**PROGRAMACIÓN DE CUBO EN JAVA**

**Rómulo Q. Altamirano Yaison H. Mallqui Alex H. Chircca**

***Universidad Nacional José María Arguedas***

**Abstract**

In this article it is presented the resolution of the CUBE or MAGIC CUBE, the first thing it is to use Java applet to encourage the CUBE. The creator of this applet was Josef Jelinek 2001-2004.

The main objective is to show the animation of the manipulation of the CUBE. Their characteristic of this applet tries to approach and to solve some problems.

The main characteristics of this applet are that he/she has a very flexible configuration, the manipulation it is natural and sensitive, the layers separated from the CUBE are compatible with many individual movement types.

**Palabra clave:**

Cubo, cubo mágico o cubo de colores.

**Introducción**:

Un poco de historia del cubo mágico o cubo rubik como quieran llamarlo, fue inventado por el escultor y profesor de arquitectura Húngaro Erno Rubik en 1974 es un rompecabezas mecánico tridimensional.

Gano el premio Alemán a mejor juego de rompecabezas, hasta el momento se han vendido 350 millones de cubos en todo el mundo (2009), y es el más vendido del mundo.

Bueno, el cubo Rubik tiene seis caras cubiertas por nueve pegatinas de seis colores uniformes (tradicionalmente blanco, rojo, azul, verde y amarillo).

El método consiste en resolver el cubo de Rubik por capas, primero la superior, luego la central, y por último la inferior. Con un poco de practica el cubo Rubik se puede armar en menos de 2 minutos, y con mucha practica incluso se puede bajar treinta segundos. (Fuente: Wikipedia).

Crear una aplicación para cubo Rubik es un desafío debido a la amplitud del problema. Las aplicaciones actuales buscan resolver el cubo de manera óptima, eso implica una exploración de un espacio de estados enorme, lo que consumiría mucho tiempo. Hoy en día se usan distintos métodos para encontrar la solución óptima en el menor tiempo posible. El resultado será una cadena de movimientos que resuelve un cubo dado, y dicha solución la obtendremos casi al instante.

**Métodos:**

En el presente trabajo de investigación se realizó utilizando metodología programación orientada a objetos de programación utilizadas en la creación de proyecto CUBO usando applet.

**La Programación Orientada a Objetos (POO).**

En un programa orientado a objetos tendremos a un conjunto de objetos colaborando entre ellos, La orientación a objetos es paradigma de que está de moda para el desarrollo de software, un objeto es una abstracción conceptual del mundo real que se puede traducir a un lenguaje de programación orientado a objetos, un objeto del mundo real tiene características y comportamientos, y de la misma manera, un objeto del mundo del software tiene variables y métodos, una Clase es una plantilla que define las variables y métodos a ser incluidas en un tipo de objeto específico, los objetos también son llamados instancias de la Clase. Los objetos sólo almacenan su estado, los objetos se comunican entre ellos usando los mensajes. Un mensaje es la invocación de un método del objeto, la orientación a objetos requiere de una metodología que integre el proceso de desarrollo y un lenguaje de modelamiento con herramientas y técnicas adecuadas.

**Parámetros:**

**Config** **parámetro** esta configuración establece los valores por defecto de los colores de fondo y permite la visualización de texto para todos los applets con esta configuración de parámetros.

**Bgcolor** **parámetro** determina el color de fondo de applet. Se establece en el color de fondo de la página con el fin de omitir las colisiones de color.

**Butbg** **parámetro** determina el color de los botones. Es similar a la bgcolor y tiene el mismo formato.

Esquema de color **de parámetros** se utiliza para establecer el esquema de colores de cubo resuelto.

**Colores** **parámetro** se utiliza para establecer nueve elementos de color configurables a colores personalizados.

**Posición de** **parámetro** se utiliza para establecer la posición inicial del cubo (su rotación).

**Velocidad** **parámetro** permite personalizar la velocidad de la animación. El valor debe constar de dígitos decimales.

**Doublespeed** **parámetro** permite personalizar la velocidad de la cara se pone por separado de cuartos de vuelta. El valor debe constar solo de dígitos decimales.

**Perspectiva** **parámetro** permite personalizar la deformación perspectiva del cubo. El valor debe constar solo de dígitos decimales.

**Escala** **de parámetro** permite personalizar el tamaño del cubo. El valor debe constar solo de dígitos decimales.

**Alinear** **parámetro** permite posicionar el cubo verticalmente.

**Pista** **parámetro** permite mostrar algunos de los facilidades que enfrentan fuera el usuario.

**Buttonbar** **parámetro** contiene controles de animación.

**Edición** **de parámetros** se desactiva la posición de torcer capas cubo con un ratón.

Hay dos valores posibles “o” y “1”.

**Movetext** **parámetro** se habilita la representación textual de la secuencia de movimientos dentro de la ventana applet.

**Fonttype** **parámetro** se ajusta a la apariencia del texto mostrado hay dos valores posibles: “0” y “1”.

**Métrica**  **parámetro** es el contador de movimientos se ajusta para diferenciar métricas.

Mover **el parámetro** secuencia de movimientos se define en notación extendida de singmaster. La base para las vueltas son seis cartas de la siguiente.

**Conclusiones:**

Se ha presentado el resultado del trabajo de una manera satisfactoria sus características sus objetivos. Se han incluido las figuras con que se ilustran los problemas del cubo.

Los problemas se han presentado con 3 estructura que son los **Paquete buscar, Paquete Figura, Paquete CuboForma** y la utilizacion del algoritmo IDA

**Agradecimientos**

Quiero aprovechar esta oportunidad para dar las gracias a las personas que nos ayudaron con este trabajo, ya que nos permitieron desarrollar este trabajo de una manera satisfactoria.

Agradecer a los amigos y conocidos por la colaboración y por las guías que me brindaron.

**Referencias**

[**http://www.ciberaula.com/articulo/tecnologia\_orientada\_objetos/**](http://www.ciberaula.com/articulo/tecnologia_orientada_objetos/)

[**http://es.wikipedia.org/wiki/Cubo\_de\_Rubik**](http://es.wikipedia.org/wiki/Cubo_de_Rubik)

[**http://programacion.net/articulo/java\_3d\_169/2**](http://programacion.net/articulo/java_3d_169/2)

[**http://ruwix.com/widget/AnimCube.java**](http://ruwix.com/widget/AnimCube.java)

[**http://panamahitek.com/java-3d-programando-en-la-tercera-dimension/**](http://panamahitek.com/java-3d-programando-en-la-tercera-dimension/)

[**http://panamahitek.com/java-3d-programando-en-la-tercera-dimension/**](http://panamahitek.com/java-3d-programando-en-la-tercera-dimension/)

[**http://www.java3d.org/**](http://www.java3d.org/)

**minipaint-mconde.googlecode.com/.../Modelado%20Gráfico%20(**Java**%20...**

**www.jaimegarciaserrano.com/archivos/**RUBIK**\_PDF.pdf**

**Nota**

El juego de aplicación esta implementada para proyección al futuro, también puede ser modificado por otros programadores.