

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Introducción a la Programación y Computación 1

Catedrático: Ing. Moisés Eduardo Velásquez Oliva

Práctica #2

Manual Técnico

Nombre: Javier Alexander Girón Donis

Registro Académico: 202000510

Instructor: William Adolfo Corado Cabrera

Sección de Laboratorio: C

Fecha de Realización: 05/10/2021

Fecha de Entrega: 07/10/2021

Índice

Objetivos y alcances del sistema	1
Objetivos generales:	1
Objetivos específicos	1
Especificación Técnica	2
Requisitos de Hardware.....	2
Requisitos de software	2
Lógica del programa	4
Paquetes	4
Clases	5
Variables globales	6

Objetivos y alcances del sistema

Objetivos generales:

1. Crear un juego inspirado en las torres de Hanoi.
2. El manejo y comprensión de una clase recursiva.
3. Comprender y practicar el manejo de hilos en JAVA.

Objetivos específicos:

1. Elaborar funciones específicas donde se apliquen tanto los hilos como un método recursivo el cual dará animación a cada uno de los componentes donde sea solicitado.
2. Comprender las funciones que puede llegar a tener la recursividad en un programa elaborado en el código de programación JAVA.
3. Poner en práctica todos los conocimientos previamente adquiridos sobre POO para poder aplicarlos en los objetos que se les debe crear una función en específico.

El juego de las torres de Hanoi está elaborado en el código de programación JAVA está dirigido un público menor para poder desarrollar un pensamiento lógico desde una temprana edad, sin embargo, este en específico por las diferentes funcionalidades extra que contiene como el tiempo y los potenciadores lo vuelve un juego dirigido a todo tipo de público que desee aprender sobre recursividad e hilos.

Especificación Técnica

Requisitos de Hardware

- **Procesador:** Intel Core i5 (Intel Core i5 o equivalente)
- **RAM:** 2 GB, 4 GB (2 GB (32-bit), 4 GB (64-bit))
- **Disco duro:** 1.5 GB HDD
- **Tarjeta gráfica:** No necesario.
- **Resolución de la pantalla:** 1024 x 728

Requisitos de software

- **Sistemas Operativos:**
 - **Windows:** Windows 10, Windows 7, Windows XP, Windows Vista (Windows XP Professional SP3/Vista SP1/Windows 7 Professional).
 - **Mac:**
 - Mac con Intel que ejecuta Mac OS X 10.8.3+, 10.9+
 - Privilegios de administrador para la instalación
 - Explorador de 64 bits
 - **Linux:**
 - Oracle Linux 5.5+¹
 - Oracle Linux 6.x (32 bits), 6.x (64 bits)²
 - Oracle Linux 7.x (64 bits)² (8u20 y superiores)
 - Red Hat Enterprise Linux 5.5+¹ 6.x (32 bits), 6.x (64 bits)²
 - Red Hat Enterprise Linux 7.x (64 bits)² (8u20 y superiores)
 - Suse Linux Enterprise Server 10 SP2+, 11.x
 - Suse Linux Enterprise Server 12.x (64 bits)² (8u31 y superiores)
 - Ubuntu Linux 12.04 LTS, 13.x
 - Ubuntu Linux 14.x (8u25 y superiores)
 - Ubuntu Linux 15.04 (8u45 y superiores)
 - Ubuntu Linux 15.10 (8u65 y superiores)

- Exploradores: Firefox
- **Java SE Development Kit 8u301:** En su versión mas actual, se adjunta un enlace de descarga [Java Downloads | Oracle](#)
- **NetBeans 8.2:** Para poder revisar correctamente el código es necesario el uso de un IDE de programación el cual se recomienda NetBeans en su versión 8.2 al ser elaborado en este, se adjunta un enlace de descarga seguro [Apache NetBeans Releases](#)

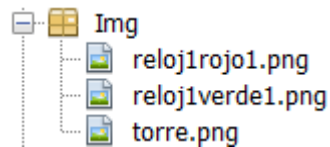
Nota: En caso no se cuente con ninguna experiencia para poder instalar ninguno de los siguientes requerimientos, como lo sería NetBeans y el JDK se adjunta el siguiente enlace de un tutorial de instalación.

