MANUAL

0810

Jogo do Galo

Lalesca Gonçalves

Prog02

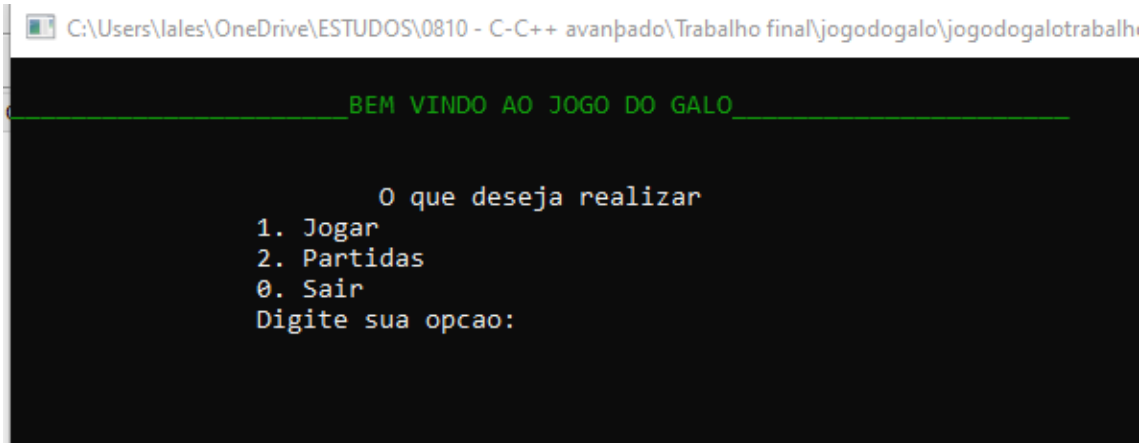
Formador:

Ricardo Farinha

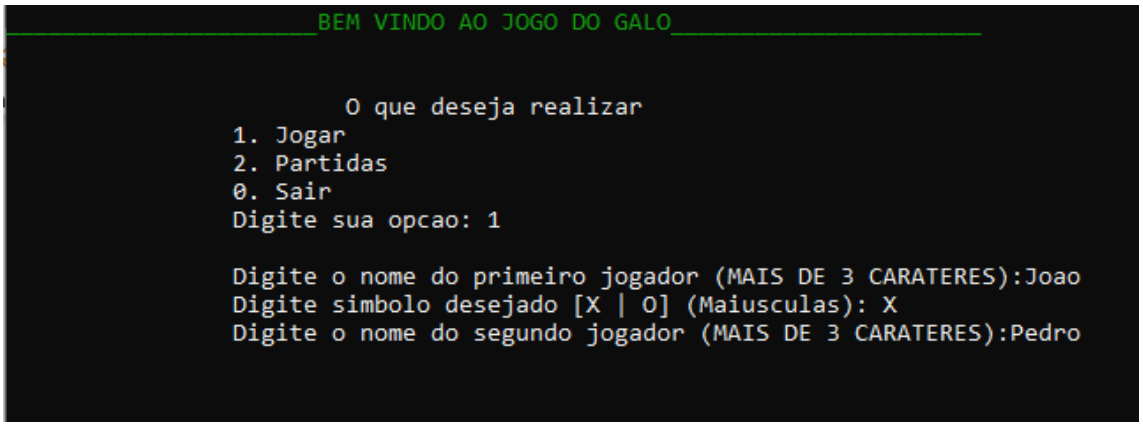
08-2020

IEFP

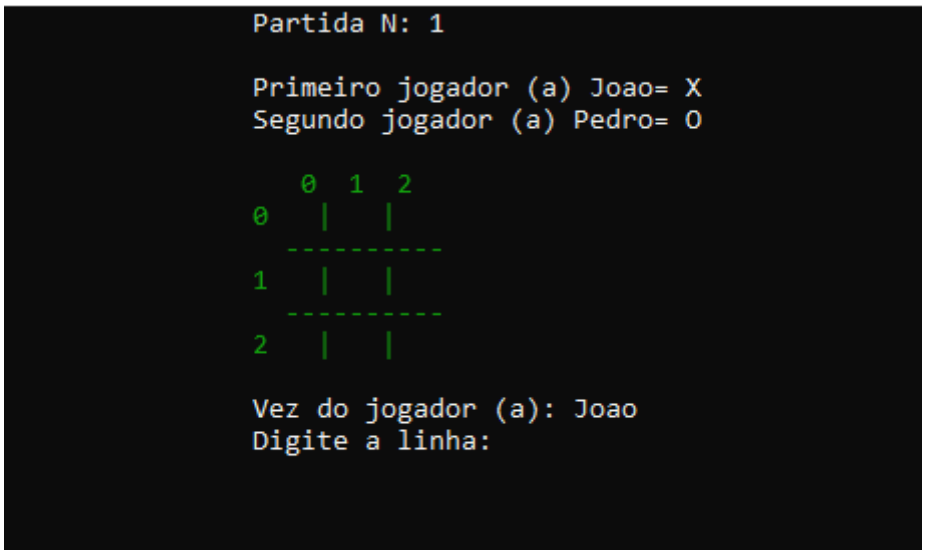
Menu de opções



Ao clicar em jogar pela primeira vez deve inserir o nome de cada jogador, o nome precisa ter mais de três caracteres. O primeiro jogador inserido tem direito de escolher o símbolo que deseja continuar a jogar. Ao escolher o X o outro jogador automaticamente terá o símbolo O e vice-versa.



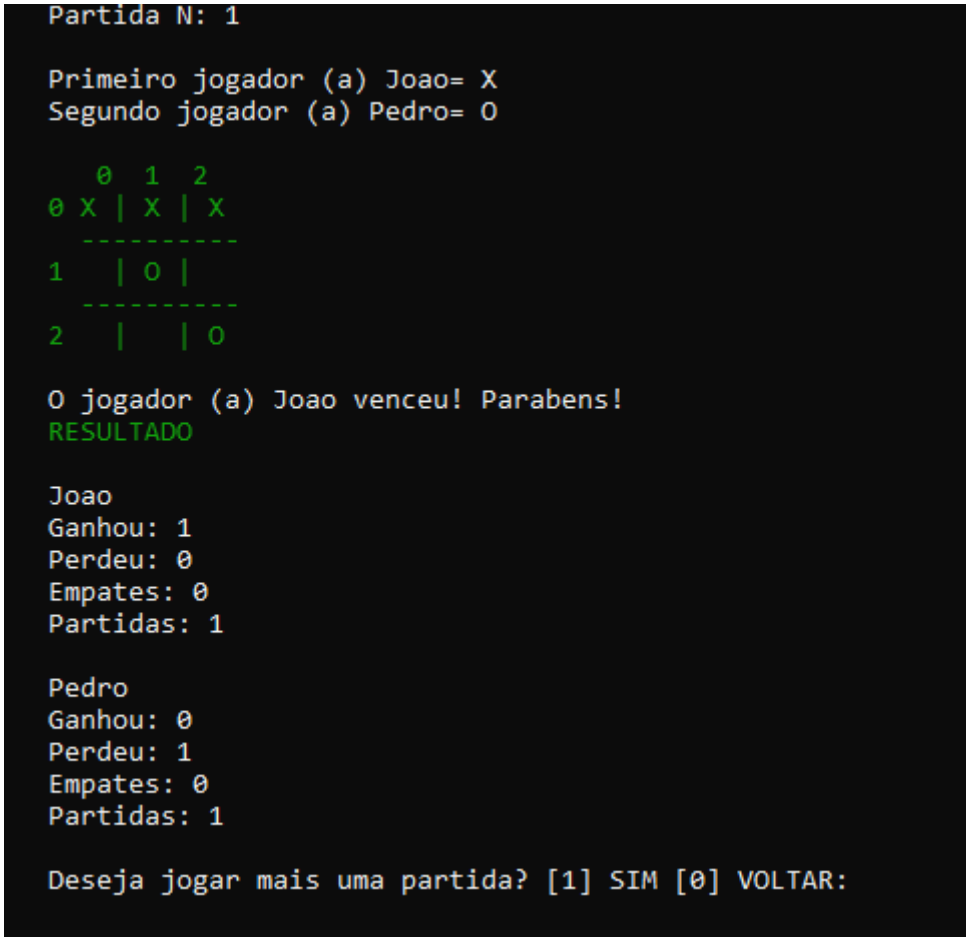
Após inserido os nomes dos jogadores e o símbolo a partida inicia.



