



simcosmos
virtual dreams

NASA Ares I SC (multistage2.dll version)

for Orbiter v060929 (2006P1)

7th, January 2007

Ares I generic implementation / dev doc:
(3D, textures, research, design)

Main addon / further dll coding:
(CEV-Orion)

António Humberto Ferreira Maia
<http://simcosmos.planetaclix.pt>
<http://www.flickr.com/photos/simcosmos/simcosmos@clix.pt>

Franz Berner
<http://francisdrake.freewebspace4all.de/>
(<http://orbit.m6.net/Forum/default.aspx?q=posts&t=11685>)



Ares I launching Orion CEV (SRB phase)

This Ares I is an in-development Orbiter addon:
high possibility of missing, incomplete, incorrect info / pictures / components!

0. Contents

1. Introduction	2
2. Installation (and list of files)	5
3. Uninstalling... ..	8
4. The SRB Launcher Concept: Ares I	9
4.1 5 segment SRB	9
4.2 Interstage	11
4.3 Second Stage / Other Components	11
4.4 Performance Data	12
5. Flight Manual	13
5.1 Automatic Guidance and Ascent Modes.....	13
5.2 Ares I CLV: CEV Mission to ISS (or beyond).....	17
6. Legal Stuff, Credits, Thanks and Feedback	20

Appendix: Payload Guide and Customisation Tips .. 23

1. Introduction

NASA is currently between two eras of manned space exploration: the **Space Transportation System (STS)** and its famous winged Orbiter vehicle are in their final operational years (ending operations ~2010) and new launchers / mission objectives are being prepared. Low Earth Orbit objectives might still be around but more ambitious goals lay ahead: NASA wants to send astronauts to the Moon, Mars and beyond as per the goals of the **Vision for Space Exploration (VSE)**.

NASA's initial chosen path for the new VSE launchers was to go an almost 100% straight forward Shuttle Derived way (4 seg. SRB for CLV, modified SSME, heavy lifter's core diameter equal to STS ET and then 5 segment SRB, etc). Some of those conceptual ideas came from recommendations of the **Exploration Systems Architecture Study (ESAS)**:

http://www.nasa.gov/mission_pages/exploration/news/ESAS_report.html

http://www.nasa.gov/mission_pages/exploration/spacecraft/ares_naming.html

<http://www.fbo.gov/spg/NASA/GMSFC/POVA/RFI%2DCLV%2DUS%2D2006/Attachments.html>



The designs are still being tweaked (and changed from the initial concepts) but the basic idea is, at least currently, to still produce a +/- 22t (up to 24/26t?) smaller payload **Crew Launch Vehicle (Ares I)**, now composed by a 5 segment SRB and a 5.5m diameter second stage powered by a J-2X engine (Apollo heritage) and...

There are also the **Ares V** plans: a big heavy lifter with some STS / EELV / Apollo heritage, powered by 5 x RS-68 on a 10m larger core, two 5 segment SRB, EDS with J-2X and capable of lifting about 125t up to ~130t into LEO.

Note: the image at left is not 100% correct regarding a number of details. (particularly the Ares V representation, where I just "clumsily" changed some parts dimensions from other addon project).

From some time now that I have been producing 3D models related with the VSE and also implementing them into Orbiter Space Flight Simulator, one of the best space simulators / visualization tools around... and free! (thanks to Dr. Martin).

The current work is the result of a few (but significant) corrections / upgrades to visuals, performance and documentation that I have been doing to some of my past (and kind of outdated) addons about this topic (have slightly improved my skills and there is now a lot more information available than when comparing with the "pre-ESAS" days). In particular, this work is the result of upgrades being slooowly prepared for my **NASA VSE SC v2.0** addon (see my flickr space, link in the first page) in which, together with other options, I plan to use the SRB launcher as CLV and an heavy lifter also as CLV / for big cargo but where the specific designs are a bit different from real world (for example, my SRB launcher kept the fins in the first stage, its second stage might be powered by two standard J-2S engines and it would be used also in non-CLV roles, the heavy lifter kept the 8.4m diameter core, CEV / LSAM designs are also different from real life plans, etc).

The specific addon related with this pdf is, however, something closer with the current real life plans and is an attempt to at least provide a fun and interesting **Ares I** 'conceptual toy' for people to play with in the comfort of their home computers.

This Ares I is an in-development Orbiter addon:
high possibility of missing, incomplete, incorrect info / pictures / components!

Ares I multistage2.dll addon for Orbiter Simulator... **...what is it and what it is not?**

Due to several constraints, this addon / simulation does not pretend to be ultra-realistic: such objective would only be possible with a much more careful and professional study / implementation and...with some real life tests!

So, when "playing" with this Orbiter simulator addon please have in mind the following aspects:

a) The current Ares I version is implemented using Vinka's generic dlls. These generic dlls are a very nice tool to easily and quickly implement rocket and spacecraft designs inside Orbiter simulator but there are a few limits to what can be done with them, in particular if wanting to introduce not directly supported extra features without using some workarounds.

b) I'm using parts of the current document as a kind of **open development letter** to Franz Berner (aka francisdrake) so that he can implement a custom dll (c++) for this launcher. I hereby publicly donate my current Ares I 3D models / textures to Franz so that they can then be properly integrated in his nice free CEV-Orion addon package (and thus reduce external dependencies).

