FIUBA - 7507

Algoritmos y programación 3

Trabajo práctico 2: BombitaRodriguez

1er cuatrimestre, 2012

(trabajo grupal)

Nombre	Padrón	email
Franco Negri	93817	franconegri2004@hotmail.com
Matias Barro	92970	matias.barro@gmail.com
Federico Di Rocco	92853	fede.dirocco@hotmail.com
Francisco Disalvo	91545	fran_disa@hotmail.com

Página en blanco (intencionalmente)

Introducción

Objetivo del trabajo

Aplicar los conceptos enseñados en la materia a la resolución de un problema, trabajando en grupo utilizando Java o C#.

Consigna

Desarrollar la aplicación completa, incluyendo el modelo de clases, la interfaz con el usuario, con sus pruebas correspondientes, de un juego cuyas reglas se muestran a continuación, y se pedirá pruebas unitarias y de integración completas, y elementos de documentación que se detallan en este enunciado.

Introducción

La empresa *Unidos o dominados Inc.*, creadora de grandes éxitos en el campo de juegos de video, está encarando un proyecto para realizar una remake de un famosísimo juego de los años '80 (bomberman – a ver que es estoHYPERLINK "http://www.minijuegostop.com.mx/items/accion/0/15391 bomberman-clasico/"?!). Para levar adelante ésta tarea ha elegido a su grupo de trabajo para realizar el diseño y desarrollo el producto en cuestión.

El juego consiste en un arcade donde Bombita Rodriguez (en adelante Bombita) tiene que destruir, en sucesivos niveles a aquellos que están en contra de la Nación.

Reseña del juego y su modalidad:

Bombita deberá destruir a sus enemigos para así liberar ciudadanos del Imperialismo. Una vez liberada una ciudad pasará a la siguiente. Para ver como pasar de nivel ver sección **Paso a nivel siguiente.**

Para llevar a cabo la liberación contará con bombas de detonación retardada que podrán ser de varios tipos, según se especifique en la **Tabla 1**. Estas bombas en función de su tipo tiene una onda expansiva y un poder de destrucción dependiendo del tipo.

El ambiente en el cuál Bombita se desempeñará será un mapa de dos dimensiones. Este mapa puede tener tando espacios vacios (por los cuáles todos los personajes pueden trasladarse) y espacios con obstáculos, obviamente por estos nadie puede pasar en principio.

A modo de ejemplo, ésta sería una configuración de mapa posible:

	1	2	3	4	5	6
1	O1	O1				
2				O1	O1	O1
3			O2	O2		
4		O3			O3	
5			O2	O2		
6		O3	O3		O3	O3
7						

Los casilleros que están vacíos o en blanco pueden ser utilizados por los personajes para desplazarse. Aquellos etiquetados con una **OX** están ocupados con obstáculos, que pueden tener diferentes características. Tanto los obstáculos como sus características se detallan más adelante en la **Tabla 2.**
Aquellos casilleros ocupados pueden ser desalojados utilizando bombas. Por ejemplo: Si Bombita planta bomba en el casillero (5,2) luego de unos segundos la bomba detonará y el obstáculo puede: destruirse completamente, dañarse o no dañarse; en función de sus características (ver **Tabla 2**).

Bombita luchará contra varios enemigos, cada uno de estos tienen características/comportamiento especiales, las cuales se describen en la **Tabla 3.** La manera con la que cuenta nuestro héroe podrá destruir a sus enemigos es a través de la utilización de bombas.

Un poco más acerca de Bombita:

Inicialmente Bombita ingresa la ciudad portando un tipo de bomba (preferentemente el más inofensivo) una velocidad de acción o desplazamiento y una vida, la cual perderá en caso de ser alcanzado por cualquier proyectil, bomba u objeto contundente. Bombita podrá modificar su comportamiento y el de sus bombas tomando del mapa objetos especificados en la **Tabla 4**. Estos objetos son el resultado de la destrucción de los obstáculos.

Paso a nivel siguiente: Bombita liberará una ciudad una vez que haya eliminado a todos sus enemigos y haya encontrado la salida de la Ciudad en cuestión. La salida de una ciudad es secreta y está ocultada por un objeto del mapa, que al ser destruido dejará ver ésta salida.

