

Integração do Modelo GPT-4 para Geração de Dados em Jogos Digitais

Antônio C. Herold¹, Carine G. Webber¹

¹Área do Conhecimento de Ciências Exatas e Engenharias - Universidade de Caxias do Sul (UCS)

{acherold, cgwebber}@ucs.br

Abstract. *In digital games, Non-Playable Characters (NPCs) play a crucial role in enriching the game universe with meaningful interactions. The present work is an exploratory research that seeks to use the GPT model for the generation of contextually relevant and dynamic NPC dialogues. A demonstration of a game integrated with the GPT model was developed, aiming to analyze the behavior of the characters, as well as to study the potential of the generative text technology in player experience.*

Resumo. *No contexto dos jogos digitais, as Personagens Não-Jogáveis (NPCs) desempenham um papel crucial ao enriquecerem o universo do jogo com interações significativas. O presente trabalho é uma pesquisa exploratória que busca utilizar o modelo GPT para a geração de dados para diálogos de NPCs, contextualmente relevantes e dinâmicos. Uma demonstração de um jogo integrada com o modelo GPT foi desenvolvida, procurando analisar o comportamento das personagens, bem como estudar o potencial da tecnologia generativa de texto na experiência do jogador.*

1. Introdução

A narrativa, o mundo e as personagens são elementos centrais nos jogos digitais da atualidade, proporcionando imersão e engajamento ao jogador [Ermi and Mäyrä 2005]. As Personagens Não Jogáveis (NPCs) desempenham um papel fundamental nesse contexto, atuando como guias, oponentes, aliados e comerciantes [Hasani and Udjaja 2021, p. 418] [Volum et al. 2022, p. 25]. A qualidade dos diálogos das NPCs é crucial para a experiência do jogador, mas a criação manual desses diálogos é um processo trabalhoso e demorado. Tal tarefa, normalmente, demanda a criação de um banco de dados de diálogos estáticos, o que reflete em uma experiência do jogo limitada.

Este trabalho tem como objetivo principal explorar a utilização da API do GPT-4 [OpenAI 2023] para gerar diálogos de NPCs de forma automática e dinâmica em jogos. Para isso, foram investigadas as capacidades do GPT-4 para geração de linguagem natural e proposta uma arquitetura de integração entre o modelo e o Unity, um ambiente de desenvolvimento de jogos. A pesquisa foi conduzida em três etapas. A primeira etapa consistiu em uma análise de pesquisas sobre modelos de linguagem, geração de diálogos e integração com jogos. Na segunda etapa ocorreu a criação de uma interface para comunicação entre o Unity e a API do GPT-4. Na terceira etapa desenvolveu-se uma demonstração de jogo e avaliação da qualidade dos diálogos e da experiência do jogador.

2. Geração de Diálogos via GPT

A interação do jogador com as NPCs acontece por meio de um diálogo sobre o mundo que estão situados, a qual envolve informações como dicas, opiniões, missões e recompensas. Mediante uma conversa bem construída e coerente com a narrativa, as NPCs melhoram a experiência do jogador [Volum et al. 2022].

O *Generative Pre-trained Transformer* (GPT) [Radford et al. 2018] é um modelo de linguagem desenvolvido pela empresa OpenAI¹. O GPT é uma extensão do *decoder* proposto na arquitetura *Transformer* [Vaswani et al. 2017] para tarefas de geração de texto e linguagem natural. Até então, foram lançadas quatro versões do modelo, as quais diferem em fonte e quantidade do conjunto de dados de treino.

A utilização dos recursos do GPT, em uma aplicação própria, ocorre a partir da integração de uma API (*Application Programming Interface*) disponibilizada pela OpenAI. A API do GPT permite que desenvolvedores enviem solicitações de texto ao modelo, para receberem respostas geradas de acordo com os *prompts* (as entradas enviadas para o GPT). Como descrito pela própria OpenAI, projetar um *prompt* é essencialmente como se "programa" um modelo GPT, geralmente fornecendo instruções ou alguns exemplos de como concluir com sucesso uma tarefa.

