FIUBA - 7507

Algoritmos y programación 3

Trabajo práctico 1: Algo42

1er cuatrimestre, 2011

(trabajo individual)

Fecha de entrega: viernes 22 de abril (ver detalles de forma de entrega en este documento)

Nombre: Juan Ángel Dausá

Padrón: 92267

Email: juandausa@gmail.com

Introducción
.....
4
Objetivo del trabajo....4
Consigna....4
Entregables....6
Forma de entrega....6
Pruebas 6
Supuestos....7
Herramientas y modo de entrega....7
Modelo de dominio
.....
7
Diagramas de clases....8
La relación entre las Naves del juego....8
Bombarderos....8
Manejador....9
Detalles de implementación....10
Excepciones....11
Diagramas de secuencia....11
Control de Posición de Algo42....11
Código fuente
.....
12
Checklist de corrección
.....
40

Introducción

Objetivo del trabajo

Aplicar los conceptos enseñados en la materia para la resolución de un problema de manera indvudual y utilizando Smalltalk.

Consigna

Desarrollar el modelo de clases, con sus pruebas correspondientes, de un juego cuyas reglas se muestran a continuación, al que se le pedirá pruebas unitarias y de integración completas, y elementos de documentación que se detallan en este enunciado.

Introducción

La empresa Rezagos Militares Software Inc., creadora de grandes éxitos en el campo de juegos de video, está encarando un proyecto para realizar un simulador de batalla de aviones para ser utilizado como software de entrenamiento de la Fuerza Aérea Argentina y ha elegido a su grupo de trabajo para realizar el diseño y desarrollo de su nueva e innovadora idea.

El juego consiste en un avión de combate que debe enfrentar naves enemigas.

Los escenarios deben tener 2 componentes principales: el fondo y los elementos móviles, como aviones enemigos, armas enemigas, objetos especiales, el avión del jugador y las armas del jugador.

A continuación se describe el enunciado general con las características funcionales de la aplicación a desarrollar:

> Corre el año 2042 y nuestro país debe defenderse de una invasión extranjera que busca el control de las fuentes de agua potable de nuestras provincias. Nuestra flota aérea consta de 2 aviones, uno de los cuales no funciona por falta de mantenimiento. La flota extranjera es muy poderosa y está compuesta de miles de aviones que comienzan a sobrevolar nuestro territorio y amenazan con controlarlo por completo. Pero aun queda una esperanza si nuestro único avión (cuyo nombre clave es "Algo42") pudiera llegar hasta el porta-aviones enemigo y arrojarse en picada sobre él, destruyendo el cuartel de control de los invasores.
>
> Para cumplir con su cometido, el Algo42 deberá cumplir una serie de misiones. En cada misión se enfrentara a una flota distinta de aviones invasores. Las flotas de los enemigos están conformadas por distintos tipos de aviones.
>
> La cantidad de aviones de las flotas es variable (mínimo 15 aviones). Cada flota cuenta con un avión Guía que coordina al resto de los aviones de la flota. En

caso de destruirse el avión Guía los demás aviones detienen sus disparos instantáneamente y huyen del campo de batalla.

Una implementación de una empresa competidora puede verse en el siguiente link, y sirve para entender la dinámica del juego:
http://www.youtube.com/watch?v=xQIB-O0DZm4
Los enemigos cuentan con los siguientes modelos de naves:

Nombre	Armas	Estrategia de vuelo	Observaciones	Puntos por destruccion
Avionetas	Lasers	Idas y vueltas en línea recta	Son los aviones más rápidos.	20
Bombarderos	Lasers, cohetes y torpedos rastreadores	Zig/Zag	Son los más poderosos pero al mismo tiempo los más lentos. Al ser destruidos, el Algo42 puede tomar sus armas.	30
Exploradores	No tiene	En círculos.	Vuelan en círculos amplios recorriendo toda la superficie aérea, en búsqueda de chocar al Algo42.	50
Cazas	Torpedos simples	En grupo formando una V	Al ser destruido su tanque de energía pude ser tomado por Algo42.	30

Por su parte el Algo42 es un avión escalable. En la versión base solo cuenta con lasers, pero puede escalar aumentado su poderío apropiándose de las armas y energía de los aviones que destruye.

Consideraciones generales:

- Todo avión tiene una fuente de energía, la cual disminuye a medida que es atacado. Cuando dicha energía llega a cero el avión es destruido.
- El Algo42 va sumando puntos para su misión a medida que destruye aviones enemigos. Al llegar a 1000 puntos termina el nivel y pasa al siguiente.
- Los lasers no se gastan, pero los torpedos y cohetes sí.
- El espacio no esta vacío, además de las flotas enemigas hay aviones civiles (pasan en línea recta a poca velocidad, el Algo42 debe evitar destruirlos ó chocarlos, caso contrario pierde 300 puntos por cada avión civil destruido) y helicópteros de la policía federal (se mueven en círculos pero tienen orden de no disparar, también debe evitarse su destrucción o se pierden 200 puntos por cada helicóptero).

Entregables

Se deberá desarrollar el modelo completo de la aplicación, sin incluir la interfaz gráfica.

Deberá entregarse:

- todas las clases con sus métodos , organizados en una o mas categorías según criterio del alumno.
- conjunto de pruebas unitarias que muestren el uso de la biblioteca y su correcto funcionamiento.
- documentación completa del código fuente
- documentación completa del diseño de clases (diagramas UML de clases y secuencia)

Forma de entrega

Enviar este documento completo, junto con todos los archivos .st correspondientes, a la siguiente dirección (según el curso en que esté inscripto):

algo3fiuba+miercoles@gmail.com

algo3fiuba+juevesnoche@gmail.com

algo3fiuba+juevestarde@gmail.com

En asunto del mail deberá indicar: 2011-1C-[número de curso]-TP1-[número de Padrón]

ejemplo: si cursa el jueves a la noche, y su padrón es 85432, el asunto deberá ser:

2011-1C-2-TP1-85432

Pruebas

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

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

Supuestos

- Se ha supuesto que los enemigos no se chocan entre si ni con aviones civiles o policias.
- Las balas rastreadoras tienen un combustible, el mismo se gasta cada vez que estas avanzan. Este combustible será del valor necesario para recorrer el largo del mapa una vez.
- Se ha utilizado un método particular de vuelo para los aviones que vuelan en linea recta, al estos llegar al limite del 'mapa' aparecen nuevamente al inicio, conservando todos los demas estados.

Herramientas y modo de entrega

- Informe completo (de acuerdo al checklist de corrección)

- Todos los diagramas en formato imagen incorporados a este documento y opcionalmente en el formato de la herramienta que hayan usado

- Archivo ".st" con la(s) categoría(s) del modelo

- Archivo ".st" con la(s) categoría(s) de las pruebas

Modelo de dominio

En el trabajo se partió de la idea sencilla de crear todos los objetos involucrados en el juego.

A medida que estos se fueron completando se comenzó a dar el comportamiento a los mismo.

Primero las tareas más sencillas como moverse o disparar, luego se implementó la comunicación entre los participantes.

Siempre se intentó que los nombres de mensajes y variables fuesen claros y que den una idea de lo que hacen.

Se agruparon los objetos por su comportamiento y pertenencia a un grupo del dominio. Esto generó clases como “Naves”, “Enemigas”, “Neutrales”.

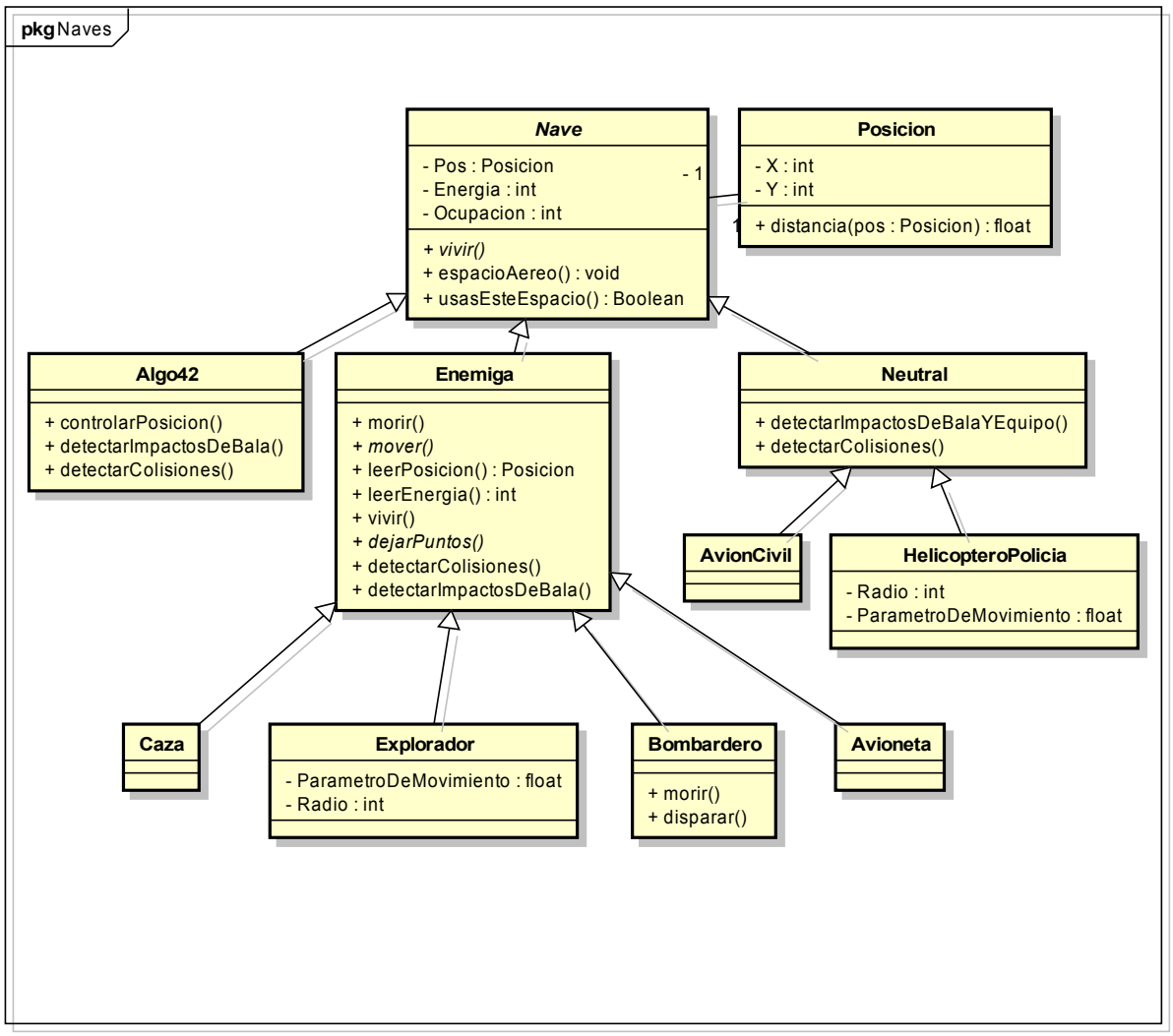
Además se crearon clases debido a la necesidad específica de representar ciertos atributos, un ejemplo es la clase “Posicion”.

