

Manual de FlightGear

Michael Basler, Martin Spott,
Stuart Buchanan, Jon Berndt, Bernhard Buckel, Cameron Moore,
Curt Olson, Dave Perry, Michael Selig, Darrell Walisser,
y otros autores



Manual de FlightGear
30 de julio de 2021
Basado en FlightGear versión 2020.3.11
Editor: Scott Giese

Foto de portada:

Cessna 172P aparcada en el Valle de Aosta [LIMV], Italia, autor: Gilberto Agostinho, licencia GPL

La Cessna 172 Skyhawk es una avioneta monomotor de ala alta con capacidad para cuatro tripulantes. Voló por primera vez en 1955 y continúa en producción, lo que la hace la aeronave con más unidades producidas de la historia.

Desarrolladores de la Cessna 172P:

David Megginson (autor original),

Gilberto Agostinho (gsagostinho), Wayne Bragg (wbragg), Juan Vera del Campo (Juanvvc),

Daniel Dubreuil (Dany93), Jonathan Redpath (legoboyvdlp), Jonathan Schellhase (dg-505),

Tuomas Kuosmanen (tigert), Anders Gidenstam (AndersG), Waldo Kitty (wkitty42),

Jarl Arntzen (jarlarntzen), algefaen, Horacio, D-ECHO, onox, thevirtualfer

Novedades respecto a anteriores ediciones:

Capítulo 4, instrucciones de la interfaz gráfica de inicio por Stuart Buchanan

Capítulo 8, nuevas imágenes por Jonathan Redpath

Capítulo 8, coordinador de giro por Scott Giese

Índice general

1. Prólogo	9
1.1. Lectura condensada	10
1.2. Instrucciones para los más impacientes	10
1.3. Lectura adicional	11
I Instalación	13
2. Te gustaría volar libre? Prueba <i>FlightGear</i>!	15
2.1. Otro simulador de vuelo más?	15
2.2. Requisitos del sistema	17
2.3. Elegir una versión	18
2.4. Modelos de Dinámica de Vuelo	19
2.5. Acerca de esta guía	19
3. Previo al vuelo: Instalar <i>FlightGear</i>	21
3.1. Instalación de escenario	21
3.1.1. Descarga de escenarios mientras vuelas (<i>TerraSync</i>)	21
3.1.2. Instalación manual de escenario	22
3.1.3. Crear tu propio escenario	24
3.2. Instalación de aeronaves	24
3.3. Instalación de la documentación	24
II Volar con <i>FlightGear</i>	25
4. Despegue: cómo arrancar el programa	27
4.1. Arrancar el simulador	27
4.2. Arrancar desde la línea de comandos	30
4.2.1. FG_ROOT	30
4.2.2. FG_SCENERY	30
4.2.3. Arrancar el simulador en Windows	30
4.2.4. Arrancar el simulador en Unix/Linux	31
4.2.5. Arrancar el simulador en Mac OS X	32

4.3.	Opciones de línea de comandos	32
4.3.1.	Opciones Generales	32
4.3.2.	Características	34
4.3.3.	Sonido	34
4.3.4.	Aeronave	34
4.3.5.	Modelos de vuelo	35
4.3.6.	Posición inicial y orientación	35
4.3.7.	Opciones de entorno	37
4.3.8.	Opciones de renderizado	38
4.3.9.	Opciones de HUD	40
4.3.10.	Opciones de Sistemas de Aeronave	40
4.3.11.	Opciones de horario	41
4.3.12.	Opciones de red	41
4.3.13.	Opciones de rutas/ <i>waypoints</i>	42
4.3.14.	Opciones de entrada/salida	42
4.3.15.	Opciones de depuración	43
4.4.	Soporte para joysticks	44
5.	En vuelo: uso de los instrumentos, teclado y menús	45
5.1.	Arrancando el motor	45
5.1.1.	Aeronaves de pistón	45
5.1.2.	Aeronaves turbopropulsadas	46
5.1.3.	Aeronaves a reacción	46
5.2.	Controles de teclado	46
5.2.1.	Controles de la aeronave	46
5.2.2.	Controles del simulador	47
5.2.3.	Controles de piloto automático	48
5.3.	Controles de ratón	49
5.3.1.	Modo normal	49
5.3.2.	Modo de control de vuelo	50
5.3.3.	Modo de dirección de vista	50
5.4.	Entradas del menú	50
5.5.	El panel de instrumentos	55
5.6.	La pantalla frontal o <i>HUD</i>)	55
6.	Funciones avanzadas	59
6.1.	Multijugador	59
6.1.1.	Guía rápida	59
6.1.2.	Otras formas de conectarse	59
6.1.3.	Solución de problemas	60
6.2.	Portaaviones	61
6.2.1.	Arrancando desde el portaaviones	61
6.2.2.	Uso de la catapulta	62
6.2.3.	Aproximación al portaaviones – TACAN	62

6.2.4. Aterrizando en el portaaviones	62
6.3. Atlas	63
6.4. Varias pantallas	63
6.5. Varias máquinas	63
6.5.1. Conceptos básicos	64
6.5.2. Configuración básica	64
6.5.3. Configuración avanzada	64
6.6. Grabaciones y reproducción	65
6.7. Soporte de Texto a Voz con Festival	65
6.7.1. Instalación de Festival	66
6.7.2. Arrancar <i>FlightGear</i> con soporte de voz	66
6.7.3. Resolución de problemas	67
6.7.4. Instalación de voces adicionales	67
6.8. Repostaje aéreo	68
6.8.1. Procedimiento	68
6.8.2. Repostaje en multijugador	69
III Tutorials	71
7. Tutorials	73
7.1. In-flight Tutorials	73
7.1.1. Cessna 172P Tutorials	73
7.2. FlightGear Tutorials	74
7.3. Other Tutorials	74
8. A Basic Flight Simulator Tutorial	77
8.1. Foreword	77
8.2. Starting Up	78
8.2.1. Using the Launcher	79
8.3. The First Challenge – Flying Straight	81
8.4. Basic Turns	88
8.5. Taxiing on the ground	90
8.5.1. Airspeed	91
8.6. Advanced Turns	94
8.7. A Bit of Wieheisterology	96
8.7.1. Engine control	96
8.7.2. Wings and speed	99
8.7.3. The flaps	101
8.7.4. The stall	102
8.7.5. The trim	103
8.7.6. What direction am I flying?	104
8.7.7. A look around the panel	105
8.8. Let’s Fly	108

8.8.1.	A realistic take off	108
8.8.2.	Landing	109
8.8.3.	Engine Shutdown	111
8.8.4.	Aborted Landing	112
8.9.	Dealing with the Wind	112
8.9.1.	Crosswind Take Off	114
8.9.2.	Crosswind Landing	115
8.9.3.	Taxiing in the Wind	115
8.10.	The autopilot	116
8.11.	What Next?	117
8.12.	Thanks	117
8.13.	Flying Other Aircraft	118
8.13.1.	How to land the Cherokee Warrior II	118
8.13.2.	How to take off and land the Piper J3 Cub	119
8.13.3.	How to take off and land a jet	120
8.13.4.	How to take off and land the P-51D Mustang	125
8.13.5.	How to take off and land the B-52 Stratofortress	125
9.	A Cross Country Flight Tutorial	129
9.1.	Introduction	129
9.1.1.	Disclaimer and Thanks	130
9.2.	Flight Planning	130
9.3.	Getting Up	132
9.3.1.	Preflight	132
9.3.2.	ATIS	133
9.3.3.	Radios	133
9.3.4.	Altimeter and Heading Indicator	134
9.3.5.	Take-Off	137
9.4.	Cruising	137
9.4.1.	The Autopilot	137
9.4.2.	Navigation	139
9.4.3.	Mixture	139
9.5.	Getting Down	140
9.5.1.	Air Traffic Control	142
9.5.2.	The Traffic Pattern	142
9.5.3.	Approach	144
9.5.4.	VASI	145
9.5.5.	Go Around	146
9.5.6.	Clearing the Runway	146
10.	An IFR Cross Country Flight Tutorial	149
10.1.	Introduction	149
10.1.1.	Disclaimers	149
10.2.	Before Takeoff	150

10.2.1. Flight Planning	151
10.2.2. VHF Omnidirectional Range	151
10.2.3. How High Are We Really?	154
10.3. Takeoff	154
10.4. In the Air	155
10.4.1. George I	155
10.4.2. MISON Impossible	156
10.4.3. George II	158
10.4.4. Staying the Course	158
10.4.5. Yet More Cross-Checks	159
10.5. Getting Down	160
10.5.1. Instrument Approach Procedures	160
10.5.2. Non-directional Beacons	162
10.5.3. Procedure Turn	164
10.5.4. Chasing the Needle	165
10.5.5. FOOTO Time	166
10.5.6. George III	166
10.5.7. ILS Landings	167
10.5.8. Intercepting the Localizer	168
10.5.9. Intercepting the Glide Slope	168
10.5.10.Touchdown, Almost	169
10.5.11.A Confession	169
10.5.12.Touchdown, Not	170
10.5.13.Touchdown	171
10.6. Epilogue	171
11. A Helicopter Tutorial	175
11.1. Preface	175
11.2. Getting started	176
11.3. Lift-Off	177
11.4. In the air	177
11.5. Back to Earth I	178
11.6. Back to Earth II	178
IV Appendices	181
A. Missed approach: If anything refuses to work	183
A.1. FlightGear Problem Reports	183
A.2. General problems	184
A.3. Potential problems under Linux	185
A.4. Potential problems under Windows	186

B. Landing: Some further thoughts before leaving the plane	187
B.1. A Sketch on the History of <i>FlightGear</i>	187
B.1.1. Scenery	188
B.1.2. Aircraft	189
B.1.3. Environment	190
B.1.4. User Interface	190
B.2. Those, who did the work	191
B.3. What remains to be done	199
C. Main Index	201
D. Index of Command Line Options	207

Capítulo 1

Prólogo

FlightGear es un simulador de vuelo de código abierto desarrollado de forma cooperativa en internet por un grupo de aficionados a la simulación de vuelo y la programación. El propósito de este “Manual de *FlightGear*” es el de guiar a los nuevos usuarios durante la puesta a punto de *FlightGear*, y ayudarles a comenzar a volar. El manual no pretende documentar minuciosamente todas las características y extensiones del simulador, pero es el mejor modo de comenzar a explorar el universo *FlightGear*.

Esta versión del manual está basada en la versión 2020.3.11 de *FlightGear*. La mayoría de la información es aplicable a versiones anteriores, pero puede que algunas de las características descritas no estén disponibles.

El manual está organizado en tres partes:

Parte I: Instalación

El Capítulo 2, [Te gustaría volar libre? Prueba *FlightGear*!](#), describe *FlightGear* en líneas generales, explica la filosofía detrás del proyecto y enumera los requisitos del sistema.

El Capítulo 3, [Previo al vuelo: Instalar *FlightGear*](#), contiene instrucciones de instalación del programa y explica cómo instalar paisaje y aeronaves adicionales.

Parte II: Volar con *FlightGear*

El Capítulo 4, [Despegue: cómo arrancar el programa](#), explica cómo usar la interfaz gráfica de inicio, así como las principales opciones de línea de comandos. También describe el uso de ficheros de configuración.

El Capítulo 5, [En vuelo: uso de los instrumentos, teclado y menús](#), explica el funcionamiento del simulador, es decir, cómo volar con *FlightGear*. Incluye una lista completa de las teclas predefinidas, una visión general de los menús más importantes, descripciones detalladas del panel de instrumentos y del HUD (*Head-Up Display* o Pantalla de Visualización Frontal), así como algunos trucos para usar el ratón de forma efectiva.

El Capítulo 6, [Funciones avanzadas](#), introduce algunas de las características más avanzadas de *FlightGear*.

Parte III : Tutoriales

El Capítulo 7, [Tutorials](#), propone varios tutoriales para los nuevos pilotos.

El Capítulo 8, [A Basic Flight Simulator Tutorial](#), introduce al piloto en los principios básicos de vuelo con ejemplos sobre los instrumentos, controles de vuelo y otros aparatos disponibles en *FlightGear*.

El Capítulo 9, [A Cross Country Flight Tutorial](#), describe un vuelo de travesía en los alrededores de San Francisco que puede realizarse con los paisajes y aeronaves instalados por defecto.

El Capítulo 10, [An IFR Cross Country Flight Tutorial](#), describe un vuelo de travesía similar al anterior, esta vez utilizando los instrumentos del avión. Esto nos permite volar en cielos nubosos, sujetos a las Reglas de Vuelo Instrumental o *IFR* (del inglés *Instrument Flight Rules*).

Apéndices

En el Apéndice A, [Missed approach: If anything refuses to work](#), te guiamos a través de algunos problemas típicos que puedes encontrar mientras usas *FlightGear*.

En el Apéndice B, [Landing: Some further thoughts before leaving the plane](#) nos gustaría reconocer la labor de los numerosos colaboradores que se han ganado nuestro respeto contribuyendo a crear este fantástico simulador. Asimismo, explicamos las ideas básicas del proceso de desarrollo de *FlightGear* y algunas tareas que todavía quedan por hacer.

1.1. Lectura condensada

Para los que no pretenden leer este manual de principio a fin, sugerimos leer las siguientes secciones en el orden indicado, con el fin de obtener los conocimientos básicos para comenzar a volar:

Instalación :	Previo al vuelo: Instalar <i>FlightGear</i>
Arrancando el simulador :	Despegue: cómo arrancar el programa
Uso del simulador :	En vuelo: uso de los instrumentos, teclado y menús

1.2. Instrucciones para los más impacientes

La mayoría de nosotros detesta leer manuales. Si el driver de tu tarjeta gráfica soporta OpenGL (echa un vistazo a la documentación; por ejemplo, las tarjetas NVIDIA suelen soportarlo) y tu sistema operativo es Windows, Mac OS o Linux, lo más probable es que puedas saltarte la Parte I del manual y usar un programa pre-compilado. Encontrarás enlaces de descarga, e instrucciones de instalación en:

<https://www.flightgear.org/download/>.

Si utilizas Linux, es probable que tu distribución disponga de paquetes de *FlightGear* en sus repositorios.

Una vez descargado e instalado el programa, el Capítulo 4 contiene instrucciones para arrancar el simulador.

1.3. Lectura adicional

Este manual debería darte suficiente información para usar el simulador. No obstante, recomendamos echar un vistazo a estas otras fuentes, especialmente si encuentras problemas:

- La **hoja de referencia** con las teclas más comunes te está esperando en la carpeta `FlightGear/Docs/FGShortRef.pdf` del paquete básico.
- **Documentación adicional** sobre algunas características y opciones adicionales, disponible en la carpeta `FlightGear/Docs/`.
- La **wiki** oficial de *FlightGear* está disponible en <https://wiki.flightgear.org>.
- El **foro** oficial de *FlightGear* está disponible en <https://forum.flightgear.org>.

Parte I

Instalación

Capítulo 2

Te gustaría volar libre? Prueba *FlightGear*!

2.1. Otro simulador de vuelo más?

Siempre has querido aprender a pilotar un avión, pero el alto coste o los exigentes controles te lo impiden? Eres un piloto acreditado y buscas mejorar tus habilidades sin asumir el alto coste asociado a volar? Te gustaría entrenar maniobras peligrosas sin experimentar el estrés y los riesgos asociados? O quizás sólo buscas un poco de diversión con un juego serio y sin violencia? Si la respuesta es sí a alguna de estas preguntas, los simuladores de vuelo son para ti.

Puede que ya hayas probado el *Flight Simulator* de Microsoft o algún otro simulador destinado al consumidor casual. Disponibles a un precio de aproximadamente \$50, adquirir una copia no debería ser un problema, especialmente teniendo en cuenta que cualquier simulador de vuelo serio requiere un ordenador del orden de los \$1500 para arriba.

Con tantos simuladores disponibles en el mercado, por qué invertir miles de horas de diseño y desarrollo en un simulador libre y gratuito? Bueno... aquí están las razones más importantes:

- Los simuladores comerciales tienen un gran fallo: están hechos por un pequeño grupo de desarrolladores; éstos definen las características de acuerdo a sus objetivos, exponiendo las mínimas interfaces posibles a los usuarios. Cualquiera que haya intentado contactar con un equipo de desarrollo sabrá lo difícil que es hacerse oír. A diferencia de éstos, *FlightGear* está diseñado por y para la gente, y absolutamente todo es público.
- Habitualmente, los simuladores comerciales comprenden una mezcla de realismo y facilidad de uso. La mayoría de desarrolladores comerciales deben atraer a un amplio abanico de usuarios, desde pilotos profesionales hasta jugadores esporádicos, pasando por pilotos *amateur*. En la práctica, el resultado está guiado por las fechas de entrega y el presupuesto. Al ser *FlightGear* libre y gratuito, dicho compromiso no es necesario. No hay ninguna editorial presionando a los desarrolladores, y somos todos voluntarios persiguiendo nuestras propias ideas. Asimismo, tenemos libertad para ayudar a usuarios que no interesan a las compañías, como por ejemplo la comunidad científica.

- Debido a su licencia propietaria, el acceso al código y recursos de los simuladores comerciales están celosamente protegidos. Esto limita las posibilidades de que otros desarrolladores externos, posiblemente con grandes ideas y habilidades, puedan contribuir al desarrollo y dirigir el rumbo de dichos proyectos. *FlightGear* permite que gente con habilidades muy diversas pueda contribuir y tener una gran influencia en el proyecto. Ser parte de un proyecto del tamaño y complejidad de *FlightGear* es enormemente gratificante y motivo de orgullo, al tener la oportunidad de escribir el futuro de este gran simulador.
- Y sobre todo, porque es divertido y punto! De alguna forma nos parecemos a esos pilotos de la vida real que deciden fabricar su propio avión con un kit o directamente desde cero. “Cualquiera” puede comprar un avión, pero fabricárselo uno mismo es más auténtico.

Todos estos puntos y más forman la base con la que creamos *FlightGear*. Con esta motivación, nos hemos puesto como meta el crear un simulador de vuelo de máxima calidad, de uso civil, multiplataforma, de código abierto, soportado por los usuarios mismos, y fácil de extender. Entrando más en detalle:

- **De uso civil:** El proyecto está enfocado principalmente a la simulación de vuelo civil. También es adecuado como simulador de vuelo general aparte de aeronaves civiles. Nuestro objetivo a largo plazo es certificar *FlightGear* según la FAA (*Federal Aviation Administration* o Administración Federal de Aviación de los EEUU) como dispositivo de entrenamiento de vuelo. Por desgracia para algunos usuarios, por el momento no es un simulador de combate; no obstante, no se descarta la posibilidad. Simplemente, todavía no hemos tenido ningún desarrollador lo suficientemente interesado en los sistemas necesarios para la simulación de combate.
- **Multiplataforma:** Los desarrolladores intentan escribir código fuente lo más independiente posible de la plataforma. La experiencia muestra que los usuarios de simuladores de vuelo usan una amplia variedad de hardware y sistemas operativos. El código actual soporta los siguientes Sistemas Operativos:
 - Linux (cualquier distribución reciente),
 - Windows 10/8/7/Vista (Intel/AMD),
 - BSD UNIX,
 - Mac OS X

Para obtener un rendimiento óptimo, se recomienda utilizar un SO reciente con un mínimo de 4 GB de RAM, drivers gráficos actualizados y una tarjeta gráfica con al menos 1 GB de memoria.

- **Abierto:** El proyecto no se restringe a un pequeño grupo de desarrolladores. Cualquiera con ganas de contribuir es bienvenido. El código (incluida la documentación) está bajo los términos de la licencia GPL (del Inglés GNU General Public License).

La licencia GPL no es fácil de entender. En pocas palabras, dice que se puede copiar y distribuir el programa libremente. Cualquiera puede modificar el programa, distribuirlo y

cobrar por ello. Sin embargo, al distribuir el software, se debe distribuir el código fuente con éste, y se debe mantener la licencia original. En definitiva:

"Puedes hacer cualquier cosa con el software excepto convertirlo en no-libre.

El texto completo de la licencia GPL está disponible como parte del código fuente de *FlightGear* o en el siguiente enlace:

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

- **Soportado por los usuarios y fácil de extender:** A diferencia de otros simuladores comerciales, en *FlightGear* los formatos de escenarios y aeronaves, las variables internas, APIs, etc. son visibles y están documentados. Incluso sin documentación (la cual hay que escribir en algún momento), uno puede ir al código fuente y ver cómo funciona. Es un objetivo básico de los desarrolladores el implementar un simulador fácil de extender para los diseñadores de de escenarios, ingenieros de paneles, escritores de aventuras o rutinas de ATC, compositores y demás. Esperamos que el proyecto, incluidos desarrolladores y usuarios, se beneficie de la creatividad y el ingenio de los miles de aficionados de todo el mundo.

Sin lugar a dudas, el éxito de Linux, comenzado por Linus Torvalds, ha inspirado a varios de los desarrolladores. No sólo ha demostrado que es posible desarrollar un software complejo con un equipo colaborando online, sino que además la calidad de dicho software puede superar a la de otros productos comerciales similares.



Figura 2.1: Aproximación incorrecta a PHNL (Daniel K. Inouye International, Honolulu)

2.2. Requisitos del sistema

Comparado con otros simuladores de vuelo recientes, los requisitos del sistema de *FlightGear* no son muy elevados. Un AMD x64 o Intel x64 deberían ser suficientes para correr *Flight-*

Gear, siempre y cuando se disponga de una tarjeta gráfica decente.

Es importante que el driver de la tarjeta soporte OpenGL. Si no sabes lo que es, mejor leer su introducción (en Inglés):

“OpenGL® is the most widely adopted 2D and 3D graphics API in the industry...”.

<https://www.khronos.org/opengl/>

FlightGear no funciona con tarjetas gráficas bajo Direct3D / DirectX. A diferencia de OpenGL, Direct3D tiene una interfaz propietaria restringida al Sistema Operativo Windows.

Puede que consigas correr *FlightGear* en un ordenador con una tarjeta gráfica sin soporte para OpenGL, o incluso en sistemas sin ningún tipo de hardware gráfico. No obstante, la ausencia de aceleración gráfica puede hacer sufrir a la máquina más potente. El típico síntoma de falta de aceleración gráfica es un frame rate por debajo de 1 frame por segundo.

Cualquier tarjeta gráfica con soporte para OpenGL debería valer. Puedes encontrar drivers OpenGL para Windows en la web del fabricante de tu tarjeta gráfica. Ten en cuenta que, a veces, el proveedor de los drivers de OpenGL es el fabricante del chip, no el de la tarjeta. Si estás pensando en comprar una tarjeta para correr *FlightGear*, recomendamos una NVIDIA GeForce, ya que suelen tener mejor soporte OpenGL que las AMD/ATI Radeon. 1GB de memoria gráfica dedicada es suficiente; muchos usuarios ejecutan *FlightGear* con menos que eso.

Los efectos de sonido funcionarán con cualquier tarjeta de sonido. Gracias a su diseño flexible, *FlightGear* soporta un amplio abanico de joysticks y pedales tanto en Linux como en Windows. *FlightGear* también acepta conexiones con sillas de simulación de vuelo.

El desarrollo de *FlightGear* ocurre principalmente bajo Linux, un clon libre de UNIX (junto con varios programas GNU) desarrollado de forma cooperativa de forma parecida a *FlightGear*. *FlightGear* funciona también (y está desarrollado en parte) bajo varias versiones de Windows. Es posible compilar *FlightGear* en Mac OS X y otros tipos de sistemas UNIX/X11. Mientras se tenga un compilador decente instalado, *FlightGear* puede ser compilado en todas esas plataformas. El compilador principal para todas las plataformas es el compilador GNU C++ (en Win32, Cygnus Cygwin).

Si quieres ejecutar *FlightGear* bajo Mac OS X, es necesario al menos Mac OS X 10.4. Los requisitos mínimos de hardware son un Power PC G4 1 GHz o bien un Intel Mac, aunque recomendamos un MacBook Pro, Intel iMac, Mac Pro, o Power Mac (Power PC G5) para volar sin problemas.

2.3. Elegir una versión

Recomendamos utilizar la versión oficial más reciente; cada año se lanzan nuevas versiones. Puedes usar los ejecutables pre-compilados disponibles en:

<https://www.flightgear.org/download/>

Si quieres disfrutar del código más actualizado (y, posiblemente, con más errores), puedes clonar los fuentes en:

<https://sourceforge.net/p/flightgear/flightgear/ci/next/tree/>

y seguir las instrucciones sobre cómo instalar *FlightGear* desde los fuentes en:

<https://wiki.flightgear.org/Git>

2.4. Modelos de Dinámica de Vuelo

En el pasado, *FlightGear* estaba basado en un modelo de vuelo heredado (junto con el Navion) de LaRCsim. Dicho modelo tenía varias limitaciones (especialmente el hecho de que casi todo estaba definido en el código en vez de usar ficheros de configuración), lo que condujo al desarrollo de otros modelos de vuelo. Como resultado, *FlightGear* soporta varios modelos, configurables durante el uso del simulador.

- El más popular es probablemente el modelo de vuelo JSB desarrollado por Jon Berndt. Este modelo es parte de un proyecto separado llamado *JSBSim*:

<http://jsbsim.sourceforge.net/>.

- Andrew Ross creó otro modelo llamado *YASim*, del Inglés *Yet Another Simulator*. El diseño de *YASim* es radicalmente diferente a la mayoría de modelos de vuelo, al usar información de geometría en lugar de coeficientes aerodinámicos. El modelado de helicópteros de *YASim* es especialmente avanzado.

- Christian Mayer ha desarrollado un modelo de vuelo basado en un globo de aire. Curt Olson lo ha integrado en un modo “UFO” que permite volar cómodamente entre dos puntos.

- Por último, está el modelo de vuelo de UIUC, desarrollado por un equipo en la Universidad de Illinois en Urbana-Champaign. Este modelo estaba especializado en el modelado de aeronaves en condiciones de congelación, pero actualmente incorpora aerodinámica “no lineal”, con comportamientos más realistas en condiciones extremas, como por ejemplo entrada en pérdida y ángulos de ataque límite. Dos buenos ejemplos son el ala delta Airwave Xtreme 150 y el avión de los Hermanos Wright de 1903. Puedes encontrar más detalles sobre este modelo en:

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

Asimismo, es posible conectar *FlightGear* a una FDM externa ejecutándose en otro ordenador mediante un pipe en la máquina local, aunque no recomendamos esta configuración a nadie que acabe de empezar en el mundo de *FlightGear*.

2.5. Acerca de esta guía

No hay nada en esta guía que no puedas encontrar en otra parte. Como dijo Montaigne, “simplemente he recogido un manajo de flores de otros hombres, sin haber contribuido nada de mi cosecha salvo el cordel que los mantiene juntos”. La mayoría de la información (no toda, por fortuna) está disponible en la web de *FlightGear* en:

<https://www.flightgear.org>

El *Manual de FlightGear* pretende ser un primer paso hacia la documentación completa de *FlightGear*. Está pensado para aquellos usuarios a los que no les interesan los detalles internos de OpenGL, no quieren compilar su propia librería de escenarios... Con un poco de suerte, en el futuro habrá una *Guía del Programador de FlightGear*, una *Guía de Diseño de Escenario de FlightGear*, que describa las herramientas de escenario compiladas actualmente en *TerraGear*; y un paquete para la *Escuela de Pilotos de FlightGear*.

Te rogamos nos ayudes a mejorar este documento enviándonos correcciones, mejoras, sugerencias y traducciones. Os invitamos a todos a contribuir descripciones con vuestras configuraciones (tarjetas gráficas, sistemas operativos, etc.). Estaremos encantados de incluir éstas en futuras versiones del *Manual de FlightGear* (con el merecido crédito a los autores).

Capítulo 3

Previo al vuelo: Instalar *FlightGear*

Antes de ejecutar *FlightGear*, es necesario instalarlo. Una vez hecho esto, puedes comenzar a instalar escenario y aeronaves adicionales.

Existen ejecutables pre-compilados con la última versión para varios sistemas operativos:

- Microsoft Windows
- Mac OS X
- Linux.

