Função Automática (Auto)

Código da Função Automática:

Função Automática

Função Automática:

Como funciona:

A função "auto()" começa removendo a opacidade ("desligado") da luz vermelha e em seguida a atribui às luzes verde e amarela. Posteriormente a função define um loop usando 'setInterval()'. Esse loop irá executar uma sequência que alterna as cores do semáforo. Dentro desse loop, há uma sequência de ações para controlar o estado do

Primeiro, a função torna o vermelho e o amarelo desligados (adicionando a classe 'desligado') e o verde ligado (removendo a classe 'desligado'). Isso representa o estado "Siga".



Função Automática

Em seguida, após um atraso de 5 segundos (5000 milissegundos), a função altera o semáforo para o estado de "Atenção" tornando o vermelho desligado, o amarelo ligado e o verde desligado.

Após mais 2 segundos (7000 milissegundos), o semáforo volta ao estado inicial, onde o vermelho está ligado e as outras duas cores estão desligadas, representando o estado "Pare".

O loop definido com setInterval() se repete a cada 13,5 segundos (13500 milissegundos), fazendo com que o semáforo continue alternando entre os estados "Siga", "Atenção" e "Pare" nesse intervalo.

Semáforo Interativo - Construção

Função da Luz Vermelha (Pare)

Código da Função da Luz Vermelha:

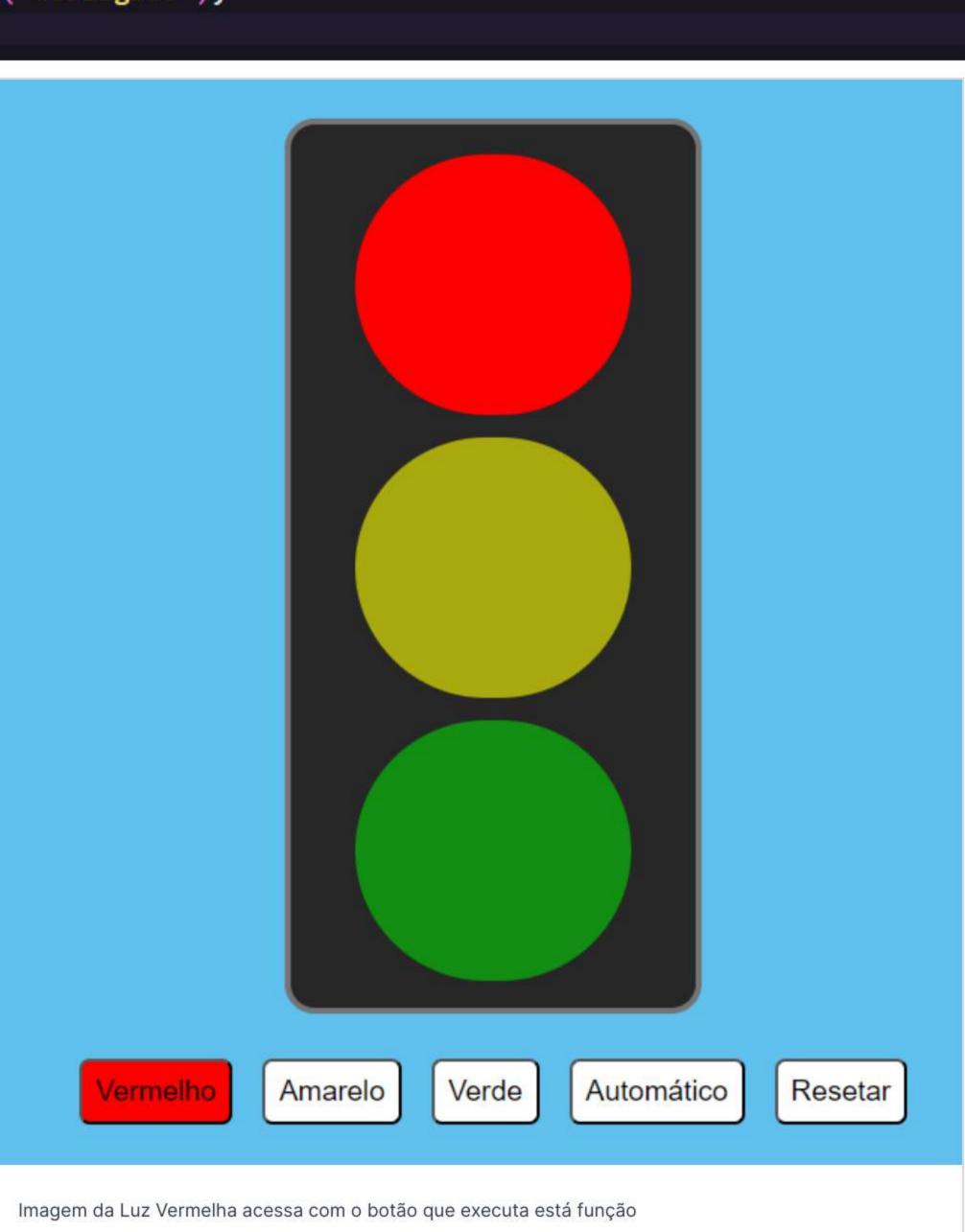
```
function pare() {
 clearInterval(autoModeInterval); // Desliga o Modo Automático (SOMENTE QUANDO A LUZ VERMELHA ESTIVER ATIVA
document.querySelector('.vermelho').classList.remove('desligado');
document.querySelector('.amarelo').classList.add('desligado');
document.querySelector('.verde').classList.add('desligado');
```

