Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Kar Méréstechnika és Információs Rendszerek Tanszék

Beágyazott rendszerek szoftvertechnológiája

Házi Feladat

Tervdokumentáció

Torpedó

Játék in Java

Készítette:

Farkas Péter
Fehérvári Áron Botond
Tímár László

Konzulens:

Erdős Csanád

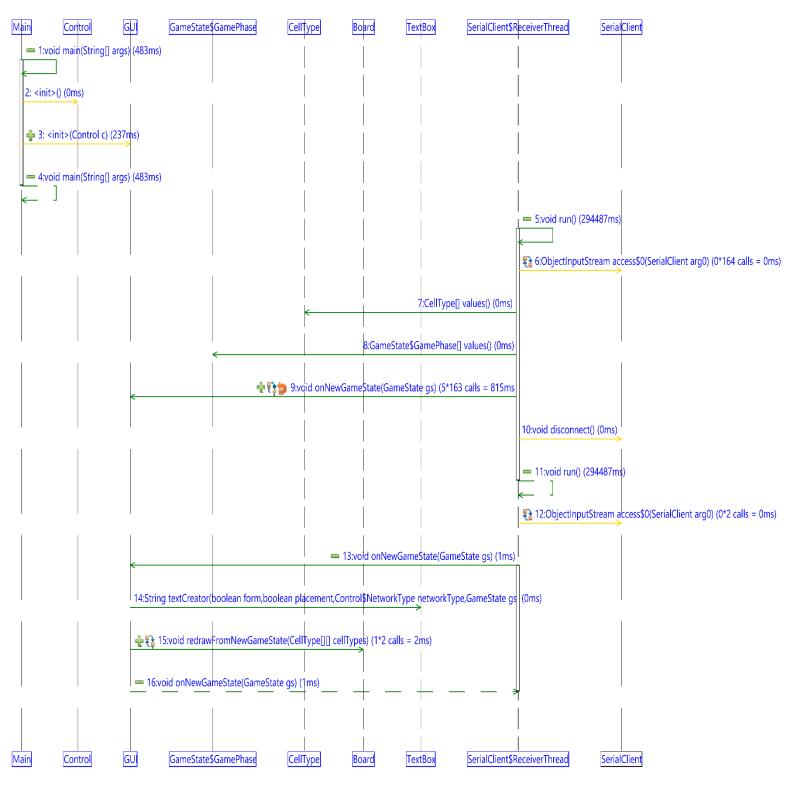
Tartalomjegyzék

UML diagram:	. 3
Szekvenciadiagramok	
GUI	
Control	
Main	
Network	
SerialServer, SerialClient	. 6
GameMessage	
Board	. 7
Ship	. 7
CellType	. 7

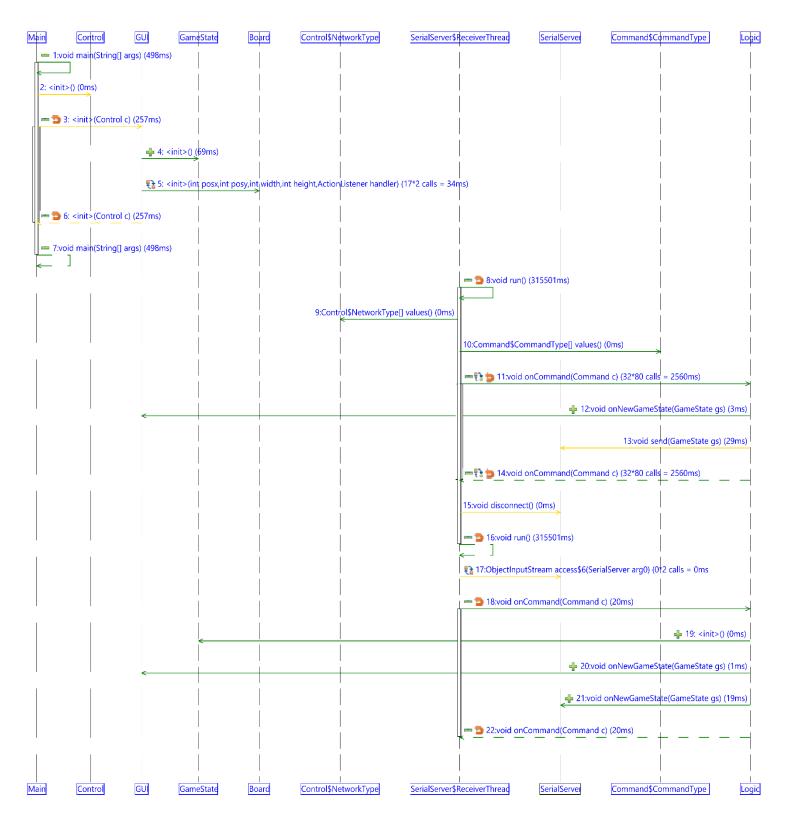
UML diagram: Main GUI Control Network - ctrl: Control gui: GUI - ctrl: Control - running: boolean net: Network + connect(String ip) + disconnect() + send(GameMessage msg) - enemyBoard: Board + sendClick(PositionMessage msg) : void - placement: boolean + sendType(TypeMessage msg) - shipToPlace: int + clickReceived(PositionMessage msg): void - enemyTurn: boolean + sendReady(ReadyMessage msg) + shootPos(Point p) + setGUI() SerialClient Serial Server + receiveType(TypeMessage msg) - serverSocket: ServerSocket + receiveReady() - socket: Socket Board - in: objectInputStream - clientSocket: Socket - enemy: boolean + startClient() out: objectOutputStream - in: ObjectInputStream ships: int out: ObjectOutputStream - buttonGrid: JButton - waterColor: Color - shipColor: Color Ship - shotWaterColor: Color - type: int - shotShipColor: Color - health: int - button[]: JButton + Board() +placeShip(Ship, JButton + Ship(type) + getPosition(JButton) + hit() + getButton(Point) + isAlive() + checkShoot(Point)

