



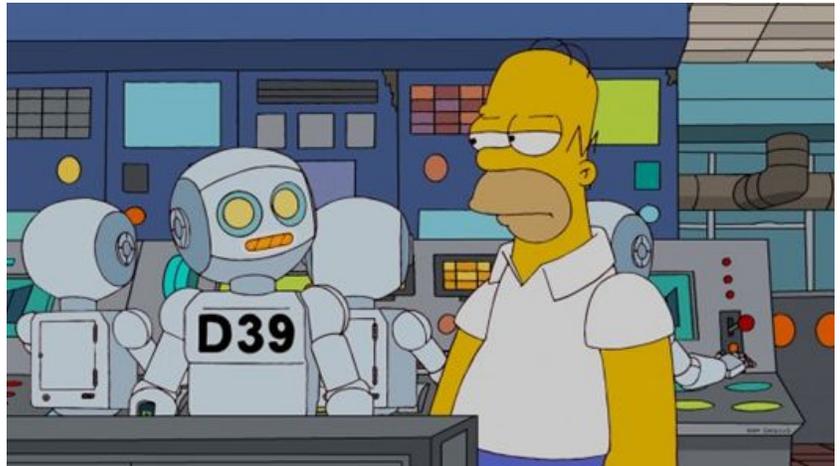
Práctico 3

Diseño Orientado a Objetos

Material de Lectura: Capítulo 7 Meyer, Guía UML

Base Starcraft

La base espacial del planeta Starcraft está habitada por diferentes seres, cada uno de ellos cumpliendo un rol distinto dentro del funcionamiento de la base. Los humanos son naturalmente quienes están a cargo de la dirección. Como todos, poseen un nombre y un número de legajo. Los robots recolectores son los encargados de explorar y recolectar el titanio de los alrededores, donde cada uno posee un número de identificación. Cuentan con un compartimiento donde almacenan provisoriamente el titanio hasta dejarlo en la Fábrica de Procesamiento. Los robots siempre llevan control del titanio que acarrean. El titanio crece en piedras de exactamente 10 Kilos, y los robots pueden juntar una piedra a la vez, operación que pueden realizar siempre y cuando no hayan superado su capacidad máxima de carga que es de 550 kilos. La Fábrica está dirigida por el Ingeniero Turing desde hace varios años. Cuenta con cinco contenedores, cada uno con un código de identificación y una especificación de la cantidad de toneladas que admite, así como otra con las que actualmente están ocupadas. Los robots siempre depositan el titanio en cualquier contenedor disponible.



- a) Realice un diagrama UML en donde se puedan observar las relaciones entre las clases que modelan este sistema. Complete cada diagrama de clases con su nombre, atributos, operaciones.

Laboratorio de Computadoras

Un laboratorio de computación de una cierta universidad cuenta con 20 escritorios aptos para colocar las computadoras que usarán los alumnos. Se dispone en stock de todo lo necesario para poder armar cada una de las máquinas: gabinetes, monitores, teclados y ratones, cada uno posiblemente de diferentes marcas o modelos. Cada vez que se arma una nueva computadora, se verifica que existan los componentes necesarios en stock, se los retira y se los ubica en su posición final correspondiente en uno de los escritorios. Usualmente se solicita a los técnicos del laboratorio que instalen una computadora específica, donde se indican tipo de gabinete, monitor, etc. Otras veces, el requerimiento puede especificar sólo el armado de la máquina, sin expresar preferencia por marcas o modelos de sus componentes, en cuyo caso se arma la computadora con los materiales disponibles.

a) Realice un diagrama UML en donde se puedan observar las relaciones entre las clases que modelan este sistema. Complete cada diagrama de clases con su nombre, atributos, operaciones.

