get on course²⁰ (it’s very hard to be lined up perfectly) and land “normally” (which for me involves a lot of bouncing around and cursing). Park the plane, then stagger out of the cockpit and have another hamburger !

Sight runway ; disengage autopilot ; cross middle marker
Land ; eat hamburger

10.5 Epilogue

That was a lot of information in a short time, a rather brutal introduction to ILS flying. Hopefully, instead of turning you off, it has whetted your appetite for more,

²⁰. Remembering, of course, to disengage the autopilot.

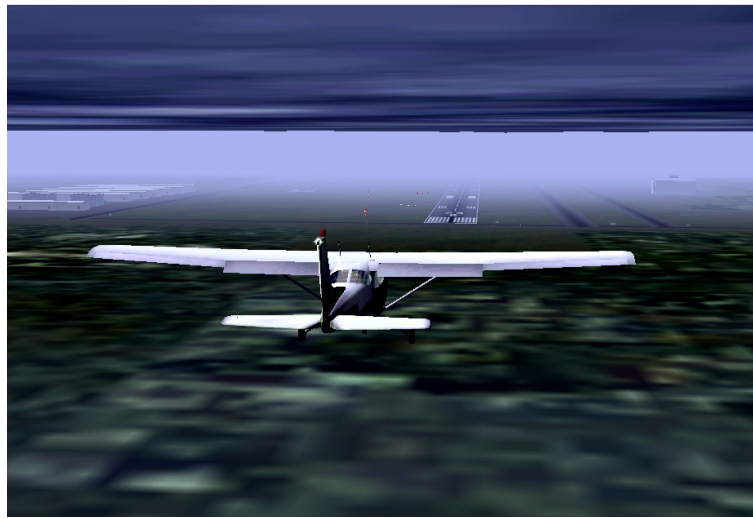


FIGURE 10.19 – On course, runway in view. We’re going to live !

because there *is* more. Some of the major issues I’ve ignored are :

Wind This is a big one. Flying IFR in a crosswind affects everything you do, and you need to be aware of it or your navigation will suffer.

Flying without the autopilot George tries his best, but he’s not completely trustworthy. You have to be prepared to go it alone.

DG precession The directional gyro in the c172p is not perfect. Over time, the values it gives you are less and less reliable — it *precesses*. It needs to be periodically calibrated against the compass (using the OBS knob on the DG to adjust it).

IFR charts We used sectionals, which are really intended for VFR flight. There are a whole set of charts devoted exclusively to IFR flight.

ATC The other people out there need to know what you’re doing. As well, they’ll probably tell you what to do, including to ignore the approach plate you so fastidiously studied.

SIDs/DPs, Airways, and STARs This tutorial introduced IAPs, which are standard ways to make approaches. In IFR flight, there are standard ways to *leave* airports (Standard Instrument Departures, *SIDs*, or Departure Procedures, *DPs*), standard ways to travel *between* airports (airways), and standard ways to go from airways to IAPs (Standard Terminal Arrival Routes, *STARs*).

Holding Patterns Most missed approaches end in a holding pattern somewhere, so you’d better know how to fly them.

GPS Our Cessna doesn’t have a GPS, but nowadays most small planes do, and GPS is rapidly replacing radio-based nav aids.

If you want to learn more, try the following resources :

- [Flight Simulator Navigation](#), written by Charles Wood. It covers everything from basic navigation to ILS approaches, with lots of examples and practice flights to improve your skills. Everything is linked together by an entertaining storyline in which you are the pilot for a fictional charter service. Two caveats, though. First, it is Microsoft Flight Simulator-based, so you'll have to translate into "FlightGear-ese" as appropriate. Second, it is a bit out of date, and things in the real world have changed since it was written. NDB beacons have been decommissioned, new approaches have replaced old ones — even an airport has disappeared (!). Treat this as a learning opportunity. You'll get better at finding more up to date information, and learn not to blindly trust your charts, just as you have learned not to blindly trust your instruments.
- If you're *really* keen and want to hear it straight from the horse's mouth, there's the official [FAA Instrument Flying Handbook](#). It's big and detailed, and there's *no* interesting storyline in which you're a pilot for a fictional charter service. It can be downloaded as multiple PDF files.
- If you'd like practice deciphering what the instruments are telling you, without the bother flying (or even virtual flying), you can try [luizmonteiro.com](#), which has Flash tutorials of various instruments, including a VOR and an ADF.
- Another simulated instrument site is [Tim's Air Navigation Simulator](#). It has a Java applet that simulates a plane flying in the vicinity of two navaids. The simulation allows you to use different kinds of instruments and navaids, so you can see their behaviour, and the advantages and disadvantages of each.
- If it's navigation information you're after, an excellent site is [AirNav.Com](#), which I've used extensively in the course of this tutorial. It has detailed airport, navaid, and fix information, and links to IAPs. Unfortunately, the information is only for the USA.
- Another source of airport and navaid information is [World Aero Data](#). Its information isn't as detailed as AirNav's, but it is international.
- [FlightSim.Com](#) has a very informative series of articles entitled "[How To . . . Use Approach Plates](#)". It starts with a *very, very* dense tutorial on how to read an approach plate, then follows with a set of approaches at Kodiak, Alaska. These are an excellent supplement to the approaches given in Charles Wood's *Flight Simulator Navigation* (see above). Most interesting, though, is section two — "Dangerous Approaches." Approaches at six airports around the world, from Penticton, BC to Kathmandu, Nepal, are described. Fly them if you dare !
Warning — the series is even more Microsoft Flight Simulator-centric than Charles Wood's, and some of it is out of date (some outside links are broken, and some of the approaches have changed).
- Also from [FlightSim.Com](#) is "[Golden Argosy](#)", a description of a flight from New York to Rome by Tony Vallillo, an American Airlines 767 captain. It

gives some interesting information about navigation that doesn't appear in the other sites mentioned here, such as the North Atlantic Tracks. However, its main appeal is that it gives a good answer to the question "What's it *really* like to be a pilot?" The author's love of flying is evident throughout the article.

- For those who are interested in the ATC side of things, and want information from an authoritative source, check out Michael Oxner's "[Aviation Topic of the Week](#)", a series of articles about flying "in many types of airspaces in many situations." Michael Oxner is a professional controller and private pilot who obviously can't get enough of airplanes, because in his spare time he's also an on-line controller with VatSim. Particularly interesting are a set of articles describing a complete IFR flight and a complete VFR flight.

Chapitre 11

A Helicopter Tutorial

11.1 Preface

First : in principle everything that applies to real helicopters, applies to *FlightGear*. Fundamental maneuvers are well described here : <http://www.cybercom.net/~copters/pilot/maneuvers.html> Some details are simplified in *FlightGear*, in particular the engine handling and some overstresses are not simulated or are without any consequence. In *FlightGear* it is (up to now) not possible to damage a helicopter in flight.



The helicopter flight model of *FlightGear* is quite realistic. The only exceptions are “vortex ring conditions”. These occur if you descend too fast and perpendicularly (without forward speed). The heli can get into its own rotor downwash causing the lift to be substantially reduced. Recovering from this condition is possible only at higher altitudes. On the Internet you can find a video of a Seaking helicopter, which got into this condition during a flight demonstration and touched down so hard afterwards that it was completely destroyed.

For all *FlightGear* helicopters the parameters are not completely optimized and thus the performance data between model and original can deviate slightly. On the hardware side I recommend the use of a “good” joystick. A joystick without springs is recommended because it will not center by itself. You can either remove the

spring from a normal joystick, or use a force feedback joystick, with a disconnected voltage supply. Further, the joystick should have a “thrust controller” (throttle). For controlling the tail rotor you should have pedals or at least a twistable joystick - using a keyboard is hard. *FlightGear* supports multiple joysticks attached at the same time.

11.2 Getting started

The number of available helicopters in *FlightGear* is limited. In my opinion the Bo105 is the easiest to fly, since it reacts substantially more directly than other helicopters. For flight behavior I can also recommend the S76C. The S76C reacts more retarded than the Bo.

Once you have loaded *FlightGear*, take a moment to centralize the controls by moving them around. In particular the collective is often at maximum on startup.