Tipo	**Características**		
	Destrucción	**Retardo**	**Onda expansiva**
Molotov	5 unidades	1 seg	3 casilleros a la redonda
ToleTole	Infinitas unidades	5 seg	6 casilleros a la redonda

Tabla 1 – Armamento (Tipos de bombas)

Tipo	**Características**	
	Durabilidad	**Observaciones**
Bloque acero	Solo se puede destruir con bomba **Toletole**	No puede ocultar salida de ciudad
Bloque Cemento	10 unidades	Puede ocultar salida de ciudad
Bloque ladrillos	5 unidades	Puede ocultar salida de ciudad

Tabla 2 – Obstáculos

Nombre	Características	
	Habilidades	**Resistencia**
Cecilio	Al igual que Bombita planta bombas con el fin de destruir a nuestro héroe.	5 Unidades
Los López Reggaé	Siempre subidos a un tanque que les permite una mayor velocidad de desplazamiento. En lugar de plantar bombas lanzan proyectiles que detonarán a una distancia y tendrán una onda expansiva a determinar.	10 Unidades
Los López Reggaé Alado	Al igual que Cecilio puede plantar bombas pero además puede atravesar cualquier obstáculo por encima sin que este esté destruido.	5 Unidades

Tabla 3 – Enemigos de Bombita

Artículo	Características	
	Efecto	**Duración**
Habano/Chala	Acelera el ritmo cardíaco de Bombita y le permite correr	Hasta que muere
Bomba Toletole	Ahora bombita plantará bombas toletole únicamente. No se agotan	Hasta que muere
Timer	Reduce el tiempo de retardo 15%	Hasta que muere

Tabla 4 – Artículos (Modifican comportamiento de las bombas y de bombita)

Entregables

Se deberá desarrollar la aplicación completa, incluyendo la interfaz gráfica. Deberá poder grabarse los puntajes altos (los n mas altos, configurable) y grabar el estado del juego para retomarlo nuevamente en otro momento.

Deberá entregarse:

- todas las clases con sus métodos , organizados en paquetes/namespaces según criterio del alumno.
- conjunto de pruebas unitarias que muestren el uso del modelo y su correcto funcionamiento.
- documentación completa del código fuente
- documentación completa del diseño de clases (ver siguientes secciones del enunciado)
- la aplicación deberá poder correrse desde consola, para lo cual deberá proveerse el archivo de Ant o NAnt correspondiente.

Entrega

Este documento se deberá completar con las secciones correspondientes.

Deberá acordarse con su ayudante la forma de entrega de los elementos que se evaluarán.

Fechas de entrega

Entrega 1 (confirmar con su ayudante asignado): se deberá entregar la primer versión de la documentación con todos correspondientes y el modelo de datos con sus pruebas unitarias.

Entrega final (semana del 25 de junio de 2012): se deberá entregar la aplicación completa junto con la documentación revisada y corregida según los comentarios realizados por el ayudante asignado.

Pruebas

Todas las clases deberán contar con sus pruebas unitarias **COMPLETAS**.

La aplicación deberá contar con pruebas de integración **COMPLETAS**.

Documentación

Supuestos

- La distancia del proyectil es fija, en nuestro caso igual a 4 casilleros y despues ecplota.
- La onda expansivas al toparse con algun personaje u obstaculo deja de expandirse, dañandolo.
- El mapa es de forma cuadrada.
- Los personajes, cualesquiera que sean al chocarse perdaran vida. En el caso de Bombita muere.
- El proyectil adopta la movilidad de los personajes.