Alfa1 y Omega6

Dos computadoras cuyos códigos son Alfa1 y Omega6 son la puerta de acceso a Internet de una fábrica de cajas. Las dos computadoras reciben mensajes de la red y utilizan un almacenamiento compartido en donde los mensajes se organizan bajo una política de "último en entrar, primero en salir". Los mensajes contienen un texto (es decir, una cadena de caracteres), y el nombre del usuario al cual está destinado el mensaje. Cuando una computadora recibe un mensaje, construye un paquete conteniendo el mensaje, junto con su propio código de identificación, más un número de serie que cada computadora asigna a los mensajes que van arribando, enviando finalmente este paquete al medio de almacenamiento compartido. Las computadoras pueden retirar un mensaje de este medio, o bien devolverlo, pero siempre siguiendo la política de manejo de una pila. Para ello se requiere el siguiente procedimiento: supongamos que la computadora X quiere retirar un mensaje. Si en el tope del medio de almacenamiento existe un paquete propio, lo retira y devuelve el mensaje. De no ser así, le solicita a la otra computadora Y que retire temporalmente todos los paquetes que le correspondan a Y, hasta que en el tope quede un paquete de X. Luego de esto, X ya puede retirar el paquete que será devuelto. Inmediatamente después, la computadora Y regresa nuevamente sus paquetes al medio de almacenamiento. De no existir ningún paquete para X en todo el medio de almacenamiento compartido, X debe devolver un mensaje especial con el texto "no se encontraron mensajes" y con el código de la computadora X como nombre de usuario. El medio de almacenamiento compartido mantiene, por razones de seguridad, una copia del último mensaje eliminado y un indicador de qué computadora realizó esa extracción. Sólo esta computadora puede solicitar un "deshacer", revirtiendo la operación de extracción del mensaje.

a) Realice un diagrama UML en donde se puedan observar las relaciones entre las clases que modelan este sistema. Complete cada diagrama de clases con su nombre, atributos, operaciones.

Zapp Brannigan

Un ejército liderado por Zapp Brannigan está organizado en regimientos y cuarteles generales. Cada cuartel general puede tener a su cargo hasta 3 regimientos. En el cuartel general conviven soldados y peones. Cada regimiento está identificado por un número, y por el cuartel general del que dependan. Los peones son los encargados de conseguir oro y madera para el cuartel, elementos que son necesarios para el entrenamiento de los soldados. Cada peón tiene un número que lo identifica, y la ocupación actual: minero o leñador. Los peones pueden cambiar de ocupación de acuerdo a las necesidades del momento, y viven en el cuartel al que pertenezcan. Pueden ser expulsados o incorporados del cuartel. Los peones cargan siempre 10 Kg. de mercadería (naturalmente, los mineros cargan oro y los leñadores cargan madera), y pueden estar libres u ocupados en su tarea. Los soldados son quienes pelean las batallas, y viven dentro de los regimientos. Cada soldado está identificado por un número de serie y un rango. Algunos soldados tienen mecha corta y al enfurecerse suelen saquear al primer peón que vean. Si el peón es leñador no le hacen nada, pero si es minero y tienen la bolsa de mineral llena les piden la mitad del oro que tengan en ese momento, para gastarlo en porrones de Duff (un antiguo brebaje con alto contenido alcohólico que sirve para aplacar los ánimos). Un regimiento puede expulsar a cualquier soldado de sus filas. También puede solicitar al cuartel general la incorporación de nuevos soldados. Si el reclutamiento fue positivo, los nuevos soldados se incorporan al regimiento de forma inmediata. El cuartel general es el único organismo capaz de formar o disolver regimientos dentro de su jurisdicción. El cuartel general también almacena todos los materiales que traigan los peones. Cuando el cuartel recibe a un peón, éste deja su carga y pasa a estar desocupado. El cuartel general puede producir soldados y peones. Para producir un peón es necesario contar con 20 Kg. de oro. Para producir un soldado, son necesarios 30 Kg. de oro y 20 Kg. de madera. Si no se cuenta con estas cantidades, no podrá reclutar nuevos soldados ni incorporar nuevos peones.



a) Realice un diagrama UML en donde se puedan observar las relaciones entre las clases que modelan este sistema. Complete Cada diagrama de clase con su nombre, atributos, operaciones y responsabilidades.

b) Realice los siguientes diagrama UML de interacción: Un soldado saquea a un peón (que puede ser minero o leñador), el cuartel general crea un regimiento e intenta incorporar 3 soldados.

