RFC - Quests and missions

Metacosm Development Team

Revision History

Revision 0.8 March, 13th 2001 Revised by: Ruffy
Detailed QuestEditor part
Revision 0.7 March, 5th 2001 Revised by: Ruffy
(Elkine Birthday version) Implementation section rewritten
Revision 0.6 April, 11th 2000 Revised by: Ruffy
First version in English - add of Implementation section
Revision 0.5 March, 5th 2000 Revised by: Ruffy
Conversion to Metacosm-DTD
Revision 0.4 January, 29 2000 Revised by: Ruffy
Adds in Credibility section - first translation
Revision 0.3 November, 23th 1999 Revised by: Ruffy
Definitions of Quest and Mission, rewrite of Quests, Missions and Plausible sections
Revision 0.2 October, 27th 1999 Revised by: Ruffy
Corrections and updates in Multiple Questers section
Revision 0.1 September, 7th 1999 Revised by: Ruffy
First try

This document describes quests and missions available in Metacosm.

1. License

2. Metacosm

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

3. Notations

Terms beginning with an uppercase letter are important concepts of the game. For easiness to understand and to translate, authors have tried to avoid long and complex sentences. Translators should do the same, by always choosing simplicity.

Some used terms are defined in the Vocabulary-RFC; look at the official web site.

Missions and Quests have names: these aren't available to players and are only used to help in discussions about specifications.

4. Missions

Mission

A Mission is in fact a moral contract: the one who accepts the Mission signs to accomplish the Mission objective, by respecting given conditions. The one who proposes the Mission signs to give the reward to the one who succeeds in the Mission.

A Mission is given by a bot or an other player.

We only care in this document about Missions given by bots; the ones given by players are managed by the player who proposes them, and the game doesn't interfere in this case, it's pure roleplay.

The one who proposes the Mission is a Missions giver; to give Missions is his/ her job, or more often this bot decides to propose a Mission after one or more events happen (friend death, recent gossip, etc).

Missions are time limited or not: after a certain date (duration or happening of a particular event (end of another Mission, etc)), the bot who has launched the Mission can decide it's cancelled (with or without informing questers), choose other questers in addition or replacement, send questers to kill or punish bad questers, etc.

Missions type:

- Murder
  - To kill a given being or a group or a category of beings.
  - In a given place or not.
  - To make corpse(s) disappear or not.
  - To send claim or not.
- Steal
  - To steal a special object or not, a max of objects from a given type (money for example), or a max of objects in general (all possessions from a being for example).
  - To steal a particular being or anybody.
  - To steal in a given place or not.
  - To be allowed to kill or not.
  - To have to be stealth or not.
  - To send claim or not.
  - Only steal what needed or not.

- Delivery
  - To deliver something to a being.
  - A special object (a message for example) or a quantity of something or a set of given objects.
  - No to kill the receiver.
  - The receiver gives or not something to pay. In the first case, pay should or not be got back to the Mission giver.
  - To have the right to steal the receiver or not.
- Recovery
  - To go somewhere to get back something.
  - A particular object or a quantity of something or a set of given objects.
  - On a being or at home of somebody in particular or anybody.
  - In a given place or not.
  - Not to kill the owner (he/she/it agrees, if not see Steal).
- Rumour
  - To propagate a given idea among people.
  - A group of people, a given number of people, one person in particular or anybody.
- Buy (variation of Recovery)
  - To have to buy or pay in some manner (barter, service, etc).
  - Not to steal.
- Recruiting
  - To find somebody to do something.
  - A particular being, a being matching criteria or anybody.
  - To do a Quest or a Mission or to enter in a group (clan, religion, etc)

- Spying
  - To obtain information about something or somebody.
  - For a place: to discover this place whose we only know some properties (its name, its inhabitants, its buildings, etc), to get back data about this place (population, building, etc).
  - For a person: to find his/her/its location, to learn rumours about the person, to know about his/her/its possessions, to discover the group whose the person is a member, his/her/its objectives (Missions, Quests, etc), etc.
  - For a group of people (clan, religion, town, state, etc): to find meeting place(s), members, group goals, possessions, characteristics, etc.
  - For an object: to find its location, its price, its characteristics, to know about the number of existent objects and their location, etc.
- Building
  - To build or to make, or to make build or make make something.
  - For an object: a special object or a quantity of some object or a set of given objects.
  - For a place: to build a special building, to modify a place (digging, irrigation, drying, farming, plantation, etc).
  - In a given place or not.
- Destruction
  - Contrary of Building Mission.
  - To kill or not some people (place inhabitants, object makers, etc).
- Enriching
  - Modified version of Recovery Mission: to get back money or valuables.
- Protection
  - To protect from somebody or something while a certain duration.

