

Simulador Collision Warning with Emergency Brake (CWEB)

Johnny Zaet Agüero Sandí

Junio 2020

Instituto Tecnológico de Costa Rica

Ingeniería en computadores

Taller de programación

Tabla de contenido

Capítulo 1. Introducción	3
Capítulo 2. Descripción del problema	3
Capítulo 3. Análisis de resultados	4
Capítulo 4. Dificultades	7
Capítulo 5. Bitácora	7
Capítulo 6. Informe de horas.....	8
Capítulo 7. Conclusión personal	8

Capítulo 1. Introducción

En el siguiente proyecto, se desarrollará un simulador para Volvo Trucks, que se encarga del diseño y fabricación de los vehículos industriales de la marca. Dentro de los sistemas de seguridad que equipan los camiones, destaca el Collision Warning with Emergency Brake (CWEB), el cual consiste en un sistema que detecta la proximidad de los vehículos circundantes y le genera una alerta al conductor. Si el conductor no toma ninguna acción que evite el impacto, el vehículo automáticamente genera una frenada de emergencia.

El simulador cuenta con sus propias instrucciones de uso, una vez iniciado el simulador deberá de reiniciarlo para volverlo a usar. Contando con la función de leer los datos aun así hayan pasado días de su último uso.

Capítulo 2. Descripción del problema

El simulador consiste en una calle de dos carriles, en la cual se generan vehículos aleatoriamente en ambos carriles.

Al inicio del simulador se mostrará una pantalla de presentación (splash animado) del desarrollador por cinco segundos o diez segundos (tiempo configurable).

Utilizando conocimientos adquiridos vía investigación. Se desarrollará el sistema para que se configure el sistema con frenado de emergencia o cambio de carril mientras así las condiciones lo permita. La única condición que permitirá el cambio de carril es cuando no exista un vehículo en el carril contrario.

Se adjuntará un parámetro de proximidad, el cual es configurable y es el indicador para que el carro a velocidad constante, logre hacer cambio de carril a tiempo o pueda usar el freno de emergencia. El parámetro de proximidad debe ser configurable por el usuario sin tener que modificar el código al inicio de la ejecución. El simulador generará vehículos aleatoriamente en algún carril cada vez que se de una condición. Si se genera un vehículo en el mismo carril en el que se encuentra el camión, se debe aplicar la condición previamente configurada por el usuario.

Capítulo 3. Análisis de resultados

Se obtiene un simulador a pura interfaz gráfica, donde se inicia se obtiene la ventana principal llamada “menú”(imagen 1) la cual se encarga de mostrarle al usuario las instrucciones de uso así como en esta misma ventana se encuentran los botones necesarios para usar el simulador.

