ISW1-AutomaticCar V2.0

La primera implementación del DrivingAssistant fue todo un éxito. Logramos convencer a los clientes de la factibilidad de ponerlo en producción, sin embargo para hacerlo es necesario ampliar la funcionalidad de la siguiente manera:

1) Agregar un sensor de “posición en carril”, que cada vez que se lo lee devuelve tres posibles valores indicando la posición del auto respecto del carril: *#centered, #shiftedLeft, #shiftedRight*
2) Poder indicarle al sistema de manejo si se quiere doblar a la izquierda o derecha usando la luz de giro.
3) Agregar un modo de consumo de nafta que puede ser *Agresivo, Normal o Económico*, que indica el tiempo que se debe tardar en llegar a la velocidad objetivo del modo de manejo cuando se está subiendo la velocidad. Solo se indica cuando se está en modo de manejo Automático o Manual-Asistido.

Las nuevas acciones que se le puede pedir al sistema de manejo son:

- Que mueva el volante para izquierda, derecha o ir derecho (*steerLeft*, *steerRight*, *goStraight*).
- Que prenda el guiño para doblar a izquierda o derecha (*signalTurnLeft*, signalTurnRight) o que lo apague (*turnOffTurnSignal*)
- Qué aceleración debe usar cuando tiene que subir la velocidad (*acelerate*:)

El comportamiento esperado del modo de manejo respecto del sensor de carril es el siguiente:

- Manual
    - Sensor de Carril:
        - Si se está saliendo del carril, tanto para la izquierda como derecha debe emitir **beep** de “saliendo de carril” siempre y cuando no tenga prendido el guiño correspondiente (o sea si está yendo a izquierda y está el guiño de doblar a izquierda, no debería emitir el beep).
        - Si está yendo derecho con el guiño prendido para doblar a cualquier lado por 2 minutos o más, debe emitir **beep** de “saliendo de carril” hasta que se apague el guiño.
- Automático
    - Sensor de Carril:
        - Si se está saliendo hacia izquierda y no tiene prendido el guiño de doblar a izquierda, mover el volante a derecha. Si tiene prendido el guiño de doblar a la izquierda, debe seguir en estado “ir derecho” (going straight)
        - Si se está saliendo a derecha y no tiene prendido el guiño de doblar a derecha, mover el volante a izquierda. Si tiene

prendido el guiño de doblar a la derecha, debe seguir en estado “ir derecho” (going straight)
        - Si está yendo derecho con el guiño prendido para doblar a cualquier lado por más de 1 minuto, debe emitir beep de “saliendo de carril” hasta que se apague el guiño.
- Manual-Asistido
    - Sensor de Carril:
        - Si se está saliendo del carril, debe combinar las acciones de manual y automático
        - Si se está yendo derecho, debe actuar como si estuviese en modo automático

El tiempo esperado para llegar a la velocidad objetivo según el modo de consumo es:

- Agresivo: 10*second
- Normal: 17*second
- Economico: 23*second

Este tiempo debe ser utilizado para calcular la aceleración que se le debe indicar al sistema de manejo por medio del mensaje *#acelerate*:, cuando debe aumentar la velocidad usando el mensaje *#keepSpeedAt*:. Esto sucede sólo cuando hay un objeto en el frente en la zona segura o cuando no hay objeto por delante. Cuando se debe bajar la velocidad y por default, la aceleración del sistema de manejo debe ser de 0*kilometer/(hour^2).

El modo de consumo se le indica al modo de manejo Automático o Manual-Asistido.

La fórmula para conocer la aceleración a aplicar es la de movimiento rectilíneo uniforme: velocidadFinal = velocidadInicial + aceleración * tiempo

Cuando se prende el auto, al estado inicial ya definido se le agrega que está yendo derecho, con el guiño apagado, con una aceleración de 0*kilometer/hour y en caso de estar en modo Automático o Manual-Asistido, el consumo de nafta debe ser Normal.

