

CERTIFICATE No. RNG.BR.LTENG.1002.01.01
RNG
‘759 Gaming’ RNG version V1.0.0

Date of Report:	05 September 2025			
Certifying Laboratory:	BMM Spain Testlabs, s.l.u. Edificio Vinson del Parque Empresarial Vallsolana Camí de Can Camps, 17-19 08174 Sant Cugat del Vallés, Barcelona – Spain			
Recipient of the report:	Secretariat for Gaming and Prizes of the Ministry of Finance			
Jurisdiction:	Brazil Federal			
Technical requirements for testing:	ORDER SPA/MF No. 722, of 2nd MAY 2024			
Supplier of the technological solution:	Name		Address	
	Caribbean Vista Media B.V.		Zuikertuintjeweg Z/N, Curacao	
Contracting’s legal entity:	Same as listed under 'Supplier of the Technological Solution'			
Party responsible for sending the report:	N/A			
Products tested:	Test Object	Component Name	RNG or Server	Game type
	RNG	759 Gaming V1.0.0	RNG	N/A
Certificate reference number:	RNG.BR.LTENG.1002.01.01			
Test Results:	Pass			

The contents of this document are strictly confidential. It has been prepared by BMM Spain Testlabs s.l.u. (BMM) exclusively for the reading of the Caribbean Vista Media B.V. and the regulatory authority of Secretariat for Gaming and Prizes of the Ministry of Finance and may not be disclosed to any other party without the prior written consent of the Caribbean Vista Media B.V.

1. Details of the Product Tested

Name of the Game or Component	Programme Identifier	Server Version	Software Version	Operating System Version	Configuration Version
759 Gaming	759 Gaming	Node v16.15.0	1.0.0	CentOS 7.9	N/A

2. Software

2.1. 759 Gaming RNG

The class imports the functions randomBytes and randomInt from Javascript's 'crypto' library. It then uses them to produce other random functions for multiple purposes. It is written in TypeScript and is intended to be used in a server running CentOS.

Description	File / directory name	Version
Server	\app.slo\sharing.base\helper\RandomHelper.ts	1.0.0
SHA-1	0D110B5852FEC30EF40E0C62FD453CC41DDBB6D8	
MD5	D043A1741449C5A414E7C52E570FABEC	
Server	\app.slo\server.app.slo\games\common\slo\helper\SloRandomHelper.ts	1.0.0
SHA-1	E92C8288D445AA250D7B12470197BBFE4B0251ED	
MD5	6B7BB043A1697B4A022BE4C7B9B368F6	

3. Game Information

N/A. RNG Evaluation.

4. Notes

Following the clarifications provided by the Secretariat for Gaming and Prizes of the Ministry of Finance in their FAQ, question 84, BMM Spain Testlabs s.l.u. has tested that all communication with bettors through the system in the scope, whether textual, auditory or audiovisual are in Brazilian Portuguese.

Below is a list of the requirements within the scope with an N/A result, along with their respective justifications:

Requirement	Comment
Mechanical RNG (Random Number Generator)	Software RNG.
Durability	Software RNG.
Manipulation/Tampering	Software RNG.
Data Collection	Software RNG.

5. Verification Procedures

The following sections describe the implementation of the RNG in the source code.

5.1 Seeding.

The RNG seeds through the operative system's /dev/urandom.

5.2 Cycling

The RNG is cryptographically strong, and no cycling method is required.

5.3 Scaling.

The RNG scaling method doesn't introduce bias.

5.4 Unpredictability.

The RNG is cryptographically strong.

5.5 Statistical Evaluation.

Statistical tests were performed on the output from the RNG. Raw output from the RNG was subjected to a range of tests in the Empirical, Diehard and NIST test suites.

Each test tests the hypothesis that the RNG is a random source of numbers. A “p-value” is produced for each test run, which is the probability that a truly random process would produce the same or a more extreme result. P-values are expected to be uniformly distributed between 0 and 1.

The p-values for each test are evaluated using an Anderson-Darling test. This produces a single p-value, which is the probability that the individual p-values have been produced from a uniform distribution. Finally, the p-values from each test in the same test suite are combined using the Holm-Bonferroni method to provide an overall p-value. This process adjusts each p-value to ensure that the overall probability of accepting the RNG as random matches the confidence interval used. The overall p-value, equal to the minimum of the adjusted p-values, is compared to a specific alpha value to determine if the RNG is accepted or rejected as being random for a specific confidence interval.

