

## TALLER DE LENGUAJES II

### ENTREGA MÓDULO I: MISSILE COMMAND

#### IMPORTANTE:

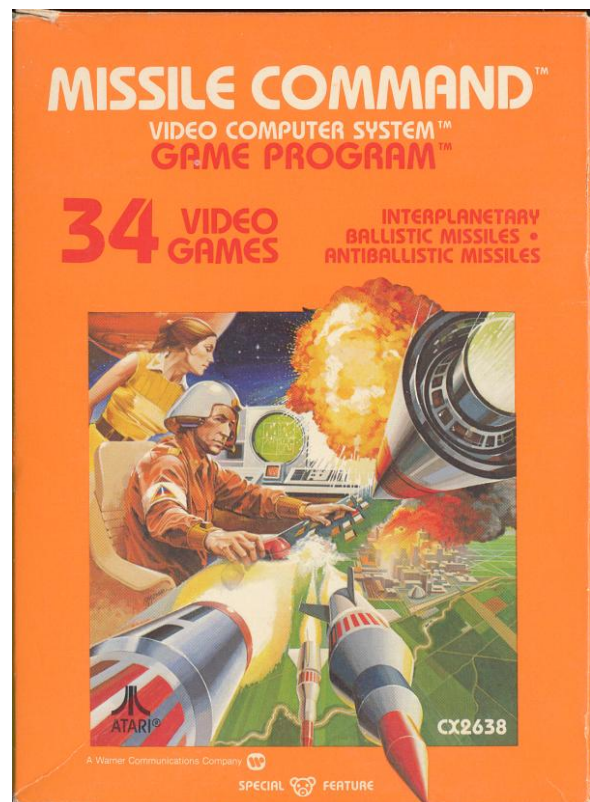
- ❑ **ÚLTIMO DÍA DE RECEPCIÓN DE TRABAJOS: 14/10/2018.**
- ❑ **En esta entrega NO se contempla lo correspondiente a visualización gráfica, ni control de tiempo en las jugadas.**

Missile Command es un juego de arcade de 1980 de Atari. Es un juego muy popular que combina diversión con un mensaje sobre los peligros de la guerra. Originalmente llamado 'Armageddon', Missile Command fue diseñado durante la época de la "guerra fría" entre Estados Unidos y Rusia.

La idea de desarrollar Missile Command comenzó con una historia en una revista sobre satélites que capturó la atención del presidente de Atari, quien pasó la nota a Lyle Rains. Rains le pidió a Dave Theurer que dirija los esfuerzos en crear este clásico juego lleno de acción. La "regla principal" para los desarrolladores fue que éste sería enteramente un juego defensivo, no un juego ofensivo.

Missile Command le da al jugador el control sobre 3 silos de misiles que protegen 6 ciudades en tierra. Cada silo tiene 10 antimisiles, y el jugador decidía desde que silo dispararía el próximo antimisil. El objetivo de cada antimisil era indicado con un cursor que era posicionado mediante una pequeña bola que que rotaba y luego con 3 botones en la consola de juego se elegía que silo disparaba el antimisil.

Es interesante notar que el juego no menciona una época o lugar, aunque el diseño original se refiere a 6 ciudades de la costa oeste de Estados Unidos: Eureka, San Francisco, San Luis Obispo, Santa Barbara, Los Angeles, y San Diego, las cuales forman parte del "anillo de fuego" y que en esa época se pensaba serían las ciudades más vulnerables en caso de un ataque nuclear soviético, y que de hecho fueron rodeadas por silos de misiles de defensa. Sin embargo, el manual del Atari 2600 indica una historia enteramente de ficción para el juego, la cual se presenta aquí.



¡Que se divierta!

