

Geração de Mapas para Jogos Roguelike a Partir de Descrições Textuais Utilizando Modelos de Linguagem de Larga Escala

Gustavo Gurgel (Orientador: Cristiano Bacelar)

gusgurgel@alu.ufc.br

Universidade Federal do Ceará

19 de janeiro de 2026

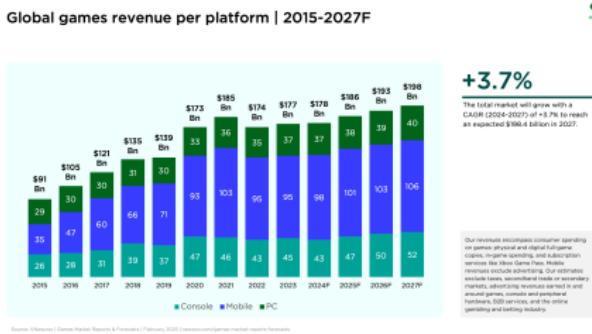
Sumário

1 Introdução

- 2 Fundamentação
- 3 Trabalhos Relacionados

Panorama Atual do Mercado de Jogos

Figura: Global games market: growth drivers and challenges for 2025-2027



Fonte: Newzoo

O gráfico demonstra o crescimento constante do mercado global de jogos de 2015 a 2027, com uma previsão de aumento de 3.7% da sua receita entre 2025 e 2027. Com uma renda prevista de **\$198 bilhões** em 2027, evidencia a importância desse mercado na economia atual.

Crescimentos dos Jogos Roguelikes



Fonte: Wikipedia

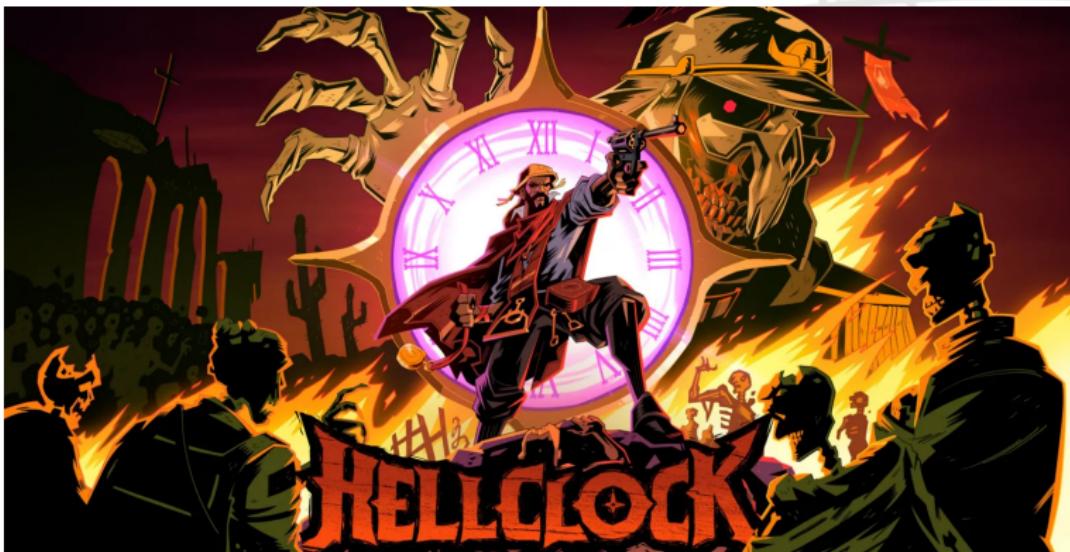
Balatro é um roguelike de construção de baralhos. **Foi um dos 6 jogos indicados para jogo do ano no The Game Awards 2024**. Isso mostra que, mesmo sendo um gênero antigo, ainda possui grande destaque no mercado de jogos.



UNIVERSIDADE
FEDERAL DO CEARÁ

Crescimentos dos Jogos Roguelikes

Figura: Hell Clock (2025): Roguelike brasileiro inspirado na Guerra dos Canudos



Fonte: Steam

Geração Procedural de Conteúdo (PCG)

Figura: Mapas gerados utilizando PCG (The Binding of Isaac)



Fonte: Level Generation by Joining Geometry [1]

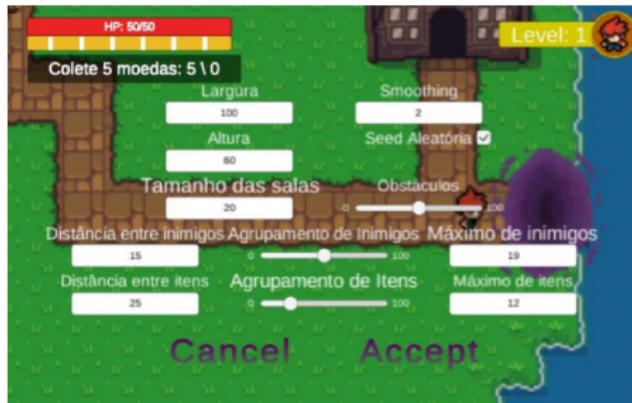
Uma das principais características dos jogos Roguelike é a presença de conteúdos gerados proceduralmente. Ou seja, conteúdos gerados por algoritmos, geralmente utilizam geradores de números pseudo aleatórios.



