Práctica 1: Mastermind

1. Introducción

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

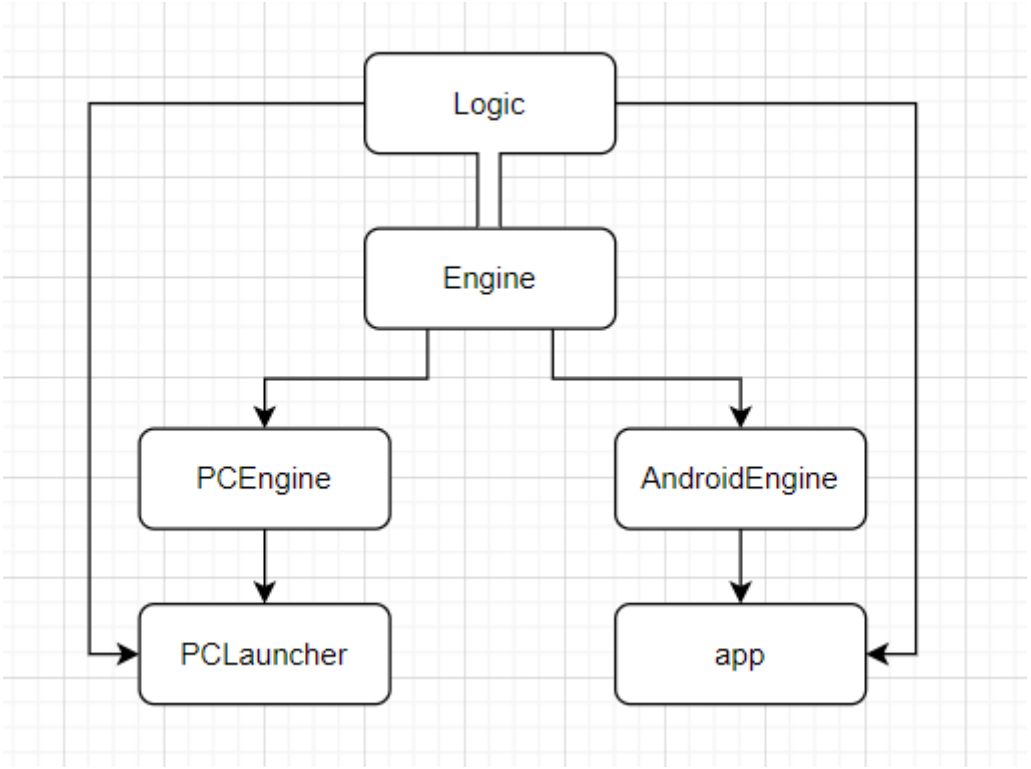

Diagrama de dependencias de los módulos del proyecto

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

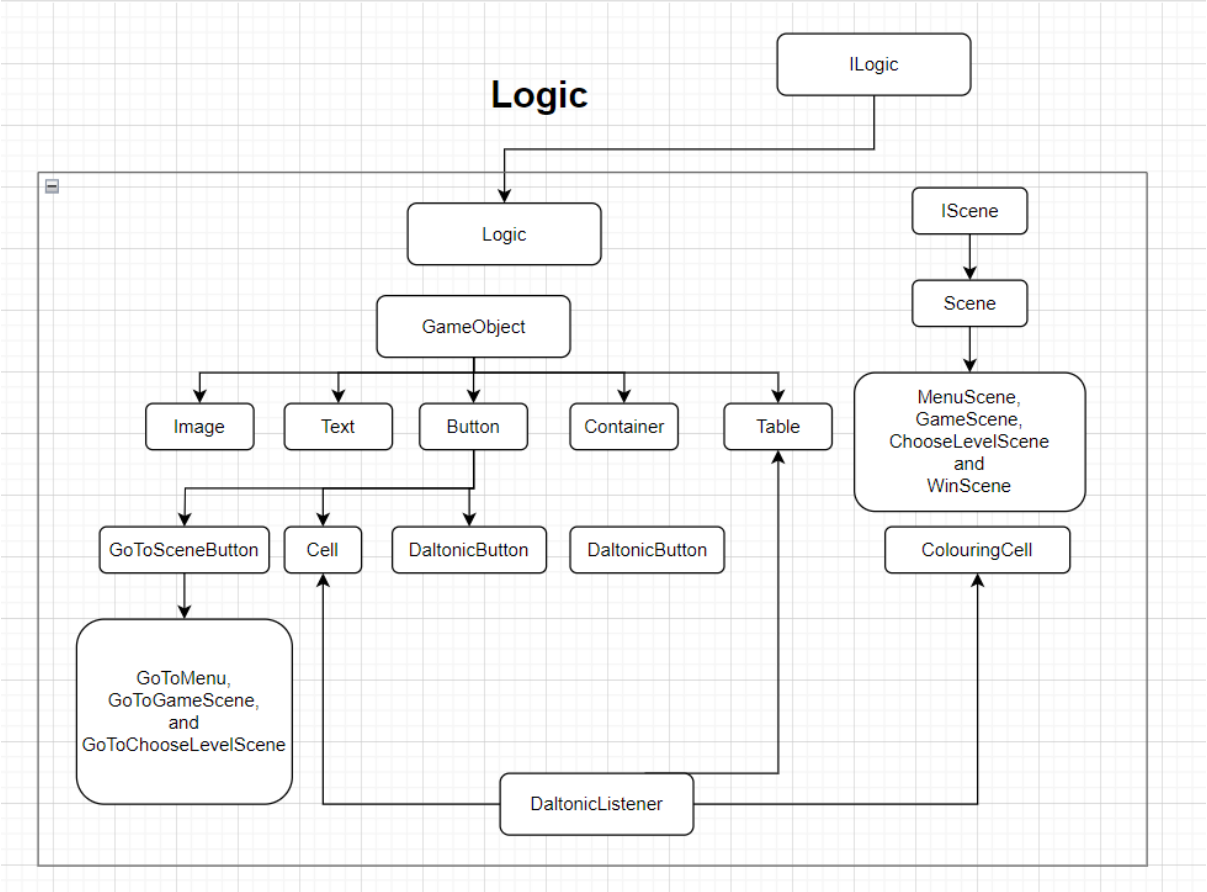

Diagrama de clases de Logic

- **Clases de Lógica**
- ***Logic:*** Implementa la interfaz de ILogic del módulo de engine para funcionar como gestor de escenas y base del juego al motor.

Escenas

- **IScene**: Interfaz que define los métodos de una escena (init: donde se crean los elementos, render: donde se dibujan, update: donde se actualizan ,y handleInput: donde se gestionan los eventos)
- ***Scene:*** Clase abstracta que funciona como contenedor de Game Objects
- ***Menu Scene:*** Escena del menú principal
- ***Choose Level Scene:***Escena de elegir dificultad
- ***GameScene:***Escena del juego en sí
- ***Win Scene:***Escena de ganar o perder

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)
  - ***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".
- ***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)

