

# TRABALHO 1 – Programação Modular

INF1301

Planejamento da segunda camada do cubo mágico

Data de entrega: 28/03/2018

Turma: 3WB



Pilar Fernandez – 1521720

Mariana Salgueiro – 1510988

Bruno Teixeira – 1520066

Gabriel Faria – 1512424

\*\*\*\*\*Funções direita e esquerda das cores\*\*\*\*\*

Vermelha: Esquerda-> U' B' U B U L U' L'

Direita -> U F U' F' U' L' U L

Verde: Esquerda-> U R U' R' U' F' U F;

Direita -> U' L' U L U F U' F';

Laranja : Esquerda-> U' F' U F U R U' R'

Direita -> U B U' B' U' R' U R

Azul: Esquerda-> U' R' U R U B U' B'

Direita -> U L U' L' U' B' U B

\*\*\*\*\*

Pseudo código para futura implementação:

```
//Utilizando a face fornecida pelo programa.
//Verificar a cor da peça do meio.
/***** Considerando a face branca embaixo *****/
/*****INÍCIO DA FACE VERMELHA*****/

if (peça do centro == vermelha){
//Sem girar o cubo, apenas considerar que estamos tratando do lado vermelho
procurar a peça vermelha e azul;
/*****LOCALIZAÇÃO DA PEÇA*****/
if (peça não estiver em cima (3ª camada)){//peça se encontra, então, na 2ª
camada
    if (peça estiver alinhada com a sua cor) //checando se a peça já se
encontra na posição correta
        não faz nada;
    elseif (peça estiver em outro lugar){
        utilizando a face que ela estiver:
        if (a peça estiver à direita)
            aplicar o algoritmo para a DIREITA(em relação a face dela)

        else
            aplicar o algoritmo para a ESQUERDA(em relação a face dela)

    }
}
//com isso, a peça terá ido para cima (3ª camada) e então precisamos encontrá-
la para fazer o alinhamento

elseif (peça estiver em cima a direita)
    girar parte superior para a esquerda;
elseif (peça estiver em cima a esquerda)
    girar parte superior para a direita;
elseif (peça estiver em cima atrás)
    girar parte superior duas vezes para a direita;

/*****/
```

```

//peça agora alinhada com o seu centro
if (peça da face vermelha superior == peça do centro) // F[0][1] == F[1][1]
    aplicar algoritmo para a ESQUERDA U' B' U B U L U' L'; //pois U[2][1] ==
L[1][1]

procurar a peça vermelha e verde;
/*****LOCALIZAÇÃO DA PEÇA*****/
if (peça não estiver em cima (3ª camada)){//peça se encontra, então, na 2ª
camada
    if (peça estiver alinhada com a sua cor) //checando se a peça já se
encontra na posição correta
        não faz nada;
    elseif (peça estiver em outro lugar){
        utilizando a face que ela estiver:
        if (a peça estiver à direita)
            aplicar o algoritmo para a DIREITA(em relação a face dela)
        else
            aplicar o algoritmo para a ESQUERDA(em relação a face dela)
    }
}

//com isso, a peça terá ido para cima (3ª camada) e então precisamos encontrá-
la para fazer o alinhamento

elseif (peça estiver em cima a direita)
    girar parte superior para a esquerda;
elseif (peça estiver em cima a esquerda)
    girar parte superior para a direita;
elseif (peça estiver em cima atrás)
    girar parte superior duas vezes para a direita;

/*****/

//peça agora alinhada com o seu centro
if (peça da face vermelha superior == peça do centro) // F[0][1] == F[1][1]
    aplicar algoritmo para a DIREITA(Vermelho) U F U' F' U' L' U L;; //pois
U[2][1] == R[1][1]
}

/*****FIM DA FACE VERMELHA*****/

/*****INÍCIO DA FACE VERDE*****/

if (peça do centro == verde){
    procurar a peça verde e vermelha;
    /*****LOCALIZAÇÃO DA PEÇA*****/
    if (peça não estiver em cima (3ª camada)){//peça se encontra, então, na 2ª
camada
        if (peça estiver alinhada com a sua cor) //checando se a peça já se
encontra na posição correta
            não faz nada;
        elseif (peça estiver em outro lugar){
            utilizando a face que ela estiver:
            if (a peça estiver à direita)
                aplicar o algoritmo para a DIREITA(em relação a face dela)
            else
                aplicar o algoritmo para a ESQUERDA(em relação a face dela)
        }
    }
}