UNIVERSIDADE
FEDERAL DO CEARÁ

Geração Procedural de Conteúdo (PCG)

Figura: Gerador de Mapas com Diversos Parâmetros de Configuração

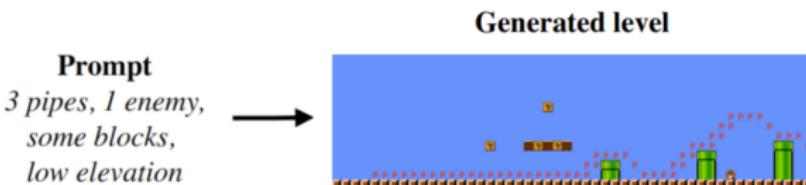


Fonte: Regressão para Predição de Mapas Gerais Proceduralmente. [2]

A geração desses conteúdos geralmente pode ser controlada por parâmetros como tamanho do objeto gerado, quantidades de elementos ou até sementes para o gerador pseudo aleatório.

PCG + LLMs

Figura: Geração de Mapas Utilizando Descrições Textuais



Fonte: MarioGPT: Open-Ended Text2Level Generation through Large Language Models [3]

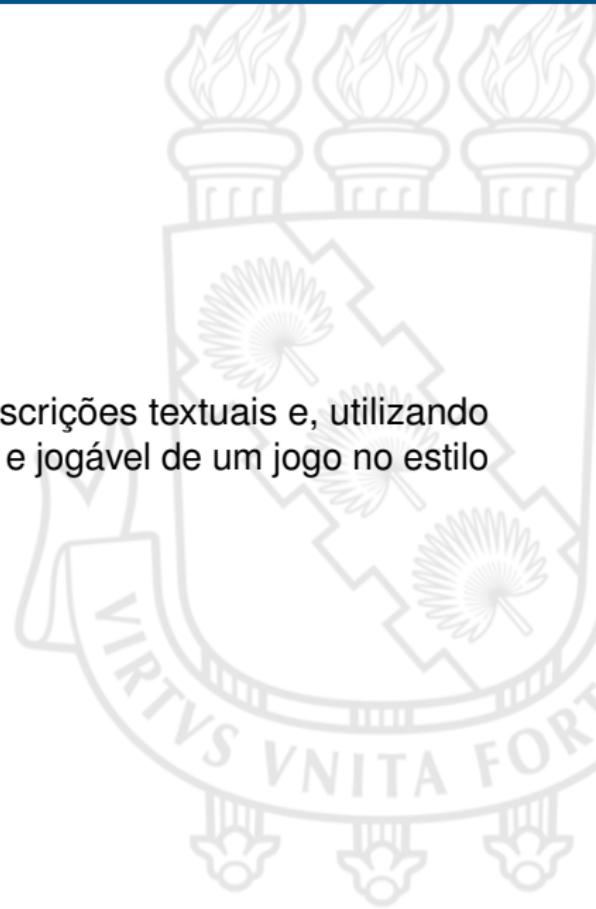
Com o surgimento dos Modelos Transformers Generativos, surgem grandes oportunidades de integrar essa tecnologia na geração procedural de conteúdo em jogos. Permitindo que o controle da geração seja feito por prompts textuais, tornando a geração mais expressiva do que a configuração de parâmetros predefinidos.