Una clase particular es: 'Constantes', que tiene solo métodos de clase, estos retornan los valores usados por las naves y balas, como por ejemplo velocidad, vida, danio causado, etc.

A medida que se fue avanzando en el trabajo fue necesaria la existencia de un objeto “Manejador” que es el encargado de establecer el dialogo entre los jugadores.

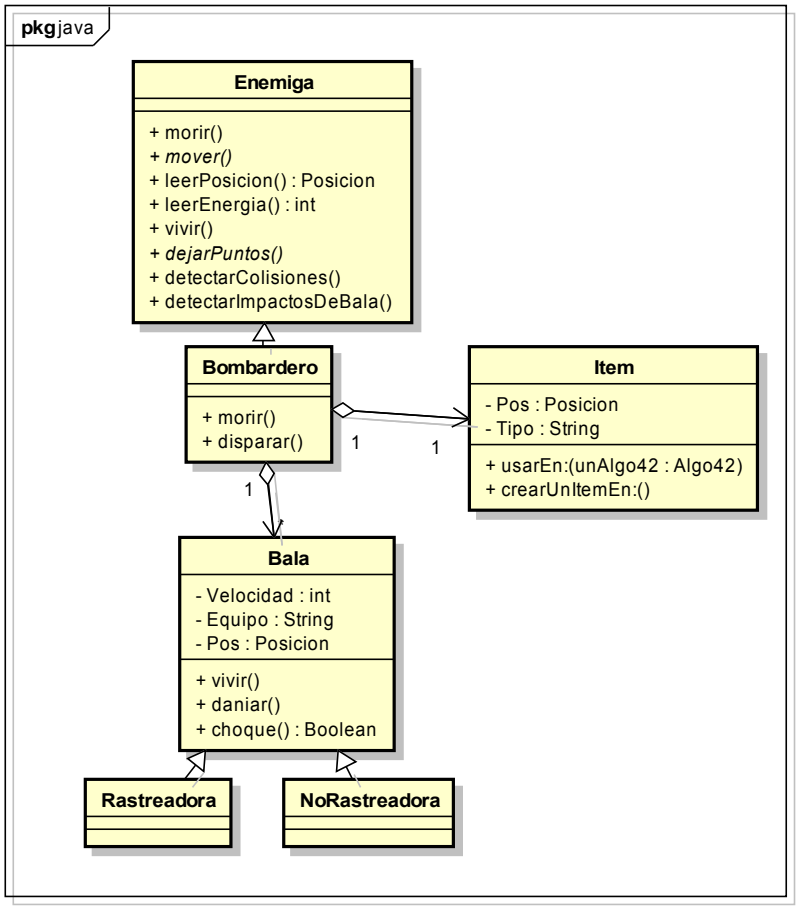
Diagramas de clases

La relación entre las Naves del juego.

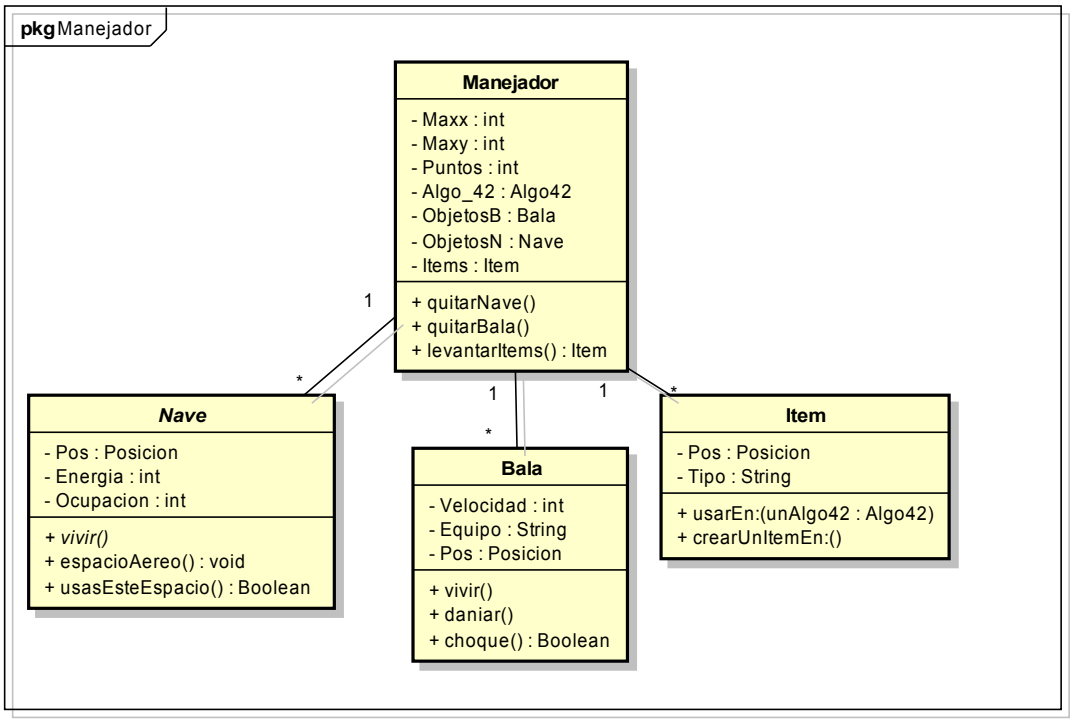


Bombarderos.

Se puede ver que tiene objetos del tipo 'Item', los 'Caza' también los tienen.


pkg java
Enemiga
+ morir()
+ mover()
+ leerPosicion() : Posicion
+ leerEnergia() : int
+ vivir()
+ dejarPuntos()
+ detectarColisiones()
+ detectarImpactosDeBala()
Bombardero
+ morir()
+ disparar()
Item
- Pos : Posicion
- Tipo : String
+ usarEn:(unAlgo42 : Algo42)
+ crearUnItemEn:()
Bala
- Velocidad : int
- Equipo : String
- Pos : Posicion
+ vivir()
+ daniar()
+ choque() : Boolean
Rastreadora
NoRastreadora
1
1
1
*

Manejador.


pkg Manejador
Manejador
- Maxx : int
- Maxy : int
- Puntos : int
- Algo_42 : Algo42
- ObjetosB : Bala
- ObjetosN : Nave
- Items : Item
+ quitarNave()
+ quitarBala()
+ levantarItems() : Item
Nave
- Pos : Posicion
- Energia : int
- Ocupacion : int
+ vivir()
+ espacioAereo() : void
+ usasEsteEspacio() : Boolean
Bala
- Velocidad : int
- Equipo : String
- Pos : Posicion
+ vivir()
+ daniar()
+ choque() : Boolean
Item
- Pos : Posicion
- Tipo : String
+ usarEn:(unAlgo42 : Algo42)
+ crearUnItemEn:()
1
*
1
*
1
*

Detalles de implementación

Se usaron además de las clases que modelan el problema una clase 'Posicion', que es una dupla de enteros. Se implementaron para esta setter y getters además de un método que calcula la distancia entre dos puntos.

Otra fue la clase 'Items', que a pesar de su pequeño código, en longitud, es de mucha utilidad para una fácil representación de estos.

Para la detección de colisiones contra Algo42 se pasa una referencia de todas las naves a este. Algo42 pregunta a cada nave si comparten el 'espacioAéreo' verificando la colisión.

Como se dijo anteriormente las naves enemigas no chocan entre si, por lo tanto para resolver las colisiones de éstas se les envía una referencia de Algo42 y las mismas resuelven el problema.

Un método muy similar se utiliza para las naves neutrales, pero éstas al colisionar con Algo42 modifican el puntaje.

Para que los impactos de bala sean detectados se envía una referencia de todas las balas a cada nave y estas indagan a las primeras si el 'espacioAéreo' es compartido, de ser así, se les indica a las balas que actúen.

La implementacion de Nivel no me resulta sencilla de armar, sin embargo trataré de esbozar una posible representación.

Tenemos una instancia de la clase nivel, supongamos la del nivel uno, esta misma se encargara de crear un grupo de naves y agregarlas al Manejador.

```
|unManejador|

( (unManejador cantidadDeElementos < 15) or: [(unManejador cantidadDeNaves)])
whileTrue: [

        Cazas crearFlotaDeCazasEn: unManejador.

        1 to: (3 atRandom) do: [ :i |

                Avionetas crearAvionetaEn: unManejador.

                Bombarderos crearBombarderoEn: unManejador.

                Exploradores crearExploradorEn: unManejador.

                ].

        ].

[Guia := unManejador devolverGuia: (cantidadDeNaves atRandom)] on: GuiaIncorrecto do:
[“Empezar de nuevo”].
```

Cuando el avión guía es destruido el 'Nivel' debería indicar que se retiren todos los aviones.

Cuando los Puntos del 'Manejador' superen los 1000 se efectúa el paso de nivel y se retiran los aviones.

Claramente cuando la Energia de Algo42 es 0 el juego termina.

