PLANEJAMENTO DA TERCEIRA CAMADA

Trabalho 2 - Modular(INF1301)
Data de entrega: 03/05/2018
Turma: 3WB

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Por: Ana Carolina Da Hora
Felipe Alexandre Metson
Michel Anísio Almeida

```
inserir cubo.h
************* INÍCIO DA FUNÇÃO EXIBE ************
interna exibeMov(char *mov)
{
	se (mov == “cimadir”)
		print “Gire a face de cima do seu cubo para a direita\n” ;
	senão (mov == “cimaesq”)
		print “Gire a face de cima do seu cubo para a esquerda\n” ;
	senão (mov == “baixodir”)
		print “Gire a face de baixo do seu cubo para a direita\n” ;
	senão (mov == “baixoesq”)
		print “Gire a face de baixo do seu cubo para a esquerda\n” ;
	senão (mov == “esqtra”)
		print “Gire a face da esquerda do seu cubo para tras\n” ;
	senão (mov == “esqfre”)
		print “Gire a face da esquerda do seu cubo para frente\n” ;
	senão (mov == “dirtra”)
		print “Gire a face da direita do seu cubo para tras\n” ;
	senão (mov == “dirfre”)
		print “Gire a face da direita do seu cubo para frente\n” ;
	senão (mov == “fredir”)
		print “Gire a face da frente do seu cubo para a direita\n” ;
	senão (mov == “freesq”)
		print “Gire a face da frente do seu cubo para a esquerda\n” ;
	senão (mov == trasdir)
		print “Gire a face de tras do seu cubo para a direita\n” ;
	senão (mov == trasesq)
		print “Gire a face de tras do seu cubo para a esquerda\n” ;
}
************* FIM DA FUNÇÃO EXIBE ************

************* INÍCIO FUNÇÃO FAZ CRUZ **********

fazCruz()
{
	procurar primeira borda();
	procurar segunda borda();
	procurar terceira borda();
	procurar quarta borda();
	se(bordas estiverem na posição)//Checa se peças ja estão na posição
	{
		saia da função
	}
	se (nenhuma estiver na face de baixo)
	{
		gira frente horário();
		exibeMov(“fredir”);
		gira esquerda horário();
		exibeMov(“esqfre”);
		gira baixo anti-horário();
		exibeMov(“baixoesq”);
		gira esquerda anti-horário();
		exibeMov(“esqtra”);
		gira baixo horário();
		exibeMov(“baixodir”);
		gira frente anti-horário();
```

```
        exibeMov(“freesq”);
}
se (duas bordas estiverem na face de baixo
                        && em formato de L)
{
        se (a peça não estiver na face da frente
                        && na face da esquerda)
        {
                gira frente horário();
                exibeMov(“fredir”);
                gira esquerda horário();
                exibeMov(“esqfre”);
                gira baixo anti-horário();
                exibeMov(“baixoesq”);
                gira esquerda anti-horário();
                exibeMov(“esqtra”);
                gira baixo horário();
                exibeMov(“baixodir”);
                gira frente anti-horário();
                exibeMov(“freesq”);
        }
        se (a peça não estiver na face da esquerda
                        && na face da tras)
        {
                gira esquerda horário();
                exibeMov(“esqfre”);
                gira trás horário();
                exibeMov(“trasdir”);
                gira baixo anti-horário();
                exibeMov(“baixoesq”);
                gira trás anti-horário();
                exibeMov(“trasesq”);
                gira baixo horário();
                exibeMov(“baixodir”);
                gira esquerda anti-horário();
                exibeMov(“esqtra”);
        }
        se (a peça não estiver na face da tras
                        && na face da direita)
        {
                gira trás horário();
                exibeMov(“trasdir”);
                gira direita horário();
                exibeMov(“dirtras”);
                gira baixo anti-horário();
                exibeMov(“baixoesq”);
                gira direita anti-horário();
                exibeMov(“dirfre”);
                gira baixo horário();
                exibeMov(“baixodir”);
                gira trás anti-horário();
                exibeMov(“trasesq”);
        }
        se (a peça não estiver na face da direita
                        && na face da frente)
        {
                gira direita horário();
                exibeMov(“dirtras”);
                gira direita horário();
```

```
                exibeMov(“dirtras”);
                gira baixo anti-horário();
                exibeMov(“baixoesq”);
                gira direita anti-horário();
                exibeMov(“dirfre”);
                gira baixo horário();
                exibeMov(“baixodir”);
                gira trás anti-horário();
                exibeMov(“trasesq”);
        }

}
senão (duas bordas estiverem na face de baixo
                && em formato de linha)

{
        se (existe uma borda na camada da frente)
        {
                gira frente horário();
                exibeMov(“fredir”);
                gira esquerda horário();
                exibeMov(“esqfre”);
                gira baixo anti-horário();
                exibeMov(“baixoesq”);
                gira esquerda anti-horário();
