

## TALLER DE LENGUAJES II

### ENTREGA MÓDULO II: BOULDER DASH

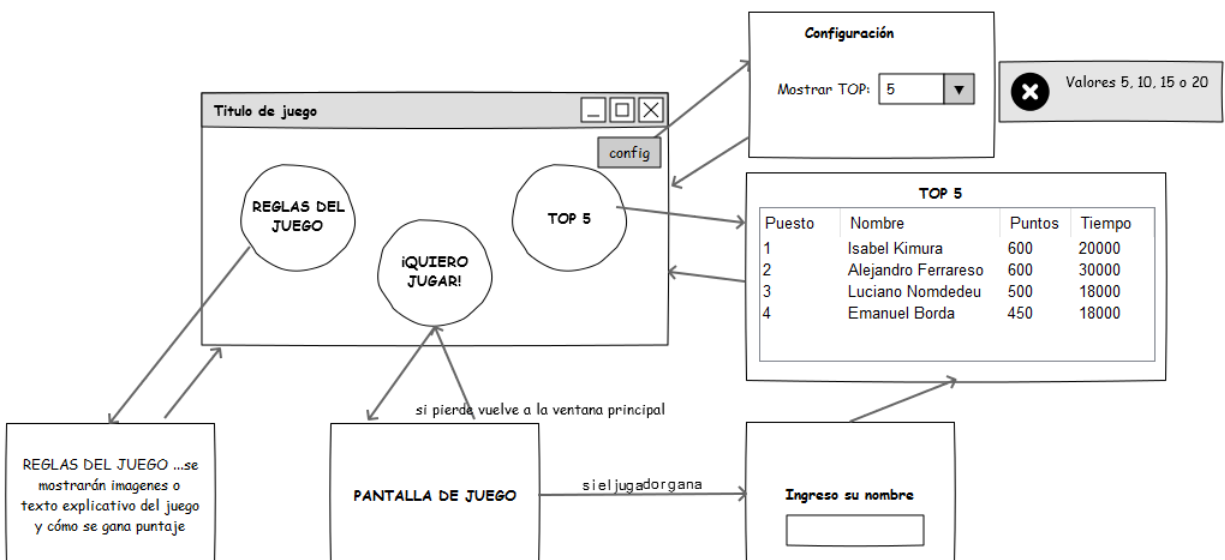
#### IMPORTANTE:

- ❑ **ÚLTIMO DÍA DE RECEPCIÓN DE TRABAJOS: 04/12/2017.**
- ❑ **En esta entrega además se entregan las correcciones del Módulo I.**
- ❑ **Para el diseño de las ventanas se sugiere que se realicen SIN el uso de plugins, sin embargo, si lo desea puede usar cualquiera de los plugins disponibles para eclipse:**
  - **Windows Builder** (<https://eclipse.org/windowbuilder/>)
  - **Jigloo:**  
<http://marketplace.eclipse.org/content/jigloo-swtswing-gui-builder#.VF3FD5G9hs>
- ❑ **Tendrá a disposición un conjunto de imágenes que puede o no usar en la entrega.**

Considerando que el modelo del Boulder Dash ya se encuentra resuelto, los cambios en el mismo deberían ser mínimos. Esta entrega se enfoca en la visualización gráfica del juego y la expiración por tiempo del juego.

## 1. DESCRIPCIÓN DE LA APLICACIÓN

El prototipo que se muestra a continuación contiene la ventanas que aparecerán en la aplicación Boulder Dash, así como el flujo entre ellas.



1. **La ventana principal** cuenta con 4 opciones: Reglas del Juego, ¡Quiero Jugar!, Top X y Configuración

2. La ventana **Reglas del Juego** muestra información acerca de cómo jugar, los personajes y sus reglas. Puede ser tan sencillo como una ventana que muestra un texto que explica cómo se comporta cada personaje que participa en el juego (puede copiar la información del enunciado del módulo I) ó ser más gráfica.

Cualquier información adicional (gráficos o consejos para el juego) es bienvenida.

3. La ventana **Top X** muestra una grilla con 4 columnas: posición en el ranking, nombre del jugador y puntaje obtenido y tiempo realizado, ordenado por puntaje de manera descendente y de haber empate, desempata por tiempo ascendente. Esta información además debe ser persistida en un archivo auxiliar, de modo que al ejecutar la aplicación se mantengan los mejores puntajes de la ejecución anterior.

Llamamos a la ventana **TOP X** porque el X depende de los parámetros de configuración, por ejemplo: mostrará TOP 5 si están configurados como máximo 5 resultados.