```

```

//com isso, a peça terá ido para cima (3ª camada) e então precisamos encontrá-
la para fazer o alinhamento

elseif (peça estiver em cima a direita)
    girar parte superior para a esquerda;
elseif (peça estiver em cima a esquerda)
    girar parte superior para a direita;
elseif (peça estiver em cima atrás)
    girar parte superior duas vezes para a direita;

/*****/

//peça agora alinhada com o seu centro
if (peça da face verde superior == peça do centro) // F[0][1] == F[1][1]
    aplicar algoritmo para a ESQUERDA U' L' U L U F U' F'; //pois U[2][1] ==
L[1][1]

    procurar a peça verde e laranja;
    /*****LOCALIZAÇÃO DA PEÇA*****/
    if (peça não estiver em cima (3ª camada)){//peça se encontra, então, na 2ª
camada
        if (peça estiver alinhada com a sua cor) //checando se a peça já se
encontra na posição correta
            não faz nada;
        elseif (peça estiver em outro lugar){
            utilizando a face que ela estiver:
            if (a peça estiver à direita)
                aplicar o algoritmo para a DIREITA(em relação a face dela);
            else
                aplicar o algoritmo para a ESQUERDA(em relação a face dela);
        }
    }

//com isso, a peça terá ido para cima (3ª camada) e então precisamos encontrá-
la para fazer o alinhamento

elseif (peça estiver em cima a direita)
    girar parte superior para a esquerda;
elseif (peça estiver em cima a esquerda)
    girar parte superior para a direita;
elseif (peça estiver em cima atrás)
    girar parte superior duas vezes para a direita;

/*****/

//peça agora alinhada com o seu centro
if (peça da face verde superior == peça do centro) // F[0][1] == F[1][1]
    aplicar algoritmo para a DIREITA U R U' R' U' F' U F; // F[0][1] ==
R[1][1]

}
/*****FIM DA FACE VERDE*****/

/*****INÍCIO DA FACE LARANJA*****/

if (peça do centro == laranja){
    procurar a peça laranja e verde;
    /*****LOCALIZAÇÃO DA PEÇA*****/
    if (peça não estiver em cima (3ª camada)){//peça se encontra, então, na 2ª
camada

