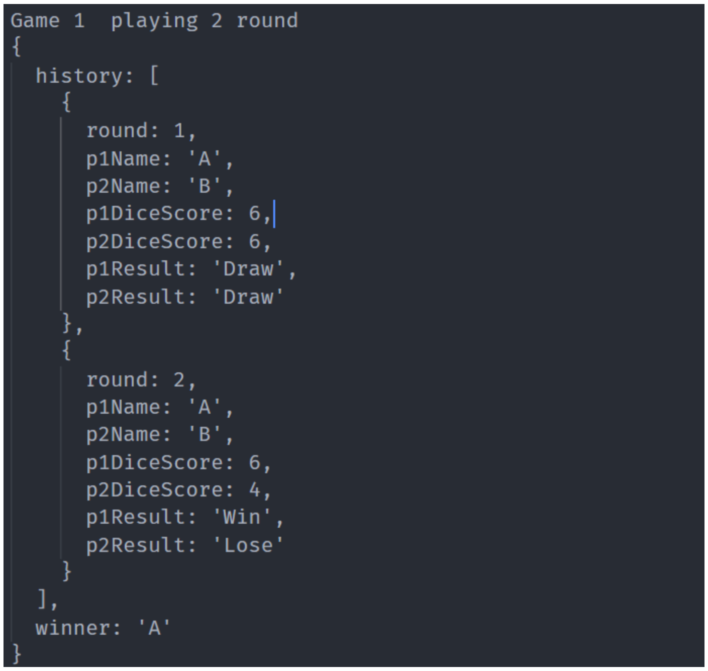
dice.js

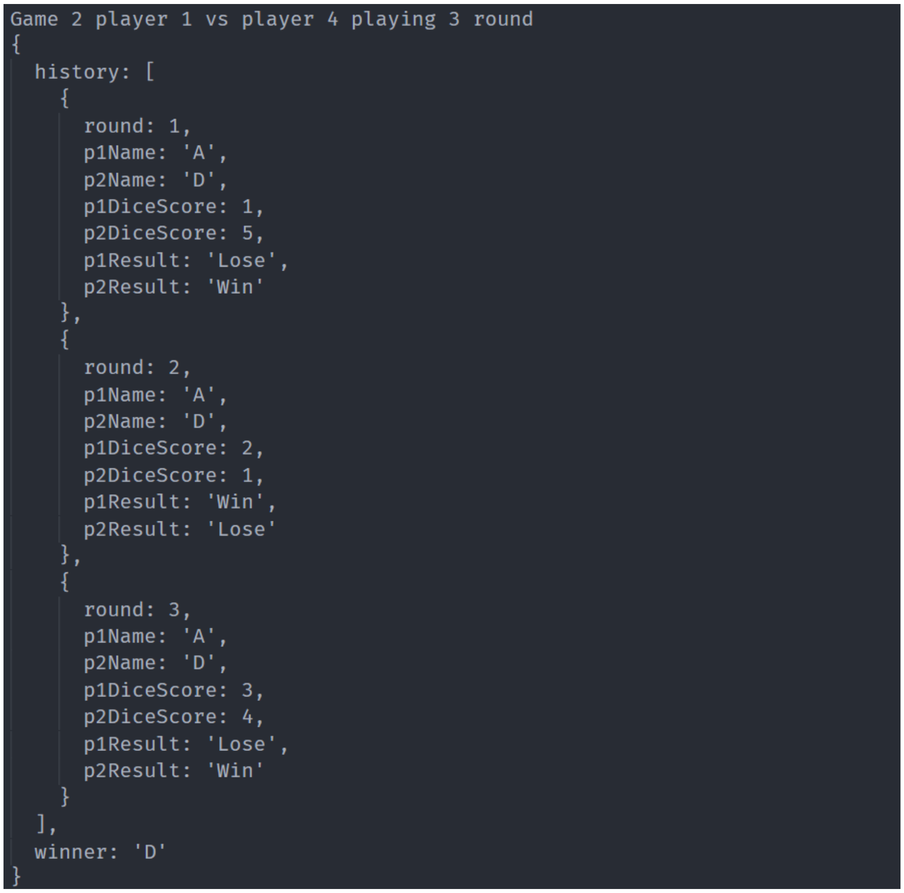
```
// Math.random : random number [0,1) for example 0.44221325113158483
// Math.floor : return integer without float
// this function return number between min and max inclusive
function randomNum(min, max) {
    return Math.floor(Math.random() * (max - min + 1) + min);
}

//create dice object
let dice = {
    // attribute "value" store number 1->6 as array
    value: [1, 2, 3, 4, 5, 6],

    // attribute "function" random() return value from value attribute
    // by random index from 0 to 5
    random() {
        return this.value[randomNum(0, 5)];
    }
}

// create function play
/* @params  p1 = player1 object
            p2 = player2 object
            round = play round     */
function play(player1, player2, round) {

    // create p1 with player1 name and point to 0
    let p1 = { name: player1.name, point: 0 };
    // create p2 with player2 name and point to 0
    let p2 = { name: player2.name, point: 0 };
    // create history array store history object in each round
    let history = [
        // demo data
        // {
        //     round: 1,
        //     p1Name: 'A',
        //     p2Name: 'B',
        //     p1DiceScore: 3,
        //     p2DiceScore: 5,
        //     p1Result: 'Lose',
        //     p2Result: 'Win'
        // }
    ];

    // loop each round
    for (let i = 0; i < round; i++) {
        // create p1DiceScore & p2DiceScore for store value of dice score
        // dice.random() is random dice value from dice object
        let p1DiceScore = dice.random();
        let p2DiceScore = dice.random();

        // create p1Result & p2Result for store result value as string "Win", "Draw",
"Lose"
```

```
        let p1Result, p2Result;
        // if p1DiceScore > p2DiceScore it's mean p1Result = "Win" , p2Result =
"Lose" and add p1 point by 1
        if (p1DiceScore > p2DiceScore) {
            p1Result = "Win";
            p2Result = "Lose"
            p1.point++;
        }
        // if p1DiceScore equal to p2DiceScore it's mean p1Result and p2Result =
"Draw"