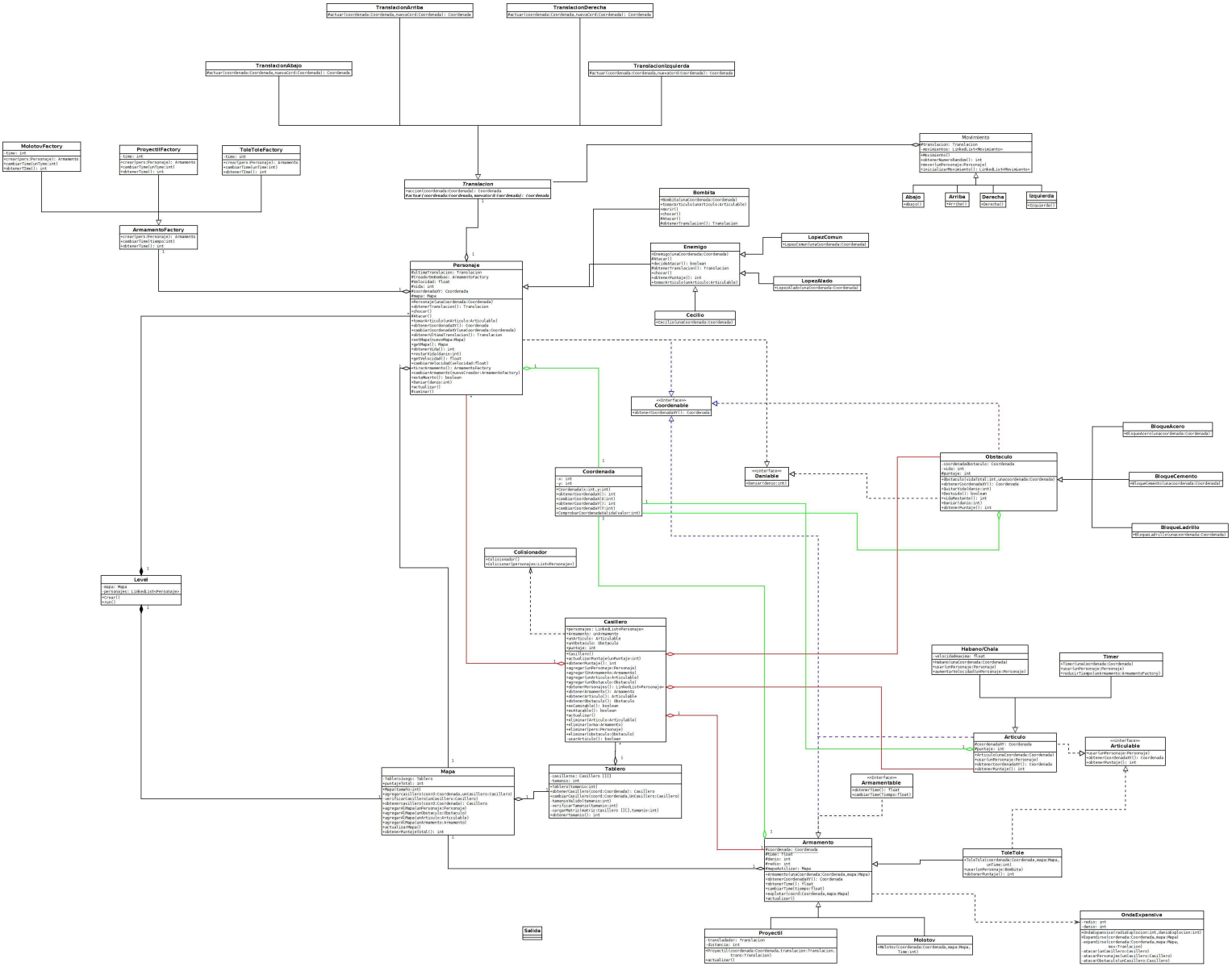
Elección del modelo entre los disponibles del TP1

El primer modelo del juego se realizó un miércoles por la tarde entre los 4 integrantes luego de que cada uno tenga su parte a realizar. Fue un modelo sin comportamiento para ver las relaciones entre las clases. No hubo un modelo por cada uno de los integrantes sino que uno unificado desde el principio. Cambios hubo varios aunque la estructura básica del sistema es la misma a la del modelo actual (por ejemplo se agregaron las clases Coordenadas, Translación y Movimiento que ayudan al movimiento de los personajes, las cuales no habían sido tenidas en cuenta en el primer modelo).

Modelo de dominio

Se dividieron las funcionalidades en 4 partes para poder comenzar con el trabajo práctico. Luego se creo el repositorio en Google Code para poder trabajar. Costo un poco al principio el tema de tenerlo organizado y así poder commitear sin problemas. Pero de a poco le fuimos encontrando la vuelta y fue de gran utilidad.

