RFC sur les dialogues

Équipe de développement de Metacosm v0.3, 20 mars 2000

Ce document spécifie le système de perceptions.

Contents

1 Licence

```
Ce document est libre, vous pouvez le redistribuer et/ou le
modifier selon les termes de la Licence Publique Générale GNU
publiée par la Free Software Foundation (version 2 ou ultérieure).

Ce document est distribué car potentiellement utile, mais
SANS AUCUNE GARANTIE, ni explicite ni implicite, y compris les
garanties de commercialisation ou d'adaptation dans un but
spécifique. Reportez vous à la Licence Publique Générale GNU pour
plus de détails.
```

Vous pouvez obtenir une copie de la Licence Publique Générale GNU
sur le site de la Free Software Foundation (http://www.fsf.org)
ou en écrivant à
Free Software Foundation, Inc., 59 Temple Place, Suite 330,
Boston, MA 02111-1307, États-Unis.

2 Metacosm/JAMBON

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

3 Historique

- Version 0.3˜: Ajouts dans la section 'descriptions'. Création de la section 'Comment faire... ?'. Passage en Metacosm-DTD
- Version 0.2˜: suppression de la section 'Implantation' remplacée par la section 'StimuliDispatcher'. Rajout d'un paragraphe sur la mémoire sensitive dans la section 'Sens'. Rajouts dans la section 'Réflexion'
- Version 0.1˜: première version reprenant les idées la liste de diffusion utopie et les premières ébauches de documents.

4 Notations

Certains termes sont définis dans le Vocabulary-RFC, disponible sur le site officiel.

5 Les Stimuli

Le système de perceptions est entièrement basé sur l'émission de stimuli par les entités et la réception de ces stimuli par d'autres entités.

Les stimuli sont de deux sortes. On distingue˜:

- Les stimuli descriptifs (ou d'état) qui décrivent l'état de l'entité associée. Ces stimuli ont une durée dans le temps qui peut être infinie.

  Ex.˜:

  ```
  La torche émet de la lumière˜: durée 6 heures Le drapeau est rouge˜: durée infinie
  ```

- Les stimuli d'actions décrivent les actions des entités associées. Leur durée est instantanée.

  Ex.˜:

  ```
  Janselmeer allume la torche. La torche s'éteint. Le dragon émet un grondement
  assourdissant.
  ```

Les stimuli peuvent aussi varier en forme et en intensité. Un stimulus peut ainsi emprunter les média suivants˜:

- Lumière

- Infrarouges
- Ultra-violets
- Son
- Infrason
- Ultra-son
- Mécanique (toucher)
- Saveur (goût)
- Odeur
- Champ magnétique
- Champ électrique
- Vie
- Magie
- Invisible
- Télépathie
- ...

Son intensité est donnée par un nombre réel positif. La valeur 1 correspond à une intensité normale (ou moyenne). En dessous de 1, l'intensité est faible. Au dessus de 1, l'intensité est forte. Un échelle peut être construite en appliquant un logarithme décimal (base 10)˜:

```
<0.001 -> <-3 : intensité exceptionnellement faible
0.001  -> -3 : intensité très faible.
0.01   -> -2 : intensité faible.
0.1    -> -1 : intensité basse.
1      -> 0 : intensité moyenne.
10     -> 1 : intensité élevée.
100    -> 2 : intensité forte.
1000   -> 3 : intensité très forte.
>1000  -> >3 : intensité exceptionnellement forte.
```

6 Les sens

Pour pouvoir recevoir des stimuli, les entités possèdent des sens. Chaque sens est spécialisé dans la réception de stimuli dans un médium spécifique. Par exemple un humain normal possède 5 sens˜:

- La vue -> lumière
- L'ouïe -> son
- Le toucher -> stimulus mécanique
- Le goût -> aliment
- L'odorat -> odeur

Chaque sens possède aussi une puissance intrinsèque. Plus un sens est fort et plus il sera capable de recevoir des stimuli faibles. Comme pour les stimuli, la capacité d'un sens est donnée par des valeurs réelles positives. Une échelle logarithmique identique à celle des stimuli peut être construite˜:

```
<0.001 -> <-3 : puissance exceptionnellement faible
0.001  -> -3 : puissance très faible.
0.01   -> -2 : puissance faible.
0.1    -> -1 : puissance basse.
1      -> 0 : puissance moyenne.
10     -> 1 : puissance élevée.
100    -> 2 : puissance forte.
1000   -> 3 : puissance très forte.
>1000  -> >3 : puissance exceptionnellement forte.
```

Pour savoir si un stimulus a été bien reçu par un sens, il faut multiplier les deux puissances (ou ajouter les 2 valeurs sur l'échelle logarithmique). Pour une valeur de 1, le stimulus est normalement reçu. En dessous de 1, le stimulus est mal reçu. Au-dessus de 1, le stimulus est très bien reçu.

