

Code Mode Documentation - 3/3

[roulette](#), [scarab_spin](#), [wheel](#), [video_poker](#), [rock_paper_scissors](#), [snakes](#), [tome_of_life](#)

system 1 June 10, 2025, 10:00am

Rock Paper Scissors

Variables

- **betSize**
Type: float
Required: true
- **pattern**
Type: array
Required: true
This must be an array, which holds 'rock', 'paper' or 'scissors' values. The length of the array implicitly defines the amount of rounds you want to play. The maximum amount of items is 20.

lastBet example

```
{
  "id": "XXXX-XXXX-XXXX-XXXX-XXXX",
  "iid": "casino:888888888",
  "rollNumber": 4,
  "nonce": "N/A",
  "active": false,
  "game": "rock-paper-scissors",
  "win": false,
  "amount": 0,
  "fiatAmount": "$0.00",
  "payoutMultiplier": 0,
  "b2bMultiplier": 0,
  "payout": 0,
  "fiatPayout": "$0.00",
  "currency": "eth",
  "dateTime": "2022-05-01T00:00:00.000Z",
  "deepLink": "https://stake.com/?betId=XXXX-XXXX-XXXX-XXXX-XXXX&modal=bet",
  {
    "payoutMultiplier": 31.36,
    "playerRounds": [
      "rock",
      "rock",
      "paper",
      "rock",

```

```
"scissors"
],
"rounds": [
  "rock",
  "paper",
  "scissors"
]
```

Code example

```
// Randomly change pattern after every win

// resetStats();
// resetSeed();

function randomPattern(rounds) {
  const pattern = [];

  for (let i = 0; i < rounds; i++) {
    const flip = ['rock', 'paper', 'scissors'][Math.floor(Math.random() * 3)];
    pattern.push(flip);
  }

  return pattern;
}

game = 'rock-paper-scissors';
betSize = 0.00000000; // Bet size is always specified in crypto value, not USD!

const rounds = 5;
pattern = randomPattern(rounds);

engine.onBetPlaced(async (lastBet) => {
  if (lastBet.win) {
    pattern = randomPattern(rounds);
  }
});
```

Roulette

Variables

- **selection**

Type: object

Required: true

Possible indexes: number0, number1, number2, number3, number4, number5, number6, number7, number8, number9, number10, number11, number12, number13, number14, number15, number16, number17, number18, number19, number20, number21, number22, number23, number24, number25, number26, number27, number28, number29, number30, number31, number32, number33,

number34, number35, number36, row1 row2, row3, colorRed, colorBlack, parityEven, parityOdd, range0112, range1324, range2536, range0118, range1936

Functions

- `rouletteNumberColor(number)`

Example: `numberColor = rouletteNumberColor(lastBet.state.result); // returns 'red', 'black' or 'green'`

Returns the color of a roulette number

lastBet example

```
{
  "id": "XXXX-XXXX-XXXX-XXXX-XXXX",
  "iid": "casino:888888888",
  "rollNumber": 10,
  "nonce": 1027,
  "active": false,
  "game": "roulette",
  "win": false,
  "amount": 0,
  "fiatAmount": "$0.00",
  "payoutMultiplier": 0.5,
  "b2bMultiplier": 0,
  "payout": 0,
  "fiatPayout": "$0.00",
  "currency": "eth",
  "dateTime": "2022-05-01T00:00:00.000Z",
  "deepLink": "https://stake.com/?betId=XXXX-XXXX-XXXX-XXXX-XXXX&modal=bet",
  "state": {
    "result": 3,
    "numbers": [
      {"amount": 1e-8, "value": "number0"},
      {"amount": 1e-8, "value": "number1"},
      {"amount": 1e-8, "value": "number2"},
      {"amount": 1e-8, "value": "number3"},
      {"amount": 1e-8, "value": "number4"},
      {"amount": 1e-8, "value": "number5"},
      {"amount": 1e-8, "value": "number6"},
      {"amount": 1e-8, "value": "number7"}
    ]
  }
}
```

Code example

```
// Simple martingale on number color

// resetStats();
// resetSeed();

game = 'roulette';
```

```

baseBet = 0.00000000;
targetColor = 'black';
selection = {'colorBlack': baseBet}

engine.onBetPlaced(async (lastBet) => {
  if (rouletteNumberColor(lastBet.state.result) !== targetColor) {
    selection.colorBlack *= 2;

    return;
  }

  selection.colorBlack = baseBet;
});

engine.onBettingStopped((isManualStop, lastError) => {
  playHitSound();
  log(`Betting stopped!`);
});

```

Scarab Spin

Variables

- **betSize**
Type: float
Required: true
- **lines**
Type: integer
Required: true
Numbers in the range from 1 to 20.

lastBet example

```

{
  "id": "XXXX-XXXX-XXXX-XXXX-XXXX",
  "iid": "casino:8888888888",
  "rollNumber": 11,
  "nonce": 1027,
  "active": false,
  "game": "slots-scarab",
  "win": false,
  "amount": 0,
  "fiatAmount": "$0.00",
  "payoutMultiplier": 0.7777778,
  "b2bMultiplier": 0,
  "payout": 0,
  "fiatPayout": "$0.00",

