Pick A Number

We look at how computers pick random numbers and whether they're really random at all

We're not sure when it happened, but at some point, the National Lottery added an option to its Instant Win online games to skip the actual game part and just see the result. Why? Because, as you can probably guess, there is nothing random about these games at all. Shock horror, right?

Well, no. The fact that these things are predetermined shouldn't come as a surprise, really, because if you think about it, even physical scratch cards are exactly the same: a card is either going to be a winner or it's not.

Oddly, this fact hasn't stopped some users suggesting some untoward is going on (**goo.gl/iW9B0**). It's not, of course. In fact, just like with real scratch cards, where the location of winning cards is already essentially set (because once they're in a location, that's where they stay), the randomness comes from us. The National Lottery operator, Camelot, doesn't determine who will go into a shop or onto the lottery website, and it can't decide who will buy a card or a game at any particular time.

Therefore, in that sense it is random, even if the game is predetermined. But even if such games don't need to use true randomness, it raises an interesting question: if an online game wanted to create a random result, how would it actually do it?

Pretty easily, you might think. After all, such games are based on software, and it's an everyday thing for us to request random results from computers. For example, when you put your music player in shuffle mode, it seems to play the tracks in a random order. You don't even think about it.

Additionally, millions of us play games of chance on computers, whether for fun or for money, and we assume that the results are random. However, as we're about to explain, what we perceive as being random might actually be predetermined. And that all comes down to the fact at the most fundamental level, it's impossible for a computer to generate random numbers.

Cannot Compute

Although we've now got to the point where AI is fairly sophisticated and computers can beat the greatest chess players in the world, there are still a few things that humans do better than machines. For a start, we can understand love without our heads exploding (even though it might not feel like it sometimes), and we also possess a greater ability to learn and, of course, to feel.

Mixed up in all this is randomness. With potentially thousands or even millions of external variables acting on us at any one time, how we feel or think can be completely different from one day or even one minute to the next.

A computer, meanwhile, assuming it's functioning correctly, will follow sheer, unmitigated logic. Ask it a question, and you'll either get the same answer every time or one from a set of predetermined answers. Whether the sun is shining, whether it's night or day, it doesn't matter: the software does its job and that's it.

How then, can a computer generate a random number? The short answer is it can't. That might not seem a big deal, but think just how often we rely on computers to do this very thing. Everything they do is based, at the most fundamental level, on two numbers, zero and one, so whenever they're required to be random, whether it's to choose a random digit, letter or even an image, it will involve numbers. Ultimately, then, any kind of apparent randomness is created through the use of numbers. Whether it's dice throws, AI in a videogame or a lottery number picker, everything they do follows some sort of pattern of numbers.

A human, of course, can simply come up with a number. Whatever pops into our heads first can be the one we state when asked. But then again, it might not be, because we might change our minds at the last second and choose something else. That, most of us will agree, is fairly random.

Yet, as we've already stated, humans are affected by external stimuli. Our brains don't exist in isolation like computers, and everything around us affects our decisions (read the 'Is Life Really Random?' boxout for more on this). This use of outside forces is key to the creation of true randomness in computers as well, and we'll be explaining that shortly. First, though, let's look at what computers to do fake it.

Pseudo Randomness

As most of us already know, computers, in spite of their apparent 'intelligence', can only do what they're told to do by humans, using preset patterns. Such rigid logic is naturally not conducive to randomness, so programmers have to find ways to get around this. One particularly common method is to use a pseudo-random number generator (PRNG) which, as the name suggests, picks a number in a way that appears to be random, but without genuinely being so.

There are several different ways to create a PRNG, but what they share is that they use algorithms to produce sequences of numbers that appear to be random. These algorithms might be based simply on precalculated tables, or they might use some form of mathematical formulae, but the result is the same: a list of numbers, which the computer runs through as you make your requests.

Clearly, this is not truly random, but the good news is that PRNGs can be so effective that they appear random to humans and, for many purposes, they are perfectly sufficient. However, because they're based on mathematical logic, it's theoretically possible to use formulas to repeat a particular sequence or to spot patterns. All you need to know is the algorithm and the starting point (the seed number). This, as you can imagine, makes PRNGs less than ideal for applications such as electronic gambling or encryption.