Ex.˜:

```
Stimulus | Sens | réception
----------------------------
   1        1        1       -> reçu
   1       0.1      0.1      -> non reçu
   1       10       10       -> bien reçu
  0.1       1       0.1      -> non reçu
  0.1      0.1      0.01     -> non reçu
  0.1      10       1        -> reçu
  10        1       10       -> bien reçu
  10       0.1      1        -> reçu
  10       10       100      -> très bien reçu
```

Suivant le niveau de réception, la description du stimulus peut varier.

Une fois reçu, le stimulus est conservé temporairement dans une mémoire sensitive. Ainsi l'entité peut consulter les stimuli reçus à tout moment sans devoir pour autant demander leur réémission à la source. Cette mémoire est tout à fait indépendante de la mémoire de l'entité et est géré de manière totalement autonome. Elle fonctionne en fait de la même manière qu'une mémoire cache, en conservant les derniers stimuli reçus ou les plus consultés tandis que les autres seront effacés (ou oubliés) lorsqu'il y aura besoin de libérer de la place. La durée de séjour des stimuli dans cette mémoire sera de toute façon limitée. Il est en effet assez naturel de ne plus ressentir un stimulus lorsqu'il n'évolue pas au bout d'un certain temps. Par exemple, lorsque l'on rencontre une odeur inhabituelle, on la remarque immédiatement et on finit par ne plus y faire attention quelques minutes plus tard..

7 Focalisation

En tant normal, une entité reçoit tous les stimuli de manière égale. Cependant, il lui est possible en se concentrant de se focaliser sur des stimuli particuliers. Sa perception en est alors accrue, tandis que celle des autres stimuli s'en trouve diminuée.

Il est possible de se focaliser˜:

- Soit sur tous les stimuli provenant d'une entité en particulier,

- Soit sur tous les stimuli d'un médium particulier.

La commande 'focus' permet de réaliser cela de la façon suivante˜:

- focus <sens> -> l'entité se focalise sur les stimuli associés à <sens>
- focus <entité> -> l'entité se focalise sur les stimuli provenant de <entité>
- focus -> l'entité reprend une perception normale.

8 Le phénomène d'éblouissement

Le phénomène d'éblouissement apparaît lorsque un stimulus très fort cache des stimuli plus faibles. Le système proposé dans cette section doit permettre de simuler ce phénomène.

Il s'agit de réduire ponctuellement la puissance d'un sens en fonction de la puissance du stimulus le plus puissant reçu. Ce stimulus est forcément d'une durée positive, c'est-à-dire qu'il n'est pas instantané. La règle est la suivante˜:

Puissance effective du sens˜= puissance intrinsèque du sens˜/ intensité du stimulus reçu le plus fort

ou

Puissance effective du sens˜= puissance intrinsèque du sens˜- intensité du stimulus reçu le plus fort en prenant les valeurs sur l'échelle logarithmique.

Dans tous les cas le stimulus reçu le plus intense doit avoir une intensité supérieure à la moyenne (soit 1 ou 0 sur l'échelle logarithmique). Sinon la puissance du sens serait accrue et non pas diminuée.

Il est à noter que ce système privilégie les sens les plus puissants qui sont éblouis moins facilement que les sens plus faibles. Ex.˜:

```
Stimulus     | Stimulus |      |
éblouissant |     reçu | Sens | réception
----------------------------
   10            1        1       0.1 -> non reçu
   10            1       10       1-> reçu
   10            1       100      10-> bien reçu
```

9 La réflexion

Cette section s'appuie sur le phénomène physique du même nom. Voici un exemple simple concernant la lumière˜: imaginons un objet de forme quelconque (un cube par exemple) de couleur blanche (c'est à dire qu'il réfléchit la lumière sur l'ensemble du spectre lumineux). Si on plonge cet objet dans le noir, on ne verra rien˜: il n'émet pas de lumière. Si on allume une lumière rouge, il prend la couleur rouge. Si on allume une lumière bleu, il devient bleu et ainsi de suite. Il s'agit en fait d'une réflexion˜: l'objet renvoie une lumière de la couleur reçue.

Les entités du jeu seront tout à fait capable de réfléchir les stimuli reçus de la même façon. En fait chaque stimulus reçu peut influer sur l'état de certaines entités et modifier leur émission de stimuli. Cependant il faut bien faire attention qu'une entité qui reçoit de la lumière bleue et de la lumière jaune ne renverra pas deux stimuli des deux couleurs mais un seul de couleur verte. En fait chacun des stimuli à modifié l'état de l'entité et c'est à partir de cet état modifié que l'entité émet un nouveau stimulus.