## LA HISTORIA

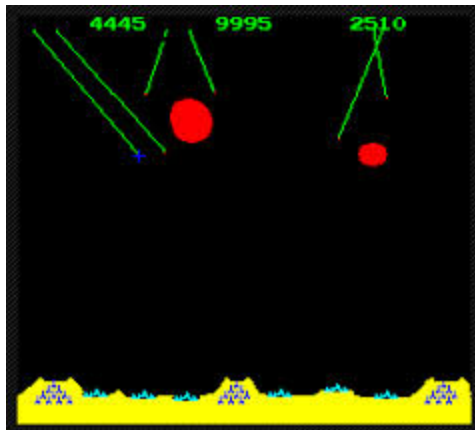
Aliens del planeta Krytol han comenzado un ataque sobre el planeta Zardon. Los Krytolianos son guerreros que buscan destruir y aprovecharse el planeta Zardon. Zardon es el último de los planetas pacíficos. Los Zardonianos son personas hábiles y trabajadoras, ellos mismos construyeron sus ciudades y son ricas en recursos. Además, Zardon es realmente un planeta libre de crímenes y violencia.

Zardon ha construido un sistema de defensa poderoso. Varias bases de misiles antibalísticos han sido establecidas dentro de las ciudades de Zardon. Los Zardonianos están listos para un ataque, y están preparados para pelear -si es necesario- para salvar sus ciudades.

Como comandante de la base, es tu responsabilidad proteger y defender las 6 ciudades del planeta de Zardon. Los Krytolianos han empezado a disparar misiles balísticos interplanetarios, están apuntando a tus ciudades y hacia tus 3 bases de misiles: Alpha, Delta y Omega. Tu única defensa es devolver el fuego con misiles antibalísticos. Pero, ¡cuidado!, los Krytolianos son astutos, también tienen misiles crucero. Los misiles crucero se ven como satélites, pero son tan dañinos como los misiles balísticos interplanetarios.

Deberás usar tus misiles antibalísticos (MABs) para detener al enemigo, antes que tu alegre y armonioso planeta sea destruido.

## EL JUEGO



El objetivo del juego es defender tus ciudades y bases de misiles. El enemigo dispara misiles balísticos interplanetarios (MBI) y misiles crucero, ambos tienen como objetivo destruir tus ciudades y bases militares. Hay 2 tipos de misil crucero: misiles cruceros tontos (que caen en línea recta) y misiles crucero inteligentes, los cuales tratarán de evadir tus misiles antibalísticos (MABs).

El enemigo ataca en una serie de oleadas que puede variar en el número de misiles balísticos interplanetarios que atacan. Cada oleada de misiles se mueve más rápido que la oleada anterior. Mientras más rápida es la oleada, más difícil es defender las ciudades. Por este motivo, mientras más rápida es la oleada, más alto será el puntaje ganado.

Con cada oleada, tenés 30 MABs para defenderte, 10 en cada uno de los silos, los cuales están ubicados en la parte inferior de la pantalla, a la izquierda, al centro y a la derecha. Una vez que has disparado los 30 MABs, estarás indefenso hasta que comience una nueva oleada.






Cada vez que un MAB llega a su destino o alcanza algún misil enemigo ocurre una explosión. La onda expansiva de esta explosión puede ocasionar que otros misiles balísticos interplanetarios o crucero que se encuentran en la zona de impacto también sean destruidos. Si la explosión del MAB no alcanza la cabeza del misil, éste seguirá con su trayectoria, sólo se verá afectada la estela que deja a su paso.

Debes tener en cuenta que hay un umbral (línea horizontal imaginaria) en el campo de juego, debajo de la cual no podrás disparar misiles. De esta manera se protege a los silos y las ciudades de la autodestrucción.

Si logras sobrevivir a la oleada, pasarás al siguiente nivel, si no logras sobrevivir a la oleada, tendrás sólo otra oportunidad para volver a defenderte.

El juego finaliza cuando todas las ciudades son destruidas.

