

Práctica 2: Scala Bloques Mayas (Antique Blocks)

PARADIGMAS AVANZADOS DE PROGRAMACIÓN

Práctica 2: Scala

Bloques Mayas (Antique Blocks)

Dinámica del juego “Bloques Mayas”

El tablero es una malla formada por cuadrados (“bloques”) que inicialmente cubren todo el tablero

Los bloques pueden tomar 8 estados, que son relacionados con 8 colores.

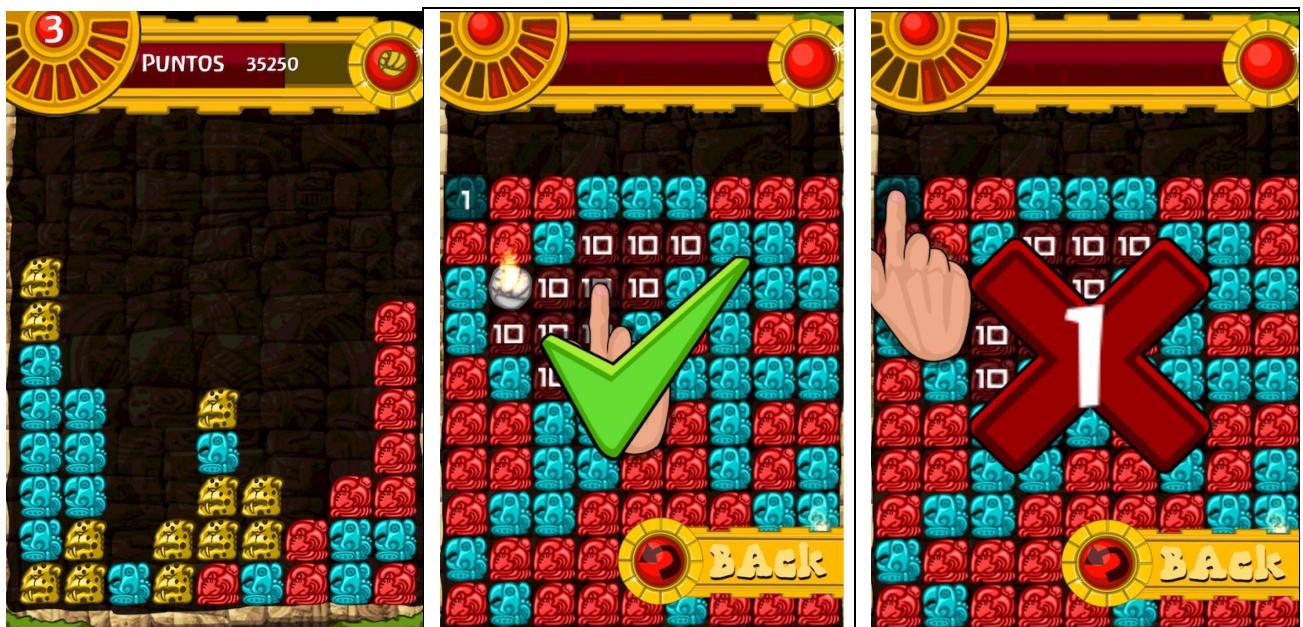
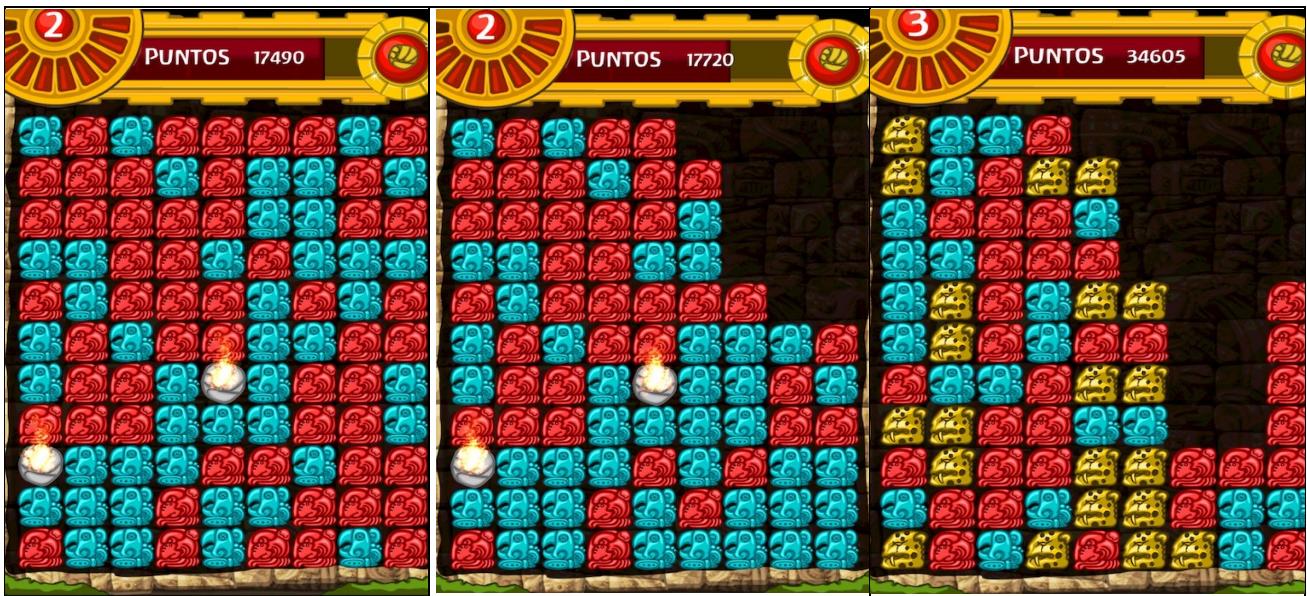
El estado 8 es una bomba que permite eliminar todos los bloques que están alrededor incluidas las diagonales