Função da Luz Vermelha

Função da Luz Vermelha:

Como funciona:

A função "pare()" começa desligando o Modo Automático (somente quando a luz vermelha estiver ativa), em seguida é removida a opacidade ("desligado") da luz vermelha e atribuída às luzes verde e amarela.



Vermelho Amarelo Verde Automático Resetar Imagem da Função Automática com o botão que executa está função

Semáforo Interativo

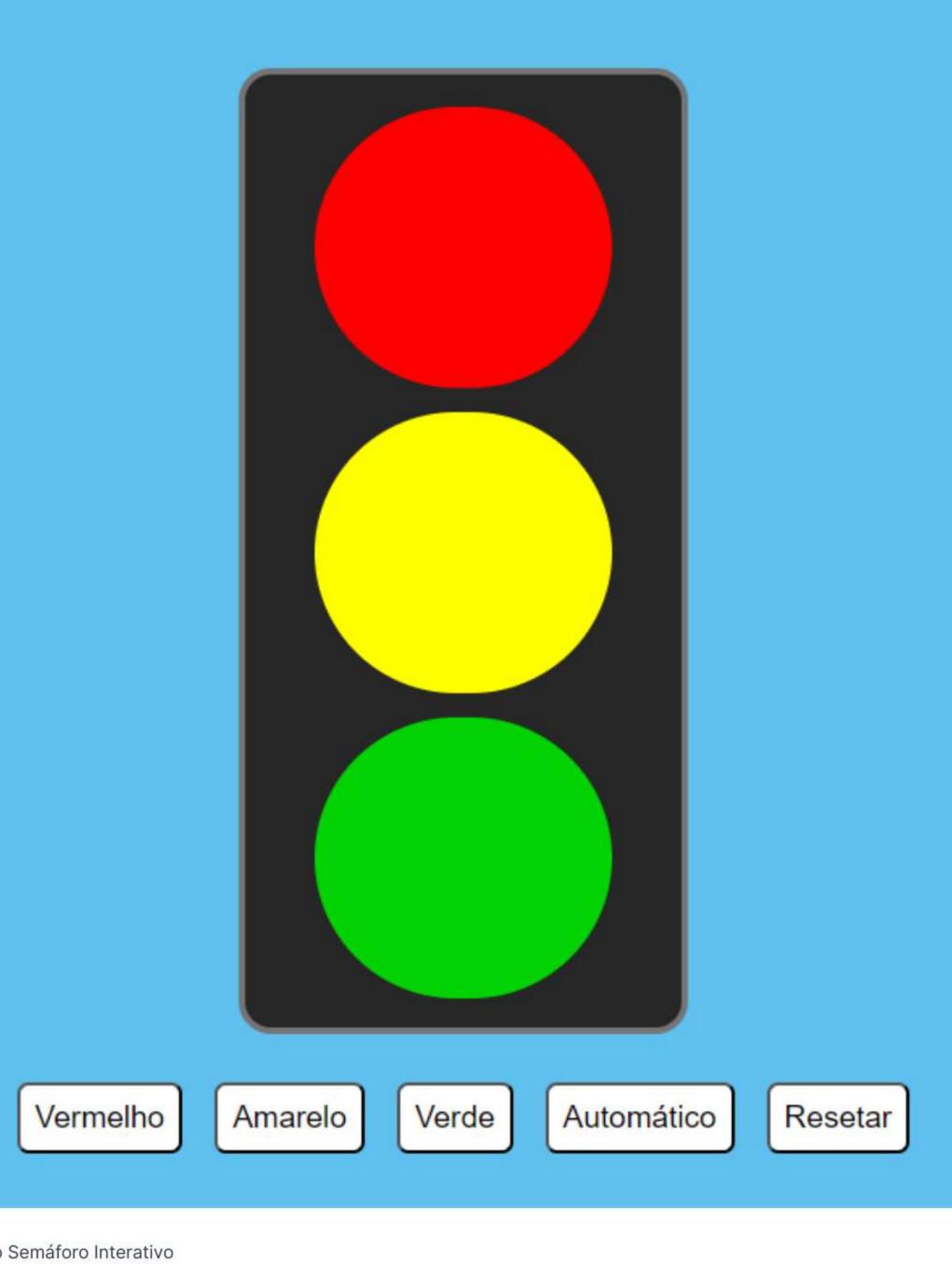


Imagem do Semáforo Interativo

Função da Luz Verde(Siga)

