Trabalho Prático em React – "Ultimate Tic-Tac-Toe"

> *Âmbito*

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 nas tecnologias necessárias ao desenvolvimento de uma solução coerente e consistente, nomeadamente JavaScript, HTML e CSS.

Os alunos devem organizar-se, obrigatoriamente, **em grupos de 2 ou 3 alunos.** Qualquer alteração a esta regra, deverá ser previamente acordada com a docente responsável pela disciplina, através do endereço de email cris@isec.pt. Todos os alunos devem colaborar na implementação do trabalho, e na implementação de componentes React, sob pena de, na defesa, lhes ser atribuído 0 na nota do trabalho. Encontra-se um formulário no *inforestudante*, secção "Submissão de Trabalhos", para especificarem a constituição do grupo de trabalho. Esta submissão é obrigatória e deverá ser efetuada até ao **dia 14 de maio**. Apenas é necessário um dos elementos do grupo especificar o grupo de trabalho no formulário, devendo, no entanto, associar os colegas na opção disponível no sistema (conforme figura abaixo). O não cumprimento das regras detalhadas no enunciado, implica penalização na nota final do trabalho prático. Para o ficheiro exigido na submissão da informação do grupo, devem submeter um ficheiro com o nome dos elementos do grupo.

Todos os detalhes do trabalho encontram-se especificados nas secções seguintes e, como tal, **sugere-se a leitura com a devida atenção** sob pena de não implementarem o trabalho na sua totalidade, por descuido. Recorda-se que o trabalho tem um peso de **8 valores** da nota final da disciplina, como especificado na FUC de Linguagens Script.

Valoriza-se a atuação autónoma do aluno na construção de uma solução para o tema proposto, tendo em consideração que esta deve cumprir, no mínimo, os requisitos gerais apresentados. **Submissão de trabalhos da autoria de terceiros, para além de 0 na nota do trabalho prático, serão ativados os devidos processos legais de plágio, aplicados no ISEC/IPC.**

Tema

O tema do trabalho prático é uma variante do tradicional jogo do galo, designado como **"Ultimate Tic-Tac-Toe",** no qual devem ser efetuadas jogadas alternadas entre dois jogadores ou um jogador e o computador até se encontrar um vencedor ou um empate.

O jogo está organizado por 9 mini-tabuleiros do jogo do galo, dispostos numa grelha 3x3, sendo que o **vencedor do jogo** é aquele que fizer 3 em linha (seja na horizontal, vertical ou diagonal) **do tabuleiro geral**. Para isso, o jogador deve ganhar mini-tabuleiros, efetuando 3 em linha em cada mini-tabuleiro e dessa forma tentar a vencer o jogo, com 3 em linha no tabuleiro geral. Em caso de empate no tabuleiro geral, a vitória pertence ao jogador com mais mini-tabuleiros vencidos, informação essa que deverá ser apresentada no final do jogo.

As figuras seguintes **apresentam exemplos do jogo** a implementar, não estando nessas figuras a opção de seleção do nível de jogo e outras das funcionalidades referidas na secção seguinte.