UNIVERSIDADE
FEDERAL DO CEARÁ

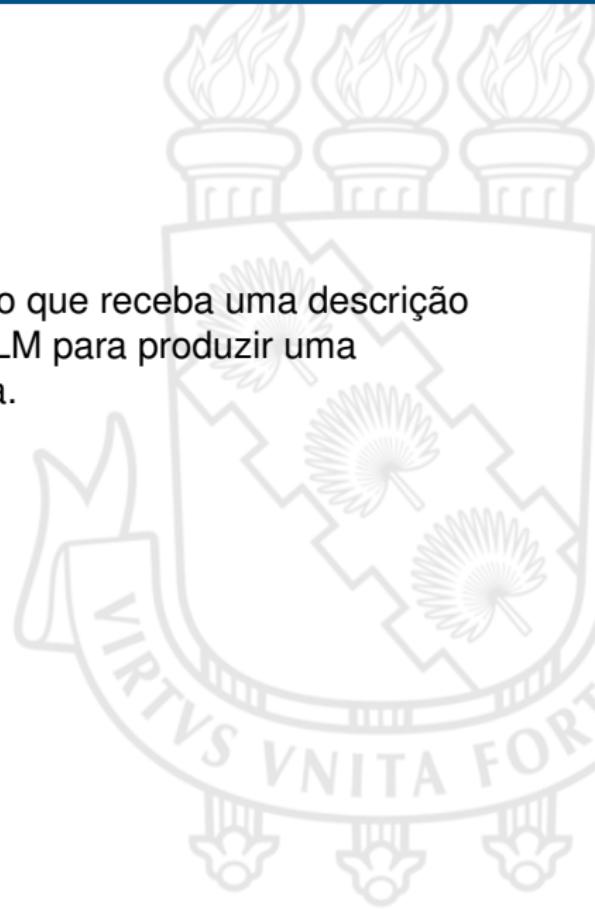
Objetivo Geral

Desenvolver um sistema que recebe descrições textuais e, utilizando um LLM, traduz em um nível estruturado e jogável de um jogo no estilo roguelike



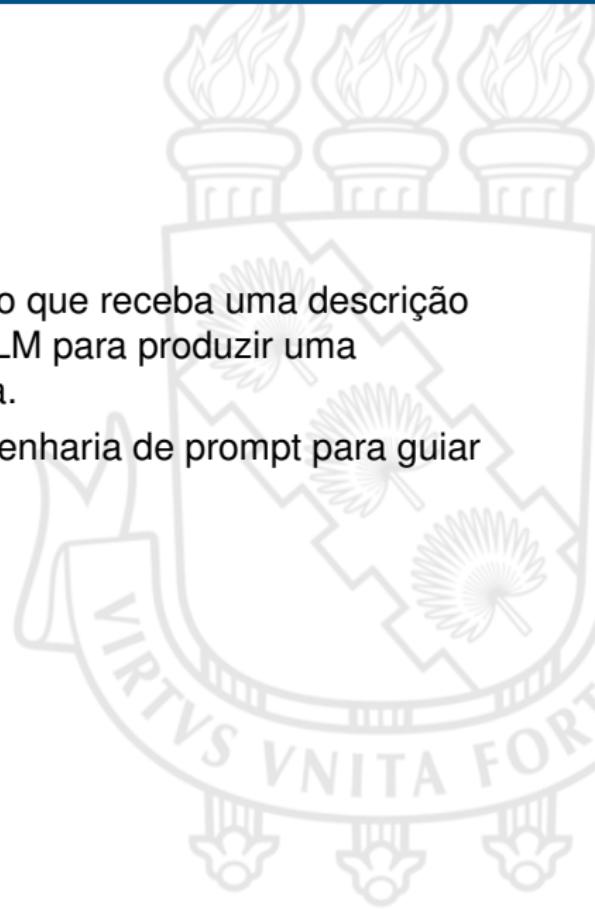
Objetivos Específicos

- Desenvolver um pipeline de geração que receba uma descrição textual como entrada e utilize um LLM para produzir uma representação de mapa estruturada.



Objetivos Específicos

- Desenvolver um pipeline de geração que receba uma descrição textual como entrada e utilize um LLM para produzir uma representação de mapa estruturada.
- Investigar e aplicar técnicas de engenharia de prompt para guiar o LLM de forma eficaz.

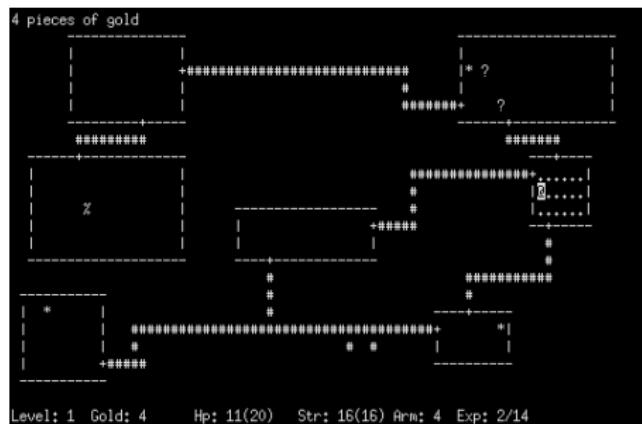


Objetivos Específicos

- Desenvolver um pipeline de geração que receba uma descrição textual como entrada e utilize um LLM para produzir uma representação de mapa estruturada.
- Investigar e aplicar técnicas de engenharia de prompt para guiar o LLM de forma eficaz.
- Implementar um protótipo funcional que integre o pipeline de geração, capaz de processar os dados do mapa e renderizá-los visualmente.