    - For a being: to avoid the given being is killed or stolen.
    - For a place: to avoid it’s altered.
    - For an object: to avoid it’s stolen or destroyed.
- Sell
    - Do not mix up with Delivery Mission in the case of recipient give money for delivery. Sell Mission is selling the highest price something to somebody.
    - A special object or a quantity of a given object or a set of given objects, a place (building, etc) or a person (slave, servant, etc).
    - The buyer should or not respect some criteria (member or not of groups, discrimination about race, etc).
    - No to kill the buyer.
    - To get back or not the payment to the Mission giver.
- Kidnapping
    - To lead a given being to a given place, by using violence if necessary.
    - Not to kill the victim.
    - Possibility of hurting the victim or not.
- Investigation
    - To discover guilty(ies) of a given action.
    - By following some rules (not to kill, not to steal, etc) or not.
- Counter Mission
    - To make a given Mission fails (so all questers working on this Mission) or only one quester on a given Mission.
    - To kill or not the quester(s).
    - Time aspect is important for this mission: the goal is to success before the others did.

The generic pattern is something like doing/successing a given Action (or several) with more or less constraints.

5. Quests

Quest

> Like the Mission, the Quest is a moral contract, but it plans to succeed in a more complex objective. The Quest is a longer Mission, with a primary objective and several secondary ones.
>
> To succeed, the quester should complete all the secondary objectives.

Like the Mission, the Quest can be limited in time.

At the implementation level, Quests are a logical sequence of 4 or 5 Missions. They should be linked together, even if the link isn't obvious: they all should work for the same objective. We should avoid a Quest is simply several disparate Missions one after an other.

Quests types:

- Sequence of same type Missions (Murder Missions for all a category of people, several Delivery Missions, ...).
- Bad trick: Steal Mission(s) + Delivery Mission(s), with or without claim.
- Transaction: Recovery Mission(s) + Delivery Mission(s)
- Conspiracy: miscellaneous Mission(s) + Rumour Mission(s), to make Mission(s) attributed to somebody else, to send somebody to jail, etc.
- Resell: Bought Mission(s) + Delivery Mission(s)

- Harassing: Steal Mission(s) and/or Murder Mission(s) and/or Destruction Mission(s) against interests of a person or a group.
- Destitution: Conspiracy Quest to obtain a change at some position.
- Justice: Investigation Mission(s) + punishment against guilty(ies). (function of cases, Murder Mission(s) or Kidnapping Mission(s) to lead them in a court of justice).
- Rivalry: Spying Mission(s) to know about Quests of an enemy group, followed by Counter-Mission.

6. Plausible

Quests and Missions should be diverse and avoid repetition. Warning: it should be plausible. Why such a person ask for such a Quest or Mission?

Quests and Missions are showed in a plausible form to the player: a Murder Mission will not be asked like this

```
Robert: "Murder mission against Raoul the baker"
```

but more like this

```
Robert: "Raoul the baker slept with my wife. Make him dead."
```

or like this

```
Robert: "I want Raoul the baker is dead before the sunset.
   My reasons are private."
```

A Quest composed of a Murder Mission, a Kidnapping Mission, a Recovery Mission and a other Murder Mission will not be proposed like this

```
Sedecias: "Kill Raymonde. Kid-
nap Gertrude. Ask 10,000 gold pieces
  to Raoul. Kill Gertrude."
```

but more like this

```
Sedecias: "I hate the Raoul the baker suc-
cess. I want to make him
  suffer. *Niark* *Niark*. Kill his wife Raymonde, kid-
nap his daughter Gertrude,
  ask 10,000 gold pieces as ran-
som and then kill Gertrude. *Niark* *Niark*"
```