Cada bloque tiene 4 bloques vecinas, que son las estás próximas a ella, sin incluir las diagonales.

Las transiciones dependen del número de bloques vecinos:



- Un bloque pulsado eliminará todos los bloques vecinos similares si están formados por 3 o mas bloques iguales (al turno siguiente los huecos vacíos serán cubiertos por los bloques que están por encima y a la izquierda).
- Un Bloque bomba permitirá la eliminación de todos los bloques vecinos incluidos sus diagonales (al turno siguiente los huecos vacíos serán cubiertos por los bloques que están por encima y a la izquierda).



Figuras 1-6

El Objetivo de la práctica.

Esta práctica tiene por objetivo desarrollar una aplicación en lenguaje funcional con Scala.

Scala es un lenguaje de programación general diseñado para expresar patrones de programación comunes de manera concisa, elegante y segura. Destaca el uso conjunto del paradigma orientado a objetos y el funcional. Sus principales ventajas son la integración perfecta con java y la seguridad de su compilador.

Para ello, se debe implementar el juego de “Bloques Mayas” en Scala, en el que como ya es por todos conocido, es un juego que se lleva a cabo por un jugador o por el propio ordenador.

Trabajo a Realizar – Obligatorio (hasta 8 puntos)

Consiste en alinear tres o más “bloques” de igual color de forma vertical, horizontal.

El juego dispondrá de tres niveles de dificultad:

- a. El nivel 1 del juego otorgará 8 vidas, y mostrará tableros de 9x11 en el que se presentará un tablero lleno de bloques de hasta 3 colores aleatorios y 2 bombas ubicadas aleatoriamente en él.
- b. El nivel 2 del juego otorgará 10 vidas, y mostrará tableros de 12x16 en el que se presentará un tablero lleno de bloques de hasta 5 colores aleatorios y 3 bombas ubicadas aleatoriamente en él.
- c. El nivel 3 del juego otorgará 15 vidas, y mostrará tableros de 25x15 en el que se presentará un tablero lleno de bloques de hasta 7 colores aleatorios y 5 bombas ubicadas aleatoriamente en él.

Las bloques pueden ser de ocho colores que se identifican a continuación:

- a. Glifo “1” representa el color Azul
- b. Glifo “2” representa el color Rojo
- c. Glifo “3” representa el color Naranja
- d. Glifo “4” representa el color Verde
- e. Glifo “5” representa el color Plata
- f. Glifo “6” representa el color Morado
- g. Glifo “7” representa el color Gris
- h. Glifo “8” representa el color Blanco (bomba)

por pantalla se mostraran identificando cada bloque en el tablero como A, R, N, V, P, M, G, B.

