F2L Algorithms – All Four Slot Angles

Developed by Feliks Zemdegs
and Andy Klise

Images sourced from Conrad Rider's VisualCube - http://cube.crider.co.uk/visualcube.php

Algorithm Presentation Format

Suggested algorithm here
Alternative algorithms here

Set up F2L pair / / Solve F2L pair

It is not recommended to learn any of these algorithms before learning intuitive F2L.

The black part of each algorithm sets up the pieces to a basic insertion case, which is then written in blue.

Basic Inserts

U (R U' R')

y' U' (R' U R)
y U' (L' U L)

y' (R' U' R)
y (L' U' L)

(R U R')

F2L Case 1

U' (R U' R' U) y' (R' U' R)
y' U (R' U' R U') (R' U' R)

U' (R U R' U) (R U R')

U' (R U2' R' U) y' (R' U' R)
U' (R U2' R') d (R' U' R)

R' U2' R2 U R2' U R
y' U (R' U2 R) U' y (R U R')
(R U' R' U) (R U' R') U2 (R U' R')

y' U (R' U R U') (R' U' R)

U' (R U' R' U) (R U R')

F2L Case 2

(U' R U R') U2 (R U' R')

y' (U R' U' R) U2' (R' U R)
d (R' U' R) U2' (R' U R)
Note – (y' U) and (d) are interchangeable

U' (R U2' R') U2 (R U' R')

y' U (R' U2 R) U2' (R' U R)
d (R' U2 R) U2' (R' U R)

F2L Case 3

U (R U2' R') U (R U' R')

y' U' (R' U2 R) U' (R' U R)

U2 (R U R' U) (R U' R')
(R U' R') U2 (R U R')

y' U2 (R' U' R) U' (R' U R)
F' L' U2 L F
Note – The second algorithm is fewer moves, but less intuitive and less finger-friendly.

Incorrectly Connected Pieces

y' (R' U R) U2' y (R U R')
(R U R') U2 (R U' R' U) (R U' R')

(R U' R' U2) y' (R' U' R)
U F (R U R' U') F' (U R U' R')

(R U2 R') U' (R U R')

y' (R' U2 R) U (R' U' R)

U (R U' R' U') (R U' R' U) (R U' R')
(R U R' U2') (R U R' U') (R U R')

y' U' (R' U R U) (R' U R U') (R' U R)
F (U R U' R') F' (R U' R')

Corner in Place, Edge in U Face

U' F' (R U R' U') R' F R
R' F' R U (R U' R') F

U (R U' R') U' (F' U F)
U (R U' R') (F R' F' R)

(R U' R' U) (R U' R')

y' (R' U R U') (R' U R)

y' (R' U' R U) (R' U' R)
(R' F R F') U (R U' R')

(R U R' U') (R U R')

Edge in Place, Corner in U face

(R U' R' U) y' (R' U R)
U' (R' F R F') (R U' R')

(U R U' R') (U R U' R') (U R U' R')

(U' R U' R') U2 (R U' R')

U (R U R') U2 (R U R')

(U' R U R') d (R' U' R)

U (F' U' F) U' (R U R')

Edge and Corner in Place

Solved Pair

(R U' R') d (R' U2 R) U2' (R' U R)

(R U' R' U') R U R' U2 (R U' R')
(R U R' U') R U2 R' U' (R U R')

(R U' R' U) (R U2' R') U (R U' R')
(R U R') U2' (R U' R' U) (R U R')

(F' U F) U2 (R U R' U) (R U' R')
(R U' R') F (R U R' U') F' (R U' R')

(R U R' U') (R U' R') U2 y' (R' U' R)

F2L Algorithms – All Four Slot Angles

Developed by Feliks Zemdegs
and Andy Klise

Images sourced from Conrad Rider's VisualCube - http://cube.crider.co.uk/visualcube.php

Algorithms for slot in back-right position.

