aircraft.dll

A generic aircraft addon module for ORBITER.

Written by Russell Smith (MaverickTech)

ORBITER is written and maintained by Martin Schweiger http://www.orbitersim.com/

Table of Contents

Introduction

Aircraft DLL is a generic addon for developing and flying aircraft within Orbiter.

This addon work in a similar way to Vinka's fantastic Spacecraft DLL. I have borrowed from Vinka's ini and config design to implement aircraft in a manner that will be familiar to existing spacecraft.dll developers. While much of the basic structure remains of the spacecraft ini files, there have been many aircraft specific functions added.

Make no mistake, this addon does not seek to replace functionality spacecraft.dll. In fact some orbiter functionality is deliberately missing, because it does not relate directly to aircraft.

Aircraft DLL Features :

- Simulation of air breathing engines : you cannot fly in a vacuum.
- Afterburners
- Animated control surfaces.
- Aerodynamic surfaces, both wings and flight controls.
- User definable drag elements, for landing gear, air brakes, and more...
- Atmospheric auto pilot, with the ability to load simple flight plans.
- Ground steering.
- Wheel brakes

System Requirements :

This addon has been developed and tested on a PIII-900 system with an 8MB Rage Mobility graphics card. It seems to fly OK.

Unless you have a high spec machine, do not try time acceleration. You may get away with 10X, but after that you are in your own. It appears that atmospheric flight places some burden on Orbiter.

Acknowledgments...

Special Thanks To :

Martin Schweiger for a way cool simulations and a cooler API ! The source code for the Delta Glider formed the basis of this addon during the early development. While much of it is unrecognisable now, the sample code in the SDK is such a great basis to learn to code vessels and MFD's ☺

Kev33 for cool models, patience while this all took shape over 4 months. Kev provided numerous test vehicles, animated and otherwise, as well as a huge amount of testing and suggestions that got the project up to the beta stage.

David "Hendo" Henderson for testing, code and assistance above and beyond. Especially for a huge amount of help sorting realistic lift routines, and helping fine tune the code to allow the use of near-real numbers for the aircraft parameters

Dave Rowbotham from whom I ripped much code and assistance. Davers' open source CCCP addons were a great source of inspiration. His Collision Detection Engine is also included in this addon.

Hendo, Sputnik & Simonpro for work on the flight dynamics of the T-38, to produce a model, that from all accounts, flies close to the real thing !

Vinka for being cool about me developing a generic addon using his structure, and for sending me his spacecraft documentation that forms part of this document

Aladin Kinach & David Hopkins who contributed code on the forums that I modified for use the Ground Steering routine.

Thanks to the following guys for all their help, with beta testing, flying, and uncovering numerous ~~bugs~~ features ☺

Burncycle
C3PO
Enricod
John "MrBatman" Wilson
MattW
VoyagerVI

Finally… A big thanks to Vash and all the gang on the Orbiter IRC server. This is where the action is for me ! Without the whole zoo of loonies there, this addon would simply not be…

Aircraft DLL Flight Model

A significant amount of time (nearly 4 months) has been put in to making this addon as configurable and (hopefully) as realistic as possible.

All of the aircraft aerodynamics have been implemented using the atmospheric improvements added to Orbiter in 031105 release of Orbiter.

All of the flight surfaces are modelled with the use of Airfoils, ControlSurfaces, and variable drag elements have been added as a component of the animations to simulate real drag effects from landing gear, spoilers and speed brakes for example.

Aircraft.dll is fully atmospheric - there is no RCS

Engine Models

```
[AIRCRAFT]
FLIGHT_MODEL=1 ; Support for multiple flight models
```

Model 1 - Engine performance is modified based on Atmospheric params (currently density)
Model 2 - Uses Logarithmic scale and ceiling to limit altitude

```
EFFICIENCY=0.8 ; Engine limiting factor
```

Scaling factor applied to current thrust performance to further limit altitude

```
CEILING=50000  ; Set aircraft ceiling
```

Aircraft ceiling for Model 2

Summary of Keys

General Keys

J - Jetison payload, and causes CTD at random
G – Toggle landing gear

Autopilot commands

1 - Toggle autopilot
2 - Toggle heading hold mode (by default this toggles all modes)
3 - Toggle altitude hold mode
4 - Toggle wing leveller mode

P - Toggle autopilot
7 - Set heading
8 - Set Altitude
9 - Set Wing Level - a/p currently overrides this value
0 - Set Pitch level - not implemented

Aircraft Specific Keys

F – Toggle flaps down/up in 25% increments

Flight Plan commands

L - Load flight plan (sample included 'test')
1 - Enagage auto pilot – flight plan requires autopilot active
S - Step forward the flight plan to next phase

Other Keys

Keystrokes for other key events in the aircraft may be defined on a case by case basis in the animations section of the ini file – see [AMIN_SEQ_x]

Aircraft INI File Settings

<Some of this section is not finished – where possible sample values are used in lieu of an explanation>

The Aircraft.dll module gives support to Aircraft that can be configured through initialisation file. Most of the conventional Aircrafts will be supported by this module. You will be able to create your own Aircraft without the need to write a specific DLL.

