Programming project - Geometry Ball Tournament

In this programming project, you will translate a simple single-site game to a distributed game that uses client/server architecture and supports Java clients. The game must actively fight LAG and cheating in order to improve the gaming experience.

Overview

The centralized version of the tool is a very simple game where two teams try to push the white ball to the opponent's side of the playing field.

Interface

You turn your ship either to the left or to the right and accelerate. There is no break or reverse.

Assignment goals

The overall goal of the assignment is to turn the single-site game into a competitive distributed game across multiple platforms. Specifically, it should be possible to play the game with clients that may be on different types of machines (desktop/android).The gaming experience should not be impeded by LAG and you should take active measurements to prevent cheating.

The LAG and cheating objectives should have the highest priority, but you are welcome to modify the game experience with graphics or other similar types of improvements in gameplay/interaction design.

You must use client/server architecture, but you may use either UDP or TCP as transport protocol. Your design decisions should be well motivated in your report.

Getting started

First of all make sure that the provided source code compiles and run in your development environment.

- Java project and source code

Once you are familiar with the code, start thinking of the architecture and interfaces for the distributed application. *Remember to document in this step!* If you are unsure about anything in this step, feel free to discuss it with your supervisor or other students. When you have a clear idea of your architecture and have specified all interfaces, you can start implementing the application.

Examination

Each student should turn in a report, a presentation, well-commented source code (including any interface definitions), as well as working jar-files containing a runnable version of your desktop-client and server. Your will demonstrate your project in seminar form.

You can either work alone or work in groups of two.

You may **not** cooperate with other student groups when writing your code or report, but you are encouraged to discuss any problems you encounter with your fellow students.

The deadline for the report, presentation and the code is available in SCIO. The seminar is in your schedule.

Deliverables

The project has the following deliverables:

- Jar files with your runnable solution
- Source files for code review (zip the whole project)
- Report
- Presentation

Examination criteria

To pass the assignment, you must have working program that follows the specification under "Assignment goals" above, and it must be possible to run the application with the clients and/or any servers located on different machines. The report must be written properly - sloppy language and formatting is not acceptable. Badly written reports will be returned immediately without comments on report or code.

You must also present and demonstration your solution in the seminar.

Testing and verification are important aspects of building any application. In your report, you should have a thorough discussion about how you have verified and tested your program for LAG and cheating prevention.

Report tips

When writing your report, make sure that you have covered at least the following things:

- A short description of the problem
- Brief descriptions of alternative approaches to solving the problem that you considered during the project.

- A more detailed description of your solution approach, including a motivation for why you chose that particular approach. You need to consider failure, security, LAG in this section.
- A post-project reflection on the advantages and disadvantages of your solution.
- A discussion of possible future work and improvements. Would you have approached the problem differently if you could redo it from the beginning?
- A reflection on what you have learned during the project.
- Brief feedback on the assignment itself.

Presentation tips

Here are some requirements and tips regarding the seminar and demo session.

- The "demo devil" has a tendency to show up. Prepare some backup slides with screen shots to use if things fail on demo day.
- The presentation and demo should not be longer than 5-10 minutes. This means that you only have time for a few remarks in addition to demonstrating your program. No more than two (2) slides with information!
- On the actual seminar we will have two computers available so that you can prepare your demo at the same time as a presentation is underway. We will provide a presentation list.

You may use either Swedish or English for your report but the presentation should be in English. Regardless, use grammar- and spell checking tools!

Good luck!