```

```

        if (peça estiver alinhada com a sua cor) //checando se a peça já se
encontra na posição correta
            não faz nada;
        elseif (peça estiver em outro lugar){
            utilizando a face que ela estiver:
            if (a peça estiver à direita)
                aplicar o algoritmo para a DIREITA(em relação a face dela)
            else
                aplicar o algoritmo para a ESQUERDA(em relação a face dela)
        }
    }
    //com isso, a peça terá ido para cima (3ª camada) e então precisamos encontrá-
    la para fazer o alinhamento

    elseif (peça estiver em cima a direita)
        girar parte superior para a esquerda;
    elseif (peça estiver em cima a esquerda)
        girar parte superior para a direita;
    elseif (peça estiver em cima atrás)
        girar parte superior duas vezes para a direita;

    /*****/

    //peça agora alinhada com o seu centro
    if (peça da face laranja superior == peça do centro) // F[0][1] == F[1][1]
        aplicar algoritmo para a ESQUERDA U' F' U F U R U' R'; //pois U[2][1] ==
L[1][1]

    procurar a peça laranja e azul;
    /*****LOCALIZAÇÃO DA PEÇA*****/
    if (peça não estiver em cima (3ª camada)){//peça se encontra, então, na 2ª
camada
        if (peça estiver alinhada com a sua cor) //checando se a peça já se
encontra na posição correta
            não faz nada;
        elseif (peça estiver em outro lugar){
            utilizando a face que ela estiver:
            if (a peça estiver à direita)
                aplicar o algoritmo para a DIREITA(em relação a face dela);
            else
                aplicar o algoritmo para a ESQUERDA(em relação a face dela);
        }
    }

    //com isso, a peça terá ido para cima (3ª camada) e então precisamos encontrá-
    la para fazer o alinhamento

    elseif (peça estiver em cima a direita)
        girar parte superior para a esquerda;
    elseif (peça estiver em cima a esquerda)
        girar parte superior para a direita;
    elseif (peça estiver em cima atrás)
        girar parte superior duas vezes para a direita;

    /*****/

    //peça agora alinhada com o seu centro
    if (peça da face laranja superior == peça do centro) // F[0][1] == F[1][1]
        aplicar algoritmo para a DIREITA U B U' B' U' R' U R; // F[0][1] ==
R[1][1]

```

```

}

/*****FIM DA FACE LARANJA*****/

/*****INÍCIO DA FACE AZUL*****/

if (peça do centro == azul){
    procurar a peça azul e laranja;
    /*****LOCALIZAÇÃO DA PEÇA*****/
    if (peça não estiver em cima (3ª camada)){//peça se encontra, então, na 2ª
camada
        if (peça estiver alinhada com a sua cor) //checando se a peça já se
encontra na posição correta
            não faz nada;
        elseif (peça estiver em outro lugar){
            utilizando a face que ela estiver:
            if (a peça estiver à direita)
                aplicar o algoritmo para a DIREITA(em relação a face dela)
            else
                aplicar o algoritmo para a ESQUERDA(em relação a face dela)
        }
    }
    //com isso, a peça terá ido para cima (3ª camada) e então precisamos encontrá-
la para fazer o alinhamento

    elseif (peça estiver em cima a direita)
        girar parte superior para a esquerda;
    elseif (peça estiver em cima a esquerda)
        girar parte superior para a direita;
    elseif (peça estiver em cima atrás)
        girar parte superior duas vezes para a direita;

    /*****/

    //peça agora alinhada com o seu centro
    if (peça da face azul superior == peça do centro) // F[0][1] == F[1][1]
        aplicar algoritmo para a ESQUERDA U' R' U R U B U' B'; //pois U[2][1] ==
L[1][1]

    procurar a peça azul e vermelha;
    /*****LOCALIZAÇÃO DA PEÇA*****/
    if (peça não estiver em cima (3ª camada)){//peça se encontra, então, na 2ª
camada
        if (peça estiver alinhada com a sua cor) //checando se a peça já se
encontra na posição correta
            não faz nada;
        elseif (peça estiver em outro lugar){
            utilizando a face que ela estiver:
            if (a peça estiver à direita)
                aplicar o algoritmo para a DIREITA(em relação a face dela)
            else
                aplicar o algoritmo para a ESQUERDA(em relação a face dela)
        }
    }

    //com isso, a peça terá ido para cima (3ª camada) e então precisamos encontrá-
la para fazer o alinhamento

    elseif (peça estiver em cima a direita)
        girar parte superior para a esquerda;
    elseif (peça estiver em cima a esquerda)

```

```

        girar parte superior para a direita;
elseif (peça estiver em cima atrás)
    girar parte superior duas vezes para a direita;

/*****/

//peça agora alinhada com o seu centro
if (peça da face azul superior == peça do centro) // F[0][1] == F[1][1]
    aplicar algoritmo para a DIREITA U L U' L' U' B' U B; // F[0][1] ==
R[1][1]
}

/*****FIM DA FACE AZUL*****/

```