You have to implement base logic for a simple slot game by using TypeScript.

This game contains (details in excel)

* set of symbols (A2:A7) with non-zero payouts (for 5 elements in a line, C2:C7)
* set of 5 reels (G2:K43)
* 3 lines (N3:R5)
* 5x3 view (E3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | 2 | 2 | 4 | 5 |
| 3 | 1 | 3 | 2 | 2 |
| 1 | 2 | 5 | 5 | 3 |

1. **Your task is to implement base game interface “game.ts”.**

**Sample output:**

*No win*

const result = Game.play({bet: 1});

result === {

stopPositions: [0, 1, 2, 3, 4],

view: [

[5, 1, 5, 2, 1],

[3, 2, 4, 4, 4],

[1, 2, 2, 1, 1]],

rewards: [],

bet: 1,

win: 0

}

*Win*

const result = Game.play({bet: 2});

result === {

stopPositions: [2, 1, 6, 5, 4],

view: [

[1, 1, 1, 1, 1],

[4, 2, 3, 5, 4],

[5, 2, 3, 4, 1]],

rewards: [{

lineId: 1,

symbol: 1,

payout: 10

}],

bet: 2,

win: 10

}

1. **Test your implementation by using BDD (mocha, chai,).**
2. **Run simulation for this model to check user’s win/lose value (1B simulation).**