

PROYECTO 2: AutosLocos EndGame

Versión 0.1

El objetivo de este proyecto es desarrollar una versión extrema del proyecto anterior de los carros locos.

Al igual que el proyecto anterior la carrera puede tener de 2 a 20 carros, pero cada carro corre por separado. Durante la carrera el carro se puede encontrar con puntos de control u obstáculos que entorpecen su camino. La idea es que el carro llegue al final del camino en el menor tiempo posible.

Descripción general:

El juego consiste en definir una serie de vehículos que competirán en una carrera EXTREMA simulada, el objetivo es determinar qué vehículo es el ganador, mediante una apuesta. Cada vehículo estará representado por la imagen de un vehículo de los autos locos o un personalizado por el usuario.

Características del juego:

- El juego es usando librerías AWT y SWING para crear pantallas, la cual incluye etiquetas, botones u otro elemento que sea necesario.
- El juego comienza definiendo la cantidad de vehículos que van a correr que serán seleccionados de la lista de autos locos y/o de carros que el usuario ingreso en la aplicación de gestión de vehículos, se debe mostrar todas las características del vehículo, para que el usuario pueda tener todos los detalles.
- La carrera solo es realizada por un carro a la vez.
- El orden de la carrera es el mismo orden en que fueron seleccionados los carros.
- Los vehículos tienen velocidad fija.
- Los vehículos tienen un factor de resistencia igual que el proyecto anterior.

Descripción del trayecto de la carrera.

- La carrera presenta un conjunto de paradas, en cada parada el vehículo tiene que decidir de manera aleatoria por uno de dos caminos.
- Existen paradas que pueden ser trampas, estas pueden ser:
 - Piso mojado representado por agua.
 - Explosivo representado por una bomba.
 - Obstáculos que obstruyen la carretera, como árboles o piedras.

Cuando el vehículo cae en una trampa el punto de control se elimina, pero eso implica que la carrera debe REORDENARSE para que toda la pista se mantenga siempre JUSTA, el vehículo debe regresar al punto de control anterior en el mismo camino por donde avanzo.

- Cada avance entre un punto de control y otro tiene una distancia que representa el recorrido hasta ese punto, a medida que el vehículo avance se va sumando la distancia que va recorriendo.
- Se debe llevar el tiempo que le toma al vehículo completar la carrera.
- No es necesario que el vehículo simule la velocidad, todos avanzan a la misma velocidad, en función de esto, el tiempo debe calcularse por la velocidad que tiene configurado cada vehículo.
- Cada parada tiene un identificador que único que permite tener una relación de orden en la pista.

Generación de la pista.

- La pista debe ser generada y se configura utilizando estructura de datos más adecuada.
- La pista debe graficarse en pantalla, cada parada puede representarse con un círculo y el camino a recorrer por una línea.
- La pista tiene niveles, al inicio cada uno tendrá más paradas que el anterior, la cantidad está determinado por 2^n , donde n es el nivel.
- La cantidad de niveles mínima es 8 PERO PUEDEN EXISTIR MAS.
- La cantidad de trampas que existen en toda la pista, la determina el usuario, mínimo son 5 y el máximo
- Las trampas están distribuidas lo más equitativo posible en toda la pista, esto significa que si el vehículo comienza la carrera y decide irse por el camino derecho las trampas no pueden estar todas de ese lado.
- Al inicio de la carrera el vehículo ya tiene que decidir aleatoriamente por qué camino desea ir.
- Las distancias de los caminos entre una parada y otra es en kilómetros debe estar en el siguiente rango $[1 - 10]$
- Cada parada tiene un identificador que es mayor que la parada siguiente del lado izquierdo y mayor que la parada siguiente del lado derecho. El identificador comienza con el valor 10.000, las paradas se van generando aleatoriamente con un numero entre 1 y 10.000-1, ya que el identificador 10.000 lo tiene el inicio. NO PUEDE HABER DOS PARADAS CON EL MISMO IDENTIFICADOR.