The helicopter is controlled by four functions. The stick (joystick) controls two of them, the inclination of the rotor disc (and thus the inclination of the helicopter) to the right/left and forwards/back. Together these functions are called “cyclic blade control”. Next there is the “collective blade control”, which is controlled by the thrust controller. This causes a change of the thrust produced by the rotor. Since the powering of the main rotor transfers a torque to the fuselage, this must be compensated by the tail rotor. Since the torque is dependent on the collective and on the flight condition as well as the wind on the fuselage, the tail rotor is also controlled by the pilot using the pedals. If you push the right pedal, the helicopter turns to the right (!). The pedals are not a steering wheel. Using the pedals you can yaw helicopter around the vertical axis. The number of revolutions of the rotor is kept constant (if possible) by the aircraft.



11.3 Lift-Off

First reduce the collective to minimum. To increase the rotor thrust, you have to “pull” the collective. Therefore for minimum collective you have to push the control down (that is the full acceleration position (!) of the thrust controller). Equally, “full power” has the thrust controller at idle. Start the engine with }. After few seconds the rotor will start to turn and accelerates slowly. Keep the stick and the pedals approximately centered. Wait until the rotor has finished accelerating. For the Bo105 there is an instruments for engine and rotor speed on the left of the upper row.

Once rotor acceleration is complete, pull the collective very slowly. Keep your eye on the horizon. If the heli tilts or turns even slightly, stop increasing the collective and correct the position/movement with stick and pedals. If you are successful, continue pulling the collective (slowly !).

As the helicopter takes off, increase the collective a little bit more and try to keep the helicopter in a leveled position. The main challenge is reacting to the inadvertent rotating motion of the helicopter with the correct control inputs. Only three things can help you : practice, practice and practice. It is quite common for it to take hours of practice to achieve a halfway good looking hovering flight. Note : The stick position in a stable hover is not the center position of the joystick.

11.4 In the air

To avoid the continual frustration of trying to achieve level flight, you may want to try forward flight. After take off continue pulling the collective a short time and then lower the nose a slightly using the control stick. The helicopter will accelerate forward. With forward speed the tail rotor does not have to be controlled as precisely due to the relative wind coming from directly ahead. Altogether the flight behavior in forward flight is quite similar to that of an badly trimmed airplane. The “neutral” position of the stick will depend upon airspeed and collective.

Transitioning from forward flight to hovering is easiest if you reduce speed slowly by raising the nose of the helicopter. At the same time, reduce the collective

to stop the helicopter from climbing. As the helicopter slows, “translation lift” is reduced, and you will have to compensate by pulling the collective. When the speed is nearly zero, lower the nose to the position it was when hovering. Otherwise the helicopter will accelerate backwards !

11.5 Back to Earth I

To land the helicopter transition to a hover as described above while reducing the altitude using the collective. Briefly before hitting the ground reduce the rate of descent slowly. A perfect landing is achieved if you managed to zero the altitude, speed and descent rate at the same time (gently). However, such landing are extremely difficult. Most pilots perform a hover more or less near to the ground and then decent slowly to the ground. Landing with forward velocity is easier, however you must make sure you don’t land with any lateral (sideways) component to avoid a rollover.



11.6 Back to Earth II

It is worth mentioning autorotation briefly. This is a unpowered flight condition, where the flow of air through the rotors rotates the rotor itself. At an appropriate altitude select a landing point (at first in the size of a larger airfield) and then switch the engine off by pressing `{`. Reduce collective to minimum, place the tail rotor to approximately 0° incidence (with the Bo push the right pedal about half, with As350 the left). Approach at approximately 80 knots. Don’t allow the rotor speed to rise more than a few percent over 100%, otherwise the rotor will be damaged (though this is not currently simulated). As you reach the ground, reduce the airspeed by lifting the nose. The descent rate will drop at the same time, so you do not need to pull the collective. It may be the case that the rotor speed rises beyond the permitted range. Counteract this by raising the collective if required. Just above the ground, reduce the descent rate by pulling the collective. The goal is it to touch down with a very low descent rate and no forward speed. With forward speed it is easier, but there is a danger of a roll over if the skids are not aligned parallel to the flight direction. During the approach it is not necessary to adjust the tail rotor, since without power there is almost no torque. If you feel (after some practice),

that autorotation is too easy, try it with a more realistic payload via the payload menu.



Quatrième partie

Appendices

Annexe A

Approche manquée : si rien ne fonctionne

Dans la section suivante, nous avons essayé de trier quelques problèmes qui peuvent être rencontrés, en fonction des systèmes d'exploitation. Cependant, si jamais vous rencontrez un problème, sachez qu'il peut parfois être une bonne idée de regarder plus loin que "votre" système d'exploitation - au cas où. Si vous rencontrez des difficultés, nous vous recommandons vivement de consulter en premier lieu la FAQ maintenue par Cameron Moore à l'adresse suivante :

<http://www.flightgear.org/Docs/FlightGear-FAQ.html>.

De plus, le code source comporte un répertoire `docs-mini` qui contient de nombreuses idées et solutions pour des problèmes spécifiques. Il s'agit donc également d'un bon endroit pour trouver d'avantage d'informations.

A.1 Signaler des problèmes relatifs à FlightGear

Le meilleur endroit pour obtenir de l'aide est généralement de faire appel aux listes de diffusion, et tout particulièrement à la liste de diffusion [**Flightgear-User**], ainsi que les forums. Si jamais vous utilisez une version Git de *FlightGear*, vous pourriez vouloir vous inscrire à la liste de diffusion [**Flightgear-Devel**]. Les informations pour s'y inscrire peuvent être trouvées à l'adresse :

<http://www.flightgear.org/mail.html>.

Bien souvent, vous n'êtes pas le premier à rencontrer ce type de difficulté. Donc une recherche sur les archives des listes de diffusion devrait vous permettre de trouver une solution rapide. Ces archives peuvent être consultées à l'adresse :

<http://www.mail-archive.com/flightgear-users%40flightgear.org/>
<http://www.mail-archive.com/flightgear-devel%40flightgear.org/>.

Vous devriez également considérer visiter les forums FlightGear pour rechercher de l'aide, des instructions et des archives à l'adresse :

<http://www.flightgear.org/forums>.

De nombreux développeurs et utilisateurs lisent ces listes et forums, donc les questions trouvent généralement une réponse. Cependant, avouez qu'il est difficile de répondre à des messages du type : *Je n'arrive pas à compiler FlightGear sur mon système, que dois-je faire ?* si vous ne donnez pas plus de détails, non ? Voici donc quelques éléments qu'il serait bon d'inclure dans votre message lorsque vous signalez un problème :

- **Système d'exploitation** : (Linux Fedora 17.../Windows Seven 64 bits...)
- **Ordinateur** : (Pentium Dual Core, 2,3 GHz...)
- **Carte graphique/processeur** : (ATI Radeon HD 770 XT/Nvidia GeForce GTX 590...)
- **Compilateur/version** : (GCC version 4.6.3...)
- **Versions des bibliothèques concernées** : (PLIB 1.8.5, OpenSceneGraph 3.0.1...)
- **Type de problème** : (Le compilateur s'arrête avec le message suivant...)
- **Étapes pour reproduire le problème** : Démarrer à KSFO, lâcher les freins...)

Afin d'obtenir une trace de la sortie que *FlightGear* produit, la commande suivante peut s'avérer utile (elle devra éventuellement être adaptée sur certains systèmes d'exploitation ou peut ne pas fonctionner du tout sur d'autres, d'ailleurs) :

```
%FG_ROOT/BIN/fgfs >log.txt 2>&1
```

Une dernière petite remarque : Merci d'essayer d'éviter de poster du code binaire sur ces forums ou sur ces listes ! Il y a de nombreux abonnés et personnes consultant ces informations, et certains disposent de bandes passantes limitées et/ou facturées. Des messages trop volumineux pourraient être refusés par l'administrateur des listes de diffusion. Merci.