Código da Função da Luz Verde:

clearInterval(autoModeInterval); // Desliga o Modo Automático (SOMENTE QUANDO A LUZ VERMELHA ESTIVER ATIVA document.querySelector('.vermelho').classList.add('desligado'); document.querySelector('.amarelo').classList.add('desligado'); document.querySelector('.verde').classList.remove('desligado');

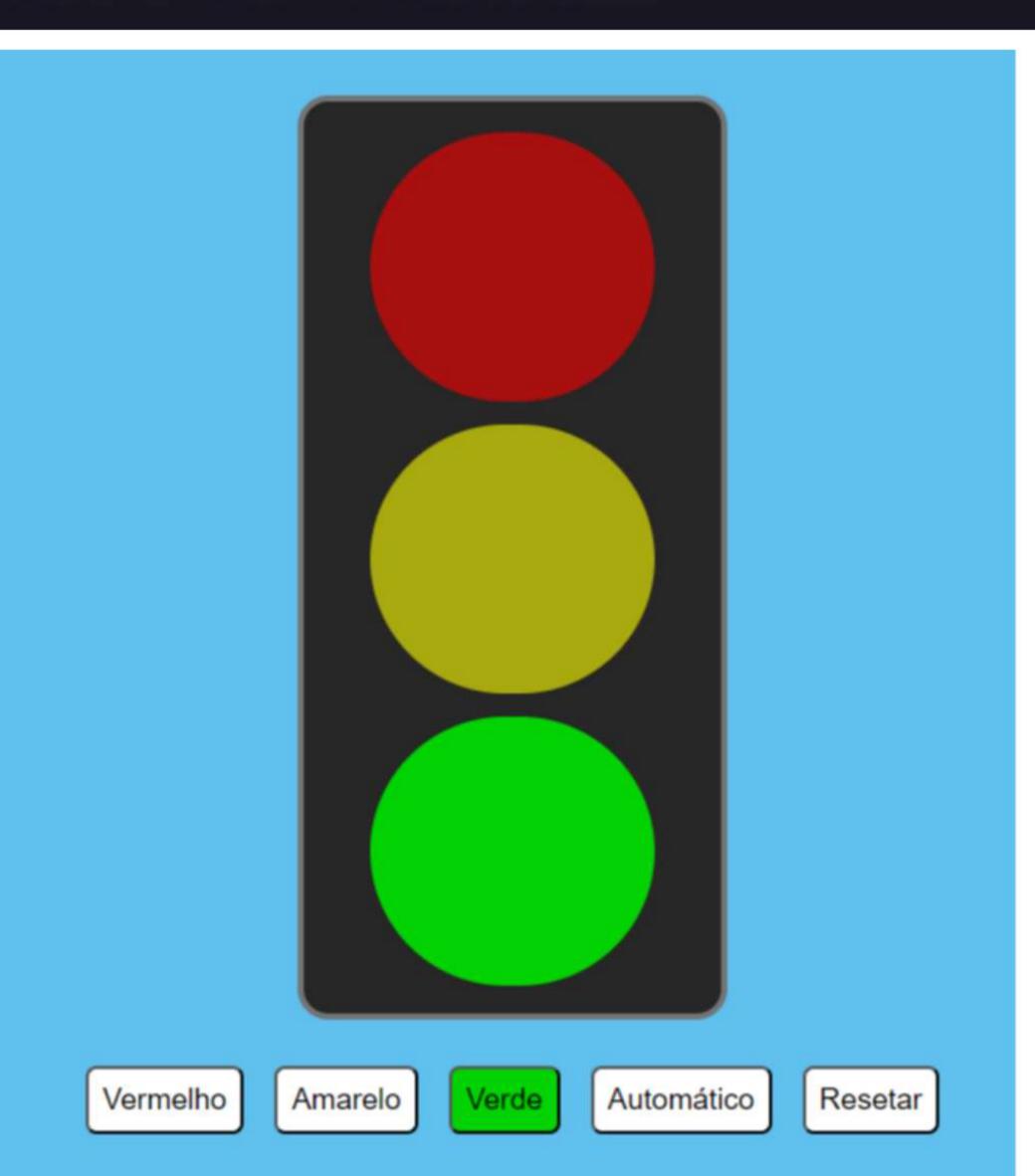


Imagem da Luz Verde acessa com o botão que executa está função

Função de Reset

Código da Função de Reset:



Função de Reset

Função de Reset: Como funciona:

A função "reset()" começa desligando o Modo Automático (somente quando a luz vermelha estiver ativa), em seguida é atribuída a opacidade ("desligado") para todas as luzes do semáforo.

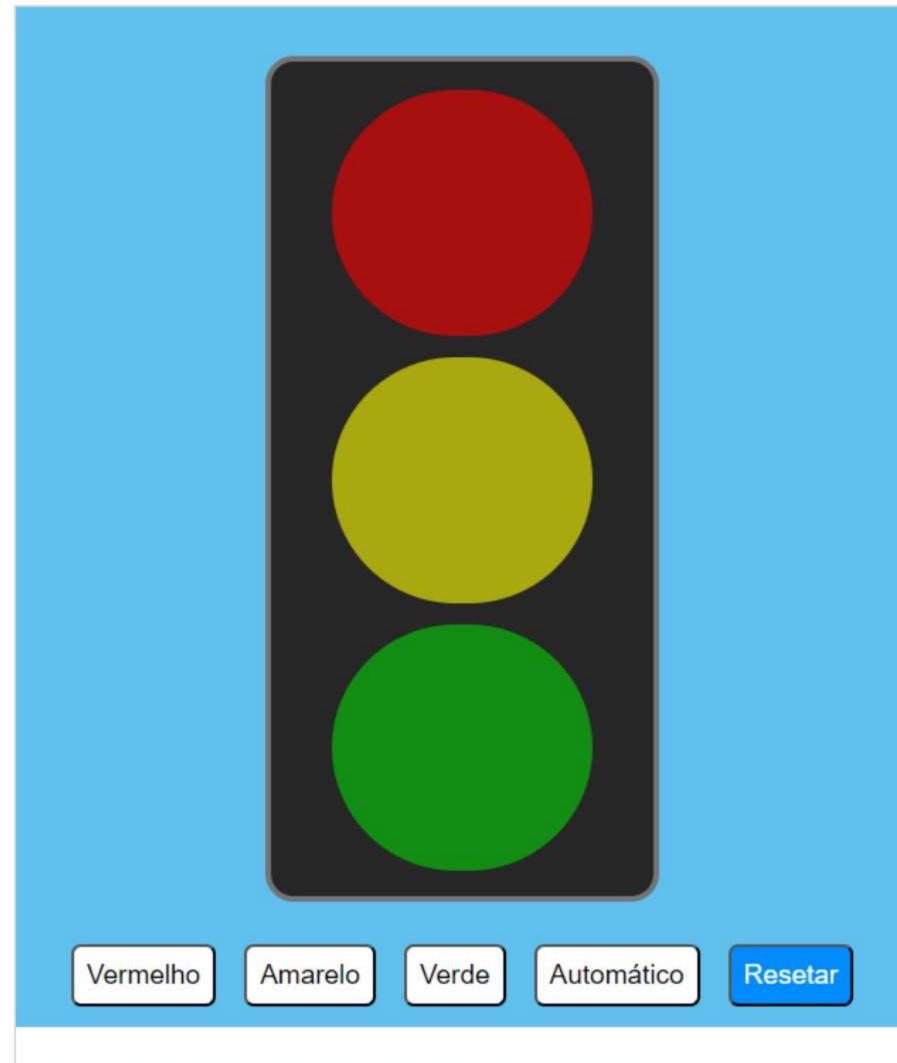


Imagem da Função de Reset com o botão que executa está função

Função da Luz Amarela (Atenção)

Código da Função da Luz Amarela:

nction atencao() { clearInterval(autoModeInterval); // Desliga o Modo Automático (SOMENTE QUANDO A LUZ VERMELHA ESTIVER ATIV document.querySelector('.vermelho').classList.add('desligado'); document.querySelector('.amarelo').classList.remove('desligado'); document.querySelector('.verde').classList.add('desligado');

