

Estructura Libgdx

LibGDX es un framework de desarrollo de videojuegos en Java diseñado para crear juegos que funcionen en múltiples plataformas utilizando una única base de código. Este sistema organiza los proyectos en módulos independientes que permiten compilar el mismo juego para escritorio, Android, iOS y navegadores web mediante GWT. El módulo más importante es el 'core', ya que contiene toda la lógica esencial del juego, incluyendo entidades, pantallas, recursos y comportamiento general. Los demás módulos funcionan como lanzadores específicos para cada plataforma, permitiendo ejecutar el contenido del módulo principal sin modificar su código.

La arquitectura interna de LibGDX se basa en una estructura clara que separa la lógica del renderizado y la gestión del flujo del juego. La clase principal suele extender de 'Game', que permite administrar de forma ordenada las diferentes pantallas a través de objetos que implementan la interfaz 'Screen'. Estas pantallas representan los distintos estados del juego, como el menú, la partida, la pausa o el game over. Cada pantalla dispone de métodos específicos como 'show()', 'render()', 'resize()' o 'dispose()', que LibGDX ejecuta automáticamente según el ciclo de vida del juego. Este sistema facilita la organización y escalabilidad del proyecto incluso en juegos pequeños como Snake, donde solo se requieren una pantalla de juego y otra para el game over.

LibGDX ofrece una amplia variedad de herramientas integradas para simplificar el desarrollo. El sistema de dibujo mediante 'SpriteBatch' permite renderizar imágenes en pantalla de forma eficiente, aprovechando la GPU y reduciendo el consumo de recursos. Además, el 'AssetManager' facilita la carga y gestión de recursos como texturas, sonidos o fuentes, evitando errores y mejorando el rendimiento. Para manejar la entrada del usuario, LibGDX incorpora 'Gdx.input', que permite detectar eventos de teclado, ratón, pantalla táctil o incluso gamepads de manera unificada.

A nivel visual, LibGDX incorpora cámaras y viewports que permiten controlar qué parte del mundo del juego es visible en pantalla y adaptar la presentación a diferentes resoluciones sin deformaciones. Para juegos 2D, la cámara 'OrthographicCamera' es la más utilizada, mientras que viewports como 'FitViewport' o 'ExtendViewport' aseguran que el juego mantenga su proporción original independientemente del dispositivo. También incluye soporte para animaciones mediante secuencias de imágenes, spritesheets y 'TextureAtlas', lo que facilita la creación de objetos animados con un uso eficiente de los recursos.

El ciclo de renderizado en LibGDX es continuo y se ejecuta decenas de veces por segundo. En cada frame se procesan tres etapas fundamentales: la entrada del usuario, la actualización de la lógica interna del juego y el renderizado en pantalla. Este flujo constante asegura una experiencia fluida y controlada, permitiendo al desarrollador actualizar las posiciones de los objetos, detectar colisiones, reproducir animaciones y dibujar los elementos visuales en el orden correcto.

Entre sus ventajas más destacadas, LibGDX ofrece un alto rendimiento, una API madura, gran flexibilidad y la posibilidad de compilar el mismo juego para varias plataformas sin esfuerzo adicional. Al ser open source, cuenta con una comunidad activa y abundante documentación. Sin embargo, también presenta algunas limitaciones como la ausencia de un editor visual, la necesidad de conocimientos de Java y aspectos básicos de OpenGL, y la configuración adicional requerida para compilar en iOS. Aun con estas limitaciones, se mantiene como una de las herramientas preferidas para quienes desean aprender programación de videojuegos desde un enfoque técnico y completo.