# PROCEEDINGS

of the Union of Scientists - Ruse

Book 5

## Mathematics, Informatics and Physics

Volume 7, 2010

RUSE

ISSN 1311-9184

**The Ruse Branch of the Union of Scientists in Bulgaria** was founded in 1956. Its first Chairman was Prof. Stoyan Petrov. He was followed by Prof. Trifon Georgiev, Prof. Kolyo Vasilev, Prof. Georgi Popov, Prof. Mityo Kanev, Assoc. Prof. Boris Borisov, Prof. Emil Marinov. The individual members number nearly 300 recognized scientists from Ruse, organized in 13 scientific sections. There are several collective members too – organizations and companies from Ruse, known for their success in the field of science and higher education, or their applied research activities. The activities of the Union of Scientists – Ruse are numerous: scientific, educational and other humanitarian events directly related to hot issues in the development of Ruse region, including its infrastructure, environment, history and future development; commitment to the development of the scientific organizations in Ruse, the professional development and growth of the scientists and the protection of their individual rights.

The Union of Scientists – Ruse (US – Ruse) organizes publishing of scientific and popular informative literature, and since 1998 – the "Proceedings of the Union of Scientists- Ruse".

**SERIES 5**

**"MATHEMATICS, INFORMATICS AND PHYSICS"**

**VOLUME 7**

## CONTENTS

# AN APPROACH TO 3D ON THE WEB USING JAVA OPENGL

**Tzvetomir Vassilev, Stanislav Kostadinov**

*Angel Kanchev University of Ruse*

***Abstract:*** *This paper describes an approach to visualizing three dimensional scenes using Java OpenGL and XML. Several existing technologies are analyzed and their advantages and disadvantages discussed. New file format, based on X3D and Open Inventor, is proposed and a Java application was developed for its visualisation. Using this approach, applets can be developed to visualize 3D objects. The size of the application is small and its downloading is rather fast. The only requirement is that a Java plugin to the corresponding browser is installed on the client machine.*

***Key words:*** *3D visualization, OpenGL, XML.*

## INTRODUCTION

Web technologies developed very quickly in the recent years and penetrated many different fields in human life. More and more applications require visualization of three-dimensional (3D) objects. Different graphics libraries and packages have been developed, like OpenGL, Direct3D, Open Inventor, etc. However, there is no common standard and format for the visualization and storage of 3D objects on the Web.

This paper describes several existing file formats and technologies for the visualization of 3D data on the web, addressing their features and disadvantages. It proposes a new format, based on Open Inventor and XML and describes a web application, based on Java OpenGL (JOGL). The user does not have to install any additional web browser plug-in for the visualization, apart from the widely used Java standard edition, which on many machines is pre-installed. The rest of the paper is organized as follows. Next section describes existing technologies and file formats for 3D visualization of the web. Section 3 describes Java OpenGL, section 4 addresses our approach. The results section shows some images produced by our software, and the last section concludes the paper.

## TECHNOLOGIES FOR 3D ON THE WEB

### *VRML*

Virtual Reality Modelling Language (VRML) [1] is a standard file format for representing 3-dimensional (3D) interactive vector graphics, designed particularly with the World Wide Web in mind. It has been superseded by X3D. VRML is a text file format where, vertices and edges for a 3D polygon can be specified along with the surface colour, UV mapped textures, shininess, transparency, and so on. URLs can be associated with graphical components so that a web browser might fetch a web-page or a new VRML file from the Internet when the user clicks on the specific graphical component. The latest version is VRML97 or VRML 2.0. In order to view a VRML file one should install a plug-in to the internet browser, several plug-ins are available on the internet. VRML has no good integration with HTML and XML and has been criticized by many researchers.

### *X3D*

Extensible 3D Graphics (X3D) [2, 3] is one of the best development efforts in creating and supporting the next-generation Web 3D standard. It is a successor of VRML and is open standard file format and run-time architecture to represent and communicate 3D scenes and objects. The markup language is a XML family member and it consists of a small set of rules which define a structured, text-based syntax for representing data. X3D isn't a language as such (that’s why it is called mark up); rather it is a meta-language, a common syntax that can be shared across diverse standards and data models. It is easy

to learn and use, especially for users familiar with HTML and even non-developers. XML can be used for communication protocol or exchange of raw data. That's why it's child X3D is perfect for transferring graphics data.

X3D language can be validated by DTD or XML Schema and parsed by using DOM, SAX or their derivative parsers. It is flexible for both client side and server side by its compression characteristics. Compressed GZIP format file is 10 to 20 times smaller that normal X3D file. Every application server allows compression of the communication resources by tuning the web container. For example JBoss AS web container allows compressing all transferred contents by setting the HTTP connector property ***compression="force"***.

