Dos and Don'ts of RNGs
iGaming Business Magazine: January / February – 2009

Random Number Generators (RNGs) are arguably the most critical aspect of any gaming system.

There are two basic types of RNGs: hardware and software. Improper design of your RNG, be it software or hardware, could leave you wide open to a number of vulnerabilities and exposures, including predictability in your game outcomes!

These prediction attacks can be costly and embarrassing to Operators and Software Suppliers alike. In order to defend yourself against these attacks, you have to think like an attacker. In doing so, your RNG should be designed defensively.

Software RNGs

At the heart of every software RNG is an algorithm. Software RNGs are also known as *pseudo-RNGs* because these algorithms generate outcomes that only *appear* to be random.

Don't assume that software RNGs are inferior!

Some would argue that the pseudo-random behaviour of software RNGs makes them inferior to hardware RNGs. This is a classic misconception. Just because software RNGs are only pseudo-random, doesn't mean they can't do the job. When implemented correctly, software RNGs can be sufficiently random to thwart even the most expert and informed attacks.

Do choose the right algorithm!

The first matter to consider is the choice of algorithm for the RNG. An algorithm is a complex mathematical equation, and not all algorithms are created equal. Some algorithms are entirely inadequate for high-risk applications like gaming. Other algorithms may be able to support simple games, but will fail miserably when called upon to support the more complex ones.

The algorithm will generate an ongoing string of outcomes for use in your games. As each outcome is generated and sent to a game, that same value is also fed back into the algorithm, which will in turn be used as the 'seed' to produce the next outcome. This feedback process will eventually dictate the 'period' of the RNG, which basically means how long it will take before the sequence starts over again.

The period of stronger algorithms is typically far greater than that of weaker ones. Since more complex games characteristically demand a greater period from the RNG, the types of games you have should directly impact your choice of algorithm.

Don't be careless about your source of seeding!

Since each outcome must be fed back into the algorithm to determine the next one, how do you create the very first outcome? Initializing a software RNG upon start-up is called 'seeding'. In order to get things going, the software RNG must look somewhere to find its initial seed value. This value must be sufficiently random and secure to be effectively hidden from attackers.

If an attacker can learn the initial seed value for your RNG, they may be able to predict the resultant string of outcomes.

<u>Do</u> implement background cycling!

Another critical layer of defence against RNG predictability is called 'background cycling'. Unfortunately, this is also the most common area for costly mistakes in software RNG design. With background cycling implemented properly, a software RNG will generate outcomes, at a highly accelerated and variable rate, whether or not outcomes are actually required by a game at any given point in time.

Effectively, what this means is that the algorithm is in a constant state of motion, thus preventing an attacker from determining what the next outcome will be.

<u>Don't</u> forget about scaling and mapping!

The final major issue to be considered in your design is 'scaling & mapping'. Most software RNGs output extremely large numbers; 32-bit binary values are not uncommon (2^{32} = 4,294,967,296). Naturally, these titanic numbers must be scaled down to more useable values, such as 52 for a deck of cards.

Each scaled number must then be mapped to a symbol used in a game. For example, the number 52 could be mapped to the Ace of Spades.

This is another common area for errors in RNG design. Scaling & mapping often wreak havoc with the quality of output, causing biases to particular outcomes.

Hardware RNGs

Hardware RNGs are altogether very different. Hardware RNGs are comprised of a physical hardware device (usually an electronic card that plugs into a computer) complete with special interface software.

Hardware RNGs are capable of truly random output. Since hardware RNGs do not depend on an algorithm, such factors as period, seeding and background cycling simply do not apply. However, there are other issues that must be addressed when using hardware RNGs.

<u>Do</u> choose the right hardware device!

The first matter to consider is the choice of hardware device. There is a wide selection of devices on the market today.

Much like algorithms, not all hardware devices are created equal. Some may not actually exhibit the advertised randomness, or may have strict requirements as to their compatibility with computer systems. Your best bet is to choose a reputable manufacturer with a tried and tested product.

<u>Don't</u> plug it in and assume that everything will be okay!

The second issue that must be addressed is hardware to software interfacing. Care must be taken to ensure that the hardware device outcomes are not being adversely manipulated by the interface software. Each operating system will have different port drivers, which will in turn interact differently with hardware devices.

<u>Don't</u> forget about scaling and mapping!

The final matter to consider is again scaling & mapping. Just like software RNGs, hardware RNGs also require the numbers to be scaled down to more useable values, and then mapped to symbols used in the game.

Bottom Line

RNGs may not be likely to earn you any profits, but they'll certainly put you in the poorhouse if not designed safely and securely.

Do your research and design them carefully, and don't forget to get them tested!

Bio

Mr. Noah Turner is the Chief Technical Officer (CTO) of Technical Systems Testing (TST), an internationally recognized Accredited Testing Facility (ATF) offering evaluation and consultation services for both the land-based (traditional / terrestrial) and Interactive gaming, lottery and Information Technology (IT) industries.

Office: +1 (604) 873-5833
Email: nturner@tstglobal.com

OFFICES:
Vancouver *– Suite #420, 1367 West Broadway, Vancouver, British Columbia, Canada, V6H 4A7 //* ***O:*** *+1 (604) 873-5833 //* ***F:*** *+1 (604) 873-1075*
London *– Swan Centre, Fishers Lane, Chiswick, London, England, United Kingdom, W4 1RX //* ***O:*** *+44 (0)2087 474 956 //* ***F:*** *+44 (0)2087 427 967*
Sydney *– Suite #305 / 306, 30 – 40 Harcourt Parade, Rosebery, New South Wales, Australia, 2018 //* ***O:*** *+61(2) 9700 7023 //* ***F:*** *+61(2) 9700 7024*
Melbourne *– Level 28, 303 Collins Street, Melbourne, Victoria, Australia, 3000 //* ***O:*** *+61 (3) 9678 9095 //* ***F:*** *+61 (2) 9700 7024*
Macau *– Macau Number 39, 17F Central Plaza, 61 Avenida de Almeida Ribeiro, Macau, China //* ***O:*** *+853 8291 3992 //* ***F:*** *+853 8291 3889*