VideoCaptureMFD

(VCMFD)

Version 1.0a

for Orbiter

Spaceflight Simulator 2005/2006
(050216 – 060929)

November 4, 2006

Contents

1 Introduction

The VCMFD plug-in for Orbiter Spaceflight Simulator is intended to facilitate the production and capture to disk of Orbiter screen image frames in a manner suitable for their use in the preparation of video media sequences. To do this, VCMFD alters the behaviour of Orbiter so that it does not operate in real-time. The behaviour of Orbiter, and all other add-ons / plug-ins – including auto-pilot programs – should be otherwise normal, (i.e. Orbiter is not aware that it is not running in real-time).

In many cases Orbiter will still appear to be running in real-time, however the simulation-time speed will slow or increase depending on the computational load at each point in the scenario. In other words, with VCMFD activated, Orbiter does not vary its internal frame rate according to the load on the system whilst maintaining a real-time speed of simulation time. Rather, the internal frame rate is held constant – as specified by VCMFD (e.g. 24 or 48 fps) – and Orbiter takes the required amount of actual time to generate each frame.

This ensures that frames are produced at exactly equidistant intervals in simulation time. When formed into a video sequence (e.g. using "Virtual Dub"), the sequence will play smoothly in real-time. VCMFD enables the video capture of Orbiter sequences of unlimited polygon-count and otherwise high computational load, at theoretically unlimited frame rate speeds, on any computer capable of running Orbiter.

The VCMFD plug-in also includes a script facility which enables the user to run a scenario without any user input (e.g. keyboard or joystick). This allows the video developer to develop and view an entire sequence before capturing it to disk. This script includes control of vessel thrusters and MFDs but VCMFD is not meant to be an Orbiter auto-pilot as such. The use of the script for vessel control is intended primarily for "set piece" manoeuvres.

2 Installation

The VCMFD plug-in “VideoCaptureMFD.dll” is installed as normal in the Orbiter “Plugin” folder. It is also necessary to create a folder entitled **"ovf"** in the main Orbiter folder, i.e. the one in which the "Orbiter.exe" program file is stored. VCMFD can then be activated from the Orbiter Launchpad. N.B. with VCMFD activated, Orbiter will not run in real-time even if the scenario does not use VCMFD. VCMFD must be deactivated to restore Orbiter to its entirely normal function.

When Orbiter is recording, frames will be written to the "ovf" folder as bitmap files. They are about 1.5 MBytes in size. A scenario which captures several thousand frames will require substantial disk space.

3 Using VCMFD

VCMFD has been built as on Orbiter MFD. To use it in a scenario it should be included as an MFD in the scenario file as follows:

```
BEGIN_MFD Left/Right
  TYPE User
  MODE VideoCaptureMFD

  ... Script Lines

END_MFD
```

It is not necessary for VCMFD to be active in an MFD window for it to record or execute its script. It can be loaded or unloaded from an MFD window like other MFDs, but it must be loaded in a window to receive direct control input from the user during execution.

Although VCMFD can be used to manually capture sequences by starting and stopping recording using the keyboard commands SHIFT-B and SHIFT-E, it will often be operated as a sequence of script commands without any user input throughout the scenario. The script allows the user to program practically any sequence that could be achieved using keyboard/joystick input. This is possible since the VCMFD script allows the direct input of data to MFD's that normally requires the completion of an Orbiter input/selection box by the user.

4 Recommended Orbiter Settings

VCMFD currently produces bitmap captures of Orbiter frames at 16-Bit colour depth only. In order to get the best results, Orbiter video colour should be set to 16-bits (see the Video tab in the Launchpad). Similarly, set the computer display to 16-bit colour (click on the Control Panel Display icon). Using 16-bit colour does not result in any significant reduction in quality of bit-maps produced compared with higher colour depth settings.

VCMFD produces bitmaps of the entire screen when Orbiter is running. Orbiter must therefore be run in full-screen mode when using VCMFD. Orbiter frames can be captured at different screen resolutions, however 1024 X 768 is recommended even if the intention is to produce a video at 800 X 600 resolution (Virtual Dub provides a built in filter to reduce frame size automatically).

