

IDATT2003 Project Report

Vetle Nilsen and William Holtsdalen

26. february 2025

Abstract

This report ...

Contents

1	Introduction	4
1.1	Background	4
1.2	Requirements	4
1.2.1	Functional requirements	4
1.2.2	Non-functional requirements	4
1.3	Constraints	4
1.3.1	Development constraints	4
1.3.2	Functionality constraints	5
1.4	Dictionary	5
2	Theory	5
3	Method	6
3.1	Development process	6
3.2	Tools	6
3.3	Use of AI-tools	6
4	Results	6
4.1	Technical design	6
4.2	Implementation	6
4.3	Testing	8
4.4	Deployment to end user	8
5	Discussion	8
5.1	Solution	8
5.2	Process	8
5.3	AI-tools	8
6	Conclusion	8
	References	8
7	Appendix	8
A	KI-deklarasjon	8

Figures

1	Use-case-diagram.	5
2	Class diagram.	7

Tables

1	Dictionary	6
---	----------------------	---

Code Listings

1	Java-code from file	6
---	-------------------------------	---

1 Introduction

1.1 Background

This project aims to create board game applications by writing code in Java.

1.2 Requirements

1.2.1 Functional requirements

- The board game applications should be able to be played by the same rules and mechanics as the original board games.
- Players must be able to move their game piece, view the state of the game, and restart the game.
- The application should provide a GUI.
- Multiplayer option should be provided locally.
- The user should be able to add Bot-players to the gaming-lobby.
- The gaming board should contain event tiles where something happens if a player lands on these specific tiles.
- There should be one or more random number generators, which should simulate die rolls.

1.2.2 Non-functional requirements

- The GUI should be user-friendly.
- The Gaming-lobby should consist of 2-5 players, with a minimum requirement of 1 real life player.
-

1.3 Constraints

The project has several constraints that need to be followed. These can be divided into 2 main groups: development constraints and functionality constraints.

1.3.1 Development constraints

The development of the project must follow specific technological and structural constraints. Firstly, the application must be made in Java, with a focus on object-oriented design principles. The graphical user interface GUI must be built with JavaFX, and the use of FXML is not allowed.

Version control is a mandatory aspect of the project. All code must be maintained in a Git repository, with regular commits. Additionally, the project is required to be set up as a Maven project.