Basic Inserts

y U (R U' R')
y' U (L U' L')

(R' U' R)

U' (R' U R)

y (R U R')
y' (L U L')

F2L Case 1

U (R' U' R U') (R' U' R)

U (R' U R U') y (R U R')

R U2' R2' U' (R2 U' R')

U (R' U2 R) U' y (R U R')

U (R' U R U') (R' U' R)

y U' (R U' R' U) (R U R')

F2L Case 2

y (U' R U R') U2 (R U' R')
U r' (U R U' R') U' r

(U R' U' R) U2' (R' U R)

y U' (R U2' R') U2 (R U' R')

U (R' U2 R) U2' (R' U R)

F2L Case 3

y U (R U2' R') U (R U' R')

U' (R' U2 R) U' (R' U R)

y U2 (R U R' U) (R U' R')

U2 (R' U' R) U' (R' U R)
R' F' U2 F R
Note – The second algorithm is fewer moves, but less intuitive and less finger-friendly.

Algorithms for slot in back-right position.

Incorrectly Connected Pieces

(R' U R) U2' y (R U R')
(R2' F R F' R) U2' (R' U R)

y (R U' R' U2) y' (R' U' R)
U (R U' R' U) (R' U' R)
Note – the second algorithm should only be used when the front-right slot is empty.

y (R U2 R') U' (R U R')

(R' U2' R) U (R' U' R)

(R' U' R U' y) (R U' R' U) (R U' R')
(U R' U2' R) y (R U2' R' U) (R U' R')

U' (R' U R U) (R' U R U') (R' U R)
y F (U R U' R') F' (R U' R')

Corner in Place, Edge in U Face

(U' R' U R) y U (R U' R')

y U (R U' R') U' (F' U F)
y U (R U' R') (F R' F' R)

y (R U' R' U) (R U' R')
R' U2 R' F R F' R

(R' U R U') (R' U R)

(R' U' R U) (R' U' R)

y (R U R' U') (R U R')

Edge in Place, Corner in U face

(R' U R' F) (R F' R)
(R' U R U') y (R U' R')

(U' R' U R) (U' R' U R) (U' R' U R)

(U' R' U' R) U2 (R' U' R)

U (R' U R) U2' (R' U R)

y (U' R U R') d (R' U' R)

U (R' U' R) y U' (R U R')

Edge and Corner in Place

Solved Pair

(R' U R) d' (R U2' R') U2 (R U' R')

(R' U R U') (R' U2 R U') (R' U R)

(R' U' R U) (R' U2' R) U (R' U' R)
(R' U R U) (R' U' R U2') (R' U R)

(R' U R) U2' y (R U R' U) (R U' R')

(R' U R U) (R' U R U') y (R U R')

F2L Algorithms – All Four Slot Angles

Developed by Feliks Zemdegs
and Andy Klise

Images sourced from Conrad Rider's VisualCube - http://cube.crider.co.uk/visualcube.php

Algorithms for slot in front-left position.

Basic Inserts

y' U (R U' R')
y U (L U' L')

U' (L' U L)

(L' U' L)

y' (R U R')
y (L U L')

F2L Case 1

U (L' U' L U') (L' U' L)

U (L' U L U') y' (R U R')

L U2' L2' U' (L2 U' L')

U (L' U2 L) U' y' (R U R')

U (L' U L U') (L' U' L)

y' U' (R U' R' U) (R U R')

F2L Case 2

y' (U' R U R') U2 (R U' R')
(L U L') y' (U R U' R')
Note – the second algorithm should only be used when the back-left slot is empty.

(U L' U' L) U2' (L' U L)

y' U' (R U2' R') U2 (R U' R')

U (L' U2 L) U2' (L' U L)

F2L Case 3

y' U (R U2' R') U (R U' R')

U' (L' U2 L) U' (L' U L)

F R U2' R' F'

U2 (L' U' L) U' (L' U L)

Algorithms for slot in front-left position.