A.2 Problèmes généraux

- *FlightGear* fonctionne siiiiientement.
Si *FlightGear* fonctionne, disons à quelque chose comme une image par seconde (*fps*, *frame per second*), ou moins, c'est que vous n'avez pas de matériel prenant en charge OpenGL. Il peut y avoir plusieurs raisons à cela. Tout d'abord, il peut effectivement n'y avoir aucun pilote matériel OpenGL disponible pour des cartes anciennes. Dans ce cas, il vous est vivement recommandé d'envisager l'achat d'une nouvelle carte.
Ensuite, vérifiez que vos pilotes sont correctement installés. Plusieurs cartes nécessitent des pilotes complémentaires pour faire fonctionner OpenGL en complément des pilotes "natifs" du gestionnaire de fenêtres.
- `configure` ou `make` échouent car ils ne trouvent pas les en-têtes ou bibliothèques *PLIB*.
Soyez certains de disposer de la dernière version de *PLIB* (> version 1.8.4) compilée et installée. Ses en-têtes comme `pu.h` doivent se situer dans le répertoire `/usr/include/plib` et ses bibliothèques, comme `libplibpu.a`,

dans le répertoire `/lib`. Vérifiez à nouveau qu'il n'y a pas d'autre en-têtes ou bibliothèques *PLIB* parasites présentes ailleurs !

Enfin, vérifiez attentivement les messages d'erreur(s) de `configure`. Dans de nombreux cas, ils donnent des informations précieuses sur les éléments manquants.

A.3 Problèmes potentiels sous Linux

Comme nous n'avons pas accès à toutes les versions possibles des distributions Linux, voici quelques-unes des causes possibles de problèmes sous cet environnement. (Cette section comprend des contributions de Kai Troester.)

- Mauvaises versions des bibliothèques

C'est une origine assez commune de griefs tout spécialement lorsque vous préférez installer les bibliothèques nécessaires à *FlightGear* à la main. Vérifiez bien que, en particulier, la bibliothèque Mesa comprend bien la prise en charge de la carte 3DFX et que les bibliothèques GLIDE sont installées et qu'elles peuvent être trouvées. Si un `ldd which fgfs`` se plaint de bibliothèques manquantes, alors vous aurez des difficultés.

Soyez également certain de *toujours* disposer de la *dernière* version de *PLIB* sur votre système. De nombreuses personnes ont lamentablement échoué à compiler *FlightGear* simplement à cause d'une version trop ancienne de `plib`.

- Droits manquants

Si vous utilisez XFree86 d'une version antérieure à 4.0, le binaire *FlightGear* peut nécessiter de disposer du bit `setuid root` afin de pouvoir accéder à certaines cartes d'accélération (ou un module spécial du noyau comme décrit précédemment dans ce document) basées sur des puces 3DFX. Vous pouvez alors essayer un :

```
chown root.root /usr/local/bin/fgfs ;
chmod 4755 /usr/local/bin/fgfs
```

pour donner au binaire *FlightGear* les droits appropriés ou installer le module 3DFX. Cette dernière solution étant la plus "propre" et donc fortement recommandée !

- Options d'installation particulières

FlightGear affichera un nombre important d'informations de diagnostic lors de son lancement. S'il se plaint de fichiers incorrects ou manquants, vérifiez que vous les avez installés de la manière dont ils sont supposés être installés (c'est-à-dire dans leur version la plus récente et à l'emplacement prévu). L'emplacement canonique de *FlightGear* attend ses données dans le répertoire `/usr/local/lib`. Soyez certain de récupérer les dernières versions de tout ce qui peut être nécessaire !

- Problèmes plus généraux de compilation

Soyez certain de disposer de la dernière version officielle de `gcc`. D'an-

ciennes version de gcc la cause de nombreux problèmes ! D'un autre côté, certaines version de RedHat 7.0 sont connues pour avoir des difficultés de compilation de *FlightGear*, car elles incluent une version préliminaire de gcc.

A.4 Problèmes potentiels sous Windows

- *FlightGear* ignore les paramètres de ligne de commande.
Il peut y avoir une difficulté à passer des options de ligne de commande contenant un caractère "=" sur la ligne de commande. Préférez plutôt la création d'un fichier *batch* pour y inclure vos options et lancez plutôt celui-ci.
- Je ne parviens pas à compiler *FlightGear* avec MSVC/MS DevStudio.
Par défaut, *FlightGear* est compilé avec GNU GCC. Le portage Win32 de GNU GCC est connu sous le nom de Cygwin. Pour obtenir des astuces sur les fichiers *Makefile* nécessaires pour MSVC ou MSC DevStudio veuillez consulter :

<https://www.gitorious.org/fg/flightgear/trees/next>.

En principe, il devrait être possible de compiler *FlightGear* à l'aide des fichiers de projet fournis avec le code source.

- La compilation de *FlightGear* échoue.
Il peut y avoir plusieurs raisons à cela, y compris l'existence de véritables anomalies. Cependant, avant de tenter quoi que ce soit ou de signaler un problème, soyez certain de disposer de la dernière version du compilateur *Cygwin*. En cas de doute, lancez à nouveau *setup.exe* et téléchargez et installez la version la plus récente de l'ensemble, car il est possible que cette version ait changé.

Annexe B

Atterrissage : quelques réflexions complémentaires avant de quitter l'avion

B.1 Une ébauche de l'histoire de *FlightGear*

L'histoire peut rapidement devenir un sujet barbant. Cependant, de temps en temps il y a des personnes qui s'intéressent à l'histoire de *FlightGear*. En conséquence, nous allons en dresser un aperçu rapide.

A l'origine, le projet *FlightGear* remonte à une discussion entre un groupe de citoyens du net en 1996 qui a eu pour résultat l'écriture d'une proposition par David Murr qui, malheureusement, a quitté le projet (ainsi que le net) plus tard. La proposition d'origine est toujours disponible à partir du site Internet de *FlightGear* à l'adresse :

<http://www.flightgear.org/proposal-3.0.1>.

Bien que les noms des personnes et des détails aient évolué au fil du temps, l'esprit de cette proposition a clairement été maintenu au fil du temps jusqu'à aujourd'hui.

La véritable écriture du code a débuté à l'été 1996 et, dès la fin de l'année, l'essentiel des routines graphiques était écrit. A cette époque, la programmation était essentiellement réalisée et coordonnée par Eric Korpela de l'université de Berkeley. Le tout premier code fonctionnait sous Linux ainsi que sous DOS, OS/2, Windows 95/NT, et Sun-OS. C'était un projet assez ambitieux car il nécessitait notamment d'écrire, en partant de rien, toutes les routines graphiques d'une manière indépendante du système.

Le développement ralentit pour finalement s'arrêter au début de l'année 1997 lorsque Eric termina sa thèse. A ce moment, le projet semblait mort et le trafic sur la liste de diffusion était quasiment réduit à néant.

Curt Olson, de l'Université du Minnesota, relança le projet au milieu de l'année 1997. Son idée était aussi simple que puissante : pourquoi réinventer la roue une

seconde fois ? Il y avait déjà eu plusieurs simulateurs de vol libres disponibles sur des stations de travail sous différentes moutures d'UNIX. L'un de ceux-ci, LaRCsim (développé par Bruce Jackson de la NASA), semblait être particulièrement approprié à cette approche. Curt s'y intéressa et en ré-écrivit plusieurs routines afin de les faire compiler et fonctionner sur les plate-formes ciblées. Ce-faisant, l'idée maîtresse était d'exploiter une plate-forme graphique indépendante du système : OpenGL.

En complément, la (bonne) décision du choix des données des scènes de base a été prise dès la toute première version. Les scènes de *FlightGear* sont générées en se basant sur les données satellite publiées par le U. S. Geological Survey. Ces données de terrain sont disponibles respectivement à partir de :

<http://edc.usgs.gov/geodata/>

pour les Etats-Unis et

<http://edcdaac.usgs.gov/topo30/topo30.html>,

pour les autres pays. Ces données de scènes accessibles librement, en conjonction avec les outils de construction des scènes inclus dans *FlightGear*, sont des fonctionnalités importantes permettant à chacun(e) de créer ses propres scènes.

Ce nouveau code de *FlightGear* - toujours largement basé sur le code original de LaRCsim - a été mis à disposition en juillet 1997. A partir de cet instant, le projet reprit de l'inertie. Voici les principales étapes dans l'histoire récente du développement.

