

# Proyecto: Juego de Carrito con Obstáculos Dinámicos usando Árbol AVL

Se desarrollará un **videojuego 2D** en Python con interfaz gráfica, en el cual un carrito avanza automáticamente sobre una carretera lineal con una distancia total de **N km**. El escenario contará con **obstáculos dinámicos** que se gestionan y organizan mediante un **árbol AVL**, utilizando como criterio de inserción las **coordenadas (x, y)** de cada obstáculo.

El sistema debe permitir la carga inicial de configuraciones y obstáculos desde un archivo **JSON**, así como la inserción de obstáculos manualmente antes de iniciar el juego. El árbol de obstáculos deberá ser graficado (pueden usar los nombres, íconos, etc) para representar cada nodo. A este árbol deben implementarle la funcionalidad de mostrar los recorridos en anchura y en profundidad vistos en clase. Al realizar las inserciones se debe considerar el comportamiento natural del AVL y los siguientes puntos:

- Comparar primero el valor de **x** del primer punto.
- En caso de empate en **x**, se compara el valor de **y** del mismo punto.
- No se permiten coordenadas repetidas (dos obstáculos no pueden estar en el mismo punto).
- Cada obstáculo tendrá un tipo, cada tipo puede quitar energía del carrito.

La energía del carrito se inicia en 100%, y cada tipo quitará unos puntos configurados en la interfaz gráfica (cada equipo debe proponer los tipos).

## Mecánica del Juego

### 1. El carrito:

- Se mueve de manera automática en el eje **x** hacia adelante, avanzando **5 metros cada 200 ms** (configurable desde el json para cambiarlo).
- El jugador puede controlar al carrito únicamente en el eje **y** con las flechas **↑** y **↓**.
- El carrito puede **saltar** con la tecla **ESPACIO**.
  - La distancia del salto es configurable.
  - Durante el salto, el carrito debe cambiar de **color** y volver a su color original al aterrizar.

### 2. La carretera:

- Es lineal, con una distancia total de **X metros** (configurables en el JSON y en interfaz gráfica).
- El avance del carrito y la aparición de obstáculos depende de la posición **x**.
- Los obstáculos pueden estar en diferentes posiciones **y** (arriba, medio, abajo), lo cual obliga al jugador a moverse o saltar.

### 3. Obstáculos:

- Cada obstáculo tiene una posición definida por **coordenadas (x1, y1) y (x2, y2)**:
  - **x**: representa la distancia sobre la carretera.
  - **y**: representa la altura o carril del obstáculo.

Cada obstáculo tendrá un área rectangular donde podrá chocar al carrito.

- A medida que el carrito avanza, se consultan en el AVL los obstáculos que se encuentran dentro del área visible de la pantalla para dibujarlos en tiempo real. Ustedes deben plantear una estrategia para avanzar automáticamente la pantalla, teniendo en cuenta que el carrito siempre estará en el borde izquierdo de la pantalla a la misma velocidad de avance de la pantalla, es decir, el carrito siempre deberá ser visible para el jugador.

## Requisitos del Proyecto

### 1. Árbol AVL de obstáculos:

- Implementar inserción, eliminación y recorridos.
- Mantener balanceado automáticamente el árbol.
- Permitir consultar rápidamente los obstáculos dentro de un rango de coordenadas ( $x_{min}$ ,  $x_{max}$ ,  $y_{min}$ ,  $y_{max}$ ), correspondiente al área visible de la pantalla.

### 2. Carga desde JSON:

- Al inicio, el juego cargará un archivo JSON con:
  - Configuraciones del juego (distancia total, velocidad de avance, altura del salto, tiempo de refresco, color inicial del carrito, etc).
  - Lista inicial de obstáculos predefinidos con sus coordenadas (x,y).

### 3. Interfaz Gráfica (GUI):

- Mostrar en pantalla el carrito, la carretera y los obstáculos que aparecen según la posición.
- Refrescar la pantalla cada **200 ms** (configurable).
- Permitir que en cualquier momento el jugador visualice gráficamente la estructura del **árbol AVL de obstáculos**.

### 4. Dinámica de juego:

- El carrito pierde energía si choca con un obstáculo que no logra esquivar o saltar.
- El juego termina cuando se alcanza la meta (distancia en metros de la carretera) o cuando el carrito se queda sin energía.

### 5. Trabajo en equipo:

- Proyecto desarrollado en equipos de **2 personas**.
- Código entregado en un repositorio **Git** con historial de commits.
- Manual técnico y de usuario para el juego (en inglés).
- Videotutorial explicativo del juego en ejecución, se debe ver la cara de cada miembro del equipo en el video cuando esté hablando (cada miembro del equipo debe participar).

Calificación:

1. Carga y representación del árbol: 0,5
2. Recorridos gráficos del árbol: 1,0
3. Administración del árbol antes de iniciar el juego: 1,0
4. Juego del carrito usando el árbol AVL: 2
5. Documentación y video: 0,5



**Tejiendo  
Universidad**

Autoevaluación Institucional 2018 - 2026

### Consideraciones importantes:

- Sin la entrega de la documentación y el video no se podrá entregar la funcionalidad ni presentar la sustentación.
- El profesor estará en la libertad de exigir sustentación individual a cada miembro del equipo o podrá elegir a una de las dos personas para que represente al equipo, siendo la nota resultante asignada a cada persona.
- La nota de funcionalidad no podrá ser mayor en una unidad a la de la sustentación.
- Explicar el juego en el video en inglés tendrá 3 décimas adicionales a la nota final.
- Fecha de entrega (envío de documentación, link del video – Drive, Youtube, etc –, repositorio público de github sin modificaciones posteriores): lunes 28 de septiembre a más tardar 6:00 pm. Cada 10 minutos de demora en la entrega tendrá penalización de 1 décima. La dirección de correo electrónico destino es: jeferson.arango@ucaldas.edu.co
- La hora y fecha de sustentación se acordará mediante un archivo que será compartido antes de la entrega. Después de la fecha de entrega del punto anterior y hasta finalizada la sustentación no se podrá realizar modificaciones sobre el proyecto y todos sus componentes.

Ejemplo de la estructura del JSON, aunque ustedes pueden usar la que deseen:

```
{
  "config": {
    "distancia_total": 2000,
    "velocidad": 10,
    "refresco_ms": 300,
    "salto_altura": 50,
    "color_carrito": "azul"
  },
  "obstaculos": [
    {"x": 150, "y": 0, "tipo": "roca"},
    {"x": 320, "y": 1, "tipo": "cono"},
    {"x": 500, "y": 2, "tipo": "hueco"},
    {"x": 1200, "y": 1, "tipo": "aceite"}
  ]
}
```