5 Script Commands

The VCMFD script contains a number of different commands which take various combinations of numerical and alphanumeric parameters. This manual describes the syntax and semantics of these commands. The reader is directed to the benchmark scenarios provided with the distribution for detailed examples of how to use these commands.

All script commands (except "Render") take at least one parameter – the start time – which indicates when the command is to be activated. This value refers to the frame number – not the time in seconds – when the command will execute. This allows precise control over each frame in the video sequence. Since Orbiter is not running in real-time, the simulation time can be calculated exactly from the frame number reference, e.g. frame 240 is precisely 10 seconds into a 24 fps sequence.

Many commands (e.g. thruster commands) also take a "finish time" parameter. The command will cease to execute on the finish time frame, i.e. "command 240 480" will execute for frames 240 to 479.

In the description that follows, command parameters are described using a C format descriptor followed by the parameter name, e.g. "%d-start time" indicates an integer parameter called "start time". Optional parameters are placed in brackets. Script command reserved words are written in upper-case, although case is not actually significant when the script is parsed.

Although VCMFD does not sort commands before execution, they do not have to be in exact temporal order to execute correctly. This allows groups of commands for a particular vessel or of a particular type to be grouped together for clarity. Commands of the same type (e.g. main engine commands for a particular vessel) do have to be in order.

5.1 General Commands

RENDER %f-framerate %f-scaling [%d-framechop]

The "Render" command is probably the most import command in the VCMFD script. All VCMFD scripts should start with this command. The *framerate* parameter indicates the required framerate e.g. 24 fps. The *scaling* parameter refers to the scaling of all subsequent command start and finish time parameters. In order to facilitate the recording and testing of scenarios at different frame rates, the "scaling" parameter (normally set to 1, 2 or 3), will scale all subsequent timing parameters by this factor. By adjusting this parameter, a scenario can be played at different frame rates without any further alteration.

All of the example scenarios contain two Render commands (one or other is commented out). They are:

```
RENDER 24.0 1.0
RENDER 48.0 2.0 2
```

The first indicates a fps of 24 with time scalings of 1, i.e. the scenario is described with timings for 24 fps. The second indicates a fps of 48 with a scaling of 2. This will play the same scenario at 48 fps with timings doubled. This will give exactly the same video sequence at 48 fps instead of 24. If the scaling is left at 1.0 the events in the sequence will play/occur at twice the speed.

The *framechop* parameter refers to the "frame-chop" rate. Orbiter appears to try and compensate for the heavy load of some high polygon scenarios by halving the real frame-rate and displaying each frame twice. In order to circumvent this and get the actual frame rate required, it is necessary to record with the frame rate at double the required rate and then chop every second

frame. In other words to get 24 fps guaranteed, record at 48 and take every second frame. Since this involves writing twice the required number of frames to disk, the "framechop" parameter (set to 2) specifies that only every second frame will be written. In this way only the frames actually required are written to disk.

In creating a video sequence the first Render command is often used while developing the scenario, with final testing and recording done using the second, higher fps command.

N.B. Only one RENDER command should be active in any given script. If more than one is used when the script runs, the last one specified will overwrite the earlier ones. RENDER can be placed anywhere in the script but the beginning is usually most convenient.

RECORD %d-start-time %d-finish-time

This command writes frames to the "ovf" directory from frames *start-time* to *finish-time* - 1. The frames will be written to the "ovf" directory and will be called 000xxxx.bmp where xxxx is the frame number / chop factor. If a chop factor higher than 1 is used, the frames will still be numbered consecutively. This allows for easier loading into Virtual Dub.

More than one RECORD command can be placed in a script. Normally, as with the RENDER command, only one RECORD command will be used for any given recording.

5.2 Camera Commands

CAMERA MODE [%s-vessel] [%s-mode] %d-start-time

Changes the camera mode and/or target vessel at *start-time*. The *vessel* and *mode* parameters are both optional but at least one must be used. If the *vessel* parameter is not used, *mode* applies to the current focus vessel. If *mode* is not used, the camera is attached to the indicated vessel without changing mode.