Puedes descargarlos en el siguiente enlace y seguir sus instrucciones de instalación:

<https://www.flightgear.org/download/>

3.1. Instalación de escenario

FlightGear dispone de ficheros de escenario para el mundo entero, permitiéndote volar a cualquier parte, desde los Himalayas hasta el Kansas rural. El paquete básico de *FlightGear* contiene escenario para Keflavik, Islandia. Para volar en otras zonas, es necesario descargar ficheros de escenario adicionales. Una vez escogido la zona de vuelo, puedes descargar los ficheros de escenario correspondientes o bien activar la descarga automática mediante *TerraSync*.

3.1.1. Descarga de escenarios mientras vuelas (*TerraSync*)

FlightGear puede descargar de Internet el escenario adecuado mientras vuelas. *FlightGear* se encargará de descargar el escenario a una carpeta separada del escenario instalado. Una de las mayores ventajas de *TerraSync* es que siempre descarga el escenario más actual del proyecto de Escenario del Mundo de *FlightGear*, y por tanto te permite descargar actualizaciones a los escenarios del Mundo, los cuales se sincronizan con versiones de *FlightGear*, de forma incremental.

Puedes activar *TerraSync* desde el lanzador o bien directamente desde dentro del simulador:

- **Desde el lanzador:** activa la opción *Descargar escenario automáticamente* en la pestaña *Configuración*, Sección *Descargas*.
- **Desde el simulador:** abre el menú *Archivo* → *Descargar escenarios*. Después, activa la opción *Enable automatic scenery download*.

Encontrarás más información acerca de *TerraSync* en la Wiki:

<https://wiki.flightgear.org/Es/TerraSync>

3.1.2. Instalación manual de escenario

Cada zona está disponible en un fichero comprimido *tar*, en piezas de 10x10 grados. El nombre de cada fichero corresponde a los grados de la zona, por ejemplo *w130n50.tgz*.

Para descargar los escenarios de la zona de vuelo actual, puedes activar la descarga automática mediante *TerraSync* a través del lanzador. También es posible descargar escenario fuera del simulador mediante un mapa interactivo en Internet, o usando la herramienta *TerraMaster*:

<https://www.flightgear.org/download/scenery>

<https://wiki.flightgear.org/TerraMaster>

Como alternativa, puedes ayudar al proyecto *FlightGear* adquiriendo el escenario del mundo al completo en el siguiente enlace:

<https://store.flightgear.org>

Una vez descargado el fichero a tu disco, busca el directorio *Scenery* en la instalación de *FlightGear*.

- En Windows, debería ser algo parecido a:

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

- En sistemas Unix, suele estar en:

```
/usr/local/share/FlightGear/data/Scenery.
```

- En Mac OS X, normalmente es:

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

Para instalar el escenario, descomprime los archivos en la carpeta *Scenery*. La mayoría de sistemas operativos proveen herramientas para descomprimir ficheros. Si no puedes descomprimirlos, instala una herramienta de descompresión, como por ejemplo (<https://www.7-zip.org>).

Por favor, no descomprimas los ficheros de escenario numerados dentro del fichero principal (por ejemplo, *958402.gz*). *FlightGear* se encarga de esto automáticamente.

Una vez hayas descomprimido el fichero, las carpetas *Terrain* y *Objects* deberían contener subdirectorios adicionales con el nuevo escenario.

Para usar el nuevo escenario, simplemente selecciona un aeropuerto de inicio dentro de la zona cubierta por éste. Si usas el lanzador, haz clic en el botón **Refrescar** antes de seleccionar el aeropuerto.

MS Windows Vista/7

Si estás en Windows Vista o Windows 7, es posible que el escenario (y las aeronaves) se descarguen en tu Tienda Virtual:

```
C:\Users\tu nombre\AppData\Local\
    VirtualStore\Program Files\FlightGear\Scenery
```

Si ocurre esto, deberás copiar las carpetas `Terrain` y `Objects` manualmente a la carpeta `Scenery` de *FlightGear* como se ha descrito anteriormente.

Mac OS X

Puedes instalar el escenario y las aeronaves descargadas desde el lanzador. Haz clic en **Instalar escenario adicional** en la pestaña *Complementors* y se abrirá una ventana de explorador. Selecciona uno o más ficheros y se instalarán en:

```
/Users/tu nombre/Library/Application Support/FlightGear/Scenery
```

Los formatos de escenario permitidos son zip, tar.gz, tgz, tar, y carpeta extraída. Si la instalación desde el lanzador falla por alguna razón, todavía tienes otra posibilidad. Abre la carpeta de datos haciendo clic en “Abrir carpeta de datos” en la pestaña “Otros” y se abrirá una ventana de búsqueda en la carpeta de datos. Arrastra una carpeta de aeronave a la carpeta `data/Aircraft` (o una carpeta de escenario a `data/Scenery`) y listo.

FG_SCENERY

Si prefieres mantener tu escenario descargado separado de tu instalación principal, puedes usar la variable de entorno `$FG_SCENERY`.

FlightGear utiliza esta variable para buscar los ficheros de escenario. Consiste en una lista de carpetas, que el simulador analiza en orden. Las carpetas están separadas por “:” en Unix (incluido Mac OS X) y “;” en Windows.

En Linux por ejemplo, el valor de `$FG_SCENERY` puede ser:

```
/home/jsmith/WorldScenery:
    /usr/local/share/Flightgear/data/Scenery
```

busca el escenario primero en:

```
/home/jsmith/WorldScenery
```

seguido de:

```
/usr/local/share/Flightgear/data/Scenery.
```

En Windows, `$FG_SCENERY` con un valor de:

```
C:\Program Files\FlightGear\data\Scenery;C:\data\WorldScenery
```

busca escenario primero en:

```
C:\Program Files\FlightGear\data\Scenery
```

seguido de:

```
C:\data\WorldScenery
```

Este documento no recoge la configuración de variables de entorno en otras plataformas.

3.1.3. Crear tu propio escenario

Si quieres crear tu propio escenario, echa un vistazo a TerraGear - la herramienta con la que se generan los escenarios de *FlightGear*:

<https://wiki.flightgear.org/Es/TerraGear>

El código de TerraGear, en continuo desarrollo, está disponible junto con el proyecto *FlightGear* en SourceForge:

<https://sourceforge.net/p/flightgear/terragear/ci/next/tree/>.

3.2. Instalación de aeronaves

El paquete básico de *FlightGear* contiene únicamente unos pocos aviones de los disponibles para *FlightGear*. No obstante, existen multitud de aeronaves desarrolladas por miembros de la comunidad, desde cazas de la Segunda Guerra Mundial hasta aviones de pasajeros como el Boeing 747.

Puedes descargar aeronaves en:

<http://home.flightgear.org/download/download-aircraft/>

Descarga el fichero y descomprímelo a la carpeta `data/Aircraft` de la instalación. Las aeronaves se descargan como ficheros `.zip`. Una vez descomprimidos, verás una subcarpeta en tu directorio `data/Aircraft` con las aeronaves dentro. La próxima vez que ejecutes *FlightGear*, podrás seleccionar la nueva aeronave.

También puedes instalar nuevas aeronaves desde el lanzador en cualquiera de las plataformas.

3.3. Instalación de la documentación

La mayoría de los paquetes mencionados anteriormente contienen una completa documentación acerca de *FlightGear*, incluyendo una versión en PDF del *Manual de FlightGear*, el cual puede ser abierto con un lector de PDFs como por ejemplo Adobe Reader, disponible en <http://get.adobe.com/reader/>

Una vez instalado, la versión HTML está disponible desde el menú *Ayuda*.

De forma adicional, el repositorio con los fuentes contiene una carpeta `docs-mini` con ideas y soluciones a problemas comunes. Éste es un buen lugar para encontrar lectura adicional.

Parte II

Volar con *FlightGear*

Capítulo 4

Despegue: cómo arrancar el programa

4.1. Arrancar el simulador



Figura 4.1: Listo para el despegue: *posición de inicio en Honolulu Intl., PHNL.*

FlightGear viene con un lanzador integrado para arrancar *FlightGear*. En Windows, haz doble clic en `FlightGear Launcher` dentro del menú Inicio o el icono de escritorio. Como alternativa, en cualquier sistema, puedes ejecutar la siguiente línea de comandos: `fgfs --launcher --language=es`. El lanzador te permite seleccionar tu aeronave, la posición de inicio (incluso puedes comenzar a diez millas de la pista de aterrizaje, o sobre un sistema de radio específico!), hora del día, activar o desactivar *TerraSync* o cambiar la meteorología, y mucho más.

Al ejecutar el lanzador por primera vez, aparecerá un diálogo que te permite configurar tu \$FG_ROOT, por ejemplo:

C:\Archivos de Programa\FlightGear\data ó

C:\Archivos de Programa\FlightGear 2020.3.11\data

Una vez configurada dicha variable, aparecerá la siguiente pantalla:



Figura 4.2: Resumen de la configuración del lanzador. Pulsa Volar! para arrancar.

Por defecto, el lanzador escoge la Cessna 172P en el aeropuerto de Keflavik. Simplemente pulsa el botón **Volar!** para arrancar el simulador. También puedes cambiar cualquier opción mediante los botones de la izquierda.

Puedes seleccionar un avión diferente haciendo clic en el botón **Aeronave** a la izquierda. *FlightGear* inicialmente dispone de dos aviones: la Cessna 172P y el UFO. Puedes instalar otros aviones fácilmente haciendo clic en el botón **Instalar**. También puedes instalar aeronaves descargadas en la web oficial o en hangares privados.

Ubicación te permite seleccionar la posición de inicio: una posición de parking, una pista de despegue, un punto en la senda de aproximación ILS, o un punto relativo a un VOR o FIX. El mapa muestra la posición seleccionada. Para seleccionar una posición diferente, pulsa el botón **Volver** e introduce el nombre del punto desde el cual deseas comenzar. Si la descarga automática está seleccionada en la página de *Configuración*, *FlightGear* descargará el escenario necesario automáticamente.

La página de *Ambiente* te permite seleccionar la hora del día, estación y modelo meteorológico. Puedes volar de acuerdo a las condiciones en tiempo real, o seleccionar condiciones meteorológicas específicas, como una región de alta presión o una tormenta.

La página *Configuración* te permite seleccionar de entre varias opciones de simulación, co-

4.1. cmdline-opts]LAUNCHER@\FlightGearDashDash LAUNCHER\textitARRANCAR EL SIMULADOR29

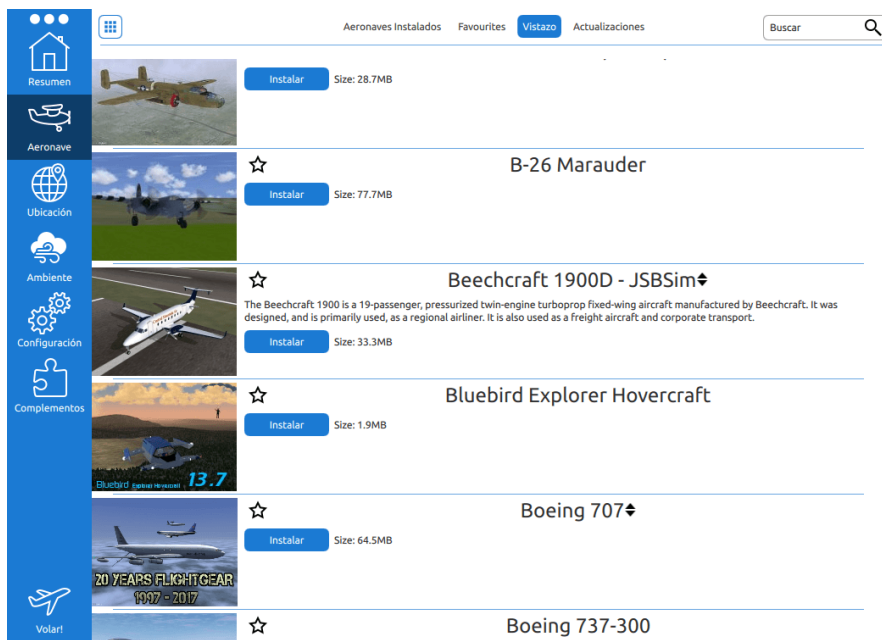


Figura 4.3: Selección de aeronave: *Selecciona entre una variedad de aeronaves y descárgalas de forma automática.*

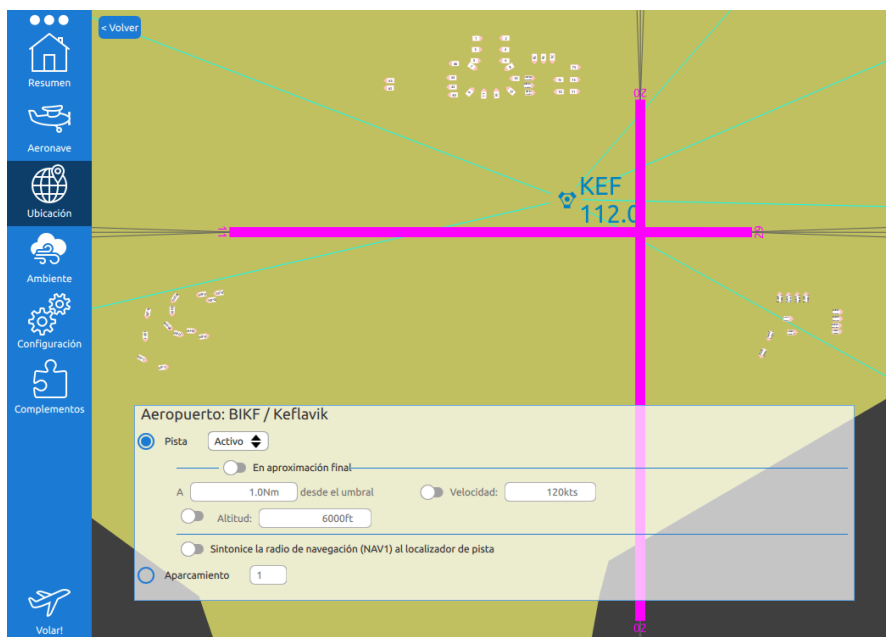


Figura 4.4: Posición inicial: *Selecciona una posición inicial en tierra o en el aire.*

mo por ejemplo el modo Multijugador, descarga automática de escenario y opciones gráficas. Se puede acceder a las opciones avanzadas pulsando **Mostrar Más** en la derecha.

Por último, la página *Complementos* te permite seleccionar hangares de aeronaves adicionales.

Una vez satisfecho con la configuración, pulsa el botón **Volar!** para arrancar el simulador.

4.2. Arrancar desde la línea de comandos

Como alternativa, puedes arrancar *FlightGear* desde la línea de comandos. Para ello, primero es necesario configurar las variables de entorno `$FG_ROOT` y `$FG_SCENERY` de forma manual.

Puedes configurar dichas variables de varias formas, dependiendo de tu plataforma y requisitos.

4.2.1. FG_ROOT

FlightGear utiliza esta variable para encontrar archivos de datos, como por ejemplo aeronaves, VOR y NDBs, o frecuencias de radio de los aeropuertos. Esta variable debería apuntar a la carpeta `data` de tu instalación de *FlightGear*, p. ej.

```
/usr/local/share/FlightGear/data o
C:\Archivos de programa\FlightGear\data.
```

4.2.2. FG_SCENERY

FlightGear utiliza esta variable para buscar ficheros de escenario. Consiste en una lista de directorios, los cuales son analizados en orden. Puedes separar los directorios mediante “:” en Unix y “;” en Windows. p. ej.

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

o

```
C:\Program Files\FlightGear\data\Scenery;C:\Program
↪ Files\FlightGear\data\WorldScenery
```

4.2.3. Arrancar el simulador en Windows

Abre una terminal de comandos, muévete al directorio que contiene el ejecutable (algo similar a `C:\Archivos de programa\FlightGear\bin\Win32`), y configura las variables de entorno de la siguiente forma:

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

y ejecuta *FlightGear* (dentro de la misma terminal, ya que las variables de entorno son válidas únicamente dentro de la misma terminal) así:

```
fgfs --option1 --option2...
```

El Capítulo 4.3 describe las diferentes opciones de línea de comandos. Siempre puedes crear un fichero *batch* con un editor de texto (p. ej. notepad) con tus opciones favoritas. Para obtener el máximo rendimiento, se recomienda minimizar la ventana de salida de texto durante la ejecución de *FlightGear*.

4.2.4. Arrancar el simulador en Unix/Linux

Antes de ejecutar *FlightGear*, es necesario configurar un par de variables de entorno:

- Añade `/usr/local/share/FlightGear/lib` a tu `LD_LIBRARY_PATH`
- Apunta `$FG_ROOT` a la carpeta de datos de tu instalación de *FlightGear*.
p. ej. `/usr/local/share/FlightGear/data`.
- `$FG_SCENERY` debe contener una lista de carpetas de escenario, separadas por `:"`. Funciona de una forma parecida a `PATH` pero para escenario.
p. ej. `$FG_ROOT/Scenery:$FG_ROOT/WorldScenery`.

Para configurar estas variables en la terminal Bourne (y compatibles):

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

o en la terminal C (y 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
```

Una vez configuradas las variables, arranca *FlightGear*:

```
fgfs --option1 --option2...
```

El Capítulo 4.3 explica las diferentes opciones de línea de comandos.

4.2.5. Arrancar el simulador en Mac OS X

Puedes arrancar el simulador mediante línea de comandos en Mac OS X. Para ello, abre Terminal.app (en /Applications/Utilities) y ejecuta lo siguiente:

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

Echa un vistazo al Capítulo 4.3 para más detalles acerca de las opciones de línea de comandos. A diferencia de otros sistemas operativos, no hace falta especificar variables de entorno como FG_ROOT and FG_SCENERY siempre y cuando uses un paquete precompilado.

4.3. Opciones de línea de comandos

A continuación encontrarás una lista con breves explicaciones de las múltiples opciones de línea de comandos disponibles en *FlightGear*.

Si utilizas algunas opciones de forma habitual, puedes crear un fichero de preferencias con una lista de opciones que se activarán automáticamente. Para ello, puedes usar cualquier editor de texto (notepad, emacs, vi o similares).

- En sistemas Unix (incluido Mac OS X), añade las opciones a un fichero en tu carpeta de usuario y llámalo `.fgfsrc`.
- En Windows, pon las opciones en un fichero llamado `system.fgfsrc` en el directorio `$FG_ROOT`.

4.3.1. Opciones Generales

- `--launcher`
Arranca el lanzador (descrito más arriba)
- `--help`
Muestra las opciones de línea de comandos más importantes.
- `--help --verbose`
Muestra las opciones de línea de comandos al completo.
- `--version`
Muestra la versión actual de *FlightGear*.
- `--fg-root=ruta`
Configura *FlightGear* para que busque los ficheros de datos principales, en caso de que el simulador no esté compilado con las variables por defecto.

- `--fg-scenery=ruta`

Permite especificar la ruta de acceso a los ficheros de escenario, en caso de que el escenario no esté en la ruta por defecto en `$FG_ROOT/Scenery`; esto es especialmente útil en caso de tener los escenarios en CD-ROM.
- `--fg-aircraft=ruta`

Permite especificar la ruta a los ficheros de aeronaves.
Por defecto, es `$FG_ROOT/Aircraft`.
- `--language=código`

Selecciona el lenguaje para esta sesión, p. ej. es, pl, nl, it, fr, en, de.
- `--restore-defaults`

Resetea la configuración de usuario a su valor por defecto.
- `--enable-save-on-exit, --disable-save-on-exit`

Activa o desactiva el guardado de la configuración de usuario entre sesiones.
- `--enable-freeze, --disable-freeze`

Controla el inicio en pausa. Por defecto, el simulador arranca no pausado.
- `--enable-auto-coordination, --disable-auto-coordination`

Activa o desactiva la coordinación automática entre alerón y timón de cola. Se recomienda activar la coordinación automática si no dispones de pedales o joystick que te permita controlar el timón. Por defecto, está desactivado.
- `--browser-app=ruta`

Especifica la ruta a tu navegador web. P. ej.: `--browser-app="C:\Program Files\Internet Explorer\iexplore.exe"`
(Fíjate en los `"..."` para poder usar espacios en la ruta!).
- `--config=ruta`

Carga propiedades adicionales desde el fichero especificado. P. ej.:
`--config=./Aircraft/X15-set.xml`
- `--units-feet`

Medir distancia en pies.
- `--units-meters`

Medir distancia en metros.

4.3.2. Características

- `--enable-ai-models, --disable-ai-models`
Activa o desactiva otras aeronaves/modelos IA en el simulador
- `--ai-scenario=escenario`
Selecciona un escenario IA específico (p. ej. `--ai-scenario=vinson-demo`). Puede usarse varias veces.

4.3.3. Sonido

- `--enable-sound, --disable-sound`
Activar o silenciar el sonido.
- `--show-sound-devices`
Mostrar dispositivos de sonido disponibles.
- `--sound-device=dispositivo`
Especificar el dispositivo de sonido a utilizar.

4.3.4. Aeronave

- `--aircraft=aeronave`
Seleccionar una aeronave específica (p. ej. `--aircraft=c172p`). Para ver las aeronaves disponibles, abre el directorio `$FG_ROOT/Aircraft`, y busca archivos con la extensión `"-set.xml"`. El nombre de la aeronave es el mismo que el del fichero, sin la terminación `"-set.xml"`. También puedes usar la opción `--show-aircraft` descrita más abajo para ver una lista de aeronaves disponibles. Si quieres descargar más aeronaves, lee la Sección 3.2.
- `--show-aircraft`
Ver una lista con los tipos de aeronave disponibles.
- `--min-status=calidad`
Mostrar únicamente aeronaves con una calidad por encima de un mínimo: `alpha`, `beta`, `early-production`, `production`. Usar en combinación con `--show-aircraft`.
- `--aircraft-dir=ruta`
Seleccionar carpeta de aeronaves, relativa al ejecutable.
El valor por defecto es `$FG_ROOT/Aircraft`.
- `--vehicle=aeronave`
Sinónimo de `--aircraft`.

- `--livery=nombre`
Seleccionar la librea de la aeronave.

4.3.5. Modelos de vuelo

- `--fdm=nombre`
Selecciona el modelo de dinámica de vuelo (FDM) de entre los siguientes valores: `jsb`, `larcsim`, `yasim`, `magic`, `balloon`, `external`, `pipe`, `ada`, `null`. Normalmente no es necesario especificar esta opción, ya que la opción `--aircraft` configura el FDM correctamente.
- `--aero=aeronave`
Especifica el modelo aeronáutico de la aeronave a cargar. Esta opción no suele ser necesaria, ya que la opción `--aircraft` suele configurar el modelo correctamente.
- `--model-hz=n`
Especifica la frecuencia a la que ejecutar el Modelo de Dinámica de Vuelo (iteraciones por segundo).
- `--speed=n`
Ejecuta el Modelo de Dinámica de Vuelo a una velocidad n veces mayor que tiempo real.
- `--trim`, `--notrim`
Ajustar “trim” (o no) al inicializar JSBSim. EL valor por defecto es ajustar.
- `--on-ground`, `--in-air`
Comenzar en tierra (por defecto) o en el aire. La opción `--in-air` requiere una inicialización de altitud inicial mediante `--altitude`, y probablemente quieras inicializar la velocidad inicial con `--vc`. Algunas aeronaves (por ejemplo el X15) necesitan comenzar directamente en el aire.
- `--enable-fuel-freeze`, `--disable-fuel-freeze`
Desactiva el consumo de combustible (*enable fuel freeze*), o lo activa (por defecto).

4.3.6. Posición inicial y orientación

- `--airport=ABCD`
Comenzar en un aeropuerto específico. Puedes especificar el aeropuerto mediante su código ICAO, p. ej. `--airport=KJFK` para JFK en Nueva York. Para aeropuertos en EEUU sin códigos ICAO, intenta añadir una 'K' al inicio del código local de 3 caracteres.
- `--parking-id=ABCD`
Comenzar en una posición de aparcamiento específica dentro del aeropuerto.

- `--runway=NNN`
 Comenzar al inicio de una pista de despegue específica (p. ej. 28L). Si no se especifica una posición de aparcamiento o una pista, el simulador comenzará en la pista más adecuada de acuerdo a la dirección del viento.
- `--vor=ABCD, --ndb=ABCD, --fix=ABCD`
 Seleccionar la posición de inicio respecto a un VOR, NDB o FIX. Esta opción es idónea para practicar aterrizajes.
- `--carrier=nombre`
 Comenzar en un portaaviones. Encontrarás más información acerca de operaciones sobre portaaviones en la Sección 6.2.
- `--parkpos=nombre`
 Comenzar en una posición de aparcamiento específica sobre el portaaviones. Debe usarse junto a la opción `--carrier`. Por defecto, se selecciona la posición de lanzamiento en la catapulta.
- `--offset-distance=millas náuticas, --offset-azimuth=grados`
 Comenzar a una distancia y rumbo específicos respecto a una posición especificada mediante `--airport, --vor, --ndb, --fix, --carrier`.
- `--lon=grados, --lat=grados`
 Comenzar a una longitud y latitud específicas, en grados decimales (sur y oeste adoptan valores negativos).
- `--altitude=pies`
 Comenzar a una altitud específica. Implica la opción `--in-air`, no es necesario añadirla por separado. La altitud es en pies, excepto si se añade la opción `--units-meters`, en cuyo caso la altitud es en metros. Posiblemente quieras configurar una velocidad inicial con `--vc` para evitar una entrada en pérdida inmediata.
- `--heading=grados, --roll=grados, --pitch=grados`
 Selecciona la orientación inicial de la aeronave. Todos los valores por defecto son 0 - rumbo norte, en ausencia de giro y manteniendo altitud.
- `--uBody=X, --vBody=Y, --wBody=Z`
 Configura la velocidad inicial en los ejes X, Y, Z. La velocidad es en pies por segundo, a no ser que se utilice junto a `--units-meters`, en cuyo caso la velocidad es en metros por segundo.
- `--vNorth=N, --vEast=E, --vDown=D`
 Configura la velocidad inicial en los ejes Norte-Sur, Este-Oeste and vertical. La velocidad es en pies por segundo, a no ser que se use en conjunto con `--units-meters`, en cuyo caso la velocidad es en metros por segundo.

- `--vc=nudos`, `--mach=num`

Selecciona la velocidad inicial en nudos o en Mach. Es importante usar esta opción si se usa `--altitude`, a no ser que quieras entrar inmediatamente en pérdida!

- `--glideslope=grados`, `--roc=ppm`

Selecciona el ángulo de la senda de descenso en grados o en pies por minuto. Puede ser positivo o negativo.

4.3.7. Opciones de entorno

- `--ceiling=pies_ASL[:grosor_pies]`

Configura el techo nuboso a la altitud seleccionada, con la opción de configurar el grosor (por defecto 2000 pies).

- `--enable-real-weather-fetch`, `--disable-real-weather-fetch`

Controla el uso de información climatológica en tiempo real.

- `--metar=cadena METAR`

Usar una cadena METAR específica, p. ej. `--metar="XXXX 012345Z 00000KT 99SM CLR 19/M01 A2992"`.

Es posible especificar METAR en los formatos más comunes (EE.UU., Europeo)

No se puede usar junto con `--enable-real-weather-fetch`.

- `--random-wind`

Configura la dirección y velocidad del viento de forma aleatoria.

- `--turbulence=n`

Configura la turbulencia en un rango desde calma (0.0) a severa (1.0).

- `--wind=direccion@velocidad`

Especifica el viento en la superficie. La dirección es en grados, y la velocidad es en nudos. Los valores se pueden especificar como un rango mediante el uso de punto y coma; p. ej.

`--wind=180:220@10:15`.

- `--season=param`

Selecciona la estación simulada. Su valor puede ser `summer` (por defecto) o `winter`.

- `--visibility=metros`, `--visibility-miles=millas`

Configura la visibilidad en metros o millas.

4.3.8. Opciones de renderizado

- `--aspect-ratio-multiplier=N`
Selecciona un multiplicador para la relación de aspecto de la ventana.
- `--bpp=profundidad`
Especifica los bits por píxel.
- `--enable-clouds, --disable-clouds`
Activa (por defecto) o desactiva las capas de nubes.
- `--enable-clouds3d, --disable-clouds3d`
Activa (por defecto) o desactiva las nubes 3D. Muy vistoso, pero tu tarjeta debe soportar shaders GLSL, lo cual puede no ser cierto si tu tarjeta es antigua o de gama baja.
- `--enable-distance-attenuation, --disable-distance-attenuation`
Activa o desactiva la atenuación realista de las luces de pista y aproximación.
- `--enable-enhanced-lighting, --disable-enhanced-lighting`
Activa o desactiva las luces realistas de pista y aproximación.
- `--enable-fullscreen, --disable-fullscreen`
Activa o desactiva (por defecto) el modo de pantalla completa.
- `--enable-horizon-effect, --disable-horizon-effect`
Activa (por defecto) o desactiva la ilusión de incremento de tamaño en objetos celestes cerca del horizonte.
- `--enable-mouse-pointer, --disable-mouse-pointer`
Activa o desactiva (por defecto) el puntero de ratón extra. Útil en tarjetas gráficas antiguas del tipo Voodoo.
- `--enable-panel, --disable-panel`
Activa (por defecto) el panel de instrumentos.
- `--enable-random-buildings, --disable-random-building`
Activa o desactiva (por defecto) los edificios aleatorios. Ten en cuenta que los edificios aleatorios consumen una gran cantidad de memoria.
- `--enable-random-objects, --disable-random-objects`
Activa (por defecto) o desactiva los objetos aleatorios de escenario.

- `--enable-random-vegetation`, `--disable-random-vegetation`

Activa (por defecto) o desactiva la vegetación aleatoria, como por ejemplo árboles. Requiere de una tarjeta gráfica con soporte para shaders GLSL, lo cual puede no ser aplicable a tarjetas antiguas o de gama baja.
- `--enable-rembrandt`, `--disable-rembrandt`

Activa o desactiva (por defecto) una opción experimental que incluye iluminación mejorada y sombras en tiempo real.
- `--enable-specular-highlight`, `--disable-specular-highlight`

Activa (por defecto) o desactiva los reflejos especulares.
- `--enable-splash-screen`, `--disable-splash-screen`

Activa o desactiva (por defecto) el logo 3DFX rotatorio que aparece al inicializar la tarjeta aceleradora (sólo con 3DFX).
- `--enable-wireframe`, `--disable-wireframe`

Activa o desactiva (por defecto) la muestra de *wireframes*. Prueba esta opción si lo quieres es ver cómo se ve el mundo de *FlightGear* por dentro!
- `--fog-disable`, `--fog-fastest`, `--fog-nicest`

Configura el nivel de niebla. Para simplificar las necesidades de renderizado, el escenario en la lejanía se muestra brumoso por defecto. Si desactivas la niebla podrás ver más lejos, pero tu *framerate* caerá. El uso de `--fog-fastest` muestra una niebla menos realista, aumentando el *framerate*. El valor por defecto es `--fog-nicest`.
- `--fov=grados`

Configura el campo de visión en grados. El valor por defecto es 55.0.
- `--materials-file=fichero`

Especifica el fichero de materiales utilizado para renderizar el escenario. Por defecto es `Materials/regions/materials.xml`.
- `--geometry=<ancho>x<alto>`

Define la resolución de la ventana o pantalla. P. ej. `--geometry=1024x768..`
- `--shading-smooth`, `--shading-flat`

Usar el suavizado de sombras (por defecto) o el suavizado plano, que es más rápido pero de peor calidad.
- `--texture-filtering=N`

Configurar el filtrado anisotrópico de texturas. Los valores permitidos son 1 (por defecto), 2, 4, 8, ó 16.

- `--view-offset=xxx`

Permite seleccionar la dirección por defecto de la vista delantera como un valor respecto del punto de referencia representado por “vista puramente hacia adelante”. Los valores posibles son `LEFT`, `RIGHT`, `CENTER` (izquierda, derecha, centro), o un valor en grados. Esto es útil en configuraciones con pantallas múltiples.

4.3.9. Opciones de HUD

- `--enable-anti-alias-hud, --disable-anti-alias-hud`

Controla el anti-aliasing del HUD (**H**ead-**U**p **D**isplay).

- `--enable-hud, --disable-hud`

Muestra o no el HUD. Por defecto, no se muestra el HUD.

- `--enable-hud-3d, --disable-hud-3d`

Muestra o no el HUD 3D. Por defecto, no se muestra.

- `--hud-culled, --hud-tris`

Muestra el porcentaje de triángulos escondidos, o el número de triángulos renderizados en el HUD. De interés para desarrolladores de gráficos.

4.3.10. Opciones de Sistemas de Aeronave

- `--adf=[radial:]frecuencia`

Configura la frecuencia y radial del ADF.

- `--com1=frecuencia, --com2=frecuencia`

Configura la frecuencia de radio COM1/COM2.

- `--dme=nav1|nav2|frecuencia`

Configura el DME a NAV1, NAV2 o una frecuencia y radial específica.

- `--failure=sistema`

Avería un sistema específico.

Los valores aceptados son `pitot`, `static`, `vacuum`, `electrical`. Puedes especificar varios sistemas a la vez.

- `--nav1=[radial:]frecuencia, --nav2=[radial:]frecuencia`

Configura la frecuencia y radial de la radio NAV1/NAV2.

4.3.11. Opciones de horario

- `--enable-clock-freeze`, `--disable-clock-freeze`

Activa el avance del tiempo, o lo congela.

- `--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`

Especifica la fecha y hora exacta de comienzo. La primera opción utiliza el huso horario de Greenwich (GMT), la segunda acepta la hora local en tu posición actual, y la tercera usa la hora de tu ordenador.

Esta opción es incompatible con `--time-match-local` y `--time-match-real`.

- `--time-match-local`, `--time-match-real`

`--time-match-real` es la opción por defecto: la hora del simulador es la misma que el reloj del sistema. Es una buena opción cuando tu vuelo virtual es en el mismo huso horario que tu sistema, ya que los relojes están sincronizados. En cambio, cuando vuelas en otra parte del mundo, puede haber una diferencia horaria considerable entre dicha posición y tu máquina.

La opción `--time-match-local` se encarga de arreglar esto calculando la diferencia horaria entre la hora de tu máquina y tu posición en el simulador, y sincroniza el reloj local.

Esta opción es incompatible con:

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

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

Especifica una diferencia de tiempo relativa a una de las opciones de tiempo mencionada anteriormente.

- `--timeofday=param`

Configura la hora. Los posibles valores son: `real`, `dawn`, `morning`, `noon`, `afternoon`, `dusk`, `evening`, `midnight`.

4.3.12. Opciones de red

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

Configura las opciones de multijugador y la señal de llamada. Ver Sección 6.1.

- `--httpd=puerto`, `--telnet=puerto`

Activa el servidor HTTP o el servidor Telnet en el puerto específico para dar acceso al árbol de propiedades.

- `--jpg-httpd=puerto`
Activa el servidor HTTP de captura de pantalla en el servidor especificado.
- `--proxy=[usuario:contraseña@]host:puerto`
Especifica un *proxy*.

4.3.13. Opciones de rutas/waypoints

- `--wp=ID[@alt]`
Permite especificar un *waypoint* en el piloto automático. Puedes especificar múltiples *waypoints* (es decir, una ruta) mediante múltiples usos de esta opción.
- `--flight-plan=fichero`
Esta opción es más amigable si tienes varios *waypoints*, ya que puedes leerlos desde fichero.

4.3.14. Opciones de entrada/salida

Estas opciones están destinadas al usuario avanzado.

Puedes encontrar descripciones más detalladas sobre los diferentes parámetros de entrada/salida en el archivo README.IO dentro del directorio *Docs* de tu instalación de *FlightGear*.

- `--atlas=params`
Abrir conexión usando el protocolo Atlas (usado por Atlas y *TerraSync*).
- `--atcsim=params`
Abrir conexión usando el protocolo ATC Sim (atc610x).
- `--AV400=params`
Abrir conexión para controlar un GPS Garmin 196/296.
- `--AV400Sim=params`
Abrir conexión para controlar un GPS Garmin 400.
- `--generic=params`
Abrir conexión usando el protocolo genérico (definido en XML).
- `--garmin=params`
Abrir conexión usando el protocolo Garmin GPS.
- `--joyclient=params`
Abrir conexión a un joystick Agwagon.

- `--jsclient=params`
Abrir conexión a un joystick remoto.
- `--native-ctrls=params`
Abrir conexión usando el protocolo de controladores nativos FG.
- `--native-fdm=params`
Abrir conexión usando el protocolo nativo para FDMs de FG.
- `--native-gui=params`
Abrir conexión usando el protocolo nativo para GUI de FG.
- `--native=params`
Abrir conexión usando el protocolo nativo de FG.
- `--nmea=params`
Abrir conexión usando el protocolo NMEA.
- `--opengc=params`
Abrir conexión usando el protocolo OpenGC.
- `--props=params`
Abrir conexión usando el gestor de propiedades interactivo.
- `--pve=params`
Abrir conexión usando el protocolo PVE.
- `--ray=params`
Abrir conexión usando el protocolo de la silla de movimiento RayWoodworth.
- `--rul=params`
Abrir conexión usando el protocolo RUL.

4.3.15. Opciones de depuración

Estas opciones están destinadas al usuario avanzado.

- `--enable-fpe`
Habilitar interrupción causada por una excepción de punto flotante.
- `--fgviewer`
En vez de cargar el simulador al completo, cargar un visor OSG mínimo. Útil para probar modelos de aeronave.

- `--log-level=nivel`

Configurar el nivel de reporte. Los posibles valores son `bulk`, `debug`, `info`, `warn`, `alert`.

- `--prop:[tipo:]name=valor`

Configurar la propiedad `nombre` con un `valor`.

Ejemplo: `--prop:/engines/engine[0]/running=true` arranca el simulador con los motores encendidos.

El siguiente ejemplo llena los tanques para un vuelo corto:

