

Taller 4 - Diseño y Programación Orientada a Objetos

ACLARACIONES : El juego original y según lo que averigüe era que todas las luces estén apagadas pero en el documento dice que el jugador gana cuando todas las luces estén prendidas entonces hice que el jugador ganará cuando todas las luces estuvieran prendidas. A continuación evidencia de lo dicho anteriormente:

El juego Lights Out se basa en un tablero con luces que pueden estar prendidas o no. Cuando el jugador selecciona una luz, esa luz cambia de estado: si estaba apagada se enciende y si estaba prendida se apaga. Lo mismo ocurre con las casillas adyacentes a la seleccionada. El objetivo del juego es, a partir de una configuración inicial con luces encendidas y apagadas, llegar en la menor cantidad de jugadas a una configuración donde todas las luces estén encendidas ¹.

El botón **Reset**, reinicia al tablero inicial antes de cualquier jugada que haya hecho el usuario. El botón de **New** genera una nueva distribución de luces. El botón **Change Player** cambia el nombre del jugador generando uno aleatoriamente, el usuario no puede escoger un nombre(no estaba especificado en el documento), se hizo así por practicidad del programa y se intentó imitar juegos de antaño donde el nombre era aleatorio; además el nombre solo es registrado si está en el top10.

1. Organización de Componentes y Layouts:

- **JPanel panel** → Utiliza un '*BorderLayout*' el cual ayuda a dividir el panel en 5 regiones (NORTE, SUR, ESTE, OESTE, CENTRO).
- **SectionUp** → Se ubicaría en el NORTE del panel y podría contener opciones como dificultad y tamaño del tablero. Utiliza un '*FlowLayout*' para organizar botones y menús desplegables.
- **SectionDown** → Localizado en SUR del panel, este podría mostrar información del jugador y el número de juegos, usando un '*BoxLayout*' para un alineamiento horizontal.
- **SectionRight** → Colocado en el ESTE del panel, maneja la lista de records y botones(ejemplo: reiniciar el juego). Un '*BoxLayout*' en el eje Y sería adecuado aquí.
-
- **PBoard** → Este es el tablero del juego y se sitúa en la región CENTRO. No necesita un layout ya que se personaliza su '*paintComponent*' para dibujar las celdas.

2. Clases y Responsabilidades:

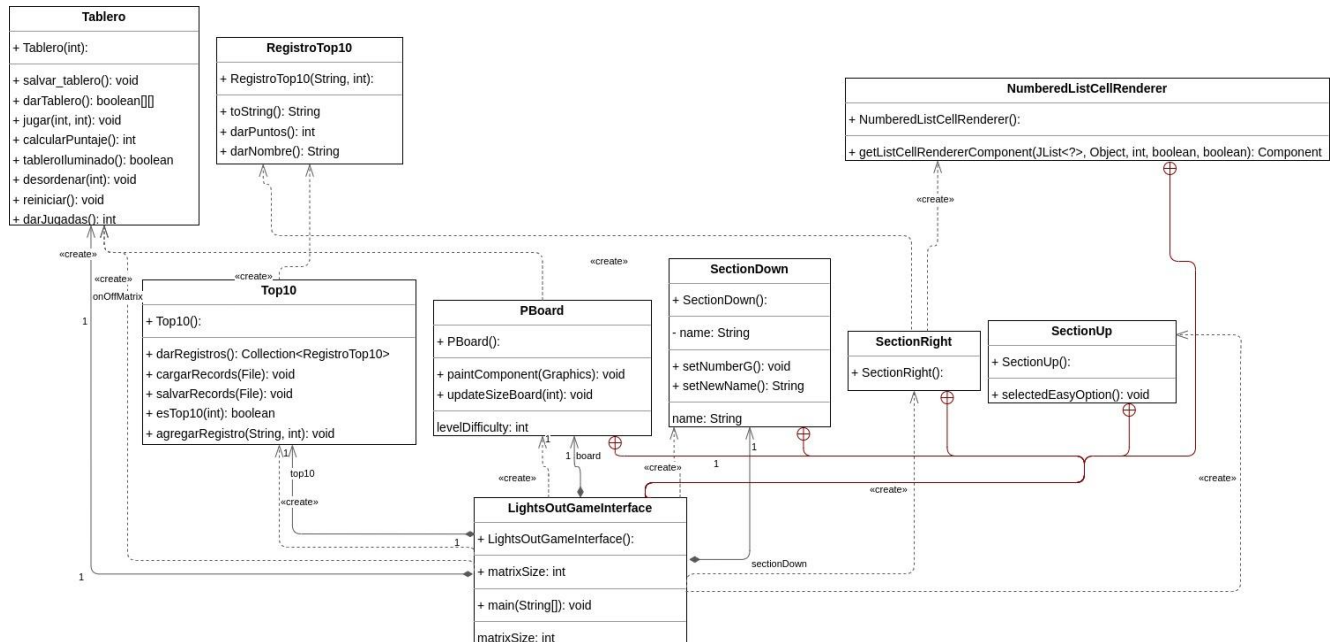
- **'LightsOutGameInterface'**: Maneja la creación y organización de los componentes principales (paneles y tablero).
- **'PBoard'**: Se encarga de dibujar el tablero y manejar eventos de clic del ratón.
- **'SectionUp'**: Gestiona la configuración del juego, como el tamaño y la dificultad del tablero.
- **'SectionDown'**: Muestra el nombre del jugador y el contador de jugadas por partida.
- **'SectionRight'**: Controla la visualización de los records y las interacciones para cambiar de jugador o reiniciar el juego.

3. Relación entre la interfaz y la lógica:

- **'Tablero'**: Es la clase del modelo que gestiona la lógica del tablero del juego (estado de las celdas, jugadas, etc.).
- **'Top10'**: Maneja la lógica tras el registro y formación del top 10, así como calcula el puntaje de acuerdo a parámetros dados.
- **'RegistroTop10'**: Almacena y gestiona la información de los mejores puntajes.
- La interfaz gráfica (**'LightsOutGameInterface'**) instancia y utiliza estas clases del modelo para reflejar el estado del juego y responder a las interacciones del usuario.

4. Interacción con el Usuario:

- Mouse event en **'PBoard'**: Detecta click en el tablero y cambia el estado de las celdas.
- Botones en **'SectionUp'**: Cambian la dificultad y el tamaño del tablero.
- Botones en **'SectionRight'**: Incluyen acciones como reiniciar el juego o cambiar de jugador.
- Botones en **'SectionDown'**: Permite el cambio del nombre del jugador de forma aleatoria.



****La imagen en jpg está adjunta en el repositorio.****