

# Projet TAL - Chatbot

Vincent Doisneau, Florian Le Pallec, Oummar Mayaki

May 5, 2018



Polytech Paris Sud

Maison de l'Ingénieur - bât 620 - Centre Scientifique d'Orsay - 91405 Orsay - France

Tel : +33 (0)1 69 33 86 00 - fax : +33 (0)1 69 41 99 58 - [www.polytech.u-psud.fr](http://www.polytech.u-psud.fr)

## Description

Ce document correspond au rapport du **projet de Chatbot** réalisé pour le cours d' "**Introduction au TAL**" ayant lieu durant le second semestre de la 4ème année Informatique à l'école d'Ingénieur **Polytech Paris-Sud**. Il fut réalisé durant l'année 2017-2018 par le trinôme d'étudiants suivant :

- DOISNEAU Vincent
- LE PALLEC Florian
- MAYAKI Oummar

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Contexte . . . . .	1
1.2	Objectifs . . . . .	1
1.3	Choix du sujet . . . . .	1
1.4	Outils de développement utilisés . . . . .	1
<b>2</b>	<b>Réalisation du projet</b>	<b>1</b>
2.1	Organisation et hiérarchie . . . . .	1
2.2	Priorités de développement . . . . .	2
2.3	Répartition des objectifs . . . . .	2
<b>3</b>	<b>Instructions du programme</b>	<b>3</b>
3.1	Lancement du programme . . . . .	3
3.2	Choix des modes . . . . .	3
3.3	Mode 1 - Backchannels . . . . .	3
3.4	Mode 2 - Basic mode . . . . .	3
3.5	Mode 3 - Advanced mode, Devinette animaux . . . . .	3
<b>4</b>	<b>Avancement et possibilités d'amélioration</b>	<b>4</b>
4.1	Etat actuel du projet . . . . .	4
4.2	Limitations . . . . .	5
4.3	Possibilités d'améliorations et d'expansions . . . . .	6

# 1 Introduction

## 1.1 Contexte

Le but de ce projet est de créer un chatbot (soit en français, soit en anglais) et présentera **3 modes distincts** selon le choix de l'utilisateur.

Dans le **1er mode**, il sera demandé au bot de répondre par des “backchannels” aux entrées de l'utilisateur, c'est à dire des réponses génériques montrant qu'il a compris et/ou entendu.

Pour le **2ème mode**, le bot devra adopter un comportement semblable au chatbot psychologue Eliza, il sera capable de reprendre l'entrée de l'utilisateur : “Je suis X” -> “Pourquoi est ce que tu es X”. Certains mots clés seront aussi identifiés et leur utilisation donnera lieu à une question liée à ce sujet.

Enfin pour le **3ème mode**, il s'agira cette fois ci d'un mode “libre” dont la réalisation sera laissée au choix de chacun des groupes pour aller au-delà de ce qui a pu être proposé dans le mode 2.

-> **Pour plus de précisions quant aux instructions et demandes liées à ce projet, veuillez vous référer au document “projet.pdf” disponible sur le répertoire Dokeos dédié à la matière.**

## 1.2 Objectifs

Les enjeux liés à ce projet sont multiples. D'une part, nous avons tous les “**enjeux techniques**” liés au chatbot, c'est à dire, de pouvoir créer une situation de dialogue entre un système automatique et un utilisateur humain qui doit être suffisamment crédible et répondre aux attentes de l'utilisateur. L'implémentation doit nous permettre de refléter nos acquis du cours d' “Introduction au TAL” et devra se concentrer sur les aspects d'analyse, Les points s'éloignant de cet objectif, comme les interfaces graphiques, seront considérés superflus.

D'autre part, nous avons ce que nous pourrions considérer comme les “**enjeux organisationnels**”, liés directement à la réalisation, l'organisation et la présentation des résultats de ce projet. Il s'agira cette fois ci de montrer que nous sommes capables de mener un projet à son terme, de façon méthodique mais surtout que nous pouvons mettre en valeur notre travail, les résultats et l'avancée actuelle dans un format structuré, ici LaTeX, pour permettre à tous et chacun d'en comprendre les notions évoquées.