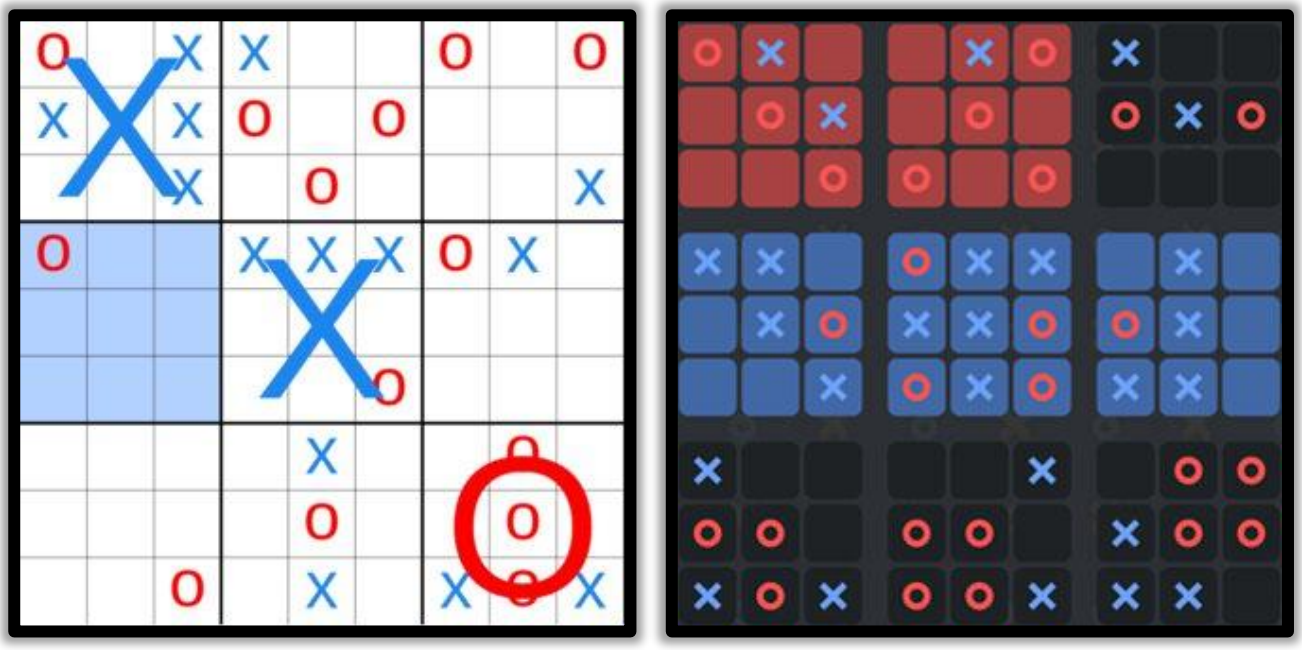


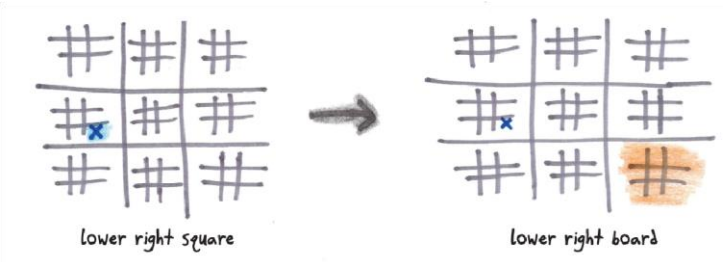
Figura 1 – Exemplos do Jogo Ultimate Tic-Tac-Toe

Funcionalidades

O jogo a implementar deverá ser implementado em **React**, acessível através do browser e deverá disponibilizar, pelo menos, as seguintes **funcionalidades**:

- Solicitar o nome dos jogadores;
- Decidir de forma aleatória o primeiro jogador bem como o símbolo associado, e apresentar essa informação;
- Apresentação do tabuleiro geral de jogo;

- Identificação clara do jogador que deve efetuar a jogada, seja por cor ou outra forma que o aluno desejar;
- Permitir a seleção da posição no mini-tabuleiro seja com o rato ou então navegando com o teclado pelas células disponíveis;
- A escolha do mini-tabuleiro em que se joga:
    - **Opção 1:** é livre, isto é, a posição escolhida pelo adversário pode ser no mesmo mini-tabuleiro ou de outro qualquer à escolha, não dependendo da jogada anterior do adversário desde que ainda esteja livre.
    - **Opção 2 (mais valorizada):** não é livre, sendo determinada pela jogada anterior do adversário. Isto é, a posição escolhida pelo adversário identifica o mini-tabuleiro onde deve ser continuado o jogo, como no exemplo seguinte:



Neste exemplo vê-se que o jogador azul colocou uma peça na posição (3,3) do mini-tabuleiro selecionado para a sua jogada. Na próxima jogada, o oponente tem que colocar uma peça no mini-tabuleiro que se encontra nesta posição na área global do jogo. Pode dar-se o caso do mini-tabuleiro escolhido para continuar o jogo já estar terminado, i.e., já ter um vencedor. Se isso acontecer, fica ao critério do aluno delinear uma regra para determinar o próximo tabuleiro em que continua o jogo.

- Identificação do vencedor de um mini-tabuleiro, seja com cores (como apresentado na figura 1) ou outra alternativa que o aluno desejar;
- Quando um mini-tabuleiro ficar fechado em termos de vitória ou empate, este deverá ficar bloqueado, isto é, sem permitir novas jogadas/navegação.
- **2 níveis de jogo** em que a forma de jogar varia de acordo com os jogadores em jogo, nomeadamente: **1 contra 1 jogador** e 1 **contra computador (Não se pretende um algoritmo elaborado, apenas um valor aleatório nas células disponíveis)**;
- Tempo de jogo limitado, sendo o vencedor quem tiver mais vitorias dos mini-tabuleiros;

- → Identificação de fim de jogo, quando todos os mini tabuleiros estiverem concluídos, assim como o vencedor (Como já referido, em caso de empate, o vencedor é o jogador com mais mini-tabuleiros vencidos);
- → Permitir jogar novamente;

› *Implementação*

A implementação do trabalho, como anteriormente referido, **deverá ser efetuado em React**, recorrendo à versão 18. Em relação à implementação, é dada a liberdade ao aluno para seguir a estratégia que considerar melhor, no entanto, especificam-se abaixo algumas características e limitações que devem ter em consideração.

