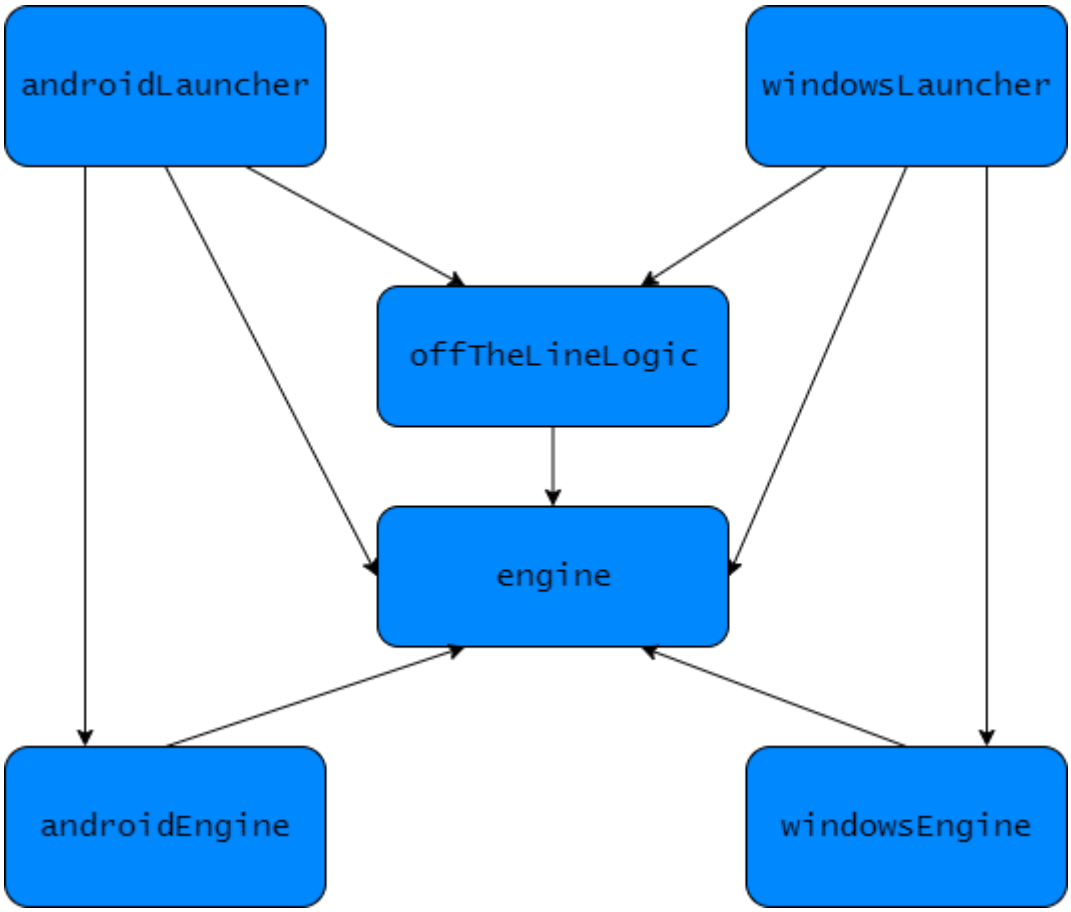
OFF THE LINE

VIDEOJUEGOS PARA DISPOSITIVOS MÓVILES: ANDROID STUDIO

GONZALO ALBA DURÁN Y NURIA BANGO IGLESIAS

Contenido

Dependencias



Módulos

Engine

Este módulo contiene el conjunto de interfaces y clases abstractas que serán implementadas en AndroidEngine y PCEngine. También contiene una interfaz llamada Logic que se implementará en la lógica, ya que el bucle del juego se encuentra dentro del motor.

- **Engine**: interfaz con los métodos necesarios para obtener el motor gráfico, el gestor del input, lectura de ficheros y establecer la lógica.
- **Graphics**: define los métodos descritos en la práctica y hemos añadido métodos para obtener el tamaño de la ventana, el de las bandas negras y el factor de escala. También se defien el método para cambiar la resolución lógica de la pantalla
- **AbstractGraphics**: es una clase abstracta que implementa el interfaz Graphics y desarrolla algunos de los métodos descritos en dicha interfaz. Los métodos son los encargados devolver el factor de escala, el tamaño lógico de la pantalla y el tamaño de las bandas negras. En esta clase es donde se hacen los cálculos para el escalado de la pantalla.
- **Font**: es una interfaz vacía que implementarán los motores de cada plataforma
- **Input**: interfaz que contiene los tipos de input y la clase TouchEvent donde se guarda el información del evento. También contiene el método de lista de eventos, sin implementar claro.
- **AbstractInput**: clase abstracta que implementa el interfaz Input y que contiene la lista de TouchEvents, el método que la devuelve y el método addEvent para añadir un evento. Estos métodos son synchronized ya que acceden y modifican la lista de eventos desde hebras diferentes, el get desde la logica y el add desde la hebra de input.
- **Logic**: interfaz que contiene tres métodos: init, update y render.

