



MODELOS DISCRETOS PARA INGENIERÍA EN ITI

NRC 9901

ALGORITMO DE RESOLUCIÓN PARA UN CUBO RUBIK

Profesor: WASHINGTON EDUARDO LOZA HERRERA

GRUPO

ALISON TAMAYO

MERLYN ULCUANGO

KAREN YANEZ

JUNIO 2023

Índice

| | |
|---|----------|
| Resumen..... | 3 |
| 1. Introducción | 3 |
| 1.1. Definiciones básicas | 4 |
| 1.1.1. Piezas del cubo..... | 4 |
| 1.1.2. Notación y movimientos básicos | 4 |
| 2. Metodología | 5 |
| 3. Implementación..... | 5 |
| 3.1. Lenguaje de programación | 6 |
| 3.2. Funciones de mezcla y orden del cubo rubick (JavaScript)..... | 6 |
| 3.2.1. Función mezclar | 6 |
| 3.2.2. Función ordenar | 7 |
| 4. Conclusiones | 8 |
| 5. Bibliografía | 8 |

Resumen

En este trabajo, se tuvo como objetivo principal el desarrollo y la implementación de un algoritmo capaz de resolver el cubo Rubik, también conocido como cubo mágico. Se consideró el hecho de que existen más de 43 trillones de posibles combinaciones para este cubo, y se buscó encontrar una solución efectiva para cualquier configuración dada. Para lograr esto, se llevó a cabo una exhaustiva investigación que involucró la definición de pasos específicos basados en la técnica de Backtracking. Durante el proceso de investigación, se exploraron diversos procedimientos con el fin de reducir el espacio de búsqueda y obtener así un algoritmo más eficiente en términos de tiempo y recursos utilizados.

Palabras claves: Implementación, cubo Rubik, combinaciones, algoritmo.

1. Introducción

Ernő Rubik, profesor de arquitectura en la Universidad de Budapest en Hungría, inventó el famoso cubo de Rubik en 1974 (Wikipedia, 2005). Aunque se dice con frecuencia que el cubo fue construido como herramienta escolar para ayudar a los estudiantes a comprender objetos tridimensionales, su verdadero objetivo era resolver el problema estructural que requería la capacidad de mover las partes independientemente sin que el mecanismo completo se desmoronara.

El cubo de Rubik es un rompecabezas de forma cúbica que consta de 26 piezas móviles que se pueden girar de lado en lado. Cada cara del cubo es de un color en su estado original, pero girar cada una de estas permite que los cubos más pequeños se combinen de varias maneras. Esto se debe a que, por una parte, ¡podemos combinar todas las esquinas de cualquier manera, lo que da como resultado $8!$ posibilidades. ¡Lo mismo ocurre con las aristas, es decir, que podemos combinarlas de todas las formas posibles da $12!$ posibilidades, pero la permutación total de esquinas y aristas debe ser igual, lo que elimina la mitad de las posibilidades. Además, tenemos la capacidad de girar todas las esquinas de la forma que deseemos, excepto una, sin alterar el resto del cubo. La orientación de la última esquina dependerá de la que tengan las otras siete, lo que nos da 3^7 opciones. Lo mismo se aplica a las aristas, lo que significa que tenemos 2^{11} opciones adicionales (Hernández, 2005). En total tendremos que el número de permutaciones posibles en el Cubo de Rubik es de:

$$(8! 12! 3^7 2^{11})/2 = 43.252.003.274.489.856.000$$

1.1. Definiciones básicas

1.1.1. Piezas del cubo

Las esquinas, las aristas y las centrales son las tres clases de piezas que componen el cubo. Cada una de estas piezas tiene 1, 2 o 3 colores distintivos. Es importante destacar que en realidad son las aristas y las esquinas las que se mueven, ya que las partes centrales siempre se encuentran en la misma posición relativa entre sí. El único movimiento que el cubo puede realizar es girar una o más veces sus caras sin mover las partes centrales de su posición original (Ridao & Vidal, 2023).

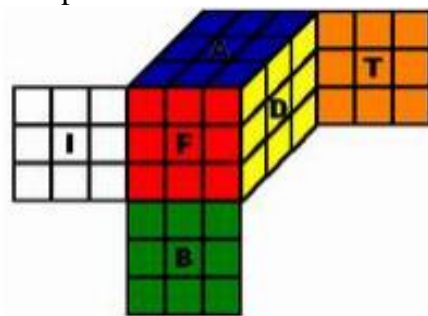
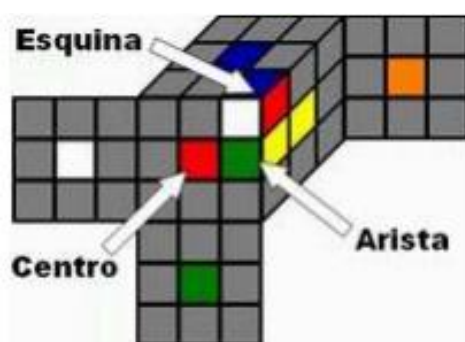