“At the most fundamental level, it's impossible for a computer to generate random numbers”

In the real world, when we shuffle a deck of cards, statistically speaking, a pack of cards will never have been in that particular order before. Indeed, the *QI* website states that the number of possible permutations looks like this: 80,658,175,170,943,878,571,660,636,856,403,766,975,289,505,440,883,277,824,000,000,000,000 (**qi.com/infocloud/playing-cards**).

Is Life Really Random?

If you believe in fate, then everything you do is predetermined. For some, that's a source of inspiration, but personally, we find this idea a bit depressing. Regardless, for the sake of argument, we'll say there's no such thing as fate – at least not in the romantic sense. Does that mean life is random?

On the face it, yes it is, because there are so many conditions and factors that can change at any moment, all interacting with and being influenced by each other. If you roll a die for example, not only is the result determined by the angle and speed at which you roll it, but also the weight of the die and the presence of changes in air pressure (i.e. a breeze). And if it's a hot day, the material of the die and the surface on which it lands might have expanded, further affecting the result. There are probably millions or even billions of variables, which gives us an obviously high degree of randomness.

Now imagine you could reproduce every single possible condition for the next roll and the next and the next after that. Would you get the same number each time? Possibly.

Similarly, a human brain could be seen as a biological computer with numerous external sensors, measuring electrical impulses (touch), changes in air pressure (sound) and radio waves (light), among other things. Without these or any other external sources of randomness, would a human pick the same number each time, if they could relive a particular moment? Could humans, like computers, be deterministic? Are we simply the product of a set of conditions, seemingly infinite as they may be?

These are questions of both philosophy and psychology, and you'll find plenty of views from these disciplines if you Google 'freewill and determinism'. And in more recent times, it's a question that's grown to encompass theories of quantum mechanics. As you can imagine, reading this stuff is a real minder bender, but it's never short of fascinating.

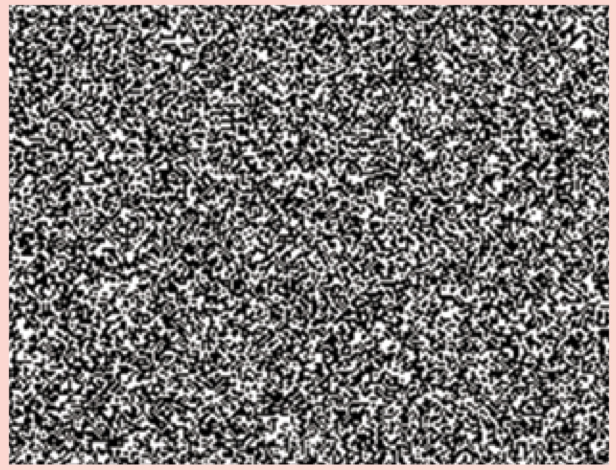
Here are some links to get you started:

- Stanford Encyclopedia of Philosophy: **goo.gl/HkTQPf**
- Simply Psychology: **goo.gl/SlWHL3**
- Fact/Myth: **goo.gl/f7p0ch**

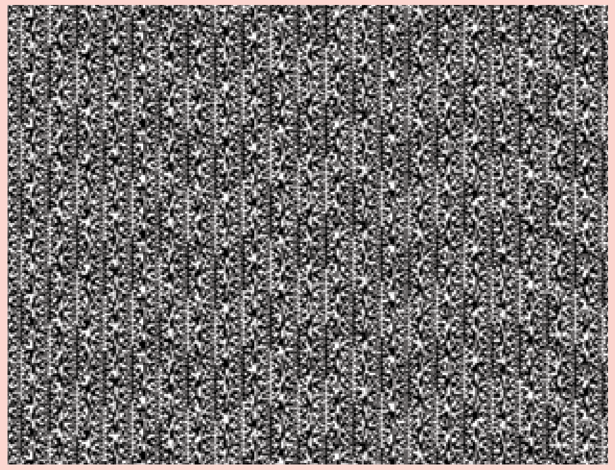
What Does Random Look Like?

