**PROGRAMACION ORIENTADA A OBJETOS**

**Viaje Espacial**

**Manual del Programador**

**Semestre 2016-2017/I**

* No de proyecto: 113
* Nombre del proyecto: Viaje Espacial.
* Integrantes del equipo:

Hernández Cuellar José Javier (237506).

Amaya García José de Jesús (220083).

* Objetivo del proyecto:

La nave debe cruzar el campo de asteroides para llevar reservas a la estación espacial.

* Descripción del proyecto:

El escenario se encuentra en el espacio exterior y la nave entra en un campo de asteroides los cuales poseen características distintas en cuanto a tamaño y material del que se componen.

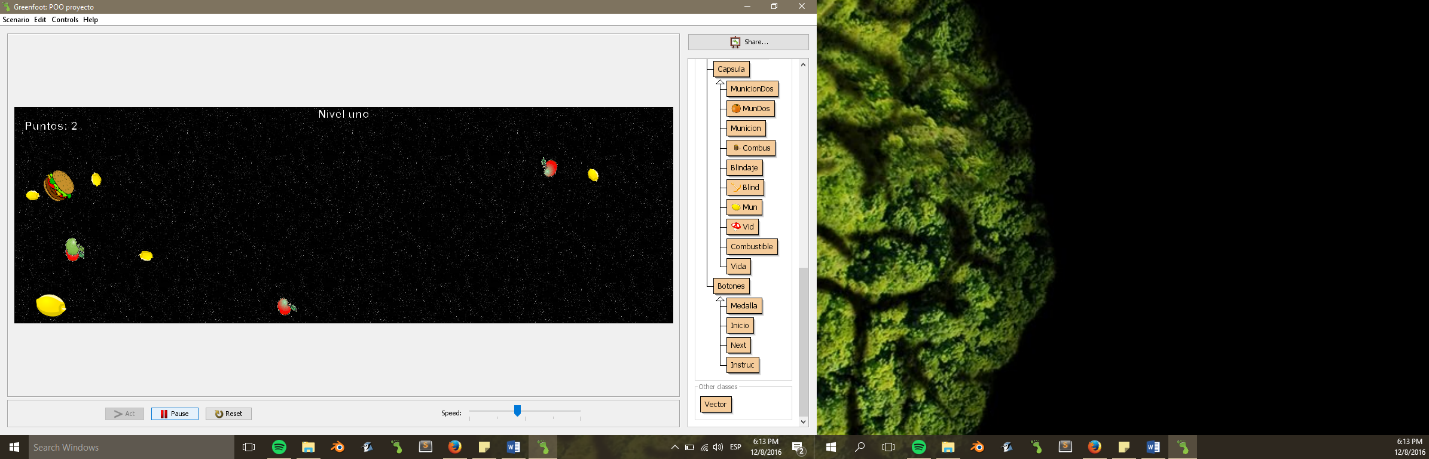
En el juego también puede encontrar objetos que le dan puntos. Dependiendo del desempeño se calificará la partida con medallas. Además el juego es de eliminación directa a menos que obtenga capsulas de vida. El jugador puede elegir una de entre dos naves para jugar.

Naves:

* NaveUno: La nave y las balas que lanza son más grandes pero lo nave también es más lenta.
* NaveDos: La nave se mueve más rápido pero el tamaño es más pequeño al igual que las balas.

Descripción e imágenes de cada nivel:

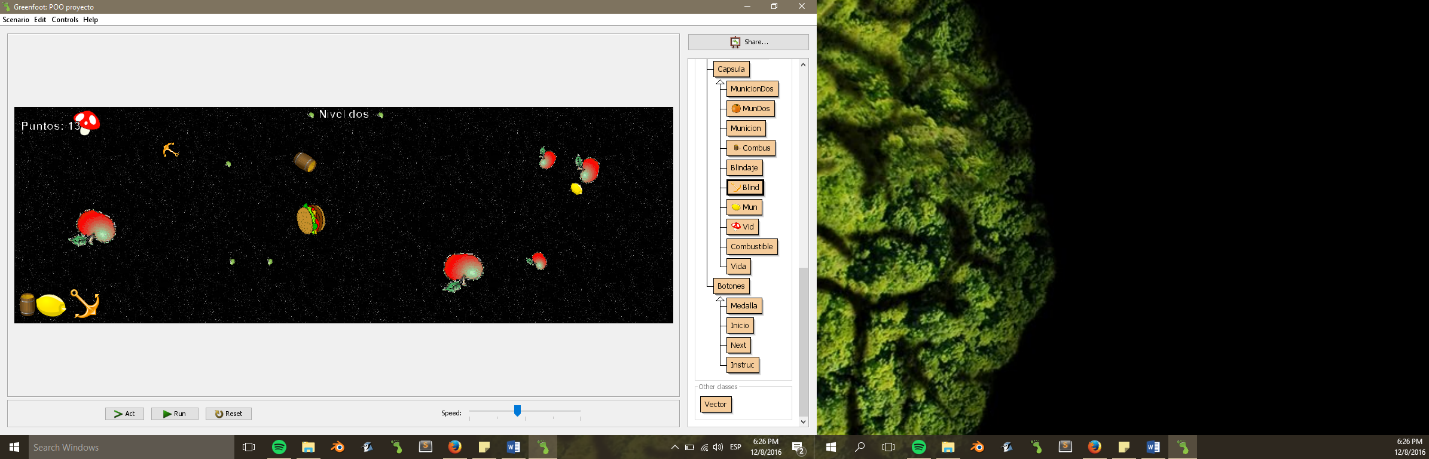
* Nivel Principiante



El jugador necesita más de 7 puntos para pasar al siguiente nivel, no debe ser tocado por los asteroides rojo y verde (blindado), puede disparar con la techa “space” y moverse con las flechas “izquierda”, “derecha” y “arriba” para acelerar, el limón en la parte inferior izquierda indica que tienes munición y por lo tanto no puedes recargar hasta que se hayan agotado las balas.

Para recargar munición debes tener contacto con alguno de los limones que aparecen florando en el espacio.

En la esquina superior izquierda se muestran los puntos acumulados y en el centro arriba el nivel en el que te encuentras.

* Nivel Intermedio

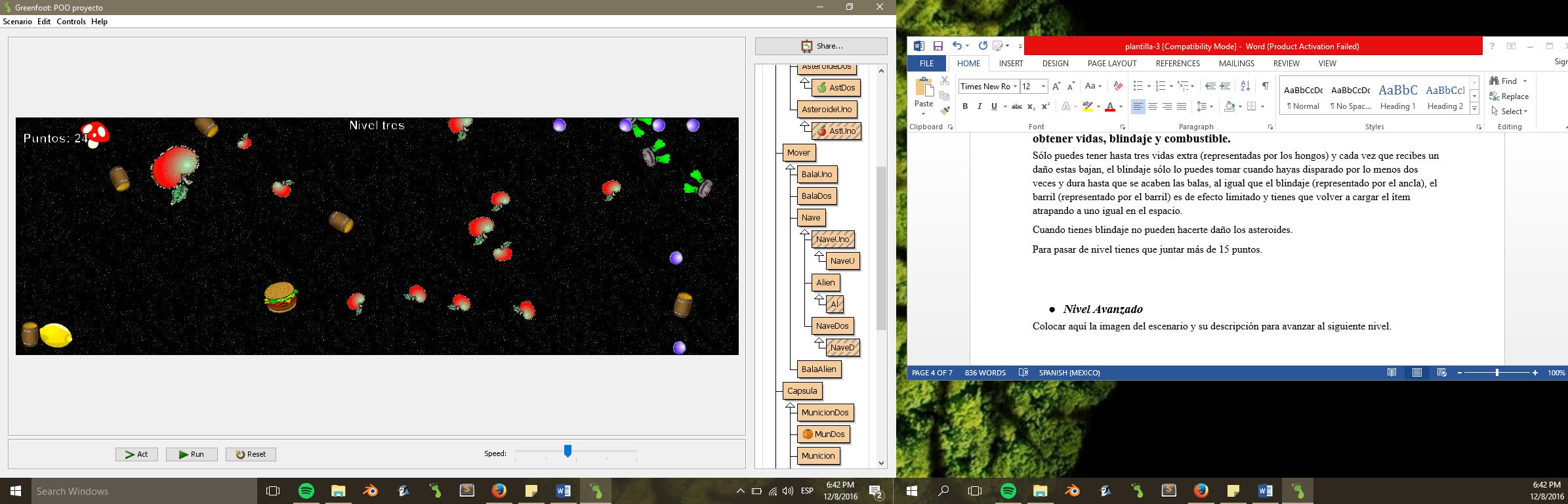
En este nivel parece que sólo salen asteroides rojos pero cuando les disparas se convierten en pequeños verdes, aquí la velocidad aumenta pero ahora puedes obtener vidas, blindaje y combustible.