Values for *mode* are:

EXTERNAL – switch to external view.
INTERNAL – switch to internal view.
NOCHANGE – equivalent to omitting the mode parameter.

CAM_COCKPIT – switch to Cockpit View.
CAM_TARGETRELATIVE – switch external view to target relative mode.
CAM_ABSDIRECTION – switch external view to absolute direction mode.
CAM_GLOBALFRAME – switch external view to global frame mode.
CAM_GROUNDOBSERVER – switch external view to ground observer mode.
CAM_PRESET_XX – switch to preset camera view number XX.

COCKPIT_GENERIC – switch internal view to generic cockpit mode.
COCKPIT_PANELS – switch internal view to 2-D panels mode.
COCKPIT_VIRTUAL – switch internal view to 3-D virtual cockpit mode.

PANEL_LEFT – switch to left neighbour panel in 2-D mode.
PANEL_RIGHT – switch to right neighbour panel in 2-D mode.
PANEL_UP – switch to upper neighbour panel in 2-D mode.
PANEL_DOWN – switch to lower neighbour panel in 2-D mode.

CAMERA %s-scroll-direction [%s-vessel] %d-start-time %d-finish-time [%f-scroll-speed]

See scenarios 11 – 20 for exmples of CAMERA MODE commands.

This command applies to ground observer mode and internal 2-D panels views only. *Scroll-direction* can take the following values:

2-D panels view:

SCROLL_LEFT – scroll panel to the left.
SCROLL_RIGHT – scroll panel right.
SCROLL_UP – scroll panel up.
SCROLL_DOWN – scroll panel down.

Ground observer view, target-lock enabled:

TRACK_LEFT – track to left.
TRACK_RIGHT – track to right.
TRACK_UP – track up.
TRACK_DOWN – track down.
TRACK_FORWARD – track forward.
TRACK_BACK – track backward.

Ground observer view, target-lock disabled:

TILT_LEFT – pan camera to the left.
TILT_RIGHT – pan camera the right.
TILT_UP – pan camera up.
TILT_DOWN – pan camera down.

The scrolling/tracking commands depend on the PanelScrollSpeed and the CameraPanSpeed parameters that are normally set in the Orbiter configuration file. The *scroll-speed* parameter overrides these values for a specific CAMERA command. They can also be set in the VCMFD script using the SET Command as follows.

SET %s-variable %d-start-time %f-speed

Values for *variable* are:

PANELSCROLLSPEED – set the Panel scrolling speed in 2-D Panels view.
CAMERAPANSPEED – set the Camera Panning speed in Ground Observer mode.

TOGGLE TARGETLOCK %d-start-time

The TOGGLE TARGETLOCK command is also available to switch Target lock on and off in Ground Observer mode.

See scenarios 18 and 19 for examples of CAMERA *scroll-direction* commands

CAMERA %s-external-direction [%s-vessel] %d-start-time %d-finish-time [%f-speed]

This applies to external views only. Values for *external-direction* are:

AZIMUTH – Horizontal rotation about the target.
POLAR – Vertical rotation about the target.
ZOOM – Zooms towards or away from the target.

The speed parameter is in radians for AZIMUTH and POLAR. For ZOOM it refers to a percentage change of distance to target at each step e.g. 0.95 decreases distance by 5% at each step, 1.05 increases by 5%.

See scenarios 4 and 11 for examples of CAMERA *external-direction* commands.

CAMERA INTDIR [%s-vessel] %d-start-time [%d-finish-time] %f-x1 %f-y1 %f-z1 %f-x2 %f-y2 %f-z2] [GRAD]

This command sets or moves the internal camera direction for all Cockpit Views. The camera direction is specified using vectors. The camera can be set immediately to a given direction by specifying only the *start-time* and the first vector i.e. *x1*, *y1* and *z1*. A progressive camera movement can be achieved by including a *finish-time* and either a second vector – *x2*, *y2*, *z2*, or by specifying only the first vector and including the GRAD keyword. This will use the current camera position as the first vector and *x1*, *y1* and *z1* as the second vector.

CAMERA OFFSET [%s-vessel] %d-start-time [%d-finish-time] %f-x1 %f-y1 %f-z1 %f-x2 %f-y2 %f-z2] [GRAD]

Similar to the previous command, this command moves the camera position in Virtual Cockpit mode.