No se debe tener en cuenta el caso de pasar de un modo de manejo a otro, siempre se empieza con un modo de manejo y se mantiene.

Se debe asumir que los sensores siempre devuelven valores válidos.

Recuerde que este sistema es un sistema de tiempo real que va generando acciones según lo leído de los sensores y el tipo de asistencia elegido, indicado esto por medio del mensaje #tick.

También, debido a que es un sistema crítico, es importantísimo que esté bien testeado, en pocas palabras, que tenga en cuenta todos los posibles casos funcionales probados y que no haya código sin testear.

Los tests provistos deben seguir pasando sin afectar la funcionalidad preexistente. Solo pueden ser modificados si es necesario hacer algo nuevo en la creación de los objetos o para agregar aserciones.

Resolver el ejercicios usando las heurísticas de diseño vistas durante el cuatrimestre y de manera iterativa e incremental por medio de TDD a partir de la solución ya provista.

Los que deban recuperar solo el primer parcial deben implementar la funcionalidad del sensor de carril, sin tener en cuenta el modo de consumo.

Ayudas:

- Si se está yendo a 60*kilometer/hour y se quiere llegar a 100*kilometer/hour en 10*second, se debe aplicar una aceleración de 14400*kilometer/(hour^2).
  En Smalltalk se puede ver esta equivalencia en el siguiente código:

(100*kilometer/hour) = ((60*kilometer/hour)+((14400*kilometer/(hour^2))*(10*second)))

- Para el día y hora actual pueden usar GregorianDateTime. Se pueden crear instancias con:
  July/17/2023 at: 17:00:00.
- GregorianDateTime theEndOfTime devuelve un día y hora que representa el fin de los tiempos, o sea que siempre es mayor que cualquier otro día y hora.
- GregorianDateTime>>#next: devuelve un día y hora adelantado en el tiempo pasado como colaborador. Ejemplo:
  (July/17/2023 at: 17:00:00) next: 10*minute → July/17/2023 at: 17:10:00
- GregorianDateTime>>#distanceTo: devuelve la diferencia entre un día y hora y otro. Ejemplo:
  (July/17/2023 at: 17:00:00) distanceTo: (July/17/2023 at: 17:10:00) → 10*minute
- Los eventos generados por los sensores pueden ser modelados con ReadStream. Por ejemplo, para el sensor de velocidad se puede usar
  *ReadStream on: { 100*kilometer/hour. 110*kilometer/hour}.*
- No hay problema de si el beep está apagado en volver a apagarlo, si el acelerador está conectado en volver a conectarlo.

Entrega:

1. Entregar por mail el fileout de la categoría de clase **ISW1-2023-1C-Recuperatorio** que debe incluir toda la solución (modelo y tests). El archivo de fileout se debe llamar: **ISW1-2023-1C-Recuperatorio.st**
2. Entregar también por mail el archivo que se llama **CuisUniversity-nnnn.user.changes**
3. Probar que el archivo generado en 1) se cargue correctamente en una imagen “limpia” (o sea, sin la solución que crearon) y que todo funcione correctamente. Esto es fundamental para que no haya problemas de que falten clases/métodos/objetos en la entrega.
4. Realizar la entrega enviando mail a la lista de Docentes: ingsoft1-doc@dc.uba.ar con el **Subject: LU nnn-aa - Solución Recuperatorio 1c2023**
5. Subir a sus repos grupales los archivos **CuisUniversity-nnnn.image** y **CuisUniversity-nnnn.changes**. Debe **zipearlos** previamente para reducir su tamaño o podría dejar sin espacio disponible a sus compañeros. **Pueden eliminar las imágenes del 1er parcial para liberar espacio.**
6. Deberán subirlos al main branch de sus respectivos repos (tenga en cuenta hacer pull antes de ser necesario), y al subdirectorio **/Recuperatorio/LUnnn-aa/**

IMPORTANTE:

No retirarse sin tener el ok de los docentes de haber recibido el mail con la resolución.