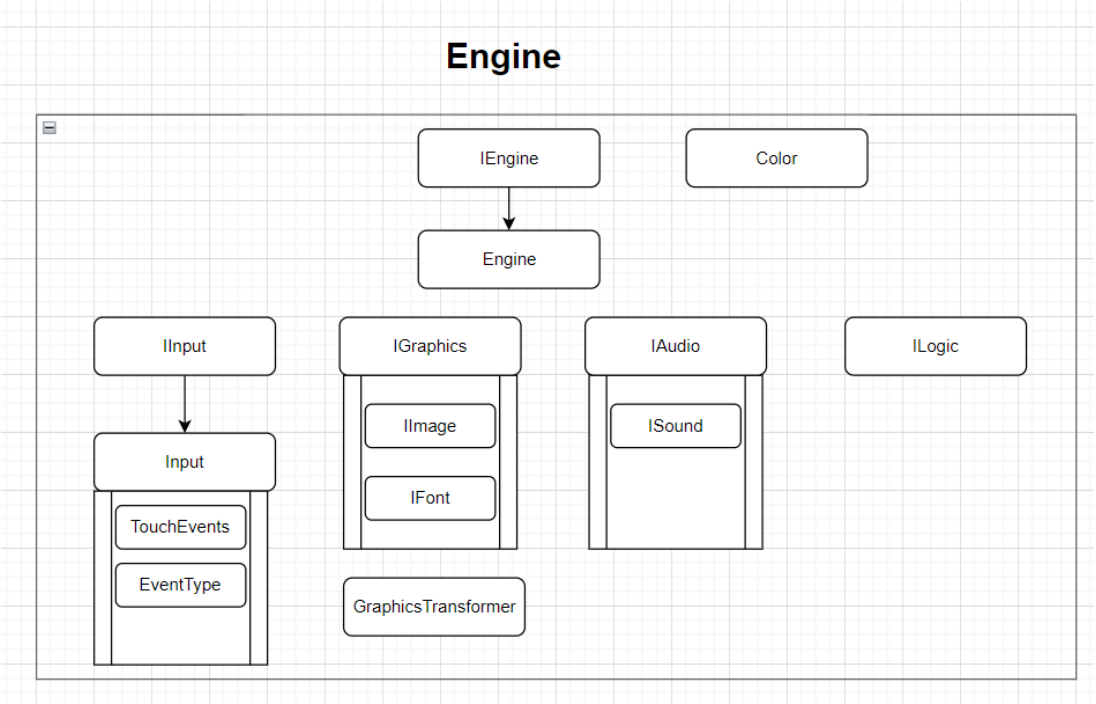

Diagrama de clases que están en el módulo de Engine

Clases del motor común

- ***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.

2.3. Motor de Android (Android Engine)

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

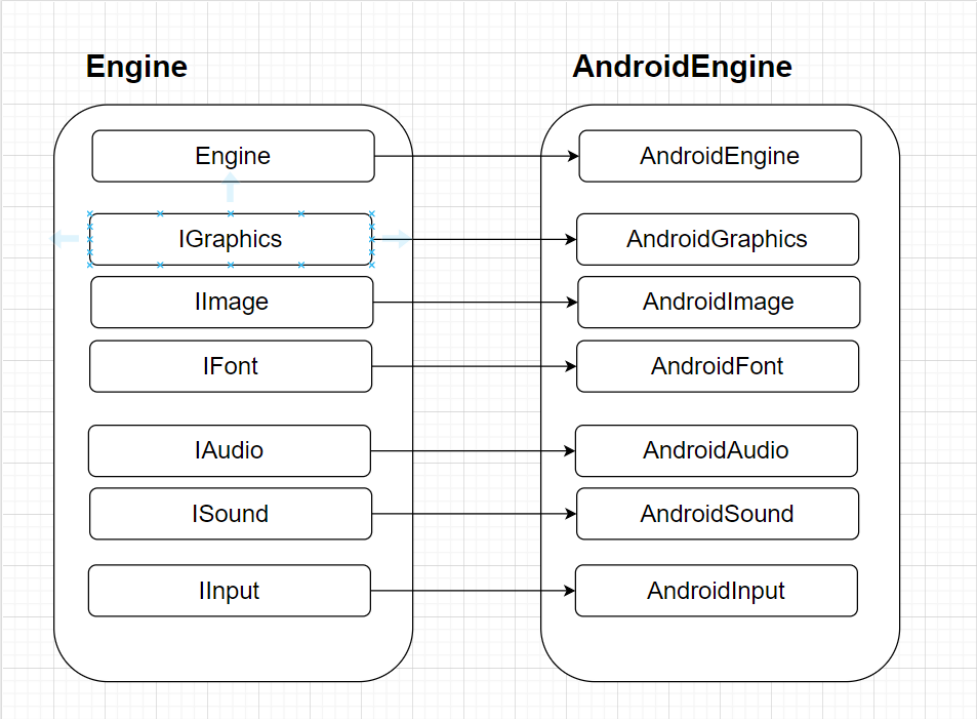

Diagrama con la herencia de las clases que están en el módulo de Android Engine

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 toque de pantalla(pulsar y levantar)

