GERAÇÃO DE NÍVEIS PARA JOGOS UTILIZANDO REINFORCEMENT LEARNING (RL)

Bruno Simões, Eduardo Silva e Pedro Antônio

Universidade Federal Rural de Pernambuco (UFRPE)

Recife (PE) - Brasil

eduardo.fsilva3@ufrpe.br, … , …

Abstract. *This article focuses on the application of Reinforcement Learning (RL) for the automatic generation of Dungeons & Dragons (D&D) maps. The study involves the development of a Q-learning-based algorithm that generates functional maps, connecting a starting point to an endpoint while passing through a predefined number of treasures. The algorithm is trained in a simulated environment where the agent learns to create rooms, corridors, and place treasures and enemies, while being rewarded for ensuring map connectivity and functionality. The results demonstrate the effectiveness of the approach, highlighting the potential of RL for automated and adaptable level generation in games. Additionally, the project emphasizes the importance of RL techniques for procedural content generation, enabling developers to create dynamic and personalized experiences for players.*

Keywords: Procedural generation. Reinforcement learning. Digital games. Maps. Dungeons & Dragons.

Resumo. *Este artigo tem como foco a aplicação de Reinforcement Learning (RL) para a geração automática de mapas de Dungeons & Dragons (D&D). O estudo envolve o desenvolvimento de um algoritmo baseado em Q-learning que gera mapas funcionais, conectando um ponto de início a um ponto de fim e passando por uma quantidade pré-definida de tesouros. O algoritmo é treinado em um ambiente simulado, onde o agente aprende a criar salas, corredores e posicionar tesouros e inimigos, enquanto é recompensado por garantir a conectividade e a funcionalidade do mapa. Os resultados obtidos demonstram a eficácia da abordagem, evidenciando o potencial do RL para a criação de níveis em jogos de forma automatizada e adaptável. Além disso, o projeto destaca a importância de técnicas de RL para a geração de conteúdo procedural, permitindo que desenvolvedores criem experiências dinâmicas e personalizadas para os jogadores.*

Palavras-chave: Geração procedural. Aprendizado por reforço. Jogos digitais. Mapas. Dungeons & Dragons.

1. Introdução

A geração procedural de conteúdo é uma área da computação amplamente utilizada no desenvolvimento de jogos digitais. Entre os diversos métodos, a utilização de Reinforcement

Learning (RL) para a geração de mapas tem se destacado pela capacidade de criar ambientes dinâmicos e adaptáveis. Este projeto implementa um algoritmo de RL para gerar mapas de Dungeons & Dragons (D&D) considerando parâmetros que serão detalhados mais a frente. O algoritmo utiliza uma abordagem de Q-learning para aprender a gerar mapas que atendem aos critérios estabelecidos.

2. Objetivo

O objetivo deste projeto é desenvolver um modelo baseado em RL que gere mapas de D&D, garantindo a conexão entre o ponto inicial e o ponto final, bem como a passagem obrigatória pelos tesouros, respeitando uma resolução predefinida. Além disso, a implementação considera a presença de inimigos e a necessidade de criar um equilíbrio entre desafios e recompensas para o jogador.

3. Metodologia

A metodologia deste projeto é baseada em Reinforcement Learning (RL), com foco na aplicação do algoritmo Q-learning para a geração procedural de mapas de Dungeons & Dragons (D&D). O ambiente de simulação é modelado como um grid bidimensional de tamanho fixo (GRID_SIZE = 15), onde cada célula pode ser modificada pelo agente para representar diferentes elementos, como paredes, salas, corredores, tesouros e inimigos. O objetivo do agente é gerar um mapa que conecte um ponto de início a um ponto de fim, passando por todos os tesouros especificados.

