

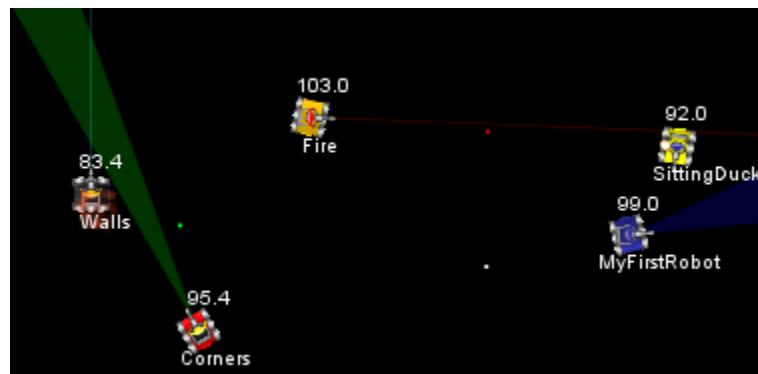
Práctica nº 1 - Programación Orientada a Objetos

Objetivo: Identificar los objetos (con sus respectivos atributos y comportamiento) que forman parte de la solución de un problema y la relación entre ellos.

Realice esta práctica en papel, **no** esperamos que use alguna herramienta digital ni que invierta tiempo en sintaxis UML, al menos para **los ejercicios 1, 2 y 3**.

1. **Escuadrón de tanques.** Se quiere modelar una nueva variante del juego **Robocode**. En este juego tanques de combate pertenecientes a distintos jugadores son puestos a prueba en un campo de batalla limitado a un área de 600 x 800 px). La nueva modalidad permitirá que **nuestro tanque** se enfrente a un **escuadrón enemigo de tanques** que circula esporádicamente sobre el campo de batalla. El escuadrón enemigo siempre se mueve en conjunto en la misma dirección (la dirección la determinará un ángulo, a modo de ejemplo: 0° corresponde al Norte, 90° al Oeste, 180° al Sur, 270° al Este), los tanques del escuadrón nunca se separan ni detienen su marcha, y cuando encuentran al adversario empiezan a disparar balas de distintos calibres. Nuestro tanque debe tratar de sobrevivir esquivando el escuadrón y a la vez tratar de dispararles, de modo de poder eliminar alguno y, con un poco de suerte, resultar ganador en el juego.

Los tanques son todos del mismo tipo, todos poseen una torreta que permite disparar las balas y sobre esta torreta se encuentra un radar con un ángulo de visualización que permite detectar al enemigo. La torreta puede sobrecalentarse si se producen disparos continuos, esto provoca que no pueda continuar con sus disparos hasta que la torreta se enfríe, después de algunos segundos.



Juego original sin los escuadrones

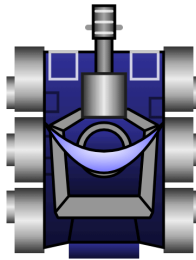


Gráfico ilustrativo de un tanque. Note la torreta y el radar sobre la misma.

Los tanques también mantienen información de la energía que poseen durante la batalla, la cual empieza en un valor de 100 y va mermando con los ataques que recibe hasta llegar a 0, en ese momento el tanque queda fuera de combate. El tanque también puede quedar fuera de combate *temporalmente*, es decir “deshabilitado”, esto ocurre como penalización para el jugador cuando realiza muchos disparos de manera continua o si no realiza movimientos.

Realice el diagrama de clases de acuerdo a lo especificado.

2. **Librería de Gráficos 2D.** Se quiere desarrollar una librería de gráficos básica que permita crear diferentes gráficos 2D (círculos, rectángulos y cuadrados). Estos gráficos 2D deberían tener un color asociado, y además el sistema debe permitir también rotarlos X grados y moverlos a otra ubicación.
 - a. Realice el diagrama de clases de acuerdo a lo especificado.
 - b. Si se quisiera agregar la funcionalidad “anidamiento de figuras” ¿deberían realizarse cambios en su modelo de clases? (ej: que un cuadrado pueda incluir un círculo, de modo que cuando se dibuje el cuadrado también se dibuje el círculo)
3. **Getting there from here.** Los juegos del tipo “getting there from here” permiten que un agente se mueva desde un origen hacia un destino.

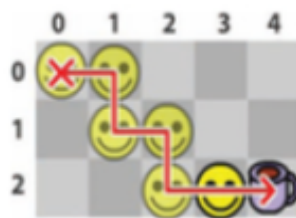


Figura a modo de ejemplo. Un agente quiere alcanzar su destino, una taza de café.

