A Quantum Adaptation to Roll Truly Random Dice in Role Playing Games

# A Quantum Adaptation to Roll Truly Random Dice in Role Playing Games

Antonio Costantino Marceddu, Nicola Dilillo, Marco Russo, Renato Ferrero, Bartolomeo Montrucchio
*Department of Control and Computer Engineering*
*Politecnico di Torino*
Turin, Italy
{antonio.marceddu, nicola.dilillo, marco.russo, renato.ferrero, bartolomeo.montrucchio}@polito.it

***Abstract*—In role-playing games (RPGs), players are called upon to assume the role of a character moving in an imaginary environment and facing several challenges. Their success or failure often depends on randomizers like cards or dice. Regarding the latter, the most commonly used in RPGs are the Platonic solids with the addition of the ten-sided die. They are commonly simulated through classical computers, however, since true randomness is not in their nature, they can only generate pseudorandom numbers. On the contrary, quantum computers exploit the nondeterministic nature of quantum mechanics, so they are perfect candidates for truly random simulations in games of chance. For this reason, this paper proposes and tests various quantum circuits for sampling uniformly distributed discrete values within a fixed range, corresponding to the number of faces of the dice. The simulations reveal the pure randomness of the output of the implemented circuits. They were then used to generate random numbers within a three-dimensional dice-rolling game.**

***Index Terms*—Games, Programming, Quantum computing, Random number generation, Simulation**

## I. Introduction

In a role-playing game (RPG), players create and develop characters with unique abilities, personalities, and backgrounds, and then participate in a shared narrative [1]. Typically players use their character's abilities to overcome obstacles, such as combat or other challenges, occurring during the exploration of a fictional world. The outcome of the game is affected only in part by the choices and actions of the players. Other influencing factors are the directions given by the game master, who acts as a referee and storyteller, and the system of rules that govern how the game is played. Moreover, randomness plays a key role in the game, because it determines the success or failure of the players' actions. In particular, randomness is usually generated by means of dice. Besides the 6-sided die, all convex regular polyhedra, known as *Platonic solids*, are used: tetrahedron (4 faces), cube (6), octahedron (8), dodecahedron (12), and icosahedron (20). Another common die is the pentagonal trapezohedron (10). In this context, also random number generator (RNG) can be a useful support for the players, but at the same time is a critical tool, due to its frequent use and its impact on the game mechanics. Commonly, random values are generated by means of algorithms implemented in computer programs. Some of the most relevant generators are the linear congruential generator, Mersenne Twister, XOR shift, and well equidistributed long-period linear (WELL) [2], [3]. More properly, these algorithms are called pseudorandom number generators (PRNG), because they generate pseudorandom values, i.e., values that have some statistical properties similar to truly random numbers but are computed in a deterministic way. In particular, PRNGs work by starting with a seed value and using a mathematical formula to generate the pseudorandom sequence. Besides predictability, pseudorandom numbers have other shortcomings compared to true random numbers [4], some of which include periodicity, bias towards certain values, and correlation of successive values. In this paper, the weakness of pseudorandom numbers is solved by exploiting quantum computing to generate truly random numbers. Quantum mechanics is based on probabilistic results, making it a perfect randomness source that can be harnessed to generate truly random numbers [5].

Although several quantum random number generators have been recently discussed in the scientific literature [6], [7], to the best of the authors' knowledge, this paper is the first one to describe quantum circuits optimized for the generation of random numbers within the range of the admissible values of RPG dice. They were subsequently utilized for generating random numbers in a three-dimensional dice game.

## II. Methodology

The Internet offers some examples regarding the quantum implementation of RPGs dice [8], [9]. Such approaches rely on the simple use of superposition obtained through *Hadamard* gates applied to all $N$ qubits so that the generated number is between $0$ and $2^N$. Subsequently, techniques such as normalization, executing the circuit multiple times to count the states equal to 1, are used to scale values to the correct range. The problem with these techniques is that such interval transformation creates an imbalance in number assignment, causing some of them to be more likely than others.