Farnsworth

El científico Farnsworth ha construido una serie de máquinas capaces de crear diferentes átomos y moléculas. Sin embargo, como la puesta en marcha de tanta maquinaria es bastante costosa, ha decidido construir un programa que simule su comportamiento, con el objetivo en mente de ahorrar tiempo y dinero. En este sistema los átomos se representan por medio de un nombre y un número real indicando su masa atómica asociada. El Atomizador es la increíble máquina capaz de crear átomos de forma instantánea. Por el momento el prototipo es bastante simple: permite la creación de átomos de hidrógeno (masa 1,007), oxígeno (masa 15,99) y carbono (masa 12,01). El Atomizador posee un interruptor de tres posiciones que indica qué tipo de átomo se está produciendo actualmente. Las moléculas son agrupaciones de dos o más átomos. Cada molécula tiene un nombre específico y posee un número de masa molecular que se obtiene sumando el número de masa atómica de los átomos que la componen. El Moleculizador es la máquina encargada de unir conjuntos de átomos para formar moléculas. Dadas las limitaciones iniciales, sólo puede producir un determinado tipo de moléculas, reseñadas en la siguiente tabla:



Molécula	Composición
Agua	2 átomos de hidrógeno y 1 átomo de oxígeno
Agua Oxigenada	2 átomos de hidrógeno y 2 átomos de oxígeno
Dióxido de Carbono	1 átomo de carbono y 2 átomos de oxígeno
Oxígeno	2 átomos de oxígeno

Finalmente, el Materializador es el aparato que engloba las dos máquinas anteriores. En su interior, el Atomizador puede cambiarse con facilidad por un prototipo mejor, más eficiente, pero el Moleculizador es siempre el mismo. Es más debido a cómo el profesor Farnsworth contruyó el Materializador, para desarmar el Materializador hay que desarmar el Moleculizador, no es el caso del Atomizador. El Materializador devuelve moléculas de los tipos indicados en la tabla de arriba. Es el aparato con el que trabaja en forma directa el científico, pues los demás están encerrados en el interior del Materializador. Además de suministrar moléculas, el Materializador es capaz de realizar las siguientes tareas complejas:

Oxigenación de agua: recibe muchas moléculas de agua y le agrega oxígeno a cada una de ellas, devolviendo agua oxigenada.

Electrólisis del agua: recibe muchas moléculas de agua y dos repositorios vacíos en los cuales retorna los átomos de hidrógeno y de oxígeno por separado (todos provenientes de las moléculas de agua).

Respiración artificial: recibe muchas moléculas de oxígeno y devuelve una igual cantidad de moléculas de dióxido de carbono (creando todos los átomos de carbono que fueran necesarios).

a) Realice un diagrama UML en donde se puedan observar las relaciones entre las clases que modelan este sistema. Complete cada diagrama de clases con su nombre, atributos, operaciones y responsabilidades.

b) Realice un diagrama UML de interacción en donde se vea la siguiente secuencia de interacciones: Crear tres moléculas de oxígeno y luego realizar una respiración artificial. Crear tres moléculas de agua y realizar una oxigenación del agua.

Grand Theft Auto: Vice City

Bienvenidos a Vice City. Bienvenidos a los '80. Luego de un largo tiempo en una cárcel de máxima seguridad, Tommy Vercetti está de nuevo en las calles de Vice City. Durante un trabajo para su jefe, Sonny Forelli, Tommy cae en una emboscada y pierde todo su dinero y mercancía. En total, una suma de 23.021.980 dólares, monto que Sonny quiere de regreso y obligará a Tommy a conseguirlo, sea como sea. Tommy Vercetti se encuentra solo y sin dinero en una gran ciudad, con mafiosos, gansters y policías corruptos



que quieren su cabeza, su única opción es hacer lo que él mejor hace, adueñarse de la ciudad. Tommy comenzará a cobrar dinero, lo que que aumentará su tesoro personal. Tommy puede utilizar este dinero para realizar diferentes operaciones.

Desde el punto de vista de los negocios, en la ciudad de Vice City (V.C.) dos elementos son importantes: los autos y los inmuebles. De ambos hay para todos los gustos. En V.C. hay muchos autos y Tommy puede adquirir cualquiera. Para lograr esto utiliza a su viejo amigo Carl Johnson. Carl es un dealer, un proveedor de autos. Cuando Tommy quiere un auto se lo solicita a él, ya sea un auto cualquiera o con una capacidad específica. Tommy no es exigente y la marca del auto y color pueden ser cualquiera. Ante el pedido, Carl recorre los autos de la ciudad buscando uno que se adapte a la especificación de Tommy. Si el pedido fue por un auto cualquiera, sin indicar ninguna característica especial, en cuanto Carl encuentra uno, le consulta a Tommy si puede pagar el costo de dicho auto. Si el pedido fue con requerimientos, Carl le cobra a Tommy \$200 más. Tommy verifica simplemente si tiene el dinero suficiente para pagarlo, de ser así le responde afirmativamente, Carl cobra ese dinero y le entrega el auto. Si Tommy no cuenta con el dinero suficiente, Carl continúa buscando por todos los autos. Antes de solicitar un auto, Tommy debe asegurarse de contar con lugar disponible en alguno de sus inmuebles. Cuando el pedido es por un auto con requerimientos, en caso de que Carl no encuentre uno que se ajuste a lo especificado, armará uno especial para Tommy. Todos los autos que Carl arma son negros, tienen patente BAT-MAN-X, donde X es un número siempre diferente, y tiene capacidad para cuatro personas. Por esto Carl no le cobra a Tommy. Solo Carl puede recorrer los autos de V.C. en busca del indicado y la única forma en que Tommy puede obtener un auto es a través de Carl.

