Projet Fin d'Etude

Modélisation par apprentissage du comportement non verbal entre deux chatbots.

Rapport Bilan de Gestion de Projet

Membres :

Begum Bekiroglu
Cherif Igui
Enzo Lopez
Yasmine Djabri
Haoyu Wang
Sara Nait Atman
Nacereddine LADDAOUI

Encadrants de projets :

Catherine Pelachaud
Directrice de Recherche CNRS, ISIR
Sorbonne Université

Nicolas Obin
Maître de conférences, IRCAM
Sorbonne Université

2022 - 2023

Le 3 février 2023

Table des matières

1 Introduction

Dans ce rapport bilan de projet, nous allons détailler les stratégies que nous avons adoptées et des implémentations de codes, puis nous allons présenter un bilan comparatif sur ce qui a été prévu initialement et ce qui a été réalisé lors de la durée de projet. Le but de notre projet étant d'améliorer la communication verbale et non-verbale des agents virtuels qui se trouvent sur la plateforme de contrôle Greta, notre rapport est constitué de deux parties principales dédiees, respectivement, à l'intégration d'un chatbot et l'amélioration des mouvements de l'agent virtuel.

La consigne initiale pour notre projet était d'intégrer un modèle de conversation sur Greta, rendre les conversations plus naturelles en ajoutant des propriétés phonétiques comme des pauses et synchroniser les paroles en ajoutant des comportements multimodaux. Néanmoins, la description de notre projet a été modifiée dès notre première réunion avec notre encadrante. Étant donné le nombre de personnes qui se trouve dans notre groupe, notre projet s'est complexifié en devenant l'intégration d'un chatbot ainsi que la conversion des mouvements de Greta, qui est en 2D à 3D.

2 Organisation

Nous avons divisé notre groupe en deux parties pour réaliser nos deux tâches principales, d'intégration de chatbot et transformation de pose. Nous avons assuré une bonne communication entre ces deux équipes en s'informant des avancés de nos taches à chaque étape. Nacereddine Laddaoui, ayant fait son stage de M1 à ISIR sur Greta, a également été consultant pour le modèle conversationnel même si il faisait partie du groupe de conversion.

Tâches \ Acteurs	Nacereddine Laddaoui	Begum Bekiroglu	Yasmine Djabri	Cherif Igui	Enzo Lopez	Haoyu Wang	Sara Nait Atman
Phase 1: Phase préliminaires							
Rapport Avant Projet	R	R	R	R	R	R	R
Rapport Ingeniérie Système	R	R	R	R	R	R	R
Recherche pour le Modèle Conversationnel	I	I	I	R	R	R	R
Recherche de base de données	R	R	R	I	I	I	I
Recherche pour le Passage de 2D à 3D	R	R	R	I	I	I	I
Comprehension de Greta	R	R	R	I	R	I	I
Phase 2: Phase de développement							
Modèle Conversationnel							
Trouver un modèle fonctionnel	C	C	C	R	R	R	R
Integration de modele conversationnel à Greta	C	I	I	R	R	I	I
Integration des mouvements selon speech	C	I	I	R	R	I	I
Integration de speach-to-text	I	I	I	R	I	R	R
Interface graphique	I	I	I	R	I	I	R
Conversion de 2D à 3D							
Creation de base de donnée de pose en 2D	R	C	C	I	I	I	I
Implementer les modèles de conversion	C	C	R	I	I	I	I
Integration sur Greta	C	R	C	I	I	I	I
Phase 3: Phase de Presentation							
Rapport Bilan Projet	R	R	R	R	R	R	R
Film	R	R	R	R	R	R	R

FIGURE 1 – Rôle des étudiants. R : responsable de la réalisation de la tâche, C : les personnes consultées et qui ont apporté leur aide, I : les personnes informés. Il y a aussi le label A qui signifie "accountable", nous avions Madame Catherine Pelachaud et Monsieur Nicolas Obin qui avaient ce rôle dans notre projet.

