

WARPMAP

r.0.1 – 3/2010

by Eloi Maduell Garcia
www.playmodes.com

WarpMap [WM] it's a simple media-player application that let's you play several layers of images or videos with a quad-warping and mesh-warp deformation effect for each layer.

Interface

WM interface is mainly operated trough keyboard and mouse. Keyboard short cuts are distributed to be used with one hand, while holding binoculars on the other hand for looking at the projection precisely. Some OSC commands are implemented for launching the videos. If you monitor the console you'll see some debug information usefull in case or errors or problems.

Configuration of layers

WM layers are configured from XML files present on the ./data/xml folder.

On the XML, inside the **<config>** tag you can specify a list of images and videos which will be part of that layer. Inside the **<quad_config>** and **<mesh_config>** there are the quad and mesh point positions.

If you add more XML files with the correct format, different layers are available to play, at the same time or separetly. Different layers are blended with GL_MAX function.

On the folder ./data/appData there's a default XML file to start with. Copy it to the ./data/XML folder, rename it properly and put your data in.

images

The images are specified trough **<imgs>** tag. The files should be placed on the ./data/img folder. Accepted format is JPG. For example if you want to have 3 images, you'll have something like :

```
<config>
...
  <imgs> image1.jpg </imgs>
  <imgs> image2.jpg </imgs>
  <imgs> image3.jpg </imgs>
...
</config>
```

videos

The videos could have audio included on the same file or not. We encourage and recommend to separate audiovisual clips into two different files, one for audio and one for video. The reason of this issue is to have better sincronization skills between both streams in intense looping and triggering.

For each layer you could specify which video loop mode will apply to the whole list of videos. Available loop modes are :

- 1 no loop
- 2 palindrome loop
- 3 normal loop

The audiovisual clips are specified through **<videoFile>** and **<audioFile>** pairs. The video and audio files should be placed on the ./data/video folder and have the same total time size. Accepted formats are AVI and Quicktime files for video and WAV files for audio. For example if you want to have 2 different audiovisual clips with normal loop mode you'll have something like :

```
<config>
...
  <videoFile> video1.mov </videoFile>
  <audioFile> audio1.wav </audioFile>
  <videoFile> video2.mov </videoFile>
  <audioFile> audio2.wav </audioFile>
  <loopMode> 3 </loopMode>
...
</config>
```

Operation

Use the quad warping mode to position the texture where you need. Then use mesh grid mode to modify precisely the internal warping issues as needed.

In mesh edit mode, if you press **'m'** you could toggle between **single point** edit or **multiple point edit**. When the mesh dots are drawn cyan you're in multiple point selection mode, so then you could click over the desired dots to move them together. Clicking over selected dots or pressing 'm' twice unselects the selected dots.

License

This binary files are registered under [GNU-GPL 3.0](#) license. There's absolutely no warranty for this piece of software, so use it under your own risk.

WarpMap is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

WarpMap is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with WarpMap. If not, see <http://www.gnu.org/licenses/>.

Observations

Don't change the directory structure for any reason.

Currently just available as it is for Windows XP and OSX 10.5.

Your computer needs to be able to work with OpenGL fragment and vertex shaders (GLSL).

There's a limit for textures up to 2048x2048 pixels.

Future ?

- OSC full integration with more commands.
- Synchronization of several computers for multi-projection displays.
- Edge-Blending shader for overlapping adjacent video projections.
- Lens Distortion patterns for mesh-warping
- RealTime 3D objects and cameras.

... and many more if donations or co-productions come in ;-)

Please donate through PayPal system at eloi@telenoika.net account (you don't need to have a PayPal account, just a credit card) or through bank account transfer detailed here :

Eloi Maduell Garcia
IBAN: ES32 3025 0001 1814 3314 1745
Caixa d'Enginyers
C / Via Laietana 39
Barcelona

Credits

Software piece developed by Eloi Maduell Garcia(*) in C++ with OpenFrameworks libs. Thanks to [Los Corderos](#) spanish theater company for producing the very first version of WarpMap. Thanks to [SQUAD company](#) from Mexico DF for co-producing this first release.

(*) Thanks to the whole OpenFrameworks project and users for it's active and helpfull community and forums.

(*) Texture warping GLSL shader based on [Memo.tv](#) shader for QuartzComposer.

Contact

For any further question visit www.playmodes.com or email us at warpmap@playmodes.com.

keyboard shortcuts

key	description	notes
F1	show/hide help screen	
F2	show/hide FPS counter	
F3	toggle full-screen on/off	
<SPACE>	set next active layer	
1	input from images	
2	input from video	
n	next image or video (depending on input)	
z	quad points edit mode	to move each corner separately
x	mesh points edit mode	to move mesh points, simple or multiple (m) modes
c	layer shift edit mode	to move the whole layer
v	play mode	to play the video clean
X	save XML file with current deformation	
R	reset mesh points deformation to zero	
p	play current video	
P	stop current video	
o	pause current video	
O	restart current video	
l	change loop mode of current video	this property does not get saved into XML file save command. Define it directly on the XML file.
!! doesn't work on play mode [v]		
q	select next point	
e	select previous point	
q	◀ move point left (1 pixel)	
d	▶ move point right (1 pixel)	
w	▲ move point up (1 pixel)	
s	▼ move point down (1 pixel)	
Q	◀◀ move point fast left (5 pixel)	just into quad points and layer shift edit modes
D	▶▶ move point fast right (5 pixels)	just into quad points and layer shift edit modes
W	▲▲ move point fast up (5 pixels)	just into quad points and layer shift edit modes
S	▼▼ move point fast down (5 pixels)	just into quad points and layer shift edit modes

OSC commands

OSC commands should be sent to the computer's IP address on port **12345**.

Imagine you have a layer defined with a XML file called "layer_name.xml" ...

command	address	value
play video	/wm/layer_name.xml/play	any value not used
stop video	/wm/layer_name.xml/stop	any value not used
pause video	/wm/layer_name.xml/pause	any value not used
set video frame	/wm/layer_name.xml/frame [float_value]	float value [0.0 ... 1.0]