RFC sur les dialogues

Équipe de développement de Metacosm v1.5 17 avril 2000

Ce document spécifie les dialogues.

Contents

1 Licence

2 Metacosm/JAMBON

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

3 Historique

- Version 1.5˜: correction mineure dans la section 'Modificateurs d'expression'
- Version 1.4˜: amélioration des sections 'Bots actifs' et 'Le dialogue', ajout des modificateurs d'expression
- Version 1.3˜: conversion en dtd metacosm
- Version 1.2˜: ajout des codes de retour et extension des dialogues à plusieurs créatures
- Version 1.1˜: mises à jour suite aux discussions
- Version 1.0˜: premières ébauches

4 Notations

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

Concernant la casse des dialogues, les mots-clés peuvent être entrés en minuscules, en majuscules voire en panaché sans que cela pose un problème.

5 Mode d'interrogation des bots

5.1 Bases

Il est question ici de spécifier le mode d'interrogation des bots. Il a été choisi un mode d'interrogation par mots-clés. Chaque bot est associé à un ensemble de couples (mot-clé, réponse).

Ex.˜: Pour un bot qui s'appelle Raoul et qui est boulanger.

```
Name -> My name is Raoul.
Job  -> I am the baker.
```

Chaque bot devra ainsi répondre aux mots-clés suivants˜:

- Name
- Job
- les noms de créatures que connaît le bot
- les noms des lieux que connaît le bot

Si les réponses aux deux premiers mots-clés sont statiques, elles sont dynamiques pour les autres et dépendent du bot. Il faut néanmoins nuancer la staticité des deux premiers˜: une créature peut mentir ou changer de métier.

Ex.˜: Raoul est un boulanger qui vit au nord de la ville de Midgaard.

```
>Raoul
Bot: Raoul the baker is a vile enemy of mine.
      You can find it in the vicinity of North Midgaard.
>Marcel
Bot: I don't know that.
```

Les mots-clés doivent aussi permettre d'élargir la conversation˜:

```
>Job
Raoul: I am the baker.
>baker
Raoul: I cook breads, pies and cakes.
>cakes
Raoul: My cakes are the best of Midgaard.
```

Mais ils peuvent aussi servir de fil rouge dans une quête˜:

```
>treasure
Old Pirate: I had a map but I sold it to an adventurer named Raoul.
(...)
>treasure
Raoul: I have only 1 half of the map. Robert has the other part.
(...)
>treasure
Robert: I have the other part of the map.
```

5.2 Bots actifs

Jusqu'à présent, nous avons considéré les bots comme étant complètement passifs, répondant uniquement aux interrogations des joueurs. Mais il n'y aucune raison, et même au contraire, que les bots ne puissent pas être plus actifs dans la conversation. Ils peuvent proposer des sujets de discussion (parler de la pluie et du beau temps, de l'actualité locale), entamer une discussion avec un passant (joueur ou non), notamment si le bot à quelque chose à vendre.

Ex.˜: Un garde aborde Janselmeer. Il ne le connaît pas, donc il l'interroge pour savoir qui il est.

```
         A guard is approaching.
         Guard: What's your name, stranger?
case 1: >Janselmeer
```

```
        Guard: I am watching you, Janselmeer.
case 2: >Oumph
        Guard: I am watching you, Oumph.
```

Janselmeer peut dire son véritable nom (1) ou mentir (2). Les deux réponses seront acceptées par le garde. Janselmeer sera alors connu sous le nom qu'il a donné à ce garde (Janselmeer ou Oumph), et aussi par les autres personnes en ville de manière potentielle.

Maintenant, reprenons à nouveau cet exemple mais cette fois-ci le garde connaît déjà Janselmeer.

```
        A guard is approaching.
        Guard: What's your name, stranger?
case 1: >Janselmeer
        Guard: I am watching you, Janselmeer.
case 2: >Oumph
        Guard: You lie! I know that you are not Oumph.
        A guard attacks you.
```

Ici, le garde attendait 'Janselmeer' comme réponse. Toute autre réponse , 'Oumph' par exemple, peut être considérée comme un mensonge par le garde.

