



Trabalho Prático Linguagens Script 2023/2024

Minesweeper



Realizado por:

- Rodrigo Bernarda nº 2021136740 email: a2021136740@isec.pt
- Gonçalo Fontes nº 2021144224 email: a2021144224@isec.pt
- Bruno Correia nº 2015015880 email: a21250666@isec.pt

Coimbra, 02 de junho de 2024

Índice

1. Introdução	3
2. Diagrama de Componentes	4
3. Funcionalidades implementadas	7
4. Limitações conhecidas e Desafios	10
5. Webgrafia	10

Introdução

Neste relatório, abordaremos detalhadamente o projeto que desenvolvemos na unidade curricular de “Linguagens Script”, que consiste no tradicional jogo Minesweeper. Ao longo de relatório serão exploradas as funcionalidades implementadas, o diagrama de componentes que constituem o projeto, bem como serão abordados os desafios e limitações que tivemos ao longo do projeto.

Diagrama de Componentes

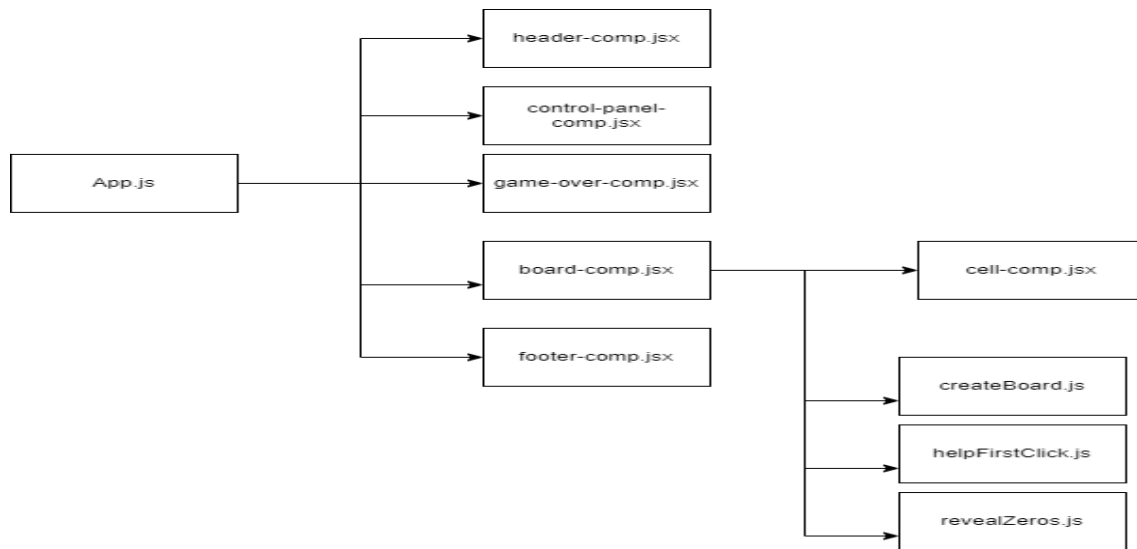


Fig.1 – Diagrama de Componentes usado no projeto

Para a realização deste projeto foram utilizadas ao todo 6 componentes funcionais, mais 3 funções para assegurar o bom-funcionamento dos mesmos, sendo estas:

- **Header-comp.jsx:** Esta primeira componente tem a simples tarefa de mostrar o header.
- **Control-panel-comp.jsx:** Esta componente é responsável por mostrar as informações do jogo, tais como o da partida e o número de minas a identificar, e pela seleção do nível e iniciar da partida.
- **Game-over-comp.jsx:** Aparecendo apenas quando uma partida termina em vitória/derrota, esta componente é responsável por mostrar a tela de fim de jogo, onde é mostrado o resultado, bem como o tempo usado, a pontuação e a nota a si associada.
- **Board-comp.jsx:** Esta é a componente principal do projeto, aqui é onde estão presentes as interações necessárias para o funcionamento do jogo, tais como a criação do tabuleiro, a colocação de bandeiras/pontos de

interrogação, identificar vitória/derrota, revelar as células do tabuleiro, entre outras funcionalidades.