The approach proposed in this paper aims to overcome this problem by simulating RPGs dice through circuits that return values falling directly into the desired range. Creating $4$ and $8$ sided dice is straightforward as these numbers are powers of two, so *Hadamard* gates can be used as noted above. Things become a bit complicated for dice with $6$, $10$, $12$, and $20$ sides,

979-8-3503-2277-4/23/$31.00 ©2023 IEEE

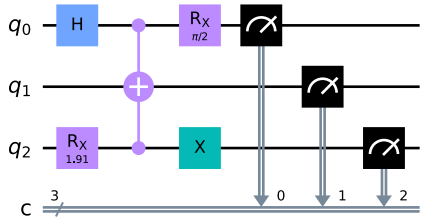


Fig. 1. Quantum circuit generating an equiprobable value between 1 and 6. Starting from the left and going to the right, it is possible to see the Hadamard gate (denoted with a blue square with an H inside), the rotation gate with respect to the X axis (denoted with a purple square with an Rx), the Toffoli gate (denoted with a plus contained inside a purple circle and connected with two other points) the NOT gate (denoted by a green square with an X inside) and finally the measurement gate (denoted by a black square).

since these numbers do not have this property. Therefore, operations must be performed to reduce to 0 the probability that the unwanted states are obtained.

### A. *Quantum circuit for 6-sided die*

Six states can be represented using a minimum of 3 qubits since they allow to represent $2^3 = 8$ different states, a higher number than necessary. If these qubits are put into superposition, 8 equally probable states will be produced, so 2 more than needed. For this reason, to describe the behavior of a *6-sided die* it is necessary that the probability of obtaining them must be equal to 0, while all the others have the same probability. Fig. 1 shows a recently proposed circuit that is capable of representing 6 equally probable states [10] and that can be adapted for this case. It was developed using a Monte Carlo technique with the aim of implementing a quantum random player for the Morra game. The probabilities of obtaining the different states are the following:

$$[0.16666666 \quad 0.16666666 \quad 0.16666666 \quad 0.16666666$$
$$0.16666667 \quad 0.16666667 \quad 0.00000000 \quad 0.00000000]$$

It can be found that starting from the 7th decimal place, the probability begins to diverge slightly. Changing the rotation made by the $RX$ gate from $\frac{\pi}{1.6442678}$ to $\frac{\pi}{1.644267775}$ improves the chances of getting states 000, 001, 010, 011, 100, and 101, which are now equal up to the 7th decimal place:

$$[0.16666667 \quad 0.16666667 \quad 0.16666667 \quad 0.16666667$$
$$0.16666667 \quad 0.16666667 \quad 0.00000000 \quad 0.00000000]$$

Therefore, the circuit having this small modification will be used to roll the *6-sided die*. Fig. 2 shows the results obtained following the execution of the circuit on the *IBM® Aer simulator*. It was executed twice, each for 16384 shots: the first time without noise (Noiseless), while the second time it was executed with *IBMQ Jakarta* noise model, a real quantum computer (Noisy). As it is possible to see, noise allows for states that would otherwise be absent. However, its effect can be reduced thanks to mitigation algorithms (Mitigated), which were added to the previously obtained Noisy results.

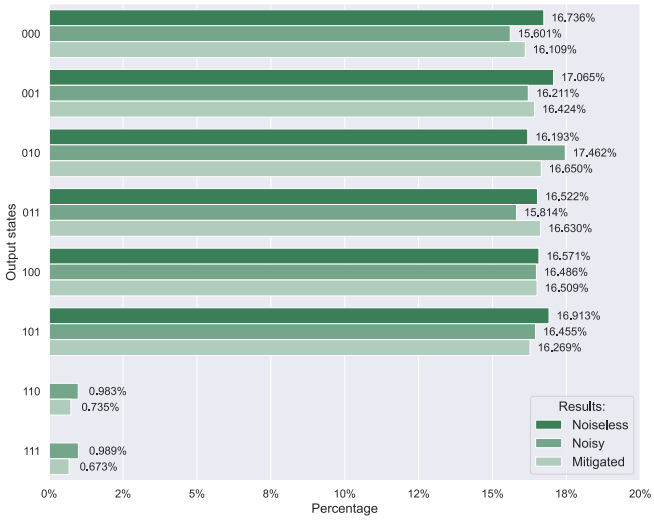


Fig. 2. Comparison of the Noiseless, Noisy, and Mitigated results obtained by running the 6-sided die quantum circuit for 16384 shots on the *IBM® Aer Simulator*.

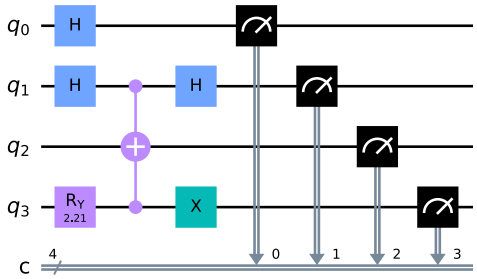


Fig. 3. Quantum circuit generating an equiprobable value between 1 and 10.