3 Modèle conversationnel

Dans le cadre de ce projet, nous avons choisi d'utiliser un modèle conversationnel[1] pré-entraîné développé par OpenAI appelé "ChatGPT" [6]. Ce modèle utilise des techniques d'apprentissage automatique pour générer des réponses à des questions posées en langage naturel. Il a été développé en utilisant le modèle Davinci-003 de OpenAI. Nous avons intégré ce système pour permettre une interaction avec un utilisateur humain. Lorsque l'utilisateur parle, ChatGPT génère une réponse qui est ensuite transmise à la plateforme Greta qui utilise NVBG [1] (Non Verbal Behavior Generator) pour associer une animation à la réponse et la jouer et prononcer le discours de la réponse.

1. NVBG est un système qui utilise un logiciel appelé SmartBody pour permettre à un utilisateur de réaliser des gestes calculés. SmartBody est utilisé pour exécuter les gestes trouvés par NVBG.

3.1 L'interface

Afin d'intégrer ChatGPT avec Greta, nous avons conçu une interface graphique utilisateur permettant une interaction directe et intuitive avec le chatbot, sans nécessité d'utiliser un terminal. Nous avons utilisé l'interface graphique TKinter en Python pour créer une zone de saisie de texte pour les utilisateurs et une zone de visualisation des réponses du chatbot. Cette intégration facilite l'émission des demandes par les utilisateurs et la consultation des réponses fournies par le chatbot.

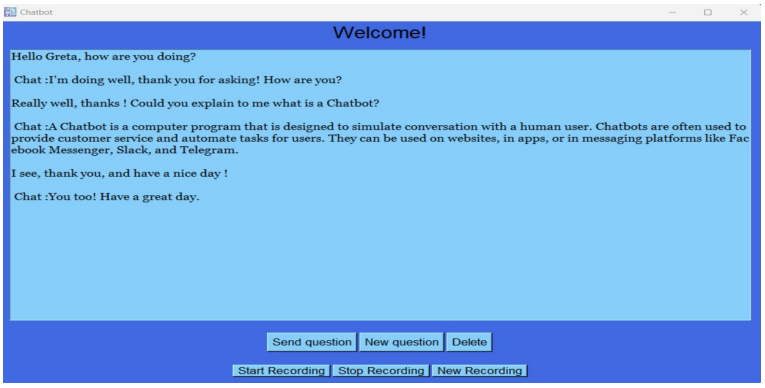

FIGURE 2 – Interface de chatbot : une vision globale.

3.2 Reconnaissance Vocale

Nous avons intégré une fonction de reconnaissance vocale pour permettre aux utilisateurs de soumettre leurs demandes par la voix. Cette fonctionnalité permet une communication plus naturelle avec le chatbot, en lui parlant directement. Nous avons utilisé la reconnaissance vocale de Google Speech Recognition pour cette implémentation.

3.3 Intégration sur Greta

FIGURE 3 – Greta : l'agent virtuel en action.

Pour intégrer notre interface de chatGPT à Greta, nous avions deux possibilités : soit créer un module sur Greta qui puisse communiquer avec notre code Python, soit directement modifier les fichiers lisibles par Greta pour lui faire dire ce que nous voulions. Étant donné que les modules sont écrits en Java 8 et que personne dans l'équipe n'avait les compétences nécessaires pour reprogrammer ces codes, nous avons choisi la seconde option. Greta est capable de lire des fichiers FML et BML, qui sont des fichiers XML qui lui permettent de lire un message et d'agir en conséquence. Pour cela, elle utilise les modules "FML/BML reader". Notre code Python relie les deux systèmes en modifiant un fichier XML directement lié à Greta et en y ajoutant les réponses de ChatGPT à nos questions sur l'interface. Ainsi, lorsque le chatbot nous répond, Greta reprend la réponse de manière appropriée.