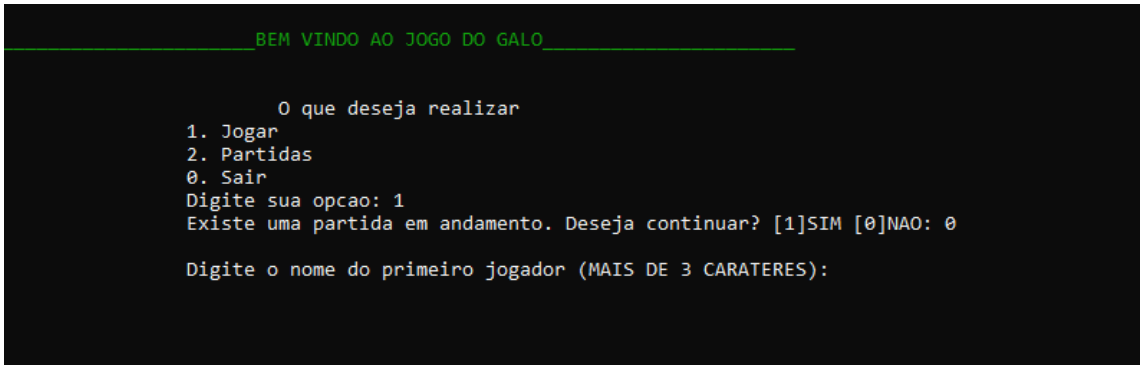
O primeiro jogador inserido inicia a partida e deve escolher a linha e a coluna que deseja. Sendo estás 0,1,2.

Após inserção correta dos valores, o tabuleiro automaticamente preenche os espaços vazios e passa a vez para o próximo jogador.

Quando um jogador ganha é mostrado o nome do ganhador e o placar do jogo.



O jogador tem opção de jogar quantas vezes quiser. A cada jogada os valores são assim atualizados. E pode retornar novamente ao Menu, onde poderá salvar partidas se assim desejar ou jogar novamente. Se o jogador clicar no jogar novamente e ainda tiver uma partida em curso, será questionado se deseja continuar a partida anterior (voltando assim para o tabuleiro) ou não deseja fazê-lo, iniciando assim uma partida do zero, onde novamente será solicitado os nomes dos jogadores.



Regras e objetivos do jogo do Galo

As regras do jogo são bem simples. Em suma, dois jogadores escolhem dois símbolos com que querem jogar. O programa aceita apenas as letras X e O. O jogador deve tentar criar uma linha em sequência com o caractere que estiver jogando, seja em linha, coluna ou na diagonal. Quem completar vence.

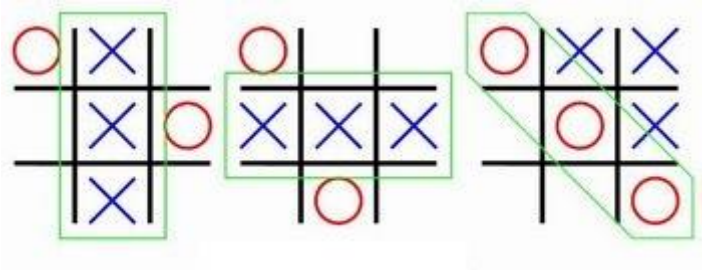
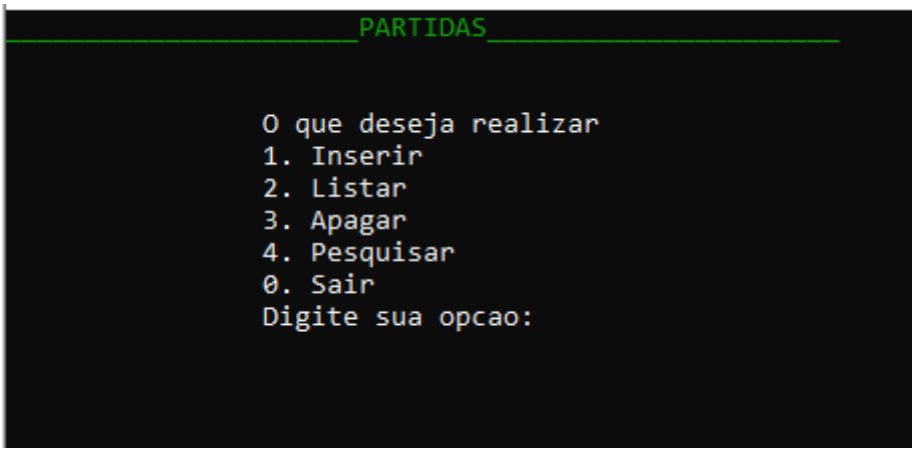


Figura 1: Situações de vitoria - Fonte: http://www.ppgia.pucpr.br/~radtke/jogos/velha/projeto-jogo_da_velha.pdf (Acesso: 27/07/2020).