## 1.3 Choix du sujet

Ce projet présentant un mode 3 dit **mode “libre”**, nous avons donc eu la possibilité de choisir un sujet qui serait capable d'aller au-delà du mode 2. Pour cela, nous avons décidé de modéliser un jeu concernant les animaux qui se réaliserait en **2 parties distinctes**, le premier mode qui permettrait au bot de deviner l'animal que le joueur a en tête et le second mode où il s'agit cette fois ci du joueur qui devra deviner l'animal choisi par le bot. Le scénario du jeu sera fait sous la forme d'un jeu télévisé pour donner plus d'immersion au participant : on ne peut pas faire oublier le fait qu'il s'agisse d'un robot à notre niveau mais nous pouvons tout de même lui fournir une histoire.

Par ailleurs, il est à noter que nous avons choisi d'implémenter notre chatbot en Anglais.

## 1.4 Outils de développement utilisés

La réalisation du chatbot s'est fait au moyen du Langage de programmation **Python**, en version 3.5. Le langage ne demandant pas de spécifications particulières, nous avons pu écrire notre programme sur nos éditeurs personnels (Atom, SublimeText...) et les tester au moyen du terminal.

Le point auquel nous avons cependant accordé de l'importance est l'utilisation d'un répertoire Git pour synchroniser rapidement et facilement notre avancement tout en isolant les modifications actuelles par l'utilisation de branches. Notre **répertoire Git** est disponible sur GitHub à l'adresse: [https://github.com/Vinnvi/TAL\\_Project](https://github.com/Vinnvi/TAL_Project)

# 2 Réalisation du projet

## 2.1 Organisation et hiérarchie

Étant un trinôme, il a fallu nous organiser de façon méthodique pour nous permettre de maximiser nos efforts et notre temps. Pour cela, nous sommes passés par une **phase de réflexion** durant la première séance qui fut dédiée uniquement à la prévision et la conception de notre projet. Cela nous a permis d'identifier les “milestones”, choisir un sujet pour le mode libre qui convienne à chacun et aussi de répartir le travail de façon efficace pour la suite. Suite à cela, il a été décidé que Vincent s'occuperait d'implémenter le mode 1 et 2 tandis que Florian et Oummar commenceront le mode 3. *(rejoint par la suite par Vincent)*

Au niveau de la **hiérarchie**, nous avons préféré adopter une **forme souple et collaborative**, il n'y a donc pas de chef de projet comme ce rôle n'était pas particulièrement nécessaire pour nos objectifs. Nous

restons cependant en communication constante sur une discussion Messenger et la rédaction du rapport s'est faite par des éditeurs collaboratifs. (*Google Doc puis Overleaf pour garder le format LaTeX*)

## 2.2 Priorités de développement

Chaque projet présentant une quantité de travail importante, il nous a fallu pour celui-ci comme pour d'autres identifier les points importants sur lesquels nous devons travailler en priorité, les **points "critiques"**, ainsi que les sujets d'améliorations qui pourraient quant à eux venir par la suite.

Les points sur lesquels nous avons décidé d'accorder davantage d'importance sont :

- Le déroulement sans accroc des **2 modes de jeu**, le bot est capable de deviner l'animal du joueur et vice versa sans l'apparition de problèmes techniques ou de lourdeurs dans le déroulement de la conversation.
- L'étendue raisonnable de la **compréhension des réponses** de la part de notre bot, notamment l'identification des tournures négatives ou positives dans une réponse ou encore des mots clés ou des informations nécessaires pour le déroulement du jeu.
- La **base de données** des animaux, qui doit être suffisamment fournie et présentant des informations facilement identifiables par les joueurs.

À contrario, les points que nous avons considérés moins "critiques" et pouvant être relégués après la réalisation des objectifs cités précédemment :

- **L'environnement** autour du bot, notamment **l'histoire** que l'on accorde à notre scénario pour donner une expérience plus immersive.
- Une **analyse plus complexe** des **réponses** pour permettre au bot de réagir au-delà de simplement identifier des mots clés ou des tons positifs ou négatifs.
- Le **comportement du bot**, son éventail de réponses et de remarques, ainsi que tout outil qui pourrait lui permettre de paraître plus naturel, plus "humain" et ainsi faire oublier l'aspect machine au joueur.

