Tuttminx solution

Overview
Preliminary

- Choose top & bottom pentagons (⬟s)
- *NB.* Hexagon (⬢) sides only allow 120° moves, thus hexagons should never have pentagon pieces (checkered stickers) within them. If not, tuttminx can lock up and/or get damaged.
- Use pentagons to reorient corners (3 possible orientations). 5 ⬟s in lower layer, 5 in upper.
- Use hexagons to transport

Notation
With hexagon face as F and pentagon face as U:
R and L are hexagon faces to the right and left of F
BR is the hexagon behind R
BL is the hexagon behind L

Algorithms
Cycle 3 corners between 2 pentagons: **F2' MF2' F2 U F2' MF2 F2 U'**

Cycle 3 edges on U pentagon with two good edges on F and L:
R-BL-BR rotate CW: **(R2 U2) (R2' U') (R2 U' R2')**
R-BR-BL rotate CCW: **(R2 U) (R2' U) (R2 U2' R2')**
Note: the above wreck one corner of the R face, see the video for fix.

Cycle 3 edges on U pentagon with two good edges on F and BR:
R-L-BL rotate CW : **(R2 U2') (R2' U') (R2 U2' R2')**
R-B-L rotate CCW: **(R2 U2) (R2' U) (R2 U2 R2')**

Steps

1. **Bottom cross & first 5 ⬢s**.

   Complete bottom pentagon by adding corner and adjacent edge pieces
   A. NB. Need to create corner & edge blocks, before moving them into place. Can not use megaminx algorithms to move edge pieces.
   B. Identify nearby working face (hexagon) to build blocks. Closer to top may be easiest/safest
2. **Solve bottom layers**
   A. *Bottom layers: bottom edges:*
      i. Insert bottom edges for all ⬢
      ii. Insert one bottom edge for a ⬟
      iii. Build and insert other bottom edge & corner pieces for ⬟ (*i.e.* 3–piece block)
      iv. Fill in middle edge & corner blocks (vertical) for ⬟s, OR:
      • continue building bottom ⬟ 3–piece blocks

B. *Bottom layers: upper edges*

- Fill in top half of ⬟s & ⬣s

NB. Do not solve bottom row in upper layer ⬟s or anything on or above that level – different strategy there

3. Solve upper layers

A. Fill in bottom edge piece for all 5 upper layer ⬟s

B. Fill in 2-piece vertical blocks (lower section of ⬟, both left & right) for 5 ⬟s

C. Fill in all ⬣ to ⬣ edges

i. Bottom edges (vertical: from uppermost ⬣ layer to layer below)

1. To flip an edge: orbit it around the tuttminx via the other 4 ⬣s

   Then, correct the 4 displaced ⬣ bottom edge pieces (in reverse order)

ii. Side edges in uppermost ⬣ layer. Complete 1 ⬣ first, then continue..

1. Use same strategy (as above) to flip the orientation of pieces (if necessary)
2. Rotate ⬣ appropriately to receive incoming edge pieces
3. Rotate ⬣ appropriately to fix other issues...
4. If side pieces of final ⬣ are flipped, use above flipping strategy
5. If end up with 3 mis-oriented pieces, use a 3-cycle (27 min)

D. Fill in ⬣ to ⬟ edges

i. Strategy: use top layer to hold target edges. Place on opposite side of target ⬣

E. Fill in remaining ⬟ corners

i. Use top ⬟ layer to stage corners

ii. Use algorithm to 3 cycle of corners from 1 ⬟ to another ⬟

- **F2' MF2' F2 U F2' MF2 F2 U'**
- Do set up move to move target corner into algo position (⬟ bottom left)

4. Solve top/final pentagon

A. Solve edges: use megaminx algorithm to cycle edges

~~~~~~
~~~~~~

Step 1, Bottom cross & first 5 ⬢s.

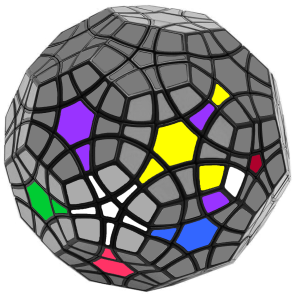
Complete bottom pentagon by adding corner and adjacent edge pieces

