

TALLER DE LENGUAJES II

ENTREGA MÓDULO I: BOULDER DASH

IMPORTANTE:

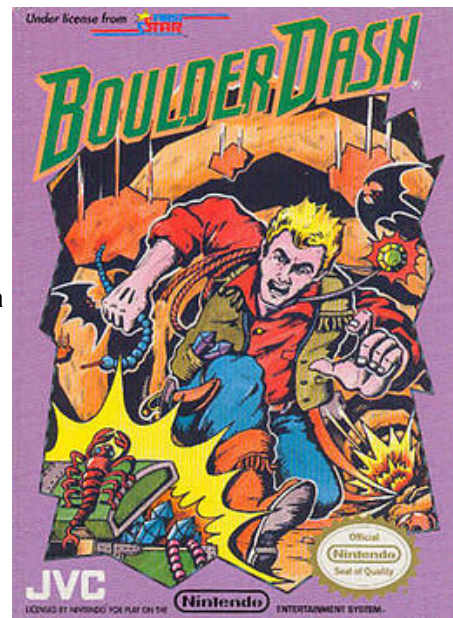
- ❑ **ÚLTIMO DÍA DE RECEPCIÓN DE TRABAJOS: 23/10/2017.**
- ❑ **En esta entrega NO se contempla lo correspondiente a visualización gráfica, ni expiración de tiempo en las jugadas.**
- ❑ **El material adicional entregado no deberá ser alterado!**

Boulder Dash es un juego de computadora creado por Peter Liepa y Chris Gray durante principios de los años 80 y luego adquirido por First Star Software.

Boulder Dash fue originalmente desarrollado para computadoras Atari, y rápidamente se convirtió en un juego para computadoras como Apple II, Commodore 64, etc.

El protagonista de Boulder Dash es Rockford, el escenario es una cueva. Rockford debe cavar para encontrar diamantes en un tiempo limitado, lo que le habilitará una puerta de salida al próximo nivel. En su recorrido, Rockford debe evitar obstáculos, criaturas peligrosas, terminar aplastado por una roca o envuelto en una explosión.

¡ Que se divierta !



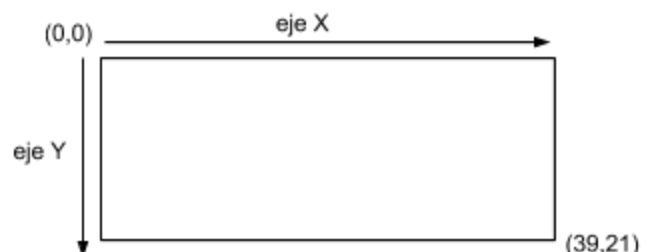
A. LAS REGLAS

1. Del Juego en sí

- Cada nivel es completado con éxito cuando el personaje principal, Rockford, logra alcanzar la puerta de salida. La puerta de salida sólo será habilitada si Rockford encuentra una cantidad determinada de diamantes que varía según el nivel.
- Cada nivel puede jugarse empleando un máximo de 4 vidas.

2. El escenario

Los personajes se desenvuelven en un tablero de juego de 40 x 22 casilleros, de acuerdo a la disposición que se vé en el gráfico.