Limitation

-maximum of 10 different Aircraft types in the same scenario.
-maximum of 16 thrusters exhaust rendering for main engines
-maximum of 10 payloads
-maximum of 16 docking ports
-maximum of 8 animation sequence
-maximum of 128 components for the animation sequences

The best way to get a feel for how aircraft parameters should be set is yo use the sample aircraft as a guide. Information regarding these can be found at the end of the document.

Modules\Aircraft.dll : the module implementing the generic Aircraft class
Config\Aircraft\Aircraft.cfg : the class configuration file

Config\Aircraft\<vessel name>.ini : the ini file which define the parameters of the Aircraft called "vessel name" in the scenario file (see below)

Scenario Files

The scenario file must define the ship as follow :

```
...
BEGIN SHIP
<vessel-name>:Aircraft\Aircraft
...
```

The Aircraft module will search for an ini file with the name of the vessel defined in the scenario file only in the Config/Aircraft directory.

The following lines must be added to the scenario file

CONFIGURATION 0 (=launch, 1=in flight, 2=landed)
CURRENT_PAYLOAD (current payload not jettison yet)

The Load/Save fully supported

INI File format

This file follow the format of the standard windows ini files. The following sections and items can/must be defined…

Lift Parameters

[VLIFT]

lift function definition (if not defined, no lift will be produced)
PTx=(AoA,0)

PT0 to PT8 can be defined the angle must range from –180° to 180°, the second parameter is the lift coefficient

[HLIFT]

lift function definition (if not defined, no lift will be produced)
PTx=(-180,0)

PT0 to PT8 can be defined the angle must range from –180° to 180°, the second parameter is the lift coefficient

General Parameters

[CONFIG]

Aircraft configuration parameters section, most of these parameters correspond to the definition given by Martin schweiger in is documentation. Not all the parameters must be defined, other parameters will get default values when not specified.
MESHNAME="atlantis" : the mesh name to be loaded
SIZE=19.6 : the vessel radius size (m)
EMPTY_MASS=104326 : the vessel empty mass (kg)
FUEL_MASS=20000 : the vessel fuel mass (kg)
MAIN_THRUST=53400 : the vessel main engine thrust (N)
RETRO_THRUST=0 : the vessel retro engine thrust (N)
HOVER_THRUST=0 : the vessel hover engine thrust (N)
ATTITUDE_THRUST=7740 : the vessel attitude engine thrust (N)
ISP=5000 : the fuel specific impulsion (N/kg/sec)
PMI=(78.2,82.1,10.7) : the principal moment of inertia (kgm²)
CW_Z_POS=0.2 : the drag coefficient on the z axis when the vessel is moving forward
CW_Z_NEG=0.5 : the drag coefficient on the z axis when the vessel is moving backward
CW_X=1.5 : the drag coefficient on the x axis
CW_Y=1.5 : the drag coefficient on the y axis
CROSS_SECTION=(234.8,389.1,68.2) : the cross section in x,y,z axis (m²)
COG=8 : the center of gravity of the vessel above ground when landed (m)
PITCH_MOMENT_SCALE=0.00001 : the pith moment scale
BANK_MOMENT_SCALE=0.00002 : the bank moment scale
ROT_DRAG=(0.5,1.0,1.0) : the rotational drag
WING_ASPECT=0.7 : the wing aspect
WING_EFFECTIVENESS=2.5 : the wing effectiveness
LAUNCH_PT1=(1,0,0), LAUNCH_PT2=(-1,0,0), LAUNCH_PT3 = (0,0,1) : the coordinates of the touchdown points in launch configuration
LAND_PT1=(1,0,0), LAND_PT2=(-1,0,0), LAND_PT3 = (0,0,1) : the coordinates of the touchdown points in landing configuration. If you want both launch and land to be the same, just define LAND_PT1,2,3. The landing configuration is automatically selected after launch when a altitude of 100 meter is reached.
FOCUS=1 : specify which payload will get the focus when jettisoned, to keep the focus on the mothercraft specify –1. Payloads valid values start at 0 and goes to number of payloads-1.
VISIBLE=1 : specify if the internal mesh rendering must be done in cockpit view
CAMERA=(0,0,0) : specify the position of the point of view in cockpit view (should correspond to the cockpit position in the mesh). The default value will set the cockpit at the front of vessel.

