

Informe del Proyecto 2048

Richard Alejandro Matos Arderí
Mauricio Sunde Jiménez

Universidad de La Habana
Licenciatura en Ciencia de la Computación
Asignatura: Programación Declarativa

1 Funcionamiento del Juego

El juego 2048 es un juego de rompecabezas de deslizamiento. El objetivo del juego es mover las fichas numeradas en una cuadrícula para combinarlas y crear una ficha con el número 2048. Los movimientos posibles son hacia arriba, abajo, izquierda y derecha. Cuando dos fichas con el mismo número se tocan, se combinan en una sola ficha con la suma de sus valores.

2 Uso de la Programación Declarativa y Haskell

Este proyecto utiliza Haskell como lenguaje de programación funcional para implementar el juego 2048. Haskell es conocido por su enfoque en la programación declarativa, donde se describe qué se quiere lograr en lugar de cómo hacerlo. Esto permite un código más conciso y fácil de razonar.

2.1 Uso de Gloss para Animaciones e Interfaz Visual

Gloss es una biblioteca de Haskell que facilita la creación de gráficos y animaciones. En este proyecto, Gloss se utiliza para renderizar la cuadrícula del juego y las fichas, así como para manejar la interacción del usuario. La biblioteca proporciona una abstracción de alto nivel para trabajar con gráficos, lo que simplifica el desarrollo de la interfaz visual del juego.

3 Estructura del Proyecto

El proyecto está organizado en varios archivos y carpetas, cada uno con un papel específico:

3.1 game/

- **stack.yaml**: Archivo de configuración para Stack, incluyendo las dependencias necesarias para compilar y ejecutar el juego.

3.2 assets/

Esta carpeta almacena todos los recursos utilizados en el juego.

- **images/**: Imágenes utilizadas en el juego (por ejemplo, sprites de las fichas).
- **sounds/**: Archivos de sonido para efectos o música de fondo.
- **fonts/**: Fuentes utilizadas para mostrar texto en el juego.

3.3 app/

- **Main.hs**: Este es el punto de entrada del programa. Aquí se inicializa el juego, se configuran los bucles de eventos y comienza la ejecución.

3.4 src/

Esta carpeta contiene el código fuente del juego.

- **Game/**: Contiene la lógica principal del juego.
 - **Game.hs**: Controlador principal que gestiona el estado del juego y las interacciones entre los diferentes componentes.
 - **GameLogic.hs**: Implementa la lógica del juego, como mover y combinar fichas, así como verificar si el jugador ha ganado o perdido.
 - **Grid.hs**: Representa el tablero del juego, incluyendo su inicialización y actualizaciones.
 - **Tile.hs**: Define la estructura de una ficha en el juego, incluyendo su valor y posición.
 - **Score.hs**: Gestiona el sistema de puntuación del juego.
- **Graphics/**: Maneja la parte gráfica del juego.
 - **Graphics.hs**: Configuraciones generales relacionadas con la biblioteca Gloss.
 - **Render.hs**: Funciones para renderizar el tablero, las fichas y otros elementos visuales en la pantalla.
 - **InputHandler.hs**: Gestiona los eventos de entrada del usuario, como las pulsaciones de teclas.
- **Config/**: Contiene configuraciones generales del juego.
 - **Config.hs**: Define constantes y configuraciones como el tamaño del tablero, colores, etc.

3.5 tests/

Esta carpeta contiene pruebas unitarias y de integración para asegurar la calidad del código.

- **GameTests.hs**: Pruebas relacionadas con la lógica del juego.
- **GraphicsTests.hs**: Pruebas relacionadas con la parte gráfica.