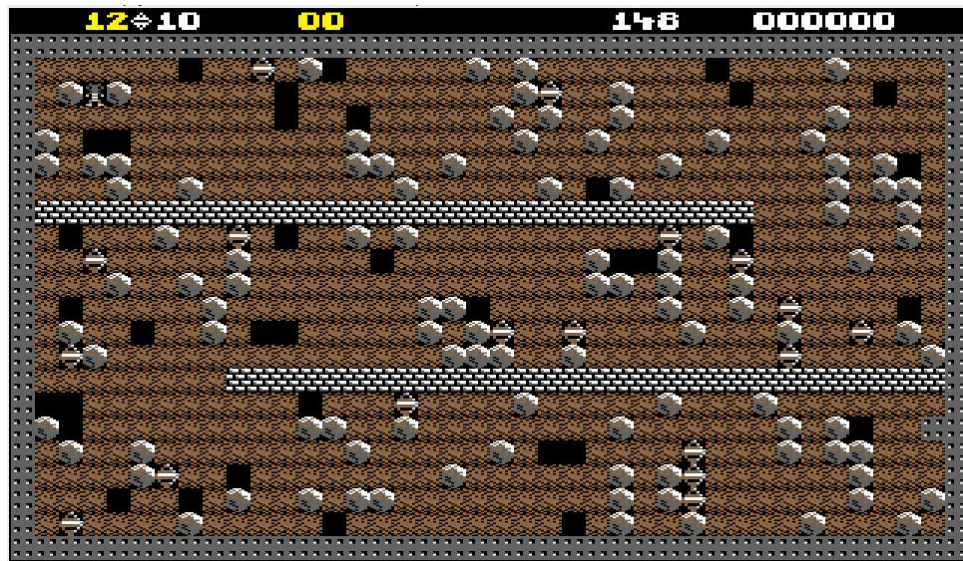
El tablero es un conjunto de casilleros donde Rockford puede moverse. Un casillero del tablero puede estar ocupado por los siguientes personajes cuyo comportamiento se encuentra detallado en el punto 3:

- Rockford
- Una roca
- Un diamante
- Una mariposa
- Una luciérnaga
- Una ameba
- Suciedad
- Un muro común, de titanio o mágico
- Una puerta de salida
- o simplemente estar vacío

Los casilleros con rocas o muros no son accesibles para Rockford.

Los casilleros con suciedad o vacíos **SI** son accesibles para Rockford, y de hecho el mismo hará su camino a través de estos.

A continuación una imagen general de cómo se visualiza el juego en el nivel 1.



3. Los Personajes



Rockford: es el personaje principal, debe recorrer el laberinto tratando de juntar una cantidad mínima de diamantes que le habiliten la puerta de salida al próximo nivel. Rockford puede escarbar en la basura así como empujar las rocas, las cuales se desplazarán sólo si el casillero contiguo está vacío.



Rocas (Rocks o Boulder): Cada roca puede estar en uno de dos estados: “estacionario” o “cayendo”. Si la roca está en estado “estacionario” y el casillero (x,y+1) justo abajo está...

1. vacío: la roca se convierte en una roca “cayendo”. (Tener en cuenta que aún no se mueve)
2. muro, roca o diamante: si los casilleros $(x-1,y)$ y $(x-1,y+1)$ inmediatamente a la izquierda y abajo-izquierda están ambos vacíos, la roca “se desliza” al casillero $(x-1, y)$ a la izquierda y se convierte en una roca “cayendo” (La regla también aplica si consideramos el lado derecho).
3. cualquier otra cosa: la roca permanece “estacionaria”.

Si la roca está actualmente “cayendo” y el casillero justo debajo es....

1. vacío: la roca se mueve al casillero que se encontraba abajo de la misma.
2. jugador (Rockford): el jugador explota y muere.
3. luciérnaga: la luciérnaga explota y cualquier cosa que se encuentre en un área de $3*3$ que no sea un muro, se transforma en un espacio vacío. Las explosiones NO continúan recursivamente aún si hay otra luciérnaga o mariposa en el área de la explosión.
4. mariposa: la mariposa explota y cualquier cosa que se encuentre en un área de $3*3$ que no sea un muro se convierte en diamante. (Las explosiones NO continúan recursivamente.)
5. cualquier otra cosa: la roca se vuelve “estacionaria”.



Diamantes (Diamond): Los diamantes se comportan de la misma manera que las rocas, de modo que cada diamante se encuentra “estacionario” o “cayendo”, excepto que el diamante caiga en el jugador (Rockford), luciérnaga o mariposa es inofensivo y simplemente termina en un estado “estacionario”.



Luciérnaga (Firefly): Si el jugador se encuentra en uno de los cuatro casilleros vecinos adyacentes a la luciérnaga (arriba abajo, izquierda o derecha), la luciérnaga explota convirtiendo un área de $3*3$ en espacio vacío, y el jugador muere. De otro modo, la luciérnaga trata de moverse. Cada luciérnaga tiene una “dirección actual” (arriba abajo, izquierda o derecha). Si el siguiente casillero en su dirección está vacío, la luciérnaga se mueve allí. Si el siguiente casillero no está vacío, la luciérnaga cambia su dirección de acuerdo a las agujas del reloj, por ejemplo: si la dirección actual es “arriba”, la nueva dirección será a la “derecha”. (La luciérnaga no trata de moverse en la nueva dirección hasta que le toca nuevamente el turno.)