Meanwhile, and as mentioned above, I decided to release this implementation because of the following reasons:

b.1) To allow an easier work for Franz, with all the information necessary for a first custom dll version concentrated in a single place: this document (and related configuration files). **Note: this information concentration isn't done yet (time constraints)... Will try to complete it, when possible.** I also used Franz's "CEV-Orion" directory structure. Please visit the following link for the work in progress regarding such integration:

<http://orbit.m6.net/Forum/default.aspx?q=posts&t=11685>
(Orbiter Forums: Orion CEV Development)

b.2) Meanwhile and as a side consequence / despite the documentation isn't quite as I would wish, other orbinauts will be able to use the current multistage2.dll version to make some flights (remember that, for *now*, you need to also install Franz **CEV-Orion**, else will have a CTD when trying to run my scenarios) and, who knows (?), some of you might even have a look at the way I implemented this launcher design and perhaps use it as inspiration to start producing some addons ;-)

b.3) Even after Franz properly includes the needed launcher files inside his addon structure / unique zip I still might keep the multistage2.dll version online, for a given amount of time, because of **b.1)** + **b.2)** and also because it might be an easier way to share development efforts with Franz if, meanwhile, we decide to implement extra features such as simulation of SRB recovery or perhaps a cargo version of the Ares I launcher (by using my related custom fairings and perhaps even one of my Centaur upper stages, similar to the ones launched in Atlas rockets)...

c) Orbiter addons are made in the free times and the authors aren't usually rocket scientists neither pretend to be ones. Please respect these last two aspects of the add-on's development work when / if analyzing it or when / if providing feedback, in particular, if you work in the aerospace industry, NASA, etc: any constructive criticism is always welcomed!

Also please remember that, as what happens in real life, this is a work in progress: NASA's CLV is still in a heavy design phase and many configuration details or even broader changes can perhaps happen in VSE's implementation path (until operational hardware really starts to be produced). Beyond that, there is always space to fine-tune the simulation!

Having written the above *blablablablabla*, please do have an attentive look at all the rest of the documentation in order to better use the current multistage2.dll implementation of Ares I.

Hope you have fun ;-)

Thanks,
António Maia

7th, January 2007
(Portugal)

2. Installation

It should be relatively easy to install these files:

Step1 – Orbiter2006P1

If not having one yet, you really need an install of **Orbiter2006P1 (060929)**:

<http://orbit.medphys.ucl.ac.uk/download.html>

Note: do not mind about additional packages if this is the very first time you try Orbiter or if you have a lower computer specification. See the above link for download / installation instructions: you need to install, at least, the Base (full) 060929 (orbiter060929_base.zip).

Step 2 – NASA_AresI_SC_20070107dev + CEV-Orion Addon

After installing Orbiter:

Unzip my Ares I zip contents (**NASA_AresI_SC_20070107dev.zip**) into your main Orbiter directory (keeping the zip's folder structure).

Also remember that to run the current multistage2.dll version scenarios (marked with **VinkaSC**) you need to have Franz Berner's **CEV-Orion** installed. Please see the already mentioned above development thread or visit his site.

Step 3 - Required / Recommended or Included Addons

Daniel Polli aka Dansteph's Orbiter Sound:

<http://orbiter.dansteph.com/index.php?disp=d>

(<http://orbiter.dansteph.com>)

- Required only **IF wanting to have sound while running Orbiter**: highly recommended if you wish to greatly improve your Orbiter simulator experience.

Orbiter Sound comes with an executable: just run it and point to your Orbiter folder.

Vinka's Multistage and Spacecraft Generic dlls:

<http://users.swing.be/vinka/>

- All required **generic dlls and respective .cfg / .INI files are already included** inside this Ares I package's own directory structure (thanks Vinka!).

Note: these generic dlls are a method of implementing launcher configurations and also to implement spacecraft inside the simulator without the need to code c++ (it is all done via simple text files - .cfg and structured .INI - that can be opened with notepad). Vinka's generic dlls documentation is also distributed under \Doc\CEV-Orion\

However, if wanting to have the original complete generic dll packages with related documentation, examples and other files, please download them from Vinka's site. But again, all what is needed to run Ares I multistage2.dll version is already included: I'm writing this note just in case you wish to have a look at the original generic dll packages / site.

Next is the complete list of files included in [NASA_AresI_SC_20070107dev.zip](#):

[\README_NASA_AresI_SC_20070107dev.htm](#)
[Doc\CEV-Orion\AresI-SC_for_CEV-Orion_20070107dev.pdf](#)
[Doc\CEV-Orion\VINKA_multistage02_050721.pdf](#)
[Doc\CEV-Orion\VINKA_spacecraft03_060302.pdf](#)

[Scenarios\CEV-Orion\Description.txt](#)
[Scenarios\CEV-Orion\CEV-Orion on Ares I \(VinkaSC - ISS\).scn](#)
[Scenarios\CEV-Orion\CEV-Orion on Ares I \(VinkaSC - Moon\).scn](#)

[Modules\CEV-Orion\Vinka_genericDLL.txt](#)
[Modules\CEV-Orion\multistage01.dll \(+ 02.dll\)](#)
[Modules\CEV-Orion\spacecraft01.dll \(+ 02.dll +03.dll\)](#)
[Modules\CEV-Orion\Stage.dll](#)

[Config\Vessels\CEV-Orion\Stage.cfg](#)
[Config\Vessels\CEV-Orion\StageINV.cfg](#)

[Config\CEV-Orion\AresI_SC.cfg](#)
[Config\CEV-Orion\AresI_SC_CEV.cfg](#)
[Config\CEV-Orion\AresI_SC_LAS.cfg](#)
[Config\CEV-Orion\AresI_SC_SRB5.cfg](#)
[Config\CEV-Orion\AresI_CEV-Orion_ISS.ini](#)
[Config\CEV-Orion\AresI_CEV-Orion_ISS.txt](#)
[Config\CEV-Orion\AresI_CEV-Orion_Moon.ini](#)
[Config\CEV-Orion\AresI_CEV-Orion_Moon.txt](#)

[Config\Spacecraft\\[CEV-Orion\]CEV-Orion.ini](#)