[AIRCRAFT]

```
FLIGHT_MODEL=1        ; Support for multiple flight models
EFFICIENCY=0.8  ; Engine limiter - playing...
MACH_NUMBERS=(0.96,1.0,1.18)
WAVE_DRAG=0.02
```

[AUTOPILOT]

```
MAX_BANK=45   ; Maximum bank angle in turn
DAMP_BANK=6
ROC=1500
ROD=1500
DAMP_BANK=5
DAMP_ELEVATOR=1
DAMP_RUDDER=0.2
ELEVATOR_ASSIST=(0.1,0.0001,3.0)
RUDDER_LIMIT=0.1
ELEVATOR_LIMIT=0.5
```

[DOCK_x]

docking ports list (maximum 16 docking ports per vessel)

```
POS=(0,2.44,10.44) : the docking port position
DIR=(0,1,0) : the docking port approach direction
ROT=(0,0,-1) : the docking port longitudinal rotation alignment vector
```

Thrust Parameters

[EX_MAIN_x]

main engine exhaust rendering definition sections (as many section as engines rendering are required)
OFF=(-2.05,3.45,-14.2) : the offset of the rendering
DIR=(-0.050, 0.099, -0.994) : the direction of the rendering
LENGTH=4 : the length of the rendering at full thrust
WIDTH=0.5 : the width of the rendering at full thrust

[EX_BURN_x]

As for main engine exhausts, but only apply when thrust is greater than 99%

Aerodynamics

Aerofoils

```
[WING]
;wing and body lift and drag components
OFFSET=(0,0,-0.9)
CHORD=5
AREA=17
ASPECT_RATIO=3.88
```

```
[FIN]
;vertical stabiliser and body lift and drag components
OFFSET=(0,0,-7.5)
CHORD=1.8
AREA=4.1
ASPECT_RATIO=1.5
```

Control Surfaces

[RUDDER] - [ELEVATOR] - [FLAPS]

```
; rudder
OFFSET=(0,0,-7.5)
AREA=0.8
LIFT=1.5
```

[ELEVATOR]

```
; elevators
;OFFSET=(0,0,-3.5)
OFFSET=(0,0,-7.5)
AREA=1.2
LIFT=2.0; 1.5
```

[FLAPS]

```
; flaps
SEQ=3
OFFSET=(0,0,0)
AREA=0.9
LIFT=1.0
```

[AILERON]

```
; ailerons
LEFT_OFFSET=(-10.5,0,-3.5)
RIGHT_OFFSET=(10.5,0,-3.5)
AREA=0.2
LIFT=1.5
```

Ground Control

```
[STEERING]
SEQ=2
FACTOR=0.1
POSITION=0.5
FRICTION=(0.01,3.0)

[BRAKES]
FACTOR=1e4
KEY=B
```

Animation Sequences (and Drag Elements)

[ANIM_SEQ_x]

animation sequence definition sections, this corresponds to the definition of the ANIM_SEQ given by Martin in the sdk documentation

The same key can be used to start/stop one or more sequences.

<This portion of the manual is yet to be completed>

[ANIM_COMP_x]

animation component definition sections.

FOR ROTATION
SEQ=0 : the animation sequence containing this component
GROUPS=91, 92, 93 : the mesh group list that must be animated
RANGE=(0.0,1.0) : the range value when the animation takes place within the sequence (between 0 and 1)
TYPE=ROTATION : specify that the animation is a rotation
ROT_PNT=(-2.8,1.35,0.0) : the point of rotation (one point where the axis of rotation passes through it)
ROT_AXIS=(0,0,1) : the rotation axis
ANGLE0=0. : the angle of rotation that is applied from 0 to begin of range
ANGLE=160. : the angle of rotation (negative value for rotation in other direction) that is applied from end of range to 1. A linear transformation of the angle value is computed from angle0 to angle within the range defined.