Mariposa (Butterfly): Las mariposas se comportan de la misma manera que las luciérnagas, excepto que se mueven en sentido contrario a las agujas del reloj cuando su intento de movimiento se encuentra bloqueado por algo.



Ameba (Amoeba): La ameba no se mueve y su toque es inofensivo para cualquiera, pero esta lentamente se va expandiendo por los casilleros vacíos, haciendo eventualmente imposible moverse al jugador. En cada turno, la ameba tiene “pocas probabilidades” de expandirse. La expansión busca un casillero vecino en una dirección elegida de manera random, y si ese casillero está vacío o sucio, se convierte en una ameba.



Muro de Titanio (Titanium Wall): Un muro indestructible.



Muro Mágico (Magic wall): Es un tipo especial de muro que permite convertir rocas en diamantes y viceversa sólo durante un tiempo determinado.



Basura (Dirt): Rockford escarba a través de la basura para dejar espacios vacíos.



Puerta de Salida (EXIT): Ésta se habilita cuando Rockford pudo recolectar una determinada cantidad de diamantes.

Notas Finales

Para tener una idea intuitiva acerca de cómo funciona el juego, pruebe jugarlo accediendo a: <http://codeincomplete.com/projects/boulderdash/>. En esa página tiene todos los niveles disponibles.

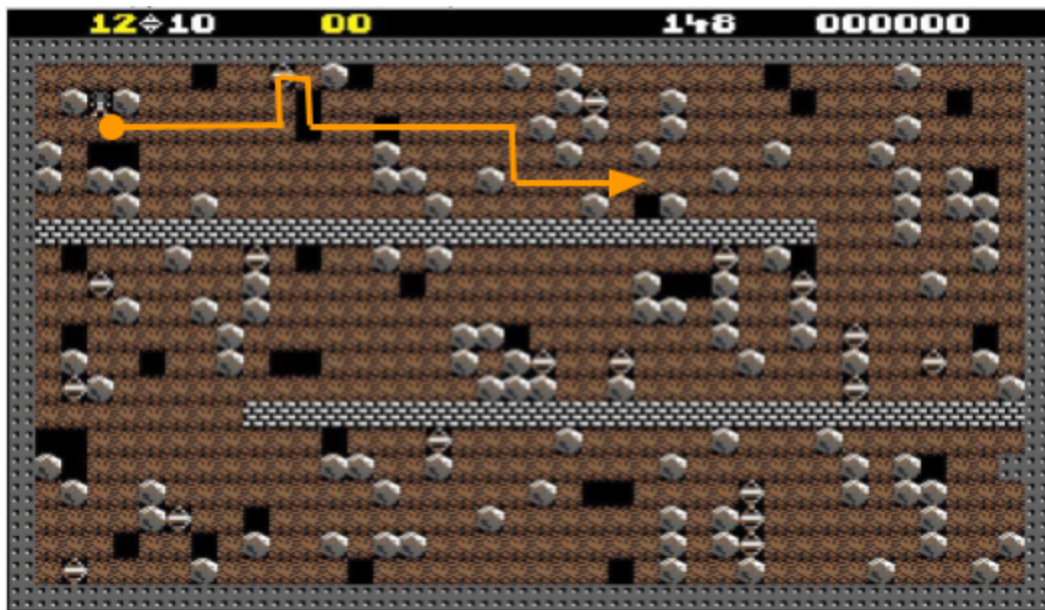
B. IMPLEMENTACIÓN

Para esta entrega deberá completar el **Nivel 1 y Nivel 2** de Boulder Dash. La solución deberá ser lo suficientemente flexible para posteriormente permitir la habilitación de otros niveles.

