

Proyecto

X-Kating

Grupo

WASTED HORCHATA

Documento de diseño de toma de decisión: espacio
y métodos

Hito: 1

Fecha entrega: 15-11-2017

Versión: 1

Componentes:

- Luis González Aracil
- Laura Hernández Rielo
- Adrián Francés Lillo
- Pablo López Iborra
- Alexei Jilinskiy

1. Introducción

En X-Kating tendremos NPCs durante las partidas. Serán otros jugadores con las mismas mecánicas de movimiento y acción que el player. Para una mejor experiencia de juego deben comportarse lo más realista y competitiva posible.

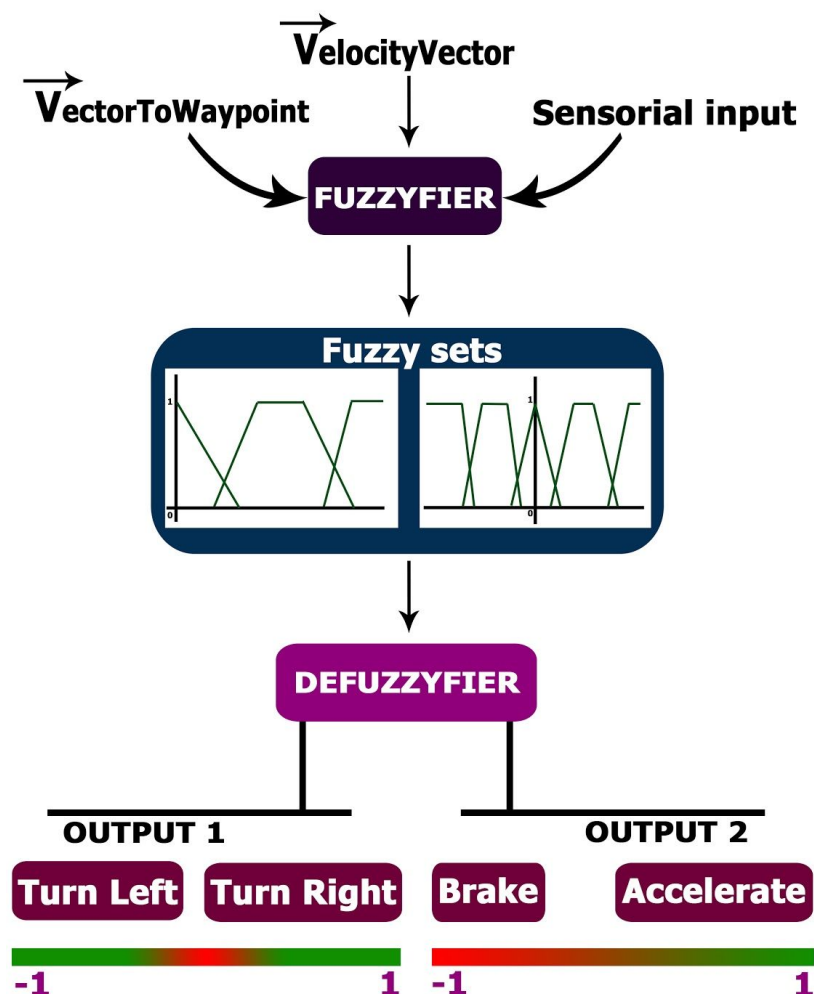
2. Espacio de decisión

a. Lógica difusa

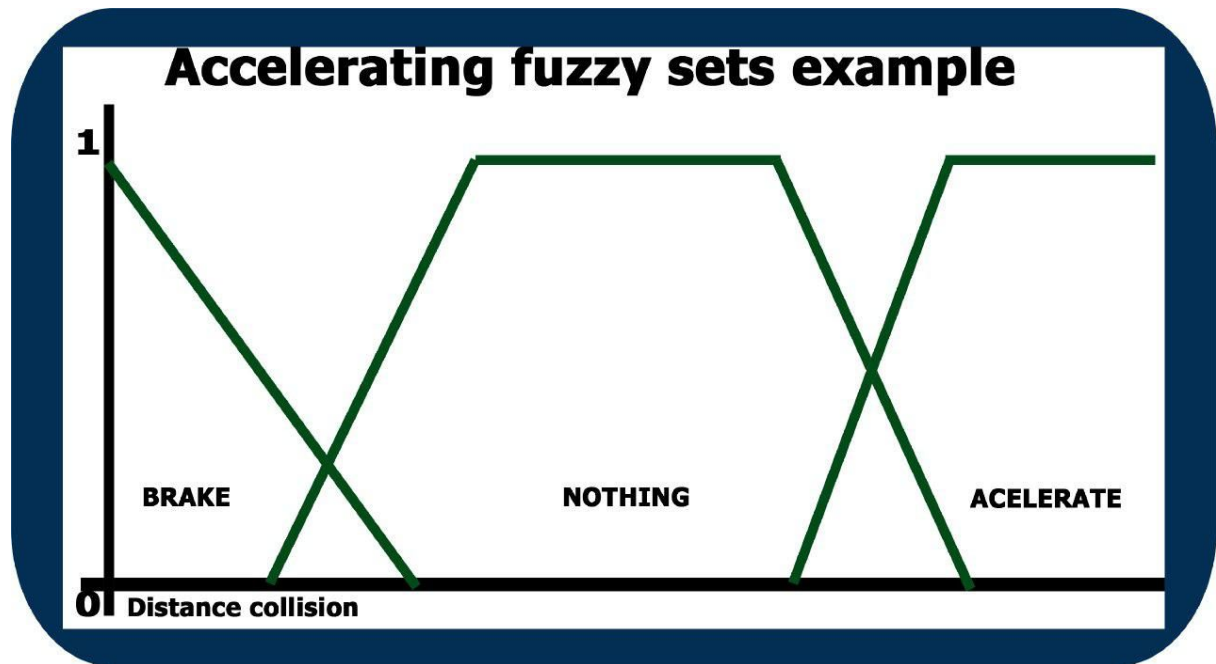
Para mover al NPC, le daremos **inputs al sistema de control** para que se dirija al punto escogido por el sistema de planificación, usando un sistema de lógica difusa.