- **Cell-comp.jsx:** Utilizada pela componente *board-comp.jsx*, aqui são constituídas as células do tabuleiro e onde ajuda a identificar *clicks* com o botão esquerdo e direito nas células.
- **Footer-comp.jsx:** À semelhança do *header-comp.jsx* esta componente mostra simplesmente o footer utilizado no projeto.

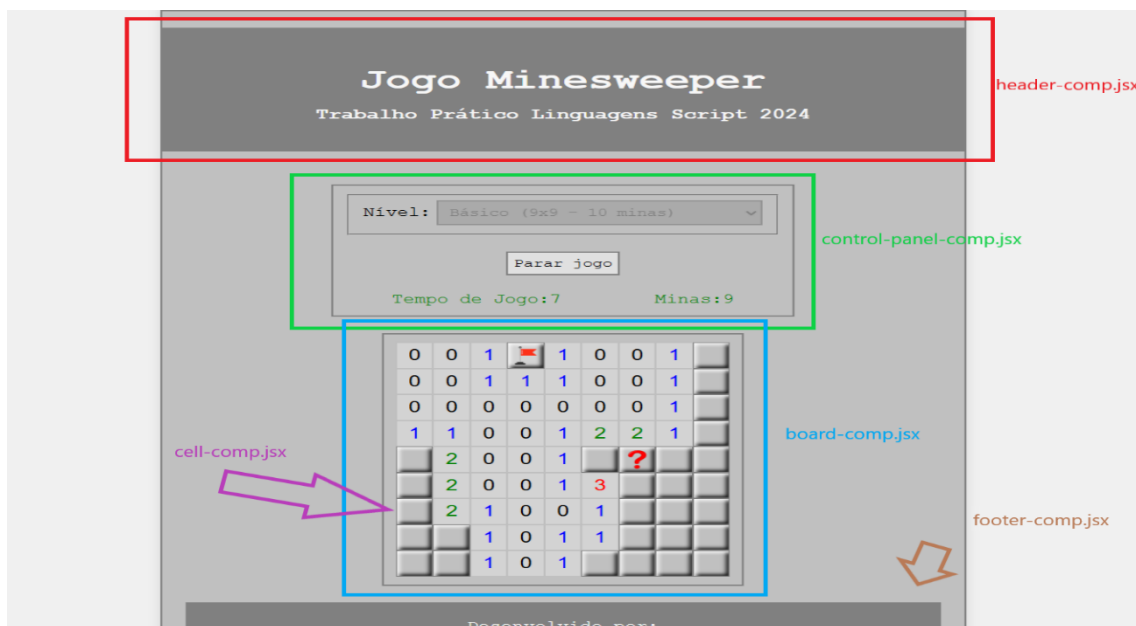


Fig. 2 - Representação visual das componentes utilizadas



Fig.3 – Representação visual da componente game-over-comp.jsx

Como mencionado, foram também utilizadas 3 funções à parte que contribuem para o bom funcionamento do jogo, sendo estas:

- **createBoard.js:** Esta função é utilizada para criar o tabuleiro de jogo, sendo capaz de distribuir aleatoriamente as minas pelo tabuleiro e de atribuir o número de minas adjacentes às células sem minas.
- **revealZeros.js:** Esta função é chamada quando uma célula com o valor 0 é selecionada, revelando todas as células vizinhas com minas adjacentes.
- **helpFirstClick.js:** Esta função é responsável por garantir a impossibilidade de perder o jogo ao primeiro *click*, reposicionando a mina selecionada num sítio aleatório do tabuleiro, atualizando depois o tabuleiro.

Funcionalidades Implementadas

Neste capítulo serão abordadas as funcionalidades implementadas nesta aplicação, bem como as soluções utilizadas para os problemas.

