TECHNISCHE
UNIVERSITÄT
DARMSTADT

Prof. Dr. Max Mühlhäuser
Dr. Immanuel Schweizer

Dr. Benedikt Schmidt
Dipl.-Wirtsch.-Inform. Axel Schulz

TELEKOOPERATION
Fachbereich Informatik
Hochschulstr. 10
64289 Darmstadt

TK1: Distributed Systems - Programming & Algorithms

1. Programming Assignment
Submission Date: 10.11.13

By handing in a solution you confirm that you are the exclusive author(s) of all the materials. Additional information can be found here: http://www.informatik.tu-darmstadt.de/de/studierende/studium-alt/plagiarismus/

Shared Fly Hunting Game via Java RMI (20 P.)

Implement a game as a Client/Server application based on **Java RMI**. The game is about hunting a fly with a fly flap. On the GUI a fly randomly appears. By pressing the mouse on top of the fly, the fly was "hunted". The player who caught the fly first gets a point. Once the fly was hunted it re-appears at a different position. All players see the same fly at the same position. Also, all players see the current points of all other players.

Requirements:

- The GUI for the client should be Swing based.
  - The GUI shows the fly
  - The GUI shows a list of all players with their current points scored
  - The GUI notifies the player when a fly was hunted
  - Use the Model-View-Controller pattern e.g. changes to window size should not delete the model etc.

Server:

- The server sends the current position of the fly to all clients
- The server distributes all changes to all players (points, fly hunted, participants)

Client:

- The client sends all changes (one point gained, fly hunted, etc.) to the server.
- The client should load the stubs from the server codebase (file-url should suffice)

Please keep the following in mind:

- Please generate a working Ant script starting two clients and a server.

The interface for client and server could (but not has to) look like this:

```
IGameServer:
      void login(String playerName, IGameClient client)
      void logout(String playerName)
      void huntFly(String playerName)

IGameClient:
      void receiveFlyHunted(String playerName, int newPoints)
      void receiveFlyPosition(int x, int y)
```

The following scoring scheme is applied:
-Model-View-Controller realized 2P (GUI, Controller, Client)
-Executable Server 6P (login, logout, huntfly, update/broadcast der clients)
-Executable Client 6P (Visualization/Event Handling etc.) + 2P if second client is executable
-Game features 4P (Fly displayed, Fly moves, Highscore list, logout)

Important:
If the ANT script is not executable -> 0P
Project of a different group submitted (Plagiarism) -> 0P for both groups