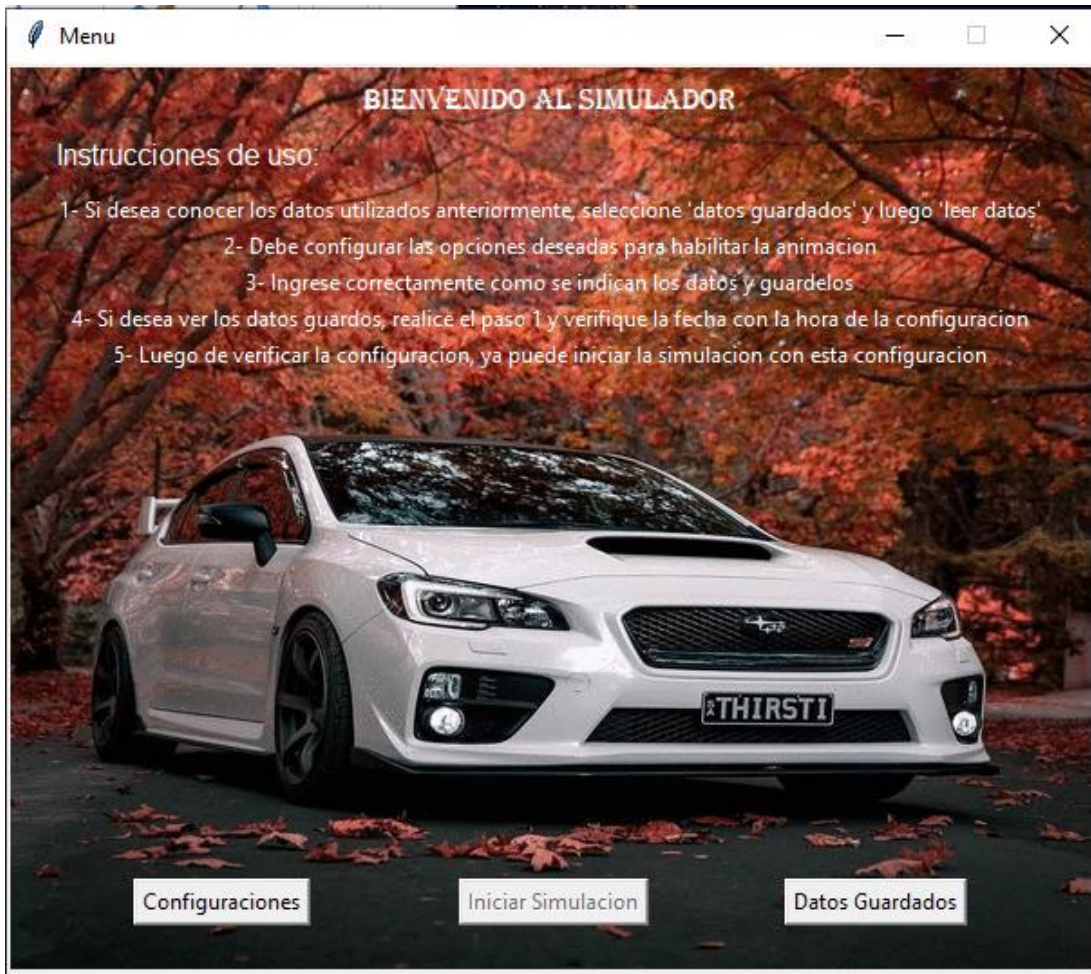


Imagen 1. Ventana Menú

Como se puede apreciar en las instrucciones y en la ventana del menú, el botón del simulador está desactivado hasta que el usuario configure correctamente, en la ventana de configuración (imagen 2), se le brindarán opciones al cliente las cuales pueden ser modificadas, si y solo si están correctas. En la ventana se indican las opciones y existe una función que valida dichas opciones. Al igual que se presenta una ventana para leer los datos (imagen 3),

utilizados en el último uso del simulador. Dicha información es posible mostrarla inclusive cerrando el programa, porque se guarda en un .txt.

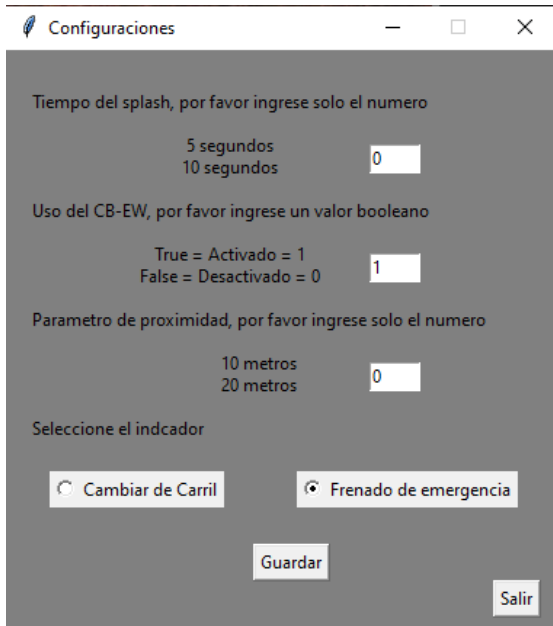


Imagen 2. Ventana de configuraciones

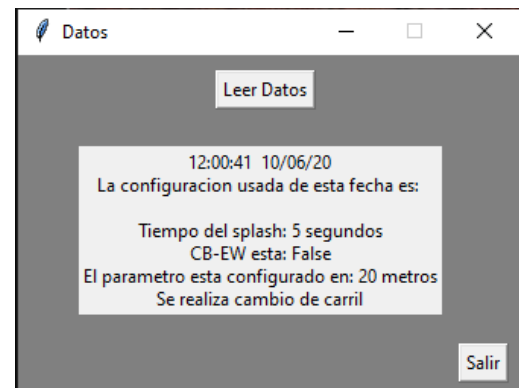


Imagen 3. Ventana de datos

Una vez configurado y revisados los datos, se puede iniciar el simulador. Este simulador cuenta con un splash animado (imagen 4) el cual da inicio al simulador, luego de este splash automáticamente se presenta el simulador con las configuraciones anteriormente dadas. Dentro de la simulación se cuenta con: el manejo normal (imagen 5) y cuando no se puede realizar un cambio por las circunstancias (imagen 6). Esto, para cambio de carril.

