UCGO 2.0 Cargo SDK
for Orbiter space simulator 2006 & 2010

©Daniel Polli 2010 All rights reserved http://orbiter.dansteph.com

(This doc is not a complete guide on modeling or texturing.
There are plenty of tutorials about that on Internet. Please see Google.)

Content:

1-Introduction

The cargo modules of UCGO are extremely versatile. There are so much features available that the only limit is your imagination. Doing new cargo does not require any programming (Except for Cargo->vessel), and it is really simple to make.

I wish you a lot of enjoyment while making cargo!

Dansteph

2-Cargo types

Cargo can be one of three types:

Inert cargo	Only here for eye-candy. One mesh, texture, config file and you are done! Want to make a fun container of "fried pink ornitorink"? You can do it.
Resource cargo	Contains one resource: oxygen, fuel, hydrogen, gold, fuel cell etc. (see notes on resource keywords) These resources can be consumed by vessel (orbiter module). This opens a world of possibilities: Ships may refuel of course, but you can also write a complete "Orbinomic add-on" from the factory or extracting derrick that "pumps" new cargo to a final base that "purchases" and consumes them. The only limitation is your imagination!
Unpackable cargo	Once unpacked by an UMmu or automatically (timer) this cargo can transform into several things: 1-Inert objects: table, chair, solar panel, light etc. Etc. (simple or multiple) 2-Base modules: Airlock, inflatable module (UMmus can breath inside it). 3-UCGO vehicles: rover, rescue boat, car...etc. 4-Orbiter vessel: probe, whatever is packed that can fit into a cargo box.

UMmus can move type 1 and 2 and pack them again; Type 3 allows all kinds of exploration rovers like Spirit, but UMmus cannot pack them again; and Type 4 gives absolutely unlimited possibilities- Once unpacked, the cargo becomes a full Orbiter vessel with endless possibilities: probe, a bomb that nukes an asteroid, laser, small droid, mining device that "drills" for new resources. If it can be done in C++, you can do it!

You are limited by your imagination and the dimension and weight to maintain some realism. (see below)

3-Cargo dimensions

Packed rule is as follows: Once packed, cargo must **absolutely** fit in a 1.3m cube. This is **mandatory!!** Not following this rule makes your cargo **not** UCGO standard. (If too many people go crazy and choose not to follow this rule, I may add size check in future versions that would simply dismiss cargo that exceeds this size)

This is because other authors expect that all cargo will fit in their cargo bay without giving odd results.

Also the container bottom must touch the "Y zero", otherwise it would float in the air in the cargo bay.

See "modeling cargo" for a template and check.

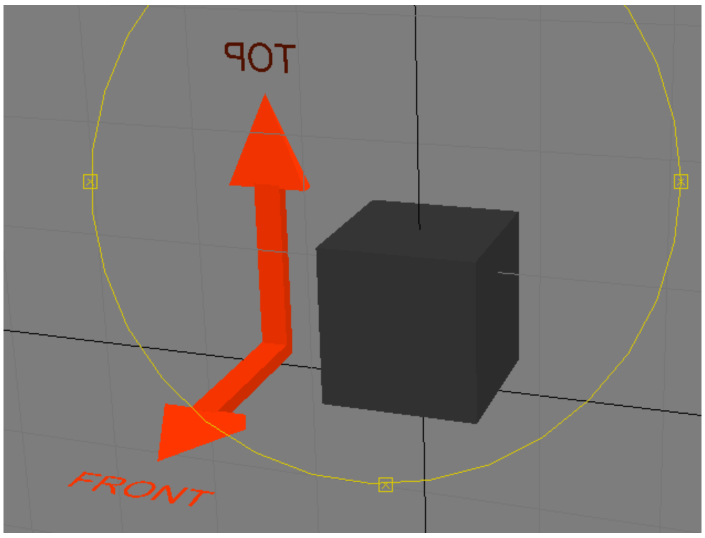
For unpacked cargo, the rule is more loose and depends on your idea of "realism". Probes, rovers, base modules or whatever should logically fit once disassembled into a packed container and it's weight set accordingly. I took some liberty with my "base modules" because they are inflatable, but I don't think we could make such things with our current technology. This said, you may avoid obviously impossible things as packing the shuttle into 1.3m cube. (unless your add-on is in the "fun" category)

4-Modeling Cargo

Packed cargo:

Packed cargo must fit in a 1.3 meter cube you can find a bounding box template in 3ds and max format in the folder «Doc/UniversalCarsAndCargos/SDK/ModelizingTemplate»

« PackedCargoSDKMaxShape.3ds » or *« PackedCargoSDKMaxShape.max »*



The bottom of your Cargo object must touch the ground (Y zero)

Note: Poly count, Textures:

Vessels can carry up to 40 cargo, so you must keep the packed cargo poly count and texture as small as possible. A 1000 poly cargo would give 40'000 poly with a full loaded ship. This would be unacceptable.
I recommend, 50 to 200 poly count maximum with one 512x512 or 1024x1024 textures only!

