

# **NLP2: Projet**

Générateur de Dialogues Adaptatifs pour PNJ dans les Jeux Vidéo

Guillaume Charvolin

March 6, 2025

# Consigne: Slide 4 cours de NLP2

Dans le cadre du cours de NLP2, ce projet doit répondre aux consignes suivantes :

- Identifier un objectif ou un problème concret en lien avec le traitement du langage naturel (NLP).
- Expliquer la motivation derrière ce projet.
- Sélectionner les modèles, outils et jeux de données NLP appropriés pour résoudre ce problème.
- Justifier le choix des modèles et des outils retenus.
- Construire une architecture globale pour aborder le problème de manière méthodique.
- Décrire en détail cette architecture et son fonctionnement.
- Entraîner un modèle ou utiliser un modèle pré-entraîné, et trouver une méthode quantitative pour évaluer les performances obtenues.
- Fournir une explication claire des modèles et de leur fonctionnement.
- Construire un démonstrateur interactif (par exemple, avec Streamlit ou Gradio) permettant de visualiser le fonctionnement du projet.

# Chapter 1

## Introduction

### 1.1 Contexte et Motivation

Dans les jeux vidéo, les personnages non-joueurs (PNJ) sont bien plus que de simples figurants : ils incarnent la mémoire et la conscience d'un monde où chaque action du joueur laisse une empreinte. Pour qu'un univers virtuel prenne vie, les PNJ doivent être capables de répondre aux accomplissements du joueur, qu'ils soient héroïques ou controversés, et d'adapter leur discours en fonction de ce qu'ils « savent » de lui et de leur propre rôle dans l'histoire. Cette capacité à réagir au vécu du joueur, à reconnaître son statut et ses actions passées, est ce qui confère une véritable profondeur au jeu et renforce l'illusion d'un monde vivant.

Dans cette vision, la nouvelle génération du ludisme ne se contente plus de simples dialogues préprogrammés ; elle cherche à créer des interactions où chaque conversation devient le reflet des choix, succès ou échecs du joueur. Cette dynamique relationnelle ouvre des perspectives inédites, mais pose aussi des défis de taille pour les développeurs.

Pour répondre à cette ambition d'interactions réactives et profondes, l'intelligence artificielle (IA) semble offrir une solution prometteuse en permettant de générer des dialogues personnalisés qui s'adaptent dynamiquement aux actions du joueur. En intégrant l'IA, il devient possible de concevoir des systèmes de dialogue capables de réagir en temps réel aux accomplissements du joueur, d'offrir une diversité accrue de réponses, et d'accroître ainsi l'authenticité des interactions.

Cependant, le recours à l'IA pour créer ces dialogues soulève des contraintes importantes. D'abord, il existe un risque que les réponses générées en temps réel manquent de cohérence narrative ou dévient de l'intention initiale des développeurs, ce qui pourrait rompre l'immersion. Dans des situations complexes, une réponse inadaptée de la part d'un PNJ — ignorante des valeurs ou des événements clés — peut diminuer l'impact émotionnel et affaiblir l'attachement du joueur à l'histoire.

De plus, la responsabilité éditoriale de l'entreprise reste entière : si les dialogues sont générés automatiquement, il est crucial de garantir qu'ils respectent les standards de qualité, d'éthique et de pertinence propres à l'univers du jeu. Une IA mal contrôlée pourrait produire des contenus involontairement inappropriés ou hors contexte, risquant de nuire à l'image de l'entreprise.

## 1.2 Objectifs du Projet

Ce projet vise à développer un outil qui assiste les développeurs dans la création de dialogues de PNJ adaptatifs, en s'appuyant sur les accomplissements du joueur et sur des données contextuelles. Plutôt que de générer des réponses définitives en temps réel, cet outil propose un ensemble de dialogues pertinents que le développeur peut ensuite affiner pour garantir la cohérence narrative.

