ChatScript : le tutoriel

Traduction Mathieu Rigard 02/12/2017 pour Utopia : m.rigard@utopia-french.tech

A faire : correction et suite de la traduction (20 % effectués seulement), ancrer les liens.

Ce tutoriel est conçu pour vous montrer comment utiliser ChatScript pour créer un Bot simple et fonctionnel - un Bot capable d'aider un humain à planifier un voyage.

Pour commencer téléchargez ChatScript, lisez le manuel de base à la partie présentation et assurez-vous que vous pouvez échanger avec le Bot par défaut au sujet de son enfance.

Maintenant, afin de créer un tout nouveau script de chat : Créer un dossier et nommez-le TEST au sein du dossier RAWDATA. Créez un fichier vide et nommez-le tutoriel.top, au sein de votre dossier TEST.

Copiez le fichier RAWDATA/filesHarry.txt présent dans le .zip de ChatScript et renommez le RAWDATA/filesmine.txt.

Ouvrez filesmine.txt, et inscrivez-y le chemin vers votre nouveau projet TEST après le chemin destiné à HARRY. Votre filesmine.txt dois ressembler à ceci :

```
RAWDATA/HARRY/simplecontrol.top # default bot control
RAWDATA/TEST/tutorial.top # tutorial bot data
```

A partir de maintenant nous allons travailler dans votre fichier tutorial.top.

NOTE: Ce tutoriel ne couvre qu'une infime partie des possibilités de ChatScript - seulement celles qui nous permettront de construire notre Bot agent-de-voyage. Vous trouverez d'autres fonctionnalitées dans le reste de la documentation.

Sommaire

Dialoguer (topic: , t:)

Le script de Chat le plus simple possible doit posséder au moins un topic (sujet) et une phrase à prononcer, par exemple :

```
topic: ~introductions []

t: Salut, je t'écoutes !
```

Où :

- Le nom (ou mot-clé) attribué à topic: définit le sujet de la conversation. Nous appellerons notre topic ~introductions (le nom d'un topic commence toujours par ~).
- Le mot clé attribué à t: présente une phrase à dire dans un topic (nous l'appellerons un "gambit", c'est de que dit le Bot sans attendre qu'on l'interpelle).
- Insérez ce texte dans tutorial.top. Ensuite, dans l'invite de commande du serveur local (en mode client, donc) tapez :build mine. Vous devriez voir quelque-chose comme :

```
>:build mine

----Reading file simplecontrol.top
Reading outputmacro: ^harry
Reading outputmacro: ^georgia
Reading table tbl:defaultbot
Reading topic ~control
Reading topic ~alternate_control

----Reading file tutorial.top
Reading topic ~introductions
No errors or warnings in build
Read 302,955 Dictionary entries
Read 304,588 Dictionary facts
Read 110,851 Build0 facts
Read 1 Build1 facts
Concept sets: 1421
Free space: 123,252,280 bytes FreeFacts: 4,584,560

Hello, talk to me!
>
```

Le chat avec ce Bot devrait donner un dialogue du type :

```
 t: Salut, je t’écoutes !

 >hi
 I don't know what to say.

 >why?
```

```
I don't know what to say.
```

Le bot dit seulement ce qu'on lui a dit de dire, puis il est coincé, car il n'a rien d'autre à dire.

Si vous êtes fatigué de parler avec lui, vous pouvez quitter le client en utilisant la commande :quit.

En dire plus (^keep, sélection aléatoire [], ^repeat)

Rendons notre Bot plus intéressant :

```
topic: ~introductions []

t: ^keep() [Salut] [Hé] [Tiens], [alors] [vas-y] [dis-moi quelque-chose] je t’écoute !
```

Ici :

- Nous avons utilisé la fonction `^keep`, qui demande au Bot de ne pas effacer une règle après qu’elle ait été utilisée (normalement, une règle n’est utilisée qu’une fois puis elle est mise au rebut).
- Nous avons également ajouté quelques options entre crochets, parmis lesquelles le Bot peut choisir au hasard. Dans n'importe quelle séquence continue de phrases entre crochets, le robot peut choisir une au hasard. Lorsqu’il a une série phrase entre crochets, le Bot en sélectionnera toujours une au hasard.

Une fois que nous avons produit le :build de notre chatbot, le dialogue pourra devenir le suivant :

```
Salut, dis-moi quelque-chose je t’écoute !
```

```
>Bonjour

Hé , vas-y je t’écoute !

>Quoi ?

Tiens, alors je t’écoute !

>Pourquoi ?

Alors, vas-y je t’écoute !

>who

Alors, dis-moi quelque-chose je t’écoute !

>abc

I don't know what to say.

>def

I don't know what to say.
```

Le Bot essaie certaines combinaisons de notre modèle de sortie aléatoirement, puis se bloque à nouveau alors qu'il commence à se répéter. Normalement, un Bot n'aime pas se répéter, sauf si on le lui demande expressément à l'aide de la macro ^repeat() :

```
topic: ~introductions []

t: ^keep() ^repeat() [Salut] [Hé] [Tiens], [alors] [vas-y] [dis-moi quelque-chose] je t’écoute !
```

Avec un tel Bot on peut Chatter à l’infini.

---------- Le tutoriel original s’arrête ici ----------

à faire...