

Simulador de un cubo de Rubik

Programación Orientada a Objetos

2CM1

Humberto Alejandro Ortega Alcocer

Tarea 1

Opción #3: Simulador de un cubo de Rubik.

7 de octubre de 2020

Pocos juguetes son lo suficientemente entretenidos y complejos como para atraer nuevos jugadores, sin embargo, solamente unos pocos son capaces de conservar adeptos aún cuando éstos tengan una gran cantidad de práctica en los mismos. Ocupando uno de los lugares a lado de juegos y rompecabezas más grandes de todos los tiempos como el Ajedrez, las Damas o Go, el cubo de Rubik es uno de los rompecabezas más exitosos (y más vendidos) del siglo XX.

Su propuesta simple y elegante, presenta un cubo con seis colores correspondientes a las caras disponibles en la figura (amarillo, rojo, azul, verde, naranja y blanco) en dónde cada cara, a su vez, se subdivide en 9 pequeños recuadros que son manipulables e intercambiables mediante la rotación de los distintos ejes presentes en el cubo, distribuyendo aleatoriamente los colores presentes en cada cara. El objetivo es simple: agrupar los pequeños recuadros por color de tal forma que, finalmente, el cubo cuente con un solo color por cara.

La popularidad del cubo de Rubik puede ser apreciada por la cantidad de algoritmos que se han descubierto para la resolución del mismo, buscando cada vez un menor número de pasos y tiempo requerido para resolver el rompecabezas. Con mas de 43,252,003,274,489,856,000 (¡43 trillones!) combinaciones posibles (para un cubo de Rubik estándar de 3x3), no se trata de una tarea trivial, y es muy común que los jugadores memoricen los algoritmos mas simples a pesar de que quizá no sean los más rápidos o eficientes.

La simulación de un cubo de Rubik, por consiguiente, no representa en sí una solución, sino que funge como una herramienta que permite verificar y evaluar el desempeño de los algoritmos planteados para resolver el cubo. Mientras la propuesta obvia de simulación responde a una representación gráfica del cubo en dónde sea posible realizar los movimientos a petición del usuario, considero que exhibir un simulador dónde sea posible redactar, ejecutar y analizar distintos algoritmos para resolver el rompecabezas, representa una propuesta de valor

mucho más concreta y ambiciosa. Es factible, bajo una óptica que parte del ámbito pedagógico, presentar dicho simulador con una interfaz gráfica opcional que permita al usuario visualizar, avanzar o retroceder, en cada uno de los pasos que, el algoritmo en cuestión, está llevando a cabo para resolver el cubo.

Un simulador de tal estilo puede sonar a una tarea imposible de realizar, más aún si consideramos las complejidades que conllevan las tareas de simulación viéndolas desde una perspectiva de *programación estructurada*, dónde el modelado se centraría en las rutinas y sub-rutinas que conforman la interacción del cubo resultando en un código extremadamente complejo que requeriría un extenso dominio para poder realizar cambios aún siendo minúsculos. Sin embargo, uno de los beneficios evidentes de cambiar a un paradigma *orientado a objetos* es la capacidad de poder descomponer nuestro problema en segmentos de código mantenibles y reutilizables que simplifiquen la tarea de adaptación del mundo real y tangible, es decir, un cubo de Rubik, a un plano digital y virtual como lo es nuestro simulador propuesto.

Para comenzar con el planteamiento del simulador, hagamos un ejercicio de descomposición del cubo en sí mismo para poder entender las partes involucradas así como empezar a entender la interacción que tienen entre sí. La intención es partir de lo más general y extendernos hasta lo más específico sin perder de vista el objetivo general que es: simular la operación de un cubo de Rubik. En primer lugar, entonces, hablaremos del cubo como entidad y es que, el cubo en sí, presenta una serie de características particulares, como la homogeneidad de su figura, volviendo su apropiada manipulación física uno de los retos primarios del acertijo, así como la condición de que el cubo siempre debe terminar cada movimiento con el cubo conservando su figura original, no se permiten movimientos intermedios de ningún estilo.

Si bien el cubo es un objeto con características específicas, dentro de un cubo de Rubik encontramos en el concepto de “cara” podemos comenzar a definir algunas reglas de comportamiento de nuestro simulador, como incluir seis de estas caras, las cuales permitirán determinar el estado del juego ya que, en cuanto todas las caras se encuentren completamente llenas de un sólo color, el rompecabezas estará resuelto, o dado que los movimientos de una cara impactan en todas las caras aledañas, siempre debemos asegurarnos de realizar los cambios adecuadamente en todas las caras colindantes.

