Universidad Autónoma de Baja California.

Facultad de Ciencias Químicas e Ingeniería.

Manual técnico de software para MathOperations

Integrantes: Rico Guzman Edgar Yahel, Velazquez Angulo Jose Miguel, De La Cruz Estrada Steven Nicolae.

Version 1.0

Fecha de elaboracion: Junio/2024

Materia: Taller de documentacion de software

1. Introducción

Propósito del manual

Este manual técnico está dirigido a desarrolladores, personal de soporte y cualquier perfil técnico que necesite comprender la estructura interna, instalación y mantenimiento del software "Juego Matemático". El propósito es proporcionar una guía detallada que permita implementar, mantener y extender la funcionalidad del sistema de manera eficiente.

Breve descripción del software

El producto es un juego de escritorio simple desarrollado en Java, diseñado para poner a prueba las habilidades aritméticas básicas del usuario (suma, resta y multiplicación) bajo un límite de tiempo. El juego incluye una interfaz gráfica intuitiva, gestión de puntuación por sesión y un sistema de persistencia de datos para mantener un "Top 5" de las mejores puntuaciones, fomentando la rejugabilidad.

Alcance del documento

Este documento cubre los aspectos técnicos del software, incluyendo:

- Una descripción de la arquitectura general del sistema.
- Las tecnologías, lenguajes y bibliotecas utilizadas en su desarrollo.
- Los requisitos de hardware y software necesarios para la compilación y ejecución del programa.

Este manual no cubre tutoriales de usuario final sobre cómo jugar, ni entra en detalles de diseño de la experiencia de usuario (UX).

2. Descripción general del sistema

Objetivo principal del software

Los objetivos principales del software "Juego Matemático" son:

- **Entretener y Educar:** Ofrecer un desafío rápido enfocado en operaciones matemáticas básicas.
- **Gestión de Puntuación:** Registrar la puntuación del jugador durante una sesión de juego cronometrada de 10 segundos.
- **Persistencia de Datos:** Mantener un "Top 5" de las mejores puntuaciones en un archivo local (scoreboard.txt).

- **Interfaz Gráfica:** Proporcionar una GUI simple con un menú principal, pantalla de juego y sección de créditos.

Arquitectura general

El sistema es una aplicación de escritorio monolítica que no requiere conexión a un servidor. La arquitectura sigue una separación de responsabilidades básica, dividida en tres clases principales:

1. **MenuInterfaz (Vista/Controlador Principal):** Es el punto de entrada de la aplicación (main method). Actúa como el menú principal y gestiona la navegación. Es responsable de crear la instancia de Scoreboard y lanzarla ventana del juego (JuegoMatematico).
2. **JuegoMatematico (Vista/Controlador del Juego):** Es una JFrame que contiene toda la lógica del juego, incluyendo la generación de preguntas, la verificación de respuestas y el temporizador. Al finalizar el tiempo, se comunica con la instancia de Scoreboard para guardar el resultado.
3. **Scoreboard (Modelo/Lógica de Datos):** Es una clase no visual (sin GUI) responsable de gestionar la persistencia de las puntuaciones. Carga los puntajes del archivo scoreboard.txt , añade nuevos puntajes (manteniendo solo el Top 5) y guarda la lista actualizada en el mismo archivo.

Tecnologías utilizadas

El software fue desarrollado utilizando las siguientes tecnologías y bibliotecas del ecosistema Java:

- **Lenguaje de Programación:** Java.
- **Bibliotecas de GUI (Frameworks):**
    - **Java Swing (javax.swing.*):** Utilizado para la construcción de toda la interfaz gráfica, incluyendo ventanas (JFrame), botones (JButton) y etiquetas (JLabel).
    - **Java AWT (java.awt.*):** Utilizado para la gestión de layouts (como BorderLayout y GridLayout), eventos (ActionEvent), fuentes y colores.
- **Manejo de Audio:**
    - **Java Sound API (javax.sound.sampled.*):** Usada para cargar y reproducir los archivos de sonido .wav para las respuestas correctas e incorrectas.
- **Persistencia de Datos:**
    - **Java I/O (java.io.*):** Utilizado para la persistencia de datos mediante la lectura (BufferedReader) y escritura (BufferedWriter) en el archivo de texto plano scoreboard.txt.
- **Otros:**
    - **java.util.Timer y java.util.TimerTask:** Implementan el temporizador de 10 segundos del juego.

