Project: Jarl
Design document

Group Number: 18

Magnus Andermo
Mattias Frånberg
Carl Johan Gustavsson
Pontus Stenetorp

Contents

1 Introduction

2 System Overview

2.1 General Description

2.2 Overall Architecture Description

2.2.1 Lobby server

The lobby server is responsible for keeping data about games that not yet have been started. It's also responsible for delivering data to the lobby clients. When a new game session is created the lobby server is responsible for spawning a new game server for that session.

The lobby server consists of a network module, a game server handler, a database module and the lobby handler.

2.2.2 Game server

The *Game server* consists of a network module, a client handler and a proxy module.

Client handler
The *client handler* filters game commands arriving from the clients and forwards them to the *client controller* or the *proxy*. Client control commands are commands regarding the overall game, for example leaving, winning and loosing. Proxy commands are commands broadcasted to *all* clients for example commands that the user issues in the GUI that changes the game world.

2.2.3 Lobby client

The lobby client is responsible for connecting a user to the lobby server. Through the lobby client the user is able to communicate and join non-started game sessions and then be able to join game sessions.

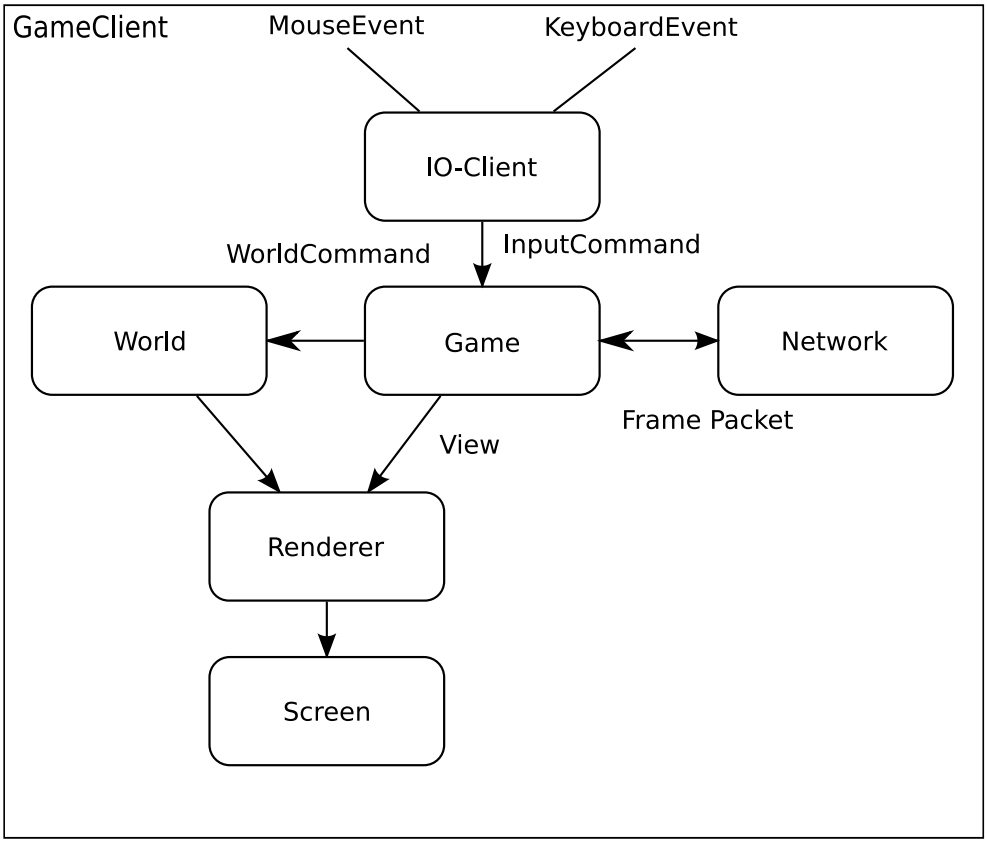

Figure 1: Picture describing how the Lobby server sub systems interact.

2.2.4 Game client

The game client is responsible for holding a current game state and present it to the user. The user can affect the game state by giving the game client input which is then sent to the game server. The game client will continuously receive game state updates from the game server and apply the changes to it's game state.

