

Segunda Lista - Módulo 1
Tipo 2
Jogo da Força e Tratamentos de Dados

Márcio Fantini Miranda
Programação no Octave
Cursos Técnicos
Automação e Eletrônica

4 de julho de 2021

Regras

1. Conceitos necessários:
 - Vetores e matrizes no Octave. Multiplicação de matrizes e de matriz por vetor.
 - Loops for e while. Strings.
 - Entrada e saída com arquivos.
 - Tipos de dados e suas conversões. Strings.
 - Ajuste polinomial. Uso das funções `polyfit()` e `polyval()`
2. Essa lista contém duas tarefas principais.
3. Essa lista, além das tarefas principais que devem ser entregues, contém explicações, exemplos e exercícios de fixação. Todos devem ser usados como preparação para a execução da tarefa. Não é necessário enviar os exercícios de fixação, os exemplos ou scripts que porventura tenha feito para se preparar para a execução da tarefa.
4. Deve-se enviar o zip com todos os arquivos necessários para a execução das duas tarefas. O nome do arquivo zip deve ser formado pelo seu nome e sobrenome, sem espaços, assentos, etc.), caso seja um trabalho individual e com os primeiros nomes dos integrantes dos grupo, caso seja trabalho em dupla ou tripla.
5. Coloque comentarios no seus scritps. Todo script deve começar com um cabeçalho, contendo seu nome (ou nomes dos integrantes) e o número da questao.
6. Organize seus códigos e programas. Planeje seus programas. Organize-os em scripts e funções.
7. Para enviar a tarefa lembre-se: envie os scripts que fazem parte do programa da tarefa. Você pode optar por organizar em funções e scripts ou fazer em um único arquivo.
8. O trabalho deve ter um programa principal que chama, usando um menu, pelas duas tarefas, mais uma opção de sair.

Capítulo 1

Jogo da Forca

Nessa tarefa você deve fazer um menu para chamar os jogos da forca da versão 3 e versão 4 (explicados abaixo). O jogo da forca é aquele em que um jogador escolhe uma palavra e pede para o outro acertá-la, informando as letras (ou podendo fazer um chute a qualquer momento). O jogador que está acertando tem um número limitado de tentativas, antes de ser "enforcado". Nesse jogo no Octave/Matlab não é necessário fazer o desenho da pessoa e da forca (como fazemos no papel).

A sugestão aqui é fazer jogos em diferentes versões para você ir pensando no problema e ir fazendo as partes mais simples primeiro, para no final ter um jogo com mais recursos.

As versões 1 e 2 então servem como preparação para as partes 3 e 4, que são o objetivo dessa tarefa.

1. **Jogo da Forca, versão 1: Dois jogadores.** Esse jogo, na versão 1, deve ser jogado com duas pessoas. O primeiro jogador define uma palavra que deve ser acertada pelo segundo jogador.

- Entrada: palavra a ser acertada pelo segundo jogador.
- Saída: mensagens de erro e/ou apresentação da palavra em formação

Detalhamento:

- O primeiro jogador deve entrar com uma palavra, que será armazenada pelo computador. Em seguida o jogo inicia, com o segundo jogador tentando acertar a palavra "secreta". O jogo consiste num *loop* que fica sendo executado enquanto as tentativas para acertar a palavra não terminarem ou até o jogador acertar a palavra.

- O jogador que deseja acertar a palavra tem que informar, a cada vez, uma letra. Se a palavra tiver a letra o jogador não perde a tentativa. Não havendo a letra ele perde a tentativa. Ele tem um número fixo de tentativas. A cada vez ele pode “chutar” a palavra toda. Se errar perde o jogo.
- Essa dinâmica de ler a letra e verificar se ela pertence à palavra deve continuar enquanto as tentativas não terminarem ou até o jogador arriscar a palavra toda.
- A cada iteração o computador deve mostrar o que já foi completado pelo usuário, ou seja, na medida em que a palavra for sendo preenchida ela deve ser mostrada. Ao mostrar a palavra incompleta, coloque caracteres que representem o espaço vazio.

2. Jogo da Forca, versão 2: Homem x Máquina

- Nessa variante do jogo anterior, quem escolhe a palavra deve ser o computador. Portanto você deve programar um banco de dados com várias palavras e o programa, no início, deve sortear uma delas.
- A dinâmica do programa é a mesma do anterior.
- O programador deve criar um arquivo texto com as palavras que serão sorteadas.

3. Jogo da Forca, versão 3: Homem x Máquina

- Incremente o programa anterior para que ele fique rodando enquanto o usuário desejar. Ou seja ao final de cada jogo o programa deve perguntar se o usuário quer parar ou continuar. O programa pode então anotar quantas vitórias e derrotas o jogador teve numa série.

4. Jogo da Forca, versão 4: Homem x Máquina

- Pode-se melhorar mais ainda a versão 2 fazendo um banco de dados com a estatística de cada usuário. Essa banco de dados deve ser criado em arquivo(s) texto.

Dicas e Recursos

Esses programas requerem recursos para trabalhar com strings. Lembre-se que uma string é um vetor de caracteres. Cuidado com as letras maiúsculas

e minúsculas, pois as duas devem ser aceitas indistintamente. Atenção na lógica de buscar a palavra sorteada no arquivo (que é o seu “banco de dados”).

QUando falamos em ter um “banco de dados” com as palavras a serem sorteadas, estamos nos referindo a é um arquivo texto. Você pode criar um script (ou função) para carregar palavras no seu “banco de dados”.

