

DS-Monopoly

Daniele Loverre



Concept

The system is a turn-based multiplayer Monopoly clone.

Players (2 to 4) can play a simplified version of the board game.

They can buy properties or houses, roll dice and

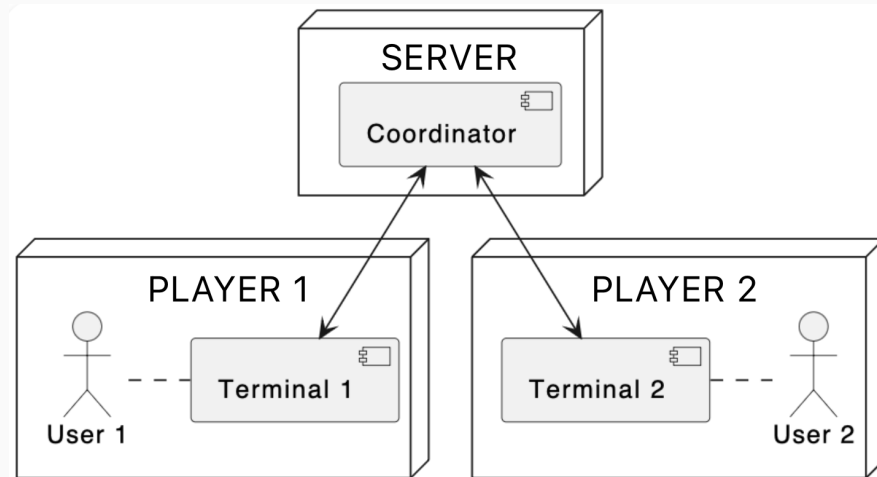
see the board and the players information from a GUI.

Technologies

- **Language:** Python3
- **Network:** TCP Sockets
- **GUI:** PyGame
- **Data format:** JSON
- **Deployment:** Docker

Architecture

The system adopts a **Client-Server** architecture, where the Clients are the players and the Server is the game coordinator.

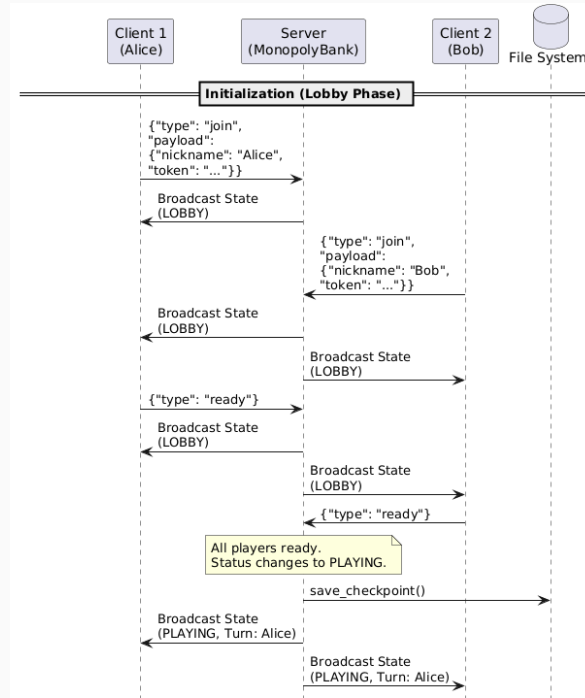


Architecture

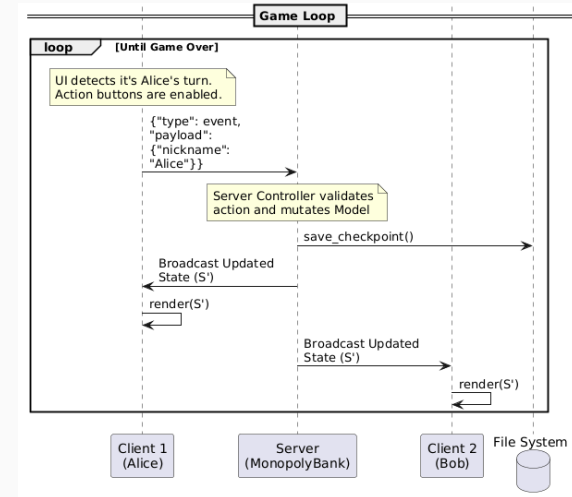
The system is divided into three distinct tiers:

- **Presentation Tier (Client):** Responsible exclusively for the user interface (PyGame).
- **Application/Logic Tier (Server):** It contains the game logic and acts as the authoritative coordinator.
- **Data Tier (Filesystem):** The components interact directly with the local filesystem to store information.

