TETRIS

PELLERITI MARIA
PALTA PABLO

ÍNDICE

Resumen ---------- 3

Introducción Objetivos ---------- 3

Palabras clave ---------- 3

Marco teórico ---------- 4

Método y desarrollo ---------- 6

Ejecución----------7

Resultados ---------- 7

Conclusiones ---------- 7

Bibliografía ---------- 8

TETRIS

Resumen

Este proyecto consiste en el desarrollo de un juego de tetris con una interfaz gráfica agradable al usuario utilizando el lenguaje de programación en java. Además de intentar utilizar la máquina virtual de java para emularlo en múltiples sistemas operativos

Introducción

La finalidad del presente proyecto es diseñar y montar el juego Tetris.

Tetris es un juego diseñado por Alexey Pajitnov en 1984, en la Unión Soviética. Debido a sus gráficos simples y su alto nivel de abstracción, está disponible para virtualmente cualquier plataforma y sistema operativo que ha surgido desde entonces. Por lo general es un gran juego para comenzar a aprender el desarrollo de videojuegos, porque contiene todos los elementos básicos que puede tener un videojuego, simplificados hasta su esencia.

Objetivo General

Desarrollar en código el juego Tetris para ser ejecutado en Linux.

Objetivos Específicos

Elaborar la programación correspondiente, para el funcionamiento del tetris.

Generar un programa fácil de interpretar.

Investigar sobre la lógica pertinente al juego.

Implementar los conocimientos adquiridos durante la carrera.

Crear una aplicación capaz de correr en múltiples sistemas operativos.

Mejorar las habilidades de programación y desarrollo de software.

Aprender a utilizar Java como lenguaje de programación.

Palabras Clave

Tetris, videojuego, Java.

Marco teórico

Lenguaje de programación:

Un lenguaje de programación es un idioma artificial diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras.

Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila y se mantiene el código fuente de un programa informático se le llama programación.

También la palabra programación se define como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos, a través de los siguientes pasos:

El desarrollo lógico del programa para resolver un problema en particular. Escritura de la lógica del programa empleando un lenguaje de programación específico (codificación del programa). Ensamblaje o compilación del programa hasta convertirlo en lenguaje de máquina. Prueba y depuración del programa. Desarrollo de la documentación.

JAVA:

Java es un lenguaje de programación y la primera plataforma informática creada por Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Java se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como dispositivos móviles y aparatos de televisión.

Herencia:

Se puede construir una clase a partir de otra mediante el mecanismo de la herencia. Para indicar que una clase deriva de otra se utiliza la palabra extends, como por ejemplo: class CirculoGrafico extends Circulo {...}

Cuando una clase deriva de otra, hereda todas sus variables y métodos. Estas funciones y variables miembro pueden ser redefinidas (overridden) en la clase derivada, que puede también definir o añadir nuevas variables y métodos. En cierta forma es como si la sub-clase (la clase derivada) "contuviera" un objeto de la super-clase; en realidad lo "amplía" con nuevas variables y métodos.

Java permite múltiples niveles de herencia, pero no permite que una clase derive de varias (no es posible la herencia múltiple). Se pueden crear tantas clases derivadas de una misma clase como se quiera.

Todas las clases de Java creadas por el programador tienen una super-clase. Cuando no se indica explícitamente una super-clase con la palabra extends, la clase deriva de

java.lang.Object, que es la clase raíz de toda la jerarquía de clases de Java. Como consecuencia, todas las clases tienen algunos métodos que han heredado de Object.

Encapsulamiento:

El encapsulamiento (o encapsulación), u ocultación de la información, es el proceso de ocultar todos los detalles de un objeto que no contribuyen a sus características esenciales. Esencialmente, significa que aquello que está en el interior de la clase está oculto; sólo las interfaces externas se pueden ver por otros objetos. El usuario de un objeto nunca necesitará conocer el interior de la clase.

Una de las ventajas principales del encapsulamiento es que proporciona al programador libertad en la implementación de los detalles de un sistema. La única restricción que tendrá el programador es mantener la interfaz abstracta que ven los usuarios externos.

En realidad, es el mecanismo que enlaza juntos código y los datos que los manipulan y los mantienen seguros de interferencias externas y el mal uso.

En Java, los fundamentos de la encapsulación están en la clase. Una clase define la estructura y el comportamiento (datos y código) que se compartirá por un conjunto de objetos. Por esta razón, los objetos se conocen como a veces como instancias de una clase. Por consiguiente, una clase es una construcción lógica; un objeto tiene una realidad física.

Polimorfismo

El polimorfismo es un concepto más difícil de entender que el encapsulamiento o la herencia. Básicamente, significa que los objetos que pertenecen a la misma ramificación de una jerarquía, cuando se envía el mismo mensaje (es decir, cuando se le indica que realice lo mismo), pueden manifestar ese comportamiento de modo diferente.

Para entender cómo el polimorfismo se aplica a un contexto de aplicación de negocio, regrese al ejemplo de Person. ¿Recuerda indicarle a Person que formatee sus atributos en una String? El polimorfismo hace que sea posible para Person representar sus atributos en una variedad de formas, dependiendo del tipo de Person que sea.

