Remote Desktop Sharing for Collaborative Work in Virtual World

Rustam Rakhimov Igorevich [1], Dugki Min [2]

[1,2.] DMS Laboratory, Department of Computer Science, Konkuk University, 120, Neungdong-ro, Gwangjin-gu, Seoul 143-701, Republic of Korea

[1.] rusyasoft@gmail.com, [2.] dkmin@konkuk.ac.kr

Abstract: There are many tools to make collaboration easy and accessible. Remote Desktop Sharing tools become one of the key essences of collaboration systems. But applying the Remote Desktop Sharing tools into the virtual world can become even more useful, since the virtual world technologies provide the remote presence applications. Sharing a screen inside the virtual world can derive a lot of novel applications such as collaboration, smart and remote education, and multiple screen control systems. In this work we describe remote desktop sharing tool with the duplex communication channel, where the user can control and visualize the remote desktop inside the virtual world.

[Rustam Rakhimov Igorevich, Dugki Min. **Remote Desktop Sharing for Collaborative Work in Virtual World.** *Life Sci J* 2014;11(7):730-735] (ISSN:1097-8135). http://www.lifesciencesite.com. 107

Keywords: Remote Desktop; Virtual World; RealXtend; Virtual World Viewer; Collaboration

1. Introduction

Collaboration is vital in problem-solving. Creating collaborative environment will increase the effectiveness of finding solution to defined problem. Especially, participators of collaboration are located in far distance. For example, dislocated different majored specialists can gather in a collaborative environment to achieve solution for a problem. Education and sharing knowledge can come as another simple and a conventional example of collaboration. Teams that work collaboratively can obtain greater resources, recognition and reward. Particularly when facing competition with finite resources. In this work we talk about providing collaborative environment inside the virtual space, where distantly located people can see each other in their avatar representations. If the avatar is under real time direct control of remote user, the presence ratio of remote user can be particularly increased. Main purpose of this work devoted to provide collaborative interaction between users who want to use remote desktop in a virtual environment. Remote desktop inside the 3D virtual world can be used in many useful purposes, such as in business meetings, educational and many other similar purposes where the desktop of remotely located PC's are shared.

This paper contains our work results and implementation results about Remote Desktop Screen inside 3D Virtual World.

In the second section the related works and references are mentioned. Section three contains the overall process of embedding Remote Desktop Sharing tool into Virtual World is described in detail. Mouse pointer mapping and rotated object calculation processes are described in a section four. Section four is one of the core ideas of the paper. The solution suggested in Section four can be considered as light hack. There were no ordinary suggestions and tutorials that is the reason why hack method was exploited. Section five contains experimental results and the analysis of those results. At the end paper ends with a short conclusion and possible impact and out comings from this research.

2. Related Works

The virtual reality service is built on the base of client-server technology, where the server provides virtual environment. The client receives all the information about virtual world from the server and draws it for the local user. Those processes called rendering the virtual world. Based on rendering client can see virtual space environment. Rendering is one of the most important issues in virtual reality related area, even nowadays it is a research issue and many commercial and non-commercial organizations are still working on it. The main questions related to rendering are how fast and how clear virtual reality can be described in client side. In some cases it can depend on server side also but mainly it is the client's job. There are some experimental conclusions. When Second Life-viewer and a RealXtend - viewer were sequentially connected to the same Virtual world server (in our experiment case OpenSIM server), and their result was different in graphics. The main reason for that is Second Life uses OpenGL graphic library [1] for rendering virtual world, but RealXtend-viewer use OGRE graphic library [2], that's why RealXtend-viewer has way better quality about describing virtual world.