Excepciones

Se creo una excepción de limites, 'LimitesIncorrectos' , esta se lanza si los parámetros de creación del 'Manejador' no son lo suficientemente grandes ya que me pareció importante asegurar que todos los objetos puedan moverse, en particular los exploradores y los cazas que necesitan más espacio que el resto.

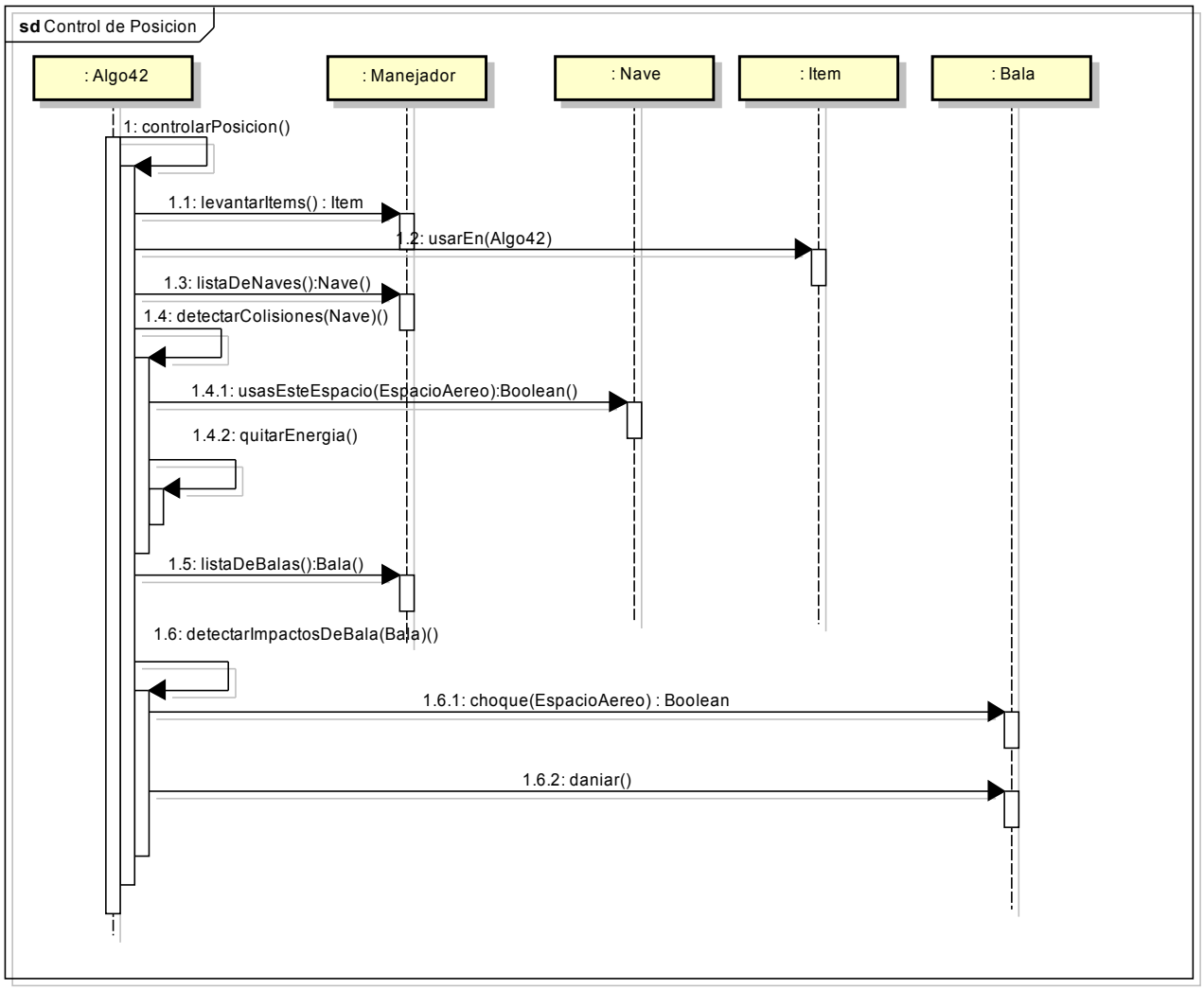
Otra excepción que me pareció importante fue 'MasDeUnAlgo42' ya que esta se lanza cuando por alguna razón se trata de reemplazar el Algo42 existente.

Se introdujo una excepción una tal que sea lanzada si no se encuentra el avión guía, aunque por la forma del programa esta no debería presentarse nunca :es bueno que se controle esto evitando así un desastre inminente.

Por último se creo una excepción que se lanza cuando el contenedor de los objetos no concuerda con lo que se espera, esta se llama ErrorDeReferencia, la misma indica que el juego deberá iniciarse nuevamente.

Diagramas de secuencia

Control de Posición de Algo42.