See senarios 13 and 17 for examples of CAMERA INTDIR and CAMERA OFFSET commands.

5.3 MFD Commands

MFD [%s-ID] MODE [%s-mfd-name] %d-start-time

Switches an MFD to a different mode. Values for *ID* are:

LEFT, **RIGHT**, **USER1**, **USER2**, **USER3**

These are the same for all MFD commands.

Mfd-name specifies the name of the MFD to be activated. This can be either a standard built-in Orbiter MFD or a plug-in MFD. The required name is the title of the MFD that will appear in Orbiter when the SEL button is pressed on an MFD window. (For the "ALIGN PLANES" MFD only "ALIGN" is used, and for the "SYNC ORBIT" MFD only "SYNC" is used.)

Alternatively the format **NUM_**xx can be used to specify MFD number xx. This refers to MFD number xx in the list of MFDs displayed in the MFD selection screen.

MFD [%s-ID] KEY %s-key-id %d-start-time

Sends the key-stroke specified by *key-id* to the MFD. *Key-id* is in the form OAPI_KEY_xx where xx is the actual key – A,B,C, ... 0,1,2 ... etc.

A full list of key-strokes in this format is found in the "OrbiterAPI.h" file of the OrbiterSDK suite.

MFD [%s-ID] INPUT %s-key-id %s-input-string %d-start-time

This is used when *key-id* corresponds to a key-stroke which would normally open an Orbiter input box. VCMFD passes the value specified by *input-string* directly to the MFD without opening the input box. If the value passed via *input-string* is not a string, i.e. an integer or floating point value,

it must be prefixed by the # sign. More than one value can be passed in this way via *input-string*.

See scenarios 3 – 9 for exmples of MFD commands.

5.4 HUD Commands

HUD REF %d-start-time

Toggles the HUD between different Nav references.

HUD COLOUR %d-start-time

Toggles the HUD between different colours.

HUD INTENSITY %d-start-time %d-finish-time %d-speed

Changes the brightness of the HUD display. *Speed* indicates the rate of change with a negative value indicating decreasing brightness.

HUD MODE %s-mode-type %d-start-time

Switches the HUD to the required mode. Values for *mode* are:

ORBIT
SURFACE
DOCKING
NONE

HUD TARGET %s-input-string %d-start-time

Specifies the HUD Target.

See scenario 15 for examples of HUD commands

5.5 Event Commands

EVENT [%s-vessel] %s-event-id %s-event-type %d-start-time [%d-finish-time] %f-x1 %f-y1 [%f-x2 %f-y2]

This command allows the programming of Orbiter cockpit mouse events. *Event-id* specifies the ID of a mouse event defined for a 2-D panel or Virtual Cockpit. *Event-type* specifies the type of event required. Values for *Event-type* are:

MOUSE_IGNORE
MOUSE_LBDOWN
MOUSE_RBDOWN
MOUSE_LBUP
MOUSE_RBUP
MOUSE_LBPRESSED
MOUSE_RBPRESSED
MOUSE_DOWN
MOUSE_UP
MOUSE_PRESSED

X1, *y1* and *x2*, *y2* specify the mouse-coordinates of the requested event. Use only *start-time* and *x1*, *y1* to program a single event. Using *finish-time* and *x2*, *y2* allows the programming of progressive events, e.g. the movement of a throttle or brake control.

See scenario 16 for examples of EVENT commands.

5.6 NAVMODE Commands

NAVMODE %s-navmode-type %d-start-time

Toggles the specified Orbiter auto-pilot function at *start-time*. Values for *navmode-type* are:

KILLROT
HLEVEL
PROGRADE
RETROGRADE
NORMAL
ANTINORMAL
HOLDALT

See scenario 2 for examples of NAVMODE commands.

