Fachhochschule Nordwestschweiz
Hochschule für Technik

Software Architecture

Exercise – Gaming Platform (iteration #1)

BSc

Ingo Arnold

Exercise Opening

Overall Motivation

In this unit, we will **practically design a small system** as a solution to a given problem. The focus is less on functionality or final maturity of the system than on its architecture design.

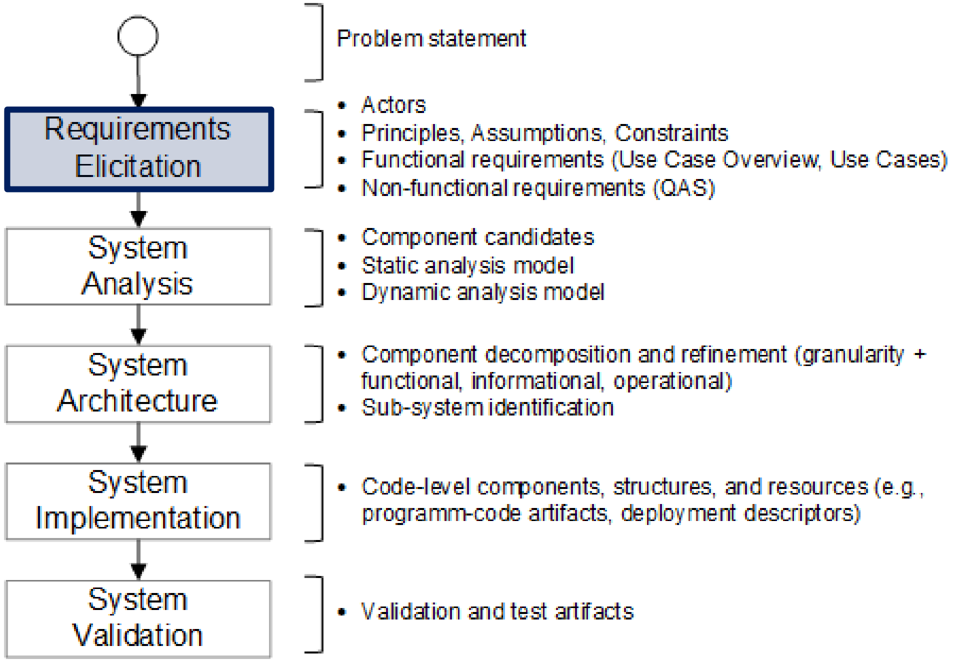
Purpose is to understand how you **systematically follow a path from a given problem to a corresponding solution** and that you understand the evolutionary nature of architecture (i.e., architecture as an iterative process that is accompanied by uncertainties).

This solution will accompany us throughout the semester. We will use it to **go through the architecture process** once completely – albeit rudimentarily. This also means that at the end of the process we will have an executable version of the system in Java.

Exercise Opening

Motivation

You **create an initial design** for a **gaming platform** by incrementally following the process outlined, below. Note that your design should focus less on the algorithmic and more on the structural solution aspects.



Problem statement

- Actors
- Principles, Assumptions, Constraints
- Functional requirements (Use Case Overview, Use Cases)
- Non-functional requirements (QAS)

- Component candidates
- Static analysis model
- Dynamic analysis model

- Component decomposition and refinement (granularity + functional, informational, operational)
- Sub-system identification

- Code-level components, structures, and resources (e.g., programm-code artifacts, deployment descriptors)

- Validation and test artifacts

Exercise Opening
Motivation

A gaming platform is to be developed—i.e. a platform to which players can log in and subsequently play the games available.

- The gaming platform provides players with a unified gaming ecosystem and allows players to access all games registered with the platform.
- At the same time, the gaming platform provides game providers with access to a large group of players.
- Players must first register on the gaming platform.
- Registration requires the player to enter their first and last name, a unique account name, a password of their choice, and a role (i.e., either player or administrator or game provider).
- After players have registered with the platform, they can enter it.
- For this purpose, a player or administrator logs in to the platform by providing account name and password.
- After entering the platform, players will find a catalogue that they can view and select games from.

Exercise Opening

Motivation

A gaming platform is to be developed—i.e. a platform to which players can log in and subsequently play the games available.

