

Práctica 1: Clon de *Off the line*

Guillermo Sánchez de Lamadrid Martínez

Gonzalo María Cidoncha Pérez

Módulos:

Tal y como se nos fue indicado en clase, se hace uso de 6 módulos distintos:

- **desktopGame:** Donde se inicializa la ventana en PC. Depende de "PCEngine", "Engine" y "Logica".
- **androidGame:** Donde se inicializa la app en Android. Depende de "AndroidEngine", "Engine" y "Lógica".
- **androidEngine:** Posee el AndroidEngine, AndroidGraphics, AndroidInput y Font (de Android). En AndroidEngine se encuentra el bucle principal en Android. Posee una única dependencia, del módulo "Engine".
- **PCEngine:** Posee el PcEngine, PcGraphics, PcInput y Font (de PC). En PC engine se encuentra el bucle principal de PC. Depende de "Engine".
- **Engine:** Módulo que posee las interfaces "Engine", "Font", "Graphics", "Input" y "Lógica". No posee ninguna dependencia.
- **Lógica:** Posee una lista extensa de clases usadas para la lógica del juego. Absolutamente todo el código que pueda ser común para ambas versiones del juego se encuentra en este módulo. Depende únicamente del módulo "Engine", haciendo uso de sus interfaces. También incluimos la librería "json-simple:1.1.1" para el parseo de JSON.

Arquitectura de clases

1. Lógica:

- **Lógica.** Llama al renderizado y actualizado del menú o los niveles según convenga.
- **LógicaMenu.** Se encarga de renderizar y actualizar los elementos del menú.
- **LógicaNiveles.** Se encarga de renderizar y actualizar los elementos del juego.
- **GameObject.** Clase básica de dibujado y actualizado de elementos en pantalla, de esta heredan: **Character**, **Cross**, **Enemy**, **Line**, **Path**, **Square** y **Boton**.
- **Player:** Hereda de Character al igual que **Coin**.

- **JsonHandler:** Realiza la lectura de los JSON
- **Nivel:** Contiene la información del nivel.
- **Collision:** La clase "Utils". Contiene métodos estáticos.
- **Coordenada:** Clase que contiene X e Y.
- **Segmento:** Clase para almacenar segmentos y otra información necesaria.
- **Vector2D:** Clase vector 2D y su normalización.

2. Engine:

- Interfaz **Engine**.
- Interfaz **Font**.
- Interfaz **Graphics**.
- Interfaz **Input**.
- Interfaz **Lógica**.

3. AndroidEngine:

Implementa la versión para android de las interfaces de engine, así como el bucle principal.

4. PCEngine:

Implementa la versión de pc de las interfaces de engine y el bucle principal.

5. DesktopGame:

- **Main**

6. AndroidGame:

- **MainActivity**