```
--aircraft=c172p
--prop:/consumables/fuels/tank[0]/level-gal=10
--prop:/consumables/fuels/tank[1]/level-gal=10
```

Opcionalmente puedes especifica el tipo de datos de la propiedad (`double`, `string`, `boolean`).

- `--trace-read=params`

Monitoriza las lecturas de una propiedad. Esta opción se puede usar varias veces.

- `--trace-write=params`

Monitoriza las escrituras de una propiedad. Esta opción se puede usar varias veces.

4.4. Soporte para joysticks

Te imaginas a un piloto controlando su Cessna con un teclado? Difícil! Para experimentar la sensación de volar de verdad, necesitas un joystick o unos cuernos, y pedales de timón.

FlightGear incorpora soporte para joysticks y los detecta automáticamente, así como cuernos y pedales. Sólo tienes que enchufar tu joystick y arrancar el simulador.

Puedes ver la configuración de tu joystick seleccionando el menú *Archivo* → *Configuración de joystick*. Este diálogo muestra el nombre de tu joystick y la función asignada a cada uno de tus botones y ejes. Puedes apretar un botón o mover el joystick para ver la función de cada uno de los botones.

Si tienes un joystick común, es muy posible que alguien haya añadido un fichero de configuración a *FlightGear* para éste. Si deseas cambiar la configuración de un botón o eje en concreto, puedes editarlo en el diálogo de Configuración de joystick.

Si tu joystick no es tan típico, *FlightGear* utilizará una configuración básica. Para cambiar dicha configuración, simplemente usa el diálogo de Configuración de joystick y selecciona la función deseada para cada botón. Los cambios tienen efecto inmediato y se guardan para tus siguientes vuelos.

Capítulo 5

En vuelo: uso de los instrumentos, teclado y menús

A continuación encontrarás una descripción de los principales sistemas que te permiten controlar el programa y pilotar tu aeronave. Asumimos que el lector tiene unas nociones básicas de vuelo, y quizás experiencia con otros simuladores. Si eres un piloto novato, los tutoriales en la Sección 7, [Tutorials](#) son una forma más fácil de iniciarse en el vuelo con *FlightGear*.

Encontrarás una tarjeta imprimible con los principales atajos de teclado en el directorio de instalación:

