

Trabalho Prático I - Programação II - Jogo da Forca

Prof. Vinicius Mota DI/UFES

Data de entrega: 10/03/2021

Locais de entrega: - submetido pelo Repl.it; - E também pelo AVA.

1. Objetivos

O objetivo geral deste trabalho é aprimorar as habilidades de programação básica, organização de funções e documentação correta de código, compilação automática de código. O tema proposto para este trabalho prático é o desenvolvimento de um **jogo da forca**. A descrição, funcionalidades detalhadas do sistema serão descritos e exemplificados nas próximas seções.

2. Descrição do sistema

Este trabalho visa simular um jogo da forca com multijogadores. Para isto, o usuário do programa deve fornecer a quantidade de jogadores que irão jogar (de 1 a 4). Onde cada jogador adiciona seu nome e terá sua vez de tentar acertar a letra da palavra. Cada jogador iniciará o jogo com 6 pontos, se o jogador acertar a letra, ele não perde ponto, caso erre a letra, ele perderá 1 ponto. Ao final o jogador com a maior pontuação vencerá, e em caso de empate, o critério poderá ser quebrado pelo número de acertos das letras.

Será fornecido 1 arquivo .csv onde serão disponibilizadas as palavras e as dicas para o jogo. O programa deve ler o arquivo e armazenar as informações em uma matriz para ser utilizada no jogo.

O usuário deve ser capaz de visualizar a pontuação de cada jogador em uma breve interface, a quantidade de letras da palavra a ser acertada definida por (..) e a dica da palavra escondida. Deve ser feito o tratamento da entrada do usuário, para caso o usuário digite alguma letra que já foi utilizada o programa saiba lidar com isso.

IMPORTANTE:

- O código deve ser devidamente separado em funções com as devidas implementações, sempre levando em consideração as boas práticas.
- É seu dever como desenvolvedor filtrar erros de entrada dos usuários. Mesmo que não estejam listados aqui. Sempre comente essas verificações.
- Código deve estar bem comentado.
- É opcional "limpar o terminal" entre as passagens do menu de opções (mas fica bem melhor se conseguir fazer).

2.1 Funcionalidades

O programa funcionará com uma interface do jogo. Isso deve ser feito no terminal, inicialmente será feita a leitura dos nomes dos jogadores e sorteado quem será o primeiro a começar, logo depois, cada jogador terá a chance de escolher uma letra sucessivamente.

Ao executar o programa, deve ser carregado em uma matriz as palavras e dicas do arquivo .csv. Essa matriz contém apenas 2 colunas: Dica e a Palavra. O formato dos arquivos é discutido na próxima seção.

2.1.1 Menu

A primeira "tela" que o usuário deverá visualizar é a tela do menu com as opções:

```
Informe a quantidade de jogadores disponíveis:
```

```
1- Modo solo (Treinamento)
2- 2 jogadores
3- 3 jogadores
4- 4 jogadores
0- Sair do Jogo
```

Opção 1, 2, 3 ou 4 - Inicia-se o jogo

Uma nova interface aparecerá solicitando o(s) nome(s) do(s) jogador(es) em questão, sendo 'X' um número de 1 a 4. Use a opção 9 para voltar ao menu anterior ou 0 para sair do jogo.

```
Informe o nome do jogador X:
```

```
Nome: <nome do jogador> ou <numero>
```

```
9- Voltar
0- Sair do Jogo
```

Jogando o jogo

Depois das escolhas dos nomes, uma palavra e sua respectiva dica será sorteada, isto é, uma linha da matriz de palavras. Em seguida, é sorteado qual jogador irá começar, em nosso exemplo cada jogador tem a chance de 1/4 de começar, a ordem de jogada segue a sequência numérica do jogador, se começar o jogo com o jogador 2, depois dele será o jogador 3 e assim sucessivamente.

O MENU ABAIXO CONSISTE EM UM EXEMPLO COM 4 JOGADORES:

```
##### Forca #####
```

```
Jogador 1: <nome do jogador> -> <pontuacao>
Jogador 2: <nome do jogador> -> <pontuacao>
Jogador 3: <nome do jogador> -> <pontuacao>
Jogador 4: <nome do jogador> -> <pontuacao>
```

```
A M A _ E _ O
```

```
Dica: Cor      Letras Já Escolhidas: A O E B C I J K M
```

```
Jogador 1 digite uma letra a sua escolha: <entrada><char>
```

Se a letra que o jogador escolher já tiver sido escolhida, deve ser informado a mensagem e pedir uma nova entrada ao jogador:

```
A letra ja foi escolhida. Favor digite uma outra letra: <entrada><char>
```

Pontuação: Erros e Acertos

Todos os jogadores iniciam com 6 pontos.

- Se errou uma letra: perde 1 ponto.
- Se acertou uma letra: Mantém pontuação.

Em ambos os casos, cada jogador só terá uma chance e passará para o próximo jogador.

Vencerá o jogador com mais pontos remanescentes.

```
##### Forca #####
```

```
Jogador 1: Pedro -> 5
Jogador 2: Felipe -> 3
Jogador 3: Alvaro -> 0 - Perdeu!
Jogador 4: Jose -> 1
```

```
A M A _ E _ O
```

```
Dica: Cor      Letras Já Escolhidas: A O E B C I J K M
```

```
Jogador 2 digite uma letra a sua escolha: U
```

Caso o jogador tenha seus pontos zerados, ele não jogará mais. No exemplo acima, após Felipe jogar, quem joga é o Jose.

