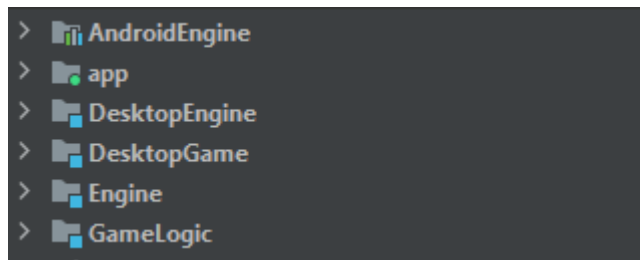
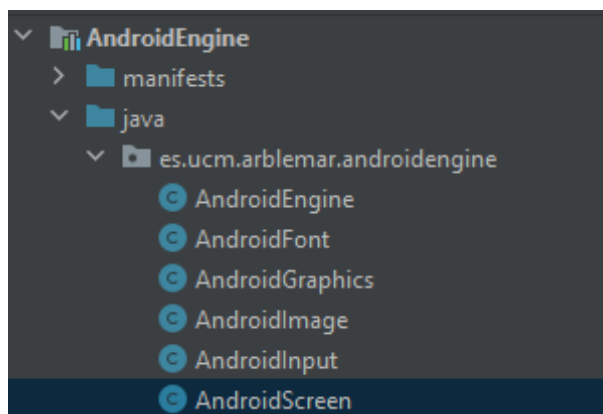


Arquitectura 0hn0

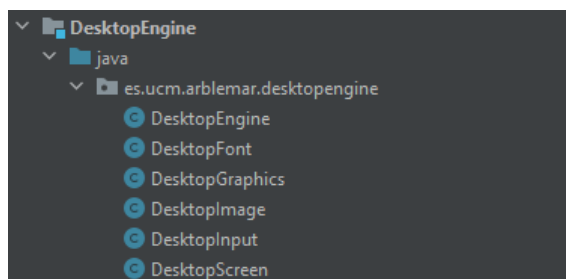
Para realizar la práctica multiplataforma se dispone de los siguientes módulos:



- **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 **State** 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**.



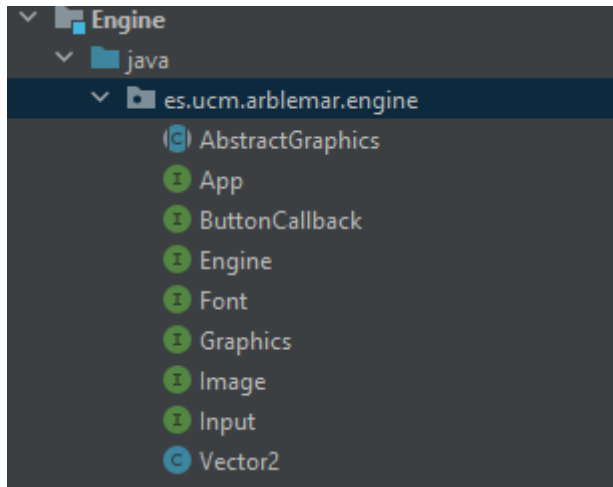
- **DesktopEngine:** análogamente igual que AndroidEngine, pero para que funcione en PC. También con la dependencia de **Engine**.



- **app / DesktopGame:** sirven únicamente para inicializar el juego en dispositivos Android y en PC respectivamente, de manera que dependen del módulo **GameLogic** y **Engine** puesto que hace falta poner en marcha el motor. Luego, cada uno tiene una dependencia de su respectiva plataforma, app de **AndroidEngine**, DesktopGame de **DesktopEngine**.

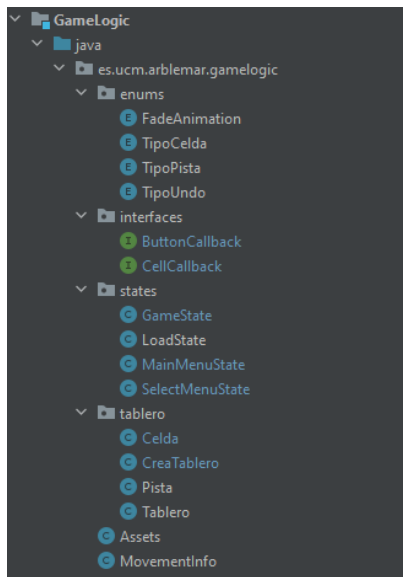
Módulo: Engine

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



- **AbstractEngine:** implementa la interfaz Engine, de manera que contiene aquella funcionalidad común para ambas plataformas: getGraphics, getInput, reqNewState(para solicitar un cambio de estado) y updateDeltaTime (para actualizar el deltaTime). Además, declara aquellos atributos comunes en ambas plataformas que sean necesarios.
- **AbstractGraphics:** implementa la interfaz de Graphics, de manera que contiene cálculos comunes para las transformaciones de los objetos en Android y PC. También contiene todos aquellos atributos que se necesiten para los cálculos mencionados.
- **AbstractInput:** implementa la interfaz Input y posee los atributos y métodos comunes para ambas plataformas: GetTouchEvents (devuelve una lista de los eventos del input) y onTouchDownEvent(que convierte las coordenadas del input en coordenadas lógicas y las añade a la lista de eventos).

Módulo: GameLogic



En la carpeta **enums** se han guardado todos aquellos enumerados que se han utilizado para la lógica del juego y en la de **interfaces**, las interfaces auxiliares usadas.

Posteriormente, en la carpeta **states** se han guardado cada uno de los estados del juego. Todos ellos implementan la interfaz State. Desde el **Engine** principal, tanto de Android como de PC, se está llamando continuamente al **handleInput**, **render** y **update** del estado actual del juego, que puede ser cualquiera de los que se ven en la imagen.

En el paquete **tablero**, están aquellas clases necesarias para llevar a cabo la creación del tablero, así como la funcionalidad para gestionar la lógica del tablero en sí.

Finalmente, aquellas clases que no pertenecen a un lugar en concreto no se han acomodado en ningún paquete adicional.

También se ha adjuntado un documento **correccionesP1.txt** donde se resaltan cada una de las correcciones comentadas en el campus virtual y la forma de solucionarlas.