BLENDERCAVE: A FLEXIBLE OPENSOURCE AUTHORING TOOL DEDICATED TO MULTIMODAL VIRTUAL REALITY

DAVID POIRIER-QUINOT, DAMIEN TOURAINE & BRIAN F.G. KATZ

EADS Astrium Services, Suresnes, France
ETIS/ENSEA - Univ. Cergy-Pontoise, France
LIMSI-CNRS, Orsay, France
david.poirier-quinot@limsi.fr

LIMSI-CNRS, Orsay, France
VENISE & AA teams
damien.touraine@limsi.fr
brian.katz@limsi.fr

Abstract. This paper presents the current state of the BlenderCAVE project, which extends the 3D creation content software Blender and its Game Engine (BGE) to Virtual Reality (VR) applications. BlenderCAVE integrates a complete framework dedicated to Virtual Reality, compatible with the three main Operating Systems for any given VR architecture configuration. Acting as a Scene Graph, BlenderCAVE handles multi-screen/multi-user tracked stereoscopic rendering through an efficient low-level master/slave synchronization process while controlling spatial audio rendering events through OSC and VRPN protocols. BlenderCAVE allows VR users to benefit from the high level scene editing, game logic, and physics engine of Blender as well as Blender's large user community.

Keywords: Virtual reality, scene graph, multimodal, CAVE, open source.

1. Introduction, Scene Graph Editors for Research in VR

Research commonly exploits Virtual Reality (VR) for two kinds of applications: fast development for *contextualized* proof of concepts, and situation simulation. For the latter, more *realistic* Virtual Environments can offer a better sense of presence leading to better results and feedback with full perceptive environments.

Over the last decade, realism has predominantly been associated with graphics. Led by animation and video game industries the scientific VR community has acquired numerous ready to use 3D creation content software that would later provide multi-screen tracked stereoscopic rendering solutions. As for public licensed software, many developers have proposed solutions to adapt open source software to VR architectures, or created their own from scratch, resulting in functional yet seldom supported solutions requiring low-level coding to create rich/interactive VR scenes. See [8, 12] for summary studies on 3D creation software for VR research.

Recent improvements regarding VR technologies have extended realism requirements to other modalities [6], with audio as an obvious immersion enabler. Commercial solutions generally support cross-software communication protocols such as VRPN and OSC to delegate these modalities, or handle them through dedicated internal solutions, often at additional costs. As of yet, no open source scene graph editor has been accepted as a durable solution for multi-modal VR scenes development, typically due to the lack of maintenance on the open source side and the scarce yet demanding research community on the non-public licensed side.

While an optimal VR Suite does not exist yet, one can list many requirements for such. Above all, it should support the three main Operating Systems exploited in VR architectures (Linux, Windows, and Mac OS). Its main development should be based

on programming languages or software with active communities, potentially open source, where independent communities could ensure ready to use SDK and future developments. It should include a main scene graph editor coupled with an expendable audio (or other modality) front end, which can be *independently extended* or removed as need be. It should facilitate simple scene graph creation by *non-programmers*, offer straightforward *scene portability* between architectures, and allow *external developers* to implement new features.

The BlenderCAVE project is an attempt to address these requirements. Benefiting from the graphical quality and scene editing capabilities of Blender, the game logic and real-time performance of the BlenderGameEngine, and the associated physics engine and user communities, BlenderCAVE aims to bring these capabilities to the VR community to provide an easy to use, flexible architecture, capable of handling requirements such as easy cross VR platform support of a given scene, multi-user adaptive stereoscopy, modularity to integrate different multimodal supports, etc. BlenderCAVE has been developed and tested on 2 VR platforms presenting very different architectures.

2. BlenderCAVE, A VR Scene Graph Editor

Blender is a multi-platform open source 3D creation content software [1] with enough functionalities to create photorealistic pictures, high quality animations and, most of all, video games. Blender based games use the real-time Blender Game Engine (BGE) handling various physical interactions through the Bullet Physics Library while general game logic may be defined through blocks and/or embedded python scripts. Blender boasts a large support community, a dedicated professional network [3], and several scene repositories containing plenty of reusable material.