Código fuente

```
Object subclass: #Item
      instanceVariableNames: 'Tipo Pos'
      classVariableNames: ''
      poolDictionaries: ''
      category: '1942'!

!Item methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
00:12'!
darPosicion: unaPosicion
      "Fija la posicion inicial de un item"

      Pos darX: (unaPosicion leerX).
      Pos darY: (unaPosicion leerY).
      ! !

!Item methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/7/2011
19:11'!
darTipo: unTipo
      "Otorga el tipo al item"
      " 1 Armas"
      "2 Energia"
      ( unTipo = 1 or: [ unTipo = 2 ] ) ifFalse: [ ErrorDeTipoItem new
signal ]
      ifTrue: [ Tipo := unTipo ].! !

!Item methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/7/2011
18:54'!
initialize
      ""
      Tipo := 0.
      Pos := Posicion new.! !

!Item methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
00:13'!
posicion
      "Devuelve la posicion del item"
      ^Pos.! !

!Item methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/7/2011
18:38'!
usarEn: unAlgo42
      "Hace que item se incorpore a Algo42"
      ( Tipo = 1) ifTrue: [unAlgo42 sumarArmas].
      ( Tipo = 2) ifTrue: [unAlgo42 sumarEnergia: 50].
! !

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!

Item class
      instanceVariableNames: ''!

!Item class methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/3/2011
21:58'!
```

```
crearUnItemEn: unManejador
      |item|
      item := Item new.
      unManejador agregarItem: item.
      ^item.! !

Object subclass: #Nave
      instanceVariableNames: 'Pos Energia Ocupacion'
      classVariableNames: ''
      poolDictionaries: ''
      category: '1942'!

!Nave methodsFor: 'lectura' stamp: 'JuanDausa 4/19/2011 16:59'!
leerEnergia
      "Devuelve la energia de una nave."
      ^Energia! !

!Nave methodsFor: 'lectura' stamp: 'JuanDausa 5/7/2011 17:20'!
leerPosicion
      "Devuelve la posición de la nave"
      ^Pos.! !


!Nave methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
17:31'!
darPosicion: unaPosicion
      "Esto no debe usarse en el curso normal del juego, se usa sólo para
pruebas"
      Pos := Posicion crearPosicion: unaPosicion.
! !

!Nave methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
12:05'!
espacioAereo
      "Devuelve una lista con el espacio ocupado por la nave"
      | espacio li |
      espacio  := OrderedCollection new.
      li:= Ocupacion*(-1).
      (li) to: Ocupacion do: [ :j | espacio add: (Posicion
crearPosicionX: (Pos leerX+j)  Y: (Pos leerY+j)) ].
      ^espacio.! !

!Nave methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
13:07'!
morir: unManejador
      "Hará lo necesario para que la nave muera"
      self subclassResponsibility.! !

!Nave methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/4/2011
00:47'!
quitarEnergia: unaCantidad
      "Quita la energia de la nave"
      ( unaCantidad < 0 ) ifTrue: [ EnergiaAQuitarIncorrecta new
signal.].
      Energia := ( Energia - unaCantidad ).
      (Energia < 0) ifTrue: [Energia :=0].! !
```

```
!Nave methodsFor: 'consulta' stamp: 'JuanDausa 4/21/2011 18:05'!
posicion
      "Devuelve la posición de la nave, uso Interno"
      ^Pos! !

!Nave methodsFor: 'consulta' stamp: 'JuanDausa 5/4/2011 00:29'!
usasEsteEspacio: unEspacioAereo
      "Verifica si dado un espacio aereo, este se superpone con el de la
nave,"

      (self espacioAereo) do: [ :posicion | unEspacioAereo do:
            [ :otraPosicion |
                  ( ( posicion distancia: otraPosicion ) = 0 ) ifTrue:
                  [^true.]
            ]
      ].
      ^false.
! !

Nave subclass: #Enemiga
      instanceVariableNames: 'Guia'
      classVariableNames: ''
      poolDictionaries: ''
      category: '1942'!

!Enemiga methodsFor: 'uso' stamp: 'JuanDausa 5/6/2011 19:07'!
detectarColisiones: unaNave
      "Detecta si una nave colisiono con otra, si es así ocasiona  daño a
la primera"

      ( unaNave usasEsteEspacio: ( self espacioAereo ) ) ifTrue:
            [ self quitarEnergia: ( Constantes danioPorChoque ).  ].

! !

!Enemiga methodsFor: 'uso' stamp: 'JuanDausa 5/7/2011 18:21'!
detectarImpactosDeBala: unaListaDeBalas
      "Detecta los impactos de bala sobre una nave enemiga, estas solo
son afectadas por balas de algo42"
      | unEspacioAereo |
      unEspacioAereo := self espacioAereo.
      unaListaDeBalas do: [ :unaBala | ( unaBala choque: unEspacioAereo )
ifTrue:
            [ ( unaBala leerEquipo = 1 ) ifTrue: [unaBala daniar:
self ] ].
      ].! !

!Enemiga methodsFor: 'uso' stamp: 'JuanDausa 5/9/2011 00:04'!
vivir: unManejador
      "Deja hacer a una nave todo lo que tiene que hacer."

      self mover.
      (1 atRandom = 0) ifTrue: [self disparar: unManejador.].
      self detectarColisiones: ( unManejador devolverAlgo42 ).
      self detectarImpactosDeBala: ( unManejador listaDeBalas ).
      (self leerEnergia = 0) ifTrue: [
```

```
            self morir: unManejador.
            self sumarPuntos: unManejador.
            ].
      ! !


!Enemiga methodsFor: 'consulta' stamp: 'JuanDausa 4/22/2011 17:40'!
choque: unEspacioAereo
      "Dado un espacio aereo verifica si ambos aviones comparten el
espacio"
      | otroEspacioAereo |
      otroEspacioAereo := self espacioAereo.
      1to: (unEspacioAereo size) do: [ :i |
            1to: (otroEspacioAereo size) do: [ :j |
                  ( ( ( unEspacioAereo at: i ) distancia:
( otroEspacioAereo at: j ) ) = 0  ) ifTrue: [^true.]
            ].
      ].
      ^false.! !


!Enemiga methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
18:48'!
morir: unManejador
      "Quita a la nave de las referencias"
      |  |
      unManejador quitarNave: self.! !

!Enemiga methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/19/2011
17:02'!
mover: unManejador

      "Responsabilidad de la subclass"

      self subclassResponsibility.! !

!Enemiga methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
20:01'!
serGuia
      "Establece a la nave como guia"
      Guia := 1.! !

!Enemiga methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
20:19'!
sumarPuntos: unManejador
      "Suma los puntos despectivos por la muerte de la nave"

      self subclassResponsibility.! !

Enemiga subclass: #Explorador
      instanceVariableNames: 'ParametroDeMovimiento Radio'
      classVariableNames: ''
      poolDictionaries: ''
      category: '1942'!
!Explorador commentStamp: '<historical>' prior: 0!
Vuelan en círculos amplios recorriendo toda la superficie aérea, en
búsqueda de chocar al Algo42.
```

```
Instance Variables:
	ParametroDeMovimiento <Number>
	Radio <Integer>!


!Explorador methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
19:53'!
disparar: unMaejador
	"No dispara, se usa por practicidad."

! !

!Explorador methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/11/2011
13:57'!
initialize
	Pos:= Posicion new.
	Radio := 0.
	Energia := ( Constantes energiaExplorador ).
	ParametroDeMovimiento := 1.
	Ocupacion := ( Constantes ocupacionExplorador ).
	Guia := 0.! !

!Explorador methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
20:22'!
mover: unManejador
	"Mueve al Explorador en circulos grandes, de un radio menor al
menor de los limites"
	|auxiliar centro|
	centro := ( (unManejador menorLimite) / 2 ).
	(Radio = 0)
	ifTrue: [ Radio := ((unManejador menorLimite / 2) - 10 ).
		Radio := (Radio)sqrt asInteger].
	ParametroDeMovimiento := (ParametroDeMovimiento + (0.1)).
	(ParametroDeMovimiento >= 2) ifTrue: [ParametroDeMovimiento := 0].
	auxiliar := (ParametroDeMovimiento * Constantes pi).
	Pos darX: (( ( ( auxiliar cos) * Radio ) + centro ) asInteger).
"Avance parametrizado, Pi se genera por la clase Constantes"
	Pos darY: (( ( ( auxiliar sin) * Radio ) + centro ) asInteger). "De
ese resultado tomo el valor entero."! !

!Explorador methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
16:14'!
posIinicialConLimite: unLimite
	"Define una posición inicial dado cierto limite"

	Pos darX:  ( unLimite / 2).
	Pos darY: 1.
	! !

!Explorador methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
20:21'!
sumarPuntos: unManejador
	"Suma los puntos despectivos por la muerte de la nave"

	unManejador sumarPuntos: 50.! !
```

```
"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!

Explorador class
	instanceVariableNames: ''!

!Explorador class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/6/2011 19:25'!
crearExploradorEn: unMotor
	"Crea un explorador y lo inicializa"

	|a|

	a := Explorador new.
	a posIinicialConLimite: (unMotor menorLimite).
	unMotor agregarEnemigo: a.

	^a.! !

Object subclass: #Bala
	instanceVariableNames: 'Velocidad Pos Equipo Danio Estado'
	classVariableNames: ''
	poolDictionaries: ''
	category: '1942'!
!Bala commentStamp: 'JuanDausa 5/7/2011 18:25' prior: 0!
Bala has not been documented yet. The class comment should describe the
purpose of the class, its collaborations and its variables.

Instance Variables:
	Velocidad  <Integer>
	Pos  <Posicion>
	Equipo	1 'Buenos' ; 2 'Malos'
	Danio <Integer>
	Estado	0 'Usada' ; 1 'Activa'!


!Bala methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
14:50'!
choque: unEspacioAereo
	"Devuelve true o false dependienso si ocupan el mismo espacio
aereo"
	| |

	1 to: (unEspacioAereo size) do: [ :i |
		( ( ( unEspacioAereo at: i ) distancia: Pos ) = 0) ifTrue:
[^true]
	].
	^false.! !

!Bala methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/7/2011
18:25'!
daniar: unaNave
	"Indica a la nave el danio a realizarse"
	unaNave quitarEnergia: Danio.
	Estado := 0.! !

!Bala methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/7/2011
18:01'!
```

```
darEquipo: unEquipo
      ( unEquipo = 1 or: [ unEquipo = 2 ] ) ifFalse: [ ErrorDeEquipo new
signal ].
      "Otorga un Equipo a la bala, esto se hace para que los contrarios
no se maten entre si."
      "El equipo de Algo42 es '1' y el de los contrarios es '2'."
      Equipo := unEquipo! !

!Bala methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
12:02'!
darPosicionX: unx Y: uny
      "Inicializa la posición de una bala."

      Pos darX: unx.
      Pos darY: uny.! !

!Bala methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/7/2011
18:26'!
initialize
