# Explicación del Código del Juego

Este documento explica el funcionamiento de nuestro juego, detallando la estructura de paquetes y clases, el propósito de cada una y cómo funciona el código. También se incluyen mejoras futuras para agregar enemigos, animaciones y mecánicas de disparo.

## 1. Estructura de paquetes y clases

El proyecto está dividido en los siguientes paquetes:  
  
- \*\*juegoprog.escenarios\*\*: Contiene clases relacionadas con los escenarios del juego.  
- \*\*juegoprog.sistema\*\*: Contiene las clases de control del juego.  
- \*\*resources\*\*: Carpeta de recursos gráficos como imágenes de escenarios y colisiones.  
  
Dentro de estos paquetes, tenemos las siguientes clases:  
  
- \*\*BaseEscenario\*\*: Clase base para gestionar escenarios.  
- \*\*ColisionesPanel\*\*: Maneja la detección de colisiones en el juego.  
- \*\*EscenarioDistritoSombrio\*\*: Implementación específica de un escenario.  
- \*\*Main\*\*: Punto de entrada del juego.  
- \*\*MenuPrincipal\*\*: Gestiona el menú del juego.  
- \*\*Movimiento\*\*: Controla el movimiento del personaje.  
- \*\*Pantalla\*\*: Gestiona la pantalla del juego.

## 2. Explicación de las clases

### BaseEscenario

Es la clase base para los escenarios. Contiene métodos que permiten gestionar la visualización y el comportamiento común de los escenarios.

### ColisionesPanel

Maneja la detección de colisiones basadas en la imagen de colisiones. Usa Pixel-Perfect Collision Detection para detectar dónde hay colisión en función del Alpha de la imagen PNG.

### EscenarioDistritoSombrio

Extiende BaseEscenario y define un escenario concreto en el juego.

### Main

Es el punto de entrada del juego, inicializa la pantalla y carga los recursos.

### MenuPrincipal

Muestra el menú inicial y gestiona la navegación.

### Movimiento

Controla la lógica del movimiento del personaje.

### Pantalla

Administra la ventana del juego, dibujando los elementos en pantalla.

## 3. Mejoras futuras y ampliaciones

En el futuro, queremos mejorar el juego con las siguientes implementaciones:  
  
- \*\*Enemigos\*\*: Se puede crear un nuevo paquete 'enemigos' y definir clases para los diferentes tipos.  
- \*\*Animaciones\*\*: Se pueden usar sprites animados en lugar de imágenes estáticas.  
- \*\*Disparos y acción\*\*: Se puede crear una clase 'Disparo' para gestionar el disparo del personaje y sus colisiones.  
  
Para estos cambios, sería recomendable estructurar el código en paquetes adicionales como 'enemigos' y 'acciones'.