3.1 Modelagem do Ambiente

O ambiente é inicializado com um grid preenchido por paredes, exceto pelos pontos de início e fim, que são posicionados em cantos opostos. O agente interage com o ambiente selecionando ações para modificar células específicas do grid. As ações disponíveis incluem:

- Criar paredes: Bloqueia a célula, impedindo a passagem.
- Criar salas: Define áreas abertas que podem ser conectadas por corredores.
- Criar corredores: Conecta salas e outros elementos do mapa.
- Posicionar tesouros: Adiciona pontos de interesse que devem ser coletados.
- Posicionar inimigos: Introduz desafios adicionais ao mapa.

3.2 Linguagem de Programação

A linguagem de programação utilizada foi Python, devido à sua vasta gama de bibliotecas e frameworks que suportam o desenvolvimento de algoritmos de RL, como NumPy, Matplotlib, e Gym.

3.3 Algoritmo de Reinforcement Learning (RL)

O algoritmo de RL foi implementado utilizando Q-learning, uma técnica clássica de RL que aprende a política ótima para maximizar a recompensa acumulada ao longo do tempo. O agente é treinado para gerar mapas que conectam o ponto de início ao ponto de fim, passando por todos os tesouros. A função de recompensa foi projetada para incentivar a criação de caminhos válidos e a coleta de tesouros.

4. Implementação

4.1 Configurações Iniciais

O código define um grid de tamanho fixo (GRID_SIZE = 15) e uma lista de ações possíveis (ACTIONS = ["wall", "room", "corridor", "treasure", "enemy"]). O grid é inicializado com paredes, e os pontos de início e fim são posicionados nos cantos opostos do grid.

4.2 Q-learning

O algoritmo de Q-learning é utilizado para treinar o agente. A Q-table é inicializada com valores aleatórios e é atualizada a cada passo do treinamento. O agente escolhe ações com base em uma política ε-greedy, que balanceia exploration e exploitation.

4.3 Função de Recompensa

A função de recompensa foi projetada para guiar o agente na criação de mapas funcionais e interessantes. Recompensas são atribuídas com base em critérios como:

- Conectividade: Recompensas por conectar salas a corredores e garantir que o caminho do início ao fim seja contínuo.
- Coleta de tesouros: Incentivos para posicionar tesouros ao longo do caminho principal.
- Eficiência do caminho: Recompensas por criar caminhos diretos e evitar redundâncias.

Além disso, o algoritmo verifica a validade do mapa por meio de uma busca em largura (BFS), que confirma se todos os tesouros estão conectados ao caminho principal e se o início e o fim estão devidamente ligados.

4.4 Verificação de Caminho Válido

O algoritmo verifica a validade do caminho por meio de uma busca em largura (BFS), que confirma se todos os tesouros estão conectados ao caminho principal e se o início e o fim estão devidamente ligados.

4.5 Renderização do Mapa

O mapa gerado é renderizado utilizando a biblioteca Matplotlib. Cada tipo de célula (parede, sala, corredor, tesouro, inimigo, início, fim) é representada por uma cor específica.

5. Resultados

O algoritmo foi capaz de gerar mapas que atendem aos critérios especificados, demonstrando a viabilidade do uso de RL para a geração de níveis em jogos. Os mapas gerados foram avaliados tanto qualitativamente quanto quantitativamente, com base na eficiência do caminho e na distribuição dos tesouros.

6. Conclusão

Este projeto demonstra o potencial do uso de Reinforcement Learning para a geração automática de níveis em jogos, especificamente mapas de Dungeons & Dragons. O código desenvolvido é flexível e pode ser adaptado para diferentes tipos de jogos e cenários, oferecendo uma ferramenta poderosa para desenvolvedores de jogos.

7. Referências

SUTTON, R. S.; BARTO, A. G. Reinforcement Learning: An Introduction. MIT Press, 2018.

MNIH, V. et al. Human-level control through deep reinforcement learning. Nature, 2015.

BROCKMAN, G. et al. OpenAI Gym. arXiv preprint arXiv:1606.01540, 2016.