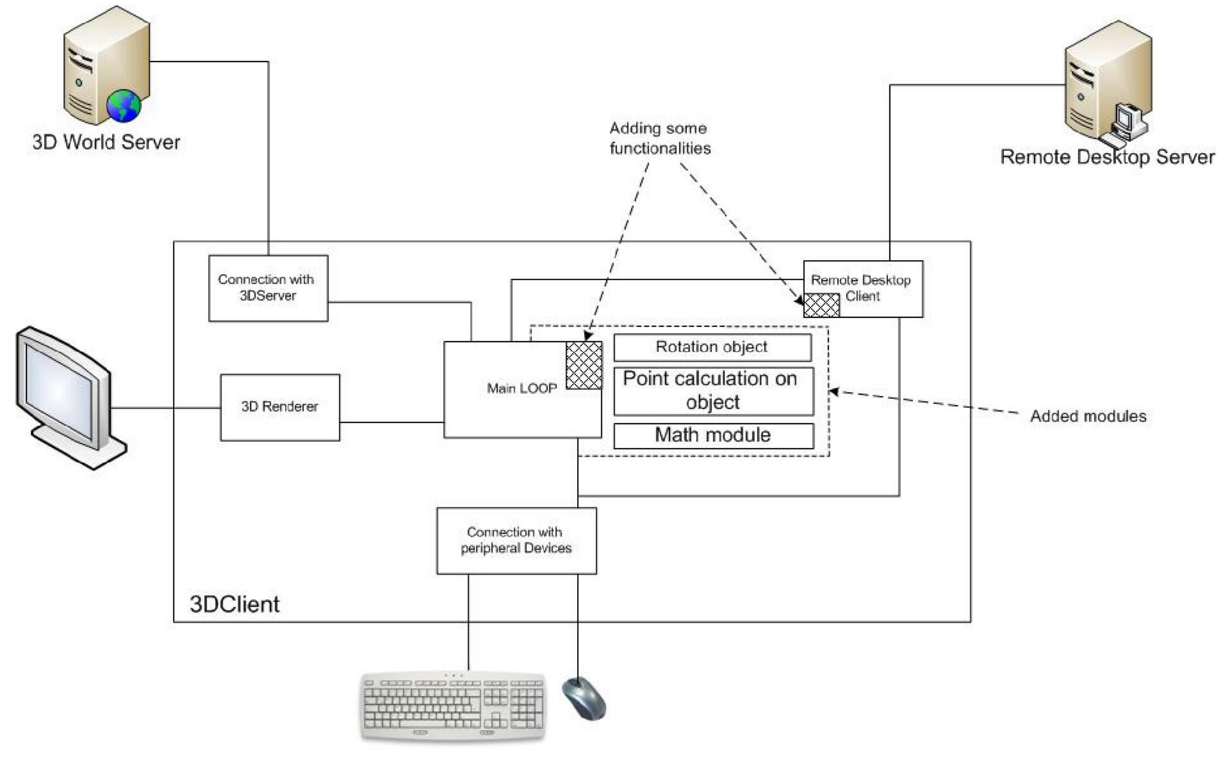

Figure 1. Overall Architecture of 3D Client(Virtual World Viewer) and added Remote Desktop Control Functionalities.

In the [3] the authors have presented VNC Ultra-thin client system, based on a simple display protocol. It provides access to computing environments from the remote computers in order to control available hardware and software resources.

In [4] advantages and disadvantages of the universities in a virtual environment are discussed. It compares Virtual University and traditional universities' opportunities. They are giving list of advantages of Virtual University such as time-place flexibility, cost effective for learners, potentially available to global audience, unlimited access to knowledge and so on.

The authors from [5] have given an equation where the interactivity and immersion together affects to the engagement quality of the learner to make study process more active. They have introduced a theory that interactivity and immersion values are multiplies to give a best engagement result.

3. Embedding Remote Desktop Sharing Tool into Virtual World.

Embedding Desktop sharing tool into the virtual world descends new sort of novel applications, such as: collaboration, smart and remote education, and multiple screen control systems. Virtual Conferences can be held inside the virtual 3D world by sharing one big screen where the presenter could demonstrate its prepared work for the crowd. This experience is not new for the virtual world experts and users.

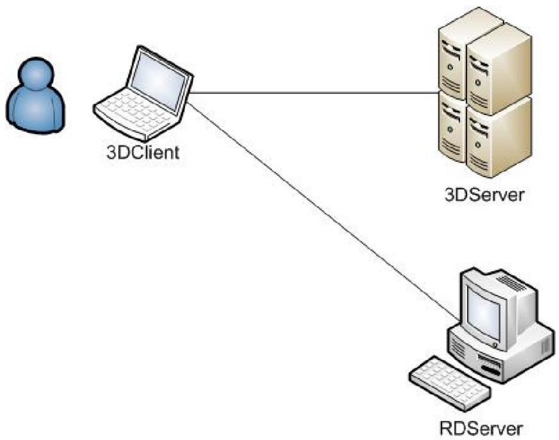

Figure 2. Overall Deployment of Remote Desktop Sharing Tool integration into Virtual World

As it is illustrated in Figure 2, the 3DClient (or Virtual World Viewer) is an application with necessary libraries installed on user PC. 3DClient is interacting with 3DServer (or Virtual World Server) and RDServer(Remote Desktop Server) to provide for user 3D world environment and remote Desktop services in virtual environment. 3DClient handling mouse and keyboard tools to send their events to 3DServer or RDServer depend on events types. Events can be distinguished as a mouse events and keyboard events. 3D Client (or Virtual World Viewer) application is complex and contains many subprojects, and Remote Desktop Client is one of them. Specific issues to control remotely located desktops are enlisted in following section of this work.

