Introduction

In this test, you need to do a small part of a simulated slot machine using Lumen (https://lumen.laravel.com/docs/5.5).

To make it as simple as possible you need to create a console command inside the project created using Lumen.

Help

To install Lumen, use the following:

composer create-project --prefer-dist laravel/lumen blog

You can use the artisan CLI in Lumen as the same way as in Laravel but with fewer built-in commands. To see all built-in commands, use the php artisan command in Lumen.

Although there is no make:command command at Lumen, you can create your custom command:

- Add new command class inside the app/Console/Commands folder, you can use the sample class template of the framework serve command
- Register your custom command by adding your created class to the $commands member inside the app/Console/Kernel.php file.

Delivery instructions

Once you are happy with the result, you only need to send back to us the files created. If you want to include everything, please remove the vendor folder before sending the result of the test.

The slot command

1.) Assume that every bet as the value of 1 Euro

2.) Generate a random board with 5 columns and 3 rows using 10 different symbols (9, 10, J, Q, K, A, cat, dog, monkey and bird).

3.) Use the following logic for numbering each symbol on the board:

0	3	6	9	12
1	4	7	10	13
2	5	8	11	14

Example: [J, J, J, Q, K, cat, J, Q, monkey, bird, bird, bird, J, Q, A]

These 15 random symbols generate this:

J	J	J	Q	K
Cat	J	Q	Mon	Bir
Bir	Bir	J	Q	A

4.) A pay out happens when 3 or more consecutive symbols of the same kind are present in a payline, always starting from the first column (0,1,2). For the test consider the following paylines:

- "0 3 6 9 12"
- "1 4 7 10 13"
- "2 5 8 11 14"
- "0 4 8 10 12"
- "2 4 6 10 14"

From the example above 2 paylines are matched "J J J" from "0 3 6 9 12" and "0 4 8 10 12".

5.) Pay out return to the player the following amount:
3 symbols: 20% of the bet.

4 symbols: 200% of the bet.

5 symbols: 1000% of the bet.

6.) Print from the console command the following:

- board: [0,….,14]
- paylines: Array with matched payline and number of matched symbol.
- bet_amount: monetary numbers in cents 1€ = 100cents. In you case is always 100
- total_win: amount won.

Considering the above example the result will be:

```
{
        board: [J, J, J, Q, K, cat, J, Q, monkey, bird, bird, bird, J, Q, A],

        paylines: [{"0 3 6 9 12": 3}, {"0 4 8 10 12":3}],

        bet_amount: 100,

        total_win: 40
 }
```

There is no need for a database or session state, that would end up in a version 2 outside the scope of this test.