In the previous example, the quester know by advance all of the Quest. It's also possible to give information to quester only progressively. Information can be given by the Quest giver or by an other person.

```
   Raoul: "My wife Raymonde has disap-
peared, help me to get her back.
           I suggest you begin with Sedecias. He has al-
ways hated me."
   [The player goes to Sedecias...]
   Sedecias: "I know where Raymonde is but I'll tell it to you only
             if you get me back a shark teeth."
   [The player comes back with a shark teeth]
   Sedecias: "Damn', you succeed..."
   [Sedecias throw negligently the shark teeth.]
   Sedecias: "All the same, I'll keep my promise: take this letter.
             I stole it in the Raoul mailbox."
   [Sedecias gives the letter to the player who reads it.]
   "Dear Raoul,
        I feel tired, so I decided to go one week in tha-
lasso. I'm sorry
   not to have told it to you, but you were work-
ing and my friend Denise
   had just given her place to me, because she can't use it(...)."
   [The player goes to the thalasso centre]
```

```
   Receptionist: "Sorry I can't give you our clients name.
                    But you can work here to-
day, you could meet them.
                    The pay is 3 silver pieces a day.
   [The player works but don't find Raymonde.]
   [He decides to meet Denise.]
   Player: "Where is Raymonde?"
   Denise: "Ha, ha. Too late! I always hated Raymonde since she won
   the applepie contest. Now, I'll have my re-
venge. I put her in a package
   and send her to Klouguirec, and no-
one could prevent me from winning the
   applepie contest next week. Under-
stand that I can't let you go now you
   know the truth, you'll die!"
   [Denise attacks the player.]
   [The player kill Denise and go back to Raoul.]
   Raoul: "Thanks a lot. Now I know Ray-
monde is at Klouguirec, I'm better.
           I'll go immediately to recover her. As a re-
ward, my bread is
           free for you for one year."
```

It brings a dynamical and non linear aspect to the Quest: if the player would have been killed by Denise, Raoul would have sent an other quester; the player could have killed Sedecias because he laughed at him; the player could have lied to Raoul, done no research and said his wife was at Schmuziskthan; the reward given by Raoul wasn't determined by advance, it can be function of time spent to complete the Quests or of some others factors; etc.

Events before the Quest could have never existed, as the stolen letter (even if it is always possible such events happened without witnesses). But the Quest could also be based on true events or facts (for example the hate between Raoul and Sedecias).

Future events discovered during the Quest can also happen 'really'. For example, the applepie contest could really be organised (a 'cook the best pie' Quest is possible, with a research of ingredients, cookbook, etc), or player could see Raoul to close his bakery and to go.

7. Multiple Questers

Quests and Missions can be realised by players groups (and at long term by non-players groups). It's recommended to use at the maximum this ability to create a true social interaction between players.

A players group can work on a Quest or a Mission together (all the players work on the same Mission, eventually each one on one special objective if multiple objectives) or in cooperation (if several Missions should be realised, the players should act in parallel to accelerate the Quest).

Moreover, it's also recommended to use at the max the oppositions between players or between groups ('bodyguard' Mission against 'murder' Mission (attack/ defence) or 'find the treasure' Missions in parallel (competition), for example).

8. Implementation

To implement Quests and Missions, we can heavily used functionalities given by Interactions, Dialog and Perceptions systems. Cf related RFC to more info about these systems.

The Quests system is heavily flexible and generic.

8.1. Quests creation

Quests can be created with the QuestsEditor (manual creation) or automagically by the game itself (in reaction to some events).

8.2. Quest object

The game manages a list of Quest objects.

A Quest object contains:

- QuestID
- Quest giver (EntityID)
- Time limit (date or event)
- Questers list (EntityID, beginning date, Quest progression state)
- Quest state machine
- Requested criterions
- Forbidden criterions

Requested and forbidden criterions determine who can do a given Quest and who can't (for example, murders are for evil people).

> It's possible a quester doesn't match anymore the criterions during or at the end of the Quest.

The states in the state machine are used to keep questers progression.

The Quest can be not linear (several ways to one or several exits).

For Quests for groups, it's possible to divide the group (several states should be accomplished at the same time) or not (action to difficult for one person).

A Quest can be unique (only one quester can succeed). It's removed after the first success.

