Chapter Objectives:

- Introduce Web3D and distributed interactive simulation in a networked environment
- Compile a list of Web3D tools

17.1 Web3D

Web3D tools are graphics tools that deliver graphics through web browsers over the Internet. Many web-based 3D modeling, rendering, and animation tools emerged recently. It is not difficult to foresee that Web3D will be the future of education, visualization, advertising, shopping, communication, and entertainment. Currently, most Web3D tools are individual plug-ins for a general web browser. Most of the tools, such as VRML browser and the Java3D programming environment, are built on OpenGL or Direct3D. Individuals and companies have developed many new Web-based 3D API engines (similar to Java3D) on top of lower-level graphics libraries.

17.2 Distributed Interactive Simulation

In addition to Web3D tools, networked virtual environments have been developed to simulate highly interactive activities in critical mission training. Unlike Web3D tools, which develop applications for independent users without real-time constraints, distributed interactive simulation (DIS) systems immerse networked users in the same virtual environments across the network in real time.
In a distributed interactive virtual environment, multiple nodes (computer simulators) at different locations have the same entities (objects) and activities (behavior). NPSNET (www.npsnet.org/), MUVE (http://www.virtual.gmu.edu/muves/), ExploreNet (http://www.cs.ucf.edu/ExploreNet/) are examples of such kind of environments. Today, most DIS environments call for a centralized infrastructure to control and manage information. The High Level Architecture (HLA) with a Run Time Infrastructure (RTI), which builds upon and generalizes the results in DIS, is advocated by the US government. HLA allows for nodes to coordinate the execution and exchange of information through the RTI.

There are two layers of communications in DIS/HLA: communicating between the multiple nodes at the network communication layer and synchronizing physical activities on top of the network communication. The low-level communication protocol determines the efficiency and reliability of the message transmission. The high-level time synchronization is vital for achieving fast DIS. Better solutions to these problems will improve the usability and speedup the simulation in DIS/HLA. For example, in Doom — a simple distributed multi-player game system — each node simply broadcasts the location of each entity that it maintains. Communication delay for time synchronization is ignored.

17.3 Synchronization in a DIS

When implementing a DIS/HLA, the commonly accepted approach to limit the rate of simultaneous updating of multiple nodes on a distributed simulation network is termed Dead Reckoning. Dead Reckoning is a method of position/orientation estimation that predicts and approximates the behavior of simulated entities among the networked nodes. Dead Reckoning’s estimations eliminate the need for sending every change in position/orientation until a pre-specified threshold is exceeded; then, the behavior of the entities that changed is updated by new data sent across the network. In a DIS/HLA, an entity is either an object or a ghost. An object is the master entity running on its host node where the user controls its activities. Its copies running on other networked nodes are called its (Dead Reckoning) ghosts. The user has no control over the activities of ghosts, which proceed according to their object’s original parameters (position, orientation, velocity, acceleration, etc). A ghost is running on the host node as well, so that Dead Reckoning algorithm can compare the parameters of the object and the ghost on the host node to estimate the errors in the networked ghosts, and update the ghosts with the object’s parameters if necessary.