      Pos := Posicion new.
      Estado := 1. "Activa"! !

!Bala methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
16:52'!
leerEquipo
      "Devuelve el equipo de la bala"
      ^Equipo.! !

!Bala methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/7/2011
17:20'!
leerPosicion
      "Devuelve la posicion de la bala"
      ^Pos.! !

Bala subclass: #Rastreadora
      instanceVariableNames: 'Combustible'
      classVariableNames: ''
      poolDictionaries: ''
      category: '1942'!
!Rastreadora commentStamp: '<historical>' prior: 0!
Las balas rastreadoras de los Malos siguen a algo42. Las del equipo
Buenos siguen al enemigo más cercano.
Se agrego una atributo combustible ya que sino estas balas podrian estar
girando en el mapa indefinidamente.

Instance Variables:
      Combustible     <Number>!


!Rastreadora methodsFor: 'as yet unclassified' stamp: 'JuanDausa
4/21/2011 17:49'!
darCombustible: unCombustible
      "Registra el combustible disponible de la bala"

      | |
      Combustible := unCombustible.! !
```

```
!Rastreadora methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 19:25'!
initialize
      super initialize.
      Danio := ( Constantes danioTorpedoRastreador ).
      Combustible := 0.
      Velocidad := 0.! !

!Rastreadora methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/7/2011
18:08'!
mover: unManejador
      "Mueve a una bala rastreador. Dependiendo del equipo el movimiento
sera distinto,"
      | posASeguir  |
      (Equipo = 2)
      ifTrue: [ "Las balas rastreadoras enemigas siguen a Algo42"
            posASeguir := unManejador posAlgo42.
      ] ifFalse: [
            posASeguir := unManejador posMasCercana: ( Pos ).
      ].
      (posASeguir leerX = Pos leerX) ifFalse: [
            Combustible := (Combustible - 1 ).
            (posASeguir leerX < Pos leerX)
            ifTrue: [
                  Pos darX: (Pos leerX - 1).
            ] ifFalse: [
                  Pos darX: (Pos leerX +1)
            ].
      ].
      (posASeguir leerY = Pos leerY) ifFalse: [
            Combustible := (Combustible - 1 ).
            (posASeguir leerY < Pos leerY)
            ifTrue: [
                  Pos darY:(Pos leerY - 1).
            ] ifFalse: [
                  Pos darY:(Pos leerY + 1)
            ].
      ] ! !

!Rastreadora methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/9/2011
00:12'!
vivir: unManejador
      "Hace lo que bala rastreadora quiera hacer"
      ( Estado = 0 ) ifTrue: [
             unManejador quitarBala: self
            ].
      self mover: unManejador.

      (Combustible = 0) ifTrue: [
            unManejador quitarBala: self
            .].

! !

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!

Rastreadora class
```

```
	instanceVariableNames: ''!

!Rastreadora class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/3/2011 21:58'!
crearBRConPos: unaPosicion YCombustible: unCombustible
	"Crea una bala rastreadora con posicion unaPosicion"
	| bala |
	bala := Rastreadora new.
	bala darPosicionX: (unaPosicion leerX) Y: (unaPosicion leerY).
	bala darCombustible: unCombustible.
	^bala! !

Bala subclass: #NoRastreadora
	instanceVariableNames: ''
	classVariableNames: ''
	poolDictionaries: ''
	category: '1942'!

!NoRastreadora methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 19:22'!
esUnCohete
	"Da a la bala los parametros de Laser"
	Velocidad := ( Constantes velocidadCohete).
	Danio := ( Constantes danioCohete ) .! !

!NoRastreadora methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 19:23'!
esUnLaser
	"Da a la bala los parametros de Laser"
	Velocidad := ( Constantes velocidadLaser ).
	Danio := ( Constantes danioLaser ).! !

!NoRastreadora methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 19:23'!
esUnTorpedo
	"Da a la bala los parametros de Laser"
	Velocidad := ( Constantes velocidadTorpedo ).
	Danio := ( Constantes danioTorpedo ).! !

!NoRastreadora methodsFor: 'as yet unclassified' stamp: 'JuanDausa
4/22/2011 18:11'!
initialize
	super initialize.
! !

!NoRastreadora methodsFor: 'as yet unclassified' stamp: 'JuanDausa
4/22/2011 13:08'!
mover: unManejador
	"Mueve la bala en linea recta"
	Pos darY: (Pos leerY + Velocidad).! !

!NoRastreadora methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/9/2011 00:12'!
vivir: unManejador
	"Deja hacer lo que la bala tenga que hacer."
	( Estado = 0) ifTrue: [
		unManejador quitarBala: self
```

```
            ].
      self mover: unManejador.
      ( ( Pos leerX >= unManejador limitex ) or: [ Pos leerY >=
unManejador limitey ] )
            ifTrue: [
                  unManejador quitarBala: self
                  ].

! !

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!

NoRastreadora class
      instanceVariableNames: ''!

!NoRastreadora class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/3/2011 21:57'!
crearBNRConPos: unaPosicion
      |b|
      b := NoRastreadora new.
      b darPosicionX: unaPosicion leerX Y: unaPosicion leerY.

      ^b! !

!NoRastreadora class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/3/2011 21:57'!
crearBNRConX: unx ConY:uny
      |b|
      b := NoRastreadora new.
      b darPosicionX: unx Y: uny.

      ^b
      ! !

Nave subclass: #Neutral
      instanceVariableNames: ''
      classVariableNames: ''
      poolDictionaries: ''
      category: '1942'!

!Neutral methodsFor: 'Uso' stamp: 'JuanDausa 4/22/2011 01:37'!
vivir: unManejador
      "Hace lo que tiene que hacer una nave civil"
      "·Se implentara en subclase"
      self subclassResponsibility.! !


!Neutral methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/7/2011
20:20'!
detectarColisiones: unaNave
      "Detecta si unaNave colisiono con la invocante, si es así devuelve
true"

      ( unaNave usasEsteEspacio: ( self espacioAereo ) ) ifTrue:
            [ self quitarEnergia: ( Constantes danioPorChoque )  ].
      ! !
```

```
!Neutral methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/8/2011
20:14'!
detectarImpactosDeBalaYEquipo: unaListaDeBalas
      "Verifija si la nave ha chocado con alguna bala e indica que debe
efectuarse el daño. Ademas si la nave no posee más Energia devuelve el
Equipo que la destruyó"
      | espacio balasUsadas auxiliar|
      balasUsadas := OrderedCollection new.
      espacio := self espacioAereo.
      auxiliar := 0.
      1 to: ( unaListaDeBalas  size ) do: [ :i | ( ( unaListaDeBalas at:
i ) choque: espacio)
            ifTrue:
                  [ ( unaListaDeBalas at: i ) daniar: self.
                  ( self leerEnergia = 0 )
                  ifTrue: [
                        auxiliar := ( unaListaDeBalas at: i ) leerEquipo.
                        espacio := OrderedCollection new ]. "Esta como
doble codición de corte"
                        balasUsadas add: ( unaListaDeBalas at: i ).
            ].
      ].
      1 to: ( balasUsadas size) do:
            [ :i | unaListaDeBalas remove: ( balasUsadas at: i ) ].

      ( self leerEnergia = 0) ifTrue:
            [ ^auxiliar ].! !

!Neutral methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
14:56'!
morir: unManejador
      "Quita a la nave de las referencias"
      |  |

      unManejador quitarNave: self.! !

!Neutral methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/20/2011
13:02'!
posIinicialConLimite: unLimite
      "Define una posición inicial dado cierto limite"

      Pos darX: ((unLimite-5) atRandom + 2).
      Pos darY: 0.
      ! !

Enemiga subclass: #Bombardero
      instanceVariableNames: 'Direccion'
      classVariableNames: ''
      poolDictionaries: ''
      category: '1942'!
!Bombardero commentStamp: '<historical>' prior: 0!
Son  los más poderosos pero al mismo tiempo los más lentos. Al ser
destruidos, el Algo42 puede tomar sus armas.

Instance Variables:
      Direccion  <Integer>!
```

```
!Bombardero methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/3/2011
21:58'!
disparar: unManejador
      "Dispara el conjunto de armas del Bombardero"

      |b1 b2 b3|
      b1  := NoRastreadora crearBNRConPos: (self posicion).
      b1  darEquipo: 'Malos'.
      b1  esUnCohete.
      unManejador agregarBala: b1 .

      b2  := NoRastreadora crearBNRConPos: (self posicion).
      b2 darEquipo: 'Malos'.
      b2  esUnLaser.
      unManejador  agregarBala: b2.

      b3 := Rastreadora crearBRConPos: (self posicion) YCombustible:
(unManejador menorLimite).
      b3 darEquipo: 'Malos'.
      unManejador agregarBala: b3.! !

!Bombardero methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/11/2011
13:57'!
initialize
      Pos := Posicion new.
      Energia := ( Constantes energiaBombardero ).
      Direccion := 1.
      Ocupacion := ( Constantes ocupacionBombardero ).
            Guia := 0.! !

!Bombardero methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/7/2011
18:45'!
morir: unManejador
      "Quita la referencia de la nave y lanza los items respectivos."
      |item|
      super morir: unManejador.
      item := Item crearUnItemEn: unManejador.
      item darTipo: 1.
      item darPosicion: Pos.! !

!Bombardero methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
12:04'!
mover: unManejador
      | aux1 aux2 |
            aux1 := Pos leerX.
            aux2 := Pos leerY.
            (Pos leerX \\ 2 = 0)
            ifTrue: [ Pos darX: ( Pos leerX + 1)]
            ifFalse: [ Pos darX: ( Pos leerX -1 )].
            Pos darY: (Pos leerY + 1 ).
            (Pos leerY >= unManejador limitey)
            ifTrue: [Pos darY: 0].! !

!Bombardero methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/19/2011
17:45'!
posIinicialConLimite: unLimite
```

```
	"Define una posición inicial dado cierto limite"
	"Dejo espacio para el movimiento en zig-zag"
	Pos darX: ((unLimite-10) atRandom + 3).
	Pos darY: 0.! !

!Bombardero methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
20:21'!
sumarPuntos: unManejador
	"Suma los puntos despectivos por la muerte de la nave"

	unManejador sumarPuntos: 30.! !

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!

Bombardero class
	instanceVariableNames: ''!

!Bombardero class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/3/2011 21:59'!
crearBombarderoEn: unMotor