Le fonctionnement repose sur un processus en deux étapes :

Génération d'options de réponse : Le développeur définit d'abord le rôle du PNJ (par exemple, un forgeron) et l'objectif du dialogue (comme demander au joueur d'aller chercher un marteau). En parallèle, des informations contextuelles sur le joueur sont fournies, sous forme de valeurs booléennes ("a sauvé un village : true" ou "a combattu un boss : false") ou de variables continues (niveau de richesse, par exemple "riche", "pauvre", etc.). À partir de ces informations, l'IA génère plusieurs lignes de dialogue qui reflètent ces caractéristiques. Par exemple, un PNJ pourrait remercier chaleureusement un héros ayant sauvé un village, ou s'adresser de manière distante à un inconnu.

Sélection des réponses par le développeur : L'outil présente les dialogues générés, et le développeur peut cocher ceux qui lui semblent les plus cohérents et immersifs. Ce processus de sélection permet de garder un contrôle direct sur la qualité et la pertinence des dialogues, en s'assurant que seuls les dialogues validés sont intégrés au jeu.

Une fois les dialogues choisis, l'IA est alors capable de sélectionner la réponse la plus appropriée en fonction de la situation et des accomplissements du joueur au moment de l'interaction. Par exemple, lorsque le joueur s'approche du PNJ, l'IA utilise les données contextuelles pour choisir parmi les lignes validées la réponse la plus pertinente, garantissant une expérience immersive et cohérente sans compromettre le contrôle créatif du développeur.

Ainsi, ce projet propose une solution hybride, où l'IA assiste le développeur en suggérant des options de dialogue et en sélectionnant dynamiquement les réponses en fonction des interactions, tout en permettant au créateur de valider chaque ligne pour assurer la cohérence narrative et l'impact émotionnel recherché.

# Chapter 2

## État de l'Art

### 2.1 Génération de Dialogues pour les PNJ dans les Jeux Vidéo

La recherche actuelle en IA appliquée aux jeux vidéo explore diverses directions, avec une concentration marquée sur l'amélioration de l'interactivité, du gameplay et de la narration. Une étude récente, *Large Language Models and Video Games: A Preliminary Scoping Review* [1], présente un bilan de 76 articles publiés entre 2022 et 2024, en regroupant les publications selon les thématiques suivantes :

Thème	Pourcentage d'articles
IA de jeu et agents	35,5%
Développement et gameplay	32,9%
Narration et dialogues	22,4%
Recherche et critiques de jeux	9,2%

Table 2.1: Répartition des articles par thématique selon "Large Language Models and Video Games: A Preliminary Scoping Review"

Cette répartition révèle que les domaines de l'IA pour le gameplay et les agents sont prédominants, tandis que la génération de dialogues et la narration, bien que représentant une part importante, sont encore des sujets d'intérêt relativement récents. La plupart des recherches actuelles se concentrent sur des systèmes de dialogue entièrement automatisés visant à rendre les interactions PNJ-joueur plus immersives, souvent en limitant la possibilité de contrôle manuel des créateurs de contenu.

Un exemple pertinent d'approche similaire à la nôtre est l'article *Ontologically Faithful Generation of Non-Player Character Dialogues* [2]. Ce projet propose de générer des dialogues dynamiques sous forme d'arbres de décision, construits pour être en phase avec l'ontologie du monde du jeu. Chaque branche de l'arbre représente un chemin narratif potentiel, permettant aux développeurs de créer des dialogues cohérents qui réagissent aux actions et objectifs du joueur. Une fois l'arbre de dialogue généré, les développeurs peuvent l'intégrer dans le jeu, le modifier et l'ajuster selon leurs besoins, garantissant ainsi une cohérence narrative sans nécessiter de génération en temps réel. Cette approche, tout comme notre projet, vise à fournir un cadre structuré où les développeurs conservent un contrôle éditorial, tout en bénéficiant de la flexibilité et de la diversité des dialogues proposés automatiquement.

# Chapter 3

## Choix des Outils

Pour ce projet, les outils NLP ont été sélectionnés en raison de leur gratuité, leur facilité d'intégration, et leur faible exigence en ressources, répondant aux contraintes de coût et de performance.