En V.C. la oferta inmobiliaria es muy grande. La ciudad mantiene un registro de todos sus inmuebles, ya sea que se hallen en venta o no. Para comprar un inmueble, Tommy debe hallar uno que se encuentre en venta y su precio no supere el dinero en efectivo con el que Tommy cuenta.. Al comprar un inmueble el capital de Tommy aumenta el equivalente a un 95% del valor del inmueble. Esto es, si el inmueble vale \$100 y Tommy lo adquiere, pierde \$100 de su dinero en efectivo, pero en el fondo ganará \$95 en el cálculo del total de su capital, pues posee ahora ese inmueble. Algunos inmuebles permiten guardar autos pero tienen una capacidad máxima. No todos los inmuebles tienen la misma capacidad por lo que antes de guardar un auto en alguno de ellos, Tommy debe asegurarse que haya lugar disponible. Cuando un auto es guardado, se considera que el auto es también parte del capital de Tommy. El valor capital de un auto es el 80% de su precio.

Gran parte de los robos de autos en V.C. son denunciados a la policía. Si Tommy posee un auto en uno de sus garajes que fue denunciado por robo, el valor capital del mismo resulta ser sólo del 10% de su valor original. Por suerte, Tommy logró conseguir un handy policial de uno de los autos de Vice City y puede saber si un auto ha sido denunciado o no. Para esto sólo debe indicar por handy el número de patente y se le informará si el auto ha sido denunciado como robado o no. Tommy sólo puede tener acceso a sus autos a través de los inmuebles en donde se encuentran.

a) Modele el juego Vice City con las características antes mencionadas. Considere implementar una clase Ciudad que de alguna forma mantenga todos los inmuebles de la ciudad (en venta o no), todos los vehículos que circulan y lo que usted considere necesario de acuerdo al enunciado. Debe existir una operación (ubicada en la clase correcta) que permita conocer cuál es el capital total de Tommy Vercetti al momento; esto es, la suma del efectivo que tenga más la del capital. Recuerde que el valor capital de un auto puede variar de acuerdo al estado del mismo por lo que al momento de realizar el cálculo se deben considerar todas las posibles situaciones. Asuma que cuenta con la clase Handy que contiene una operación estaRobado(Patente:String) : booleano que retorna verdadero si el auto cuya patente es Patente ha sido declarado como robado, se retorna falso en caso contrario.

b) Realice un diagrama UML de interacción para la operación CalcularCapital. Realice un diagrama UML de interacción en donde se vea la siguiente secuencia de interacciones: Tommy le pide un auto a Carl con una especificación, Carl no encuentra un auto disponible para Tommy por lo que le arma uno nuevo.