	"Crea un bombardero y la inicializa"

	|a|

	a := Bombardero new.
	a posIinicialConLimite: (unMotor limitex).
	unMotor agregarEnemigo: a.

	^a.! !

Enemiga subclass: #Caza
	instanceVariableNames: 'Stop'
	classVariableNames: ''
	poolDictionaries: ''
	category: '1942'!
!Caza commentStamp: '<historical>' prior: 0!
Vuelan en grupo formando una V.
Al ser destruido su tanque de energía pude ser tomado por Algo42.


Instance Variables:
	Stop <Number>!


!Caza methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
23:51'!
darPosicionInicialX: unX Y: unY
	"Fija la posición inicial de un caza"
	Pos darX: unX.
	Pos darY: unY.! !

!Caza methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
23:44'!
darStop: unEntero
	"Fija el stop de un caza, este se usará para su movimiento en V"
	Stop := unEntero.! !
```

```
!Caza methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/3/2011
21:57'!
disparar: unMotor
      "Crea una bala y la inicializa. Ademas deja un referencia en el
motor"
      |b|

      b := NoRastreadora crearBNRConPos: (self posicion).
      b darEquipo: 'Malos'.
      b esUnTorpedo.
      unMotor agregarBala: b! !

!Caza methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/11/2011
13:57'!
initialize
      Pos := Posicion new.
      Energia := ( Constantes energiaCaza ).
      Ocupacion := ( Constantes ocupacionCaza ).
      Stop := 0.
      Guia := 0.! !

!Caza methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/7/2011
18:45'!
morir: unManejador
      "Quita la referencia de la nave y lanza los items respectivos."
      |item|
      super morir: unManejador.
      item := (Item crearUnItemEn: unManejador).
      item darTipo: 2.
      item darPosicion: Pos.


! !

!Caza methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/7/2011
17:20'!
mover: unManejador
      "Mueve al caza en linea recta, el stop indica que debe esperar para
avanzar"
      |auxiliar|
      auxiliar := (self leerPosicion) leerY.
      (Stop > 0) ifTrue: [Stop := (Stop - 1)]
      ifFalse: [
            (unManejador limitey < (auxiliar + 1))
                        ifTrue: [ Pos darY: auxiliar + 1 ]
                        ifFalse: [ Pos darY: 0 ]
                  ].
      ! !

!Caza methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
23:49'!
sumarPuntos: unManejador
      "Suma los puntos despectivos por la muerte de la nave"

      unManejador sumarPuntos: 30.! !

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!
```

```
Caza class
	instanceVariableNames: ''!

!Caza class methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/3/2011
21:59'!
crearFlotaDeCazasEn: unManejador
	"Crea una flota de cazas en unManejador"
	| caza listaDeCazas auxiliar |
	auxiliar := ((unManejador limitex)/2) floor.
	listaDeCazas := OrderedCollection new.
	caza := Caza new.
		caza darPosicionInicialX: (auxiliar) Y: (unManejador limitey).
		caza darStop: 0.
		unManejador agregarEnemigo: caza.
		listaDeCazas add: caza.
	1 to: 2 do: [:i |
		caza := Caza new.
		caza darPosicionInicialX: (auxiliar - (i*3)) Y: (unManejador
limitey).
		caza darStop: i.
		unManejador agregarEnemigo: caza.
		listaDeCazas add: caza.
		caza := Caza new.
		caza darPosicionInicialX: (auxiliar + (i*3)) Y: (unManejador
limitey).
		caza darStop: i.
		unManejador agregarEnemigo: caza.
		listaDeCazas add: caza.
	].
	^listaDeCazas.! !

Object subclass: #Constantes
	instanceVariableNames: ''
	classVariableNames: ''
	poolDictionaries: ''
	category: '1942'!

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!

Constantes class
	instanceVariableNames: ''!

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/8/2011 22:35'!
danioCohete
	"Devuelve el danio que ocasiona un Cohete"
	^20.

! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/8/2011 22:34'!
danioLaser
	"Devuelve el danio que ocasiona un laser"
	^15.
```

```
! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/4/2011 00:34'!
danioPorChoque
      "Devuelve el valor de danio por un choque entre naves"

      ^20! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/8/2011 22:35'!
danioTorpedo
      "Devuelve el danio que ocasiona un Torpedo"
      ^25.

! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/9/2011 00:13'!
danioTorpedoRastreador
      "Devuelve el danio que ocasiona un Torpedo Rastreador"
      ^50.

! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 20:23'!
energiaAlgo42
      ^350.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 20:27'!
energiaAvionCivil
      ^100.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 20:24'!
energiaAvioneta
      ^100.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 20:27'!
energiaBombardero
      ^100.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 20:26'!
energiaCaza
      ^50.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 20:25'!
energiaExplorador
      ^100.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 20:28'!
```

```
energiaHelicopteroPolicia
	^100.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/11/2011 13:53'!
ocupacionAlgo42
	^2.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/11/2011 13:54'!
ocupacionAvionCivil
	^3.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/11/2011 13:53'!
ocupacionAvioneta
	^2.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/11/2011 13:54'!
ocupacionBombardero
	^6.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/11/2011 13:54'!
ocupacionCaza
	^2.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/11/2011 13:54'!
ocupacionExplorador
	^3.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/11/2011 13:56'!
ocupacionHelicopteroPolicia
	^3.! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
4/22/2011 16:25'!
pi
	"Devuelve Pi"
	^(3.14159265)! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 19:22'!
velocidadCohete
	"Devuelve el danio que ocasiona un Cohete"
	^3.

! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 19:22'!
velocidadLaser
	"Devuelve el danio que ocasiona un Cohete"
	^4.
```

```
! !

!Constantes class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/10/2011 19:23'!
velocidadTorpedo
      "Devuelve el danio que ocasiona un Cohete"
      ^4.

! !

Neutral subclass: #HelicopteroPolicia
      instanceVariableNames: 'ParametroDeMovimiento Radio'
      classVariableNames: ''
      poolDictionaries: ''
      category: '1942'!

!HelicopteroPolicia methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/11/2011 13:56'!
initialize
      Pos := Posicion new.
      Energia := ( Constantes energiaHelicopteroPolicia ).
      Ocupacion := ( Constantes ocupacionHelicopteroPolicia ).! !

!HelicopteroPolicia methodsFor: 'as yet unclassified' stamp: 'JuanDausa
4/22/2011 20:25'!
mover: unManejador
      "Mueve al Policia en circulos grandes, de un radio menor al menor
de los limites"

      |auxiliar centro|
      centro := ( (unManejador menorLimite) / 2 ).
      (Radio = 0)
      ifTrue: [ Radio := ((unManejador menorLimite / 2) - 10 ).
            Radio := (Radio)sqrt asInteger].
      ParametroDeMovimiento := (ParametroDeMovimiento + (0.1)).
      (ParametroDeMovimiento >= 2) ifTrue: [ParametroDeMovimiento := 0].
      auxiliar := (ParametroDeMovimiento * Constantes pi).
      Pos darX: (( ( ( auxiliar cos) * Radio ) + centro ) asInteger).
"Avance parametrizado, Pi se genera por la clase Constantes"
      Pos darY: (( ( ( auxiliar sin) * Radio ) + centro ) asInteger). "De
ese resultado tomo el valor entero."! !

!HelicopteroPolicia methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/12/2011 09:53'!
vivir: unManejador
      "Deja hacer lo que tiene que hacer un Avion civil"
      | equipo algo42|
      algo42 := ( unManejador devolverAlgo42 ).
      equipo := ''.
      self mover: unManejador.
      self detectarColisiones: algo42.
      "Aunque se sabe que solo choca contra algo42, esta comprobación se
realiza porque esto puede cambiar"
      ( ( self leerEnergia = 0 ) and: [ algo42 usasEsteEspacio: ( self
espacioAereo ) ] )
            ifTrue:
```

```
            [
            unManejador quitarPuntos: 200.
            self morir: unManejador]
      ifFalse: [
            equipo := self detectarImpactosDeBalaYEquipo: ( unManejador
listaDeBalas ).
            ] .
      ( equipo = 1 or: [ equipo = 2 ] )
      ifTrue: [
            self morir: unManejador
            ].

      ( equipo = 1 ) ifTrue: [ unManejador quitarPuntos: 200 ].! !

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!

HelicopteroPolicia class
      instanceVariableNames: ''!

!HelicopteroPolicia class methodsFor: 'as yet unclassified' stamp:
'JuanDausa 5/3/2011 22:00'!
crearPoliciaEn: unManejador
      |auxiliar|
      auxiliar := HelicopteroPolicia  new.
      auxiliar posIinicialConLimite: unManejador limitex.
      unManejador agregarEnemigo: auxiliar.
      ^auxiliar ! !

Nave subclass: #Algo42
      instanceVariableNames: 'Armas'
      classVariableNames: ''
      poolDictionaries: ''
      category: '1942'!
!Algo42 commentStamp: '<historical>' prior: 0!
Algo42 es del jugador, es escalable a todas las armas, inicialmente posee
una vida de 400, pero puede ser ampliada a 500 mediante items de energía.

Instance Variables:
      Armas<Integer>!


!Algo42 methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/11/2011
13:58'!
initialize
      Pos := Posicion new.
      Energia := ( Constantes energiaAlgo42 ).
      Armas := 1.
      Ocupacion := ( Constantes ocupacionAlgo42 ).! !


!Algo42 methodsFor: 'consulta' stamp: 'JuanDausa 4/22/2011 11:55'!
leerArmas
      "Devuelve el número representativo de Armas"

      ^Armas.! !
```

```
!Algo42 methodsFor: 'accion' stamp: 'JuanDausa 5/3/2011 21:58'!
dispara: unManejador
      "dispara las armas del Algo42"

      |b1 b2 b3|
      b1  := NoRastreadora crearBNRConPos: (self posicion).
      b1  darEquipo: 'Malos'.
      b1  esUnCohete.
      unManejador agregarBala: b1 .
      (Armas = 2)       ifTrue: [
            b2  := NoRastreadora crearBNRConPos: (self posicion).
            b2 darEquipo: 'Buenos'.
            b2  esUnLaser.
            unManejador  agregarBala: b2.

            b3 := Rastreadora crearBRConPos: (self posicion) YCombustible:
(unManejador menorLimite).
            b3 darEquipo: 'Buenos'.