Cet exemple permet d'effleurer deux nouveaux aspects. Le premier, c'est l'ajout de nouveaux mots-clés (ici 'Janselmeer' a été rajouté). Le deuxième, plus intéressant, est l'intégration des réponses du joueur à la mémoire du bot. Sur l'exemple, le garde va associer le nom du joueur Janselmeer à son apparence. Bien sûr, rien n'empêche de mentir mais l'exemple montre de façon simplifiée les risques encourus.

Il peut arriver que l'on ne désire pas répondre aux questions. Dans ce cas, une créature peut utiliser le mot-clé JOKER pour le signifier. Une absence de réponse (au bout d'un temps limité) pourra aussi être considérée comme un refus de communiquer. Les joueurs préféreront peut être la deuxième solution tandis que les bots utiliseront la première. Si les joueurs doivent utiliser JOKER, il serait mieux de lui donner un alias comme '...' pour l'interface utilisateur, ce qui est plus explicite.

5.3 Communication entre bots

Appliquons ce que l'on a défini précédemment à la communication entre 2 bots. Reprenons l'exemple du garde.

Vu par le joueur, cela devrait donner ça˜:

```
Guard to Bot: What's your name, stranger?
Bot to Guard: My name is Bot.
Guard to Bot: I am watching you Bot.
```

Vu par un bot˜:

```
Guard to Bot: Name
Bot to guard: Bot
Guard to Bot: ""
```

La principale difficulté pour réaliser ceci est que le garde ne sait pas qu'il parle à un bot. Celui à qui il parle doit pouvoir comprendre qu'il soit bot ou joueur. Pour être certain de se faire comprendre, le garde doit donc s'adresser à son interlocuteur de manière compréhensible par les joueurs et les non-joueurs. De plus, s'il y a d'autres créatures qui écoutent la conversation, elles pourront aussi comprendre qu'il s'agisse de joueurs ou non. La solution que suggère l'exemple est d'intégrer dans un message à la fois une version courte interprétable par les bots (correspondant aux mots-clés), et une version longue du message pour les joueurs.

Exemple du garde, vu par le jeu˜:

```
Guard to Bot: ("Name", "What's your name, stranger?")
Bot to Guard: ("Bot", "My name is Bot.")
Guard to Bot: ("", "I am watching you Bot.")
```

Lorsqu'un bot ou un joueur envoie un message qui ne peut être compris ou qui ne nécessite aucune réponse, la version courte sera vide. Aucun traitement ne sera alors fait pour interpréter le message. Par contre il pourra être répété.

Ex.˜:

```
Guard: Now you die, Raoul !
Message ("", "Now you die, Raoul !")
```

Ex.˜:

```
      Épicier: Elles sont belles mes outres ! Achetez-les !
(a priori destiné uniquement aux joueurs, ou alors les bots sont
sacrement intelligents)
      Message ("", "Elles sont belles mes outres ! Achetez-les")
```

5.4 Application au niveau de l'interface utilisateur

Le joueur peut envoyer des messages de 2 façons˜:

classique

Ex˜: tell Oumph "Ou étais-tu passe?" Ceci générera ce message ("", "Ou étais-tu passe?"). Il ne sera compris par aucun non-joueur˜; néanmoins n'importe quelle créature est capable de rapporter les paroles qu'elle entend (espion, commère, etc).

avec des alias

Ex˜: tell Oumph Job Ceci générera ce message ("Job", "What's your job?") qui pourra être compris par les bots et les joueurs. Une liste d'alias devra être implémentée dans le contrôleur. Le joueur aura bien sûr la possibilité de les modifier à sa guise. Si l'alias n'existe pas, 'tell Oumph cloporte' générera le message ("cloporte","cloporte").

5.5 Les codes de contrôle

Pour que les bots puissent mieux se comprendre, nous rajoutons les codes de retour NONE, ANS, WRONG_ANS et BAD_ANS. Ces codes ne sont pas lisibles par les joueurs et ces derniers ne peuvent pas non plus en utiliser directement (la commande 'answer' permettra d'envoyer un message avec le code de contrôle ANS). Leur usage est donc limité aux bots.

