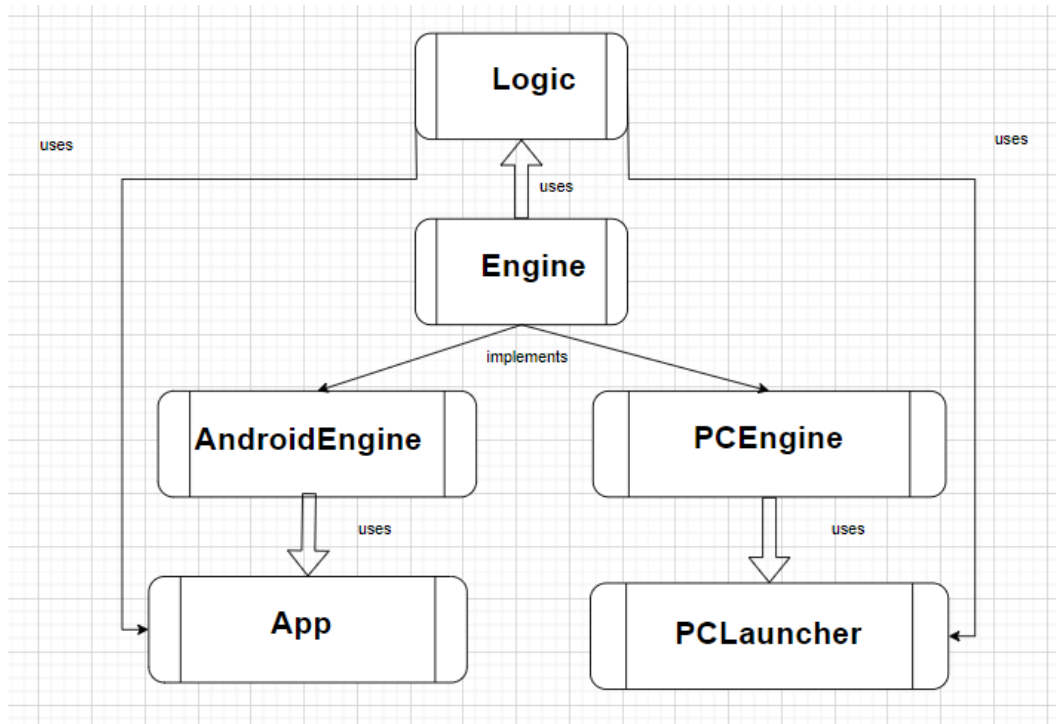


Práctica 2: Nonogramas

1. Introducción

Esta práctica se compone en un total de 6 módulos distintos, dos de ellos siendo launchers para diferentes plataformas (PC y Android respectivamente), tres ellos siendo el motor básico y sus adaptaciones para cada una de las versiones, y por último tenemos la lógica del juego en sí.



En esta imagen se puede observar los diferentes módulos y la comunicación entre ellos, como se puede ver la lógica se comunica con los launchers pero es para después definirse en los diferentes motores como el "juego" que se corre como se ve en la imagen de abajo.

```
engine = new AndroidEngine(renderView, context: this);
engine.getGraphics().setResolution( width: 400, height: 600);

Logic logic = new Logic(engine);
engine.setLogic(logic);
```

Fragmento de código del launcher de Android

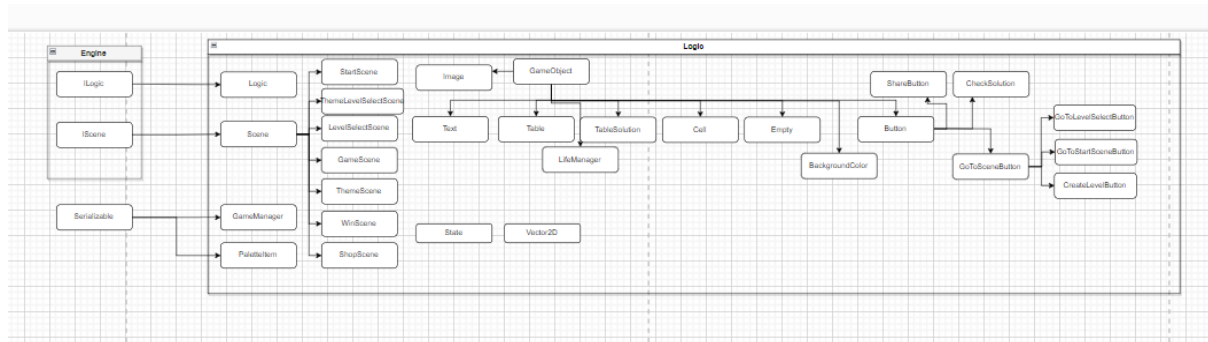
Y aunque realmente solo sean necesarios 3 de ellos (lógica, motor y launcher) hemos continuado con la arquitectura de módulos de la práctica anterior por comodidad y se ha adaptado el juego para que siga funcionando en PC pero sin lo único de android (compartir contenido, uso de sensores, notificaciones y anuncios) .