Origem dos Roguelikes

Figura: Rogue (1980)



Fonte: Wikipedia

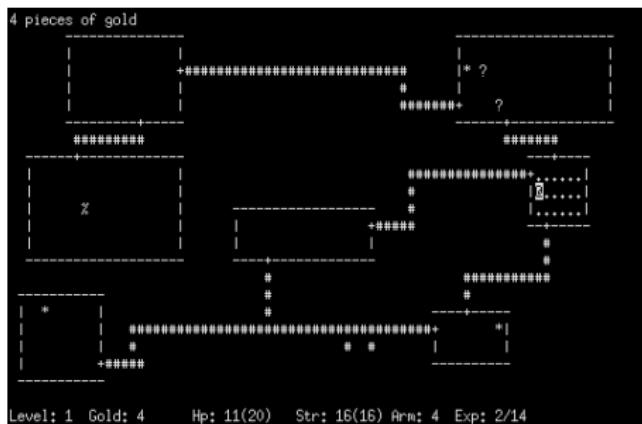
Em 1980 surge **Rogue**, um jogo de terminal onde o jogador explorar masmorras geradas aleatoriamente, enfrentando monstros, coletando itens e tentando chegar ao final em uma única vida.



UNIVERSIDADE
FEDERAL DO CEARÁ

Origem dos Roguelikes

Figura: Rogue (1980)



Fonte: Wikipedia

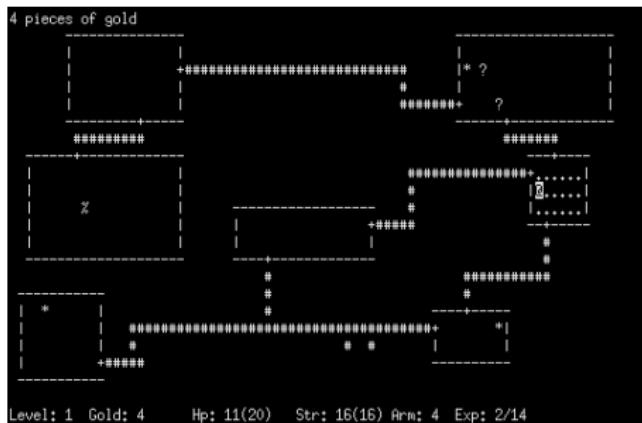
Esse jogo foi tão marcante que deu origem a um gênero inteiro chamado Roguelike, que se baseia em elementos como:



UNIVERSIDADE
FEDERAL DO CEARÁ

Origem dos Roguelikes

Figura: Rogue (1980)



Fonte: Wikipedia

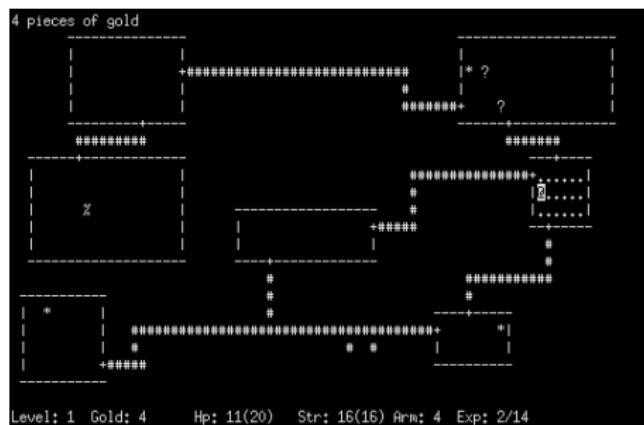
Esse jogo foi tão marcante que deu origem a um gênero inteiro chamado Roguelike, que se baseia em elementos como:

- Geração procedural.



UNIVERSIDADE
FEDERAL DO CEARÁ

Origem dos Roguelikes



Fonte: Wikipedia

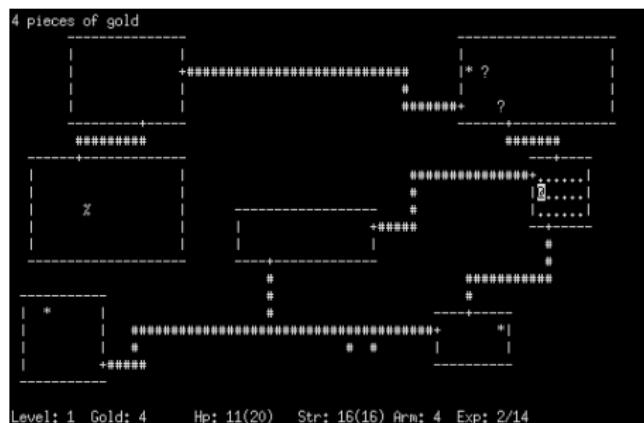
Esse jogo foi tão marcante que deu origem a um gênero inteiro chamado Roguelike, que se baseia em elementos como:

- Geração procedural.
- Permadeath (morte permanente).