WindowsEngine

Contiene cuatro clases:

- **Engine:** Es la clase que implementa el interfaz Engine. Contiene dos métodos adicionales, uno para crear la ventana y otro es donde está el bucle del juego.
- **Font:** implementa el interfaz Font y contiene un objeto Font de java.awt.
- **Graphics:** hereda de la clase AbstactGraphics e implementa los métodos del interfaz.
- **Input:** hereda de AbstractInput. Contiene dos clases, una que implementa MouseListener y otra MouseMotionListener. En ellas se implementan los métodos necesarios para recoger eventos específicos y guardarlos en la lista de eventos de la clase AbstractInput.

AndroidEngine

Este módulo contiene cinco clases:

- **Engine:** Esta clase implementa el interfaz Engine y guarda la instancia del motor gráfico e input. Al contrario que en Windows, esta clase no contiene el bucle del juego, si no que hay una clase específica que lo contiene, y es el Engine el que guarda la

instancia de esa clase. También contiene la surfaceview y los métodos resume y pause que llaman al resume y pause de la clase MainLoop (runnable).

- **MainLoop:** Es la clase que implementa la interfaz Runnable y contiene tres métodos.
    - **Resume**: que sirve para solicitar que se continue con el active rendering, es decir, para volver a poner en marcha el juego (o poner en marcha por primera vez).
    - **Pause**: que se llama cuando el active rendering tiene que ser detenido
    - **Run:** Ejecuta el bucle del juego, entre otras cosas. Establece el canvas y el factor de escala.
- **Font:** Implementa el interfaz Font del módulo Engine y en el constructor se crea la fuente con los parámetros especificados.
- **Graphics:** Hereda de AbstractGrafics, contiene el canvas, el paint y el assetmanager, además de los métodos de la interfaz.
- **Input:** Hereda de la clase AbstractInput y contiene la clase TouchEvents que implementa OnTouchListener que se encarga de captar los eventos e introducirlos en la la lista de eventos.

AndroidLauncher

Es la clase encargada de crear la actividad en Android y donde se crea el Engine y la lógica del juego. Hereda de AppCompatActivity e implementa los métodos onCreate, onResume y onPause para crear, reanudar y pausar la aplicación. Esta clase entra en el bloque de la lógica del juego, ya que es donde se crea.

WindowsLauncher

Es el lanzador de Windows. Es una clase con un main donde se crea el Engine y la lógica del juego y donde se llama al método run de engine que contiene el bucle del juego.

Lógica

Contiene la lógica del juego Off The Line. Tiene las siguientes clases:

- **OffTheLineLogic:** Es la clase de entrada donde se establece el tamaño lógico de la pantalla y se crean los diferentes “estados” del juego. Gestiona el cambio de estados o escenas del juego.
- **StartMenu y Game:** Son clases que implementan el interfaz Logic, por lo tanto desarrollan los métodos init, update y render. En StartMenu crea los diferentes títulos del menú de inicio y los botones de de dificultad.

  En Game se crean los objetos necesarios para leer el .json de los niveles y los guarda en una lista. También crea al player, lleva la cuenta del nivel en el que estamos y gestiona el final del juego.
- **JSONReader:** Es la clase encargada de leer el .json y crear los objetos del juego en función de los datos leídos.
- **Level:** Es el objeto que contiene la información del nivel. Tiene una lista de enemigos, ítems y paths. En esta clase es donde se llama al update y render de los objetos de las listas.
- **Item, Enemy, Path y Player:** estas clases contienen la información sobre su posicion, rotacion, traslación y color. El player también comprueba las colisiones con el resto de elementos y gestiona la logica del salto al recibir una pulsación.

- **Utils:** Contiene dos clases, point y vector, y dos métodos estáticos que se utilizan para detectar las colisiones en el player.

Tareas pendientes

Nos falta la lógica matemática para hallar la posicion en cada momento del enemigo para poder detectar la colisión del player con él. La infraestructura está pero nos faltan los calculos porque no hemos sido capaz de sacarlos. Lo mismo pasa en el NIVEL 3 – ORBIT, los ítems son colocados y luego rotados. Para detectar la colision con ellos hay que calcular los puntos a mano.