## LOS MISILES KRYTOLIANOS

	<b>Misil Balístico Interplanetario de trayectoria recta:</b> aparecen en todos los niveles tratando de atacar las ciudades zardonianas y los silos de misiles. Siguen una trayectoria recta hacia uno de los silos o ciudades, y a partir del nivel <b>3</b> , luego de llegar cierta altura mínima se bifurcan en dos o más misiles balísticos interplanetarios.
	<b>Satélite:</b> es un bombardero. Tiene una forma circular, y ataca disparando 2 o más misiles balísticos interplanetarios de trayectoria recta a partir del nivel <b>2</b> del juego. A medida que se avanza en los niveles, aparece cada vez más cerca de las ciudades (más abajo).
	<b>Avión:</b> es un bombardero. Como su nombre lo indica tiene la forma de un avión y ataca disparando 2 o más misiles balísticos interplanetarios de trayectoria recta a partir del nivel <b>2</b> del juego. A medida que se avanza en los niveles, aparece cada vez más cerca de las ciudades.
	<b>Misil crucero tonto:</b> Es una pequeña esfera con puntas que cae en línea recta de la misma manera que los misiles balísticos interplanetarios de trayectoria recta.
	<b>Misil crucero inteligente:</b> Se ve exactamente igual que un misil crucero tonto, pero es capaz de esquivar las explosiones cercanas a éste. Es decir, una vez que detecta una explosión, se mueve a derecha o izquierda alejándose de la onda expansiva de la explosión. Un misil crucero inteligente tiene un sensor que barre una semicircunferencia hacia abajo de radio igual al de la explosión de un MAB, y es capaz de detectar cualquier MAB o explosión en el área de esa semicircunferencia. Esta semicircunferencia es siempre “hacia abajo”: Un misil crucero inteligente no detectará un MAB que explotara cerca de ella pero por arriba. Cuando el sensor de un misil crucero inteligente detecta un misil que va a explotar o una explosión, cambia de dirección tratando de escapar del área de la explosión, y luego retoma su objetivo.

## PUNTAJE

Ganarás puntuación cuando destruyas misiles balístico interplanetarios y misiles crucero. También podés sumar puntaje por los misiles antibalísticos no usados y ciudades salvadas. Dado que cada oleada de

misiles balísticos interplanetarios se mueve más rápido, el puntaje ganado también será mayor. Por ejemplo, oleadas 9 y 10 valen 5 veces el puntaje original.

Los puntos se anotan de la siguiente manera:

Objetivos	Puntaje
Misiles balísticos interplanetarios	25 puntos por misil
Misiles crucero enemigos	125 puntos por misil
Misiles antibalísticos sin usar	5 puntos por misil
Ciudades salvadas	100 puntos por ciudad
Bombardero	25 puntos por misil

Tenga en cuenta almacenar estos puntajes en contadores independientes, no simplemente el total para cuando los tenga que mostrar al final de cada nivel.

Por otro lado, anteriormente se mencionó que a medida que el juego avanza, las oleadas atacan más rápidamente, lo que influye en el puntaje obtenido, a continuación la tabla de progreso de los puntajes

Nivel	Puntaje
1-2	Puntaje simple
3-4	Puntaje x2
5-6	Puntaje x3
7-8	Puntaje x4
9-10	Puntaje x5
11 y más	Puntaje x6

Entre los consecutivos niveles, las ciudades caen y se levantan nuevamente (cuando la siguiente oleada empieza). Un bonus por ciudades (bonus city) es ganado cada 10000 puntos. Es decir, si tu puntuación alcanza 10000 al final de la oleada y las 6 ciudades son destruidas, aún así se recibirás el bonus city (una ciudad como sobreviviente al ataque) y otra oleada de misiles balísticos interplanetarios atacará.

## APARICIÓN DE ELEMENTOS EN EL JUEGO

Todo el juego transcurre en un área de 525px de ancho y 480px de alto (este es un tamaño sugerido que puede ser modificado).

Durante todo el juego caen Misiles Balísticos Interplanetario, ya sea desde el cielo ó desde los bombarderos. La cantidad de misiles puede variar entre 12 y 17, y no aparecen todos juntos de una vez, sino en oleadas o grupos. Pueden aparecer en grupos de 3 o 4.

En el siguiente cuadro, se muestra la aparición de los misiles krytolianos según se avanza en los niveles.