2.3 Detailed Architecture

2.3.1 Lobby server

Lobby
The Lobby handles requests from the client regarding game session states. It provides the client with information about the game sessions. When a game session should start the Lobby mediates between clients and the game server handler and is responsible for that the game server handler has all the information required to start the game session.

Game server handler

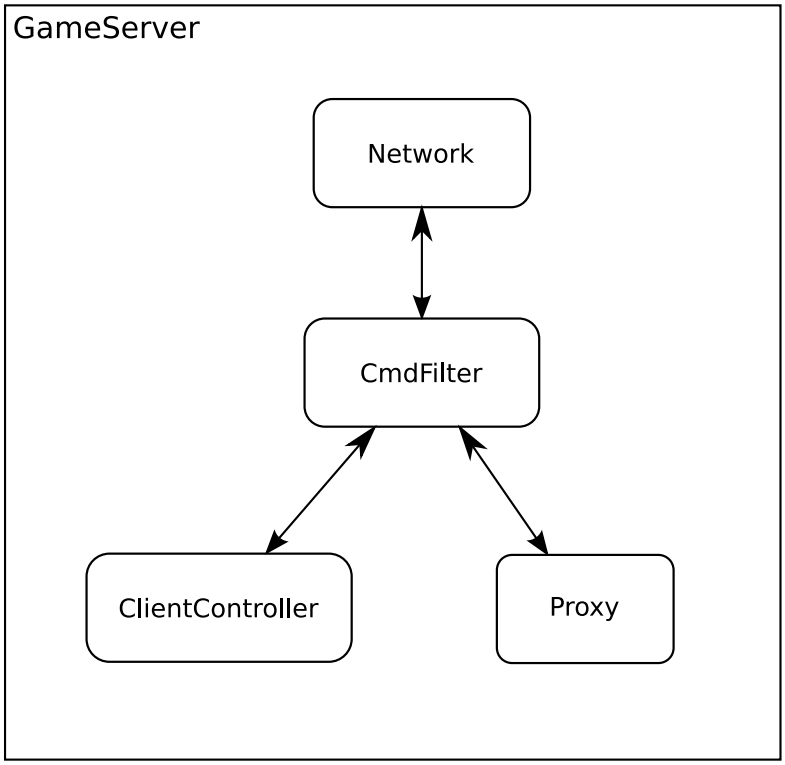

Figure 2: Picture describing how the Game server sub systems interact.

The game server handles the interaction between the Lobby Server and the Game Servers. The Game server handler is responsible for starting up Game Servers. The game server handler is part of the Lobby Server.

User database
Contains information about all users and their wins and losses.

2.3.2 Game server

Network
The commands coming through the external networks are either Game commands or commands that handles interaction between the client and the server, for example a client disconnecting.

When a client sends a game command that command will get executed at the same frame on all clients, because all clients have the same state and random seed the command will result in the same new state for all clients. The Artificial Intelligence will act based on the same state and random seed on all clients, so the AI decisions will be the same for all clients.

Command filter
The CmdFilter distinguishes between interaction and game commands and for-

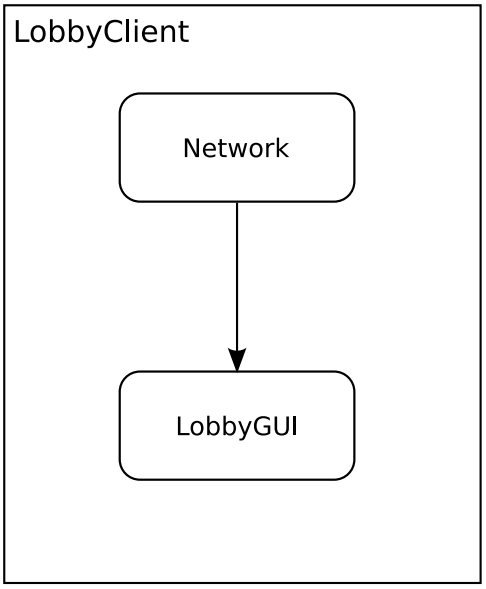

Figure 3: Picture describing how the Lobby client sub systems interact.

wards them to either the ClientController or the Proxy.

Client controller
The ClientController handles all the management of the clients.

