

## Trabalho Prático - Roda a Roda Jequití

### PLE 2021-1

Você deverá implementar um programa que simule o jogo Roda a Roda Jequití, transmitido diariamente pela emissora SBT. O jogo é uma extensão da Forca, clássico jogo onde, dado o número de letras de uma palavra secreta, o jogador deve escolher letras que ele imagine que componham a palavra. Se ele acertar, todas posições da palavra que possuem aquela letra são reveladas. Se errar, ele deve escolher uma nova letra, podendo ocorrer no máximo  $n$  erros.

No Roda a Roda Jequití, três participantes giram uma roleta. Essa roleta pode 1) dar uma certa quantia de pontos para o jogador, 2) passar a vez ou 3) perder todos os pontos. Se ele conseguir uma pontuação, ele deve escolher uma letra qualquer de três palavras que pertençam a um tema específico. Assim como na versão clássica, se o jogador acertar a letra, os locais onde a letra aparece são revelados. Se ele errar, passa a vez para o próximo jogador.

Quando faltarem três letras, o jogador da vez tem a possibilidade de dizer as três palavras. Se ele acertar, ganha uma pontuação bônus. Ao final de três rodadas, o jogador com a maior pontuação ganha a competição e tem a possibilidade de participar da rodada final. Nela, ele escolhe um número de 1 a 10 que contém uma palavra pertencente a um tema. Ele deve então escolher 5 consoantes e uma vogal de uma única vez. Com base nas escolhas, as letras que pertençam à palavra serão reveladas. O jogador deverá então tentar acertar a palavra. Se acertar, ele dobra a pontuação que obteve na competição.

#### Regras de implementação:

1) seu programa irá trabalhar um dicionário chamado `base_de_dados`, cujas chaves são strings com os temas das palavras e os valores são listas de strings contendo as palavras daquele tema. O conteúdo do dicionário será disponibilizado por mim.

2) dado o dicionário, seu programa deve realizar dois tipos de sorteio: um tema dentre os possíveis e três palavras dentro do tema.

3) o programa deve respeitar as regras de alteração de jogadas. Os nomes dos jogadores estão definidos como Ana, Bárbara e Carlos, e as passagens são definidas da seguinte forma: Ana -> Bárbara, Bárbara->Carlos e Carlos->Ana. O nome do jogador deverá ser exibido a cada turno e o programa deve mostrar informações de rodada e turno.

4) o turno começa com a roleta sendo girada. Os resultados da roleta são definidos a seguir: pontos de 100 a 950, passo 50 uma vez cada. Duas vezes o valor 1000; duas vezes Passa a Vez; duas vezes Perdeu Tudo. Os resultados são equiprováveis.

5) com base no resultado, o contador de pontos do jogador deve ser atualizado de acordo.

6) o resultado "Passa a Vez" troca do jogador ativo e roda a roleta novamente. O resultado "Perdeu Tudo" zera a pontuação e passa a vez.

7) as palavras devem ser representadas por uma sequência de caracteres `_` (underscore), tantos quantos forem os caracteres da palavra. No terminal, deve ser mostrado uma sequência de `_` por linha.

8) após as exibições solicitadas pelos itens anteriores, o jogador deve escolher uma letra, que pode ser digitada minúscula ou maiúscula. Seu programa deve verificar se a letra está ou não nas palavras sorteadas. Em todo índice que a letra aparecer, o `_` exibido deve ser substituído pela letra em formato maiúsculo.

9) se a letra estiver em pelo menos um índice, a roleta deve ser girada novamente. Se não estiver em nenhuma palavra, o jogador passa a vez para o próximo. As letras erradas que foram escolhidas devem sempre estar visíveis. Se a letra já tiver sido escolhida anteriormente, o programa deve informar um erro e pedir uma nova escolha.

10) quando faltarem 3 ou menos letras para serem escolhidas, o jogador da vez tem a possibilidade de dizer quais são as palavras. Seu programa deve passar a perguntar se o jogador quer ou não dizer as palavras. Se ele quiser, o programa deve perguntar palavra a palavra qual a resposta do jogador. Se ele não quiser, deve seguir o procedimento da roleta. Caso o resultado leve a outro jogador, o mesmo terá a chance de dizer as três palavras.

11) Após 3 rodadas de jogo, o programa deve computar os pontos e verificar o vencedor da competição. Esse vencedor deverá participar da rodada final. Um segundo dicionário chamado rodada\_final será utilizado. Nele, as chaves são os temas da palavra, enquanto os valores são as palavras.

12) o programa deverá solicitar 5 consoantes e 1 vogal de uma única vez e computar o resultado das escolhas, substituindo os \_ pelas letras certas. A seguir, deve pedir para o usuário digitar a resposta final. Se ele acertar, o programa deve dobrar os pontos e informar a vitória. Se errar, deve informar a derrota e qual era a palavra correta.

13) Seu código deve ser capaz de analisar entradas inválidas. Se o usuário cometer algum erro na escolha, o programa deve informar o erro e pedir uma nova entrada. O código também precisa ignorar a entrada com caracteres maiúsculos e minúsculos. Além disso, caracteres especiais devem (acentos, hífen, etc) devem ser retirados.

14) o código deve ser modularizado e documentado de acordo com as diretrizes que foram trabalhadas ao longo do curso. A qualidade da sua implementação será avaliada e pontuada.

15) o trabalho é individual. Qualquer indício de cola resultará em nota zero tanto para quem passou a quem recebeu.

#### **Exemplo de layout de exibição por turno**

```
+=====
| RODADA X - TURNO Y
+=====
| ANA - XXXXX | BARBARA - YYYYY | CARLOS - ZZZZZ
+=====
Jogador ativo: <nome_jogador>
Pontuação atual: <pontuacao>
Roleta: <resultado>
Nova pontuacao: <pontuacao_atualizada>
+=====
Tema: <tema_das_palavras>
P1) _ _ _ _ _
P2) _ _ _ _
P3) _ _ _ _
Letras erradas: <lista_de_letras_erradas>
+=====
```

#### **Pontuação bônus**

Tratamento de erros fora da especificação

Dinamismo da interface

Montagem da base de dados a partir de arquivos

Interface gráfica

Outros: me surpreendam :)