NONE est le code de contrôle par défaut. Il ne signifie rien. Il est employé quand aucun autre code de contrôle n'est utilisé.

Le code ANS signifie que le message est une réponse à une précédente question.

WRONG_ANS sert à indiquer que la réponse est considérée comme fausse et mensongère. L'interlocuteur peut donner ainsi une autre chance de dire la vérité (ou plutôt ce que le bot considère comme la vérité), changer de sujet de conversation ou bien rompre la discussion avec des conséquences fâcheuses (combat, baisse de la réputation...).

Ex.˜: un garde interroge un joueur.

```
      A guard is approaching.
      Guard: What's your name, stranger?
      (name, "What's your name, stranger?",NONE)
      >Janselmeer
      ("Janselmeer", "Janselmeer",ANS)
      Guard: You lie! I know you are not Janselmeer.
      ("", "You lie! I know you are not Janselmeer.", WRONG_ANS)
      >Oumph
      ("Oumph", "Oumph", ANS)
cas 1 :
      Guard: You lie! I know that you are not Oumph.
      ("", "You lie! I know that you are not Oumph.", WRONG_ANS)
      A guard attacks you.
cas 2 :
      Guard: I'm watching you Oumph.
      ("", "I'm watching you Oumph.", NONE)
```

Il convient bien sûr de limiter le nombre d'essais possibles... BAD_ANS indique que la réponse est incorrecte, non conforme au type de réponse souhaité.

Ex.˜:

```
A guard is approaching.
Guard: What's your name, stranger?
(name, "What's your name, stranger?", NONE)
>43
("43", "43", ANS)
Guard: "I said 'What's your name?'."
("", "I said 'What's your name?'." BAD_ANS)
```

Un code BAD_ANS généré par un bot est une erreur grave dans le code du jeu ou du bot. Un enregistrement de l'erreur doit être effectué pour débogage et correction.

Un code BAD_ANS généré par un joueur signifie une faute de frappe, une mauvaise compréhension de la question, un mensonge volontaire, une tentative pour perturber le bot, etc.

Le codeur est libre de faire réagir son bot comme bon lui semble (poli et compréhensif, énervé et agressif, etc). Le bot peut donc ou non reposer sa question.

Note˜: le bot ne peut et ne doit pas savoir s'il parle à un autre bot ou à un joueur. Pour cette raison, le nombre de réponses BAD_ANS doit être limité, afin qu'aucun bot bogué ne provoque de boucles infinies...

5.6 Les modificateurs d'expression.

Un modificateur d'expression est une information contextuelle qui accompagne le message. Cette information est signifiante pour le bot qui reçoit le message et lui permet de réagir de manière plus appropriée.

Par exemple, il y a plusieurs façons de demander son nom à quelqu'un.

- politesse: "Would you mind to tell me your name, please ?"
- neutralité: "What is your name ?"

- insulte: "What is your name, alien scum ?"

Dans chaque cas, seul le modificateur d'expression a changé. Voici ci-dessous les mêmes messages entendus par un bot :

- politesse: (name, NONE, POLITE)
- neutralité: (name, NONE, NEUTRAL)
- insulte: (name, NONE, INSULTING)

POLITE, NEUTRAL et INSULTING sont des modificateurs d'expression. Voici une liste de modificateurs d'expressions :

- OBSEQUIOUS
- POLITE
- NEUTRAL
- RUDE
- THREATENING
- INSULTING
- CHARMING

Cette liste n'est figée.

L'exemple suivant explique comment les modificateurs d'expressions peuvent être utilisés. Janselmeer rencontre Robert le boucher (le frère de Raoul) et lui demande des informations sur Raoul le boulanger.

```
Janselmeer: (RAOUL, "Do you know RAOUL?", NONE, NEUTRAL)
Robert:  ("", "I am not concerned with that RAOUL.", ANS, RUDE)
Janselmeer: (RAOUL, "O MIGHTY ONE, may I beg you a little of your
             knowledge about Raoul?", NONE, OBSEQUIOUS)
Robert: ("", "RAOUL? This thing exists only in your sick mind.",
         ANS, INSULTING)
Janselmeer: (RAOUL, "Tell me something about Raoul or I'll kill you!",
             NONE, THREATENING)
Robert: ("", "I swear I don't know Raoul.", ANS, NEUTRAL)
Janselmeer punches at Robert.
Janselmeer: (RAOUL, "Tell me something about Raoul or I'll kill you!",
             NONE, THREATENING)
Robert: ("", "Raoul is the baker. He lives North of here. Please don 't
         kill me!", ANS, POLITE)
```