As it was already mentioned 3DClient is one of the most complicated parts of Virtual World technologies chain. Even though the Virtual World Server here just demonstrated as a one server but in a real systems it contains mass of distributed computers. They have been represented as a grid, even basic OpenSim Servers has a grid configuration where the multiple Servers can be joined together to recreate the complicated Virtual World. But 3D Virtual Worlds Client embraces dozens of sub projects. The complexity of the 3DClient (Figure 1) grows by the requirements of the User. Requirement list contains mostly quality related features such as speed, quality of 3D meshes, rendering, streaming, security (message exchange and authentication) and other kind of additional features. Additional features usually implemented as a plugin. For example to apply haptic controllers or simply controlling avatar with the special kind of controllers new plugins should be implemented. Remote Desktop Client is the sort of additional plugin. RealXTend developers have implemented Remote Desktop Client feature by embedding TightVNC open source project. It was good trial but it didn't work well and wasn't complete work. They have faced the problem related to control the Remote Desktop Server from the virtual world. TightVNC is the open source implementation of VNC protocol. Our main goal was understand the internal architecture of the VNC client and 3DVirtual Client. VNC Client does not have complex architecture, and quite simple implementation algorithm. As it was mentioned understanding 3DClient implementation architecture was more challenging and tedious work. The reason is that there was no clear explanation about the internal source code. The source code analysis job was performed, while most of the remote desktop related functions and HCI (Human Control Interface) parts of code were analyzed. As a starting point to analyze the source of human interface we have tracked the mouse pointer. Especially it was key factor of tracking mouse events, such as hovering on top of specific primitive object. There are multiple and complex layers in the source code that are handling mouse pointer inside the virtual world. Hovering event functions were key factors to find the control embedding points into the source code.

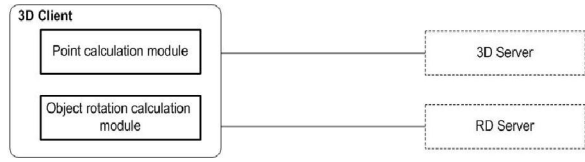

Figure 3. Core Interconnecting Modules of Collaborative Screen Sharing in a Virtual World

The core of this work can be described in a four blocks interconnecting to each other (Figure 3). 3D Server generates the virtual 3D world environment. RDServer is the computer whose screen should be shared and rendered inside the virtual world. The 3D Client machine is the block which connects to both servers and merges their contents to provide user with the screen sharing experience inside the virtual world.

4. Point and Object Rotation Calculation Modules.

The main purpose of the point calculation module is mapping mouse point which is located on the virtual object to remotely located desktop. For a mapping mouse pointer to the remote desktop pointer, calculation O (0,0,0) point on top of virtual object is important. Because O (0,0,0) point is served as a relative point for mapping mouse pointer to the remote desktop. For the calculation of coordinate of the mouse pointer on new defined O (0,0,0) coordinate system, global coordinate of the mouse pointer of the object should be used. We have decided to use camera ray as a solution to gain necessary global coordinate parameters. Virtual camera has a NxM ray points, which starts from the camera's position and ends in a frustum defined distance. Every pixel ray intersects with some object located in the frustum area. Object where the mouse pointer hovers can be extracted from the objects array. By implementing specific functions to the mouse pointer we can extract mouse hovering object from the bunch of other objects. When object id was extracted it becomes possible to access object's properties.

Extraction of object property gives us an advantage to calculate the O (0,0,0). To do this object coordinate and scale data is needed. By dividing scale vector information to 2, we can get the central point of the prim object. Since the coordinate of object points to the center of an object, the value of the prim object can be subtracted to the half of the scale.

O.x = object_coord.x - scalevector.x/2;
O.y = object_coord.y - scalevector.y/2;
O.z = object_coord.z - scalevector.z/2;

here: O.x,y,z – values are global coordinates of 0(0;0;0) point, object_coord.x,y,z – global coordinates of 3D object, scalevector.x,y,z – size of 3D object.