            unManejador agregarBala: b3.
      ].! !

!Algo42 methodsFor: 'accion' stamp: 'JuanDausa 5/7/2011 18:00'!
disparar: unManejador
      "dispara las armas del Algo42"

      |b1 b2 b3|
      b1  := NoRastreadora crearBNRConPos: (self posicion).
      b1  darEquipo: (1).
      b1  esUnCohete.
      unManejador agregarBala: b1 .
      (Armas = 2)       ifTrue: [
            b2  := NoRastreadora crearBNRConPos: (self posicion).
            b2 darEquipo: (1).
            b2  esUnLaser.
            unManejador  agregarBala: b2.

            b3 := Rastreadora crearBRConPos: (self posicion) YCombustible:
(unManejador menorLimite).
            b3 darEquipo: (1).
            unManejador agregarBala: b3.
      ].! !

!Algo42 methodsFor: 'accion' stamp: 'JuanDausa 4/20/2011 12:17'!
moverAbajo: unManejador
      ( Pos leerY = 1)
      ifTrue: [Energia := (Energia - 10)]
      ifFalse: [Pos darY: (Pos leerY - 1)].! !

!Algo42 methodsFor: 'accion' stamp: 'JuanDausa 4/22/2011 04:01'!
moverArriba: unManejador
      ( Pos leerY = ( unManejador limitey - 1 ) )
      ifTrue: [Energia := (Energia -10)]
      ifFalse: [Pos darY: (Pos leerY + 1)].! !

!Algo42 methodsFor: 'accion' stamp: 'JuanDausa 4/20/2011 12:18'!
moverDerecha: unManejador
      ( Pos leerX = (unManejador limitex -1))
```

```
      ifTrue: [Energia := (Energia - 10)]
      ifFalse: [Pos darX: (Pos leerX + 1)].! !

!Algo42 methodsFor: 'accion' stamp: 'JuanDausa 4/20/2011 12:19'!
moverIzquierda: unManejador
      ( Pos leerX = 1)
      ifTrue: [Energia := (Energia - 10)]
      ifFalse: [Pos darX: (Pos leerX - 1)].! !


!Algo42 methodsFor: 'controlesDePosicion' stamp: 'JuanDausa 5/10/2011
19:36'!
controlarPosicion: unManejador
      "Verifica la posicion del Algo42 y detecta daños ocasionados por
colisiones e impactos de bala. Ademas si encuentra items los usa"
      | listaDeItems|
      listaDeItems := unManejador levantarItems: (self espacioAereo).
      1 to: (listaDeItems size) do:[ :i |
            (listaDeItems at: i) usarEn: self ].
      self detectarColisiones: ( unManejador listaDeNaves ).
      self detectarImpactosDeBala: ( unManejador listaDeBalas ).

      ! !

!Algo42 methodsFor: 'controlesDePosicion' stamp: 'JuanDausa 5/4/2011
00:43'!
detectarColisiones: unaListaDenNaves
      "Verifica si la posicion de Algo42 es tambien ocupada por otras
naves, en caso de ser positivo daña a Algo42"

      | cantidadDeChoques espacioAereo |
      espacioAereo := self espacioAereo.
      unaListaDenNaves do: [ :nave |
            ( nave usasEsteEspacio: ( espacioAereo ) ) ifTrue:
            [ self quitarEnergia: (Constantes danioPorChoque ). ].
      ].
      1! !

!Algo42 methodsFor: 'controlesDePosicion' stamp: 'JuanDausa 5/7/2011
18:21'!
detectarImpactosDeBala: unaListaDeBalas
      "Detecta los impactos de bala sobre algo42"
      | unEspacioAereo |
      unEspacioAereo := self espacioAereo.
      unaListaDeBalas do: [ :unaBala | ( unaBala choque: unEspacioAereo )
ifTrue:
             [ unaBala daniar: self ].
      ].
      ! !

!Algo42 methodsFor: 'controlesDePosicion' stamp: 'JuanDausa 4/21/2011
20:39'!
posicionInicialEnX: unLimiteX EnY: unLimiteY

      Pos darX: (unLimiteX / 2).
      Pos darY: (unLimiteY -20).! !
```

```
!Algo42 methodsFor: 'Items' stamp: 'JuanDausa 4/21/2011 20:42'!
sumarArmas
      "Habilita todas las armas en Algo42"
       Armas := 2.! !

!Algo42 methodsFor: 'Items' stamp: 'JuanDausa 4/22/2011 11:50'!
sumarEnergia: unaCantidad
      "Aumenta en 'unaCantidad' la energia de Algo42"

      (Energia = 500) ifFalse: [Energia := (Energia + unaCantidad ).].! !

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!

Algo42 class
      instanceVariableNames: ''!

!Algo42 class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/7/2011 20:05'!
crearAlgo42En: unManejador
      "Insatancia un Algo42 y lo referencia en unManejador"
      | auxiliar |
      auxiliar := Algo42 new.
      auxiliar posicionInicialEnX: (unManejador limitex) EnY:
(unManejador limitey).
      [unManejador agregarAlgo42: auxiliar.] on: MasDeUnAlgo42 do:
[MasDeUnAlgo42 new signal].
      ^auxiliar.


! !

Object subclass: #Posicion
      instanceVariableNames: 'X Y'
      classVariableNames: ''
      poolDictionaries: ''
      category: '1942'!

!Posicion methodsFor: 'setear' stamp: 'JuanDausa 4/21/2011 15:13'!
darX: unx
      "Graba un y en 'Posicion'"

      X := unx.
! !

!Posicion methodsFor: 'setear' stamp: 'JuanDausa 4/21/2011 15:13'!
darY: uny
      "Graba un y en 'Posicion'"

      Y := uny.
      ! !


!Posicion methodsFor: 'leer' stamp: 'JuanDausa 4/21/2011 15:13'!
leerX
      "Devuelve el valor de x"
      ^X! !
```

```
!Posicion methodsFor: 'leer' stamp: 'JuanDausa 4/21/2011 15:13'!
leerY
      "Devuelve el valor de y"
      ^Y! !


!Posicion methodsFor: 'uso' stamp: 'JuanDausa 4/21/2011 15:13'!
distancia: otroPunto
      "Devuelve la distancia entre un punto y otro"
      | dx dy |
      dx := otroPunto leerX - X.
      dy := otroPunto leerY - Y.
      ^ (dx * dx + (dy * dy)) sqrt! !

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!

Posicion class
      instanceVariableNames: ''!

!Posicion class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
4/22/2011 16:27'!
crearPosicion: Pos
      "Crea una posicion"
      |a|
      a := Posicion new.
      a darX: (Pos leerX).
      a darY: (Pos leerY).
      ^a! !

!Posicion class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
4/17/2011 18:08'!
crearPosicionX:unx Y:uny
      "Crea una posicion"
      |a|
      a := Posicion new.
      a darX: unx.
      a darY: uny.
      ^a! !

Object subclass: #Manejador
      instanceVariableNames: 'Maxx Maxy ObjetosB ObjetosN Algo_42 Puntos
Items'
      classVariableNames: ''
      poolDictionaries: ''
      category: '1942'!
!Manejador commentStamp: '<historical>' prior: 0!
Esta clase contiene información general del nivel, limistes de
movimiento, cantidad de objetos, una referncia a ellos, etc. Esta clase
no se encarga del comportamiento sino de la comunicación entre objetos.!


!Manejador methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
20:04'!
devolverGuia: unEntero
      "Establece la nave guia"
      ( unEntero > ( ObjetosN size ) ) ifTrue: [ GuiaIncorrecto new
```

```
signal ].
      ( ObjetosN at: unEntero ) serGuia.
      ^(ObjetosN at: unEntero). ! !

!Manejador methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
03:46'!
initialize
      ObjetosB := OrderedCollection new.
      Items := OrderedCollection new.
      ObjetosN := OrderedCollection new.
      Puntos := 0.
      Algo_42 = nil.! !

!Manejador methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
12:31'!
levantarItems: unEspacioAereo
      "Inspecciona el espacio aereo en busca de items que puedan ser
adquiridos."

      | lista |
      lista:= OrderedCollection new.
      1 to: (unEspacioAereo size) do: [:i |
            1 to: (Items size) do: [ :j |
                  ( ( ( unEspacioAereo at: i ) distancia: ( ( Items at:
j ) posicion ) )= 0) ifTrue:
                        [ lista add: (Items at: j).]
            ]
      .].
      1 to: (lista size) do: [:i | Items remove: (lista at: i).].
      ^lista.
! !


!Manejador methodsFor: 'agregarobjetos' stamp: 'JuanDausa 4/20/2011
12:13'!
agregarAlgo42: unAlgo42
      "Agrega un Algo42 en el manejador"
      (Algo_42 = nil)
      ifTrue: [  Algo_42 := unAlgo42] ifFalse: [MasDeUnAlgo42 new
signal].
            ! !

!Manejador methodsFor: 'agregarobjetos' stamp: 'JuanDausa 4/18/2011
12:33'!
agregarBala: unaBala
      "Agrega una bala a la lista de objetos"
      ObjetosB add: unaBala.! !

!Manejador methodsFor: 'agregarobjetos' stamp: 'JuanDausa 4/17/2011
18:00'!
agregarEnemigo: unaNave
      "Agrega naves enemigas al motor"

      ObjetosN add: unaNave.
! !

!Manejador methodsFor: 'agregarobjetos' stamp: 'JuanDausa 4/21/2011
```

```
20:30'!
agregarItem: unItem
      "Agrega un item a la lista"
      Items add: unItem.
! !


!Manejador methodsFor: 'quitarobjetos' stamp: 'JuanDausa 4/22/2011
13:19'!
quitarBala: unaBala
      "Quita la referencia de la bala"
      (ObjetosB size = 0) ifTrue:[ErrorDeReferencias new signal].
      1 to: (ObjetosB size) do: [ :i | (unaBala = (ObjetosB at: i))
            ifTrue: [ObjetosB removeAt: i ].
      ].! !

!Manejador methodsFor: 'quitarobjetos' stamp: 'JuanDausa 4/22/2011
03:21'!
quitarNave: unaNave
      "Quita la referencia de la Nave"
      |aux|
      (ObjetosN size = 0) ifTrue:[ErrorDeReferencias new signal].
      1 to: (ObjetosN size) do: [ :i | (unaNave = (ObjetosN at: i))
            ifTrue: [aux := i ].
      ].
      ObjetosN removeAt: aux.
      ! !


!Manejador methodsFor: 'consulta' stamp: 'JuanDausa 4/22/2011 03:37'!
cantidadDeBalas
      "Devuelve la cantidad de elementos (objetos) que esta siendo
usados. Uso Interno"

      ^(ObjetosB size).! !

!Manejador methodsFor: 'consulta' stamp: 'JuanDausa 4/22/2011 03:36'!
cantidadDeElementos
      "Devuelve la cantidad de elementos (objetos) que esta siendo
usados. Uso Interno"

      ^((ObjetosB size) + (ObjetosN size)).! !

!Manejador methodsFor: 'consulta' stamp: 'JuanDausa 4/22/2011 03:15'!
cantidadDeItems
      "Devuelve la cantidad de items en el mapa."
      ^(Items size).! !

!Manejador methodsFor: 'consulta' stamp: 'JuanDausa 4/22/2011 20:11'!
cantidadDeNaves
      "Devuelve la cantidad de Naves"
      ^(ObjetosN size).! !

!Manejador methodsFor: 'consulta' stamp: 'JuanDausa 5/7/2011 20:01'!
devolverAlgo42
      "Devuelve una referencia a algo42"
```

```
      ^Algo_42.
! !

!Manejador methodsFor: 'consulta' stamp: 'JuanDausa 4/17/2011 03:10'!
limitex
      "Devuelve el limite en x"

       ^Maxx! !

!Manejador methodsFor: 'consulta' stamp: 'JuanDausa 4/17/2011 03:10'!
limitey
      "Devuelve el limite en y"

      ^Maxy! !

!Manejador methodsFor: 'consulta' stamp: 'JuanDausa 5/7/2011 18:10'!
listaDeBalas
      "Privado"
      ^ObjetosB.! !

!Manejador methodsFor: 'consulta' stamp: 'JuanDausa 5/3/2011 23:45'!
listaDeNaves
      "Devuelve una lista con las naves que estan participando."
      "Privado"
      ^ObjetosN.
! !

!Manejador methodsFor: 'consulta' stamp: 'JuanDausa 4/21/2011 12:13'!
menorLimite