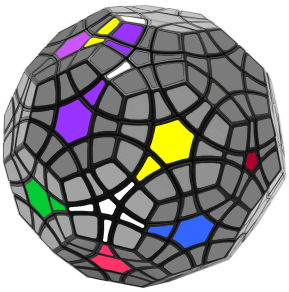
Method 1

CW 72° to join edge & corner →

CW 120° to build 3-piece block

CW 120° to put block in place

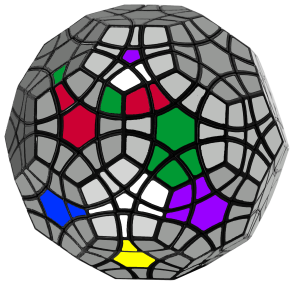


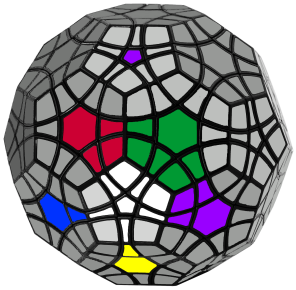


Method 2

CW 72° to join block with red-green edge

CW 120° to put 3-piece block in place →





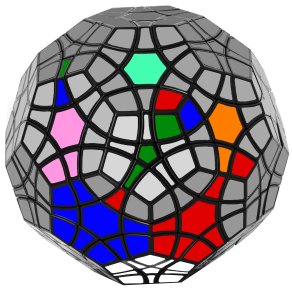
Step 2, Solve bottom layers

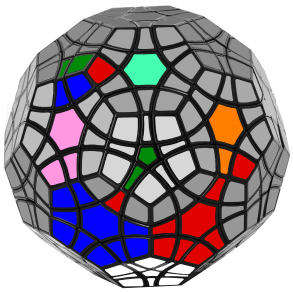
2.a.iii, Build ⬟ bottom 3-piece block

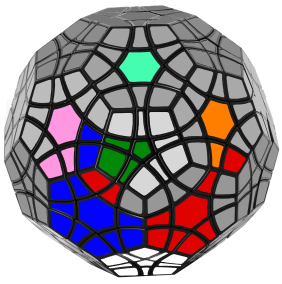
CW 120°
build 2-piece block →

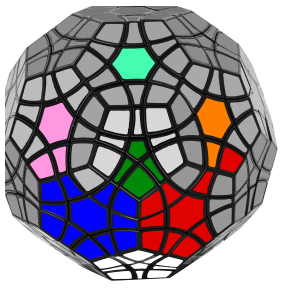
CW 120°
build 3-piece block →

CCW 72°
put block in place →



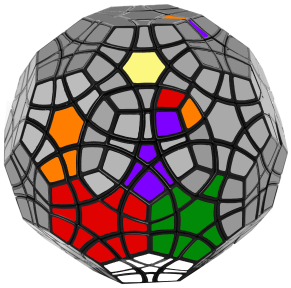




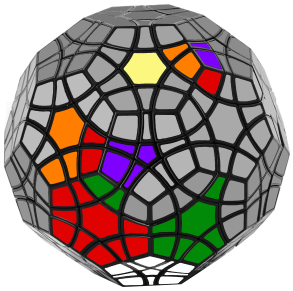


2.a.iv, Build ⬟ vertical 2-piece block (left side)

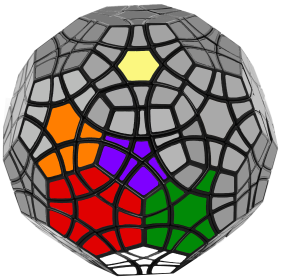
⬢ **CCW** 120°
build 2-piece block →



⬟ **CW** 72° to set up ⬟
⬢ **CW** 120° to join block with ⬟ →

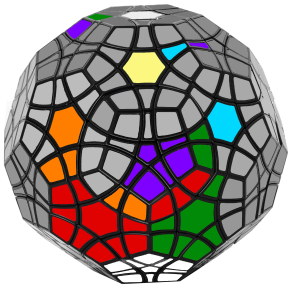


⬟ **CCW** 72° to complete

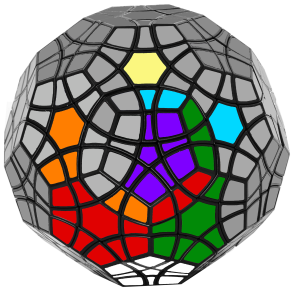


2.a.iv, Build ⬟ vertical 2-piece block (right side)

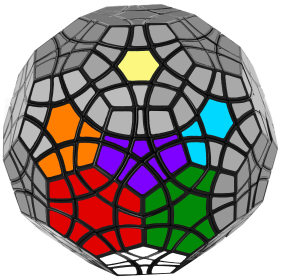
⬢ (top face) **CCW** 120°
build 2-piece block →



⬟ **CCW** 72° to set up ⬟
⬢ **CW** 120° to join block with ⬟ →



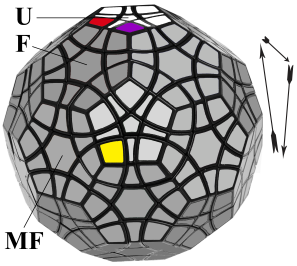
⬟ **CW** 72° to complete



Step3, Solve upper layers

Fill in remaining ⬟ corners:

- Use top ⬟ layer to stage corners
- Do set up move to target bottom left corner of target ⬟
- Use 3-cycle algorithm to cycle corners from 1 ⬟ to another ⬟
  F2' MF2' F2 U F2' MF2 F2 U'



Step 4, Solve top/final pentagon

A. *Solve edges:* use megaminx algorithm to cycle edges
- Identify correct edges – see algorithms above.
  NB. 2020.06.03 solve, had only *one* correct edge. Used:
  R-L-BL rotate CW : **(R2 U2') (R2' U') (R2 U2' R2')** which yielded two correct edges,
  Then used appropriate algorithm from above..
- If 3-cycle algorithm displaces a corner piece, use **F2' MF2' F2 U F2' MF2 F2 U'** to correct it immediately afterwards.

B. *Permute corners*: use 3-cycle algorithm: **F2' MF2' F2 U F2' MF2 F2 U'**
NB. May need to do 3-cycle twice to get a particular corner in its correct position..

Sources

1. Pete the geek
   Part 1: youtube.com/watch?v=rqX1Lf3ZH1Y
   Part 2: youtube.com/watch?v=Ug-9dPB5w5g
   Part 3: youtube.com/watch?v=MuB70ER4_Mo
2. Falcon
   youtube.com/watch?v=mbqDqudFsSg

Video notes

Pete
Part 1
15 min, method for joining corner & edge pieces

Part 2
18-19 min, use pentagons to reorient corners (3 possible orientations).

Part 3

16-21 min, fill in all ⬢ to ⬢ edges

27 min, If end up with 3 mis-oriented pieces, use a 3-cycle (27 min)
34-38 min, 3-cycle algorithm

50 min, 3-cycle algorithm for edges
53 min, permute corners

Falcon
Part 1
Stopped at 11 min

Unicode symbols
https://www.unicode.org/charts/nameslist/n_2B00.html
https://unicode.org/charts/PDF/U2B00.pdf