Incorrectly Connected Pieces

(L' U L) U2' y' (R U R')

y' (R U' R' U2) y' (R' U' R)
(U' R U' R') U' (L' U' L)
Note – the second algorithm should only be used when the front-right slot is empty.

y' (R U2 R') U' (R U R')

(L' U2 L) U (L' U' L)

y' U (R U' R' U') (R U' R' U R U' R')
y' (R U R' U2') (R U R' U') (R U R')

U' (L' U L U) (L' U L U') (L' U L)
y' F (U R U' R') F' (R U' R')

Corner in Place, Edge in U Face

(U' L' U L) d (R U' R')

y' U (R U' R') U' (F' U F)
y' U (R U' R') (F R' F' R)

y' (R U' R' U) (R U' R')

(L' U L U') (L' U L)

(L' U' L U) (L' U' L)

y' (R U R' U') (R U R')
(r U' r' F) (r U' r' F)

Edge in Place, Corner in U face

(L' U L U') y' (R U' R')

(U' L' U L) (U' L' U L) (U' L' U L)

(U' L' U' L) U2 (L' U' L)

U (L' U L) U2' (L' U L)

y' (U' R U R') d (R' U' R)

U (L' U' L) y' U' (R U R')

Edge and Corner in Place

Solved Pair

(L' U L U') y' (R U2' R' U2 R U' R')

(L' U L U') (L' U2 L U') (L' U L)

(L' U' L U) (L' U2' L) U (L' U' L)
(L' U L U) (L' U' L U2') (L' U L)

(L' U L) F R U2' R' F'

(L' U L U) (L' U L U') y (L U L')

F2L Algorithms – All Four Slot Angles

Developed by Feliks Zemdegs
and Andy Klise

Images sourced from Conrad Rider's VisualCube - http://cube.crider.co.uk/visualcube.php

Algorithms for slot in back-left position.

Basic Inserts

U (L U' L')

y (R' U' R)

y U' (R' U R)
y' U' (L' U L)

(L U L')

F2L Case 1

y U (R' U' R U') (R' U' R)

y R U2' R2' U' (R2 U' R')

y U (R' U R U') (R' U' R)

U' (L U L' U) (L U L')

L' U2 L2 U (L2' U L)

U' (L U' L' U) (L U L')

F2L Case 2

(U' L U L') U2 (L U' L')

U' (L U2 L') U2 (L U' L')

y (U R' U' R) U2' (R' U R)

y U (R' U2' R) U2' (R' U R)

F2L Case 3

U (L U2' L') U (L U' L')

y' F R U2' R' F'

y U' (R' U2 R) U' (R' U R)

y U2 (R' U' R) U' (R' U R)
y R' F' U2 F R
Note – The second algorithm is fewer moves, but less intuitive and less finger-friendly.

Algorithms for slot in back-left position.

Incorrectly Connected Pieces

y (R' U R) U2' y (R U R')
U' (L' U L U') (L U' L')
Note – the second algorithm should only be used when the front-left slot is empty.

(L U' L' U2) y (R' U' R)

(L U2 L') U' (L U L')

y (R' U2' R) U (R' U' R)

U (L U' L' U') (L U' L' U) (L U' L')

y U' (R' U R U) (R' U R U') (R' U R)

Corner in Place, Edge in U Face

[y2]
y' (U' L' U L) d (R U' R')

[y2]
U (L U' L') d' (R' U R)

[y2]
(L U' L' U) (L U' L')

[y2]
y (R' U R U') (R' U R)

[y2]
y (R' U' R U) (R' U' R)

[y2]
(L U L' U') (L U L')

Edge in Place, Corner in U face

[y2]
y (R' U R' F) (R F' R)

[y2]
(U L U' L') (U L U' L') (U L U' L')

[y2]
(U' L U' L') U2 (L U' L')

[y2]
U (L U L') U2 (L U L')

[y2]
(U2' L U L') d' (R' U R)

[y2]
y U2' (R' U R) d (L U L')

Edge and Corner in Place

[y2]
Solved Pair

[y2]
y (R' U R) d' (R U2' R' U2 R U' R')

[y2]
(L U L' U') (L U2 L' U') (L U L')

[y2]
(L U' L' U) (L U2' L') U (L U' L')

[y2]
y (R' U R) U2' y (R U R' U R U' R')

[y2]
(L U' L') d' (U' R' U' R U') (R' U R)