Lobby phase:



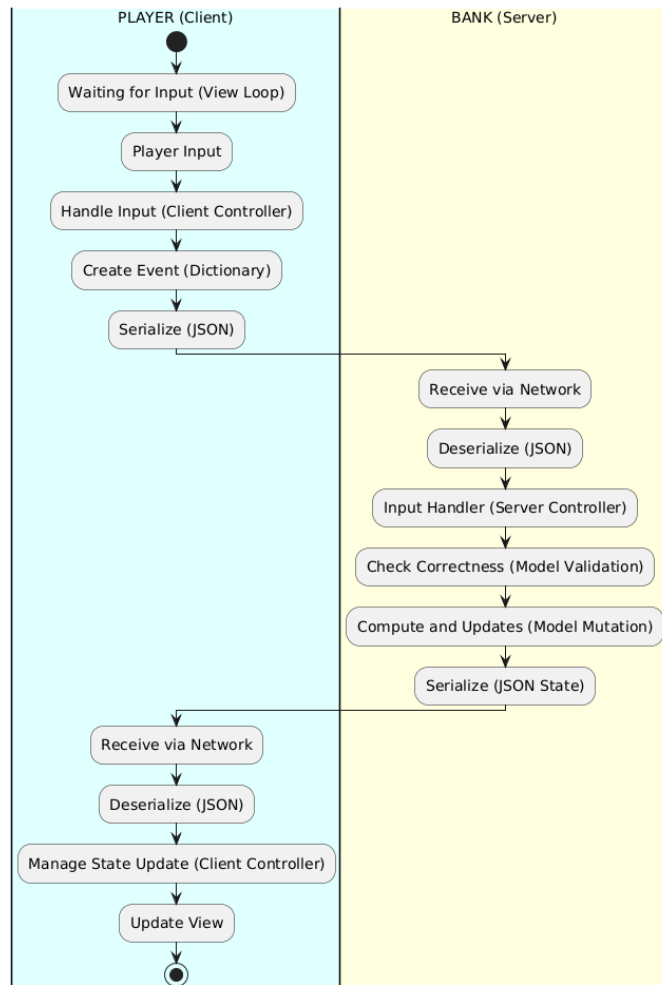
Game loop:



Architecture

Event-Driven Interaction:

1. The server waits passively for network events.
2. Valid events mutate the state and trigger a JSON broadcast of the updated state to all clients.



Distributed system feature

Consistency

The problem:

- Synchronize game state for across up to 4 nodes.
- Preventing cheating.

The solution:

- Centralized server act as a single source of truth that broadcast the game state to clients.
- Clients cannot change the game state directly, they only receive and render the state.

Fault Tolerance and Recovery

The problem:

- Unreliable networks.
- Sudden disconnections.
- Process crashes.

The solution:

- **Checkpointing:** The game state is dumped to an autosave.json file.
- **Authentication:** UUID tokens are stored on the client to prove identity upon reconnection.
- **Timeouts:**
 - 30 seconds on clients drops before player bankruptcy.
 - 60 seconds on server drops before game closed.

CAP Theorem

- **Partition Tolerance:** Network drops are inevitable and detected via TCP timeouts.
- **Consistency (Prioritized):** All players must observe the exact same state. Desynchronization is not acceptable.
- **Availability (Sacrificed):** Upon a network partition, the system temporarily halts the game rather than allowing disconnected clients to continue playing in a diverged state.

Main Weaknesses

- **Single Point of Failure:** There is a single server that compute and keeps the game state, so if the host machine dies or lose connection (for more than 90 seconds) the entire game halts and the player cannot continue that game anymore.
- **Security:** Players tokens are stored and sent in clear text without any form of encryption.
- **Manual Network Discovery:** When connecting across different physical machines, especially if the server is hosted via Docker, users must manually look up and input the host's IP address, which makes finding the server less user-friendly.

