

TAREA INTEGRADORA 2: Sistema de Gestión y Monitoreo de Movilidad y Seguridad (SGMMS)

Contexto:

La ciudad de Palmira necesita capacitar a sus operadores para gestionar el **Sistema de Gestión y Monitoreo de Movilidad y Seguridad (SGMMS)**. Para ello, se desarrollará un simulador interactivo donde los jugadores asumirán el rol de operadores municipales, resolviendo incidentes en tiempo real, optimizando rutas y garantizando la seguridad ciudadana.

Condiciones del Simulador:

1. Vehículos Autónomos (automatas):

- Los vehículos de emergencia (patrullas, ambulancias, bomberos) se generarán aleatoriamente en el mapa y se moverán automáticamente usando hilos. Cada tipo de vehículo tendrá una velocidad y prioridad distinta (ejemplo: ambulancias son más rápidas).

2. Incidentes Dinámicos:

- Aparecerán incidentes en el mapa a los cuales se les debe prestar atención:
 - Accidentes: Estos no deben ser aleatorios, sino que deben tener una razón de ser con el comportamiento entre los autómatas definidos. (Usted debe pensar en al menos 3 criterios)
 - Robos e incendios: Deben ser aleatorios

El jugador debe asignar entonces vehículos disponibles para resolverlos. Cada incidente se almacena en un árbol binario de búsqueda ordenado por gravedad (prioridad alta a baja).

3. Gestión de Rutas:

- Se debe definir un mapa que defina las rutas a seguir a través de la ciudad.

4. Movilidad del Operador:

- El jugador navegará en un mapa 2D con desplazamiento en 4 direcciones (teclas WASD). El mapa incluirá zonas residenciales, comerciales y vías principales, delimitadas por bordes gráficos.

5. Escenas Conectadas:

- Tres escenas: Centro de Monitoreo, Mapa de Tráfico y Panel de Incidentes. El jugador alternará entre ellos mediante interfaces gráficas (botones o teclas).

6. Indicadores en Tiempo Real:

- La interfaz mostrará:
 - Número de incidentes activos (por cada tipo debe haber un indicador gráfico distinto).
 - Puntaje actual

7. Puntuación:

- El jugador ganará puntos al resolver incidentes. La puntuación se calculará según la eficiencia (tiempo y recursos usados).
-

Restricciones Técnicas:

1. **JavaFX Obligatorio:** Diseño gráfico con Canvas, imágenes para vehículos/incidentes, y paneles interactivos.
 2. **Hilos:** Movimiento autónomo de vehículos y actualización de indicadores en tiempo real.
 3. **Árboles Binarios de Búsqueda:**
 - Gestión de incidentes (orden por prioridad).
 - Almacenamiento de rutas activas (orden por tiempo).
 4. **Prohibido:**
 - **NO** se permite el uso de librerías como OpenGL, o cualquier Game Engine, este tipo de librerías o herramientas, abstraen el uso de hilos o elementos gráficos que deben ser evaluados en este curso.
 5. **Anotaciones:**
 - Cualquier lógica o estrategia de programación que parezca estar por fuera de los lineamientos del curso debe poder ser sustentada de forma completa y correcta.
-

Entregas:

- **Implementación (70%):** Código funcional en JavaFX con los requisitos anteriores.
 - Commits: Evaluación y evidencia del aporte individual de cada estudiante en el proyecto a través de el sistema de control de versiones (Git)
- **Diseño (15%):** Actualización del diagrama de clases
 - Aplicación de los patrones de diseño vistos en clase y vistos en el curso de Ingeniería de software 2.
- **Documentación (15%):**
 - Manual de usuario: Incluirá teclas de control, explicación de indicadores y estrategias para gestionar incidentes. Se mostrará en una ventana emergente (JavaFX).
- **Sustentación (multiplicador):** Demostración en semana 18 explicando el funcionamiento de todo el proyecto.

Entrega: Semana 18