→ Método

Os alunos **devem criar um projeto** de forma a facilitar a gestão de dependências de todos os módulos necessários à implementação de uma aplicação react. Assim, recomenda-se o uso da aplicação creact-react-app para criação do projeto, mas é dada liberdade aos alunos para usarem outra abordagem, desde que devidamente explicada no relatório.

Em relação à criação dos componentes, o React permite a implementação dos componentes usando abordagens diferentes, seja *functional components* (em conjunto com hooks) ou *class components*, no **entanto, o trabalho deverá ser implementado, na sua maioria, com componentes funcionais.** Os alunos que assim o desejarem, poderão também explorar componentes de classe (alerta-se, no entanto, que quem seguir esta opção, não será considerada como extra na nota do trabalho prático).

→ Interface

Fica à responsabilidade do aluno a especificação da interface da aplicação. Esta não tem de ter o aspeto do exemplo apresentado nas figuras anteriores. Deve haver uma preocupação no sentido de desenvolver uma aplicação visualmente agradável com uma imagem consistente. Poderão ser usadas bibliotecas e *frameworks* CSS que considerarem úteis e possam facilitar a implementação da interface, como por exemplo, Bootstrap, Tailwind CSS, entre outros. No entanto, **não é permitido** recorrer a bibliotecas e/ou frameworks que disponibilizem componentes React já implementados, como por exemplo, *Material UI*, *React Bootstrap*,

Sematic UI, *React Toolbox*, *Ant Design*, *React Foundations,* entre outros. Todos os componentes necessários à implementação do jogo devem ser implementados pelos alunos.

Nesse sentido, sempre que o aluno utilize componentes previamente implementados por terceiros (bibliotecas/frameworks, entre outros), será considerado que o aluno não implementou esse componente nem as respetivas funcionalidades.

> *Relatório*

O trabalho prático deve ser acompanhado de um pequeno relatório técnico (que não ultrapasse as 10 páginas) onde devem **apresentar o diagrama de componentes**, especificar as funcionalidades implementadas bem como as soluções utilizadas no desenvolvimento da aplicação. Além disso, devem ser especificados os componentes, ou outros elementos que tenham sido utilizados no desenvolvimento da aplicação.

> *Condições e Data de Entrega*

A **data limite** para entrega do trabalho prático **é o dia 15 de Junho de 2023, quinta-feira, até às 23:59.** A entrega do trabalho prático é realizada em formato digital, **no *inforestudante*, apenas por um elemento do grupo**, devendo ainda especificar no momento da submissão os outros elementos do grupo de trabalho. O trabalho prático deve ser submetido num ficheiro **ZIP** com o nome **`LS2122_numAluno1_numAluno2_NumAluno3.ZIP`**. **Trabalhos práticos submetidos depois do horário terão uma penalização de 5% da nota final, por cada hora.**

NOTA IMPORTANTE: Antes de criar o ficheiro ZIP, os alunos devem **obrigatoriamente** remover a pasta "*node_modules*" e o ficheiro "*package-lock.json*" existente no estrutura do projeto. Para além do projeto implementado, que deve incluir o código fonte (sem a pasta), o ficheiro Zip deve incluir o relatório em formato pdf.

Recomenda-se que os alunos testem o trabalho que submetem, sob pena, de não entregarem a versão correta ou pretendida, como frequentemente acontece….

Defesa

O trabalho terá uma defesa obrigatória, que será efetuada **em data a definir**, através de uma pré-inscrição no *inforestudante*. **A não comparência** à defesa na data (dia e hora) combinada, implica reprovação do aluno à disciplina.

Critérios de Avaliação

- Aspeto Gráfico (15% da percentagem total do trabalho);
- Apresentar nome do jogador e símbolo associados (5%);
- Identificação clara dos jogadores em cada jogada alternando o respetivo símbolo (10%)
- Seleção das células dos mini-tabuleiros (40% na opção 2):
  - ***(opção 1)*** Permitir jogar de forma livre e bloquear quando um mini-tabuleiro concluído;
  - ***(opção 2 – mais valorizada)*** Próxima Jogada determinada pela jogada anterior, bloqueando as outras células e bloquear quando um mini-tabuleiro concluído;
- Identificação do vencedor do mini-tabuleiro, seja com cores ou como o aluno desejar (10%)
- Identificação de fim de jogo com temporizador ou com mini-tabuleiros concluídos apresentando vencedor (10%)
- 2 níveis de jogo em funcionamento (10%)
- ***Critérios gerais a aplicar à nota final:***
  - Organização do código e metodologia usada na implementação da aplicação, nomeadamente, forma de estruturação da aplicação em vários componentes React, o uso correto de hooks (quando aplicável) na aplicação, entre outros.
  - Defesa do aluno.