

Trabalho de Modular 2018.1

Terminologia:

Nota:

Todas as rotações são feitas com o cubo fixo, mantendo a perspectiva de estar olhando para a face frontal. Ao mudar a face frontal, as novas instruções já estarão de acordo.

"Face [cor X]" se refere à face do cubo que possui como peça central um painel de cor X.

X vs X':

X representa uma rotação de 90° no sentido horário.

X' representa uma rotação de 90° no sentido anti-horário.

Movimentos:

U & U' : Gira a face de cima.

E & E' : Gira a linha horizontal do meio.

D & D' : Gira a face de baixo.

R & R' : Gira a face direita.

L & L' : Gira a face esquerda.

F & F' : Gira a face frontal.

B & B' : Gira a face traseira.

M & M' : Gira a linha vertical do meio.

S & S' : Gira a face do meio, a que possui 3 peças de cada uma das faces: cima, direita, de baixo e esquerda.

Cubo:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

| D7 D8 D9 |
+-----+

Peças:

1. Peças de Canto: Existem 8 peças de canto no Cube. Elas possuem 3 painés de cores que podem estar em quaisquer ordens. São elas:
 1. U1/B3/L1
 2. U3/B1/R3
 3. L7/B9/D7
 4. L9/F7/D1
 5. F1/L3/U7
 6. F3/U9/R1
 7. F9/R7/D3
 8. R9/D9/B7
 2. Peças de Borda: Existem 12 peças de borda. Elas possuem 2 painés de cores que podem estar em quaisquer ordens.
 1. U2/B2
 2. U4/L2
 3. U6/R2
 4. U8/F2
 5. F4/L6
 6. F6/R4
 7. F8/D2
 8. B4/R6
 9. B6/L4
 10. B8/D8
 11. D6/R8
 12. D4/L8
-

Algoritmo:

Objetivo:

- Ter a face superior completa.
- Ter a primeira linha de todas as faces completa.

Descrição:

1. Comece com a face branca para cima e a verde como frontal. Chame a cor verde como x .
2. Verifique se a cruz está formada, ou seja, se U_2 U_4 U_6 e U_8 são brancos. *Caso positivo, pule para o passo 15.*
3. Busque a peça de borda da cor X/Branca. Chame a peça de y .
4. Mova y para a face de baixo do cubo. ***Se não for possível fazê-lo sem modificar a cruz que está se formando na face de cima, ou seja, se a peça residir na mesma coluna de uma face -já iterada-, pule para o passo 7***
5. Rotacione a face de baixo de forma que a peça y esteja na face correspondente a sua cor x . (Ou seja, na face frontal, face direita, traseira e esquerda, na primeira, segunda, terceira e quarta iteração respectivamente.)