```

```
"currency": "eth",
"dateTime": "2022-05-01T00:00:00.000Z",
"deepLink": "https://stake.com/?betId=XXXX-XXXX-XXXX-XXXX-XXXX&modal=bet",
"state": {
  "lines": 9,
  "rounds": [
    {
      "offsets": [26, 16, 20, 17, 9],
      "paylines": [
        {
          "payline": "line4",
          "hits": 3,
          "multiplier": 0.5555555555555556,
```

Code example

```
// Hunt for a b2b multiplier sum

// resetStats();
// resetSeed();

game = 'slots-scarab';
betSize = 0.00000000;
initialBetSize = betSize;
lines = 9;
resetOnB2bMultiplierSum = 100;

engine.onBetPlaced(async (lastBet) => {
  if (lastBet.win) {
    if (lastBet.payout / initialBetSize >= resetOnB2bMultiplierSum) {
      betSize = initialBetSize;
    } else {
      betSize = lastBet.payout;
    }
  } else {
    betSize = initialBetSize;
  }
});

engine.onBettingStopped((isManualStop, lastError) => {
  playHitSound();
  log(`Betting stopped!`);
});
```

Snakes

Variables

- **betSize**
Type: float
Required: true
- **difficulty**
Type: string
Required: true
Possible values: easy, medium, hard, expert, master
- **rounds**
Type: integer
Required: true
Number range from 1 to 5.

lastBet example

```
{
  "id": "XXXX-XXXX-XXXX-XXXX-XXXX",
  "iid": "casino:888888888",
  "rollNumber": 6,
  "nonce": 1027,
  "active": false,
  "game": "snakes",
  "win": true,
  "amount": 0,
  "fiatAmount": "$0.00",
  "payoutMultiplier": 1.6333333333333333,
  "b2bMultiplier": 1.6333333333333333,
  "payout": 0,
  "fiatPayout": "$0.00",
  "currency": "eth",
  "dateTime": "2022-05-01T00:00:00.000Z",
  "deepLink": "https://stake.com/?betId=XXXX-XXXX-XXXX-XXXX-XXXX&modal=bet",
  "state": {
    "_rolls": [
      [6, 5],
      [2, 6],
      [5, 6],
      [6, 6],
      [3, 1]
    ],
    "rounds": [
      {
        "roll": [6, 5],
```

Code example

```
// Randomly change difficulty and rows after every bet

// resetStats();
```



```
"rollNumber": 11,
"nonce": 1027,
"active": false,
"game": "slots-tomeoflife",
"win": false,
"amount": 0,
"fiatAmount": "$0.00",
"payoutMultiplier": 0.7777778,
"b2bMultiplier": 0,
"payout": 0,
"fiatPayout": "$0.00",
"currency": "eth",
"dateTime": "2022-05-01T00:00:00.000Z",
"deepLink": "https://stake.com/?betId=XXXX-XXXX-XXXX-XXXX-XXXX&modal=bet",
"state": {
  "lines": 9,
  "rounds": [
    {
      "offsets": [26, 16, 20, 17, 9],
      "paylines": [
        {
          "payline": "line4",
          "hits": 3,
          "multiplier": 0.5555555555555556,
```

Code example

```
// Hunt for a b2b multiplier sum

// resetStats();
// resetSeed();

game = 'slots-tomeoflife';
betSize = 0.00000000;
initialBetSize = betSize;
lines = 9;
resetOnB2bMultiplierSum = 100;

engine.onBetPlaced(async (lastBet) => {
  if (lastBet.win) {
    if (lastBet.payout / initialBetSize >= resetOnB2bMultiplierSum) {
      betSize = initialBetSize;
    } else {
      betSize = lastBet.payout;
    }
  } else {
    betSize = initialBetSize;
  }
});

engine.onBettingStopped((isManualStop, lastError) => {
  playHitSound();
```



```
log(`Betting stopped!`);
});
```

Video Poker

Variables

- `betSize`
Type: float
Required: true

Functions

- `videoPokerGetOptimalHold(hand)`
Example: `videoPokerGetOptimalHold(currentBet.state.initialHand);`
Returns an array with cards which should be held, based on the optimal Video Poker strategy. This would be used as return value of the `onGameRound` function.

lastBet example

```
{
  "id": "XXXX-XXXX-XXXX-XXXX-XXXX",
  "iid": "casino:888888888",
  "rollNumber": 6,
  "nonce": 1027,
  "active": false,
  "game": "video-poker",
  "win": true,
  "amount": 2,
  "fiatAmount": "$0.00",
  "payoutMultiplier": 1,
  "b2bMultiplier": 1,
  "payout": 2,
  "fiatPayout": "$0.00",
  "currency": "trx",
  "dateTime": "2022-05-01T00:00:00.000Z",
  "deepLink": "https://stake.com/?betId=XXXX-XXXX-XXXX-XXXX-XXXX&modal=bet",
  "state": {
    "playerHand": [
      {"suit": "D", "rank": "K"},
      {"suit": "H", "rank": "10"},
      {"suit": "S", "rank": "Q"},
      {"suit": "H", "rank": "K"},
      {"suit": "C", "rank": "2"}
    ],
    "initialHand": [
      {"suit": "D", "rank": "K"},
```

