**PRACTICA 4: SUPER JUEGO DE CARRERAS “MEDAUTO”**



El juego consiste en una carrera de vehículos, en la que cada jugador elige un vehículo y una carta objetivo. Los vehículos pueden ser coche, moto o camión. Son siempre tres jugadores.

El programa comienza con la creación de la pista de carreras, que contiene un coche, una moto y un camión en las posiciones [0,0], [1,0], [2,0] respectivamente. A continuación, se pide al jugador que elija un vehículo y una carta objetivo. La carta objetivo indica el orden de llegada de dos vehículos. Las cartas objetivo son:

-Camión primero y Moto segundo

-Camión segundo y coche primero

-Camión tercero y moto primero

En cada turno, cada jugador saca una carta de un mazo de cartas. Las cartas pueden mover un vehículo hacia delante o hacia atrás, incluyendo el mismo, por lo que el jugador puede elegir mover el vehículo que desee, incluyendo el suyo propio.

La carrera termina cuando el primer vehículo llega a la meta. Después se visualiza la pista de carreras y el ganador es el jugador que haya conseguido resolver su carta objetivo.

El camión avanza y retrocede siempre 2, el coche avanza y retrocede 3 y la moto avanza y retrocede 4, siempre en el eje x (es decir se modifica las columnas). Siempre permanece en la misma pista de carreras (fila).

CÓDIGO OBLIGATORIO:

-Sobre la clase Vehículo, parte de su código es:

*public abstract class Vehiculo {*

*String nombre;*

*int posx;*

*int posy;*

*public abstract void avanzar();*

*public abstract void retroceder();*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| POS | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| VEHICULO | CAMION |  |  |  |  |  |  |  |  |  |
| MOTO |  |  |  |  |  |  |  |  |  |
| COCHE |  |  |  |  |  |  |  |  |  |

EJEMPLO CONSOLA:

*run:*

*Carta objetivo*

*El objetivo del camion es: Camion primero y Moto segundo*

*El objetivo de la moto es: Camion segundo y coche primero*

*El objetivo del coche es: Camion tercero y moto primero*

*CAMION \_ \_ \_ \_ \_ \_ \_ \_ \_*

*MOTO \_ \_ \_ \_ \_ \_ \_ \_ \_*

*COCHE \_ \_ \_ \_ \_ \_ \_ \_ \_*

*Camion coge carta*

*Quien quieres que avance: 1.camion 2.moto 3.coche*

*2*

*Moto coge carta*

*Quien quieres que avance: 1.camion 2.moto 3.coche*

*2*

*Coche coge carta*

*Quien quieres que avance: 1.camion 2.moto 3.coche*

*1*

*\_ \_ CAMION \_ \_ \_ \_ \_ \_ \_*

*\_ \_ \_ \_ \_ \_ \_ \_ MOTO \_*

*COCHE \_ \_ \_ \_ \_ \_ \_ \_ \_*

*Camion coge carta*

*Quien quieres que retroceda: 1.camion 2.moto 3.coche*

*1*

*Moto coge carta*

*Quien quieres que avance: 1.camion 2.moto 3.coche*

*3*

*Coche coge carta*

*Quien quieres que avance: 1.camion 2.moto 3.coche*

*3*

*CAMION \_ \_ \_ \_ \_ \_ \_ \_ \_*

*\_ \_ \_ \_ \_ \_ \_ \_ MOTO \_*

*\_ \_ \_ \_ \_ \_ COCHE \_ \_ \_*

*Camion coge carta*

*Quien quieres que avance: 1.camion 2.moto 3.coche*

*2*

*Moto coge carta*

*Quien quieres que avance: 1.camion 2.moto 3.coche*

*1*

*Coche coge carta*

*Quien quieres que retroceda: 1.camion 2.moto 3.coche*

*3*

*\_ \_ CAMION \_ \_ \_ \_ \_ \_ \_*

*\_ \_ \_ \_ \_ \_ \_ \_ \_ MOTO*

*\_ \_ \_ COCHE \_ \_ \_ \_ \_ \_*

*BUILD SUCCESSFUL*

EXTRA: CONSEGUIR QUE LA CARTA OBJETIVO ESTÉ ASOCIADA AL OBJETO VEHÍCULO CONCRETO Y AL FINALIZAR EL JUEGO, PUEDA COMPROBAR SEGÚN EL RESULTADO DE LA PISTA DE CARRERAS, QUIÉN HA GANADO Y SUPERADO LA CARTA OBJETIVO DE FORMA AUTOMÁTICA.

AYUDA (OPCIONAL):

-Para las cartas objetivos simplemente realizar un array de String de tamaño 3.

-Para las cartas del juego, crea una clase Carta con nombre y estado ( 0 avanza y 1 retrocede). Y en el main un array de tamaño 10 de objetos Carta, que un bucle haga 5 de avanzar (pares) y otras 5 cartas de retroceder (impares).

Por ejemplo:

if (cartas[random].getEstado()%2==0){

camion.avanzar();

}

-Para la finalización de cada turno utiliza dos métodos static del main que se llamen actualizar y mostrar. Actualizar, pone los vehículos en sus nuevas posiciones en el tablero (pista de carreras) y el resto lo pones a null. El mostrar, solo pinta donde hay vehículos y donde no lo hay que muestra ‘\_’.

-Cada turno jugarán los tres vehículos, y decidirán en función de la carta que le haya tocado (random), si retroceder o avanzar de sí mismo u otro vehículo en función de su interés.

-Comprobar si ha llegado el primero a la meta, simplemente es ver que no hay un nulo en la posición [0][9] o [1][9] o [2][9].