## 2.3 Répartition des objectifs

Une fois les priorités identifiées, il fut nécessaire de s'organiser pour pouvoir suivre le travail de chacun et ne pas se marcher dessus. Si cela peut se faire en communiquant sur nos conversations dédiées ou par le biais du répertoire git, cela se révèle assez pénible.

TASK	STATE	PERSON
Complete database animals		Florian
Handle Yes/No/Idk replies		Oummar
Complete Mode 1		Vincent
Complete Mode 2		Vincent
Guess animal (bot's side)		Florian
Guess animal (player's side)		Vincent
Improve rows of queries from bot to ask questions		Florian
Handle guessing when no further questions can be asked		Florian
Improve guessing to also take informations from player		
Possibility of restart game/change mode at the end		Florian
Handle cases like "Sometimes" for colors		
Update queries with name to make it personal		
Improve smalltalks		
Handle questions from us		Vincent
Implement mode player guessing		
Improve backstory of bot/game		
Implement explanation/help		Florian
Display list animals		Florian
Possible implementations (afterwards)		
Connect to a database of informations		
Offer further explanations on animals thanks to database		

Figure 1: Extrait du Google Sheet durant le projet

Nous avons donc cherché à faire cela par un moyen plus efficace et avons opté pour un fichier Google Sheet capable de garder une trace des objectifs du projet, à quel état se trouvent-ils et qui en est responsable.

Notre liste s'est développée au fur et à mesure que nous identifions des points à réaliser, un système de couleur était ensuite utilisé : orange "en cours", jaune "réalisé mais améliorations possible" et vert "objectif atteint". Une liste d'améliorations supplémentaires (dans la continuité du projet) a aussi été rajoutée à la fin de cette liste tandis que nous commençons à visualiser comment ce programme pourrait évoluer par la suite.

## 3 Instructions du programme

### 3.1 Lancement du programme

Pour le lancement de l'application, il suffit de compiler le programme au moyen de la ligne de commande ci dessous :

```
- python main.py
```

### 3.2 Choix des modes

Pour choisir le mode, il suffit comme cela a été demandé dans le sujet, d'ajouter un nombre entre 1 et 3 inclus à la suite de la commande ci-dessus, ce qui donne une commande de la forme : `python main.py 3` On empêche la possibilité de donner des nombres inférieurs à 1 et supérieurs à 3, grâce au paramètre "choices=range(1, 4)" du parser. Suite à cela, le programme nous envoie dans un des 3 modes disponibles, en lançant la fonction associée.

### 3.3 Mode 1 - Backchannels

Ce mode simple consiste à répondre par des "well..." "hmmm..." etc... en évitant de répéter ce qui a été dit la fois précédente. Pour faire ce mode nous avons une boucle principale, dans laquelle nous prenons en entrée l'input de l'utilisateur (*nous n'en faisons rien par la suite*), on lance ensuite une fonction "answer" : elle sélectionne au hasard une réponse parmi celles disponibles (*rangées dans une liste*) et la renvoie. Pour éviter de répéter plusieurs fois la même réponse, on stocke dans une variable la dernière réponse, qu'on envoie dans la fonction answer.

### 3.4 Mode 2 - Basic mode

Pour ce mode on se base sur le même principe que le mode précédent. On prend dans une boucle principale, l'entrée de l'utilisateur. Puis on va analyser cette suite de mots : pour chaque mot, on regarde s'il fait partie des mots-clés d'une des catégories que l'on a à notre disposition (nous en avons mis 5, avec plus d'une dizaine de mots-clés pour chacune). Si l'on détecte qu'il y a un mot-clé de cette catégorie, alors on choisit une des 2 réponses possibles pour cette catégorie, en évitant de prendre la dernière utilisée, comme cela a été fait dans le précédent mode, puis on renvoie cette réponse. Sinon, on rappelle la fonction answer du premier mode, pour renvoyer un backchannel. On détecte aussi les cas "I'm/I was ..." : on cherche d'abord, s'il y a un "I'm" ou "I" puis on regarde s'il y a "was", "am" ou "will be" après (dans le cas du I) et on renvoie une réponse ensuite avec la suite de la phrase. Nous avons voulu rendre cette phrase plus naturelle en ajustant la réponse : par exemple si l'utilisateur écrit : "I'm your father" le bot va renvoyer "Why are you my father ?" et non "Why are you your father ?"