                exibeMov(“esqtra”);
                gira baixo horário();
                exibeMov(“baixodir”);
                gira frente anti-horário();
                exibeMov(“freesq”);
        }
        se (existe uma borda na camada da esquerda)
        {
                gira esquerda horário();
                exibeMov(“esqfre”);
                gira tras horário();
                exibeMov(“trasesq”);
                gira baixo anti-horário();
                exibeMov(“baixoesq”);
                gira tras anti-horário();
                exibeMov(“trasdir”);
                gira baixo horário();
                exibeMov(“baixodir”);
                gira esquerda anti-horário();
                exibeMov(“esqtra”);
        }
        se (existe uma borda na camada de trás)
        {
                gira trás horário();
                exibeMov(“trasdir”);
                gira direita horário();
                exibeMov(“dirtras”);
                gira baixo anti-horário();
                exibeMov(“baixoesq”);
                gira direita anti-horário();
                exibeMov(“dirfre”);
                gira baixo horário();
                exibeMov(“baixodir”);
                gira trás anti-horário();
```

```
            exibeMov(“trasesq”);
      }
      se (existe uma borda na camada de direita)
      {
            gira direita horário();
            exibeMov(“dirtras”);
            gira frente horário();
            exibeMov(“fredir”);
            gira baixo anti-horário();
            exibeMov(“baixoesq”);
            gira frente anti-horário();
            exibeMov(“freesq”);
            gira baixo horário();
            exibeMov(“baixodir”);
            gira direita anti-horário();
            exibeMov(“dirfre”);
      }
se(bordas não estiverem na posição) /*Checa a necessidade de efetuar
                              * novamente a função*/
      fazCruz();

      }

}
************* FIM DA FUNÇÃO FAZ CRUZ ************

************* INÍCIO FUNÇÃO COMPLETA FACE *******

completaFace();
{
      procurar a primeira quina();
      procurar a segunda quina();
      procurar a terceira quina();
      procurar a quarta quina();
      se(quinas estiverem na posição)//Checa se peças ja estão na
posição
      {
            saia da função
      }
      /* Como ja foi ressaltado antes, é muito importante aplicar o
algarimo de sune tomando as referências corretas */
      se (faltar quina opostas) /* face da frente sendo aquela que tem
a                               * borda da diagonal superior direita
                                * da cor desejada */
      {
            algaritmo sune();
            algaritmo sune();
            algaritmo sune();
      }
      se ( faltar quina paralelas
            && elas estão na mesma face) /* face da frente sendo
aquela                                    * que tem a borda da
diagonal                                  * superior esquerda e
direita*/    {
            algaritmo sune();+U’
            algaritmo sune();
            algaritmo sune();
      }
      se ( faltar quina paralelas
```

```
                && elas estão faces diferentes)/* face da frente sendo
        aquela                                          * que tem a borda
        da diagonal                                     * superior esquerda
        */
              {
                algaritmo sune();+U
                algaritmo sune();
                algaritmo sune();
              }
              se ( faltam todas as quina
                && duas estão na mesma face
                && existe uma face sem amarelas)/* face da frente sendo
        aquela                                          * que tem a borda
        da diagonal                                     * superior direita
        */
              {
                algaritmo sune();
                algaritmo sune();
        }
        se ( faltam todas as quina
              && não estão na mesma face
              && existe uma face sem amarelas))/* face da frente sendo
                              * aquela que tem a borda da
                        * diagonal superior esquerda */
        {
              algaritmo sune();+U'
              algaritmo sune();
        }
        se ( faltam 3 as quina
              && quina estão na diagonal esquerda)/* face da frente sendo
                                    * aquela não tem borda*/
        {
              algaritmo sune();+U+U
              algaritmo sune();
        }
        se ( faltam 3 as quinas
              && quinas estão na diagonal direita)/* face da frente sendo
                                    * aquela que tem a borda da
                              * diagonal superior direita e
                        * que a face a sua direita
                                          * tenha na diagonal superior
                                          *direita outra borda */
