salvar tablero()

Copia el tablero actual sobre la matriz tablero_original

tableroIluminado()

Informa si todas las casillas del tablero están iluminadas

desordenar(int dificultad)

Desordena el tablero realizando una serie de jugadas sobre posiciones aleatorias. Si antes de ejecutar este método el tablero estaba todo iluminado, esto garantiza que se vaya a poder resolver después.

El número de jugadas realizadas se mantiene constante

reiniciar()

Reinicia el tablero para que las casillas tengan los valores almacenados en tablero_original.

Deja la cantidad de jugadas con valor 0.

PanelSuperior InterfazLightsOut Esta clase es reponsable de recibir y hacer visibles PanelIza Esta clase es la responsable de armar el Jframe los datos de juego que desea el usuario, los cuales Panel izquierdo es responsable de mostrar, recibir con todos sus componentes, para ello añade todos son el nivel de dificultad y el tamaño del tablero y realizar aspectos relazionados con la experiencia los paneles al frame y añade el tablero de juego de juego del usuario, y hacer modificaciones sobre que es PruebaTablero. el tablero, como reiniciar el tablero, o crear otro tablero con diferentes datos, a su vez, es el encargado de mostrar la lista de top 10 <u>InterfazLightsOut</u> <u>PanelSuperior</u> · size: JLabel ventana: |Panel PanelIzg comboBox: JComboBox<String> comboBox juegoIniciado: boolean dificultad: JLabel + int sizeTablero: int - agregarTablero(): void facil: JRadioButton - Tablero tablero + eliminarTablero(): void medio: |RadioButton - tableroActual: boolean[][] dificil: |RadioButton + actionPerformed(ActionEvent e): void sizeTablero: int= "3" + DialogTop10(): void dificultadTablero: int **Tablero** + actionPerformed(ActionEvent e): void + itemStateChanged(ItemEvent e): void tablero: boolean[][] tablero_original: boolean[][] - jugadas: int reiniciar (): void RegistroTop10 **PruebaTablero** - salvar tablero(): void - nombre: String - desordenar(int dificultad): void size: int - puntos: int + darJugadas(): int int width: int + darPuntos(): int + darTablero(): boolean[][] tableroActual: boolean[][] **PanelInferior** + darNombre(): String cantidades: int[][] + jugar(): void + toString(): String + tableroIluminado(): boolean ultima_columna: int + int sizeTablero: int + calcularPuntaje(): int ultima fila: int Tablero tablero tableroActual: boolean[][] paintComponent(Graphics h): void - jugadas: JLabel optionPaneWin(): void jugar(int fila, int columna) actualizarJugadas(int contador): void - convertirCoordenadasACasilla(int x, int y): int[] Cambia el estado de una casilla y el Top10 estado de las casillas que la rodean. Las registros: PriorityQueue<RegistroTop10> iluminadas se apagan y viceversa. peorPuntaie: int PruebaTablero Incrementa en 1 el número de jugadas Se espera que prueba tablero realice las actividades esTop10(int puntaje): boolean realizadas hasta el momento. + agregarRegistro(String nombre, int puntaje): void correspodientes a el funcionamiento del juego, como + darRegistros(): Collection<RegistroTop10> cambiar los colores del tablero al reconocer una + salvarRecords(File archivo): void interaccion de click, identificar si el usuario gano el + cargarRecords(File archivo): void juego y realizar la solicitud de su nombre

esTop10(int puntaje)

para entrar al top10

Indica si el puntaje indicado es suficiente

salvarRecords(File archivo)

cargarRecords(File archivo)

Persistencia

PanelInferior

inicializandolas en 0

Se espera que panel inferior muestre la informacion

del juego actual, actualizando las jugadas e