Imperio Galáctico

Hay inquietud en el Senado Galáctico. Varios miles de sistemas solares han declarado su intención de abandonar la República. Este movimiento separatista, liderado por el misterioso Conde Dooku, ha provocado que al limitado número de Caballeros Jedi les resulte difícil mantener la paz y el orden en la galaxia. Debido a esta situación de alta tensión se han vuelto populares las estaciones de defensa planetaria. Estas estaciones funcionan como estaciones espaciales cargadas de cañones de plasmas y escudos antireflex. El cañón de plasma es un dispositivo de defensa sofisticado, dispara una bola de plasma altamente explosiva. A través de los años se han desarrollado diferentes versiones de esta arma. Estas versiones varían en su peso, consumo eléctrico y capacidad de disparo. La capacidad de disparo de los cañones se mide en parsecs, y todos los cañones poseen un control para regular la potencia del cañón. Dicho control varía entre 1% y 100% y se lo denomina KGB. Los cañones necesitan para disparar cargas de plasma; todos tienen espacio para 10 cargas, solo se pueden cargar de a una a la vez y hasta que no estén las 10 no es posible realizar un disparo (cada disparo consume las 10 cargas). La primera versión del cañón de plasma fue la MIL-KHOU-SE, consume 100W de energía y pesa 450kg; es capaz de generar una explosión de 1500xKGB parsecs. Una versión nueva es el MIL-W7. El cañón MIL-W7 pesa 100kg menos que el MIL-KHOUSE-SE pero consume 35W más. La explosión que generar es bastante menor que este último, un 50% menor pero es más barato de producir. En contra partida, también se creó el MIL-WVista (otra versión del MIL-KHOU-SE, no del MIL-W7) muchísimo más lento que el MIL-KHOUSE-SE. MIL-WVista pesa media tonelada más y agrega un consumo de 1000W. Cada disparo genera una explosión de 4000xKGB parsecs. Los cañones no siempre se usan al 100% de KGB debido a que esto reduciría su vida útil. Con cada disparo se liberan residuos llamados poños. Una carga de plasma tiene un peso que varía entre 25g y 50g, además posee un consumo de 15W energía. Este consumo se debe a que el contenedor de plasma se debe mantener en un estado de equilibrio químico sino la carga se vuelve altamente explosiva. El año pasado, y debido a la necesidad de tecnologías más amigables con el planeta, se comenzó la producción del MIL-ALL-GREEN, una versión mejorada del MIL-W7 que consume un 75% menos que el MIL-W7 y mantiene el mismo peso y la potencia de disparo.

Cada cañón (cualquiera de los mencionados) se ubica sobre una plataforma de disparo móvil. La plataforma almacena hasta 20 cargas de plasma que se utilizan para cargar su correspondiente cañón, recordando que solo se puede cargar una carga de plasma a la vez. Si la plataforma se queda sin cargas entonces le tiene que solicitar a la estación de defensa que le entregue 20 cargas más y finalizado esto vuelve a intentar cargar el cañón. La plataforma de disparo móvil sirve como vínculo entre la estación espacial y el cañón, por



cuestiones de seguridad nadie puede operar el cañón directamente, solo a través de la plataforma. Si la plataforma intentase realizar un disparo con su cañón de plasma sin contar con las suficientes cargas de plasma en él, entonces se generaría una falla. Ante esta falla, lo que debe hacer la plataforma es cargar una nueva carga de plasma en el cañón y volver a disparar nuevamente. Si la plataforma no tiene más cargas disponibles entonces le solicita a la estación de defensa que le haga una entrega de 20 cargas.

Una aclaración importante, el peso real de un cañón es el peso del mismo más el peso de las cargas de plasma que contiene; algo similar pasa con el consumo eléctrico, el consumo de un cañón es su consumo más el consumo de cada una de sus cargas. En este último caso el consumo total se debe duplicar debido a cuestiones de seguridad, para evitar problemas de tensión se aumenta el consumo eléctrico. Como estamos en tiempos de guerra, el presupuesto militar es ilimitado por lo que cada estación de defensa puede crear un número ilimitado de cargas de plasmas. Existe un protocolo de seguridad conocido como VIENEN_LOS_MALOS que ejecuta un disparo masivo de todos los cañones, a través de sus plataformas.

a) Para este ejercicio, incluya las consultas Gasto y Peso donde crea conveniente. Gasto debe retornar el gasto total de energía del anillo de seguridad de la estación espacial, y peso cuánto pesa dicho anillo.

b) Para el siguiente enunciado realice su correspondiente diagrama UML, para cada clase incluya sus atributos y operaciones, recuerde incluir los modificadores de visibilidad y otros. Incluya la operación VIENEN_LOS_MALOS dentro de la clase que crea conveniente.

