Ejercicios
Java
Programación II

Ejercicios

33) Diseñe un programa que permita administrar una liga de football (ahora en Java :p):

- **Registro de equipos participantes:** Para cada equipo debe almacenar la siguiente información: Nombre, cantidad de jugadores (mínimo 15), capitán del equipo.

- **Para cada equipo:** Debe brindar la opción de buscar y editar equipos.

- **Generar calendario de partidos:** Tomando como base los equipos ingresados, debe generar el calendario de los partidos de la liga (una vuelta). Por ejemplo, si los equipos ingresados son: Barcelona, Bayern Munich, Inter y el Municipal, el calendario podría quedar de la siguiente forma:

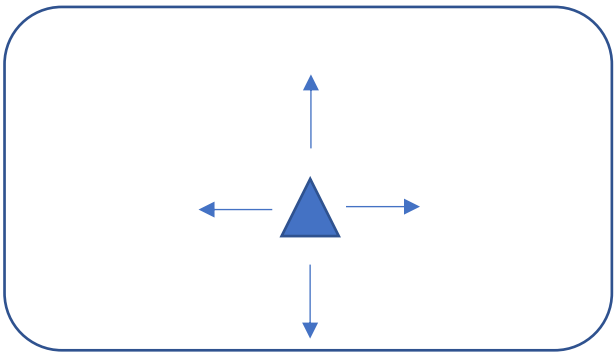
  Barcelona - Bayern
  Inter - Municipal
  Barcelona - Inter
  Bayern Municipal
  Barcelona - Municipal
  Inter - Bayern

  Su sistema debe tener la capacidad de generar el calendario de partidos para una cantidad de equipos par e impar.

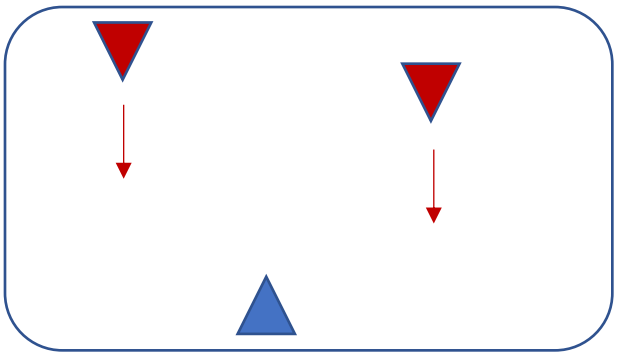
- **Ingreso de resultados:** Debe permitir el ingreso del marcador para cada uno de los partidos generados.

- **Tabla de resultados:** En base a los resultados ingresados, muestre en pantalla la tabla de posiciones que muestre el resumen de la liga.

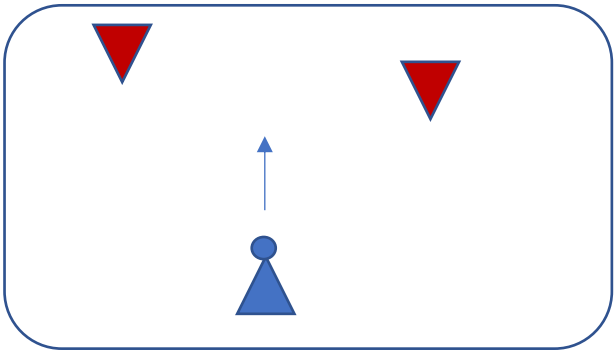
34) Implemente un juego de Naves en Unity3d donde el objetivo es detener una fuerza invasora. La mecánica de juego consiste en:
    a. El juego es visto desde arriba.
    b. El jugador puede mover su nave espacial hacia adelante, hacia los lados y hacia atrás. (Puede moverse libremente por la pantalla)



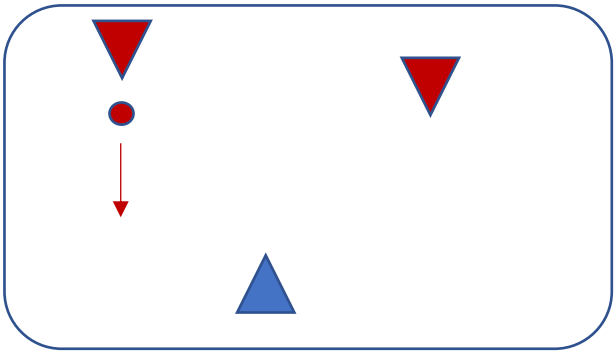
    c. Los enemigos aparecen desde la parte superior de la pantalla y se mueven en linea recta hacia abajo. (Puntos Extra si diseñan trayectorias aleatorias)



d. La nave del jugador puede disparar balas cuando se presiona la tecla Space.



e. Los enemigos pueden disparar balas también, las cuales siguen una trayectoria recta.



f. Si un enemigo es impactado por una bala del jugador, dicho evento cuenta como 10 ptos. y el enemigo desaparece de pantalla.
g. Si el jugador es impactado por una bala de los enemigos o por un enemigo, el jugador pierde el juego. (Solo hay una vida)

Las gráficas no son un factor importante para este reto, por lo que puede usar las figuras básicas de Unity3D (circulos, cubos, esferas, etc.)

35) Implemente un sistema de POS básico en Java. Para ello se requieren las siguientes funcionalidades:

a. Productos: Cada producto debe contener la siguiente información: Nombre, Precio y Cantidad en existencia. Cada objeto debe implementar los métodos que permiten establecer el valor de un atributo o consultar el valor del mismo. No se permite el acceso directo a un atributo.

b. Inventario: Debe permitir agregar nuevos productos o modificar la cantidad de productos ya existentes.

c. Venta: El sistema debe permitir el registro de ventas, donde el usuario selecciona los productos que se estan vendiendo y la cantidad vendida de cada uno de ellos. Al finalizar una venta se debe mostrar en pantalla el detalle de la factura: Un listado por producto, la cantidad ingresada, precio unitario, precio total por linea y el total de la venta. Ejemplo:

Producto	Cantidad	P. Unitario	Total
Computadoras	2	Q23.0	Q46.00
Cargadores	1	Q10.0	Q10.00

Gran Total: Q56.00

Cada venta debe quedar registrada y almacenada en el sistema. Para ello se propone la creacion de un objeto que almacene las operaciones.

No se puede ejecutar una venta de un producto que no tenga existencias.

d. Resumen de Ventas: Muestra en pantalla todas las ventas realizadas y el gran total de cada una de ellas, para el caso anterior, únicamente se mostraría:

Venta1 Total de: Q56.00
….
Y asi sucesivamente para el resto de ventas.

e. Menu: El menu principal del sistema debe mostrar las siguientes opciones:
   - Inventario
   - Venta de Productos
   - Resumen de Ventas