### B. *Quantum circuit for 10-sided die*

The 10-sided die can be described using as little as 4 qubits. If put into superposition, they allow representing up to $2^4 = 16$ different states. Since this number is higher than necessary, the probability to obtain 6 of these states must be equal to 0. One of the possible circuits that are capable to represent 10 different equally probable states is depicted in Fig. 3. It was developed starting from the *five-values number generator circuit* presented in [10] and making the following changes:

- the original $RY$ gate rotation was $\frac{\pi}{1.4187762}$. It returns the following probabilities of obtaining the various states:

  $$[2.00000011\text{E}{-1} \quad 2.00000011\text{E}{-1} \quad 2.00000011\text{E}{-1}$$
  $$2.00000011\text{E}{-1} \quad 1.99999957\text{E}{-1} \quad 8.33296990\text{E}{-34}$$
  $$0.00000000 \quad 0.00000000]$$

  Setting it to $\frac{\pi}{1.41877626883}$ instead returns the following probabilities:

  $$[2.00000000\text{E}{-1} \quad 2.00000000\text{E}{-1} \quad 2.00000000\text{E}{-1}$$

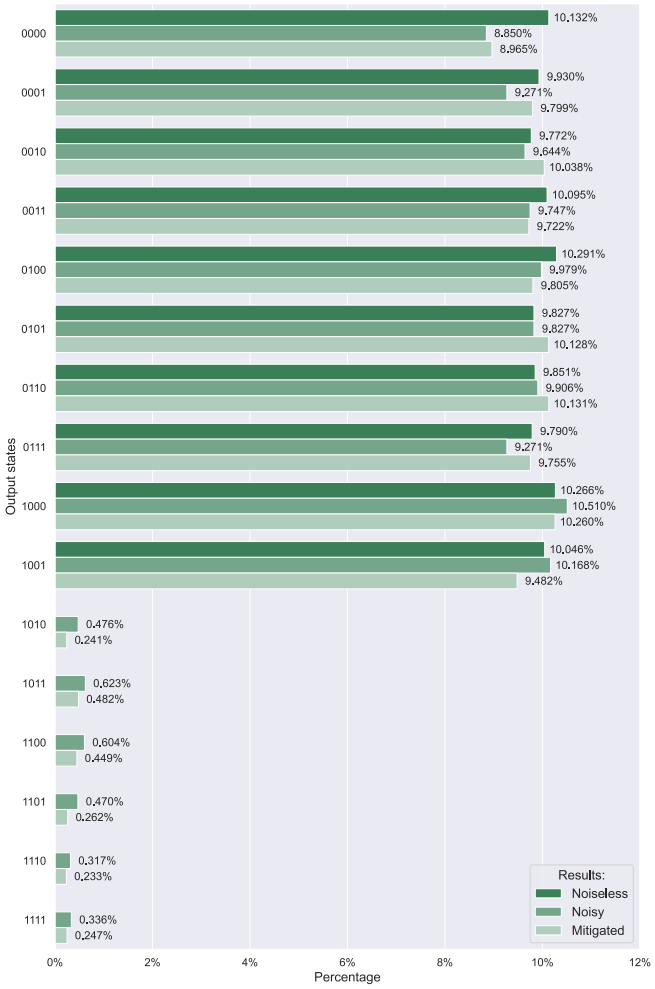


Fig. 4. Comparison of the Noiseless, Noisy, and Mitigated results obtained by running the 10-sided die quantum circuit for 16384 shots on the *IBM® Aer Simulator*.

$$2.00000000\mathrm{E}{-1} \quad 2.00000000\mathrm{E}{-1} \quad 7.53110158\mathrm{E}{-34}$$

$$0.00000000 \qquad 0.00000000]$$

The improvement is evident as the probability to obtain the states 000, 001, 010, 011, and 100 are now equal up to 8 decimal places. However, at present, it has not been possible to completely eliminate the probability of obtaining the 101 state, which is however extremely low;

- another qubit was added and then put in superposition through a *Hadamard* gate;
- the remaining gates have been adjusted to return equally probable states that are close to each other.