[Meshes\CEV-Orion\AresI_STG1_SRB5.msh](#)
[Meshes\CEV-Orion\AresI_STG2_LWTcentaurJ-2X.msh](#)
[Meshes\CEV-Orion\AresI_LAS_1.msh](#)
[Meshes\CEV-Orion\AresI_CEVadapter_1.msh](#)
[Meshes\CEV-Orion\AresI_CEVadapter_2.msh](#)
[Meshes\CEV-Orion\Zahadum\zahadum.msh](#)
[Meshes\CEV-Orion\Zahadum\zahadum_1.msh](#)
[Meshes\CEV-Orion\Zahadum\zahadum_2.msh](#)
[Meshes\CEV-Orion\Zahadum\zahadumMach1_1.msh](#)

Textures\CEV-Orion\AresI_1ststage_SRB4_side.dds
Textures\CEV-Orion\AresI_1ststage_SRB5_rec0.dds
Textures\CEV-Orion\AresI_1ststage_SRB5_rec.dds
Textures\CEV-Orion\AresI_1ststage_SRBbottom.dds
Textures\CEV-Orion\AresI_2ndstage_foamLWTcentaur.dds
Textures\CEV-Orion\AresI_gen_silver.dds
Textures\CEV-Orion\AresI_IN.dds.dds
Textures\CEV-Orion\AresI_J-2S.dds.dds
Textures\CEV-Orion\AresI_LAS.dds.dds
Textures\CEV-Orion\AresI_LOGO_Agency.dds
Textures\CEV-Orion\AresI_Misc1.dds
Textures\CEV-Orion\AresI_PLF260_NASA.dds
Textures\CEV-Orion\AresI_LOGO_Agency.dds
Textures\CEV-Orion\AresI_SRBsmoke.dds

Textures\CEV-Orion\contrail1w.dds
Textures\CEV-Orion\contrail2.dds
Textures\CEV-Orion\exhaust2_noalpha.dds
Textures\CEV-Orion\exhaust_crcs

Sound\CEV-Orion\AresI_CEV-Orion_sounds.txt
Sound\CEV-Orion\AresI_SRBsep.wav
Sound\CEV-Orion\AresI_LASsep.wav

3. Uninstalling

It should also be relatively easy to remove these files... But remember that most of the files will be integrated, in some future occasion, inside Franz Berner's CEV-Orion zip... If removing the files you must be sure that you do not wish to use Orion-CEV (either by running my Ares I implementation or Franz custom dlls).

Step1 – Deleting Ares I / CEV-Orion files...

Just search for all folders called **CEV-Orion** under your Orbiter sub-folders:

`\Doc, \Config, \Config\Vessels, \Modules, \Meshes, \Textures, \Sound, \Scenarios, etc`

See also `\Config\Spacecraft\` for files with **[CEV-Orion]** designation.

In case of doubt, please have a look at the list of files available in the previous pages (and also look at Franz addon's zip structure).

Step2 – Removal of Required / Recommended or Included Addons

Dansteph's Orbiter Sound: <http://orbiter.dansteph.com>

- **ATTENTION:** Are you sure that really want to remove Orbiter Sound!? Even if you uninstall Ares I / CEV-Orion files from your Orbiter folder I highly recommend that you keep Orbiter Sound for a much nicer simulation experience with the default spacecraft and other addons. If still wanting to uninstall Orbiter Sound, please refer to that addon support site / docs / zip / exe.

Vinka's Multistage and Spacecraft.dll: <http://users.swing.be/vinka/>

- The generic dll files included inside the current Ares I addon's directory structure are removed if deleting Modules\CEV-Orion during Step1 above.

Note: not related with DIRECT addon but just in case you, for some reason, installed the full original packages / docs (from Vinka's site) and do not wish to keep them, please refer to that addon support site / docs / zip. Remember that many addons use those generic files in their default locations: you might want to keep them anyway.

STEP3: Delete also:

`\README_NASA_AresI_SC_20070107dev.htm`
(in Orbiter's root folder)

This Ares I is an in-development Orbiter addon:
high possibility of missing, incomplete, incorrect info / pictures / components!

4. The SRB Launcher Concept: Ares I

At a first look, the SRB launcher design seemed apparently (just apparently) simple: one solid booster acting as first stage and one liquid second stage powered by a single engine on top of the SRB.

Due to a number of reasons, real life's SRB launcher will probably be a more restricted design than this virtual implementation: until word in contrary, it will be used only for ISS related missions (cargo / crew) and for crew transfer duties regarding exploration missions, in both cases, delivering a **Crew(ed) Exploration Vehicle (CEV)** into a sub-orbital trajectory (a cargo version of the CEV is also planned). For all the rest, NASA currently expects to use the - also in the plans - new **Ares V** heavy lifter, EELV (Delta / Atlas) for probes or some other specific missions, etc or any other rocket, given certain constraints. This is why the SRB Launcher is also widely known as **CLV (Crew Launch Vehicle)** aka **Ares I** aka... "**the stick**".

DEV Note: In virtual world, we are not limited by political, economical, technical and other constraints: *our* SRB launcher, although based on real life plans, could not only act as a **CLV** but also as a cargo launcher for a number of different payloads and missions (maybe in a future release?).

In the following pages I will make a global overview of some of the basic assumptions that were used to implement this Ares I simulation. These assumptions are based in real life data but they do not pretend to be a one-in-one correspondence.

- alpha doc - high probability of missing, incorrect or not final info/pictures - alpha doc -

4.1) 5 segment SRB

4.1.1) Description (to be added later: images, mass vs thrust vs performance considerations for the simulation such as SRB's TVC or other different empty, propellant, thrust numbers, depending of related interconnected aspects such as recovery of the SRB, second stage aspects, etc)

- alpha doc - high probability of missing, incorrect or not final info/pictures - alpha doc -

4.1.2) Thrust Curve

A solid booster's thrust duration, variation and intensity is adjusted by playing with the fuel grain quantity, composition and geometry. This allows, among other things, to adjust the thrust accordingly with key moments of the first part of the ascent profile such as:

- lift-off
- maximum dynamic pressure (max Q)
- 2nd part of the SRB ascent, where the throttle must decrease in order to keep a comfortable environment for astronauts and cargo
- staging event

The 5 segment thrust curve being used (thrust level vs time) is +/-based on real data. (ATK/NASA 1998 - alpha doc: more considerations about this to be included on final documentation)

The 5 seg. SRB data (~27 points) was extracted from a picture that was part of that study and will hopefully represent all the main variations. Those extracted points were later edited mainly because this SRB launcher has a few configuration variants (only one of them is readily available in this zip) and there was the need to adjust it accordingly with a common factor for various types of missions.

alpha doc: on a later occasion this data will probably be reviewed but, for now, it seems that the performance is in concordance with expected values, despite the current simulation limitations. As a side note, less detailed data (less points) were also extracted to simulate a 4 segment SRB (for other of my addons).

