---

description: History of the Void Swap blindfold solving method for the Rubik's Cube.

---

import AnimCube from "@site/src/components/AnimCube";

import ReactPlayer from 'react-player'

import ImageCollage from '@site/src/components/ImageCollage';

# Void Swap

<AnimCube params="config=../../ReconstructionConfig.txt &initmove=U' D B R' B U' D2 F' L2 D B2 L2 U' F2 L2 U D2 L2 F2 R&move={Scramble: U' D B R' B U' D2 F' L2 D B2 L2 U' F2 L2 U D2 L2 F2 R}{Special case: M' S' M2 S M'}M' S' M2 S M'.{RB: R2' F' U' (R' F2 R2 E' R' S' R' F2 R S) U F R2}R2' F' U' (R' F2 R2 E' R' S' R' F2 R S) U F R2.{LU: U' F R (R' F2 R2 E' R' S' R' F2 R S) R' F' U}U' F R (R' F2 R2 E' R' S' R' F2 R S) R' F' U.{FL: F2 R (R' F2 R2 E' R' S' R' F2 R S) R' F2}F2 R (R' F2 R2 E' R' S' R' F2 R S) R' F2.{UR: (R' F2 R2 E' R' S' R' F2 R S)}(R' F2 R2 E' R' S' R' F2 R S).{RU: U F R (R' F2 R2 E' R' S' R' F2 R S) R' F' U'}U F R (R' F2 R2 E' R' S' R' F2 R S) R' F' U'.{UB: U (R' F2 R2 E' R' S' R' F2 R S) U'}U (R' F2 R2 E' R' S' R' F2 R S) U'.{FR: R (R' F2 R2 E' R' S' R' F2 R S) R'}R (R' F2 R2 E' R' S' R' F2 R S) R'.{DB: D' R2 (R' F2 R2 E' R' S' R' F2 R S) R2 D}D' R2 (R' F2 R2 E' R' S' R' F2 R S) R2 D.{LD: D2 R2 (R' F2 R2 E' R' S' R' F2 R S) R2 D2}D2 R2 (R' F2 R2 E' R' S' R' F2 R S) R2 D2.{FD: F' R (R' F2 R2 E' R' S' R' F2 R S) R' F}F' R (R' F2 R2 E' R' S' R' F2 R S) R' F.{UF: U' (R' F2 R2 E' R' S' R' F2 R S) U}U' (R' F2 R2 E' R' S' R' F2 R S) U.{RD: R F' U' (R' F2 R2 E' R' S' R' F2 R S) U F R'}R F' U' (R' F2 R2 E' R' S' R' F2 R S) U F R'.{UB: U (R' F2 R2 E' R' S' R' F2 R S) U'}U (R' F2 R2 E' R' S' R' F2 R S) U'.{DB: D' R2 (R' F2 R2 E' R' S' R' F2 R S) R2 D}D' R2 (R' F2 R2 E' R' S' R' F2 R S) R2 D.{UBR: U (R B R' U2' r U' r B r2' U2') U'}U (R B R' U2' r U' r B r2' U2') U'.{FLU: F2 R (R B R' U2' r U' r B r2' U2') R' F2}F2 R (R B R' U2' r U' r B r2' U2') R' F2.{DLF: F2 (R B R' U2' r U' r B r2' U2') F2}F2 (R B R' U2' r U' r B r2' U2') F2.{FUR: R U (R B R' U2' r U' r B r2' U2') U' R'}R U (R B R' U2' r U' r B r2' U2') U' R'.{LBU: U R' (R B R' U2' r U' r B r2' U2') R U'}U R' (R B R' U2' r U' r B r2' U2') R U'.{DRB: R2 (R B R' U2' r U' r B r2' U2') R2}R2 (R B R' U2' r U' r B r2' U2') R2.{DFR: R2 U (R B R' U2' r U' r B r2' U2') U' R2}R2 U (R B R' U2' r U' r B r2' U2') U' R2" width="600px" height="400px" />

\*\*Example solve from Vincent Trang\*\*

## Description

\*\*Creator:\*\* [Vincent Trang](CubingContributors/MethodDevelopers.md#trang-vincent-trangium)

\*\*Created:\*\* 2021

\*\*Steps:\*\*

1. Solve edges one at a time, swapping two edges and four centers per target.

2. Solve corners one at a time, with LDB as the buffer and setting up to UFR as the target.

[Click here for more step details on SpeedSolving.com](https://www.speedsolving.com/threads/void-swap-a-unique-beginner-3bld-method.85777/)

## Origin and Development

While learning blindfold solving, Vincent Trang found that it wasn’t beginner friendly to remember the setups to solve pieces, and to also have to reverse the setups after pieces are solved [1]. Around the same time, Trang was experimenting with FMC techniques such as edge and center insertions and how they relate with the NISS technique. This led Trang to make a connection from the FMC techniques to the thought that there may be an easier way of blindfold solving. The realization was that two edges could be swapped when centers are also swapped.

On April 30, 2021, Trang posted the preliminary idea to the SpeedSolving.com forum [2]. This original version of the method used the buffer BD and the target UF while cycling the M slice centers. After further refining the process, and taking into account feedback received from the initial post, Trang posted a dedicated thread officially introducing Void Swap to the puzzle community [3]. The edge buffer and target was updated to become LB and UR, with the new edge swap algorithm now cycling the E slice centers.

<ImageCollage

images={[

{ src: require("@site/docs/BlindfoldSolving/img/VoidSwap/VoidSwap1.png").default},

{ src: require("@site/docs/BlindfoldSolving/img/VoidSwap/VoidSwap2.png").default}

]}

/>

## Previous Proposal

In November 2005, Stefan Pochmann suggested a similar idea of swapping two edges while cycling centers [4]. Pochmanns version swapped the edges at UR and UL and cycled the S slice centers. However, the idea received no further development beyond the introduction of an alternate edge swapping algorithm [5].

<ImageCollage

images={[

{ src: require("@site/docs/BlindfoldSolving/img/VoidSwap/Pochmann.png").default},

{ src: require("@site/docs/BlindfoldSolving/img/VoidSwap/Pochmann2.png").default},

{ src: require("@site/docs/BlindfoldSolving/img/VoidSwap/Pochmann3.png").default}

]}

/>

## References

# References

|  |  |
| --- | --- |
| [1] | M. Straughan and V. Trang, Personal Communication, 5 April 2025. [Online]. |
| [2] | V. Trang, "The New Method / Substep / Concept Idea Thread," SpeedSolving.com, 30 April 2021. [Online]. Available: https://www.speedsolving.com/threads/the-new-method-substep-concept-idea-thread.40975/post-1439675. |
| [3] | V. Trang, "Void Swap: A Unique Beginner 3BLD Method," SpeedSolving.com, 21 November 2021. [Online]. Available: https://www.speedsolving.com/threads/void-swap-a-unique-beginner-3bld-method.85777/. |
| [4] | S. Pochmann, "just swapping 2 pieces," Blindfold Solving Rubik's Cube Yahoo! Group, 22 November 2005. [Online]. |
| [5] | S. Pochmann, "Re: just swapping 2 pieces," Blindfold Solving Rubik's Cube Yahoo! Group, 23 November 2005. [Online]. |