PROGRAMAÇÃO ORIENTADA A OBJETOS – POO
PROVA P1

ATENÇÃO

- Todas as questões devem ser feitas a partir do repositório base do projeto MenuTemplate disponível no link <**https://github.com/Hasenfresser/MenuTemplate**>
- Cada aluno deve criar um Fork a partir do repositório base para o seu repositório local
- Ao término, o aluno deverá enviar APENAS um arquivo para o repositório do Prof. Erik contendo um **README.md** com as seguintes informações:
  - Nome do aluno
  - Link para o repositório
  - Descrição detalhada sobre como as questões foram solucionadas
  - Enviar o **README.md** para a estrutura de pasta contendo o seguinte padrão
    - **https://github.com/aceiro/poo2019/prova_p1/[NOME-DO-ALUNO]**
    - **Opcional (https://guides.github.com/features/mastering-markdown)**
  - **DATA DA ENTREGA 02/05/2019 ATÉ AS 23:59 HORAS**

O projeto **MenuTemplate** para Engines de Games 2D/3D é uma biblioteca escrita em C++ orientada a objetos que suporta – eventos, exibição de cursor em formato ASCII e alocação dinâmica em STL. Suponha que você foi contratado como programador para dar manutenção nesse projeto e também desenvolver algumas novas funcionalidades para uma nova versão do Game "Neverwinter Nights" (http://nwn.beamdog.com/). Assim, pede-se:

Q1) (1,0 ponto) Adicionar para o menu principal as seguintes opções: **Video Options; Sound Options; Controls.**

Q2) (1,0 ponto) Modificar a descrição do Menu do game para **"Neverwinter Nights Simple Menu"**

Q3) (1,0 ponto) A classe **MenuTemplate** possui inúmeros métodos (ver arquivo **MenuTemplate.hpp**). Contudo, nem todos os métodos foram implementados. Um deles é o método **getCursor()** que retorna a string associada com o cursor definido inicialmente. Sua atividade aqui nessa questão é implementar essa rotina **getCursor** para retornar e então exibi-la no programa **main.cpp**.

Q4) (1,0 ponto) Adicione para a classe **MenuTemplate** a possibilidade de contar a quantidade de caracteres que existem na **string do cursor do menu**. Para isso, crie um atributo e um método que armazene e retorne respectivamente a quantidade de caracteres existente na string do cursor. O nome da variável e método faica a critério do aluno.

Q5) (3,0 ponto) Por padrão, a biblioteca MenuTemplate monta um menu de opções que exibe caracteres ASCII através da String definida em **setCursor**. Para melhorar a apresentação do Menu é importante poder adicionar outros tipos de caracteres. Nesse caso, sua tarefa aqui é definir um método adicional para a classe **MenuTemplate** de tal forma que essa classe possa suportar além de caracteres ASCII (comportamento default) o uso de caracteres Unicode UTF-8 (vide Tabela abaixo)

U+2662	♢	\xe2\x99\xa2	WHITE DIAMOND SUIT
U+2663	♣	\xe2\x99\xa3	BLACK CLUB SUIT
U+2664	♤	\xe2\x99\xa4	WHITE SPADE SUIT
U+2665	♥	\xe2\x99\xa5	BLACK HEART SUIT
U+2666	♦	\xe2\x99\xa6	BLACK DIAMOND SUIT
U+2667	♧	\xe2\x99\xa7	WHITE CLUB SUIT
U+2668	♨	\xe2\x99\xa8	HOT SPRINGS
U+2669	♩	\xe2\x99\xa9	QUARTER NOTE

(https://www.utf8-chartable.de/unicode-utf8-table.pl?start=9728&number=128&utf8=string-literal)

Assim, pede-se que seja criado um novo método **MenuTemplate::setCursor(const string &Cursor, const bool isUnicode)** onde Cursor é uma string em Unicode e **isUnicode** é uma flag para controlar a exibição em unicode. Use a variável booleana isUnicode para controlar ou não a exibição em ASCII ou UNICODE. Para carregar o código UNICODE em string C++ use o exemplo abaixo

```
const string heart = "\xe2\x99\xa5 "
```

Q6) (3,0 ponto) A implementação atual do **MenuTemplate** considera a inserção de um novo elemento sempre no final do vetor (vide método – **MenuTemplate ::addEntry**). Em PCs com pouca capacidade de processamento isso pode ser um problema uma vez que pode leva a um uso excessivo da memória para além do limite permitido. Para resolver esse problema, uma alternativa é o uso de uma estrutura de dados do Tipo Vetor Circular – i.e., um novo elemento é sempre adicionado na cabeça do vetor considerando-se que a cabeça do vetor sempre será o item menos usado. Assim, para essa questão modifique a implementação do método **addEntry** para que esse método passe a user um vetor circular.