Back to the thrust curve: the variation in grain geometry is not as *versatile* (in the lack of better expression) as the thrust variation in liquid boosters but it provides some (pre-defined) control and can also be considered as something elegant. For the moment (Vinka's generic dlls), the best way of implementing one thrust curve with more than 10 points is probably via a guidance file.

ATTENTION! The last paragraph means that a guidance file is currently used for each Ares I mission powered by Vinka's multistage2.dll and that the first part of the guidance file (until ~140s) must be equal, with exception for the roll program heading... All this to have a better 5 segment SRB simulation with generic dlls. If launching in manual mode (and keep full thrust) with the current multistage2.dll Ares I implementation the poor virtual astronauts / rocket will face some harsh ascent conditions :-)

A custom dll can easily implement (built-in) such thrust curve into the 5 segment stage itself (outside of guidance files). Franz, please see the provided guidance files for Thrust vs MET.

After SRB vs second stage separation (~132s), the simulation of the 5 seg. SRB's full burnout (thrust quickly decaying to "zero value", which happens at ~138s or ~140s) is not currently simulated.

Please remember once more that the current SRB implementation is not perfect: burnout masses (and also fuel mass) still needs to be adjusted as well the ISP, thrust curve, etc... Extra considerations about why I chose the current numbers for the SRB (and other components) will be added in a later version of this document.

To sum up a long story, the first part of the ascent is smoother now
(when comparing with my previous Ares I implementations uploaded at Orbit hangar mods)...

G control: up to 3.2g at MECO
(J-2X* always running at 100%)

MaxQ: less than 39KPa
(depending of payload mass, roll program...)

So, at least regarding the 5 segment SRB performance, things are a little better than what happened on previous addon versions, and part of this was thanks to the thrust curve implementation, although in a clumsy way and not counting with the simulation of extra stuff like, just to give an example, thrust variations accordingly with the temperature of the day and a few other tweaks I had to do due to the implementation method being used. But the important is to make that 5 segment SRB to carry the upper stage + CEV + LAS to about 60Km altitude (apogee of about 100Km) and deliver it at something like mach6 or so :-)

4.2) Interstage

(to be added later: interstage components, images... considerations about roll control, flight loads, masses, etc)

4.3) Second Stage / Other Components

(to be added later: second stage design assumptions, J-2X* information, additional considerations about LAS, CEV and spacecraft adapter; images for all that.)

4.4) Performance Data (all units in SI)

Ares I - CLV	
1st Stage	
Configuration	5 segment SRB
Dimensions	~57.80m / ~3.7m diam.
Propellant Mass	646735.18Kg
Dry Mass	111735.18Kg
Total Mass	758470.36Kg
Max. Thrust	15794834.26 N
ISP	~265.4s
Burn time (operational)	~131.56s (~138s until full burnout)
2nd Stage	
Configuration	1 x J-2XD
Dimensions	~26m / ~5.5m diam.
Propellants Mass	130000.00Kg
Dry Mass	13350.00Kg
Total Mass	143350.00Kg
Max. Thrust (vac)	~1217700.67 N
ISP (vac)	~448.3s
Burn time (@100% thrust / vac.)	~469.0s
Vehicle Performance	
Max Q	from ~35KPa up to 39KPa (mission / payload type dependent)
SRB separation	~60Km alt., Mach~6.00 MET=132s
LAS separation	~85Km alt., Mach~7.40 MET=162s
MECO	MET=594s to 600s (mission / payload type dependent)
Max Gs	~3.2g (near MECO, J-2XD at 100%, for 22.6t payload. Max allowed: 4g)
Payload 28.5 deg (ISS)	22.6t , 30Km x 300Km alt. (~2% prop. on 2nd stg)
Payload 51.6 deg (Moon, etc)	22.6t , -56Km x 185Km alt. (<1% prop. on 2nd stg)

Extra performance / development data / sources are available inside: \Config\CEV-Orion\ (see INIs)

This Ares I is an in-development Orbiter add-on:
high possibility of missing, incomplete, incorrect info / pictures / components!

Global Comments:

a) These are +/-maximum performance numbers from Orbiter's current simulation development phase and when making ascent tests to ISS or due EAST with a common CEV test mass of 22546Kg defined in the launcher's payload section.

b) The included automatic guidance(s) is (are) not optimised yet. The residuals in the core (at MECO) are at least ~1% of total propellants (0.7%min for ISS). Of course that better safety propellant margin can be obtained if assuming something like a 19864Kg CEV for ISS or if using the 22.6t CEV with -56Km x 185Km for the lunar missions (and thus keeping upper stage impact in Indian ocean, instead of Pacific)

c) Further mass considerations: LAS is 6.2t, the CEV to 2nd stg adapter is 950Kg (and stays attached until MECO in this simulation). The 1st stg empty mass includes a 5t interstage (simplification). The 2nd stage design assumes something like a 5.5m derived Centaur (optimistic dry mass?). Higher 2nd stage and / or CEV masses might result in minimal injection objectives (-56Km x 185Km) having to be achieved with CEV's service module if J-2X is not updated (thrust and ISP) and if second stage hasn't extra propellants, which by its turn could require other assumptions for the 5 segment SRB (more notes to be added later).

