

Trabalho Prático – Parte 2

O objetivo da parte 1 deste trabalho foi aprender a usar **estruturas de dados do tipo matriz, subalgoritmos (funções), macros do pré-processador, bibliotecas diferentes das utilizadas até então, variáveis globais de forma correta.**

Agora, na parte 2 do trabalho, o aluno deverá aplicar os conhecimentos sobre **registros (structs) e arquivos (tipos texto e binário)**

Especificação da segunda etapa:

1. Replay:

Ao selecionar a opção jogar no menu principal, o jogo agora terá a opção (1) de criar um arquivo de replay, (2) utilizar um arquivo de replay ou (3) para jogar o jogo sem criar ou utilizar um arquivo de replay. Um arquivo de replay consiste em um arquivo texto, que contém as combinações das peças que foram inseridas ao tabuleiro, em ordem.

- Ao escolher a primeira opção, após criar o arquivo, tem-se que a cada momento que uma peça aparece no tabuleiro, esta deve, também, ser gravada no arquivo criado. Quando o jogo termina, o arquivo deve ser fechado. O nome do arquivo (.txt) deve ser escolhido e informado pelo usuário.
- Ao escolher a segunda opção, deve ser inserido o nome do arquivo (com a extensão .txt) a ser utilizado como replay, retornando uma mensagem de erro caso o arquivo não exista. Após escolher o arquivo, ao invés da geração aleatória das peças, deverão ser utilizadas as peças presentes no arquivo texto, na mesma ordem em que estão no arquivo, até o momento em que este acaba. Quando todas as peças do arquivo forem lidas o jogo deve ser encerrado. Ao utilizar um arquivo de replay, sua pontuação não pode ser submetida ao ranking.
- Ao escolher a terceira opção, o usuário irá jogar o jogo sem criar nem utilizar arquivo de replay como feito na parte 1 do trabalho.

2. Novas opções do menu:

Nesta segunda etapa o aluno deverá implementar as opções 3 e 4 do Menu Principal da Parte 1 do trabalho prático. Ou seja: “3. CONFIGURAÇÕES” e “4. RANKING”.

A seguir é apresentado o detalhamento para cada opção.

2.1. Configurações

As configurações do jogo devem ser salvas em um arquivo texto, e carregadas toda vez que o jogo for iniciado. Ou seja, **toda vez que a opção JOGAR for selecionada o programa deverá consultar a configuração atual que está gravada no arquivo texto.** Caso o jogador faça alguma alteração na configuração atual, então a nova configuração é que passará a valer para a próxima jogada.

O menu de configurações deverá ter as seguintes opções:

1. Peças
2. Ativar Modo Rankeado
3. Voltar

Nas opção 1 deverão ser alteradas as seguintes variáveis:

- **pontuacao** – quantidade de peças conectadas necessárias para se pontuar. Mínimo: 3; máximo 10; default: 4 (atual da parte 1 do trabalho).
- **velocidade_descida** – tempo em segundos necessário para que o tabuleiro perca uma linha. Mínimo: 10; máximo 35; default: 20 (atual da parte 1 do trabalho).

Para cada variável alterada deve ser definido e informado na tela os valores limites. Além disso, o valor digitado deve ser validado e deve ser mostrada uma mensagem de erro caso o valor fique fora dos limites estabelecidos; neste caso a leitura deve prosseguir até que um valor válido seja digitado.

Na opção 2 o usuário poderá ativar o modo ranqueado, **que não utilizará as configurações gravadas no arquivo tipo texto.** Ao invés disto, este modo usará sempre os valores padrões abaixo:

- puntuacao: 5
- velocidade_descida: 15 segundos.

ATENÇÃO: A pontuação será salva no ranking somente quando o modo ranqueado estiver ATIVO.

2.2. Ranking

Cada vez que o jogador iniciar um jogo com o modo ranqueado ATIVADO deve ser exigido que ele digite seu nome (ou *nickname*), que será utilizado para identificá-lo no ranking. O nome máximo do jogador deve ser de 10 caracteres e no mínimo de 1 caractere. Isso deve ser testado e informado mensagem de erro caso o valor esteja fora dos limites. Enquanto a leitura não estiver correta deve ser lido novamente.

Após o final de cada partida devem ser gravados no **arquivo binário do ranking** os dados do jogador se e somente se ele tiver jogado no modo ranqueado. Abaixo são fornecidas as instruções sobre o registro de uma partida no ranking.

Quando o usuário consultar a opção “4. RANKING”, o arquivo binário contendo os dados do ranking deve ser lido e os 10 jogadores com melhor pontuação devem ter seus nomes e pontuação mostrados na tela, em ordem DECRESCENTE de pontuação. Após mostrar o Ranking, coloque na tela uma mensagem “Tecle <ENTER> para continuar!” e use um `getchar()`; no programa para segurar a tela e assim que usuário visualizar o ranking.

Registro de uma partida no ranking:

O aluno deverá implementar a função sugerida `registerMatch()` para registrar a pontuação obtida durante a partida em um arquivo binário com as informações de ranking. O jogo deverá guardar apenas o ranking dos 10 melhores resultados em ordem DECRESCENTE.

Para implementar esta função, será necessário definir o seguinte tipo:

```
typedef struct {  
    char nick[11];  
    int score;  
} Player;
```

O nome do arquivo binário de ranking deve ser: `ranking.bin`

Utilizando o tipo definido, crie a função `registerMatch()` com a seguinte lógica. Tentar abrir o arquivo para leitura (opção “rb”);

- caso não seja possível, então criar e abrir o arquivo para escrita (opção “wb”) e gravar um vetor de 10 posições do tipo `Player`, em que o primeiro elemento descreve o resultado da partida que acabou de terminar e os próximos 9 elementos devem possuir a string `nick` vazia, ou seja, com `'\0'` na primeira posição, e encerrar a função;
- caso seja possível abrir o arquivo para leitura (opção “rb”), utilizar o arquivo aberto para ler um vetor de 10 posições do tipo `Player`; após ler o vetor, a função deve procurar uma posição para inserir um registro com o resultado da partida que acabou de terminar; se foi possível inserir o registro, abrir o arquivo para escrita e escrever o novo vetor de 10 posições. Lembrando para tomar cuidado para não sobrescrever em cima do registro anterior, ou seja, o novo registro de ranking deve ser INSERIDO na posição adequada, e caso o vetor já esteja com 10 elementos, o último (ou de menor pontuação) será descartado da lista do ranking.

Observe que se a pontuação obtida for menor do que a última pontuação do ranking, esta NÃO deverá ser gravada no ranking atual.

3. Loop de menu

É importante clarificar que, nesta parte, é necessário que após a entrada em qualquer uma das opções do menu (Jogar, Instruções, Configurações e Ranking), é necessário que o jogo volte para a tela de menu após a execução desta opção. É recomendado que no fim da execução da opção, seja mapeada alguma tecla para que, ao pressioná-la, se retorne ao menu principal. **É importante não esquecer de deixar evidente na tela de retorno qual tecla será utilizada.**

IMPORTANTE: deverá ser enviado também um arquivo texto chamado **README (.txt)** contendo as informações sobre alterações e extras que forem implementados, se for o caso.

Observações Gerais:

1. Incluir cabeçalho como comentário (ou seja, entre `/* */`), no programa fonte, de acordo com os **critérios de avaliação dos trabalhos (Disponível no Moodle)**.
2. A data de entrega do programa é: **12/07/2019 (6a-feira) até às 23:55 hs.**