### 3.5 Mode 3 - Advanced mode, Devinette animaux

Le choix du mode 3 entrainera le lancement du chatbot de devinettes d'animaux sous la forme d'un jeu télévisé. Il sera demandé à l'utilisateur d'entrer son nom, après quoi le bot lui proposera un rappel des règles du jeu. Ensuite l'utilisateur décide s'il veut choisir l'un des deux modes de jeux ou laisser le hasard décider à sa place.

Les règles du jeu changent selon le mode sélectionné :

"Mode deviner" : Dans ce mode, on part du principe que l'utilisateur a un animal en tête, et le bot devra deviner de quel animal il s'agit grâce à une série de questions. Celles-ci porteront sur certains attributs de l'animal tels que son environnement de vie, sa classe, sa couleur, etc. L'utilisateur répond à ces questions par des réponses positives ou négatives.

Une grande importance a été attribuée à l'interprétation des réponses de l'utilisateur aux questions du bot :

- En effet, nous avons pris en compte le fait qu'il existe de nombreux moyens de répondre affirmativement/négativement à des questions de ce type et avons pris le soin de faire en sorte que la majorité de ces moyens soient interprétables pour le bot. Ainsi, il est possible au bot d'interpréter des réponses de complexités et de registres différents.

*Quelques exemples : yeah/nah, of course (not), I (don't) think so, without a doubt, why not*

- Le bot est aussi capable de réagir lorsqu'il n'arrive pas à interpréter la réponse de l'utilisateur. Il y a 3 grands cas de figure auxquels le bot sait réagir :
  - La réponse de l'utilisateur n'a pas d'éléments permettant au bot de déterminer si sa réponse est positive ou négative
  - L'utilisateur répond à une question du bot par une autre question, ce qui est malpoli !
  - L'utilisateur ne répond pas (input vide)

Dans chacun de ces cas, le bot réagit de manière différente mais toujours de façon à dire que la réponse n'était pas interprétable, puis il est demandé à nouveau à l'utilisateur de saisir une réponse. Dans l'optique de rendre l'échange plus vivant, les réactions du bot dans les cas précédemment cités ont été désignées pour être de plus en plus fortes, afin de simuler une colère grandissante. Au bout d'un certain temps, il va jusqu'à "perdre patience" et mettre fin au jeu.

Mode "faire deviner" : Dans ce mode, le bot a en tête le nom d'un animal, et l'utilisateur doit deviner à quel animal pense le bot, à travers une série de questions. Ces questions, auxquelles le bot répondra oui ou non (mais de façon plus naturelle qu'un simple "yes/no"), porteront donc sur des caractéristiques des animaux (type, taille, présence ou non de critères tels que : ailes, poils, etc. ). Pour faire cette partie du mode 3, on a construit au fur et à mesure le programme :