5. Flight Manual

5.1) Automatic Guidance and Ascent Modes

There are three different ways of launching Ares I

- Automatic mode
 - Manual mode (!)
 - Mixed mode
-

5.1.1) Automatic Mode

Attention! Attention! Attention! Attention! Attention! Attention! Attention! Attention!
Do **NOT** use time acceleration if activating the Automatic Mode!

Despite being activated with just a single press on the “P” key, the automatic launch procedure is, from a certain point of view, the most realistic of the three launch methods: the hard job is left to the computers in real life! Astronauts keep a close eye in the instruments / MFD and, except for a few *minor* button pressing, only take manual control if a serious anomaly arises during the ascent (and if such manual override is even possible, it would depend of the emergency type and phase of the ascent).

The files responsible for such automatic ascents have a .txt extension and are located inside **Config\CEV-Orion** with the same name of their respective scenarios: **AresI_CEV-Orion_ISS.txt** and **AresI_CEV-Orion_Moon.txt**.

If opening those files, you will see that they are responsible for:

- engine throttling (in particular, please remember that the **SRB thrust curve** is implemented in those guidance files!)
- roll program / azimuth targeting
- pitch profile
- separation events (such as SRB, LAS)

I recommend people to use the automatic ascent program at least once: another side advantage of using the provided automatic guidance are custom Ares I sounds.

Custom sounds

These were implemented with the main objective of creating some flight ambience (cool sound factor is the equivalent to *eye-candy* in the graphics department): they do not pretend to be 100% realistic.

Note: to allow for faster uploads / downloads and given that this is a kind of an alpha development zip I just included the SRB and LAS separation sounds. Later releases might include extra sounds (those whom have tried my DIRECT SDLV addon might imagine what I’m talking about) but that will probably only happen after I update this documentation with all the – for now – missing sections. As alternative, those of you knowing what are doing can edit the guidance files and insert any sound there ;-)

Computer Specifications / “Eye-Candy” and Impact on Guidance Behaviour

Please also have in mind that the results of the automatic guidance might vary depending of computer specification and, to be more specific, if during key moments of the ascent:

- having other programs running in the background,
- taking screenshots,
- changing between external / internal views,
- using external view with the camera pointing into the SRB smoke trail (using high resolution items or graphic card settings at their maximum might also impact)...

It might be natural that people with higher computer specs than my development PC (PIII@600Mhz, 384SDRAM, GeForce 5200FX 124MB RAM, Windows XP) might end up with different apogee / perigee than what I have here, in particular if following the entire launch in *internal view*.

One particularly sensitive phase happens when the automatic guidance is trying to *hunt* the intended launch azimuth, after starting the roll procedure: depending of the above constraints, that hunting and respective azimuth *lock* can take more or less time to accomplish having then an influence regarding the execution of the pitch program... For example, the pitch program might freeze until the azimuth is *locked* and, after that, the guidance will try to compensate by quickly changing to the pitch that should be happening at that given Mission Elapsed Time (MET). Another thing to remember is that, even not considering external interferences, the roll program is far from perfect: **for example, in this Ares I simulation I simply haven't bothered too much in programming the roll control... That is why ISS ascent is done in heads-up and other Exploration missions (launching East) are done with heads-down :-)**

The default camera modes available in the demonstration scenario have been conceived to at least try to minimize the interference of *eye-candy* items (such as the SRB smoke trail). If having problems with the launch azimuth *hunt + lock* I recommend that you do not pan the external camera position (for example, with the mouse, see Orbiter manual) at least until SRB separation.

Guidance Parameters

As you probably already concluded from the above talk, the provided automatic ascent are in a very early alpha state but should (?) get people up into a suitable injection *orbit*. Such *orbit* should have the perigee below 70Km altitude (to make the 2nd stage reenter and impact in one of our oceans, Indian or Pacific, again it might depend of guidance behaviour under a number of different operating conditions) but with a high enough apogee that CEV crew can then use as a starting point to raise perigee and fine tune ISS or EDS rendezvous (see scenario's description, performance table, etc)

Note: for this version I will not provide instructions about how to select other launch azimuth or about how to edit other guidance parameters. This information is available in Vinka's multistage.dll documentation anyway, if you wish to research and play with it.
(please see \Doc\CEV-Orion\)

5.1.2) Manual Mode

Given that we are in virtual world, it is always a LOT more fun to ride a rocket or spacecraft in full manual mode, don't you think so? If you are an experienced Orbiter user or have some Flight / Astronautics / Physics background it should not be very hard to do a manual launch at the first try (and even have much better results than if using the alpha automatic guidance!). On the other hand, if being a new Orbiter user, maybe it would be best to have a look first at the full automatic mode.

To launch in manual mode is just a matter of pressing **Ctrl** and **+** until max. thrust: Then, to control the launcher's attitude, use the numeric keyboard (please see Orbiter / Vinka's manuals); please also note that the pitch / yaw commands might be a bit slow.

Attention! Attention! Attention! Attention! Attention! Attention! Attention! Attention!
Remember that without the automatic guidance, the user has to do everything!

Main Engine(s) Throttling: in real world this is needed to control maxQ and g loads.

For this simulation and if launching in manual mode, you are free to not simulate / do not even worry about these aspects: the launcher will not be destroyed and the crew will always survive but... for a better realism and warm fuzzy feeling inside, you might want to replicate something like what is seen in the automatic mode :-)

Note: use **Ctrl** and **+** or **Ctrl** and **-** to throttle the SRB motor or 2nd stage engines up and down. Remember once more that the SRB has its thrust curve currently implemented in the automatic guidance: you will have to simulate such thrust curve (in a later generic dll version I might implement a lower definition thrust curve in the booster itself). Regarding the upper stage, in theory you should not need to worry about using something other than 100% thrust for the J-2XD (if launching a 22.6t payload mass).

Roll Program / Azimuth Target: now, this might be hard for less experienced users... On the other hand, you are now completely free to choose any launch heading allowed for KSC operations (and even beyond that, hehe).