As we all know, everything created by a computer is based on numbers, and that includes images. This means, if you want to, you can take randomly generated numbers and turn them into pictures. Using a random bitmap generator, such as the one at Random.org (**goo.gl/uBLUlz**), you can quickly see a visual representation of randomness.

As you can see from this example, it looks like someone spilt a pot of salt on a black table, with no discernible pattern. This is based on data from a TRNG.



Now compare that to this pseudo-random bitmap created by web dev Bo Allen (**boallen.com/random-numbers.html**), using the rand() PHP function with Windows Bitmap.



Quite clearly, a pattern is observable. Presented with just the numbers, a human would probably not be able to spot this, but when viewed in this way, the limitation of PRNGs is beautifully illustrated.

We have no idea how to pronounce this number, but it's probably safe to say that pseudo-random number generators don't even come close to producing this quantity of possible combinations.

In a gambling situation, if you had access to a poker site's algorithm and the seed number, you could theoretically use it to predict the next card, but in reality, you won't be able to. Seed numbers are closely guarded, as are the algorithms, and the number of permutations is likely to be immense, even if it doesn't match up to the reality of a genuine deck of cards shuffled by a human. Also, some sites might continue to generate numbers when games aren't being played, and it's only when a player enters a game that the algorithm is applied and a seed number chosen. So just like a person going into a shop and buying a scratch card, randomness can be provided by the timing of the human interaction.

For all intents and purposes, then, PRNGs are random enough to be used by gambling sites and, indeed, they are. You'll also find them in fruit machines and other software-based gambling devices.

And if you're thinking this is open to abuse from the companies behind gambling sites and machines, then you'd be right. If such firms wanted to rig the results created by their random number generators, they could, but thankfully most countries have strict regulations about how much chance players should have of winning (**goo.gl/hzOQHd**), and they're regularly audited to make sure they're complying with the law.

Introducing Entropy

As good as PRNGs are (and they're efficient too – hence their popularity among coders), they still don't create truly random numbers. In an encryption scenario, that translates into potential weakness. The strength of encryption relies largely on the creation of a strong key, so if it has been produced by a poorly coded PRNG, it could leave you vulnerable.

One solution to this problem is to introduce what's known as entropy. This refers to randomness that is 'collected' from an external source, which is then used to create an encryption key that is truly unique and non-reproducible – just like a human-shuffled deck of cards. In the case of mainstream encryption programs, such as the discontinued TrueCrypt, entropy could be created by the user moving the mouse around wherever they liked, but it was also introduced by the software tracking random changes in the operating system's API.

“ If you had access to a poker site's algorithm and the seed number, you could theoretically use it to predict the next card ”

This still plugs into a pseudo-random number generator, but with a high level of entropy and complicated enough seed data, such encryption should be virtually impossible to crack. Indeed, assuming a PRNG is secure enough, with high enough level of entropy, it can be classed as a cryptographically secure pseudo-random number generator (CSPRNG). Such encryption is still not truly random, meaning it can theoretically be cracked, but the generation of numbers through pseudo-random methods can be enormously secure and, most importantly, it's efficient.

That's good news, because we all use encryption in one form or another every day, with one particularly popular use being in wi-fi protection.

However, we still haven't answered the question at the heart of this article: can a computer ever generate a truly random number? The answer is yes, but only with some help.

Truly Random

To generate a genuinely random number, a computer has to use a true random number generator (TRNG), which takes information garnered from physical phenomena and uses that randomness in its calculations. The movements of a mouse, for example, which can be used to add entropy to a PRNG, is an example of a physical source being used to create a random number.

However, it's only used to create a key, and the encryption program then relies on pseudo-random algorithms to do the rest of its work. If you wanted to create a program that could choose a continuous string of truly random numbers based on something in the real world, then moving a mouse every time wouldn't exactly be convenient, which you'd need to do if you wanted every number to truly random. Also, the level of entropy created by the movement of a mouse's laser over what is a relatively small physical space isn't as high as you might think.