Diagramas de clases



Detalles de implementación

- Casillero: Es un contenedor de una lista de personajes, un armamento, un obstáculo, un articulo y un puntaje. Entre sus funciones se pueden destacar:
    - Contiene su propio puntaje que luego será calculado en total con un iterador en todos los casilleros cada vez que se actualiza el mapa.
    - Da información acerca si es caminable o es atacable para que los personajes puedan interactuar en el mismo.

- FactoryMethod: Patrón que fue utilizado para crear los armamentos y el level.

- Level: Crea el mapa en donde se va a llevar el juego y agrega los objetos necesarios para poder jugar. Además corre la aplicación.

- Translación/Movimiento: Se ocupan de todo el movimiento de los personajes en el juego. Por ahora el juego tiene un movimiento random (incluyendo a Bombita)

- Constantes juego: Es una clase donde se encuentran todas las constantes del juego. No tuvimos tiempo pero luego cada constante debería estar en sus clases correspondientes. Fue creada para tener una mayor legibilidad y organización a la hora de llevar a cabo la aplicación.

Excepciones

- ArmamentoNoInicializadoError: Es lanzada cuando un personaje quiere usar un armamento y este no se inicializo para ese personaje.

- VelocidadMaximaError: Se lanza cuando Bombita agarra el habano/chala, articulo que aumenta su velocidad, haciendo que su velocidad supere a la máxima posible.

- TiempoInvalidoError: Es lanzada cuando, a causa del timer, se debe reducir el tiempo de explosión de un armamento, y el mismo tiene un tiempo menor o igual a 0.

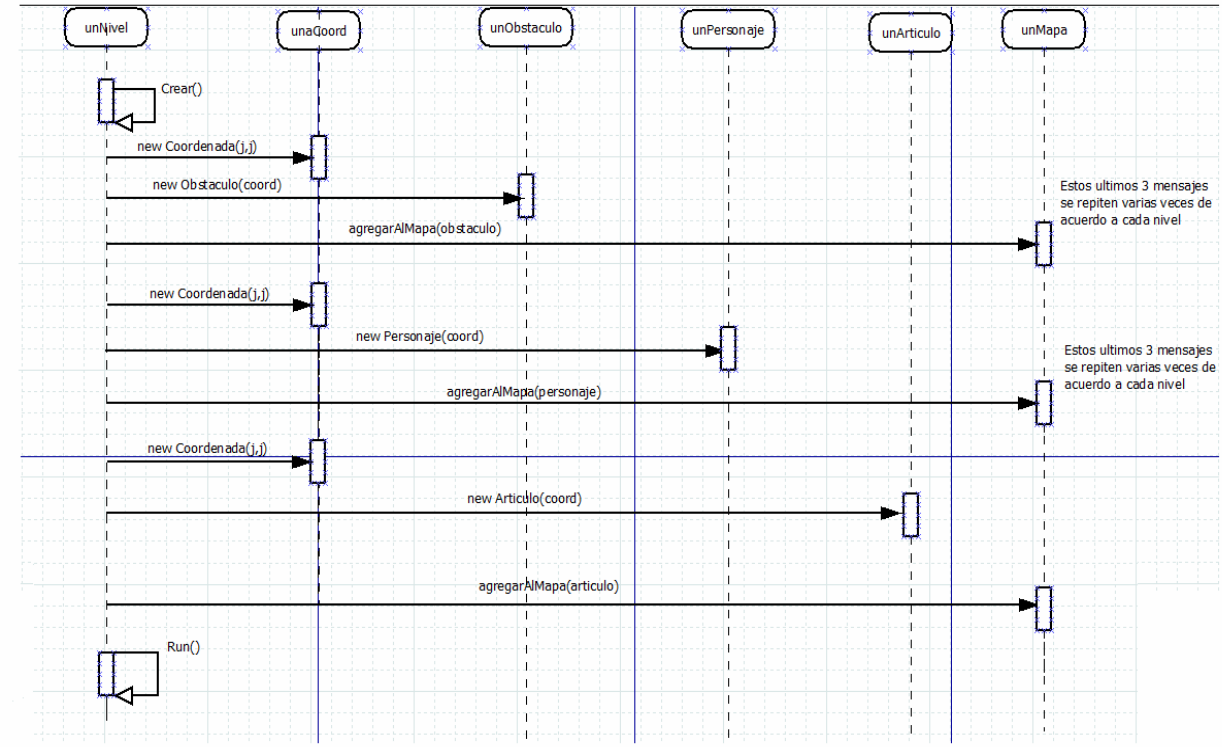
- CoordenadaInvalida: Es lanzada cuando se quiere validar alguna coordenada y esta no se encuentra dentro de los valores validos, o cuando en el caso de que la clase translación quiera generar una coordenada para un nuevo movimiento, y esta coordenada no es valida.