6. Verifique:
 1. Se o painel da face de baixo de Y for X:
 1. Se for a primeira iteração execute: D' F' R F . *Siga para o passo 6.2*
 2. Se for a segunda iteração execute: D' R' B' R *Siga para o passo 6.2*
 3. Se for a terceira iteração execute: D' B L B' *Siga para o passo 6.2*
 4. Se for a quarta iteração execute: D' L' F L *Siga para o passo 6.2*
 2. Se o painel da face de baixo de Y for da cor branca:
 1. Se for a primeira iteração execute: F F *Siga para o passo 9*
 2. Se for a segunda iteração execute: R R *Siga para o passo 9*
 3. Se for a terceira iteração execute: B B *Siga para o passo 9*
 4. Se for a quarta iteração execute: L L *Siga para o passo 8*
 3. Verifique:
 1. Se sua peça for F4/L6 execute: L D' L' . *Pule para o passo 5.*
 2. Se sua peça for F6/R4 execute: R' D R. *Pule para o passo 5.*
 3. Se sua peça for B6/L4 execute: L' D' L. *Pule para o passo 5.*
 4. Se sua peça for B4/R6 execute: R' D R. *Pule para o passo 5.*
7. Verifique se U2 U4 U6 e U8 são brancos. Caso positivo, prossiga para o *passo 10*.
8. Chame x como a cor central da próxima face, na ordem: frontal -> direita -> traseira -> esquerda. *Pule para o passo 3*
9. Buscar a peça da cor verde, branca e cor vermelha:
 1. Caso a peça esteja na face inferior: 1. Caso B9/D7/L7: Execute D D. *Prossiga com o passo 11* 2. Caso F7/D1/L9: Execute D' . *Prossiga com o passo 11* 3. Caso F9/D3/R7: *Prossiga com o passo 11* 4. Caso B7/D9/R9: Execute D . *Prossiga com o passo 11*
 2. Caso a peça esteja na face superior: 1. Caso B3/U1/L1: Execute L' D' L . *Prossiga com o passo 11* 1. Caso B1/U3/R3: Execute R D R' . *Prossiga com o passo 11* 1. Caso F1/U7/L3: Execute L D' L' . *Prossiga com o passo 11* 1. Caso F3/U9/R1: *Prossiga com o passo 11*
10. Repita R' D R D' até a peça atingir as posições U9 (branca), F3 (Verde) e R1 (Vermelha).
11. Buscar a peça da cor azul, branca e cor vermelha:
 1. Caso a peça esteja na face inferior: 1. Caso B9/D7/L7: Execute D. *Siga para o passo 13* 1. Caso F7/D1/L9: Execute D D. *Siga para o passo 13* 1. Caso F9/D3/R7 : Execute D' . *Siga para o passo 13* 1. Caso B7/D9/R9: *Siga para o passo 13*
 2. Caso a peça esteja na face superior: 1. Caso B3/U1/L1: Execute L' D L . *Siga para o passo 13* 1. Caso B1/U3/R3: *Siga para o passo 13* 1. Caso F1/U7/L3: Execute L D D L' . *Siga para o passo 13*
12. Repita R' D' R D até a peça atingir as posições U3 (branca), B3 (Azul) e R3 (Vermelha).
13. Buscar a peça da cor azul, branca e cor laranja:
 1. Caso a peça esteja na face inferior: 1. Caso B9/D7/L7: *Siga para o passo 15* 1. Caso F7/D1/L9: Execute D. *Siga para o passo 15* 1. Caso F9/D3/R7 : Execute D D . *Siga para o passo 15* 1. Caso B7/D9/R9: Execute D' . *Siga para o passo 15*
 2. Caso a peça esteja na face superior: 1. Caso B3/U1/L1:* *Siga para o passo 15** 1. Caso F1/U7/L3: Execute L D D L' D . *Siga para o passo 15*
14. Repita L' D L D' até a peça atingir as posições U1 (branca), B3 (Azul) e L1 (Laranja).
15. Buscar a peça da cor verde, branca e cor laranja:
 1. Caso a peça esteja na face inferior: 1. Caso B9/D7/L7: Execute D' *Siga para o passo 17* 1. Caso F7/D1/L9: *Siga para o passo 17* 1. Caso F9/D3/R7: Execute D . *Siga para o passo 17* 1. Caso B7/D9/R9: Execute D D. *Siga para o passo 17*
 2. Caso a peça esteja na face superior: 1. Caso F1/U7/L3: *Siga para o passo 17*
 1. Repita L D' L D até a peça atingir as posições U7 (branca), L3 (Laranja) e F1 (Verde).