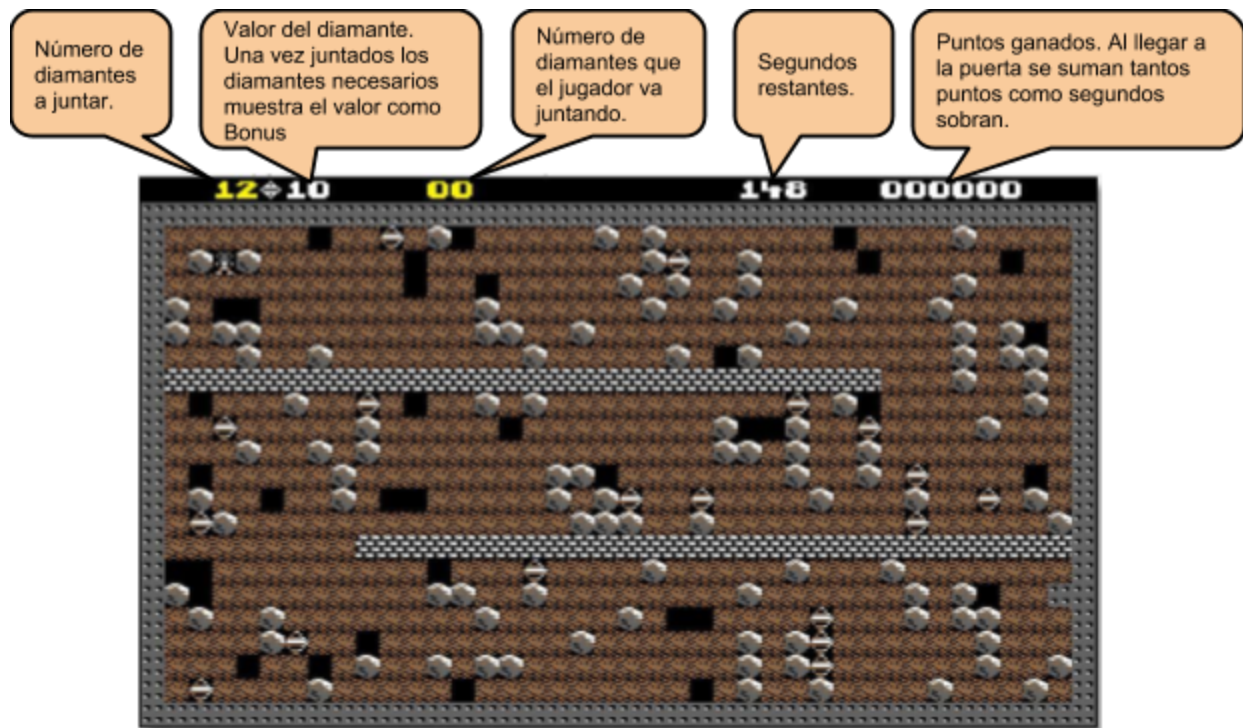
4. La ventana que se muestra cuando un usuario hace click en **¡Quiero Jugar!** es la aplicación principal. Cuando el jugador logra un puntaje que merece estar en el TOP X, se abre otra ventana donde debe ingresar el nombre del jugador. **Debe validar que el nombre del jugador sea válido:** el nombre no puede estar vacío y no puede tener menos de 2 caracteres. Si tiene más de 20 caracteres, se acepta el nombre pero se trunca a los primeros 20 caracteres. Además el nombre no puede contener espacios en blanco Si el nombre del jugador ya existe, ud. puede elegir entre 2 opciones: sobrescribir el resultado, o avisarle al jugador y permitirle ingresar otro nombre.

5. La ventana **Configuración** permite indicar cuántos resultados se mostrarán en el TOP X . Note que es un selector entre 4 posibles valores.

Adicional sólo para equipos de más de 2 integrantes:

-agregar a la ventana de configuración la opción “seleccionar nivel”, que permita indicar el nivel en el que comienza el juego. Ese valor debe persistirse y cuando el juego inicia, lo hace en ese nivel.

## 2. Detalle del Tablero de Juego



### Información de valores

Nivel	Diamantes requeridos	Valor Diamante	Valor Diamante Bonus	Tiempo (seg)
1	12	10	15	110
2	12	20	50	110
3	23	15	0	100
4	36	5	20	100
5	6	30	0	100
6	5	50	90	120

Para el resto de los niveles puede establecer ud. los valores, o tomarlos de:

[http://www.boulder-dash.nl/download/maps/PeterLiepa/BoulderDash01\\_level2.html](http://www.boulder-dash.nl/download/maps/PeterLiepa/BoulderDash01_level2.html)

### 3. ACERCA DE LA OPORTUNIDAD DE JUEGO DE CADA PERSONAJE

- ❑ Cada ronda de juego debería ejecutarse cada X veces por segundo aproximadamente, donde X es un número constante entre 10 y 20.
- ❑ Tenga en cuenta que Rockford, al igual que el resto de los personajes se mueve 1 vez por ronda.

### 4. CRITERIO DE EVALUACIÓN

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

- UML del modelo (no se pide incluir la parte gráfica).
- Debe estar claramente separado lo que corresponde a la Vista y al Modelo (entregado en el Módulo I). Esto permitirá desarrollar la aplicación respetando lo que se conoce como el patrón MVC (Model-View-Controller). Se permitirá que la Vista LEA información del Modelo, pero NO que actualice el Modelo, en este último caso deberá pasar por el Controller.
- Demuestre el uso correcto de casos excepcionales (uso y tratamiento de excepciones en el código). Debe crear al menos **2 excepciones propias** que considere condiciones excepcionales en el juego.
- Demuestre el uso correcto de clases que permitan la concurrencia en Java, en particular para el manejo de tiempos.
- La entrega será un **archivo con extensión jar ejecutable desde la línea de comando** y un archivo comprimido con su proyecto donde se encuentre el código fuente.
- **Recuerde que el código debe compilar en Java 7**
- **En caso de realizar trabajo adicional (gráfica, animación, sonido ó mejoras interesantes) la entrega podrá conseguir una puntuación de 110.**

#### IMPORTANTE SOBRE LA RE-ENTREGA MÓDULO I

Esta entrega contempla además la re-entrega del MÓDULO I. En caso de realizar correcciones sobre la entrega del MÓDULO I, indicar en un archivo de texto sobre las correcciones indicadas por el ayudante, cuales de estas correcciones fueron realizadas. Si realiza cualquier otra mejora que ud. considere hace que el código sea más orientado a objetos, más legible o más completo, por favor déjela también anotada y será verificado.

**Recuerde que en la re-entrega del MÓDULO I SOLO será revisado lo que ud. indique.**

Por otro lado, no es necesario que realice la salida en pantalla, la calificación de ese ítem se proporcional a la salida en pantalla que ud. realice en la entrega Módulo II.