GAME2016
Mathematical Foundation of Game Design and Animation

Lecture 2

Random numbers at a glance

Dr. Paolo Mengoni
pmengoni@hkbu.edu.hk
Senior Lecturer @HKBU Department of Interactive Media

Random Number Generators

Random Number Generators

- Random number generation is a method of producing a sequence of numbers that lack any discernible pattern.

- Random Number Generators (RNGs) have applications in a wide range of different fields such as:
  - **Cryptography**: RNGs ensure that encryption keys are unpredictable and secure.
  - **Computer Simulations**: RNGs help in creating random variables and scenarios, which are essential for accurate and reliable simulation results.
  - **Video Games**: RNGs play a significant role in video games by generating random events, loot drops, enemy behavior, and procedural content. This enhances the gaming experience by making each playthrough unique and engaging.

What is Random?

- What do we mean by Random?
- Uncorrelated
  - with what you're interested in
- Quantum computer systems are the only truly random
  - As far as we know
- Pseudo RNGs are what we can achieve with normal computers
  - It’s random enough for your applications
  - Sometimes we want repeatability from a random number generator (seeding)

Pseudo RNGs

- PRNGs usually generate very long sequences of random numbers.
- E.g., the commonly used Mersenne Twister MT19937 has a sequence length of 2^{19937}-1.
- This is a sequence that will not recur in any practical compute time
  - E.g., the lifetime of the known universe.

Bit Properties

- Ideally a generator will produce numbers that are made up of completely random bits.
- From these numbers we can produce any sort of number we want.
- In practice, algorithms normally generate integers. We must be careful how we transform them. Uniform floating point values don't have random bits.

Generator Algorithms

- Computer systems and programming languages often come equipped with a RNG built in.
  - E.g., rand() or random()

- While these are fast and easy to use, they may:
  - not be very random
  - have short repeat sequences
  - have strong correlation patterns

- But sometimes are good enough

Mersenne Twister

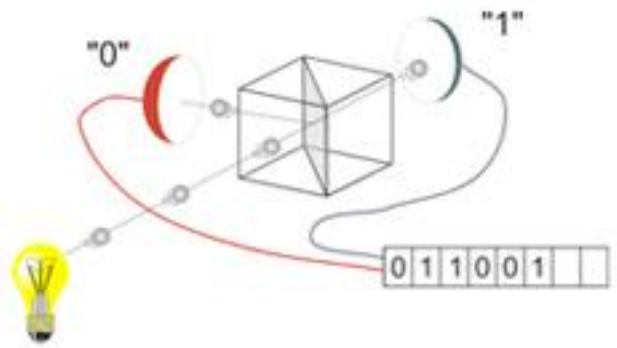
- If you want to be sure the RNG properties are supported, use an implementation of the Mersenne Twister algorithm.
  - Python standard random functions use this algorithm.
- The Mersenne Twister is a widely used PRNG, the name comes from the period length which is chosen to be a Mersenne Prime Number.
- The commonly used MT19937 algorithm stores and generates 624 numbers at a time and can extract them one by one.
- For details, please see:
  - https://www.educative.io/answers/what-is-mersenne-twister

Software Practicalities

- Applications that require high quality random number generators often require a large number of them.
  - Correlation can be a problem if there are many numbers that could be correlated.
  - Need to consider the tradeoff between using high-quality PRNGs and fast PRNGs.
- What type of data primitives does the generator use?
  - 32-bit or 64-bit, integers or floating point?
- How can we generate a random initial state that will still produce high-quality random numbers?
  - seed() function in Python

Truly Random Numbers

- Some modern devices capable of generating a truly random number.
  - The Quantis card is a physical random number generator that exploits a quantum optics process.



- A simulation that uses a truly random number is not repeatable.
  - These quantum random number generators are not seeded and each simulation will be different.
  - Any interesting occurrence or phenomena will not necessarily be reproducible.