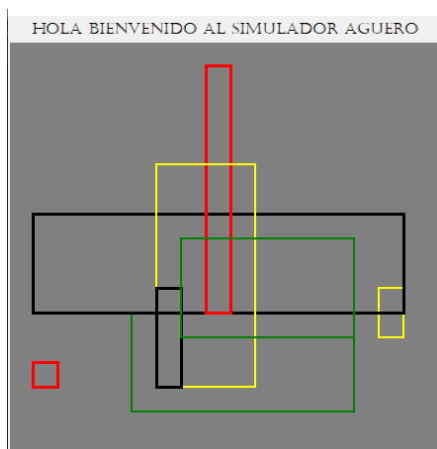


Imagen 4. Splash animado

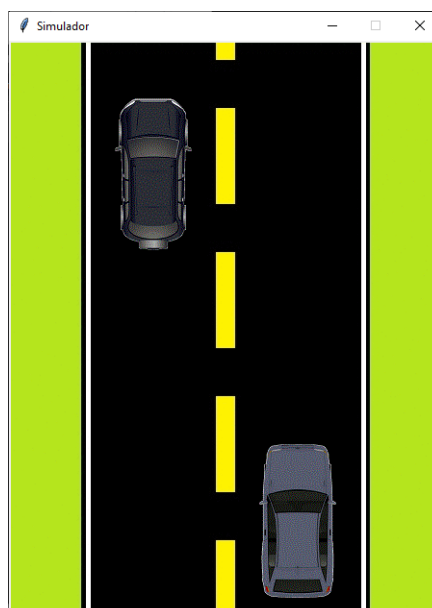


Imagen 5. Manejo normal

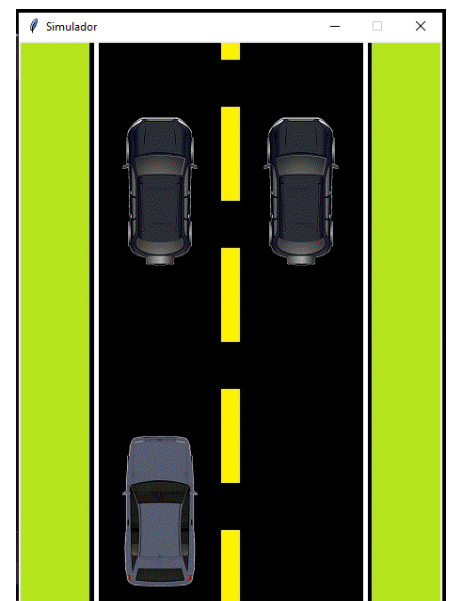


Imagen 6. No posible cambio

Para las otras 2 opciones. Solo se presenta Manejo normal y frenado de emergencia (imagen 7), cuando se desactiva el sensor de proximidad, para ambas situaciones se presentará una colisión (imagen (8)).