Como referido anteriormente, a componente funcional *control-panel-comp.jsx* é capaz de iniciar jogo, selecionar nível e mostrar o tempo de jogo e as minas a identificar. Para tal, são utilizadas funções presentes no *App.js*, como a *handleGameStart*, responsável por iniciar jogo, *handleLevelChange* que garante a mudança correta de nível que é selecionada dentro da componente *control-panel-comp.jsx*, um *useEffect* que serve de contador de tempo de jogo, e por fim a função *gameInfo* que, recebendo da componente *board-comp.jsx*, fornecerá as informações do jogo a decorrer, como o resultado atual, o número de não minas por revelar e as minas já identificadas com a bandeira, que servirão de informação para as componentes funcionais *control-panel.jsx* (minas identificadas) e *game-over-comp.jsx* (não minas por revelar).

Nas componentes funcionais *board-comp.jsx* e *cell-comp.jsx*, estão presentes todas as funções necessárias para o funcionamento correto do jogo. Cada célula tem 3 propriedades: *revealed*, que indica se a célula já foi clicada com botão esquerdo ou não, *flag*, que indica o estado da célula (0 significa estado normal, 1 se tem bandeira e 2 se tem um ponto de interrogação) e *value* que conterà o valor da célula (número de minas adjacentes ou a mina). Como já referido, a componente *board-comp.jsx* contém as funções necessárias para o funcionamento do jogo, como os 2 *useEffects* responsáveis pela constituição inicial do jogo, a função *updateRightClick* que permite colocar bandeiras ou pontos de interrogações nas células do tabuleiro e a função *updateReveal* que permite revelar corretamente as células, identificar vitória/derrota, e com ajuda das funções auxiliares *revealZeros.js* e *helpFirstClick.js* consegue dar resposta a situações inusitadas em que se clica numa célula vazia ou numa célula com mina

na primeira jogada. Para identificar vitória é utilizada o *useState nonMineFound* que conterà inicialmente o número total de não minas por encontrar, e à medida que se vai clicando em células sem minas, *nonMineFound* decrementará até chegar a 0, o que significa que o jogador venceu o jogo. A função *updateReveal* não permite *clicks* quando o jogo ainda não foi iniciado e a células com bandeira/ponto de interrogação ou que já foram reveladas (essencial para impedir batota). A função *updateRightClick* tem um comportamento semelhante ao não permitir *clicks* com o botão direito quando o jogo ainda não foi iniciado ou quando uma célula já foi revelada (essencial para garantir que o número de minas por identificar seja sempre correto).



Fig. 4 – Representação visual dos diferentes estados das células

Por fim, a componente *game-over-comp.jsx* é capaz de mostrar a tela de fim de jogo, sendo chamada apenas quando se dá vitória/derrota, recebendo como parâmetros o resultado, o nível selecionado, tempo de jogo e o número de não minas que ficaram por revelar, que serão utilizados para calcular a pontuação à performance do jogador, dando também uma classificação associada à pontuação. É capaz também de iniciar jogo novamente.



Fig.5 – Representação visual da tela de fim de jogo em caso de vitória

Limitações conhecidas e Desafios

Em geral, a implementação do jogo foi bem-sucedida, não havendo nenhum problema grave por registrar, porém ao executar a aplicação é possível inicialmente criar jogo sem um nível selecionado, mas quando um nível é selecionado, já não é possível iniciar jogo sem um nível selecionado.

O desenvolvimento deste projeto foi marcado por grandes desafios, uns mais fáceis e agradáveis e outros mais complicados, tendo sido estes a criação do tabuleiro e a garantia de impossibilidade de derrota ao primeiro *click*. No geral, sentimo-nos satisfeitos com o trabalho desenvolvido neste projeto, pois foi uma ótima oportunidade de aprendizagem de uma biblioteca tão essencial nos dias de hoje.

Webgrafia

[Minesweeper in 100 minutes - React JS Game \(youtube.com\)](https://www.youtube.com/watch?v=...)