Counter Terrorist Unit

La empresa Electronics Arts E.A. está trabajando en un nuevo juego de estrategia llamado CTU, Counter Terrorist Unit. Como su nombre lo indica el juego se basa en la amenaza terrorista al mundo y en la lucha de los agentes especiales para atraparlos. El juego todavía no ha sido terminado, pero parte de su especificación ya se ha dado a conocer al público. Un terrorista está caracterizado por un nombre real y una serie de alias o nombres falsos. Los agentes especiales se caracterizan por tener un nombre y un valor numérico entero que representa su rango. El juego agrega a cada terrorista y agente una credencial, esta es única e inamovible a cada personaje y servirá para luego determinar si el agente puede o no eliminar al terrorista. Una credencial está caracterizada por un nombre y una secuencia de enteros que representa un código numérico. Las credenciales se asignan cuando cada terrorista o agente es creado.

Un rol muy importante dentro del juego lo tienen las ciudades. Una ciudad está representada por un nombre, su cantidad de habitantes y los terroristas que se encuentran en la misma. A su vez, cada ciudad esta comunicada con dos ciudades más a las que considera ciudades vecinas. Dentro de las funciones que cumple un agente esta la de limpiar una ciudad. Cuando se le indica a un agente que limpie una ciudad determinada entonces debe buscar los terroristas que en ella se encuentren y eliminar aquellos que tengan una credencial cuya secuencia numérica sea igual a la del agente. Una ciudad puede recibir un agente, ante esta situación la ciudad le solicita al agente que la limpie y luego le solicita que limpie las dos ciudades vecinas. Un terrorista puede atacar una ciudad cualquiera. El ataque de un terrorista a una ciudad tiene como resultado una disminución del 10% en la población de la ciudad y un aumento del 4% en las poblaciones de las ciudades vecinas (la gente huye).

A un terrorista se le puede ingresar un nuevo alias, si este no se encuentra en su lista, se agrega al final. A una ciudad se le puede indicar que dos alias están representando a la misma persona. Si dentro del registro de terroristas de la ciudad, los dos alias pertenecen al mismo terrorista, no se deben tomar acciones, si no existen terroristas con esos alias tampoco. Si solo existe un terrorista que tiene uno de los alias, el otro alias se agrega a su listado de nombres falsos. Cuando cada alias representa un terrorista diferente la situación es más compleja: Se deben eliminar de la ciudad los dos terroristas y luego crear uno nuevo en la ciudad, su nombre real será la concatenación de los nombres de los dos terroristas y su lista de alias será también la concatenación de las dos listas de alias de los terroristas. La credencial que se le asigne a este nuevo terrorista debe ser un duplicado de cualquiera de las dos credenciales de los terroristas originales.

Dentro del trabajo de investigación que se podrá realizar en el juego está la posibilidad de realizar consultas. A un agente se le puede consultar, dada una ciudad, que diga qué terroristas podría eliminar considerando no solo los terroristas en la ciudad sino también en las dos ciudades vecinas. Un agregado de último momento al juego fue el de mantener para cada terrorista su nivel de peligrosidad. La forma de calcular este valor es muy simple, el nivel de peligrosidad de un terrorista es igual a la cantidad de alias que tiene. Por cuestiones de implementación del juego, este valor no puede ser almacenado en cada terrorista sino que debe ser calculado

ante cada consulta. Además se agregó una nueva consulta. A una ciudad se le puede solicitar todos los terroristas que tengan un nivel de peligrosidad igual o mayor a uno dado.

a) Realice el diagrama de relaciones UML correspondiente para el modelo del juego mencionado. Ubique en cada clase su nombre, atributos, operaciones, responsabilidades y relaciones.

b) Para concatenar los nombres de los terroristas, suponga que cuenta con una función predefinida: concatenar(cad1: string; cad2: string) : string; que retorna la concatenación de cad1 con cad2.

Road Fighter

Road Fighter es un videojuego de carreras producido por Konami y lanzado en los arcades en 1984. Fue el primer juego de carreras desarrollado por esta compañía. El objetivo era alcanzar la meta sin chocar los coches que aparecen ni quedarse sin gasolina. La particularidad de este juego de carreras, es que no se corría en una pista sino en una ruta transitada. Actualmente existe una tendencia a reestrenar antiguos éxitos de la era de los juegos 8-bits en las consolas de última generación. En este contexto, la corporación WAYNE ha decidido reeditar el juego con algunas pequeñas modificaciones. El espíritu del juego será el mismo que el anterior, pero se incorporarán nuevos elementos. En el juego se encontrarán vehículos (los que circulan por la ruta y los que corren la carrera), power-ups (los que darán algún tipo de premio al jugador) y obstáculos. Todos los obstáculos tienen un peso y un puntaje. Este último puede ser un número positivo o negativo. Los obstáculos van a afectar al auto del jugador, pero esto dependerá del tipo de obstáculos.



