RFC - Générateur aléatoire et nombres aléatoires

Équipe de développement de Metacosm v0.2, 30 mars 2000

L'objectif de ce document est de définir l'utilisation des nombres aléatoires et la façon de les générer, dans le projet Metacosm.

Contents

1 Licence

2 Metacosm/JAMBON

Pour plus d'informations sur Metacosm/JAMBON, consultez le site *Metacosm* `<http://metacosm.org>`.

3 Historique

- v0.2 30/03/1999 (en anglais, Ruffy) - Traduction par Ruffy - Passage en metacosm-DTD
- v0.1 30/01/1999 (en anglais, Ruffy) - Traduction par Ruffy
- Premières discussions sur les nombres aléatoires˜: été 1999

4 Pourquoi des nombres aléatoires˜?

Oui, c'est une bonne question... Vous voulez vraiment une réponse˜? OK, je vais essayer de vous donner des éléments de réponse. Si nous n'utilisons pas de nombres aléatoires, le jeu sera entièrement déterministe (i.e. une situation donnée évoluera toujours de la même façon) *en l'absence* de joueurs.

Nb.˜: nous considérons le temps absolu comme un nombre aléatoire, parce que nous ne le contrôlons pas. Nous supposons donc que les situations ne sont pas fonction du temps absolu. Elles peuvent être fonction de durées.

Mais cela ne serait pas un jeu s'il n'y avait pas de joueurs, n'est-ce pas˜? Avec l'arrivée du premier joueur, nous perdons le comportement déterministe du jeu (nous supposons que le joueur est actif, donc qu'il interagit avec le jeu...). Le joueur agit lorsqu'il le veut et fait ce qu'il veut. Avec le temps, le joueur peut comprendre comment le jeu fonctionne (et cela d'autant plus facilement avec le code source :), et utiliser ces informations pour tricher. Avec plusieurs joueurs, chaque joueur agit lorsqu'il le veut et fait ce qu'il veut, mais il y a autant de degrés de liberté que de joueurs, donc il est plus difficile d'avoir une compréhension et une connaissance parfaite du jeu (i.e. trouver la formule qui régit le jeu et décrit le futur). Mais, si les joueurs décident de tricher ensemble, ils contrôlent le futur. C'est la première raison pour avoir des nombres aléatoires˜: cela augmente la difficulté pour tricher.

Dans la vie réelle, une question philosophique classique est "Le monde est-il complètement déterministe mais chaotique, ou avons-nous une quelconque liberté de faire ce que nous voulons lorsque nous le voulons˜?". La réponse n'a aucune importance pour nous. Le seul point important est˜: le monde est imprévisible pour nous. Donc si nous souhaitons obtenir la même chose dans le jeu, nous pouvons soit coder des fonctions extrêmement complexes (impossibles à inverser (NdT.˜: comprendre ici trouver la fonction réciproque d'une bijection) pour tricher et capables de créer toutes ou une grande variété de situations), soit nous utilisons simplement des nombres aléatoires et nous gardons un code humain (i.e. simple mais bogué :p). C'est la seconde raison.

L'utilisation de nombres aléatoires crée un tel chaos que même nous nous serons surpris par l'évolution du jeu. Ceci est encore plus vrai pour les joueurs. C'est une bonne chose pour le plaisir du jeu˜: des situations variées et imprévisibles. C'est la troisième raison.

5 Comment˜?

Apparemment vous aimez les questions :). Nous utilisons un générateur aléatoire. Vous allez demander lequel, quel type de générateur, avec quelle période, etc. La réponse est sans importance. Même si le générateur est mauvais, les nombres aléatoires sont utilisés dans de nombreuses parties du jeu, donc cela n'a pas d'effet. Et, de plus, les joueurs sont une autre source de hasard. Ces deux points montrent qu'un générateur basique est suffisant. Mais un meilleur conviendrait aussi :).

Nous avons seulement besoin d'un générateur qui nous fournit des entiers (de différentes tailles) et des flottants (avec une précision simple ou double). Le générateur est encapsulé dans une interface, donc nous pouvons le changer si nécessaire. Cette interface fournit des fonctions de base comme le lancer d'un ou de plusieurs dés.

Un point important dans le choix du générateur est la capacité d'utiliser un germe (i.e. un nombre pour l'initialisation).

6 Pourquoi un germe˜?

Pour les joueurs, nous avons besoin d'un jeu imprévisible. Pour les développeurs, nous avons besoin d'un jeu facile à déboguer, donc nous avons besoin d'être capables de reproduire une situation, de la recréer. Nous devons être capables d'obtenir à nouveau la même liste de nombres aléatoires, et la méthode pour obtenir cela est d'utiliser un germe.

Le germe est chargé par le jeu au démarrage et sauvegardé de temps en temps pour permettre le déboguage. Avec le germe et un bon système de logs, nous sommes capables de rejouer le jeu et de le déboguer.

Nb.˜: le germe et l'état courant du générateur aléatoire ne doivent pas être accessibles aux joueurs, et même aux administrateurs (pas de tentation), pour éviter la triche.