Pitch Profile: reaching orbital injection conditions is a matter of converting vertical velocity into a given amount of horizontal velocity... This is why rocket trajectories are a kind of arc in the skies (switch to external view and zoom out when launching in automatic mode and while the SRB are still burning to see what I mean).

Different launch systems / different missions require different pitch programs... If wanting to simulate things in a more or less realistic way for **Ares I** and while in manual mode, try to aim for 60km altitude and a pitch attitude (relative to horizon) of ~12 to ~20 degrees (or so) at MET=132s. From there, tune the second stage pitch to achieve whatever injection conditions you wish and, at the same time, making sure your perigee stays below 70Km for a safe stage disposal (look at Map MFD in order to control perigee at MECO = upper stage's impact point... please make sure you do not impact the into someone's head, ok? Aim for the Indian Ocean, at west of Australia or, depending of mission constraints, to the middle of Pacific but... avoid islands!).

Separation Events:

There are three key events requiring user intervention if flying in manual mode...

MET / Alt	Events	Action Keys
132s / 60Km	SRB separation	J (as needed, for SRB + InterSTG sep))
162s / 85Km	LAS separation	F (do not press it before to see FPS! :-))
MECO++	CEV separation + activation	J (only if having Orion installed!)

5.1.3) Mixed Mode

There is also the mixed mode: you are free to start in automatic mode (press **P**) and then kill the autopilot (press **P** again) and continue the flight in manual mode as long as remembering the following important rules:

- a) Once the automatic guidance is turned off there is no way to turn it back on.
- b) In the current simulation, the autopilot can't be turned off at any moment: the launcher should be in a stable attitude. Switching to manual mode while the guidance is busy with an intensive pitch / yaw / roll change will probably mean the end of the mission given that the launcher might start to tumble without control (it depends of MET)

Constant pitch phases where auto to manual transition can be safely achieved are:

MET=1s

MET=130s to 134s, 12 degrees pitch (previous to SRB separation)

MET=142s to 460s, 30 degrees pitch (LAS is ejected during this phase)

MET=535s to 570s, 10 degrees pitch (when preparing MECO)

Note1: remember that the autopilot should only be started while at the pad!

Note2: there might be other phases of the ascent where should be possible to make an auto to manual transition, in particular if the launcher is making smooth attitude changes but try at your own risk!

Mixed mode allows a bit more flexibility to the virtual astronaut, for example:

- If simulating an ISS mission some persons might wish to let the automatic guidance do the SRB part of the ascent (**recommended!**) and then take control after SRB separation...
- Others might prefer to wait for the *30 degrees* constant pitch phase, disengage the auto and try to simulate a probably unrealistic but interesting (in virtual reality) Return To Launch Site abort with the CEV by making something similar to the STS RTLS procedure: pointing the second stage back to KSC, keep it at a safe altitude, managing the return in order to make the core hit the Atlantic but making the capsule land near the coast... All this while operating in a *safe* flight envelope. Haven't tried this with Ares I... Not sure if we have enough J-2XD thrust output for that "playtime" scenario.
- Some others might prefer to wait for the *10 degrees* phase to fine tune injection / MECO conditions... I recommend that you at least use the *10 degrees* (or the last *2 or 0 degrees*) constant pitch phase to keep an eye in **Map** or **Orbit MFD** anyway, even if making a fully automated launch! You might need to switch to manual for MECO if, in your computer, seeing that the perigee will not end up in one of our Oceans and / or if the apogee is becoming much higher than the intended!

Attention! Attention! Attention! Attention! Attention! Attention! Attention! Attention!

Remember that the automatic guidance is also responsible for the final countdown and other sounds: some users might prefer to press **P** just to hear the countdown and then kill the autopilot (**P** again) as soon as the SRB ignite. As alternative, manual mode enthusiasts / advanced users might prefer to adapt the scenario and guidance files in order to just include the sound calls in the guidance file.

This Ares I is an in-development Orbiter addon:
 high possibility of missing, incomplete, incorrect info / pictures / components!

5.2) Ares I CLV: CEV Mission to ISS (or beyond)

In this section I will make a descriptive walkthrough about the main ascent events for a nominal CEV mission to ISS (or beyond) with Ares I CLV.

By attentively reading this description, the previous pages and ~~also with the ascent table available next (perhaps you would like to print it and have at hand, when playing with Ares I)~~ I believe that you will be more than ready to fly this rocket!

Note1: in a later version I might use Orbiter's replay feature and prepare a full ISS mission demo with a few on-screen comments (unless someone wishes to contribute such good quality replay first!). Ascent tables will be included in later versions or... you can use the available automatic guidance and build your own custom table from there.

Note2: when starting Orbiter for the first time, make sure you have all realism settings ON (see Parameters tab). Then go to the Scenarios tab, search for CEV-Orion, double-click to open the folder, click in the scenario(s) (this Ares I uses scenarios with **VinkaSC** designation) to read its description, double-click to run it.

5.2.1) Ares I CLV: launch to injection (CEV – ISS / Moon, etc mission)

The demo scenarios start with an external view of Ares I, almost ready to launch. Most of the comments below will be about the ISS mission given that we have a nice one available as default in Orbiter.

As previously noted, it is only needed to press **P** to start the automatic ascent program. But attention! The moment for liftoff will depend of the orbital plane alignment with the intended target, in this case, the ISS (or an EDS, if you launched it before and edited the "Moon" scenario to include it). When the scenario starts we are not quite in the best launch opportunity yet...

Launch Preparation

Press **F1** to go into internal view and check **Map MFD** (already selected at left) or select **Align MFD**. The ISS orbital path projection should be very near KSC. (Same procedure for any other target)

Press **T** twice to time accelerate until... hummm... **20:05/06 or so UT** (look at the top right corner) should be +/- ok (press **R** twice to return to normal time); once happy, press **P** to start the automatic launch. If having Orbiter Sound installed you will then listen some cool stuff! There will be a -10s countdown... Press **F1** again to return to external view and see the launch in all its glory ;-)

Roll Program and First Moments of the Ascent

After **clearing the tower**, the **roll and azimuth alignment** program will kick-in. You can watch the launch from the ground camera but the rocket will soon become a tiny spot in the sky. At about MET=012s you might wish to select a closer external view (press **F2**).

