Design and Implementation of Dynamic and Collaborative Virtual Reality Using Croquet

Riri Fitri Sari, Abdul Khayyi
Department of Electrical Engineering Department, Faculty of Engineering
University of Indonesia, Kampus Baru UI, Depok 16424, Indonesia
riri@eng.ui.ac.id abd_khayyi@yahoo.com

Abstract: This paper reviews the issue of creating a dynamic and collaborative Virtual Reality (VR) environment. In this work we use Croquet, an open source project which enables virtual reality objects to run collaboratively on a network. We review other virtual reality programs such as VRML, in which 3D object can be shared via web using HTML script. Different from VRML, Croquet’s 3D object can be shared via web using web object located in web server.

Croquet was built on Squeak, which was developed using an object oriented language-Smalltalk. Supported by OpenGL, Croquet can be use to make two dimensional or three dimensional virtual realities. Using croquet, virtual objects can be programmed to be interactive and attractive.

In this work, we perform some experiments using croquet and 3D tools such as 3D Smax to create some objects. We analyze the results by using Ethereal to capture data packets and the average sum of Bytes per second etc. From the experiment to access two different remote spaces, we obtained that the average of 15.382 and 970.159 packets per seconds are required to be sent. The difference is due to the different complexity of the components complexity which built both of the remote spaces.

2. VIRTUAL REALITY AND CROQUET

2.1 VR definition
Aukstakalnis and Blatter [1] defined “Virtual Reality is a way for humans to visualize, manipulate and interact with computers and extremely complex data”. Other definition by Pimentel [2] stated that virtual reality is “a response to user actions, real-time 3D graphics, and a sense of immersion”. A more complete definition by Cruz-Neira [1] stated that “Virtual reality refers to immersive, interactive, multisensory, viewer-centered, three-dimensional, computer-generate environments and the combination of technologies required to build these environments”. This means that in VR user interact and experience the system.

2.2 VR Classification
Based on those definitions, VR can be classified as:
• Desktop Virtual Reality
This simplest VR system uses conventional computer monitor to display virtual reality.
Figure 1 Desktop Virtual Reality with a pair of 3D Glasses, a joystick, and a space mouse [1]

Figure 1 is an example of Desktop Virtual reality. A person using 3D glasses, a joystick and a space mouse operates VR via his monitor.

- **Immersive Virtual Reality**

  In Figure 2, the immersive Virtual Reality system puts the entire user inside the application. Immersive systems are often equipped with Head Mounted Displays (HMD), or used inside a CAVE where the user is surrounded by a 360-degree picture.

2.3 Croquet

2.3.1 Croquet definition

Croquet is an open source project which supports the development of dynamic and applicative virtual environment. It is a combination of computer software and network architecture that supports deep collaboration and resource sharing among large numbers of users.

It delivers compelling 3D visualization and simulations and focussed to enable multi-user peer-to-peer collaboration.

Croquet is not a typical 3D application program such as 3D S max, Maya, AutoCAD 3D, ArchiCAD etc. Croquet has been supported by 3D development because it was built on OpenGL engine which is used as a graphic engine for 3D programs and games [5, 6, 7, 10]. The object oriented programming methods allows significant flexibility for the design and the nature of the protocols and architectures being developed.

2.3.2 Space and Portal

In croquet, space and portal are the most important component which differentiates croquet with other virtual realities. A Space is a place. In Croquet, a space is a container of objects, 2D object and 3D object, which includes the user which is represented as “avatar”. Portals are a 3D spatial connection between spaces. If two portals in different spaces are linked together, then they can be viewed from each other. Figure 3 displays a portal in Croquet that connects a space to other space.

3. DESIGN DYNAMIC APPLICATIVE COLLABORATIVE VIRTUAL REALITY USING CROQUET

3.1 UML design (class diagram)

Figure 4 shows the structure of the class in the system.

3.2 Creating local space and remote space

Space is the container of all objects. Croquet has a component to handle graphic programming. Its derived from Morphic class. One of the classes of Morphic that handle space is TeapotMorph. In order to create a space, the InitializeDefaultSpace should be override and defined appropriately. Space can be divided in to local and remote space. The differences are the location of the storage and access privileges. Local space is the private space. It means that only a user can access the space, whereas other users connected in network can not access it. Physically, the space is located in user computer hard disk. Remote space is a non private space and is physically stored in the web server as a web object and can be accessed by other users connected to the network. Remote space that will be stored in web server should be a web object.

3.3 Creating Portal between Local and Remote space
In order to create portal and connect local space with remote space, we need to be connected to the local space and link the second portal to the remote space. Then we link other portals. Portal can be made by defining an object of TPortal class.

3.4 Creating 3D object
Croquet has the feature to enable the making of a 3D objects using available classes. The objects are ordinary 3D objects in geometry shapes or other more complicated 3D object. On the other hand croquet can support the use of 3D objects built using other tools such as 3D S Max. Some file types such as “ASE” file from 3D S Max, “WRL” file from VRML, and many more such “MDL”, etc can be used in Croquet.

3.5 Event Handling of 3D Object
3D object will response to the initiated event from user, located at derived class from TGroup class. That derived class has three basic methods, i.e. handlesKeyboard, handlesPointerDown, and isComponent. This defines the 3D object as a component that will listen to event while user press keyboard or mouse button.