```
FlightGear/Docs/FGShortRef.pdf
```

Dentro del simulador, el menú Ayuda contiene una explicación de los atajos de teclado más importantes.

5.1. Arrancando el motor

La mayoría de aeronaves en *FlightGear* comienzan con el motor apagado por defecto. Las instrucciones que encontrarás a continuación son genéricas. Echa un vistazo a la ayuda o los tutoriales de la aeronave si necesitas instrucciones más específicas.

Una vez arrancado el motor, comprueba si el freno de parking está activo. Si lo está, pulsa **B** para quitarlo.

5.1.1. Aeronaves de pistón

En aeronaves de motor de pistones, los magnetos se controlan con las teclas { and }. En la mayoría de aeronaves, el arranque se activa mediante la tecla **s**. En aviones multimotor, puedes seleccionar qué motor quieres controlar. La tecla '~' selecciona todos los motores a la vez. La mayoría de magnetos dispone de 4 opciones: OFF, LEFT, RIGHT, BOTH en este orden. Para arrancar el avión, pulsa la tecla } tres veces (posición BOTH) seguido de la tecla **s**.

Ten en cuenta que algunos procedimientos de arranque, como por ejemplo los cazas de la segunda guerra mundial, suelen ser más complejos. Utiliza el menú de ayuda de aeronave como

referencia.

5.1.2. Aeronaves turbopropulsadas

Para encender un turbopropulsor, simplemente mueve la palanca de combustible desde la posición de *Off* hasta la posición de *Idle* con la tecla **m**.

5.1.3. Aeronaves a reacción

El arranque de un jet suele ser bastante complejo, y varía considerablemente de un modelo a otro.

1. Comprueba que la palanca de *cutoff* está en *on*
2. Activa el *starter*
3. Una vez el motor alcanza aproximadamente 5 % N1, cierra la palanca de *cutoff*
4. Desactiva el *starter* una vez el motor alcanza la velocidad de operación

5.2. Controles de teclado

Aunque la mejor manera de disfrutar de *FlightGear* es usando joystick y pedales, puedes volar usando únicamente el teclado o junto con un ratón, como se describe a continuación. Cualquiera que sea tu modo preferido de controlar la aeronave, algunas acciones sólo pueden ser realizadas por medio del teclado.

Los atajos de teclado son configurables. Puedes ver la configuración actual y cambiarla mediante el fichero `keyboard.xml` que encontrarás en el directorio raíz de *FlightGear*. Este fichero es simplemente un archivo de texto fácil de leer. Aunque quizás cambiar el fichero no es la mejor idea si acabas de empezar a usar el simulador, los usuarios avanzados lo encontrarán muy conveniente para ajustar el manejo del simulador a su gusto, por ejemplo para usar los mismos atajos que otros simuladores.

5.2.1. Controles de la aeronave

Para controlar la aeronave mediante el teclado, asegúrate de que `NumLock` está activado y la ventana de *FlightGear* está activa (en otras palabras, en foco o seleccionada).

Las siguientes teclas accionan las superficies de control de la aeronave y la potencia.

Tecla	Acción
9 / 3	Potencia
4 / 6	Alerón
8 / 2	Elevador
0 / Enter	Timón de cola
5	Centrar alerón/elevador/timón
7 / 1	Compensador del elevador

Tabla 1: *Superficies primarias de la aeronave*

Las siguientes teclas controlan los motores.

Tecla	Acción
!	Seleccionar motor 1
@	Seleccionar motor 2
#	Seleccionar motor 3
\$	Seleccionar motor 4
~	Seleccionar todos los motores
hline {	Reducir magneto en el motor seleccionado
}	Aumentar magneto en el motor seleccionado
s	Arrancar el motor seleccionado
M / m	Empobrecer o enriquecer mezcla en el motor seleccionado
N / n	Reducir o aumentar las RPM del motor seleccionado

Table 2: *Motores*

Las siguientes teclas controlan los sistemas secundarios.

Tecla	Acción
b	Aplicar todos los frenos
, / .	Aplicar freno izquierdo o derecho (útil para usar frenado diferencial)
l	(Des)activa el bloqueo de la rueda de cola
B	(Des)activa el freno de parking
g/G	Sube o baja el tren de aterrizaje
Space	<i>Push To Talk</i> (PTT) o “Pulsar para hablar”
- / _	Menú de chat multijugador
[/]	Retraer o extender los flaps
j / k	Retraer o extender los spoilers
Ctrl-B	(Des)activa los aerofrenos

Tabla 3: *Controles secundarios*

5.2.2. Controles del simulador

Para mover la vista, desactiva NumLock y utiliza el teclado numérico:

Tecla	Dirección de movimiento
Shift-8	Delante
Shift-7	Izquierda/delante
Shift-4	Izquierda
Shift-1	Izquierda/atrás
Shift-2	Atrás
Shift-3	Derecha/atrás
Shift-6	Derecha
Shift-9	Derecha/delante

Tabla 4: *Dirección de la vista*

Las teclas siguientes te permiten controlar otros detalles de vista:

Tecla	Acción
P	(Des)activa el panel de instrumentos
c	(Des)activa la cabina 3D/2D (si están disponibles)
S	Cambiar entre estilo de panel completo/simple
Ctrl-C	(Des)activa el resaltado de superficies de interacción
h	(Des)activa el HUD
H	Modifica el brillo del HUD
i / I	Cambia el modo del HUD
x / X	Acercar o alejar vista
v / V	Cambiar vista
Ctrl-V	Cambiar a vista de piloto
z / Z	Incrementar/decrementar visibilidad
F10	Mostrar/ocultar menú
Shift-F10	(Des)activa la pantalla completa

Tabla 5: *Opciones de vista*

Tecla	Acción
p	Pausa el simulador
a / A	Cambia la velocidad de simulación
t / T	Cambia la velocidad de reloj
Ctrl-R	Replay instantáneo
F3	Saca una foto
ESC	Sale del simulador

Tabla 6: *Controles generales.*

5.2.3. Controles de piloto automático

FlightGear soporta dos tipos de piloto automático: uno genérico que funciona con cualquier aeronave (incluidas las que no disponen de piloto automático en la realidad), y pilotos automáticos específicos que se pueden controlar desde la cabina.

El piloto automático genérico se controla mediante las siguientes teclas:

Tecla	Acción
Retroceso	(Des)activa el piloto automático
Ctrl-A	(Des)activa el modo de mantener altitud
Ctrl-G	(Des)activa el modo de seguimiento de senda de descenso (NAV 1)
Ctrl-H	(Des)activa el modo de mantener rumbo
Ctrl-N	(Des)activa el modo de seguimiento de NAV 1
Ctrl-S	(Des)activa el control automático de velocidad
Ctrl-T	(Des)activa el seguimiento de terreno AGL
Ctrl-U	Añade 1000 pies a la altitud actual (emergencia)
F6	(Des)activa el modo de ruta
F11	Abre el diálogo de altitud
8 / 2	Ajusta la altitud
4 / 6	Ajusta el rumbo
9 / 3	Ajusta la velocidad

Tabla 7: Controles de piloto automático.

Ctrl-t es un modo curioso: hace que tu aeronave se comporte como un misil de crucero, siguiendo el terreno a una distancia constante. **Ctrl-u** es útil como emergencia ante una colisión con el terreno inminente.

5.3. Controles de ratón

Además del uso de los menús y los elementos accionables en la cabina, el ratón tiene otros usos importantes en *FlightGear*.

El ratón dispone de tres modos: normal, control de vuelo y dirección de vista. Estos modos te permiten interactuar con elementos (es decir, hacer clic en cosas), volar la aeronave y mirar a tu alrededor, respectivamente. Puedes cambiar entre modos con la tecla de Tabulador.

5.3.1. Modo normal

En modo normal, puedes controlar los menús y los paneles de control. En este modo, el cursor del ratón es una flecha.

Para accionar un elemento, simplemente haz clic con el botón izquierdo o central.

Para accionar diales y controles lineales, como por ejemplo un dial de radio o la palanca de potencia, haz clic a la izquierda para decrementar el valor, o a la derecha para incrementarlo. Haz clic con el botón izquierdo para un ajuste fino, o el botón central para un ajuste más grueso. Algunos controles, como por ejemplo las radios, también soportan el uso de la rueda del ratón, mientras que otros, como la palanca de potencia, pueden ser arrastrados con el botón izquierdo.

Si no encuentras dónde hacer clic, presiona **Ctrl-c** para resaltar los elementos accionables.

5.3.2. Modo de control de vuelo

En este modo puedes controlar la aeronave moviendo el cursor. El cursor adopta la forma de una cruz.

Un movimiento hacia la derecha o izquierda acciona los alerones, girando la aeronave. El elevador se acciona moviendo el ratón arriba o abajo, cambiando el ángulo de inclinación de la aeronave.

Presionando el botón izquierdo del ratón, se cambia el comportamiento de forma que un movimiento hacia la derecha o izquierda mueve el timón de cola. Puedes accionar el mando de potencia presionando el botón central y moviendo el ratón hacia arriba o hacia abajo.

Por último, la rueda cumple la función de ajuste del compensador de elevador

El modo de control de vuelo es especialmente útil si no tienes un joystick, al permitir un control mucho más preciso que el teclado. Si vas a usar el ratón como control principal de forma habitual, te recomendamos activar la coordinación automática, de forma que el control de los alerones están unidos al timón de cola. Puedes activar esta opción con la opción de línea de comandos `--enable-auto-coordination`, o seleccionando la opción de or auto-coordinación en el lanzador.

5.3.3. Modo de dirección de vista

Este modo te permite mirar a tu alrededor con el ratón. El cursor toma la forma de una flecha de doble dirección.

Este modo es muy útil para mirar alrededor de la cabina, o por una ventana. Puedes usar la rueda del ratón para acercar o alejar la vista. Para volver a la vista por defecto, haz clic en el botón izquierdo.

Para mover el punto de vista hacia la derecha/izquierda o arriba/abajo, mantén presionado el botón central y mueve el ratón en la dirección deseada. Para mover el punto de vista hacia adelante o hacia atrás, presiona el botón central junto con **Ctrl**.

5.4. Entradas del menú

La barra de menú te da acceso a una multitud de opciones de simulador y aeronave. La mayoría de aeronaves dispone de menús específicos, como por ejemplo para cambiar el identificador de registro o arrancar los motores automáticamente. Éstos están situados al final de la barra de menú.

Para mostrar o esconder la barra de menú, presiona F10. El menú también aparecerá de forma automática si tocas el borde superior de la pantalla con el ratón.

La barra de menú contiene los siguientes sub-menús y opciones:

- **Archivo**

- **Reiniciar (Shift-Esc)** el vuelo en la posición inicial. Útil en caso de pérdida o accidente.

- **Grabaciones de vuelo** abre el diálogo de configuración, carga y salvado de grabaciones de vuelo. **Shift-F1** y **Shift-F2** abren los diálogos de carga y salvado, respectivamente.
 - **Capturar pantalla (F3)** captura una foto y la guarda como `fgfs-YYYYMMDDHHmmSS.png`.
 - **Directorio de capturas** te permite configurar el directorio en el que se guardan las fotos.
 - **Configurar sonido** muestra opciones de configuración de varios dispositivos y canales de sonido.
 - **Configurar dispositivos de entrada** configura el comportamiento del ratón.
 - **Configuración de joystick** te permite configurar el joystick, con sus ejes y botones.
 - **Descargar escenarios** muestra el diálogo de configuración de descarga de escenarios mediante *TerraSync*.
 - **Centro de aviones (Experimental)** te permite descargar e instalar aeronaves desde el simulador. Experimental.
 - **Salir (Esc)** cierra el programa.
- **Vista**
- **Pantalla completa (Shift-F10)** alterna entre modo de ventana y pantalla completa.
 - **Opciones de renderizado** abre el diálogo de configuración de propiedades gráficas. Aquí puedes activar opciones de calidad gráfica como sombras, nubes 3D y reflejos especulares a cambio de *framerate*. Para obtener un buen balance entre calidad gráfica y velocidad, puedes activar la opción “Show Frame Rate” dentro del menú *Opciones* de vistas; esto mostrará el *framerate* actual en frames por segundo en la esquina inferior derecha de la pantalla. 20 fps debería ser suficiente para volar confortablemente. Las opciones gráficas activadas afectan al *framerate*, como por ejemplo la visibilidad (configurable con **z / Z**), el número de objetos al alcance de la vista y su nivel de detalle (LOD).
 - **Opciones de vistas** permite configurar diferentes opciones de vista, incluido qué vistas están disponibles, o mostrar el panel 2D, el *framerate* o los mensajes de chat.
 - **Opciones de vista de cabina** configura la vista de cabina, el movimiento de la cabeza del piloto, o los efectos de vista debidos a altos Gs o Gs negativos.
 - **Ajustar nivel de detalle** configura el rango de detalle de diferentes objetos. Dicha configuración define el detalle de las texturas a varias distancias.
 - **Posición de punto de vista** muestra la posición del punto de vista actual y permite ajustarlo mediante unos diales. También puedes hacer pequeños ajustes con el ratón (ver más arriba).
 - **Cambiar propiedades del HUD** configura las propiedades del HUD (del inglés *Head Up Display*), como la transparencia o el *anti-aliasing*.

- **Activar senda de planeo** muestra un túnel virtual que te guiará hasta la pista de aterrizaje a través de un sendero de aproximación estándar. Esta opción es muy útil para entrenar aproximaciones.
- **Replay instantáneo (Ctrl-R)** controla la opción de replay instantáneo, una herramienta muy útil para revisar tus aterrizajes!
- **Renderizado de órbita terrestre** controla el renderizado alternativo basado en la colección de imágenes *NASA Visible Earth*.
- **Opciones vista estereoscópica** configura la vista estereoscópica compatible con el uso de gafas 3D rojo/verde u otros métodos.

■ Localización

- **Ubicar aeronave en vuelo** sitúa la aeronave en un punto determinado en el aire. Puedes seleccionar de entre varios puntos conocidos: aeropuertos, VOR, coordenadas, y cualquier posición relativa a los puntos anteriores según la distancia, dirección y altitud. También te permite definir tu velocidad y dirección inicial. Puedes usar estas opciones para practicar aproximaciones.
- **Seleccionar aeropuerto** sitúa tu aeronave en un aeropuerto. Puedes buscar entre todos los aeropuertos instalados, ver el METAR más actual, información y mapa de aeródromo, así como seleccionar una pista o posición de parking.
- **Dirección al azar** posiciona la aeronave en una dirección, velocidad y actitud aleatorias. Útil para practicar recuperaciones de situaciones de pérdida.
- **Posición de la torre** configura la torre de control utilizada en las distintas vistas de torre.

- **Autopiloto** está disponible únicamente en aeronaves en las que se ha configurado el piloto automático por defecto. Algunas aeronaves disponen de su propio piloto automático en cabina, en cuyo caso este menú aparece deshabilitado.

- **Configurar AP (F11)** configura el piloto automático mediante varios modos, desde mantener las alas niveladas, hasta seguir una aproximación ILS.
- **Configurar rutas** configura la ruta mediante una serie de *waypoints*. Estos pueden ser aeropuertos o bien *fixes*. El HUD dispone de la opción de mostrar la dirección, distancia y tiempo restante al *waypoint* actual.
- **Waypoint previo** selecciona el waypoint inmediatamente anterior al actual.
- **Waypoint siguiente** selecciona el waypoint inmediatamente posterior al actual.

■ Ambiente

- **Clima** te permite configurar el tiempo actual, seleccionar una configuración meteorológica o usar la meteorología basada en el METAR de la estación más cercana, habitualmente un aeropuerto. Puedes usar la meteorología básica, que utiliza las condiciones locales, o la meteorología avanzada, que simula sistemas meteorológicos más extensos a cambio de una menor precisión local.
- **Configurar ambiente** te permite configurar las texturas dependiendo de la estación (verano o invierno), así como las condiciones del terreno como nieve o polvo.
- **Configurar horario** te permite definir el horario actual, acelerar la simulación y cambiar la velocidad del paso del tiempo. También puedes ver la hora UTC y local.
- **Configurar incendios** permite activar la opción de generación de incendios por accidentes aéreos, los cuales pueden extinguirse con aviones cisterna.
- **Volcanes** configura la actividad volcánica en los alrededores.

■ Equipamiento

- **Mapa** muestra un mapa en movimiento con aeropuertos, radiobalizas, etc.
- **Mapa (Canvas)** muestra un mapa alternativo basado en Canvas
- **Mapa (abre en el navegador)** muestra otro mapa alternativo, esta vez en el navegador. Es necesario tener el servidor web interno activado.
- **Cronómetro** muestra un cronómetro simple. Útil durante la aproximación.
- **Combustible y carga** permite rellenar el depósito de combustible y añadir carga. Disponible sólo en algunas aeronaves.
- **Configurar radio (F12)** ajusta las frecuencias y radiales utilizadas en las radios y otro equipo de navegación.
- **Configurar GPS** configura los waypoints y la información de ruta del GPS.
- **Configurar instrumentos** permite introducir la presión del altímetro y la corrección del indicador de rumbo (HSI).
- **Fallas al azar** configura la aparición de fallos aleatorios de diversos sistemas e instrumentos mediante dos parámetros: el tiempo medio entre fallos (MTBF) o el número medio de ciclos entre fallos (MCBF).
- **Fallas de sistemas** configura la aparición de fallos aleatorios en sistemas tales como el sistema de vacío.
- **Fallas de instrumentos** configura la aparición de fallos en instrumentos específicos.

■ IA

- **Configurar tráfico y escenarios** configura los escenarios IA activos.
- **Servicios ATC en cercanía** muestra las frecuencias de radio de los aeropuertos cercanos.
- **Controles vuelo en formación** controla otras naves IA en formación.
- **Controles reabastecimiento en vuelo** permite crear un avión cisterna, si tu avión lo permite. La Sección 6.8, [Repostaje aéreo](#) muestra más detalles.
- **Configurar manga** permite controlar los fingers de algunos aeropuertos.
- **Objetos IA** te permite seleccionar objetos IA y abrir sus diálogos de configuración.

■ Multijugador

- **Configuración multijugador** permite seleccionar una señal de llamada y un servidor.
- **Configuración de FGCom** configura las comunicaciones de voz con otros usuarios.
- **Ventana de chat** te permite chatear con otros usuarios.
- **Menú de chat (-)** te permite enviar mensajes predefinidos a otras aeronaves o ATC.

- **Lista de pilotos** muestra una lista de usuarios dentro de rango y su distancia, rumbo y altitud.
 - **Selección de portaviones multijugador** muestra una lista de portaaviones disponibles.
 - **Ajustes de retraso** abre el diálogo de corrección de retardo.
 - **Conexión Swift** abre el diálogo para arrancar un servidor Swift.
- **Depurar** contiene varias opciones avanzadas enfocadas al desarrollador.
 - **Ayuda**
 - **Ayuda (abre en el navegador)** abre la ayuda en un navegador.
 - **Navegador de documentación** abre un lector de documentación dentro del simulador.
 - **Ayuda de aeronave (?)** muestra información específica de la aeronave.
 - **Listas de verificación del aeronave** muestra listas disponibles en la aeronave actual.
 - **Teclas comunes a toda aeronave** enumera las teclas básicas de control de la aeronave actual.
 - **Teclas básicas del simulador** enumera las teclas generales de control del simulador.
 - **Tutoriales** ofrece tutoriales prácticos. Sólo disponible en algunas aeronaves.
 - **Acerca de** muestra información sobre la versión actual del simulador.

5.5. El panel de instrumentos

Una aeronave puede disponer de paneles de instrumentos en 2 y 3 dimensiones. Los paneles en 3D ofrecen una visión mucho más real, pero a veces son difíciles de leer en monitores de poca resolución.

El avión por defecto, la Cessna 172P (c172p), dispone de ambos tipos de panel. Por defecto se muestra el panel 3D, pero puedes mostrar el panel 2D mediante el menú *Vista* → *Opciones de vistas* → *Show 2D Panel*, o simplemente pulsando la tecla **P**.

Puedes usar el ratón para accionar cualquier control mediante los botones izquierdo o central. En controles con un rango de posiciones, el botón central realiza un cambio mayor que el botón izquierdo. Normalmente, el botón derecho incrementa el valor, mientras que el izquierdo lo decrementa.

Algunos instrumentos, como por ejemplo los radios, también soportan el uso de la rueda. Otros controles, como por ejemplo el control de potencia, también pueden ser arrastrados.

5.6. La pantalla frontal o HUD)

FlightGear dispone de una pantalla frontal o HUD (del inglés **H**ead-**U**p **D**isplay). El HUD es típico de aeronaves militares y algunos jets de última generación. Aun así, *FlightGear* te permite usarlo en la mayoría de aviones. Para activarlo, pulsa **h**.



Figura 5.1: La cabina 3D de una Cessna 172P Skyhawk (1982).

En la Figura 5.2 puedes ver el HUD con los principales parámetros del avión. En el centro se muestra el indicador de actitud (en grados) con el indicador de alerón encima y el indicador de timón de cola abajo. A la izquierda se encuentra el indicador de altitud respecto al suelo junto con el indicador de compensación de actitud. El inferior es un indicador de giro.

A la izquierda también puedes encontrar la velocidad en nudos y la posición del mando de potencia. A la derecha puedes ver dos escalas de altitud, la primera respecto al suelo y la segunda respecto al nivel del mar, ambas en pies.

La parte superior del HUD muestra información adicional. A la izquierda encontrarás información de GPS, velocidad respecto al suelo y Gs. En el centro se encuentran la latitud y longitud actual a ambos lados, y la dirección actual. La flecha roja apunta a la torre de control activa.

Puedes cambiar el color del **HUD** mediante las teclas **H / h**. Las teclas **I / i** lo maximizan o minimizan.

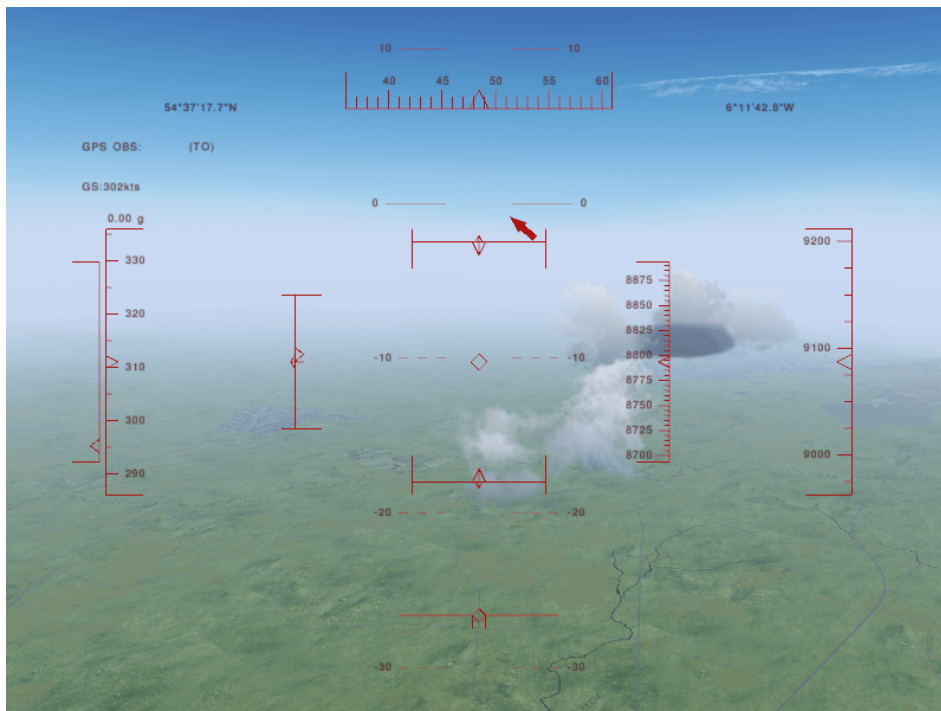


Figura 5.2: La pantalla frontal o HUD.

Capítulo 6

Funciones avanzadas

FlightGear dispone de una gran cantidad de opciones, algunas más fáciles de usar que otras. Esta sección explica cómo activar y usar algunas de las funciones avanzadas más útiles.

Al estar *FlightGear* en constante desarrollo, es posible que parte de la información ofrecida a continuación esté obsoleta. Para acceder a la documentación más actual (y nuevas funciones), puedes visitar la wiki de *FlightGear*, disponible en <https://wiki.flightgear.org>.

6.1. Multijugador

FlightGear te permite conectarte en modo multijugador y compartir el mundo virtual con otros jugadores. Puedes ver una lista de servidores, así como los pilotos conectados en cada momento, en el mapa multijugador disponible en:

<http://mpmap02.flightgear.org>

Haz clic en la pestaña ‘server’ para ver una lista de servidores disponibles.

6.1.1. Guía rápida

Puedes conectarte desde el menú multijugador haciendo clic en la opción *Configuración Multijugador*. Simplemente selecciona el servidor más cercano a ti, introduce una señal de llamada (como identificador para otros jugadores), y pulsa *Connect*.

Para ver una lista de pilotos conectados, pulsa *Lista de pilotos* en el menú *Multijugador*.

Todos los servidores multijugador están conectados entre sí, por lo que no es necesario conectarse al mismo servidor en el que están volando tus compañeros.

6.1.2. Otras formas de conectarse

Si te estás conectando a un servidor no oficial, o el método arriba indicado no funciona en tu caso, también puedes hacer lo siguiente:

Desde el lanzador

La pestaña *Configuración* del lanzador dispone de una sección *Multijugador*. Puedes seleccionar uno de los servidores disponibles en el menú desplegable *Servidor*, o configurar el acceso a un servidor manualmente mediante la entrada *Servidor personalizado*. Para esto último, debes introducir el nombre del servidor y el puerto en el siguiente formato: `nombre:puerto`.

Se recomienda configurar una señal de llamada como identificador a los demás usuarios. Dicha señal debe tener 7 caracteres como máximo.

Desde la línea de comandos

Los argumentos básicos de línea de comandos para multijugador son:

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

donde

1. `portnumber` es el puerto TCP del servidor, p. ej. 5000.
2. `server` es el nombre del servidor multijugador p. ej. `mpserver01.flightgear.org`.
3. `client` es el nombre de tu máquina, o la dirección IP de la interfaz de red que estás usando para conectarte al servidor, incluso si es una IP local, p. ej. 192.168.0.1
4. `callsign` es tu señal de llamada o identificador, máximo 7 caracteres, p. ej. N-FGFS.

Una vez la carga del simulador finalice, deberías verte en el mapa online. Si no te ves, puede que la consola muestre mensajes de error. Más abajo encontrarás una sección de solución de problemas.

6.1.3. Solución de problemas

Para conectarse correctamente a una red multijugador, es necesario saber la dirección IP de nuestra máquina y poder comunicarse con el servidor correctamente. A continuación se describe cómo hacer esto con algunas configuraciones frecuentes.

Si te conectas a internet mediante un módem USB

Primero, necesitas saber la IP de la red en la que vas a correr la sesión multijugador. Si te conectas mediante un módem ADSL enchufado directamente a tu máquina a través de USB, <https://www.whatismyip.com> debería mostrarte tu IP. Ten en cuenta que tu IP puede cambiar de forma ocasional. Si te caes de la sesión multijugador, comprueba tu IP de nuevo.

Si te conectas a internet mediante un router Ethernet

Lo más probable es que te conectes mediante un router a través de un conector RJ-45 o “Ethernet”, parecido al del cable de teléfono estándar, o mediante WiFi. Necesitas averiguar la IP de esta interfaz de red.

En Linux, sólo tienes que ejecutar el comando `ifconfig` como superusuario. Puede que veas varias interfaces que comienzan por `lo`; ignóralas. Deberías ver al menos una interfaz del tipo `eth0` o `wlan0`. En esta interfaz, busca el texto `inet`. La IP es el texto que le sigue, p. ej. `inet 192.168.0.150`.

En Windows XP, haz clic en el menú *Inicio, Ejecutar* y escribe `cmd`. En la terminal que aparece, escribe `ipconfig`. Al ejecutar este comando, deberías ver tu IP.

En Windows 98, en vez de `ipconfig`, ejecuta `winipcfg`.

Si aun así no funciona

Créeme, DEBES DAR TU IP LOCAL, no la IP pública de tu router!

Comprueba que tu firewall no está causando problemas: desactívalo temporalmente o añade una excepción para permitir conexiones al puerto 5000.

Si todavía no consigues hacerlo funcionar, pregunta en el canal IRC de *FlightGear* es muy posible que alguien te pueda ayudar.

6.2. Portaaviones

FlightGear soporta operaciones en portaaviones, e incluye varios escenarios en EEUU (como por ejemplo Nimitz y Vinson en San Francisco) y en el mar mediterráneo (como Eisenhower y Foch). Los portaaviones incluyen modelos de catapulta, cables de parada, ascensores, TACAN y FLOLS.

Varios aviones soportan operaciones en portaaviones, pero el más fácil para aprender es posiblemente el Seahawk.

6.2.1. Arrancando desde el portaaviones

Para comenzar p. ej. desde el portaaviones Nimitz, usa las siguientes opciones en tu línea de comandos:

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

Es importante escribir la “N” de “Nimitz” en mayúscula.

Si utilizas el lanzador de *FlightGear*, puedes seleccionar un portaaviones en la pestaña de *Ubicación*. Haz clic en el botón con forma de barco en la esquina superior derecha, justo a la derecha de la barra de búsqueda. Podrás ver varios portaaviones, en los cuales podrás también seleccionar la posición de arranque, bien en la cubierta o en el aire.

Si usas el lanzador de Windows u OS X, busca la sección donde puedes añadir opciones de línea de comandos, y añade las opciones arriba indicadas.

6.2.2. Uso de la catapulta

Nada más termine de cargarse el simulador, asegúrate de quitar el freno de aparcamiento. Después presiona **L** para enganchar la catapulta. Mantén presionado hasta que escuches el mensaje “Engaged”; notarás que el avión se alinea con el raíl y aparecerán unas correas unidas al tren de aterrizaje. Para que esto ocurra, es necesario estar en la posición correcta respecto a la catapulta. Como guía, para la posición de aparcamiento el morro del Seahawk debería más o menos igualarse a la burbuja de observación de la cubierta.

Para poner el portaviones en la mejor posición posible para el lanzamiento selecciona el diálogo *AI Objects* en el menú *AI*, selecciona el portaaviones actual (p. ej. Nimitz) y activa la opción *Turn to launch course*. El portaviones debería virar hacia el viento; naturalmente la superficie puede oscilar algo mientras gira. Espera a que acabe esta maniobra y se estabilice la superficie antes de pasar a la siguiente fase.

Una vez acoplado a la catapulta, pon los motores a potencia máxima, asegúrate de que los frenos están desactivados y de que la configuración de la aeronave es la de lanzamiento. Cuando estés listo presiona **C** para soltar la catapulta. Te propulsarás fuera de la cubierta, momento en que puedes subir el tren de aterrizaje y ascender lentamente, teniendo cuidado de no entrar en pérdida.

6.2.3. Aproximación al portaaviones – TACAN

Encontrar el portaaviones en mar abierto no es tarea fácil, especialmente en condiciones de poca visibilidad. Para simplificarla, el Nimitz equipa un sistema TACAN, el cual permite a aeronaves compatibles (incluido el Seahawk) obtener una distancia y rumbo al portaaviones. Para empezar, sintoniza tu radio (**F12** o menú *Equipamiento* → *Configurar radio*) al canal TACAN, que en el caso del Nimitz es 029Y. Si estás dentro del rango de cobertura, el DME te debería mostrar la distancia al portaaviones, y el ADF (justo al lado del DME en el Seahawk) indicará el rumbo en grados. Si tomas este rumbo, la distancia en el DME debería comenzar a disminuir, indicando que te estás acercando a la nave.

6.2.4. Aterrizando en el portaaviones

Ésta es, con diferencia, la parte más difícil de la operación, justo como en la vida real. El tutorial de Andy Ross sobre el A4 Skyhawk puede serte útil:

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

Una vez localizado el portaaviones con TACAN, alinéate con la cubierta de popa. La pista está en un pequeño ángulo respecto del rumbo del portaaviones, por lo que puede ser necesario corregir tu rumbo constantemente. Asegúrate de disponer tu aeronave en configuración de aterrizaje (el menú *Ayuda* → *Ayuda de aeronave* dispone de información específica para tu aeronave). Como mínimo, deberías desplegar tu tren de aterrizaje y gancho de parada.

A medida que te aproximas a la pista, verás el FLOLS (por el inglés *Fresnel Lens Optical landing System*), un grupo de luces de color. Éste indica tu posición en el sendero de aproximación. Verás una línea de luces verdes y, si te encuentras dentro del sendero, una luz anaranjada (conocida coloquialmente como “la albóndiga”) más o menos en línea con las luces verdes. Si te

encuentras a demasiada altitud respecto del sendero, la luz naranja se verá por encima de la línea verde y, si estás demasiado bajo, estará por debajo de la línea. Si estás excesivamente bajo, la “albóndiga” se vuelve roja. Si sigues un sendero de aproximación perfecto, deberías interceptar el tercer cable de freno.

Los aterrizajes en portaaviones se suelen describir como “aterrizajes forzosos”. No pierdas el tiempo intentando rotar tu avión o posarlo cuidadosamente en la pista como si fuera un aterrizaje estándar; el propósito principal es asegurarse de capturar uno de los cables.

Justo después de tocar cubierta, abre gas al máximo en caso de que no hayas capturado los cables y tengas que despegar. Los cables mantendrán el avión en su sitio, incluso a máxima potencia.

Opcionalmente, puedes utilizar uno de los ascensores para aparcar tu avión en la cubierta inferior. Abre el menú *IA* → *Objetos IA* → *<tu portaaviones>* y haz clic en la caja del ascensor deseado.

Por último, no desesperes si no te sale a la primera. Estamos hablando de una maniobra muy complicada. Si consigues controlarla y el Seahawk te parece muy fácil, encontrarás un reto mayor en el Seafire!

6.3. Atlas

Atlas es un mapa móvil para *FlightGear*. Muestra tu aeronave respecto al terreno junto con los aeropuertos, radiobalizas y frecuencias de radio.

Para más información, dirígete a su web oficial:
<http://atlas.sourceforge.net>

6.4. Varias pantallas

FlightGear soporta múltiples pantallas. Escribiendo un poco de XML, puedes configurar múltiples cámaras secundarias relativas a la vista principal. De esta forma, puedes usar varios monitores para mostrar una vista común de la simulación. Por ejemplo, puedes usar una pantalla para mostrar la vista frontal, y otras dos pantallas a los lados para mostrar una vista lateral.

Encontrarás más información sobre la configuración de múltiples pantallas en el fichero `docs/README.multiscreen` de tu instalación de *FlightGear*.

6.5. Varias máquinas

FlightGear te permite conectar varias instancias del simulador a través del subsistema de E/S de innumerables formas, por ejemplo para mostrar diferentes vistas y controles en máquinas separadas. Puedes usar esta técnica junto con el soporte de múltiples pantallas para crear entornos de simulación de gran complejidad, por ejemplo con pantallas separadas para paneles o incluso una máquina separada donde un instructor puede fallar instrumentos, cambiar la meteorología, etc.

El siguiente proyecto es un buen ejemplo de uso:

https://ftp.igh.cnrs.fr/pub/flightgear/www/Projects/747-JW/747_project.html

6.5.1. Conceptos básicos

Debido a la gran complejidad de las FDM y los gráficos, *FlightGear* requiere de una gran cantidad de potencia de procesamiento. Por ello, es necesario tener una única instancia de *FlightGear* por cada máquina.

Por tanto, cuenta con una máquina por cada vista de la simulación que quieres configurar, incluido el panel de instrumentos. Por razones obvias, las máquinas deben estar conectadas en red, preferiblemente a la misma subred para simplificar la configuración.

Elige tu máquina primaria: ésta ejecutará la FDM y se conectará a los controles tales como el joystick o los pedales. Al estar ejecutando la pesada FDM, conviene mostrar una única vista simple, como por ejemplo el panel de instrumentos, con el fin de optimizar el rendimiento.

Llamaremos a las otras máquinas “secundarias”, y las utilizaremos para mostrar otras vistas basadas en la información de FDM recibida desde la primaria.

6.5.2. Configuración básica

Es fácil crear una configuración básica con múltiples máquinas. La primaria debe comunicar datos de FDM y control a las secundarias, por lo que la arrancaremos con las siguientes opciones de línea de comandos:

```
--native-fdm=socket, out, 60, <IP-secondary>, 5505, udp
--native-ctrls=socket, out, 60, <IP-secondary>, 5506, udp
```

Reemplaza el parámetro `<IP-secondary>` con la IP de tu máquina secundaria. Si no sabes cómo averiguar la IP de una máquina, una búsqueda rápida en Google debería darte la respuesta. Si tienes varias máquinas secundarias, puedes repetir las opciones anteriores tantas veces como máquinas tengas, cambiando el parámetro `<IP-secondary>` en cada una de ellas y dejando el resto de parámetros igual.

Una vez has arrancado la máquina primaria, puedes arrancar las secundarias. Éstas deben recibir la información de la primaria, así como desactivar su FDM:

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

6.5.3. Configuración avanzada

Las opciones anteriores hacen que tanto la máquina primaria como la secundaria muestren la misma vista. Añade alguna de las opciones siguientes a la máquina primaria o secundaria para modificar el comportamiento:

```
--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
```

Si vas a usar la máquina primaria para mostrar únicamente el panel de instrumentos, asegúrate de que tu aeronave dispone de un panel a pantalla completa (la Cessna 172 ya tiene uno) o crea uno tú mismo, y usa las siguientes opciones:

```
--enable-panel
--disable-hud
# switch off rendering the world:
--prop:/sim/rendering/draw-mask/terrain=false
--prop:/sim/rendering/draw-mask/aircraft=false
--prop:/sim/rendering/draw-mask/models=false
--prop:/sim/rendering/draw-mask/clouds=false
```

6.6. Grabaciones y reproducción

Aparte del menú *Menú/Grabaciones de vuelo*, también puedes grabar tu vuelo mediante el sistema de E/S. Puedes encontrar más información sobre cómo grabar partes específicas de la FDM en el archivo `$FG_ROOT/Docs/README.protocol`.

Para grabar un vuelo, usa la siguiente opción de línea de comandos:

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

Con esta opción, el simulador grabará a 20 Hz (20 muestras por segundo) usando el protocolo `playback` y guardando los datos en el fichero `flight.out`.

Para reproducir la captura, arranca con las siguientes opciones:

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

El fichero de protocolo `playback.xml` no incluye información tal como el modelo de aeronave u hora. Intenta arrancar con las mismas opciones de línea de comandos que usaste al grabar.

6.7. Soporte de Texto a Voz con Festival

FlightGear soporta Texto a Voz (TTS por sus siglas en inglés *Text To Speech*) para crear mensajes de control de tráfico y mensajes para tutoriales mediante el motor Festival (<http://www.cstr.ed.ac.uk/projects/festival/>). Dicho motor está disponible en varias distribuciones Linux. También se puede instalar en Windows sobre Cygwin. Se desconoce el estado del soporte en plataformas distintas de las mencionadas.

6.7.1. Instalación de Festival

1. Instala Festival. Algunas distribuciones Linux (como p. ej Ubuntu 20.04) te permiten instalarlo desde su gestor de paquetes. Si no, dirígete a su web oficial <http://www.cstr.ed.ac.uk/projects/festival/>
2. Comprueba que funciona correctamente. Festival se maneja desde una interfaz de consola, como se muestra aquí. Observa que los comandos se escriben entre paréntesis!

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

3. Comprueba que MBROLA está instalado. De nuevo, quizás esté disponible en tu gestor de paquetes. Si no, dirígete a la página oficial del proyecto:

<https://github.com/numediart/MBROLA>

MBROLA es opcional: sáltate este paso si no quieres usar las voces más realistas que ofrece. :(

Ejecuta MBROLA para ver si puedes acceder al programa. Deberías ver su mensaje de ayuda.

```
$ mbrola -h
```

6.7.2. Arrancar *FlightGear* con soporte de voz

Para empezar, arranca Festival en modo servidor:

```
$ festival --server
```

Después, arranca *FlightGear* con el soporte de voz activado. Para ello, necesitas añadir la siguiente opción a tu línea de comandos:

```
--prop:/sim/sound/voices/enabled=true
```

Puedes poner esto en tu fichero de configuración aunque no vayas a usar Festival siempre. Únicamente verás algunos mensajes de error en la terminal, pero nada más. No puedes arrancar el subsistema de voz una vez has arrancado *FlightGear*.

Para probar que todo funciona, contacta con el ATC de KSFO mediante la tecla “”. Deberías escuchar tu voz primero (y ver el mensaje en amarillo en la parte superior de la pantalla), seguido de la respuesta de ATC con una voz diferente (acompañado del mensaje en texto verde).

6.7.3. Resolución de problemas

En algunas distribuciones de Linux, el acceso a Festival está restringido, y verás mensajes como estos:

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

Encontrarás más información en:

http://www.festvox.org/docs/manual-2.4.0/festival_28.html.

Puedes desactivar las restricciones a localhost y localhost.localdomain añadiendo lo siguiente a un fichero `.festivalrc` en `$HOME`:

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

O también puedes deshabilitar la lista de acceso directamente:

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

Esto último permite cualquier conexión. No importa mucho siempre y cuando estés protegido por un firewall.

6.7.4. Instalación de voces adicionales

Este proceso puede ser un poco aburrido, por lo que puedes saltártelo si estás a gusto con las voces por defecto. Para empezar, todos los datos en Festival van a un directorio común, como en *FlightGear*. Necesitas encontrar este directorio. Por ejemplo, `/usr/local/share/festival/`. Llamaremos a este directorio `$FESTIVAL`.

1. Busca las voces disponibles; puedes probarlas añadiendo `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. Puedes descargar las voces de Festival y los *wrappers* para MBROLA aquí:

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

La voz *don_diphone* es un poco primitiva, pero ocupa poco y suena bien en aeronaves IA. Debería instalarse bajo un directorio `$FESTIVAL/voices/english/don_diphone/`.

También necesitas instalar `festlex_OALD.tar.gz` y `$FESTIVAL/dicts/oald/`, y ejecutar el Makefile en dicho directorio. Es posible que necesites añadir `--heap 10000000` a la línea de comandos de Festival en dicho Makefile.

3. Recomendamos las voces “us2_mbrola”, “us3_mbrola” y “en1_mbrola”. Es necesario instalar MBROLA (ver más arriba) así como los siguientes *wrappers*:

```
festvox_us2.tar.gz
festvox_us3.tar.gz
festvox_en1.tar.gz
```

Se crearán directorios `$FESTIVAL/voices/english/us2_mbrola/` etc. El archivo de voz debe descargarse desde otra página (más información en el punto 4).

4. Puedes descargar las voces de MBROLA desde la página de MBROLA (ver más arriba). Busca las voces “us2” y “us3”. Descomprímelas en los directorios donde se han creado los *wrappers*: `$FESTIVAL/voices/english/us2_mbrola/` y similares.

6.8. Repostaje aéreo

Como su nombre indica, el repostaje aéreo (RA) implica repostar una aeronave, típicamente un caza de corto alcance) desde un avión cisterna. *FlightGear* soporta varios aviones cisterna, y cada uno de ellos soporta a su vez dos tipos de sistemas de repostaje. El primero despliega un brazo rígido que se conecta a un puerto en el avión receptor. El segundo sistema consiste en una manguera en la que el avión receptor inserta una sonda.

Hay varios aviones que soportan RA, incluidos el T-38, Lightning, A-4F, Vulcan, Victor (el cual es a su vez un avión cisterna) y A-6E. Si quieres saber si un avión soporta RA, fíjate en el menú *IA/Controles reabastecimiento en vuelo*. Si está accesible, la aeronave soporta RA.

6.8.1. Procedimiento

Para empezar, arranca *FlightGear* con un avión con soporte para RA, despegue y asciende a unos 15,000 pies. Una vez estable a dicha altitud, abre el menú *IA* → *Controles reabastecimiento en vuelo*, selecciona un modelo de avión cisterna en la lista desplegable y haz clic en **Request**. Se desplegará un avión cisterna en el aire.

FlightGear comunicará la altitud, rumbo, velocidad e identificador TACAN del avión cisterna. Ahora es momento de configurar tu TACAN con el identificador recibido, bien mediante el menú *Equipamiento* → *Configurar radio* o directamente en tus radios de cabina. Dependiendo de tu aeronave, el avión cisterna aparecerá en tu radar. Si necesitas ayuda para localizar el avión haz clic en el botón **Get Position** para recibir información relativa a tu posición.

Toma el rumbo adecuado con ayuda del TACAN (aproxímate desde la cola del avión cisterna) y busca el avión en el radar o la pantalla de navegación. Aproximadamente a 5 millas náuticas, reduce tu velocidad a 20 nudos por encima de la velocidad de la cisterna. Algunos

aviones de grandes dimensiones son visibles desde las 10 millas, pero otros más pequeños son invisibles más lejos de una milla. Si ves que vas a adelantar a la cisterna, activa tus aerofrenos.

Acércate a unos 15 metros de la cisterna, con cuidado con no chocar con ella. Habrá alguna indicación en cabina de que estás recibiendo combustible; algunos aviones disponen de una luz en el indicador de combustible, o simplemente la aguja del indicador subirá. Si aun así te es difícil adivinarlo, puedes activar la opción *Report refueling* en la ventana *Controles reabastecimiento en vuelo*. Si todavía te es difícil tomar contacto con el avión cisterna, prueba a aumentar el control *Contact radius* para prevenir desconexiones entre los dos aviones.

Una vez hayas llenado tus tanques, cierra un poco el gas para desengancharte y prosigue tu vuelo. La maniobra en sí es muy difícil y te llevará algo de práctica, como en la vida real. Aquí tienes algunas pistas para hacerla más fácil:

1. Acércate a la cisterna despacio. Es fácil adelantarla y perderla de vista.
2. Si tienes dificultades ajustando tu velocidad a la de la cisterna con tu palanca de potencia, activa tus aerofrenos. Así, necesitarás más potencia, lo que significa que la palanca será menos sensible.
3. Es posible usar el piloto automático para mantener altitud y/o velocidad, pero que sepas que es hacer trampa! No es como se hace en la vida real, aunque la NASA demostró hace tiempo que es posible repostar haciendo uso del piloto automático sin intervención humana.
4. Ten en cuenta que, al recibir combustible, el centro de gravedad de tu aeronave cambia, afectando el compensador de elevación.
5. El avión cisterna vuela siguiendo un circuito predefinido en el sentido de las agujas del reloj, y avisará poco antes de ejecutar un giro. Es posible seguir conectado en el transcurso de dichos giros, pero te será más fácil si esperas a que el giro haya terminado para conectarte.

6.8.2. Repostaje en multijugador

Es posible repostar en el aire en multijugador pilotando el KC135 o el Victor. El piloto de dicho avión debe usar uno de los siguientes identificadores: “MOBIL1”, “MOBIL2” o “MOBIL3” si se quiere usar TACAN. Es posible usar otros identificadores, pero sólo estos tres están asociados a canales TACAN aire-aire, en concreto 060X, 061X y 062X respectivamente.

Si el avión que está repostando está basado en la FDM YASim, esto es todo. Si, por el contrario, está basado en JSBSim, es necesario asegurarse de que no hay aviones cisterna IA activos, incluidos los activados mediante escenarios, por ejemplo dentro de `aircraft-set.xml` o `preferences.xml`.

El repostaje en multijugador funciona exactamente igual que el repostaje IA, y es muy recomendable! Para disfrutar al máximo, intenta que tu conexión esté libre de interrupciones; el sistema multijugador predice a veces algunos movimientos cuando la red pierde paquetes, lo cual puede complicar bastante el vuelo en formación.

Parte III
Tutorials

Capítulo 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 reality, 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 *Help* → *Tutorials* from the menu and click **Start Tutorial**.

The tutorial system works particularly well with the Festival Text To Speech system (6.7, [Soporte de Texto a Voz con Festival](#)).

For simplicity, run tutorials with AI aircraft disabled. Otherwise, Air Traffic Control (ATC) messages may make it difficult to hear your instructor. Via *AI* → *Traffic and Scenario Settings*, uncheck option *Enable AI traffic*.

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

Within a tutorial, you may request your instructor to repeat any instructions, by tapping **+**. You may pause the tutorial at any time using the **p** key. To stop the tutorial, select **Stop Tutorial** from the *Help* → *Tutorials* 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 Hilo International Airport (PHTO), Hawaii and Daniel K Inouye International Airport (PHNL), Honolulu, Hawaii. Both these airports are provided in the base package. To start the tutorials, select the *Cessna 172P* aircraft, and a starting airport of PHTO or PHNL, using the Launcher, or the command line:

