Ankara Yıldırım Beyazıt University
Department of Computer Engineering

CENG 201 – Object Oriented Programming
Course Project

G19: The Traitor

Analysis Report

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

Instructor: Muhammed Abdullah Bülbül
Teaching Assistant: Elif Şanlıalp
Date: 14/11/2025

Table of Contents

1. Introduction

The Traitor is a 5-player multiplayer game. There is one randomly chosen player which is the traitor, and four others are innocents. Initially, all players have one country. The innocents goal is to find the traitor and destroy its country. The traitors goal is to destroy all 4 innocent countries.

Each country has 3 stats: economy, health, education. They are all 100 at start. Innocent players have actions such as trade pact (+10 own +10 target economy), trade embargo (-10 own -15 target economy) etc. The traitor also has secret actions like spreadplague (-30 target health).

Game starts with discussion phase, then action phase, and then resolution phase. After that, game continues again starting with discussion phase until winning conditions meet.

In this report, user interface (UI) , UML class diagram and game mechanics explained in detail.

2. Requirements

2.1. Functional Requirements

- The system must be able to start a game session with 5 players.
- The system must be able to give a role ("Traitor" or "Innocent") to each player at the beginning of the game.
- The system must show an information screen to each player showing only their own role.
- The system must provide the "Traitor" role player the ability to select another player as a target for attacks.
- The system must have the option for all players to make “actions” (e.g., trade pact, trade embargo) to other players.

- The system must give all players the action to do a "treaty embargo" targeting another player.
- The system must provide all players the ability to send an "intelligence" message (about other players) .
- If a "virus attack" occurs, the system must publicly display the identity of the player who was attacked to all players.
- The system must give incoming "intelligence" only privately to the receiver of the intelligence.
- When a "treaty embargo" is applied to a player, the system must display this status as a notification only to the player under the embargo.
- The system must activate treaties and automatically update the relevant players' statistics.
- The system must pause the game flow after all actions and treaties are resolved.
- When the game pauses, the system must open a "discussion screen" allowing all players to communicate.
- When the discussion phase ends, the system must proceed to a new turn (a new "year").

2.2. Non-Functional Requirements

- Pace: Pace must be dependent to a variable in order to optimize (and let the player change) pace of the game.
- Capacity: The system must be able to run a game session with exactly 5 players.

3. System Models

3.1. Scenarios

Scenario Title: The First Year in the Game

Players and Roles (Assigned secretly by the system):
* Player 1: Traitor
* Player 2: Innocent
* Player 3: Innocent
* Player 4: Innocent
* Player 5: Innocent

Stage 1 (Game Start)

System: The game has started.
System: Each player’s role has been shown to them.

Stage 2 (Actions)

Player 1: Launches a virus attack on Player 2.
Player 2: Sends a trade pact to Player 3.

Player 3: Imposes a trade embargo on Player 1.
Player 4: Sends a trade pact to Player 2.
Player 5: Spreads misinfo to Player 4.

Stage 3 (Reveals)

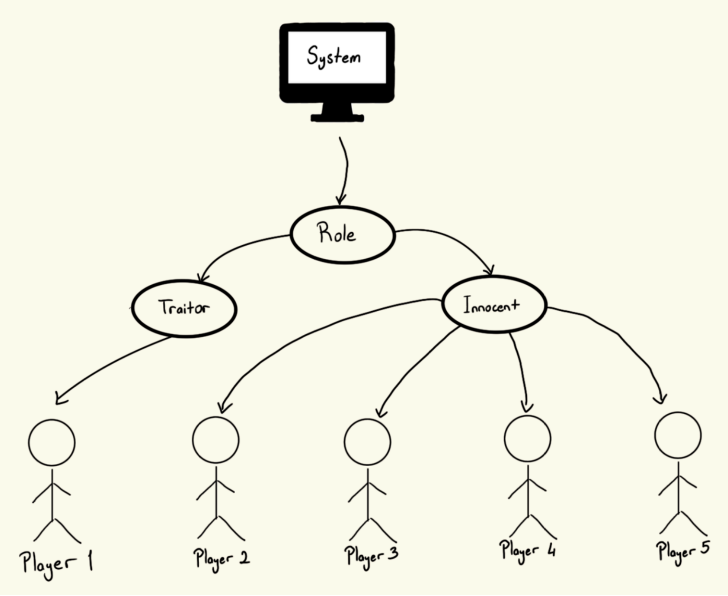
System: Executes the actions between players and adjusts the necessary statistics.
System: Everyone sees that Player 2 was attacked with a virus.
System: Player 4's trade pact with Player 2 shown to all players.
System: Player 2's trade pact with Player 3 shown to all players.
System: Player 5's spreading misinfo to Player 4 is shown to all players.
System: Player 1's screen shows the message "Trade embargo from Player 3 applied."

