Práctica 2: Mastermind

1. Introducción

Esta práctica se compone en un total de 3 módulos: Android Engine, Engine y Logic

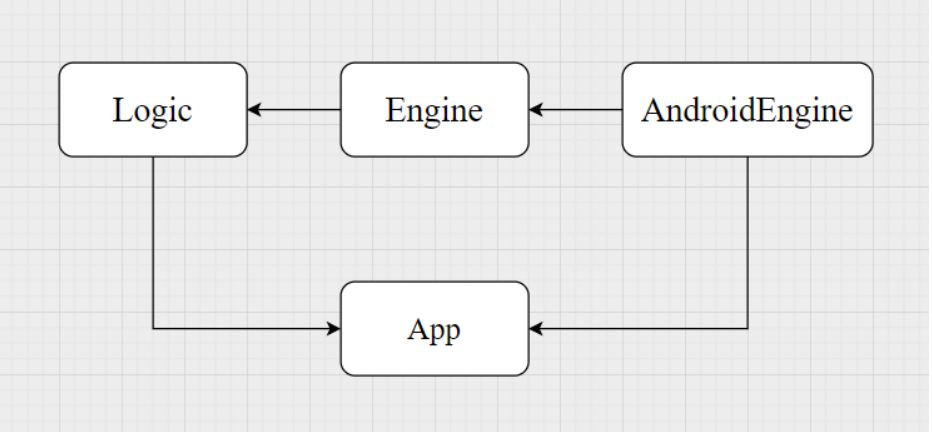


Diagrama de dependencias de los módulos del proyecto

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 lanza.

A continuación se explicará en detalle el contenido de cada módulo así como las clases que los forman.

2. Módulos e implementación

2.1. Lógica

- **Clases de Logica**
- ***Logic:*** Implementa la interfaz de ILogic del módulo de engine para funcionar como gestor de escenas y base del juego al motor.
- **PlayerData**: Almacena los datos que queremos que persistan, como las monedas, las skins desbloqueadas de la tienda, el progreso….También se encarga del guardado y cargado

Escenas

- ***Scene:*** Clase abstracta que funciona como contenedor de Game Objects
- ***Menu Scene:*** Escena del menú principal
- **ExploreWorldsScene**: Escena del modo historia
- ***Choose Level Scene:***Escena de elegir dificultad
- ***GameScene:***Escena del juego en sí
- ***Win Scene:***Escena de ganar o perder
- **ShopScene**: Escena de personalización: compra de skins, fondos y paletas

GameObjects

- ***GameObject:*** Clase abstracta básica que representa un objeto de juego
- ***Image:*** Objeto que renderiza una imagen
- ***Text:*** Objeto que renderiza un texto con una determinada fuente
- ***Button:*** Objeto abstracto que al tocarlo hace una acción (de la cual heredan muchos objetos)
  - ***Go To Scene:*** Clase que al pulsar va de una Escena a otra(hay varios de estos)
  - **BuyItemButton**: Clase abstracta que sirve para comprar cosas y desbloquear cosas.
    - **SetAnimalButton**: Botón que te desbloquea un animal
    - **SetPaletteButton**: Botón que te desbloquea una paleta de colores.
  - ***Cell:*** Clase que representa cada una de las casillas del juego (círculo gris o imagen gris)
  - ***Colouring Cell:*** Clase que al pulsar, colorea la primera celda disponible, internamente la rellena con un valor del 0 a n° de colores posibles -1.
  - ***Daltonic Button:*** Botón con dos imágenes que al pulsarlo activa el modo daltonismo. El cual hace que sobre las casillas rellenas y los colouring buttons se ponga además de la imagen o circulo, un número. Para la comunicación de este modo se ha implementado una interfaz *Daltonic Listener* la cual al informar a la escena de haberse activado, informa a todos los elementos "escuchantes".
  - **ShowRewardButton**: Clase abstracta que te muestra un anuncio recompensado, la recompensa que te da se implementa en las clases hijas de esta.

- -**ShareContentButton:** Clase que comparte una captura de pantalla de la app.

