

Frogger

Daniel Ramirez Martinez

3 de noviembre de 2011

Introducción

Implementación mediocre hecha en Java del juego 'Frogger' de Konami para Atari 2600.

Implementación

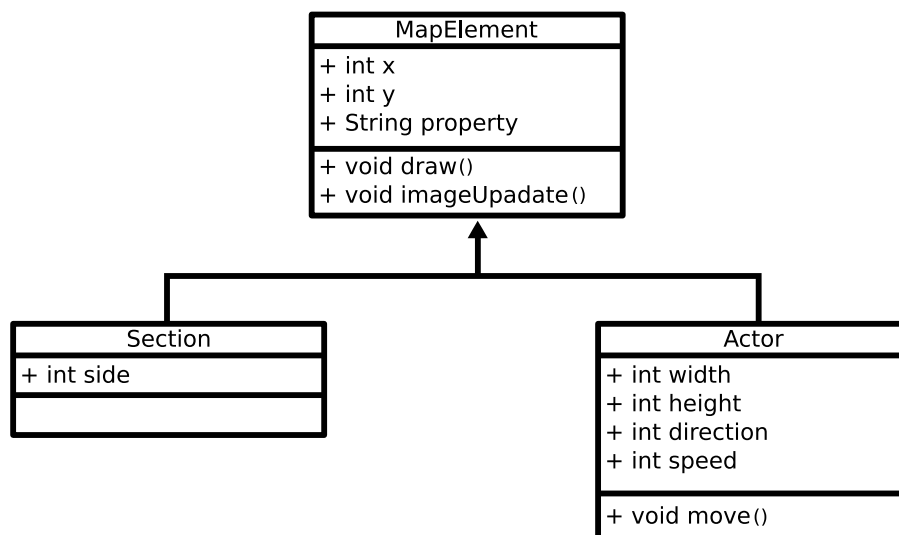
Mapa

El juego consta de un entorno, que a su vez está compuesto por un montón de elementos que decenden de la superclase abstracta 'MapElement'.

Las secciones (clase 'Section') es una subclase abstracta de 'MapElement' que determina a los objetos estáticos del entorno.

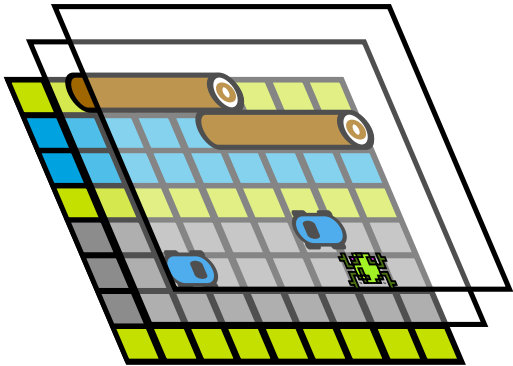
Los actores (clase 'Actors') es una subclase abstracta de 'MapElement' que determina a los elementos dinámicos del mapa.

A continuación se muestra el diagrama de clases.



El orden jerarquico de los elementos del entorno esta dado en orden ascendente como sigue:

- Terreno
- Actores
- Jugador



El terreno es un arreglo bidimensional de secciones cuadradas, que yace bajo las demas capas.

Los actores son un arreglo unidimensional de listas, la posición en el arreglo determina la coordenada en y de el mapa a la que se asocia una lista de actores.

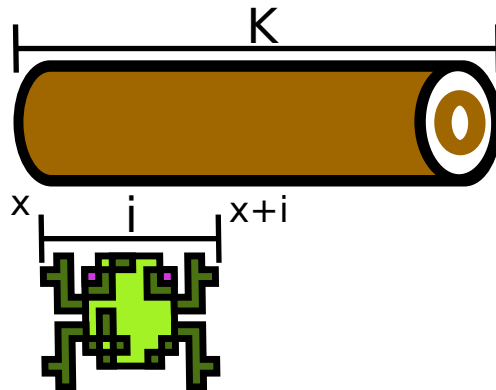
El jugador simplemente esta sobre todas las demas capas.

Detección de colisiones

La rana es capaz de moverse solo sobre casillas, es decir que siempre avanzara una casilla completa por lo que la colisión mientras esta sobre terreno es muy sencilla ya que solo hay que verificar la propiedad de la casilla que le precede al jugador antes de realizar el movimiento. Por otro caso cuando el jugador esta sobre un actor el movimiento de la rana ya no es exacto, ya que los actores son capaces de moverse libremente.

La función de chequeo de colisión en la clase 'Map' verifica si el arreglo de actores tiene una lista en la posición correspondiente a la nueva coordenada 'Y' de la rana. En caso de que si la posea, entonces procedera a verificar sobre todos los elementos de dicha lista:

- si x esta dentro de $K + \langle \text{alguna tolerancia} \rangle$
- si $x + i$ esta dentro $K - \langle \text{alguna tolerancia} \rangle$



La función realizara este chequeo para todos los elementos de la lista a excepción de que en efecto colisione, entonces retornara en ese elemento con algún código de salida.

En caso de que el arreglo de actores no posea una lista en la posición correspondiente a la nueva coordenada 'Y' de la rana, entonces retornara inmediatamente con el estado de no colisión y se verificara el elemento del terreno en la casilla de terreno correspondiente.

Cuando un actor esta sobre otro este adopta su velocidad, por lo que cuando la rana sube sobre un tronco esta se mueve con el.

Juego

El juego esta regido por un 'Timer' que realiza la acción cada 90 milisegundos.

El jugador adquiere:

- 10 puntos: por cada casilla que suba
- 100 puntos: por tomar una rana bonus
- 200 puntos: por cada rana salvada (que llegue a la meta).

Por cada 1000 puntos o cada rana salvada el jugador adquiere mas tiempo de juego.

Si se acaba el tiempo de juego el jugador pierde una vida.

Problemas

Sonido

Culpen a open-jdk-1.6.0_22, jdk-1.7 y a mis escasos conocimientos sobre sonido en java (y en todos los lenguajes).

Eventos con el teclado

no fue posible restringir el abuso en el movimiento de la rana por parte del usuario.

Gestión de tiempo

No se pudo implementar un contador para el tiempo de juego basado en una constante de tiempo real. Todo era relativo al 'Timer'.