When O (0,0,0) point was found next step is simply extraction of difference of x, y coordinates. Since primitive object does not have any rotation angle, the plane which is organized by mouse pointer and 0 (0,0,0) is collinear and lays on top of the surface of the prim. If primitive object has even small rotation angle then additional Object Rotation operation should be applied.

Object Rotation operation contains simple rotation of O(0,0,0) point, that are could be optimized by necessity. To rotate 0 (0,0,0) point simple rotation algorithms can be used, and the rotation degree can be extracted from the object property list.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \end{aligned}$$

New rotated O (0,0,0) coordinate can be calculated every time, or just can be saved as one of the properties of the remote desktop object. In optimal manner it could be better to save O (0,0,0) coordinate in somewhere object properties. Changes in O (0,0,0) value can be done only when object movement or rotation procedure event is triggered.

Those algorithms were added as an additional function to the main loop of the 3D Client module. To perform those mathematical operations it is possible to use existing mathematical functions from the main source code of the Main Loop. Otherwise those codes can be implemented as additional mathematical operations. The advantage of adding new property for the object that keeps those new calculated values would be even more optimized. Because when the multiple users on the same space are sharing the same screen object they all would share same value. That means each client doesn't have to recalculate those rotation related values on their machines. It would be calculated only once, at the beginning of initialization of the object. And it will be recalculated every time when the position and location of the object is changed.

In the next step ratio of x,y coordinates should be derived in order to send it to the remote desktop server. To implement remote desktop collaborative environment we add additional modules, rather than designing everything from the scratch. Implementation flowchart illustrated in Figure 4, can be applied to other virtual world implementations. Mouse and Keyboard events can be intercepted from the Client machine. Location of mouse pointer will activate point calculation and rotation algorithm at once. Hovering mouse cursor on remote desktop object will cause trigger two-way communication functionality. Main workflow of this added function is plugging in and running additional three modules. After that these three modules will calculate x, y coordinate for the remote desktop. With the result of x,y coordinates another function located on Remote Desktop client module will be invoked.

When it calls that function (DMSWndProc) from Remote Desktop client module x,y coordinates will be sent to the Remote Desktop Server. There are additional coordinate convert operations were also performed on the Remote Desktop Client module. To keep it simpler and easily convertible we have kept ratio based coordinate information. Ratio based coordinates contains values between [0;1].

