

Atividade avaliativa assíncrona 2

Atividade avaliativa assíncrona 2

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define SIZE 3

void imprimirTabuleiro(char tabuleiro[SIZE][SIZE]) {
    int i, j;
    for (i = 0; i < SIZE; i++) {
        for (j = 0; j < SIZE; j++) {
            printf("%c", tabuleiro[i][j]);
            if (j < SIZE - 1) {
                printf("|");
            }
        }
        printf("\n");
        if (i < SIZE - 1) {
            printf("-----\n");
        }
    }
}

int verificarVitoria(char tabuleiro[SIZE][SIZE], char jogador) {
    int i;
    for (i = 0; i < SIZE; i++) {
        if ((tabuleiro[i][0] == jogador && tabuleiro[i][1] == jogador &&
tabuleiro[i][2] == jogador) ||
            (tabuleiro[0][i] == jogador && tabuleiro[1][i] == jogador &&
tabuleiro[2][i] == jogador)) {
            return 1;
        }
    }

    if ((tabuleiro[0][0] == jogador && tabuleiro[1][1] == jogador &&
tabuleiro[2][2] == jogador) ||
        (tabuleiro[0][2] == jogador && tabuleiro[1][1] == jogador &&
tabuleiro[2][0] == jogador)) {
        return 1;
    }
}
```

```

    }

    return 0;
}

int verificarEmpate(char tabuleiro[SIZE][SIZE]) {
    int i, j;
    for (i = 0; i < SIZE; i++) {
        for (j = 0; j < SIZE; j++) {
            if (tabuleiro[i][j] == ' ') {
                return 0;
            }
        }
    }
    return 1;
}

void jogadaComputadorFacil(char tabuleiro[SIZE][SIZE]) {
    int linha, coluna;
    do {
        linha = rand() % SIZE;
        coluna = rand() % SIZE;
    } while (tabuleiro[linha][coluna] != ' ');

    tabuleiro[linha][coluna] = 'O';
}

void jogadaComputadorDifícil(char tabuleiro[SIZE][SIZE]) {
    int i, j;
    int verificaVitoriaPossivel(char jogador) {
        int i, j;
        for (i = 0; i < SIZE; i++) {
            for (j = 0; j < SIZE; j++) {
                if (tabuleiro[i][j] == ' ') {
                    tabuleiro[i][j] = jogador;
                    if (verificarVitoria(tabuleiro, jogador)) {
                        tabuleiro[i][j] = ' ';
                        return 1;
                    }
                    tabuleiro[i][j] = ' ';
                }
            }
        }
    }
}

```

```

        return 0;
    }
    for (i = 0; i < SIZE; i++) {
        for (j = 0; j < SIZE; j++) {
            if (tabuleiro[i][j] == ' ') {
                tabuleiro[i][j] = 'O';
                if (verificarVitoria(tabuleiro, 'O')) {
                    tabuleiro[i][j] = 'O';
                    return;
                }
                tabuleiro[i][j] = ' ';
            }
        }
    }
}

for (i = 0; i < SIZE; i++) {
    for (j = 0; j < SIZE; j++) {
        if (tabuleiro[i][j] == ' ') {
            tabuleiro[i][j] = 'X';
            if (verificarVitoria(tabuleiro, 'X')) {
                tabuleiro[i][j] = 'O';
                return;
            }
            tabuleiro[i][j] = ' ';
        }
    }
}

int linha, coluna;
do {
    linha = rand() % SIZE;
    coluna = rand() % SIZE;
} while (tabuleiro[linha][coluna] != ' ');

tabuleiro[linha][coluna] = 'O';
}

int main() {
    srand(time(NULL));

    int opcao, vez, linha, coluna, nivel;
    char tabuleiro[SIZE][SIZE] = {{' ', ' ', ' ', ' '}, {' ', ' ', ' ', ' '}, {' ', ' ', ' ', ' '}, {' ', ' ', ' ', ' '}}, jogador1[20], jogador2[20], jogadorHumano[20], simboloJogador;

```

