

Relatório Linguagens Script – “4 em linha Especial”



Figura 1 – Interface do jogo 4 em linha

> Resumo

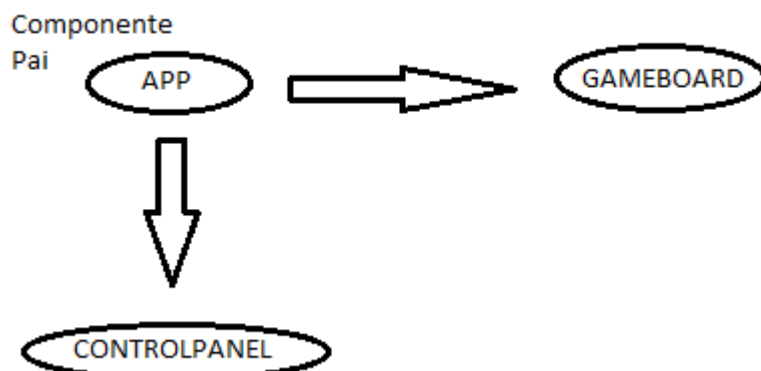
Este trabalho prático tem como objetivo o desenvolvimento de uma aplicação em **React JS**, na qual os alunos devem aplicar todos os conhecimentos adquiridos ao longo das aulas. Como tal, devem demonstrar o domínio na tecnologia React assim como das tecnologias necessárias ao desenvolvimento uma solução coerente e consistente, nomeadamente JavaScript, HTML e CSS.

> Equipa de trabalho

O trabalho prático foi realizado pela seguinte equipa:

- Guilherme Rico
- Micael Santos

> Diagrama de Componentes



> Componentes

→ App (Component Principal): Trata do estado geral do jogo (Modo de jogo, Jogadores e Tempo), renderiza condicionalmente diferentes etapas do jogo (Seleção de modo de jogo, Inserção de nomes dos jogadores e Tabuleiro de jogo e painel de controlo) e passa o *props* para os componentes filhos

- GameModeSelection: Exibe os botões para selecionar o modo de jogo (1vs1 ou vsCPU), mostra informações sobre os desenvolvedores, atualiza o estado *gameMode* no componente App
- PlayerNameInput: Guarda os nomes dos jogadores, faz com que o jogo comece depois de meter um nome para o jogador e atualiza o estado *players* no componente App

→ ControlPanel;

- Barra Superior: Botão “Alterar modo de jogo” que faz com que volte à seleção inicial e o Temporizador visual mostra o tempo restante e fica vermelho quando passar dos 3s.
- Barra de Status: Mostra o modo de jogo atual (1vs1 ou vsCPU) e Mostra os nomes dos jogadores (ou “CPU” no modo vsCPU)
- Gestão de Estado: Manipula os estados *gameMode* e *gameStarted* do componente App.

→ GameBoard;

- Startup do Tabuleiro: Cria um tabuleiro 6x7 e cria aleatoriamente 5 células especiais;
- Lógica de Jogo: Gerencia o estado do tabuleiro e peças dos jogadores, faz os turnos entre jogadores, implementa temporizador de 10 segundos por jogada e deteta vitórias usando padrões pré-definidos;
- Modo vs CPU: Implementa IA básica com jogadas aleatórias e respeita o tempo de espera para simular “pensamento”;

- Células Especiais: Permitem ao jogador jogar novamente se cair em uma célula especial;
- Renderização: Cria 42 células (6 linhas x 7 colunas) com classes dinâmicas;
- Cell: Representa cada célula do tabuleiro e tem classes dinâmicas para o jogador atual (player1/player2), células especiais e estado vazio/ocupado;

> **Limitações conhecidas**

Durante o desenvolvimento do jogo “4 em Linha” em React, deparamo-nos com algumas limitações técnicas e desafios como por exemplo:

- **Temporizador:** A implementação do temporizador de 10 segundos por jogada foi mais complexo do que estávamos à espera quando toca à sincronização com o estado do jogo, especialmente quando combinada com a lógica das células especiais que permitem jogadas consecutivas.
- **IA da CPU:** A inteligência artificial implementada para o modo contra o computador segue uma abordagem aleatória simples, sem coisas avançadas ou conseguir prever jogadas, o que limita o desafio para jogadores mais experientes.
- **Responsividade:** O tabuleiro foi desenvolvido com dimensões fixas (6x7), o que talvez limite a adaptação a diferentes tamanhos de ecrã sem ajustes adicionais.
- **Compatibilidade:** O jogo foi testado principalmente em navegadores mais recentes, e pode apresentar comportamentos diferentes em versões mais antigas.
- **Performance:** Em dispositivos móveis com menos capacidade de processamento, pode ocorrer pequenos atrasos na renderização durante as animações das jogadas.

> **Desafios**

O desenvolvimento deste projeto apresentou vários desafios interessantes:

- **Lógica do Jogo:** Implementar as regras completas do "4 em Linha", tipo a deteção de vitórias em todas as direções possíveis (horizontal, vertical e diagonais) precisou de um algoritmo cuidadoso para verificar todas as combinações.
- **Células Especiais:** A implementação das células especiais que permitem jogadas consecutivas acrescentou complexidade à gestão de turnos e à lógica do jogo.
- **Temporizador Integrado:** Coordenar o temporizador com os estados do jogo e as transições entre jogadores e o computador foi um bocado desafiador.
- **Modo CPU:** Criar um comportamento razoável para o jogador automático, mesmo que básico, precisou da implementação de um sistema de pausas para simular "pensamento".
- **Gestão de Estado:** Manter sincronizados os múltiplos estados do jogo (tabuleiro, jogadores, tempo, modo de jogo) em diferentes componentes foi um desafio constante.

- **Testes:** Garantir que todas as combinações de jogadas fossem testadas adequadamente, especialmente nos casos de vitória e células especiais, precisou da criação de vários cenários de teste.