- Dans un premier temps, nous avons mis en place la structure du programme, à savoir l'initialisation, dans laquelle l'ordinateur choisit aléatoirement un animal à partir de la base de données (et sa classe associée) et les instructions pour le joueur. Puis, une boucle infinie dans laquelle chaque tour de boucle correspond à un tour de jeu. Première étape : l'utilisateur donne sa question, deuxième étape, on analyse sa phrase, troisième étape : on renvoie le résultat associé à celui de l'analyse.
- Nous avons ensuite commencé à mettre en place le système d'analyse, ce système requiert de la part de l'utilisateur, pour le moment, qu'il ne s'y trouve pas de fautes d'orthographe : on analyse le premier mot de l'utilisateur, et à partir de ce premier mot et des suivants, on va commencer à déduire sa question : par exemple si la phrase commence par un sujet, on peut ensuite s'attendre à un verbe d'action (swim, live, ...) ou bien le verbe "être", qui peut alors déboucher sur la question 'he is tall ?' par exemple ou alors le verbe "avoir", qui peut lui déboucher sur la question 'he has wings ?' En reconstituant ainsi au fur et à mesure les possibilités de questions on va pouvoir répondre à de plus en plus de questions. Bien évidemment, nous ne pourrons pas répondre, à la fin du projet, à toutes les questions, car les possibilités sont très nombreuses voire quasi-infinies, d'autant plus que l'on peut formuler les questions de plusieurs manières à l'écrit : he has wings ? / Does he have wings ? he is small / is he small ? etc. Néanmoins, nous nous efforçons de proposer le maximum de questions qui sont pertinentes et qui sont les plus communes (taille, lieu de vie, couleur, ...). Nous mettrons avec ce rapport l'ensemble des questions auxquelles le bot peut répondre, pour que cela soit plus facile à essayer.

## 4 Avancement et possibilités d'amélioration

### 4.1 Etat actuel du projet

- **Un jeu fonctionnel**

À la date actuelle, notre programme assure les objectifs principaux que nous avons choisi de fixer, le jeu se lance, introduit la storyline entourant le bot et permet de choisir le mode auquel on veut jouer pour ensuite deviner l'animal du chatbot, ou au contraire que le bot essaye de deviner notre animal. Ces modes fonctionnent de manière correcte, il existe des points que nous aurions pu affiner mais les parties remplissent leurs rôles.

- **Un chatbot mais pas seulement**

Une attention particulière s'est portée sur l'environnement de la storyline et les dialogues proposés, tout cela dans le but d'offrir une expérience plus naturelle, une conversation plus immersive, ces détails aidant à mettre en place une atmosphère pour le joueur. Cela se fait notamment par le scénario du jeu télévisé "Wild Guess", l'utilisation de dialogues interpellant le joueur, l'utilisation de son prénom, divers types de réponses pour une seule situation...

Pour donner des exemples pratiques, le bot va notamment demander le nom du joueur au début



de la partie, et va ensuite ajouter le nom à certains dialogues pour rajouter une touche personnelle durant la conversation. Les questions posées au joueur et les déclarations durant la partie sont tirées au hasard parmi plusieurs propositions

- **Des améliorations sur certains points obligatoires**

Nous avons tenté d'ajouter quelques touches sur certains points pour essayer généralement de rendre le bot plus réaliste ou le système plus précis. Nous avons par exemple, pour la partie "I'm x" du mode 2, essayé de redonner une réponse cohérente vis à vis de la question ainsi que donner la possibilité de mettre la structure "I'm x" n'importe où dans une phrase et pas seulement au début. Par exemple on peut dire "Oh sorry, I'm so tired" ce qui donnera lieu à la réponse "Why are you tired ?". Nous avons aussi ajouté un début de vérification d'erreur dans le mode 3, pour indiquer les fautes quand le joueur donne une question

- **Structure de programme réfléchi**

Nous avons conçu le programme de manière à pouvoir l'améliorer rapidement et façon efficace, tout d'abord en séparant au maximum ce qui peut être séparé : les différents modes sont dans des fichiers différents, la base de données des animaux aussi, etc. Puis nous avons fait en sorte que les parties de programmes qui peuvent toujours être améliorées dans le temps soit faites de façon modulaire pour être le moins rigide possible et ainsi permettre à de futurs développeurs de pouvoir rapidement reprendre le travail, ou alors ajouter facilement du contenu si l'on souhaite. Par exemple dans la partie compréhension du texte par le bot, ou le joueur doit deviner l'animal, nous avons structuré le programme de manière à ce que les développeurs puissent l'améliorer c'est à dire lui permettre de comprendre plus de questions, sans avoir à toucher au reste.