En el juego se utilizarán tres tipos de obstáculos distintos: la anciana, el perro y el bache. La anciana pesa 50 Kg. y tiene valor -10. Cuando el auto del jugador golpea a la anciana, ésta no le hace nada al auto. El perro pesa 20 Kg. y tiene puntaje -50. Cuando el auto del jugador golpea al perro, éste le reduce la velocidad del auto a la mitad. Finalmente está el bache, el bache no pesa nada y tiene valor -5. Cuando el auto del jugador golpea el bache, éste le quita al auto todos sus power-ups. Los power-ups son "poderes" o "extras" que afectan el juego cuando el jugador encuentra uno. Todos los power-ups tienen un puntaje y en total el juego tendrá 3 power-ups: Misil, Nitro y Nafta. El misil tendrá puntaje 50, el nitro 20 y la nafta 10. El auto del jugador juntará un power-up cuando lo choque en la ruta o cuando choque un vehículo que contiene dentro del mismo un power-up. Cuando el auto del jugador se encuentra con el nitro, duplicará su velocidad. La velocidad del auto del jugador no puede ser cambiada por el jugador sino sólo por las cosas que ocurran durante el juego. Si se encuentra con un misil, se guardará dentro del vehículo del jugador y si se encontrara con la nafta entonces se sumará 10 litros al tanque de combustible.

La parte más complicada del juego serán los vehículos. Todos los vehículos del juego tendrán una velocidad y un peso. El único vehículo que podrá cambiar de velocidad será el auto de carrera del jugador. Todos los vehículos tendrán también una patente, que será una cadena de caracteres. Los vehículos del juego se dividirán en dos grupos, aquellos que forman parte de la carrera y los que sólo transitan por la ruta. Los vehículos de carrera tendrán una posición, que será un número entre 1 y 12. El vehículo de carrera del jugador es un caso especial porque, a diferencia de los otros vehículos de carrera, éste almacenará los misiles que se encuentra. Todos los vehículos que sólo transitan por la ruta pueden llevar en su interior un power-up. Este power-up se libera cuando el vehículo del jugador choca a un vehículo de los que sólo transitan por la ruta. Estos vehículos también tienen una masa, la masa es el resultado que se obtiene al multiplicar el peso del vehículo por el puntaje del power-up que contienen, si es que contienen uno. De estos vehículos se tendrán tres tipos, las motos, los autos familiares y los camiones. Las motos viajarán a 80 km/h y pesarán 70 kg. Los autos viajarán a 160km/h y pesarán 150 kg. Finalmente los camiones viajarán a 90 km/h y pesarán 2 toneladas. A diferencia de los dos primeros casos, los camiones pueden llevar en su interior hasta 2

power-ups, el cálculo de la masa del camión será igual a su peso multiplicado por la suma de los puntajes de sus power-ups. El auto del jugador comenzará el juego con una velocidad de 90 km/h y 100 litros de combustible. Finalmente el juego tendrá muchas rutas para que el jugador seleccione. Cada ruta tendrá un nombre, los obstáculos que en ella se encontrarán y los vehículos de tránsito que estarán en la misma. Cada ruta tendrá asociada una complejidad, la complejidad de una ruta se obtiene al sumar la cantidad de vehículos de tránsito dentro de ella y la cantidad de obstáculos dentro de ella, todo esto luego se multiplica por la mayor masa presente en la ruta. El nombre de la ruta se forma al concatenar todas las patentes de los vehículos que transitan por ella. El juego tendrá también los 12 vehículos de carreras que competirán en una ruta. Un agregado de último momento es el de un nuevo tipo de obstáculos, el lobo. El lobo será un tipo especial de perro, el cuál duplicará el peso de éste último pero mantendrá igual el puntaje. Cuando el auto del jugador golpee al lobo, éste hace lo mismo que haría un perro y además le dejará el puntaje del jugador en 0.

a) Realice un diagrama UML de clases que modele los elementos descritos en este enunciado relacionados con el juego "Road Fighter". El atributo de masa de los vehículos que sólo transitan por la ruta debe ser calculado. Los atributos de peso y puntaje de los obstáculos también serán atributos calculados.

b) Realice el diagrama de secuencia para obtener la complejidad de una ruta.