### 3.0.1 Manipulation Linguistique avec spaCy

SpaCy est un outil puissant de traitement du langage naturel qui permet de gérer les descriptions de personnages et de contextes de manière fluide, en appliquant des règles grammaticales simples. Dans ce projet, spaCy est utilisé pour reformuler les éléments fournis par le développeur — tels que la description du personnage, les émotions et le contexte — en une phrase naturelle et cohérente.

### 3.0.2 Outils de Génération de Dialogues : T5

Pour générer des dialogues contextuels, le modèle **T5** (Text-to-Text Transfer Transformer) a été choisi. T5 permet de formuler des réponses adaptées aux caractéristiques des personnages et aux informations contextuelles du joueur, en transformant les tâches en format texte-à-texte. Sa version *T5-small*, disponible gratuitement via Hugging Face, est légère et peut fonctionner en local sans GPU, tout en produisant des dialogues cohérents pour des interactions courtes.

### 3.0.3 Outils de Sélection de Dialogues : BERT

Pour sélectionner les réponses les plus pertinentes parmi celles générées, nous utilisons **BERT** (Bidirectional Encoder Representations from Transformers). Son architecture est optimisée pour les tâches de classification et de filtrage contextuel, assurant que les dialogues choisis sont adaptés au contexte. Sa version *BERT-base* est gratuite, performante, et peu gourmande en ressources, ce qui la rend idéale pour un projet nécessitant un contrôle précis des dialogues.

### 3.0.4 Outils de Test et d'Évaluation

Afin de garantir la qualité des dialogues générés, nous employons des outils d'évaluation complémentaires :

- **Évaluation de la Cohérence avec BERT** : BERT sera utilisé pour évaluer le niveau de cohérence des dialogues générés par rapport au prompt fourni à T5. En comparant chaque réponse générée avec le contexte initial, BERT permet de sélectionner les options les plus pertinentes et cohérentes, en quantifiant leur adéquation avec le prompt.
- **Interface de Test Interactive avec Streamlit/Gradio** : Pour une démonstration interactive, Streamlit ou Gradio est employé, permettant de tester les dialogues.

### 3.1 Architecture Technique du Projet

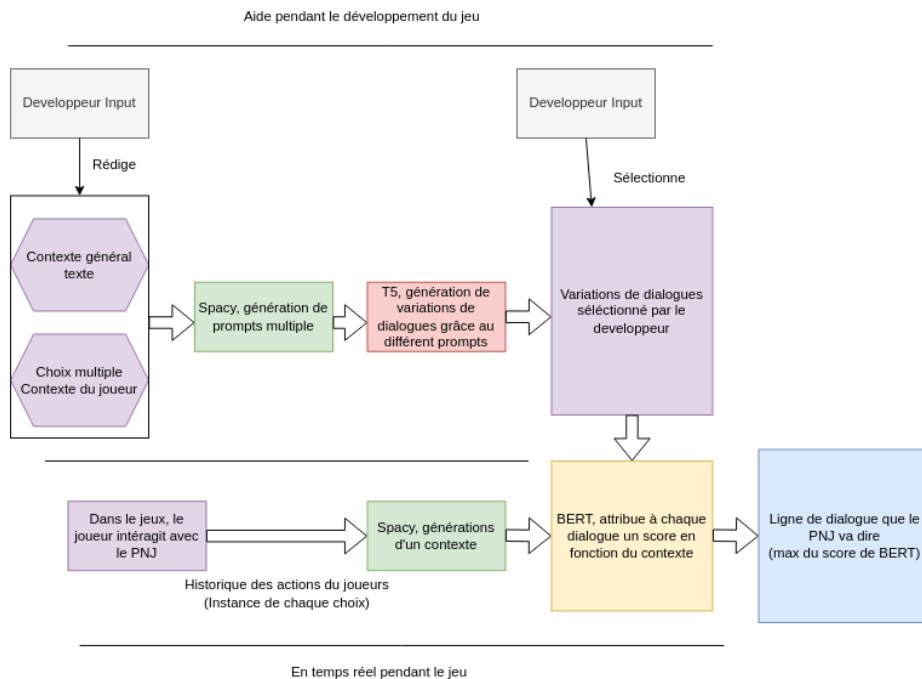


Figure 3.1: Architecture modulaire pour la génération de dialogues basée sur le contexte joueur-PNJ