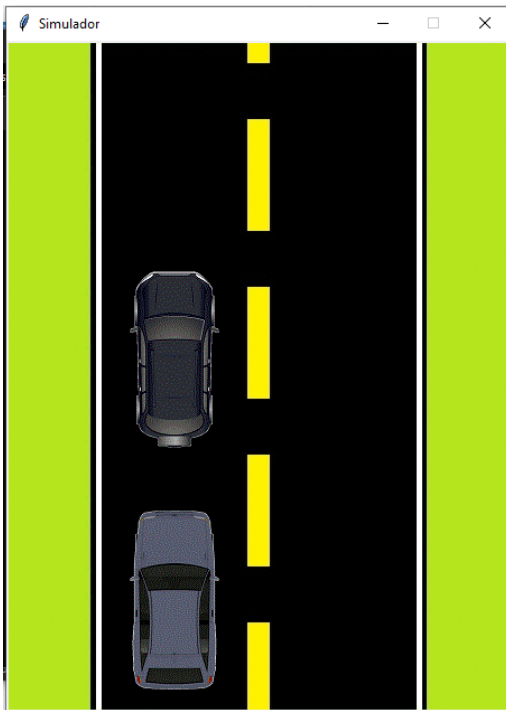


Imagen 7. Freno de emergencia

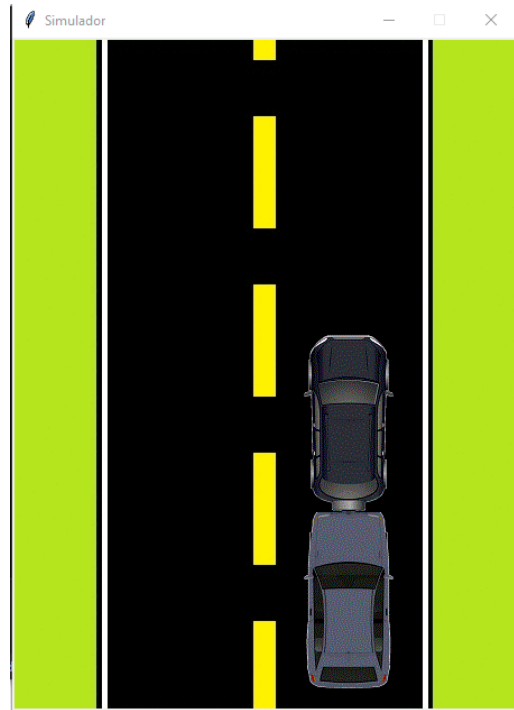


Imagen 8. Colisión

NOTA: Cabe destacar que para todas las opciones, es necesario contar con audio y disponerlo para el simulador. Ya que se usó la función para reproducir sonidos según lo que ocurra.

Capítulo 4. Dificultades

Las dificultades más grandes encontradas en el proyecto, fueron:

- 1- La forma de poder detectar cuando venían carros en ambos carriles. Si bien es cierto, se logró el objetivo, el hecho de intentar desarrollar proyectos con librerías que no se prestan para, es complicado. Teniendo en cuenta que existen otras para desarrollar estas.
 - a. Esta problemática se resolvió implementando una función para el caso donde el random generara carros en ambos carriles. De esta forma el simulador ya tomaba en cuenta evaluar ambos carros en la condicional de cambio de carril.
- 2- La parte de estética y lograr acoplar las imágenes de carros en las posiciones correctas con sus dimensiones, fue todo un reto.
 - a. Se implementaron algunos conocimientos básicos en edición de fotografías y algunas apps para redimensionar las imágenes.
- 3- El tiempo estimado de desarrollo del proyecto es una limitante muy grande, ya que al parecer no se toma en cuenta el hecho de la modalidad que se está impartiendo, no se tiene consciencia del problema porque al parecer no se ha pensado en otras materias. Lo que lo lleva a uno a dedicar menos tiempo en el proyecto. Ya que también se tiene que dedicar mucho tiempo a la parte de investigación, porque en su mayoría, todo lo del proyecto se ha aprendido fuera del curso.

Capítulo 5. Bitácora

Requerimientos / Diseño: En esta actividad, se analizó el diseño que se iba a utilizar, así mismo se pensó en lo que esto requería (imágenes, código, etc). Se tomaron imágenes y se redimensionaron a lo necesario. TIEMPO TOTAL: 3 horas

Investigación de funciones: En este punto, se buscaron soluciones para dichos problemas de movimiento de imágenes, repetición de movimientos, y funciones necesarias para lograr el funcionamiento del sensor. TIEMPO TOTAL: 5 horas.

Programación: En esta actividad, se pasó a implementar las funciones y las ideas de código que se tenían, durante 3-4 días seguidos se le dedicó más 2.5 horas diarias, corrigiendo errores, buscando soluciones a funciones no compatibles con el programa, etc. TIEMPO TOTAL: 11 horas

Documentación interna: Con forme se iba avanzando con la programación se iba haciendo cierta documentación interna, al finalizar el proyecto. Se sacó tiempo para poner entradas, salidas y restricciones a las funciones. Así como su funcionalidad. TIEMPO TOTAL FUERA DE PROGRAMACION: 1 hora

Pruebas: Muchas pruebas se fueron realizando con forme se programa, por lo tanto, al final del proyecto se tomaron unos minutos para probar todas las situaciones, que el audio funcionara correctamente y tomar las evidencias necesarias para el documento. TIEMPO TOTAL: 30 minutos

Elaboración documento: Luego de finalizar y obtener las evidencias necesarias para la documentación, se procedió a elaborar el documento final de documentación. Dicho documento necesita de tiempo para ser redactado. TIMEPO TOTAL: 1 hora

Capítulo 6. Informe de horas

FUNCIÓN	TOTAL
Requerimientos / Diseño	3 horas
Investigación de funciones	5 horas
Programación	11 horas
Documentación interna	1 hora
Pruebas	30 minutos
Elaboración documento	1 hora
TOTAL	21:30 horas

Capítulo 7. Opinión personal

Al realizar el proyecto, investigué las funciones y me enteré de métodos que posiblemente me funcionarán a futuro. Lastimosamente no se toman en cuenta los demás cursos llevados por los estudiante y la nueva modalidad que se está efectuando, por lo que limita un poco el tiempo y esto hace que el proyecto baje de calidad.