- Before this catalogue shows games, they must have been previously registered for the platform.
- If an actor finds an interesting game in the catalogue, he can select it and register for the game.
- In doing so, the game is entered into the actor's favourites list so that the player can start the game directly from there in the future.
- After a game is started, players play the game.
- We limit the games for now so that all games are played against the computer.
- A game ends when one of the players (human or computer) wins the game or the human player ends the game early.
- For the beginning, it is enough for us to simulate games.

Exercise Opening
Motivation

A gaming platform is to be developed—i.e. a platform to which players can log in and subsequently play the games available.

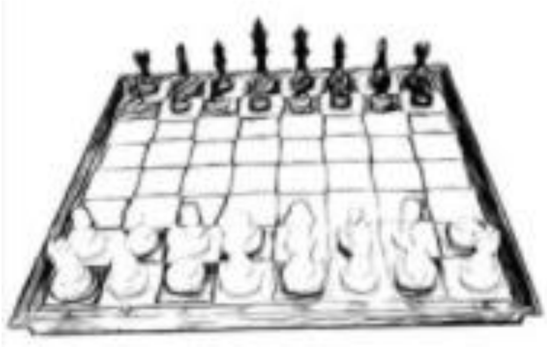
- A game simulation is to randomly determine a winner (i.e., player or computer). The duration of the simulated game (between 1 and 100 minutes) and the score resulting from the victory (between 1 and 100 points) should also be determined randomly.
- For each game there is a high score list.
- At the end of a (simulated) game, the winner, his opponent, the score as well as the duration are entered into the high score list of the respective game.
- The high score list of a game can be viewed by all players at any time.
- It is sufficient for screen outputs as well as possible screen inputs to direct these into a console output.
- Initially, a first prototype of the game platform is to be developed in a very short time.
- At the same time, it is expected that the original expectations for the game platform will be successively expanded.

Exercise Opening
Motivation

A gaming platform is to be developed—i.e. a platform to which players can log in and subsequently play the games available.

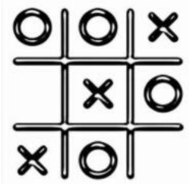
- Three games are to be initially registered on the platform.
- However, the games do not have to be implemented algorithmically – i.e., we assume the games to be closed entities.

chess



connect four

tic tac toe



Exercise Opening

Motivation

Start by **refining the given requirements** where this seems reasonable to you. Distinguish between functional and non-functional requirements, constraints and assumptions.

Exercise Agenda

- Requirements Elicitation
- System Analysis
- System Architecture
- System Implementation

Requirements Elicitation

Use Case Overview and Use Cases

In a first step, use the gaming platform description and derive the functional requirements inherent in it. In other words, examine the description for *actors* and *functionalities* (i.e., *use cases*) expected from the platform.

Create a *use case overview* and elaborate descriptions for all identified *use cases* based on the schema below. Focus on the attribute *flow of events*. However, also think about pre- and post-conditions as well as use case specific quality attributes already. Note them where useful.

Use Case Name
Actor
Flow of Events
Precondition
Postcondition
Qualities

Requirements Elicitation

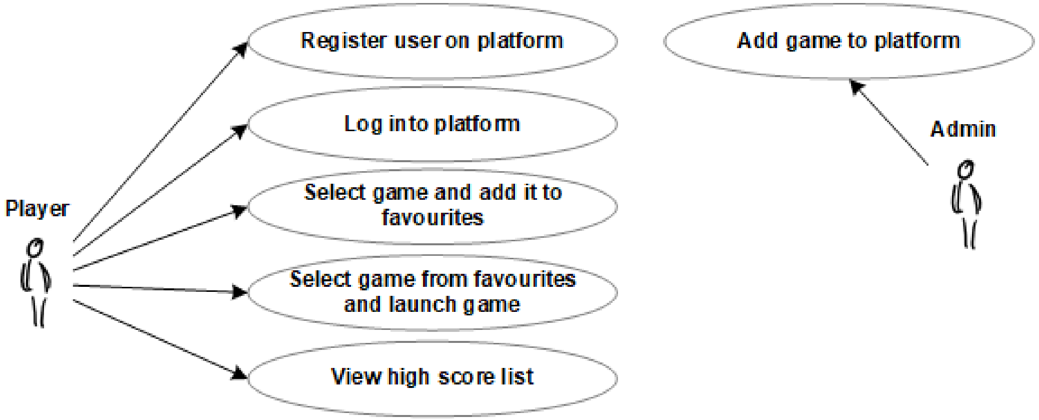
Use Case Overview and Use Cases

Create an **use case overview diagram** for the gaming platform. The use case overview diagram should include all major actors and use cases.