> *Of course, it depends of the type of cargo. A vessel may take several "oxygen" cargo containers but only one or two probes. If your cargo looks like a folded version of your probe, you might push the poly count up to 1000 poly. This should remain an <u>exception</u> and only for <u>special</u> cargo that will never be loaded in mass. Be careful!!!!!*

Once done you must save your packed cargo mesh in **«Meshes/UniversalCars»** folder, and your textures in **«Textures2/UniversalCars»** This is mandatory. (see below configuration file)

Once done, carefully check with existing cargo that your cargo dimensions are correct. You can compare with B34 connect module cargo it has the maximum dimensions except that the corners that are rounded.

Unpacked Cargo:

As said above the dimensions and poly counts are less important, but your object must still be in the middle of your modeling scene with a bottom that touches the Y zero. The exception is if your unpacked cargo is a Universal Car or an Orbiter Vessel. For cars, please see the cars pdf, For vessels you are free to design as desired.

Once done, you must save your unpacked cargo mesh in the **«Meshes/UniversalCars»** folder, and your textures in **«Textures2/UniversalCars».** This is mandatory. (see below configuration file)

This do not apply if your unpacked cargo is an Orbiter vessel, in this case, you are free to save them where you want.

5-Cargo resource keywords

UCGO SDK functions allow vessels to consume cargo resources. This is based on content keywords. Common examples would be to consume «fuel» or «oxygen» cargo to refill a vessel's tanks.

There is no limit on resource keywords. You can even define a cargo that contains «barbies» and vessel authors would use this keyword to refill their «barbies» tank. However, in order to define a common standard for both vessel author and cargo author, I propose to all the following list of basic keywords:

oxygen, fuel, food, water, hydrogen, gold, helium, metal, rock, powercell

Using these keywords gives a greater chance that other authors have made cargo or a vessel that can use them too.

6-Configuration file

The configuration file of cargo must go in the «**config/vessels/UCGO/Cargos**» folder. The simpler way to make one is to copy an existing configuration file of the type of resource you want (resource, packable, cars, vessel). Rename the copy and simply modify parameters into it. It's pretty self explanatory.

In the default distribution, there is almost one cargo of each type possible:

Inert example:	CargoBarrelFuel.cfg
Resource example:	CargoSpaceFuel.cfg
	CargoOxygen.cfg
Simple unpackable example:	CargoBaseTank.cfg
Multi-object unpackable example:	CargoBaseTable.cfg
Beam light example:	CargoBaseLight.cfg
Beacon example:	CargoLandingBeacon.cfg (XPDR frequency once unpacked)
Automatic unpack after delay:	CargoRescueBoatAuto.cfg
Spawn a universal cars:	CargoRescueBoatUMmu.cfg
Spawn an Orbiter vessel:	CargoProbeUMmu.cfg
Breathable base module for UMmu:	CargoBaseModule.cfg

Note on filename:

You may not be the only one that wants to do a «CargoFuel.cfg» cargo. I recommend to name your config file with at least 3-4 unique characters so there is no risk of overwriting between add-ons. If your name is John Doe, you may name it for example «JD66CargoFuel.cf» Following these rules for naming your mesh & textures with help avoid this problem.

Don't forget to add an bitmap image for the scenario editor.

7-Skin existing Cargo

It may be cool to have different space fuel cargo or other containers with different capacities (small ship), different colors, shapes, textures or company markings, so you are free to skin and release any existing cargo mesh or texture given it complies to license rules (see the html documentation license).

Basically the two main rules are:
1-Give credit where it's due (skin of ... by ...)
2-Don't overwrite one default file.

Copy config file mesh and texture giving them all a new names (see note on file name). Skin them as you wish, test them, and release them on OrbitHangar.

Remember anyway to respect the author and the Orbiter Community: Don't release one “skin” if you only added a text with a bitmap editor, or you used the the paint tool to badly color the texture. Instead, learn how to make great textures or meshes. Train and release only your best work when it really adds something and looks cool. Zip your add-on carefully. Add-ons that ask users to move files one-by-one into orbiter probably aren't worth even one upload.

Happy Cargoing ! :)