Assignment

Design Patterns - 2025

Tutor: Prof. Dr. Doğan Aydın

İzmir, October 27, 2025

Contents

1 | Project Overview

Project Objective

This project aims to teach how design patterns [1, 2, 3, 4, 5] can be applied in a real software project. Students will develop an online multiplayer **Bomberman** game to practice object-oriented programming principles and design patterns. During the project, at least four different design patterns must be implemented.

Technology Stack

- Programming Language: Java, C#, C++, or Python

- Game Framework: Unity, Pygame, or manual development

- Database: MySQL, PostgreSQL, MongoDB, MS SQL, or SQLite

- Network Technology: WebSocket, Socket.IO, SignalR, MQTT, or native socket programming

2 | Game Requirements

Core Game Mechanics

- Two-player online gameplay

- Classic Bomberman rules must be applied.

- Players place bombs, which explode after a fixed time delay.

- Explosions propagate in four directions.

- A player dies when hit by an explosion or when touching an enemy.

2.1 | Map and Walls

The map must include at least three wall types:

- Unbreakable Walls: Cannot be destroyed.

- Breakable Walls: Destroyed with a single explosion.

- Hard Walls: Require multiple explosions to be destroyed.

Theme system:

- Desert Theme: Sand and stone walls

- Forest Theme: Green and tree-like walls

- City Theme: Concrete or brick walls

2.2 | Power-up System

When breakable walls are destroyed, random power-ups may appear:

- Increased Bomb Count

- Increased Bomb Power

- Speed Boost

- (Optional) Skateboard, Wall-Pass, Bomb-Push, etc.

2.3 | Enemy System

At least one type of enemy is required:

- Static Moving Enemies

- Chasing Enemies (simple AI)

- Intelligent Enemies (shortest path logic)

2.4 | Database Requirements

- User information (username, password hash)

- Game statistics (wins, losses, total games)

- High scores (leaderboard)

- Player preferences (theme, etc.)

Design Pattern Requirements

Mandatory Pattern Distribution

- 1+1 [1] Creational Patterns

- 1+1 Structural Patterns

- 2+1 Behavioral Patterns

- Repository Design Pattern

- Architectural Pattern: MVC, MVP, MVVM, etc. (choose one)

3 | Deliverables

- Source Code (70 points)

- Design Document (30 points)

4 | Evaluation Criteria

Category	Criterion	Points
Source Code	Pattern Implementation	50
	Code Quality	10
	Functionality	10
Design Document	Pattern Explanation	20
	UML Diagrams	10

5 | Additional Rules and Submission

Projects can be done individually or in pairs. AI assistance is allowed, but students must understand the logic of their code. Submission: GitHub link or ZIP + PDF document via the university LMS (UBS). The submission deadline will not be extended.

---

[1] If individual project: 1; if group of two: 2

6 | Bonus Points

- Each additional design pattern (+5)
- Advanced AI (A* Pathfinding) (+5)
- Professional UI/UX (+5)
- Multiplayer Lobby System (+5)

7 | Contact and Submission Details

Deadline: **28.12.2025 24:00**
Submission Platform: IKÇÜ UBS
Email: dogan.aydin@ikc.edu.tr

Good Luck!

8 | References

[1] Eric Freeman, Elisabeth Robson, Bert Bates, and Kathy Sierra. *Head First Design Patterns: A Brain-Friendly Guide.* " O'Reilly Media, Inc.", 2004.

[2] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, and Design Patterns. Elements of reusable object-oriented software. *Design Patterns*, 1:417, 1995.

[3] Robert Nystrom. Game programming patterns. https://gameprogrammingpatterns.com/contents.html, 2014. Online version accessed: 2025-10-27.

[4] Refactoring.Guru. Refactoring.guru – design patterns refactorings. https://refactoring.guru/, 2025. Accessed: 2025-10-27.

[5] SourceMaking. Design patterns — sourcemaking. https://sourcemaking.com/design_patterns, 2025. Accessed: 2025-10-27.