Il n'y a aucune obligation d'utiliser un modificateur d'expression plutôt qu'un autre. C'est purement une question de comportement. Certaines personnes sont plutôt insolentes tandis que d'autres se montrent polies tout simplement parce qu'elles sont comme cela. Mais évidemment, les créatures n'auront pas les mêmes réactions avec les premiers qu'avec les seconds. Cela dépend d'elles. Par exemple, un seigneur est normalement habitué à ce qu'on s'adresse à lui de manière obséquieuse et pourrait s'offusquer de quelqu'un qui ne le ferait pas. Le choix du modificateur d'expression dépend donc à la fois de la personnalité du du personnage qui parle et de la réaction attendue de celui qui écoute.

5.7 Extensions futures du langage des bots

Le système des mots-clés a été choisi pour des raison de simplicité de mise en oeuvre. Néanmoins, cela devrait évoluer et tendre vers un vrai dialogue avec des phrases (sujet verbe complément). La première extension aux mots-clés pourrait être l'utilisation des pronoms et adverbes interrogatifs˜: what, who, where, why, how, how many, which, when.

Ex.˜:

```
>Who Raoul
Bot: Raoul is the baker.
>Where Raoul
Bot: You can find it in the vicinity of North Midgaard.
>What Raoul
Bot: I've never heard of this.
```

6 Les différents modes de la communication verbale

Le contenu de cette partie peut paraître évident, mais pourtant il est tout à fait essentiel. Nous distinguons 2 modes de communication, le dialogue et le monologue, qui seront utilisés à des fins différentes dans le jeu.

6.1 Le monologue

Le monologue est la situation où une créature émet un message qui se suffit à lui-même et par conséquent n'attend pas de réponse. Ce message peut être destiné à une créature en particulier, mais la plupart du temps il n'est pas dirigé.

Ex.˜: Un garde attaque Janselmeer.

```
Guard: Now you die, Janselmeer!
A guard attacks Janselmeer.
```

Le message est destiné à Janselmeer, mais toutes les créatures de passage peuvent l'entendre.

Ex.˜: Une caravane arrive sur la place du marché.

```
The caravan: The caravan has arrived! Look at my beautiful vegetables!
```

Le message est destiné à toutes les créatures de passage.

Ex.˜: Janselmeer viens de faire un achat.

```
Janselmeer: That was my last silver...
```

Le message n'est adressé à personne mais peut être entendu par toutes les créatures présentes (qui vont s'interroger sur la santé de Janselmeer :).

6.2 Le dialogue

Le dialogue est un échange d'informations par des messages entre 2 ou plus créatures.

Il s'effectue en mode connecté, avec une phase de connexion, une phase de conversation et une phase de déconnexion.

Voyons comme exemple de dialogue un touriste qui demande sa route à un passant˜:

```
-phase de connexion :
touriste : S'il vous plaît, monsieur...
Le passant s'arrête
passant : Oui ?...

-phase de conversation :
touriste : Je cherche le camping du coin.
passant : Vous tournez à gauche dans 500 m, vous continuez tout droit
          jusqu'à la fontaine et tournez à droite.

-phase de déconnexion :
touriste : Merci, bonne journée.
passant : C'était avec plaisir.
```

C'est ce genre de dialogue que l'on va tenter de reproduire. Un dialogue se déroule de la façon suivante˜:

```
La créature C1 veut parler aux créatures C2, C3, ..., Cn.
1- C1 demande à parler avec C2, C3, ..., Cn.
2- Certains Ci acceptent la demande, d'autres la refuse.
3- C1 parle avec ceux qui ont accepté.
4- Les participants se retire peu à peu. L'avant dernier retrait met fin
à la conversation.
```

L'avantage d'un mode connecté est de faciliter la gestion des créatures. Par exemple, si on parle à une créature qui est en train de se déplacer, celle-ci s'arrêtera le temps de la conversation (donc de la connexion).