Capítulo 2

Tratamento de Dados e Ajustes de Curvas

2.1 Corrida

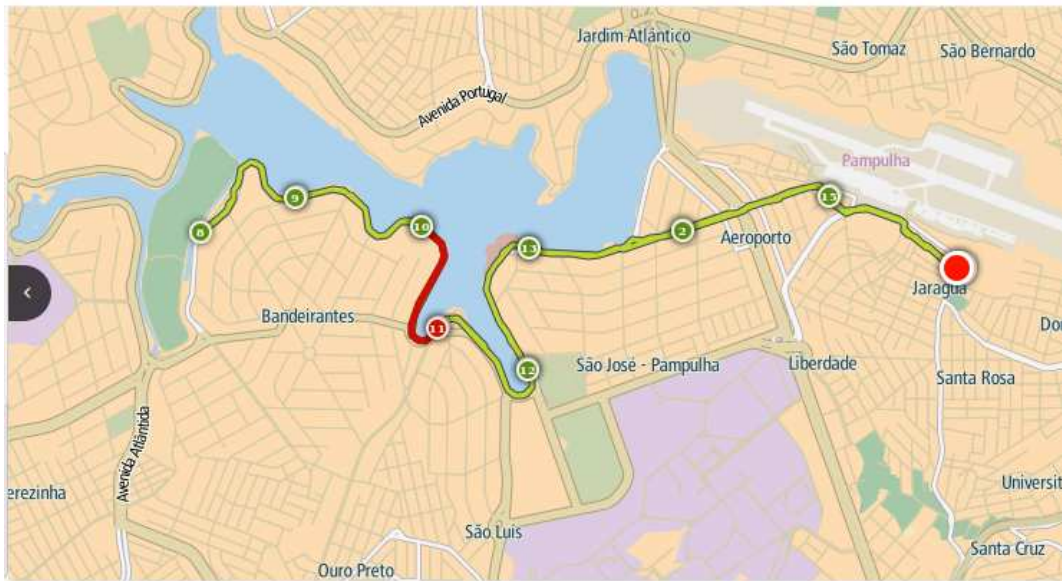


Figura 2.1: Mapa mostrando a corrida realizada em 16 trechos de 1Km (aproximadamente). Destaca-se (em vermelho) o trecho 10-11, da volta. A partir dos dados dessa corrida são feitas várias análises

A figura 2.1 apresenta o trecho de uma corrida, que foi registrada pelo aplicativo mysports, que salva os dados de tempo, distância, velocidade ins-

tantânea, calorias, elevação e batimento cardíaco. Esses dados são dados em 6 tabelas, nos arquivos `corrida1.dat` (do tempo 0 até o tempo 3355 segundos) e `corrida2.dat`, dentro do arquivo `corrida1.zip`. A seguir é pedido um estudo desses dados. Faça um script (ou quantos achar necessário) para responder às questões.

Seu(s) script(s) deve fazer:

1. Carregue os dois arquivos, cada um em uma matriz (digamos $M1$ e $M2$). Crie uma única matriz, concatenando $M1$ com $M2$. Chame-a, por exemplo, de M .
2. Atenção: O que cada coluna (de $M1$ e $M2$) informa foi explicado acima. A coluna de tempo está em segundos, a coluna de distância está em metros, a coluna de velocidade está em m/s , de calorias em Kcal, de elevação em metros e de batimento cardíaco está em batimentos por minuto (BPM).
3. Seu script deve informar:
 - (a) Qual o tempo total gasto na corrida?
 - (b) Qual a distância total percorrida?
 - (c) Qual a velocidade média total da corrida?
 - (d) Qual a variação da elevação máxima que ocorreu durante a corrida?
 - (e) Qual a caloria total gasta na corrida?
 - (f) Qual o maior BTM?
 - (g) Em qual (ou quais) instante(s) ocorreu o máximo BTM?
 - (h) Quanto tempo durou o máximo BPM? (em segundos)?
 - (i) Qual o BPM médio da corrida?
 - (j) Qual a elevação média? e a máxima? e a mínima?
4. Plotar os gráficos pedidos (você decide a melhor forma: se cada um em uma figura, se dois gráficos na mesma figura (usando `subplot()`), etc). Coloque sempre label nos eixos e título no gráfico:
 - (a) $t \times d$ (tempo em segundos, distância em metros).
 - (b) $t \times d$ (tempo em horas, distância em metros).
 - (c) $t \times d$ (tempo em horas, distância em Km).

- (d) $t \times v$ (t em segundos e velocidade instantânea, dada na coluna 3, em m/s).
 - (e) $t \times v$ (t em segundos e velocidade instantânea, agora convertida para Km/s).
 - (f) $t \times$ calorias (t em segundos e calorias em Kcal)
5. Calcular a velocidade instantânea a partir dos dados do tempo (s) e da distância (m). Plote o vetor gerado em função do tempo (s). Compare com a velocidade instantânea dada na coluna 3. Plote as duas velocidades (a calculada e a informada na coluna 3) no mesmo gráfico.
 6. Calcular o erro entre as duas velocidades e plote o gráfico desse erro.
 7. Criar um vetor de aceleração instantânea que pode ser calculado a partir da equação (2.1)

$$a(i) = \frac{v(i+1) - v(i)}{t(i+1) - t(i)} \quad (2.1)$$

- (você pode usar a velocidade calculada ou a informada na coluna 3).
8. Obter os polinômios de ajuste para as relações temporais da distância, velocidade e aceleração.
 9. Plotar o resultado dos polinômios em função do tempo no mesmo gráfico dos dados reais (para cada caso separadamente, obviamente).