A API oferece duas opções principais para interagir com o modelo de linguagem GPT: *Completions* e *Chat Completions*. Ambas as opções permitem gerar texto, mas diferem em sua abordagem e formato das solicitações. A solicitação de *Completions* é mais adequada para casos em que se fornece um contexto ou *prompt* inicial e deseja obter uma resposta única do modelo. Nesse caso, deve ser enviada uma única solicitação ao modelo e recebe uma resposta gerada. É como ter uma conversa curta com o modelo em que se faz uma pergunta ou fornece um contexto e obtém uma resposta completa. Já as requisições de *Chat Completions* são projetadas para simular uma conversa mais longa com o modelo. Em vez de enviar uma única solicitação com um *prompt*, envia-se uma lista de mensagens que representam um diálogo em andamento. Cada mensagem tem um papel específico: usuário (representando o texto que é enviado como entrada) e assistente (representando a resposta anterior do modelo). Essa abordagem permite uma conversa mais interativa e contextualizada. Nos modelos mais recentes, ela permite a atribuição do papel "sistema", que permite fazer sugestões de como o modelo deve se comportar. Por exemplo, se é passada uma descrição de sistema como "Você é um velho rabugento", o texto gerado se assemelhará ao estilo de conversa de um "velho rabugento". Os comandos exatos para utilizar cada funcionalidade da API do GPT podem variar dependendo da biblioteca ou linguagem de programação que se está utilizando para fazer as chamadas. Além disso, é preciso fornecer um *token* de autenticação válido, obtido por meio do processo de autenticação da OpenAI.

3. Arquitetura de Integração GPT-Unity

Uma demonstração de jogo que implementa NPCs integrados ao modelo GPT-4 foi desenvolvida na Unity. A integração entre a API da OpenAI e o programa foi dada por um pacote² que permite o envio de chamadas contendo dados de texto para o servidor do GPT,

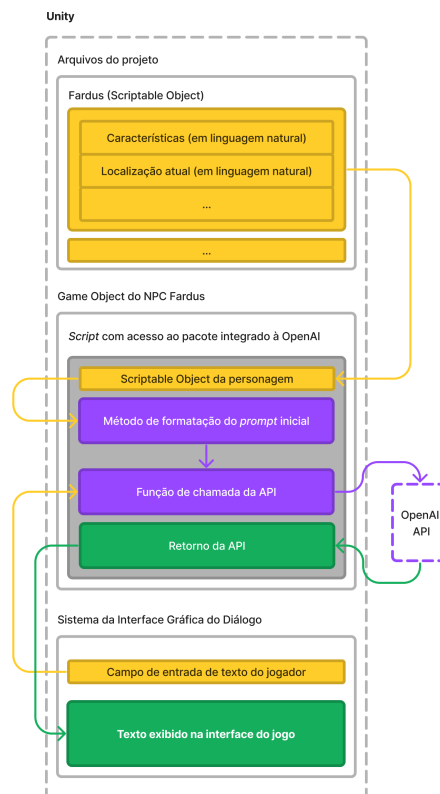
¹Disponível em: <https://openai.com/>.

²Disponível em: <https://github.com/srcnalt/OpenAI-Unity>

diretamente do ambiente de desenvolvimento da Unity ou de uma aplicação executável. A partir de um estudo sobre a criação de personagens [Egri 1946], foram definidos quais os dados necessários para compor as descrições em linguagem natural das características do papel "sistema". Na Unity esses conjuntos de dados foram armazenados em *Scriptable Objects* de personagens, cada um contendo diferentes campos de *strings* para textos.