UNIVERSIDADE
FEDERAL DO CEARÁ

Origem dos Roguelikes



Fonte: Wikipedia

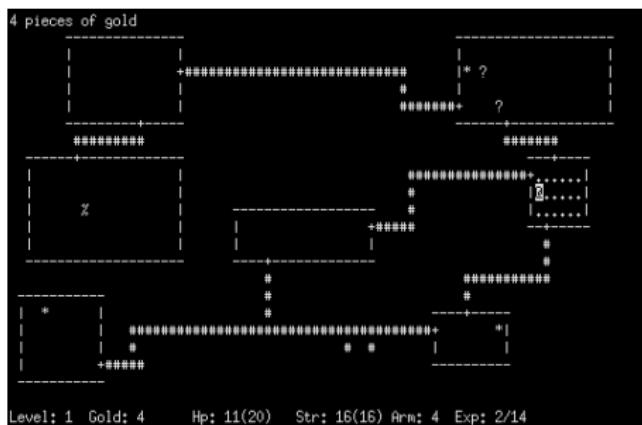
Esse jogo foi tão marcante que deu origem a um gênero inteiro chamado Roguelike, que se baseia em elementos como:

- Geração procedural.
- Permadeath (morte permanente).
- Exploração estratégica.



UNIVERSIDADE
FEDERAL DO CEARÁ

Origem dos Roguelikes



Fonte: Wikipedia

Esse jogo foi tão marcante que deu origem a um gênero inteiro chamado Roguelike, que se baseia em elementos como:

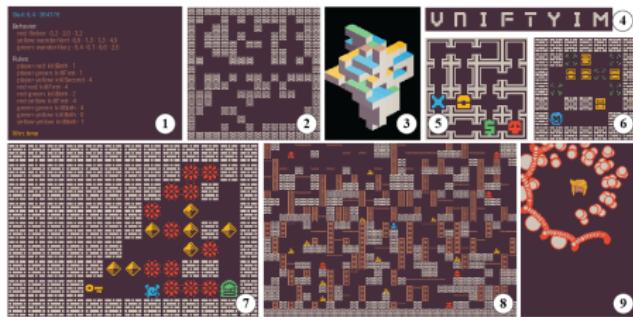
- Geração procedural.
- Permadeath (morte permanente).
- Exploração estratégica.
- Alta rejogabilidade.



UNIVERSIDADE
FEDERAL DO CEARÁ

Procedural Content Generation (PCG)

Figura: Diferentes Métodos de PCG



Fonte: antoniosliapis.com

Procedural Content Generator (PCG) é uma técnica utilizada no desenvolvimento de jogos para **criar conteúdo de forma automática, por meio de algoritmos e regras predefinidas**.

Geração de texto com LLMs

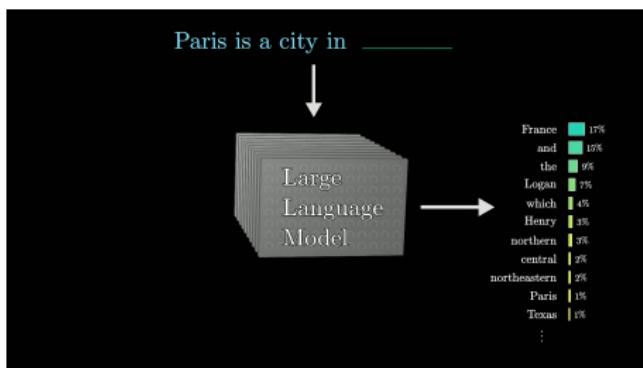
Figura: LLMs



Fonte: blog.n8n.io

Em contraste com algoritmos simples que marcam espaços bidimensionais de forma aleatória, temos a **geração de texto utilizando LLMs**.

Geração de texto com LLMs



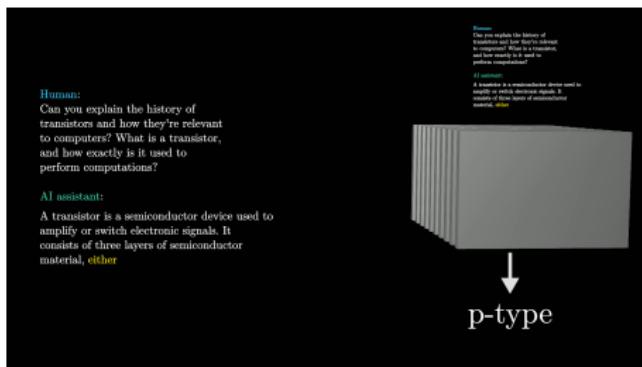
Fonte: 3Blue1Brown

Em uma visão de alto nível, LLMs são **funções matemáticas extremamente sofisticadas** que recebem um trecho de texto como entrada e retornam os tokens com maior probabilidade de continuar esse texto de forma coerente.