- ***Table:*** Clase que representa cada uno de los intentos en sí
- ***Colouring Table:*** Clase que contiene los *Colouring Cells*
- ***HintObject:*** Clase que representa las pistas
- **HintElem:** Clase que representa cada una de esas pistas

2.2. Motor Común (Engine)

El motor común consta de la definición básica de las funcionalidades que tiene que ir para cada uno de los motores concretos (aunque en este caso sea solo android), así como de la comunicación con lógica (por ejemplo: dibujar una imagen de un botón)

Diagrama de clases que están en el módulo de Engine

- **Clases del motor común**

Aquí tenemos las clases comunes de la anterior práctica:

- ***IEngine:*** Interfaz que contiene cada uno de los managers de las funcionalidades(graphics, audio, logic e input)
- ***Engine:*** Implementa la interfaz anterior para poder asignar y obtener estos managers.

- ***IGraphics:*** Interfaz que define los métodos de dibujado
- ***IImage:*** Interfaz que define una imagen
- ***IFont:*** Interfaz que define una fuente de escritura.
- ***Color:*** Clase que nos presenta un modelo de color rgba y nos lo abstrae al renderizado (que utiliza argb con hexadecimales y para colores concretos nos manejamos mejor con rgb y rango de 0 a 255)
- ***GraphicsTransform:*** clase que se encarga del redimensionado de la "pantalla lógica" respecto a la pantalla del dispositivo (actualizado cuando se gira el móvil o cuando se redimensiona la pantalla en PC)

- ***IAudio:*** Interfaz que define las funcionalidades de sonido
- ***ISound:*** Interfaz que representa un efecto de sonido

- ***IInput:*** Interfaz que define la cola de eventos de input (toques en la pantalla y scroll en la pantalla)
- ***Input:*** Clase que implementa la interfaz IInput
- ***TouchEvent:*** Clase que representa un evento de input contiene las coordenadas de la pantalla, las coordenadas de la pantalla lógica y el tipo de evento

- ***ILogic:*** Interfaz que define el core del juego.

Y a continuación tenemos las nuevas clases de esta práctica:

- **IAdsManager:** Interfaz que gestiona los anuncios banners y recompensados
- **ISensorsManager:** Interfaz que comunica los sensores con la lógica. Tiene métodos para registrar los oyentes de los sensores que queremos(en este

caso de los acelerómetros, aunque el motor está preparado para registrar otros tipos de sensores).

- **ISensorListener**: Interfaz que define un escuchante de los sensores del juego

- **Notification**: Representación de una notificación en el motor, encapsula los datos que vamos a enviar.
- **INotificationHandler**: Interfaz que define un gestor de notificaciones

- **IFileManager**: Interfaz que maneja la lectura de archivos
- **IJsonObject**: Interfaz que define un archivo JSON.
- **IShareContentManager**: Interfaz que se encarga de compartir contenido

---

2.3. Motor de Android (Android Engine)

Este módulo consiste en la implementación del motor adaptado para la plataforma de android

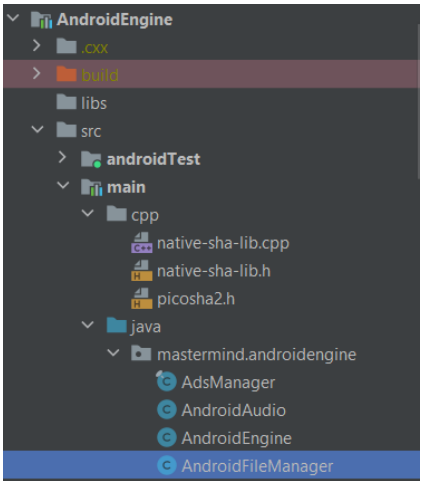
Diagrama con la herencia de las clases que están en el módulo de Android Engine

Clases de Android Engine:

- **Android Engine:** Hereda de Engine e implementa Runnable, crea los otros managers de android y representa la clase con el bucle principal del juego