```
fgfs --aircraft=c172p --airport=PHTO
```

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

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:

<https://ftp.igh.cnrs.fr/pub/flightgear/www/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:

https://www.faa.gov/air_traffic/publications/.

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 difficult, 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 want to spend bucks on getting a published pilot 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:

[Flight Simulator Navigation \(PDF\)](#)¹.

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

<https://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.

¹<https://static1.1.sqspcdn.com/static/f/1003521/14592887/1318406663403/Flight+Navigation.pdf>

Capítulo 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 bit of training, flying a small aircraft becomes easy. Yet if a problem occurs, you must be able to resolve it within 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](#) (see figure 8.1). This is the aircraft used in many real life flight schools and it is a great airplane to fly.

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

- <https://www.gleimaviation.com/resources/free-downloads/>
- http://www.pilotfriend.com/training/flight_training/aero/principa.htm
- <https://en.wikipedia.org/wiki/Aircraft>
- https://en.wikipedia.org/wiki/Flight_control_surfaces
- https://en.wikipedia.org/wiki/Airplane_flight_mechanics
- https://en.wikipedia.org/wiki/Aircraft_engine_controls



Figura 8.1: The Cessna 172P

- [Flight Simulator Navigation \(PDF\)](#)¹
- <http://www.free-online-private-pilot-ground-school.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. I will refer you back to Chapter 4, [Despegue: cómo arrancar el programa](#).

¹<https://static1.1.sqspcdn.com/static/f/1003521/14592887/1318406663403/Flight+Navigation.pdf>

8.2.1. Using the Launcher

FlightGear has a Launcher in which you can choose your aircraft and starting position. From along the left edge, select the *Aircraft* tab and then choose the *Cessna 172P Skyhawk* airplane (see figure 8.2). Click on the **Back** button if you need to choose a different aircraft.

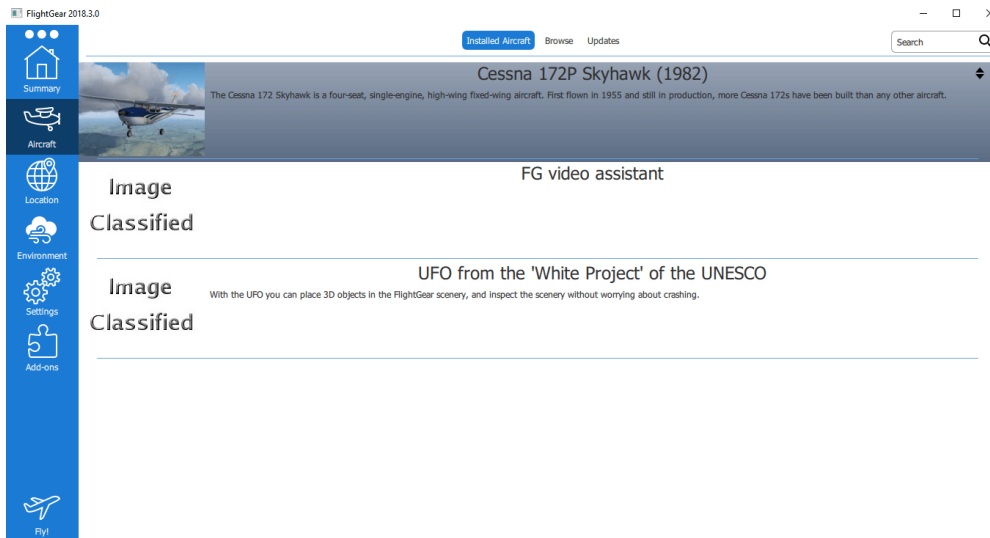


Figura 8.2: Choosing the aircraft

You can start from any airport for this tutorial, but I will assume that you will start from Daniel K. Inouye International airport at Honolulu (PHNL, see figure 8.3).

From the *Location* tab, once you have selected PHNL (click on the **Back** button if you need to change it), you can then set any number of options for the simulator. For your first flight, I suggest setting the *Time of day* to *Noon* within the *Environment* tab (see figure 8.4).

I also recommend that you start with a small resolution of 800×600 . You can change the resolution via the *Settings* tab by clicking on the **Show more** button of section **View and Window** (see figure 8.5). Later on, you can play around with the options and use a higher resolution, but this obviously adversely affects performance. Click on the **Fly!** button and *FlightGear* will start with the options you have selected.

If you have problems running the latest version *FlightGear* on your system, you may want to try an earlier version with lower graphics requirements. You can find previous releases on the FTP mirrors on the web page: <https://www.flightgear.org/download/mirror/>.



Figura 8.3: The airport diagram for PHNL in the built-in launcher

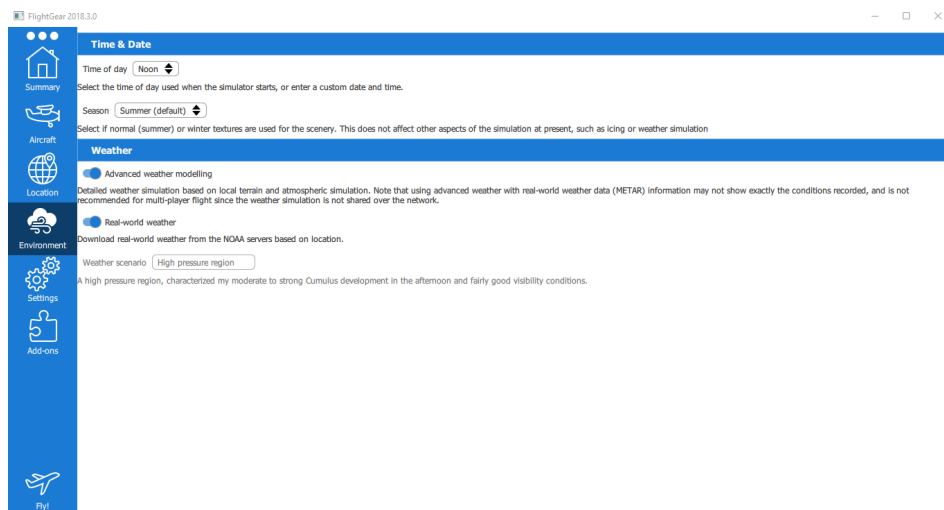


Figura 8.4: The *Environment* tab of the built-in launcher

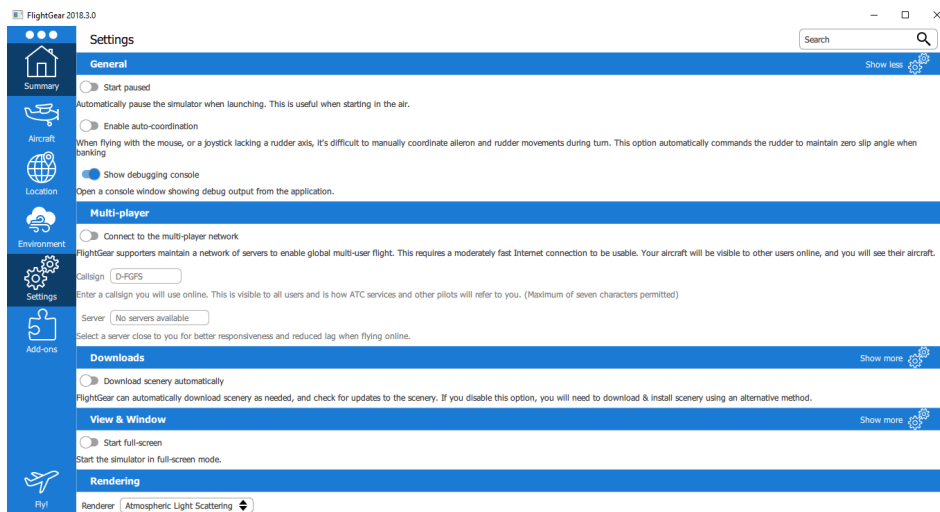


Figura 8.5: The *Settings* tab of the built-in launcher

8.3. The First Challenge – Flying Straight

Once *FlightGear* is started, you will see the scene shown in figure 8.6. Your first task is to get the engine running, which is most easily performed via the menu bar *Cessna C172P* → *Autostart*.

The aircraft is at the start of the runway with the engine running at low power. With the propeller now spinning, the airplane may occasionally tremble a little, but it should not move (see figure 8.7).

About the keyboard.

- In this tutorial, a lowercase letter key indicates you should simply press that key. An uppercase means you must press **Shift** (marked with blue in figure 8.8) and that key. In other words: if you are told to type “v”, simply hit the **v** key briefly. If you are told to type **V**, first press the **Shift** key and hold it down; while it is held down, press the **v** key, release it then release the **Shift** key. (In short: **V** is the same as **Shift-v**.)
- To be sure that the keys will work regardless of the platform on which you run *FlightGear*, this tutorial will assume you have the **NumLock** key on. When this is the case, you should see a small green LED at the right of your keyboard. Press the **NumLock** key once to turn the LED on in case it is switched off (marked with green in figure 8.8).

Having NumLock turned on, we can use the **Home/End** (for trim) and **PageUp/PageDown** (throttle control) keys, located above the arrow keys (marked with red in figure 8.8). Normally for the Cessna 172P, you should be able to use these keys from the numeric keypad as well, but depending on the platform and operating system it may be different.



Figura 8.6: Automatic starting of the Cessna 172P



Figura 8.7: Cessna 172P on runway with engine started

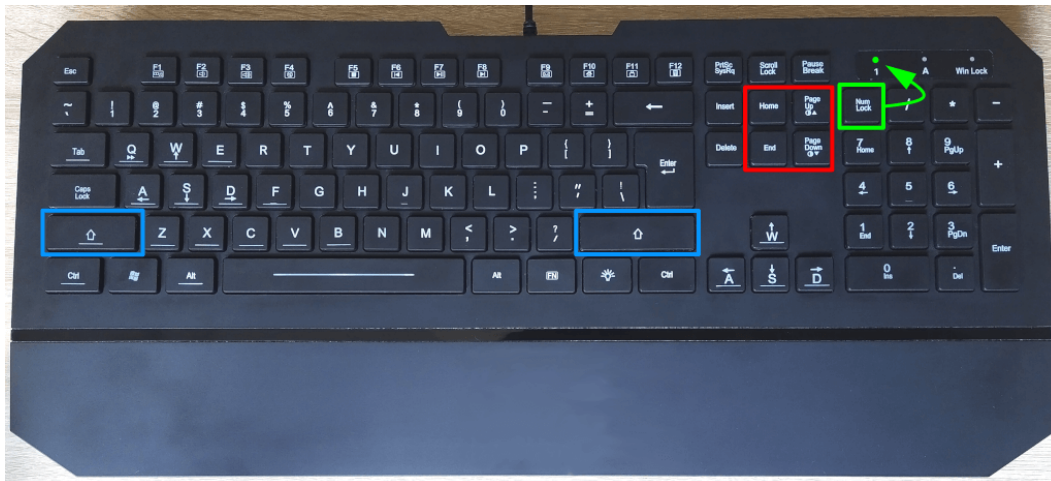


Figura 8.8: The **Shift** keys (boxed in blue), **NumLock** key (boxed in green), **Home**, **End**, **PageUp** and **PageDown** keys (boxed in red)

Press **v** to view the aircraft from the outside, as shown in figure 8.9. Type **v** repeatedly to cycle through a number of different available views until you return to the cockpit. Pressing **V** will cycle backwards through the views. **Ctrl-v** will return you directly back to the cockpit view.



Figura 8.9: External views

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

Hold the **PageUp** (or **NumPad-9**) key down for about eight seconds. You will hear the engine sound rise as the throttle is advanced to full power.

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. Figure 8.10 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, reset *FlightGear* using *File* → *Reset* (or the keyboard shortcut

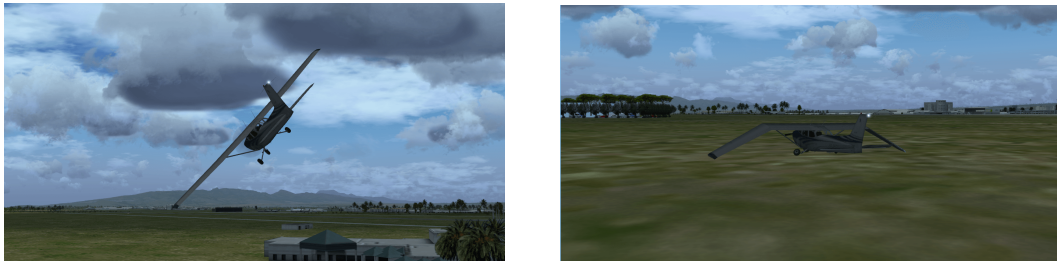


Figura 8.10: Viewing the crash using *Instant replay*

Shift-Esc), to return to the pre-crash condition.

In order to fly straight, we will need to use the airplane control yoke (see figure 8.11).



Figura 8.11: The 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 into that mode by pressing **Tab**. The mouse cursor becomes a + symbol. 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 movement of the rudder, elevator (tail) and the ailerons (wings). 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. Use **Ctrl-v** to change the view back into the cockpit.

Pressing **Tab** again gets you into mouse view mode. In this mode the mouse cursor will be a ↔ symbol. This allows you to look around easily by moving the mouse. Clicking the left mouse button will re-center the view. You can also change your view direction in the normal and yoke modes by holding down the right mouse button and moving the mouse. A further press of **Tab** will return you to the normal mouse mode.

To summarize, the **Tab** key 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).
- *View mode.* The mouse controls the view direction (\leftrightarrow pointer).

Try taking off again using the mouse to control the yoke. Press **Tab** to put the mouse in yoke mode (+ pointer) and raise the engine throttle to maximum by holding the **PageUp** key down. Do not try too hard to keep the airplane rolling straight on the runway using the mouse / yoke. It is okay to let it drift a bit leftwards. Wait until it rises into 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 (cf. figure 8.12)... or from plunging toward the ground (figure 8.13). Try to fly more or less straight, with the horizon line slightly above the airplane nose (see figure 8.14).

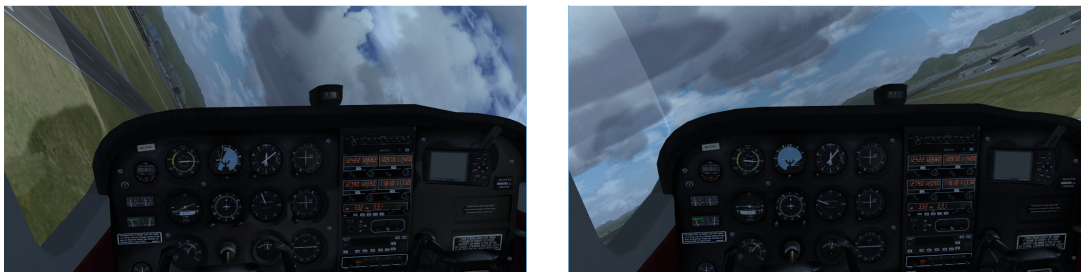


Figura 8.12: Left or right banking



Figura 8.13: An intentional descent?

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 it is common for the novice pilot to begin to despair 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.



Figura 8.14: Straight and level flight

Common Mistakes

The most common errors are moving the controls too much, which lead to the aircraft getting out of your control, or moving the mouse forward in an attempt to bring the nose up.

You will learn that small yoke movements have a fairly large effect on the aircraft's reaction. Go easy on the controls by using small yoke movement. You may always add a touch more control input if you see the aircraft is responding too subtly. Inversely, too much yoke movement can cause a severe response from the aircraft and then you find yourself in a poor situation as you desperately attempt to regain control. If over-control is especially difficult for you, then you may find that decreasing your mouse sensitivity could help during your initial training.

Remember, pull the nose up and push the nose down. You must pull the yoke aft, by moving the mouse backwards, to raise the nose. 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.

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 forward the ball will roll forward and your fingers will point toward the desk. If you move your hand backward the ball will roll backward and your fingers will now point up at the ceiling. Your hand is the airplane.

Another common mistake 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), do not move it close to the FlightGear window edge. 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 pressing **Tab** twice. Or use *FlightGear* in full screen mode.

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 represented in figure 8.15 shows an altitude of 800 ft, i.e. approximately 245 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



Figura 8.15: The altimeter

mouse mode) inside the black knob and drag the mouse left or right. A drag toward the left makes the altimeter decrease in indicated altitude. Dragging toward the right will result in an increase in indicated altitude. An even more convenient method is to use the mouse wheel on the black altimeter knob. Use that little knob to set the altimeter to the field elevation of your airport. The principle is that you use the knob when you know your present altitude. If you know you are at 1100 ft altitude, tune in 1100 ft on the altimeter. Type **Ctrl-c** to see the knob highlighted so that you understand the allowable click region.

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 mean sea level” and “altitude above ground level”. If you fly near Mount Everest at an altitude of 24 000 ft above mean sea level (MSL), your altitude above ground level (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:

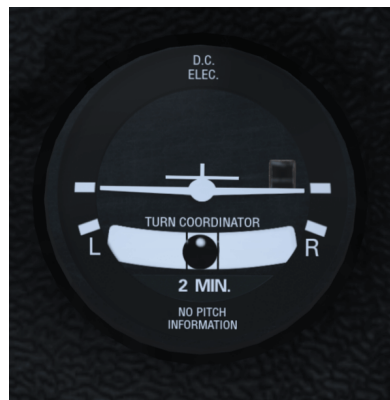


Figura 8.16: Turn coordinator

- When the airplane is banked to the left, it turns to the left.
- When 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. Besides, passengers find excessive bank angles very disheartening. They don't like it!

The turn coordinator does NOT represent bank angle.



The inquisitive pilot may be interested in the fact that the bank angle required to establish a 3°, standard rate turn varies with True Airspeed (TAS). You could calculate an approximation with the formula:

$$\text{angle} \approx \frac{\text{TAS(knots)}}{10} + 7$$

TAS (knots)	formula	angle
80	8 + 7	15°
100	10 + 7	17°
120	12 + 7	19°

The turn coordinator represents Rate of Turn. The standard rate of turn is defined as 3° of travel per second. Why is this important? Frankly, because it can save your life! If you even find yourself suddenly in poor visibility, your smart reaction is to turn around 180°. In poor visibility, how are you suppose to know when you have turned halfway around? Hopefully, the standard rate of turn is now making sense to you. If you perform a standard rate turn, covering 3° of the turn every second, then a 180° turn will complete in one minute. After two minutes, you would have turned in a full circle and returned back to where you started your turn. This explains the “2 minute” placard you often see printed on the face of the instrument.

With this knowledge, let us explore how to properly utilize the turn coordinator instrument. As you bank the aircraft to the right, notice how the airplane depicted on the instrument lowers

its right wing. Likewise, banking the aircraft to the left is duplicated on the face of the turn coordinator. When the wingtip aligns with the tickmark along the rim, at that point neutralize the control yoke to maintain that angle. Glance quickly at the clock and roll out of the turn one minute later. I know you were curious as to why airplanes had clocks.

Try the following: keep the airplane banked in a standard rate turn 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 demonstrates that 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 hold a bank at the standard rate you always need 60 seconds to make a 180° turn in the Cessna 172P (or any other aircraft).

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, in relation to the horizon, establishes you into a straight and level flight path.

The little ball, at the bottom of the turn indicator, shows the sideways forces acting upon the aircraft. In real life you would feel these forces as you turn. However, it is not possible to simulate these, so you must simply keep an “eye on the ball”. Think of the ball as representing the position of the tail of the airplane. If you turn neatly (a.k.a. a coordinated turn), the ball will remain centered. If the ball is pushed, say rightwards, this means that you the pilot too are pushed rightwards. During a coordinated turn, even a strong turn, the plane occupants do not endure a sideways force. They are only pushed a little harder into their seats by the centrifugal force. To maintain coordination, simply “step on the ball”. When you see the ball deflected toward the right, push on the right pedal to return the ball to the center. Likewise, the left pedal will return the ball to the center when deflected to the left.

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 to aircraft such as the Cessna.

8.5. Taxiing on the ground

While *FlightGear* is able to start you off conveniently lined up on the runway, you may be wondering how to get your aircraft from its hangar, along the taxiways, to the runway. This is taxiing.

Figure 8.17 shows the tachometer instrument. It displays how fast the engine is turning in hundreds of revolutions per minute (RPM).

Tap the **PageUp** key a few times, until the tachometer is showing 1,000 RPM (as shown above). If required tap the **PageDown** 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.

Tap 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 tap 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.



Figura 8.17: Tachometer

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 ramp and taxiways. 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 toward. 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), 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. Press **Tab** to get in yoke control mode (+ pointer). 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 the surface.

8.5.1. Airspeed



Figura 8.18: Airspeed Indicator

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 to nautical miles per hour (knots).



One nautical mile is the distance covered by 1 arc minute of latitude (1852 meters.)
 One knot is equivalent to the speed required to travel one nautical mile per hour.
 (60 knots = 60 nm/hour)

A knot is 1.852 km/h. 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 kt 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.

Figure 8.19 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 shown in figure 8.20. At airspeeds above 40 knots, the rudder has enough airflow 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.



Figura 8.19: The front wheel is slightly lifted.



Figura 8.20: The tail rudder

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. Do not 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 main 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 engine torque. As the propeller rotates clockwise, an opposite force is rolling the aircraft counter-clockwise. This transfers a force onto the left main gear and tire. This force results in a slight increase in drag and a slightly smaller tire radius. A secondary 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 the front of the aircraft to yaw a bit 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/ailerons 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 figure 8.21, 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.



Figura 8.21: Turn coordinator

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/ailerons 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 (see figure 8.22).



Figura 8.22: Magneto

To better view the magneto switch, click the place where the yoke column enters the cockpit to hide the yoke. You can also press **x** key several times 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 to 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 preflight 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 with the mouse by simply clicking the left and middle mouse button. Click and hold near “START” to start the engine.

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



Figura 8.23: Throttle & mixture

You already know that you increase the engine power by pushing that throttle rod in (**PageUp** key). You decrease the power by pulling the control out (**PageDown** key). You can also use the mouse wheel while its cursor is on the throttle, or you can hold down the left mouse button and drag the mouse.

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 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. Figure 8.23 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 (**m** key). 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**. 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 or **M** key.) 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).)

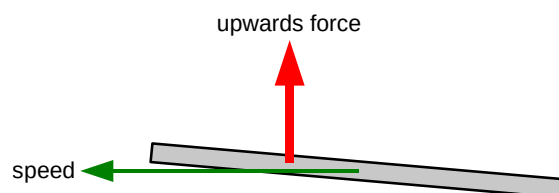


Figura 8.24: Lift

- 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!



Figura 8.25: Airfoil

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 section 8.7.5), 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 attack, 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 (see figure 8.26).

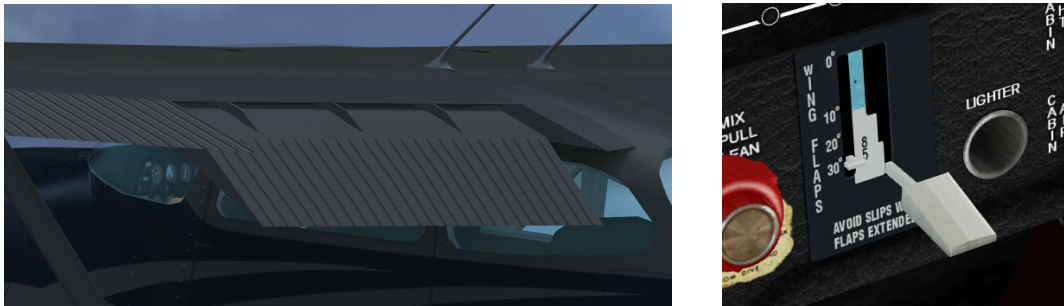


Figura 8.26: Flaps & flaps lever

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 stage 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**-→ to shift the view to the right and then quickly **Shift**-↑ 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



Figura 8.27: Trim

The trim is the dark big vertical wheel with gray dots located at the middle below the instrument panel (see figure 8.27).

In *FlightGear*, the keys **Home** and **End** adjust the trim. **Home** rolls the wheel upwards while the **End** rolls the wheel downwards. You can also left-click the trimmer wheel to rotate it up, or middle-click to rotate it down.

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.
 - Hold down the right mouse button in normal or yoke mode and move the mouse to change the view direction.
 - Use the mouse in view mode (**Tab**, ↔). 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 or “heading indicator” (see figure 8.28). 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”).)



Figure 8.28: Magnetic compass & heading indicator

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



Figure 8.29: Attitude indicator

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. 5.1), 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 (see figure 8.29).

Left of the artificial horizon, you'll see the airspeed indicator shown in figure 8.18. Not only does it provide a speed indication in knots but also several arcs showing characteristic velocity

rages 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 exceed, unless you want to break up the plane in mid-flights. . .

Below the airspeed indicator, you can find the turn indicator (cf. figure 8.16). 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 and 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 (cf. figure 8.15). Below the altimeter is the vertical speed indicator (see figure 8.30) 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.



Figura 8.30: Vertical Speed Indicator

Below the turn coordinator is the propeller tachometer, or RPM (rotations per minute) indicator, which displays the rotations per minute in hundreds (cf. figure 8.17). 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; both are shown in figure 8.28.

Four of these gauges being arranged in the form 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. 8.22). 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 purposes 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 () , 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 *AI* → *ATC Services in Range* 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 ft. 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 1000 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 – 0 ft.
2. Check aileron and elevator are neutral by observing the yoke position.
3. Change the mouse to control mode by pressing **Tab**.
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 ft, turn to your required heading, staying away from buildings until you are over 1000 ft.

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 500 ft until on final approach to the runway.
- Approach the runway at approximately 70 knots.
- Touch down on the two main wheels at 55 kts.
- Let the nosewheel touch down at 40 kts.

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 (see figure 8.31 (a)). 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. 1500 ft above the airport, and a couple of miles out, an in-line with the runway, reduce power to approximately 1500 RPM. This will slow you down somewhat and start a gradual descent.
2. Once below 115 knots, apply one step of flaps (1). This will increase lift, but also slow you down.
3. Re-trim the aircraft so you continue to descend.
4. At around 1000 ft, apply another step of flaps (1). This increase drag significantly, but also improve the view over the nose.

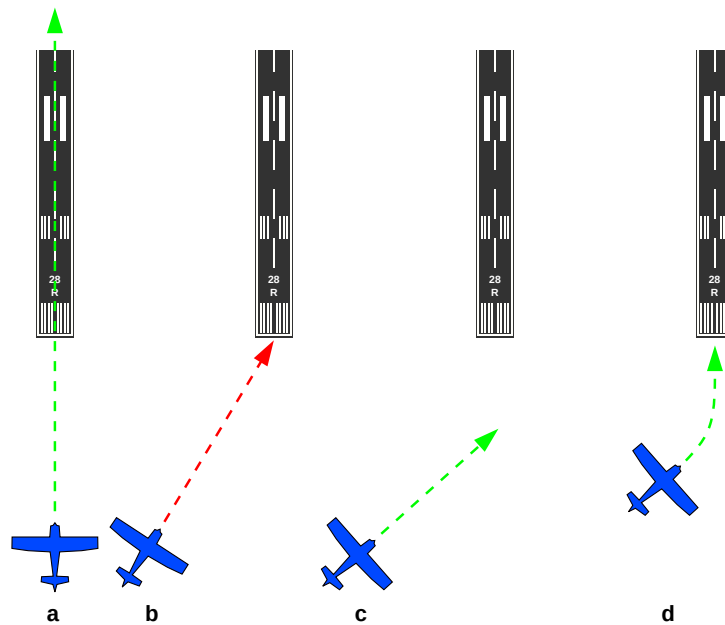


Figura 8.31: Alignment to the runway

5. Tune the speed using the elevator and trim: push the yoke if you are flying below 70 kt, 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 500 ft, 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.



Figura 8.32: Landing

8.8.3. Engine Shutdown

To shut the engine down:

- Set the parking brake, type **B**.
- Engine throttle to minimum (hold **PageDown** 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 does not 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 (**L**).

Landing is much harder than taking off. Landing on a large runway, such as KSFO (San Francisco) 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 figure 8.33.

- 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.

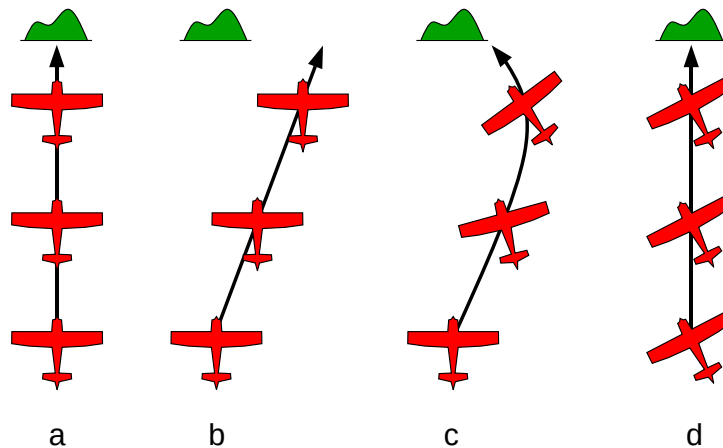


Figura 8.33: Side wind

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 figure 8.34 shows an airspeed of 5 knots.



Figura 8.34: Windsock

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 (toward 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



Figura 8.35: 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 figure 8.35 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 modelled in the Cessna 172: [Bendix King website](#).

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 taught 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 flare 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:

- https://en.wikipedia.org/wiki/Piper_Cherokee
- <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 ft 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 (see the arrow on the right side of the vertical line shown in figure 8.36).
- It is not mandatory to use the rudder to keep on the runway. The airplane will take off before it drifts off the runway. (For sure it is better and more “secure” to keep in the middle of the runway. But using the rudder can make things hectic for a beginner.)
- Once above about 160 knots, the plane rises its nose in the air. Immediately push the yoke back to neutral or almost and stabilize at 200 knots airspeed (which makes a fair climb angle) (I've no idea whether 200 knots is the right climb speed for a real A-4. What's more I suppose one should rather use the AOA (see below).)



Figura 8.36: Yoke indicator on the HUD

- Retract the landing gear using key **g**.
- Either maintain $\frac{1}{2}$ engine power and a speed of 200 knots to get above the clouds, or reduce the engine power to less than $\frac{1}{4}$ and fly normally. (Off course you can “fly normally” with full engine power. Great fun.)

The “nervous” take off procedure is the same but you push in full engine power. The plane takes off quickly and you need to settle a very steep climb angle to keep 200 knots. Best retract the landing gear immediately.

You don’t land a jet the same way you land a little propeller airplane. My way to land the A-4, inspired by some texts I found on the Web, is this:

- Really far from the runway, keep below 2000 ft and get the speed below 200 knots. Then lower the landing gear (key **G**) and I deploy full flaps (all three steps, by hitting **J** three times.)
- Keep a steady altitude of about 1000 ft and a speed of “exactly” 150 knots. Use the mouse / yoke / elevator to tune the altitude and the engine throttle to tune the speed. (The opposite from the Cessna.)
- Try to align with the runway.
- When do you know the dive towards the runway must begin? For this you need the HUD; the full default HUD with lots of features. Look at the figure 8.37. When you see the “distance” between the green 0° lines and the runway start is 25 % the distance between the green 0° lines and the green -10° dotted line, it is time to dive, aiming at the runway start. (In the figure 8.37, that “distance” is 64 %, far too much to start a landing.)

Let’s explain this. The two horizontal lines labeled “0” show the horizon line. Rather they show where the horizon would be if the Earth was flat. When your eyes aim at those “0” lines, you are looking horizontally. Look at the dotted green lines labeled “-10”. A feature

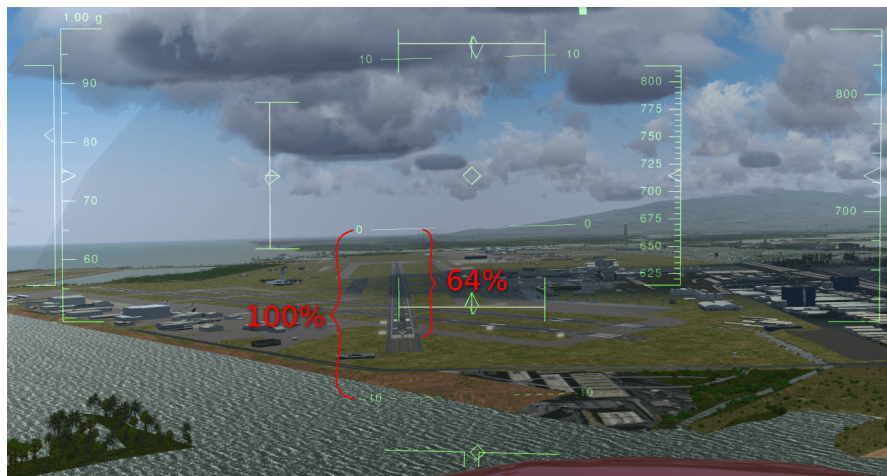


Figura 8.37: Aiming at the threshold of the runway with the HUD

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 figure 8.37, the start of the runway is situated at 64 % of the way towards the green “ -10° ” dotted lines. That means you have to lower your eyes of 6.4° 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° (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° (up to 3°), that is 25 % of -10° (up to 30 %).

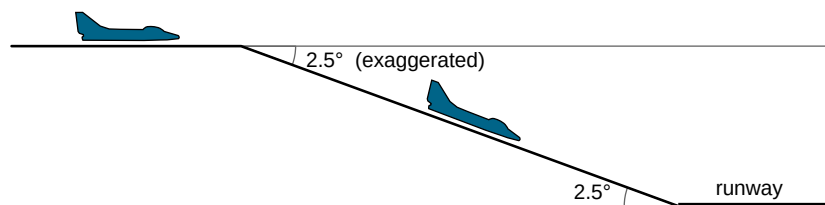


Figura 8.38: Glideslope

- 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° (that is 25 % of -10°):
 - If the angle increases above 2.5° , you are above the desired path and you must loose

altitude faster. Both decrease the engine power and dive the nose a little.

- If the angle decreases below 2.5° , 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 figure 8.39. 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.)

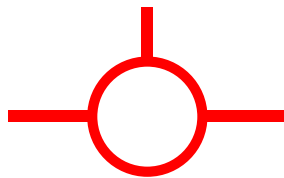


Figura 8.39: Aim point symbol

Also, a real HUD shows a dotted line at -2.5° , to help find the correct descent path. Simply keep that dotted line on the runway thresh-hold.

In additional 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 (see figure 8.40). When the upper \vee 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



Figura 8.40: The HUD in F-14B

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° 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 loose 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 1000 ft 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 kinds 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.

Capítulo 9

A Cross Country Flight Tutorial

9.1. Introduction

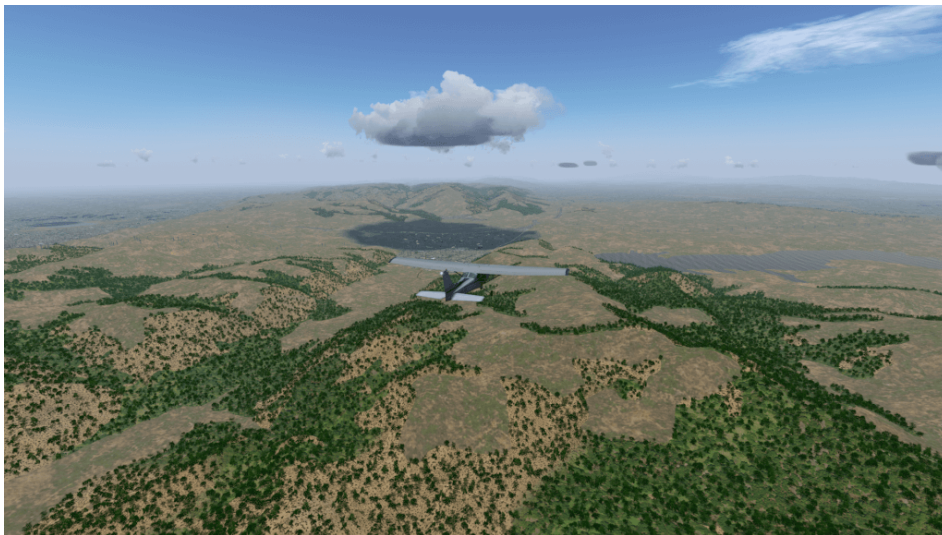


Figura 9.1: Flying over the San Antonio Dam to Livermore

This tutorial simulates a cross-country flight from Reid-Hillview (KRHV) to Livermore (KLVK) under Visual Flight Rules (VFR). Both airports are not included in the standard *FlightGear* package, therefore make sure that in the Launcher you have enabled the option *Download scenery automatically*, which is in the *Settings* tab.

I'll assume that you are happy taking off, climbing, turning, descending and landing in *FlightGear*. If not, have a look at the tutorials in chapters 7 ([Tutorials](#)) and 8 ([A Basic Flight Simulator Tutorial](#)). 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](mailto:stuart_d_buchanan@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 an aeronautical map which shows 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:

<https://skyvector.com>

Simply click "Flight Plan" and enter KRHV to "Departure" field. 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.

An even simpler option is to use "Phi", which is a map in the browser. Just run *FlightGear* with the command line option `--httpd=8080` and then in the simulator choose *Equipment* → *Map* (*opens in browser*) from main menu.

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 you can find them in Launcher in *Location* tab. (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° magnetic towards Livermore (KLVK). We'll fly at about 3500 ft about sea-level. This puts us at least 500 ft 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 Antonia 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.

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 (which is recommended to be able to listen the ATIS.) To do this, in the Launcher, go to the *Environment* tab and check the *Real-world weather* option.



Figura 9.3: On the runway at KRHV

9.3.1. Preflight

Before we take off, we need to preflight 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. If you want, you can do this by selecting *Walker* → *Toggle Walker outside* from the main menu.

However, 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 *AI* → *ATC Services in Range* menu. Then, in the dialog box, click the button with the ICAO code of our airport, i.e. KRHV. 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 125.2 MHz.

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.



Figura 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 6 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.



Figura 9.5: COMM1 adjustment knob

To change the frequency, use the mouse wheel on the black knob below the standby frequency (highlighted in the Figure 9.5), just to the right of the 'STBY'. Turning the mouse wheel over the small (inner) knob changes the decimal point (kHz), and turning the wheel over the large (outer) knob changes the number before the decimal place (MHz). Turning the wheel in both ways – you will see the values go up and down. Alternatively, you can click each knob. Left click mouse button increasing the frequency, and middle click decreasing.

As for the COMM radio, note that the frequency before the decimal point (MHz) can be set in the range from 118 to 136, in 1 MHz step. However, the frequency after the decimal point can work in two standards, the older one with a spacing of 25 kHz, or the new one with a spacing of 8.33 kHz. The default is 8.33 kHz spacing, giving you more frequencies to choose from. If you want you can switch the radio between 25 kHz and 8.33 kHz spacing, just hold down the **Shift** key and click the little knob. When you look closely, you will notice that a small, inner knob is pushed in.

If you are having difficulty clicking on the correct place, press **Ctrl-c** to highlight the hot-spots for clicking.

Once you have tuned the frequency to 125.2 MHz, press the white button with arrows, located 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. If you can't hear ATIS, make sure the 'COMM1' switch above the frequencies is in the *Speaker* or *Phone* position (see Figure 9.7).

9.3.4. Altimeter and Heading Indicator



Figura 9.6: COMM1 transfer key to swap the frequencies



Figura 9.7: Speakers switch for COMM1



Figura 9.8: Altimeter calibration knob

Listen for the ‘Altimeter’ setting. If you are not using real weather, the atmospheric pressure will depend on the weather conditions set, according to the selected weather scenario. Then listening to ATIS will be impossible. Therefore, to check the pressure, go to menu *Environment* → *Weather*. In the bottom of new window, there is a METAR message with atmospheric pressure. If you don’t know how to read the METAR please use the "METAR Description" button.

As you can see, the atmospheric pressure can be different every time. Therefore, we 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 9.8), in the same way as you changed the radio frequency. Turning the knob changes the value in the little window on the right of the altimeter. You must set the same value as given by ATIS (or METAR). Notice that the altitude indicated by the altimeter is changing too.

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 133 ft. This means you can double-check the pressure value reported over ATIS.



As you may have noticed, the pressure on the Cessna altimeter can only be set in units of inHg. Fortunately, we fly in the USA, where ATIS also gives us the pressure in inHg. But when flying in Europe, for example, the pressure will be given in hPa. Then we have to convert hPa to inHg, or simply use the "METAR Description" window mentioned above. Another way is to use the menu *Equipment* → *Instrument Settings*, where you can set the pressure for the altimeter, both in hPa and inHg.

We will also take the opportunity to set the on the heading indicator to 350° – our bearing from KRHV to KLVK. To do this, use the amber knob on the heading indicator housing (highlighted in Figure 9.9), just as you’ve done before. For faster rotation, hold down the **Shift** key to increase the increment value by 5°. The value of 350° is just anti-clockwise of the labeled



Figura 9.9: Heading adjust knob

value of N (North – 0°).

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 1000 ft, make a gentle turn to the right to a heading of 350°. As we've set the heading bug, it will be easy to follow. We're aiming for a fairly prominent valley.

Continue climbing to 3500 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.

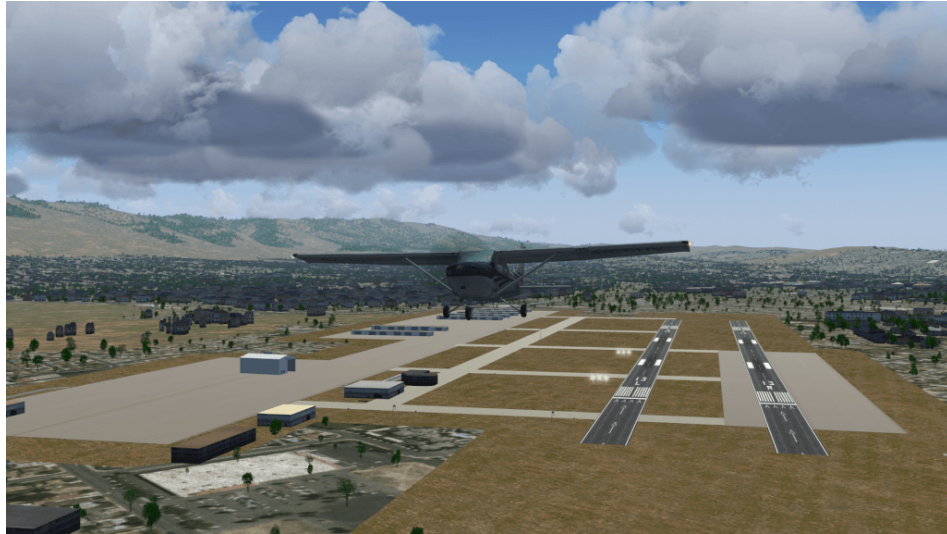


Figura 9.10: Take-off from KRHV



Figura 9.11: The C172 Autopilot

The autopilot panel is located towards the bottom of the radio stack (highlighted in Figure 9.11). It is easily distinguishable as it has many more buttons than 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 heading indicator, 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 ailerons 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

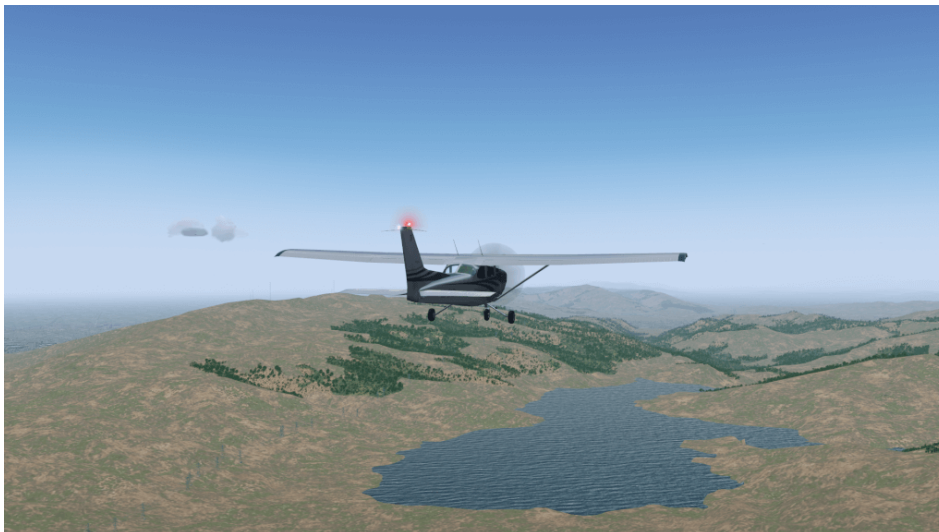


Figura 9.12: The Calaveras Reservoir

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

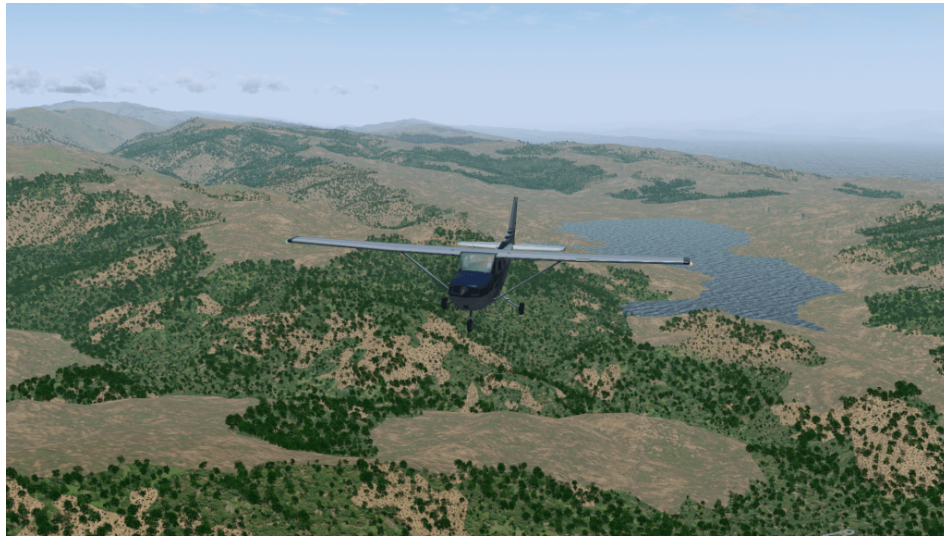


Figura 9.13: The Calaveras Reservoir

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 (Figure 9.14). 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.

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, hold down the right mouse button. Moving the mouse now pans the view. Once you can see the mixture lever clearly, release the right mouse button.

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 (see Figure 9.15). We're now running slightly rich of peak. While at 3500 ft 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, assu-



Figura 9.14: Mixture Control



Figura 9.15: EGT gauge

ming 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 Frequencies dialog from the *Equipment* → *Radio Settings* menu. The Livermore ATIS frequency is 119.65 MHz.

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 118.1 MHz. 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.

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 click **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 callsign 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 9.16. The 'downwind' mentioned above refers to one of these, the one with the number 3.

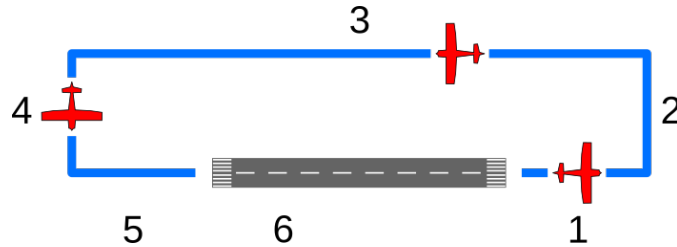


Figura 9.16: The Traffic Pattern

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 500 ft and 1000 ft 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 500 fpm 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.

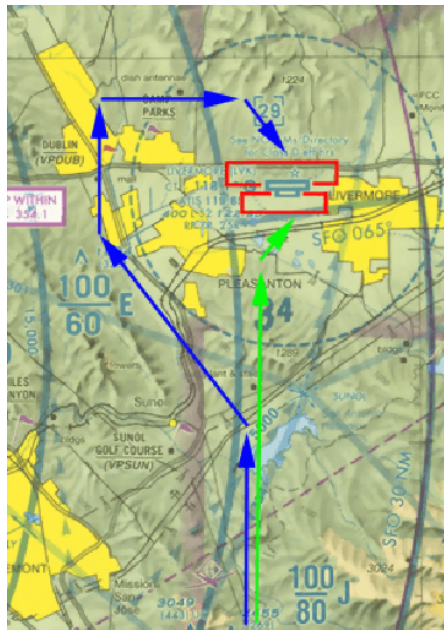


Figura 9.17: Sectional extract showing approaches to Livermore

9.5.3. Approach

We're approaching Livermore airport from the South, while the runways align East to 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 pattern. Both the patterns are illustrated in Figure 9.17. 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 1000 ft above ground level (AGL). Livermore airport is at 400 ft above sea-level (ASL), so we need to descend to an altitude of 1400 ft 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 appropriate pattern as shown in the figure 9.17.

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



Figura 9.18: 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 9.18). This is the VASI (Visual Approach Slope Indicator) 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 60 kts, a descent rate of about 500 fpm should be fine. If you are too high, just decrease power to increase your descent rate to 700 fpm. If you are too low, increase power to decrease your descent rate to 200 fpm.

9.5.5. Go Around



Figura 9.19: Missed approach at Livermore

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.
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).

Capítulo 10

An IFR Cross Country Flight Tutorial

10.1. Introduction



Figura 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 ft above ground level, and 800 metre visibility. This tutorial assumes you've completed the cross country flight tutorial.

10.1.1. Disclaimers

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.



Figura 10.2: On runway 31R at KRHV

This flight was flown using FlightGear 3.0. Newer or older versions of FlightGear might be slightly different.

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** ⇒ **Weather** to bring up the weather dialog. In the **Weather Conditions** list, select **CAT I minimum**.

This will give us a low ceiling and reduced visibility. Unfortunately, it will also give us rather stiff winds. If you don’t want to deal with them, then you can easily turn off the winds:

- Click on **Weather Conditions** again, 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°).

Hit **OK** to make FlightGear accept the changes and close the dialog.

Finally, I find that the reduced visibility situations are rendered best when atmospheric light scattering is turned off: click **View** ⇒ **Rendering Options** and make sure the **Atmospheric light scattering** box is unchecked.

10.2.1. Flight Planning

When you look out the window, 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 $\cdots \text{-----} \text{---}\cdot$).

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, 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? NAV1 \Rightarrow 114.1³

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. Why? Check the audio panel (see Figure 10.4). You'll note there's a switch for all the instruments that produce useful sounds, and NAV1 is one of them. Flip the switch up (or down

¹See https://en.wikipedia.org/wiki/VHF_omnidirectional_range for more information.

²Operation of the COMM receivers was covered in the cross country flight tutorial.

³All 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.

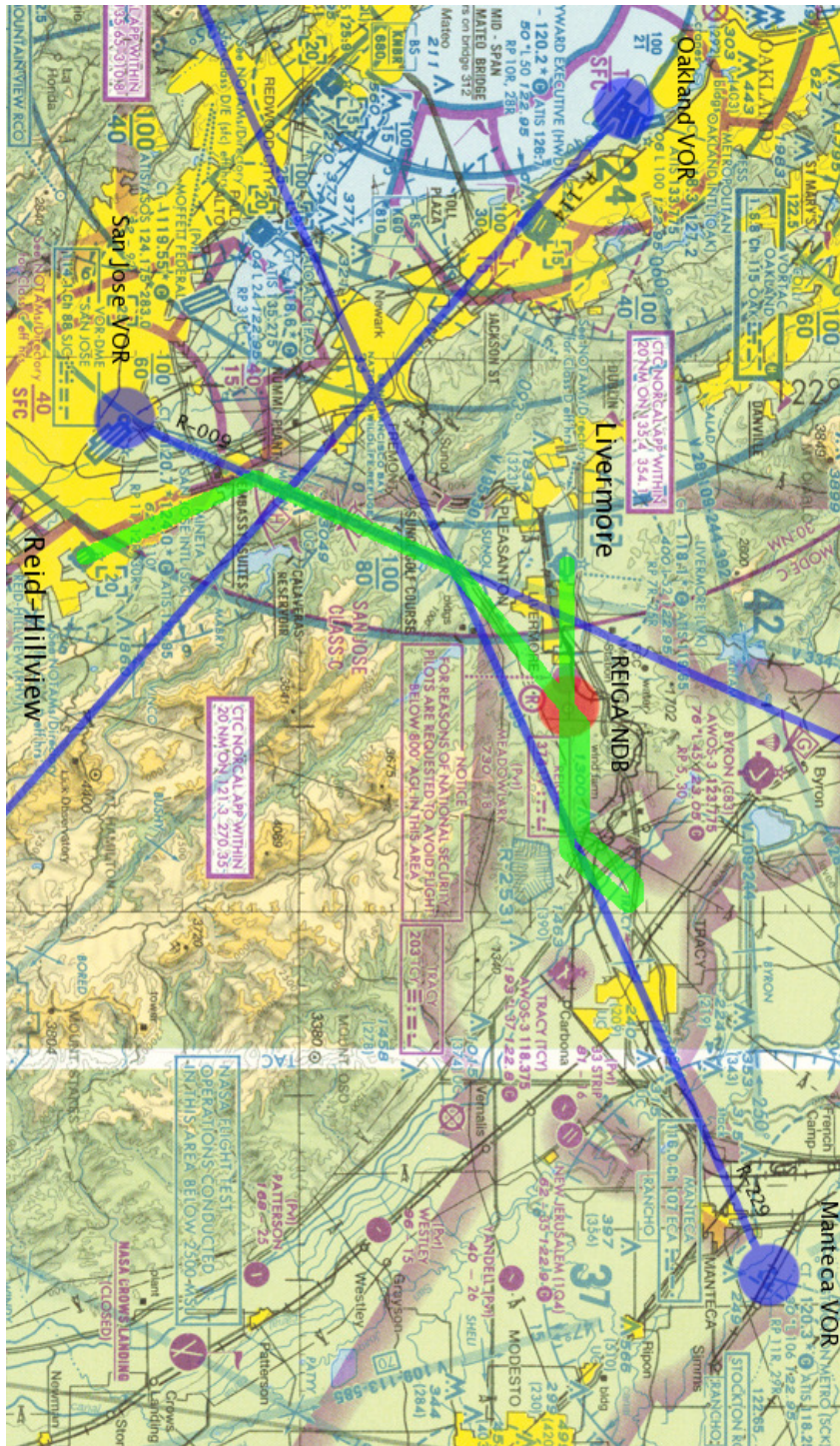


Figura 10.3: Green: our route, Blue: VORs and radials, Red: NDBs

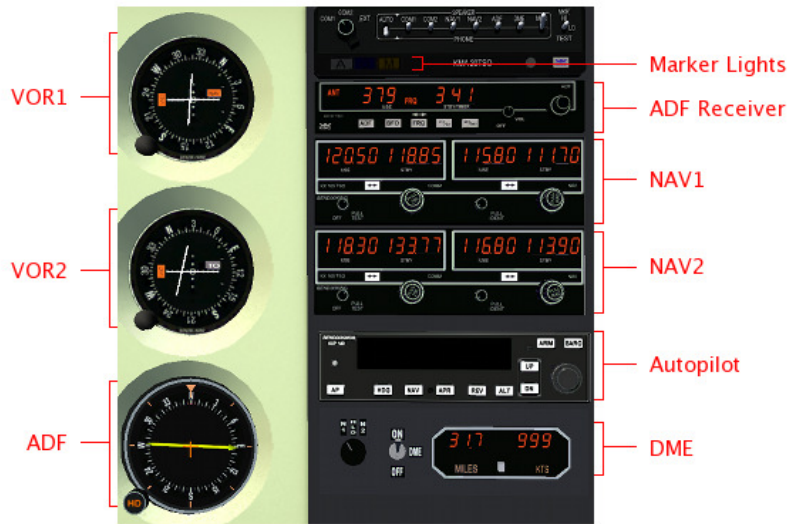


Figura 10.4: IFR navigation instruments



Figura 10.5: VOR1, before and after tuning

– it doesn't matter), and you should hear this: ... ·---- ····.⁴ Nice. Flip the switch back to the centre when you get tired of listening to dots and dashes.

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.⁵ 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 "9°" (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°.

One final thing – set the heading bug on the directional gyro to our current heading (about 310°).

10.2.3. How High Are We Really?

One effect of our changing the weather conditions is that the barometric pressure is no longer the standard value of 29.92. Our altimeter needs to know the correct value, otherwise it will report the wrong altitude. This isn't critical at takeoff, but it can make a *huge* difference when descending through the clouds (can you say "controlled flight into terrain"?).

As described in the cross-country flight tutorial, we get the current barometric pressure via ATIS. To recap, click **AI** ⇒ **ATC Services in Range**, select our airport, and look up the ATIS frequency (it should be 125.2 MHz). Dial this frequency into COMM1 or COMM2 (remembering to flip the appropriate switch on the audio panel), listen to the ATIS report, and set the altimeter to the given barometric pressure.

We are going to be using the autopilot (see Figures 10.4 and 10.6) to hold an altitude (more on that later), so it also needs to know the barometric pressure. To do so, click the BARO button on the autopilot. You should see "29.92" displayed – this is what the autopilot thinks the barometric pressure is. Before the "29.92" disappears (within about 3 seconds), rotate the big dial to change it to the correct value.

10.3. Takeoff

We're ready to take off. There are other 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. Take off, keeping a heading of 310° for now. Establish a steady rate of climb. We plan

⁴Still can't hear it? Check the volume control on the NAV1 receiver. If that has no effect, click **File** ⇒ **Sound Configuration** and adjust the settings. If that doesn't work, check the volume on your computer. If that doesn't work, and you have external speakers, adjust the volume on the speakers. And if that doesn't work, check your ears.

⁵The horizontal needle is used in ILS landings, which will be explained later.

R1 OBS ⇒ 009

heading bug ⇒ 310

Take off; climb on runway heading



Figura 10.6: Autopilot after engaging

to climb to 4000 ft. There's just one problem though – those ugly-looking clouds are standing in our way.

10.4. 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.

10.4.1. 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 10.6, 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.

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.

Engage autopilot;
set vertical speed;
engage heading
mode



Figura 10.7: Typical IFR scenery

10.4.2. 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 10.7), 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 10.8 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 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 in the audio panel and make sure you're getting the correct ident.

V2 ⇒ 116.8

We need to adjust the OBS, to tell VOR2 which radial we're interested in. Set the OBS to 114.⁶ 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.

R2 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.

⁶If you get tired of clicking on the knobs, much of this can be done more easily using the **Equipment** ⇒ **Radio Settings** dialog.

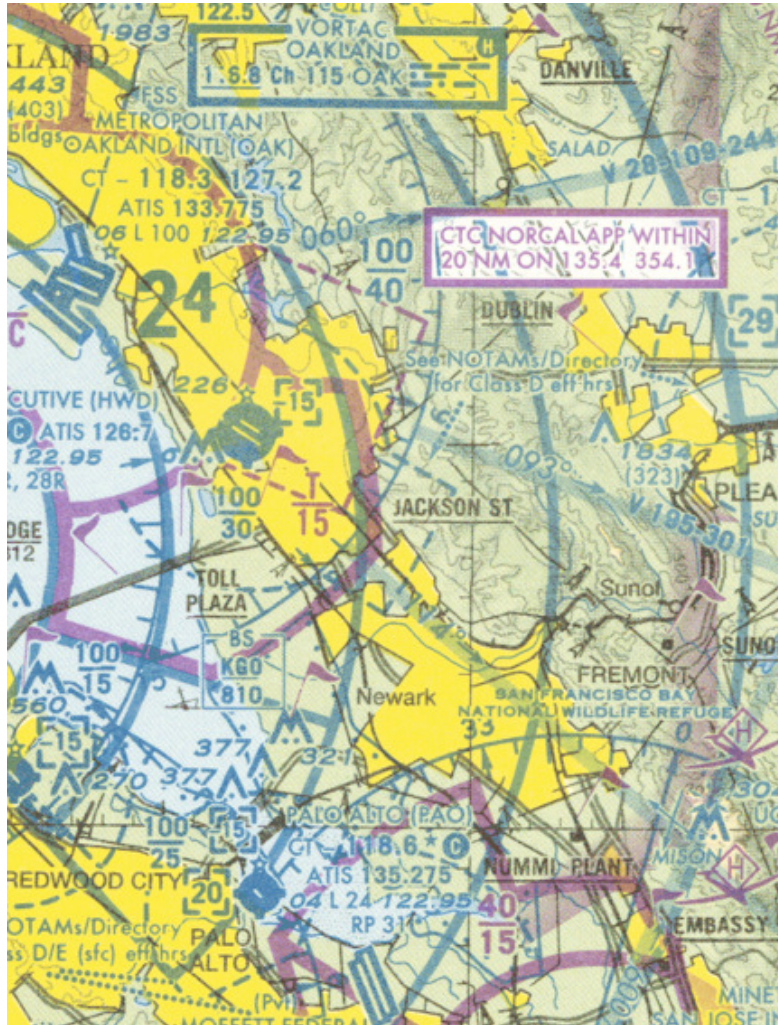


Figura 10.8: Oakland VOR and 114 radial to MISON intersection



Figura 10.9: Autopilot with altitude armed

10.4.3. 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 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 ft, displayed on the right. When you started turning it, "ALT ARM" should have appeared in the autopilot display (as in Figure 10.9). 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 ft).⁷ It will also politely beep 5 times when you cross 3000 ft to remind you that you're within 1000 ft the armed altitude.

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.

10.4.4. 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.

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 slomping 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.⁸ But if the needle is *moving* towards us, that means we're going to cross the radial, sooner or later, so our

⁷Of course, you don't really have to do this – you could just watch the altimeter, and when it gets to 4000 ft, 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.

⁸Unless you're heading in the opposite direction, but that's another story.

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.

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.

10.4.5. 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 (· -·-· ·-).

Cross OAK 114 radial

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.

NAV2 ⇒ 116.0

VOR2 OBS ⇒ 229

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 DME⁹ (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.

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.

DME ⇒ N2

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,¹⁰ 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 us with a very precise fix here because Manteca is at such an oblique angle. But it does give us a good warning of

⁹See https://en.wikipedia.org/wiki/Distance_Measuring_Equipment for more information.

¹⁰For example, from <http://www.airnav.com/airspace/fix/SUNOL>.

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.

10.5. 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 ft 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.

10.5.1. 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 10.10 (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

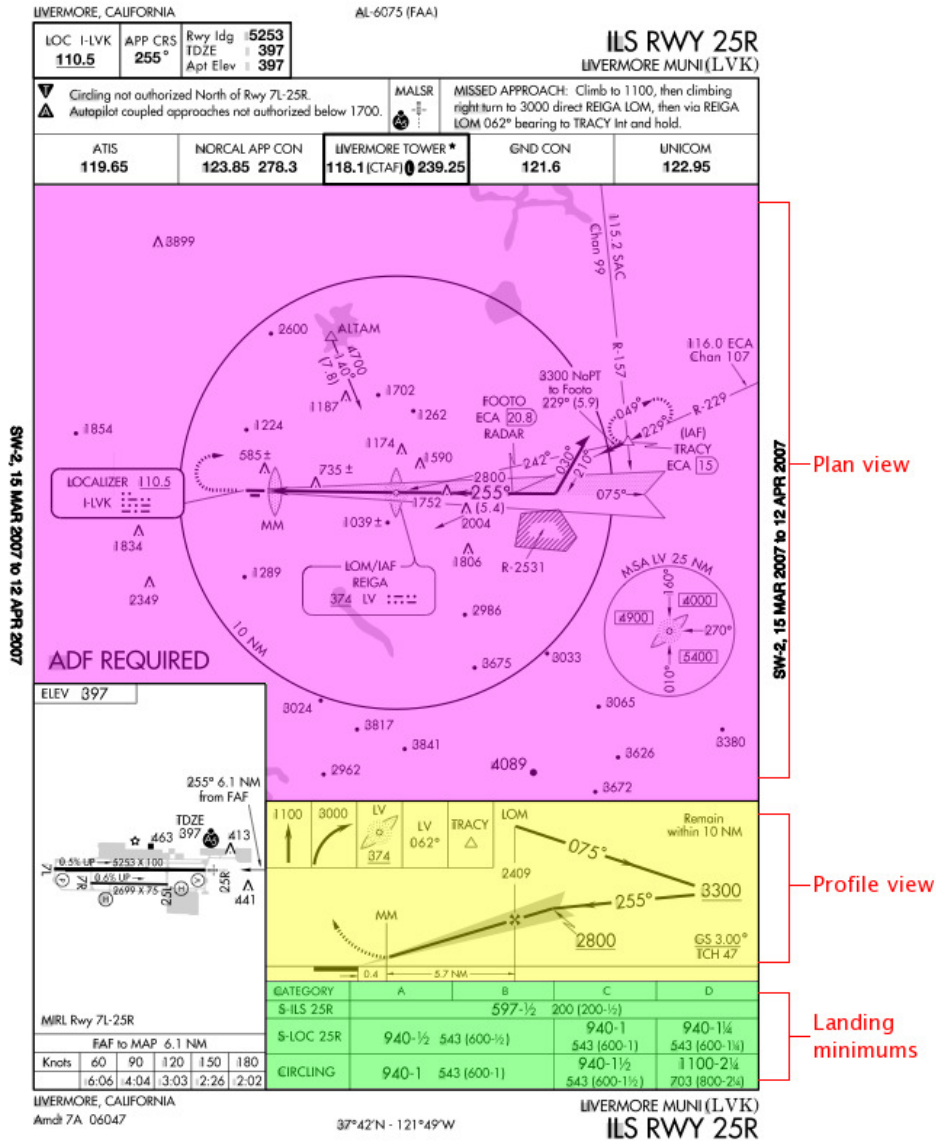


Figura 10.10: ILS approach plate for Livermore runway 25R

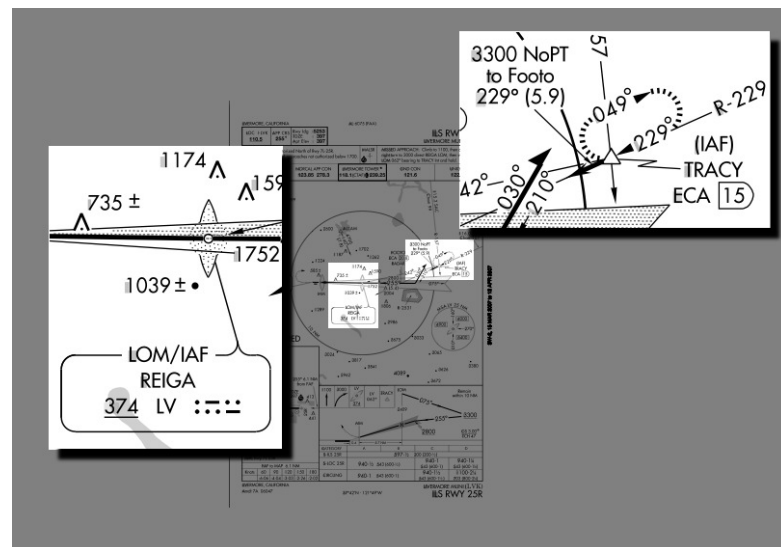


Figura 10.11: Initial approach fixes

“plan view”, which I’ve coloured purple in Figure 10.10. Our IAP lists two, one in the middle and one on the right (see Figure 10.11 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.

10.5.2. Non-directional 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 NDB¹¹ (non-directional 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 $\cdot\cdot\cdot\cdot \cdot\cdot\cdot\cdot$ 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 10.12 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.

¹¹See https://en.wikipedia.org/wiki/Non-directional_beacon for more information.



Figure 10.12: REIGA non-directional beacon

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 might not. Why? Because the receiver might be in antenna mode (as show by the “ANT” in the upper-left portion of the display).¹² If it is in antenna mode, 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, so hit the ADF switch on the audio panel and check.

ADF ⇒ 374

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.*

¹²Antenna 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.

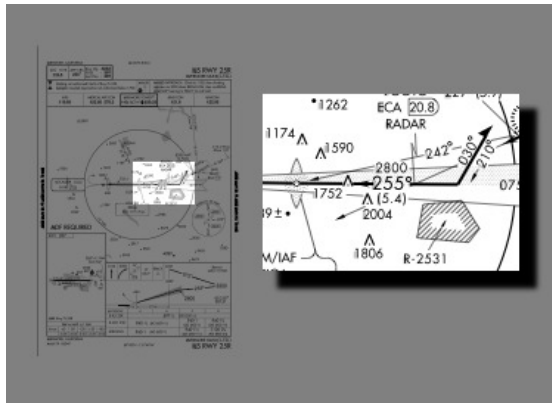


Figura 10.13: Livermore ILS procedure turn

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.¹³

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 . . .

10.5.3. 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 10.13). As you can see if you follow the arrow, we need to fly away, on a heading of 75°, then turn left 45° to a heading of 30°. 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 10.10). You can see that at the top is the LOM, our IAF. Now follow the arrows. After the IAF, we head out at 75°. During the procedure turn we can descend to 3300 ft, 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 ft, 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

¹³Which is actually bad technique in the presence of a crosswind, but I’m ignoring the wind to simplify the tutorial.



Figura 10.14: ADF with timer running

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 75°, and two minutes at 30°. On the way back we don't care about times – we just want to intercept 255°.

So, after we pass REIGA, turn right to 75°. 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 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, labelled "SET/RST". The timer should reset to 0, then start counting up (see Figure 10.14).¹⁴ 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.

10.5.4. 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 75°*.

Now, in an ideal world, after we turned to 75°, 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 75°, we'll be right on course.

Figure 10.15 shows what I mean. In the figure, the plane, flying along the green line, is initially off course.¹⁵ The heading is correct, 75°, 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 75°, and readjust the ADF compass card.¹⁶ 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

Cross REIGA;
Fly at 75° away
from REIGA for
two minutes

¹⁴The timer can also be set to count down from a time you specify – except that feature has not yet been implemented.

¹⁵Way off course, actually. I've exaggerated the angles to make the explanation clearer.

¹⁶You 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).

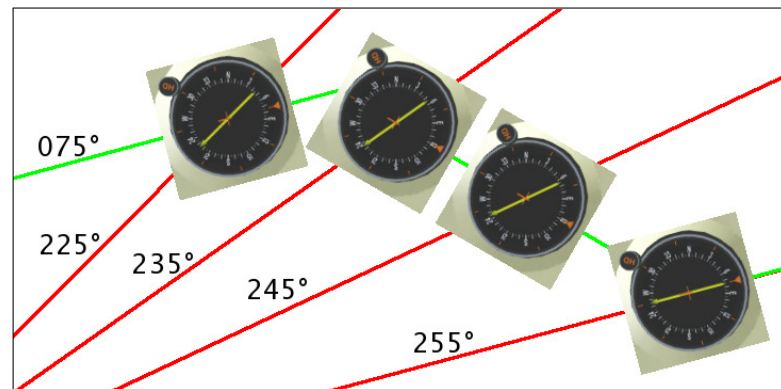


Figura 10.15: Getting back on course

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 30° as part of our procedure turn, at which point we'll just ignore the NDB anyway. Sigh. All that effort for just 2 minutes. Hardly seems worth it.

10.5.5. 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 10.10), 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 75° are up. At the end of two minutes, turn left 45° to 30°. Reset the timer, and fly for another two minutes on this heading.

10.5.6. George III

This leg is relatively uneventful, so we'll take advantage of the lull in the action to descend on left to 30°; fly for to 3300. Before descending, check the KLVK ATIS (it should be 119.65 MHz) and make sure minutes while des- your altimeter is correct. nding to 3300

Assuming you're using the autopilot, you will need to do a few things to descend:

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.

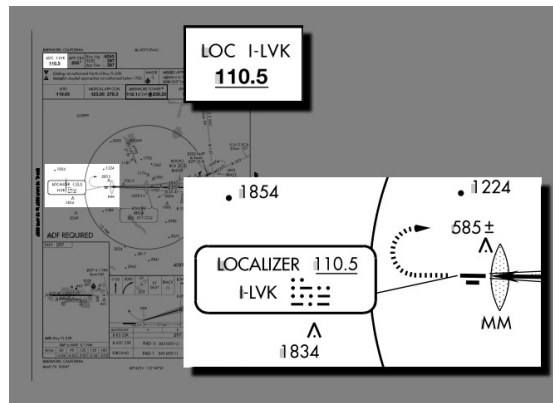


Figura 10.16: Livermore 25R localizer

10.5.8. Intercepting the Localizer

We're now ready to intercept the ILS localizer. When the two minutes on the 30° 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.

Now that you're heading back inbound at 255°, slow to 75 knots, drop a notch of flaps, and descend to 2800 ft (but no lower). And check for the inbound passage of FOOTO to confirm your position. And pat your head and rub your stomach.

10.5.9. 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.¹⁸ 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:

¹⁸Maybe. 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.

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.

Intercept glideslope;
cross outer marker;
drop second notch
of flaps

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.

10.5.10. 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?). If you want, you can lower the flaps completely as you get closer.

10.5.11. 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 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.

10.5.12. 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,¹⁹ 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 10.10). 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 ft above the runway). At 597 ft we make our decision – if we can't see the runway, then we have to execute a missed approach. 597 ft 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.²⁰

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 10.17): 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:

1. Climb straight ahead to 1100 ft
2. Make a climbing right turn to 3000 ft
3. Fly to REIGA
4. Fly outbound from REIGA at 62°
5. Fly a holding pattern at the TRACY intersection

¹⁹Well, unless it's a Category IIIC ILS approach.

²⁰As 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 "I"? Because in ancient times, it was also used to identify passage over "airway" markers along radio range tracks.

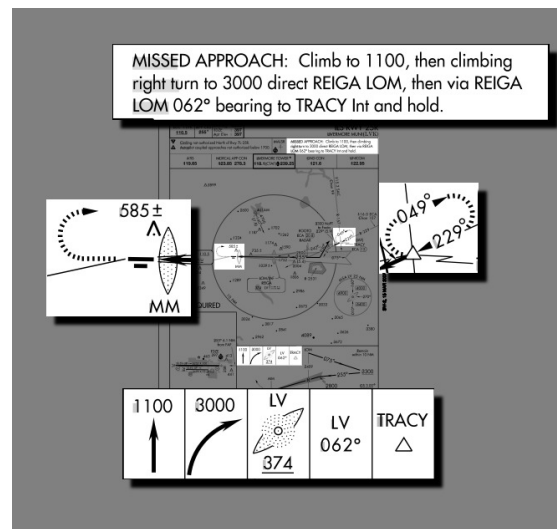


Figure 10.17: Missed approach procedure

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 . . .

10.5.13. 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 course²¹ (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 mid-marker

Land;
eat hamburger

10.6. 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, 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.

²¹Remembering, of course, to disengage the autopilot.



Figura 10.18: On course, runway in view. We're going to live!

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. More documents can be found at their [Aviation Handbooks & Manuals](#) page.
- 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 [Fergo IFR Simulator](#), where you can learn navigation by playing NAVAIDs finding missions.
- 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.
- [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 – this series is even more Microsoft Flight Simulator focused 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.

Capítulo 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: <https://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 **J**. 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.



Parte IV
Appendices

Apéndice A

Missed approach: If anything refuses to work

In the following section, we tried to sort some problems according to operating system, but if you encounter a problem, it may be a wise idea to look beyond “your” operating system – just in case. If you are experiencing problems, we would strongly advise you to first check the FAQ maintained by Cameron Moore at:

https://wiki.flightgear.org/Frequently_asked_questions.

Moreover, the source code contains a directory `docs-mini` containing numerous ideas on and solutions to special problems. This is also a good place to go for further reading.

A.1. FlightGear Problem Reports

The best place to look for help is generally the mailing lists, specifically the **[Flightgear-User]** mailing list. If you happen to be running a Git version of *FlightGear*, you may want to subscribe to the **[Flightgear-Devel]** list. Instructions for subscription can be found at:

https://wiki.flightgear.org/Mailing_lists.

It is often the case that someone has already dealt with the issue you are dealing with, so it may be worth your time to search the mailing list archives at:

https://sourceforge.net/mailarchive/forum.php?forum_name=flightgear-users
https://sourceforge.net/mailarchive/forum.php?forum_name=flightgear-devel.

You should also consider searching the FlightGear forums for help, instructions and archives at:

<https://forum.flightgear.org>.

There are numerous developers and users reading those lists and forums, so questions are generally answered. However, messages of the type *FlightGear does not compile on my system. What shall I do?* are hard to answer without any further detail given, aren't they? Here are some things to consider including in your message when you report a problem:

- **Operating system:** (Linux Fedora 17.../Windows Seven 64 bits...)
- **Computer:** (Pentium Dual Core, 2.3 GHz...)
- **Graphics board/chip:** (ATI Radeon HD 770 XT/NVidia GeForce GTX 590...)
- **Compiler/version:** (GCC version 4.6.3...)
- **Versions of relevant libraries:** (PLIB 1.8.5, OpenSceneGraph 3.0.1...)
- **Type of problem:** (Linker dies with message...)
- **Steps to recreate the problem:** Start at KSFO, turn off brakes ...

In order to analyze what happened during the last *FlightGear* session, the following command can be used (the `~/fgfs` path corresponds to `$FG_HOME` on Unix-type systems; it needs to be adapted on other systems, or you can use option `--log-dir` to make *FlightGear* write the log elsewhere):

