Steven Stern
STS.035
Reading Response, Week 3

In the early days of computing, the definition of "random" became interchangeable with the definition of "pseudo-random". Computer scientists of the time agreed that their functions were not random, since they knew an upper bound on when the random numbers generated would begin repeating. In fact, this upper bound was rather small. If a pseudo-random function, which repeated every 10^{10} digits, were to be used in a cryptographic scheme today, the scheme could be easily broken on a personal computer. Perhaps a comparison to today's computers isn't reasonable, since it is possible that in the year 2050, today's cryptographic schemes will seem trivial. However, despite the fact that it served their purposes then, I think their definition of randomness was very weak.

I would accept a computer-generated sequence as random. Computers use many physical events to generate random numbers. For example, some computers use the time between keystrokes, variations in white-noise detected by the computer's microphone, or the speed that the hard drive spins. This can, of course, be improved. If the actual randomness of the computer's randomly generated numbers is of extreme importance, such as when generating the private key for the VeriSign root certificate, more extreme inputs can be used. Pure randomness, from a physics point of view, can be achieved by simply attaching a Geiger counter to a computer.

While reading about the uncertainties people had in the Monte Carlo method, one particular event came to mind. If you ask a person to determine whether a coin is fair, it seems very intuitive to flip the coin a thousand times and record the results. Nobody would object to that method of calculating this result, despite the fact that it is possible (though terribly unlikely) that an unfair coin would appear fair in this test. However, if the computer wishes to perform the equivalent of this test, people object. Instead, people believe the computer should study the density and exact shape of the coin to determine whether it will flip fairly or not. I can't see any difference between this example, and the modeling the computers were doing with nuclear reactions.

The main advantage of a computer simulation, as compared to a laboratory experiment, is that a computer simulation is a controlled environment. A computer can model a nuclear chain reaction that, if it were to happen in reality, would level an entire Pacific island, without causing any real damage at all. I do believe that a simulation is a step away from reality, but by studying reality well enough, it is possible to make this step extremely small. Perhaps this argument is getting a bit too metaphysical, but when a person throws a ball, he can't tell if the motion of the ball is "reality" or an extremely good simulation. Perhaps we are all in a simulation, and when we throw a ball, an extremely fast computer calculates the trajectory of the ball, taking into account the force of the throw, the spin on the ball, and the air resistance around the ball.

I do believe that computer simulation should count as a science. In fact, there are 2 distinct fields of computer simulation that should count as a science: using the simulation, and developing better simulations. So long as the simulation is close enough to reality, I do not see any difference between work done in a simulation and work done in reality. I also believe that work done to make better simulations should be worthy of a PhD. If a mechanical engineer develops a fundamentally superior method for building skyscrapers, that significantly reduce the cost of building one, it is considered very significant. However, that engineer only developed a better way to do what can already be done. Developing a near-perfect computer simulation.