Code example

```
// Increase bet size by 20% on win greater than 1x, reset on new all-time-high p

// resetStats();
// resetSeed();

game = 'video-poker';
balanceDivider = 100000;
betSize = balance / balanceDivider; // Bet size is always specified in crypto va

engine.onBetPlaced(async (lastBet) => {
  if (lastBet.payoutMultiplier > 1) {
    betSize *= 1.2;
  }

  if (profit >= highestProfit) {
    betSize = balance / balanceDivider;
  }

  /**
   * Possible values for lastBet.state.handResult:
   * - none
   * - onePair
   * - twoPair
   * - threeOfAKind
   * - straight
   * - flush
   * - fullHouse

```

Wheel

Variables

- **betSize**
Type: float
Required: true
- **risk**
Type: string
Required: true
Possible values: low, medium, high
- **segments**
Type: integer
Required: true
Possible values: 10, 20, 30, 40, 50

lastBet example

```
{
  "id": "XXXX-XXXX-XXXX-XXXX-XXXX",
  "iid": "casino:888888888",
  "rollNumber": 12,
  "nonce": 1027,
  "game": "wheel",
  "win": false,
  "amount": 0,
  "fiatAmount": "$0.00",
  "payoutMultiplier": 0,
  "b2bMultiplier": 1,
  "payout": 0,
  "fiatPayout": "$0.00",
  "currency": "eth",
  "dateTime": "2022-05-01T00:00:00.000Z",
  "deepLink": "https://stake.com/?betId=XXXX-XXXX-XXXX-XXXX-XXXX&modal=bet",
  "state": {
    "result": 17,
    "segments": 50,
    "risk": "high"
  },
  "clientSeed": "yourClientSeed",
  "serverSeed": "XXX", // Simulation Mode only
  "serverSeedHashed": "XXX" // Live Mode only
}
```

Code example

```
// Hunt 49.5x with pre-rolls

// resetStats();
// resetSeed();

game = 'wheel';
betSize = 0;
baseBetSize = 0.00000001;
risk = 'high';
segments = 50;

engine.onBetPlaced(async (lastBet) => {
  if (lastBet.win) {
    betSize = 0;
  }

  if (!lastBet.win && betSize > 0) {
    betSize *= 1.0207;
  }
})
```

```

    if (currentStreak === -50) {
      log('Loss streak of 50 reached. Setting bet size...');

      betSize = baseBetSize;
    }
  });

  engine.onBettingStopped((isManualStop, lastError) => {

```

Bars

Variables

- **betSize**
Type: float
Required: true
- **difficulty**
Type: string
Required: true
Possible values: easy, medium, hard, expert
- **tiles**
Type: array
Required: true
Possible values: An array of numbers from 0 to 29 with 5 numbers at maximum.

lastBet example

```

{
  "id": "XXXX-XXXX-XXXX-XXXX-XXXX",
  "iid": "casino:888888888",
  "rollNumber": 12,
  "nonce": 1027,
  "game": "bars",
  "win": true,
  "amount": 0,
  "fiatAmount": "$0.00",
  "payoutMultiplier": 2.4,
  "b2bMultiplier": 2.4,
  "payout": 0,
  "fiatPayout": "$0.00",
  "currency": "eth",
  "dateTime": "2022-05-01T00:00:00.000Z",
  "deepLink": "https://stake.com/?betId=XXXX-XXXX-XXXX-XXXX-XXXX&modal=bet",
  "state": {
    "board": [
      1.2,
      0.3,
      0.3,

```

```
0.3,  
0.3,  
0.6,  
0.6,  
1.4,  
0.6,
```

Code example

```
// Add 3 random tiles on win, remove one random number on loss.  
// Keep minimum of 1 and a maximum of 5 numbers.  
  
// resetStats();  
// resetSeed();  
  
function randomTiles(amount) {  
  // Create array with a number range from 0 to 29  
  const numbers = Array.from(Array(30).keys());  
  
  return shuffle(numbers).slice(0, amount);  
}  
  
game = 'bars';  
betSize = 0.00000000; // Bet size is always specified in crypto value, not USD!  
initialBetSize = betSize;  
difficulty = 'medium';  
amount = 1;  
tiles = randomTiles(amount);  
minNumbers = 1;  
maxNumbers = 5;  
  
engine.onBetPlaced(async (lastBet) => {  
  if (lastBet.win) {  
    // Add 3 tiles on win, make sure we have 5 tiles max.  
    amount = Math.min(maxNumbers, amount + 1);  
  } else {
```

[Changelog 16/07/2025](#)