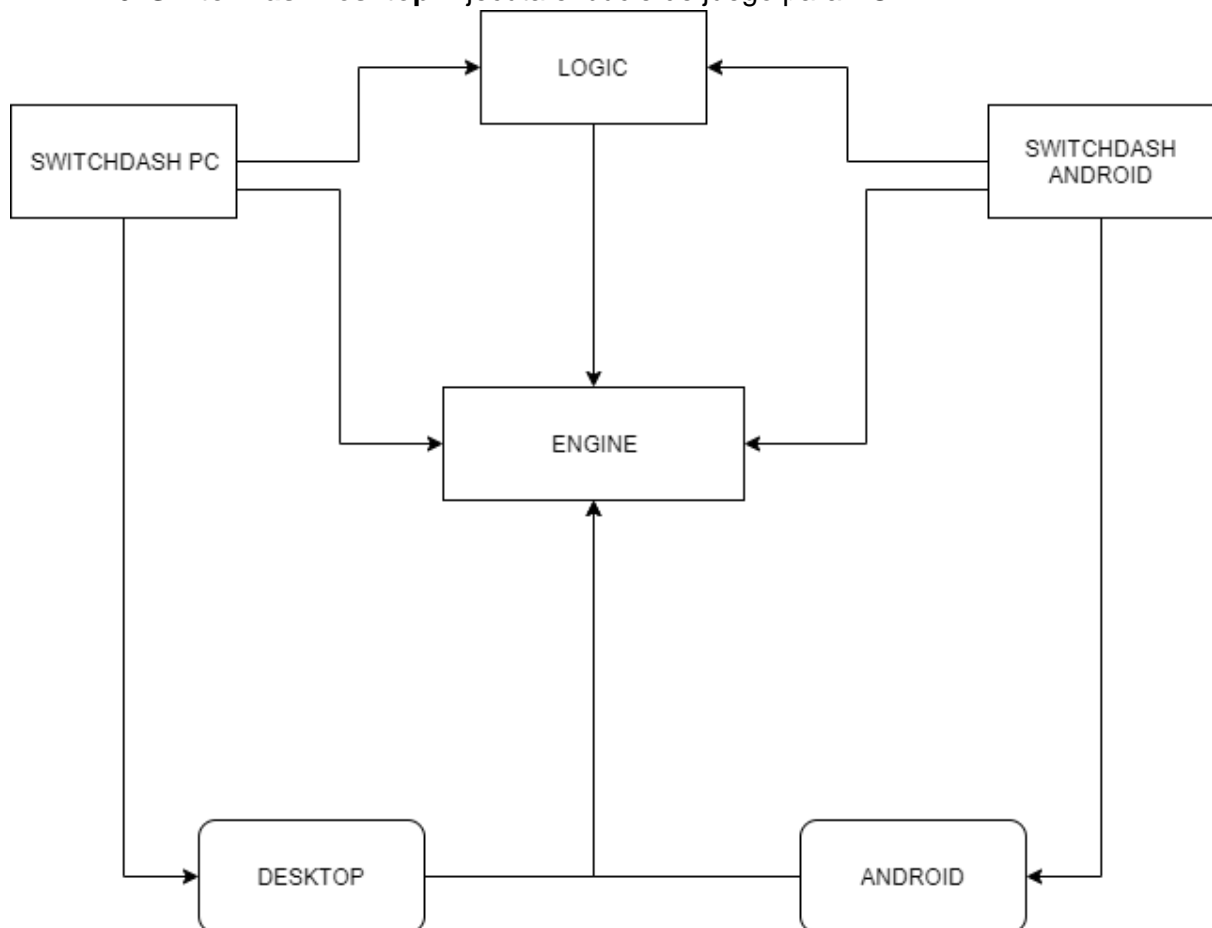


PRÁCTICA DE APLICACIONES MÓVILES SWITCH DASH

Alumnos: *Carlos Llames Arribas y Gonzalo Guzmán del Río*

La práctica contiene 6 módulos diferentes:

- 1: **EngineInterface:** Módulo que contiene las interfaces y métodos abstractos las/os cuáles serán implementados por el motor correspondiente a cada plataforma.
- 2: **Desktop:** Módulo encargado de implementar los/as las interfaces y métodos del motor, con código específico para ordenador.
- 3: **Android:** Módulo encargado de implementar los/as interfaces y métodos del motor, con código específico para ordenador.
- 4: **Logic:** Se encarga de implementar las interfaces del motor. (Lógica del juego, cambio de estados, botones...).
- 5: **SwitchDashAndroid:** Ejecuta el bucle de juego para Android.
- 6: **SwitchDashDesktop:** Ejecuta el bucle de juego para PC.



ENGINEINTERFACE:

- IGame**: Interfaz con diferentes métodos para la ejecución del juego.
- IGameState**: Interfaz para el bucle de un estado del juego.
- IGraphics**: Encargada de pintar
- Image**: Interfaz para la creación de imágenes.
- IInput**: Interfaz encargada de administrar los eventos del jugador(Pulsación sobre la pantalla, botón...)

DESKTOP:

- DGame**: Creación del juego para PC.
- DGraphics**: Pintado de imágenes en PC.
- DImage**: Creación de imágenes en PC.
- DInput**: Detección de eventos en PC.

ANDROID:

- AGame**: Creación del juego para Android.
- AGraphics**: Pintado de imágenes en Android.
- AImage**: Creación de imágenes en Android.
- AInput**: Detección de eventos en Android.

LOGIC:

-Managers:

- GameManager**: Encargado de gestionar diferentes aspectos como la velocidad, el color de fondo, la puntuación, o record.
- TextManager**: Clase que se encarga de crear textos(o puntos) a partir de un sprite font dado.

-States:

- GameState**: Clase Base para la gestión de eventos. En esta clase se incluye lo común a todos los estados, como el objeto Screen(más abajo).
- StartState**: Estado inicial del juego. De aquí se pasa al estado Help si es la primera vez que se juega. Si no, se pasa automáticamente al estado Play.
- HelpState**: Estado que te muestra cómo funciona el juego. De este estado se va al PlayState.
- PlayState**: Estado “principal” del juego. Este estado se mantiene mientras no falles una bola. Hay que poner la pala del color de la bola que cae. De este estado se pasa al GameOver, cuando el jugador falle.
- GameOverState**: Estado al que se pasa cuando el jugador pierde la partida. Muestra la puntuación y el record actual. De aquí puede pasar al help, start o

iniciar una nueva partida.

-Ball: Clase que renderiza una bola, con su lógica interna como el cambio de color, reseteo de posición...

-Button: Clase que representa un botón, con su posición XY, y su ancho y alto. Métodos para comprobar si hemos pinchado dentro.

-Particle: Clase que utiliza Ball para la representación de una partícula (alpha, ancho, y alto). Se “invoca” cuando el jugador destruye una bola.

-ParticleSystem: Clase encargada de la gestión de las partículas. Se crea siempre el mismo número de partículas, definido como constante.

-Player: Representación del jugador. Gestión del cambio de color del mismo.

-Resources: Clase para la gestión de recursos. Carga de todas las imágenes, con sus propiedades, y cálculo de lo que ocupa cada imagen.

-Screen: Clase utilizada para la representación física de la pantalla. El screen siempre renderizará y actualizará el fondo y las flechas.

-Sprite: Clase que “simboliza” una imagen con posición XY, y un ancho y alto calculados a partir de la imagen pasada como parámetro (cargada desde resources).

SWITCHDASHANDROID:

-SwitchDashAndroid: Clase encargada de lanzar el juego para Android.

SWITCHDASHDESKTOP:

-SwitchDashDesktop: Clase encargada de lanzar el juego para PC.