Empirical Tests

Test	P-values	95% Confidence	99% Confidence
Frequency Test	1.000000	PASS	PASS
Serial Correlation Test	0.722130	PASS	PASS
Runs Test	1.000000	PASS	PASS
Gap Test	1.000000	PASS	PASS
Coupon Collector Test	0.308054	PASS	PASS
Subsequences Test	1.000000	PASS	PASS
Poker Test	1.000000	PASS	PASS
Overall	0.308054	PASS	PASS

Conclusion: The RNG is **ACCEPTED** as random at the 95% confidence interval.

Conclusion: The RNG is **ACCEPTED** as random at the 99% confidence interval.

The Empirical Tests are based on the tests described by Donald Knuth in The Art of Computer Programming Volume 2: Semi numerical Algorithms (1968, revised in 1997). They test sequences of numbers scaled to specific ranges.

Frequency Test	Counts of each number occurring across the sample set.
Serial Correlation Test	Counts of non-overlapping groups of numbers occurring together. Group sizes of two, three, and four are tested separately.
Runs Test	Counts of ascending and descending sequences of numbers. Note that this is a different test to the Runs Test in the Diehard and NIST Tests.
Gap Test	Counts of the size of gaps between successive occurrences of a given number. Each number in the range is tested separately.
Coupon Collector Test	Counts of sequence lengths required to complete a full set of each number in the range.
Subsequences Test	Similar to the Serial Correlation Test for pairs of numbers, except looking at numbers separated by a specific gap. Step sizes of 5, 10, 15, and 20 are tested separately.
Poker Test	The sequence is split into groups of five. The number of unique values in each group is counted.

Diehard Tests

Test	P-values	95% Confidence	99% Confidence
Binary Rank 32x32 Test	0.834645	PASS	PASS
Binary Rank 6x8 Test	1.000000	PASS	PASS
Birthday Spacings Test	0.401317	PASS	PASS
Bitstream Test	1.000000	PASS	PASS
Count The 1's Stream Test	1.000000	PASS	PASS
Count The 1's Specific Test	1.000000	PASS	PASS
Runs Test	1.000000	PASS	PASS
Squeeze Test	1.000000	PASS	PASS
Overall	0.401317	PASS	PASS

Conclusion: The RNG is **ACCEPTED** as random at the 95% confidence interval.

Conclusion: The RNG is **ACCEPTED** as random at the 99% confidence interval.

The Diehard Tests are based on the test suite published by George Marsaglia in 1995. They test sequences of raw binary output from the RNG.

Binary Rank 32x32 Test	Matrices are created using 32 32-bit words. The ranks of the resulting matrices are counted.
Binary Rank 6x8 Test	Same as the Binary Rank 32x32 Test, except each matrix is formed using 6 values, each taking 8 bits from successive 32-bit words with a specific offset. All possible offsets are tested separately.
Birthday Spacings Test	32-bit words are taken as values, sorted, and the spacings between them calculated. The number of spacings of the same size are counted.
Bitstream Test	Blocks of 2^{18} values are treated as a stream of overlapping 20-bit values. The number of possible 20-bit values that are not found in each block is counted.
Count The 1's Stream Test	8-bit values are taken and assigned a "letter" based on the number of one's appearing in the binary representation of each value. Overlapping groups of 5 "letters" are counted.
Count The 1's Specific Test	Similar to the Count The 1's Stream Test, except 8-bit values are taken from successive 32-bit words with a specific offset. All possible offsets are tested separately.
Runs Test	Counts sequences of increasing and decreasing 32-bit words. Note that this is a different test to the Runs Test in the Empirical and NIST Tests.
Squeeze Test	A value of 2^{31} is repeatedly multiplied by 32-bit words, dividing by 2^{32} and taking the ceiling of the result each time. The number of successive words that are required to reduce the value down to 1 is counted. The value is reset to 2^{31} and the process is repeated.