PARTIDAS

Nessa opção do menu o jogador encontra o que deseja realizar com a partida jogada, ou visualizar partidas anteriores que foram salvas.



Caso o jogador deseje inserir a partida para ser salva, só poderá fazê-lo se já tiver jogado ao menos uma vez. Automaticamente sera salvo o nome de cada jogador e o seu placar.

Para a opção **apagar** e **pesquisar** será questionado o nome de algum jogador adicionado anteriormente que deseja pesquisar, o mesmo deve ter mais de três caracteres.

Bom jogo!!

MANUAL

PROGRAMADORES

Jogo do Galo

Lalesca Gonçalves

Prog02

Formador:

Ricardo Farinha

08-2020

IEFP

O jogo do galo foi desenvolvido utilizando a linguagem C++, com a IDE DevC++.

O programa foi iniciado criando uma struct com o nome de *jogador*.

```
using namespace std;
struct jogador
{
    string nome;
    string simbolo;
    int ganhos;
    int perdas;
    int empates;
    int partidas;
    bool vez = false;
};
```

Esses são as funções usadas, bem como as variáveis globais. Foi criado a variável *hConsole* para podermos adicionarmos cor dentro no console. E foi criado duas variáveis a partir da struct como supra citado *j1, j2*.

```
HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE); //mudanca de cor das letras
jogador j1, j2;
string tabuleiro[3][3];
int contador=0;
int coluna, linha, partida =0;
fstream ficheiro;

string lerNomeJogadores();
void iniciarJogo();
void menu(int op);
void menuPartida (int opcao);
void inicioFicheiro();
void mostrartabuleiro();
void inserirTabuleiro();
void verificaganhador();
void vencedorx();
void vencedor0();
void inicializarVetor();
void inicializarJogo();
void atribuirSimbolo();
int verificarCampo();
void iniciarPartida();
void maisPartida();
void mostrarResultado();
void inserirFicheiro();
void listarFicheiro();
void pesquisarFicheiro();
void apagarFicheiro();
void verificarJogo();
```

No main é chamado o inicializar vetor para que seja possível verificar mais a frente se o espaço se encontra vazio.

```
main()
{
    inicializarVetor();
    iniciarJogo();
}
void inicializarVetor(){//inicializando o tabuleiro para depois verificar na funcao insere tabuleiro
    for (int i=0;i<3;i++)
    {
        for (int k=0; k<3;k++){
            tabuleiro[i][k] = "";
        }
    }
}
void inicializarJogo(){//inicializando os jogadores todos
    j1.nome = "\0";
    j1.simbolo ="\0";
    j1.ganhos =0;
    j1.perdas =0;
    j1.empates=0;
    j1.partidas=0;
    j1.vez=false;

    j2.nome = "\0";
    j2.simbolo ="\0";
    j2.ganhos =0;
    j2.perdas =0;
    j2.empates=0;
    j2.partidas=0;
    j2.vez=false;

    partida =0;
    inicializarVetor();
    contador =0;//zera o contador
    iniciarPartida();
}
```

Funções

A função verificarJogo verifica se há uma partida em andamento através da variável global *partida*.

```
void verificarJogo(){//inicializando os jogadores todos
    int op;
    if(partida > 0){
        cout<<"\t\tExiste uma partida em andamento. Deseja continuar? [1]SIM [0]NAO: ";
        cin >> op;
        while(cin.fail() || op != 0  && op != 1)
        {
         cin.clear();
         cin.ignore();
         cout << "\t\tCampo invalido, digite novamente: ";
         cin >> op;
        }
        if(op==1){
            iniciarPartida();
        }else{
            inicializarJogo();
        }
    }else{
        inicializarJogo();
    }
}
```