3. Requisitos del sistema

Hardware

El software es una aplicación Java Swing ligera y no tiene requisitos de hardware específicos más allá de los necesarios para ejecutar el Entorno de Ejecución de Java.

- **CPU:** Cualquier procesador moderno
- **RAM:** Mínimo 256 MB
- **Espacio en disco:** Menos de 10 MB para los archivos de la aplicación (código compilado, imágenes y archivos de sonido).

Software

- **Sistema Operativo:** Cualquier sistema operativo de escritorio que soporte Java, incluyendo Windows, macOS o distribuciones de Linux.
- **Dependencias:**
    - **Para Ejecución (Usuario final):** Se requiere el **Java Runtime Environment (JRE)**, versión 8 o superior.
    - **Para Desarrollo (Mantenedor):** Se requiere el **Java Development Kit (JDK)**, versión 8 o superior, para compilar el código fuente (.java).
- **Bibliotecas Externas:** No se requiere ninguna biblioteca externa. Todas las dependencias (Swing, AWT, Sound, IO) son parte de la biblioteca estándar de Java (Java SE).

Red

- **Conexión:** No se requiere conexión a Internet ni a ninguna red local.
- **Puertos:** La aplicación no abre ni utiliza ningún puerto de red. Toda la operación es local y la persistencia se maneja a través del sistema de archivos local (scoreboard.txt).

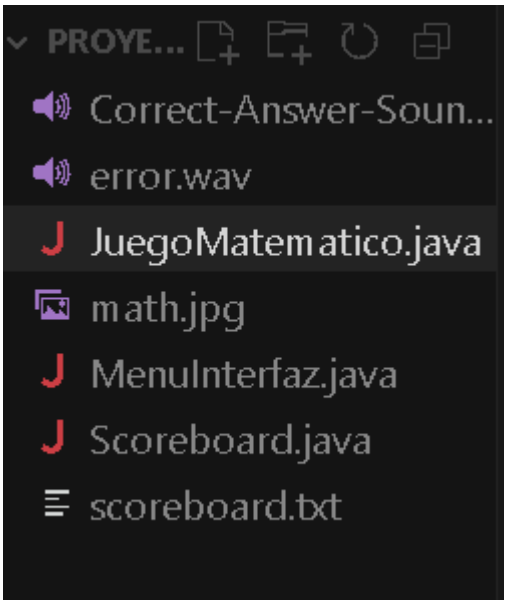
Para cubrir esos dos nuevos requisitos, necesitas agregar dos secciones principales a tu manual.

Basándome en tu proyecto, aquí tienes exactamente lo que necesitas documentar para cumplir con esos puntos.

4. Instalación y Configuración (Comandos y Capturas)

4.1. Estructura de Archivos

Para que la aplicación funcione correctamente, todos los archivos deben estar en el mismo directorio. El archivo scoreboard.txt se creará automáticamente en esta misma ubicación la primera vez que se complete una partida.



4.2. Compilación del Código

Para compilar el software, es necesario tener instalado el JDK de Java (versión 8 o superior). Abra una terminal o línea de comandos en la carpeta del proyecto y ejecute el siguiente comando:

javac *.java

4.3. Ejecución de la Aplicación
Este es el comando para que un usuario (o el desarrollador) inicie el juego.

Una vez compilado el código (o si se distribuyen los archivos .class), la aplicación se puede ejecutar desde la misma terminal usando el siguiente comando, que llama a la clase principal MenuInterfaz:

java MenuInterfaz

5. Mantenimiento y Control de Versiones

5.1. Sistema de Control de Versiones (Git)

El código fuente de este proyecto se gestiona mediante el sistema de control de versiones Git. El repositorio central está alojado en github

5.2. Obtención del Código Fuente

Para obtener la última versión del código fuente y contribuir al proyecto, primero debe clonar el repositorio. Ejecute el siguiente comando en su terminal:

git clone https://github.com/Flaky7/MathOperations.git