Figura 1. Piezas del cubo

1.1.2. Notación y movimientos básicos

Primero se debe asignar una anotación adecuada, donde se nombrará cada una de sus caras con una letra mayúscula, la cual se describirá los movimientos que tendrá el cubo:



- La cara de Arriba: A
- La cara de aBajo: B
- La cara de la Izquierda: I
- La cara de la Derecha: D
- La cara del Frente: F
- La cara de aTrás: T

Figura 2. Notación Cubo de Rubik

El cubo lo resolveremos por capas, que son 3, cada capa tiene sus propios pasos, lo que da como resultado un total de 7 pasos.

Primera capa:

- **Cruz:** Resolución de las aristas de la primera capa.
- **Esquinas:** Resolución de las esquinas de la primera capa.

Segunda capa:

- **Aristas:** Resolución de las aristas de la segunda capa.

Tercera capa:

- **Orientación de aristas:** Orientación correcta de las aristas de la tercera capa.
- **Permutación de aristas:** Colocación de las aristas en su lugar correspondiente.
- **Permutación de esquinas:** Colocación de las esquinas de la tercera capa en su lugar correspondiente.
- **Orientación de esquinas:** Orientación de las esquinas de la tercera capa.



Figura 3. Movimientos del cubo

2. Metodología

La metodología aplicada se basa en un enfoque sistemático y disciplinado para el desarrollo de software, utilizando técnicas de programación rigurosas y principios de diseño orientados a objetos. Esta metodología se enfoca en la construcción de algoritmos eficientes y en el uso de estructuras de datos apropiadas para resolver problemas complejos. Algunos aspectos clave de esta metodología incluyen:

- **Abstracción y especificación:** Se busca desarrollar una especificación clara e independiente de la representación de los datos y del lenguaje de programación. Esto permite un análisis predictivo de la complejidad temporal y espacial, ayudando a seleccionar las mejores estructuras de datos y algoritmos.
- **Análisis de complejidad:** Se realiza un análisis detallado de la complejidad temporal y espacial de los algoritmos propuestos. Esto permite evaluar su eficiencia y rendimiento antes de implementarlos, facilitando la toma de decisiones informadas.
- **Reutilización de componentes:** Se fomenta la reutilización de componentes existentes, como bibliotecas de código o módulos previamente desarrollados, para aprovechar soluciones probadas y acelerar el proceso de desarrollo.

3. Implementación

El Cubo Rubik ofrece una gran cantidad de combinaciones posibles en sus colores. Para abordar la resolución del Cubo Rubik, se han desarrollado diversos métodos basados en

una serie de movimientos. Para ordenar el cubo se necesita un Sistema de Resolución, es lo que se denomina Algoritmo, por lo que en este trabajo nos enfocamos en el algoritmo que tengan las funciones para mover un número determinado de aristas o vértices desde un punto central sin deshacer el resto del cubo y de igual manera que se pueda volver a ordenar.

3.1. Lenguaje de programación

El lenguaje utilizado en el contexto del Cubo Rubik, se utiliza una combinación de HTML, CSS y JavaScript para crear una representación interactiva del Cubo Rubik en un entorno web.

- El HTML se utiliza para estructurar y definir los elementos de la página web, como los controles, la escena y el pie de página.
- El CSS se utiliza para aplicar estilos y dar formato visual a los elementos de la página, como el color de fondo y el diseño del pie de página.
- El JavaScript se utiliza para agregar interactividad y funcionalidad al Cubo Rubik, como los movimientos y la lógica de resolución. Tomando en cuenta el uso de librerías.

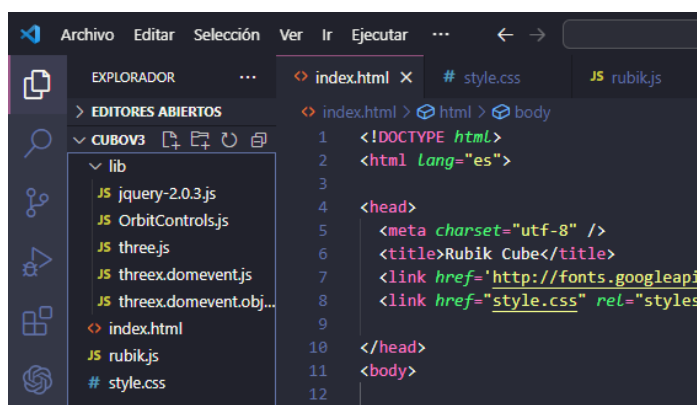


Figura 4 Proyecto CuboV3 programados

3.2. Funciones de mezcla y orden del cubo rubick (JavaScript)

3.2.1. Función mezclar

Aquí está el funcionamiento del método shuffle():

Dentro del método shuffle(), se definen tres funciones de utilidad:

randomAxis(): Devuelve una cadena aleatoria que representa uno de los ejes de rotación del cubo ('x', 'y' o 'z').

randomDirection(): Devuelve un número aleatorio que representa la dirección de rotación (-1 o 1).