La **lógica difusa** de los NPCs se **utilizará** para las acciones de **aceleración, frenado y giro**. El sistema decidirá en qué grados ejecutar estas acciones, devolviendo un input con **grado de pertenencia** hacia uno de ellos.

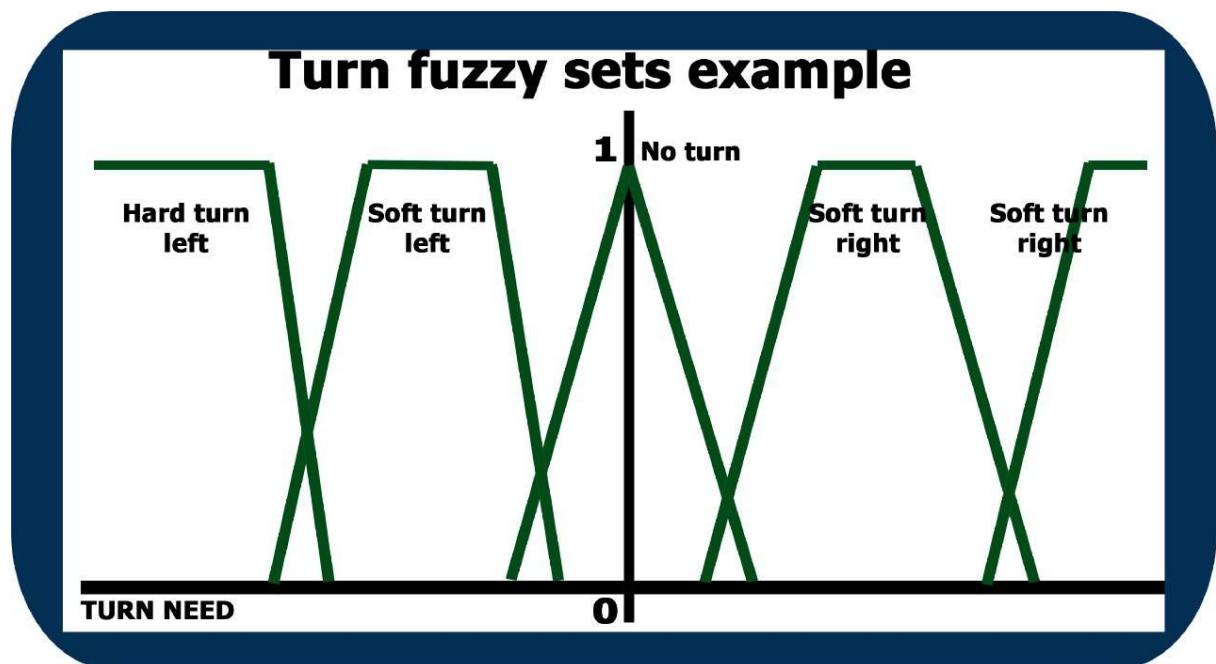
Necesitaremos como entradas del sistema las variables del **vector velocidad**, el **vector del próximo waypoint** y los **inputs sensoriales**. El esquema general del sistema de lógica difusa es el siguiente:



El fuzzificador **tratará las variables para coger los valores** acordes que **usaremos en los sets difusos**. Los sets difusos cogen una variable y determinan su pertenencia a diferentes sets. Inicialmente tendremos **dos sets difusos: set acelerador y set de giro**.