- **Android Graphics:** Implementa los métodos de *IGraphics* con las librerías de Android
- **Android Image:** Implementa la interfaz *IImage* usando los Bitmaps de Android.
- **Android Font:** Implementa la interfaz *IFont* usando las Typefaces de Android.
- **Android Audio:** Implementa la interfaz *IAudio* usando SoundPool para efectos de sonido cortos.
- **Android Sound:** Implementa la interfaz *ISound* para efectos de sonido.
- **Android Input:** Hereda de *Input* e implementa *View.OnTouchListener* y recoge los eventos de scroll y toque de pantalla(pulsar y levantar)

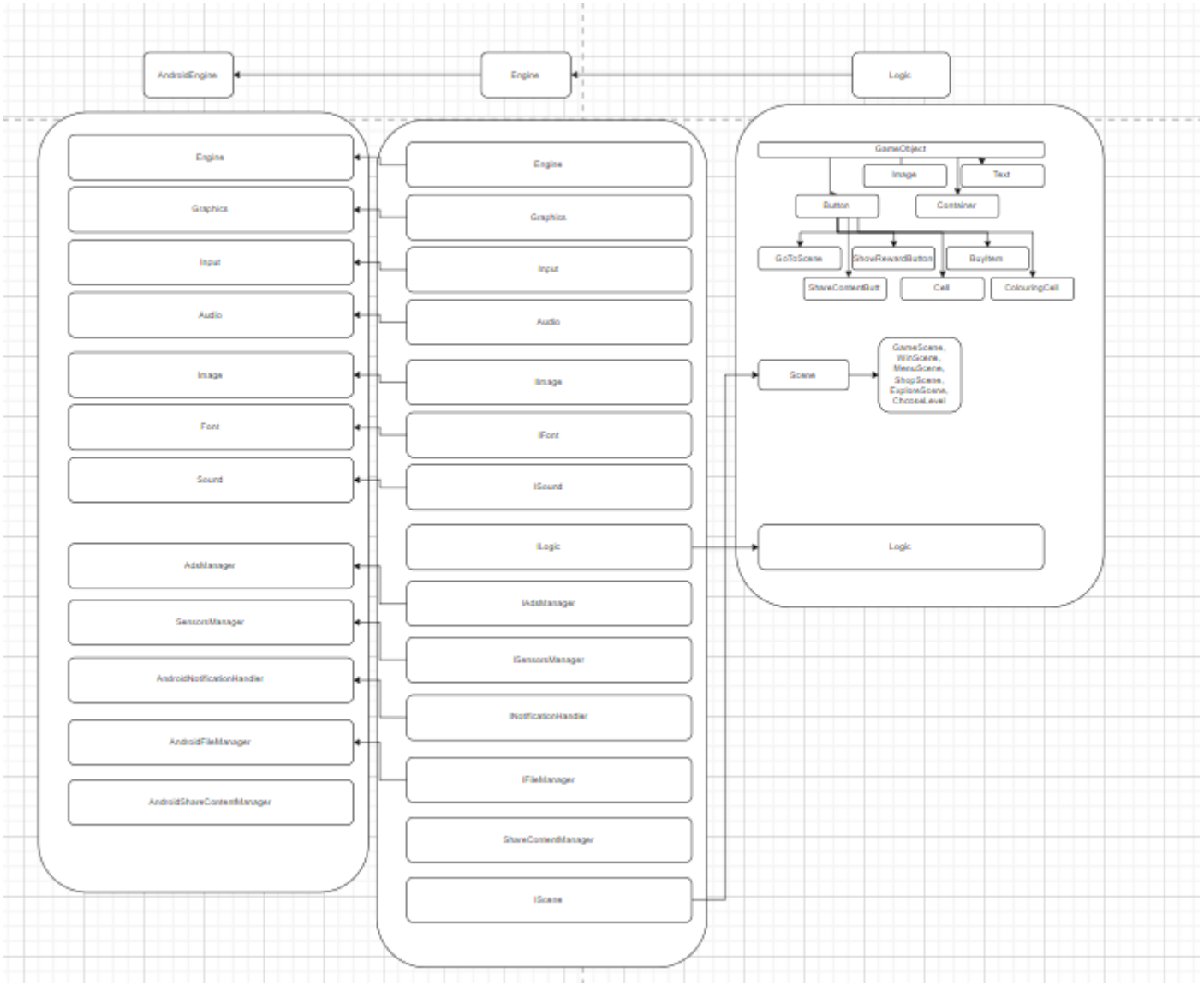
- **AdsManager:** Implementa la interfaz *IAdsManager* que gestiona la vista de los anuncios de banner (*AdView*) y los anuncios recompensados.
- **SensorsManager:** Implementa la interfaz *ISensorsManager*, aunque solo escuche el acelerómetro está preparado para meter otros sensores.
- **AndroidFileManager:** Implementación de la interfaz *IFileManager*, contiene metodos para gestionar los archivos así como encriptarlos mediante una NDK con el algoritmo SHA



