

***Engenharia Informática***

***Programação I***

The Goose Game

Alunos:

Afonso Fernandes

Rafael Ferreira

1. **Índice**

Objetivos........................................................................................2

Introdução.....................................................................................3

Desenvolvimento...........................................................................3

Conclusão......................................................................................7

1. **Objetivos**

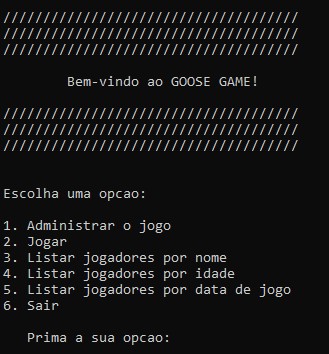
* Utilizar os conteúdos lecionados durante as aulas
* Utilizar Listas Ligadas e Duplamente Ligadas
* Utilizar linguagem C

1. **Introdução**

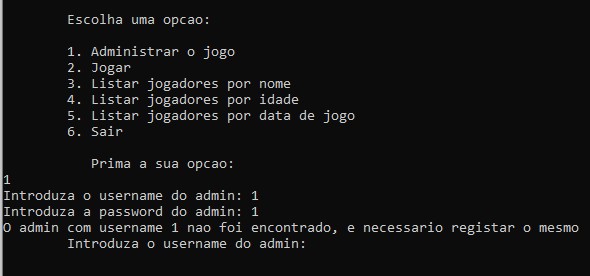
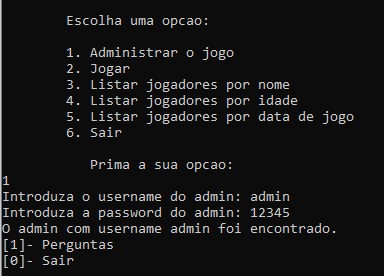
O principal objetivo deste trabalho é desenvolver um jogo em linguagem C, onde um administrador criado pelo utilizador com nome e password consegue criar perguntas de escolha multiplas e fazer a listagem de todas as perguntas já criadas anteriormente. Ao longo do desenvolvimento do projeto foram implementadas várias ideias até o resultado final.

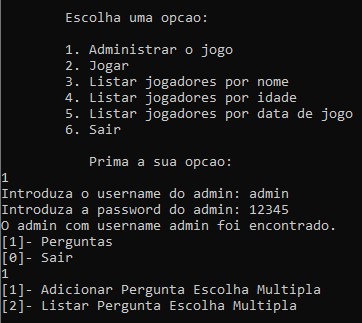
1. **Manual**

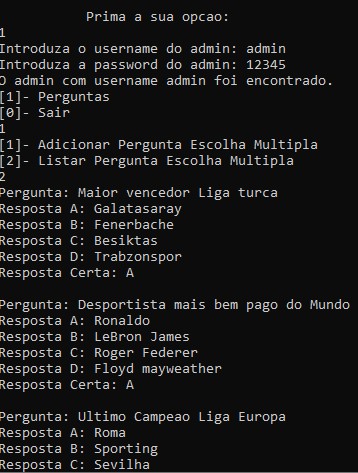
O código inicialmente abre em um menu onde o utilizador escolhe entre administrar o jogo, jogar, listar jogadores por nome, idade ou data de jogo.



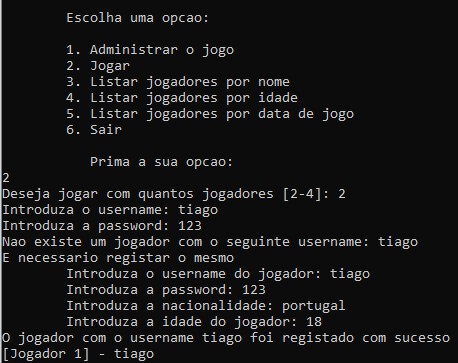
Caso o utilizador decida administrar o jogo vai aparecer para fazer login, ou seja introduzir o username do admin e a sua password. Se estiver corretas essas informações abre-se o menu de administrador, se estiver errado abre um menu para criar o admin, sendo que só é possivel haver um administrador.

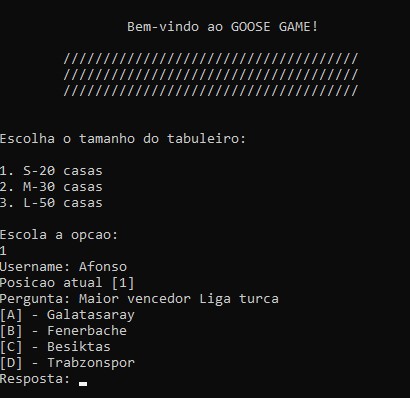


No menu de administrador é possivel criar perguntas de escolha multipla e listar todas as perguntas de escolha multipla já criadas anteriormente que ficaram guardadas em um ficheiro binario.



Caso o utilizador decida jogar, uma pergunta ira aparecer a perguntar quantos jogadores vão jogar, sendo a possibilidade entre 2 a 4. Após o utilizador selecionar quando vão jogar irá aparecer opcao para introduzir o username e a sua password, caso esse username não exista irá aparecer opcao de criar um novo username, password e introduzir a sua nacionalidade e a sua idade. Depois do nº de jogadores darem login irá aparecer uma opção do tamanho do tabuleiro sendo elas S(20 casas), M(30 casas), L(50 casas). Depois de selecionar a opção desejada o jogo começa automaticamente com perguntas de escolha multipla. Caso o utilizador acerte na pergunta certa vai avancando nº de casas estipulado no código referente á casa onde acertou á pergunta. Até chegar ao fim onde quem chegar primeiro á última casa é o vencedor.





Voltando no menu é possivel também listar jogadores por nome, idade, ou data de jogo. Sendo estes guardados quando um utilizador termina um jogo.

