BLENDERCAVE

MULTIMODAL SCENE GRAPH EDITOR
FOR VIRTUAL REALITY

D. Poirier-Quinot,
D. Touraine and B.F.G. Katz

ICAD 2013 – Lodz, Poland
Session 7 - HRTF and Spatial Audio
Presentation Plan

1. Framework
   - Scene Graph Editors in VR
   - BlenderCAVE context

2. BlenderCAVE
   - Architecture
   - Sound Rendering Engine

3. Usage Considerations
   - Performances
   - Scene Creation
1. Framework– SGE in VR research

- Key features
  - Scene content creation
  - Sustainability
  - Portability
  - Price
  - Internal Logic edition
  - External Logic edition
# 1. Framework– SGE in VR research

- **Commercial and Public Domain Solutions: Pro and Cons**

<table>
<thead>
<tr>
<th>Commercial</th>
<th>Public Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Sustainability</td>
<td>✓ Price</td>
</tr>
<tr>
<td>✓ Available Features</td>
<td>✓ Improvable Features</td>
</tr>
<tr>
<td>✓ Community and After-Sale</td>
<td>✓ Community and Software developers</td>
</tr>
<tr>
<td>Service.</td>
<td></td>
</tr>
</tbody>
</table>

- **✗ Portability**
- **✗ External Logic Edition**

- **CalVR**
- **QuestVR**
- **3dvia virtools**
- **MiddleVR for Unity**
1. FrameWork – BlenderCAVE Context

- Smarti-2 system
  - 2 screens / 4 projectors
  - Wave Field Synthesis and Ambisonic
  - 2.6m

- EVE system
  - 4 screens / 7 projectors
  - 2 users adaptive stereoscopy
  - HOA Ambisonic and Binaural

- Laptop
  - 1 screen
  - Stereo and Binaural
1. Framework – BlenderCAVE Context

Open Source

Game Engine

Physic Engine

Developers Community

Animations

Professional Network

BLENDER
1. FrameWork – BlenderCAVE Context

- From Blender to 2013 BlenderCAVE

**2011** BlenderCAVE, by GMRV
- ✔ Video Wall Display
- ✗ “Manual” synchronization
- ✗ Stereoscopy

**2013** BlenderCAVE Improved, by LIMSI
- ✔ Adaptive Stereoscopic Rendering
- ✔ Master / Slave synchronization process
- ✔ External Messages processing
- ✗ Patching Blender sources

www.gmrv.es/~jgascon/BlenderCave/
2. BlenderCAVE – Architecture

- **Embedded OSC API** – Easy communication with the Sound Rendering Engine

**Implemented Classes:**

- **Global**
  - start
  - mute
  - volume
  - configuration

- **Object**
  - sound
  - position

- **User**
  - position
  - HRTF
  - room acoustics

- **ObjectUser**
  - mute
  - volume
2. BlenderCAVE – SRE

- Max/MSP based Sound Rendering Engine Implementation

- **Transparent**

- **Dynamic** object instantiation (poly~)

- **Commutable** Spatialization Engine
3. Usage – Performances

- Work in progress

- Enhance synchronization process: FPS stress-test (980 cubes)

- Complex scenes (Features and Sound)

- Portability
3. Usage – Scene Creation

1. Create / Download a Blender Scene (Content & Logic)

2. Setup / Activate BlenderCAVE rendering

```python
import blender_cave
Blender_cave.run()
```

Configuration file

Any Architecture

or

...
3. Usage – Scene Creation

3. Setup / Activate BlenderCAVE Sound Rendering Engine

```python
import blender_cave
OSC = blender_cave.getOSC()

## Access / set OSC User
user_OSC = OSC.getUser(blender_cave.getUserByName('Binaural1'))
user_OSC.volume('%10')

## Access / set OSC Object
object_OSC = OSC.getObject(scene.objects['Target'])
object_OSC.sound('micro2')

## Access / set OSC ObjectUser Linker
linker = OSC.getObjectUser(object_OSC, user_OSC)
linker.mute(False)
```
Thanks

- BlenderCAVE sources, tutorial and associated Max/MSP Sound Rendering Engine are available at:
  
  http://blendercave.limsi.fr

- Next step: BlenderCAVE integration into Blender official trunk

**Beta Testers are Welcome!**

BlenderCAVE 2013