Requirements Elicitation

Use Case Overview and Use Cases

Create an **use case overview diagram** for the gaming platform. The use case overview diagram should include all major actors and use cases.


Player
Register user on platform
Log into platform
Select game and add it to favourites
Select game from favourites and launch game
View high score list
Add game to platform
Admin

Requirements Elicitation

Use Case Overview and Use Cases

Create *use case descriptions* for all identified use cases (focus on *flow of events*).

Use Case Name
Actor
Flow of Events
Precondition
Postcondition
Qualities

Requirements Elicitation

Use Case Overview and Use Cases

Use case **add game to platform**.

Use Case Name	Add game to platform
Actor	Admin
Flow of Events	1. Actor creates game and sets its name. 2. Actor adds game to the gaming platform.
Precondition	Actor is logged into platform as admin.
Postcondition	New game is added to the gaming platform.
Qualities	

Requirements Elicitation
Use Case Overview and Use Cases

Use case **register user on platform**.

Use Case Name	Register user on platform
Actor	Player, Admin
Flow of Events	1. Actor initiates self-registration. 2. Actor registers with platform providing his name, first name, unique id, password (must be entered twice), and role (either «player» or «admin»). 3. Platform validates attributes and registers actor.
Precondition	
Postcondition	New player or admin is registered on platform.
Qualities	

Requirements Elicitation
Use Case Overview and Use Cases

Use case **log into platform**.

Use Case Name	Log into platform
Actor	Player, Admin
Flow of Events	1. Actor initiates login. 2. Actor provides gaming platform credentials (i.e., unique id and password). 3. Platform validates credentials and establishes user session.
Precondition	Actor is successfully registered on platform.
Postcondition	User session for actor is established.
Qualities	

Requirements Elicitation
Use Case Overview and Use Cases

Use case **select game and add it to favourites**.

Use Case Name	Select game and add it to favourites
Actor	Player, Admin
Flow of Events	1. Actor opens the game catalogue. 2. Actor views the game catalogue contents. 3. Actor selects game from game catalogue. 4. Selected game is added to actor’s favourites list.
Precondition	Player is successfully logged into platform. Game catalogue is not empty. Selected game is not yet in the player’s favourites list.
Postcondition	Game is added to actor’s favourites list.
Qualities	

Requirements Elicitation
Use Case Overview and Use Cases

Use case **select game from favourites and launch game**.

Use Case Name	Select game from favourites and launch game
Actor	Player
Flow of Events	1. Actor opens the favourites list. 2. Actor views the favourites list contents. 3. Actor selects game from favourites list. 4. Actor launches a selected game. 5. Actor plays the game till it ends. 6. Game results are added to the game’s high score list.
Precondition	Player is successfully logged into platform. Game was played and high score list is not empty.
Postcondition	Favourites list was viewed by an actor.
Qualities	

Requirements Elicitation

Use Case Overview and Use Cases

Use case **view high score list**.

Use Case Name	View high score list
Actor	Player
Flow of Events	1. Actor opens the high score list. 2. Actor views all game results in the high score list.
Precondition	Player is successfully logged into platform. High score list is not empty.
Postcondition	High score list was viewed by an actor.
Qualities	

Requirements Elicitation

Qualities

Reflect on *qualities* that were already mentioned in the gaming platform’s description.

Requirements Elicitation
Qualities

Reflect on *qualities* that were already mentioned in the gaming platform’s description.

- Use case **add game to platform**: games must be checked for several aspects (e.g., code-review to exclude malicious code from entering the gaming platform) and authorized by several parties as part of the process of adding them.
- Use case **register user on platform**: each user can be registered only once (e.g., unique identifier). If a registered user attempts registration, the platform rejects the request.
- Use case **log into platform**: credentials are securely transmitted—secured against replay, tampering, interception attacks.

Questions