El siguiente paso en este análisis consiste en tratar de ir, aún más, hacia lo específico, llegando finalmente al “cuadro”. Los cuadros son cincuenta y cuatro pequeñas figuras cuadradas, rellenas de un color inmutable durante el juego, que se distribuyen en agrupaciones de nueve elementos que comparten el mismo color para cada una de las caras (seis). Estos pequeños elementos, son la base del rompecabezas. En ellos se centra el gigantesco número de posibles combinaciones y también se centra la posibilidad de completar el acertijo. Las reglas son simples. Un cuadro inicia con un color y mantendrá tal color hasta el fin de la simulación en cuestión. Un cuadro nunca podrá desplazarse en movimientos que no estén comprendidos por los ejes de intersección de las caras exteriores del cubo. Un cuadro siempre tendrá ocho cuadros a su alrededor en cualquier momento dado, esto despreciando las limitantes que existan entre caras colindantes.

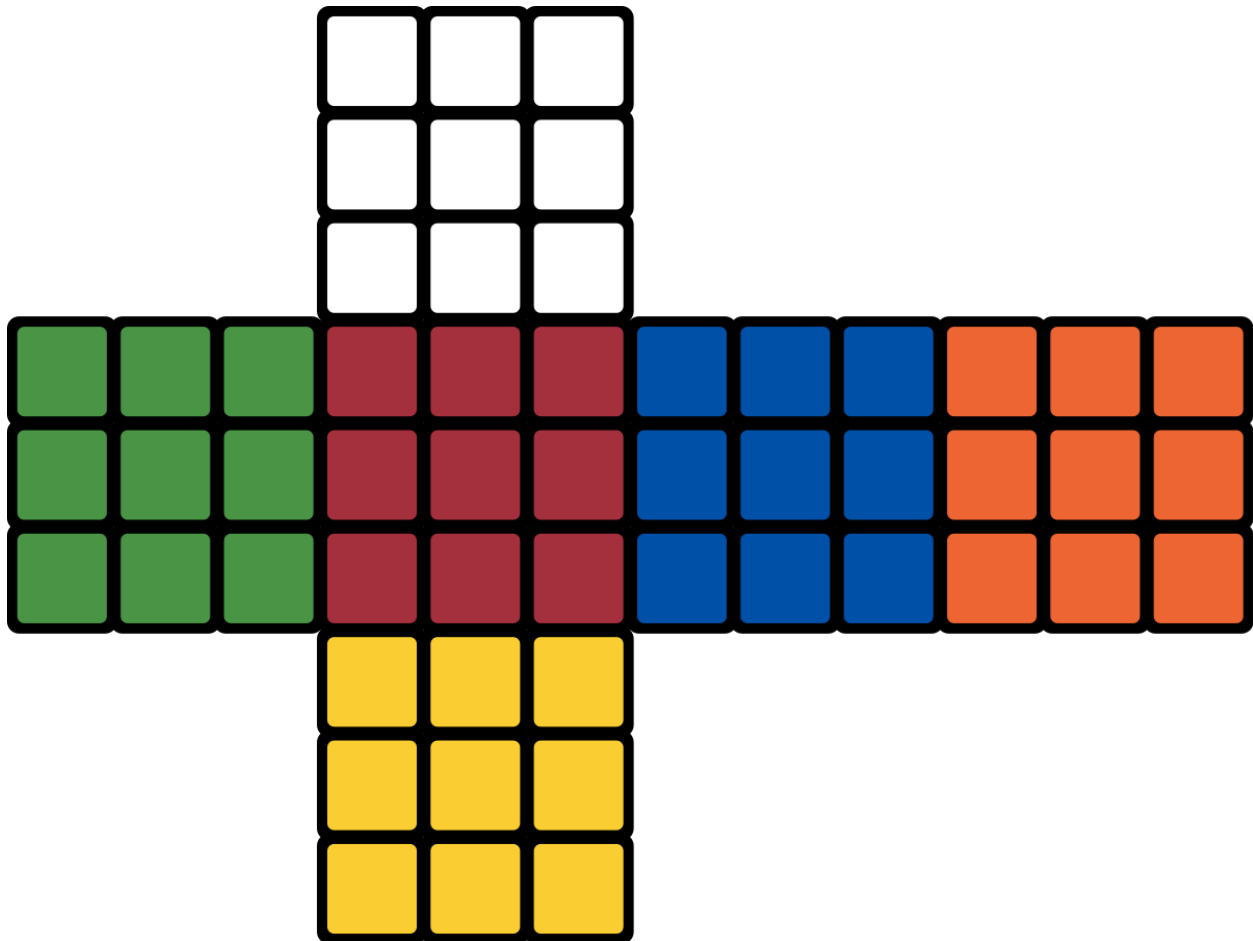
Si bien hemos definido algunas de las reglas de comportamiento de nuestras entidades hasta ahora, las operaciones disponibles para cada una de éstas, ha sido omitida. Esta omisión responde a la dependencia que tienen las operaciones sobre un elemento básico y fundamental que ya fue mencionado anteriormente, pero al no ser un elemento tangible del cubo, evade nuestro primer intento de análisis: la perspectiva. Y es que, aunque simular un cubo de Rubik pudiera parecer una tarea