+ showShot(JButton) + canPlaceShip(Ship, x, y)

Szekvenciadiagramok



1. ábra Kliens játékos szekvencia diagramja



2. ábra Szerver játékos szekvencia diagramja

GUI

Példányosítja a játékosokhoz tartozó pályákat. Csak a játék aktuális állapotának megfelelő akciókat engedélyezi (például amíg az ellenfél nem lőtt, mi sem lőhetünk újból.

- shootPos()
 - Az ellenfél által lőtt mező kezelése

Control

A játékosok közti kapcsolat logikáját írja le.

- sendClick(PositionMessage msg)
 - o Az ellenfélnek elküldi, hogy milyen pozícióra kattintottunk
- clickReceived(PositionMessage msg)
 - Az ellenfél által kattintott pozícióinformációt tartalmazza. Válaszul visszaküldi a pozíción található mező típusát (sendType)
- sendType(TypeMessage msg)
 - o Az ellenfélnek visszaküldi az általa lőtt mező típusát
- receiveType(TypeMessage msg)
 - Az ellenfél által visszaküldött információ fogadása, arra vonatkozólag, hogy milyen mezőt lőttünk
- sendReady(ReadyMessage msg)
 - o Az ellenfélnek jelezzük, hogy készen állunk a játék megkezdésére
- receiveReady()
 - Az ellenfél jelezte, hogy befejezte a hajók lerakását

Main

A program belépési pontját tartalmazza. Példányosítja és egymáshoz rendeli a GUI és Control objektumokat.

Network

A szerver és kliens közti kommunikációt leíró absztrakt osztály

- connect(String ip)
 - Csatlakozás az ellenfélhez
- disconnect()
 - A kapcsolat lebontása
- send(GameMessage msg)
 - Üzenet küldése az ellenfélnek

SerialServer, SerialClient

A Network absztrakt osztály metódusainak konkrét implementációit tartalmazza. Felelős a kapcsolat felépítéséért, lebontásáért és az üzenetek küldéséért, fogadásáért. Az üzenetek típusa is itt kerül vizsgálatra, majd ennek függvényében kerülnek az üzenetek feldolgozásra.

GameMessage

Absztrakt osztály, a kliens-szerver kommunikáció ezen osztályból leszármazott osztályokat használ fel, mit például kattintás pozícióját tartalmazó PositionMessage, vagy a lőtt mező típusát (víz, vagy hajó) tartalmazó TypeMessage.

Board

A játék mezeit tartalmazó pálya. Egy mátrixban tárolja a mezőket, illetve a mezőkhöz tartozó hajókat.

Ship

A játékos által felhasznált hajó tulajdonságait tartalmazza, mint a hajó még kilövendő mezőinek száma (a hajó "élete"), illetve a hajó teljes hossza.

CellType

A mezők típusát leíró enumerátor. Értekei WATER, illetve SHIP.