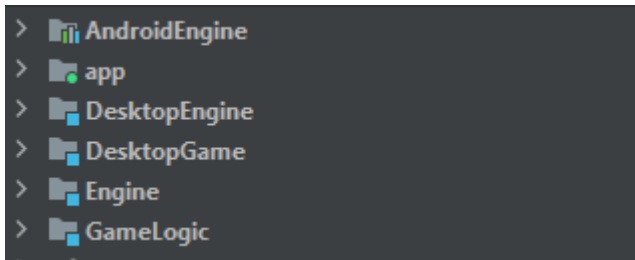
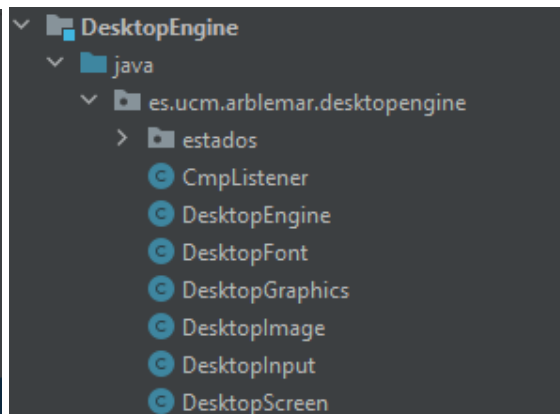
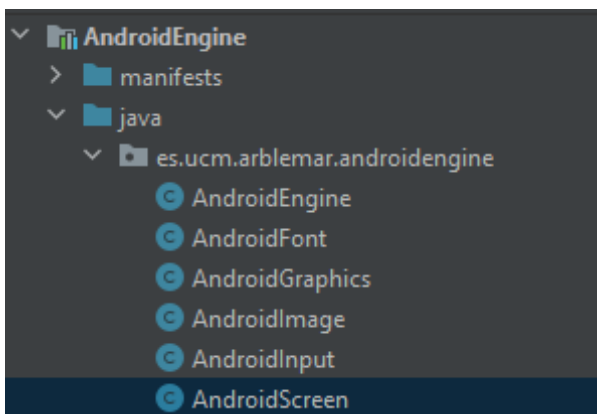


Arquitectura Ohn0

Para realizar la práctica multiplataforma se dispone de los siguientes módulos:



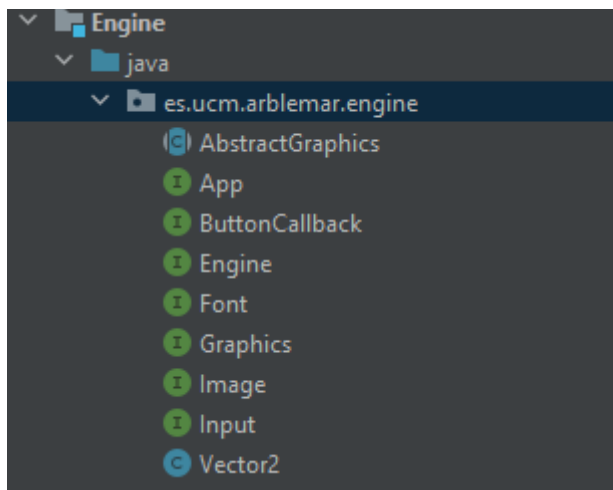
- **Engine:** es el módulo que contiene las interfaces y clases comunes para ambas plataformas. No tiene dependencias.
- **GameLogic:** módulo contenedor de todas las clases necesarias para implementar la lógica del juego, es decir, contiene cada uno de los estados del juego (clases que implementan la interfaz **App** de **Engine**) así como todos los objetos, funciones y algoritmos necesarios para la resolución del tablero, pistas, renderizado, etc. Tiene una dependencia de **Engine**.
- **AndroidEngine:** contiene las clases que implementan cada una de las interfaces del módulo Engine para que funcione en dispositivos Android. Tiene una dependencia con **Engine**
- **DesktopEngine:** análogamente igual que AndroidEngine, también con la dependencia de **Engine**.



- **app / DesktopGame:** sirven únicamente para inicializar el juego en dispositivos Android y en PC respectivamente, de manera que dependen del módulo **GameLogic** y de **Engine**, puesto que hace falta poner en marcha el motor.

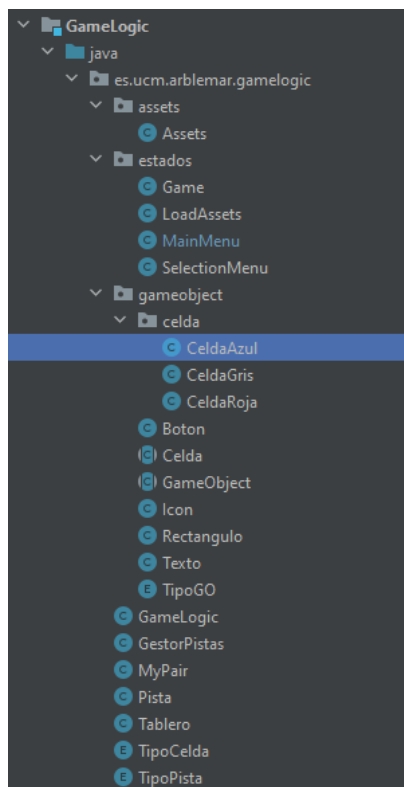
Módulo: Engine

Se han creado las interfaces pedidas en el enunciado de la práctica, ampliando debidamente con **AbstractGraphics**, **ButtonCallback** y **Vector2**.



- **AbstractGraphics**: implementa la interfaz de Graphics, de manera que contiene cálculos comunes para las transformaciones de los objetos en Android y PC.
- **ButtonCallback**: interfaz que contiene un método **doSomething** diseñado para crear callbacks en los botones del juego.
- **Vector2**: contiene dos enteros (x, y) para ser utilizados a forma de vector o **Pair**.

Módulo: GameLogic



Para la lógica hemos utilizado un sistema polimórfico a través de la herencia de **GameObject** de manera que en los **estados**, los cuales implementan la interfaz de **App**, poseen una lista de cada uno de los objetos del juego para procesar su input, render y update.

De esta misma manera funcionan los estados, es decir, desde el **Engine** principal, tanto de Android como de PC, se está llamando continuamente al **handleInput**, **render** y **update** del estado actual del juego.