**Instituto Tecnológico de Costa Rica**

**Curso IC-2101 Programación Orientada a Objetos**

**Dennis Valverde Pacheco**

**Descripción del segundo proyecto programado**

**Valor: 25%**

**Bomberman**

**Objetivos**

* Aprender el lenguaje Java y familiarizarse con la Programación Orientada a Objetos.
* Aplicar conceptos de diseño de aplicaciones utilizando la teoría de Orientación a Objetos.
* Generar una herramienta que canalice todos los conceptos estudiados del tema.

**Requerimientos funcionales**

Se requiere el desarrollo de un videojuego en java basado en el existente BOMBERMAN, (para referencia <https://www.youtube.com/watch?v=3smytj9Bu_E>), el juego consiste en un heroe que mediante el uso de bombas debe descubrir la puerta secreta para salir de un territorio que se encuentra plagado de una especie de globos y barriles vivientes que desean eliminarlo.

El territorio será representado con una matriz en tres tamaños disponibles 20x20 (facil), 40x40 (intermedio) y 60x60 (dificil), en el cual de manera aleatoria se colocaran un total de 75 bloques (facil), 200 bloques (intermedio) y 500 bloques (avanzado). Es necesario aclarar que a excepción de las filas y columnas de las orillas de la matriz el resto de la misma cuenta con bloques barrera los cuales no pueden ser destruidos por las bombas, estos bloques se encuentran distribuidos de manera intermitente (de uno por medio) ya sea de manera horizontal y vertical.

Se deberá hacer uso de objetos para representar cada uno de los elementos dentro de la matriz, de este modo se deberá implementar los objetos: bloque, bloque barrera, espacio vacío, héroe, globos, barriles, bomba.

Todos estos objetos heredan de la superclase Element, la matriz contiene solo datos de ese tipo.

Bloque barrera

Su función es simple pero vital, son bloques indestructibles que se encuentran de manera intermedia en toda la matriz, al no poder ser destruidos permite darle una mayor estructura a los “túneles” o canales por donde avanza el heroe y los enemigos, su existencia es útil para atrincherar los enemigos y destruirlos con bombas.

Bloque

Son bloques de concreto que pueden ser destruidos mediante bombas. Obstruyen el paso al heroe y a los enemigos. Solamente uno esconde la puerta por donde el heroe termina el juego.

Espacio vacío

Son espacios disponibles para que transite el heroe o los villanos.

Héroe

Único en el juego, es el personaje con el que jugamos la partida, debe ir por la matriz realizando dos tareas específicas con sus bombas: romper bloques en búsqueda de la puerta y eliminar sus enemigos.

Los movimientos del héroe se realizan con los direccionales arriba, abajo, derecha e izquierda. No puede atravesar ningún otro componente excepto la puerta final.

Cuenta con bombas ilimitadas. Gana el juego cuando elimina a todos sus enemigos, encuentra la puerta y la cruza.

Bombas

Son colocados por el héroe al presionar la tecla “espacio”. Se instalan en la misma posición donde se encuentra el heroe y explotan a los 3 segundos, solo es permitido instalar una bomba a la vez.

La bomba destruye bloques, elimina enemigos o puede causar la muerte al héroe. El alcance de una bomba es de dos espacios a la redonda.

Globos

Son los enemigos más torpes, recorren la matriz en busca del héroe para eliminarlo, lo logran una vez se colocan en la misma posición que él. No pueden atravezar paredes y se mueven sin razonamiento alguno de una manera aleatoria.

Se encuentran 6 globos en el modo fácil, 10 en el intermedio y 15 en el difícil.

SE ASIGNARÁ 5 PUNTOS EXTRA SI SE IMPLEMENTA, DOCUMENTA Y DEFIENDE ALGÚN ALGORITMO PARA IMPLEMENTAR LA BÚSQUEDA “INTELIGENTE” DEL HEROE POR PARTE DE LOS GLOBOS.

**Otras funcionalidadades**

* Se deben implementar hilos para el movimiento independiente de los enemigos.
* Se deben implementar sonidos para eventos importantes como explosiones, muertes de enemigos, triunfos o pérdidas.
* Se implementarán imágenes para cada elemento de la matriz, a su vez se deberá implementar una ventana que permita seleccionar las configuraciones iniciales de juego como por ejemplo el nivel de funcionalidad.
* Se cronometrará el tiempo total que un usuario use para lograr pasar el juego, cuando este termina una partida aparecerá una ventana solicitando el nombre del mismo para almacenar su tiempo en un archivo de texto, esto con el fin de hacer una ventana que muestre la tabla de posiciones con los 5 usuarios más rápidos.

**Requerimientos de programación orientada a objetos**

* Es importante recordar utilizar adecuadamente los modificadores de acceso public, private, protected, etc de manera efectiva y eficiente.
* Los métodos set y get deben ser desarrollados dentro de cada clase cuando correspondan.
* El diseño de las clases para la solución del proyecto será propuesta por cada grupo de trabajo, sin embargo, dicha implementación debe contemplar ya sea clases abstractas o interfaces y la justificación de su creación y uso debe ser documentada.
* Se deberán definir y justificar también la creación de métodos abstractos creados donde corresponda.
* Las clases generadas deberán encontrarse clasificadas en al menos dos paquetes que permitan diferenciar la funcionalidad de estas dentro de la solución.
* Es necesario la implementación excepciones. Al menos una excepción debe ser customizada por el equipo del proyecto (creación propia).
* Finalmente, debe elegir 2 clases para que elabore las pruebas unitarias respectivas. Debido a que la clase reservación es de suma importancia, debe estar incluida entre las clases seleccionadas para el proceso de pruebas.

**Documentación**

Se deberá documentar cada uno de los métodos desarrollados para obtener la solución del sistema. A esto se le conoce como documentación interna y se hace mediante comentarios dentro del código Java al estilo Javadoc para generar al finalizar el HTML correspondiente.

Se deberá generar adicionalmente una documentación externa del proyecto en formato PDF con la estructura definida en el proyecto anterior.

**Evaluación**

La nota del proyecto se obtendrá de la siguiente forma:

|  |  |
| --- | --- |
| **Aspecto** | **Valor** |
| Diseño y funcionalidad de interfaz gráfica | 20% |
| Requerimientos funcionales del proyecto (jugabilidad) | 50% |
| Requerimientos de Programación Orientada a Objetos | 15% |
| Documentación incluyendo informes de avance | 15% |

**Aspectos administrativos**

* La tarea será desarrollada en grupos de trabajo de máximo 2 personas. Cualquier acción de plagio será sancionada con un **0** en la nota y la carta al expediente respectiva para todos los implicados.
* La fecha de entrega será el viernes 13 de Noviembre de 2015 a las 22:00.
* Se entregarán informes de avance vía email los días 27 de Octubre, y 3 de Noviembre de 2015 en donde se indique el grado de avance, detalles técnicos o screenshots del sistema y dificultades presentadas.
* La entrega será mediante la plataforma Tec-Digital.
* Cualquier archivo infectado con virus no será revisado.
* Se recomienda que se empiece a trabajar lo más antes posible.