Funcionamiento del Juego

1. La dificultad del juego será indicado por el usuario a través del teclado.
2. Una vez presentado el tablero inicial se debe buscar una estrategia ganadora, mediante la elección de un “bloques” posición (x,y) que tenga tres o más bloques alrededor.
3. Cada vez que se consiguen alineaciones de tres fichas o más estas se eliminan, ya sea horizontal o vertical, en estos casos los bloques que estén por encima caerán hasta completar las casillas vacías y se desplazarán a la izquierda.
4. Hay que llevar el conteo de los bloques que se han eliminado. Cada bloque eliminado en grupo de 3 o más, se consiguen 10 puntos por cada bloque, a modo de ejemplo cada línea horizontal o vertical de 3 bloques puntúa 30 puntos. Cada bloque eliminado de forma individual dejará a cero el conteo de la partida.(ver figura 5)
5. Cada bloque eliminado de forma individual restara una vida. (ver figura 6)
6. En el caso de seleccionar una posición imposible esta será comunicada mediante un mensaje “Posición Imposible” y esperará a proponer un movimiento correcto.
7. Al final de cada partida el programa indicará los puntos conseguidos.
8. Este proceso (pasos 2-6) se repetirán hasta que finalice el tiempo de juego asignado, indicando la puntuación total obtenida y el número de partidas jugadas.

Trabajo de Optimización a Realizar (hasta 4 puntos)

Consiste en implementar un algoritmo que facilite la opción de juego más óptima al computador en cada jugada que realice. (**hasta 4 puntos**)

Trabajo Avanzado a realizar (4 puntos)

A partir del código obtenido en la optimización del juego en el apartado anterior, implementar mediante Colecciones Paralelas de Scala las modificaciones necesarias para su correcto funcionamiento, analizando previamente los problemas que plantea su uso (**hasta 4 puntos**).

Para la obtención de esta puntuación, será necesario haber realizado la parte anterior de optimización.

Trabajo a Entregar

Los alumnos deben entregar las siguientes fases de la práctica:

Desarrollo obligatorio de la práctica mediante listas (**este apartado es obligatorio**)

Desarrollo e implementación de un algoritmo de optimización en la práctica mediante Listas y funciones de orden superior, etc

Desarrollo avanzando de la práctica

Todas las entregas deben llevar asociadas el código del programa realizado, un txt con los componentes del grupo (2 alumnos máximo) y un video individual donde se evidencie la autoría, se explique el problema y la solución adoptada, y se describa el funcionamiento del programa, así como una explicación de los problemas encontrados y como se han resuelto.

Defensa de las prácticas y examen del Laboratorio – Obligatorio – (Hasta 16 puntos)

La defensa de las prácticas será en la fecha y la forma indicada por el profesor del laboratorio, pudiendo ser esta oral o escrita, en el caso de no contestarse correctamente a las cuestiones presentadas, la práctica presentada podrá considerarse como suspensa.

La nota máxima a la que puede acceder en cada apartado se pondrá en función de:

- a) Calidad del material entregado por el alumno.
- b) Documentación de seguimiento presentada durante su realización
- c) Realización e implementación de la práctica.
- d) Defensa e implementación de las modificaciones solicitadas
- e) Manejo del software utilizado para el desarrollo de la práctica.
- f) Código desarrollado.

Notas:

Se debe tener en cuenta que para la realización de los programas en Scala, **se deben usar vals para la definición de variables, objetos inmutables y métodos sin efectos colaterales.**

Todo el código entregado deberá estar perfectamente comentado. Aquella práctica que no tenga el código bien documentado puede ser suspendida.

Se valorará positivamente el código bien estructurado y óptimo

La fecha tope para su entrega es el 19 de abril de 2022 a las 23:59 horas con el fin de que el profesor de la asignatura pueda preparar la prueba para su defensa.

Bibliografía

1. **Odersky, Spoon & Venners, "Programming in Scala – A Comprehensive Step-by-step Guide", Artima, 2008**
2. Wampler & Payne, "Programming Scala", O'Reilly, 2009
3. Odersky, M, "Scala By Example", Programming Methods Laboratory – EPFL Switzerland
4. Maya Antique Blocks (Android)
<https://www.youtube.com/watch?v=cEV0zTg3BeM>