- **Le souci de performance**

Réaliser des chatbots peut très vite devenir complexe au niveau de l'analyse et des calculs, d'autant plus que ce qui fonctionne d'aujourd'hui peut devenir la chose de trop une fois que des améliorations/expansions ont été rajoutées. Pour cela, nous avons toujours essayé de choisir des méthodes et des algorithmes qui restent relativement peu coûteux. La structure de la base de données qui ne garde ainsi que certaines informations populaires auprès de la population, ou l'utilisation des "dictionary" qui permet très rapidement de limiter les possibilités (*et donc les calculs*) sont représentatifs de cette optique. Cela s'est aussi fait par des petits détails comme l'utilisation de la fonction "join" lorsque l'on veut concaténer plus de deux string à la fois car en effet, dans ce cas là, ses performances sont meilleures que la concatenation par l'opérateur "+" de Python.

## 4.2 Limitations

Si les fonctionnalités principales sont assurées, quelques problèmes restent cependant à régler pour atteindre une utilisation fiable.

- **Relance du jeu en construction**

Nous donnons au joueur la possibilité de recommencer la partie avec un mode ou un autre, néanmoins il existe des bugs au redémarrage de la partie avec notamment des listes d'animaux raccourcis et des valeurs qui ne sont pas correctement réinitialisées. Ces problèmes provoquent donc un mauvais fonctionnement du programme qui peut aller jusqu'à son arrêt complet. Néanmoins cela ne suggère pas la présence de problèmes considérables, un peu de temps supplémentaire aurait permis d'isoler le problème.

- **Une base de données à compléter**

Nous avons choisi d'écrire nous même la base de données sur les animaux, cela nous permettait d'une part d'avoir un plus grand contrôle sur le format mais aussi de ne retenir que les informations que nous jugeons utiles pour le déroulement du jeu. Nous avons essayé de mettre les animaux les plus populaires mais il en reste encore beaucoup pour véritablement compléter cette base. Le problème apparaît lorsque l'utilisateur choisit un animal en dehors de cette base, le bot ne sera donc pas capable de le trouver puis qu'il n'y est pas présent. Le point positif est que ce problème n'est pas complexe et demande simplement du temps pour trouver et remplir les données.

- **Une compréhension du langage limitée**

Nous avons donc un mode dans lequel le joueur pose des questions et le bot répond, ce qui nécessite une compréhension des questions. Nous n'avons pas opté pour une méthode dans laquelle nous analysons des mots-clés, mais on regarde d'abord quel est le premier mot, à quelle classe grammaticale il appartient, puis en fonction on regarde le deuxième, puis le troisième etc. Cela est long et fastidieux car il y a beaucoup de types de mots possibles, même si on ne reste que dans des questions, et l'ordre des mots peut être multiple. C'est pourquoi la compréhension des questions de notre bot reste très limitée. Mais l'architecture mise en place permet cependant une amélioration très simple et rapide du programme et n'est pas rigide.

### 4.3 Possibilités d'améliorations et d'expansions

Bien que nous ayons essayé de remplir tous les objectifs que nous avons identifiés, il reste toujours des **éléments à perfectionner** et implémenter dans un projet, nous allons donc évoquer ici les points auxquels nous souhaiterions revenir si nous devions affiner notre rendu :

- **Améliorer la partie "deviner"**

Nous avons attribué beaucoup d'importance à l'interprétation des réponses, mais des améliorations dans ce domaine sont encore possibles et notre programme comporte encore quelques failles. En effet, notre bot est capable de réagir à l'éventualité où les critères de l'animal choisi par l'utilisateur ne correspondent à rien qui soit présent dans sa base de données. Et pourtant, il ne sait pas encore réagir si l'utilisateur ne fait que répondre "non" à toutes les questions, très certainement à une condition mal placée dans notre code.

