Projeto Árvores Binárias - MP 2018/1

Nesse projeto está contida uma biblioteca de árvores binárias representada pelos arquivos **arvore.hpp** e **arvore.cpp** e um Jogo de 20 Perguntas no arquivo **jogo_20_perguntas.cpp**.

A biblioteca foi desenvolvida 100% na linguagem *C++* com padrao **C++17** e técnica de desenvolvimento orientada a testes utilizando o framework Catch2, o módulo de testes está no arquivo **testa_arvore.cpp**.

O desenvolvimento foi focado em um uso genérico da biblioteca de forma que fosse possível armazenar diversos tipos de dados nas células da árvore, para isso, foram utilizados templates do **C++** como uma forma de metaprogramação.

A biblioteca teve sua compilação testada em sistema Ubuntu 17.10, Solus Linux e macOS usando compilador **g++** de versão 7.0 ou superior.

Na pasta **doc** está contida uma versao em PDF deste documento e outro com a descrição dos testes.

Acesso às células da árvore:

O acesso às celulas da árvore nao é feito de forma direta pelo usuário. Para ler e modificar os
valores das células da árvore, o usuário deverá usar as funcoes **LerCelula** e **ModificarCelula**
da classe **ArvoreBinaria**.

Para facilitar o acesso, foi usado um sistema de coordenadas
de **“nível”** e **“posição”**:

"Nivel" representa os níveis (ou andares) da árvore binária, sendo nível 0 a raíz da arvore.
“Posição” representa a posição da célula em relação a cada nível, sendo a posição mais à
esquerda de cada nível considerada posição 0.

```
Exemplo: (Pos. indica Posicao)
Nivel 0              (Pos.0)
                   /         \
Nivel 1         (Pos.0)     (Pos.1)
               /     \      /     \
Nivel 2    (Pos.0)(Pos.1) (Pos.2) (Pos. 3)
...                   ...
```

Arquivo de jogo:

O arquivo que guarda o estado do jogo é um arquivo de texto puro que contém, em cada linha, um

comando de insercao de uma célula na árvore que gerencia o jogo.

Os comandos de inserçao seguem a seguinte sintaxe: **(nivel, posicao)=“frase”**.

Exemplo:

```
(0,0)="Seu objeto é vermelho?"
(1,0)="Seu objeto é uma fruta?"
(1,1)="Maçã"
            .
            .
            .
```

Como Compilar:

Para compilar o módulo de teste é necessário ter o arquivo **catch.hpp** no diretorio padrao de bibliotecas do *C++*.

Em seguida, deve se executar o seguinte comando:

```
$ cd src
$ make testa_arvore
```

Será gerado um arquivo executável **testa_arvore** na pasta raiz do diretório projeto1mp2018.

Para compilar o jogo das 20 perguntas:

```
$ cd src
$ make jogo_20_perguntas
```

Será gerado um arquivo executável **jogo_20_perguntas** na pasta raíz do diretório projeto1mp2018.

Como Usar:

Módulo de Teste de Arvore.hpp

Para rodar os teste especificados no arquivo *testa_arvore.cpp*, deve se executar o arquivo **testa_arvore**.

Jogo 20 Perguntas

O jogo começa ao rodar o executável **jogo_20_perguntas**, o jogo pode tanto abrir/salvar um arquivo de árvore
especificado pelo usuário, quanto abrir/salvar o arquivo padrao **jogo_padrao.txt**.

Para ler e salvar o estado do jogo no arquivo padrao (**jogo_padrao.txt**), o usuário deve executar
jogo_20_perguntas sem parâmetros na linha de comando:

```
$ ./jogo_20_perguntas
```

Para ler e salvar o estado do jogo em um arquivo específico, o usuário deve executar
jogo_20_perguntas seguido do nome do arquivo que deseja ler e salvar o estado do jogo:

```
$ ./jogo_20_perguntas arquivo.txt
```

Para ler e salvar o estado do jogo em arquivos diferentes, o usuário deve executar
jogo_20_perguntas seguido do nome do arquivo que deseja ler e do arquivo no qual
se deseja salvar o estado do jogo:

```
$ ./jogo_20_perguntas arquivo_entrada.txt arquivo_saida.txt
```

Nao é possível rodar o jogo sem um arquivo de árvore válido.

Como Jogar:

```
O usuário deve executar o jogo e responder as perguntas feita
 pelo programa com SIM ou NAO.

No final do jogo, caso o computador não tenha acertado o obje
to do usuário, o programa perguntará
qual o objeto escolhido e pedirá para o usuário inserir uma p
ergunta nova que ajude o computador
a adivinhar o objeto nas próximas jogadas.
```

```
A arvore será salva em um arquivo como especificado anteriorm
ente.
```