Puntualmente el juego que se quiere modelar consiste en una región o tablero representado por una grilla de NxM celdas, donde cada celda puede representar tierra, montaña o agua. El agente sólo puede caminar sobre la tierra -por lo pronto no puede escalar una montaña ni nadar- por tanto, estos representan obstáculos en su camino.

El juego debe permitir posicionar al agente, indicando su posición inicial y su avance hasta alcanzar su posición final, indicar qué celdas son tierra, montaña o agua para inicializar el tablero, e indicar la posición de inicio del juego.

Una cuestión final y muy importante: una opción del juego permite saber cual es el camino con menor “costo” que el agente podría recorrer. El valor de costo puede variar según el juego, en este caso vamos a considerar como costo que atraviese la menor cantidad de celdas posibles para alcanzar su destino. Si hubiera varios caminos con el mismo costo, es indistinto cuál de todos esos caminos elija.

Realice el diagrama de clases de acuerdo a lo especificado.

4. **Sistema de Administración de Aerolíneas (SAA).** El SAA permitirá registrar información de aeropuertos, aerolíneas y vuelos de las mismas. La información con la que se cuenta es la siguiente:

“Cada aeropuerto se identifica por un código de tres letras (por ejemplo BUE identifica al aeropuerto Ministro Pistarini de Buenos Aires o LIM que identifica al aeropuerto Jorge Chávez de Lima) y tiene datos de su georeferenciación (el posicionamiento geográfico se describe con una latitud y una longitud). Por otro lado, cada aerolínea tiene asociado un conjunto de vuelos que inicialmente puede ser cero. Acerca de los vuelos que pertenecen a una aerolínea, siempre están identificados por un código de vuelo formado por el código de la aerolínea y un número (LAN 1256 o TA 24). Cada vuelo tiene asociado un aeropuerto origen y un aeropuerto destino, fecha/hora de partida y fecha/hora de llegada, puerta de embarque, un estado de la partida (confirmado, demorado, en tiempo), etc. Cada vuelo también tendrá asignado un avión, el cual tiene 1 o más secciones. Por ejemplo: primera clase, ejecutiva y económica, siempre hay sólo UNA sección de cada tipo. Cada avión tiene un conjunto de asientos organizados como N filas y M columnas. Las filas están representadas por números, mientras que las columnas se representan por letras (A, B, etc. dependiendo de la cantidad de asientos que tenga una fila).

El SAA permite realizar diversas operaciones: crear aeropuertos, aerolíneas y vuelos, así como administrar la relación entre ellos. También permite encontrar vuelos entre dos aeropuertos y realizar la reserva de asientos o incluso, cuando un pasajero no quiere pagar el costo extra de la reserva de asiento (algunos vuelos tienen este costo

extra), el SAA sabe elegir qué asiento asignar, teniendo en cuenta el balanceo de la carga en el avión).

El SAA además permite emitir un reporte general y completo acerca de todos los aeropuertos, aerolíneas, vuelos, secciones de vuelos y asientos reservados, así como también reportes específicos, como encontrar todos los asientos disponibles para una sección de un vuelo en particular o cuantos asientos libres quedan en total en un vuelo.”

Realice el diagrama de clases de acuerdo a lo especificado.

5. Analice y verifique si su modelo se adapta a las siguientes consideraciones, y de ser necesario modifique su diagrama:
 - a. Así como existe “primera clase” o “clase turista”, se ha creado una nueva clase: “Tránsito” donde se ubicaran a los pasajeros que no vuelan hacia su destino final, sino que usan el vuelo como un punto de conexión.
 - b. El sistema debe permitir realizar el cambio de asiento reservado
 - c. El sistema debe permitir registrar información de la tripulación de vuelo (piloto, co-piloto, comandante de abordaje y auxiliar de vuelo) asociado a un vuelo en particular.
 - d. El sistema debe poder recuperar todos los vuelos de un piloto en particular y el historial de aerolíneas en las que trabajó.
 - e. El sistema debe permitir dejar registro de denuncias realizadas por los pasajeros, quienes pueden indicar en qué vuelo ocurrió el incidente, sus comentarios al respecto e incluso podrían subir una foto si la hubiere.
 - f. El sistema debe permitir hacer una estadística de los países más visitados.
 - g. El sistema debe determinar cuántos pasajeros de un determinado vuelo ya pasaron por el check-in.