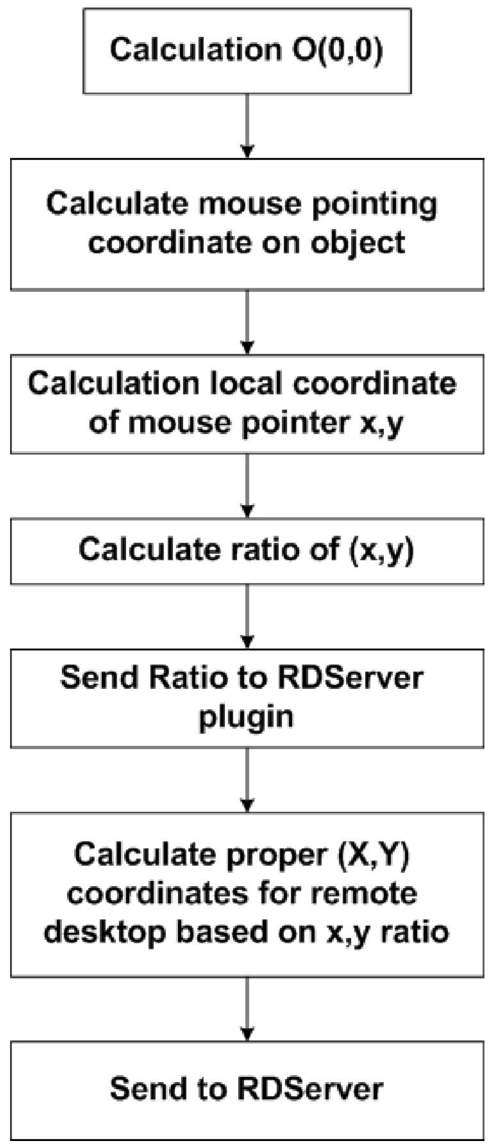

Figure 4. Flowchart of Collaborative Virtual Environment

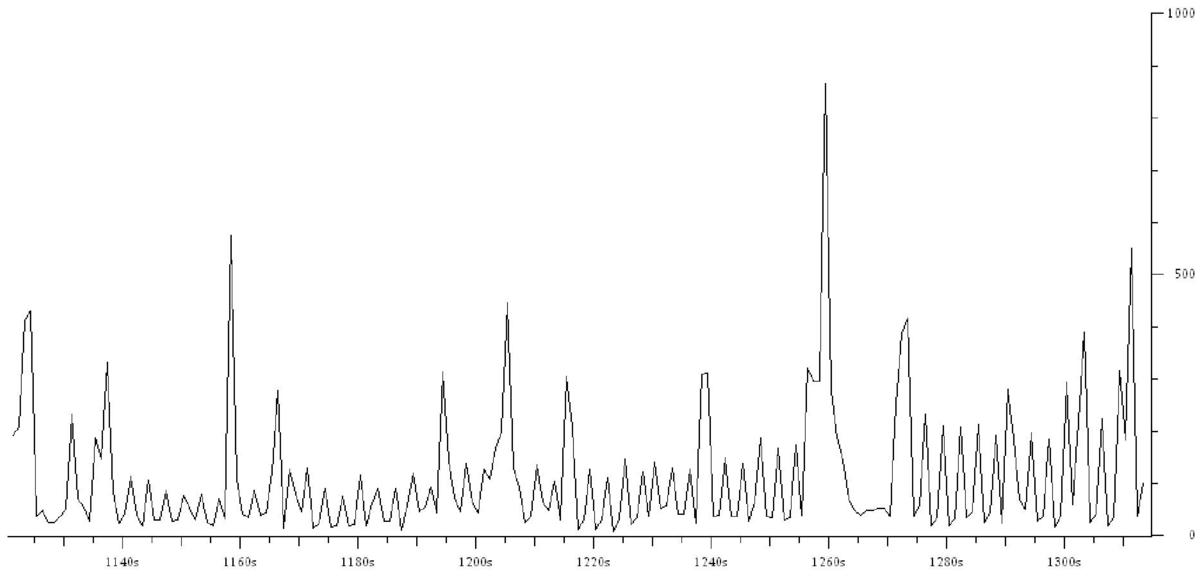

Figure 5. Remote Desktop Server Performance with seven real-time connected users

5. Implementation and Experimental Results

Open Source project RealXtend Viewer was selected as an implementation platform for Remote Desktop Sharing screen inside the virtual world. Main reason of the selection this viewer among others is that, it is open source under GNU license, and it already has a VNC implementation. Developers of RealXtend viewer already have implemented VNC, but they made it only simplex way, which means remote desktop can be just shown in the virtual world. From the virtual world it was impossible to affect remote located desktop. On the other hand our goal is implementation remote desktop to make it full duplex connected. Implementation process was followed steps mentioned in previous section by applying point calculation and object rotation calculation operations.

Mouse movements and keyboard events were implemented and tested. The demo screen is illustrated in Figure 6, contains how the virtual world clients connects to remote desktop server which installed on the laptop located on the right side. It shows how movements and actions on the virtual world primitive object manipulates screen of laptop on the right.

The security side of the system relies on Remote Desktop Server authentication service. That means when the Remote Desktop Client object embedded primitive object is clicked the authentication window will be showed. User can pass password to the Remote Desktop Server through that authentication dialog window. Security could be improved by applying different algorithms between VNC Client and Server. It is well known fact that original security of VNC is not sufficient, since by default they use only 8 character based password secure key. Another security issue is that by default frame buffers do not support encryption, which means all screen information flows as an open binary message structure. We have tried to secure it by opening SSH channel between VNC Server and Client. That way the secure communication between nodes could be installed. The experiment to create SSH secure channel between Virtual World Viewer and Remote Desktop Server was installed through using PuTTy Tool. It is a good and easily manageable tool for creating secure logical channel

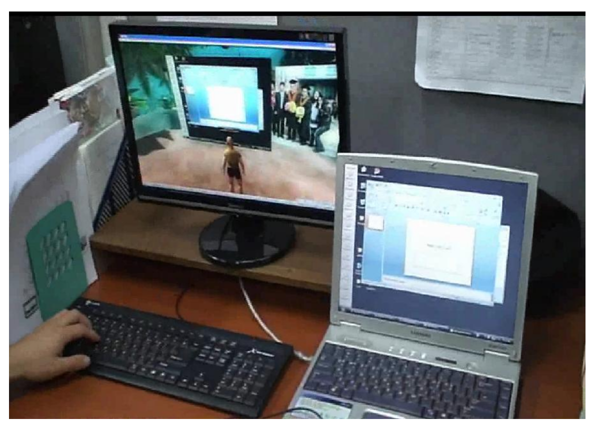

Figure 6. Demo View.