Sólo puedes tener hasta tres vidas extra (representadas por los hongos) y cada vez que recibes un daño estas bajan, el blindaje sólo lo puedes tomar cuando hayas disparado por lo menos dos veces y dura hasta que se acaben las balas, al igual que el blindaje (representado por el ancla), el barril (representado por el barril) es de efecto limitado y tienes que volver a cargar el ítem atrapando a uno igual en el espacio.

Cuando tienes blindaje no pueden hacerte daño los asteroides.

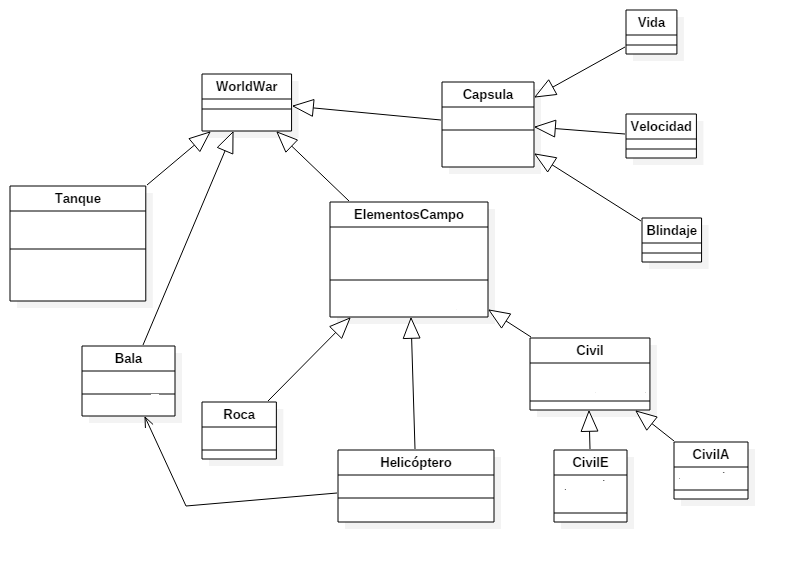
Para pasar de nivel tienes que juntar más de 15 puntos.

* Nivel Avanzado



Para llegar a este nivel sólo se necesitan más de **15 puntos, aparecen únicamente** asteroides rojos de diferente tamaño y robots alienígenas que quieren destruirte, sólo puedes deshacerte de los robots con las balas de oro que puedes escoger oprimiendo la tecla “b”, cuando te hayan matado aparece un mensaje con Game Over y las medallas que obtuviste con tu puntuación.

* Diagrama de clases UML

1. 

