

Autonomic

A computer implementation of the game of Nomic

Antoine Amarilli et Pablo Rauzy

École normale supérieure

Le jeu¹ de Nomic

- On a plusieurs joueurs et un ensemble de règles initiales.
- Les règles initiales décrivent comment jouer à un jeu simple et sans intérêt, et (surtout !) comment modifier les règles (souvent à base de vote par majorité).
- Le jeu consiste à modifier les règles et à les exploiter pour gagner d'une façon tordue à laquelle personne n'avait pensé.

Nomic

Players	variable
Setup time	variable, usually low
Playing time	variable, from minutes to years
Random chance	variable
Skill(s) required	imagination, logic, politics, lawyering, variable

All aspects of Nomic are variable. The players can vote to change the rules to whatever sort of game they want to play.

¹Vous avez tous perdu.

- On veut faire une version *informatique* de Nomic (pour éviter les problèmes d'interprétation, et parce que c'est cool).
- L'état courant des règles ne sera plus du texte mais le *code* courant du jeu.
- Les joueurs pourront soumettre des *amendements* permettant de modifier l'état courant pour changer les règles. Ces amendements seront des programmes prenant l'état courant en argument et renvoyant un nouvel état.

Votre mission, si vous l'acceptez :

Construire Autonomic, un programme fonctionnel pur qui soit un état initial raisonnable pour une partie de Nomic sur ordinateur. Par exemple :

- Un joueur quelconque propose un amendement qui est une modification à appliquer aux règles.
- Tous les joueurs votent “oui” ou “non”.
- Si la majorité a voté “oui” alors la modification est appliquée.

(Pas de panique si ça semble flou, des explications plus détaillées suivent.)

Un exemple de partie

<Autonomic> [Current code of the program.]

<Player1> Modify: (lambda (x) x)

<Autonomic> OK, please vote.

<Player1> Vote: yes

<Player2> Vote: yes

<Player3> Vote: no

<Autonomic> Modification accepted.

<Autonomic> [Current code of the program.]

Idées de modifications possibles

Les règles proposées sont très simples. Voici quelques exemples d'amendements que les joueurs pourraient vouloir adopter :

- Faire les propositions d'amendement chacun son tour plutôt que de permettre aux gens de le faire quand ça leur chante.
- Considérer que le proposant vote toujours "oui" à sa propre proposition.
- Exiger une majorité au deux tiers pour modifier certaines parties critiques du programme.
- Gérer les exceptions lors de l'application d'une proposition pour revenir à l'état précédent en cas de problème.

On suppose fixé un ensemble de joueurs identifiés par des pseudos distincts, et une condition de victoire (par exemple, faire afficher un certain message).

Entrée Un *message*, c'est-à-dire un couple pseudo, corps du message.

Sortie Une liste de *messages* à adresser aux joueurs (destinataire et corps) et un nouvel état du programme.

- On traite alors les messages des joueurs dans leur ordre d'arrivée en faisant évoluer le programme (un peu comme un `fold`) et on renvoie les messages de sortie au fur et à mesure.
- On ne s'occupera pas de la partie réseau, ie. il suffit de prendre en entrée les messages avec leur émetteur et de renvoyer les messages avec leur destinataire sur l'entrée et la sortie standard. (Ensuite, on peut facilement utiliser IRC.)

- Il est *fortement* recommandé d'utiliser un langage de programmation avec homoiconicité (*code is data*) tel qu'un Lisp.
- Le programme doit être construit comme une quine pour récupérer son propre code et le renvoyer modifié :
 - soit pour stocker une proposition ou un vote ;
 - soit pour appliquer une proposition acceptée.
- Il n'y a pas d'état partagé entre les différents messages d'entrée à part le code courant. C'est purement fonctionnel, et l'état courant des règles est *exactement* le code courant.

- Pour rendre le jeu effectivement jouable, il serait pertinent de fournir dans l'état initial la définition d'un DSL pour pouvoir facilement écrire des modifications.
- Ce DSL serait également modifiable, évidemment, mais la plupart des amendements l'utiliseraient sans le modifier.

Pourquoi c'est cool ?

- Il y a effectivement des fous qui jouent à Nomic, et à notre connaissance personne n'a réussi à l'implémenter.
- C'est totalement abstrait et incompréhensible et pourtant tellement naturel.
- Si c'est vraiment jouable, ce sera clairement le jeu le plus méta du monde.