- TamanioMatrizInvalidoError: Es lanzada cuando la matriz tiene un tamaño invalido, o cuando un personaje se quiere mover hacia un casillero fuera del rango.

Diagramas de secuencia

Nos pareció interesante realizar la secuencias de como se genera un nivel, y de como un personaje realiza un movimiento.

1- Nivel:



2- Movimiento:

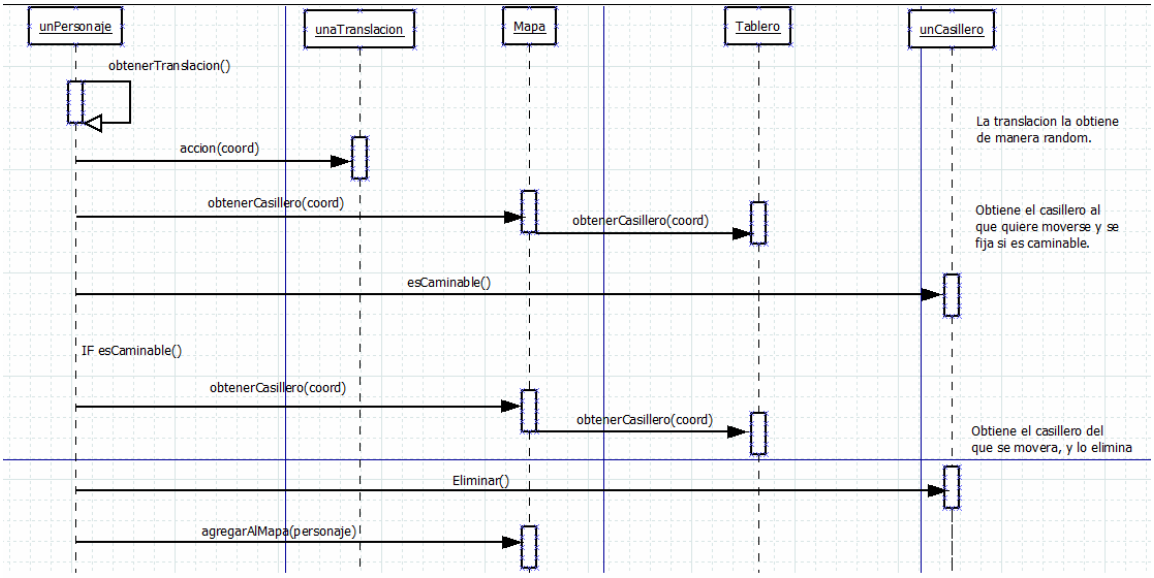


Diagrama de estados

**Lo agregaremos más tarde*

Diagrama de paquetes

**Lo agregaremos más tarde*

Checklist de corrección

Esta sección es para uso exclusivo de la cátedra, por favor no modificar.

Carpeta

Generalidades

- ¿Son correctos los supuestos y extensiones?
- ¿Es prolija la presentación? (hojas del mismo tamaño, numeradas y con tipografía uniforme)

Modelo

- ¿Está completo?¿Contempla la totalidad del problema?
- ¿Respeta encapsulamiento?
- ¿Hace un buen uso de excepciones?
- ¿Utiliza polimorfismo en las situaciones esperadas?

Diagramas

Diagrama de clases

- ¿Está completo?
- ¿Está bien utilizada la notación?

Diagramas de secuencia

- ¿Está completo?
- ¿Es consistente con el diagrama de clases?
- ¿Está bien utilizada la notación?

Diagramas de estado

- ¿Está completo?
- ¿Está bien utilizada la notación?

Código

Generalidades

- ¿Respeta estándares de codificación?
- ¿Está correctamente documentado?