Ganhando o Jogo

Há duas maneiras de finalizar o jogo:

1- Terminando a Palavra: ao terminar a palavra por completo ganha aquele jogador que conseguiu manter mais pontos. Quebra de empate de quem mais acertou letras, caso esse numero também seja igual, considere um empate.

```
##### Forca #####
```

```
Jogador 1: Pedro -> 5
Jogador 2: Felipe -> 3
Jogador 3: Alvaro -> 0 - Perdeu!
Jogador 4: Jose -> 1
```

```
A M A R E L O
```

```
Dica: Cor      Letras Já Escolhidas: A O E B C I J K M R
```

```
Jogador 2 digite uma letra a sua escolha: L
```

Neste exemplo Pedro foi o vencedor do jogo.

2- A segunda maneira de terminar o jogo é se todos os jogadores, exceto um, tenham seus pontos zerados:

```
##### Forca #####
```

```
Jogador 1: Pedro -> 3
Jogador 2: Felipe -> 0 - Perdeu!
Jogador 3: Alvaro -> 0 - Perdeu!
Jogador 4: Jose -> 0 - Perdeu!
```

```
A M A _ E _ O
```

```
Dica: Cor      Letras Já Escolhidas: A O E B C I J K M
```

```
Jogador 2 digite uma letra a sua escolha: U
```

Em todos os dois exemplos quem venceria seria o Pedro, deve-se então imprimir a seguinte mensagem:

```
O jogador 1 venceu o jogo!!! Pedro se salvou da forca!!
```

```
Jogar novamente?
```

```
1- sim
2- nao
```

Neste menu, a opção 2 encerra o jogo. Já a opção 1, sorteia uma nova palavra. Quem começa o jogo é o ganhador da última partida, no nosso exemplo, o Pedro.

Em caso de empate, como já citamos anteriormente, deve-se imprimir a seguinte mensagem:

```
Os jogadores 1 e 2 empataram!!! Pedro e Felipe se salvaram!!
```

```
Jogar novamente?
```

```
1- sim
2- nao
```

Neste caso, é feito o sorteio entre todos os jogadores para decidir quem inicia.

Uso de Structs

O trabalho também tem como objetivo o uso correto de structs e funções. Portanto, você deverá organizar seu código corretamente.

Exemplo: é o jogador que armazenará diversas informações que precisaram ser acessadas pelo programa, nome, pontuação, vitórias, etc. Logo é um struct obrigatório.

O próprio Jogo também é um candidato a ser um tipo definido pelo usuário, contendo jogadores, último vencedor, placar geral, etc.

Organize bem o seu código.

2. Arquivos de entrada

O sistema realiza a leitura de um arquivo `.csv`: o arquivo de palavras. Suas informações são separadas por vírgula simples.

Para o arquivo de palavras, são dispostas as informações

```
dica1,palavra1
dica2,palavra2
dica3,palavra3
```

onde dica e palavra são strings de, no máximo, 25 caracteres.

3. Regras Gerais

O trabalho deverá ser feito **em dupla** e pelos próprios alunos, isto é, os alunos deverão necessariamente conhecer e dominar todos os trechos do código implementados.

Cada dupla deverá trabalhar independente das outras, não sendo permitido a cópia ou compartilhamento de código. O professor irá fazer verificação automática de plágio. Trabalhos identificados como iguais, em termos de programação, serão penalizados com a nota zero. Isso também inclui a dupla que forneceu o trabalho, sendo, portanto, de sua obrigação a proteção de seu trabalho contra cópias ilícitas.

4. Entrega do Trabalho

O trabalho deverá ser submetido pelo Repl.it e também pelo AVA até as 23:59 do dia 10/03/2022.

O arquivo enviado pelo AVA deve seguir o padrão: TEP-2021_MATRICULA1_MATRICULA2.zip, substituindo a matrícula pelos respectivos números de matrícula da dupla.

O arquivo zip deve conter: `README.md` contendo nome da dupla e com considerações de projeto que julgue necessário, incluindo as decisões de projeto tomadas. Por exemplo, etc.

O código fonte deverá estar bem comentado, isto é, todas as structs e funções devem ter descrições claras do que ela faz, o que representa cada parâmetro e o que ela retorna (se houver). Pode-se adicionar também comentários que ajude a entender a lógica da questão.

5. Avaliação

O trabalho será corrigido em dois turnos: correção do professor e entrevista. A correção do professor (CP) gerará uma nota entre 0 e 10 e as entrevistas (E) serão realizadas posteriormente à data de entrega do trabalho e será atribuída uma nota entre 0 e 1.

Alunos que fizeram o trabalho em dupla realizarão a entrevista juntos, porém terão seus desempenhos na entrevista avaliados de forma individual. Pequenas e pontuais correções de código serão permitidas no momento da entrevista, o que poderá acarretar em uma CP maior.

A nota final (NF) do trabalho será dada pela seguinte fórmula: $NF = E * CP$

O trabalho será pontuado de acordo com sua funcionalidade e outros aspectos de implementação, conforme as listas abaixo.

Aspectos funcionais

1. Inicialização. Capacidade de carregar os arquivos. (10%)
2. Uso correto e eficiente de tipos definidos pelo usuário. (20%)
3. Uso correto e eficiente de funções. (20%)
4. Qualidade da documentação do código. (10%)
5. Jogo segue o fluxo do jogo descrito acima corretamente. (40%)

5.1 Penalizações (-20% sobre a nota do item 5 para cada falha). Lista, não exaustiva, de exemplos de falhas:

- Não permite 4 jogadores;
- Não circula corretamente entre jogadores;
- Não mostra vencedores corretamente;
- Não é possível reiniciar a partida após terminada;

Importante: Neste trabalho não é permitido o uso de bibliotecas, além das padrões. Por exemplo, `string.h`