Language format and semantics are described by Web 3D Consortium. The extension of XML is a combination of VRML Meta data and XML structure. Definition of the language format and data representation is published by Web 3D Consortium as X3D specifications [4]. A lot of 2D and 3D base components can be defined by only one tag: Arc2D, Circle2D, Rectangle2D, NurbsCurve2D, Box, Cone, Sphere, Cylinder, NurbsCurve, NurbsPatchSurface and so on. Another part of the language contains Meta tags for definition of colors, view point, lights and component material. X3D allows invocation of scripts and the Script tag is dedicated for that purpose. Transformation tag is used for all 9 real life types of movements: scale, translate and rotate in three dimensions: by abscissa, ordinate and applicate. That functionality provides viewer flexibility in scene generation and makes it more apprehensive.

In order to view X3D scenes, one has to again install additional browser plug-in or standalone viewer. Usually X3D viewer can also display VRML content.

Xj3D is a project of the Web3D Consortium focused on creating a toolkit for VRML97 and X3D content written completely in Java. It uses Java OpenGL for rendering the images. This toolkit may be used to the import VRML content into a custom application, or to create a fully-fledged browser. If there is java installed on the machine, the installation of the packages can be done automatically, when viewing a web page for the first time. However, the whole system is quite bulky, and it takes time to download and install everything, so the user might not be so patient to wait for the whole process to finish.

***O3D***

O3D is developed by Google[5]. It is an open-source web API for creating rich, interactive 3D applications in the browser. This API is shared at an early stage as part of a conversation with the broader developer community about establishing an open web standard for 3D graphics. One has to install a Google plug-in for Windows or Mac. The Linux support is still at an experimental phase. Different technologies are used on different platforms: Direct3D on Windows and OpenGL on MacOS and Linux. Installation of Java is also required.

All of the technologies, described above, require that the user install an additional plug-in to view the file format. Many of the users nowadays are not willing to do this.

***JAVA OpenGL***

Java OpenGL (JOGL) is a wrapper library that allows OpenGL graphics library [6] to be used in the Java programming language [7]. It is currently being developed by the Game Technology Group at Sun Microsystems, and is the reference implementation for JSR-231 (Java Bindings for OpenGL). It allows access to most features available to C programming language programmers, with the notable exception of window-system related calls in GLUT (as Java contains its own windowing systems, AWT and Swing), and some extensions. It is part of a suite of open-source technologies initiated by the Game Technology Group at Sun Microsystems. 3.0 is the latest version of the API. It is java archive (or jar) that invokes few DLL libraries. The base Open GL C API is accessed in

JOGL via Java Native Interface (JNI) calls. JOGL differs from some other Java OpenGL wrapper libraries in that it merely exposes the procedural OpenGL API via methods on a few classes: GL, GLU, GLUT, rather than attempting to map OpenGL functionality onto the object-oriented programming paradigm. Indeed, the majority of the JOGL code is auto generated from the OpenGL C header files via a conversion tool named Gluegen, which was programmed specifically to facilitate the creation of JOGL. One of the most successful experimental projects using JOGL library is “NASA World Wind Java”. It supplies a suite of open-source components that developers include in their own applications, providing virtual globe functionality to any application that can benefit from it.

It is easy to embed JOGL applets in html page using the JNLPAppletLauncher. The new JNLPAppletLauncher [8] enables the creation and deployment of applets using 3D graphics via OpenGL without requiring the applet to be signed or performing any manual installation of software on users' computers.

## FILE FORMAT AND IMPLEMENTATION IN JOGL

### *File format*

The file format used in this work is XML based and is similar to X3D. However it does not support all features of X3D and also is influenced by some of the tags in Open Inventor file format [9].

### *JOGL Implementation*

The implementation of the file format in Java is structured as the main XML schema file. There is only one main object in the data model. There are lots of java classes which are equivalent to the X3D structure types. Some of them are enumerations that contain specific String values other are only POJO classes. Model is a tree structure with strictly defined type of objects. Class Head contains data about the implemented 3D scene. Its fields are Meta and Component class instances. They define Meta data like other X3D files connected with that one, version and profile of the file. They are optional not as Scene data. Scene contains fields about:

- Viewpoint – contains data about the position, orientation and the angle of view of the view point.
- 3D components – there are few implemented 3D structures – cone, box, cylinder, sphere, curve and patch. All of them are implemented in two modes – wire and solid mode. Switching between modes is made by boolean flag ***solid***;
- LightPoint – custom structure provides functionality for positioning the light point, light intensity, attenuation and colour;
- Transformation components – that meta data allows moving object in the three dimension perspective: scaling, translating and rotating by abscissa, ordinate and applicate. It also provides functionality for setting the center of group of 3D components;
- Material characteristics – consists of ambient intensity field, emissive colour field dedicated to the reflecting light, shininess field, transparency dedicated to the transparent outlook of the components and diffuse colour field.

