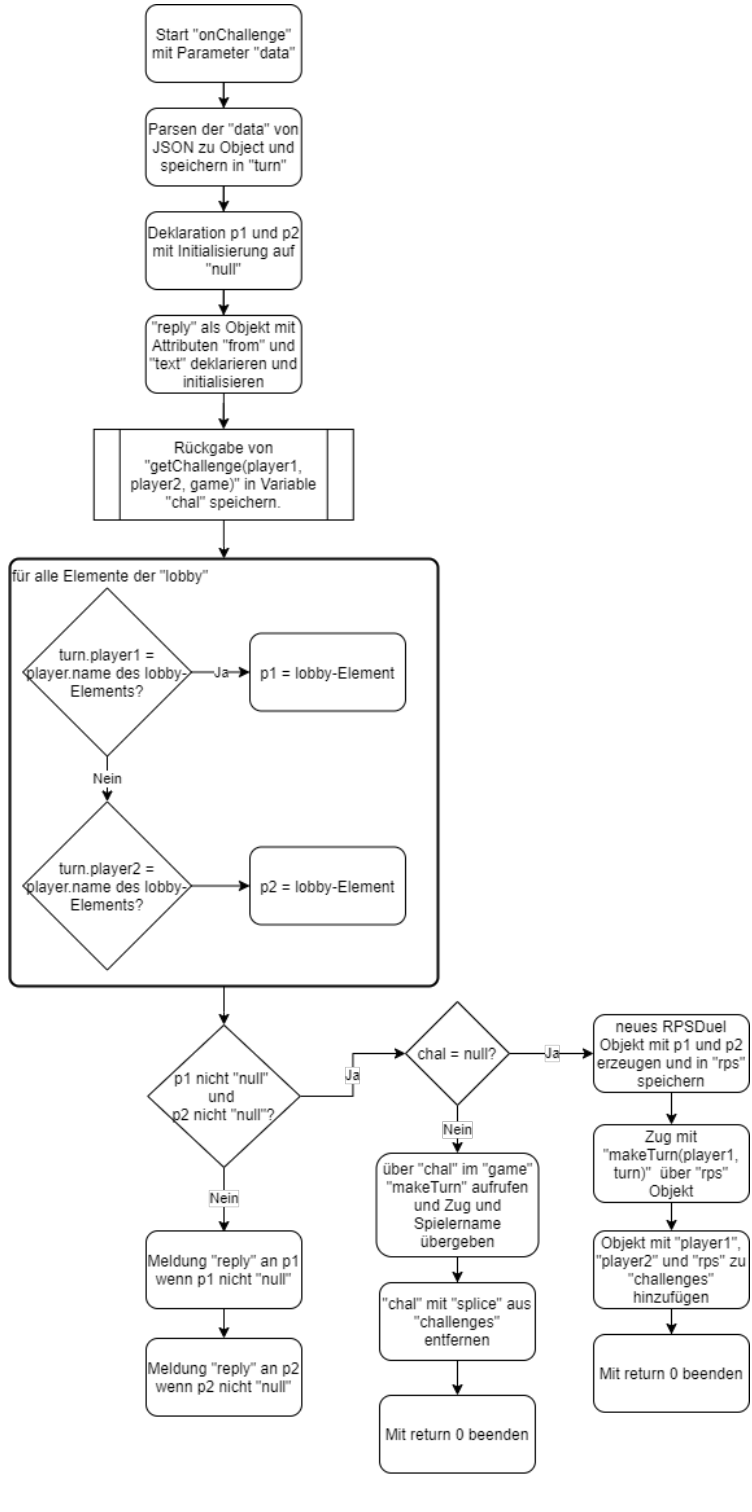


Die Funktion "onChallenge" wird im folgenden Flussdiagramm dargestellt. Daneben findet sich ihr Code. Zunächst implementiert sie nur den Aufbau einer Herausforderung zwischen zwei Spielern für das Stein-Schere-Papier. Es sind aber schon Grundlagen für die Erweiterung mit anderen Spielen geschaffen.



```

function onChallenge(data) {
    let turn = JSON.parse(data);
    let p1, p2 = null;
    let reply = { from: 'Server', text: "" };
    let chal = getChallenge(turn.player1, turn.player2, turn.game);

    for (let i = 0; i < lobby.length; i++) {
        if (turn.player1 === lobby[i].player.name) {
            p1 = lobby[i];
        } else if (turn.player2 === lobby[i].player.name) {
            p2 = lobby[i];
        }
    }

    if (p1 !== null && p2 !== null) {
        if (!chal) {
            let rps = new RPSDuel(p1, p2);
            rps.makeTurn(turn.player1, turn.turn);
            challenges.push({ player1: turn.player1,
                player2: turn.player2, game: rps });
            return 0;
        } else {
            chal.game.makeTurn(turn.player1, turn.turn);
            challenges.splice(challenges.indexOf(chal), 1);
            return 0;
        }
    } else {
        reply.text = "Spieler nicht vorhanden. Herausforderung ("
            + turn.game + " : "
            + turn.player1 + " vs. "
            + turn.player2 + ") gescheitert.";
        challenges.splice(challenges.indexOf(chal), 1);
    }

    if (p1 !== null) {
        p1.socket.emit('msg', JSON.stringify(reply));
    }
    if (p2 !== null) {
        p2.socket.emit('msg', JSON.stringify(reply));
    }
}
    
```

Versuche den Code so gut es geht zu verstehen und zu kommentieren.

getChallenge ( )