Future Works

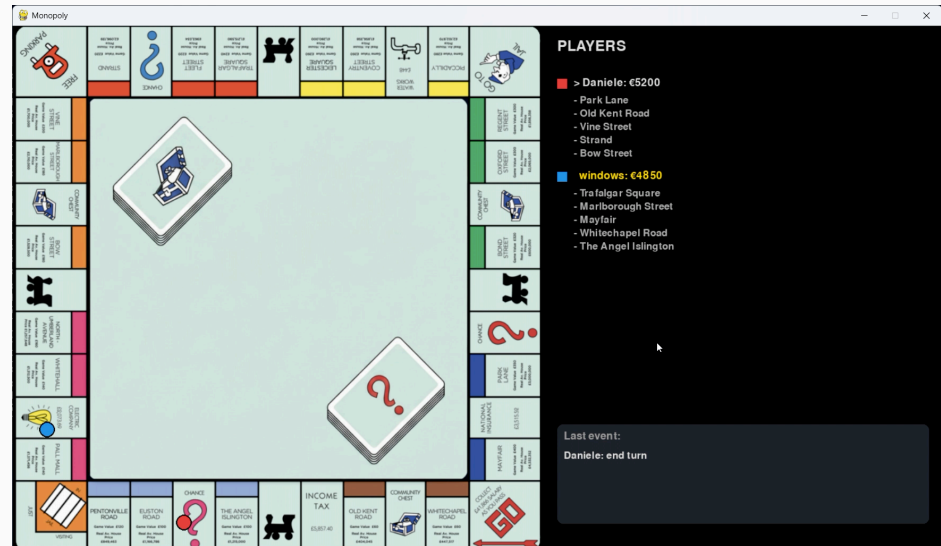
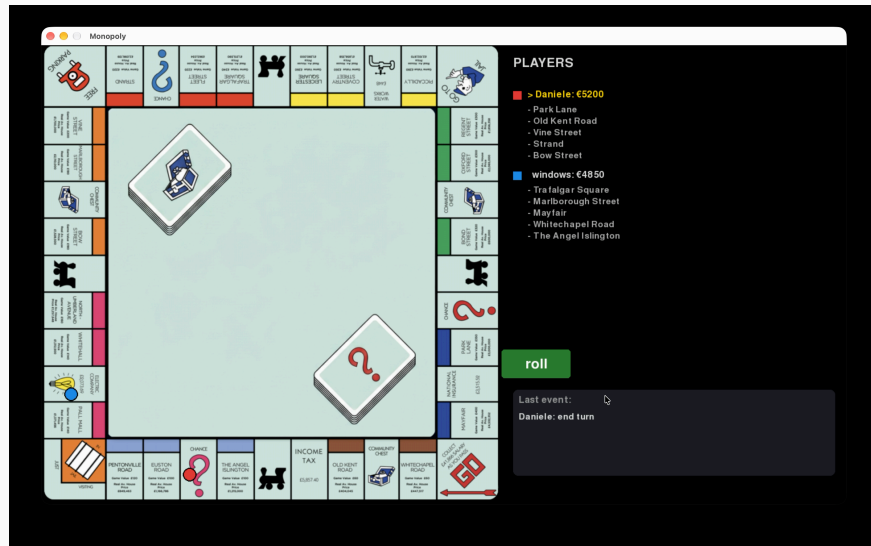
- **Removing the Single Point of Failure:** A more robust and highly available setup would involve deploying multiple server nodes, so that if one server node crashes another can pick up the ongoing session.
- **Concurrent Game Sessions (Matchmaking):** Upgrading the server into a central hub where users can create and join multiple concurrent public or private game rooms (lobbies), each running as an isolated game thread.
- **Adding a Database:** Adding a database could be useful for both increasing availability and fault tolerance by saving on it checkpoints and logging, and implementing an account management system that allows players to register with a username and password, keeping track of their game history, win rates, and friends lists.

Live Demonstration

Standard Game Loop



Client Failure



```
[NETWORK] connection lost: Network partition detected (timeout)
[SYSTEM] Reconnection timer of 30.0 seconds started...
[NETWORK] connection lost: None
Open ingoing connection from: ('192.168.1.13', 55069)
[SYSTEM] Authentication succeeded for 'windows'.
[SYSTEM] All the player are back. Reconnection timer cancelled.
```

Server Failure



```
[NETWORK] connection lost: Network partition detected (timeout)
[SYSTEM] Reconnection timer of 90.0 seconds started...
[NETWORK] connection lost: None
Open ingoing connection from: ('192.168.1.9', 53027)
[SYSTEM] Authentication succeeded for 'Daniele'.
Open ingoing connection from: ('192.168.1.9', 53028)
[SYSTEM] Authentication succeeded for 'Dani'.
Open ingoing connection from: ('192.168.1.13', 52174)
[SYSTEM] Authentication succeeded for 'windows'.
[SYSTEM] All the player are back. Reconnection timer cancelled.
```

Thanks!