(25) Descarga e Instala NetBeans IDE 8.2 y Java Development Kit para Windows 7/8/8.1/10 - YouTube

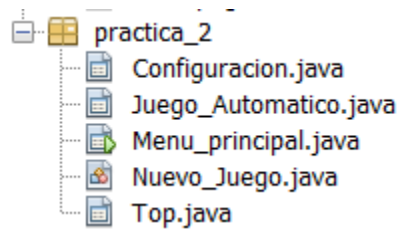
Lógica del programa

Paquetes:

- **Img:** En este paquete se almacena todo el contenido multimedia utilizado en el programa, para ser específicos la torre en formato png, el potenciador verde y el potenciador rojo.

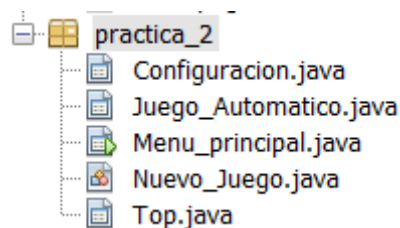


- **Practica_2:** En dicho paquete se almacenan todos los menús utilizados para el funcionamiento del programa, de igual manera contiene una clase donde se crea el menú de “nuevo juego” sin draw and drop.



Clases:

- **Configuración:** Perteneciendo al paquete practica_2, cumple con la función de desplegar el menú de configuración del juego, de igual manera contiene en forma de JFrame todos sus contenidos.
- **Juego_Automatico:** Este JFrame contiene el menú de juego automático, este cumple la finalidad de mostrar los pasos que deben seguirse para poder completar el juego. Este obedece un algoritmo aplicado con recursividad donde tomara los valores heredados por la clase Menu_principal donde se sabrá el número de discos que este utilizara (en caso de no contener una herencia se usara por defecto el valor de 3).
- **Menu_principal:** Es la clase encargada de iniciar el juego, esta contiene todo el menú principal del juego, así como distribuir los datos heredados de la clase Configuraciones a las clases Nuevo_Juego y Juego_Automatico. Esta clase contiene el método public static void main(String args[]).
- **Nuevo_Juego:** La clase Nuevo_Juego es de las más complejas del juego ya que es la que contiene la mayor interacción con el jugador, hereda de la clase Menu_Principal para obtener cuantos discos se desean utilizar y el tiempo que se utilizara para la partida. Contiene un hilo que ejecutara la cuenta regresiva y a la vez dará el movimiento de los potenciadores los cuales cambiaran cada segundo por medio de ciclos if. Por último, contiene múltiples botones donde recibirán los valores del disco que se desea mover heredando al método derecha_1 donde se comparará el disco si el movimiento es legal o no conforme a las reglas establecidas.
- **Top:** La clase top contiene 5 JTable que usaran serialización de datos para almacenar el top 5 de jugadores que logren completar el juego en la menor cantidad de tiempo posible, ya que por cada disco la dificultad aumenta se crearon 5 tablas en las cuales generara el top dependiendo del número de discos con los que se jugó la partida.



Variables globales:

- **Int reloj:** La variable reloj es una de las variables más declaradas en las distintas clases ya que será la que almacene el tiempo inicial de la partida, será heredada de configuración a Menu_Principal y por ultimo a Nuevo_Juego, de igual manera se declara en las otras clases ya que por defecto debe contener que se ejecute con el valor de 120 segundos.
- **Int discos/disc:** Nuevamente mencionamos otra variable de tipo int, cumple prácticamente la misma función que la variable reloj, pero esta hereda el valor de cuantos discos se utilizaran en la partida y se hereda también a Juego_Automatico la cual la usara para saber qué algoritmo aplicar. Como valor por defecto será 3 discos.
- **Object [][] clasifiacion:** Teniendo el primer array mencionado, clasificación almacena la serialización de cada una de las clasificaciones, así mismo cumple la función tanto de guardar como cargar sobre él.
- **String clasifiacion_tablas:** Cuando se menciona una variable de tipo String no es una variable que sea tan fundamental en el programa, sin embargo, esta cumple un papel importante ya que es la que dependiendo del número de discos que se utilice tomara el nombre del archivo.dat de la clasificación que corresponde al número de discos utilizada.
- **Int [] torre:** Nuevamente tenemos un array pero este se divide en torre1, torre2 y torre3 las cuales almacenan un los valores de cada disco que se mueve, para poder comparar que los valores iniciales de torre1 sean los mismos de torre3 y esto causa que el juego termine, en cada botón se implementa la comparativa para que el juego este evaluando en cada movimiento.
- **JPanel pieza:** Para la creación de los discos se decidió que se harían por medio de paneles ya son fáciles de manipular y posicionar dentro de un panel mayor, cada disco está declarado como pieza y el número de pieza que corresponde como por ejemplo pieza1 que corresponde a la pieza número 1.