Élargissons maintenant le concept˜: supposons qu'une entité renvoie un stimulus d'un type différent. Prenons l'exemple du téléphone. On peut considérer qu'il s'agit d'une entité qui est capable de recevoir ou d'émettre des stimuli électromagnétiques ou sonores. Cependant, s'il est plongé dans un système isolé il ne fera rien. Il ne fait en fait que transformer des ondes sonores en ondes électromagnétiques et vice-versa. Le téléphone est donc en quelque sorte un miroir sur lequel nous pouvons appliquer le principe de la réflexion. On peut aussi imaginer une entité d'un type spécial constituée de plusieurs parties réparties à différents lieux géographiques. Cela reste hypothétique, mais cela pourrait donner un émetteur/récepteur à distance pouvant recevoir des stimuli dans un lieu et les réémettre dans un autre. Les perspectives données par de telles entités sont très intéressantes, comme des relais de stimuli d'un conteneur vers un autre (voir la section "StimuliDispatcher").

10 Les descriptions

Une entité reçoit toutes les informations sur son environnement grâce aux stimuli. Par exemple, elle pourra obtenir en recevant un stimulus visuel provenant d'une créature, son identité, sa taille, sa race, son âge approximatif, son poids approximatif... Un stimulus sonore pourra aussi la renseigner sur d'autres informations. Ces données brutes suffiront à un bot mais pour un joueur humain, il faut mettre en forme ces données de manière textuelle, graphique, sonore ou autre... Ceci est l'objectif du mécanisme de description dynamique. La suite traite uniquement des descriptions textuelles mais cela s'adapte dans la plupart des cas aux autres types de descriptions.

Voici un exemple d'une entité avec quelques attributs et les descriptions qui pourraient être générées˜:

```
identité : EntityID
race : loup
taille : 2 m
couleur : gris
position : x y z (coordonnées 3D)
```

La description générée au vol est˜: "Un énorme loup gris se trouve à quelques mètres devant vous.". Si le loup était déjà connu sous le nom de "Médor" par le personnage (c'est-à-dire si l'entité EntityID était dans la mémoire du personnage), elle pourrait devenir˜: "Médor se trouve à quelques mètres devant vous.".

Un stimulus transporte donc un certain nombre d'informations qui peuvent ensuite être utilisées par les entités qui les reçoivent. Cependant, on a vu dans les sections précédentes qu'un stimulus pouvait être plus ou moins bien perçu. Ces informations ne seront donc pas toujours accessibles et pourront aussi être déformées. Certaines informations peuvent aussi nécessiter une perception excellente tandis que d'autres seront facilement obtenues.

Voici un autre exemple concernant cette fois-ci une action˜:

```
Entité sujet˜: EntityID (Raoul)
Action˜: ActionID (manger)
Entité objet˜: EntityID (du pain)
...
```

La description sera˜:

```
"Raoul mange du pain"
```

Si le stimulus est mal reçu, cette description pourrait être remplacée par l'une des suivantes˜:

```
"Quelqu'un mange du pain"
```

```
"Raoul fait quelque chose avec du pain"
"Raoul mange quelque chose"
"Quelqu'un fait quelque chose avec quelque chose"
...
```

Ou bien la description pourrait même être tout à fait fausse˜:

```
"Raoul vous menace avec du pain"
"Raoul mange du bois"
```

Le problème qui se pose alors est de savoir quelle partie des descriptions sera altérée. Le problème se pose non seulement pour les joueurs mais aussi pour les bots puisque cela concerne les attributs à la source. Nous associons pour cela un indice de réception minimal à chaque attribut. Le niveau de réception du stimulus doit être supérieur à la valeur de cet indice pour une lecture optimale de l'attribut. Pour mieux comprendre, reprenons l'exemple précédent˜:

```
Entité sujet˜: EntityID (Raoul) : 0.01
Action˜: ActionID (manger) : 7
Entité objet˜: EntityID (du pain) : 1

réception 10 : "Raoul mange du pain"
           1 : "Raoul mange quelque chose"
         0.1 : "Raoul vous menace avec du pain"
        0.01 : "Raoul mange du bois"
```

Cependant, contrairement à l'exemple, il ne sera pas aussi simple en pratique de donner une valeur fausse aux attributs sauf pour le cas où ils sont numériques (taille, longueur, poids...). On pourra alors déterminer aléatoirement la valeur fictive dans un intervalle centré sur la valeur réelle et de diamètre croissant lorsque le niveau de réception diminue.