Proxy
The proxy module of the game server is responsible for distributing game commands to the clients. The clients send commands that represent user commands. The server runs frames with a fixed frequency. All commands that where received during a server frame should be executed at the same frame on all game clients. The server signals the end of a server frame by sending a frame end command. Consequently the frame end commands will be sent from the server at equally spaced intervals, but the commands will not arrive at equally spaced intervals, due to difference in network latency. If we execute the frames the moment we receive them from the server, the time the frames will be shown to the user will variate. If that variation in frame time will be clearly observable and affects the gameplay experience, we will compensate for that by having a buffer of frames and use heuristic that will decide when to execute the next frame.

2.3.3 Lobby client

Lobby
The lobby client presents a lobby which consists of data received from the lobby server. The data is presented using a GUI. The data presented is the current joinable games and the players currently connected to the server. It also allows the user to create a new game using the GUI.

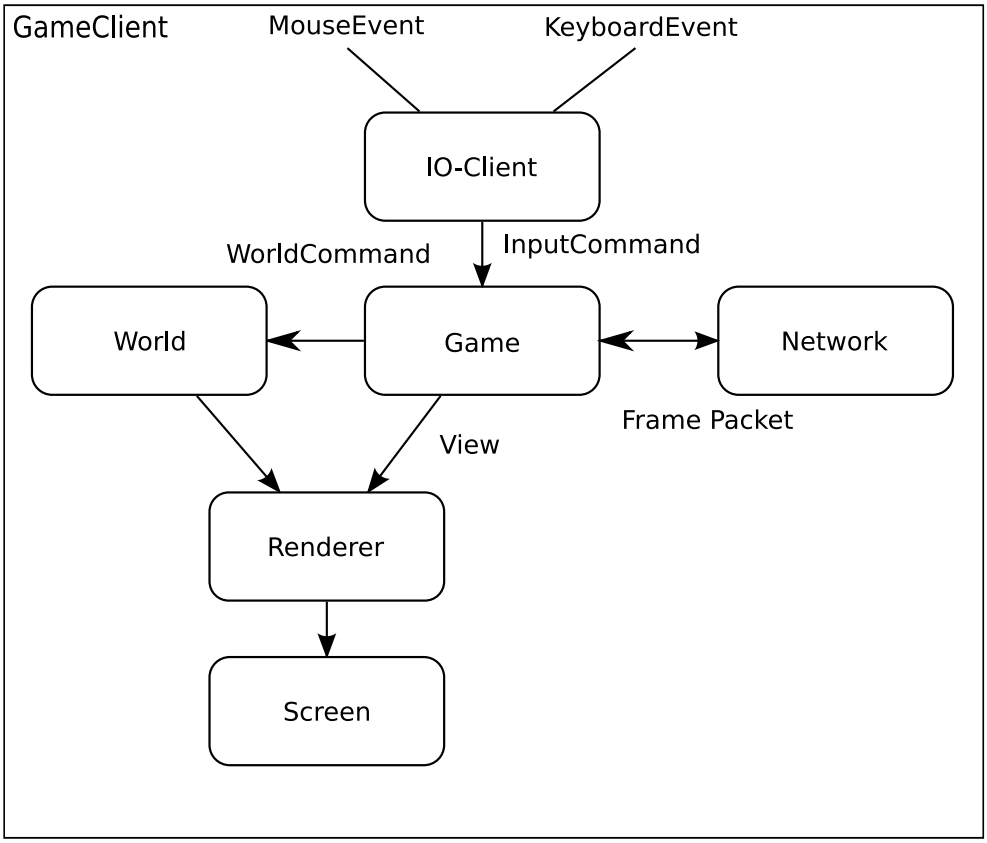

Figure 4: Picture describing how the Lobby client sub systems interact.

2.3.4 Game client

IO-client
The IO-client takes input from the user via the keyboard or mouse and transforms them into game commands.

Game
Game forwards commands coming from the IO-client to the Network. Game also keeps track of which commands that should be executed in their respective rendering frame. Each frame is a set of commands that shall be executed. Game commands that changes the game world will alter the world repository.

Network
Handles game commands coming from the server and appends them to the frame which they belong to.

World
World is a data repository that holds all the objects in the world and the data corresponding to them.

Renderer
The renderer will on request read the world repository and create an image of the current game state based on information from View. The information about the clients current view determines which part of the game world that should be rendered.