```
less ~/fgfs/fgfs.log
```

The default log level is `info`; it corresponds to the `fgfs` option `--log-level=info`. If you pass option `--log-level=debug` to `fgfs`, *FlightGear* will write a lot more messages to its log file.

Some messages, *a priori* originating from *FlightGear*'s dependencies, aren't written to the log file. One way to capture all messages directly or indirectly coming from *FlightGear*, is to redirect its standard output and standard error streams:

```
fgfs --log-level=debug >log.txt 2>&1
```

One final remark: please avoid posting binaries to these lists or forums! List subscribers are widely distributed, and some users have low bandwidth and/or metered connections. Large messages may be rejected by the mailing list administrator. Thanks.

A.2. General problems

- *FlightGear* runs SOOO slow.
When *FlightGear* indicates that it is running with something like 1-fps (frame per second) or below you typically don't have working hardware-OpenGL support. There may be several reasons for this. First, there may be no OpenGL hardware drivers available for older cards. In this case it is highly recommended to get a new board.

Second, check if your drivers are properly installed. Several cards need additional OpenGL support drivers besides the "native" windows ones.

- Either the `configure` or the `make` script display errors due to missing *PLIB* headers or libraries. Ensure you have the latest version of *PLIB* compiled and installed. The headers, like `puAux.h`, have to be within a `plib` subfolder (e.g. `/usr/include/plib`) and its libraries, like `libplibpu.a`, are typically at `/lib`. Double-check that there is not a second set of *PLIB* headers and libraries sitting elsewhere! Check carefully the error messages of `configure`. In several cases, it specifies what is missing.

A.3. Potential problems under Linux

Since we don't have access to all possible flavors of Linux distributions, here are some thoughts on possible causes of problems. (This Section includes contributions by Kai Troester.)

- Wrong library versions
This is a rather common cause of grief especially when you prefer to install the libraries needed by *FlightGear* by hand. Be sure that especially the Mesa library contains support for the 3DFX board and that GLIDE libraries are installed and can be found. If a `ldd` which `fgfs`` complains about missing libraries you are in trouble.

You should also be sure to *always* keep the *latest* version of *PLIB* on your system. Lots of people have failed miserably to compile *FlightGear* just because of an outdated `plib`.

- Missing permissions
In case you are using XFree86 before release 4.0 the *FlightGear* binary may need to be setuid root in order to be capable of accessing some accelerator boards (or a special kernel module as described earlier in this document) based on 3DFX chips. So you can either issue a