Changeons de sujet pour la fin de cette section. Dans les exemples précédents, les informations sont réduites et les descriptions sont donc courtes. Cependant, pour des stimuli d'état les descriptions (au moins textuelles) risquent d'être trop longues. Par conséquent, il faudra pouvoir générer une description détaillée (longue) et une description plus courte. La description courte est celle proposée par défaut mais l'utilisateur peut à loisir consulter la description longue. Ceci peut s'appliquer aussi bien à des textes ou d'autres types de descriptions. Par exemple, on peut imaginer un interface graphique dans laquelle les entités apparaissent sous forme d'images (la description courte) et le fait de cliquer dessus fait apparaître une fenêtre comportant des informations supplémentaires sur l'entité.

Ex.˜:

```
description courte:  "Raoul est derrière son comptoir" description longue:  "Raoul est
derrière son comptoir.

C'est un homme corpulent de taille moyenne.  Il porte un marcel et un pantalon blanc
tenu par des bretelles."
```

11 StimuliDispatcher

En français cela signifie "répartisseur de stimuli". Le StimuliDispatcher est en fait le grand intermédiaire entre les entités émettrices de stimuli et celles qui les reçoivent. Il existe un StimuliDispatcher associé à chaque conteneur y compris bien sûr les scènes puisque c'est là que les échanges de stimuli sont les plus nombreux.

Le fonctionnement est le suivant˜: lorsque une entité entre dans un nouveau conteneur, celle-ci est enregistrée auprès du StimuliDispatcher local en tant que nouvelle entité pouvant émettre des stimuli (on considère que toute entité peut potentiellement en émettre même si cela pourrait ne pas être toujours le cas), et pour la réception sur chacun de ses sens. Le StimuliDispatcher demande alors la réémission des stimuli pour toutes les entités enregistrées de sorte que la nouvelle entité puisse avoir un aperçu de son environnement. Par la suite, lorsque un stimulus est émis par n'importe quelle entité, le StimuliDispatcher l'achemine vers tous les sens enregistrés qui peuvent le recevoir. Le rôle du StimuliDispatcher est donc seulement limité à la livraison des stimuli; en aucun cas il ne doit les modifier.

12 Comment Faire... ?

Cette section a pour but de montrer comment le système de perceptions défini dans les sections précédentes peut s'appliquer dans des cas inhabituels. Il s'agit de cas de figure qui ne peuvent pas apparaître dans le monde réel et qui seront notamment créés avec des effets magiques.

- **Comment réaliser un stimulus ambiant ?** Il suffit de définir le lieu comme la source du stimulus.
- **Comment réaliser un stimulus délocalisé ?** Par exemple, on peut vouloir émettre un stimulus dans un ville A alors que l'on est dans une ville B. Il suffit pour cela que l'entité émettrice s'enregistre auprès du StimuliDispatcher associé à B, émette son stimulus, et se désenregistre.
- **Comment faire de la ventriloquie ?** Il suffit d'émettre un stimulus sonore en spécifiant que la source est l'entité visée au lieu de celle qui crée le stimulus.
- **Comment faire un stimulus mondial ?** Il suffit de relayer un stimulus diffus sur tous les StimuliDispatcher.
- **Comment modifier par une illusion l'apparence d'une entité ?** Par exemple, Il peut s'agir de donner l'apparence d'un lingot d'or à un lingot de plomb ou donner l'apparence d'une souris à un éléphant. Pour cela, l'entité visée doit émettre un stimulus qui ne correspond pas à son état interne réel (le lingot de plomb a toujours le même poids et non pas celui d'un lingot d'or). Pour cela, on peut imaginer qu'un sort puisse en faire en sorte que l'entité visée modifie les stimuli avant de les envoyer, ou bien intercepter les stimuli émis et les modifier ensuite (voire les remplacer par d'autres).
- **Comment créer l'illusion da la présence d'une entité ?** C'est plus compliqué que le cas précédent puisqu'il faudra créer spécialement une entité possédant l'apparence souhaitée (mais uniquement l'apparence, cette entité pouvant n'avoir aucune existence physique).

13 Résumé

Un Stimulus, c'est:

- Une entité source
- Une intensité intrinsèque
- Un médium
- Une action le cas échéant
- Des attributs pour décrire la source ou l'action.

Un sens, c'est:

- Un médium
- Une sensibilité (puissance)
- Une mémoire des derniers stimuli reçus.

Un StimuliDispatcher, c'est:

- Un ensemble d'entités sources.
- Un ensemble de sens récepteurs par médium différent.

Nouvelles commandes:

- focus˜: permet de se concentrer sur des stimuli particuliers