B.1.1 Scènes

- La prise en charge des textures a été ajouté par Curt Olson à l'automne 1998. Ceci a marqué un tournant important en termes de réalisme. Quelques textures de haute qualité pour le projet *FlightGear* ont été ajoutées par Eric Mitchell. Depuis, d'autres textures de haute qualité ont été ajoutées par Erik Hofman.
- Après avoir amélioré la prise en charge des scènes et des textures, le de l'image diminua à tel point qu'il était impossible de faire voler un avion dans *FlightGear* au printemps 1998. Cette difficulté fut contournée en exploitant la prise en charge matérielle d'OpenGL, qui devint disponible à cette date, et en implémentant la vue frustum culling (une technique de rendu qui ignore la partie du paysage non visible d'une scène), réalisée par Curt Olson. En ce qui concerne le taux de rafraîchissement, il convient de se rappeler que le code, tel qu'il est aujourd'hui, n'est en aucun cas optimisé, ce qui laisse des possibilités pour de futures améliorations.
- En septembre 1998, Curt Olson parvint à créer un modèle de terrain complet pour les Etats-Unis. Les scènes sont aujourd'hui mondiales, et disponibles via une carte cliquable ; située à l'adresse :

<http://www.flightgear.org/download/scenery/>.

- Scenery was further improved by adding geographic features including lakes, rivers, and coastlines later, an effort still going on. Textured runways were added by Dave Cornish in spring 2001. Light textures add to the visual impression at night. To cope with the constant growth of scenery data, a binary scenery format was introduced in spring 2001. Runway lighting was introduced by Curt Olson in spring 2001. Finally, a completely new set of scenery files for the whole world was created by William Riley based on preparatory documentation by David Megginson in summer 2002. This is based on a data set called VMap0 as an alternative to the GSHHS data used so far. This scenery is a big improvement as it has world wide coverage of main streets, rivers, etc., while its downside are much less accurate coast lines. *FlightGear's* base scenery is based on these new scenery files since summer 2002.
- There was support added for static objects to the scenery in 2001, which permits placing buildings, static planes, trees and so on in the scenery. However, despite a few proofs of concept systematic inclusion of these landmarks is still missing.
- The world is populated with random ground objects with appropriate type and density for the local ground cover type since summer 2002. This marks a major improvement of reality and is mainly thanks to work by D. Megginson.

B.1.2 Aircraft

- A HUD (head up display) was added based on code provided by Michele America and Charlie Hotchkiss in the fall of 1997 and was improved later by Norman Vine. While not generally available for real Cessna 172, the HUD conveniently reports the actual flight performance of the simulation and may be of further use in military jets later.
- A rudimentary autopilot implementing heading hold was contributed by Jeff Goeke-Smith in April 1998. It was improved by the addition of an altitude hold and a terrain following switch in October 1998 and further developed by Norman Vine later.
- Friedemann Reinhard developed early instrument panel code, which was added in June 1998. Unfortunately, development of that panel slowed down later. Finally, David Megginson decided to rebuild the panel code from scratch in January 2000. This led to a rapid addition of new instruments and features to the panel, resulting in nearly all main instruments being included until spring 2001. A handy minipanel was added in summer 2001.
- Finally, LaRCsims Navion was replaced as the default aircraft when the Cessna 172 was stable enough in February 2000 - as most users will welcome. There are now several flight model and airplane options to choose from at runtime. Jon Berndt has invested a lot of time in a more realistic and versatile flight model with a more powerful aircraft configuration method.

JSBSim, as it has come to be called, did replace LaRCsim as the default flight dynamics model (FDM), and it is planned to include such features as fuel slosh effects, turbulence, complete flight control systems, and other features not often found all together in a flight simulator. As an alternative, Andy Ross added another flight dynamics model called *YASim* (Yet Another Flight Dynamics Simulator) which aims at simplicity of use and is based on fluid dynamics, by the end of 2001. This one bought us flight models for a 747, an A4, and a DC-3. Alternatively, a group around Michael Selig from the UIUC group provided another flight model along with several planes since around 2000.

- A fully operational radio stack and working radios were added to the panel by Curt Olson in spring 2000. A huge database of Nav aids contributed by Robin Peel allows IFR navigation since then. There was basic ATC support added in fall 2001 by David Luff. This is not yet fully implemented, but displaying ATIS messages is already possible. A magneto switch with proper functions was added at the end of 2001 by John Check and David Megginson.. Moreover, several panels were continually improved during 2001 and 2002 by John and others. *FlightGear* now allows flying ILS approaches and features a Bendix transponder.
- In 2002 functional multi-engine support found it's way into *FlightGear*. *JSB-Sim* is now the default FDM in *FlightGear*.
- Support of "true" 3D panels became stable via contributions from John Check and others in spring 2002. In addition, we got movable control surfaces like propellers etc., thanks to David Megginson.

B.1.3 Environment

- The display of sun, moon and stars have been a weak point for PC flight simulators for a long time. It is one of the great achievements of *FlightGear* to include accurate modeling and display of sun, moon, and planets very early. The corresponding astronomy code was implemented in fall 1997 by Durk Talsma.
- Christian Mayer, together with Durk Talsma, contributed weather code in the winter of 1999. This included clouds, winds, and even thunderstorms.

B.1.4 User Interface

- The foundation for a menu system was laid based on another library, the Portable Library *PLIB*, in June 1998. After having been idle for a time, the first working menu entries came to life in spring 1999. *PLIB* underwent rapid development later. It has been distributed as a separate package by Steve Baker with a much broader range of applications in mind, since spring 1999. It has provided the basic graphics rendering engine for *FlightGear* since fall 1999.

- In 1998 there was basic audio support, i. e. an audio library and some basic background engine sound. This was later integrated into the above-mentioned portable library, *PLIB*. This same library was extended to support joystick/yoke/rudder in October 1999, again marking a huge step in terms of realism. To adapt on different joystick, configuration options were introduced in fall 2000. Joystick support was further improved by adding a self detection feature based on xml joystick files, by David Megginson in summer 2002.
- Networking/multiplayer code has been integrated by Oliver Delise and Curt Olson starting fall 1999. This effort is aimed at enabling *FlightGear* to run concurrently on several machines over a network, either an Intranet or the Internet, coupling it to a flight planner running on a second machine, and more. There emerged several approaches for remotely controlling *FlightGear* over a Network during 2001. Notably there was added support for the “Atlas” moving map program. Besides, an embedded HTTP server developed by Curt Olson late in 2001 can now act a property manager for external programs.
- Manually changing views in a flight simulator is in a sense always “unreal” but nonetheless required in certain situations. A possible solution was supplied by Norman Vine in the winter of 1999 by implementing code for changing views using the mouse. Alternatively, you can use a hat switch for this purpose, today.
- A property manager was implemented by David Megginson in fall 2000. It allows parsing a file called `.fgfsrc` under UNIX/Linux and `system.fgfsrc` under Windows for input options. This plain ASCII file has proven useful in submitting the growing number of input options, and notably the joystick settings. This has shown to be a useful concept, and joystick, keyboard, and panel settings are no longer hard coded but set using *.xml files since spring 2001 thanks to work mainly by David Megginson and John Check.

During development there were several code reorganization efforts. Various code subsystems were moved into packages. As a result, code is organized as follows at present :

The base of the graphics engine is **OpenGL**, a platform independent graphics library. Based on OpenGL, the Portable Library *PLIB* provides basic rendering, audio, joystick etc routines. Based on *PLIB* is *SimGear*, which includes all of the basic routines required for the flight simulator as well as for building scenery. On top of *SimGear* there are (i) *FlightGear* (the simulator itself), and (ii) *TerraGear*, which comprises the scenery building tools.

