



Ankara Yıldırım Beyazıt University  
Department of Computer Engineering

---

## CENG 201 – Object Oriented Programming Course Project

# G19: The Traitor

## Class Design

23050111051, Ahmet Yasin Çetinkaya  
23050111060, Enes Can Bozkurt  
23050151035, Muhammed Yıldız  
23050111049, Muhammed Enes Karaca

**Instructor:** Muhammed Abdullah Bülbül  
**Teaching Assistant:** Elif Şanlıalp  
**Date:** 5/12/2025

## Table of Contents

1. Introduction	2
2. Class Responsibility Collaboration (CRC) Cards	2
3. Class Diagram	3
4. Conclusion	4

## 1. Introduction

This report includes the Class Responsibility Collaboration (CRC) cards and the detailed UML class diagram for the game.

## 2. Class Responsibility Collaboration (CRC) Cards

GameHost	
Manage network connectons.	GameManager
Recieve action requests from all connected clients.	ClientApp
Send game state to players	GameHost

GameManager	
Maintain the central game logic and rules	Player
Manager the game clock and transitions between phases	Phase
Validate and process ActionPackets received from the Host	Action
Execute actions and update the game state	

ClientApp	
Establish and maintain tcp connection with the server	GameHost
Manage the main game loop	Game View
Receive GameState updates from the server	InputHandler
Send ActionPackets to the server.	
Manage the application window	

GameView	
Render the visual representation of the game	
Display the event log and toggle its visibility	ClientApp
Render available action options based on the current state	

InputHandler	
Listen for raw user inputs	
Specify inputs into game-understandable ActionPackets	

Figure 1&2 - View and Controller Classes

Phase	
Define a specific period of gameplay	GameManager

Country	
Track resource values	
Track the state of anarchy	
Provide getters for current stats	

(Abstract) SecretAction	
Execute hidden game logic on target players	Player
Generate a log string	Country
	SpreadPlague, SabotageFactory, Destroy School

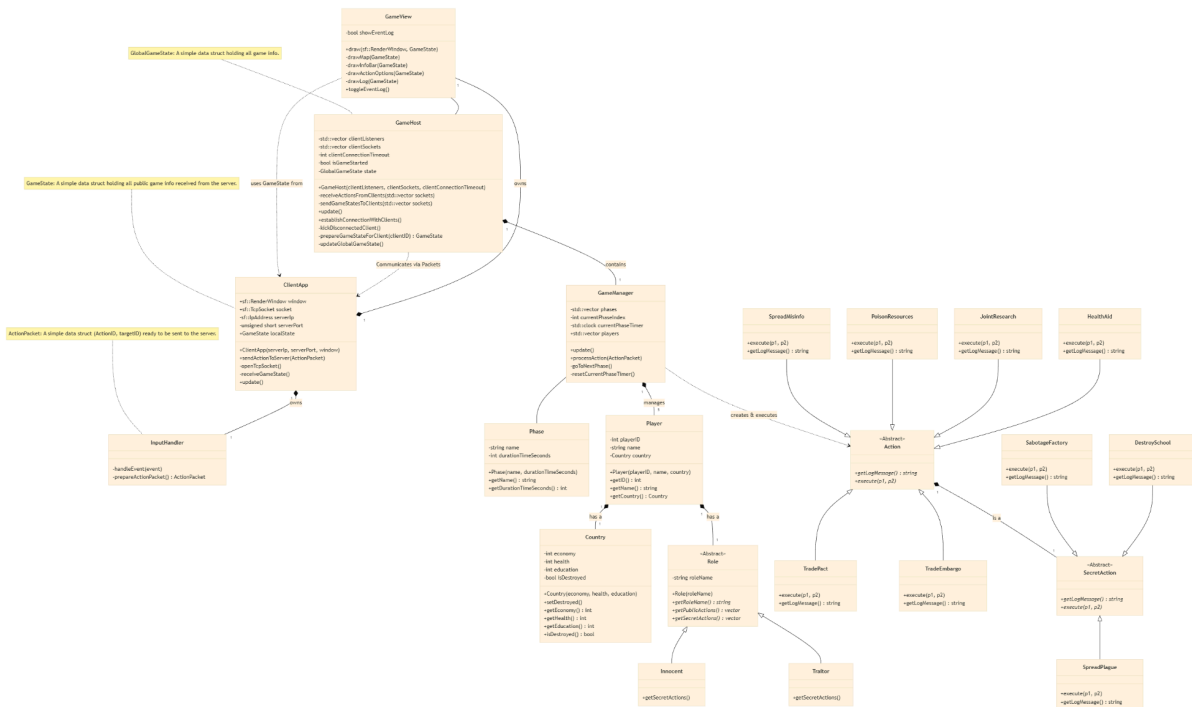
(Abstract) Role	
Define the archetype of the player	Action
Provide list of allowed public and secret actions.	SecretAction

(Abstract) Action	
Execute specific game logic on target players	Player
Generate a log string describing the event	Log
	TradePact, TradeEmbargo, JointResearch, SpreadMisinfo, PoisonResources

### Figure 3 - Game Model Classes

### 3. Class Diagram

The following diagram illustrates the Client-Server architecture, identifying the separation between the GameHost (Server logic) and the ClientApp (Player view and input), as well as the inheritance hierarchy for Roles and Actions.



## 4. Conclusion

The class design and responsibility and collaborations of the game classes are explained in this report. The architecture relies on a GameHost to manage the GlobalGameState while individual ClientApp instances receive GameStates. ClientApp handles rendering and input via GameView and InputHandler. The action system allows the traitor to do secret actions along with public actions.

Work Distribution:

Class Diagram: Muhammed Enes Karaca, Ahmet Yasin Çetinkaya

CRC Diagram, Introduction, Conclusion: Muhammed Yıldız, Enes Can Bozkurt