These modifications, shown in Fig. 3, have allowed both to improve the results and to obtain a circuit capable of representing the roll of a *10-sided die*. Fig. 4 shows the results obtained executing this circuit on the *IBM® Aer simulator*. As for the *quantum circuit for 6-sided die*, it was executed a first time for 16384 different shots without the addition of noise (Noiseless), and a second time for another 16384 shots

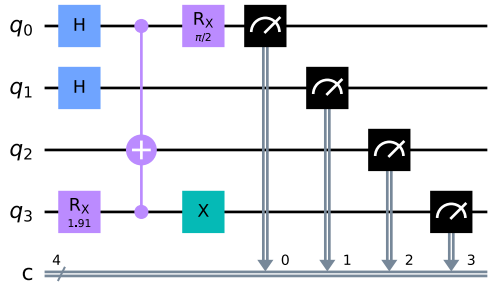


Fig. 5. Quantum circuit generating an equiprobable value between 1 and 12.

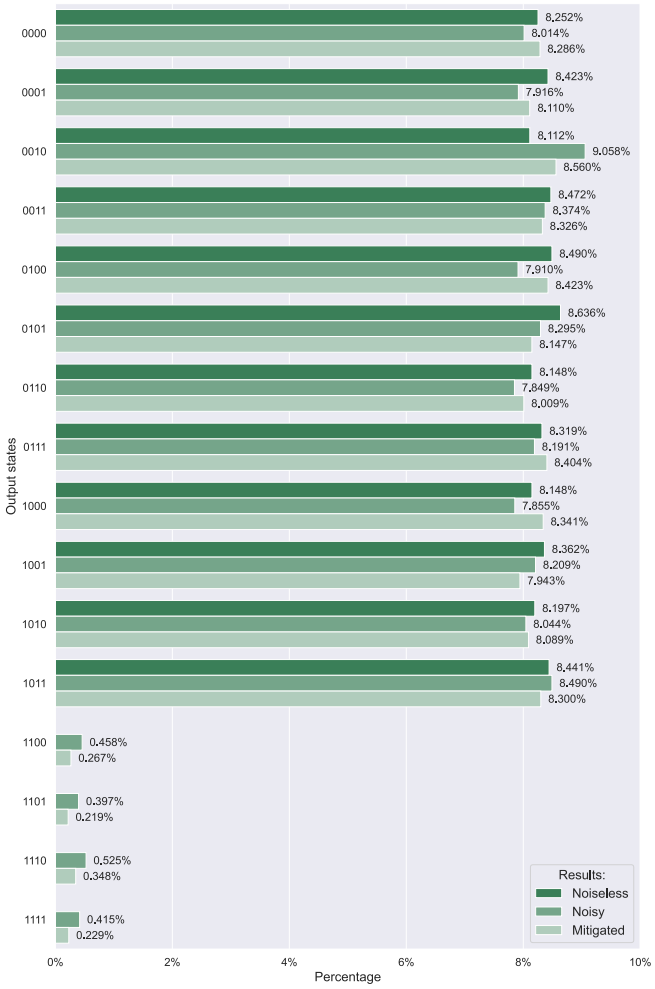


Fig. 6. Comparison of the Noiseless, Noisy, and Mitigated results obtained by running the 12-sided die quantum circuit for 16384 shots on the *IBM® Aer Simulator*.

following the addition of the noise present in *IBMQ Jakarta* (Noisy). Again, noise brings up states that should not normally be present. Mitigation algorithms (Mitigated) can be used to reduce its effect.

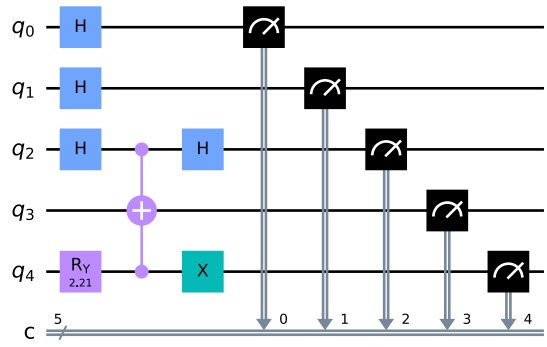


Fig. 7. Quantum circuit generating an equiprobable value between 1 and 20.

### C. Quantum circuit for 12-sided die

As with the previous case, the representation of twelve states also requires the use of a minimum of 4 qubits. If pushed to superposition, 16 equally probable states will be obtained. Since 4 of these states are unwanted, the probability of obtaining them must be set to 0. The same technique used for the *6-sided die* can be used to obtain a circuit capable of doing this. In fact, the number 12 can be obtained by multiplying 6 by 2. It is therefore possible to obtain a circuit able to generate 12 equally probable numbers which can be used to represent the 12-sided die starting from the one used to generate 6, adding another qubit put in superposition through a *Hadamard* gate, and adapting the remaining ports. The final circuit and the results obtained by running it on the *IBM® Aer simulator* are shown in Figs. 5 and 6.