2.4. Motor de PC (PC Engine)

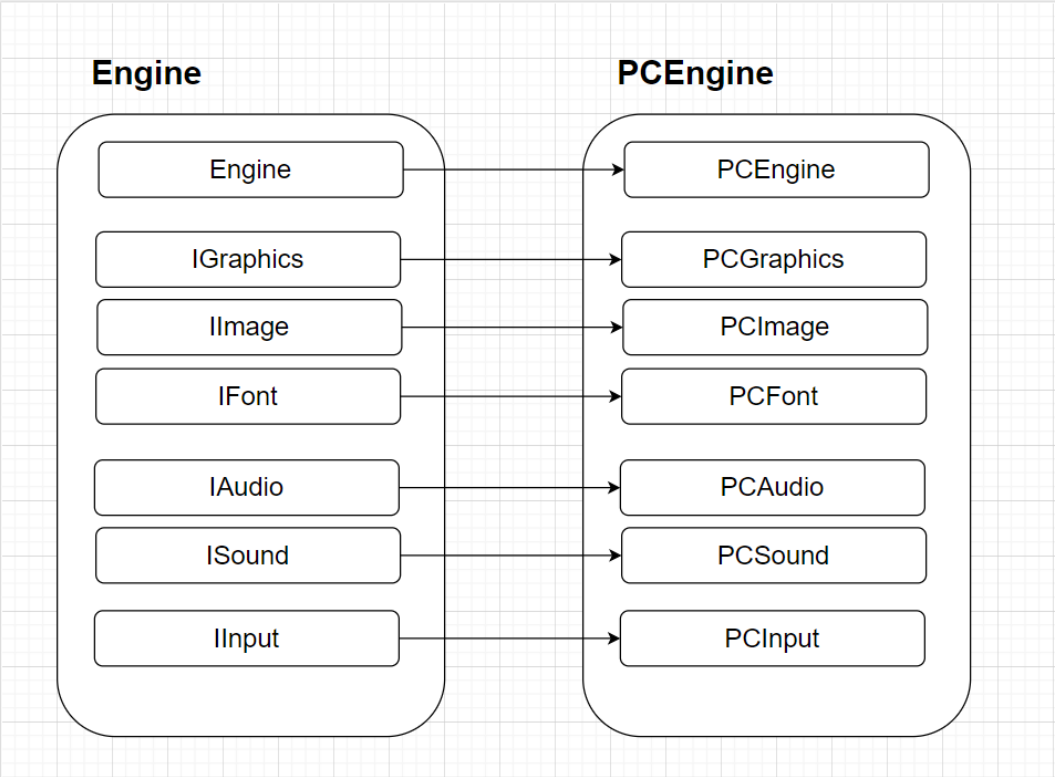

Diagrama con la herencia de las clases que están en el módulo de PC Engine

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

- **PC Graphics:** Implementa los métodos de *IGraphics* con las librerías de java swing y java awt para la gestion de imágenes, colores y fuentes
- **PC Image:** Implementa la interfaz *IImage* usando los Buffered Image de java.awt.
- **PC Font:** Implementa la interfaz *IFont* usando las Fonts de la libreria java.awt
- **PC Audio:** Implementa la interfaz *IAudio* usando *AudioSystem de la librería javax.sound.sampled.AudioSystem* para efectos de sonido cortos.
- **PC Sound:** Implementa la interfaz *ISound con javax.sound.sampled.Clip* para efectos de sonido.
- **PC Input:** Hereda de *Input* e implementa *MouseInputListener, KeyListener* y recoge los eventos de toque de pantalla(click izquierdo y derecho respectivamente; además cuando se presiona la tecla F se pone la ventana completa

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:** sacados de flaticon
- **Sonidos:** 1 sonidos sacado de freeSounds: uno para el click de las cells
- **Fuentes:** 2 fuentes sacadas de dafont