Une autre amélioration serait de traiter les cas exprimés par l'exemple suivant :

*Bot : "Your animal... Is it small?"*

*User : "pretty small"*

Dans un cas comme celui-ci, le bot ne saura pas interpréter la réponse. Cela pourrait facilement être corrigé en ajoutant aux mots interprétables comme des mots positifs le mot correspondant au critère demandé (ici "small")

- **Améliorer la partie "faire deviner"**

Nous pouvons toujours améliorer ce que peut comprendre le bot, quand l'utilisateur pose des questions pour deviner un animal. Actuellement, nous avons une base, qui permet de poser les questions les plus pertinentes, mais nous pourrions améliorer tout cela, en ajoutant les phrases négatives, en permettant de comprendre plus de types de questions et de tournures de phrases, et éventuellement de comprendre malgré les fautes d'orthographe (*ce que le bot peut faire mais pour certaines erreurs simples, comme le s à la troisième personne du singulier*). On peut retrouver l'ensemble des questions disponibles actuellement dans la partie wiki sur la page Github du projet.

- **Augmenter l'interaction**

Si notre bot permet de suivre une histoire et une conversation, cette conversation tend justement à être très (*trop*) scénarisée et revient finalement à suivre un chemin dicté au joueur. Ce chemin repose sur beaucoup de questions attendant une réponse positive ou négative, ce qui est assez dommage pour l'intérêt de la discussion. Nous aurions donc voulu jouer d'avantage sur l'interaction, peut être par le biais d'un autre mode de jeu pour permettre finalement au joueur et au bot d'entamer une conversation plus active. Même si ce point nous éloigne du sujet que nous avons choisi, cela aurait représenté un point très intéressant pour rester davantage sur le traitement de langage.

- **Diversifier les données sur les animaux**

Nous avons pour le moment une base simple d'animaux et, comme expliqué précédemment, nous pourrions augmenter la variété proposée mais cela pourrait aussi se jouer sur d'autres détails, notamment les caractéristiques qu'ils possèdent pour donner plus de choix à la réflexion du joueur. Par exemple : sur quels continents vivent-ils principalement, sont-ils diurnes, sauvages ou domestiques, en voie d'extinction, ...

Par ailleurs, il existe aussi des éléments qui ne faisaient pas partie de nos objectifs et ne représentent pas des parties essentielles de l'expérience utilisateur mais qui constituent tout de même de **potentielles expansions** à notre chatbot, nous allons donc évoquer ici ces points :

- **En faire un jeu éducatif**

S'il est amusant de deviner les animaux, il serait aussi intéressant d'aborder l'aspect éducatif que l'on pourrait implémenter, notamment en proposant des informations supplémentaires sur les animaux évoqués, cela pourrait notamment se faire en reliant le programme à une base de données en ligne qui contiendrait des informations sur les animaux. Cet objectif de relier une base de données extérieure et incorporer un autre aspect à notre chatbot pourrait permettre d'offrir une facette différente à l'expérience utilisateur.

- **Interagir autrement que par du texte**

Si la communication avec le chatbot se fait par du texte, on pourrait imaginer utiliser d'autres biais. L'utilisation de commandes vocales est aujourd'hui beaucoup plus simplifiée et se fait dans nombre de domaines, le bot pourrait ainsi durant la partie produire quelques interactions vocales (*sans forcément attendre de réponses*) pour à la fois surprendre et immerger le joueur. Ce canal serait aussi être utile pour d'autres objectifs, on pourrait imaginer que le bot soit capable de produire les sons de certains animaux pour donner des indices ou simplement dans un but éducatif.

- **Proposer d'autres modes de jeu**

Seuls 2 scénarios sont actuellement proposés mais on peut imaginer ajouter d'autres modes de jeu



par la suite avec par exemple un mode à 3 joueurs : le bot fait deviner le même animal à 2 joueurs en même temps ou bien chacun à leur tour et le but est de deviner l'animal en premier. On peut aussi ne pas limiter ce mode à seulement 2 joueurs, mais éventuellement encore plus et ainsi permettre à une famille de jouer ensemble si on veut permettre à des enfants de jouer avec leurs parents.

- **Passer à la reconnaissance vocale**

Si l'on voit encore plus loin dans le futur en restant sur notre idée de devinettes d'animaux, on peut imaginer développer ce produit avec des outils de reconnaissance vocale. On sait par exemple qu'avec *Alexa* et les différentes API mises en ligne par Amazon (Alexa Skills kit), on pourrait s'en servir pour porter notre jeu sur ce produit et ainsi proposer notre jeu sur une autre plateforme.