Le dialogue devrait donc être le mode de communication privilégié avec les bots. Une fois un dialogue ouvert, certaines actions sont interdites (comme s'éloigner par exemple). Ces actions interrompent automatiquement la conversation. Un dialogue peut être interrompu par un événement extérieur si celui-ci nécessite obligatoirement une réaction de la part d'une créature engagée dans la conversation (un combat par exemple).

```
   Touriste                      |             Passant
--------------------------------------------------------------------------
>talk passant                    |
touriste: S' il vous plaît       | touriste: S'il vous plaît, monsieur...
  monsieur...                    | a tourist wants to talk with you.
                                 | Do you accept? (Y/N)
                                 | >Y
passant: Oui ?...                | passant: Oui ?...
>tell passant "Je cherche le     |
  camping du coin."              | touriste: Je cherche le camping du coin.
                                 | >tell touriste "Vous tournez à gauche dans
                                 |  500 m, vous continuez tout droit jusqu'à
                                 | la fontaine et tournez à droite."
passant: Vous tournez à gauche   |
  dans 500 m, vous continuez     |
  tout droit jusqu'à la          |
  fontaine et tournez à droite.  |
bye                              |
touriste: Merci, bonne journée   | touriste: Merci, bonne journée.
```

Il est à noter que la commande 'talk' permet l'ouverture du dialogue et que 'bye' met fin à la conversation. remarquons aussi qu'une interface utilisateur peut fournir un certain nombre de raccourcis pour le dialogue. Par exemple, il n'est pas nécessaire de demander à un joueur si il veut répondre. Il suffit qu'il réponde dans le cas positif ou bien qu'il fasse autre chose en cas de réponse négative. De la même façon, Il suffirait de faire simplement autre chose que parler pour mettre fin à un dialogue plutôt que de taper 'bye'.

Pour ouvrir un dialogue avec plus d'une créature, on rajoute les noms des créatures souhaitées en paramètre de 'talk' ou l'on utilise 'talk all' pour parler à toutes les créatures présentes.

Ex.~: Embauche dans une taverne

```
          Recruteur> talk all
          Recruteur: "Votre attention s'il vous plaît."
     [A ce stade, certains écoutent, d'autres pas.]
          Recruteur: "J'ai du travail à proposer, dangereux et mal payé."
     [Deux ou trois créatures continuent à écouter...]
          Recruteur: "3 pièces d'argent pour aller tuer Skarkash."
     [Gros silence dans la pièce]
          Recruteur: "Bon, tant pis."
          Recruteur> bye
          Recruteur: "Merci de votre attention."
```

Ex.~: Contrôle d'identité dans la rue. Le garde recherche un elfe évadé.

```
Garde> talk elfe1 elfe2
Garde: "Eh toi et toi !"
Garde: "Quels sont vos noms ?"
Elfe1: "Celeborn"
Elfe2: "Galadriel"
Garde> bye
Garde: "Très bien, circulez."
```

Remarque~: cet exemple met en avant les difficultés de codage sous-jacentes. La phrase associée à un mot-clé dépend du niveau de langage de celui qui parle (politesse, élocution, etc), de son métier, du nombre de personnes à qui l'on s'adresse, du sexe de ces personnes, etc.

6.3 Les messages et la perception

Le système de dialogue dépend des perceptions. En effet, un message n'est rien d'autre qu'un stimulus particulier qui contient beaucoup d'informations. Un message est associé à un ou plusieurs Stimuli, ce qui permet de choisir le médium (son, télépathie, etc) et l'intensité du message. Il est donc possible de choisir sa façon de communiquer.