This is by no means an exhaustive history and most likely some people who have made important contributions have been left out. Besides the above-named contributions there was a lot of work done concerning the internal structure by : Jon S. Berndt, Oliver Delise, Christian Mayer, Curt Olson, Tony Peden, Gary R. Van Sickle, Norman Vine, and others. A more comprehensive list of contributors can be found in Chapter B as well as in the `Thanks` file provided with the code. Also, the *FlightGear* Website contains a detailed history worth reading of all of the

notable development milestones at

<http://www.flightgear.org/News/>

B.2 Those, who did the work

Did you enjoy the flight? In case you did, don't forget those who devoted hundreds of hours to that project. All of this work is done on a voluntary basis within spare time, thus bare with the programmers in case something does not work the way you want it to. Instead, sit down and write them a kind (!) mail proposing what to change. Alternatively, you can subscribe to the *FlightGear* mailing lists and contribute your thoughts there. Instructions to do so can be found at

<http://www.flightgear.org/mail.html>.

Essentially there are two lists, one of which being mainly for the developers and the other one for end users. Besides, there is a very low-traffic list for announcements.

The following names the people who did the job (this information was essentially taken from the file `Thanks` accompanying the code).

A1 Free Sounds

Granted permission for the *FlightGear* project to use some of the sound effects from their site. Homepage under

<http://www.a1freesoundeffects.com/>

Syd Adams

Added clipping for 2D instruments, ATC volume control and created a wide variety of aircraft.

Raul Alonzo

Mr. Alonzo is the author of `Ssystem` and provided his kind permission for using the moon texture. Parts of his code were used as a template when adding the texture. `Ssystem` Homepage can be found at :

<http://www1.las.es/~amil/ssystem/>.

Michele America

Contributed to the HUD code.

Michael Basler

Author of Installation and Getting Started. Flight Simulation Page at

<http://www.geocities.com/pmb.geo/flusi.htm>

Jon S. Berndt

Working on a complete C++ rewrite/reimplimentation of the core FDM. Initially he is using X15 data to test his code, but once things are all in place we should be able

to simulate arbitrary aircraft. Jon maintains a page dealing with Flight Dynamics at :

<http://jsbsim.sourceforge.net/>

Special attention to X15 is paid in separate pages on this site. Besides, Jon contributed via a lot of suggestions/corrections to this Guide.

Paul Bleisch

Redid the debug system so that it would be much more flexible, so it could be easily disabled for production system, and so that messages for certain subsystems could be selectively enabled. Also contributed a first stab at a config file/command line parsing system.

Jim Brennan

Provided a big chunk of online space to store USA scenery for *FlightGear*!.

Bernie Bright

Many C++ style, usage, and implementation improvements, STL portability and much, much more. Added threading support and a threaded tile pager.

Stuart Buchanan

Updated various parts of the manual and wrote the initial tutorial subsystem.

Bernhard H. Buckel

Contributed the README.Linux. Contributed several sections to earlier versions of Installation and Getting Started.

Gene Buckle

A lot of work getting *FlightGear* to compile with the MSVC++ compiler. Numerous hints on detailed improvements.

Ralph Carmichael

Support of the project. The Public Domain Aeronautical Software web site at

<http://www.pdas.com/>

has the PDAS CD-ROM for sale containing great programs for aeronautical engineers.

Didier Chauveau

Provided some initial code to parse the 30 arcsec DEM files found at :

<http://edcwww.cr.usgs.gov/landdaac/gtopo30/gtopo30.html>.

John Check

John maintains the base package CVS repository. He contributed cloud textures, wrote an excellent Joystick Howto as well as a panel Howto. Moreover, he contributed new instrument panel configurations. *FlightGear* page at

<http://www.rockfish.net/fg/>.

Dave Cornish

Dave created new cool runway textures plus some of our cloud textures.

Oliver Delise

Started a FAQ, Documentation, Public relations. Working on adding some networking/multi-user code. Founder of the FlightGear MultiPilot

Jean-Francois Doue

Vector 2D, 3D, 4D and Matrix 3D and 4D inlined C++ classes. (Based on Graphics Gems IV, Ed. Paul S. Heckbert)

http://www.animats.com/simpleppp/ftp/public_html/topics/developers.html.

Dave Eberly

Contributed some sphere interpolation code used by Christian Mayer's weather data base system.

Francine Evans

Wrote the GPL'd tri-striper we use.

<http://www.cs.sunysb.edu/~stripe/>

Oscar Everitt

Created single engine piston engine sounds as part of an F4U package for FS98. They are pretty cool and Oscar was happy to contribute them to our little project.

Bruce Finney

Contributed patches for MSVC5 compatibility.

Olaf Flebbe

Improved the build system for Windows and provided pre-built dependencies.

Melchior Franz

Contributed joystick hat support, a LED font, improvements of the telnet and the http interface. Notable effort in hunting memory leaks in *FlightGear*, *SimGear*, and *JSBSim*.

Jean-loup Gailly and Mark Adler

Authors of the zlib library. Used for on-the-fly compression and decompression routines,

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

Mohit Garg

Contributed to the manual.

Thomas Gellekum

Changes and updates for compiling on FreeBSD.

Neetha Girish

Contributed the changes for the xml configurable HUD.

Jeff Goeke-Smith

Contributed our first autopilot (Heading Hold). Better autoconf check for external timezone/daylight variables.

Michael I. Gold

Patently answered questions on OpenGL.

Habibe

Made RedHat package building changes for SimGear.

Mike Hill

For allowing us to concert and use his wonderful planes, available form

<http://www.flightsimnetwork.com/mikehill/home.htm>,
for *FlightGear*.

Erik Hofman

Major overhaul and parameterization of the sound module to allow aircraft-specific sound configuration at runtime. Contributed SGI IRIX support (including binaries) and some really great textures.

Charlie Hotchkiss

Worked on improving and enhancing the HUD code. Lots of code style tips and code tweaks.

Bruce Jackson (NASA)

Developed the LaRCsim code under funding by NASA which we use to provide the flight model. Bruce has patiently answered many, many questions.

Maik Justus

Added helicopter support, gear/ground interaction and aerotow/winch support to the YASim FDM.

Ove Kaaven

Contributed the Debian binary.

Richard Kaszeta

Contributed screen buffer to ppm screen shot routine. Also helped in the early development of the "altitude hold autopilot module" by teaching Curt Olson the basics of Control Theory and helping him code and debug early versions. Curt's Boss Bob Hain also contributed to that. Further details available at :

<http://www.menet.umn.edu/~curt/fgfs/Docs/Autopilot/AltitudeHold/AltitudeHold.html>.

Rich's Homepage is at

<http://www.kaszeta.org/rich/>.

Tom Knienieder

Ported the audio library first to OpenBSD and IRIX and after that to Win32.

Reto Koradi

Helped with setting up fog effects.

Bob Kuehne

Redid the Makefile system so it is simpler and more robust.

Kyler B Laird

Contributed corrections to the manual.

David Luff

Contributed heavily to the IO360 piston engine model.

Sam van der Mac

Contributed to The Manual by translating HTML tutorials to Latex.

Christian Mayer

Working on multi-lingual conversion tools for fgfs as a demonstration of technology. Contributed code to read Microsoft Flight Simulator scenery textures. Christian is working on a completely new weather subsystem. Donated a hot air balloon to the project.

David Megginson

Contributed patches to allow mouse input to control view direction yoke. Contributed financially towards hard drive space for use by the flight gear project. Updates to README.running. Working on getting fgfs and ssg to work without textures. Also added the new 2-D panel and the save/load support. Further, he developed new panel code, playing better with OpenGL, with new features. Developed the property manager and contributed to joystick support. Random ground cover objects

Cameron Moore

FAQ maintainer. Reigning list administrator. Provided man pages.

Eric Mitchell

Contributed some topnotch scenery textures being all original creations by him.

Anders Morken

Former maintainer of European web pages.

Alan Murta

Created the Generic Polygon Clipping library.

<http://www.cs.man.ac.uk/aig/staff/alan/software/>

Phil Nelson

Author of GNU dbm, a set of database routines that use extendible hashing and work similar to the standard UNIX dbm routines.

Alexei Novikov

Created European Scenery. Contributed a script to turn fgfs scenery into beautifully rendered 2-D maps. Wrote a first draft of a Scenery Creation Howto.

Curt Olson

Primary organization of the project.

First implementation and modifications based on LaRCsim.