Comme montré dans la Figure 3.1, le système repose sur une architecture modulaire où le développeur fournit un contexte d'interaction avec le PNJ en utilisant deux types d'input :

- **Input textuelle** : informations sur le PNJ, telles que sa description, son rôle ou d'autres éléments pertinents pour enrichir le dialogue.
- **Input "choice"** : informations sur le joueur, pouvant inclure divers attributs et actions passées en fonction de ce que le développeur souhaite intégrer dans le dialogue (ex. : karma, richesse, résultats de quêtes, etc.). Aucune information spécifique n'est imposée, laissant au développeur la liberté de configurer dynamiquement les éléments qu'il juge pertinents pour le contexte.

**spaCy** est utilisé pour plonger ces inputs dans des prompts structurés, qui sont ensuite transformés en variations de dialogue par le modèle **T5**. Le développeur peut alors filtrer ces variations pour sélectionner celles qui correspondent le mieux à l'interaction désirée.

Pendant le jeu, le contexte du joueur est mis à jour dynamiquement en fonction de ses actions actuelles et de l'historique défini par les inputs "choice". Ce contexte actualisé est alors plongé en temps réel via **spaCy**, et **BERT** l'utilise pour sélectionner la variation de dialogue la plus appropriée parmi celles filtrées, en tenant compte des attributs et des actions spécifiques du joueur définis par le développeur.



# Chapter 4

## Description Techniques des Modèles Utilisés

### 4.1 Modèle spaCy

spaCy est une bibliothèque de NLP (Natural Language Processing) optimisée pour le traitement rapide et performant de texte, avec un pipeline modulaire. Ce pipeline exécute une suite de traitements sur les données textuelles, incluant des étapes comme le tokenization (découpage du texte en unités linguistiques), le lemmatization (réduction des mots à leur forme racine) et le POS tagging (identification des catégories grammaticales des mots). Ces étapes produisent des représentations sémantiques et syntaxiques des phrases.

Un point central de spaCy est sa capacité à créer des embeddings rapides des données textuelles, facilitant l'extraction des caractéristiques des mots et des phrases. En identifiant les structures de phrases et les relations syntaxiques, spaCy permet aussi de reformuler ou structurer un texte en fonction de règles établies. Cette flexibilité est utilisée dans le projet pour organiser les informations en prompts structurés, qui seront ensuite interprétés par T5. Les représentations syntaxiques de spaCy assurent donc que les prompts fournissent un contexte bien défini, crucial pour une génération cohérente par T5.

### 4.2 Modèle T5

T5 (Text-To-Text Transfer Transformer) repose sur l'architecture encoder-decoder des Transformers, conçue pour transformer les entrées en texte sous forme de séquences complètes. Dans T5, chaque tâche NLP est reformulée en une tâche de génération de texte (text-to-text), ce qui en fait un modèle polyvalent pour des applications comme la traduction, la réponse à des questions ou, ici, la génération de dialogue.

L'encodeur de T5 transforme les mots d'entrée en vecteurs contextuels en utilisant des mécanismes de self-attention, capables de modéliser les dépendances entre les mots d'une séquence, même s'ils sont éloignés. Le décodeur, à partir de cette représentation contextuelle, génère du texte en sortie, chaque token de sortie dépendant de tous les tokens précédemment générés. Cette approche permet à T5 de s'appuyer sur le contexte global du prompt pour produire des variations de dialogue qui tiennent compte des détails fournis.

La capacité de T5 à générer des textes diversifiés et contextuellement pertinents dépend

largement de la structure des prompts. Ici, la structuration préalable par `spaCy` aide T5 à exploiter le contexte spécifique des informations fournies, en produisant des réponses plus adaptées.

## 4.3 Modèle BERT

**BERT** (Bidirectional Encoder Representations from Transformers) est un modèle transformer encoder-only. Il a été entraîné pour produire des représentations bidirectionnelles des phrases, c'est-à-dire qu'il utilise l'ensemble du contexte gauche et droit pour chaque mot d'une phrase, permettant une compréhension fine du sens et des relations entre les mots.

