Torres de Hanói

INTRODUCCIÓN A LA PROGRAMACIÓN

RECURSIVIDAD

PROFESOR: CARLOS BENAVIDES

Jason Latouche Jiménez
2015146294
TECNOLÓGICO DE COSTA RICA | ESCUELA DE COMPUATCIÓN

Contenido

Introducción

Este trabajo fue realizado con el propósito de familiarizarse mejor con la programación recursiva y su implementación en problemas de lógica, con lo es el juego de las Torres de Hanói. A lo largo de este documento se explicará lo que son las Torres de Hanói, en que se basan, sus reglas y procedimientos para resolver fácil y eficientemente. También se comentará porque la solución recursiva es mejor implementarla en estos problemas que la iterativa.

Objetivos del proyecto

- Conocer el juego de las Torres de Hanói
- Lograr resolver el juego por medio de implementación de recursividad
- Implementar interfaz gráfica agradable para mostrar por medio de animaciones el funcionamiento de las Torres de Hanói

Torres de Hanói

Es un juego de mesa para una persona, el cual consistía originalmente en una serie de 8 discos cuyo radio incrementa con forme se acercan a la base. Estos discos están apilados en una columna inicialmente, la cual esta incrustada en la misma base donde los discos se apoyan. Hay también otras 2 columnas más; una de ellas es para acomodar todos los discos ahí, la otra es para usarse como apoyo mientras se pasan los discos de una columna a otra.

El objetivo de este entretenido juego es muy sencillo: buscar la manera de mover todos los discos de la columna de origen otra columna cualquiera. No obstante, lo que vuelve el juego aún más interesante son las reglas: solo se puede mover un disco a la vez, antes de mover una siguiente pieza, el disco que se estaba moviendo debe estar en alguna columna. El disco que se vaya a mover debe estar encima de la columna correspondiente. Por último, pero no menos importante, los discos solo se pueden poner encima de la base o de otro disco con radio mayor a él.

Esta última regla es la que complica el juego, y además, el que le permite implementar la solución recursiva. En la próxima sección de este documento veremos cuáles son las técnicas necesarias a tomar en cuenta cuando se valla a resolver este juego.

Solución al juego

La cantidad de anillos a mover es indispensable conocerla para poder resolver el juego en la menor cantidad de movimientos. La fórmula más común usada para conocer la cantidad de movimientos mínimos a realizar para resolver el juego es la siguiente: $2^n – 1$; en donde *n* quiere decir la cantidad de anillos que hay en el juego. Por ejemplos, con tres anillos, la cual es, por cierto, la cantidad de anillos mínima razonable para jugar, podemos aplicarle la fórmula para conocer sus movimientos mínimos: $2^3 – 1 = 7$. Como podemos ver, se tiene toda la razón. Pero ¿Cómo podemos comprobarlo?

Es en este punto es donde la magia de la programación de código y recursión se unen para formar una solución efectiva al juego de las Torres de Hanói.

```
DEF HANOI(LARGO, ORIGEN, AUXILIAR, DESTINO):
  IF LARGO != 0:
    HANOI(LARGO - 1, ORIGEN, DESTINO, AUXILIAR)
    PRINT "MOVER EL DISCO %D DE LA TORRE %D A LA TORRE %D" %(LARGO, ORIGEN, DESTINO)
    HANOI(LARGO - 1, AUXILIAR, ORIGEN, DESTINO)
```

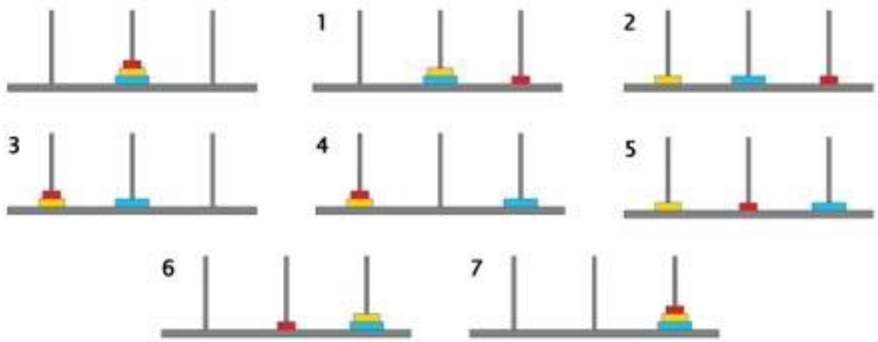
Lo que vemos arriba es el segmento de codigo recursivo que se encarga de dar la instrucción necesaria, en este caso imprimir, para guiar al usuario a resolver el juego existosamente. Explicaremos basicamente porque este codigo tan pequeño es el necesario para resolver el juego.