### D. Quantum circuit for 20-sided die

The 20-sided die can be described using a minimum of 5 qubits: if put into superposition, they allow representing $2^5 = 32$ different states. To achieve the desired goal, the probability of getting 12 of these states must be set to 0. It is possible to use the same technique used previously as the number 20 can be obtained by multiplying 5 by two by two times or by multiplying 10 by two. It can therefore be obtained starting from the *quantum circuit for 10-sided die*, adding a further qubit put in superposition and adapting the remaining gates. The resulting circuit is depicted in Fig. 7. For this last case, the results obtained by executing it on the *IBM® Aer simulator* have been omitted due to the large size of the resulting image, induced by the large number of states present.

### E. Interactive 3D Quantum Dice Roller

The described quantum circuits were used to generate random numbers for a dice rolling game, called *Interactive 3D Quantum Dice Roller* [11]. It is based on a modified version of the original Anton Natarov's 3D dice roller, where number generation has been replaced with a quantum implementation [12] [13]. The user can select one or more dice according to how many dice it wants to roll. The same die can be chosen multiple times by clicking on it repeatedly. The *Clear* button removes a previously chosen die. Dice are thrown by clicking on the *Throw* button, after which a loading interface will appear, hiding the operations necessary to execute the circuits on the *IBM®* server. When the results are received, an animation with dice moving shows the random number generated by the quantum device. For practical reasons, the application is set to be executed using the *IBM® QASM simulator*, which reduces the waiting time and avoids errors due to quantum noise.

## III. Conclusions

In this paper, 6 different circuits for the representation of some of the commonly used dice for role-playing games have been discussed. While those with 4 and 8 faces are of immediate realization, the others have required a more careful study. The random number generation efficiency was subsequently tested using the *IBM® Aer* simulator with and without the noise from a real quantum computer. Therefore, an attempt was made to reduce noise at least in terms of the component induced by the measurement phase. Finally, they were used within an application to generate the random numbers of RPGs dice rolled by players. In the future, the authors of this paper plan to continue their research on the adaptation of games to the world of quantum computing.

## References

[1] W. J. White *et al.*, "Tabletop role-playing games," in *Role-Playing Game Studies*. Routledge, 2018, pp. 63–86.
[2] K. Bhattacharjee and S. Das, "A search for good pseudo-random number generators: Survey and empirical studies," *Computer Science Review*, vol. 45, p. 100471, 2022.
[3] P. Kietzmann, T. C. Schmidt, and M. Wählisch, "A guideline on pseudorandom number generation (PRNG) in the IoT," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–38, 2021.
[4] A. M. Frieze, R. Kannan, and J. C. Lagarias, "Linear congruential generators do not produce random sequences," in *25th Annual Symposium on Foundations of Computer Science, 1984.* IEEE, 1984, pp. 480–484.
[5] M. Herrero-Collantes and J. C. Garcia-Escartin, "Quantum random number generators," *Reviews of Modern Physics*, vol. 89, no. 1, p. 015004, 2017.
[6] V. Mannalath, S. Mishra, and A. Pathak, "A comprehensive review of quantum random number generators: Concepts, classification and the origin of randomness," *arXiv preprint arXiv:2203.00261*, 2022.
[7] A. Saini, A. Tsokanos, and R. Kirner, "Quantum randomness in cryptography—a survey of cryptosystems, RNG-based ciphers, and QRNGs," *Information*, vol. 13, no. 8, p. 358, 2022.
[8] Russell Huffman, "Truly Quantum Dice," http://www.jrussellhuffman.com/quantumdice/, [Online]. Accessed on May 12, 2023.
[9] Erika Bulger, "Quantum Dice," https://www.erikabulger.com/quantum-dice, [Online]. Accessed on May 12, 2023.
[10] A. C. Marceddu and B. Montrucchio, "A quantum adaptation for the morra game and some of its variants," *IEEE Transactions on Games*, pp. 1–9, 2023.
[11] Nicola Dilillo, "Interactive 3D Quantum Dice Roller," https://github.com/CARDIGANSPoliTo/quantum-dice-roller, [Online]. Accessed on May 12, 2023.
[12] Anton Natarov, "Online 3D dice roller," http://a.teall.info/dice/, [Online]. Accessed on May 12, 2023.
[13] Matteas Eden, "Interactive 3D dice roller," https://github.com/Matteas-Eden/dice-roller, [Online]. Accessed on May 12, 2023.