El polimorfismo es uno de los conceptos más complejos con los que se encontrará en OOP en la plataforma Java y no dentro del ámbito de un tutorial introductorio. Vea Recursos si quiere aprender más acerca del polimorfismo.

Método y desarrollo

Planificación:

Elegir el juego a desarrollar.

Seleccionar el lenguaje a utilizar

Seleccionar el IDE a utilizar

Definir las clases del juego

Distribuir el desarrollo de las mismas entre los integrantes del grupo

Unir las partes para lograr la correcta ejecución

Solucionar problemas que pueden presentarse

Elaborar el informe sobre el desarrollo del juego

Crear el ejecutable del juego para correrlo en Linux.

Reglas de inicio

Al iniciar una partida, se dispone de un espacio cuadriculado, de 10 bloques de ancho por 14 de alto. Una pieza de las 7 disponibles se escoge, y se coloca en el tope del espacio, centrado horizontalmente.

Reglas del progreso

-piezas disponibles

El jugador cuenta con 7 tipos de piezas, cada una con un color característico: el bloque cuadrado (2x2 bloques), la S (3x2 bloques), la Z (3x2 bloques), la L (3x2 bloques), la L invertida (3x2 bloques), la T (3x2 bloques), y la línea (4x1 bloques).

-movimiento de los bloques

El jugador puede controlar un solo bloque a la vez. El bloque puede ser rotado en sentido de las agujas del reloj o al contrario libremente y en cualquier momento por el jugador. El bloque cada cierto tiempo baja un espacio.

Cuando el bloque toca el fondo del espacio, o toca un bloque ya fijo, pasado cierto tiempo el bloque se fijará como parte del espacio, se escogerá una nueva pieza al azar y se colocará al tope de la pantalla, como está especificado en las reglas de inicio.

El bloque puede rotarse libremente mientras haya espacio para hacerlo. Si no hay espacio para rotarse, el bloque se mantiene en su orientación previa. Cuando el bloque entra en contacto con el fondo de la pantalla o algún bloque ya fijo, el bloque puede rotar todas las veces que el jugador quiera antes de fijarse.

Después de fijado un bloque, se debe comprobar en cada fila del espacio si todos los espacios de esa fila están cubiertos por un bloque. En tal caso, se desparecen todas las

líneas que hayan sido completadas por ese bloque, y se bajan todas las filas superiores un número de espacios iguales a la cantidad de filas desaparecidas.

Reglas de resolución

Si al surgir un bloque nuevo en el tope-centro horizontal del espacio, ya existe un bloque fijado, se da por terminada la partida.

Interfaz

En todo momento, el juego debe mostrar, además del espacio de juego, el puntaje de la partida actual (el número de líneas completadas).

Ejecución

Para ejecutar es necesario contar con una JVM

En caso de no poseerla, se pude instalar bajo el siguiente comando en el terminal:

sudo apt-get install default-jre

Una vez instalada, para ejecutar un archivo .jar hay que seguir los siguientes pasos:

dar click derecho sobre el archivo

abrir con

abrir con otra aplicación

usar una orden personalizada

escribir la siguiente orden en el recuadro: "java -jar "

Resultados

Como resultado logramos crear un juego de tetris mediante el lenguaje de Java. En éste implementamos lógica de vectores y posicionamientos para lograr hacer funcionar el juego. Aparte de esto se obtuvo una aplicación capaz de correr en todos los sistemas operativos de Windows y de llamativos colores e interfaz fácil de utilizar al usuario.

Conclusiones

Al realizar este proyecto logramos entender más sobre la metodología que se utiliza para el diseño de software. Aquí utilizamos el lenguaje de java que es muy versátil y fácil de aprender debido a que viene derivado de una compilación de lenguajes populares como C++ y C. Gracias a la utilización de este lenguaje fuimos capaces de crear una interfaz gráfica agradable para los usuarios dado que pudimos utilizar objetos en este caso imágenes para desarrollar una aplicación llamativa. En este tipo de desarrollo de programas es importante que sea del agrado del usuario final.

Resultados Detallados

Utilizando el lenguaje Java, con el IDE NetBeans, se crearon distintas clases. En las cuales se desarrollaron los métodos para dar las siete formas utilizadas en el juego Tetris, las dimensiones del tablero, el modo de interactuar entre las formas, los límites y reglas preestablecidas, un contador de líneas destruidas y los menús de entrada y salida.

Se logró conocer más acerca del lenguaje Java y sus clases, se profundizó en el uso de la API de Java y la creación e interacción entre objetos.

A través de la investigación, se descubrió el mecanismo de funcionamiento del Tetris y la lógica incluida en este.

Una vez concluida el desarrollo del juego, se consiguió elaborar un ejecutable .jar para que pudiese ser ejecutado en el sistema operativo Linux.

Bibliografía

API de java: https://docs.oracle.com/javase/7/docs/api/

Historia del tetris: https://www.neoteo.com/la-historia-de-tetris-25-anos-de-tetris-16135/

Más sobre Java: http://www.aprendeaprogramar.com/mod/forum/view.php?id=203

Interfaces gráficas en Java, Libro de María del Soto Montalvo Herranz y Micael Gallego Carrillo.

Ejecutar .jar en Linux: https://andalinux.wordpress.com/2008/12/17/ejecutar-aplicaciones-jar-en-ubuntu-linux/