Another advantage of this remote desktop implementation is that the user doesn't have to push some additional button for locking the screen.

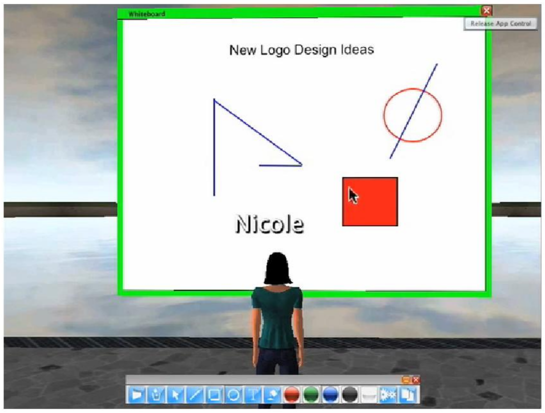

Figure 7 Avatar Interacts with Object to Draw On it in OpenWonderland

Solutions such as OpenWonderland [W1], requires locking button, when avatar interacts with whiteboard objects (Figure 7). Sometimes it makes the user uncomfortable, especially when the user interacts with two or multiple boards simultaneously. Our implementation locks movement of the avatar when the mouse hovers on top of the remote desktop object. The controller lock is triggered by OnMouseHover event function from the 3DClient. Some additional implementations were performed for specific buttons. Because the software architeture of the RealXtend (or Second Life Client) does not tracks all keyboard events equally. There are some special keboard buttons that are triggered and captured in different layer of the source code. This kind of design mostly was related to the accelaration and optimization of some parts of the 3DClient Application.

Experiment on the overload performance of remote desktop was checked by connecting multiple users. Since the remote desktop server transfers desktop's screen view to the Virtual World Viewer directly, where the desktop view will be mapped onto a virtual object. This solution requires a powerful remote desktop server, because it should care all the users who want to use a remote desktop service. Multiple users' connection to remote desktop proportionally effects on performance. I/O performance diagram of remote desktop server with seven connected clients were illustrated in Figure 5.

From the comparison of two and seven client connections it can be concluded that network i/o performance increases twice. Since there is no any pre-defined strict policy about screen control, users can access to screen anytime. It will cause the same result when clients are trying to control same PC, mouse and keyboard simultaneously by using ordinary remote desktop sharing tools. Remote Desktop Server performance diagrams contain some extreme points, and the main reason of that is because of user interaction with the screen. When a user interacts with virtual screen intently, the remote desktop server receives more packets compared to the situation with no interaction.

Conclusion

Applying the Remote Desktop tool into the virtual world is quite important and useful for the future collaborative systems. In this work Remote Desktop Sharing tool for Virtual World was designed and implemented. Disadvantage of the previous implementation of the Remote Desktop tool for virtual world was considered. The main key issues are related to embedding Remote Desktop into virtual world was considered in this work.

Acknowledgements:

This paper was supported by Industrial Source Technology Development Programs funded by the Ministrty of Trade, Industry & Energy of Korea.

Corresponding Author:
Dr. Dugki Min
DMS Laboratory, Department of Computer Science, Konkuk University, 120, Neungdong-ro, Gwangjin-gu, Seoul 143-701, Republic of Korea
E-mail: dkmin@konkuk.ac.kr

References

1. http://www.opengl.org/about/. http://www.opengl.org. [Online] Gold Standard Group, 1997.
2. http://www.ogre3d.org/about. http://www.ogre3d.org. [Online] 2000-2009.
3. Richardson, T. ; Olivetti & Oracle Res. Lab., Cambridge, UK ; Stafford-Fraser, Q. ; Wood, K.R. ; Hopper "Virtual Network Computing", Internet Computing, IEEE Jan/Feb 1998:33-38.
4. Maryam Mirzakhani, Hossein Ashrafzadeh, Azadeh Ashrafzadeh., "The Virtual University: Advantages and Disadvantages." s.l.: 4th International Conference on Distance Learning and Education, 2010.
5. KARL TAPP, TONY O' DRISCOLL, "Escaping Flatland", Guild Research 360 Report on Synchronous learning systems.

5/26/2014