NIST Tests

Test	P-values	95% Confidence	99% Confidence
Approximate Entropy Test	1,000000	PASS	PASS
Block Frequency Test	1,000000	PASS	PASS
Cumulative Sums Test	1,000000	PASS	PASS
Discrete Fourier Transform Test	1,000000	PASS	PASS
Frequency Test	1,000000	PASS	PASS
Linear Complexity Test	1,000000	PASS	PASS
Longest Run of Ones Test	1,000000	PASS	PASS
Non-Overlapping Template Matchings Test	1,000000	PASS	PASS
Overlapping Template Matchings Test	1,000000	PASS	PASS
Random Excursions Test	1,000000	PASS	PASS
Random Excursions Variant Test	1,000000	PASS	PASS
Rank Test	1,000000	PASS	PASS
Runs Test	1,000000	PASS	PASS
Serial Test	1,000000	PASS	PASS
Universal Test	1,000000	PASS	PASS
Overall	1,000000	PASS	PASS

Conclusion: The RNG is **ACCEPTED** as random at the 95% confidence interval.

Conclusion: The RNG is **ACCEPTED** as random at the 99% confidence interval.

The NIST Tests are based on the suite of tests released by the National Institute of Standards and Technology in Special Publication 800-22, Revision 1a (revised April 2010). They test sequences of raw binary output from the RNG.

Approximate Entropy Test	Similar to the Serial Test, count each possible m-bit value, except it does so for two adjacent m bit lengths and compares the two.
Block Frequency Test	Similar to the Frequency Test, except the data is split into equally sized blocks. The number of ones and zeroes in each block is counted.
Cumulative Sums Test	Random walks are created by converting the data to +1 / -1 for 1 / 0 respectively and summing consecutive values.
Discrete Fourier Transform Test	The data is transformed using a Discrete Fourier Transform. The number of peaks within the 95% threshold are counted.
Frequency Test	The number of ones and zeroes in the binary output is counted.
Linear Complexity Test	The length of the linear complexity of the random sequence is determined.
Longest Run of Ones Test	The data is split into equally sized blocks. The longest run of ones in each block is determined and counted.
Non-Overlapping Template Matchings Test	The data is split into equally sized blocks. Each block is searched for a specific pattern of bits and counted. A separate test is run for various bit patterns. Each bit pattern searched does not overlap with itself. That is, when the pattern is matched, the end of the pattern cannot be the start of another match.
Overlapping Template Matchings Test	Similar to the Non-Overlapping Template Matchings Test, except only one pattern is searched, which may overlap with itself.

Random Excursions Test	As with the Cumulative Sums Test, random walks are created by converting the data to +1 / -1 for 1 / 0 respectively and summing consecutive values. The number of times a given state is visited between returns to zero are counted. Separate tests are run for various states from -4 to +4, not including 0.
Random Excursions Variant Test	Similar to the Random Excursions Test, except the number of times the given state is visited is counted for the entire sequence. Separate tests are run for various states from -9 to +9, not including 0.
Rank Test	Matrices are created using 32 32-bit words. The ranks of the resulting matrices are counted. Note that this is fundamentally the same test as the Binary Rank 32x32 Test in the Diehard Tests, although the implementation may differ.
Runs Test	Runs of consecutive bits of the same value of various lengths are counted.
Serial Test	Counts of each possible m-bit values. Separate tests are run for various m bit lengths.
Universal Test	Distances between repeated patterns of bits are counted.

6. Test Environment

Software details	
Delivery mechanisms	N/A. The RNG is an external module isolated from the games. It has been evaluated using source code, without platforms or other mechanisms.
Client installer package(s)	
Type(s) of delivery mechanisms	
Test details	
Platform(s) and version(s) tested	N/A. The RNG is an external module isolated from the games. It has been evaluated using source code, without platforms or other mechanisms.
Browser(s) and version(s) tested	
Operating system(s) with version	
Mobile device(s)	
Client Operating System	

Sant Cugat del Vallés, Barcelona, Spain, 05 September 2025.

Rubén Baptista
SVP Operations EURSAM

Annex
RNG:

ORDER SPA/MF No. 722, OF 2nd MAY 2024:

Requirement	Result
Requirements of the Random Number Generator (RNG)	Compliant
Source Code Requirements	Compliant
Statistical Analysis	Compliant
RNG Strength for Determining Results	Compliant
Cryptographic Attacks on the RNG	Compliant
Mechanical RNG (Random Number Generator)	N.A.
Durability	N.A.
Manipulation/Tampering	N.A.
Data Collection	N.A.