Besides putting together all the pieces provided by others mainly concentrating on the scenery subsystem as well as the graphics stuff. Homepage at

<http://www.menet.umn.edu/~curt/>

Brian Paul

We made use of his TR library and of course of Mesa :

<http://www.mesa3d.org/brianp/TR.html>, <http://www.mesa3d.org>

Tony Peden

Contributions on flight model development, including a LaRCsim based Cessna 172. Contributed to *JSBSim* the initial conditions code, a more complete standard atmosphere model, and other bugfixes/additions.

Robin Peel

Maintains worldwide airport and runway database for *FlightGear* as well as X-Plane.

Alex Perry

Contributed code to more accurately model VSI, DG, Altitude. Suggestions for improvements of the layout of the simulator on the mailing list and help on documentation.

Friedemann Reinhard

Development of an early textured instrument panel.

Petter Reinholdtsen

Incorporated the GNU automake/autoconf system (with libtool). This should streamline and standardize the build process for all UNIX-like platforms. It should have little effect on IDE type environments since they don't use the UNIX make system.

William Riley

Contributed code to add "brakes". Also wrote a patch to support a first joystick with more than 2 axis. Did the job to create scenery based on VMap0 data.

Andy Ross

Contributed a new configurable FDM called *YASim* (Yet Another Flight Dynamics Simulator, based on geometry information rather than aerodynamic coefficients.

Paul Schlyter

Provided Durk Talsma with all the information he needed to write the astro code. Mr. Schlyter is also willing to answer astro-related questions whenever one needs to.

<http://www.welcome.to/pausch/>

Chris Schoeneman

Contributed ideas on audio support.

Phil Schubert

Contributed various textures and engine modeling.

<http://www.zedley.com/Philip/>

Jonathan R. Shewchuk

Author of the Triangle program. Triangle is used to calculate the Delauney triangulation of our irregular terrain.

Gordan Sikic

Contributed a Cherokee flight model for LaRCsim. Currently is not working and needs to be debugged. Use configure `--with-flight-model=cherokee` to build the cherokee instead of the Cessna.

Michael Smith

Contributed cockpit graphics, 3D models, logos, and other images. Project Bonanza

Martin Spott

Co-Author of The Manual.

Jon Stockill

Maintains a database of objects and their location to populate the worldwide scenery.

Durk Talsma

Accurate Sun, Moon, and Planets. Sun changes color based on position in sky. Moon has correct phase and blends well into the sky. Planets are correctly positioned and have proper magnitude. Help with time functions, GUI, and other things. Contributed 2-D cloud layer. Website at

<http://people.a2000.nl/dtals/>.

UIUC - Department of Aeronautical and Astronautical Engineering

Contributed modifications to LaRCsim to allow loading of aircraft parameters from a file. These modifications were made as part of an icing research project.

Those did the coding and made it all work :

Jeff Scott

Bipin Sehgal

Michael Selig

Moreover, those helped to support the effort :

Jay Thomas

Eunice Lee

Elizabeth Rendon

Sudhi Uppuluri

U. S. Geological Survey

Provided geographic data used by this project.

<http://edc.usgs.gov/geodata/>

Mark Vallevand

Contributed some METAR parsing code and some win32 screen printing routines.

Gary R. Van Sickle

Contributed some initial GameGLUT support and other fixes. Has done preliminary work on a binary file format. Check

<http://www.woodsoup.org/projs/ORKiD/fgfs.htm>.

His *Cygnwin* Tipspage might be helpful for you at

<http://www.woodsoup.org/projs/ORKiD/cygnwin.htm>.

Norman Vine

Provided more numerous URL's to the "FlightGear Community". Many performance optimizations throughout the code. Many contributions and much advice for the scenery generation section. Lots of Windows related contributions. Contributed wgs84 distance and course routines. Contributed a great circle route autopilot mode based on wgs84 routines. Many other GUI, HUD and autopilot contributions. Patch to allow mouse input to control view direction. Ultra hires tiled screen dumps. Contributed the initial goto airport and reset functions and the initial http image server code

Roland Voegtli

Contributed great photorealistic textures. Founder of European Scenery Project for X-Plane :

<http://www.g-point.com/xpcity/esp/>

Carmelo Volpe

Porting *FlightGear* to the Metro Works development environment (PC/Mac).

Darrell Walisser

Contributed a large number of changes to porting *FlightGear* to the Metro Works development environment (PC/Mac). Finally produced the first Macintosh port. Contributed to the Mac part of Getting Started, too.

Ed Williams

Contributed magnetic variation code (impliments Nima WMM 2000). We've also borrowed from Ed's wonderful aviation formulary at various times as well. Website at <http://williams.best.vwh.net/>.

Jim Wilson

Wrote a major overhaul of the viewer code to make it more flexible and modular. Contributed many small fixes and bug reports. Contributed to the PUI property browser and to the autopilot.

Jean-Claude Wippler

Author of MetaKit - a portable, embeddible database with a portable data file format previously used in *FlightGear*. Please see the following URL for more info :

<http://www.equi4.com/metakit/>

Woodsoup Project

While *FlightGear* no longer uses Woodsoup services we appreciate the support provided to our project during the time they hosted us. Once they provided computing resources and services so that the *FlightGear* project could have a real home.

<http://www.woodsoup.org/>

Robert Allan Zeh

Helped tremendously in figuring out the Cygnus Win32 compiler and how to link with dll's. Without him the first run-able Win32 version of *FlightGear* would have been impossible.

Others

The following individuals have contributed to the scenery object database : Jon Stockill, Martin Spott, Dave Martin, Thomas Foerster, Chris Metzler, Frederic Bouvier, Melchior Franz, Roberto Inzerillo, Erik Hofman, Mike Round, Innis Cunningham, David Megginson, Stuart Buchanan, Josh Babcock, Esa Hyytia, Mircea Lutic, Jens Thoms Toerring, Mark Akermann, Torsten Dreyer, Martin C. Doege, Alexis Bory, Sebastian Bechtold, Julien Pierru, Bertrand Augras, Gerard Robin, Jakub Skibinski, Morten Oesterlund Joergensen, Carsten Vogel, Dominique Lemesre, Daniel Leygnat, Bertrand Gilot, Morten Skyt Eriksen, Alex Bamesreiter, Oliver Predelli, Georg Vollnhals, and Paul Richter.

B.3 What remains to be done

If you read (and, maybe, followed) this guide up to this point you may probably agree : *FlightGear* even in its present state, is not at all for the birds. It is already a flight simulator which sports even several selectable flight models, several planes with panels and even a HUD, terrain scenery, texturing, all the basic controls and weather.

Despite, *FlightGear* needs – and gets – further development. Except internal tweaks, there are several fields where *FlightGear* needs basics improvement and development. A first direction is adding airports, buildings, and more of those things bringing scenery to real life and belonging to realistic airports and cities. Another task is further implementation of the menu system, which should not be too hard with the basics being working now. A lot of options at present set via command line or even during compile time should finally make it into menu entries. Finally, *FlightGear* lacks any ATC until now.

There are already people working in all of these directions. If you're a programmer and think you can contribute, you are invited to do so.

Acknowledgements

Obviously this document could not have been written without all those contributors mentioned above making *FlightGear* a reality.

First, I was very glad to see Martin Spott entering the documentation effort. Martin provided not only several updates and contributions (notably in the OpenGL section) on the Linux side of the project but also several general ideas on the documentation in general.

Besides, I would like to say special thanks to Curt Olson, whose numerous scattered Readmes, Thanks, Webpages, and personal eMails were of special help to me and were freely exploited in the making of this booklet.

Next, Bernhard Buckel wrote several sections of early versions of that Guide and contributed a lot of ideas to it.

Jon S. Berndt supported me by critical proofreading of several versions of the document, pointing out inconsistencies and suggesting improvements.

Moreover, I gained a lot of help and support from Norman Vine. Maybe, without Norman's answers I would have never been able to tame different versions of the *Cygwin – FlightGear* couple.

We were glad, our Mac expert Darrell Walisser contributed the section on compiling under Mac OS X. In addition he submitted several Mac related hints and fixes.