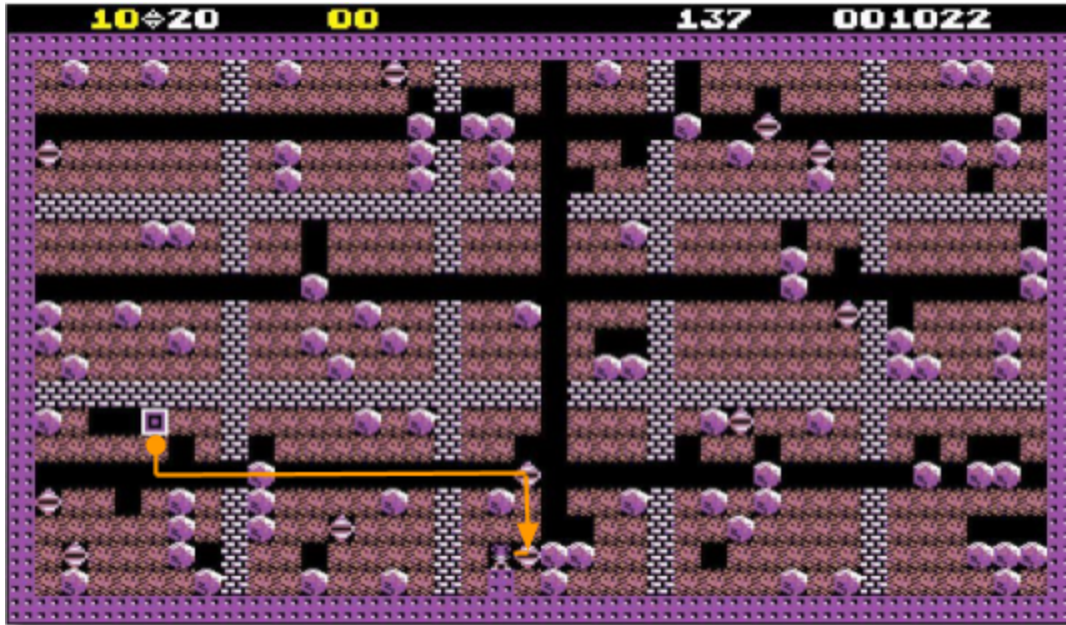
C. TEST DE LA APLICACIÓN

Si bien la aplicación no tiene aún interfaz gráfica, deberá emitir una salida por pantalla de lo que va ocurriendo en el juego. Considerando que aún el jugador no puede mover mediante el teclado a Rockford, éste personaje deberá realizar las rutas indicadas en los gráficos. Durante su camino, **cada elemento que realiza una acción deberá imprimirlo en pantalla.**

CASO NIVEL 1



CASO NIVEL 2



D. MATERIAL ADICIONAL

Se encuentra a disposición un archivo llamado mapa.rar que contiene un paquete java llamado “mapa” donde se provee la disposición original de los elementos en el mapa según los distintos niveles de juego. Este paquete consiste de 3 archivos:

- un archivo XML descriptor del mapa,
- un tipo de datos enumerativo que representa a las componentes del mapa
- una clase que se encarga de leer el archivo XML y que responde consultas acerca de la distribución en el mapa. Esta clase además posee un método “main” que le servirá de ejemplo de uso de esta clase.

ATENCIÓN: Estos archivos NO deberán modificarse

E. CRITERIO DE EVALUACIÓN

Se establece una calificación máxima de 100. A continuación las pautas a considerar en la calificación de la entrega:

- a. Codificación estándar
 - Escritura de código de acuerdo a convención y usando buenas prácticas de programación en Java.
 - El código debe estar debidamente documentado, con comentarios explicativos acerca de la funcionalidad realizada.
- b. Documentación

- Adjuntar el diagrama de clases UML. En este punto, no es necesario que estén completos TODOS los métodos de cada clase, SI es importante que se señalen en el diagrama los métodos representativos del comportamiento y las relaciones entre las clases. El diagrama UML debe mostrar un diseño orientado a objetos, indicando clases y responsabilidades. **Importante:** No se aceptará un documento UML generado a partir de herramientas de generación automática a partir de código Java.
 - Entregar archivos en formato HTML como documentación respaldo de las clases creadas.
- c. Correcto Funcionamiento
- d. Entrega en formato .jar **incluyendo el código fuente** y el compilado. El archivo .jar indicará como clase principal a la clase donde se realiza el Test del juego.
- e. **IMPORTANTE : Èl còdigo entregado debe ser compatible con Java 1.7.**