Stage 4 (Execution)

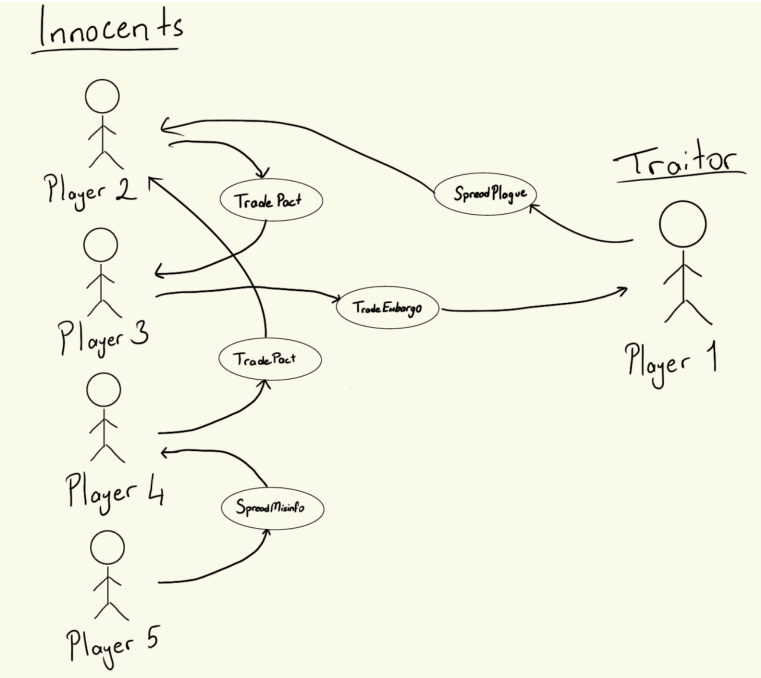
System: Freezes the game.
System: Displays the discussion screen to all players.

(Players discuss among themselves, and the next round begins again from Stage 2.)

