Ankara Yıldırım Beyazıt University
Department of Computer Engineering

CENG 201 – Object Oriented Programming
Course Project

G9: Empires of Ages

Analysis Report

Cahit Onur Enoğlu 24050111040,
Yusuf Yücel 23050111013,
Hacı Salih Toker 23050111008,
Eren Köse 23050111059

Instructor: Muhammed Abdullah Bülbül
Teaching Assistant: Elif Şanlıalp
Date: 13/11/2025

Table of Contents

1. Introduction

This document presents the analysis report for Group 9's project, a Real-Time Strategy (RTS) game developed in C++ using the SFML library. The game involves base-building, resource management, and unit control, with support for LAN multiplayer gameplay. This report outlines the requirements, system models, and design considerations based on Object-Oriented Programming principles.

2. Requirements

2.1. Functional Requirements

• Player can create or join a LAN multiplayer session.
• Players can control units on a tile-based map.
• Units can move, attack, and gather resources.
• Players can construct and upgrade buildings.
• The system manages resource generation and unit production.
• The game synchronizes states between clients via network communication.

2.2. Non-Functional Requirements

• The system should maintain smooth real-time gameplay with minimal latency.
• The user interface should be responsive and visually intuitive.
• The architecture should be modular and maintainable, following OOP principles.
• Codebase should follow SOLID principles for scalability.

3. System Models

3.1. Scenarios

Scenario 1: The player starts a new game session, gathers resources, constructs a base, trains units, and defeats the opponent by destroying their main building.

Scenario 2: A player joins a LAN match, coordinates with teammates, and uses tactical decisions to outmaneuver enemy units and secure victory.

3.2. Use Cases

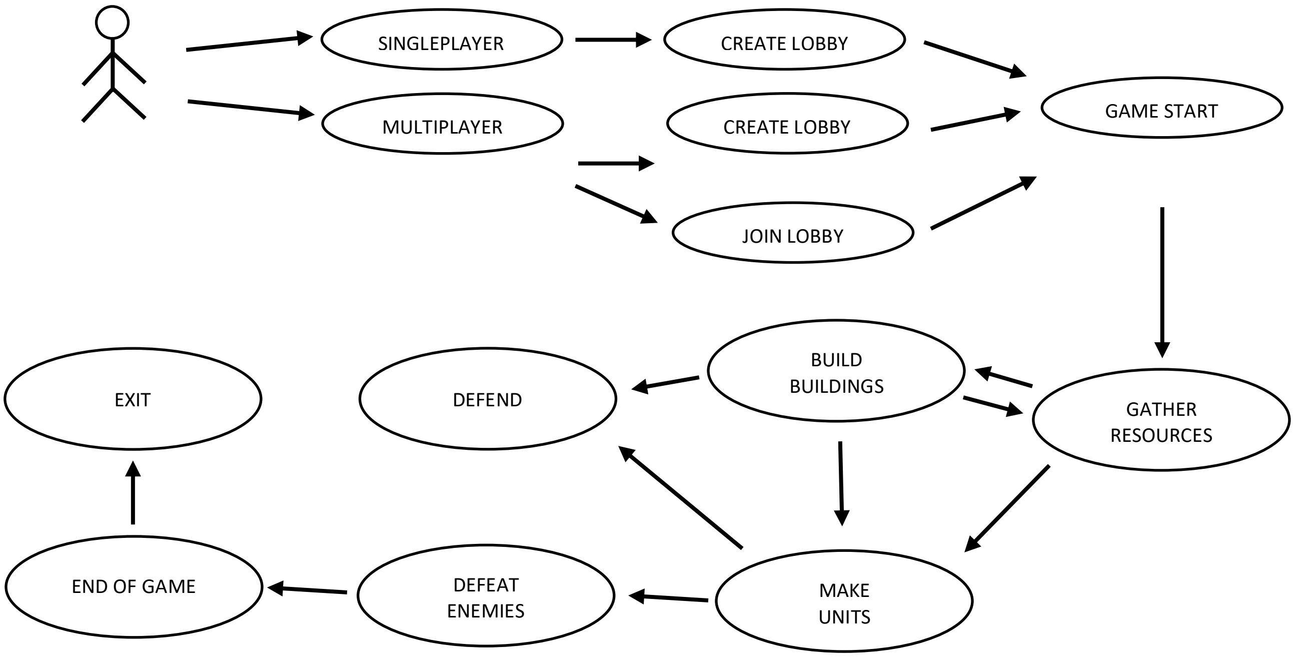

Figure 1: Simple use case

3.3. Object and Class Model

The core of the system is based on several interacting classes:

- Entity (base class): Common properties like position, health, and ownership.
- Unit (derived): Movable, attack-capable entities.
- Villager (derived): Movable, workable.
- Building (derived): Produces units and upgrades.
- Map: Manages a grid of Tile objects.
- Tile: Represents terrain data (grass, water, forest, etc.).
- PathFinder: Implements A* algorithm for movement.
- UI: Manages buttons, panels, and HUD elements.
- NetworkManager: Handles multiplayer synchronization.
- GameState: Maintains current world and unit states.