Further contributions and donations on special points came from John Check, (general layout), Oliver Delise (several suggestions including notes on that chapter), Mohit Garg (OpenGL), Kyler B. Laird (corrections), Alex Perry (OpenGL), Kai Troester (compile problems), Dave Perry (joystick support), and Michael Selig (UIUC models).

Besides those whose names got lost withing the last-minute-trouble we'd like to express our gratitude to the following people for contributing valuable 'bug fixes' to this version of The FlightGear Manual (in random order) : Cameron Moore, Melchior Franz, David Megginson, Jon Berndt, Alex Perry,, Dave Perry,, Andy Ross, Erik Hofman, and Julian Foad.

Index

- .fgfsrc, [37](#), [193](#)
- équipement, [64](#)
- joystick, [38](#)
- 2D cockpit, [59](#)
- 3D clouds, [41](#)
- 3D panels, [192](#)
- 3DFX, [187](#)

- A1 Free Sounds, [194](#)
- aérofrenes, [57](#), [59](#)
- abort, [118](#)
- Adams, Syd, [194](#)
- additional scenery, [19](#)
- ADF, [113](#)
- Adler, Mark, [196](#)
- Aeronautical Information Manual, [84](#)
- Affichage, [62](#)
- aileron, [57](#), [112](#)
- Air Traffic Control, [143](#)
- air traffic facilities, [113](#)
- Air-Air Refuelling, [78](#)
- aircraft
 - installation, [23](#)
 - selection, [38](#)
- aircraft aeronautical model, [39](#)
- Aircraft Carrier, [71](#)
- airport, [39](#), [202](#)
- airspeed indicator, [112](#)
- Airwave Xtreme 150, [17](#)
- Alonzo, Raul, [194](#)
- Altimeter, [139](#)
- altimeter, [95](#), [112](#)
 - tuning, [96](#)
- altitude, [67](#)
 - absolute, [96](#)
- America, Michele, [191](#), [194](#)
- angle de vue, [58](#)
- artificial horizon, [112](#)
- assiette, [67](#)
- astronomy code, [192](#)
- ATC, [192](#), [202](#)
- ATIS, [136](#)
- ATIS messages, [192](#)
- Atlas, [73](#), [193](#)

- attitude indicator, [112](#)
- audio library, [197](#)
- audio support, [193](#)
- auto coordination, [112](#)
- auto-co-ordination, [38](#)
- autopilot, [122](#), [140](#), [191](#), [196](#), [197](#)
 - modes
 - heading mode, [122](#)
 - roll control mode, [122](#)
 - vertical speed mode, [122](#)
- Avionics, [43](#)

- Baker, Steve, [192](#)
- bank, [112](#)
- Basler, Michael, [194](#)
- Bendix transponder, [192](#)
- Berndt, Jon, [203](#)
- Berndt, Jon, S., [191](#), [193](#), [194](#), [203](#)
- binaries
 - pre-compiled, [7](#)
- binary distribution, [5](#)
- Bleisch, Paul, [195](#)
- brakes, [98](#), [199](#)
 - left wheel [,] (colon), [98](#)
 - right wheel [.] (dot), [98](#)
- Brennan, Jim, [195](#)
- Bright, Bernie, [195](#)
- BSD UNIX, [12](#)
- Buchanan, Stuart, [195](#)
- Buckel, Bernhard, [195](#), [203](#)
- Buckle, Gene, [195](#)

- capture d'écran, [62](#)
- Carmichael, Ralph, [195](#)
- carte graphique, [14](#), [15](#)
- carte, cliquable, [190](#)
- Cessna, [200](#)
- Cessna 172, [191](#)
- Chauveau, Didier, [195](#)
- Check, John, [54](#), [192](#), [193](#), [195](#), [203](#)
- Cherokee flight model, [200](#)
- clock, [113](#)
- cloud layers, [41](#)
- clouds, [192](#), [200](#)

- cockpit, 59
- cockpit 3D, 59
- code source, 13
- COM transceiver, 113
- COMM1, 113
- COMM2, 113
- command line options, 37
- communication radio, 113
- compilateur, 15
- contrôles moteurs, 57
- contrôles clavier, 56
 - affichage, 59
 - général, 59
 - vue, 58
 - aéronef, 57
 - base, 56
 - contrôles du simulateur, 58
 - moteur, 57
 - pilote automatique, 58, 59
- contrôles du pilote automatique, 60
- contributeurs, 194
- control mode, 38
- control surface, movable, 192
- Cornish, Dave, 191, 195
- Cygnus, 15, 202
- Cygwin, 15, 188

- débogage, 65
- Démarrer Flightgear
 - Linux, 28
 - Mac OS X, 30
 - Windows, 29
- default settings, 37
- Delise, Oliver, 193, 196, 203
- deltaplane, 17
- Denker, John, 85
- Direct3D, 14
- direction de la vue, 58
- DirectX, 14
- documentation, 13
 - installation, 23
- DOS, 189
- Doue, Jean-Francois, 196

- Eberly, Dave, 196
- Ecole de pilotage FlightGear, 17
- effets sonores, 15
- engine
 - shut down, 105
- Entrées du menu, 62
- espace disque, 15
- Evans, Francine, 196
- Everitt, Oscar, 196

- FAA, 84
- FAA Training Book, 84
- FAQ, 185
- FDM, 192, 194
 - external, 17
- FG_ROOT, 27
- FG_SCENERY, 21, 27
- field of view, 42
- Finney, Bruce, 196
- flaps, 108, 112
 - control lever, 108
 - steps, 108
- Flebbe, Olaf, 196
- flight dynamics model, 39, 192
- flight model, 16, 39, 191
- flight planner, 193
- Flight simulator
 - civilian, 12
 - multi-platform, 12
 - open, 12, 13
 - user-extensible, 13
 - user-supported, 12, 13
- FlightGear, 193
 - versions, 16
- FlightGear documentation, 17
- FlightGear Programmer's Guide, 17
- FlightGear Scenery Design Guide, 17
- FlightGear Website, 193
- Foad, Julian, 203
- fog, 42
- fog effects, 197
- forums, 185
- forums FlightGear, 185
- frame rate, 42
- Franz, Melchior, 196, 203
- FreeBSD, 196
- freinage différentiel, 57, 59
- freins, 57, 59
- freins de parking, 55, 57, 59
- FS98, 196
- fuel indicator, 113
- full screen display, 41

- Gailly, Jean-loup, 196
- GameGLUT, 201
- Garg, Mohit, 196, 203
- gaz, 57
- Gellekum, Thomas, 196
- geographic features, 191
- Girish, Neetha, 196
- GLIDE, 187
- GNU C++, 15
- GNU General Public License, 13
- Go Around, 147

- Goeke-Smith, Jeff, [191](#), [196](#)
- Gold, Michael, I., [197](#)
- gouverne de direction, [57](#)
- gouverne de profondeur, [57](#), [67](#)
- GPL, [13](#)
- GSHHS data, [191](#)
- gyro compass, [113](#)

- Habibe, [197](#)
- head up display, [67](#), [191](#)
- help, [65](#)
- Hill, Mike, [197](#)
- histoire, [189](#)
 - scènes, [190](#)
- history
 - aircraft, [191](#)
 - environment, [192](#)
 - user interface, [192](#)
- Hofman, Erik, [190](#), [197](#), [203](#)
- hot air balloon, [198](#)
- Hotchkiss, Charlie, [191](#), [197](#)
- HTTP server, [193](#)
- http server, [44](#)
- HUD, [43](#), [67](#), [191](#), [194](#), [197](#)
- HUD (Head Up Display
 - full mode), [126](#)
- HUD (Head Up Display), [126–128](#)

- IA, [65](#)
- icing
 - modelling, [16](#)
- IFR, [85](#)
- ignition switch, [113](#)
- inclinometer, [112](#)
- indicateur de la gouverne de direction, [67](#)
- indicateur de taux de virage, [67](#)
- indicateur des ailerons, [67](#)
- installer des aéronefs, [23](#)
- instrument panel, [42](#), [191](#)
- Internet, [193](#)