5.7 Thrust Commands

THRUST %s-thrust-group [%s-thrust-direction] [%s-vessel] %d-start-time %d-finish-time %f-thrust-value [GRAD]

Sets the required thruster group to *thrust-value* for *start-time* to *finish-time* - 1 frames. If GRAD is specified, the thruster setting increases progressively from its current value at *start-time*. The required thruster group is determined from the *thrust-group* and *thrust-direction* parameters. If *thrust-group* is specified as MAIN, RETRO or HOVER, the *thrust-direction* parameter is not required. Values for *thrust-group* are as follows.

MAIN – fire the main engines.
RETRO – fire the retro engines if available.
HOVER – fire the hover engines if available.
PITCH – rotate the craft about its lateral axis.
ROLL – rotate the craft about its longitudinal axis.
YAW – rotate the craft about its vertical axis.
LONG – engage fore/aft translational thrusters.
LAT – engage lateral translational thrusters.
VERT – engage vertical translational thrusters.

Values of *thrust-direction* for each value of *thrust-group* are as follows:

MAIN, RETRO, HOVER:
    none
PITCH:
    UP, DOWN
ROLL:
    LEFT, RIGHT
YAW:
    LEFT, RIGHT
LONG:
    BACK, FORWARD

LAT:
LEFT, RIGHT
VERT:
UP, DOWN

See scenarios 1 and 14 for examples of THRUST commands.

5.8 Control Surface Commands

AIRCTRL %s-control-group [%s-control-direction] [%s-vessel] %d-start-time %d-finish-time %f-control-position %d-speed

Moves the *control group* in *control-direction* to *control-position* at *speed* times the default value for *start-time* to *finish-time* - 1 frames. Values for *control-group* are as follows:

ELEVATOR
RUDDER
AILERON
FLAP
ELEVATORTRIM
RUDDERTRIM
WHEELBRAKE

Values of *control-direction* for each value of *control group* are as follows:

ELEVATOR:
UP, DOWN
RUDDER:
LEFT, RIGHT
AILERON:
LEFT, RIGHT
FLAP:
UP, DOWN
ELEVATORTRIM:
FORWARD, BACK
RUDDERTRIM:
LEFT, RIGHT
WHEELBRAKE:
LEFT, RIGHT, BOTH

See scenario 14 for examples of AIRCTRL commands.

5.9 Miscellaneous Commands

TIME %d-start-time %f-value

Sets Orbiter time acceleration to *value*.

FOCUS %s-vessel %d-start-time

Sets the current focus vessel to *vessel*.

VESSEL [%s-vessel] KEY %s-key-stroke %d-start-time

Sends *key-stroke* to *vessel* or to the current focus vessel if *vessel* is not specified. This allows the script to activate vessel animations or general functions like undocking.

TOUCHDOWN [%s-vessel] %d-start-time %f-x1 %f-y1 %f-z1 %f-x2 %f-y2 %f-z2 %f-x3 %f-y3 %f-z3

Sets touchdown points to the vectors specified by *x1, y1, z1, x2, y2, z2, x3, y3, z3*.

DUST [%s-vessel] %d-start-time [%d-finish-time] %d-thruster-index %f-pos.x %f-pos.y %f-pos.z %f-srcsize %f-srcrate %f-v0 %f-srcspread %f-lifetime %f-growthrate %f-atmslowdown %s-ltype %s-levelmap %f-lmin %f-lmax %s-atmsmap %f-amin %f-amax [%s-texnam]

Dust effects can be achieved by adding an Orbiter exhaust stream as specified above to thruster *thruster-index* at postion vector *pos.x, pos.y, pos.z*.

Values for *ltype* are:
EMMISIVE, DIFFUSE

Values for *levelmap* are:
LVL_FLAT, LVL_LIN, LVL_SQRT, LVL_PLIN, LVL_PSQRT

Values for *atmsmap* are:
ATM_FLAT, ATM_PLIN, ATM_PLOG

See scenarios 10 and 20 for examples of the commands in this section.

Copying / Warranty

This software is freeware and can be distributed without any fee. All files must be included. The software is the property of David Harley. It is Windows software and there is NO WARRANTY of any kind. There is no guarantee against any system damage or error arising from its use. Use it at your own risk. This software has been created for the 2006 edition of Martin Schweiger's Orbiter spaceflight simulator.

Download

VCMFD can be downloaded from *http://www.orbithangar.com/*