- **AndroidJSONObject:** Representación de un JsonObject en nuestro motor.
- **AndroidNotificationHandler:** Implementación del gestor de notificaciones. Contiene un ArrayList de notificaciones pendientes de enviar, al salir

de la app estas notificaciones se ponen en una cola de tareas del NotificationWorker en el módulo de app.

- **AndroidShareContentManager:** Implementación de la interfaz *IShareContentManager*, tiene la funcionalidad de sacar una captura de la surfaceView y compartirla mediante un Intent.

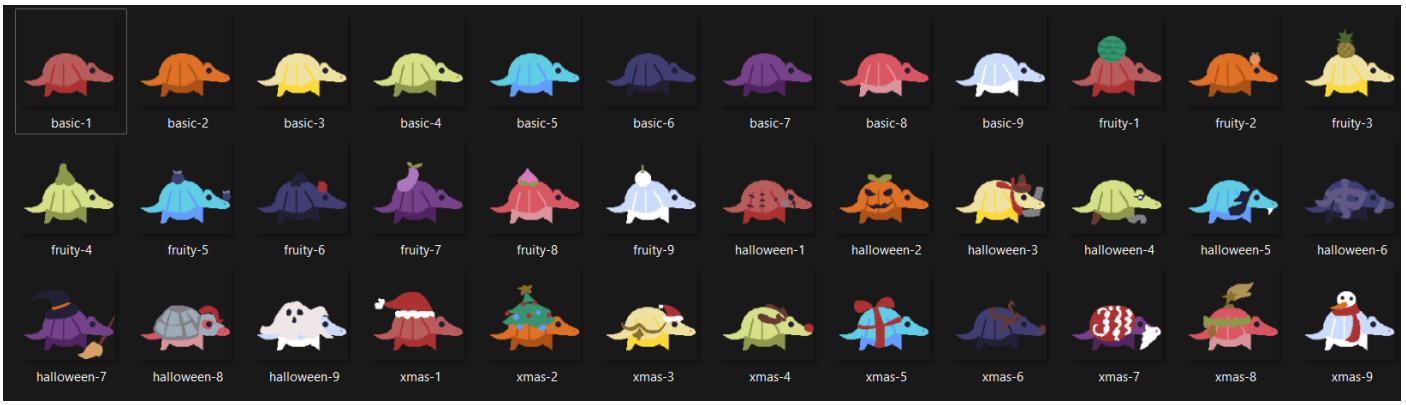


3. Referencias

Este apartado se usa como créditos a los recursos usados en la práctica.

Para el juego, hemos seguido una temática de animalitos de colores con diferentes disfraces para cada uno de los mundos, en la tienda tienes la opción de jugar con otros animalitos para más personalización.

- **Imágenes:** Creados a mano a excepción de algunos iconos (flecha de volver atrás por ejemplo) sacados de flaticon.



Una armada de armadillos

- **Sonidos:** 4 sonidos sacados de freeSounds: uno para el botón de daltonico, otro sonido cuando gastas dinero, otro para la pantalla de ganar y otro para la de perder.
- **Fuentes:** 2 fuentes sacadas de dafont