3.4 Résultats

Greta est désormais apte à interagir avec les utilisateurs, comme l'indique ce graphique :

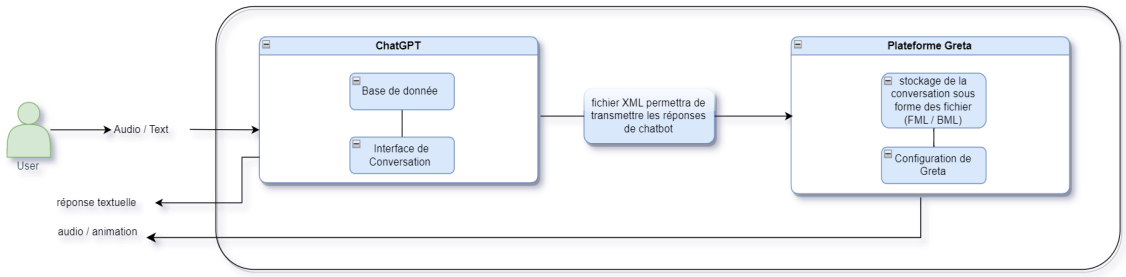

FIGURE 4 – Schéma de l'intégration de ChatGPT avec Greta : une mise en relation efficace.

Actuellement, Greta ne fait que reproduire les réponses de ChatGPT ainsi que les comportements non verbaux associés à ces réponses, telles que les expressions faciales et les mouvements de la bouche. Elle accomplit cela en examinant un fichier FML, où la réponse de GPT est enregistrée.

Pour être capable de réaliser des mouvements des mains en parlant, Greta a besoin de fichiers BML, où les descriptions de ces mouvements sont indiquées.

Dans notre étude sur Greta, nous avons découvert des modules "Meaning Miner" capables de comprendre les métaphores dans le texte d'entrée pour générer des mouvements aléatoires. Malheureusement, ces mouvements ne fonctionnaient que pour un groupe de mots très restreint, ce qui rendait rapidement le module inefficace.

3.5 Améliorations Possibles

Il existe plusieurs idées d'amélioration pour notre modèle de conversation comme la création d'un nouveau module pour générer des fichiers BML avec des mouvements aléatoires (Comme Meaning Miner), l'ajout d'un système automatisé pour lancer la reproduction de la réponse de ChatGPT par Greta, l'intégration de données supplémentaires pour améliorer la compréhension du système, l'ajout de fonctionnalités de sécurité, la reconnaissance faciale pour une interaction plus personnelle et la reconnaissance visuelle pour communiquer avec le chatbot en utilisant une image.

On a réussi à créer un module en mettant en place toutes les dépendances nécessaires au bon fonctionnement d'un tel module sur NetBeans [2], mais il nous reste encore à programmer le code java, et à construire une bibliothèque de mots/poses qui y fassent référence. Ceci serait une bonne façon de faire un lien avec l'autre sous-groupe de projet.

3.6 Limites

Les chatbots conversationnels ont des limites dans la compréhension des questions subtiles et des nuances émotionnelles, ainsi que dans la personnalisation des réponses. Les avancées en intelligence artificielle promettent des améliorations, mais nécessitent une maintenance continue et une surveillance humaine pour maximiser leur performance.

2. NetBeans est un environnement de développement intégré (IDE) pour le développement Java. Il s'agit d'un logiciel libre et open-source qui fournit une grande variété d'outils pour écrire, déboguer et tester du code.

4 2D-3D Conversion

Dans le cadre de la génération du mouvement de l'agent virtuel, la pose permet d'estimer la position des jointures de l'agent. Une méthode dejà existante permet de générer la posture en 2D, en utilisant le modèle cinématique inverse, à partir des textes, Néanmoins, la nature purement procédurale rend les mouvements de Greta peu naturels et sont en 2D. C'est dans le but d'améliorer cette partie que s'intègre notre tâche.

Notre but, pour cette partie du projet, était donc d'améliorer le mouvement des agents de Greta en assurant une meilleure modélisation en 2D et aussi passer à une modélisation en 3D pour obtenir une résultat encore plus réaliste. Pour ce faire, notre stratégie est de commencer par extraire les poses en 2D et 3D en appliquant un modèle sur une base de donnée adaptée à notre projet. Puis utiliser un deuxième modèle capable de calculer la pose 3D à partir des poses 2D.

4.1 Dataset

Le choix de la base d'entraînement est lié au choix des mouvements que l'on souhaite apprendre à l'agent virtuel. Le modèle permettant de générer la pose à partir du text, a été entraîné sur la base de données Pose, Audio, Transcript, Style (PATS) qui est constituée de séquences vidéos d'orateurs. Cette base de données est utilisée comme une référence pour les agents virtuels pour la génération de mouvement naturel correspondant à un discours.