NIVEL NRO.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
MISIL CRUCERO TONTO	X	X	-	-	X	X	-	-	X	X	-	-	X	X	-	-
MISIL CRUCERO INTELIGENTE	-	-	X	X	-	-	X	X	-	-	X	X	-	-	X	X

## IMPLEMENTACIÓN

Para esta entrega deberá reflejar el comportamiento de todos los elementos de juego. Como puede deducir, eso le permitirá implementar todos los niveles mencionados.

## TEST DE LA APLICACIÓN

Si bien la aplicación no tiene aún interfaz gráfica, se provee una librería: `graficador.jar` la cual debe usar para testear su aplicación.

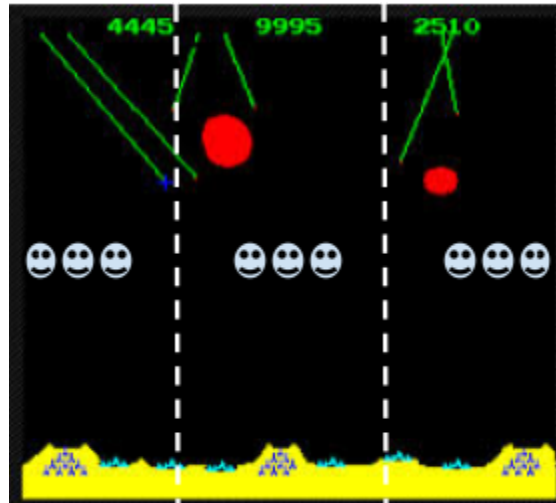
En su aplicación, cada elemento del juego tiene “un turno para realizar un movimiento”, estas acciones se ejecutan en N rondas de juegos hasta que el jugador gana o pierde en esa jugada. La idea del graficador provisto es que al final de cada ronda de juego, ud. invoque al método `graficador.refrescarDownTop(...)` ó `graficador.refrescarTopDown(...)`. **Realice las adaptaciones necesarias en sus clases para que puedan ser usadas por el graficador.**

Considerando que aún el jugador no puede indicar gráficamente -mediante el mouse- donde lanzará sus MABs, se deberá “hardcodear” el destino de los MABs que son disparados.

El Test de la aplicación deberá realizar lo siguiente para los **Niveles 1, 2 y 3**, donde deberá:

1. Calcular la cantidad de MIBs que atacarán en el nivel.

2. Cada MBI deberá elegir aleatoriamente su destino, esto le permitirá definir su trayectoria.
4. Defina 3 MBA con 3 direcciones destino diferentes, pero que cubran el primer tercio de la pantalla (izquierda, ver figura)
5. Defina 3 MBA con 3 direcciones destino diferentes, pero que cubran el segundo tercio de la pantalla (centro, ver figura)
6. Defina 3 MBA con 3 direcciones destino diferentes, pero que cubran el tercer tercio de la pantalla (derecha, ver figura)



7. Cada MBI tendrá una oportunidad para moverse, es decir, realizar un movimiento.
8. Cada MBA tendrá una oportunidad para moverse, es decir, realizar un movimiento.
9. Los puntos 7 y 8 se repiten hasta que el juego termina. Tenga en cuenta que no todos los MBI salen de una sólo vez.

## CRITERIO DE EVALUACIÓN

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

### 1. 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.

### 2. Documentación

- Adjuntar el diagrama de clases UML (**PDF o Imagen, no se aceptará otro formato y no se tomarán en cuenta diagramas UML auto-generados a partir del código fuente con algún plugin de eclipse**). 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.

- Entregar archivos en formato HTML como documentación respaldo de las clases creadas.
3. Correcto Funcionamiento. Incluyendo el uso del graficador provisto por la cátedra.
  4. 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.

**5. IMPORTANTE : El código entregado debe ser compatible con Java 1.7.**

### Notas Finales

- Para tener una idea intuitiva acerca de cómo funciona el juego, pruebe jugarlo accediendo a:  
<http://my.ign.com/atari/missile-command>
- Si se le dificulta pasar de nivel, puede observar este video donde un jugador experimentado le mostrará que ocurre en los sucesivos niveles de Missile Command  
<http://www.retroland.com/missile-command/>