FOR TRANSLATION
SEQ=3 : the animation sequence containing this component
GROUPS=6 : the mesh group list that must be animated
RANGE=(0.,1.) : the range value when the animation takes place within the sequence (between 0 and 1)
TYPE=TRANSLATE : specify that the animation is a translation
SHIFT0=(0,0,0) : the translation value that is applied from 0 to begin of range on the three axis X, Y, Z.
SHIFT=(0,-0.8,0) : the translation value that is applied from end of range to 1. A linear transformation of the translation components is computed from begin of range to end of range.

FOR SCALING
SEQ=3 : the animation sequence containing this component
GROUPS=6 : the mesh group list that must be animated

RANGE=(0.,1.) : the range value when the animation takes place within the sequence (between 0 and 1)
TYPE=SCALE : specify that the animation is a scaling
SCALE0=(1, 1,1) : the original values of scaling that are applied from 0 to begin of range on the 3 axis X, Y, Z.
SCALE=(0.1,0.1,1) : the final values of scaling that are applied from end of range to 1, a linear transformation of the values is performed from the value of scale0 to scale in the range defined.
REF=(-0.340,-3.1855,0) : the reference point to apply the scaling (fixed point where the mesh seems to be scaled around).

<This portion of the manual is yet to be completed>

Payloads

[PAYLOAD_x]

payloads definition sections
MESHNAME=Sidewinder : the mesh name of the payload for rendering
NAME=Sidewinder-1 : the name of the vessel that will be created when the payload is released
OFF=(0,1,2) : the offset position of the payload in the Aircraft
MASS=1200 : the total mass of the payload
MODULE=Sidewinder : the vessel class of the payload
SPEED=(0,1,0) : the release speed (relative to the Aircraft axis) in m/s
ROT_SPEED=(0.3,0,0) : the release rotation speed (relative to the Aircraft axis) in rad/s

Distribution

The distribution is taking advantage of the orbiter support for sub-directories. no file will be placed in the root directories. This should result in better classification and in no existing file being replaced.

Base package (always required)

Module/Aircraft.dll	: Main Aircraft vessel dll
Config/Aircraft/Aircraft.cfg	: Aircraft vessel class configuration file
Docs/AircraftDLL.pdf	: this documentation file

Addons Samples

Config/Aircraft/747.ini
Config/Aircraft/aircraft.cfg
Config/Aircraft/F-15-LFB.ini
Config/Aircraft/T-38.ini

Scenarios/Aircraft/Boeing 747.scn
Scenarios/Aircraft/F-15-LFB at KSC.scn
Scenarios/Aircraft/T-38.scn

Meshes/T-38.msh
Meshes/F-15-l.msh
Meshes/F-15-lfb.msh
Meshes/F-15-MS1.msh
Meshes/F-15-MS2.msh
Meshes/747.msh

Textures/F-15-l-front.dds
Textures/F-15-l-rear-wing.dds
Textures/F-15-l-side1.dds
Textures/F-15-l-tail.dds
Textures/F-15-l-top.dds
Textures/F-15-l-wing.dds

For developers

The full source code for Aircraft.dll is available in a separate package. Developer are encourages to rummage around and steal what they like.

I only ask that : -

- If you use my code that you include some suitable attribution in you addon.
- Since I have released the code for this, if you wish to keep your source closed, you send me a message to that effect. I am not stopping you from using it, just let me know ☺

Sample Aircraft

T-38 (My personal favourite)

Meshes courtesy of Kev33

The T-38 model in the aircraft package has been as a close to real life model. Care has been taken to make sure that the handling characteristics, engine performance (including afterburners) are as close to the real thing as possible.

This is a sleek flat wing fighter aircraft !

Animations are limited to landing gear (G) and canopy (J). A non-animated sequence is included for airbrakes (K). Nose wheel is animated during ground phase.

F-15-LFB (The Show-Pony of the fleet)

Meshes & Textures courtesy of Kev33

This has got the lot ! All animated control surfaces, beautiful textures, what a machine. Let attention has been taken with the realism of this model, but it is a real show piece of the new aerodynamic features in Orbiter and aircraft.dll

This is another sleek flat wing fighter aircraft ! Drag is a little low, so you will likely be landing with ya hair in FIRE !

All control surfaces are animated, including animated flaps, and a double rudder configuarion. Animated landing gear (G), animated airbrakes (K), and animated node wheel during ground phase.

Boeing 747

This is a bare bone public domain mesh.

This is implemented as an example of a thick wing heaving aircraft. A complete contrast to the 2 fighter jets in the package.

You'll need 60% fuel load to get of on the length of the KSC runway, and it flies heavy !

No animations. Flaps (F), Landing Gear (G), and Airbrakes (K) are all implemented as non animated drag elements.

License