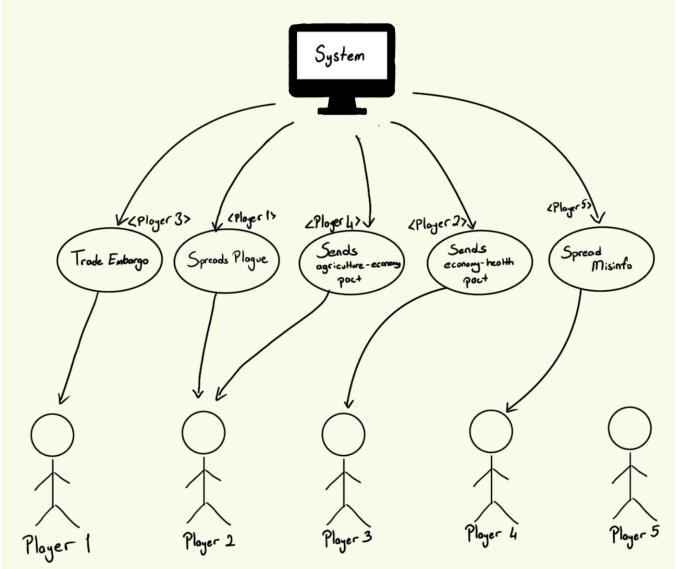
3.2. Use Cases



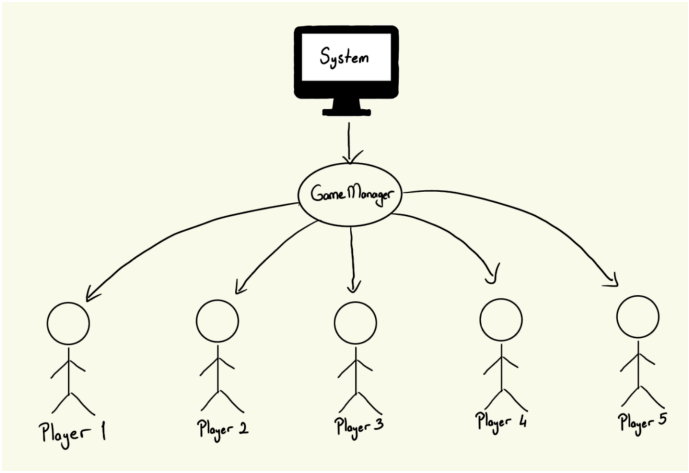
Stage 1



Stage 2

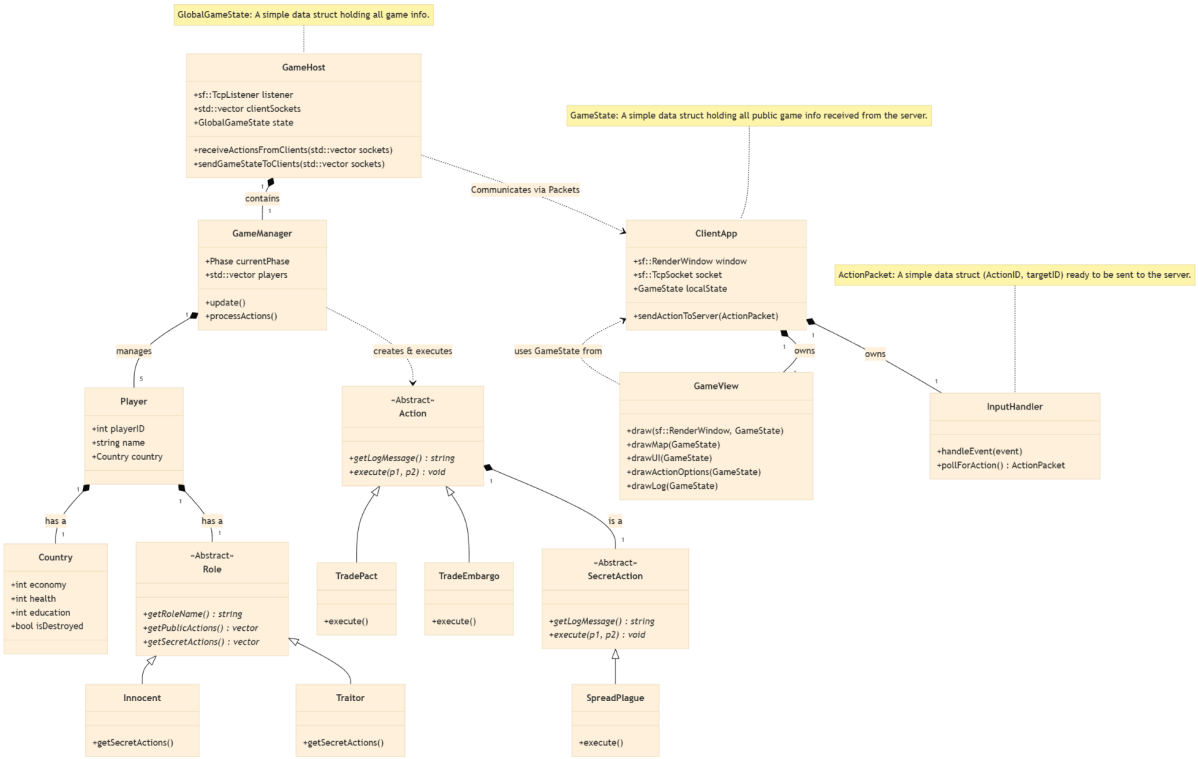


Stage 3

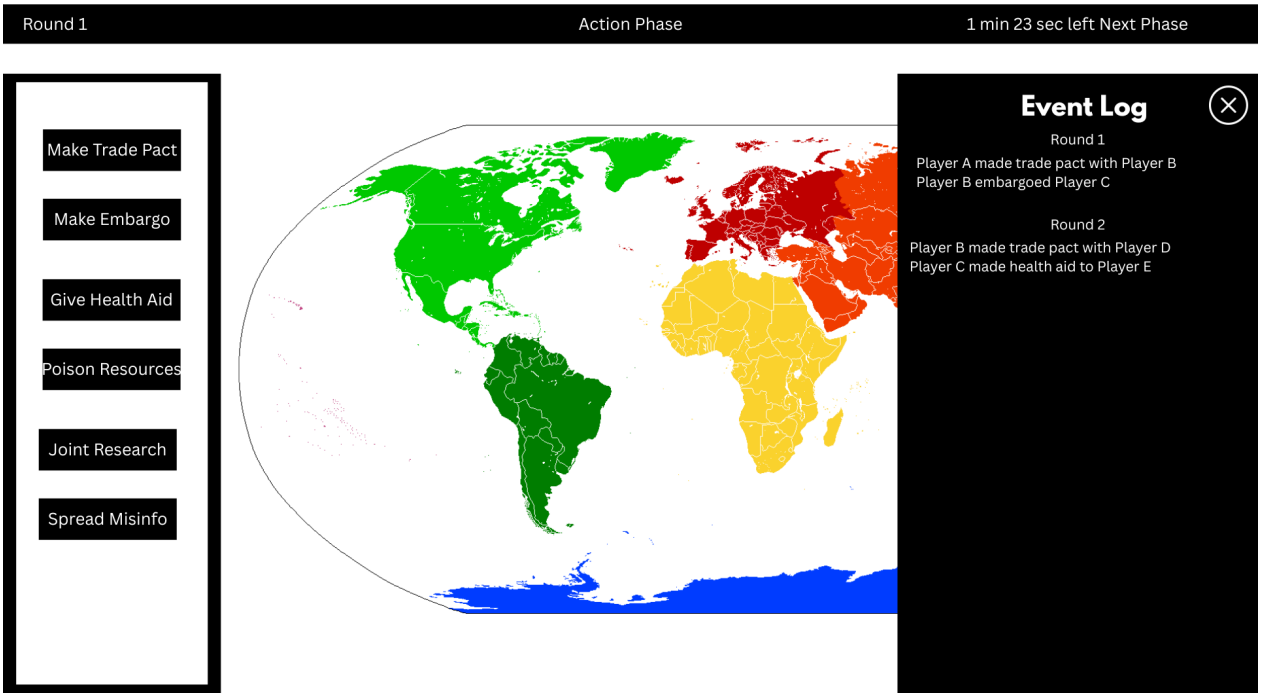