The first instance of BlenderCAVE was developed in 2010 [7], providing multi-screen support (video-wall functionality). Subsequent developments [9, 10] have extended the initial concept to a complete scene graph editor, focusing on multi-platform and multi-architecture interoperability.

BlenderCAVE is a patched version of the original Blender software. Internal routines have been added to handle the three basic functions needed in standard CAVE VR applications: (1) Master/slave synchronization, (2) Adaptive stereoscopic rendering, (3) External message processing. A basic architecture graph is shown in Figure 1. Architecture related parameters (eye separation, screen dimensions and positions, VPRN server address, etc.) are stored in a shared xml file. BGE instances are executed on all rendering nodes. External message processing and mapping into scene graph events are conducted through the Master only.

The core BlenderCAVE modifications to the BGE trunk consist in the addition of a `prerender` method prior to the basic `predraw`, the ability to redefine the projection matrix, and the possibility to redefine the aspect-ratio from a python script, simplifying portability of scenes between architectures. BlenderCAVE routines and patched modifications have been implemented to be as transparent as possible regarding BGE native processes to facilitate integration in the official Blender trunk.

Master/Slave synchronization is carried out at each frame, in the `prerender` method. Synchronization, executed via a python script, inspects every object (identified as being dynamic) in the scene to see if it has changed. Update information is passed form the Master to the Slave nodes before rendering.

Adaptive stereoscopic rendering principally requires changes of coordinate systems and projection operations. Modification of the projection matrix is necessary to take into account tracked users, typically away from the central position, and requires non-symmetric projection matrices, taking into account current head position and orientation. Such variable frustum projections are implemented so as to be computed locally on each graphical node in the `prerender` process.

External message processing represents a cornerstone in any scene graph editor as it largely impacts end users in their scene developments. BlenderCAVE to/from external user defined interactions are implemented and collected in the python script `[sceneName].processor.py`, attached to the VR scene to help differentiate BlenderCAVE from Blender related logic (object displacement, collision triggered events, etc.). Most incoming messages are currently dedicated to user integration or interaction in the VR scene relying on VRPN protocol. The introduced *Processor* python Class (different from here the `processor.py` script) gathers methods to ease VRPN related message processing, granting simple access to controllers and user parameters instantiated in the previously mentioned XML file.

Most outgoing messages are currently intended to add sound to the scene, i.e. update scene graph object properties in the Sound Rendering Engine (SRE) using the Open Sound Control (OSC) protocol. The SRE comprises gathering audio object definitions (sound source, position, etc.) and sound rendering methods in an Audio Programming Environment (e.g. Max/MSP, PureData). BlenderCAVE proposes an OSC module encapsulating three main Classes of messages: Global, Object, and User.

Global messages set general properties like main volume, room characteristics, and rendering architecture configurations. During the course of scene development on a laptop or architecture that differs from the final high performance rendering system, one could use a Global message to load a basic audio engine before porting the scene to the real VR architecture where a corresponding message can instantiate a more sophisticated audio rendering method. An Object represents a simple sound emitter requiring messages to define its nature (sound file, microphone input, etc.), position, orientation, etc. A User encompasses individuals interacting within the scene graph as well as their associated spatialization engines. One User could be defined statically at the CAVE center associated with an Ambisonic/VBAP rendering, while others could be real tracked Users with headphones receiving individual binaural renderings.

Installation Sources and a step-by-step installation guide are available on the BlenderCAVE webpage [2]. The developed SRE, fully compatible with IRCAM's spatialization solutions [5] for Max/MSP, is also available.

Scene graph edition is accomplished using the full power of Blender's creation and animation tools. Audio rendering related events are written in the Processor script. The overall process being architecture independent, most scene development can easily be carried out on standard laptops, before being ported to the actual VR systems. Once installed, basic import of a Blender scene (without tracking or audio rendering) on a VR system requires the addition of only 2 lines of code in an embedded python script.