* Características y comportamiento de cada clase

|  |  |
| --- | --- |
| Nombre de la clase: | Mover |
| Características: | Ésta es la clase que permite el movimiento vectorial de las naves. |
| Comportamiento: | [getLocacionX](about:blankMover.html#getLocacionX())() Método getter para la locación en x. |
|  | [getLocacionY](about:blankMover.html#getLocacionY())() Método getter para la locación en y. |
|  | [getMovimiento](about:blankMover.html#getMovimiento())() Retorna el movimiento. |
|  | [getx](about:blankMover.html#getx(int))(int xn) |
|  | [gety](about:blankMover.html#gety(int))(int yn) |
|  | [incrementaVelocidad](about:blankMover.html#incrementaVelocidad(Vector))(Vector s) Incrementa la velocidad con el vector dado. |
|  | [mover](about:blankMover.html#mover())() Método del movimiento para las naves uno y dos. |
|  | [mover](about:blankMover.html#mover(double, double))(double xi, double yi) Permite settear la locación donde empieza el movimiento. |
|  | [moverAlien](about:blankMover.html#moverAlien())() Metodo que permite el movimiento de los robots. |
|  | [setLocacion](about:blankMover.html#setLocacion(double, double))(double x, double y) |

|  |  |
| --- | --- |
| Nombre de la clase: | Asteroide |
| Características: | Es la clase que contiene a todo el comportamiento de cualquiera de los dos asteroides. |
| Comportamiento: | [detectaBala](about:blankAsteroide.html#detectaBala(int))(int selec) Aquí se indican las condiciones para que el asteroide pueda ser eliminado cuando una bala de la nave los toque o que lleguen al final del mundo. |
|  | [setVelocidad](about:blankAsteroide.html#setVelocidad(int))(int vel) Método setter para la velocidad del asteroide. |

|  |  |
| --- | --- |
| Nombre de la clase: | Capsula |
| Características: | Ésta es la clase padre de todos los ítems y controla el movimiento y que desaparezcan cuando llegan al final del mundo. |
| Comportamiento: | [comportamiento](about:blankCapsula.html#comportamiento())() |
|  | [setVelocidad](about:blankCapsula.html#setVelocidad(int))(int vel) Método que setter para la velocidad del ítem. |

|  |  |
| --- | --- |
| Nombre de la clase: | Nave |
| Características: | La nave se puede mover con "up","left","right". |
| Comportamiento: | [comandos](about:blankNave.html#comandos(boolean))(boolean var) |
|  | [detectarObjeto](about:blankNave.html#detectarObjeto())() |
|  | [getBanderaMunicion](about:blankNave.html#getBanderaMunicion())() |
|  | [getBlindaje](about:blankNave.html#getBlindaje())() |
|  | [getVida](about:blankNave.html#getVida())() |
|  | [recargaContador](about:blankNave.html#recargaContador())() Método que aumenta en uno la variable recargaContador bala uno. |
|  | [recargaContadorAlien](about:blankNave.html#recargaContadorAlien())() Método que aumenta en uno la variable recargaContadorAlien. |
|  | [recargaContadorNaranja](about:blankNave.html#recargaContadorNaranja())() Método que aumenta en uno la variable recargaContador bala dos. |
|  | [remueveItem](about:blankNave.html#remueveItem())() |
|  | [setAceleracion](about:blankNave.html#setAceleracion(Vector))(Vector aceleracion) Método setter para la aceleración. |
|  | [setArmaTiempo](about:blankNave.html#setArmaTiempo(int))(int arma) Método setter para el arma de tiempo bala uno. |
|  | [setArmaTiempoAlien](about:blankNave.html#setArmaTiempoAlien(int))(int arma) Método setter para el arma de tiempo bala alien. |
|  | [setBanderaCombustibleNave](about:blankNave.html#setBanderaCombustibleNave(boolean))(boolean daro) Método setter para la bandera de combustible de la nave. |
|  | [setBoolean](about:blankNave.html#setBoolean(boolean))(boolean bandera) Para aceptar la munición. |
|  | [setIncrementaVelocidad](about:blankNave.html#setIncrementaVelocidad(Vector))(Vector velocidad) Método que incrementa la velocidad de la nave. |
|  | [setLimiteBalas](about:blankNave.html#setLimiteBalas(int))(int limiteBalas) Método setter para el límite de balas que puede disparar la nave. |
|  | [setLimiteBalasNaranja](about:blankNave.html#setLimiteBalasNaranja(int))(int limiteBalasNaranja) |
|  | [setLimiteMov](about:blankNave.html#setLimiteMov(int))(int limiteMov) Método setter para el límite de movimientos que puede tener la nave. |