Defining which 3D object will response the user’s event is done by initialize method. This method is a constructor from the class of Smalltalk programming. Different from the constructor of java programming, where constructor’s name is same with the class’s name, constructor of Smalltalk is defined with the name initialize. Definition of event from user is done using keyDown method. On creating event handling program of this croquet, keyDown defines some keyboard character buttons which will give some behaviors if the buttons is pressed.

3.6 The 3D object animations
The 3D objects which will have animation should be derived from TGroup class. This class has three basic methods mentioned previously. In our work Croquet, a 3D object of plane has been animated. It can fly and turn around on the sky. For handling this turning around movement of Pesawat, a spin method has been used.

4. CROQUET IMPLEMENTATION
The 3D programs generally require high powered computer resources with the latest specification. This is also the case with Croquet. To enable the VR environment to work well, a state of the art computer resources with high specification of VGA card, memory and processor speed is required. Software called Jasmine has been used in this work to run Croquet [10].

4.1 Local Space and Remote Space
Private local space displays a virtual environment which has a lot of component inside. Therefore the loading process will be relatively heavy. Private local space consists of some components such as portal, window, floor, light, sky effect and 3D object. Private local space can be explored by an avatar. Figure 5 displays a Private local space. From figure below, we see many components like house, toy soldier, sky effect, Makara (University of Indonesia’s logo), portal, and so on.

The components of remote space are not as a many as the private local space, and the access is not as heavy. The components of remote space are dominant for applications that can be used together by all user connected on network. There are two remote spaces in this work, i.e. remotespR.tea and remSp2.tea. Remote space should be a web object with “TEA”, “ASE”, “MDL” or “WRL” files. remSp2.tea file is a remote space which has a horizon component beside general component such as floor, light, sky etc. The remoteSpR.tea file has a legoic 3D object (like robot) which is different from general components.

4.2 Portal
Figure 7 describes condition in which an avatar stands up in front of portal in private local space to enter remote space remotespR.tea. Figure 8 displays when avatar stands up in front of portal to leave remote space.
4.3 The 3D object, animation and event handling

To make an attractive virtual environment attractive, some 3D objects are programmed to enable animation or to accept response of user’s event. The object of an aeroplane in our work can fly around on the sky and the Toy soldier can move forward, backward or sideward by pressing current buttons of keyboard.

Figure 9 shows 3D object of Toy soldier before the event of user response (object which stands up next to door of the house). Figure 10 displays the toy soldier after the response of the user’s event (stands up close to the avatar).

4.4 Squeak applications

Figure 10 and 11 shows some squeak applications being made in this work such as the response event testing and creation of a calculator in the VR environment.

4.5 Peer To Peer Croquet

Peer to peer Croquet connection can be experienced by two users connected via the network. Both users should be connected by setting the name, party name, and the network to enable the collaboration (refer to Figure 11). Once it is connected both user can meet each other as shown in Figure 12.

5. EVALUATION OF PERFORMANCE TO ACCESS REMOTE SPACE

On the accessing remote space, at the first time, downloading will be done automatically when the object of a remote space is being rendered. The access speed depends on download speed.

The evaluation of the number of packet sent from a computer with a remote space in its web server to other computer is conducted from the starting time in accessing a remote space, rendering and until it can be displayed perfectly.

5.1 Evaluation on accessing remote space remSp2.tea

Table 1 shows the average number of packets sent per second on the access of a remote space.
5.2 Evaluation on accessing remote space remotSpR.tea

On accessing remote space - remSp2.tea and remotSpR.tea, the result of packet average per second shows different value. On accessing remSp2.tea, the average of ten of packet average per second is 15.382 and remotespR.tea is 970.159. The difference is caused by complexity of components which built the remote space. remotSpR.tea has less complexity than remSp2.tea. To obtain a range of middle value from the data used, we should use statistical method with confidence interval parameters. The confidence interval is formulated in [9]

\[
x \pm t_s y
\]

\[
x - t_s y < \eta < x + t_s y
\]

Where \( s_y \) is standar error, which is formulated [9]:

\[
s_y = \frac{s}{\sqrt{n}}
\]

Variable s is a standard deviation. The confidence interval calculation is done with confidence level 95%. From the calculation, we obtained the confidence interval of packet average per second on accessing remsp2.tea is 14.97113 < \( \eta \) < 15.79287 and on accessing remotespR.tea is 951.3022 < \( \eta \) < 992.0158.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>The average Packet per second</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15.005</td>
</tr>
<tr>
<td>2</td>
<td>15.409</td>
</tr>
<tr>
<td>3</td>
<td>14.453</td>
</tr>
<tr>
<td>4</td>
<td>16.538</td>
</tr>
<tr>
<td>5</td>
<td>14.989</td>
</tr>
<tr>
<td>6</td>
<td>15.162</td>
</tr>
<tr>
<td>7</td>
<td>15.911</td>
</tr>
<tr>
<td>8</td>
<td>16.302</td>
</tr>
<tr>
<td>9</td>
<td>15.016</td>
</tr>
<tr>
<td>10</td>
<td>15.135</td>
</tr>
</tbody>
</table>

5.3 Future Work

We plan to run Croquet on a network by accessing a croquet script. On work with Croquet will give us a richer experience in implementing a user friendly VR environment using an Integrated Development Environment (IDE).

6. CONCLUSION

In this work we review the Virtual Reality (VR) specifically using croquet to produce an immersive, interactive, multi-sensory, viewer-centered, three-dimensional, computer-generated environments. The future work will include more users and remote spaces to be connected via the network.

REFERENCES