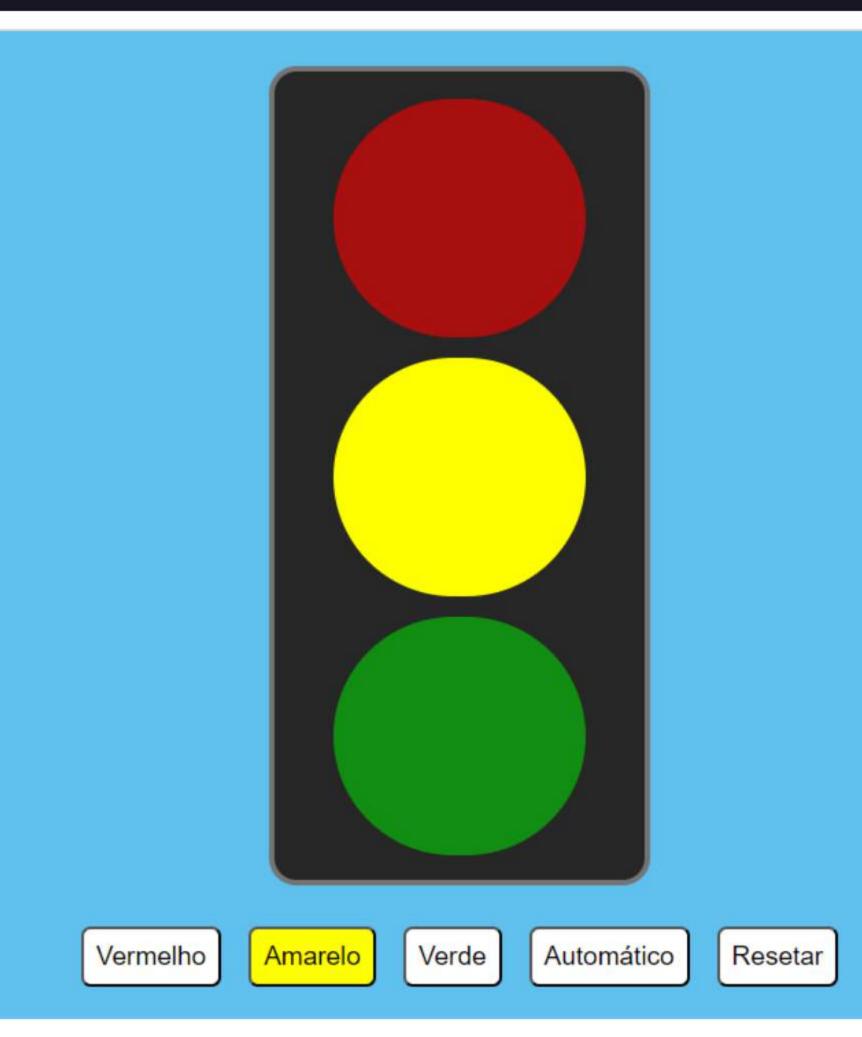


Imagem da Luz Amarela acessa com o botão que executa está função

Função da Luz Amarela:

Função da Luz Amarela

Como funciona:

A função "atencao()" começa

desligando o Modo Automático (somente quando a luz vermelha estiver ativa), em seguida é removida a opacidade ("desligado") da luz amarela e atribuída às luzes verde e vermelha.

Função de Verificar A Vitória

Código da Função de Verificar a



```
//define matrizes com as combinacoes de vitorias
const winPatterns = [
   [0, 1, 2], [3, 4, 5], [6, 7, 8], //linhas
  [0, 3, 6], [1, 4, 7], [2, 5, 8], //colunas
   [0, 4, 8], [2, 4, 6] //diagonais
 ui o codigo dentro do for vai ser executado uma vez para cada combinacao
for (const pattern of winPatterns) {
  const [a, b, c] = pattern;
     //aqui verifica se as 3 células nas posições (a b c) tem o mesmo, ou seja se as 3 são X ou O
   if (selected[a] && selected[a] === selected[b] && selected[a] === selected[c]) {
       //mensagem mostrada no jogadorVez é atualizada p/ mostrar q o jogador com o simbolo mostrado em selected[a] venceu o jogo
       jogadorVez.innerHTML = "JOGADOR ${selected[a]} VENCEU!";
       //aqui como o gamerOver esta true mostra que ele terminou e pode ser reiniciado
       //se ja foi encontardo um ganhador ele interrompe para nao continuar procurando combinacoes
       return;
 //verifica se nao tem mais casas vazias, sem X ou O
 //se for verdade mostra que nao teve ganhador
 f (!selected.includes("")) {
   //ent mostra essa mensagem
   jogadorVez.innerHTML = "Deu Velha!";
   //e toca esse som
   risada.play();
   //indica que o jogo terminou
   gameOver = true;
```