Furthermore,

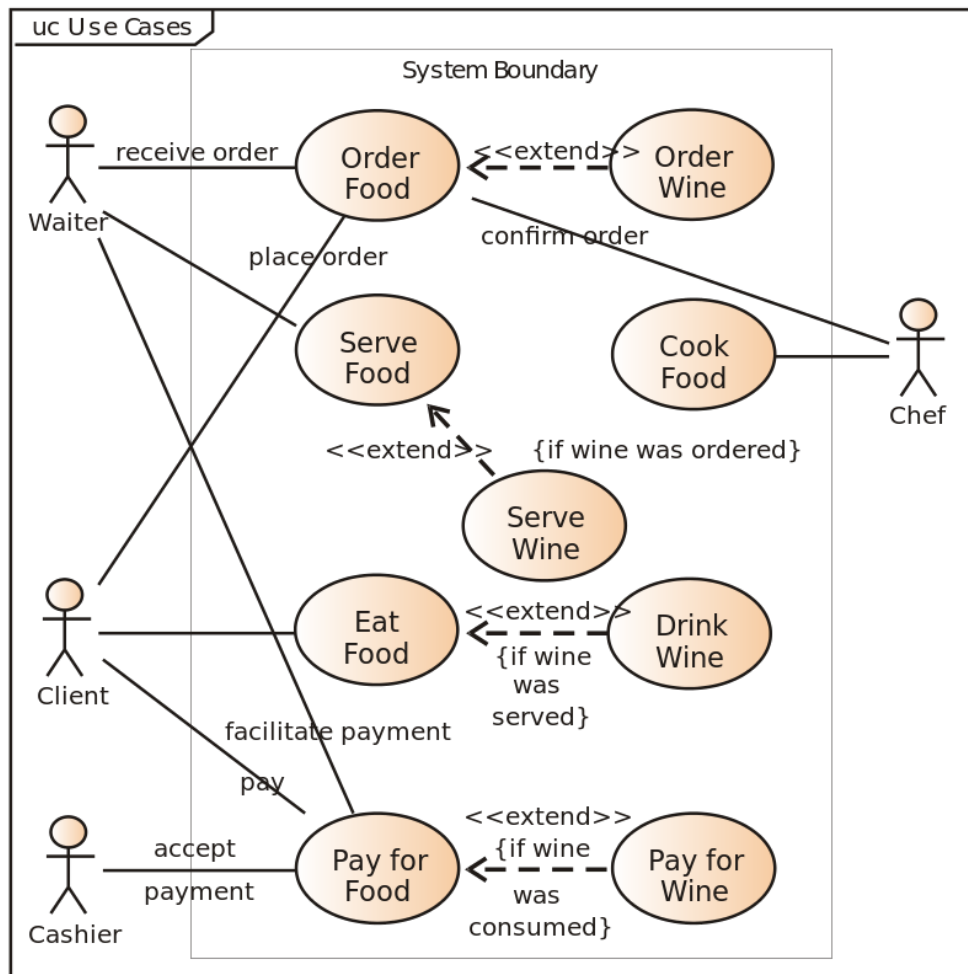


Figure 1: Use-case-diagram.

1.3.2 Functionality constraints

1.4 Dictionary

2 Theory

Example reference: [1]

Term	Abbreviation	Description
Game		Desc. here
Die		Desc. here
Dice		Desc. here
Board		Desc. here
Player		Desc. here
Tile		Desc. here
Action / Tile action		Desc. here

Table 1: Dictionary

3 Method

3.1 Development process

3.2 Tools

3.3 Use of AI-tools

4 Results

4.1 Technical design

4.2 Implementation

Code listing 1: Java-code from file

```
1  /**
2   * Hello World
3   */
4  class HelloWorld {
5      public static void main(String[] args) {
6          System.out.println("Hello, World!");
7      }
8  }
```



Figure 2: Class diagram.

4.3 Testing

4.4 Deployment to end user

5 Discussion

5.1 Solution

5.2 Process

5.3 AI-tools

6 Conclusion

-
-
-
-

Terminology

Bot-players Computer controlled players. 4

gaming-lobby the party of players playing the game, it can contain both real life players and Bot-players. 4

GUI Graphical User Interface, its when a user can view and interact with an application. 4

References

- [1] D. J. Barnes and M. Kölling, *Objects First with Java: A practical introduction using BlueJ*, Sixth. Pearson/Prentice Hall, 2017.

7 Appendix

A KI-deklarasjon

[KI-deklarasjonen som *skal* leveres inn utfylt, er vedlagt på neste side. Last ned Word-malen og fyll den ut, og last opp generert PDF.]

Deklarasjon om KI-hjelpemidler

Har det i utarbeidingen av denne rapporten blitt anvendt KI-baserte hjelpemidler?

☐ Nei

☐ Ja

Hvis *ja*: spesifiser type av verktøy og bruksområde under.

Tekst

☐ **Stavekontroll.** Er deler av teksten kontrollert av:
Grammarly, Ginger, Grammarbot, LanguageTool, ProWritingAid, Sapling, Trinkai.ai eller lignende verktøy?

☐ **Tekstgenerering.** Er deler av teksten generert av:
ChatGPT, GrammarlyGO, Copy.AI, WordAi, WriteSonic, Jasper, Simplified, Rytr eller lignende verktøy?

☐ **Skriveassistanse.** Er en eller flere av ideene eller fremgangsmåtene i oppgaven foreslått av:
ChatGPT, Google Bard, Bing chat, YouChat eller lignende verktøy?

Hvis *ja* til anvendelse av et tekstverktøy - spesifiser bruken her:

Kode og algoritmer

☐ **Programmeringsassistanse.** Er deler av koden/algoritmene som i) fremtrer direkte i rapporten eller ii) har blitt anvendt for produksjon av resultaterslik som figurer, tabeller eller tallverdier blitt generert av: *GitHub Copilot, CodeGPT, Google Codey/Studio Bot, Replit Ghostwriter, Amazon CodeWhisperer, GPT Engineer, ChatGPT, Google Bard* eller lignende verktøy?

Hvis *ja* til anvendelse av et programmeringsverktøy - spesifiser bruken her:

Bilder og figurer

☐ **Bildegenerering.** Er ett eller flere av bildene/figurene i rapporten blitt generert av: *Midjourney, Jasper, WriteSonic, Stability AI, Dall-E* eller lignende verktøy?

Hvis *ja* til anvendelse av et bildeverktøy - spesifiser bruken her:

Andre KI-verktøy

☐ **Andre KI-verktøy.** har andre typer av verktøy blitt anvendt? Hvis *ja* spesifiser bruken her:

☐ Jeg er kjent med NTNUs regelverk: *Det er ikke tillatt å generere besvarelse ved hjelp av kunstig intelligens og levere den helt eller delvis som egen besvarelse.* Jeg har derfor redegjort for all anvendelse av kunstig intelligens enten i) direkte i rapporten eller ii) i dette skjemaet