

**Programación Orientada a Objetos**  
**Proyecto del Segundo Parcial**  
**Segundo Término 2017-2018**

## Bomb it!

Su tarea consiste en programar el juego **Bomb it** que consiste en una adaptación del tradicional juego Bomberman.

Para ello utilizará las herramientas aprendidas durante el curso de Programación Orientada a Objetos, incluyendo:

- Manejo de Archivos
- Threads
- Interfaz gráfica con Java FX
- Herencia e interfaces
- Manejo de excepciones
- Estructuras de control
- Polimorfismo

El juego consiste en eliminar todos los fantasmas que aparecen en el laberinto de obstáculos, utilizando bombas.

Al inicio del juego se presentará un menú de opciones:

1. Nuevo Juego.
2. Historial de jugadores.
3. Salir.

### Nuevo Juego

Al elegir esta opción se presentará primero una pantalla donde se solicite el nombre del jugador. Luego ya podrá iniciar el primer nivel juego.

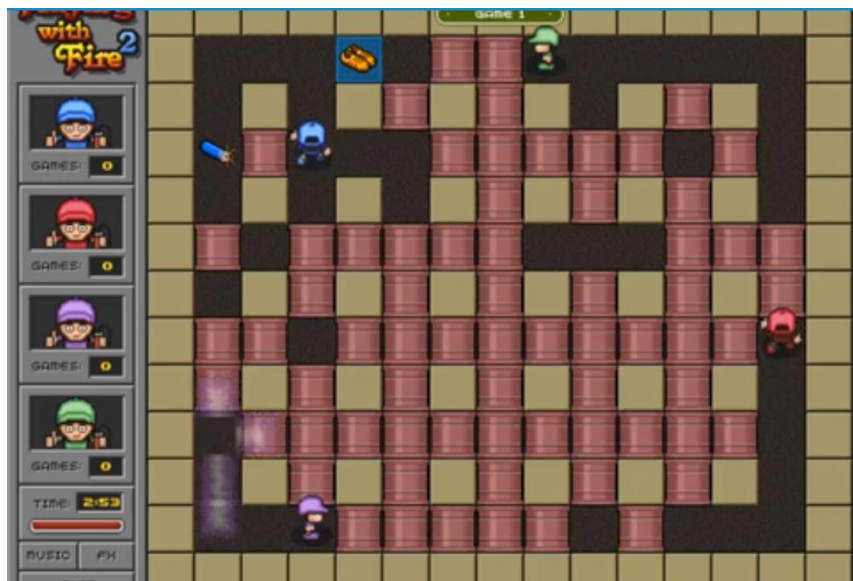
### Desarrollo del juego:

Cada nivel del juego tendrá una duración de dos minutos.

Además, el juego estará compuesto por dos niveles. Cada vez que se logré derribar a todos los fantasmas se podrá acceder al siguiente nivel.

En cada nivel aparecerán 5 fantasmas que se mueven automáticamente por el laberinto poniendo bombas cada tres movimientos que explotan dos segundos después de ser ubicadas en una posición. Solo pueden ubicarse bombas en las áreas libres del laberinto.

- Para el nivel básico, debe aparecer un laberinto similar al que se muestra en la imagen a continuación:



- Para el nivel Intermedio, cambia el diseño del tablero como se muestra en la imagen. En este laberinto habrá túneles que transportan al jugador de una posición a otra justo cuando se ubica sobre él.



Considerando que el tablero será siempre de 9 filas por 11 columnas.

En ambos laberintos habrá partes que se pueden explotar con bombas para abrir caminos, y otras partes que serán muros indestructibles.

Tanto el jugador como los fantasmas pueden moverse solo por las áreas libres del tablero en los sentidos: arriba, abajo, derecha e izquierda.

Para el jugador utilizará las teclas direccionales del teclado y los fantasmas se mueven de forma independiente.

Para ubicar una bomba el jugador usará la barra espaciadora. Solo se pueden ubicar bombas en las áreas libres.

Una bomba que explota tiene siempre un alcance de 3x3 y se expande por cada extremo (arriba, abajo, derecha e izquierda).

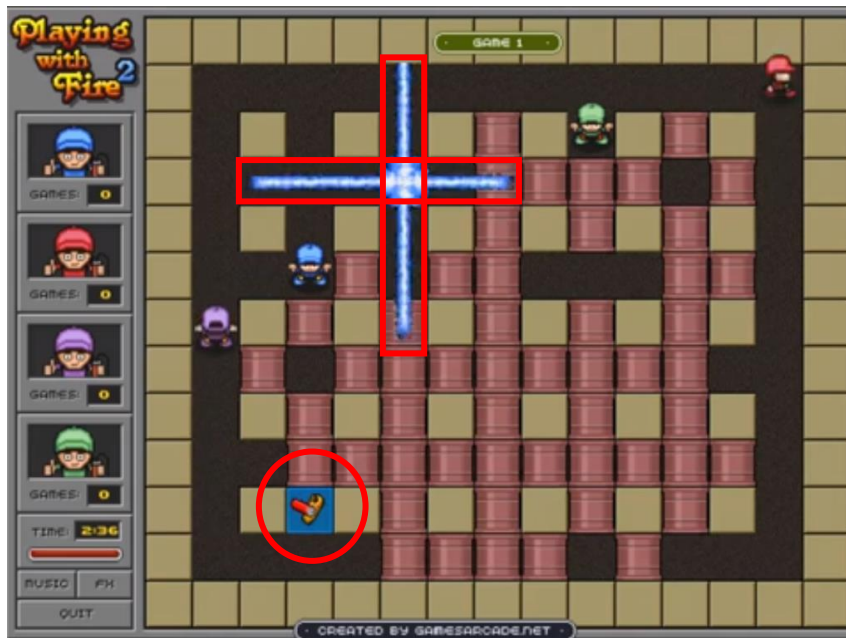
- Si el alcance de la bomba llega a la ubicación de un fantasma o de un barril, los elimina.
- Si el alcance de la bomba llega al jugador, este pierde una vida. Un jugador tiene tres vidas inicialmente.

