**MILESTONE 1**

About a Game 20

Outline of the game with its goals, rules and mechanics

About a Game (advanced) 5

Have a very clear and well thought out concept of your game

Ahead of Schedule and Under Budget! 10

You have already started producing relevant code for your project.

Dear Diary 5

Project diary exists and is readable

Dear Diary (advanced) 5

The diary has at least one meaningful entry

Mockup 5

Show a mockup of your game during the presentation

Networking 10

Overview of the functionality of client and server

Requirement Analysis 10

Overview of the software requirements (not required software)

The Name of the Game 5

Have a name for your game, ideally also for your group

Who? What? When? 15

Project timeline and responsibilities

Who? What? When? (advanced) 10

Have a detailed and well thought out project plan

**MILESTONE 2**

.gitignore 5

There is a gitignore file containing entries relevant for major operating systems and the used tools

/\*\* ... \*/ 10

Source code is sufficiently documented

3nc0d1️⃣ng ER2ÖR 15

Protocol and content is properly encoded, decoded and validated

Call Me Bob 5

Players can change their nicknames

Chat 15

Client to client chat is working

Dear Diary 10

Project diary is up to date and filled with meaningful entries

Login 10

Arbitrarily many clients can log on to the server

Logout 5

The server handles a client logout in a meaningful way

Ping 10

A ping message is regularly sent from server to client and processed meaningfully (connection losses are detected and handled)

Pong 10

A ping message is regularly sent from client to server and processed meaningfully (connection losses are detected and handled)

Protocol Code 10

The human-readable network protocol is defined and documented in the source code

Protocol Document 10

The human-readable network protocol is defined in a dedicated document

Protocol Validator 15

Network protocol is being used correctly

QA Concept 10

Reasonable Software Quality concept

bob\_001 5

The server assures that player nicknames are unique and changes duplicates in a consistent manner by assigning appropriate names

whoami 5

The client suggests a nickname based on the system username

**MILESTONE 3**

/\*\* ... \*/ 10

Source code is sufficiently documented

About a Game 5

Outline of the game with its goals, rules and mechanics

Broadcast 5

Broadcast to all clients across all games and lobbies is working and has a dedicated command

Build Script 5

The build script succeeds and produces an executable jar & javadoc for the task "build-cs108"

Command Line 5

Command line parameters are parsed correctly: (client <hostadress>:<port> [<username>] | server <port>)

Dear Diary 10

Project diary is up to date and filled with meaningful entries

Demo! 10

Present a working prototype of the game in the presentation

GUI 15

The chat is available via a basic GUI

Game List 5

There is a way to list open, ongoing as well as finished games. Their status is indicated

Game Logic 25

The main logic of the game as well as its fundamental mechanics are present and the game is playable

Game State 10

Game state is kept on server

Librarian 10

Use at least one external library besides JUnit. JavaFX does not count as an external library.

Lounging 10

Have support for multiple lobbies (one per game) with their respective internal chats

Manual 10

There is a manual which describes how the game is to be played

Player List 5

There is a way to list all players currently connected to the server

Progress Report 15

How things are going - project timeline and responsibilities, changes and problems

Protocol Code 10

Network protocol is completely defined and documented in source code

Protocol Document 5

The definition of the network protocol in the external document is up to date

QA 10

Present your QA measures which can also be found in the updated document

QA (advanced) 10

Have a well thought out and detailed written QA concept and a solid and realistic plan on how to implement it. If possible, have a first result of your metrics.

Rules to Code 5

How is the gamestate and game logic represented in the code

Shall We Play a Game 30

Present a working prototype of the game logic in the exercise slot by playing the game (Terminal or GUI)

Technology! 10

Describe tools, libraries and processes which you use that are not introduced in the lecture and why you use them

Whisper 5

Whisper-chat between two clients is working

Who? What? When? 10

Update and adjust your project plan for milestones 4 and 5

**MILESTONE 4**

/\*\* ... \*/ 10

Source code is sufficiently documented

5 out of 7 5

You will receive the playable .jar from two other groups. Assess all MS4 achievements which pertain to the game for these two groups. Handin procedure specified on Slideset 06 and via Mail after MS4.

Dear Diary 10

Project diary is up to date and filled with meaningful entries

GUI 20

The game is represented graphically

GUI (advanced) 30

The game is fully playable from the UI. Text input is only used where absolutely necessary.

High Score 5

There is a high score list which is stored persistently, updated when needed and availble for clients

Peer Precision 5

Your assessment of the other games matches our assessment with a deviation of at most one achievement

Referee 15

The rules of the game are checked and enforced correctly

Shall We Play A Game 20

Present the working game in the excercise slot by playing the game

Unit-Test 5

All relevant features of a core component of your game are tested

Unit-Test (advanced) 15

Unit-Tests are sufficient and meaningful

Victory! 10

The winner and the win state is correctly determined

**5**

/\*\* ... \*/ 15

Source code is sufficiently documented

About a Game 10

Outline of the game with its goals, rules and mechanics

About a Game (advanced) 10

Have a clear introduction to your game explaining its fundamentals in such a way that a first-time listener can understand it

Architecture 15

Have documentation outlining the overall program architecture

Archiving 5

Your repository contains an outreach/ folder with the specified content

Command Line 5

Command line parameters are still parsed correctly

Dear Diary 10

Project diary is up to date and filled with meaningful entries

Demo! 35

Have a bugfree demo of the game in the final presentation

GUI 30

The GUI works without any errors and all client functionality is accessible by it

Game Logic 15

The entire logic of the game as well as all its mechanics are present and the game is fully playable

High Score 10

There is a high score list which is stored persistently, updated when needed and shown when requested

Login 5

Arbitrarily many clients can still log on to the server

Logo 5

The game and/or team has a proper Logo

Manual 10

There is a manual which describes how the game is to be played

Pachydermatous Librarian 10

All external libraries in your project are managed by gradle via maven central

Picturesque 10

Have a representative screenshot of the game

Protocol Code 5

Final version of network protocol is completely defined and documented in source code

Protocol Document 5

The final version of the network protocol is current in external documentation

QA 15

Results of Quality Assurance measures and lessons learned from them

QA Report 15

QA report with goals, methods and results

QA Report (advanced) 15

Detailed and relevant analysis of QA measures over time, discussion of results, well-written report

Shall We Play A Game 35

Present the working final game in the excercise slot by playing the game

Technology! 10

Describe tools, libraries and processes which you use that are not introduced in the lecture and why you use them

Technology! (advanced) 15

Use at least two libraries besides JUnit actively for a significant / impactful task in your project. JavaFX does not count.

Twitchy 5

Have a representative video showing somewhere between 20 seconds and 2 minutes of gameplay

Unit-Test 10

All unit-tests are meaningful and 'green'

We Are Smarter Now 20

Lessons Learned