El **set acelerador** se encarga de **decidir si acelerar o frenar**. La primera estrategia que implementaremos será que el **vehículo acelere siempre**, y si se encuentra en rumbo de colisión **frenará según la distancia a la colisión en línea recta**, en primera instancia.



El **set de giro** decide cuánto y dónde gira el NPC. De momento, determinará esta acción **comparando el ángulo** entre el **vector dirección actual** y el **vector** que apunta al

siguiente destino del **waypoint**. Si está en rumbo de **colisión** se **acentuará el giro (Derrape)**.

Después de inferir estas variables en los sets difusos, el defuzzificador transformará los datos en un output de giro, freno y aceleración, y en qué grado.

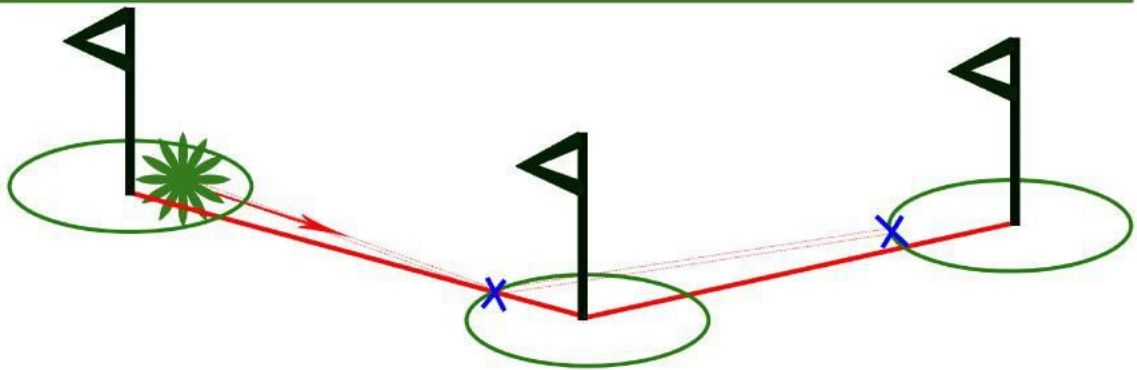
b. WayPoints

El sistema de **WayPoints** de nuestro proyecto se desarrollará **en varias iteraciones**.

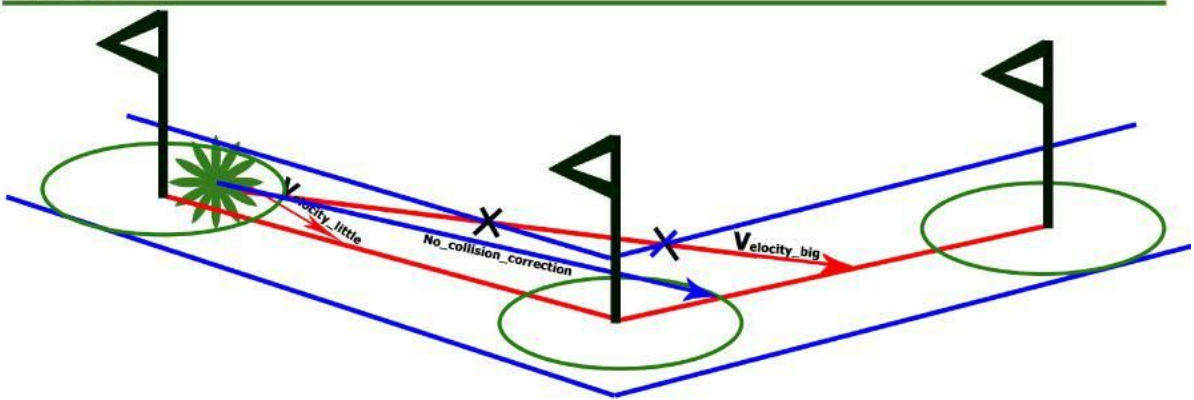
1. **Fase inicial: Incluir** un WayPoint en el mapa con **puntos ordenados y conectados** entre sí, y **con un rango** de detección. Probamos que el NPC va de un punto al siguiente, **usando lógica difusa** para acelerar y girar.
2. **Segunda fase: Esquivar objetos u otros jugadores**. El vector del WayPoint ahora apunta a un punto entre la línea interpolada de los próximos puntos/nodos del WayPoint. Apuntará **más lejos** en el WayPoint cuanto **más rápido** vaya. **Si colisiona, recalcula** el vector del WayPoint, y manda la señal al coche de frenar.
3. **Tercera fase: WayPoints paralelos** en el mismo trozo de ruta. El NPC puede elegir trazados. Se tendrá en cuenta **el derrape** para el giro en curvas (**interpolación circular del movimiento**). Ahora los **WayPoints** no tienen que ser líneas rectas, pueden ser **líneas curvas**.

Esto se detalla en el esquema siguiente:

Fase 1



Fase 2



Fase 3