Note: depending of the already mentioned constraints, the guidance program will try to do its best to properly align the rocket into the intended launch heading (northeast direction for ISS, East for Moon, etc) but the roll program + azimuth alignment might be a little clumsy in the current implementation. If that is the case there, please try to not change too much the default external camera (do not focus too much the SRB trail), at least not until the launch heading is locked and you feel that the pitch program is running ok (else the launcher will have some delay regarding the start of pitch program and that will impact in things like AOA, perigee / apogee, etc)

MaxQ

At some time of the ascent, the SRB motor will reduce its thrust in anticipation of **maxQ** (you can change to internal view, select the **Surface MFD** and read there the dynamic pressure): more or less at about the same time you will also see the launcher going transonic (if going back to external view, you will see that I tried to implement something like we see in STS flights, even if not 100% correct).

Note: multiply by 20.8854 if wanting to convert maxQ value from KPa to psf. Exceeding safe maxQ values in this simulation does not have any structural impact in Ares I... maybe in a later (specific dll) version(?)

After maxQ, the SRB thrust go up again and the pitch program will be a bit more "aggressive" in order to prepare the launcher for SRB separation attitude (its thrust will decay before that... Astronauts should not feel much beyond 3g during SRB phase)

Note: please do not pay too much attention to AOA / SRB phase, etc... The guidance is really in an alpha state (and there are also a few limitations). Better results might be achieved in manual mode!

SRB Separation / 2nd Phase of the Ascent

SRB separation will happen at MET~132s / 60Km altitude / mach6

Note: separation dynamics are not simulated; after separation, the SRB is just a dummy (inactive) 3D model and there is not a simulation interstage + frustum separation neither of 5 segment SRB recovery... although it would be a nice thing to have in a later specific dll version. For a first version, active separation motors would be a nice addition (see INI for positions).

After SRB separation, the second stage will ignite and will keep a constant pitch attitude of ~30 degrees: one of the several reasons why this happens is because this simple simulation has not functional TVC (Thrust Vector Control) for any of its components (it is all simulated by "fake" forces)... Perhaps the pitch program could be smoother with a few extra tweaks in the guidance, do not know (still researching)...

Anyway, it is in this "nose up" attitude that **LAS separation** will happen at MET=162s and at about 85Km altitude.

Note: the LAS is just a dummy 3D part (it is not functional) and its separation is automatic in Ares I multistage2.dll version.

That "30 degrees constant pitch phase" is also a good time to turn off the autopilot if wanting to simulate a kind of flight challenge... hummm, RTLS procedure? (see comments about Mixed Mode in previous pages) or if wanting to fight against gravity losses in a personal manner :-)

3rd Phase of the Ascent: building up speed and preparing MECO...

After that, it will be impossible to make a safe and probably crazy try to return to KSC: Ares I upper stage will start reducing its pitch and building up horizontal speed faster.

At a given moment, but more to the end of the ascent, the g count will start to increase until a maximum of ~3.2g. Attention! There should not be the need to throttle down the J-2XD unless for MECO, at least if flying something like a 22.6t payload mass. I'm also not sure yet about the throttling capabilities of the J-2XD.

Yet another note for this phase of the ascent: keep an eye at **Align MFD** even if using the automatic guidance (there might be the need for some manual corrections – yaw commands – in order to reduce at maximum the relative inclination between CEV and its orbital target).

Concerning MECO: for both ISS + Moon missions you might want to go for -56Km x ~185Km injection. Please see the notes in the Performance Table.

Notes: several abort modes are not quite well implemented or even studied yet;

If wanting to have an easy reading of the G your virtual astronauts are experiencing, I recommend any of the available custom MFD addons for that effect such as kwan3217's Accelerometer (<http://www.orbithangar.com/searchid.php?ID=2556>) or Duncan's Reentry MFD (<http://www.orbitermars.co.uk>).

After MECO, it should be just a matter of pressing **J** (only if having CEV-Orion installed) and you will be automatically aboard a functional CEV. Use its resources to raise the perigee. From such safe orbit: start rendezvous procedures necessary to accomplish the rest of the mission: go to ISS or... dock with a previously launched EDS and go to the Moon or... Rendezvous with a Mars Transfer Vehicle and go explore the Red Planet, etc, etc (you will need to install extra addons but that is a thing that exists in abundance / quality in our community ;-)

Happy Flights,
António

6. Legal Stuff, Credits, Thanks and Feedback

6.1) First, about my work (António Maia) and legal “stuff”

Except for the cases that will be referenced in the next page (6.2) or for cases of unintentional mistake or forgotten credits **I'm the author of all contents included in this document's file list.**

Making a few generic comments and sometimes being more specific about the 3D models (Ares I launcher, etc: original files are in .an8 format (anim8or)): they are not a masterpiece of art but I would like to write some “terms of use”:

- People do not need to contact or give me credit when uploading screenshots of these models / addon to public web spaces or forums unless those images are intensively and recursively used for a presentation, multimedia production or any other project. On such cases please let me know about that project and give at least the following credit (or any other credit to the respective authors)

António Maia (<http://simcosmos.planetaclix.pt>)

- The contents of this package must remain together. Unless for private use or other special occasions (where I should be informed and agree about and this is, for example, the case with integration and coding of my Ares I files inside Franz Berner's CEV-Orion addon) people are not allowed to change and / or extract parts of this package and redistribute that derived work without my previous permission. For parts that are not from me, please contact the respective authors and / or read their documentation.
- To be explicit: from my side, it's OK to redistribute this zip package (in compressed state) as long as the original zip name, folder structure and all contents are kept exactly the same as I prepared them. However, have in mind that I only commit myself to quickly support the non-dev marked package versions (the current one is a dev marked zip btw) of the specific Orbiter version they were made for (read: current Orbiter version, in this case 20060929) and that, except for special occasions, only support the packages that I personally uploaded to public distribution places.

