

CS3524 Assessment

This assessment will examine your ability to design and implement distributed systems using the Java API. You need to use these skills to develop a game server providing clients' access to a MUD (Multi-User Dungeon or Dimension or Domain). Very basic functionality is required for a CGS D mark, but solutions with further more challenging structure and functionality attract better marks. Marks are also given for style, documentation, etc., but normally within the same CGS band as is indicated by the functionality implemented.

You are expected to use and adapt code provided in practical exercises to construct your solution to the assessment. When you do, however, you must acknowledge the classes that your solution is based on using a comment in your source code.

Requirements

CGS D

The code that should be written is as follows:

- Implement a MUD game server, creating a remote interface, extending `java.rmi.Remote`, an implementation of this interface, and a server mainline for the MUD game server. In developing your solution to this, start from the java code provided in `mud.tgz`. You can download this file from MyAberdeen.

The `mud` package contains three classes:

- `MUD`, which provides a graph-based representation of a MUD world;
- `Vertex`, which represents a vertex in the graph or a location in the MUD; and
- `Edge`, which represents an edge in the graph or a path between locations in the MUD.

The archive also contains three files that capture information about an example MUD world: `mymud.edg` contains the edges in the graph representing the MUD; `mymud.msg` contains the messages or information associated with each location; and `mymud.thg` contains the things at each location. You can define your own MUD using these file formats and check that it parses the MUD specification by running the class on its own (see the `main()` method in `MUD.java`).

- Implement a client that utilises the methods provided by the remote object to provide the user with the ability to make at least one move in at least one direction with the MUD and display information about the change of location.

CGS C

You should extend your solution to the CGS D requirements so that:

- Users can move around in any direction in your MUD world
- Users can see other users in the MUD
- Users can pick up things in the MUD

CGS B

You should extend your solution to the CGS C requirements so that the server provides access to multiple MUDs. To do this, you need to modify your server so that it:

- Generates more than one instance of your remote MUD object
- Provides the user with the ability to find out what MUDs are running on the system and select a MUD to join

CGS A5

You should extend your solution to the CAS 15-17 requirements so that a user can create new MUDs at run time. The server should also restrict the number of MUDs that can be running at any time and the total number of users logged on to MUDs.

CGS A4 – A1

For a CGS > A5, you are expected to add additional functionality to your CGS A5 solution, using techniques learned in the course, such as the handling of clients joining / leaving games, clients being able to create any number of new MUDs, how to handle clients aborting etc.

Submission Procedure

You are required to submit in electronic as form:

- one zip file called **cs3524_assessment_<your_studentID>.zip**

Submit your work, using the following method:

- In MyAberdeen, go to “Assessment”, and do the following:
 - click on CS3524 Assessment Submission
 - click on “Browse My Computer” (this is underneath “Assignment Submission”)
 - find your zip file and complete this upload

Please use your University email address to submit your assessment.

All submissions be documented, and this documentation must include your name and userid at the top of each file. Please submit (a) the complete source code (plus any necessary make files, text files, configuration files etc to compile and run your submission) and (b) a report describing your submission and how to operate your application. Make a zip file containing all this information and send it to the email address above.

For generating the PDF version of your source code: To ensure that the source code is not excessively long, I suggest that you print the code in two-column landscape mode. The unix utility `enscript` provides this and more; for example, try out the following on some Java source file, e.g. `MyClass.java`, and look at the resulting PDF (`MyClass.pdf`): `enscript -2 -r -Ejava -o - MyClass.java | ps2pdf - MyClass.pdf`

Deadline

The deadline for handing in your in-course assessment is specified in MyAberdeen.

Standard [lateness penalties](#) apply.

Important Note

This exercise is part of the formal assessment of CS3524 and the work done must be your own. You should look at the section on [Cheating and Plagiarism](#) in your [student handbook](#).