Características de los vehículos

Cada vehículo cuenta con los siguientes atributos: Velocidad y resistencia.

Una vez que se indica la cantidad de carriles, se colocan las características de cada vehículo, imagen que lo representa.

En un archivo están las configuraciones de los vehículos, estas son:

- **Velocidad:** perezoso, crucero, el super Ferrari, Delorean.
- **Resistencia:** es un valor que determina la capacidad de resistir a obstáculos que se pueden encontrar en el camino, los cuales puede ser: bombas, obstáculos, líquido resbaladizo, cada atributo tiene un valor que determina el nivel de resistencia para cada tipo de obstáculo, pero eso depende de las características del carro como tipos de cauchos y tamaño.
- **Tipos de cauchos:**
 - Normales
 - Anti-coleo (permiten rodar sobre líquidos que pueden ocasionar pérdida de la dirección sino cuenta con esta característica)
 - Todo terreno (tiene mayor resistencia a los obstáculos, esta va en una escala de 1 a 5, mientras mayor el número podrá evitar mejor los obstáculos) pero no tienen propiedad de anti-coleo.
- **Tamaño de los cauchos:**
 - Pegado al piso (son los carros más rápidos, su velocidad puede ser El súper Ferrari, Delorean),
 - Normales (su velocidad puede ser: perezoso, crucero),
 - MonsterTruck (son los más altos, pero también son muy lentos, su velocidad es perezoso)
- En cada carril puede aparecer MÁXIMO un obstáculo solamente de cada tipo.
- De acuerdo a las características anteriores, se pueden configurar muchos tipos de vehículos, existen 11 vehículos denominados “Autos locos”, que se describen a continuación:

- El Rocomóvil (The Bouldermobile) conducido por los Hermanos Macana, Pedro y Roco (The Slag Brothers, Rock and Gravel)2.
- El Espantomóvil (The Creepy Coupe) pilotado por Los Tenebrosos (The Gruesome Twosome).
- El Auto/Súper Convertible (Convert-a-car) al volante el profesor Locovich (Pat Pending).
- El Stuka Rakuda (The Crimson Haybailer) conducido por el Barón Hans Fritz (The Red Max).
- El Compact Pussycat conducido por Penélope Glamour (Penelope Pitstop).
- El Súper Chatarra Special (The Army Surplus Special) conducido por el Sargento Blast y el Soldado Meekly.
- El Antiguoóvil (The Bulletproof Bomb) conducido por los Hermanos Anticuados (The Ant Hill Mob).
- El Alambique Veloz (The Roaring Plenty) conducido por Lucas el Loco (Lazy Luke) y Osvaldo el Oso (Blubber Bear).
- El Superheterodino (The Turbo Terrific) conducido por Peter Perfecto (Peter Perfect).
- El Troncoswagen (The Arkansas Chugabug) conducido por Tío Tomás (Rufus Ruffcut) y Oso Miedoso (Sawtooth).
- Súper Ferrari (The Mean Machine) conducido por Pierre Nodoyuna (Dick Dastardly) y Patán (Muttley).

En la siguiente tabla se muestra el tipo de vehículo que representa sus características.

TIPO	NOMBRE
5	Rocomóvil (Los hermanos macana)
6	Espantomóvil (Los tenebrosos)
2	Auto/Súper Convertible (El profesor Locovitch)
6	Stuka Rakuda (El barón Hans Fritz)

8	Compact Pussycat (Penélope Glamour)
7	Súper Ferrari (Pierre Nodoyuna y Patán)
4	Antiguomóvil (Mafio y sus siete pandilleros)
2	Alambique Veloz (Lucas el granjero y el Oso miedoso)
9	Superheterodino (pedro Bello)
3	Troncoswagen (Brutus y listus)
1	El súper Chatarra Special (El sargento Blast y el soldado Meekly)

En esta versión, cada vehículo tiene asignado los atributos por defecto.