3.4. User Interfaces

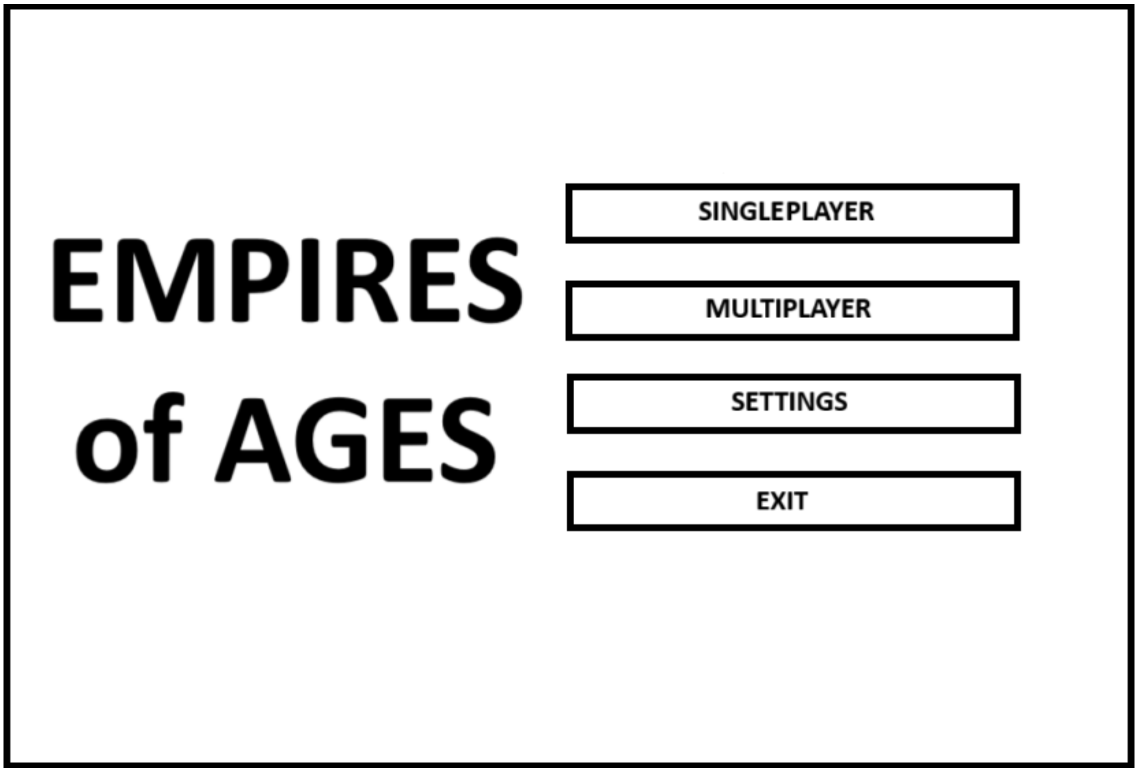

Figure 2: Main menu

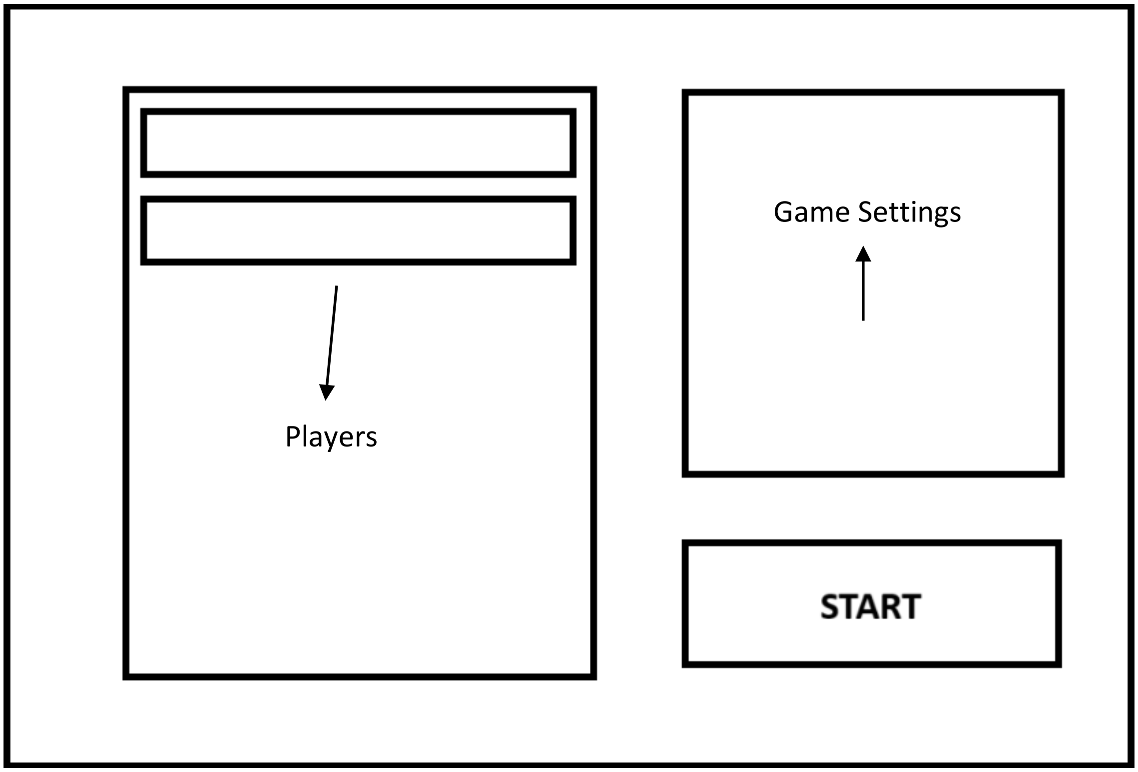

Figure 3: Lobby

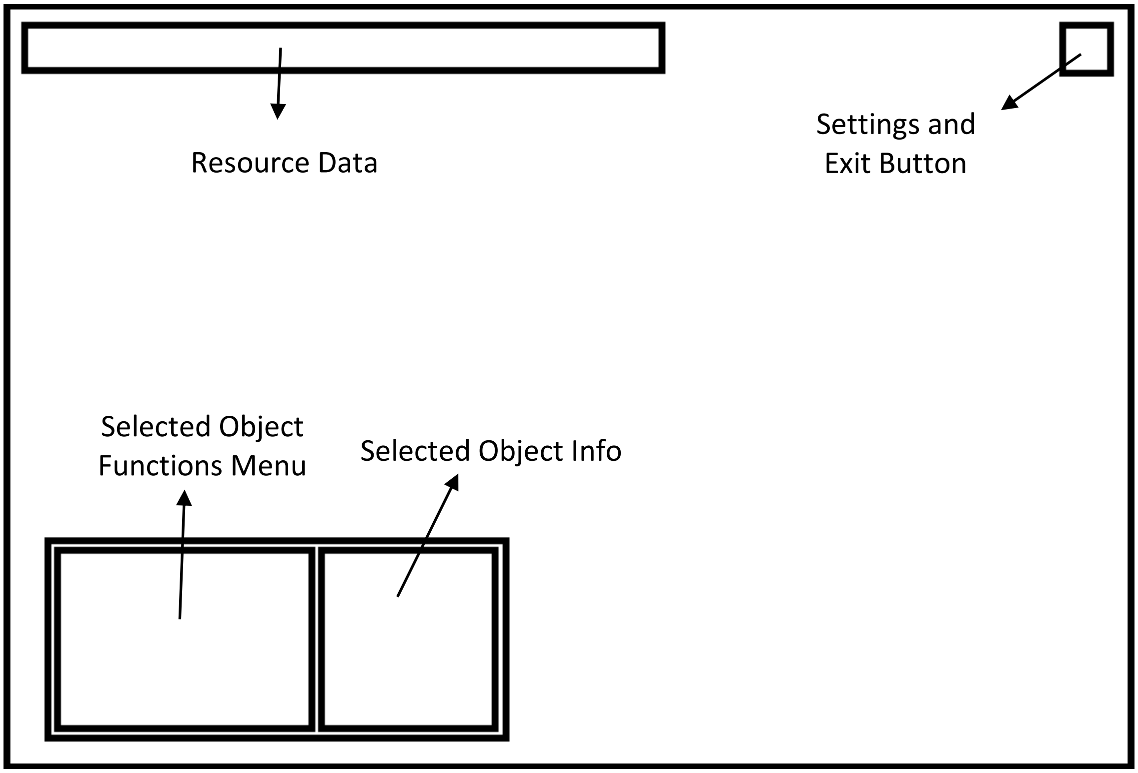

Figure 4: In game

4. Conclusion

This report presents the overall design and analysis of the Group 9 RTS game project. It defines the system requirements, design models, and gameplay flow. Each member contributed to different parts of this report:

- Cahit Onur Enoğlu: System and game management
- Yusuf Yücel: Interface implementation and data management
- Hacı Salih Toker: Network management and LAN systems
- Eren Köse: Map system and pathfinding

All members contributed collaboratively to the final structure and review of the report.