A função lerNomeJogadores tem como intuito verificar os campos em que será necessário ler os nomes dos jogadores, será usada também nas funções apagar e pesquisar em ficheiro.

```
string lerNomeJogadores(){
    string nome;
    cin.ignore();
    getline( cin, nome);
    while(nome.length() < 3){
        cout << "\t\tDigite o novamente o nome: ";
        getline( cin,nome );
    }
    return nome;
}
```

Função para construir o tabuleiro. Essa função é chamada a cada vez de um jogador e insere no tabuleiro o respetivo símbolo.

```
void mostrartabuleiro(){
    system ("CLS");
    cout<<"\t\tPartida N: "<<partida<< "\n";
    cout << "\n\t\tPrimeiro jogador (a) "<<j1.nome << "= "<<j1.simbolo<<endl;
    cout << "\t\tSegundo jogador (a) "<<j2.nome << "= "<<j2.simbolo<<"\n"<<endl;
    int l,c;
    SetConsoleTextAttribute(hConsole, 2);
    cout << "\t\t   0  1  2\n";
    for (l=0;l<3;l++)
    {
        cout << "\t\t" << l;
        for (c=0; c<3;c++){
            if (c ==0 ){
                if(tabuleiro[l][c].length() == 0){
                    cout << "   ";
                }else{
                    cout << " "<<tabuleiro[l][c]<<" ";
                }
            }else{
                if(tabuleiro[l][c].length() == 0){
                    cout << "|   ";
                }else{
                    cout << "| "<<tabuleiro[l][c]<<" ";
                }

            }
        }
        cout << "\n";
        if(l!=2){
            cout << "\t\t  ----------";
            cout << "\n";
        }
    }
    SetConsoleTextAttribute(hConsole, 15);
    verificaganhador();
    inserirTabuleiro();
}
```

verificaCampo é chamado para validar dados do primeiro menu de opções e para validar linhas e coluna.

```
}
int verificarCampo(){
    int n;
    cin >> n;
    while(cin.fail()|| n > 2 || n <0 )
       {
        cin.clear();
        cin.ignore();
        cout << "\t\tCampo invalido, digite novamente: ";
        cin >> n;
       }
    return n;
}
```

A função apagar nos ficheiros cria um ficheiro auxiliar onde será copiado todo o conteúdo do partidas.txt menos o que se deve apagar. Após o ficheiro partidas.txt é apagado e o ficheiro auxiliar é renomeado para o nome de partidas.txt.

O ficheiro auxiliar só é criado caso o ficheiro possa ser aberto e o nome inserido realmente for encontrado dentro do partidas.txt.

```
void apagarFicheiro(){
    bool flag = false;
    string nome = "";
    string linha;
    fstream ficheiroAux;
    ficheiro.open("partidas.txt",ios::in);
    cout<< "\t\tQual jogador deseja apagar? Digite o nome do jogador [ATENCAO, ISSO AFETARA TODOS OS NOMES CORRESPONDENTES -MAIS DE 3 CARATERES-]\n"<<endl;
    cout<< "\t\tNome: ";
    nome = lerNomeJogadores();
    if (ficheiro.is_open()){//abrir ficheiro
        while(getline(ficheiro,linha)){
            if(linha.find(nome) != string::npos){
                flag=true;
            }
        }
        if (flag==true){
            ficheiroAux.open("auxiliar.txt",ios::app);//so criar o ficheiero auxiliar se o ficheiro for possivel ser aberto, e se realmente o nome inserido existe
            ficheiro.clear();
            ficheiro.seekp(0, ios::beg);
            while(getline(ficheiro,linha)){//mandar cada linha para variavel linha
                if(linha.find(nome) == string::npos){//se o nome digitado for diferente, escreve no ficheiro aux, se for igual nao faz nada, dessa forma, ele nao escreve
                    ficheiroAux << linha << endl;
                }
            }
            cout<< "\t\tPartida apagada com sucesso!";
                ficheiro.close();
                ficheiroAux.close();
                remove("partidas.txt");//removo ficheiro anterior
                rename("auxiliar.txt", "partidas.txt");//renomeio o auxiliar para o nome do arquivo original
        }else{
            cout<< "\t\tNome inexistente, verifica se digitou corretamente!" << endl;
            getch();
        }
    }else{
        cout<< "\t\tAinda nao existe nenhuma partida salva";
    }
    ficheiro.close();
    ficheiroAux.close();
    getch();
```