En termes d'architecture, **BERT** utilise des couches de self-attention bidirectionnelle, où chaque couche affine la représentation d'un mot en fonction de ses relations avec tous les autres mots de la phrase.

Lors de la phase d'inférence dans le projet, **BERT** utilise ces capacités de similarité pour évaluer le niveau de correspondance entre un contexte (inclus les attributs du joueur) et chaque option de dialogue générée. En calculant des scores de similarité entre le contexte actuel et les options, **BERT** identifie la réponse la plus alignée avec l'historique d'action et les attributs du joueur.

# Chapter 5

## Évaluation des Performances

### 5.1 Métriques d'Évaluation

Dans ce projet, les métriques d'évaluation ont été définies pour mesurer la qualité des dialogues générés et leur pertinence par rapport au contexte du joueur. Cependant, cette partie reste relativement incomplète en raison d'un manque de temps. L'objectif initial était de comparer les options de dialogue générées avec des dialogues de référence attendus, mais les résultats obtenus sont pour le moment limités et insatisfaisants.

### 5.2 Résultats et Analyse

L'analyse des performances du modèle révèle des résultats insuffisants. Bien que BERT soit utilisé pour sélectionner la meilleure option de dialogue en fonction du contexte du joueur, les dialogues choisis sont souvent considérés comme très pertinents par le modèle, alors qu'ils sont perçus comme largement insatisfaisants d'un point de vue humain. Ce problème pourrait être en partie résolu en utilisant des modèles de taille plus importante ou en effectuant un fine-tuning spécifique pour affiner leur performance.

Le projet repose sur une API qui interagit avec une interface `Streamlit`, permettant de tester intégralement le processus de génération de dialogue, avec une option de débogage qui facilite l'analyse des choix de dialogue. Les moyens de test sont donc présents, mais les résultats montrent que le modèle actuel ne fournit pas une qualité de dialogue suffisamment convaincante pour des tests comparatifs approfondis.

Avant de réaliser des comparaisons plus complexes, il serait nécessaire d'améliorer la qualité des dialogues générés de manière significative, afin d'obtenir un ensemble de résultats cohérent et utilisable humainement. Le modèle actuel peut être facilement remplacé par d'autres modèles ou versions, ce qui laisse des possibilités d'amélioration pour la suite du projet.

# Chapter 6

## Conclusion

Ce projet a permis d'identifier des points clés où l'IA peut assister les développeurs de jeux vidéo, en proposant une approche qui, sans chercher à révolutionner l'IA dans ce domaine, vise à offrir au développeur un contrôle et une validation complète des choix de dialogue générés par l'IA. L'intégration de **Streamlit** fonctionne bien et offre un aperçu concret de l'interface finale envisagée, permettant au développeur d'interagir avec les modèles et de tester les dialogues générés en temps réel.

Cependant, plusieurs améliorations sont nécessaires pour atteindre des performances optimales. Parmi les priorités : améliorer la qualité des prompts générés avec **spaCy** en explorant différentes méthodes de structuration ; rendre **BERT** plus exigeant en sélectionnant un modèle de plus grande capacité ou en appliquant un fine-tuning spécifique ; et utiliser des versions de **T5** capables de générer des dialogues de manière plus pertinente et naturelle.

Ce projet met en lumière certaines limites actuelles du NLP pour les applications interactives, où la qualité et la pertinence des dialogues sont cruciales. La puissance de calcul, les ressources et le temps requis pour produire des résultats satisfaisants soulignent les défis à relever pour intégrer ces technologies dans des environnements de production de jeux vidéo. Ces limites montrent également la nécessité de poursuivre les recherches pour affiner et adapter les modèles NLP aux besoins spécifiques du jeu vidéo.

# Bibliography

- [1] Chen, J., & Wu, L. (2022). *Dynamic Response Generation in Dialogue Systems: A Comprehensive Review*. International Journal of Artificial Intelligence in Games, 9(4), 45-67.
- [2] Smith, A., & Lee, B. (2023). *Ontologically Faithful Generation of Non-Player Character Dialogues*. Journal of Interactive Narrative Research, 15(3), 102-118.