En la primera linea se define la función de manera que reciba parametros como el largo o cantidad de anillos del juego, un valor u objeto de Origen, Auxiliar y Destino. Recordemos que una varialbe puede almacenar cualqueir elemento que necesitemos. Por eso es que esta solución recursiva servirá tanto asigandole valores enteros a Origen, Auxiliar y Destino, o asignandole valores de objetos. En este ejemplo que se muestra se presenta con números, indicando con el 1, 2 y 3 la columna 1, 2 y 3 respectivamente.

Cada vez que se valla entrando en la misma funcion, se va a ir decrementando el valor de Largo, hasta llegar a 0. En ese caso base se detendrá, para imprimir la posición de destino del primer disco hasta ahora.

La primera función decide los movimientos que basado en la cantidad de discos, hacia que columna dependiendo si es impar o par. En la segunda recursión, al final de codigo, se decide cual será la columna provisional o auxiliar a usar.

Despues de comprender la implementación de esta solución recursiva, es muy sencillo hacerla funcionar para lo que queramos. En nuestro caso, veremos mas adelante que nos fue muy útil para lograr decidir de donde vienen y para donde van las animaciones que se encargan de hacer notar la solución de las Torres de Hanói.



Como vemos, si siguieramos los pasos de codigo anteriormente presentado, y lograramos comparar el resultado que nos dio junto al de la imagen de arriba, notaríamos que es realmente lo mismo.

Historia del Juego

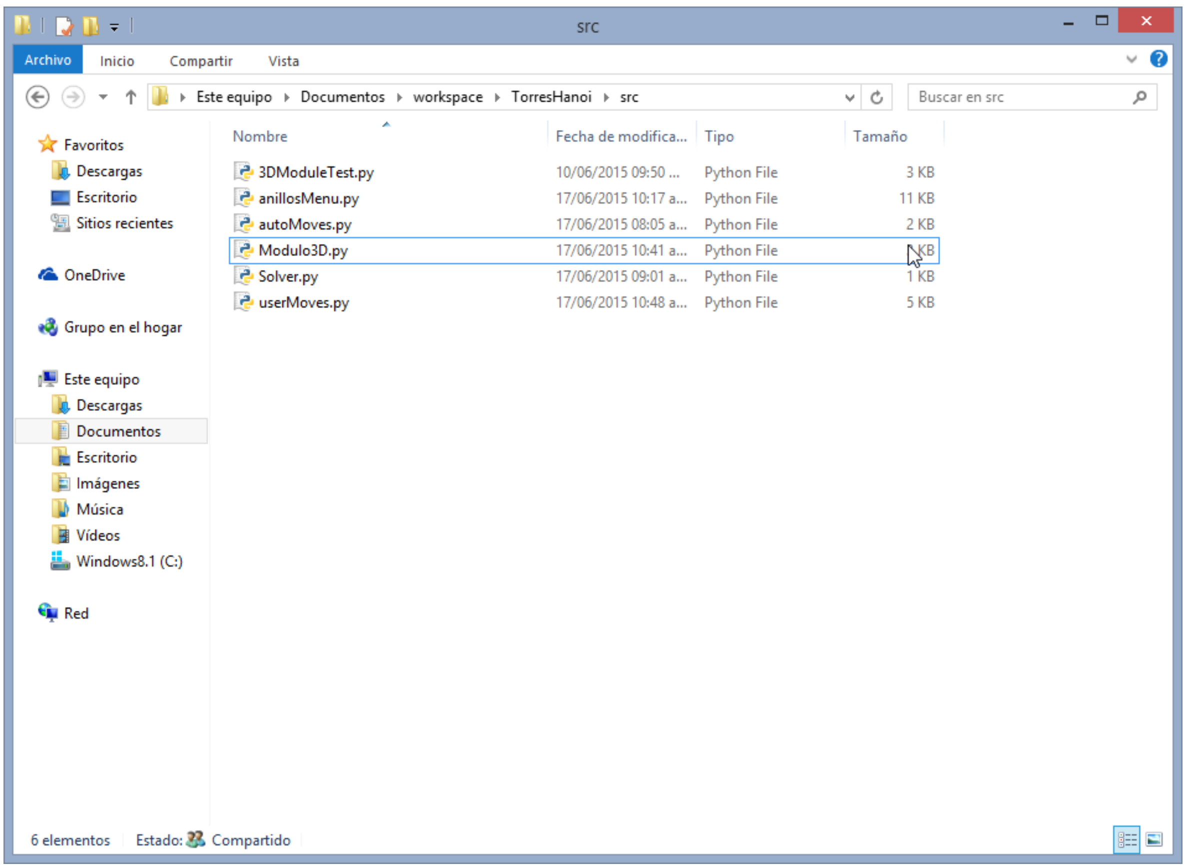
Este insteresante juego fue inventado en 1883 por el matemático francés Édouard Lucas, el cual incluso creo leyendas para obtener atención del público y así ganarse la vida. Se cuenta que en un templo de Benarés se encontraba una cúpula que señalaba el centro del mundo. Allí estaban tres agujas de diamante. Un dia un rey mandó a poner 64 discos de oro ordenados por tamaño. Los sacerdotes del templo intentaron mover los discos entre las agujas, según las reglas que se presentaron. Hoy no existe tal templo, pero el juego aún perdura en el tiempo.