Debe considerar una lista de vehículos. Cada vehículo tiene los siguientes atributos:

- Nombre en español
- Nombre el inglés
- Nombre del o los conductores
- Tipo de caucho
- Tamaño de caucho
- Velocidad
- Resistencia a obstaculos
- Resistencia a liquido
- Resistencia a explosivos

Cada campo debe ser validado, no pueden dejarse en blanco o colocar valores incorrectos.

Es importante destacar que, en esta versión, los vehículos que son creados por el usuario, los valores de resistencia a las distintas trampas son colocados por él, por lo tanto, son independientes del tipo y tamaño de caucho, y los valores mmínimos y máximos son los siguientes:

- La velocidad puede variar entre 80km/h y 200 km/h
- Explosivo: 5% a 35%
- obstáculos: 5% a 40%
- Líquido: 10% a 35%

Factor de resistencia

El factor de resistencia debe calcularse y representa un valor en milisegundos, es una reducción de la velocidad del vehículo a cero por cierto tiempo. Este factor depende del tipo de obstáculo y velocidad del vehículo, se calcula multiplicando el obstáculo por velocidad por 100, el resultado está dado en milisegundos, por ejemplo:

Carro tipo crucero, con tamaño de cauchos normales, tipo de caucho normales, que choca con una bomba se frenará por un factor de resistencia calculado de la siguiente manera:

$$FR \text{ (milisegundos)} = 20\% * 120 * 100 = 1000 \text{ milisegundos (1 seg).}$$

El valor 100 es una constante que se debe colocar siempre.

Tabla de valores según las características del vehículo y valores de los obstáculos

TIPO	Velocidad	Tipo Caucho	Tamaño Caucho	Bomba	Obstac.	Líquido	Velocidad km/h
1	Perezoso	Todo Terreno	Monster Truck 1	15%	20%	30%	140
2	Perezoso	Todo Terreno	Monster Truck 2	12%	17%	25%	120
3	Perezoso	Todo Terreno	Monster Truck 3	10%	15%	20%	110
4	Perezoso	Todo Terreno	Monster Truck 4	7%	10%	15%	100
5	Perezoso	Todo Terreno	Monster Truck 5	5%	5%	10%	80
6	Perezoso	Normales	Normales	15%	20%	30%	120

7	Crucero	Normales	Normales	20%	25%	35%	140
8	El super Ferrari	Anti-Coleo	Pegado al piso	25%	30%	20%	160
9	Delorean	Anti-coleo	Pegado al piso	35%	40%	15%	180

NOTA: Los valores deben estar en un archivo de texto plano para que pueda modificarse cambiándolos en el archivo.

Consideraciones generales de la carrera.

- Al finalizar la carrera de cada vehículo seleccionado, se debe mostrar la tabla de posiciones indicando el tiempo y distancia que tardo, las posiciones se colocan en orden ascendente según el tiempo, por lo que el primer lugar es el vehículo con menos tardo.
- En un archivo de texto se guardará la tabla de posiciones.
- La aplicación debe mostrar una ventana principal con las siguientes opciones:
 - Generar pista.
 - Seleccionar vehículos.
 - Mostrar histórico de resultados de carreras anteriores indicando su fecha, posiciones, tiempo y distancia.
 - Salir.

Aplicación de gestión de vehículos.

Se debe crear una aplicación por separado que permite:

- **Agregar:** permite agregar un vehículo nuevo con la información de los atributos. Al registrar la nueva información se debe indicar al usuario con un mensaje indicando que fue realizado satisfactoriamente el registro y mostrar la información completa del mismo.
- **Modificar:** permite modificar la información de un vehículo que es previamente seleccionado de una lista. EL flujo debe ser el siguiente:

- Se muestra la lista de vehículos con un número de opción que lo identifique.
- El usuario selecciona la opción del vehículo que desea modificar.
- Se muestra toda la información del vehículo acompañado con un número de opción que identifica el atributo.
- El usuario selecciona la opción del atributo que desea modificar.
- Modificar el valor del atributo.
- Se realiza la modificación.
- Se muestra al usuario un mensaje si la modificación se hizo exitosamente o si no se pudo hacer indicando por qué.
- Si fue exitosa la modificación se muestra toda la información del vehículo.
- **Eliminar:** permite eliminar la información de un vehículo que es previamente seleccionado mediante un número de opción asociado a cada vehículo.
- Una vez eliminado el vehículo, se muestra el mensaje que se eliminó exitosamente y se muestra al usuario el vehículo eliminado junto con la información de los atributos.
- **Lista de vehículos:** muestra todos los vehículos con sus atributos incluyendo los que están configurados por defecto: los autos locos.
- **Consultar vehículo:** permite consultar **TODA** la información de un vehículo. Se muestra un listado de vehículos, el usuario selecciona el vehículo que desea consultar y la aplicación muestra toda la información.
- **Regresar:** regresa siempre a la pantalla anterior.
- **Salir:** sale del sistema.

Cada vez que se agrega, modifica o elimina, debe realizar la acción en la lista de vehículos y en el archivo de texto que mantiene la información, por lo tanto, debe haber guardado automático de los datos.

AL agregar o modificar un vehículo debe terminar la acción mostrando los datos de vehículo y el mensaje correspondiente de que la acción fue realizada exitosamente.

Al eliminar un vehículo debe mostrar el mensaje respectivo y los datos del vehículo eliminado.

Requerimientos:

Aplicación de simulación de carreras:

1. Vista grafica de la pista, incluye los vehículos, las paradas con su id, las trampas, las distancias.
2. Ajuste automático de la pista para que este equilibrada.
3. Ventana principal.
4. Generación de la pista.
5. Selección de Vehículos, se debe mostrar todas las características del vehículo, para que el usuario pueda tener todos los detalles.
6. Representación de los vehículos con imágenes.
7. Mostrar histórico de todas las carreras.
8. Reacción a los obstáculos.

Aplicación de gestión de vehículos:

1. Agregar
2. Modificar
3. Eliminar
4. Consultar
5. Listar

Toda la información que se necesite mostrar de forma detallada, por ejemplo, todos los vehículos y todas las características debe mostrar en formato de tabla.

Grupos de proyecto:

- El grupo de proyecto debe realizarlo hasta 3 personas, deben estar presentes para defenderlo y corregirlo individualmente.

Características técnicas.

- El programa debe implementarse en JAVA usando interfaces gráficas.
- Debe implementar las clases necesarias para representar cada objeto que maneje el proyecto.
- ES OBLIGATORIO EL USO DE ARBOLES PARA LA CARRERA Y LISTAS PARA LA GESTIÓN DE VEHICULOS.
- Todas las lecturas de los archivos deben almacenarse en listas dinámicas primero antes de utilizarse, cualquier nuevo elemento, modificación o eliminación debe realizarse sobre la lista y después guardar automáticamente para los archivos.
- Como buena práctica de desarrollo, no se debe repetir código.
- Cada función debe hacer una única tarea.
- El proyecto debe estar documentado usando JAVADOC

Entrega: fecha de entrega: 17/07/2023 al 21/07/2023. Porcentaje: 20%

- Documentos a entregar:
 - Código fuente del proyecto.
- **Documentos a entregar:**
 - Código fuente del proyecto.
 - Librerías propias necesarias para ejecutar el software.
 - Los archivos deben estar en una carpeta con el nombre: AlgIIProyecto2GrupoX.
 - Debe ser enviado empaquetado en formato zip con el mismo nombre por correo.
 - El asunto del correo debe ser: AlgIIProyecto2GrupoX.
 - En el cuerpo del correo deben colocar los integrantes con su cédula.

Recomendaciones generales:

- Definir un jefe de proyecto en el grupo que supervise las actividades (por supuesto, este integrante debe desarrollar también).
- Definir las actividades que va a realizar cada integrante, analicen primero, definan tareas a realizar, definan cuáles tareas son prioritarias, definir estructura de datos, clases.
- Pensar en desarrollos que sean lo más modular posible, no repetir código: principio DRY (don't repeat yourself) y KISS ("Keep it Simple, Stupid")