4.2 Modèle d'Extraction de Pose

Nous avons étudié des différentes méthodes pour extraire la pose depuis une image. Nous avons essayé de travailler avec la bibliothèque Openpose qui était référencée par la base d'entraînement PATS mais celle ci n'arrivait pas à extraire tous les joints. Nous avons donc cherché d'autres méthodes telle que MMPose, qui est une toolbox basée sur pyTorch et Mediapipe, cette dernière donnait les meilleurs résultats pour l'extraction des poses en 2D et 3D. La limitation de cette méthode est l'impossibilité d'exécuter l'extraction de poses sur GPU, celle ci prend donc du temps ce qui nous a obligé à seulement prendre une partie de la base de données.

4.3 Modèle de conversion de 2D en 3D

Après nous être documentés sur les articles qui traitent de l'estimation de pose 3D, nous avons trouvé que la plupart des modèles existants utilisent directement des images en entrée. Or, dans notre cas, nous souhaitons calculer la pose 3D à partir de la pose 2D. Pour ce faire, nous avons réalisé l'apprentissage en utilisant comme entrée les poses 2D extraites précédemment et comme sorties désirées, les poses 3D correspondantes. Pour réaliser cette conversion, nous avons implémenté l'architecture présentée dans l'article [4] et l'avons adapté à nos données d'entrée et de sortie.

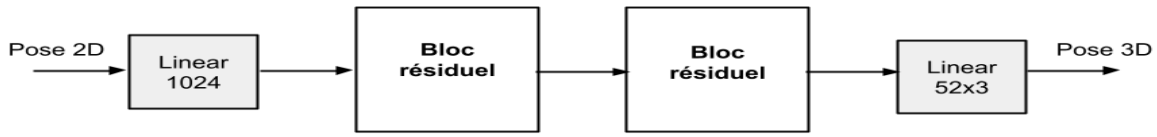

FIGURE 5 – Architecture utilisé pour la passage de 2D-3D

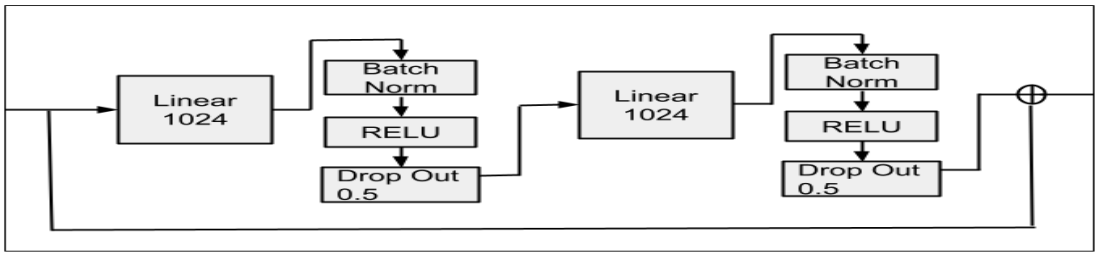

FIGURE 6 – Architecture du Bloc Résiduel

Dans le but d'améliorer les résultats obtenus par ce modèle, nous avons également implémenté un réseau de l'article [5] qui est constitué de couches de Graph convolution et structuré en blocs résiduels.

4.4 Résultats

En utilisant le premier modèle de conversion, nous obtenons une erreur de 0.12 (mean absolute error), on a testé le modèle sur de nouvelles séquences avec d'autres orateurs ainsi qu'en temps réel en utilisant une camera. un exemple des résultats obtenus est illustré ci-dessous :

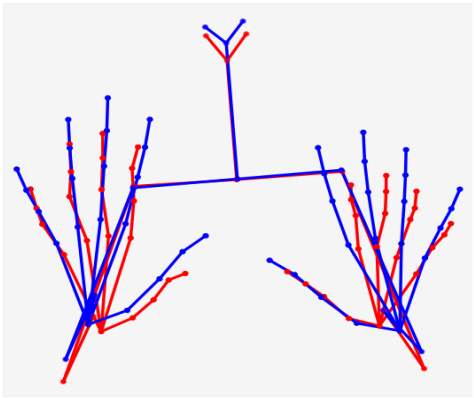

FIGURE 7 – La comparaison entre la pose estimée par notre modèle et la sortie désirée. La pose en bleu représente le résultat trouvé utilisant Mediapipe. Cela a été utilisé comme ground truth. La pose en rouge est le résultat que nous avons obtenu avec notre modèle.