en dónde pudiéramos pensar en atacar el problema únicamente simulando las matrices de cubos y realizando las operaciones necesarias para transformar dichas matrices en función de la dirección especificada, es necesario comprender cuál será la perspectiva que tendremos del cubo en cualquier momento dado para poder representar los movimientos de una forma agradable para el usuario. Un simulador de cubo de Rubik respetable (al menos bajo los estándares de hoy en día) debe ostentar, cuando menos, algún nivel de interactividad con un modelo en 2-D o 3-D, pero, considerar la perspectiva del usuario como un elemento a considerar en nuestro modelado del rompecabezas, nos permite ofrecer una interfaz gráfica sencilla y simple de implementar mediante un modelo renderizado en 3-D a la par que mantenemos el control del juego con el enfoque mencionado anteriormente. Tener la perspectiva actual del usuario, nos permite entender (y posteriormente analizar) los movimientos que toma el usuario así como ofrecer una experiencia de visualización sin igual para los que deseen únicamente observar un algoritmo ejecutándose en dicho cubo.

Entonces, tenemos un cubo, que tiene seis caras, cada una con nueve recuadros de un solo color y una perspectiva que nos permite saber qué está observando el usuario en ese momento, ahora es momento de trabajar en las operaciones que debemos realizar así como en las implicaciones que conllevan dichas operaciones.

En el cubo, las operaciones disponibles son simples: girar cubo arriba, girar cubo abajo, girar cubo izquierda, girar cubo derecha, que realmente sólo cambian el orden en el que se visualizan las caras en la ventana, para esto cambiaremos el tipo de perspectiva que estemos usando en función de estos movimientos. El siguiente elemento a analizar son las caras, aquí, los movimientos clásicos del cubo de Rubik salen a relucir: girar los ejes. Para analizar cómo se comportan estos

movimientos, a continuación una representación bi-dimensional de un cubo de Rubik clásico de 3x3:



Como podemos observar, las distintas caras cuentan con un estado inicial en donde todos los recuadros se encuentran con el mismo color. Para realizar un movimiento, tendremos que seleccionar una esquina para “empujar” con lo cual determinaremos la ubicación, la dirección y los cambios de ubicación necesarios para poder completar el movimiento.

En un cubo físico, al realizar el “empuje” de una de las esquinas para así realizar la rotación de los ejes comprendidos en los extremos del cubo, vemos cómo no únicamente los elementos colindantes que comparten la dirección del

empuje son afectados por el movimiento sino que varios colores más, pertenecientes a caras ubicadas en los extremos de la línea correspondiente a la dirección del empuje del usuario, también son modificadas. Este hecho, es el que vuelve irresistible este pequeño cubo para las personas que son adictas a resolver acertijos y problemas, cualquier movimiento puede acercarte un paso más a la solución del acertijo, o puede alejarte 50 pasos.

Para representar el movimiento, realizaremos un recorrido por las matrices de cuadros (caras) en la dirección del empuje y actualizaremos el color de los recuadros según se comporta el movimiento dentro del cubo.

Regresando al primer nivel, el cubo, aquí llevaremos el control del juego, para ello usaremos el concepto de turno, al iniciar cada turno el usuario selecciona la esquina a empujar, se realizará el movimiento y, finalmente, se realizará un chequeo de las funciones verificadoras de cada cara para saber si todas las caras ya cuentan, únicamente, con cuadros de un solo color, con lo cual habremos de declarar el juego como finalizado.

El cubo de Rubik no es un juego que tome poco tiempo. De hecho, sin un conocimiento sobre algún algoritmo solución para el problema, el acertijo puede volverse imposible, por lo que nuestro simulador deberá implementar algún método para almacenar, en un archivo, el estado del juego y así poder retomarlo cuando más conveniente le sea.

Simular un cubo de Rubik no es una tarea simple de realizar, pero definitivamente la programación orientada objetos sale a relucir en cuanto a entendimiento, implementación y escalabilidad del problema y su respectiva solución mediante el software planteado.