



État de l'art

SmartInteract

Équipe C: Alexandre Bolot, Laura Lopez, Théos Mariani et Alexanne Masson



État de l'art

SmartInteract est un outil d'aide au développement pour les jeux possédant des dialogues que l'on souhaiterait rendre plus réalistes et vivants en se basant sur du NLP et fonctionnant comme une API. Plus précisément, nous séparons deux aspects distincts au sein du projet :

- Côté Développeur: Le moteur de jeu fournit la possibilité au joueur de dialoguer par saisie clavier avec les personnages enregistrés dans le jeu. Puis il renvoie une réponse adaptée et précise permettant d'avancer dans le jeu.
- Côté Désigner: Une interface graphique qui permet au développeur de créer des personnages et leur attribuer des connaissances. Elle permet aussi de visualiser des logs du déroulement d'interactions en direct, d'utiliser un outils de debug qui permet de dialoguer avec un personnage sans passer par le jeu et de récupérer des statistiques liées aux personnages.



L'innovant

Utilise des technologies récentes de NLP qu'il met au profit de jeux vidéo ce qui n'a aujourd'hui jamais été réalisé.



Une solution accessible à tous les langages dans les jeux possédant des dialogues.

Un projet qui ne nécessite aucune connexion obligatoire.



Interface de monitoring du moteur de jeu proposée au développeur.

État de l'art



Nous cherchons donc à garantir l'innovation de notre projet en nous renseignant sur :

- Compatibilité d'un moteur de dialogue avec différents langages de programmation.
- Les technologies récentes de NLP mises au profit de jeux vidéo.
- Technologies de NLP hors ligne.

Nous n'avons pas de concurrent direct.

Cependant nous avons des concurrents indirects :

- Les assistants personnels (Google, Alexa, Siri...), mais ceux-ci ne sont pas dans l'univers des jeux vidéos et ont besoin d'une connexion permanente à Internet.
- Snips est un outil permettant de construire des assistants personnels à l'aide d'une interface graphique, de plus il peut fonctionner sans connexion Internet. Mais il n'est pas destiné aux jeux vidéos.
- Les moteurs de jeux video, le plus connu est Unity. Mais ils ne permettent pas d'utiliser du langage naturel (NLP)

Positionnement technique

Au niveau des technologies, nous avons défini que le code de nos composants sera réalisé en Java car c'est le langage avec lequel nous avons le plus d'affinité, qu'il est performant, et que fournissant une interface d'interopérabilité, nous n'avons pas à nous soucier de problèmes de compatibilité. Nous avons également besoin de faire du traitement de langage naturel, et de communiquer de manière distante hors ligne avec les systèmes externes. Pour cela, nous avons effectué les études suivantes :

Notre choix se porte sur CORBA car :

- Nous avons besoin d'un système opérant inter-langage pour permettre à notre moteur d'être utilisé par des jeux en C, C++ et Java.
- Le langage de description IDL nous permet de facilement générer des Stubs dans les trois langages cités ci-dessus, avec un seul fichier de description.

| ORBs | Langage | Spécification |
|----------|--------------|-----------------------|
| CORBA | C, C++, Java | Langage dédié IDL |
| DCOM | C++ | Utilise protocole RPC |
| Java RMI | Java | Model objet Java |

Positionnement technique

Au niveau du traitement automatique de langage naturel, nous avons choisi Spacy pour ses performances, son panel de langues et la qualité de la documentation en ligne.

*Vitesse en mots
par seconde
Ar : Arabe
Zh : Chinois
Hi : Hindi
Ur : Ourdou
El : Grec*

| Outils | Langage | Précision | Vitesse | Langues reconnues |
|--------------------------|---------------|------------|------------|--------------------------------|
| Spacy | Python/Cython | 92,6 | 13.963 | De, El, En, Fr, It, Nl, Pt, Sp |
| Natural Language Toolkit | Python | non trouvé | non trouvé | Beaucoup de langues |
| Apache Lucene | Java | non trouvé | non trouvé | non trouvé |
| Stanford Core NLP | Java | 89,6 | 8.602 | Ar, De, En, Fr, Sp, Zh |
| Clear NLP | Java | 91,7 | 10.271 | Ar, En, Hi, Ur, Zh |
| MATE | Java | 92,5 | 550 | De, En, Fr, Zh |
| Turbo NLP | C++ | 92,4 | 349 | non trouvé |



Critères d'évaluation



Intégration

- Notre moteur de jeu doit être compatible avec un maximum de langage de programmation.
- Évaluation du critère: Réussir à s'intégrer à un jeu. Ce dernier doit comporter des dialogues entre joueur et personnage.
- Nous sommes en train de nous intégrer à l'équipe Reroll the game master. Car ils font un role play connecté, comportant des dialogues.



Efficacité

- La réponse doit être adaptée, précise et rapide.
- Évaluation du critère: Vérifier que la vitesse de réponse du moteur soit inférieure à une seconde.
- A l'utilisation le temps de réponse est de l'ordre de la milliseconde.



Langage

- Le moteur doit pouvoir accepter un niveau de langage courant et avoir une tolérance aux fautes d'orthographe.
- Évaluation du critère: L'utilisateur saisit une phrase, notre moteur lui donne une réponse adaptée.
- Pour le moment nous repérons des mots clés, le type de question, des patterns de phrase et les mots qui n'existent pas.