Stage 4

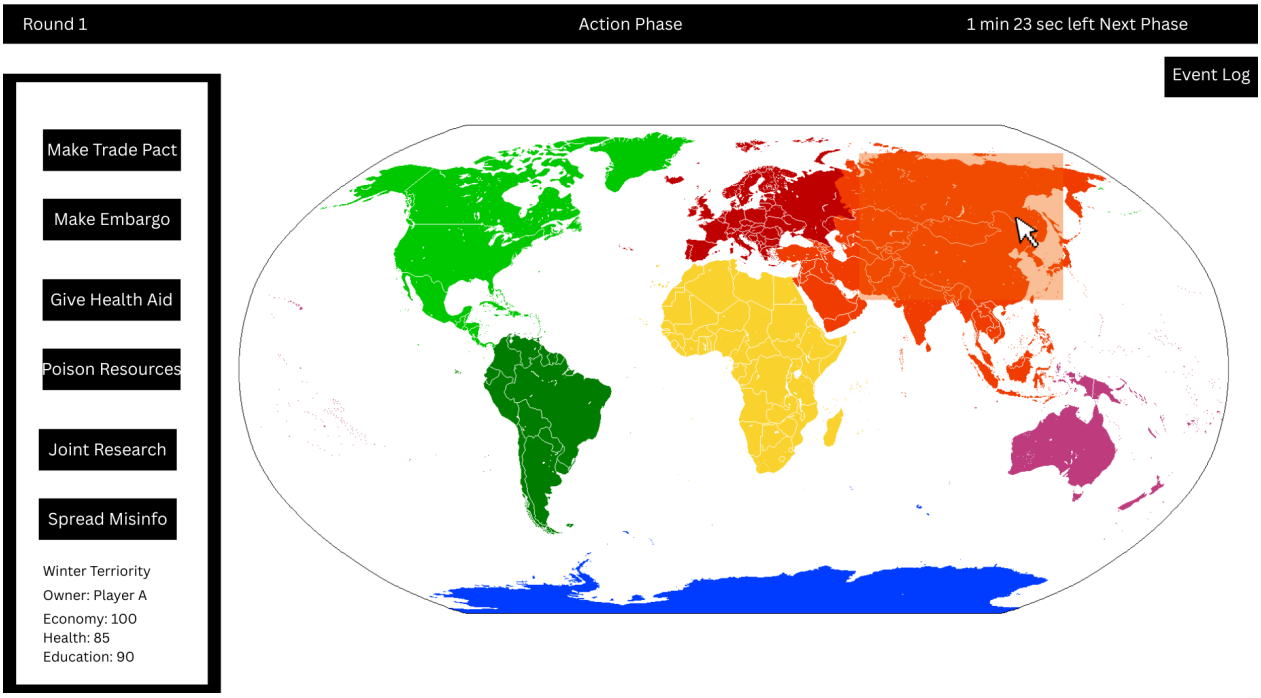
3.3. Object and Class Model



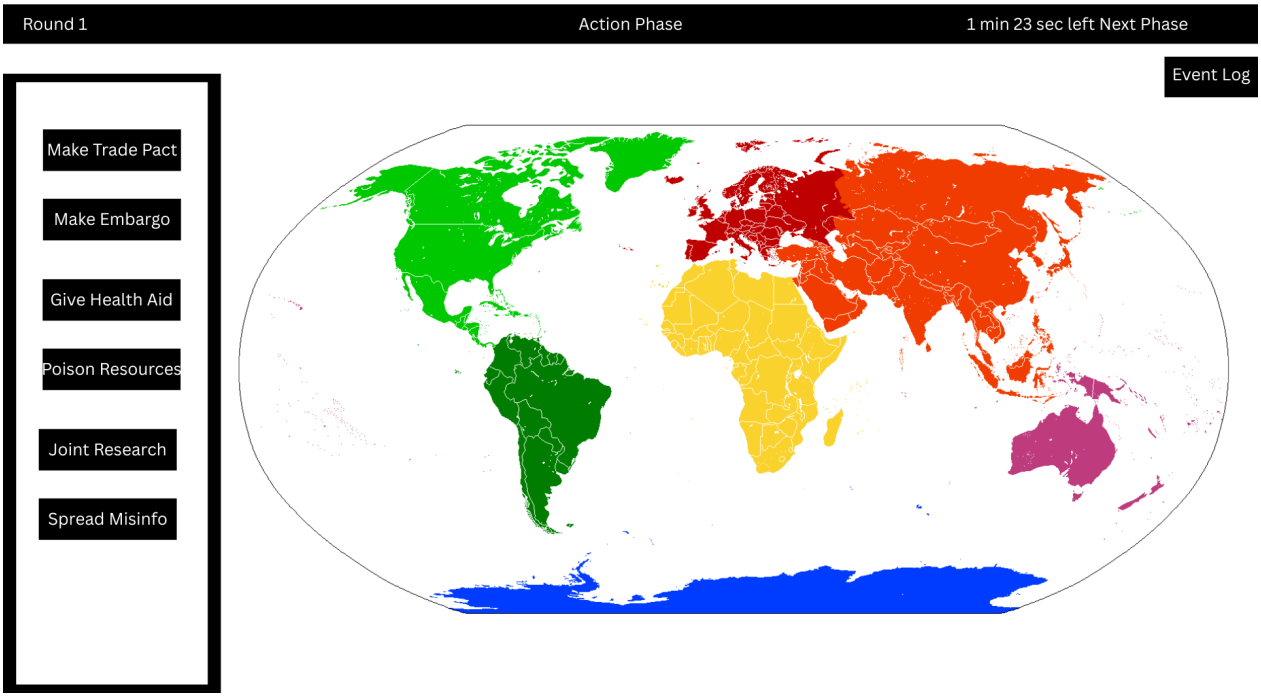
3.4. User Interfaces



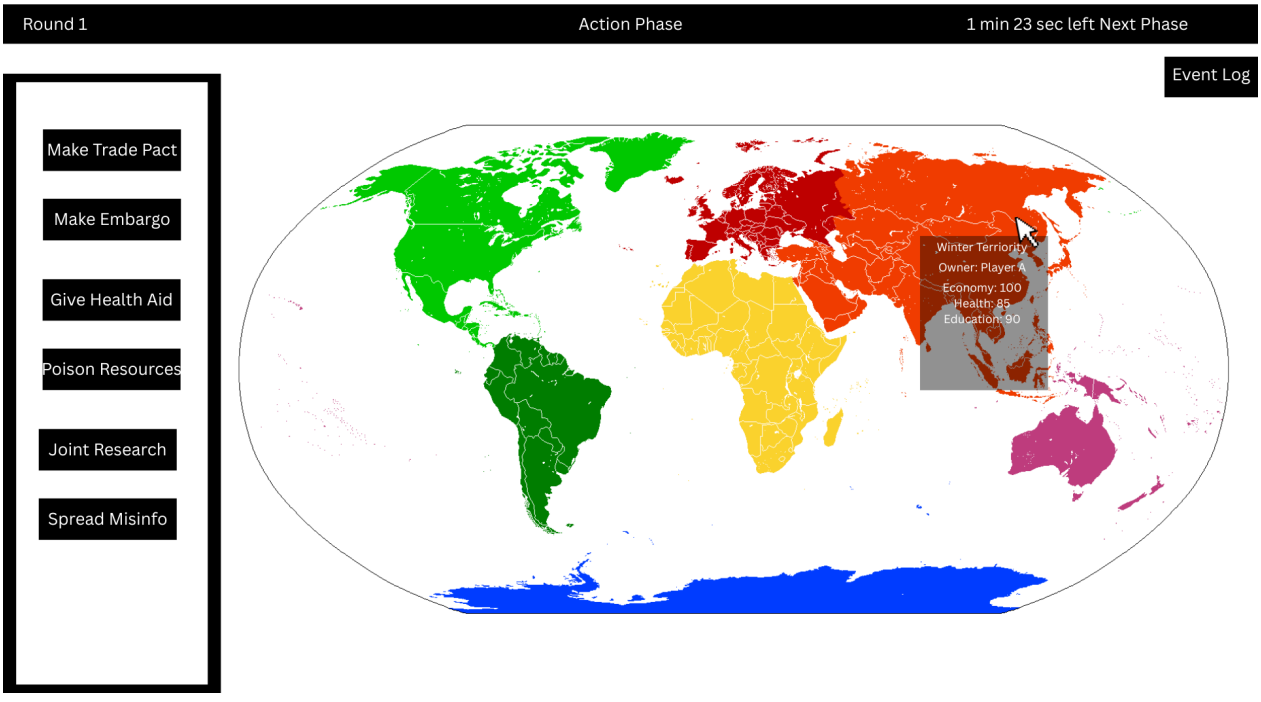
Draft 1. Event Log Panel.



Draft 2. Selecting a country.



Draft 3. Action Phase.



Draft 4. Hovering over a country.

4. Conclusion

The main idea and early design of the game is explained in this report. The small details like UI buttons or actions effects may subject to change.

Functional and non-functional requirements - Enes Can Bozkurt

Use cases and scenarios - Ahmet Yasin Çetinkaya
User interfaces and class diagrams - Muhammed Enes Karaca & Muhammed Yıldız