      "Devulve el menor de los limites"
      (self limitex < self limitey)
      ifTrue: [^self limitex] ifFalse: [^self limitey] .! !

!Manejador methodsFor: 'consulta' stamp: 'JuanDausa 4/21/2011 18:04'!
posAlgo42
      "Devuelve la posicion de Algo42"

      | pos |

      pos := Posicion new.
      pos darX:((Algo_42 posicion) leerX).
      pos darY:((Algo_42 posicion) leerY).
      ^pos.! !

!Manejador methodsFor: 'consulta' stamp: 'JuanDausa 5/7/2011 18:09'!
posMasCercana: unaPosicion
      "Devuelve la posicion enemiga más cercana"
      | pbuscada distancia names |
      (ObjetosN size > 0) ifTrue:[
            pbuscada := (ObjetosN at: 1) posicion.
            distancia := (pbuscada distancia: unaPosicion).
            "Me quedo con la primera"
            2 to: (ObjetosN size) do: [ :i |
                  ((ObjetosN at: i) distancia: unaPosicion < distancia)
ifTrue:
                        [pbuscada := (ObjetosN at: i).
                        distancia := (ObjetosN at: i) distancia:
```

```
unaPosicion.].
                ]
     ] ifFalse: [
          pbuscada := Posicion new.
          pbuscada darX: (Maxx /2).
          pbuscada darY: 0.
          ].
     ^pbuscada.! !


!Manejador methodsFor: 'colisiones' stamp: 'JuanDausa 4/21/2011 15:44'!
choqueConAlgo42: unEspacioAereo
     "Verifica si la posicion de Algo42 coincide con la del otro objeto
volador"
     | listaAuxiliar variable names |
     listaAuxiliar := Algo_42 espacioAereo.
     1 to: (listaAuxiliar size) do: [ :i |
          1 to: (unEspacioAereo size) do: [ :j |
               (listaAuxiliar at: i = unEspacioAereo at: j) ifTrue:
[^true]
          ]
     ].! !

!Manejador methodsFor: 'colisiones' stamp: 'JuanDausa 5/7/2011 18:21'!
choqueConBala: unaNave
     "Verifija si la nave ah chocado con alguna bala e indica que debe
efectuarse el daño. Ademas si la nave no posee más Energia devuelve el
Equipo que la destruyó"
     | espacio balasUsadas auxiliar|
     balasUsadas := OrderedCollection new.
     espacio := unaNave espacioAereo.
     auxiliar := ''.
     1 to: (ObjetosB size) do: [ :i | ((ObjetosB at: i) choque: espacio)
          ifTrue:
               [(ObjetosB at: i) daniar: unaNave.
               (unaNave leerEnergia = 0)
               ifTrue:[
                    auxiliar := (ObjetosB at: i) leerEquipo.
                    espacio := OrderedCollection new.].
                    balasUsadas add: (ObjetosB at: i).
          ].
     ].
     1 to: (balasUsadas size) do: [ :i | ObjetosB remove: ( balasUsadas
at: i ) ].
     (unaNave leerEnergia = 0) ifTrue: [^auxiliar ].
     ! !

!Manejador methodsFor: 'colisiones' stamp: 'JuanDausa 5/7/2011 18:21'!
choqueConBalaBuena: unaNave
     "Dada un nave inspecciona si esta choco con una bala del Algo42"
     | espacio listaDeUsadas |
     espacio := unaNave espacioAereo.
     listaDeUsadas := OrderedCollection new.
     1 to: (ObjetosB size) do: [ :i |
          ((ObjetosB at: i) leerEquipo = 'Buenos') ifTrue: [
               ((ObjetosB at: i) choque: espacio)
               ifTrue: [
```

```
                    (ObjetosB at: i) daniar: unaNave.
                    listaDeUsadas add: ( ObjetosB at: i ).
               ].
          ].
     ].

     1 to: ( listaDeUsadas size) do: [ :i | ObjetosB remove:
( listaDeUsadas at: i ) ].! !

!Manejador methodsFor: 'colisiones' stamp: 'JuanDausa 5/7/2011 18:21'!
choqueConBalaMala: unaNave
     "Dada un nave inspecciona si esta choco con una bala del Algo42,
informa esta situacion a demas objetos para que hagan lo necesario"
     | espacio listaDeUsadas|
     listaDeUsadas := OrderedCollection new.
     espacio := unaNave espacioAereo.

     1 to: (ObjetosB size) do: [ :i |
          ( (ObjetosB at: i ) leerEquipo = 'Malos' ) ifTrue: [
               (  ( ObjetosB at: i) choque: espacio )
               ifTrue: [
                    (ObjetosB at: i) daniar: unaNave.
                    listaDeUsadas add: ( ObjetosB at: i).

               ].
          ].
     ].
     1 to: ( listaDeUsadas size ) do: [ :i | ObjetosB remove:
(listaDeUsadas at: i) ].


! !

!Manejador methodsFor: 'colisiones' stamp: 'JuanDausa 4/22/2011 18:27'!
choqueConEnemigo: unAlgo42
     "Dada un nave inspecciona si esta choco con Algo42"
     | espacio |
     espacio := unAlgo42 espacioAereo.
     1 to: (ObjetosN size) do: [ :i |
          ( ( ObjetosN at: i ) choque: ( unAlgo42 espacioAereo ) )
ifTrue: [unAlgo42 quitarEnergia: 50].
     ].

! !


!Manejador methodsFor: 'usodelprogramador' stamp: 'JuanDausa 4/17/2011
18:58'!
darLimiteX: unx LimiteY: uny

     Maxx := unx.
     Maxy := uny.! !


!Manejador methodsFor: 'puntos' stamp: 'JuanDausa 4/22/2011 18:54'!
leerPuntos
     "Devuelve los puntos del manejador"
```

```
	^Puntos.! !

!Manejador methodsFor: 'puntos' stamp: 'JuanDausa 4/21/2011 15:50'!
quitarPuntos: unaCantidad
	"Quita unaCantidad de puntos"

	|  |
	(Puntos < unaCantidad) ifFalse:[
		Puntos := Puntos - unaCantidad].
	! !

!Manejador methodsFor: 'puntos' stamp: 'JuanDausa 4/21/2011 17:29'!
sumarPuntos: unaCantidad
	"Suma 'unaCantidad' a los puntos"

	Puntos := (Puntos + unaCantidad) .! !

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!

Manejador class
	instanceVariableNames: ''!

!Manejador class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
4/19/2011 16:41'!
crearConLX:unEntero LY:otroEntero
	"Crea un motor con limites x,y e inicializa variables de clase"
	| m |

	((unEntero>300) and: [otroEntero>400]) ifFalse: [LimitesIncorrectos
new signal].

	m := Manejador new.
	m darLimiteX: unEntero LimiteY: otroEntero.
	^m



		! !

Enemiga subclass: #Avioneta
	instanceVariableNames: 'Direccion'
	classVariableNames: ''
	poolDictionaries: ''
	category: '1942'!
!Avioneta commentStamp: '<historical>' prior: 0!
Vuelan en Idas y vueltas en línea recta , son los aviones más rápidos.

Instance Variables:
	Direccion  <Integer>!


!Avioneta methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/7/2011
18:02'!
disparar: unMotor
	"Crea una bala y la inicializa. Ademas deja un referencia en el
motor"
```

```
      |b|

      b := NoRastreadora crearBNRConPos: (self posicion).
      b darEquipo: 2.
      b esUnLaser.
      unMotor agregarBala: b! !

!Avioneta methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/11/2011
13:58'!
initialize
      Pos := Posicion new.
      Energia := ( Constantes energiaAvioneta ).
      Direccion := 1.
      Ocupacion := ( Constantes ocupacionAvioneta ).
      Guia := 0.! !

!Avioneta methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
02:46'!
mover: unManejador
      "Mueve a una avioneta, estas se mueven en linea recta, ida y
vuelta."

      | auxiliar |
      "Comprabaciones"
      auxiliar := Pos leerY.
      (Direccion = 1)
            ifTrue: [
                  (unManejador limitey > (auxiliar + 1))
                        ifTrue: [ Pos darY: (auxiliar + 1). ]
                        ifFalse: [ Direccion := ( 0 - 1) .]
            ] ifFalse: [
                        (Pos leerY = 1)
                        ifTrue: [ Direccion = 1]
                        ifFalse: [ Pos darY: (auxiliar - 1)]
                        ].
      ! !

!Avioneta methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/22/2011
02:44'!
posIinicialConLimite: unLimite
      "Define una posición inicial dado cierto limite"

      Pos darX: (unLimite atRandom).
      Pos darY: 1.
      ! !

!Avioneta methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
20:19'!
sumarPuntos: unManejador
      "Suma los puntos despectivos por la muerte de la nave"

      unManejador sumarPuntos: 20.! !

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!

Avioneta class
      instanceVariableNames: ''!
```

```
!Avioneta class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/3/2011 21:59'!
crearAvionetaEn: unMotor
      "Crea una avineta y la inicializa"

      |a|

      a := Avioneta new.
      a posIinicialConLimite: (unMotor limitex).
      unMotor agregarEnemigo: a.

      ^a.! !

Neutral subclass: #AvionCivil
      instanceVariableNames: ''
      classVariableNames: ''
      poolDictionaries: ''
      category: '1942'!

!AvionCivil methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/11/2011
13:56'!
initialize
      Pos := Posicion new.
      Energia := ( Constantes energiaAvionCivil ).
      Ocupacion := ( Constantes ocupacionAvionCivil ).! !

!AvionCivil methodsFor: 'as yet unclassified' stamp: 'JuanDausa 4/21/2011
13:19'!
mover: unManejador
      "Mueve a los aviones civiles en linea recta."
      Pos darX: (Pos leerX + 1).
      Pos darY: (Pos leerY + 1).
      (Pos leerY >= unManejador limitey or: [Pos leerX >= unManejador
limitex])
      ifTrue: [ self morir: unManejador]
! !

!AvionCivil methodsFor: 'as yet unclassified' stamp: 'JuanDausa 5/12/2011
09:53'!
vivir: unManejador
      "Deja hacer lo que tiene que hacer un Avion civil"
      | equipo algo42|
      algo42 := ( unManejador devolverAlgo42 ).
      equipo := ''.
      self mover: unManejador.
      self detectarColisiones: algo42.
      "Aunque se sabe que solo choca contra algo42, esta comprobación se
realiza porque esto puede cambiar"
      ( ( self leerEnergia = 0 ) and: [ algo42 usasEsteEspacio: ( self
espacioAereo ) ] )
            ifTrue:
            [
            unManejador quitarPuntos: 300.
            self morir: unManejador]
      ifFalse: [
            equipo := self detectarImpactosDeBalaYEquipo: ( unManejador
```

```
listaDeBalas ).
            ] .
      ( equipo = 1 or: [ equipo = 2 ] )
      ifTrue: [
            self morir: unManejador
            ].

      ( equipo = 1 ) ifTrue: [ unManejador quitarPuntos: 300 ].! !

"-- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- "!

AvionCivil class
      instanceVariableNames: ''!

!AvionCivil class methodsFor: 'as yet unclassified' stamp: 'JuanDausa
5/6/2011 19:29'!
crearAvionEn: unManejador
      |auxiliar|
      auxiliar := AvionCivil new.
      auxiliar posIinicialConLimite: unManejador limitex.
      unManejador agregarEnemigo: auxiliar.
      ^auxiliar ! !
```

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?

Código

Generalidades

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