A Quest can be 'one times only' (a given quester can do it only one times). A mark is kept on the quester for remembering.

Example 1. State machine of a Murder Mission

```
(0) --> (1) --|--> (2) |--> (3)
```

State 0: the Quest giver explains the Mission/Quest. 0 -> 1: the quester kills the victim. State 1: (nothing) 1 -> 2: the quester brings the corpse to the Quest giver State 2: reward 1000 gold. 1 -> 3: the quester brings just the victim ID card to the Quest giver. State 3: reward 100 gold (and if he/she/it lies, the quest giver could send new questers to kill the lier).

To detect a state change, we could wrap a detector around some Actions. For example, if the quester should deliver an object to somebody, the "to give" Action is wrapped. If the quester calls the new Action with the good object and the good receiver as parameters, the Quest state is changed and the original Action is called. If not, the original Action is directly called. The quester can also progress in the Quest just by saying a word to somebody (see Dialog-RFC).

Each Character keeps a list of QuestID (for missions/quests he/she/it's doing).

8.3. QuestsEditor

The QuestsEditor offers a GUI interface to create Quests (a text interface can also exist). To create a Quest, you need to provide a list of potential Quest givers, the requested and forbidden criterions and a state machine.

The list of potential Quest givers can contain:

- EntityID (0, 1 or several) of potential giver(s)
- Influence(s): each Entity affected by this (or at least one of these) Influence(s) is a potential giver
- Named group(s): a named group contains EntityID(s) and/or Influence(s). It has got a name because it's frequently used or because it's easier to manipulate explicit lists.

Items of the list/a named group can be affected by 4 operators:

- NOT
- AND

- OR
- ()

This list can't be empty.

Example 2. A list of potential Quest givers

[EntityID: Raoul] AND ([Group: SecurityForces] AND NOT [Type: elf]) where SecurityForces = ([Influence: Guard] OR [Influence: Watchman] OR [Influence: Fighter])

As soon as one of the Quest givers matches criterions, a Quest object is created and the Quest giver can propose the Quest.

The criterions list can contain:

- Property(ies): values to match for the quester (exact value, range or set of values)
- Influence(s): influence(s) the quester should have (or not)
- Possession(s): object(s) the quester should have (or not)
- Named group(s) of Property(ies) and/or Influence(s)

Items of the criterions list/a named group can be affected by the 4 operators.

It could also contain affects, memories, relationships, as soon as they had be defined...

Example 3. A criterions list

Quester should match [Property: Strength >= 13] OR [Influence: Guard] AND NOT [Possession: Malediction ring].

The state machine is defined by an initial state, one or more transitions and state effects.

A transition is defined by an origin state, a destination state and a transition condition.

A transition condition can contain:

- Property(ies): values the quester should match
- Influence(s): new influence(s) the quester should acquire or influence(s) he/she/it should loose.
- Possession(s): object(s) the quester should have (or not)
- Action(s): action that should be done to allow transition. Only OR and () operators can be used for Action(s). At least one Action should be provided (it'll be used to test transition).
- Named group(s) of Property(ies) and/or Influence(s) and/or Possession(s) and/or Action(s)

Items of a transition/a named group can be affected by the 4 operators (except for Action(s)).

State effects are operations to be executed when the corresponding state is reached. Operation mean a list (EntityID, Action, Parameters).

Example 4. A state machine

Initial state '0' Transition: '0' to '1' ([Action: 'kill' with victim's EntityID as parameter]) Transition: '1' to '2' ([Action: 'give' with quest giver as destination and victim's corpse as object] AND [Possession: corpse] Transition: '1' to '3' ([Action: 'give' with quest giver as destination and victim's ID card as object] AND [Possession: ID card] State 0: [EntityID: Quest giver][Action: explains][Parameters: MissionID] State 1: (nothing) State 2: [EntityID: Quest giver][Action: special_give][Parameters: 1000 gold to quester] ('special_give' is 'give' if the quest giver has 1000 gold and 'say I am sorry and give what I am’ if not) State 3: [EntityID: Quest giver][Action: defiant_special_give] [Parameters: 100 gold to quester] (’defiant_special_give’ is ’special_give’ with only 100 gold, and a quest murder is created, with rumour ’victim is alive’ as needed criterion)