1. **Relatório**

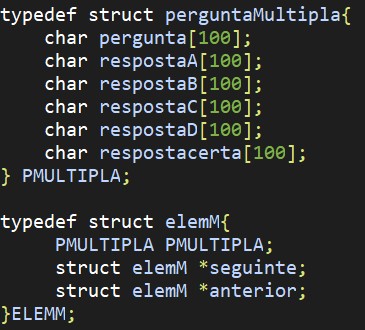
O código do programa está dividido em 9 ficheiros sendo eles “main.c, jogadores.c.h, perguntas.c.h, tabuleiros.c.h, e admin.c.h. Em cada um destes ficheiros está destinado para uma tarefa diferente. O main.c é onde está localizado o menu principal do jogo, onde pode-se escolher entre administrar o jogo, jogar, listar jogadores por nome,idade, ou data de jogo. Cada opção vai rederecionar para uma função especifica relacionada á opção selecionada.



No ficheiro “jogadores.c” estão as funções “create\_player”, função utilizada para criar um jogador novo que ainda não foi registado. A função “login\_players” que é utilizada para os utilizadores fazerem login com o seu username e password, comparando com os dados de jogadores já criados, caso seja igual o login foi feito com sucesso caso contrário entra a função “create\_player”. No fim a função “writePlayers” é invocada onde guarda jogadores no arquivo.txt, guardando o username, password, nacionalidade, idade, data do ultimo jogo e nº de perguntas respondidas. A função “readPlayers” lê a informação que está guardada no ficheiro de texto. Para terminar tem as funções “show\_name”, “show\_age” e “show\_date” que são as funções para fazer a listagem dos utilizadores por nome, idade e por data do ultimo jogo.

No ficheiro “jogadores.h” estão as structs utilizadas que é a struct JOGADOR, que contem a idade, dados, username, passoword, nacionalidade, perguntas respondidas e a data de jogo.

No fichiero “perguntas.c” está a função “imprimirPerguntas” a função imprime informações sobre a pergunta e as respostas associadas ao elemento atual. O código é usado para imprimir essas informações usando printf. A função “listarperguntasmultiplas” serve para o admin poder ver as perguntas de escolha multipla que já estão criadas. A função “verificarRespostaCorreta” percorre a lista e verifica se a resposta fornecida é igual á resposta correta. A função “criarPerguntaMultipla” faz-nos inserir uma pergunta, meter 4 respostas e inserir qual a resposta certa. No final passa a pergunta e as respostas para um ficheiro binário. A função “inserirPerguntaMultipla” cria um novo elemento ELEMM, preenche os seus campos com os valores da estrutura aux e insere esse elemento no início da lista. E a função “lerPerguntaBinario” lê os dados de um arquivo binário e cria uma lista de perguntas de escolha multipla. O ficheiro “perguntas.h” contem as structs utilizadas para o programa correr.



A primeira struct é necessária para a criação da pergunta de escolha multipla. A segunda struct é importante para percorrer a lista tanto para frente quanto para trás, acessando os elementos em ordem sequencial ou em sentido inverso.

No ficheiro “tabuleiros.c” a função “escolher\_tabuleiro” permite ao jogador escolher o tamanho de tabuleiro preferido para o jogo. Após selecionar o nº de casas pretendido o jogo irá pedir para selecionar o numero de jogadores e os jogadores fazerem login com o respetivo username e password. Após isso a função “jogar” vai ler as perguntas guardadas no ficheiro binário e imprimi-las na tela, o primeiro jogador vai jogar respondendo as perguntas. Vai jogar sempre até errar, quando errar troca de jogador. Quando um jogador chegar a ultima casa e ganhar termina o jogo.

No ficheiro “tabuleiros.h” está definido o nº de casas para cada tabuleiro, a struct jogo, que contem a data, vencedor e o tamanho do tabuleiro jogado. Também contém o nº de casas de um jogador avança ao acertar a pergunta na casa selecionada.

Para terminar o ficheiro “admin.c” contem a função “novo\_admin” que é utilizada para criar um novo administrador, colocando o username e a password e registando em um ficheiro .txt. A função “login\_admin” vai pedir um username e password. Caso o username e a password seja igual ao guardado no ficheiro.txt vai abrir o painel de gestão do admin. Caso não seja encontrado e não haja nenhum admin registado (MAX. ADMIN = 1) vai abrir a função “novo\_admin”. A função “gerir\_admin” abre o painel de gestão do administrador onde ele pode criar perguntas e ver as perguntas já criadas anteriormente.

No ficheiro “admin.h” está a struct admin onde guarda o username e a password.

1. **Conclusão**

O trabalho que nos foi atribuido foi o “Jogo do Ganso”. O nosso objetivo foi criar um jogo onde um administrador conseguisse criar perguntas, introduzindo a pergunta as possiveis respostas e a resposta certo. Sendo cada pergunta guardada em um ficheiro binário. Depois a opcção de jogar era possivel escolher quantas pessoas irão jogar dando login nas suas contas, caso não tenham seja possivel criar uma conta conta nome, password, nacionalidade e idade. Após os utilizadores darem login o jogo começa com perguntas onde o utilizador tem de acertar na resposta certa para avançar de casa. As perguntas são lidas do ficheiro binário para o ecrã. Após isso o vencedor é quem chegar em primeiro lugar á ultima casa.

Ao longo do tempo encontramos algumas dificuldades em fazer algumas funções referentes á criação de pergunta e fazer as funções duplamente ligadas bem como os ficheiro binários mas com algum trabalho e pesquisa na internet conseguimos resolve-los.

Para concluir, com a realização deste trabalho prático foi possivel desenvolver as nossas capacidades de programação em C e melhorar o nosso trabalho em equipa.