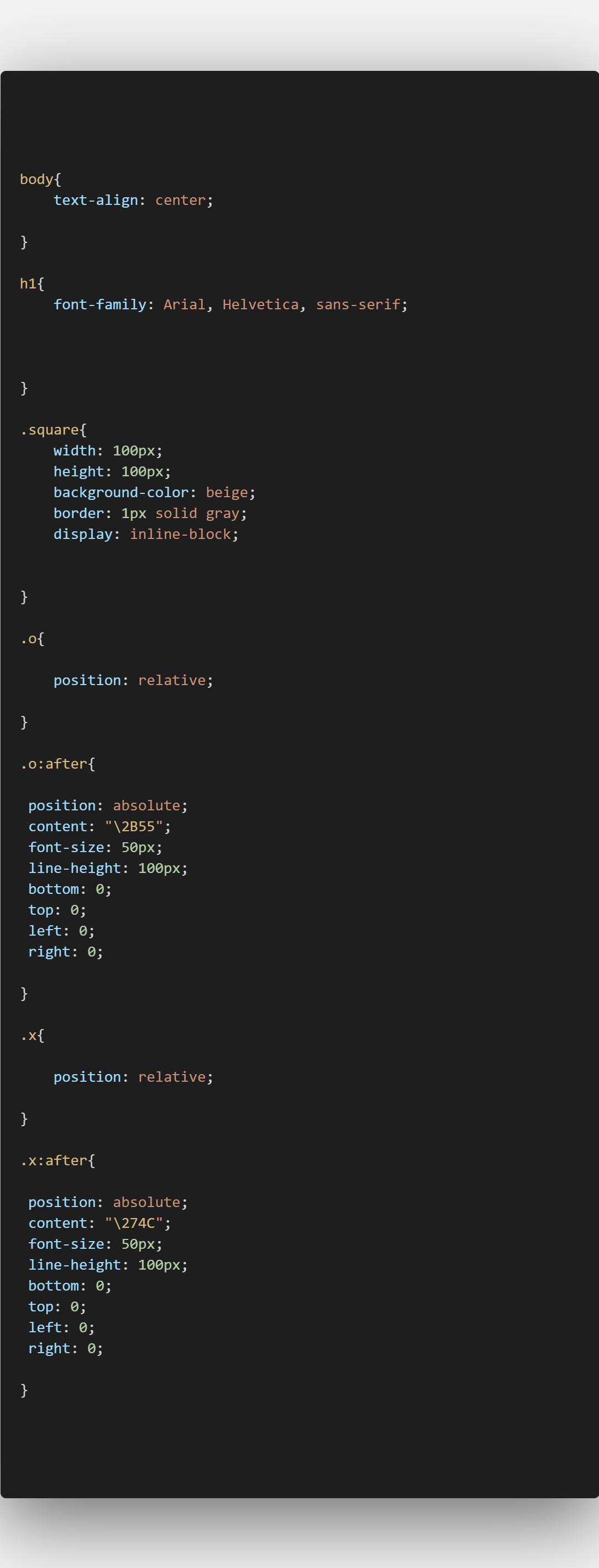
**HTML E CSS**



Quando usamos <div></div>, automaticamente ela pula uma linha para a próxima, o que facilitou para criar o formato de quadrado do tabuleiro



Peguei a imagem do “x” e do “o” no emojipedia, usando o “codepoint” da figura.

:after é um psedo-elemento que estiliza o ultimo filho do elemento selecionado. Dando a ele um “content”, que nesse caso é a imagem do x ou o.

O inline-block foi usado para que as 3 divs principais pudessem formar um quadrado maior.

A posição do pai deve ser relative, para que a posição do filho se baseie por ela.