Curves and patches are generated by using the algorithms of Bezier. One for the curve is restricted for using input arrays not bigger that 100 elements and the result array is again smaller that 100 elements. It is not a standard implementation of customized one that allows speeding up the process of components visualization. The main algorithm uses

binomial coefficients for calculation of inner curve point. The same source code can be used for generation of 2D curves or N dimensional etc. One of the Java tricks used in here is invocation of System.arraycopy that allows manipulation of array elements by using native code. That speed up the process and makes development word easier /for example the same functionality could be achieved by cycles which means more code and of course more potential errors/. Curve is drawn by connecting inner points with lines. If rang of the curve is number between 50 and 100 the shape is smooth and looks nice but of course is drawn slower.

Algorithm for the patches is similar to the one of the curves but is used for U and V direction. Tests run with different input array sizes gives as a result patch made 16 to 20 times faster than ordinary implementation. Patches are drawn with small quads. Each quad is made by setting normal vector of its angles. This is done by approximating the angles around the processed node /vector multiplication/. That is why each spot on the patch has different color, ambient intensity and so on and that makes it more realistic.

Application could be run as simple Swing application or as Java applet. Input X3D file is validated with SAX parser. Required tag attributes are marked with @Required annotation. That Java trick allows compile time and runtime validation of the model and does not overlooks field initialization. Parsing the file is done by using Java reflection. All default values are set as constants and during class instantiation all the fields are initialized with the default values got from XML schema. Abstract class XMLTag contains method for generation of its children. If there is a complex manipulation of instance of child class all the functionality could be overloaded / setAttributes() etc./.

Saving model in file is implemented by invocation of toString() method. It is overloaded from Object class. All child tags are invoked recursively.

**RESULTS**

The system was implemented in Java using Eclipse development environment. Several scenes have been created and the software was tested on several platforms: desktop, laptop, mini laptop (netbook) with the following OS: Windows Vista, XP, Ubuntu Linux and Mac OS X. It was also tested in the two most common browsers: Internet Explorer 6 and 7, and Mozilla Firefox. Some images are shown in the figures below.