Otra leyenda cuenta que Dios, al crear el mundo, colocó tres varillas de diamante con 64 discos en la primera. También creó un monasterio, en donde los monjes tenían la tarea de resolver esta Torre de Hanói divina. El día que estos monjes consiguieran terminar el juego, el mundo acabaría. No obstante, esta leyenda resultó ser un invento publicitario del creador del juego, el matemático Éduard Lucas. La mínima cantidad de movimientos para resolver este problema es de 2^{64} - 1; si los monjes hicieran un movimiento por segundo, sin equivocarse, osea 18,446,744,073,709,551,616 movimientos, los 64 discos estarían en la tercera varilla en algo menos de 585 mil millones de años. (Como comparación para ver la magnitud de esta cifra, la Tierra tiene unos 5 mil millones de años, y el Universo, unos 14 mil millones de años de antigüedad, solo una pequeña fracción de esa cifra.)

Manual de Usuario

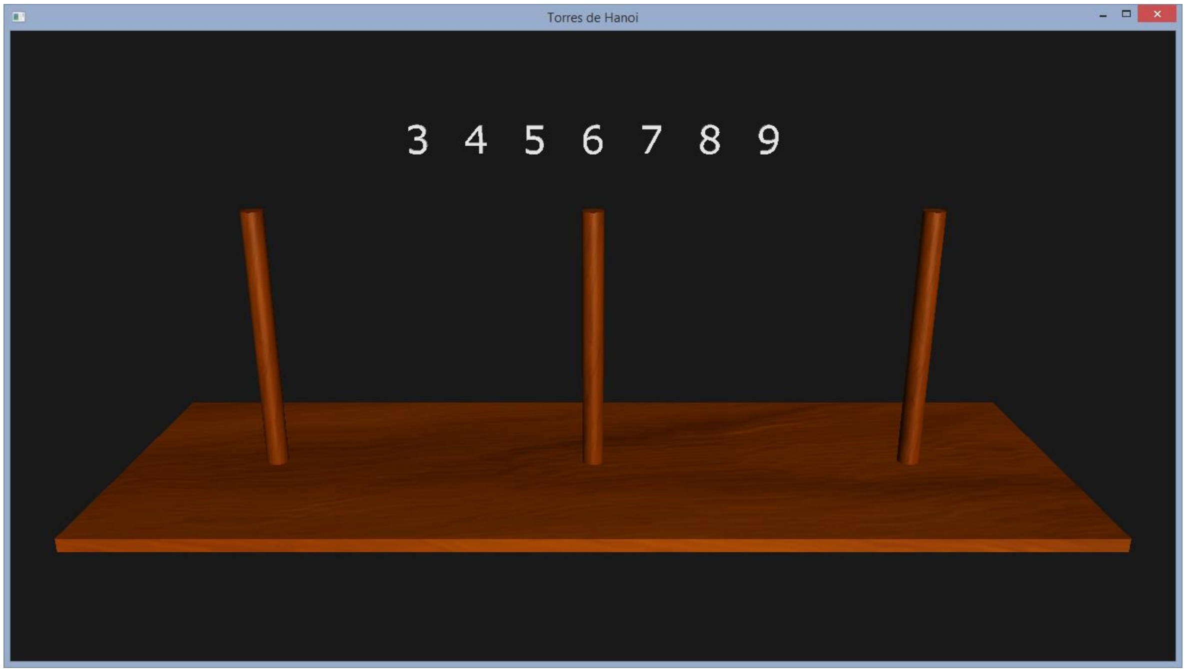
Este juego se logró realizar gracias a las librerias de VPython. Para poder correr correctamente es juego, se recomienda la instalación de estas librería en su versión mas reciente hasta el momento, VPython6. No obstante, no habrá incompatibilidad si se instala una mas nueva. Se puede econtrar en la versión mas reciente en http://vpython.org/

Teniendo el paso anterior completamente listo, ya podemos proceder a ejecutar el juego elaborado en Python 2.7. Para ello vamos a descomprimir el archivo Hanoi.zip en un directorio que nos convenga. Podemos descomprimirlo en el Escritorio o en el mismo directorio que se encuentra el archivo .zip. Entramos a la carpeta y veremos algo así:



Nuestro archivo principal será Modulo3D.py. Desde acá podremos acceder a los menus y otros elementos importantes del juego. Por ser este nuestro archivo principal, este se encargará de declarar los demas modulos para su posterior uso.

Cuando le ejecutemos veremos algo como los siguiente:

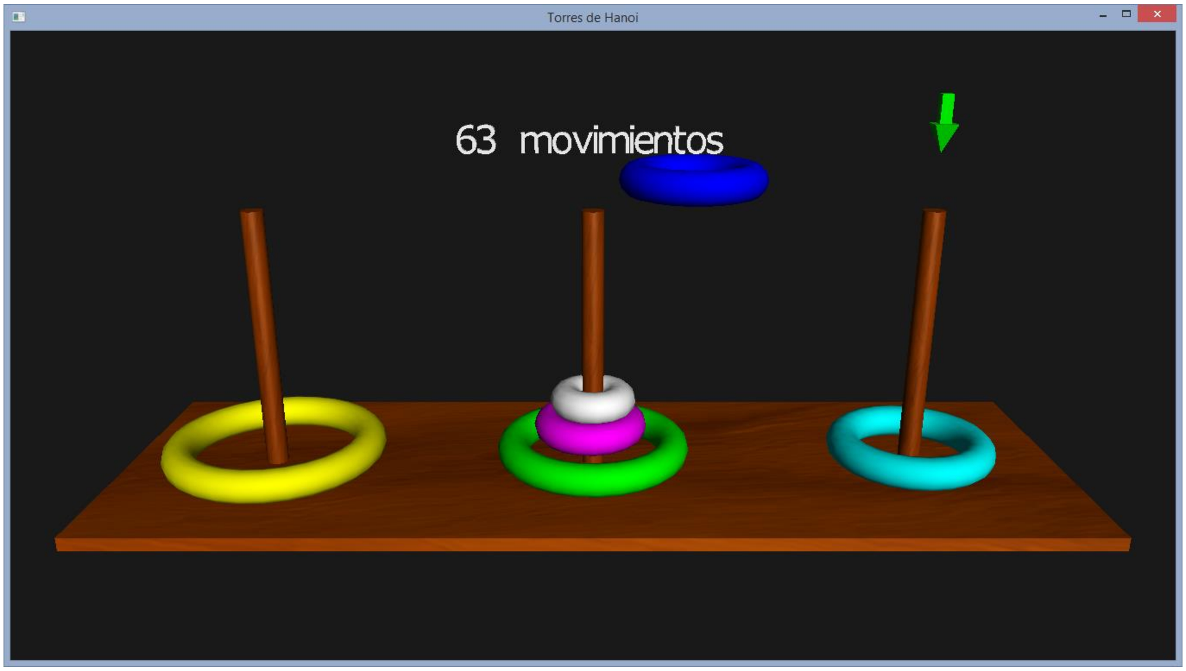


Desde esta ventana podremos escoger la cantidad de anillos que deseemos. Aunque el juego se pueden resolver con 2 anillos en 3 movimientos, se consideró que sería un opción sumamente inutil, por lo que se decidió limitarlo a 3 como minimo.

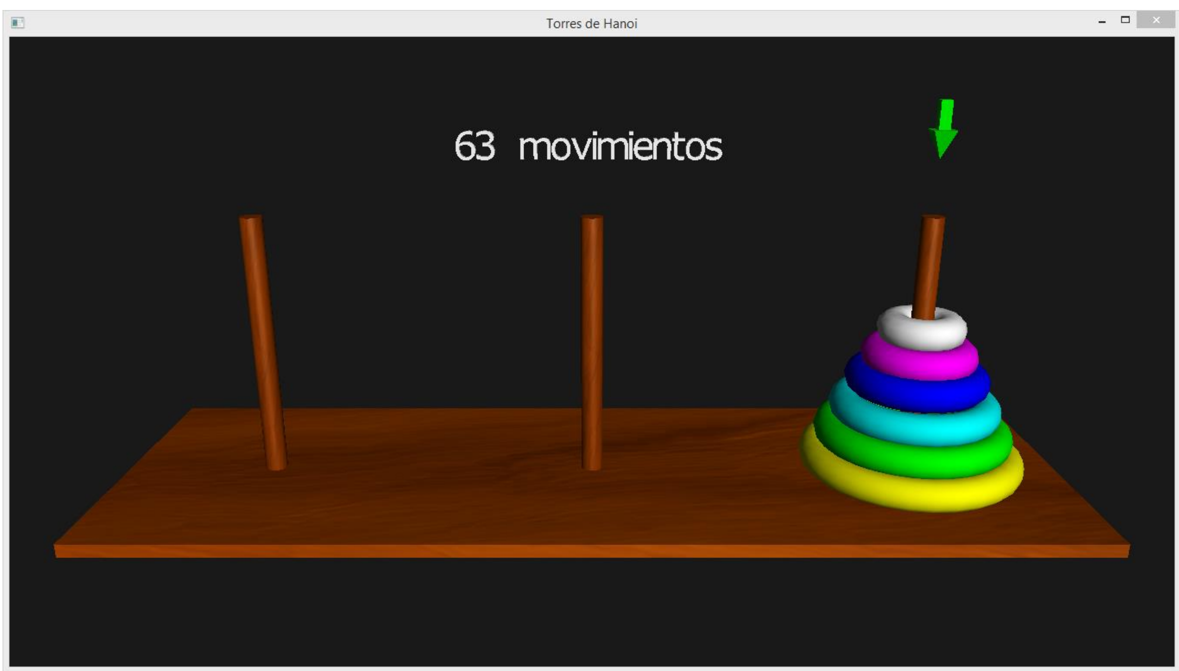
Cuando escogamos nuestros anillos veremos algo como lo siguiente:

Ya podemos ver los anillos, ademas, tambien se observan dos menus y una flecha.

Esta flecha indica a donde deben ir colocados los anillos para resolver el juego. Para continuar, presionaremos entre los dos menus que continuan. Iniciaremos con “Auto Resolver”, el cual tiene aplicada la recursión.



Presionando este botón podemos ver como nuestros anillos comienzan a mover hasta lograr colocar todos al final.



