# Explicación de la Clase Pantalla

La clase `Pantalla` se encarga de gestionar la ventana principal del juego. Utiliza `CardLayout` para cambiar entre diferentes pantallas, como el menú y el juego, y `JLayeredPane` para organizar las capas del escenario, colisiones y personaje.

## 1. Definición de la Clase

Ubicación: `juegoprog.sistema.Pantalla`

Extiende: `JFrame` (es la ventana principal del juego)

## 2. Explicación Detallada del Código

### Atributos

```java  
private CardLayout cardLayout;  
private JPanel contenedorPrincipal;  
private JLayeredPane capaJuego;  
private Movimiento movimiento;  
private EscenarioDistritoSombrio escenario;  
private ColisionesPanel colisiones;  
```

✔️ `cardLayout`: Permite cambiar entre el menú y la pantalla de juego.

✔️ `contenedorPrincipal`: Panel principal que contiene todas las pantallas.

✔️ `capaJuego`: Permite manejar diferentes capas en la pantalla de juego.

✔️ `escenario`, `colisiones`, `movimiento`: Representan el fondo, las colisiones y el personaje.

### Constructor

```java  
public Pantalla() { ... }  
```

✔️ Configura la ventana (`JFrame`) con título, tamaño y comportamiento de cierre.

✔️ Usa `CardLayout` para gestionar la pantalla del menú y la del juego.

✔️ Agrega un `JLayeredPane` con varias capas: escenario, colisiones y movimiento.

### Uso de CardLayout

✔️ `CardLayout` permite cambiar entre el menú y la pantalla del juego dinámicamente.

✔️ `contenedorPrincipal.add(menu, "MENU");` agrega el menú.

✔️ `contenedorPrincipal.add(capaJuego, "JUEGO");` agrega la pantalla del juego.

### Uso de JLayeredPane

✔️ Permite organizar elementos en capas para dibujar correctamente el juego.

✔️ Se agregan las capas en este orden:

1️⃣ `DEFAULT\_LAYER`: Escenario de fondo.

2️⃣ `PALETTE\_LAYER`: Imagen de colisiones.

3️⃣ `MODAL\_LAYER`: Movimiento del personaje.

### Método cambiarPantalla()

```java  
public void cambiarPantalla(String pantalla) { ... }  
```

✔️ Permite cambiar entre el menú (`MENU`) y la pantalla del juego (`JUEGO`).

✔️ Usa `cardLayout.show(contenedorPrincipal, pantalla);` para cambiar la vista.

✔️ Si se cambia a `JUEGO`, se usa `SwingUtilities.invokeLater(() -> movimiento.requestFocusInWindow());` para asegurar que el personaje reciba el foco.

## 3. Ejemplo de Uso

Para cambiar a la pantalla de juego desde el menú:

```java  
pantalla.cambiarPantalla("JUEGO");  
```

Esto cambiará la vista activa dentro del `CardLayout`.

## 4. Conclusión

La clase `Pantalla` gestiona la ventana del juego, el cambio entre pantallas y la organización de las capas en la pantalla de juego. Facilita la transición entre diferentes estados del juego de manera eficiente.