`randomCube()`: Devuelve un cubo aleatorio seleccionado de la matriz `allCubes` que contiene todos los cubos del cubo de Rubik.

Se genera un número aleatorio `nMoves` que representa la cantidad de movimientos aleatorios que se realizarán.

Se realiza un bucle `for` que se ejecuta `nMoves` veces. En cada iteración:

Se selecciona un cubo aleatorio llamando a la función `randomCube()`.

Se obtiene una posición inicial del cubo llamando a `cube.position.clone()`.

Se selecciona un eje de rotación aleatorio llamando a `randomAxis()`.

Se selecciona una dirección de rotación aleatoria llamando a `randomDirection()`.

Se llama a la función `pushMove()` para agregar el movimiento a la cola de movimientos.

Finalmente, se llama a `startNextMove()` para comenzar el siguiente movimiento en la cola de movimientos.

En resumen, el método `shuffle()` se utiliza para generar una secuencia aleatoria de movimientos del cubo de Rubik.

3.2.2. Función ordenar

Aquí está el funcionamiento del método `solve()`:

Primero se verifica si no hay movimientos en curso (`!isMoving`). Esto se hace para asegurarse de que no se esté realizando ningún movimiento actualmente.

Dentro del condicional, se realiza un bucle `forEach` en la matriz `completedMoveStack` que contiene los movimientos completados anteriormente.

En cada iteración, se agrega un movimiento inverso a la cola de movimientos llamando a la función `pushMove()` con los parámetros del movimiento original pero con la dirección multiplicada por `-1`. Esto es necesario para deshacer los movimientos completados y volver al estado original del cubo de Rubik.

Después de agregar los movimientos inversos a la cola de movimientos, se vacía la matriz `completedMoveStack` para restablecerla.

Se utiliza el método `one()` del objeto `moveEvents` para escuchar el evento `'deplete'` una vez. Este evento se activa cuando la cola de movimientos se vacía por completo.

Finalmente, se llama a `startNextMove()` para comenzar el siguiente movimiento en la cola de movimientos, lo que iniciará el proceso de resolución del cubo de Rubik.

En resumen, el método solve() utiliza los movimientos completados anteriormente para generar una secuencia de movimientos inversos que deshacen los movimientos realizados. Luego, inicia el proceso de resolución del cubo de Rubik ejecutando los movimientos inversos en orden.

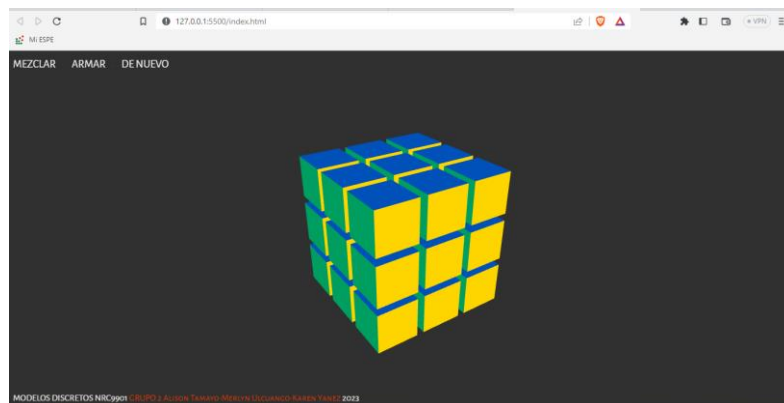


Figura 5 Funcionalidad programa

4. Conclusiones

En conclusión, el Cubo Rubik ofrece un amplio espectro de combinaciones y la resolución del mismo se basa en la utilización de algoritmos que permiten mover ciertas partes del cubo sin deshacer el resto. La longitud de los algoritmos puede variar, desde pocos movimientos hasta más de una decena, dependiendo de la dificultad del cubo y del método utilizado para resolverlo.

Los algoritmos utilizados para resolver el cubo de Rubik se basan en una combinación de movimientos y secuencias predefinidas que se aplican repetidamente hasta alcanzar la configuración deseada. Estos algoritmos están diseñados para manipular las piezas del cubo de manera específica y lograr progresivamente la solución final.

5. Bibliografía

- Hernández, C. (2005). El cubo de Rubik de la A a la Z. Obtenido de <http://usuarios.lycos.es/rubikaz/>
- Ridao, I., & Vidal, S. (2023). Algoritmos de resolución para el cubo de Rubik. Universidad Nacional del Centro de la Provincia de Buenos Aires.
- Wikipedia. (2005). Cubo de Rubik. Obtenido de http://en.wikipedia.org/wiki/Rubiks_Cube