Geração de texto com LLMs

Figura: Geração de um Saida da LLM

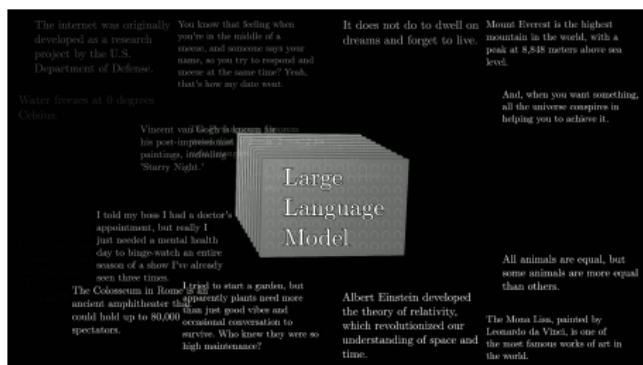


Fonte: 3Blue1Brown

Para gerar uma resposta, o texto é passado para LLM, que calcula quais palavras (ou tokens) têm maior probabilidade de vir em seguida. Uma dessas opções é **escolhida e anexada ao texto**. Esse novo texto, agora com o token adicionado, é **usado como nova entrada para a LLM**. Esse processo se **repete várias vezes** até que uma resposta completa seja construída.

Geração de texto com LLMs

Figura: Processo de Aprendizados das LLMs



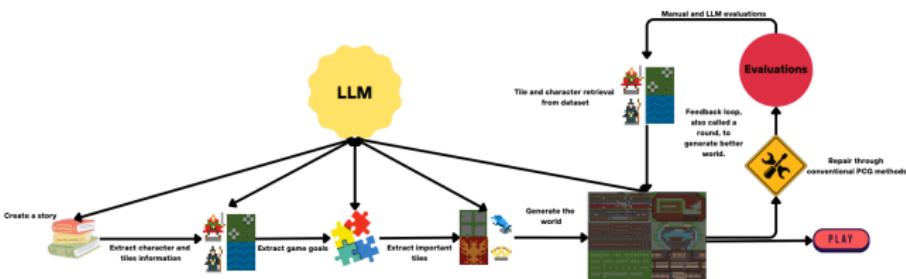
Fonte: 3Blue1Brown

Os modelos aprendem a fazer previsões através do processamento de uma quantidade **extremamente grande de texto, geralmente adquirido da internet.**



Word2World

Figura: Pipeline do Projeto Word2World



Fonte: Word2World: Generating Stories and Worlds through Large Language Models [4]

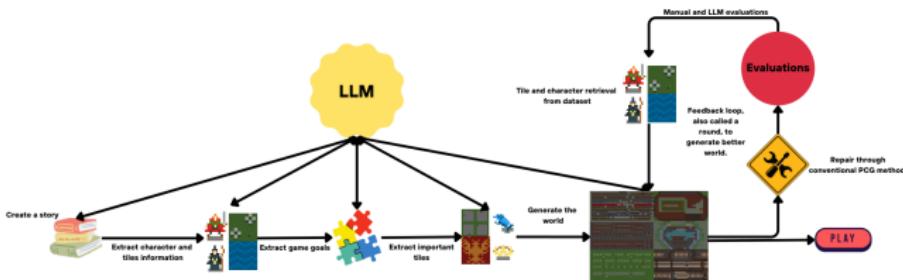
Tanto o trabalho Word2World quanto esse TCC definem um pipeline para a geração de mapas utilizando LLMs sem a necessidade de fine tune prévio.



UNIVERSIDADE
FEDERAL DO CEARÁ

Word2World

Figura: Pipeline do Projeto Word2World



Fonte: Word2World: Generating Stories and Worlds through Large Language Models [4]

A principal diferença está que no Word2World, a LLM cria a própria história, ou seja, o usuário não tem controle da geração. Além disso, o TCC visa a geração de mapas para jogos Roguelike, enquanto o Word2World não gera mapas para um gênero específico.

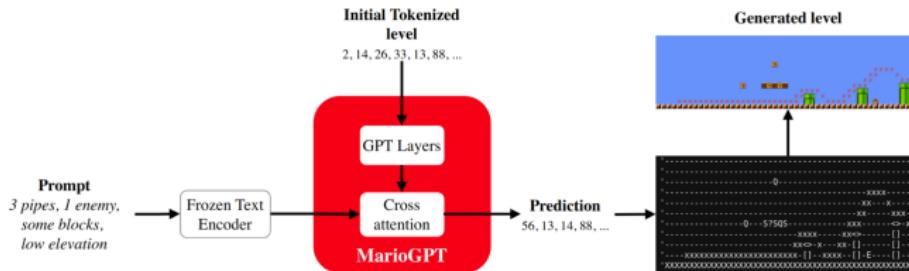


UNIVERSIDADE
FEDERAL DO CEARÁ

MarioGPT



Figura: Pipeline do Projeto MarioGPT



Fonte: MarioGPT: Open-Ended Text2Level Generation through Large Language Models[3]

MarioGPT tem o mesmo objetivo de gerar mapas baseados em descrições textuais, mas utiliza uma versão ajustada do GPT 2. A utilização de fine-tuning traz a necessidade de uma grande amostra de treino. No caso do MarioGPT foram utilizadas 200.000 amostras de mapas.

