

Description

Randomly made choices are hardly a sign of intelligence. Teach your computer to make educated decisions with the help of basic statistics. Here's how it works:

The primary objective of the AI is to determine which domino is the least favorable and then get rid of it. To reduce your chances of skipping a turn, you must increase the diversity of your pieces. For example, it's unwise to play your only domino that has a 3, unless there's nothing else that can be done. Using this logic, the AI will evaluate each domino in possession, based on the rarity. Dominoes with rare numbers will get lower scores, while dominoes with common numbers will get higher scores.

The AI should use the following algorithm to calculate the score:

1. Count the number of 0's, 1's, 2's, etc., in your hand, and in the snake.
2. Each domino in your hand receives a score equal to the sum of appearances of each of its numbers.

The AI will now attempt to play the domino with the largest score, trying both the left and the right sides of the snake. If the rules prohibit this move, the AI will move down the score list and try another domino. The AI will skip the turn if it runs out of options.

Objectives

Replace the random move generator with the algorithm described above. Let's analyze how the computer will act in two example situations:

1. The first case (see Example 1 below).

```
Computer pieces: [2,5],[3,5],[0,5]
Domino snake: [4,4],[4,2],[2,1],[1,0],[0,0],[0,2]
```

Count:

```
0: 5
1: 2
2: 4
3: 1
4: 3
```

5: 3
6: 0

Scores:

$[2,5]: 4 + 3 = 7$
 $[3,5]: 1 + 3 = 4$
 $[0,5]: 5 + 3 = 8$

Attempts:

Domino $[0,5]$ has the highest score. However, it cannot be played at this moment.

Domino $[2,5]$ has the second-highest score. We can play it by appending it to the right side of the snake.

The result:

Play the $[2,5]$ domino by appending it to the right side of the snake.

2. The second case (see example 2 below).

Computer pieces: $[1,5], [3,5], [0,5]$
Domino snake: $[4,4], [4,2], [2,1], [1,0], [0,0], [0,2]$

Count:

0: 5
1: 3
2: 3
3: 1
4: 3
5: 3
6: 0

Scores:

$[1,5]: 3 + 3 = 6$
 $[3,5]: 1 + 3 = 4$
 $[0,5]: 5 + 3 = 8$

Attempts:

Domino [0,5] has the highest score. However, it cannot be played at this moment.

Domino [1,5] has the second-highest score. However, it cannot be played at this moment.

Domino [3,5] is the last option. Unfortunately, it also cannot be played at this moment.

The result:

Take an extra piece from the stock (if it's not empty) and skip a turn.

Examples

The greater-than symbol followed by a space (>) represents the user input. Note that it's not part of the input.

Example 1

The Computer plays a domino.

```
=====
Stock size: 12
Computer pieces: 3

[4, 4][4, 2][2, 1][1, 0][0, 0][0, 2]

Your pieces:
1:[2, 2]
2:[3, 3]
3:[5, 5]
4:[6, 6]
5:[4, 5]
6:[3, 6]
7:[5, 6]

Status: Computer is about to make a move. Press Enter to continue...
>
=====
Stock size: 12
Computer pieces: 2

[4, 4][4, 2][2, 1]...[0, 0][0, 2][2, 5]

Your pieces:
1:[2, 2]
2:[3, 3]
3:[5, 5]
4:[6, 6]
5:[4, 5]
6:[3, 6]
7:[5, 6]
```

Status: It's your turn to make a move. Enter your command.
>

Example 2

The Computer skips the turn.

```
=====
Stock size: 12
Computer pieces: 3

[4, 4][4, 2][2, 1][1, 0][0, 0][0, 2]

Your pieces:
1:[2, 2]
2:[3, 3]
3:[5, 5]
4:[6, 6]
5:[4, 5]
6:[3, 6]
7:[5, 6]

Status: Computer is about to make a move. Press Enter to continue...
>
=====
Stock size: 11
Computer pieces: 4

[4, 4][4, 2][2, 1][1, 0][0, 0][0, 2]

Your pieces:
1:[2, 2]
2:[3, 3]
3:[5, 5]
4:[6, 6]
5:[4, 5]
6:[3, 6]
7:[5, 6]

Status: It's your turn to make a move. Enter your command.
>
```