7 Des dialogues multilingues

7.1 Langues différentes dans le jeu

Les créatures dans un monde n'ont pas forcément toutes le même langage ni la même façon de s'exprimer. Pour gérer la différence entre les langues, un attribut 'langue' est introduit dans les messages. A la réception d'un message, une créature vérifie d'abord qu'elle connaît la langue dans laquelle est formulé ce message avant de pouvoir le lire. Si ce n'est pas le cas, le contenu du message apparaîtra codé (et donc inintelligible) au joueur ou au script/IA qui le reçoit.

Une créature peut néanmoins mémoriser un message qu'elle ne comprend pas (problème de langues). Par contre, elle ne peut mémoriser un message si elle ne possède pas le sens requis pour sa perception.

7.2 Support multilingue pour les joueurs

Pour répondre à l'objectif d'avoir un jeu multilingue (anglais, français, espagnol, allemand...) , il est nécessaire que les dialogues le soient aussi dans la mesure du possible. La solution consisterait à intégrer dans les messages la même version du texte dans des langues différentes. Ceci concernerait les messages émis par les bots mais aussi les textes des alias des joueurs.

Ex.˜: job -> ("What's your name?", "Comment t'appelles-tu˜?", "Wie heisst du˜?")

Les joueurs pourront ainsi lire les messages dans la version qu'ils désirent, si elle est disponible, sinon ce sera la version par défaut (qui est toujours disponible). Par contre, ils n'auront bien entendu pas à formuler leurs messages en plusieurs langues. La représentation interne des mots clés standards (name, job...) restera toujours la même mais il n'est pas exclu que ceux-ci soient traduis par l'interface utilisateur. D'autres mots-clés doivent absolument être traduits en plusieurs langues par les créateurs de monde ou les traducteurs. L'exemple suivant montre bien la nécessité de traduire certains mots-clés˜:

Exemple du boulanger en version anglaise˜:

```
>Job
Raoul: I am the baker.
>baker
Raoul: I cook breads, pies and cakes.
>cakes
Raoul: My cakes are the best of Midgaard.
```

En version française˜:

```
>Job
Raoul: Je suis le boulanger.
>boulanger
Raoul: Je fait du pain, des tartes et des gâteaux.
>gâteaux
Raoul: Mes gâteaux sont les meilleurs de Midgaard.
```

Si les mots-clés

```
baker
```

et

```
cakes
```

ne sont pas traduits, un joueur en version française aura du mal à faire progresser la conversation˜:"

```
>Job
Raoul: Je suis le boulanger.
>boulanger
Raoul: Je ne connais pas.
>gâteaux
Raoul: Je ne connais pas.
```

Ceci est piège dans lequel il ne faut pas tomber pour les traducteurs.

Précisons enfin pour éviter toute confusion, qu'un mot clé en français ne signifie pas que la réponse sera uniquement en français et idem dans les autres langues. Dans tous les cas la réponse doit rester multilingue pour que les autres joueurs qui écoutent le message et qui utilisent une autre langue puissent le comprendre. En fait, il y aura juste un ensemble de mots-clés associé à un ensemble de réponses (boulanger, baker)->("Je fait du pain, des tartes et des gâteaux.", "I cook breads, pies and cakes.")

8 Résumé

Les mots-clés standards (valables pour tous les bots)˜:

- job = "What is your job?"˜: requête de l'activité de la créature.
- name = "What is your name?"˜: requête du nom de la créature.
- joker = refus de répondre à une question.

Les codes de contrôle˜:

- NONE˜: par défaut (aucune signification),
- ANS˜: réponse,
- BAD_ANS˜: réponse non valable,
- WRONG_ANS˜: réponse fausse.

Les modificateurs d'expression˜:

- OBSEQUIOUS
- POLITE
- NEUTRAL
- RUDE
- THREATENING
- INSULTING
- CHARMING

Un message est constitué de˜:

- un attribut indiquant la langue,
- un attribut indiquant le (ou les) destinataire du message,
- un texte court interprétable par les bots,
- un texte long pour les joueurs disponible en plusieurs langues (dont une langue par défaut au minimum),
- un code de contrôle.

Les commandes˜:

- tell˜: envoie un message à une ou plusieurs entités,
- talk˜: ouvre un dialogue avec une ou plusieurs entités,
- bye˜: ferme un dialogue,
- answer˜: répond à une question.