

## **JavaMOO Technical Description**

JavaMOO is a framework for building virtual worlds used for educational purposes (Figure 1), intended to replace LambdaMOO which served as the implementation platform for WWVIC's original suite of educational virtual environments.

While LambdaMOO provides the essential functionality needed for building educational virtual worlds, it has many drawbacks that inhibited the development and distribution of large-scale environments. JavaMOO was written to address those drawbacks, and further facilitate the creation and deployment of educational virtual worlds.

### ***JavaMOO Design***

The JavaMOO framework consists of 1) a suite of servers providing a reliable and robust environmental simulation and managing the interaction of dozens of connected students, 2) an application programmers' interface (API) for building an application-specific client that presents the state of the environment and allows students to interact with the environment and each other, and 3) a Data Collection Server collects server information and tracks student progress in each game. Figure 1 shows all of these components and how they communicate with each other. This framework provides teachers with a self-contained package that is easy to setup and use, along with maintenance functions that allow teachers to track student progress, and allow WWVIC to monitor the status of the server itself.

### **JavaMOO Servers**

The server of a JavaMOO virtual environment actually hosts a set of servers. The central component is the JavaMOO Server itself, which is responsible for executing the simulation, handling communication with clients that are connected to it, and communicating with the database to store and retrieve simulation and environment data. JavaMOO provides a set of foundation classes that form the base of the JavaMOO server. The game developer builds the simulated environment on top of this base. The client communication and data storage are all handled by the foundation classes, so the game developer doesn't need to worry about these features.

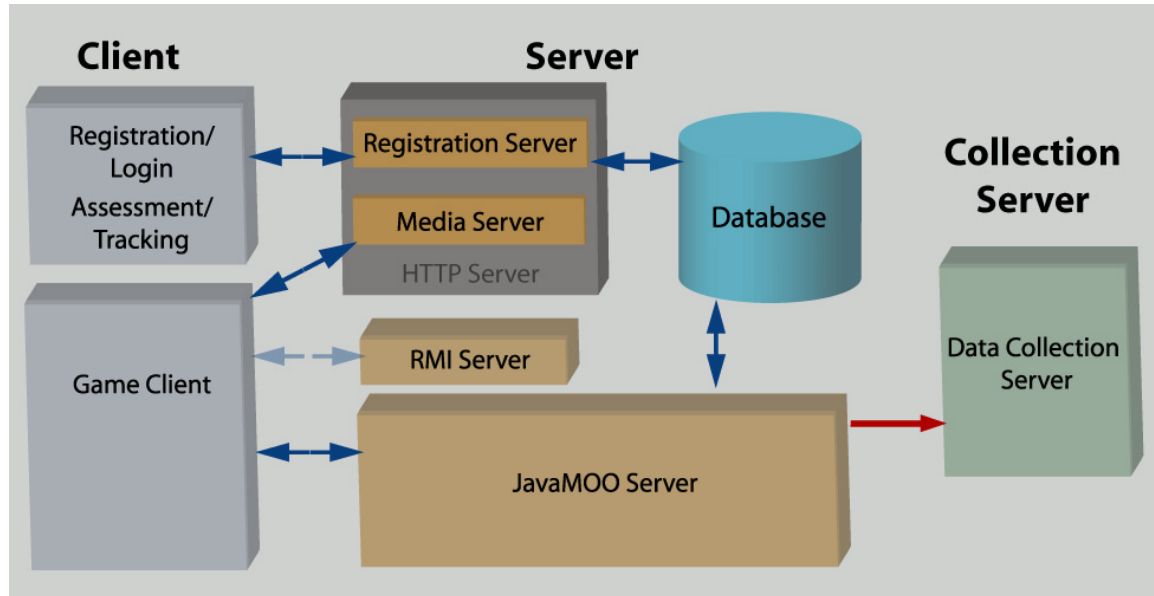
The JavaMOO server package comes with an SQL database, which the JavaMOO Server uses to store and retrieve simulation data. Account information is also stored in the database. While nearly any SQL server would work with our system, JavaMOO comes with the Apache Derby SQL database. By using an external database, JavaMOO has quick and reliable access to the data it needs.

Client/Server communication is handled using Java's Remote Method Invocation (RMI) system. In order for a client to connect to the JavaMOO Server, the server must first register with an RMI Server. The client then contacts the RMI Server to establish a connection with the JavaMOO server. Once this connection has been established, the RMI Server is no longer explicitly addressed.

The JavaMOO server package also comes with Tomcat, and HTTP server which is used to host a Registration Server and a Media Server. The Registration Server is used by the client to create new accounts, and ask students to fill out surveys with demographic information and knowledge evaluation.

This information is used to track student learning and provide data for larger studies. The Media Server holds images, animations, and other media used by the client. By storing these images on the central server, they do not need to take up storage on the client computers. This system also allows the game media to be updated without reinstalling a new client or rebooting the JavaMOO Server.

Together, these components make up the JavaMOO server package. A single server is created for each game being played by each class. Currently, this server is hosted by WWWIC.



**Figure 1: A networking diagram of the JavaMOO framework.** The Registration/Login system contacts the Registration Server to create an account in the Database and store demographic information. The Game Client then retrieves the JavaMOO Server Remote Method Invocation (RMI) address from the RMI Server, and logs into the JavaMOO Server. Media used by the Client is retrieved from the Media Server. The JavaMOO server uses the Database to store and retrieve all simulation and environment data. Periodically, the JavaMOO Server contacts the centralized Data Collection Server and submits reports on the server status and the progress of the students playing the game.

## JavaMOO Client

The JavaMOO Client is the application that is run by the student to interact with the virtual environment hosted by the JavaMOO Server. The game developer is responsible for writing the game client to suit the needs of the educational domain being modeled in the simulation. Because each game will have its own interface needs, the implementation of the client is largely left to the game developer. JavaMOO does provide a set of hooks with which the client can communicate with the JavaMOO server suite. We also provide a library for a classic two-dimensional game, which can be used or replaced with another interface library, such as one that would provide a three-dimensional interface.

The JavaMOO Client comes with a Registration Client that communicates with the Registration Server. This is where students create new accounts and take surveys, as described above with the Registration Server.

Clients are made available to the student using the school's computers.