# Módulos

* Módulo “*Interface*”:

Contiene todas las interfaces que se encargarán de implementar los módulos de PC y de Android. En este módulo encontramos las siguientes interfaces:

Engine: Proporciona métodos para controlar el ciclo de vida del juego, como iniciar, pausar y cambiar de escenas. Además, a través de esta interfaz se puede obtener una referencia a “*Graphics*”.

Font: Permite obtener el tamaño de la fuente.

Graphics: Se utiliza para dibujar gráﬁcos en la pantalla. Permite la renderización de imágenes y texto en una superﬁcie de dibujo. Además, contiene todos los métodos de conversiones entre dimensiones “reales” y dimensiones “lógicas”, para evitar el deformamiento de la pantalla al redimensionarla.

Image: Proporciona métodos para obtener las dimensiones de una imágen.

Input: Contiene un array de eventos que se usará en cada frame para determinar la entrada del usuario. Este array está formado por la clase “TouchEvent”, que contendrá toda la información de la entrada (por ejemplo, si se ha pulsado o soltado el botón del ratón).

Interface: Se utiliza para deﬁnir los métodos que tendrán que sobreescribir todos los elementos de la aplicación, como el renderizado o la actualización en cada frame. Además, contiene un método para obtener los eventos de entrada de usuario.

Scene: Tiene métodos para representar escenas en la aplicación. Al igual que “Interface”, los métodos que contiene se encargan del renderizado de la escena, y la actualización de los elementos propios de ella, además de tener un método para la entrada de usuario.

* Módulo “*LibEnginePC*”:

En este módulo se sobreescriben todas las interfaces deﬁnidas anteriormente para deﬁnir su comportamiento en PC. Por ejemplo, PC usa las librerías proporcionadas por Java.AWT, la API estándar para suministrar una interfaz gráﬁca de usuario para un programa Java.

* Módulo “*LibEngineAndroid*”:

En este módulo se sobreescriben todas las interfaces deﬁnidas anteriormente para deﬁnir su comportamiento en Android. Por ejemplo, Android usa las clases proporcionadas por Android.View y Android.ViewGroup.

* Módulo “*LogicLib*”:

Contiene todos los elementos comunes a la lógica del juego que se usan en las escenas, todo el contenido de este módulo se comparte entre Android y PC. Los elementos de este módulo son los botones, las escenas, y las clases que controlan la lógica del propio juego.

* Módulo “*Desktop*”:

Módulo que contiene el main() de la aplicación de PC. Crea la ventana, el motor de PC, y comienza la ejecución.

* Módulo “*app*”:

Módulo que contiene la actividad principal de la aplicación de Android. Al igual que en PC, crea la ventana, el motor de Android, y comienza la ejecución.

En ese mismo modulo también se controla la creación de sensores, la creación de anuncios y el request de las notificaciones periódicas con el uso de un Worker

# Arquitectura

La arquitectura de la aplicación se basa en los siguientes aspectos:

Del módulo “Interface”, heredarán los módulos de PC (LibEnginePC) y Android (LibEngineAndroid), cada uno sobreescribiendo los métodos con su propia API.

El módulo de lógica (LogicLib) se compartirá entre la aplicación de PC y Android, y no es necesario sobreescribir ningún tipo de comportamiento, ya que funciona en las dos plataformas por igual.

Por último, tendremos un módulo para lanzar la aplicación de PC, y otro módulo para lanzar la aplicación de Android.

# UML

La arquitectura a grandes rasgos está basada en la que se ha visto en clase.

<https://app.moqups.com/D8ersGczN1hWw9ZYvLenHHeTFLY1dDpg/view/page/a8bfb0d3e>

**\* No se han realizado partes opcionales.**