```

printf("Jogo da Velha\n");
printf("1 - Jogar\n");
printf("2 - Sair\n");
printf("Digite sua opção: ");
scanf("%d", &opcao);

if (opcao == 1) {
    printf("Jogo da Velha\n");
    printf("1 - Um jogador\n");
    printf("2 - Dois jogadores\n");
    printf("Digite sua opção: ");
    scanf("%d", &opcao);

    if (opcao == 2) {
        printf("Informe o nome do HUMANO 1: ");
        scanf(" %s", jogador1);
        printf("Informe o nome do HUMANO 2: ");
        scanf(" %s", jogador2);

        vez = rand() % 2;
        simboloJogador = (vez == 0) ? 'X' : 'O';

        printf("%c - %s\n%c - %s\n", jogador1[0], jogador1, jogador2[0],
jogador2);
        printf("Sorteando quem vai começar ...\n");

        if (vez == 0) {
            printf("%s começa\n", jogador1);
        } else {
            printf("%s começa\n", jogador2);
        }

        while (1) {
            printf("Selecione a linha para a jogada (1, 2 ou 3): ");
            scanf("%d", &linha);
            printf("Selecione a coluna para a jogada (1, 2 ou 3): ");
            scanf("%d", &coluna);

            if (linha < 1 || linha > SIZE || coluna < 1 || coluna > SIZE
|| tabuleiro[linha - 1][coluna - 1] != ' ') {
                printf("Jogada inválida. Tente novamente.\n");
            } else {
                tabuleiro[linha - 1][coluna - 1] = simboloJogador;
            }
        }
    }
}

```

```

        imprimirTabuleiro(tabuleiro);

        if (verificarVitoria(tabuleiro, simboloJogador)) {
            printf("O jogador %s venceu!\n", (simboloJogador ==
'X') ? jogador1 : jogador2);
            break;
        } else if (verificarEmpate(tabuleiro) == 1) {
            printf("Empate!\n");
            break;
        }

        vez = 1 - vez;
        simboloJogador = (vez == 0) ? 'X' : 'O';

        if (vez == 0) {
            printf("%s é sua vez de jogar.\n", jogador1);
        } else {
            printf("%s é sua vez de jogar.\n", jogador2);
        }
    }
}

} else if (opcao == 1) {
    printf("Informe o seu nome: ");
    scanf(" %s", jogadorHumano);
    printf("NÍVEL DO JOGO\n");
    printf("1 - Fácil\n");
    printf("2 - Difícil\n");
    printf("Digite sua opção: ");
    scanf("%d", &nivel);

    vez = rand() % 2;
    simboloJogador = (vez == 0) ? 'X' : 'O';

    printf("%c - %s\n%c - Computador\n", jogadorHumano[0],
jogadorHumano);
    printf("Sorteando quem vai começar ...\n");

    if (vez == 0) {
        printf("%s começa\n", jogadorHumano);
    } else {
        printf("Computador começa\n");
    }
}

```

```

while (1) {
    if (vez == 0) {
        printf("Selecione a linha para a jogada (1, 2 ou 3): ");
        scanf("%d", &linha);
        printf("Selecione a coluna para a jogada (1, 2 ou 3): ");
        scanf("%d", &coluna);

        if (linha < 1 || linha > SIZE || coluna < 1 || coluna >
SIZE || tabuleiro[linha - 1][coluna - 1] != ' ') {
            printf("Jogada inválida. Tente novamente.\n");
        } else {
            tabuleiro[linha - 1][coluna - 1] = simboloJogador;
            imprimirTabuleiro(tabuleiro);

            if (verificarVitoria(tabuleiro, simboloJogador)) {
                printf("O jogador %s venceu!\n", (simboloJogador
== 'X') ? jogadorHumano : "Computador");
                break;
            } else if (verificarEmpate(tabuleiro) == 1) {
                printf("Empate!\n");
                break;
            }

            vez = 1 - vez;
            simboloJogador = (vez == 0) ? 'X' : 'O';

            if (vez == 0) {
                printf("%s é sua vez de jogar.\n",
jogadorHumano);
            }
        }
    } else {
        if (nivel == 1) {
            jogadaComputadorFacil(tabuleiro);
            imprimirTabuleiro(tabuleiro);

            if (verificarVitoria(tabuleiro, 'O')) {
                printf("Computador venceu!\n");
                break;
            } else if (verificarEmpate(tabuleiro) == 1) {
                printf("Empate!\n");
                break;
            }
        }
    }
}

```

```

    }

    vez = 1 - vez;
    simboloJogador = (vez == 0) ? 'X' : 'O';
} else if (nivel == 2) {
    jogadaComputadorDifícil(tabuleiro);
    imprimirTabuleiro(tabuleiro);

    if (verificarVitoria(tabuleiro, 'O')) {
        printf("Computador venceu!\n");
        break;
    } else if (verificarEmpate(tabuleiro) == 1) {
        printf("Empate!\n");
        break;
    }

    vez = 1 - vez;
    simboloJogador = (vez == 0) ? 'X' : 'O';
}
}
}
}
}

return 0;
}

```