Además, cada vez que explota una bomba, en su lugar aparecen premios que pueden ser: monedas o vidas (que se generarán de forma aleatoria). Estos premios aparecen por tres segundos, y si no son alcanzados por el jugador, desaparecen.

- Cada moneda equivale a 20 puntos que el jugador puede ir acumulando.
- Las vidas se acumulan también y se consumen cada vez que el jugador pierde una vida por culpa de una bomba. Si el jugador se queda sin vidas termina el juego

En el lado izquierdo de la pantalla se mostrarán:

- Puntos
- Vidas
- Tiempo (debe mostrarse un cronómetro en la pantalla del juego).



Siempre que termine una partida, debe indicarse al jugador una ventana con el mensaje indicando que ha ganado o perdido.

En el caso de haber ganado se mostrará los puntos obtenidos, y se preguntará si desea jugar el siguiente nivel, en el caso de que haya terminado el primer nivel, o se mostrará un mensaje FINAL: "¡Felicidades! Has superado los dos niveles de este juego", en el caso de que haya pasado los dos niveles de manera consecutiva.

Si esto ocurre se guardará el nombre del jugador en un archivo de ganadores.

En el caso de haber perdido, se debe preguntar al usuario si desea seguir jugando o no.

- Si elige la opción Sí, entonces volverá a jugar el mismo nivel.
- Si elige no, deberá retornar al menú principal.

El juego puede finalizar por dos motivos:

- Se acabó el tiempo y no logró destruir a todos los fantasmas.

- El jugador se quedó sin vidas

Al final, solo cuando el jugador gana, se debe guardar los datos en un archivo historial.txt con el siguiente formato:

```
nombre, tiempo de juego, monedas, fecha, nivel
Carlos, 00:01:20, 150, 12/10/2017, 1, S
María, 00:00:55, 50, 13/10/2017, 1, N
José, 00:02:30, 600, 17/10/2017, 2, S
```

## Historial de Jugadores

Si elige la opción 2 del menú principal podrá ver la información detallada del historial con el siguiente formato:

Jugador	Tiempo	Monedas	Fecha	Nivel
Carlos	00:01:20	150	12/10/2017	1
María	00:00:55	50	13/10/2017	1
José	00:02:30	600	17/10/2017	2

## Salir

Si elige la opción 3 del menú se termina la ejecución del juego.

## Consideraciones

- Recuerde hacer uso de las herramientas mencionadas.
- Recuerde que debe implementar en su código el manejo de excepciones necesarias en cada caso. Esto quiere decir que su programa no debe caerse y si ocurre alguna excepción debe mostrarse al usuario algún mensaje personalizado más no la excepción en sí. Por ejemplo: "Problemas técnicos. Estamos resolviendo".
- Debe ser creativo para el diseño de la interfaz y utilizar los gráficos adecuados para el desarrollo del juego. Por ejemplo, si se habla de fantasmas, deben incluirse imágenes que representen esto.
- Incluir el uso de Hojas de Estilo externas (css) para agregar estilo al proyecto. Recurso: <https://docs.oracle.com/javase/8/javafx/get-started-tutorial/css.htm>
- Cada nivel debería variar, por su diseño (imágenes de fondo y de los fantasmas).
- Para probar el juego original, revise el siguiente link: <http://www.paisdelosjuegos.com/ec/juego/bomberman/playing+with+fire+2.html>

- El proyecto debe cumplir con el 70% de la funcionalidad para ser calificado, caso contrario tendrá una calificación de 0.
- La nota final del proyecto será igual a:  $\text{nota} \times \text{sustentación} \times \text{participación}$ .
- **Puede ganar hasta 5 puntos adicionales si su proyecto, ADEMÁS de todos los requerimientos establecidos, incluye características adicionales y novedosas que demuestren su esfuerzo en el desarrollo de este juego.**

## Entregables

- Proyecto java.
- Diagrama de clases.
- Documentación generada con Javadoc.

## Rúbrica del proyecto

Funcionalidad	Debe cumplir la funcionalidad del juego como tal y como lo explica en este documento. NO invente sus propias reglas	25
Interfaz gráfica con Java FX	Utilice su creatividad	20
Implementación de Threads de forma correcta	Implementación de Threads para el reloj y para dar independencia a los piratas	15
Manejo de excepciones	Manejar todas las excepciones de forma específica que pueden presentarse durante la ejecución del programa.	15
Implementación de Herencia e Interfaces	Identificar de forma correcta cuando necesite usar herencia y cuando necesite definir interfaces	5
Manejo de Archivos	Lectura y escritura de archivos.	10
Abstracción y uso de objetos	Creación de Clases pertinentes con sus propiedades y métodos. Correcta interacción de objetos.	5
Documentación (Javadoc y diagrama de clases)	Debe documentar todos los métodos utilizados en la programación <b>NO SOLO</b> ALGUNOS.	5

## Modo de Entrega:

- Proyecto en java
- Carpeta comprimida con la documentación del proyecto incluyendo diagrama de clases.
- Subir un zip de su proyecto al link de la tarea en el sidweb con el siguiente formato: [FP5][PY]SegundoParcial\_Apellidos\_nombres.py (incluir el proyecto y la documentación)