A continuación se explicará sobre la implementación de los diferentes módulos de Android

2. Módulos e implementación

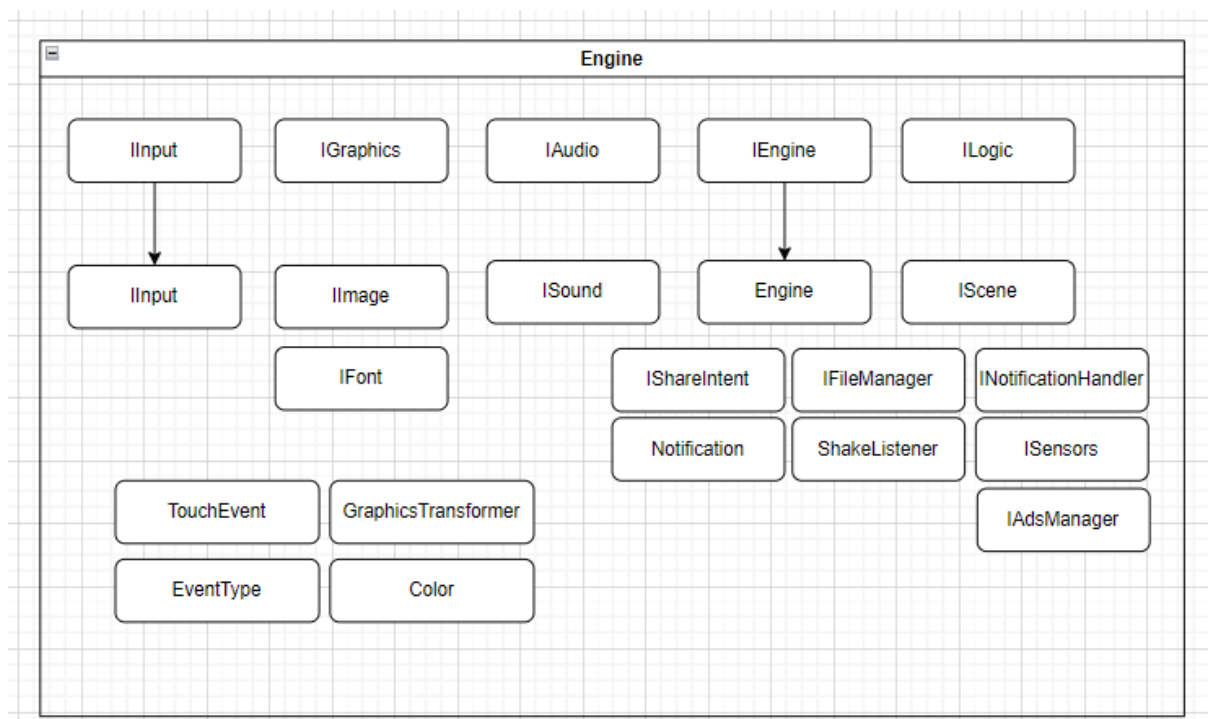
2.1. Lógica

Este módulo como se ha explicado anteriormente, contiene la definición del juego en sí, así como las diferentes escenas del mismo



Respecto a la 1ª práctica se han añadido nuevos elementos como más escenas (escena de tienda y la selección de temáticas), botones (como el botón de compartir contenido o comprar paletas, pero sobre todo han sido botones de ir a otras escenas) pero lo principal que se ha añadido es el *GameManager* que se encarga de gestionar todos los recursos y del guardado del juego.

