

Atividade: Jogo da Velha

Descrição:

Desenvolva um jogo da velha utilizando React para a interface de usuário e Redux para o gerenciamento de estado. O objetivo é criar uma aplicação onde dois jogadores possam jogar o jogo da velha, com a interface sendo controlada pelo React e toda a lógica de jogo (como o estado do tabuleiro, controle de turnos e verificação de vitória/derrota) sendo gerenciada pelo Redux.

Requisitos:

Estrutura do Projeto: O projeto deve ser estruturado de forma a separar claramente a interface (React) da lógica de negócio (Redux). Os componentes React devem ser responsáveis apenas pela exibição e interação do usuário (Clicar na posição X da Matriz para realizar sua jogada).

Estado Global com Redux: Utilize Redux para gerenciar o estado global do jogo. Isso inclui o estado do tabuleiro (posição das marcas 'X' e 'O'), o jogador atual e o status do jogo (em andamento, vitória, empate).

Ações e Redutores: Defina ações apropriadas para as diferentes interações do jogo (como fazer uma jogada ou reiniciar o jogo). Implemente redutores para atualizar o estado do jogo com base nas ações disparadas.

Interface do Usuário: Crie uma interface de usando React, incluindo o tabuleiro do jogo e uma indicação de qual jogador é o próximo, além de mostrar mensagens de vitória, derrota ou empate.

Log de Jogadas: Implemente um histórico de jogadas, permitindo que os usuários vejam as jogadas anteriores e o estado do tabuleiro em cada ponto do jogo.

Controle de Turnos: Garanta que o jogo siga a ordem de turnos corretamente e evite que um jogador faça duas jogadas consecutivas.

Determinação de Vitória ou Empate: Implemente a lógica para determinar o vencedor do jogo ou um empate, atualizando o estado global através do Redux.

Dicas:

Você pode representar a matriz (3x3) da seguinte forma:

```
import React from 'react';

export function App(props) {

  const matriz = [
    ['O', 'O', 'O'],
    [' ', 'X', ' '],
    [' ', 'X', ' ']
  ];

  return (
    <div className="App">
      {matriz.map((linha, linhaIndex) => (
        <div key={linhaIndex}>
          {linha.map((celula, celulaIndex) => (
            <span key={celulaIndex} onClick={() => console.log(linhaIndex, celulaIndex)}>{celula} </span>
          ))}
        </div>
      ))}
    </div>
  );
}
```

Porém é livre a escolha de como será a lógica da interface.