In all cases, do not claim to be the author and please give the proper credits (or make them visible, point to them in some way). If having any doubt regarding the above, just contact me (simcosmos@clix.pt) or the respective authors.

Other very important notes:

About these files: do not make money with them!

And, the usual:

Use all these files at your own risk!

I'm not liable for any bad stuff that happens!

(CTD, data loss, health problems, if your cat is sad and never stops those long meooooow, the end of the world as we know it, and so on, and so on...)

6.2) Other Credits

This is where I try to sum up the credits for all the included components that were not fully made by me (António Maia aka simcosmos). If seeing any incorrect information or if you feel there is a missing credit please email so that it can be solved as soon as possible. Thanks!

Components	Author and/or Origin and/or + Info
Modules\CEV-Orion\Stage.dll Modules\CEV-Orion\multistage01.dll (+ 02.dll) Modules\CEV-Orion\spacecraft01.dll (+ 02.dll +03.dll) Doc\CEV-Orion\VINKA_multistage02_050721.pdf Doc\CEV-Orion\VINKA_spacecraft03_060302.pdf	Vinka's generic DLL (thanks also for 3ds2msh) http://users.swing.be/vinka/
Tip for SRB particle settings: http://www.orbitersim.com/Forum/default.aspx?q=posts&t=3978 (20050802)	Randolph Vallee (aka Lambo)
RCS texture: Textures\CEV-Orion\exhaust_crcs.dds And thanks also for all the nice toys ;-)	Franz Berner (aka Francisdrake) http://francisdrake.freewebspace4all.de/
Textures\CEV-Orion\contrail1w.dds ----- Textures\CEV-Orion\contrail2.dds ----- Textures\CEV-Orion\exhaust2_noalpha.dds -----	Orbiter's default / modified files: contrail1 with small tweak contrail2 (included to avoid other addon's overwrite) exhaust2 but with no alpha
Textures\CEV-Orion\SRBsmoke Textures\CEV-Orion\AresI_PLF260_NASA.dds	Images / Sounds produced from VSE / STS / Apollo multimedia (wip): www.nasa.gov

- alpha doc - high probability of missing, incorrect or not final info/pictures - alpha doc -

6.3) Extra Credits and Thanks

Beyond all mentions made in the previous page I would also like to thank:

Dr. Martin Schweiger (and all collaborators)

Orbiter, Orbiter, Orbiter, Orbiter, Orbiter, Orbiter, Orbiter ;-D

<http://www.medphys.ucl.ac.uk/~martins/orbit/orbit.html>

Read 10Oct2006, 16Sep2004 news to learn about
Orbiter capabilities as a Spacecraft Visualization Tool:

<http://orbit.medphys.ucl.ac.uk/news.html>

Daniel Polli aka Dansteph for Orbiter Sound

<http://orbiter.dansteph.com>

R. Steven Glanville

anim8or, not the heaviest 3D editor around...

...and that is why I just love it!

<http://www.anim8or.com>

Martin Wright

DXTBmp, cool to convert .dds

<http://www.mnwright.btinternet.co.uk>

Chris Bergin from <http://nasaspaceflight.com> for such interesting site / forum contents as well to all collaborating / participating there. Thanks also to the **Orbiter Community** in general.

6.4) Feedback (and future plans / work in progress)

If wanting to provide feedback about this Ares I (generic dll to specific dll) development effort as well about the CEV-Orion addon in general and see how things progress perhaps the best would be to use the respective Orbiter forum thread:

Orion CEV Development:

<http://orbit.m6.net/Forum/default.aspx?q=posts&t=11685>

- Boring Appendix: Payload Guide and Customisation Tips -

- alpha doc - high probability of missing, incorrect or not final info/pictures - alpha doc -

Dear reader, if you are new to Orbiter space simulator or if it's the first time you wish to add your own payloads I highly recommend a good, calm and very attentive reading at Vinka's great generic dll documentation available in \Doc\CEV-Orion folder or here: <http://users.swing.be/vinka/>

The process of adding a payload might be simplified in later generic addon versions (better documentation) or if using the planned future coded version by Franz Berner.

Meanwhile here go a few tips!

There are 3 types of "vessels" for Orbiter simulator, powered by different "technologies"

- **CFG** vessels
- **Spacecraft generic DLL** based vessels
- **custom DLL** vessels

CFG vessels have all their physical parameters (3D model reference, empty and fuel masses, thrust, cross sections, principal moments of inertia, etc) defined in a simple text file - having **.cfg** as extension - located under \Config or \Config\Vessels (or other places). This is the simplest method of having something in Orbiter but it has limitations such as: not possible to define more advanced physical parameters, animations...

Custom DLL vessels use, as the name implies, a **dll** with all their properties inside (dll called by a very simple cfg). On these cases, the limits of what can be implemented in a given vessel are only imposed by the Orbiter version capabilities and the imagination of the author but... this requires programming skills (c++)... If you do not know what is a compiler or what sdk means then please advance to...

To end, the best of the two worlds: the **spacecraft generic dlls**! These are dll that can be used for a lot of different vessel implementations and that interpret the physical properties from simple text files, this time with an **.ini** extension. People do not need to code c++ but can easily add advanced features to their vessels (such as virtual cockpits, animations, particle effects, custom exhaust textures, advanced physical properties, remote arms, etc) as long as they are supported by the generic dll.

(rest will be completed / adapted from other virtual project in a later occasion)

A) Creating a launcher configuration with custom payload

A.1) About Custom Fairings

A.2) Custom Payloads

B) Guidance File

C) Last Step: the launch scenario