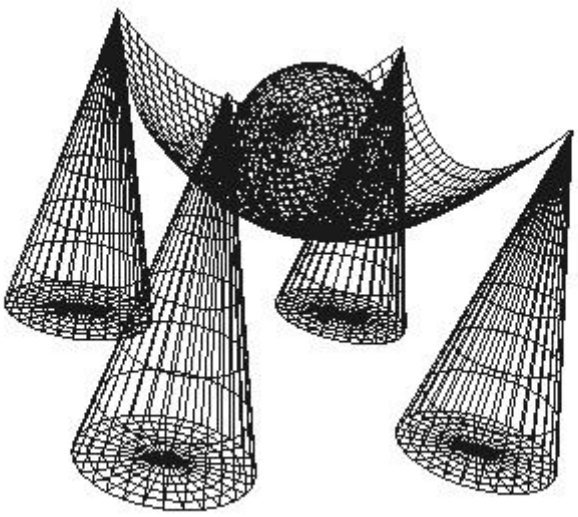

***Figure 1**. Wired components visualized on white background.*

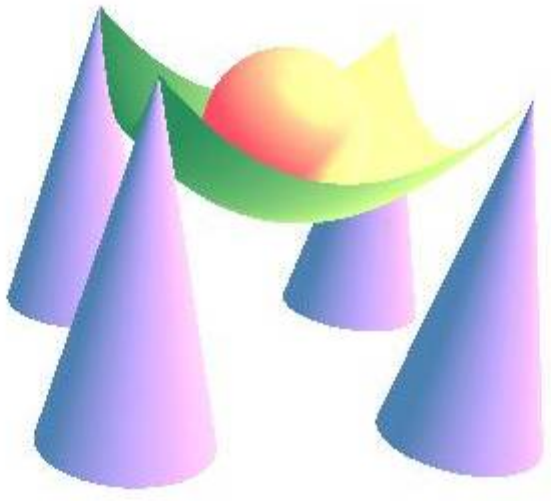

***Figure 2**. Solid components visualized on white background.*

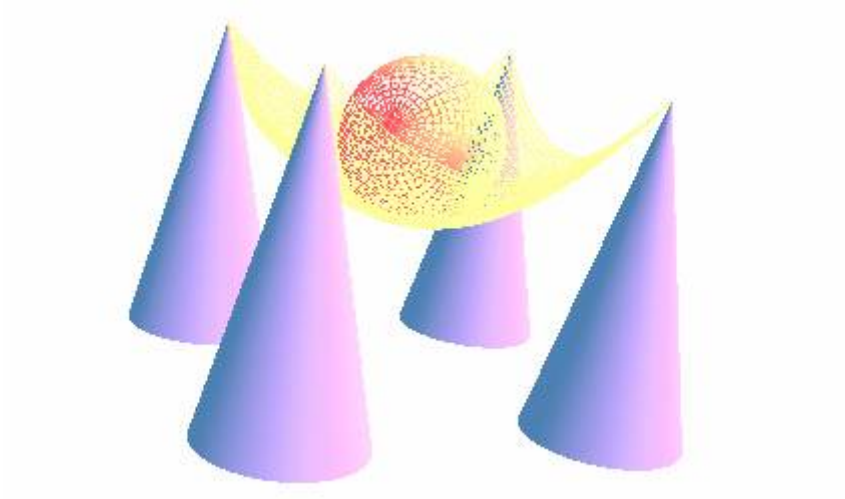

***Figure 3**. Combination of solid and wired components visualized on white background.*

**CONCLUSIONS**

The paper presented an approach to visualization of 3D content on the web using Java OpenGL and XML. The following main conclusions can be drawn:

- X3D is an extensive XML based standard for 3D graphics content. It has too many features and implementing all of them requires a huge application
- Xj3D is a java based engine for the visualization of X3D content, which is too bulky and the visualization of applets using Xj3D is quite clumsy.
- Java OpenGL is a suitable technology for implementing 3D graphics on the web. Due to the fact that it uses the native driver OpenGL calls the resulting JAR archive is small in size and fast in performance. Thanks to the built-in zip support in Java, the 3D graphics file can be zipped, which makes the transfer of large scene rather fast.

**REFERENCES**

[1] VRML, Wikipedia, 2009. http://en.wikipedia.org/wiki/VRML
[2] Brutzman, D. and Daly, L. X3D for Web Authors. Morgan Kaufmann Publishers, Elsevier, April 2007.
[3] What is X3D, Web3D consortium, 2009, http://www.web3d.org/about/overview/
[4] X3D specifications. http://www.web3d.org/x3d/specifications/
[5] O3D API, Google, 2009, http://code.google.com/intl/bg/apis/o3d/
[6] Shreiner, D., Woo, M., Neider, J., Davis, T. OpenGL Programming Guide, sixth edition, Addison-Wesley Professional, 2007.
[7] Java™ Binding for the OpenGL® API Wiki, 2009, http://kenai.com/projects/jogl/pages/Home
[8] Applet-launcher project home. https://applet-launcher.dev.java.net/
[9] Open Inventor File Format. http://web.mit.edu/ivlib/www/iv/files.html

**CONTACT ADDRESSES**

Assoc. Prof. Tzvetomir I Vassilev, PhD
Department of Informatics and Information Technologies
Angel Kanchev University of Ruse
Bulgaria
Phone: (+359 82) 888 475
E-mail: TVassilev@uni-ruse.bg

Mr. Stanislav Kostadinov,
BSc, Graduate student
Department of Informatics and Information Technologies
Angel Kanchev University of Ruse
Bulgaria

# ЕДИН ПОДХОД КЪМ 3D В ИНТЕРНЕТ С ИЗПОЛЗВАНЕ НА JAVA OPENGL

**Цветомир Василев, Станислав Костадинов**

*Русенски университет "Ангел Кънчев"*

***Резюме:*** *Статията описва един подход за визуализация на тримерни сцени с използване на Java OpenGL и XML. Разгледани са няколко съществуващи технологии и са описани техните предимства и недостатъци. Предложен е формат базиран върху X3D и Open Inventor и е разработено Java приложение за визуализацията му. С негова помощ могат да се създават аплети за визуализация на тримерни обекти. Размерът на приложението е малък и зареждането му става достатъчно бързо. Изисква се на клиетна да е инсталиран единствено Java plugin за съответния браузер.*

***Ключови думи:*** *3D визуализация, OpenGL, XML.*

ISSN 1311-9184