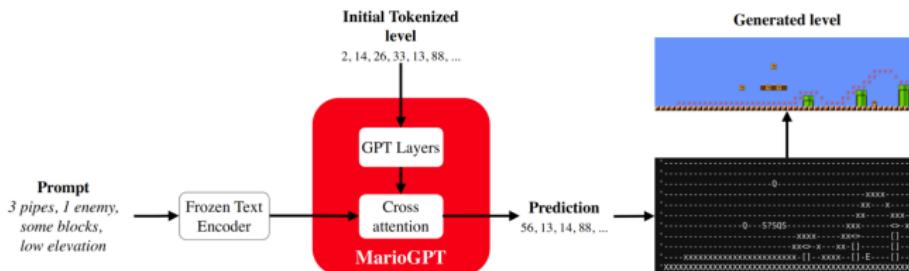


UNIVERSIDADE
FEDERAL DO CEARÁ

MarioGPT



Figura: Pipeline do Projeto MarioGPT



Fonte: MarioGPT: Open-Ended Text2Level Generation through Large Language Models[3]

A proposta do TCC remove a necessidade de fine-tuning ao se utilizar de técnicas de engenharia de prompt. Sem contar a diferença de gêneros, o MarioGPT gera mapas de platformers, enquanto o TCC se propõe

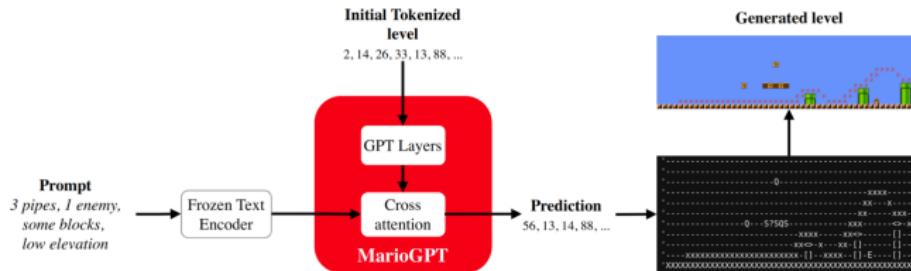


UNIVERSIDADE
FEDERAL DO CEARÁ

MarioGPT



Figura: Pipeline do Projeto MarioGPT



Fonte: MarioGPT: Open-Ended Text2Level Generation through Large Language Models[3]

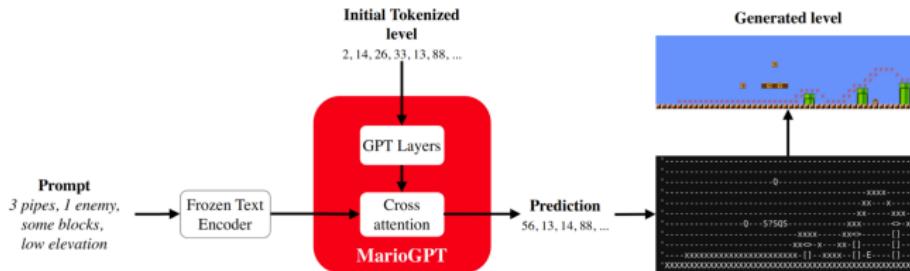
MarioGPT tem o mesmo objetivo de gerar mapas baseados em descrições textuais, mas utiliza uma versão ajustada do GPT 2. A utilização de fine-tuning traz a necessidade de uma grande amostra de treino. No caso do MarioGPT foram utilizadas 200.000 amostras de mapas.



MarioGPT



Figura: Pipeline do Projeto MarioGPT



Fonte: MarioGPT: Open-Ended Text2Level Generation through Large Language Models[3]

A proposta do TCC remove a necessidade de fine-tuning ao se utilizar de técnicas de engenharia de prompt. Sem contar a diferença de gêneros, o MarioGPT gera mapas de platformers, enquanto o TCC se propõe a gerar mapas de Roguelikes.



UNIVERSIDADE
FEDERAL DO CEARÁ

Game Environment Design Creator

Figura: Pipeline e Interface do Projeto

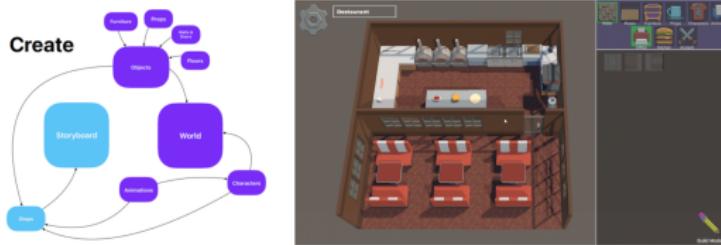


Figure 1.1: Create mode View

Fonte: Game Environment Design Creator Using Artificial Intelligence Procedural Generation [5]

O trabalho relacionado busca desenvolver uma ferramenta para auxiliar pessoas sem conhecimento de game design e programação a criarem jogos. O usuário escreve uma descrição e a LLM cria uma versão inicial da sala, o que se assemelha com o objetivo text-to-level do TCC.