```
chown root.root /usr/local/bin/fgfs ;
chmod 4755 /usr/local/bin/fgfs
```

to give the *FlightGear* binary the proper rights or install the 3DFX module. The latter is the “clean” solution and strongly recommended!

- Non-default install options
FlightGear will display a lot of diagnostics while starting up. If it complains about bad looking or missing files, check that you installed them in the way they are supposed to be installed (i.e. with the latest version and in the proper location). The canonical location *FlightGear* wants its data files under `/usr/local/lib`. Be sure to grab the latest versions of everything that might be needed!
- Compile problems in general
Make sure you have the latest (official) version of `gcc`. Old versions of `gcc` are a frequent source of trouble! On the other hand, some versions of the RedHat 7.0 reportedly have certain problems compiling *FlightGear* as they include a preliminary version of `gcc`.

A.4. Potential problems under Windows

- The executable refuses to run.

You may have tried to start the executable directly either by double-clicking `fgfs.exe` in Windows Explorer or by invoking it within a MS-DOS shell. Double-clicking via Explorer never works (unless you set the environment variable `$FG_ROOT` in `autoexec.bat` or otherwise). Rather double-click `fgrun`. For more details, check Chapter 4.

Another cause of grief might be that you did not download the most recent versions of the base package files required by *FlightGear*, or you did not download any of them at all. Have a close look at this, as the scenery/texture format is still under development and may change frequently. For more details, check Chapter 3.

Next, if you run into trouble at runtime, do not use Windows utilities for unpacking the `.tar.gz`. If you did, try it in the Cygnus shell with `tar xvfz` instead.

- *FlightGear* ignores the command line parameters.

There can be a problem with passing command line options containing a "=" on the command line. Instead create a batch job to include your options and run that instead.

- I am unable to build *FlightGear* under MSVC/MS DevStudio.

By default, *FlightGear* is build with GNU GCC. The Win32 port of GNU GCC is known as Cygwin. For hints on *Makefiles* required for MSVC or MSC DevStudio have a look into:

https://wiki.flightgear.org/Building_using_CMake_-_Windows.

In principle, it should be possible to compile *FlightGear* with the project files provided with the source code.

- Compilation of *FlightGear* dies.

There may be several reasons for this, including true bugs. However, before trying to do anything else or report a problem, make sure you have the latest version of the *Cygwin* compiler. In case of doubt, start `setup.exe` anew and download and install the most recent versions of bundles as they possibly may have changed.

Apéndice B

Landing: Some further thoughts before leaving the plane

B.1. A Sketch on the History of *FlightGear*

History may be a boring subject. However, from time to time there are people asking for the history of *FlightGear*. As a result, we'll give a short outline.

The *FlightGear* project goes back to a discussion among a group of net citizens in 1996 resulting in a proposal written by David Murr who, unfortunately, dropped out of the project (as well as the net) later. The original proposal is still available and can be found under:

https://groups.google.com/g/rec.aviation.simulators/c/ny8HFBE5_T8/m/OdtliGNGJc8J.

Although the names of the people and several of the details have changed over time, the spirit of that proposal has clearly been retained up to the present time.

Actual coding started in the summer of 1996 and by the end of that year essential graphics routines were completed. At that time, programming was mainly performed and coordinated by Eric Korpela from Berkeley University. Early code ran under Linux as well as under DOS, OS/2, Windows 95/NT, and Sun-OS. This was found to be quite an ambitious project as it involved, among other things, writing all the graphics routines in a system-independent way entirely from scratch.

Development slowed and finally stopped in the beginning of 1997 when Eric was completing his thesis. At this point, the project seemed to be dead and traffic on the mailing list went down to nearly nothing.

It was Curt Olson from the University of Minnesota who re-launched the project in the middle of 1997. His idea was as simple as it was powerful: Why invent the wheel a second time? There have been several free flight simulators available running on workstations under different flavors of UNIX. One of these, LaRCsim (developed by Bruce Jackson from NASA), seemed to be well suited to the approach. Curt took this one apart and re-wrote several of the routines such as to make them build as well as run on the intended target platforms. The key idea in doing so was to exploit a system-independent graphics platform: OpenGL.

In addition, a clever decision on the selection of the basic scenery data was made in the

very first version. *FlightGear* scenery is created based on satellite data published by the U. S. Geological Survey. These terrain data are available from:

<https://www.usgs.gov/core-science-systems/ngp/tnm-delivery/gis-data-download>

for the U.S., and

<https://www.usgs.gov/centers/eros/science/usgs-eros-archive-digital-elevation-global-30-arc-second-elevati>

resp., for other countries. Those freely accessible scenery data, in conjunction with scenery building tools included with *FlightGear*, are an important feature enabling anyone to create his or her own scenery.

This new *FlightGear* code – still largely being based on the original LaRCsim code – was released in July 1997. From that moment the project gained momentum again. Here are some milestones in the more recent development history.

B.1.1. Scenery

- Texture support was added by Curt Olson in spring 1998. This marked a significant improvement in terms of reality. Some high-quality textures were submitted by Eric Mitchell for the *FlightGear* project. Another set of high-quality textures was added by Erik Hofman ever since.
- After improving the scenery and texture support frame rate dropped down to a point where *FlightGear* became unflyable in spring 1998. This issue was resolved by exploiting hardware OpenGL support, which became available at that time, and implementing view frustum culling (a rendering technique that ignores the part of the scenery not visible in a scene), done by Curt Olson. With respect to frame rate one should keep in mind that the code, at present, is in no way optimized, which leaves room for further improvements.
- In September 1998 Curt Olson succeeded in creating a complete terrain model for the U.S. The scenery is available worldwide now, via a clickable map at:

<https://www.flightgear.org/download/scenery>.

- Scenery was further improved by adding geographic features including lakes, rivers, and coastlines later. 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 it's 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.
- 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.
- Today, the effort is still going on, with the use of *TerraSync*, the tool for on-the-fly scenery download, the powerful mapserver and scenemodels infrastructure laying behind, as well as webforms enabling to quickly add or update objects. Scenery generation tools have been updated in order to use more accurate data such as the 8.50 apt.dat format, as well as external data such as OpenStreetMap, when the licence is adequate.

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 – a move 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 navi-

gation 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 its way into *FlightGear*. *JSBSim* 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` 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* Wiki contains a detailed history worth reading of all of the notable development milestones at

https://wiki.flightgear.org/Category:FlightGear_changelogs

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

https://wiki.flightgear.org/Mailing_lists.

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.

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://openuniverse.sourceforge.net>

Michele America

Contributed to the HUD code.

Michael Basler

Author of Installation and Getting Started. Flight Simulation Page at:

<https://www.flusi.info>

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, wrote the initial tutorial subsystem, developed random vegetation and buildings.

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

<https://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:

<https://www.usgs.gov/centers/eros/science/usgs-eros-archive-digital-elevation-global-30-arc-second-elevation-gtop>

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.

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)

https://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.

<https://www3.cs.stonybrook.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, <https://zlib.net>.

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 and daylight variables.

Michael I. Gold

Patiently answered questions on OpenGL.

Habibe

Made RedHat package building changes for SimGear.

Mike Hill

For allowing us to concert and use his wonderful planes 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:

["Flight Gear Autopilot: Altitude Hold Module" on archive.org¹.](http://web.archive.org/web/20030803143303/http://www.menet.umn.edu:80/~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.

¹<https://web.archive.org/web/20030803143303/http://www.menet.umn.edu:80/~curt/fgfs/Docs/Autopilot/AltitudeHold/AltitudeHold.html>

Alan Murta

Created the Generic Polygon Clipping library.

https://en.wikipedia.org/wiki/General_Polygon_Clipper

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.

Brian Paul

We made use of his TR library and of course of Mesa:

<https://www.ssec.wisc.edu/~billh/bp/TR.html>, <https://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.

<https://stjarnhimlen.se/english.php>

Chris Schoeneman

Contributed ideas on audio support.

Phil Schubert

Contributed various textures and engine modeling.

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://www.durktalsma.nl>.

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.

<https://www.usgs.gov/core-science-systems/ngp/tnm-delivery/gis-data-download>

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.

Norman Vine

Provided 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.

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:

<https://edwilliams.org>

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:

<https://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 upon a time, they provided computing resources and services so that the *FlightGear* project could have a real home.

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 supports 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 of the contributors mentioned above for 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.

I would like to express a special thanks to Curt Olson, whose numerous scattered READ.MEs, 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 at 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.

Apéndice C

Main Index

Symbols

.fgfsrc, [32](#), [191](#)

Numbers

2D cockpit, [48](#)

3D cockpit, [48](#)

3D panels, [190](#)

3DFX, [185](#)

A

A1 Free Sounds, [192](#)

abort, [112](#)

Adams, Syd, [192](#)

additional scenery, [21](#)

ADF, [107](#)

Adler, Mark, [194](#)

Aeronautical Information Manual, [74](#)

aeronave

- seleccion, [34](#)

AI, [54](#)

aileron, [46](#), [106](#)

aileron indicator, [56](#)

Air Traffic Control, [142](#)

air traffic facilities, [107](#)

Air-Air Refuelling, [68](#)

Aircraft Carrier, [61](#)

aircraft

- installation, [24](#)

airport, [199](#)

airspeed indicator, [105](#)

Airwave Xtreme 150, [19](#)

ala delta, [19](#)

Alonzo, Raul, [192](#)

altimeter, [87](#), [106](#)

Altimeter, [136](#)

altimeter

- tuning, [87](#)

altitude

- absolute, [87](#)

ambiente, [53](#)

America, Michele, [189](#), [192](#)

Arrancar el simulador, [27](#)

Arrancar el simulador en Windows, [30](#)

Arrancar el simulador

- Linux, [31](#)

- Windows, [30](#)

Arrancar Flightgear

- Mac OS X, [32](#)

artificial horizon, [105](#)

astronomy code, [190](#)

atajos

- configuracion, [46](#)

ATC, [190](#), [199](#)

ATIS, [133](#)

ATIS messages, [190](#)

Atlas, [63](#), [191](#)

attitude indicator, [105](#)

audio library, [195](#)

audio support, [190](#)

auto coordination, [106](#)

autopilot, [48](#), [53](#), [116](#), [137](#), [189](#), [194f](#)

autopilot controls, [49](#)

autopilot

- modes, [116](#)

- heading mode, [116](#)

- roll control mode, [116](#)

- vertical speed mode, [116](#)

autothrottle, [49](#)

avión de los Hermanos Wright, [19](#)

Avionics, [40](#)

B

Baker, Steve, [190](#)

- bank, 105
- Basler, Michael, 192
- Bendix transponder, 190
- Berndt, Jon, 200
- Berndt, Jon, S., 189, 191f, 200
- binaries
 - pre-compiled, 10
- binary distribution, 9
- Bleisch, Paul, 192
- brakes, 47, 90, 196
 - left wheel [,] (colon), 90
 - right wheel [.] (dot), 90
- Brennan, Jim, 192
- Bright, Bernie, 192
- BSD UNIX, 16
- Buchanan, Stuart, 192
- Buckel, Bernhard, 192, 199
- Buckle, Gene, 193

- C**
- campo de visión, 39
- Carmichael, Ralph, 193
- Cessna, 197
- Cessna 172, 189
- Chauveau, Didier, 193
- Check, John, 190f, 193, 200
- Cherokee flight model, 197
- clock, 107
- clouds, 190, 197
- cockpit, 48
- código fuente, 17
- COM transceiver, 107
- Comenzar Flightgear
 - Mac OS X, 32
- Comenzar simulador
 - Linux, 31
- COMM1, 107
- COMM2, 107
- command line options
 - index of, *véase* [Index of Command Line Options](#)
- CommonMistakes, 86
- communication radio, 107
- Compensador, 46
- compilador, 18
- contributors, 191
- control surface, movable, 190
- controles de los motores, 47
- controles de teclado, 46
 - primarios, 46
- Controles del simulador, 47
- coordinación automática, 33
- Cornish, Dave, 188, 193

- Cygnus, 18, 199
- Cygwin, 18, 186

- D**
- debug, 55
- Delise, Oliver, 190f, 193, 200
- Denker, John, 75
- differential braking, 47
- Direct3D, 18
- DirectX, 18
- display options, 48
- documentación, 17
- documentation
 - installation, 24
- DOS, 187
- Doue, Jean-Francois, 193

- E**
- Eberly, Dave, 193
- efectos de sonido, 18
- el servidor Telnet, 41
- elevation indicator, 56
- elevator, 46
- engine
 - shut down, 98
- equipment, 54
- Evans, Francine, 193
- Everitt, Oscar, 193
- exit, 51

- F**
- FAA, 74
- FAA Training Book, 74
- FAQ, 183
- FDM, 189, 192
 - external, 19
 - pipe, 19
- Festival, 65
- FG_ROOT, 30
- FG_SCENERY, 23, 30
- Finney, Bruce, 193
- flaps, 47, 101, 106
 - control lever, 101
 - steps, 101
- Flebbe, Olaf, 193
- flight dynamics model, 189
- flight model, 19, 189
- flight planner, 190
- Flight simulator
 - civilian, 16
 - free, 187
 - multi-platform, 16
 - open, 16
 - user-extensible, 16f

- user-supported, 16f
- FlightGear, 191
- FlightGear documentation, 20
- FlightGear Flight School, 20
- FlightGear forums, 183
- FlightGear Programmer's Guide, 20
- FlightGear Scenery Design Guide, 20
- FlightGear
 - versions, 18
- FlightGear Website, 19
- FlightGear Wiki, 191
- Foad, Julian, 200
- fog effects, 195
- forum, 11
- frame rate, 18, 188
- Franz, Melchior, 194, 200
- FreeBSD, 194
- FS98, 193
- fuel indicator, 107

G

- Gailly, Jean-loup, 194
- GameGLUT, 198
- Garg, Mohit, 194, 200
- gear, 47
- Gellekum, Thomas, 194
- generic autopilot, 48
- geographic features, 188
- Girish, Neetha, 194
- GLIDE, 185
- GNU C++, 18
- GNU General Public License, 16
- Go Around, 146
- Goeke-Smith, Jeff, 189, 194
- Gold, Michael, I., 194
- GPL, 16f
- graphics routines, 187
- GSHHS data, 188
- gyro compass, 106

H

- Habibe, 194
- head-up display, 55, 189
- heading bug, 136
- heading hold, 49
- help, 55
- Hill, Mike, 194
- History, 187
- history
 - aircraft, 189
 - environment, 190
 - scenery, 188
 - user interface, 190
- Hofman, Erik, 188, 194, 200

- hot air balloon, 195
- Hotchkiss, Charlie, 189, 194
- HTTP server, 191
- HUD (Head-Up Display
 - full mode), 120
- HUD (Head-Up Display), 120ff
- HUD, 40, 55f, 189, 192, 194

I

- icing
 - modelling, 19
- IFR, 74
- ignition switch, 107
- inclinometer, 106
- installing aircraft, 24
- instrument panel, 48, 55, 189
- Internet, 190

J

- Jackson, Bruce, 187, 194
- joystick, 44, 190
- joystick settings, 191
- joystick/self detection, 190
- joysticks, 18
- Justus, Maik, 194

K

- Kaaven, Ove, 194
- Kaszeta, Richard, 194
- keyboard, 81
- keyboard controls
 - display, 48
 - general, 48
 - view, 48
 - aircraft, 47
 - autopilot, 48
 - engine, 47
- keyboard
 - numeric, 81
 - uppercase and lowercase keys, 81
- keyboard.xml, 46
- Knienieder, Tom, 195
- Koradi, Reto, 195
- Korpela, Eric, 187
- Kuehne, Bob, 195

L

- Laird, Kyler B., 195, 200
- landing, 109
 - aid signals, 87
- landing gear, 47
- LaRCsim, 187ff, 194, 196f
- latitude, 36
- leaflet, 11

- light textures, 188
- Linux, 16ff, 187
- Livermore, 130
- location, 53
- longitude, 36
- Luff, David, 190, 195
- M**
- magnetic compass, 106
- magneto, 96
- magneto switch, 190
- mailing lists, 183, 191
- map, clickable, 188
- Mayer, Christian, 190f, 195
- Megginson, David, 74, 188–191, 195, 200
- menu, 190
- menu entries, 50
- menu system, 199
- MetaKit, 198
- Metro Works, 198
- Microsoft, 15
- Mitchell, Eric, 188, 195
- mixture, 97, 140
 - lever, 98
 - optimisation, 98
- modelo aeronáutico de la aeronave, 35
- modelo de dinámica de vuelo, 35
- modelo de vuelo de UIUC, 19
- modelos de dinámica de vuelo, 19
- modelos de vuelo, 19
- Modelos de vuelo, 35
- Moore Cameron, 183
- Moore, Cameron, 195, 200
- Morken, Anders, 195
- moteur
 - avviare, 46
 - jet, 46
- motor
 - encendido, 45
 - piston, 45
 - turbopropulsores, 46
- mouse, 49
- mouse modes, 49
- mouse
 - normal mode, 85
 - rudder control, 91
 - view mode, 85
 - yoke mode, 85
- mouse, actions, 49
- mouse, controllo, 50
- mouse, normale, 49
- mouse, view, 50
- MS DevStudio, 186
- MSVC, 186, 193
- multi-engine support, 190
- multi-lingual conversion tools, 195
- multiplayer, 54
- Multiplayer, 59
- multiplayer code, 190
- Multiple Computer, 63
- Multiple Displays, 63
- Murr, David, 187
- Murta, Alan, 196
- N**
- NAV, 107
- Navion, 189
- NDB, 107
- Nelson, Phil, 196
- network, 190
- network options, 41
- networking code, 190, 193
- niebla, 39
- Novikov, Alexei, 196
- NumLock, 46
- NVIDIA, 10
- O**
- offset, 40
- Olson, Curt, 187–191, 196, 199
- opciones
 - aeronave, 34
 - características, 34
- opciones de línea de comandos, 32
- opciones de línea de comandos, 32
- opciones
 - entorno, 37
 - generales, 32
 - modelos de vuelo, 35
 - orientación, 35
 - posición inicial, 35
 - sonido, 34
- OpenGL, 10, 18, 20, 184, 187f, 191, 194
 - drivers, 18
- options
 - Aircraft Systems, 40
 - debugging, 43
 - HUD, 40
 - IO, 42
 - network, 41
 - rendering, 38
 - route, 42
 - time, 41
 - waypoint, 42
- orientación, 36
- OS/2, 187
- Others, 199

P

panel, [55](#), [195f](#)
 panel de instrumentos, [38](#)
 parking brake, [47](#)
 Paul, Brian, [196](#)
 pausa, [48](#)
 pedales, [18](#)
 Peden, Tony, [191](#), [196](#)
 Peel, Robin, [189](#), [196](#)
 permissions, [185](#)
 Perry, Alex, [196](#), [200](#)
 Perry, Dave, [200](#)
 pitch, [105](#)
 pitch indicator, [56](#)
 Playback, [65](#)
 PLIB, [190f](#)
 problem report, [183](#)
 problems, [183](#)

- general, [184](#)
- Linux, [185](#)
- Windows, [186](#)

 programmers, [191](#)
 property manager, [191](#)
 proposal, [187](#)
 puntero de ratón, [38](#)

R

radio, [133](#)
 radio stack, [107](#), [189](#)
 random ground objects, [189](#)
 Reid-Hillview, [130](#)
 Reinhard, Friedemann, [189](#), [196](#)
 Reinholdtsen, Petter, [196](#)
 requisitos del sistema, [17](#)
 reset flight, [50](#)
 Riley, William, [188](#), [196](#)
 Ross, Andy, [189](#), [196](#), [200](#)
 RPM indicator, [106](#)
 rudder, [46](#), [106](#)
 rudder indicator, [56](#)
 rudder

- keyboard control, [91](#)
- mouse control, [91](#)

 runway lighting, [188](#)

S

scenery, [21](#), [187f](#)

- additional, [21](#)

 Scenery Database, [199](#)
 scenery directory

- path, [33](#)

 scenery subsystem, [196](#)
 Schlyter, Paul, [197](#)
 Schoenemann, Chris, [197](#)

Schubert, Phil, [197](#)
 screenshot, [51](#)
 Sectional, [130](#)
 See how it flies, [75](#)
 Selig, Michael, [189](#), [200](#)
 servidor HTTP, [41](#)
 Shewchuk, Jonathan, [197](#)
 shutdown, [111](#)
 side-slip, [102](#)
 Sikic, Gordan, [197](#)
 SimGear, [191](#)
 Sistemas Operativos, [16](#)
 Smith, Michael, [197](#)
 speed, [56](#)
 speed brakes, [47](#)
 speed

- units, [92](#)
 - knot (nautical mile per hour), [92](#)

 spin, [102](#)
 spoilers, [47](#)
 Spott, Martin, [197](#), [199](#)
 stall, [102](#)
 starter, [107](#)
 starting the engine, [107](#)
 Startup latitude, [36](#)
 Startup longitude, [36](#)
 static objects, [189](#)
 Stockill, Jon, [197](#)
 Sun-OS, [187](#)
 system requirements, [17](#)
 system.fgfsrc, [32](#)

T

tachometer, [90](#)
 tail-wheel lock, [47](#)
 Talsma, Durk, [190](#), [197](#)
 tarjeta de sonido, [18](#)
 tarjeta gráfica, [18](#)
 teclado, [46](#)
 TerraGear, [191](#)
 Text To Speech, [65](#)
 texture, [188](#)
 textures, [188](#), [195](#)
 throttle, [46](#), [56](#)
 throttle lever, [97](#)
 thunderstorms, [190](#)
 time options, [41](#)
 Torvalds, Linus, [17](#)
 Traffic Pattern, [143](#)
 triangle program, [197](#)
 trim, [103](#)
 Troester, Kai, [185](#), [200](#)
 troubles

- mouse speed, [86](#)
- turn coordinator, [89](#)
- turn indicator, [56](#), [106](#)
- tutorial, [74](#)

U

- U. S. Geological Survey, [188](#), [198](#)
- UIUC, [189](#), [197](#)
- UNIX, [187](#)

V

- Vallevand, Mark, [198](#)
- van der Mac, Sam, [195](#)
- van Sickle, Gary, R., [191](#), [198](#)
- variables por defecto, [32](#)
- VASI, [145](#)
- velocity rages, [106](#)
- vertical speed indicator, [106](#)
- VFR, [74](#), [107](#)
- view, [51](#)
 - changing, [83](#)
- view directions, [47](#)
- view frustrum culling, [188](#)
- view
 - instant replay, [83](#)
- view modes, [48](#)
- views, [191](#)
- Vine, Norman, [189](#), [191](#), [198](#), [200](#)
- visual flight rules, [107](#)
- VMap0 data, [188](#)
- Voegtli, Roland, [198](#)
- Volpe, Carmelo, [198](#)

- VOR, [107](#)

W

- Walisser, Darrell, [198](#), [200](#)
- weather, [195](#)
- wiki, [11](#)
- Williams, Ed, [198](#)
- Wilson, Jim, [198](#)
- window size, [39](#)
- Windows, [18](#)
- Windows 10/8/7/Vista, [16](#)
- Windows 95/NT, [187](#)
- winds, [190](#)
- windsock, [114](#)
- Wippler, Jean-Claude, [198](#)
- wireframe, [39](#)
- Wood, Charles, [74](#)
- Woodsoup, [198](#)
- workstation, [187](#)

X

- XFree86, [185](#)

Y

- YASim, [19](#)
- yoke, [84](#)
 - mouse yoke mode, [85](#), [87](#)
 - pulling, [86](#)

Z

- Zeh, Allan, [199](#)
- zlib library, [194](#)

Apéndice D

Index of Command Line Options

Page numbers in italics indicate the reference page for the option.

A

--adf, 40
--aero, 35
--ai-scenario, 34
--aircraft, 34f, 44, 120, 125
--aircraft-dir, 34
--airport, 35f
--altitude, 35ff
--aspect-ratio-multiplier, 38
--atcsim, 42
--atlas, 42
--AV400, 42
--AV400Sim, 42

B

--bpp, 38
--browser-app, 33

C

--callsign, 41
--carrier, 36
--ceiling, 37
--com1, 40
--com2, 40
--config, 33, 33

D

--disable-ai-models, 34
--disable-anti-alias-hud, 40
--disable-auto-coordination, 33
--disable-clock-freeze, 41
--disable-clouds, 38
--disable-clouds3d, 38
--disable-distance-attenuation, 38
--disable-enhanced-lighting, 38

--disable-freeze, 33
--disable-fuel-freeze, 35
--disable-fullscreen, 38
--disable-horizon-effect, 38
--disable-hud, 40
--disable-hud-3d, 40
--disable-mouse-pointer, 38
--disable-panel, 38
--disable-random-building, 38
--disable-random-objects, 38
--disable-random-vegetation, 39
--disable-real-weather-fetch, 37
--disable-rembrandt, 39
--disable-save-on-exit, 33
--disable-sound, 34
--disable-specular-highlight, 39
--disable-splash-screen, 39
--disable-wireframe, 39
--dme, 40

E

--enable-ai-models, 34
--enable-anti-alias-hud, 40
--enable-auto-coordination, 33, 50, 106
--enable-clock-freeze, 41
--enable-clouds, 38
--enable-clouds3d, 38
--enable-distance-attenuation, 38
--enable-enhanced-lighting, 38
--enable-fpe, 43
--enable-freeze, 33
--enable-fuel-freeze, 35
--enable-fullscreen, 38

--enable-horizon-effect, 38
--enable-hud, 40
--enable-hud-3d, 40
--enable-mouse-pointer, 38
--enable-panel, 38
--enable-random-buildings, 38
--enable-random-objects, 38
--enable-random-vegetation, 39
--enable-real-weather-fetch, 37
--enable-rembrandt, 39
--enable-save-on-exit, 33
--enable-sound, 34
--enable-specular-highlight, 39
--enable-splash-screen, 39
--enable-wireframe, 39

F

--failure, 40
--fdm, 35
--fg-aircraft, 33
--fg-root, 32
--fg-scenery, 33
--fgviewer, 43
--fix, 36
--flight-plan, 42
--fog-disable, 39
--fog-fastest, 39
--fog-nicest, 39
--fov, 39

G

--garmin, 42
--generic, 42
--geometry, 39
--glideslope, 37

H

--heading, 36
 --help, 32, 32
 --httpd, 41, 130
 --hud-culled, 40
 --hud-tris, 40

I

--in-air, 35f

J

--joyclient, 42
 --jpg-httpd, 42
 --jsclient, 43

L

--language, 27, 33
 --lat, 36
 --launcher, 27, 27, 32
 --livery, 35
 --log-dir, 184
 --log-level, 44, 184
 --lon, 36

M

--mach, 37
 --materials-file, 39
 --metar, 37
 --min-status, 34
 --model-hz, 35, 124
 --multiplay, 41

N

--native, 43
 --native-ctrls, 43
 --native-fdm, 43
 --native-gui, 43
 --nav1, 40

--nav2, 40
 --ndb, 36
 --nmea, 43
 --notrim, 35

O

--offset-azimuth, 36
 --offset-distance, 36
 --on-ground, 35
 --opengc, 43

P

--parking-id, 35
 --parkpos, 36
 --pitch, 36
 --prop, 44, 44
 --props, 43
 --proxy, 42
 --pve, 43

R

--random-wind, 37
 --ray, 43
 --restore-defaults, 33
 --roc, 37
 --roll, 36
 --rul, 43
 --runway, 36

S

--season, 37
 --shading-flat, 39
 --shading-smooth, 39
 --show-aircraft, 34
 --show-sound-devices, 34
 --sound-device, 34
 --speed, 35
 --start-date-gmt, 41

--start-date-lat, 41
 --start-date-sys, 41

T

--telnet, 41
 --texture-filtering, 39
 --time-match-local, 41
 --time-match-real, 41
 --time-offset, 41
 --timeofday, 41
 --trace-read, 44
 --trace-write, 44
 --trim, 35
 --turbulence, 37

U

--uBody, 36
 --units-feet, 33
 --units-meters, 33, 36

V

--vBody, 36
 --vc, 35ff
 --vDown, 36
 --vEast, 36
 --vehicle, 34
 --verbose, 32
 --version, 32
 --view-offset, 40
 --visibility, 37
 --visibility-miles, 37
 --vNorth, 36
 --vor, 36

W

--wBody, 36
 --wind, 37
 --wp, 42