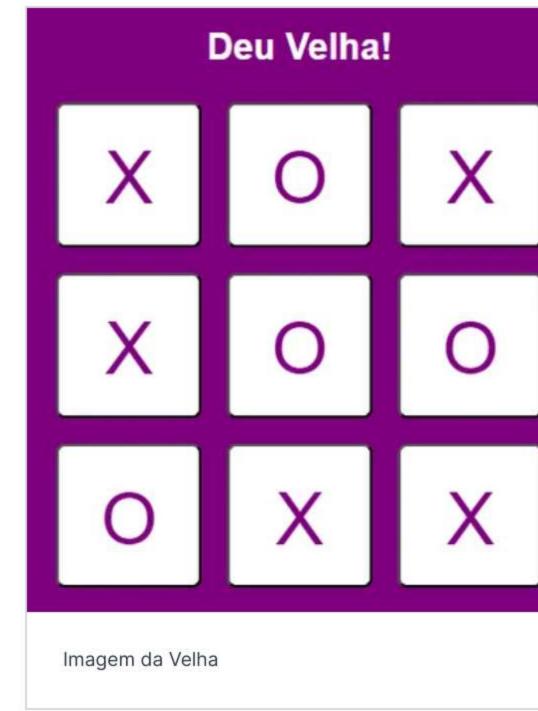
JOGADOR O VENCEU! Imagem de Vitória

Função de Verificar a Vitória

Função de Verificar a Vitória:

Como funciona:

A função checkWinner() é responsável por verificar se há um vencedor no jogo da velha com base nas combinações de vitória possíveis e se o jogo terminou em empate ("velha"). Ela atualiza a mensagem exibida no elemento jogadorVez de acordo com o resultado da verificação e controla o estado do jogo através da variável gameOver.



Jogo da Velha - Construção

Função de Jogar Novamente

Código da Função de Jogar Novamente:

```
v function JogarNovamente() {
    divSelecionarJogador.style.display = "block"; //exibe a tela de seleção de jogador de novo
    mainGame.style.display = "none"; //esconde o tabuleiro de jogo
     jogadorVez.innerHTML = ""; //limpa a mensagem do jogador da vez
    iniciar(); //reinicia o jogo
```

Função de jogar Novamente

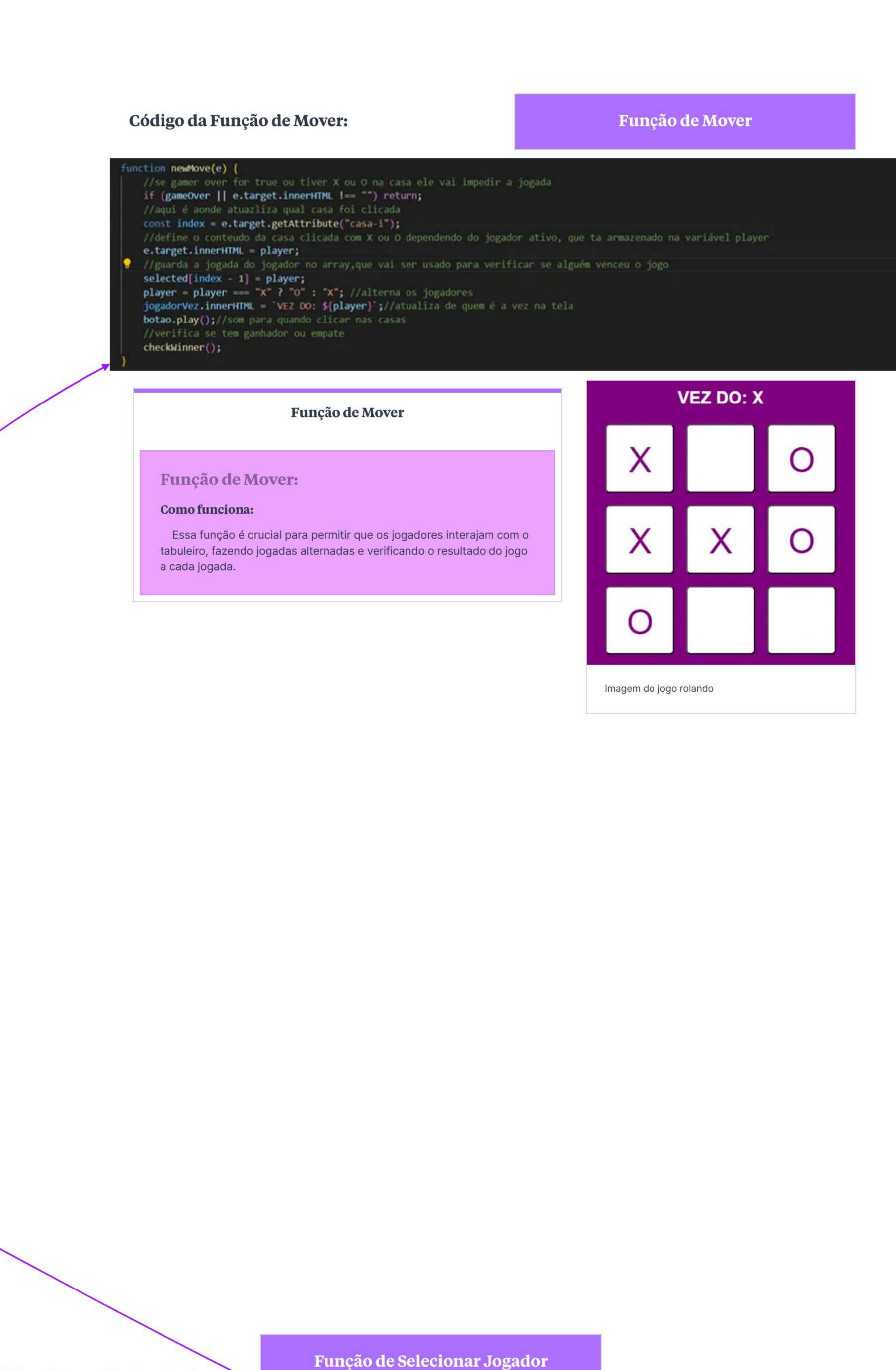
Função de Jogar Novamente:

Como funciona:

Essa função permite que os jogadores escolham jogar novamente após o término de uma partida, retornando à tela de seleção de jogador e preparando o jogo para uma nova

Jogar Novamente

Imagem de Jogar Novamente



Código da Função de Selecionar

Jogador: nction selecionarJogador(jogadorSelecionado) { player = jogadorSelecionado; //atribui o valor do jogador selecionado a variável player divSelecionarJogador.style.display = "none"; //esconde a tela de seleção de jogador mainGame.style.display = "block"; //mostra o tabuleiro de jogo na tela jogadorVez.innerHTML = `VEZ DO: \${player}`; //define o jogador ativo e mostra na tela quem é iniciar(); //inicia o jogo depois da seleção

Função de Selecionar Jogador

Função de Selecionar Jogador:

Como funciona:

A função selecionarJogador(jogadorSelecionado) atribui o jogador selecionado à variável player, esconde a tela de seleção de jogador, mostra o tabuleiro de jogo na tela, exibe qual jogador está ativo e inicia o jogo após a seleção.

Escolha seu jogador: Jogador (X) Jogador (O)

Imagem do Menu de Escolha do Jogador

Função de Iniciar:

//seleciona todos os botões da area do tabuleiro

document.querySelectorAll(".casas button").forEach((item) => {

item.innerHTML = ""; //limpa o conteúdo HTML do botão, tira os X e O

Como funciona:

Código da Função de Iniciar:

function iniciar() {

Imagem do Jogo de Velha

selecionarJogador() permite que os jogadores escolham seus símbolos e exibe o tabuleiro, enquanto iniciar() configura o tabuleiro para começar ou reiniciar o jogo, limpando as células, definindo o jogador ativo e habilitando as jogadas. Essas funções são fundamentais para estabelecer o estado inicial do jogo e permitir que os jogadores façam suas jogadas em um ambiente de jogo funcional.

selected = Array(9).fill(""); //reinicia o array de jogadas com a string vazia, ou seja sem X ou 0

gameOver = false; //define a variavel como false, ou seja o jogo só ira reiniciar quando for true

item.addEventListener("click", newMove); //esse botao "ouve" e quando for clicado vai executar

jogadorVez.innerHTML = `VEZ DO: \${player}`; //inicia com o jogador selecionado e mostra quem é