|  |  |
| --- | --- |
| Nombre de la clase: | Vector |
| Características: | Vector en 2D |
| Comportamiento: | [agrega](about:blankVector.html#agrega(Vector))([Vector](about:blankVector.html) otro) Agrega el vector dado. |
|  | [copia](about:blankVector.html#copia())() Crea una copia del vector. |
|  | [getDireccion](about:blankVector.html#getDireccion())() Devuelve la dirección (en grados). |
|  | [getLongitud](about:blankVector.html#getLongitud())() Devuelve la longitud del vector. |
|  | [getX](about:blankVector.html#getX())() Devuelve la x del vector. |
|  | [getY](about:blankVector.html#getY())() Devuelve la y del vector. |
|  | [setDireccion](about:blankVector.html#setDireccion(int))(int direccion) Establece la dirección del vector. |

|  |  |
| --- | --- |
| Nombre de la clase: | Botones |
| Características: | Ésta clase contiene cada método que define el comportamiento de los botones. |
| Comportamiento: | [gameOver](about:blankBotones.html#gameOver())() Método para el botón de las medallas que hacen que regreses a la portada. |
|  | [Inst](about:blankBotones.html#Inst())() Método para el botón que te lleva al mundo de las instrucciones. |
|  | [next](about:blankBotones.html#next())() Método para el botón que hace que avances en el mundo de instrucciones. |
|  | [play](about:blankBotones.html#play())()  Método para el botón que inicia el juego. |

* Herencia, polimorfismo y encapsulamiento.

Uno de las clases donde se aplicó la propiedad de herencia en nuestro programa es en la clase abstracta Nave donde se definen todos los métodos que utilizan en común NaveUno y NaveDos, en dichas clases sólo se tiene que escribir el método y en algunos casos pasar como parámetro la variable con la que tiene que implementar de acuerdo a las características que se mencionaron en la descripción del proyecto.

Además del ejemplo antes mencionado también se utilizaron métodos heredados en las clases AsteroideUno y AsteroideDos, en la clase padre Botones, etc

También se debe mencionar la encapsulación que se ve reflejada en todas las variables privadas de las clases y que sólo pueden ser accedidas mediante métodos getter y setter.

El polimorfismo se ve en la sobrecarga de métodos o constructores; en muchos casos nos tuvimos que definir más de un constructor por clase para poder mantener su relación con las clases hijas.

* Cronograma de actividades (plan de trabajo)

|  |  |  |
| --- | --- | --- |
| Fecha de Inicio | Fecha de Término | Actividad por realizar |
| 19/oct | 07/dic | Comienzo en la elaboración del proyecto |
| 07/dic | 09/dic | Manual de usuario |
| 07/dic | 09/dic | Manual del programador |
| 07/dic | 09/dic | Video y el resto de la documentación |

* Bitácora de actividades (historial)

Aquí se describen cada una de las actividades realizadas desde la propuesta hasta la entrega del proyecto.

|  |  |
| --- | --- |
| Fecha (dd/mm/aa) | Descripción de la actividad realizada |
| 03/10/16 | Elaboración de la propuesta de proyecto. |
| 05/10/16 | Elaboración del diagrama de clases y objetos. |
| 25/10/16 | Creación de movimiento de las naves. |
| 01/11/16 | Elaboración de la bala y los asteroides. |
| 03/11/16 | Elaboración de las capsulas que participan en el juego. |
| 20/11/16 | Creación de la portada e instrucciones. |
| 22/11/16 | Creación de la segunda nave. |
| 30/11/16 | Creación de la segunda bala. |
| 05/12/16 | Creación del robot enemigo. |
| 07/12/16 | Revisión del proyecto. |
| 08/12/16 | Creación del manual de usuario y video. |
| 09/12/16 | Documentación de código y creación del manual del programador. |