        else if (p1DiceScore == p2DiceScore) {
            p1Result = "Draw";
            p2Result = "Draw"
        }
        // if p1DiceScore < p2DiceScore it's mean p1Result = "Lose" , p2Result =
"Win" and add p2 point by 1
        else {
            p1Result = "Lose";
            p2Result = "Win"
            p2.point++;
        }

        //create result object that store value below
        let result = {
            //store round by i + 1
            round: (i + 1),
            //store player 1 name by p1.name
            p1Name: p1.name,
            //store player 2 name by p2.name
            p2Name: p2.name,
            //store player 1 dice score by p1DiceScore
            p1DiceScore: p1DiceScore,
            //store player 2 dice score by p2DiceScore
            p2DiceScore: p2DiceScore,
            //store player 1 result by p1Result
            p1Result: p1Result,
            //store player 2 result by p2Result
            p2Result: p2Result
        }
        // add result object to history array
        // The push() method adds new items to the end of an array.
        history.push(result)
    }



    // create winner variable to store winner player name
    let winner;
    // player 1 point > player 2 point it's mean player 1 is winner
    if (p1.point > p2.point) {
        // set winner by player 1 name
        winner = p1.name;
    }
    // player 1 point == player 2 point it's mean draw
    else if (p1.point == p2.point) {
        // set winner to draw
        winner = "Draw"
    }
    // player 1 point < player 2 point it's mean player 2 is winner
    else {
        // set winner by player 2 name
        winner = p2.name;
    }

    // return object that store history data and winner data
    return { history: history, winner: winner };

}

//create player1 object that store name attribute
let player1 = {
    name: "A",
}

//create player2 object that store name attribute
let player2 = {
    name: "B",
}

//create player3 object that store name attribute
let player3 = {
    name: "C",
}

//create player4 object that store name attribute
let player4 = {