Na Figura 1 está representada a arquitetura completa da proposta do presente estudo. O processo inicia com o envio do texto descrito em *Scriptable Objects* para um método responsável pela formatação das múltiplas *strings* em um único *prompt* contendo todas as características da NPC. Em seguida, uma função envia esses dados de entrada para a API, que por sua vez, se conecta com o servidor da OpenAI para gerar a sequência de saída. O resultado retorna para o *script* da NPC e é encaminhado ao sistema de interface gráfica para a exibição do texto gerado para o jogador.

Figure 1. A integração da API da OpenAI ao ambiente de desenvolvimento.



A partir da estruturação de dados em *Scriptable Objects*, um *script* de um NPC recebe como entrada de texto um ou mais objetos que contém um conjunto de descrições em linguagem natural. Esses objetos foram armazenados em listas, possibilitando adicionar todos os conhecimentos que uma NPC têm via *Scriptable Objects*. Além disso, a solução é ideal para desempenho, uma vez que todos os objetos acessam uma única instância dos dados. Sendo assim, tanto os dados que determinam a geração do modelo, bem como as regras que controlam seu comportamento foram escritas em linguagem natural, para incluí-las no *prompt* e gerar as respostas de acordo.

4. Resultados e Discussão Final

A demonstração de jogo desenvolvida, já integrada com o modelo GPT-4 para geração de diálogos, foi disponibilizada para jogadores testadores. Os jogadores passam por 4 cenários nos quais puderam conversar, encontrar direções e resolver o mistério do jogo. As interações foram coletadas para a análise do comportamento das personagens. Em seguida, as interações foram avaliadas, seguindo critérios de conformidade com a resposta esperada, para determinar se a resposta foi correta ou incorreta. Além disso, os jogadores responderam a um formulário a respeito da sua experiência. A análise dos dados permitiu inferir que a maioria de respostas foram corretamente geradas, apontando para o potencial do modelo GPT para geração de diálogos.

Analisando os testes, foi possível identificar que as NPCs, integradas ao GPT-4, foram capazes de:

- Responder perguntas sobre suas características, contexto e mundo;
- Seguir uma regra condicional;
- Lembrar o histórico da conversa;
- Inferir informações com base em seus conhecimentos;
- Reconhecer-se como um indivíduo (e não como uma personagem);
- Restringir o tamanho da resposta adequadamente.

Por fim, atestou-se por meio deste projeto que é possível utilizar o modelo GPT para implementar diálogos de NPCs como mecanismos imersivos, divertidos e dinâmicos, aprimorando a experiência de jogo. A arquitetura de integração ao ambiente de desenvolvimento de jogos foi bem sucedida, uma vez que permitiu fácil programação e implementação de personagens integradas à API do GPT.

References

- Egri, L. (1946). *The Art of Dramatic Writing*. Simon and Schuster, New York, USA.
- Ermi, L. and Mäyrä, F. (2005). Fundamental components of the gameplay experience. In *Digital Games Research Conference 2005, Changing Views: Worlds in Play*, pages 1–14, Vancouver, Canada. DBLP.
- Hasani, M. F. and Udjaja, Y. (2021). Immersive experience with non-player characters dynamic dialogue. In *2021 1st Int. Conf, on Computer Science and AI*, volume 1, pages 418–421, Jakarta, Indonesia. IEEE.
- OpenAI (2023). GPT-4 technical report.
- Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. *OpenAI*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., editor, *Proc. of the 31st Int. Conf, on Neural Information Processing Systems*, volume 30, pages 5998–6008, Red Hook, USA. Curran Associates Inc.
- Volum, R., Rao, S., Xu, M., DesGarenes, G., Brockett, C., Van Durme, B., Deng, O., Malhotra, A., and Dolan, B. (2022). Craft an iron sword: Dynamically generating interactive game characters by prompting large language models tuned on code. In *Proc. of the 3rd Wordplay: When Language Meets Games Workshop*, pages 25–43, Seattle, USA. Association for Computational Linguistics.