Thankfully, though, there are other ways to add randomness, and they generally involve some kind of extra hardware, which is used to measure a physical phenomenon such as atmospheric noise. Such devices aren't generally employed by regular computer users, but there are hardware RNGs available to consumers, such as the OneRNG (**onerng.info**), a Kickstarter-funded, open-source entropy generator that plugs into a USB port. Gaining $48,551 NZD from 440 backers, the OneRNG uses two on-board sources of randomness: an avalanche diode that generates 'quantum noise' and a detuned RF receiver that “frequency hops at random times to random channels and returns the least significant bit from the (failed) demodulator -this returns significantly closer to 8 bits per byte of entropy data”. This data can then be used to create unique, unpredictable encryption keys.

It's also possible to generate truly random numbers from a Raspberry Pi, using a circuit built into its SoC, which measures thermal noise. You read more about this at **goo.gl/ylzT8**.

If neither of these options appeals or if you want something more established, then you'll have to get in touch with specialist companies. For example, there's the Simtec Entropy Key (**www.entropykey.co.uk**). At the time of writing, however, the website was not functioning, so we're not sure if this is still available. We also found the TrueRNG (**goo.gl/wQa7Q8**) but only available from American websites. Indeed, according to the handy TRNG comparison chart at Wikipedia (**goo.gl/nGmx7b**), the only UK-based seller of this kind of hardware is Simtec. Your options, then, would be somewhat limited if you wanted to create truly random numbers at home.

But there is an answer, and it won't necessarily cost you a penny: an external service that generates truly random numbers for you and provides them on request. One excellent example of this is **www.random.org**. Using atmospheric noise to generate its numbers, Random.org provides a huge list of services, many of which are free, including simple dice rollers, lottery number pickers, password generators and list randomisers. It also offers an iOS and Android app, which offers a free coin flipper, but you can unlock more modes through in-app purchases to get access to a dice roller, a card shuffler, a lottery picker, an integer generator and a list randomiser.

There are plenty of other random number servers too, such as the Hotbits project (**www.fourmilab.ch/hotbits**), which tracks radioactive decay to create random numbers, and EntropyPool (**random.hd.org**). There are also pseudo-random services available, such as **andrew.hedges.name** and **randomnumbergenerator.com**.

Some Final Random Thoughts

If you're not in the market for a hardware RNG, a service like Random.org could be a workable solution. But it's worth considering this from Wikipedia: “Note: random numbers transferred over the public internet are not cryptographically secure for most purposes.” Indeed, even Random.org org says the following:

“We should probably note that while fetching the numbers via secure HTTP would protect them from being observed while in transit, anyone genuinely concerned with security should not trust anyone else (including RANDOM.ORG) to generate their cryptographic keys.”

Clearly, then, having your own hardware to generate numbers is preferable where security is of utmost importance. However, for most of us, regular encryption programs are more than sufficient, particularly if they're bolstered by something that introduces entropy. There's really no need to worry that you don't have a hardware RNG connected your PC.

Anyway, if you think about it, even the data taken from a hardware RNG is the result of billions upon billions of external factors coalescing into one moment. Such thinking underpins chaos theory and the idea of the butterfly effect. Even the seemingly random mess that is real life could be seen as deterministic (although once you get to the quantum level, the whole can of worms that is quantum indeterminacy gets cracked open and we're back at square one – but that's a long, complicated story for another time).

All that matters is that there are far too many variables for a human or even a computer to say what will be next in a string of truly random numbers. To do so would mean being able to predict the future accurately. For proof of our inability to do that, just look at how much difficulty we humans have with creating accurate weather forecasts.

For us, it's ultimately just an intriguing notion to think that without some kind of external, natural source, if you ask a computer to come up with a random number, it can only really respond in one way: “How should I do that?” mm

Disclaimer

We aren't experts in encryption or mathematics, and this article is not intended to provide security advice. It was written purely for entertainment purposes and because we thought it was interesting.