        {
              algaritmo sune();+U+U
              algaritmo sune();
        }
        se(quinas não estiverem na posição) /*Checa a necessidade de efetuar
                                    * novamente a função */
        {
              completaFaces();
        }


}
********** FIM DA FUNÇÃO COMPLETA FACES *********

********** INÍCIO FUNÇÃO RESOLVE QUINAS *********
resolveQuinas()
{
```

```
procurar a primeira quina();
procurar a segunda quina();
procurar a terceira quina();
procurar a quarta quina();
se(quinas estiverem na posição)//Checa se peças ja estão na posição
{
	saia da função
}

se (face da frente tem duas quinas)
{
	gira esquerda horário();
	exibeMov(“esqfre”);
	gira trás anti-horário();
	exibeMov(“trasdir”);
	gira esquerda horário();
	exibeMov(“esqfre”);
	gira frente horário();
	exibeMov(“fredir”);
	gira frente horário();
	exibeMov(“fredir”);
	gira esquerda anti-horário();
	exibeMov(“esqtra”);
	gira trás anti-horário();
	exibeMov(“trasdir”);
	gira esquerda horário();
	exibeMov(“esqfre”);
	gira frente horário();
	exibeMov(“fredir”);
	gira frente horário();
	exibeMov(“fredir”);
	gira esquerda horário();
	exibeMov(“esqfre”);
	gira esquerda horário();
	exibeMov(“esqfre”);
}
se (face da esquerda tem duas quinas)
{
	gira trás horário();
	exibeMov(“trasesq”);
	gira direita anti-horário();
	exibeMov(“dirfre”);
	gira trás horário();
	exibeMov(“trasesq”);
	gira esquerda horário();
	exibeMov(“esqfre”);
	gira esquerda horário();
	exibeMov(“esqfre”);
	gira trás anti-horário();
	exibeMov(“trasdir”);
	gira direita horário();
	exibeMov(“dirtra”);
	gira trás horário();
	exibeMov(“trasesq”);
	gira esquerda horário();
	exibeMov(“esqfre”);
	gira esquerda horário();
	exibeMov(“esqfre”);
	gira trás horário();
```

```
        exibeMov(“trasesq”);
        gira trás horário();
        exibeMov(“trasesq”);
}
se (face da direita tem duas quinas)
{
        gira frente horário();
        exibeMov(“fredir”);
        gira esquerda anti-horário();
        exibeMov(“esqtra”);
        gira frente horário();
        exibeMov(“fredir”);
        gira direita horário();
        exibeMov(“dirtra”);
        gira direita horário();
        exibeMov(“dirtra”);
        gira frente anti-horário();
        exibeMov(“freesq”);
        gira esquerda horário();
        exibeMov(“esqfre”);
        gira frente horário();
        exibeMov(“fredir”);
        gira direita horário();
        exibeMov(“dirtra”);
        gira direita horário();
        exibeMov(“dirtra”);
        gira frente horário();
        exibeMov(“fredir”);
        gira frente horário();
        exibeMov(“fredir”);
}
se (face da trás tem duas quinas)
{
        gira esquerda horário();
        exibeMov(“esqfre”);
        gira frente anti-horário();
        exibeMov(“freesq”);
        gira esquerda horário();
        exibeMov(“esqfre”);
        gira trás horário();
        exibeMov(“trasesq”);
        gira trás horário();
        exibeMov(“trasesq”);
        gira esquerda anti-horário();
        exibeMov(“esqtra”);
        gira frente horário();
        exibeMov(“fredir”);
        gira esquerda horário();
        exibeMov(“esqfre”);
        gira trás horário();
        exibeMov(“trasesq”);
        gira trás horário();
        exibeMov(“trasesq”);
        gira esquerda horário();
        exibeMov(“esqfre”);
        gira esquerda horário();
        exibeMov(“esqfre”);
}
se(quinas não estiverem na posição) /*Checa a necessidade de efetuar
```

```
                                    * novamente a função */
        {
                resolveQuinas();
        }

}
********** FIM DA FUNÇÃO RESOLVE QUINAS*********

********** INÍCIO FUNÇÃO RESOLVE BORDAS *********

resolveBordas();
{

        procura primeira borda();
        procura segunda borda();
        procura terceira borda();
        procura quarta borda();
        se(bordas estiverem na posição) //Checa se peças ja estão na posição
        {
                saia da função
        }
        /*Nessa função tome como referência a face de trás sendo aquela que já
está completo */
        se(borda estão trocadas no sentido horario)
        {
                permutação horária();
        }
        se(borda estão trocadas no sentido anti-horario)
        {
                permutação anti-horária();
        }
        se(bordas não estiverem na posição) /*Checa a necessidade de efetuar
                                    * novamente a função */
        {
                resolveBordas();
        }

}
********** FIM DA FUNÇÃO RESOLVE BORDAS *********

***************Funções predefinidas do cubo ***************
->permutação horária();: F2 U L R' F2 L' R U F2

->permutação anti-horária();:  F2 U' L R' F2 L' R U' F2

->algaritmo sune();:  R U R' U R U2 R'

/*importante ressaltar a necessidade de referenciar os movimentos para os
algoritmos funcionarem */

*********************************************************
```