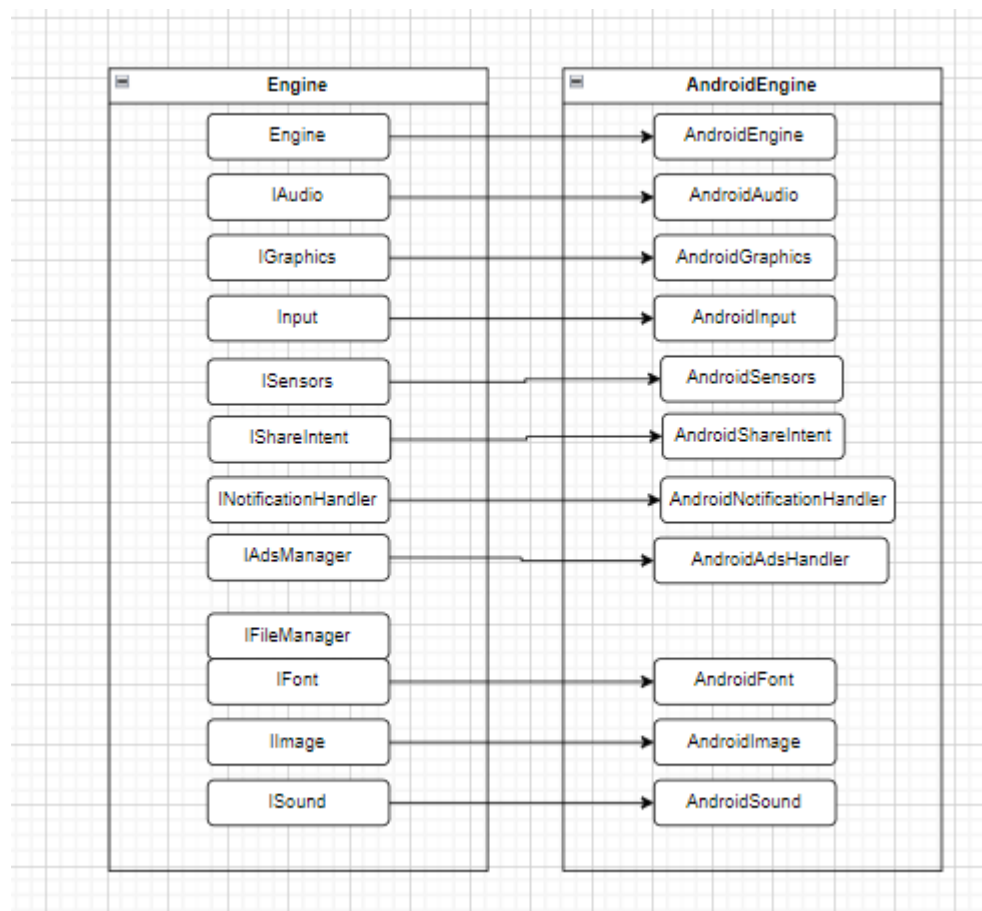
2.2. Motor básico



Se han añadido nuevos managers que se encargan de gestionar la comunicación entre el motor de Android y la lógica ya que hemos continuado con la misma arquitectura de módulos.

- **IGraphics:** Contiene métodos para manejar los gráficos dibujando diferentes elementos, imágenes(definida en la interfaz *IImage*) y textos con diferentes fuentes (definida en la interfaz *IFont*), con distintos colores (definida en la clase *Color*).
- **IAudio:** Contiene métodos para manejar el audio, es decir, reproducir sonidos(definidos en la interfaz *ISound*)
- **Input:** Maneja el input, contiene una lista de eventos(definidos en la clase *TouchEvent*, estos contienen una posición y un tipo de evento(*EventType*)
- **ILogic:** Interfaz que define el juego a correr en el motor.
- **IAdsManager:** Gestiona los anuncios.
- **INotificationHandler:** Gestiona las notificaciones.
- **IFileManager:** Gestiona la lectura y escritura de archivos.
- **ISensors:** Gestiona los sensores que se usan en la aplicación.
- **IShareIntent:** Gestiona los *Intents* y el compartimiento de contenido a través de otras plataformas.
- **Engine:** Core del módulo, contiene métodos para definir los distintos sistemas que se usarán en el bucle principal de la aplicación.

2.3. Motor Android



Respecto a la anterior práctica, se han añadido estos nuevos managers que se encargan cada uno de la adaptación respectiva de Android a su función original.