RFC - Random generator and random numbers

Metacosm Development Team

Revision History

Revision M1	July, 15th 2001	
Conversion to Docbook-DTD		
Revision 0.2	March, 30th 2000	Revised by: Ruffy
Conversion to metacosm-DTD		
Revision 0.1	January, 30th 1999	Revised by: Ruffy
First written version		
Revision First discussions	Summer 1999	

The goal of this document is to define the use of random numbers and the way of generating them, in the Metacosm project.

1. License

2. Metacosm

For more information on Metacosm, check our site: Metacosm (http://metacosm.org).

3. Why random numbers?

Yes, that's a good question... Do you really want an answer? OK, I'll try to give you some keys.

If we don't use random numbers, the game will be entirely determinist (i.e. a given situation will always evolve in the same way) *without* players.

> we consider absolute time as a random number (because we don't control it). So we suppose that situations are not a function of absolute time, or only of duration.

But it won't be a game if there aren't players, will it? With the first player, we lose the determinist behavior of the game (we suppose that the player is active, so he/she interacts with the game...). The player acts when he/she wants and does what he/she wants. With time, the player can understand how the game works (and even more with the source :) and uses these informations to cheat. With several players, each player acts when he/she wants and does what he/she wants, but there are as many degrees of liberty as players, so it would be more difficult to have a full understanding and knowledge of the game (i.e. to find the formula which manages the game and describes its future). But, if players decide to cheat together, they'll control the future. That's the first reason of having random numbers: it increases the cheat difficulty.

In the real life, a classical philosophical question is "Is the world completely determinist but chaotic, or do we have some freedom to do what we want when we want?". The answer does not matter for us. The only important point is: the world is unpredictable for us. So if we want to do the same in the game, we can either code extremely complex functions (impossible to reverse to cheat and able to create all or many various situations), either we simply use random numbers and keep human code (i.e. simple but buggy :p). That's the second reason.

The use of random numbers create such a chaos that even we will be surprised by the game evolution. It's even more true for players. That's nice for game-play: various and unpredictable situations. That's the third reason.

4. How?

It seems you like questions :). We use a random generator. You'll say which one, which type, which period, etc. The answer is we don't care. Even if the generator is bad, random numbers are used in many different parts of the game, so it has no effect. And moreover, players are an other source of randomness. These two points show that a basic generator is enough. But a better one will also work :).

We only need a generator able to provide us integers (several sizes) and floats (simple or double precisions). The generator is encapsulated behind an interface, so we can change it if needed. This interface provides basic functions such as to throw dice or dices.

An important point in the generator choice is the ability to use a seed (i.e. a number to initialize it).

5. Why a seed?

For players, we need an unpredictable game. For coders, we need an easy to debug game, so we need to be able to replay a situation, to recreate it. We need to be able to obtain again the same list of random numbers, and the way to do it is to use a seed.

The seed is loaded by the game at the beginning and saved sometimes to allow debugging. With the seed and a good logger, we are able to replay the game and debug it.

the seed and the current state of the random generator should be unavailable to players, and even to supervisors (no temptation), to avoid cheating.