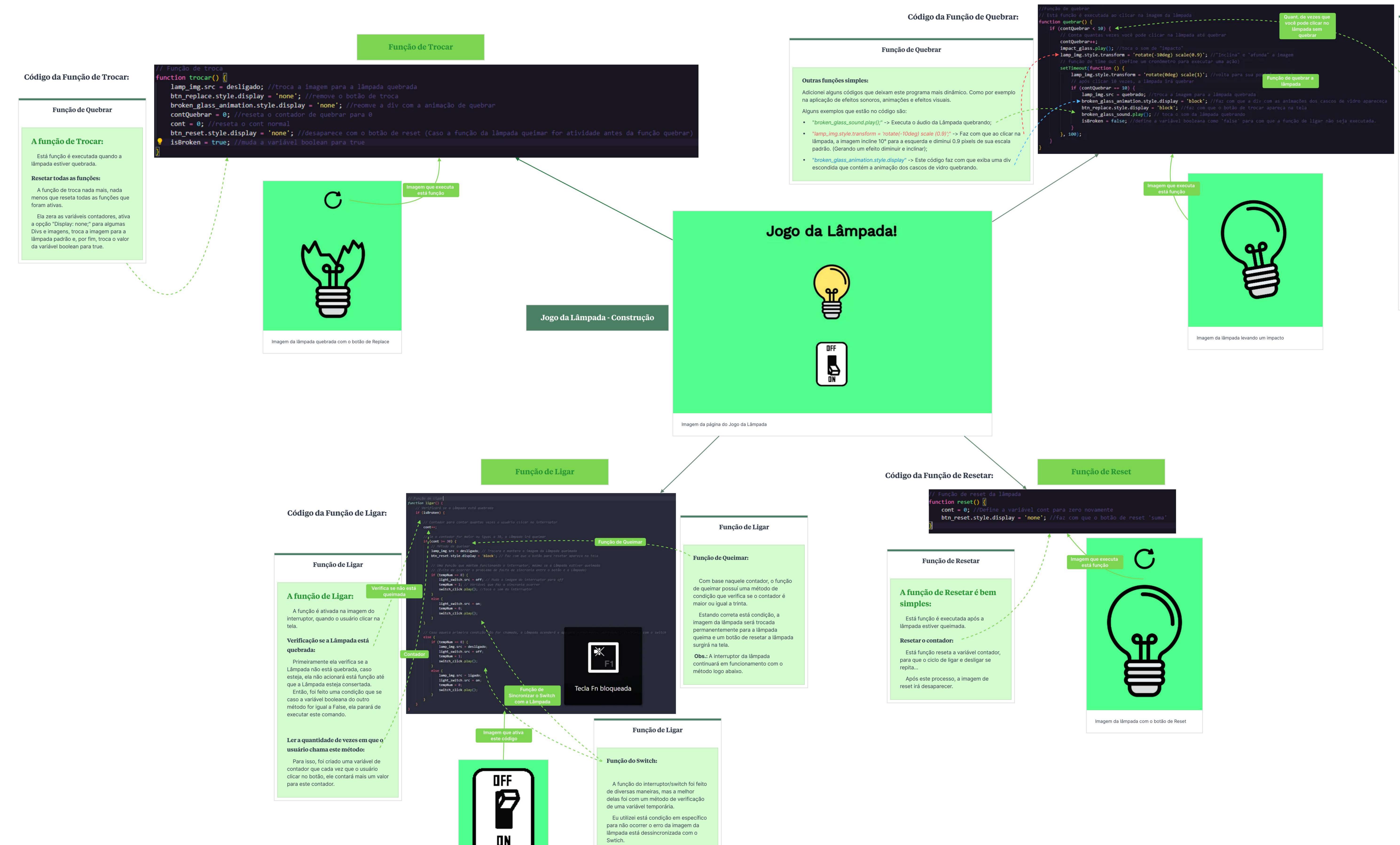
Função de Iniciar

Função de Iniciar

JOGO DA VELHA

VEZ DO: O

Jogar Novamente



Mas enfim, a condição IF verifica se

"tempNum" que vai mudando o seu valor

Caso seja 0, a imagem da lâmpada e

do switch ficaram sincronizados e o valor

da variável mudará para 1. O mesmo

uma variável temporária denominado

para 0 ou 1.

acontece para o Else.

Imagem do Interruptor

Função de Quebrar

A função de Quebrar:

Verificação de quantas vezes foram clicadas:

Primeiramente, foi criado uma condição que verifica se uma variável contadora "contQuebrar" é menor igual a 10. Caso esteja correta está afirmação, você poderá clicar na lâmpada sem ter o risco da lâmpada quebrar.

Função de quebrar:

Utilizando da explicação dada anteriormente, a função de quebrar é bem simples.

Foi criada uma condição que verifica se a variável contadora é igual a 10. Se a condição for aceita, trocará a imagem para uma lâmpada quebrada, o botão de trocar a lâmpada irá aparecer e alguns efeitos especiais irão rolar na tela.

Para evitar o problema de acender e apagar a luz quando a lâmpada estiver quebrada, a variável booleana "isBroken" é trocada para false, fazendo com que aquela condição da função de ligar a lâmpada seja ativada impedindo que o usuário acenda a luz.