BlenderCAVE has been developed on LIMSI's VR platforms: SMART-I2 [11] and EVE [4]. Ease of portability of developed scenes between such very different VR architectures has been the goal behind the BlenderCAVE project.

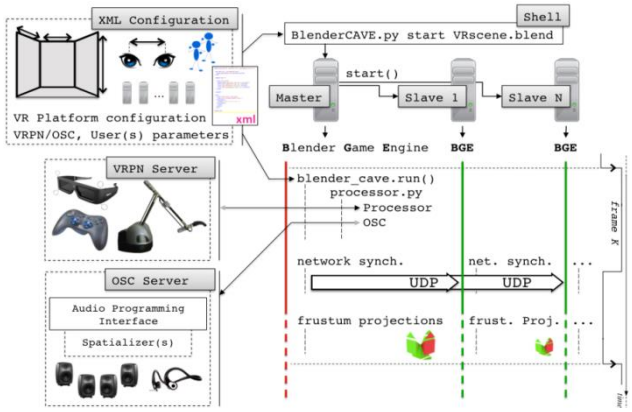

Figure 1. BlenderCAVE architecture graph.

Compatible with Windows, Linux, and MacOS for any CAVE like architecture, BlenderCAVE is released as open source. Built on top of self-evolving software accepted as references in their respective domains, BlenderCAVE is aimed at those in need of a straightforward scene development solution for multimodal VR creation. The next major step is to have the main elements natively integrated in the Blender trunk, limiting maintenance issues regarding future Blender releases. Discussions are underway with Blender community members and we are working with external developers to confirm BlenderCAVE's multi-architecture aspect. Beta testers willing to join the process are obviously welcome. (Work supported by ANR-Equipex DIGISCOPE and an ANRT-CIFRE industry research project with Astrium Services.)

References

[1] Blender. www.blender.org, last check 2013-09-24.
[2] BlenderCAVE - LIMSI. blendercave.limsi.fr, last check 2013-09-24.
[3] BlenderNetwork. www.blendernetwork.org, last check 2013-09-24.
[4] Evolutive Virtual Environment. www.limsi.fr/venise/EVEsystem, last check 2013-09-24.
[5] Spat (spatialisateur). forumnet.ircam.fr/product/spat/, last checked 2013-09-26.
[6] FERNANDO, T., MURRAY, N., GAUTIER, G., MIHINDU, S., LOUPOS, K., GRAVEZ, P., HOFFMANN, H., BLONDELLE, J., DI MARCA, S., FONTANA, M., ET AL. INTUITION IST-NMP-1-507248-2, WP1.B State-of-the-art in VR. *Tech. Rep., IST*, 2007.
[7] GASCÓN, J., BAYONA, J. M., ESPADERO, J. M., AND OTADUY, M. A. BlenderCAVE: Easy VR authoring for multi-screen displays. *SIACG 2011: V Ibero-American Symp. in Computer Graphics* (2011).
[8] JUST, C., MEINERT, K., BIERBAUM, A., AND HARTLING, P. Open source virtual reality. In *Virtual Reality, 2002. Proc. IEEE* (2002), pp. 303–303.
[9] POIRIER-QUINOT, D., TOURAINE, D., AND KATZ, B. F. BlenderCave 3D-s project, OpenSource architecture adaptation to virtual reality research expectations. In *Blender Conf.* (Amsterdam, Oct 2012).
[10] POIRIER-QUINOT, D., TOURAINE, D., AND KATZ, B. F. BlenderCAVE: A multimodal scene graph editor for virtual reality. In *19th Intl. Conf. on Auditory Display (ICAD)* (Lodz, Oct 2013), pp. 223–230.
[11] RÉBILLAT, M., CORTEEL, E., AND KATZ, B. SMART-I2 "Spatial Multi-user Audio-visual Real-Time Interactive Interface". In *125th Conv. Audio Eng. Soc.* (2008), pp. 7609_1–16.
[12] WRIGHT, T., AND MADEY, G. A survey of collaborative virtual environment technologies. *University of Notre Dame-USA, Tech. Rep* (2008).