UNIVERSIDADE
FEDERAL DO CEARÁ

Game Environment Design Creator

Figura: Pipeline e Interface do Projeto

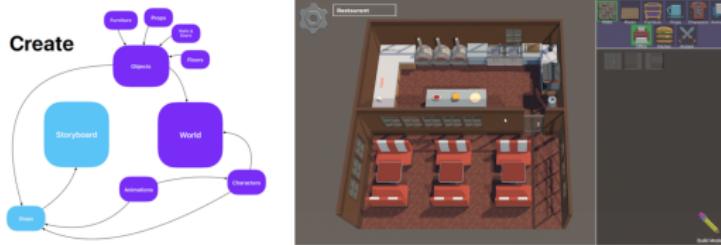


Figure 1.1: Create mode View

Fonte: Game Environment Design Creator Using Artificial Intelligence Procedural Generation [5]

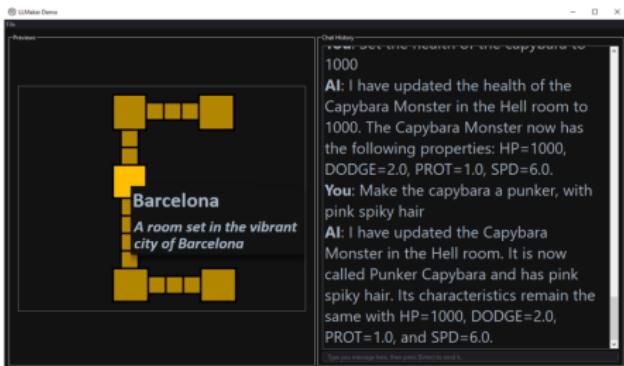
Esse trabalho relacionado foca na experiência do usuário. Enquanto esse TCC foca no pipeline de geração. Além disso, o trabalho relacionado cria pequenas salas para jogos de visão de cima e não possui um foco em um gênero específico.



UNIVERSIDADE
FEDERAL DO CEARÁ

LLMaker

Figura: Interface do LLMaker

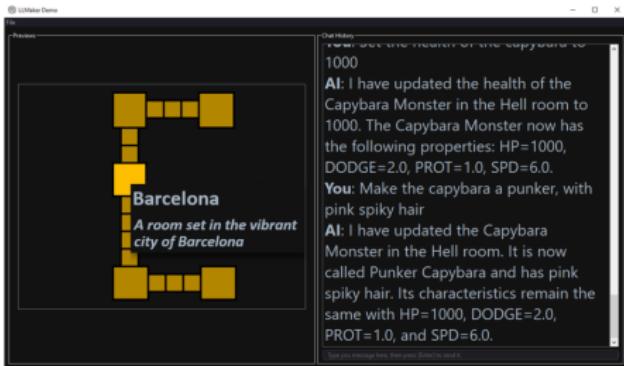


Fonte: Consistent Game Content Creation via Function Calling for Large Language Models [6]

O objetivo desse projeto é desenvolver uma ferramenta de design onde o usuário conversa em um chat LLM e então a LLM converte a linguagem natural em alterações no mapa.

LLMaker

Figura: Interface do LLMaker



Fonte: Consistent Game Content Creation via Function Calling for Large Language Models [6]

Uma semelhança ao TCC está na geração de mapas com LMMs para Dungeon Crawlers, gênero com estrutura muito parecida com os Roguelikes. Entretanto, o foco desse trabalho relacionado é converter texto do usuário e modificações no mapa gerado.

Referências I

- [1] Rafael Castro e Silva et al. "Procedural game level generation by joining geometry with hand-placed connectors". Em: (2020).
- [2] Pedro Augusto Nunes Marchand. "APLICAÇÃO DE ELEMENTOS DE STORYTELLING PARA ENTRADAS NA UTILIZAÇÃO DE REGRESSÃO PARA PREDIÇÃO DE MAPAS GERADOS PROCEDURALMENTE". Em: () .
- [3] Shyam Sudhakaran et al. "Mariogpt: Open-ended text2level generation through large language models". Em: **Advances in Neural Information Processing Systems** 36 (2023), pp. 54213–54227.
- [4] Muhammad U Nasir, Steven James e Julian Togelius. "Word2world: Generating stories and worlds through large language models". Em: **arXiv preprint arXiv:2405.06686** (2024).
- [5] André Linard Santos Auxtero. "Game Environment Design Creator Using Artificial Intelligence Procedural Generation". Diss. de mestr. Universidade NOVA de Lisboa (Portugal), 2023.



Referências II

- [6] Roberto Gallotta, Antonios Liapis e Georgios Yannakakis. “Consistent game content creation via function calling for large language models”. Em: **2024 IEEE Conference on Games (CoG)**. IEEE. 2024, pp. 1–4.

Obrigado(a) pela Atenção!