- Jackson, Bruce, [197](#)
- Jackson, Bruce, [190](#)
- joystick, [38](#), [46](#), [56](#), [193](#)
 - .fgsrc, [53](#)
- joystick settings, [193](#)
- joystick/self detection, [193](#)
- joysticks, [15](#)
- Justus, Maik, [197](#)

- Kaaven, Ove, [197](#)
- Kaszeta, Richard, [197](#)
- keyboard, [90](#)
 - numeric, [91](#)
 - uppercase and lowercase keys, [90](#)
- Knienieder, Tom, [197](#)
- Koradi, Reto, [197](#)
- Korpela, Eric, [189](#)
- Kuehne, Bob, [197](#)

- Laird, Kyler B., [198](#), [203](#)
- landing, [115](#)
 - aid signals, [95](#)
- LaRCsim, [190](#), [191](#), [197](#), [198](#), [200](#)
- latitude, [40](#), [67](#)
- leaflet, [7](#)
- light textures, [191](#)
- Linux, [12](#), [13](#), [15](#), [189](#)
- listes de diffusion, [185](#)
- Livermore, [134](#)
- longitude, [40](#), [67](#)
- Luff, David, [192](#), [198](#)

- météo, [64](#)
- magnetic compass, [113](#)
- magneto, [103](#)
- magneto switch, [192](#)
- mailing lists, [194](#)
- maintien d'altitude, [58](#), [60](#)
- maintient de cap, [58](#), [60](#)
- manches, [15](#)
- manette des gaz, [67](#)
- manette des gaz automatique, [58](#), [60](#)
- manquants, [187](#)
- Mayer, Christian, [192](#), [193](#), [198](#)
- Megginson, David, [84](#), [191–193](#), [198](#), [203](#)
- menu, [192](#)
- menu system, [202](#)
- MetaKit, [201](#)
- Metro Works, [201](#)
- Microsoft, [11](#)
- Mitchell, Eric, [190](#), [198](#)
- mixture, [104](#), [142](#)
 - lever, [104](#)
 - optimisation, [105](#)
- modèle de vol UIUC, [16](#)
- Modèles de dynamique de vol, [16](#)
- modes de vue, [59](#)
- modes souris, [60](#)
- Moore Cameron, [185](#)
- Moore, Cameron, [198](#), [203](#)
- Morken, Anders, [198](#)
- moteur
 - démarrage
 - avion à réaction, [56](#)
 - pistons, [55](#)
 - turbopropulseurs, [56](#)
 - démarrer, [55](#)

- mouse
 - normal mode, 92
 - rudder control, 98
 - view mode, 92
 - yoke mode, 92
- mouse pointer, 42
- MS DevStudio, 188
- MSVC, 188, 195
- multi-engine support, 192
- multi-lingual conversion tools, 198
- multijoueurs, 65
- Multiplayer, 69
- multiplayer code, 193
- Multiple Displays, 74
- Murr, David, 189
- Murta, Alan, 198

- NAV, 113
- Navion, 191
- NDB, 113
- Nelson, Phil, 198
- network, 193
- network options, 44
- networking code, 193, 196
- Novikov, Alexei, 198
- NumLock, 56
- NVIDIA, 6

- offset, 43
- Olson, Curt, 189–193, 198, 203
- OpenGL, 6, 14, 15, 17, 186, 190, 193, 197
 - drivers, 15
- options
 - aircraft, 38
 - Aircraft Systems, 43
 - debugging, 45
 - environment, 41
 - features, 38
 - flight model, 39
 - general, 37
 - HUD, 43
 - initial position, 39
 - IO, 44
 - joystick, 46
 - network, 44
 - orientation, 39
 - rendering, 41
 - route, 44
 - sound, 38
 - time, 43
 - waypoint, 44
- Options d'affichage, 59
- orientation, 40
- OS/2, 189

- Others, 202

- pédales de palonnier, 56
- palonniers, 15
- panel, 198, 199
- Paul, Brian, 199
- pause, 60
- pedal, 46
- Peden, Tony, 193, 199
- Peel, Robin, 192, 199
- Perry, Alex, 199, 203
- Perry, Dave, 203
- PFE, 53
- pilote automatique, 59, 60, 63
- pilote automatique générique, 59
- pitch, 112
- Playback, 75
- PLIB, 192, 193
- position, 63
- pré-requis système, 14
- problèmes, 185
 - généraux, 186
 - Linux, 187
 - Windows, 188
- programmers, 194
- property manager, 193
- proposition, 189

- quitter, 62

- réinitialiser un vol, 62
- raccourcis clavier
 - configuration, 56
- radio, 137
- radio stack, 113, 192
- random ground objects, 191
- Reid-Hillview, 134
- Reinhard, Friedemann, 191, 199
- Reinholdtsen, Petter, 199
- Riley, William, 191, 199
- Ross, Andy, 192, 199, 203
- routines graphiques, 189
- RPM indicator, 113
- rudder, 112
 - keyboard control, 98
 - mouse control, 98
- runway lighting, 191

- scènes, 19
- scènes, 190
- scenery, 191
 - additional, 19
- Scenery Database, 202
- scenery directory

- path, 37
- scenery subsystem, 198
- Schlyter, Paul, 199
- Schoenemann, Chris, 199
- Schubert, Phil, 199
- Sectional, 134
- See how it flies, 85
- Selig, Michael, 192, 203
- Shewchuk, Jonathan, 200
- shutdown, 117
- side-slip, 109
- signaler un problème, 185
- Sikic, Gordan, 200
- SimGear, 193
- Simulateur de vol
 - libre, 190
- Smith, Michael, 200
- souris, 60
- souris, actions, 60
- souris, contrôle, 61
- souris, normal, 60
- souris, vue, 61
- speed
 - units
 - knot [nautical mile per hour], 99
- spin, 109
- Spott, Martin, 200, 203
- stall, 109
- starter, 113
- starting the engine, 113
- Startup latitude, 40
- Startup longitude, 40
- static objects, 191
- stations de travail, 190
- Stockill, Jon, 200
- Sun-OS, 13, 189
- Systèmes d'exploitation, 12
- system.fgfsrc, 37, 193

- tableau de bord, 59, 66
- tachometer, 98
- Talsma, Durk, 192, 200
- taux d'images, 15
- taux de rafraîchissement, 190
- telnet server, 44
- TerraGear, 193
- Text To Speech, 76
- textures, 190, 198
- throttle lever, 104
- thunderstorms, 192
- time options, 43
- Torvalds, Linus, 13
- Traffic Pattern, 144
- train d'atterrissage, 57, 59

- triangle program, 200
- trim, 57, 110
- Troester, Kai, 187, 203
- troubles
 - mouse speed, 94
 - stuttered simulation, 89
- turn coordinator, 97
- turn indicator, 112
- tutorial, 84

- U. S. Geological Survey, 190, 200
- UIUC, 192, 200
- UNIX, 190

- Vallevand, Mark, 200
- van der Mac, Sam, 198
- van Sickle, Gary, R., 193, 201
- variables d'environnement, 27
- VASI, 147
- velocity rages, 112
- verrouillage de la roue de queue, 57, 59
- vertical speed indicator, 112
- VFR, 85, 114
- view
 - changing, 91
 - instant replay, 91
- views, 193
- Vine, Norman, 191, 193, 201, 203
- visual flight rules, 114
- vitesse, 67
- VMap0 data, 191
- Voegtli, Roland, 201
- volant, 56
- volets, 57, 59
- Volpe, Carmelo, 201
- VOR, 113
- vue frustrum culling, 190

- Walisser, Darrell, 201, 203
- weather, 198
- wiki, 7
- Williams, Ed, 201
- Wilson, Jim, 201
- window size, 42
- Windows, 15
- Windows 95/98/ME, 12
- Windows 95/NT, 189
- Windows NT/2000/XP/Seven, 12
- winds, 192
- windsock, 120
- Wippler, Jean-Claude, 201
- wireframe, 42
- Wood, Charles, 85
- Woodsoup, 202

Wright Flyer, [17](#)

XFree86, [187](#)

YASim, [16](#)

yoke, [46](#), [92](#)

 mouse yoke mode, [92](#), [95](#)

 pulling, [94](#)

Zeh, Allan, [202](#)

zlib library, [196](#)