    name: "D",
}


//test
console.log(`Game 1  playing 2 round`)
console.log(play(player1, player2, 2));

console.log(`Game 2 player 1 vs player 4 playing 3 round`)
console.log(play(player1, player4, 3));

console.log(`Game 3 player 3 vs player 4 playing 4 round`)
console.log(play(player3, player4, 4));
```

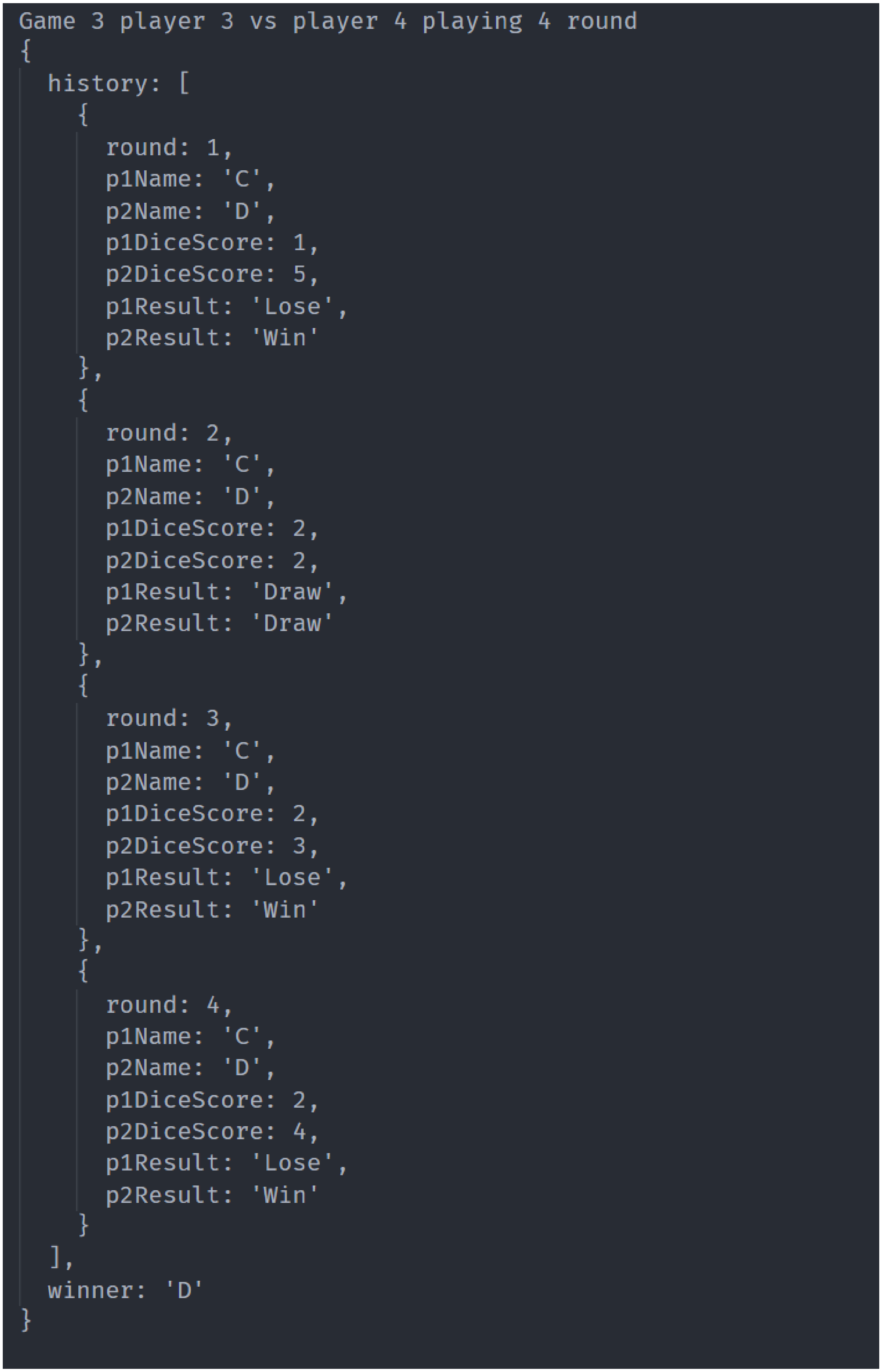
Capture Image



```
Game 1  playing 2 round
{
  history: [
    {
      round: 1,
      p1Name: 'A',
      p2Name: 'B',
      p1DiceScore: 6,
      p2DiceScore: 6,
      p1Result: 'Draw',
      p2Result: 'Draw'
    },
    {
      round: 2,
      p1Name: 'A',
      p2Name: 'B',
      p1DiceScore: 6,
      p2DiceScore: 4,
      p1Result: 'Win',
      p2Result: 'Lose'
    }
  ],
  winner: 'A'
}
```



```
Game 2 player 1 vs player 4 playing 3 round
{
  history: [
    {
      round: 1,
      p1Name: 'A',
      p2Name: 'D',
      p1DiceScore: 1,
      p2DiceScore: 5,
      p1Result: 'Lose',
      p2Result: 'Win'
    },
    {
      round: 2,
      p1Name: 'A',
      p2Name: 'D',
      p1DiceScore: 2,
      p2DiceScore: 1,
      p1Result: 'Win',
      p2Result: 'Lose'
    },
    {
      round: 3,
      p1Name: 'A',
      p2Name: 'D',
      p1DiceScore: 3,
      p2DiceScore: 4,
      p1Result: 'Lose',
      p2Result: 'Win'
    }
  ],
  winner: 'D'
}
```



```
Game 3 player 3 vs player 4 playing 4 round
{
  history: [
    {
      round: 1,
      p1Name: 'C',
      p2Name: 'D',
      p1DiceScore: 1,
      p2DiceScore: 5,
      p1Result: 'Lose',
      p2Result: 'Win'
    },
    {
      round: 2,
      p1Name: 'C',
      p2Name: 'D',
      p1DiceScore: 2,
      p2DiceScore: 2,
      p1Result: 'Draw',
      p2Result: 'Draw'
    },
    {
      round: 3,
      p1Name: 'C',
      p2Name: 'D',
      p1DiceScore: 2,
      p2DiceScore: 3,
      p1Result: 'Lose',
      p2Result: 'Win'
    },
    {
      round: 4,
      p1Name: 'C',
      p2Name: 'D',
      p1DiceScore: 2,
      p2DiceScore: 4,
      p1Result: 'Lose',
      p2Result: 'Win'
    }
  ],
  winner: 'D'
}
```

Git Link

siraom15/INT201_S3_Group-10: INT201 Group 10 Client Side Javascript (github.com)