O pseudocódigo é apenas uma (pseudo) implementação da formalização acima. Em caso (improvável) de divergências, a descrição funciona como único objeto de verdade.

```
cubo.UpFace = branca;
cubo.FrontFace = verde;

seja X = verde;
seja i = 1;                                // iterações

enquanto (U2 != Branco || U4 != Branco || U6 != Branco || U8 != Branco)
{
    se (i == 1)
        seja Y = cubo.acharPeca(Verde, Branca);
        seja faceX = frontal
    senão se (i == 2)
        seja Y = cubo.acharPeca(Vermelho, Branca);
        seja faceX = direita
    senão se (i == 3)
        seja Y = cubo.acharPeca(Azul, Branca)
        seja faceX = traseira
    senão se (i == 4)
        seja Y = cubo.acharPeca(Laranja, Branca)
        seja faceX = esquerda

    seja res = colocarPecaFaceDeBaixo(Y, i);    // retorna false se a peça
    modificaria a cruz branca já construída

    se (res == false)
    {
        se (Y == F4/L6)
            cubo.executar(L D' L);
        se (Y == F6/R4)
            cubo.executar(R' D R);
        se (Y == B6/L4)
            cubo.executar(L' D' L);
        se (Y == B4/R6)
            cubo.executar(R' D R);
    }

    enquanto Y.face != faceX
        cubo.executar(D)

    se Y.corFaceDeBaixo == X
    {
        se (i == 1)
            cubo.executar(D' F' R F);
        senão se (i == 2)
            cubo.executar(D' R' B' R);
        senão se (i == 3)
            cubo.executar(D' B L B');
        senão se (i == 4)
            cubo.executar(D' L' F L);
    }

    senão se Y.corfaceDebaixo == Branca
```

```

{
    se (i == 1)
        cubo.executar(F F);
    senão se (i == 2)
        cubo.executar(R R);
    senão se (i == 3)
        cubo.executar(B B);
    senão se (i == 4)
        cubo.executar(L L);
}

se (i == 1)
    X = Vermelho;
senão se (i == 2)
    X = Azul;
senão se (i == 3)
    X = Laranja;
}

seja cantoUm = cubo.acharPeca(Verde, Branca, Vermelha);

se (cantoUm == B9/D7/L7)
    cubo.executar(D D);
se (cantoUm == F7/D1/L9)
    cubo.executar(D');
se (cantoUm == F9/D3/R7)
    // do nothing
se (cantoUm == B7/D9/R9)
    cubo.executar(D);
se (cantoUm == B3/U1/L1)
    cubo.executar(L' D' L);
se (cantoUm == B1/U3/R3)
    cubo.executar(R D R');
se (cantoUm == F1/U7/L3)
    cubo.executar(L D' L);
se (cantoUm == F3/U9/R1)
    // do nothing

enquanto ( U9 != Branco || F3 != Verde || R1 != Vermelho)
    cubo.executar(R' D R D);

seja cantoDois = cubo.acharPeca(Azul, Branca, Vermelha);

se (cantoDois == B9/D7/L7)
    cubo.executar(D');
se (cantoDois == F7/D1/L9)
    cubo.executar(D D);
se (cantoDois == F9/D3/R7)
    cubo.executar(D');
se (cantoDois == B7/D9/R9)
    // do nothing
se (cantoDois == B3/U1/L1)
    cubo.executar(L' D L);
se (cantoDois == B1/U3/R3)
    // do nothing
se (cantoDois == F1/U7/L3)
    cubo.executar(L D D L');

```

```
enquanto ( U3 != Branco || B3 != Azul || R3 != Vermelho)
    cubo.executar(R' D' R D);

seja cantoTres = cubo.acharPeca(Azul, Branca, Laranja);

se (cantoTres == B9/D7/L7)
    // do nothing
se (cantoTres == F7/D1/L9)
    cubo.executar(D');
se (cantoTres == F9/D3/R7)
    cubo.executar(D D);
se (cantoTres == B7/D9/R9)
    cubo.executar(D');
se (cantoTres == B3/U1/L1)
    // do nothing
se (cantoTres == F1/U7/L3)
    cubo.executar(L D D L' D);

enquanto ( U1 != Branco || B3 != Azul || L1 != Laranja)
    cubo.executar(L' D L D');

seja cantoQuatro = cubo.acharPeca(Verde, Branca, Vermelha);

se (cantoQuatro == B9/D7/L7)
    cubo.executar(D');
se (cantoQuatro == F7/D1/L9)
    // do nothing
se (cantoQuatro == F9/D3/R7)
    cubo.executar(D);
se (cantoQuatro == B7/D9/R9)
    cubo.executar(D D);
se (cantoQuatro == F1/U7/L3)
    // do nothing

enquanto ( U7 != Branco || L3 != Laranja || F1 != Verde)
    cubo.executar(L D' L D);

fim
```