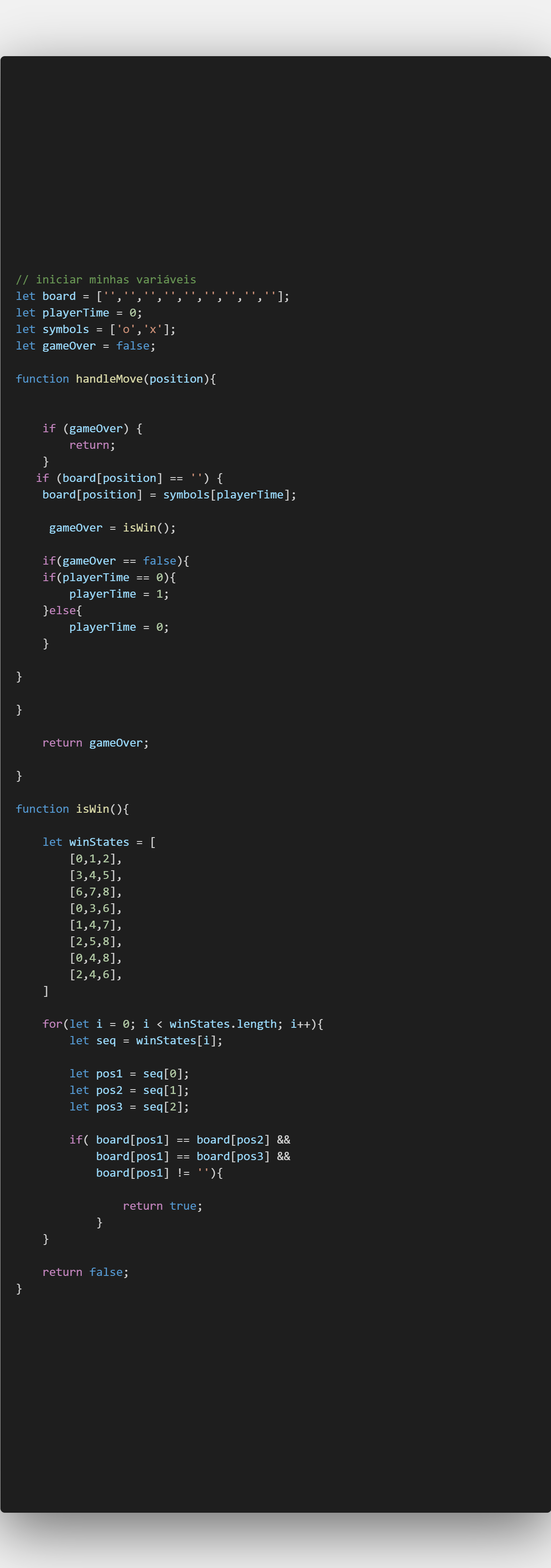
O bottom, top, left, right, foram estilizados com 0, para o elemento se concentrar no meio do quadrado.

**JAVASCRIPT**

Neste projeto, dividi o arquivo Javascript em 2. O primeiro chamado de game.js é o que vai lidar com a lógica do jogo, esse tipo de método também é chamado de regra do negócio, coisas que não tem a ver com a interface gráfica.

O segundo arquivo se chamara interface.js, que terá a função de fazer a união entre o html e o css e o game.js.

