Ankara Yıldırım Beyazıt University
Department of Computer Engineering

CENG 201 – Object Oriented Programming
Course Project

G9: Empires of Ages

Design Report

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

Instructor: Muhammed Abdullah Bülbül
Teaching Assistant: Elif Şanlıalp
Date: 07/12/2025

Table of Contents

1. Introduction

This report presents the refined design stage for Group 9's project, "Empires of Ages," a Real-Time Strategy (RTS) game developed in C++ using the SFML library. Following the analysis phase, this document details the complete object-oriented structure of the system.

The contents of this report include:

- **CRC Cards:** Detailed definitions of responsibilities and collaborations for each class in the system.
- **Class Diagram:** A comprehensive visual representation of the system's architecture, including all classes, methods, and relationships.
- **Conclusion:** A summary of the design phase and individual contributions of the group members.

The goal of this design document is to provide a clear blueprint for the implementation phase, ensuring that the codebase follows SOLID principles and remains modular and maintainable.

2. Class-Responsibility-Collaboration (CRC) Cards

Based on the analysis of our requirements and the class structure, the following CRC cards have been generated for the core classes of the system.

Class Name: Game (GameState)	
Responsibilities	**Collaborators**
- Manages the main game loop (events, update, render).	- Map
- Handles state transitions (Menu, Game, Pause).	- NetworkManager

Class Name: Game (GameState)	
- Initializes the game world and entities.	- UI
- Manages global resources.	- Entity (Unit, Building)

Class Name: Map	
Responsibilities	**Collaborators**
- Stores and manages the grid of Tiles.	- Tile
- Handles loading and rendering of the map.	- Game
- Provides terrain data for pathfinding.	- PathFinder

Class Name: Tile	
Responsibilities	**Collaborators**
- Represents a specific location on the grid.	- Map
- Stores terrain type (grass, water, obstacle).	
- Holds reference to any entity occupying it.	

Class Name: Entity (Abstract Base Class)	
Responsibilities	**Collaborators**
- Defines common properties (ID, position, health, owner).	- Map
- Handles basic rendering and updates.	- Game
- Provides interface for interaction.	

Class Name: Unit (Inherits Entity)	
Responsibilities	**Collaborators**
- Moves across the map using pathfinding.	- PathFinder
- Attacks enemy units or buildings.	- Entity (Target)
- Executes commands given by the player.	- Map

Class Name: Villager (Inherits Unit)	
Responsibilities	**Collaborators**
- Gathers resources from resource nodes.	- Resource
- Constructs and repairs buildings.	- Building
- Inherits movement and health properties from Unit.	

Class Name: Building (Inherits Entity)	
Responsibilities	**Collaborators**
- Produces units (e.g., Barracks produces Soldiers).	- Unit
- Researches upgrades.	- Player
- Provides static defense (if applicable).	

Class Name: PathFinder	
Responsibilities	**Collaborators**
- Calculates the shortest path between two points.	- Map
- Implements the A* (A-Star) algorithm.	- Tile
- Handles collision avoidance logic.	- Unit

Class Name: NetworkManager	
Responsibilities	**Collaborators**
- Establishes LAN connection (Server/Client).	- Game
- Synchronizes game state between players.	- Entity
- Handles packet transmission and reception.	

Class Name: UI (User Interface)	
Responsibilities	**Collaborators**
- Renders buttons, minimap, and resource counters.	- Game
- Handles user input (mouse clicks, key presses).	- Player
- Displays information about selected entities.	- Entity

3. Class Diagram

The following diagram illustrates the complete structure of the "Empires of Ages" system, showing the relationships (inheritance, association, composition) between the classes described in the CRC cards.