Concernant le deuxième modèle, les résultats étaient similaires au premier. Une solution pour améliorer l'entraînement et les performance de nos modèles seraient d'augmenter le nombre de données soit en faisant plus d'extraction ou en utilisant des méthodes d'augmentation de données (translation des images...)

4.5 Intégration sur Greta

Pour la description bas niveau des mouvements, Greta utilise un fichier de type BAP (Body Animation Parameters) qui définie les angles de rotation pour de différent articulations. Les fichiers BAP contiennent, au totale, les paramètres de 196 articulations [2]. Chaque articulation représentée dans des baps prennent des différentes valeurs correspondent aux changements d'amplitude des mouvements sur différents axes. Malheureusement sur Greta, ces valeurs sont affectées avec la méthode "essaie- erreur". Donc il n'existe pas un coefficient que nous pouvons utiliser pour intégrer nos résultats de manière automatique en utilisant des fichiers d'extension bap.

Les fichiers bap sont générés automatiquement dans un module de Greta à partir des fichiers .bvh. Pour intégrer nos résultats dans Greta, nous avons transformé nos poses en fichier .bvh en utilisant le github

du doctorant qui a travaillé sur Greta l'année dernière [3]. Nous avons adapté ces codes à nos résultats, puis nous avons modifié les fichiers .bvh que nous avons obtenus pour qu'ils puissent être adaptés à Greta. Malheureusement, les fichiers que nous avons obtenus en suivant le github que nous avons mentionné ci-dessus n'étaient pas exploitables par Greta. Résoudre ce problème nécessite une compréhension très approfondie de la fonctionnement de mouvements Greta. La manque de documentation et la nature chronophage de compréhension des modules de Greta a rendu cette adaptation impossible pendant la durée du projet.

5 Discussion

Dans le cadre de ce projet, nous avons essayé d'améliorer certains aspects de l'agent virtuel Greta. Ce projet nous a appris à prendre en main un projet en cours de route et essayer de l'améliorer, ce qui était une très bonne occasion de nous entraîner pour le milieu de travail. Malgré notre semestre chargé, nous avons réussi à suivre l'emploi de temps que nous avions prévu sur notre diagramme de Gantt. Nous avons réalisé presque toutes les tâches que nous avions dans le cadre de ce projet. La partie intégration de nos travaux sur Greta a été peut-être encore plus difficile que de réaliser nos taches car nous n'étions pas familiarisés avec le codage de ce genre de plateformes. Cela nous a appris que le fait notre adaptation aux nouvelles plateformes et projets est aussi important que de développer nos propres algorithmes.

Nous remercions sincèrement Madame Catherine Pelachaud et Monsieur Nicolas Obin d'avoir été constamment présent et à l'écoute lorsque nous avons eu des soucis de compréhension de nos objectives. Nous remercions également Michele Grimaldi de nous avoir aider à mieux comprendre le code de Greta ainsi que ses conseils concernant des multiples solutions que nous avons abordées.

Références

[1] modèle conversational. https://github.com/3803531/Chatbot.

[2] The character animation company. Mpeg-4 face and body animation (mpeg-4 fba) an overview. https://visagetechnologies.com/uploads/2012/08/MPEG-4FBAOverview.pdf.

[3] Michele Grimaldi. Pfe-openpose-to-vae-to-bvh. https://github.com/Michele1996/PFE-OpenPose-to-VAE-to-BVH.

[4] Javier Romero James J. Little Julieta Martinez, Rayat Hossain. A simple yet effective baseline for 3d human pose estimation. https://arxiv.org/pdf/1705.03098.pdf.

[5] Yu Tian Mubbasir Kapadia Dimitris N. Metaxas Long Zhao, Xi Peng. Semantic graph convolutional networks for 3d human pose regression. https://arxiv.org/abs/1904.03345.

[6] OpenAI. Gpt-3 modèles. 2021. https://beta.openai.com/docs/models/gpt-3.