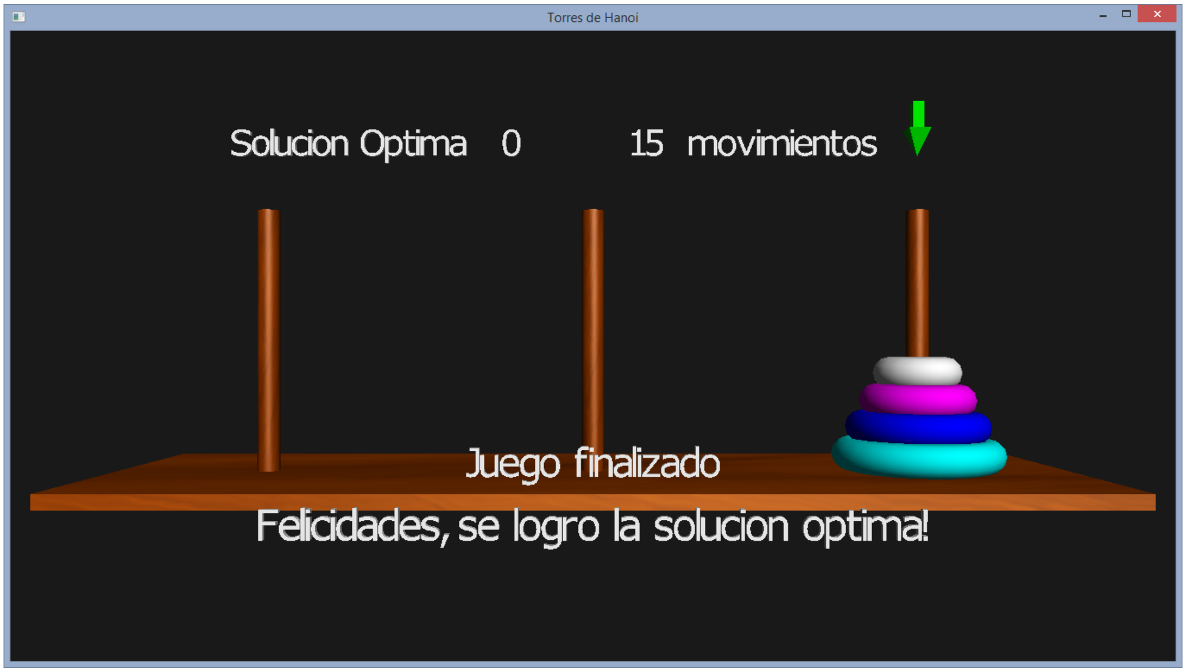
También se muestran los movimientos optimos, dada por la formula antes mencionadad.

Si en vez de “Auto Resolver” presionaramos la opción “Empezar” veremos otra pantalla que donde se nos indican dos valores: la “Solución Óptima” y los “movimientos” realizados.

Entrado a este modulo podremos tomar cualquier anillo y poverlo donde queramos. Recodemos que nuestro objetivo es mover todos los anillos y donde la flecha indica y en orden descedente, primero el mas grande, luego el mas pequeño.

Con forme vallamos haciendo moviemientos, los valores de ambas casillas cambiaran, hasta que terminemos. La “Solución Óptima” se van a ir restando con forme se avance en el juego, mientras que “movimientos” se incrementa.

Cuando el juego finalize, mostrará un pantalla de felicitaciones:



Conclusiones

Durante esta tarea se aprendió el funcionamiento de las Torres de Hanoi, además promovió la investigación de métodos para programar recursivamente la solución óptima. Usar interfaz gráfica tridimensional para simular cada uno de ellos nos condujo a la investigación de código, lo cual ayudará no solo para esta tarea o este curso, sino también para futuras investigaciones que se deban realizar acerca de estos temas en incluso para la simulación de cuerpos o particulas.

Logros

Entre los logros que se realizaron están:

1. Logran generar una interfaz gráfica amigable para simular las Torres de Hanói.
2. Importar librerías de cada interfaz gráfica 3D.
3. Generar un código fuente a base recursiva para solucionar las Torres de Hanói.
4. Se logra cumplir las espectativas del juego.
5. Se procura generar un menú fácil de entender para un usuario básico.