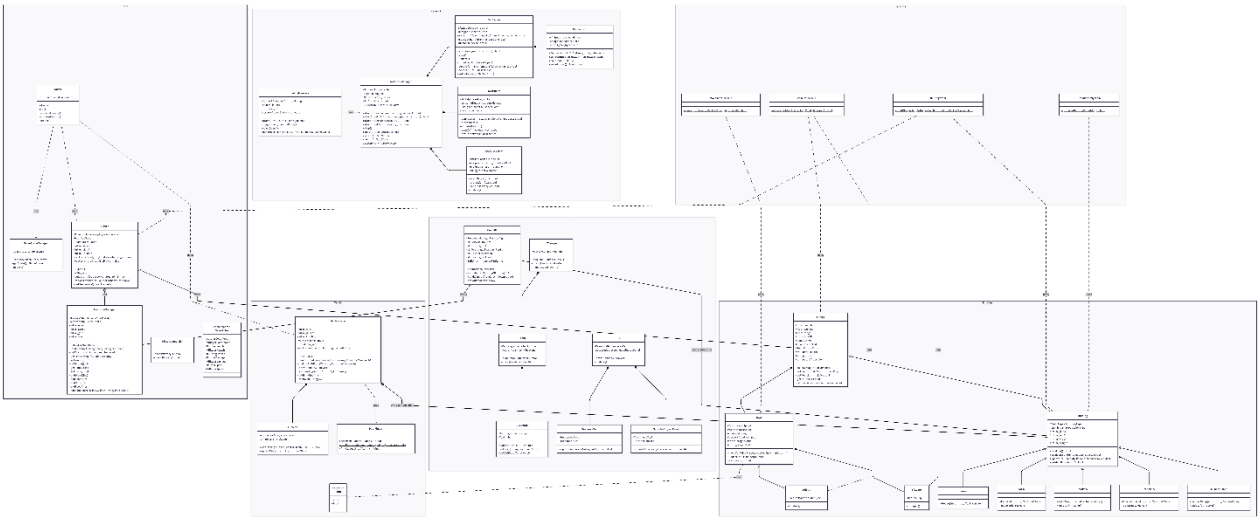

Figure 1:Complete Class Diagram

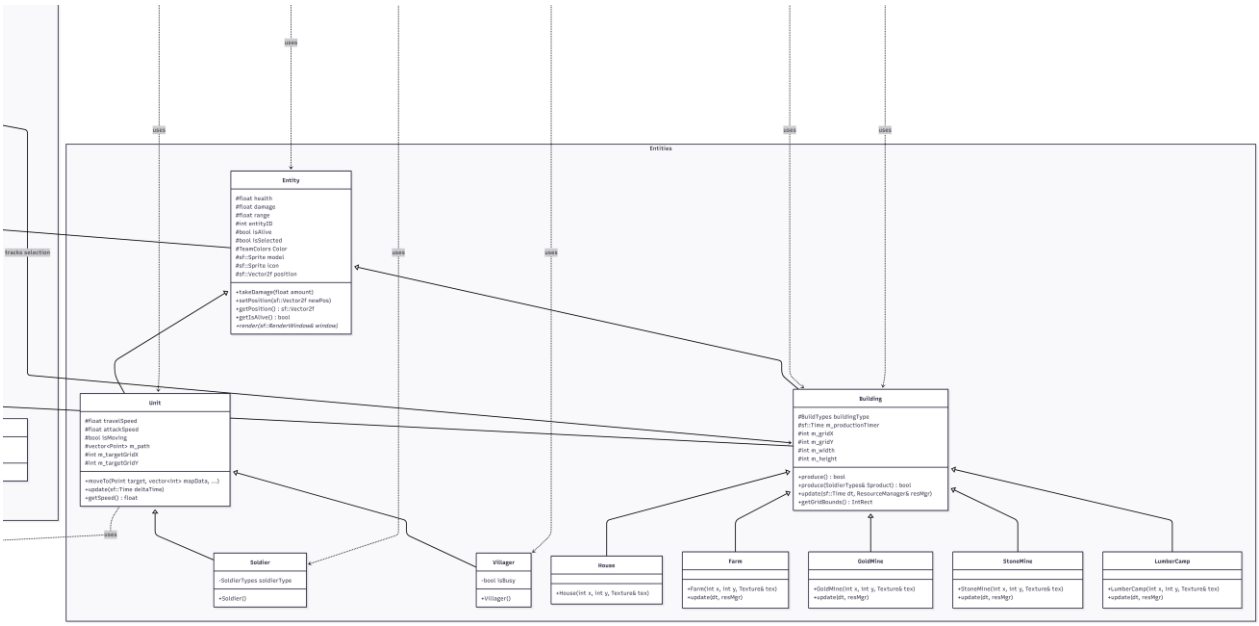

Figure 1.1:Entity & Unit Hierarchy

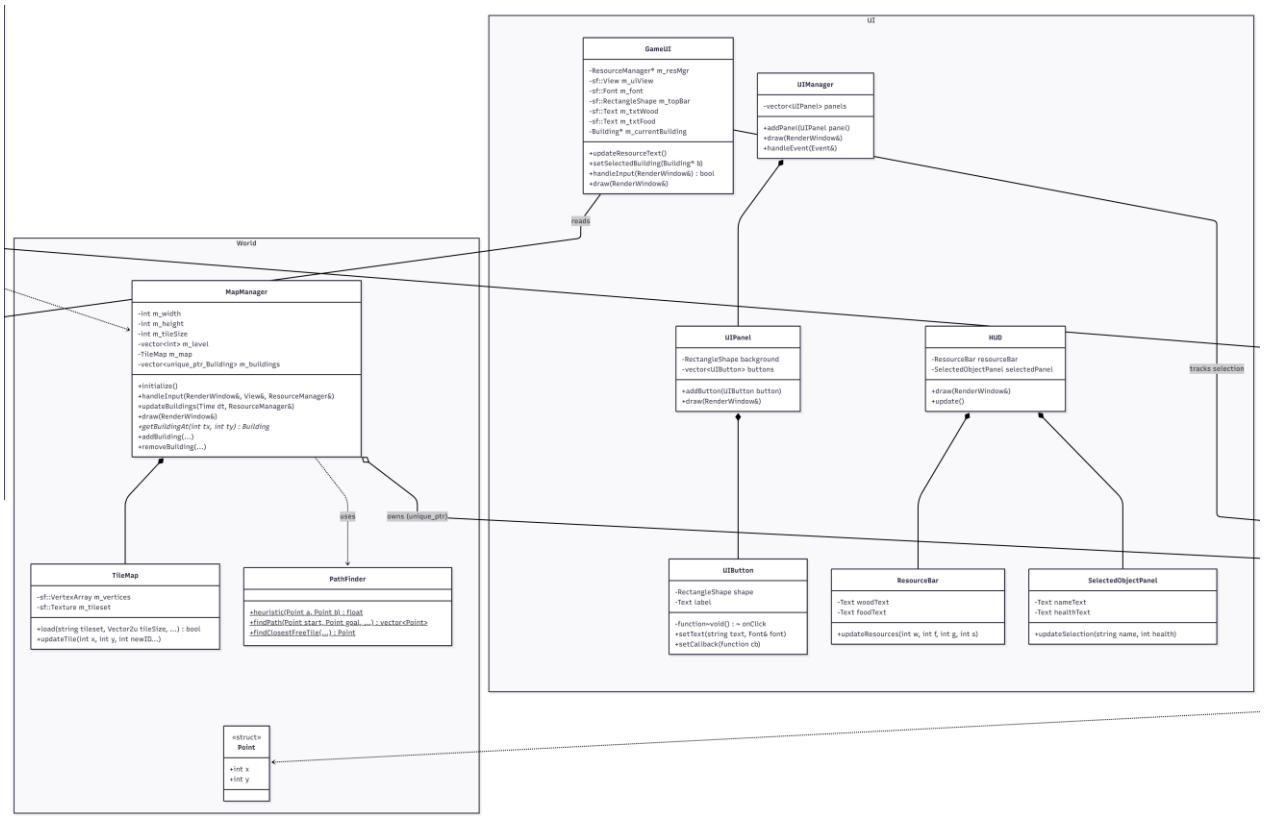

Figure 1.2:Map & UI Hierarchy

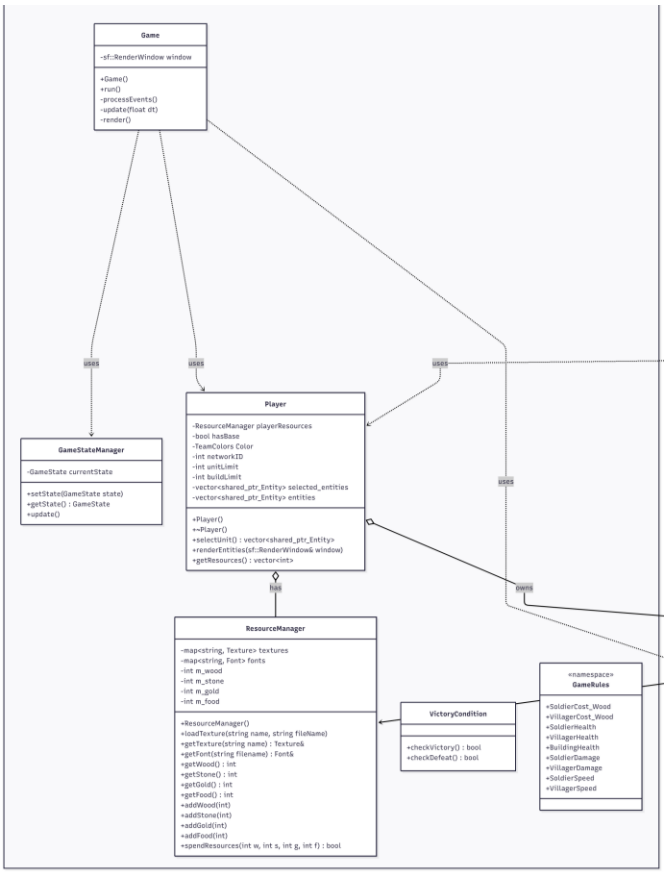

Figure 1.3:Game Hierarchy

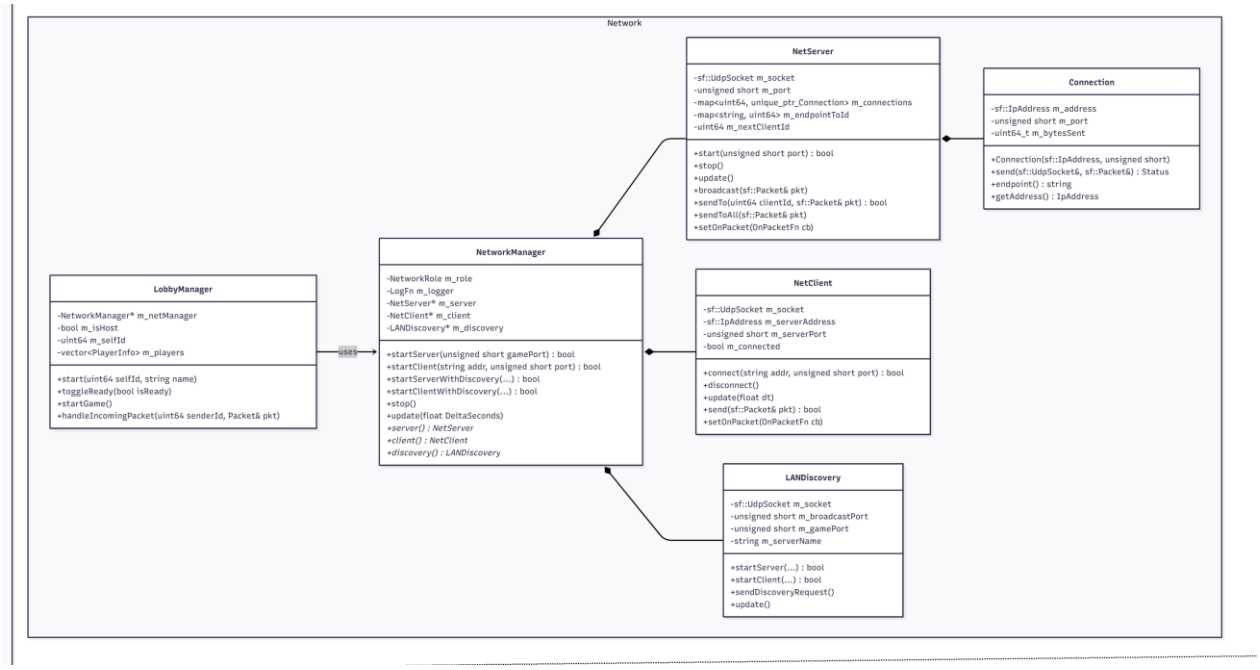

Figure 1.4:Network Hierarchy