**Game.js**



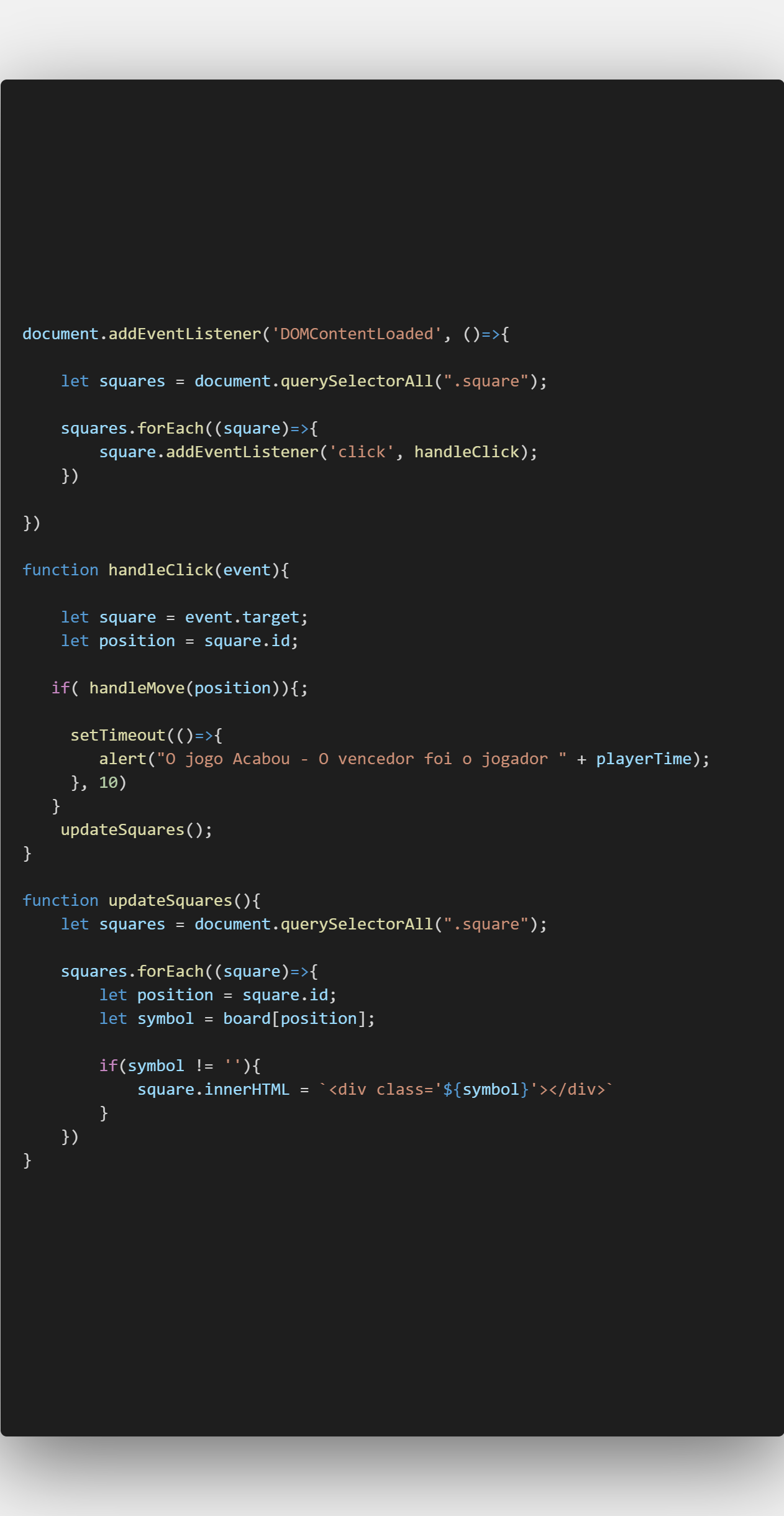
Nesta função o [position] e o [playetime], estão sendo usados como o index do array do symbols e board.

Aqui em symbols, usamos “o” e “x”, que anteriormente definimos com as imagens do emojipedia, serão esses dois que preencheram os espações vazios no board

O playerTime define a quantidade de jogadores e a vez do jogador em questão

Este array, chamado board, representa todas as 9 posições do tabuleiro, que podem ser: x, o ou vazio.

**INTERFACE.JS**

****

Como o arquivo interface foi carregado depois do arquivo game, o interface tem acesso a todas as variáveis dos arquivos html, css e do próprio game

Esta função é respondável por inserir as figuras dentro dos quadrados escolhidos

Usamos id, para identificar a posição que está sendo jogada naquele momento.

O forEach(), chama uma função para todos os elementos de um array, neste caso, para todos os quadrados do tabuleiro.

O querySelectorAll() retorna todos os seletores css desejados, neste caso, todos as class=”square”. Se quiséssemos apenas um seletor específico, usaríamos querySelector().

Este é um evento que permite que o documento HTML seja carregado imediatamente ao JS, assim conseguimos identificar a ação do usuário na hora que ele a faz.