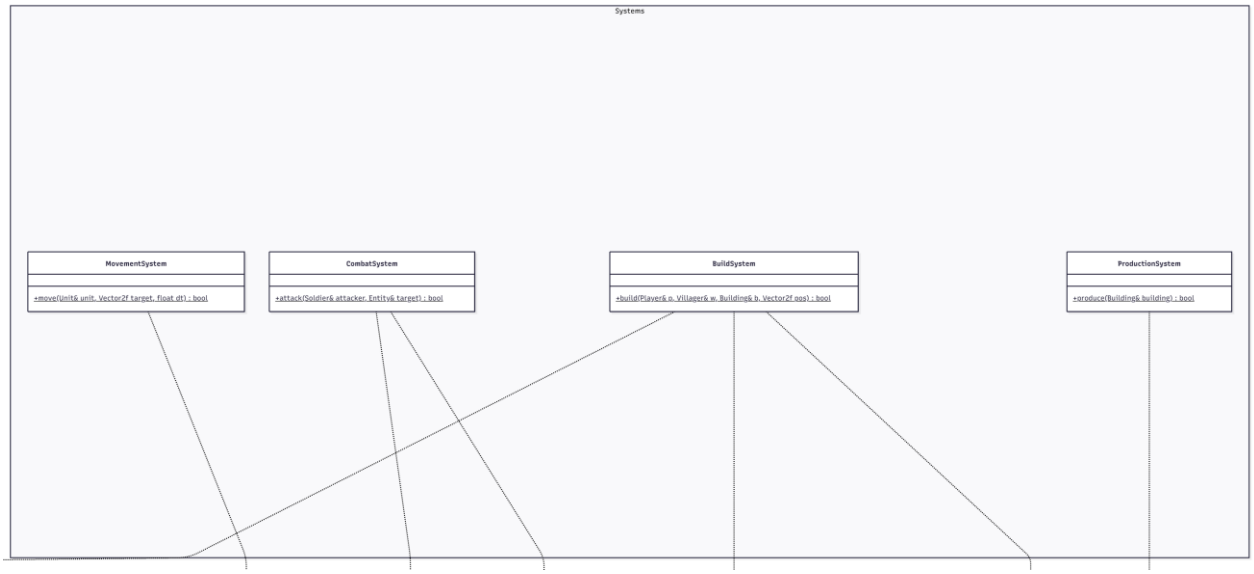

Figure 1.5:System Hierarchy

The diagram highlights the hierarchy where Unit and Building derive from the base Entity class. The Game class acts as the central controller, managing instances of Map, UI, and NetworkManager.

4. Conclusion

This report has detailed the refined design of the Group 9 RTS game project. By generating CRC cards and a complete class diagram, we have established a solid architectural foundation for the implementation phase. The design adheres to the requirements of modularity and maintainability outlined in the analysis phase.

Contributions:

- **Cahit Onur Enoğlu:** Designed the core Game and Entity class hierarchy and integrated system management logic.
- **Yusuf Yücel:** Designed the UI class structure and interaction models for user commands.
- **Hacı Salih Toker:** Designed the NetworkManager architecture for LAN synchronization and state handling.
- **Eren Köse:** Designed the Map, Tile, and PathFinder classes, detailing the algorithms for terrain management and movement.

All members collaboratively reviewed the final class relations to ensure consistency across the system.